

Шановні одинадцятикласники!

Уже протягом кількох років ви вивчаєте інформатику. За цей час ви ознайомилися зі значною кількістю прикладних програм, навчилися застосовувати різноманітні інформаційно-комунікаційні технології в своїй діяльності. В 11-му класі ви продовжите вивчення основ інформатики. Опануєте роботу з програмою створення комп'ютерних публікацій **Microsoft Publisher** і системою управління базами даних **Microsoft Access 2007**, навчитесь опрацьовувати мультимедійні дані та створювати відеокліпи, ознайомитеся з сучасними Інтернет-технологіями **Веб 2.0**. Найбільше уваги ви приділите вивченню важливих складових інформатики – алгоритмізації та програмуванню.

Підручник складається з 4 розділів, кожен з яких розділено на кілька пунктів. Вивчення кожного розділу починається з короткого вступу, в якому зазначено, про що ви дізнаєтесь під час його вивчення. Усі пункти побудовані за єдиною структурою. На початку пункту наведено кілька запитань, відповіді на які допоможуть вам пригадати вивчене раніше для кращого сприйняття нового матеріалу. У пунктах міститься теоретичний матеріал з кожної теми й опис технології опрацювання даних з використанням відповідних програмних засобів, наведено алгоритми та приклади здійснення основних способів діяльності. У кінці кожного пункту розміщено запитання для самоперевірки отриманих вами знань і перелік практичних завдань для відпрацювання навичок роботи.

Для кожного запитання й завдання визначено його рівень складності, який позначається так: ° – початковий і середній рівні; • – достатній рівень; * – високий рівень. Завдання підібрано таким чином, щоб послідовно та цілеспрямовано формувати у вас досконалі навички роботи з програмним забезпеченням, вміння самостійно розв'язувати навчальні задачі з використанням ІКТ, розвивати ваше мислення та навички індивідуальної та колективної роботи. Також у підручнику розміщено завдання 20 обов'язкових практичних робіт, результат виконання яких демонструватиме рівень засвоєння вами навчального матеріалу з певного питання.

Для полегшення сприйняття вами навчального матеріалу автори унаочнили підручник схемами, таблицями, екранними копіями, до підручника включено приклади для демонстрації основних положень інформатики. Додаткові рубрики підручника *«Для тих, хто хоче знати більше»*, *«Це цікаво знати»*, *«Цікаві факти з історії»*, *«Додаткові джерела інформації»* допоможуть вам краще зрозуміти і глибше засвоїти основи науки. Це дає змогу використовувати підручник і для самостійного опанування матеріалу.

Автори пропонують он-лайн підтримку вивчення курсу інформатики 11-го класу на сайті www.allinf.at.ua, де вчителі та учні зможуть знайти файли-заготовки для виконання практичних завдань, додаткові матеріали до окремих тем, поспілкуватися з авторами на форумі, залишити свої відгуки та пропозиції в гостьовій книзі.

Бажаємо вам успіхів у навчанні та засвоєнні сучасних інформаційних технологій!

З повагою, автори

Розділ 1

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У НАВЧАННІ

У цьому розділі ви дізнаєтеся про:

- програмні засоби навчального призначення та особливості їх використання;
- можливості педагогічних програмних засобів для навчання предметів природничо-математичного циклу;
- математичний процесор і його використання для автоматизації математичних обчислень;
- побудову графіків функцій, розв'язування рівнянь і систем рівнянь засобами математичного процесора.

1.1. Програмні засоби навчального призначення та особливості їх використання



1. Які програмні засоби навчального призначення вам відомі? Для чого вони призначені?
2. Наведіть приклади мультимедійних курсів, з якими вам довелося працювати на уроках з різних навчальних предметів.
3. Які електронні засоби навчання ви використовували на уроках природничо-математичного циклу?

Програмні засоби навчального призначення

Коли ви виконуете домашнє завдання, добираєте матеріал для написання реферату, готуетесь до контрольної роботи, вам потрібні різноманітні джерела інформації: підручники та збірники завдань, атласи, словники, хрестоматії, колекції фотографій тощо. Для виконання навчальних завдань ви використовуєте й різноманітні електронні засоби. Це можуть бути ресурси Інтернету, електронні енциклопедії, галереї зображень, а також прикладні програми загального призначення, які використовують для проведення обчислень (наприклад, **Калькулятор**), для підготовки рефератів (наприклад, текстовий процесор **Word 2007**), для створення та редагування малюнків (наприклад, графічний редактор **Paint**) та ін.

Разом з тим існують програми, які безпосередньо призначені для забезпечення навчання. Такі програми називають **програмними засобами навчального призначення**, або **педагогічними програмними засобами (ППЗ)**. Прикладами таких програм є **Програмно-методичний комплекс «Таблиця Менделєєва»**, **Електронний посібник «Біологія 8–9»**, **Програмне середовище «Система лінійних рівнянь»** та інші, які ви могли використовувати на уроках і під час підготовки до навчальних занять у попередніх класах (рис. 1.1).

Перевагою програмних засобів навчального призначення порівняно з традиційними засобами навчання є наявність зручних у використанні засобів візуалізації навчального матеріалу: статичне та динамічне представлення об'єктів, процесів, явищ, їх складових, наочне представлення закономірностей і результатів проведених експериментів, дослідів, знайдених розв'язків задач. Однією з переваг ППЗ є швидкий зворотний зв'язок між користувачем і засобами ІКТ, який забезпечує реалізацію діалогу між учнем і програмним навчальним середовищем. Такий зворотний зв'язок називають **інтерактивністю** програмного засобу. Завдяки використанню ППЗ, навчання може здійснюватись у тому темпі, який найбільше задовольняє учнів.



Рис. 1.1. Використання електронного атласу з географії у навчальному кабінеті, обладнаному мультимедійним проектором

Педагогічні програмні засоби можна розподілити на такі групи:

- **електронні посібники** – електронні навчальні видання, які доповнюють підручники та містять навчальний матеріал з певного предмета, окремих розділів навчальної дисципліни, факультативного курсу або курсу за вибором, найчастіше представлений з використанням мультимедійних засобів; наприклад, Педагогічний програмний засіб «Фізика 10–11» (рис. 1.2, а), Педагогічний програмний засіб «Алгебра, 11 клас»;
- **електронні (віртуальні) практикуми** – електронні навчальні збірки практичних завдань і вправ, у тому числі:
 - **віртуальні лабораторії**, наприклад Віртуальна хімічна лабораторія. 8–11 кл. (рис. 1.2, б), Програмно-методичний комплекс «DG – динамічна геометрія»;
 - **електронні тренажери**, наприклад «Майстер-клас». Клавіатурний тренажер з української мови;
 - **електронні задачники**, наприклад Електронний задачник «Фізика. 7–9», Програмне середовище «Система лінійних рівнянь»;
- **електронні засоби контролю навчальних досягнень учнів** – комп'ютерні програми, призначені для створення тестових завдань, проведення тестування та фіксації результатів; наприклад Система інтерактивного тестування «Школяр»;



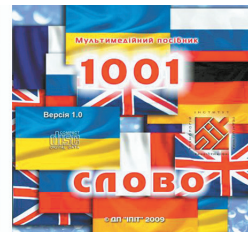
а



б



в



г

Рис. 1.2. Приклади педагогічних програмних засобів

- **мультимедійні засоби ілюстративного і довідкового призначення:**
 - **електронні атласи** – електронні колекції зображень різних об’єктів (карти, креслення, малюнки та ін.) із засобами навігації та пошуку; наприклад **Електронний атлас «Економічна і соціальна географія світу, 10–11 класи»** (рис. 1.2, в);
 - **електронні хрестоматії** – електронні навчальні видання літературно-художніх, історичних та інших друкованих творів, музичних творів, творів образотворчого чи кіномистецтва або їх фрагментів; наприклад **фонохрестоматія «Шкільна колекція». Фонохрестоматія для 10–11 класів з англійської мови**;
 - **електронні енциклопедії** – електронні довідкові видання, що містять основні відомості з однієї чи кількох галузей знань і практичної діяльності, подані у коротких статтях, доповнені аудіо- та відеоматеріалами, засобами пошуку та відбору довідкових матеріалів; наприклад **Електронна база знань «Людина, суспільство і світ», Електронна енциклопедія «Міфи народів світу»**;
 - **електронні словники** – електронні видання словників державної або іноземних мов, що містять засоби пошуку слів та словосполучень і доповнені можливістю прослуховування фрагментів словника; наприклад **Навчальне середовище «1001 слово»** (рис. 1.2, з)

та ін.

Існують комбіновані ППЗ, в яких поєднано програмні засоби з різних груп, наприклад **Інтегрований електронний комплекс «Економічна та соціальна географія світу»**. Вони отримали узагальнену назву **мультимедійні, або інтерактивні, курси**.

Найчастіше педагогічний програмний засіб має такі складові:

- **меню програми**, яке відображає зміст матеріалу в програмному засобі або надає доступ до його основних функцій;
- **гіпертекстова система навігації** між блоками навчального або довідкового матеріалу;
- **пошукова система** для швидкого переходу до потрібного блока навчального матеріалу;
- **довідка** з навчального матеріалу або щодо роботи з програмою.

Залежно від типу педагогічного програмного засобу до його складу можуть входити:

- текстовий інформаційний блок;
- колекція графічних зображень;
- колекція аудіо- та відеоматеріалів;
- блок виконання тренувальних вправ і практичних завдань;
- контролюючий блок

та ін.

Деякі ППЗ розроблені за клієнт-серверною технологією. Передбачається, що на комп’ютері вчителя встановлюється серверна частина таких засобів, на комп’ютерах учнів – клієнтська.

Вибір для роботи того чи іншого педагогічного програмного засобу залежить від навчального завдання, яке стоїть перед учнем. Наприклад, готуючи повідомлення на урок географії, ви зможете шукати потрібні матеріали в електронному посібнику, хрестоматії, енциклопедії або електронному атласі. Якщо ж вам потрібно підготуватися до контрольної роботи з хімії, корисно буде звернутися до віртуальної хімічної лабораторії, збірника задач або засобу контролю навчальних досягнень.

ППЗ можуть містити не тільки програмний матеріал з предмета, а й додатковий матеріал, наприклад теоретичний, що виходить за межі шкільної програми, задачі підвищеної складності, засоби для проведення навчальних досліджень та ін. Це може бути корисним під час підготовки до олімпіад та інших інтелектуальних і творчих змагань, написанні робіт МАН тощо.

Програмні засоби навчального призначення для вивчення різних предметів можна, наприклад, отримати безкоштовно на сайті **Острів знань** (<http://www.ostriv.in.ua>) у розділі *Сервіси* ⇒ *Файловий архів* ⇒ *Електронні засоби навчання*.



Перед тим як використовувати комп'ютерні програми, їх потрібно встановити на комп'ютер (інсталиувати), дотримуючись «Правил використання комп'ютерних програм у навчальних закладах», які затверджені Міністерством освіти і науки України. Усе програмне забезпечення повинно бути сертифіковане, про що має бути зазначено в супроводжувальній документації до програмного засобу.

Установлення (інсталяція) програмного забезпечення проводиться з носія, на якому воно знаходиться, на жорсткий диск комп'ютера. Для цього потрібно виконати таку послідовність дій:

1. Поставити диск з потрібним педагогічним програмним засобом у пристрій для роботи з компакт-дисками.
2. Відкрити вікно із вмістом папок і файлів носія.
3. Запустити файл з іменем **setup.exe** (або **install.exe**).
4. Слідувати подальшим указівкам програми інсталяції.



Перевірте себе

- 1°. Що таке педагогічні програмні засоби? Наведіть приклади навчальних програм.
- 2°. Наведіть приклади програмних засобів для підтримки навчальних предметів природничо-математичних дисциплін.
- 3°. На які види поділяються педагогічні програмні засоби? Охарактеризуйте їх.
- 4°. Які можливості мають сучасні педагогічні програмні засоби?
- 5°. Які складові наявні в педагогічних програмних засобах усіх видів? Яке їх призначення?
- 6°. Опишіть орієнтовну послідовність дій для інсталяції навчальних комп'ютерних програм.
- 7°. Яке призначення електронних посібників та з якою метою вони можуть використовуватися?
- 8°. Які засоби належать до електронних (віртуальних) практикумів? Яке їх призначення та з якою метою вони можуть використовуватись?
- 9°. Які засоби належать до інтерактивних комп'ютерних моделей? Яке їх призначення та з якою метою вони можуть використовуватися?
- 10°. Що може виконувати учень, використовуючи комп'ютерні програми для вивчення фізики, біології, хімії, географії, астрономії?
- 11°. Знайдіть в Інтернеті перелік педагогічних програмних засобів, рекомендованих МОН України до використання в навчальних закладах. Які із цих засобів можна використати для викладання предметів вашого профілю навчання?
- 12*. Які он-лайн ресурси можна використовувати на уроках для предметів вашого профілю навчання? Знайдіть їх в Інтернеті.



Виконайте завдання

1*. Установіть відповідність між групами педагогічних програмних засобів та їхніми описами.

	Педагогічні програмні засоби	Опис
1	Електронні посібники	А Електронні навчальні видання літературно-художніх, історичних та інших друкованих, музичних творів, творів образотворчого чи кіномистецтва або їх фрагментів
2	Електронні (віртуальні) практикуми	Б Електронні видання словників державної або іноземних мов, що містять засоби пошуку мовних одиниць і доповнені можливістю прослуховування фрагментів словника
3	Електронні засоби контролю навчальних досягнень	В Електронні довідкові видання основних відомостей з однієї чи кількох галузей знань і практичної діяльності, поданих у коротких статтях, доповнених аудіо- та відеоматеріалами, засобами пошуку і добору довідкових матеріалів
4	Електронні атласи	Г Електронні навчальні видання, які доповнюють підручники та містять навчальний матеріал з певного предмета, окремих розділів навчальної дисципліни, факультативного курсу або курсу за вибором, найчастіше представлені з використанням мультимедійних засобів
5	Електронні словники	Д Електронні колекції зображень різних об'єктів (карти, креслення, малюнки та ін.) із засобами навігації та пошуку
6	Електронні енциклопедії	Е Електронні навчальні видання практичних завдань і вправ
7	Електронні хрестоматії	Є Комп'ютерні програми, призначені для створення тестових завдань, проведення тестування та фіксації результатів

2*. Укажіть види педагогічних програмних засобів, які ви оберете для виконання таких навчальних завдань:

- а) підготовка до лабораторної роботи з хімії;
- б) написання реферату з математики;
- в) розв'язування задачі з економіки;
- г) написання науково-дослідницької роботи з географії;
- д) підготовка повідомлення про визначні наукові відкриття XX ст.



3*. З'ясуйте, які ППЗ, що призначені для вивчення математики, доступні на сайті **Острів знань** (<http://www.ostriv.in.ua>) у розділі **Сервіси** ⇒ **Файловий архів** ⇒ **Електронні засоби навчання**. Визначте, до яких груп ППЗ вони належать.

4*. Запустіть на виконання електронний посібник з математики, встановлений на вашому комп'ютері. Розгляньте меню програми. Перейдіть до першого розділу посібника. Перегляньте теми першого розділу. З'ясуйте призначення елементів керування та умовних позначок у тексті посібника. Складіть їх опис. Наведіть три приклади використання посібника для розв'язування навчальних завдань.

5*. Відкрийте головну сторінку сайту **Математика для школи** (<http://www.formula.co.ua>). Ознайомтеся зі змістом рубрик **Алгебра** та **Геометрія**. Знайдіть пояснення слова *тригонометрія*. Переведіть 30° у радіани, використовуючи засоби сайту, та обчисліть значення виразу $\sin 30^\circ + \cos 30^\circ$ з використанням вбудованого калькулятора. Поясніть, чим можуть бути корисними для вас матеріали цього сайту.



- 6*. Завантажте з сайту **Острів знань** (<http://www.ostriv.in.ua>) педагогічний програмний засіб «Динамічна геометрія». Проінсталуйте комплекс на своєму комп'ютері. Ознайомтеся з довідкою щодо використання програми. Перегляньте режими роботи програми. До якого типу ППЗ можна віднести цю програму? Наведіть три приклади використання програми для розв'язування навчальних завдань. Підготуйте презентацію про можливості програми.
- 7*. Завантажте з сайту **Острів знань** (<http://www.ostriv.in.ua>) з розділу **Навчання** \Rightarrow **Шкільні предмети** \Rightarrow **Хімія** \Rightarrow **Електронні засоби для вивчення хімії** програмно-методичний комплекс **Таблиця Менделєєва**. Проінсталуйте комплекс на своєму комп'ютері. Ознайомтеся з довідкою щодо використання програми. Перегляньте режими роботи програми. До якого типу ППЗ можна віднести цю програму? Наведіть три приклади використання програми для розв'язування навчальних завдань. Підготуйте презентацію про можливості програми.
- 8*. Відкрийте головну сторінку сайту **Шкільна фізика** (<http://sp.bdpu.org>). Визначте, які матеріали сайту можуть бути вам корисними під час вивчення фізики. Розгляньте в рубриці **Демонстрації** інтерактивну модель руху тіла, кинутого під кутом до горизонту. Поясніть фізичну суть продемонстрованого явища.
- 9*. Завантажте з сайту **Google Earth** (<http://earth.google.com>) безкоштовну версію програми **Google Earth** та запустіть її на своєму комп'ютері. Знайдіть на моделі Землі мапу та фотографії вашої місцевості та історичних місць вашої області.
- 10*. Відкрийте сторінку **Дистанційне навчання. Демонстраційні фрагменти курсів** сайту Сумського державного університету (<http://dl.sumdu.edu.ua/demo>). Ознайомтеся з переліком інтерактивних засобів, розроблених співробітниками лабораторії. Визначте особливості використання програмного засобу з хімії **Окислювально-відновлювальні реакції**. Складіть інструкцію щодо його використання.



Практична робота № 1. Використання програмних засобів під час вивчення фізики, хімії, біології

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

- Запустіть на виконання педагогічний програмний засіб (інтерактивний мультимедійний курс, електронний (віртуальний) практикум, бібліотеку наочностей, електронний атлас тощо) з профільного предмета (фізика, хімія, біологія тощо), встановлений на вашому комп'ютері.
 - Розгляньте меню програми.
 - Ознайомтеся зі змістом програмного засобу.
 - З'ясуйте призначення елементів керування та умовних позначок у тексті посібника.
 - Складіть їх опис у зошиті.
 - Запишіть у зошит три приклади використання курсу для розв'язування навчальних завдань.
- Відкрийте вікі-сайт wiki.ciit.zp.ua.
 - Перейдіть за посиланням **Інтернет-ресурси** в рубриці **Свободная зона**.
 - Ознайомтеся з переліком он-лайн ресурсів для вашого профільного предмета.
 - Оберіть один з них і перейдіть на відповідний сайт.
 - Ознайомтеся з можливостями розміщеного веб-ресурсу.
 - З'ясуйте його призначення, коло користувачів, особливості використання для розв'язування навчальних задач із профільного предмета.
 - Знайдені відомості запишіть у зошит.

1.2. Математичні процесори. Програмно-методичний комплекс GRAN та його використання



1. Які програмні засоби навчального призначення ви використовували на уроках математики? Для чого вони призначені?
2. Які правила введення формул у табличному процесорі **Excel 2007**?
3. Що таке графік функції; графік рівняння?
4. Які дослідження функції ви виконували на уроках математики?
5. Наведіть алгоритм розв'язування рівняння графічним способом.

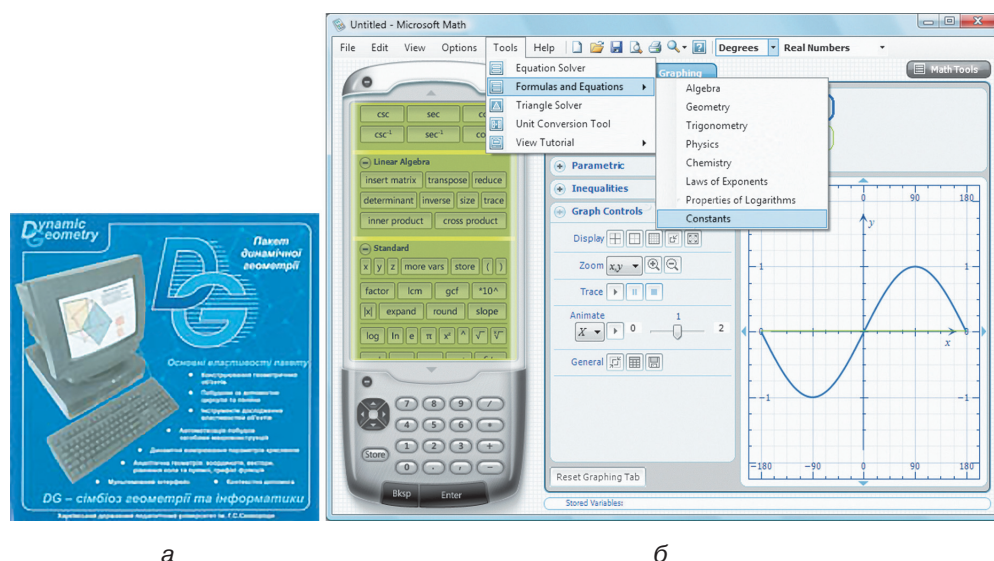
Математичні процесори

Серед програмного забезпечення навчання математики важливе місце займають *математичні процесори*. Основні можливості цих прикладних програм:

- обчислення значень числових виразів;
- побудова графіків функцій, заданих різними способами;
- трасування графіків (побудова таблиці значень функції на основі побудованого графіка);
- знаходження координат точок перетину графіків двох функцій на заданому проміжку;
- знаходження нулів і екстремумів функції на заданому проміжку;
- наближене знаходження розв'язків рівнянь та їхніх систем;
- графічне розв'язування нерівностей та їхніх систем;
- обчислення площ та об'ємів геометричних фігур

та ін.

Прикладами таких програм є **GRAN**, **DG** – Динамічна геометрія (рис. 1.3, а), **Microsoft Mathematics** (рис. 1.3, б), **MathCad**, **EUREKA**, **MathLab**, **Maple**, **DERIVE**, **Advanced Grapher** та ін. Одні з цих програм – це потужні системи комп'ютерної математики з великою кількістю



а

б

Рис. 1.3. Приклади математичних процесорів

функціональних можливостей, які розраховані на фахівців високої кваліфікації, інші мають значно менші можливості та можуть бути використані учнями середньої школи.


Використання математичних процесорів для навчання дає змогу поєднати обчислювальні можливості комп'ютера у процесі дослідження різноманітних функціональних залежностей, звільнившись від рутинних обчислень, з перевагами графічного подання інформації, врахувати індивідуальні можливості та здібності кожного учня щодо темпів навчання. Також це сприяє розвитку графічних навичок та образного мислення, геометричної інтуїції та наочних уявлень з предметної галузі.

Програмно-методичний комплекс GRAN


Одним з перших в Україні педагогічних програмних засобів був програмний комплекс для підтримки навчання математики **GRAN**, розробка якого розпочалася у 1989 році авторським колективом під керівництвом відомого українського вченого Мирослава Івановича Жалдака, академіка АПН України, доктора педагогічних наук, професора (рис. 1.4).

Програмно-методичний комплекс **GRAN**, до складу якого входять педагогічні програмні засоби **GRAN1**, **GRAN-2D**, **GRAN-3D**, забезпечує підтримку вивчення математики (планіметрії, стереометрії, тригонометрії, алгебри і початків аналізу, початків теорії ймовірностей і математичної статистики), а також окремих розділів фізики в школі (7–11 класи).

Розглянемо використання математичних процесорів для комп'ютерної підтримки навчання математики на прикладі програмного засобу **GRAN1**, який призначений для графічного аналізу функцій, що й відображає його назва **G**raphic **A**nalysis.

Запуск програми здійснюється традиційним способом: **Пуск** \Rightarrow **GRAN1** \Rightarrow **GRAN1** або з використанням ярлика . Після цього на екрані відкривається вікно програми (рис. 1.5), яке складається з трьох внутрішніх вікон: **Графік**, **Список об'єктів**, **Відповіді**.

У верхньому рядку вікна програми знаходиться **Рядок заголовка**, під ним – **Рядок меню**. У програмі реалізована *контекстна довідка*. Для її отримання треба вибрати потрібну команду та натиснути клавішу **F1**.

У вікні **Графік** будуються графічні зображення об'єктів і графіків функцій, відображаються пояснювальні написи тощо. Під час переміщення вказівника по координатній площині у вікні переміщується покажчик координат . У верхній частині вікна **Графік**, над робочим полем, відображаються поточні координати вказівника, у нижній частині (**Рядку статусу**) – максимальне та мінімальне значення координат на площині.

Вікно **Список об'єктів** складається з двох частин. У верхній частині знаходиться поле зі списком восьми типів залежностей між змінними, які можна обрати для побудови графічного об'єкта, а нижче – список усіх введених об'єктів, серед яких поточний об'єкт відмічено позначкою \checkmark . У другій частині вікна знаходяться відомості про поточний об'єкт: функція, проміжок, на якому вона розглядається, мінімальне та максимальне



Рис. 1.4.
Жалдак М.І.

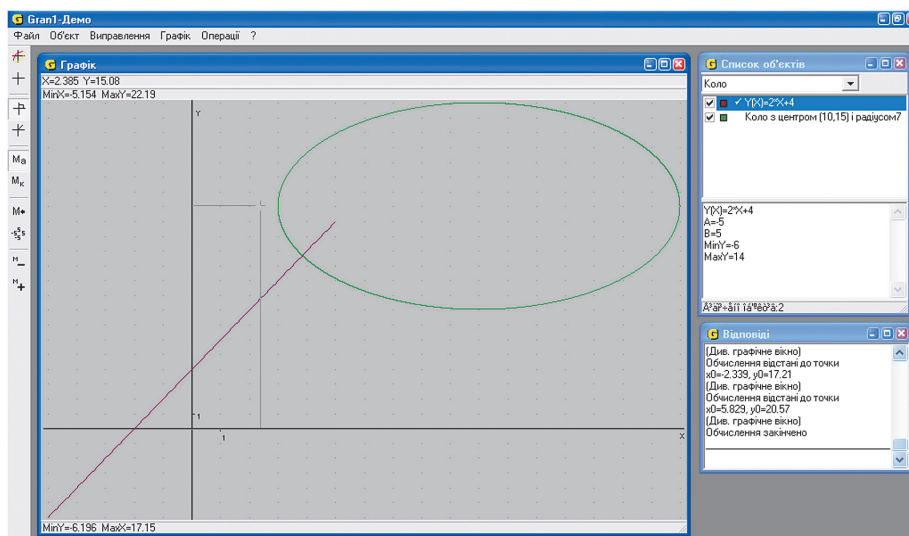


Рис. 1.5. Вікно програми GRAN1

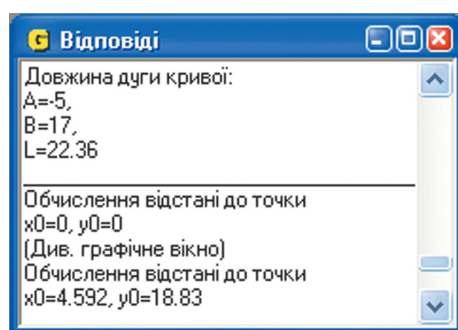


Рис. 1.6. Вікно Відповіді

значення функції або інші параметри виділеного графічного об'єкта.

У вікні **Відповіді** (рис. 1.6) подаються результати виконання різноманітних операцій, які вибираються в меню **Операції** – обчислення відстані від точки, обчислення довжини дуги, визначення довжини ламаної тощо. Це вікно можна очистити, виконавши **Операції** \Rightarrow **Відповіді** \Rightarrow **Очистити**.

Зліва у вікні програми знаходиться панель інструментів, яка містить такі елементи керування (табл. 1.1):

Таблиця 1.1. Команди панелі інструментів GRAN1

Пікто-грама	Команда	Пікто-грама	Команда	Пікто-грама	Команда
	Побудувати графік		Відкрити файл	M_K	Масштаб користувача
$+$	Очистити		Зберегти файл	M_A	Автоматичний масштаб
\oplus	Декартові координати	$f+$	Створити об'єкт	M_*	Попередній масштаб
\oplus	Полярні координати	M_- M_+	Зменшити масштаб Збільшити масштаб	$-S_+ S_-$	Початковий масштаб

Виконання обчислень у GRAN1

Однією з найпоширеніших функцій математичних процесорів є виконання різноманітних математичних обчислень. Програмний засіб **GRAN1** має розширені можливості щодо цього.

Обчислення виконуються у вікні **Калькулятор (Операції ⇒ Калькулятор)**. Введення виразів можна здійснювати або з клавіатури, або використовуючи кнопки вікна **Калькулятор**. Після завершення введення виразу потрібно натиснути **Enter** або вибрати кнопку **Обчислити** у вікні **Калькулятор**. Числові вирази подаються за правилами, близькими до правил табличного процесора **Excel 2007**. Усі допустимі функції та операції подано на кнопках **Калькулятора** (рис. 1.7).

Пропуски у записі виразу не допускаються. Дробова частина у записі чисел відокремлюється від цілої крапкою. Пріоритет операцій загальноприйнятий. Для його змінення використовують дужки. Редагування введеного виразу здійснюється традиційними способами.

У таблиці 1.2 подано умовні позначення арифметичних операцій і функцій.

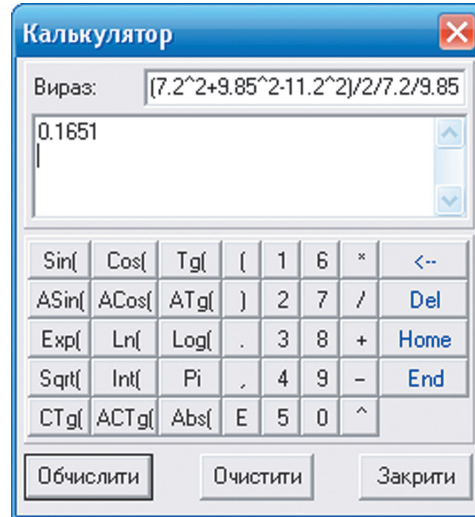


Рис. 1.7. Вікно **Калькулятор**

Таблиця 1.2. Умовні позначення арифметичних операцій і функцій

Позначення	Функція	Позначення	Функція	Позначення	Функція	Позначення	Арифметична операція
Sin	Синус	Cos	Косинус	Tg	Тангенс	*	Множення
ASin	Арксинус	ACos	Арккосинус	ATg	Арктангенс	/	Ділення
Exp	Експонента	Ln	Логарифм натуральний	Log	Логарифм з довільною основою	+	Додавання
Sqrt	Арифметичний квадратний корінь	Int	Ціла частина числа	Pi	Число π	-	Віднімання
CTg	Котангенс	ACTg	Арккотангенс	Abs	Модуль	^	Піднесення до степеня

Розглянемо використання **Калькулятора** на такому прикладі:

Задача 1. Дано сторони трикутника: $AB = 7,2$, $BC = 11,2$, $AC = 9,85$. Обчислити міри його кутів (у радіанах).

Для розв'язування даної задачі потрібно виконати таку послідовність дій:

1. Запустити програму **GRAN1**.
2. Виконати **Операції** \Rightarrow **Калькулятор**.
3. Обчислити значення косинуса кута A за формулою: $\cos A = \frac{AB^2 + AC^2 - BC^2}{2 \cdot AB \cdot AC}$, увівши в поле **Вираз** таку послідовність символів $(7.2^2 + 9.85^2 - 11.2^2) / (2 * 7.2 * 9.85)$ (рис. 1.7).
Результат обчислень: 0.1651.
4. Обчислити міру кута A (у радіанах) за формулою $A = \arccos(\cos(A))$, увівши в поле **Вираз** таку послідовність символів **ACos(0.1651)**.
Результат обчислень: 1.405.
5. Обчислити значення косинуса кута B за формулою: $\cos B = \frac{AC \cdot \sin A}{BC}$, увівши в поле **Вираз** таку послідовність символів $9.85 * \sin(1.405) / 11.2$.
Результат обчислень: 0.8674.
6. Обчислити міру кута B (у радіанах) за формулою: $B = \arcsin(\sin(B))$, увівши в поле **Вираз** таку послідовність символів **ASin(0.8674)**.
Результат обчислень: 1.05.
7. Обчислити міру кута C (у радіанах) за формулою: $C = \pi - A - B$, увівши в поле **Вираз** таку послідовність символів **Pi-1.405-1.05**.
Результат обчислень: 0.687.

Побудова графіків залежностей між змінними у GRAN1 та їх аналіз

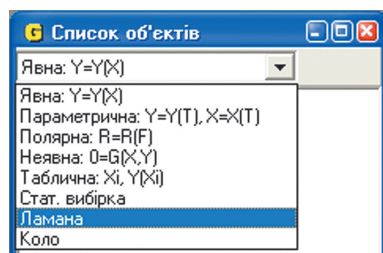


Рис. 1.8. Типи залежностей між змінними

У **GRAN1** можна побудувати графіки восьми основних типів залежностей між змінними (рис. 1.8). Одночасно у вікні можна відобразити до п'яти графіків, усі вони автоматично малюються різними кольорами. Колір ліній кожного об'єкта відображається у вікні **Список об'єктів** біля рівняння функції.

Загальний алгоритм побудови графіка залежності між змінними:

1. Вибрати у вікні **Список об'єктів** тип залежності між змінними.
2. Вибрати у меню **Об'єкт** команду **Створити**.
3. Увести у поле діалогового вікна **Введення виразу залежності** відповідний вираз і вибрати кнопку **ОК**.
4. Вибрати у меню **Графік** команду **Побудувати графік**.

Розглянемо алгоритм побудови графіка функції на такому прикладі.

Задача 2. Побудувати графік функції $y = |x^2 - 8|x| + 7|$.

Для цього потрібно виконати таку послідовність дій:

1. Запустити програму **GRAN1**.
2. Вибрати у вікні **Список об'єктів** тип залежності **Явна: $Y=Y(X)$** .
3. Вибрати у меню **Об'єкт** команду **Створити**.
4. Увести в поле **$Y(X)=$** діалогового вікна **Введення виразу залежності вираз**
 $ABS(X^2-8*ABS(X)+7)$.
5. Вибрати у меню **Графік** команду **Побудувати графік**.

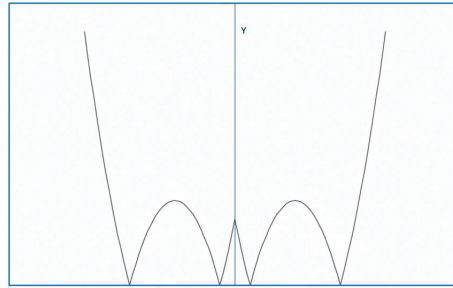


Рис. 1.9. Графік функції
 $y = |x^2 - 8|x| + 7|$

Отриманий результат подано на рисунку 1.9.

Аналізуючи побудований графік функції, можна визначити:

- *нули функції*. Для цього слід установити вказівник у точці перетину графіка функції з віссю Ox і з'ясувати координати цієї точки. Вони будуть відображатися у верхній частині вікна **Графік**;
- *точки екстремумів і екстремуми*. Для цього слід установити вказівник у найвищу чи найнижчу точку графіка і з'ясувати координати вказівника у вікні **Графік**. Максимальне і мінімальне значення функції також відображаються у вікні **Список об'єктів**;
- *проміжки зростання та спадання*. Спочатку потрібно визначити точки екстремумів, а потім записати проміжки спадання та зростання.

Розв'язування рівнянь і систем рівнянь графічним способом

Розглянемо, як можна використати програмний засіб **GRAN1** для графічного розв'язування рівняння з однією змінною. Алгоритм розв'язування даного типу завдань складається з трьох етапів:

1. Побудувати графік залежності.
2. Відмітити на координатній площині точку перетину графіка функції з віссю Ox .
3. Визначити координати вказівника, які відображаються у верхньому рядку вікна **Графік**. Це і буде наближеним коренем рівняння.

Отримані у такий спосіб значення є наближеними. Похибка виникає за рахунок того, що переміщення вказівника на **Робочому полі** має свій крок.

Розглянемо графічне розв'язування рівнянь з однією змінною.

Задача 3. Розв'язати рівняння $x^3 - 2x + 6 = 0$ графічним способом.

Для цього потрібно виконати таку послідовність дій:

1. Запустити програму **GRAN1**.
2. Вибрати у вікні **Список об'єктів** тип залежності **Явна: $Y=Y(X)$** .
3. Вибрати у меню **Об'єкт** команду **Створити**.
4. Увести в поле **$Y(X)=$** діалогового вікна **Введення виразу залежності вираз**
 $X^3-2*X+6$ (рис. 1.10).
5. Вибрати у меню **Графік** команду **Побудувати графік**. На екрані у вікні **Графік** отримаємо графік уведеної функції.
6. Виконати **Графік \Rightarrow Список точок на графіку \Rightarrow Запис**.
7. Установити вказівник послідовно в точки перетину графіка функції з віссю Ox (рис. 1.11).

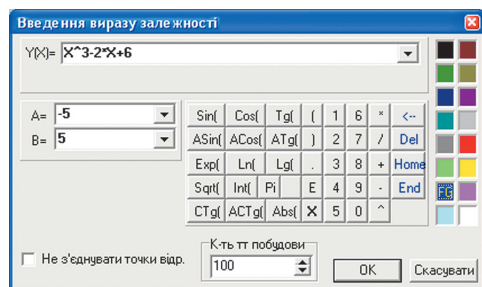
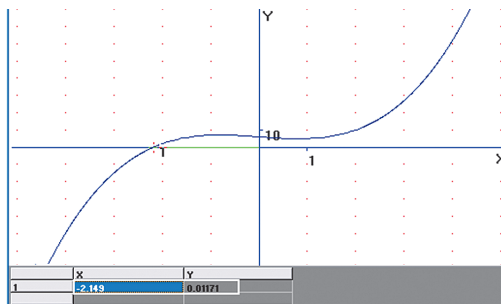
Рис. 1.10. Діалогове вікно **Введення виразу залежності**

Рис. 1.11. Отримання наближеного значення кореня рівняння

8. Визначити наближене значення кореня рівняння, яке відображається у нижній частині вікна **Графік**.

Відповідь: $x = -2.149$.

Якщо точок перетину графіка з віссю Ox буде кілька, то рівняння матиме кілька розв'язків, які й відображатимуться в таблиці значень у вікні **Графік** під координатною площиною.

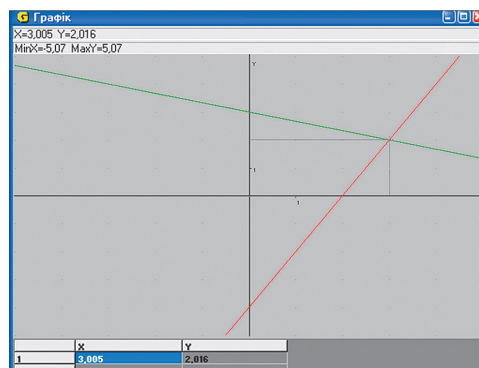
Для системи рівнянь алгоритм знаходження наближеного розв'язку виглядатиме так:

1. Побудувати графіки кожного з рівнянь системи.
2. Відмітити на координатній площині точки перетину побудованих графіків.
3. Визначити координати вказаних точок.

Задача 4. Розв'язати систему рівнянь $\begin{cases} x + 3y = 9, \\ 2x - y = 4 \end{cases}$ графічним способом.

Для цього потрібно:

1. Запустити програму **GRAN1**.
2. Вибрати у вікні **Список об'єктів** тип залежності **Неявна: $0=G(X,Y)$** .
3. Вибрати у меню **Об'єкт** команду **Створити**.
4. Увести в поле діалогового вікна **Введення виразу залежності** вираз $X+3*Y-9$.
5. Вибрати у меню **Графік** команду **Побудувати графік**.
6. Вибрати у вікні **Список об'єктів** тип залежності **Неявна: $0=G(X,Y)$** .
7. Вибрати у меню **Об'єкт** команду **Створити**.
8. Увести в поле діалогового вікна **Введення виразу залежності** вираз $2*X-Y-4$.
9. Вибрати у меню **Графік** команду **Побудувати графік**.
10. Виконати **Графік** \Rightarrow **Список точок на графіку** \Rightarrow **Запис**.
11. Вибрати вказівником точку перетину графіків функцій (рис. 1.12).

Рис. 1.12. Графічний розв'язок системи рівнянь $\begin{cases} x + 3y = 9, \\ 2x - y = 4 \end{cases}$

12. Визначити координати точки перетину у вікні **Графік**.

Відповідь: (3.005; 2.016).



Розв'язування найпростіших задач з планіметрії з використанням GRAN1

Розглянемо використання програмного засобу **GRAN1** для розв'язування геометричної задачі.

Задача 5. Визначити вид чотирикутника $ABCD$, який задано координатами вершин $A(1; 1)$, $B(2; 3)$, $C(0; 4)$, $D(-1; 2)$.

Щоб розв'язати цю задачу з використанням програми **GRAN1**, потрібно виконати таку послідовність дій:

1. Запустити програму **GRAN1**.
2. Вибрати у вікні **Список об'єктів** тип залежності **Ламана**.
3. Вибрати у меню **Об'єкт** команду **Створити**.
4. Увести в діалогове вікно **Координати вершин ламаної** (рис. 1.13) координати вершин чотирикутника і вибрати кнопку **OK**.
5. Вибрати у меню **Графік** команду **Побудувати графік**.
6. Виміряти по черзі довжини сторін і діагоналей, виконавши **Операції** \Rightarrow **Операції з ламаними** \Rightarrow **Довжина ламаної**.

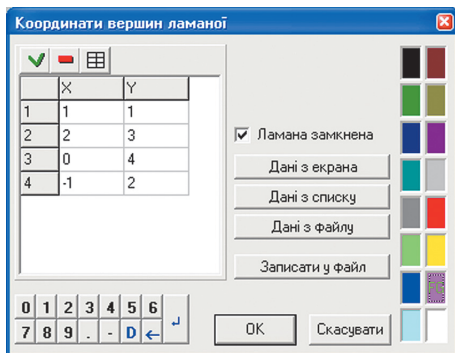


Рис. 1.13. Діалогове вікно **Координати вершин ламаної**

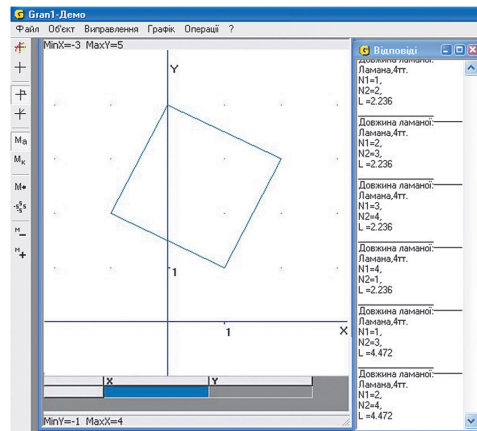


Рис. 1.14. Розв'язування задачі з геометрії з використанням програми **GRAN1**

У вікні **Графік** отримали побудований чотирикутник, а в результаті вимірювання у вікні **Відповіді** (рис. 1.14) отримали такі значення: всі сторони чотирикутника дорівнюють 2.236 лінійні одиниці, діагоналі дорівнюють 4.472 лінійні одиниці. Отже, заданий чотирикутник є квадратом.







Перевірте себе

- 1°. Яке призначення математичних процесорів? Наведіть приклади математичних процесорів.
- 2°. Опишіть інтерфейс вікна програми **GRAN1**.
- 3°. Які правила введення числових виразів у програмі **GRAN1**?
- 4*. Порівняйте правила введення формул у математичному процесорі **GRAN1** і табличному процесорі **Excel 2007**.
- 5°. Які типи залежностей між змінними використовуються в **GRAN1**?
- 6°. Скільки графіків одночасно можна побудувати в **GRAN1**?

- 7°. Як очистити вікно **Графік**?
- 8°. Наведіть алгоритм побудови графіка функції.
- 9°. Наведіть алгоритм побудови графіка рівняння.
- 10°. Які дослідження функції можна здійснити на основі її графіка?
- 11°. Поясніть, як визначити нулі функції. Екстремуми функції?
- 12°. Наведіть загальний алгоритм розв'язування рівняння графічним способом.
- 13°. Наведіть загальний алгоритм розв'язування системи рівнянь графічним способом.
- 14°. Поясніть, чому розв'язання рівняння чи системи рівнянь графічним способом і аналітичним відрізняються.



Виконайте завдання

- 1°. Виконайте обчислення виразів, використовуючи одну з математичних програм, наприклад **GRAN1**:
 - а) $y = \cos 2x$ при $x = 0; 1; 1,57$;
 - б) $y = x^2 - 4|x| + 3$ при $x = -3; 3$;
 - в) $y = 1/(x^2 - 3)$ при $x = -2; 0; 15$.
- 2°. Обчисліть периметр і площу прямокутного трикутника, якщо один з катетів дорівнює 7 см, а протилежний кут 60° , використовуючи одну з математичних програм, наприклад **GRAN1**.
-  3°. Обчисліть периметр і площу трикутника за трьома заданими сторонами ($a = 7$, $b = 5$, $c = 8$), використовуючи одну з математичних програм, наприклад **GRAN1**.
- 4°. Побудуйте графік функції $y = x^4 - 5x^2 + 3$ з використанням одного з математичних процесорів, наприклад **GRAN1**.
- 5°. Побудуйте послідовно графіки функцій $y = |x|$, $y = |x + 1|$, $y = |x| + 1$. Поясніть, як змінюється графік функції та його розташування.
-  6°. Побудуйте послідовно графіки функцій $y = \sin x$, $y = 2\sin x$, $y = \sin 2x$, $y = \sin(x + 2)$. Поясніть, як змінюється графік функції та його розташування при зміні коефіцієнтів.
-  7°. Побудуйте графіки функцій, заданих параметрично, використавши один з математичних процесорів, наприклад **GRAN1**:
 - а) коло: $x = 10\cos T$, $y = 10\sin T$;
 - б) равлик Паскаля: $x = 5\cos^2 T + 3\cos T$, $y = 5\cos T \sin T + 3\sin T$.
- 8°. Побудуйте графік функції $y = |x^3| + x + 7$, використавши один з математичних процесорів, наприклад **GRAN1**. Дослідіть функцію на основі графіка за планом і запишіть відповіді у зошит:
 - а) парність, непарність;
 - б) нулі функції;
 - в) проміжки знакосталості: 1) $y > 0$; 2) $y < 0$;
 - г) точки екстремумів;
 - д) екстремуми;
 - е) проміжки зростання та спадання: 1) зростає; 2) спадає.
- 9°. Розв'яжіть рівняння $x^3 = \sqrt{x}$ графічним способом, використавши один з математичних процесорів, наприклад **GRAN1**.
- 10°. Розв'яжіть систему рівнянь $\begin{cases} x^2 + y^2 = 25, \\ xy = 12 \end{cases}$ графічним способом, використавши один з математичних процесорів, наприклад **GRAN1**.
-  11°. Розв'яжіть задачу, використавши один з математичних процесорів, наприклад **GRAN1**. Дано точки: $A(0; 1)$, $B(1; 0)$, $C(1; 2)$, $D(2; 1)$. Доведіть рівність довжин векторів: а) AB і CD ; б) AC і BD .

Практична робота № 2. Автоматизація математичних обчислень

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Виконайте обчислення значень виразів, використовуючи одну з математичних програм, наприклад **GRAN1**. Запишіть у зошит введені вирази та результати обчислень.

а) $y = 2\sin x$ при $x = 0$;

г) $y = ||x - 1| - |x - 2||$ при $x = -2,15$;

б) $y = \cos 2x$ при $x = 0$;

д) $y = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ при $a = 2, b = 8, c = 3$;

в) $y = (x - 1)^2 / (x - 2)^3$ при $x = -3$; е) $y = \pi r^2$ при $r = -3,9$.

2. Радіус Місяця дорівнює 1740 км. Обчисліть площу його поверхні та об'єм космічного тіла, використовуючи одну з математичних програм, наприклад **GRAN1**. Запишіть у зошит введені вирази та результати обчислень.

Практична робота № 3. Побудова графіків функції та їх аналіз

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Побудуйте графіки наведених функцій, використавши один з математичних процесорів, наприклад **GRAN1**:

а) $y = 2\sin x$;

в) $y = 2x$;

д) $y = ||x - 1| - |x - 2||$;

б) $y = \cos 2x$;

г) $y = \frac{x - 1}{x - 2}$;

е) $y = (1 + x)^x$.

2. Дослідіть функції а, в, д завдання № 1 за їх графіками згідно з планом:

а) парність, непарність;

б) нулі функції;

в) проміжки знакосталості: 1) $y > 0$; 2) $y < 0$;

г) точки екстремумів;

д) екстремуми;

е) проміжки зростання та спадання: 1) зростає; 2) спадає.

Практична робота № 4. Знаходження наближених значень розв'язків рівнянь та їх систем

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Знайдіть наближені розв'язки рівнянь графічним способом, використавши один з математичних процесорів, наприклад **GRAN1**:

а) $y = x^2 - 4|x| + 3$;

б) $y = \frac{1}{x^2 - 3}$;

в) $y = \sqrt{||x| - 2|}$.

2. Знайдіть наближені розв'язки системи рівнянь графічним способом, використавши один з математичних процесорів, наприклад **GRAN1**:

а) $\begin{cases} 3x - 7y = 9, \\ 5x + 13y = 7; \end{cases}$

б) $\begin{cases} 3x^2 - 4y^3 = 12, \\ 2x^3 - 5y^2 = 8; \end{cases}$

в) $\begin{cases} x + y = 1, \\ |xy| = \frac{1}{2}; \end{cases}$

г) $\begin{cases} x + y + 1 = 0, \\ \sin(x + y) = 0. \end{cases}$

Розділ 2

ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

У цьому розділі ви дізнаєтеся про:

- поняття моделі та моделювання, типи моделей;
- створення інформаційних моделей;
- алгоритми та їх властивості, основні алгоритмічні структури;
- етапи розв'язування задач з використанням комп'ютера;
- комп'ютерне моделювання;
- комп'ютерні програми;
- створення та налагодження комп'ютерних програм в інтегрованому середовищі візуальної розробки;
- методи опрацювання даних з використанням комп'ютерних програм.

2.1. Моделі та моделювання. Типи моделей



1. Наведіть приклади об'єктів, їхніх властивостей і значень цих властивостей.
2. Хто такі *модельєри*? Що створюють представники цієї професії?
3. Які моделі об'єктів ви знаєте? Для чого вони створюються? На яких уроках і як використовувалися моделі об'єктів?
4. Які процеси або явища можуть бути описані функцією $y = kx$?

Поняття моделі. Моделювання

Для вивчення властивостей і взаємозв'язків об'єктів (предметів, процесів або явищ) люди проводять різноманітні дослідження. Але не завжди можна або доцільно досліджувати самі предмети, процеси або явища безпосередньо. У таких випадках створюють і досліджують не самі об'єкти, а їхні **моделі**. Термін *модель* походить від латинського слова *modulus* – зразок, аналог.

Моделі створюють для дослідження об'єктів тоді, коли сам об'єкт недоступний і його неможливо дослідити безпосередньо (наприклад, зірка сузір'я Великої Ведмедиці або виверження вулкана), або коли дослідження об'єкта можуть призвести до його руйнування (наприклад, мостовий перехід), або коли його виготовлення потребує значних коштів (наприклад, забудова нового мікрорайону) тощо.

Моделі об'єктів створюють не тільки тоді, коли вони недоступні або дорого коштують, а й тоді, коли потрібно дослідити конкретну властивість або групу властивостей об'єкта. У таких випадках створюють модель об'єкта, яка обов'язково має ті властивості, що досліджуються, а інші властивості, що є несуттєвими для даного дослідження, можуть бути в моделі відсутні.

Моделями є знайомі вам карта України та глобус у кабінеті географії, моделі геометричних тіл у кабінеті математики, моделі органів людини в кабінеті біології, план спортивного майданчика вашої школи, малюнок нового автомобіля, словесний опис незнайомої вам людини.

Люди створюють моделі одягу для вивчення попиту на них, моделі літаків для вивчення їхніх аеродинамічних властивостей, моделі корпусів автомобілів для вивчення їхньої міцності, моделі нових архітектурних ансамблів для вивчення їхньої сумісності з уже існуючими, моделюють хімічний експеримент для вивчення його наслідків тощо. Учені-фізики моделюють рух тіла під дією різних сил, процес ядерної реакції, створили модель так званої ядерної зими для ілюстрації наслідків застосування ядерної зброї, сейсмологи моделюють землетруси, щоб вивчити їхні наслідки для різних регіонів планети, економісти створюють моделі розвитку суспільства.



Множину всіх предметів, властивості яких і відношення між якими досліджуються, називають *предметною областю* цього дослідження.

Так предметною областю дослідження для зоологів є множина тварин, предметною областю для математиків під час дослідження подільності чисел є множина цілих чисел.



***Модель об'єкта* – це новий об'єкт, який відображає властивості об'єкта, суттєві для даного дослідження.**

Процес створення та дослідження об'єктів на основі їхніх моделей називається *моделюванням*.

Зазначимо, що в різних науках досліджують різні властивості об'єктів. І тому для кожного об'єкта можуть існувати різні моделі. Це залежить від того, які саме властивості досліджуються. Так, різними будуть моделі людини у дослідженнях фізика, хіміка, біолога, лікаря, модельєра.

З іншого боку, різні об'єкти можуть мати одну й ту саму модель. Так, прямокутний паралелепіпед може бути моделлю книжки, шафи для одягу, будинку та багатьох інших об'єктів. А функція $y = kx$ може слугувати моделлю прямолінійного руху матеріальної точки з постійною швидкістю, змінення напруги електричної мережі залежно від сили струму при постійному опорі, вартості покупки картоплі залежно від маси покупки та ін.

Можна стверджувати, що будь-яка розумова діяльність людини являє собою оперування образами предметів, процесів і явищ, які є, по суті, їхніми моделями. Дійсно, міркуючи про конкретний об'єкт, людина виділяє з усіх його властивостей лише окремі, які стосуються мети його розумової діяльності про предмети або явища, що досліджуються.

Очевидно, що правильна побудова моделей об'єктів та їхнє дослідження сприяють точності й правильності наукових та інженерних висновків, пропозицій, рішень. У сучасній науці та техніці побудова моделей, а також їхнє дослідження проводяться з використанням комп'ютерів, спеціальних комп'ютерних програм.

Крім того, створюють спеціальні комп'ютерні програми, які реалізують модель об'єкта. Такі програми називають **комп'ютерною моделлю**

об'єкта. У наш час комп'ютерні моделі широко використовуються для дослідження об'єктів, проведення **обчислювальних експериментів** у тих випадках, коли проведення реальних експериментів неможливе, або потребує багато коштів, або має непередбачувані наслідки.

Типи моделей

Моделі класифікують за різними ознаками (рис. 2.1):

- за способом подання;
- за галузями використання;
- за фактором часу

та ін.

За способом подання моделі розподіляють на **матеріальні (предметні)** та **інформаційні**.

Іграшки, опудала тварин, манекени, муляжі, глобус, модель водяного млина – все це приклади **матеріальних моделей**. Матеріальні моделі призначені для проведення **практичних досліджень**.

Фізична карта України, рівняння хімічної реакції, математична функція, розповідь про береги Дніпра – це приклади **інформаційних моделей**. Інформаційні моделі призначені для проведення **теоретичних досліджень**.



Матеріальна модель об'єкта – це модель об'єкта, подана у вигляді матеріального об'єкта (предмета).

Інформаційна модель об'єкта – це модель об'єкта, подана у вигляді його опису.

Як і матеріальні моделі, інформаційні моделі одного й того ж об'єкта будуть різні, залежно від мети дослідження. Так, наприклад, інформаційна модель об'єкта «помідор» для постачальника міститиме дані про розміри, умови зберігання, фактори і терміни дозрівання, максимальні терміни зберігання тощо. А для фермера інформаційна модель цього самого об'єкта міститиме дані про час сіяння, регулярність прополювання і поливання, раціональне використання добрив тощо.

Інформаційні моделі у свою чергу розподіляють на:

- **словесні** (усні та письмові описи);
- **графічні** (рисунок, креслення, піктограми, карти та ін.);
- **структурні** (таблиці, графіки залежностей, діаграми, схеми та ін.);
- **алгоритмічні** (правила, плани дій та ін.);
- **математичні** (формули, рівняння, нерівності, функції та ін.);
- **спеціальні** (хімічні формули і рівняння, нотні записи, записи шахових партій та ін.).

Для створення інформаційної моделі об'єкта потрібно:

1. Визначити об'єкт дослідження, для якого створюється модель, і мету дослідження.
2. Виділити ті властивості об'єкта, які є суттєвими для вказаного дослідження.
3. Установити взаємозв'язки між вибраними властивостями та виразити їх, використавши одну з форм представлення.

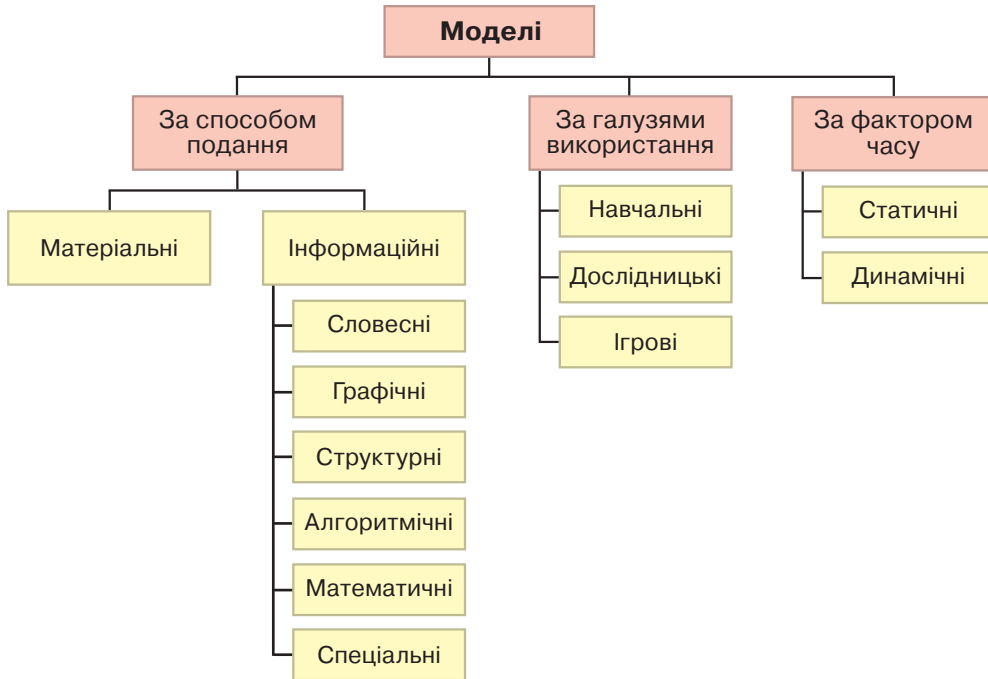


Рис. 2.1. Типи моделей

За галузями використання моделі розподіляють на:

- **навчальні** – створюються для навчання (наочні посібники, тренажери, що створюють для навчання, наприклад тренажер літака для навчання льотчиків, тренажер для імітації перевантажень під час виведення на орбіту в космічному польоті та ін.);
- **дослідницькі** – створюються для проведення досліджень (модель річкового теплохода, гелікоптера, гідроелектростанції для проведення досліджень на міцність, плавучість, аеродинамічні властивості, вплив на екологію, модель комп'ютера, прилад, що імітує розряд блискавки та ін., а також формули, рівняння, функції та ін., які можна використати для дослідження об'єктів);
- **ігрові** – створюються для моделювання ситуацій для розробки і перевірки різноманітних стратегій поведінки, адаптації до певних умов (військові, економічні, спортивні, ділові ігри та ін.).

За фактором часу моделі розподіляють на:

- **статичні** – це моделі, в яких зафіксовано стан об'єкта в певний момент часу і подальші зміни цього об'єкта не враховуються, наприклад гіпсова модель піраміди Хеопса;
- **динамічні** – це моделі, які призначені для дослідження змінення значень властивостей об'єкта з часом, наприклад модель змінення рівня води в гірських річках під час танення снігів; модель погоди в даному регіоні; модель Сонячної системи, що змінюється залежно від результатів нових досліджень.



Перевірте себе

- 1°. Що таке предметна область дослідження? Наведіть приклади.
- 2°. Що таке модель? Наведіть приклади моделей.
- 3°. Для чого створюються моделі? Наведіть приклади.
- 4°. Поясніть, у чому полягає процес моделювання.
- 5°. Чим модель об'єкта відрізняється від реального об'єкта?
- 6°. Поясніть, чому один і той самий об'єкт може мати різні моделі. Наведіть приклади.
- 7°. Поясніть, чому один і той самий об'єкт може бути моделлю різних об'єктів. Наведіть приклади.
- 8°. Як класифікують моделі за способом представлення? Охарактеризуйте їх.
- 9°. Які моделі називають матеріальними? Наведіть приклади.
- 10°. Які моделі називають інформаційними? Наведіть приклади.
- 11°. Як класифікують інформаційні моделі? Опишіть різні види інформаційних моделей.
- 11°. Як класифікують моделі за галузями використання? Наведіть приклади.
- 12°. Як класифікують моделі за фактором часу? Наведіть приклади.
- 13°. Що таке комп'ютерне моделювання?



Виконайте завдання

- 1°. До якого типу моделей (матеріальна чи інформаційна) належать такі моделі:

а) рисунок піраміди;	и) модель атома;
б) накреслений план школи;	і) запис шахової партії;
в) формула площі прямокутника;	ї) проведення фізичного досліду;
г) нотний запис пісні;	й) рівняння хімічної реакції;
д) макет забудови ділянки;	к) опудало птаха;
е) кулінарний рецепт;	л) сценарій проведення свята;
є) хімічний дослід;	м) фотографія місцевості;
ж) модель літака;	н) глобус?
з) комп'ютерна модель автомобіля;	
- 2°. До якого виду належать наведені нижче інформаційні моделі:
 - а) графік чергування;
 - б) сценарій п'єси;
 - в) формула закону Ома;
 - г) ноти нової пісні;
 - д) реклама по телебаченню;
 - е) словесне формулювання теореми Піфагора;
 - є) рівняння хімічної реакції;
 - ж) таблиця результатів чемпіонату України з баскетболу;
 - з) автобіографія?
- 3°. Створіть інформаційну модель об'єктів «квітка» і «кавун» з точки зору художника, медика, ботаніка, постачальника, покупця.
- 4°. Створіть інформаційну модель об'єкта «дачний будинок» з точки зору художника, будівельника, покупця.
- 5°. Відстань між містами А і В дорівнює 50 км. З цих міст одночасно назустріч один одному виїхали два велосипедисти зі швидкостями 15 км/год і 12 км/год відповідно. Створіть математичну модель для визначення відстані між ними у будь-який момент часу до їхньої зустрічі.
- 6°. Один оператор може набрати рукопис за 20 год, а другий – за 30 год. Створіть математичну модель для визначення частини рукопису, яку вони наберуть через x год, якщо працюватимуть одночасно.

- 7*. Назвіть процеси, математичною моделлю яких є лінійна функція; квадратична функція; обернена пропорційність.

2.2. Алгоритми та їх властивості. Форми подання алгоритмів



1. Які речення називаються спонукальними?
2. Чи готували ви якусь страву, користуючись рецептом її приготування? Як ви це робили?
3. На яких уроках ви користувалися інструкціями? Якими саме?
4. Наведіть приклади правил української мови, в яких використана конструкція *якщо ... , то ...*.

Поняття алгоритму

Люди щоденно користуються різноманітними правилами, інструкціями, рецептами тощо, що складаються з певної послідовності **команд (вказівок)**. Деякі з них настільки увійшли до нашого життя, що ми виконуємо їх майже не замислюючись, іноді кажуть, автоматично.

Так для приготування яєчнї потрібно виконати послідовність команд:

1. Поставити сковороду на плиту.
2. Покласти на сковороду шматочок вершкового масла.
3. Увімкнути конфорку.
4. Чекати, поки масло на сковороді розтане.
5. Розбити два яйця і вилити їх вміст на сковороду.
6. Посолити.
7. Чекати, поки загусне білок.
8. Вимкнути конфорку.

А для того щоб визначити вид трикутника за його кутами, якщо відомі його три сторони, потрібно виконати таку послідовність команд:

1. Визначити сторону трикутника, яка не менша за кожну з двох інших.
2. Обчислити косинус кута трикутника, що лежить проти сторони, визначеної як результат виконання команди 1.
3. Якщо визначений косинус кута від'ємний, то повідомити, що даний трикутник тупокутний, якщо ні, то якщо визначений косинус кута дорівнює нулю, то повідомити, що даний трикутник прямокутний, якщо ні, то повідомити, що даний трикутник гострокутний.

Такі послідовності команд (вказівок) називають **алгоритмами**.



Алгоритм – це скінченна послідовність команд (вказівок), що визначає, які дії та в якому порядку потрібно виконати, щоб досягти поставленої мети.

Кожна команда алгоритму є спонукальним реченням, що вказує, яку дію має виконати виконавець алгоритму. **Виконавцем алгоритму** може бути людина, тварина, автоматичний пристрій, такий як робот, верстат з програмним керуванням, електронна обчислювальна машина тощо.

Множину всіх команд, які може виконувати даний виконавець, називають **системою команд цього виконавця**. Наприклад, у систему команд виконавця, що виконуватиме другий з наведених вище алгоритмів, повинні входити такі команди:

1. Порівняти довжини сторін трикутника і вибрати з них не меншу.
2. Обчислити косинус кута трикутника за відомими трьома сторонами.
3. Порівняти число з нулем (більше за нуль, менше за нуль або дорівнює нулю).
4. Вивести повідомлення.

Звертаємо вашу увагу: розробляючи алгоритм, потрібно перш за все визначити, для якого виконавця він призначений, і використовувати в алгоритмі тільки ті команди, які входять до системи команд цього виконавця.



Слово **алгоритм** походить від імені видатного вченого середньовічного Сходу **Мухаммеда бен-Муси аль-Хорезмі** (783–850) (рис. 2.2), який у своїх наукових працях з математики, астрономії та географії описав і використовував індійську позиційну систему числення, а також сформулював у загальному вигляді правила виконання чотирьох основних арифметичних дій: додавання, віднімання, множення та ділення. Європейські вчені ознайомилися з його працями завдяки перекладам їх на латину. Під час перекладу ім'я автора було подано як **Algorithmus**. Звідси й пішло слово **алгоритм**. А розроблені ним правила виконання арифметичних дій вважають першими алгоритмами.



Рис. 2.2. Мухаммед бен-Муса аль-Хорезмі

Властивості алгоритму

Властивостями алгоритму є **дискретність, визначеність, виконуваність, скінченність, результативність і масовість**.

Дискретність (лат. *discretus* – розділений, розривний) алгоритму означає, що його виконання зводиться до виконання окремих дій (кроків) у певній послідовності. Причому, кожна команда алгоритму повинна виконуватися за скінченний інтервал часу.

Визначеність (або **детермінованість** (лат. *determinans* – визначальний)) алгоритму означає, що для заданого набору значень початкових (вхідних) даних алгоритм однозначно визначає порядок дій виконавця та результат цих дій. Алгоритм не повинен містити команди, які можуть сприйматися виконавцем неоднозначно, наприклад «Узяти 2–3 ложки цукру», «Трохи підігріти молоко», «Вимкнути світло через кілька хвилин», «Поділити число x на одне з двох даних чисел a або b » тощо. Крім того, в алгоритмах недопустимі ситуації, коли після виконання чергової команди виконавцю незрозуміло, яку команду він повинен виконувати наступною.

Виконуваність алгоритму означає, що алгоритм, призначений для певного виконавця, може містити тільки команди, які входять до системи команд цього виконавця. Так, наприклад, алгоритм для виконавця «Учень першого класу» не може містити команду «Побудуй бісектрису даного кута», хоча така команда може бути в алгоритмі, який призначений для виконавця «Учень восьмого класу».

Зазначимо, що виконавець повинен лише **вміти виконувати** кожну команду зі своєї системи команд, і не важливо, розуміє він її чи ні. Говорять, що виконання алгоритмів виконавцем носить **формальний характер**: виконавець може не розуміти жодну з команд, може не знати мети виконання алгоритму, і все одно отримає результат. Так, наприклад, верстат з програмним керуванням не розуміє жодної з команд, які він виконує, але завдяки своїй конструкції успішно виготовляє деталі.

Скінченність алгоритму означає, що його виконання закінчиться після скінченної (можливо, досить великої) кількості кроків і за скінченний час при будь-яких допустимих значеннях початкових даних. Наведені вище послідовності команд є скінченними, а наступна послідовність команд – нескінченна:

1. Взяти число 2.
2. Помножити взяте число на 10.
3. Додати до одержаного числа 5.
4. Якщо одержане число додатне, то виконати команду 3, якщо ні, то припинити виконання алгоритму.

Результативність алгоритму означає, що після закінчення його виконання обов'язково одержуються результати, які відповідають поставленій меті. Результативними вважаються також алгоритми, які визначають, що дану задачу не можна розв'язати або дана задача не має розв'язків при заданому наборі початкових даних.

Масовість алгоритму означає, що алгоритм може бути застосований до цілого класу однотипних задач, для яких спільними є умова та хід розв'язування та які відрізняються тільки початковими даними.

Таким, наприклад, є алгоритм розв'язування квадратного рівняння, який дає змогу знайти дійсні корені квадратного рівняння з довільними дійсними коефіцієнтами або визначити, що при певних значеннях коефіцієнтів рівняння не має дійсних коренів. Масовим також є, наприклад, алгоритм побудови бісектриси довільного кута з використанням циркуля та лінійки.

Однак, крім масових алгоритмів, складаються й застосовуються алгоритми, які не є масовими. Таким, наприклад, є алгоритм розв'язування конкретного квадратного рівняння (наприклад, $2x^2 + 5x + 2 = 0$) або алгоритм приготування конкретного салату (наприклад, грецького) на конкретну кількість осіб.

Форми подання алгоритмів

Розглянемо алгоритми розв'язування такої задачі.

Задача 1. Є посудина місткістю 8 л, яка заповнена рідиною, і дві порожні посудини місткістю 5 л і 3 л. Потрібно одержати в одній з посудин 1 літр рідини і повідомити в якій.

Розглянемо виконавця, який має таку систему команд:

1. Перелити рідину з однієї посудини в іншу.
2. Наповнити одну з посудин рідиною з іншої посудини.
3. Вивести повідомлення.

Для цього виконавця алгоритм розв'язування цієї задачі буде таким:

1. Наповнити 3-літрову посудину з 8-літрової.
2. Перелити з 3-літрової посудини в 5-літрову.
3. Наповнити 3-літрову посудину з 8-літрової.
4. Наповнити 5-літрову посудину з 3-літрової.
5. Вивести повідомлення: «1 л одержано в 3-літровій посудині».


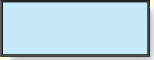
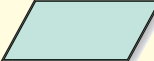
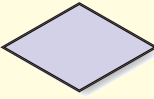
Наведений алгоритм, як і розглянуті раніше, записаний у вигляді послідовності команд, кожна з яких має свій порядковий номер і записа-

на українською мовою, тобто мовою людського спілкування. Така форма подання алгоритму називається **словесною**.

Але словесна форма подання алгоритму не завжди є зручною і наочною. Тому, крім такої форми подання, часто використовують **графічну форму подання алгоритму**, або подання алгоритму у вигляді **блок-схеми**.

У блок-схемі алгоритму кожна команда записується в геометричній фігурі (блоці) певного вигляду. Блоки з'єднуються між собою стрілками, які вказують напрям переходу для виконання наступної команди алгоритму. Наведемо деякі елементи (блоки) блок-схеми алгоритму (табл. 2.1).

Таблиця 2.1. Деякі елементи (блоки) блок-схеми алгоритму

Найменування	Позначення	Призначення
Термінатор		Початок або кінець алгоритму
Процес		Виконання однієї або кількох команд
Дані		Введення вхідних даних (аргументів) або виведення вихідних даних (результатів)
Рішення		Прийняття рішення залежно від результату перевірки вказаної умови

Блок-схема алгоритму розв'язування задачі 1 представлена на рисунку 2.3.



Характерними рисами цього алгоритму є те, що всі його команди виконуються в записаній послідовності, кожна команда алгоритму обов'язково виконується, причому тільки один раз. Такі алгоритми (або фрагменти алгоритму) називаються **лінійними.**

Розглянемо ще одну задачу – задачу на обчислення значення виразу, і складемо алгоритм її розв'язування.

Задача 2. Обчислити значення виразу $(a - b) * (c - d)$, де a, b, c, d – дійсні числа.

Розглянемо виконавця, який вміє отримувати значення змінних, виконувати арифметичні операції над дійсними числами, запам'ятовувати результати арифметичних операцій і повідомляти результат. Передавання даних виконавцеві називається **введенням даних**, а повідомлення виконавцем результатів виконання алгоритму – **виведенням даних (результатів)**.

Запишемо алгоритм розв'язування задачі 2 для такого виконавця в словесній формі.

1. Увести значення змінних a, b, c, d . (У результаті виконання цієї команди виконавець запам'ятовує введені дані як значення відповідних змінних.)
2. Обчислити значення виразу $a - b$ і результат присвоїти змінній x (запам'ятати як значення цієї змінної).
3. Обчислити значення виразу $c - d$ і результат присвоїти змінній y (запам'ятати як значення цієї змінної).

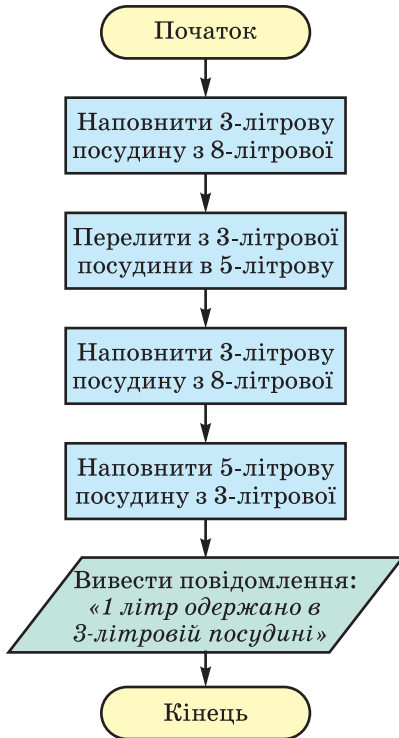


Рис. 2.3. Блок-схема алгоритму розв'язування задачі 1

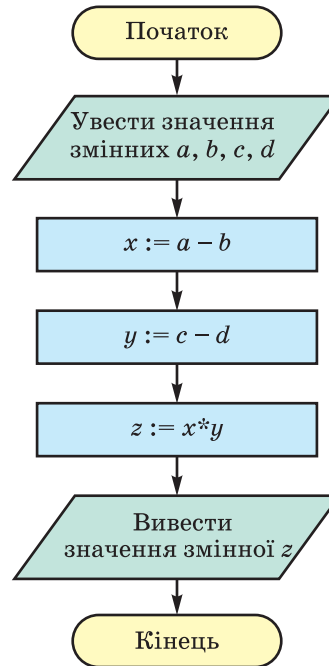


Рис. 2.4. Блок-схема алгоритму розв'язування задачі 2

4. Обчислити значення виразу $x*y$ і результат присвоїти змінній z .
5. Вивести значення змінної z .

У командах 2–4 обчислюється значення виразу і результат обчислення присвоюється (запам'ятовується як значення) певній змінній. Такі команди називаються **командами присвоювання**. Для них прийнято використовувати таку форму запису:

2. $x := a - b$ (читається: змінній x присвоїти значення виразу $a - b$).
3. $y := c - d$ (читається: змінній y присвоїти значення виразу $c - d$).
4. $z := x*y$ (читається: змінній z присвоїти значення виразу $x*y$).

Знак $:=$ називається **знаком присвоювання** та складається з двох символів: *двокрапки* і *дорівнює*, які записуються без пропуску між ними.

Блок-схема цього алгоритму наведена на рисунку 2.4.

Цей алгоритм також є лінійним, бо при кожному наборі значень змінних a, b, c, d кожна його команда обов'язково виконується, причому тільки один раз.

Проілюструємо виконання алгоритму розв'язування задачі 2 для значень змінних $a = 3, b = 4, c = -2, d = -5$.

Команда	Результат виконання
Увести значення змінних a, b, c, d	$a = 3, b = 4, c = -2, d = -5$
$x := a - b$	$x = 3 - 4 = -1$
$y := c - d$	$y = -2 - (-5) = 3$

Команда	Результат виконання
$z := x * y$	$z = -1 * 3 = -3$
Вивести значення змінної z	$z = -3$

Аналогічно можна виконати цей алгоритм при іншому наборі значень змінних a, b, c, d .



Перевірте себе

- 1°. Наведіть приклади речень, які є командами, і приклади речень, які не є командами.
- 2°. Що таке *алгоритм*? Наведіть приклади.
- 3°. Що таке *система команд виконавця*? Наведіть приклади виконавців і системи їх команд.
- 4°. Назвіть властивості алгоритму. Поясніть кожну з них.
- 5*. Наведіть приклад послідовності команд, яка не є виконуваною.
- 6*. Наведіть приклад послідовності команд, яка не є результативною.
- 7°. Поясніть відмінність між словесною та графічною формами подання алгоритму.
- 8°. Назвіть елементи блок-схеми алгоритму та поясніть їх призначення.
- 9°. Який алгоритм (фрагмент алгоритму) називається лінійним?
- 10*. Наведіть приклади виконавців з навколишнього життя та їх системи команд. Наведіть приклади алгоритмів, які виконує кожний з них.
- 11°. Що таке команда присвоювання? Як вона позначається?



Виконайте завдання

- 1°. Укажіть команди серед наведених речень:
 - а) Закрий вікно.
 - б) Котра година?
 - в) $3 + 2 = 5$.
 - г) Не заважай читати.
 - д) Якщо йде дощ, візьми парасольку.
 - е) Я живу в Києві.
- 2*. Сформулюйте лінійні правила-алгоритми, які ви вивчали на уроках:
 - а) української мови;
 - б) математики;
 - в) інших предметів.
 Подайте ці алгоритми у вигляді блок-схем.
- 3°. Виконайте алгоритм:
 1. Накреслити відрізок AB завдовжки 5 см.
 2. Поставити вістря циркуля в точку A .
 3. Побудувати коло, радіус якого дорівнює довжині відрізка AB .
 4. Поставити вістря циркуля в точку B .
 5. Побудувати коло, радіус якого дорівнює довжині відрізка AB .
 6. Провести пряму через точки перетину побудованих кіл.
 Яку б назву ви дали цьому алгоритму? Які геометричні задачі можна розв'язувати за цим алгоритмом? Складіть його блок-схему.
- 4°. Складіть алгоритм приготування вашої улюбленої страви. Подайте його в словесній формі.
- 5°. Є координатна пряма з позначеними на ній цілими числами. На цій прямій мешкає виконавець **Коник**, який вміє переміщуватися по ній, виконуючи команди: 1) стрибни на 3 одиниці праворуч; 2) стрибни на 2 одиниці ліворуч. Початкове положення **Коника** – точка 0. Складіть блок-схему алгоритму, за яким **Коник** за найменшу кількість стрибків буде в точці: а) 24; б) 7; в) -3 .
- 6°. Човняру потрібно перевезти в човні через річку вовка, козу і капусту. У човні, крім човняра, вміщується або тільки вовк, або тільки коза, або

тільки капуста. На березі не можна залишати козу з вовком або козу з капустою. Складіть алгоритм перевезення. Подайте його у словесній формі. (Ця старовинна задача вперше трапляється в математичних рукописах VIII ст.)



- 7*. Двом солдатам потрібно переправитися з одного берега річки на інший. Вони побачили двох хлопчиків на маленькому човні. У ньому можуть переправлятися або один солдат, або один чи двоє хлопчиків. Складіть алгоритм переправлення солдатів. Подайте його у словесній формі. (Після переправлення солдатів човен повинен залишитись у хлопчиків.)
- 8*. Складіть блок-схему алгоритму побудови трикутника за трьома його сторонами, довжини яких 5 см, 6 см і 4 см, використовуючи циркуль і лінійку.
- 9*. Придумайте виконавця. Задайте його систему команд. Сформулюйте задачу та складіть алгоритм її розв'язування для цього виконавця.
- 10*. Складіть блок-схему алгоритму обчислення на калькуляторі значення виразу $(81 - 12)(58 + 84)$.
- 11*. Складіть блок-схему алгоритму знаходження x з рівняння $2x + a = c$. Виконайте його при: а) $a = 5, c = 7$; б) $a = -15, c = 105$; в) $a = 5, c = 5$.

2.3. Комп'ютерні програми і мови програмування. Етапи розв'язування задач з використанням комп'ютера



1. Для чого призначене і з чого складається програмне забезпечення комп'ютера?
2. Що таке алгоритм? Що таке система команд виконавця алгоритму? У чому полягає формальність виконання алгоритму виконавцем?
3. У чому полягає процес кодування даних? Для чого воно використовується? Що таке двійкове кодування?
4. Що таке модель об'єкта? Які види моделей ви знаєте?
5. Назвіть основні галузі застосування сучасних комп'ютерів.

Комп'ютерні програми

Ви знаєте, що комп'ютер працює під управлінням програмного забезпечення, яке складається з комп'ютерних програм (далі – програм) різноманітного призначення. Працюючи з комп'ютером у школі та вдома, ви використовували текстовий процесор і графічний редактор, програми-архіватори й антивірусні програми, табличний процесор і редактор комп'ютерних презентацій, навчальні та контролюючі програми, ігрові програми та багато інших.



Програма – це набір команд (вказівок, інструкцій), призначений для виконання комп'ютером у певній послідовності.

Програми складаються для виконання комп'ютером алгоритмів. Ці алгоритми утворюють **логіку програми (програмну логіку)**.

У процесі своєї роботи програма опрацьовує дані. Дані, які вводять до програми безпосередньо користувач програми або програма їх отримує від певного пристрою (наприклад, від датчика температури), або від іншої програми, або з іншого джерела (наприклад, з текстового файлу) називаються **вхідними (початковими) даними**. Деякі програми працюють

без вхідних даних. Дані, отримання яких є метою використання програми, називаються **вихідними (результуючими) даними**. Під час виконання програми утворюються та опрацьовуються й інші дані, які називаються **проміжними даними**.

Більшість сучасних програм у процесі своєї роботи надають користувачу певний набір засобів для його взаємодії з програмою та пристроями. До цих засобів належать засоби керування (кнопки, меню та ін.), засоби введення даних (поля, лічильники та ін.), засоби виведення даних (написи, поля і т. д.) та ін. Сукупність таких засобів, а також методів їх використання утворюють **інтерфейс користувача**.



Першим у світі програмістом вважається **Ада Лавлейс** (1815–1852), дочка відомого англійського поета Джорджа Гордона Байрона. Вона працювала з Чарльзом Беббіджем (1791–1871), розробником механічної обчислювальної машини (аналітичної машини), і вперше описала основні принципи розробки програм для обчислювальних машин. На жаль, ця обчислювальна машина так і не була створена. На честь Ади Лавлейс одна з мов програмування названа **Ada**.

Мови програмування

Складаючи алгоритми, призначені для виконання людиною, користуються мовою спілкування людей: українською, російською, англійською, німецькою тощо.

Але для алгоритмів, які повинен виконувати автоматичний пристрій (зокрема, комп'ютер), мова спілкування людей складна, має неоднозначні конструкції (наприклад, слова-омоніми). Тому для запису алгоритмів, які призначені для виконання автоматичними пристроями, розробляють і використовують спеціальні мови – **мови програмування**.



Мова, яка використовується для запису алгоритмів, призначених для виконання комп'ютером, називається мовою програмування.

Кожна мова програмування має такі компоненти:

- 1) **алфавіт** – множину символів, з яких можна утворювати слова та речення цієї мови;
- 2) **словник** – набір спеціальних (зарезервованих, ключових) слів.
- 3) **синтаксис** – правила складання та запису мовних конструкцій (не словникових слів і речень);
- 4) **семантику** – встановлене однозначне тлумачення мовних конструкцій, правил їх виконання.

Використання символів, що не входять до алфавіту, неправильне написання словникових слів, порушення синтаксичних правил призводить до неможливості виконання комп'ютером відповідної команди. Такі порушення називаються **синтаксичними помилками**.

За останні 70 років створено близько трьох тисяч різних мов програмування. Деякі з них уже вийшли з користування, для деяких постійно з'являються досконаліші версії, кожна наступна з яких зручніша для складання програм і має ширші можливості, постійно створюються нові мови програмування.

Деякі мови програмування використовуються для складання програм для розв'язування задач з різних галузей науки, техніки, виробництва,

сфери побуту та ін. (наприклад, **Delphi**, **C++**, **C#**, **Java**), а деякі створені спеціально для складання програм для розв’язування спеціального кола задач (наприклад, **Prolog** (англ. *Programming in Logic* – програмування в логіці), **Lisp** (англ. *List Processing* – опрацювання списків)).

Процесор комп’ютера може виконувати команди, подані тільки **машинною мовою**. **Машинна мова** – це мова програмування, в якій команди подаються як послідовності двійкових кодів. Машинна мова програмування орієнтована на процесори конкретної архітектури, тобто машинні мови для різних процесорів можуть відрізнятися одна від одної.

Для виконання процесором програм, написаних не машинною мовою програмування, їх потрібно спочатку перекласти на машинну мову і лише потім виконати. Для перекладу програм на машинну мову створені та використовуються спеціальні програми – **компілятори**. Ці програми аналізують весь текст програми на наявність синтаксичних помилок, і якщо такі помилки відсутні, перекладають текст програми на машинну мову, формуючи **машинний код програми**. Цей код, залежно від режиму роботи компілятора, або зберігається в пам’яті комп’ютера, або записується на диск у вигляді виконуваного файлу (наприклад, **exe**-файлу). Після отримання виконуваного файлу його можна відправити на виконання процесором. При цьому сама програма-компілятор вже не використовується. Тому виконуваний файл може використовуватися й на тих комп’ютерах, де програма-компілятор відсутня.

За наявності в програмі синтаксичних помилок, компілятор або зупиняється на першій з них і виводить на екран повідомлення про неї, або аналізує програму до кінця та виводить на екран загальний список повідомлень про наявні помилки. Після цього потрібно виправити всі синтаксичні помилки і розпочати процес компіляції знову.



Для деяких мов програмування створено інші спеціальні програми – **інтерпретатори**. Ці програми не створюють виконуваних файлів, а аналізують програму покомандно й одразу ж ці команди виконують. Тому виконати програму, яка інтерпретується, а не компілюється, можна лише на тому комп’ютері, де встановлена відповідна програма-інтерпретатор.

Для деяких сучасних мов програмування використовують комбінацію компіляції й інтерпретації. Спочатку програма компілюється в деякий проміжний код (не машинний), після чого інтерпретується спеціальною програмою, написаною для цього коду.

Етапи розв’язування задач з використанням комп’ютера

Вам уже відомо, що перші ЕОМ були створені для швидкого виконання громіздких обчислень. А сучасні комп’ютери здатні розв’язувати не тільки обчислювальні задачі, а й створювати та редагувати малюнки, текстові документи, презентації, мультфільми і кінофільми, вибирати потрібні дані з великих масивів даних, керувати роботою різноманітних пристроїв та багато іншого. Використовувати комп’ютер для розв’язування деяких із цих задач ви вже навчилися в 10-му класі, деякі з них будуть розглянуті в 11-му класі.

Усі ці задачі різні за своєю суттю, кожна з них вимагає свого, індивідуального підходу. Але все ж таки хід розв’язування кожної із задач можна поділити на кілька етапів, які є спільними для всіх задач, що

розв'язуються з використанням ПК. Розглянемо ці етапи на прикладі такої задачі.

Задача 1. Використовуючи комп'ютер, розробити проект спортивного майданчика.

Для розв'язування цієї задачі спочатку потрібно з'ясувати розміри майданчика, які об'єкти потрібно на ньому розмістити (баскетбольний і волейбольний майданчики, тенісний корт, майданчик для футболу, гімнастичні тренажери тощо) та у якому вигляді потрібно цей проект надати, наприклад у вигляді плану (креслення) такого майданчика. Цей етап називається **аналіз умови задачі, визначення вхідних даних і кінцевих результатів**. Інколи цей етап називають етапом постановки задачі.

Після цього потрібно визначити, які властивості об'єктів спортивного майданчика будуть відображені на його плані, яких значень вони набуватимуть. Деякі з цих значень є стандартними, а деякі ми повинні визначати самі. Наприклад, якщо на майданчику потрібно встановити баскетбольні щити, то їхні розміри, висота нижнього краю над землею, висота кільця над землею є стандартними, а от довжину перекладин та їхню кількість у шведській стінці ми можемо визначити самі. На цьому самому етапі потрібно встановити й залежності між об'єктами, наприклад відстані між ними, та формули, за якими можуть бути обчислені результати. Цей етап розв'язування задачі називається **побудова інформаційної моделі**.

Далі потрібно вибрати засіб, який буде використано для створення цього проекту. У даному випадку це може бути одна з прикладних програм: векторний чи растровий графічний редактор, спеціальний пакет програм для креслення тощо. А можна скласти власну програму, яка за вхідними даними задачі на основі інформаційної моделі запропонує один чи кілька варіантів проектів такого майданчика. Цей етап називається **вибір програмних засобів для розв'язування задачі**.

Якщо вирішимо використати одну з прикладних програм, то далі потрібно:

1. Вибрати ті засоби цієї програми, які нададуть можливість створити потрібний проект.
2. Скласти алгоритм його створення, враховуючи вибрані засоби програми.
3. Виконати складений алгоритм і отримати комп'ютерну модель проекту майданчика.

Ці етапи називаються **вибір засобів у середовищі прикладної програми, складання алгоритму, виконання алгоритму та отримання комп'ютерної моделі об'єкта**.

Якщо ж вирішимо скласти власну програму для розв'язування цієї задачі, то перш за все треба вибрати мову програмування. Після цього потрібно:

1. Скласти алгоритм (алгоритми) розв'язування задачі.
2. Скласти програму вибраною мовою програмування відповідно до складених алгоритмів, розробити інтерфейс користувача програми.
3. Запустити складену програму на виконання, ввести вхідні дані й одержати кінцеві результати (у даній задачі – комп'ютерну модель проекту майданчика).

Ці етапи називаються *вибір мови програмування, складання алгоритму розв'язування задачі, складання програми, виконання програми та отримання комп'ютерної моделі об'єкта*.

Під час виконання алгоритму або програми може статися так, що одну з команд алгоритму або програми виконавець не може виконати. Це означає, що алгоритм або програма містять синтаксичні помилки. Потрібно визначити всі наявні синтаксичні помилки і ліквідувати їх. Цей етап називається *налагодження алгоритму або програми*. Після налагодження потрібно знову виконати алгоритм або програму розв'язування даної задачі.

Після того як ми одержимо результат або з використанням прикладної програми, або з використанням спеціально створеної програми, його потрібно дослідити на відповідність умові поставленої задачі. Якщо всі вимоги умови задачі виконуються, то можна вважати, що задача розв'язана правильно і ми одержали правильний її розв'язок, у даному разі, проект спортивного майданчика. Якщо ж ні, потрібно шукати помилки на попередніх етапах розв'язування задачі (вони називаються *логічними помилками*), виправляти їх і знову шукати правильний розв'язок задачі, тобто такий її розв'язок, який повністю відповідає умові задачі. Цей етап називається *дослідження отриманих результатів на реальність і на відповідність умові*.

Звертаємо вашу увагу, що дослідження отриманих результатів найчастіше проводиться не на одному наборі вхідних даних, а цілій серії таких наборів (тестові набори даних), які підбираються таким чином, щоб якомога повніше, з усіх сторін, у різних ситуаціях дослідити отримані результати на їх реальність і відповідність умові задачі.

З розглянутого прикладу можна виділити такі основні етапи розв'язування задач з використанням комп'ютера (рис. 2.5):

Прослідкуємо тепер, як реалізуються розглянуті вище етапи під час розв'язування іншої задачі.

Задача 2. На торговельній базі зберігаються товари, які виготовлені в Україні й імпортуються до різних країн. Відомі їхні назви та ціни у гривнях. Перевести ціни товарів у долари, євро і фунти стерлінгів.

I етап. Аналіз умови задачі, визначення вхідних даних і кінцевих результатів

Вхідні дані:

- назви товарів;
- ціни товарів у гривнях;
- курси долара, євро і фунта стерлінгів відносно гривні.

Кінцеві результати:

- ціни товарів у доларах, євро і фунтах стерлінгів.

II етап. Створення інформаційної моделі

Очевидно, інформаційною моделлю розв'язування цієї задачі є математична модель, яка складається з формул, за якими ціна з гривень переводиться в долари, євро і фунти стерлінгів.

Ці формули виглядатимуть так:

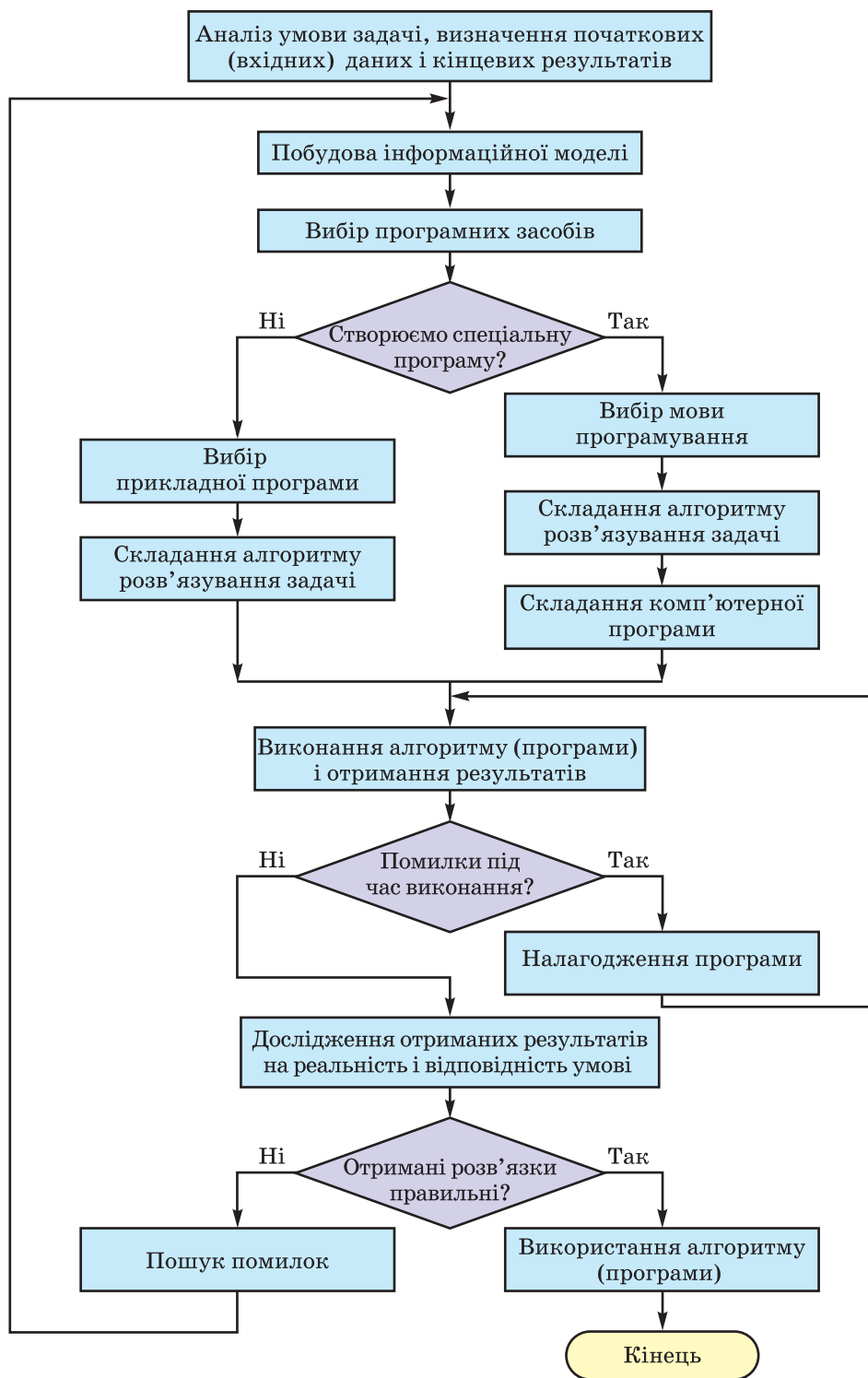


Рис. 2.5. Схема етапів розв'язування задач з використанням комп'ютера

$$D = \frac{G}{KD}, E = \frac{G}{KE}, F = \frac{G}{KF},$$

де G – ціна товару в гривнях, D, E, F – ціни товару відповідно в доларах, євро і фунтах стерлінгів, KD, KE, KF – курси долара, євро і фунта стерлінгів відносно гривні відповідно.

Крім того, одержані ціни бажано округлити до сотих.

III етап. Вибір програмних засобів для розв'язування задачі

Прикладним програмним засобом, який можна використати для розв'язування задачі, може бути, наприклад, програма-калькулятор або табличний процесор. Або можна скласти спеціальну програму для розв'язування саме цієї задачі, використовуючи одну з мов програмування.

IV етап. Створення алгоритму розв'язування задачі

Якщо буде вибрано прикладний програмний засіб табличний процесор **Excel 2007**, то алгоритм розв'язування цієї задачі може бути такий:

1. Увести в клітинки **A3:A5** текст відповідно до таблиці 2.2
2. Увести в клітинки **B3:B5** поточні курси відповідних валют.
3. Увести в клітинки **D2:H2** текст відповідно до таблиці 2.2.
4. Увести в клітинки стовпця **D**, починаючи з клітинки **D3**, назви товарів.
5. Увести в клітинки стовпця **E**, починаючи з клітинки **E3**, ціни товарів у гривнях.
6. Увести в клітинку **F3** формулу **=ROUND(E3/\$B\$3;2)**.
7. Увести в клітинку **G3** формулу **=ROUND(E3/\$B\$4;2)**.
8. Увести в клітинку **H3** формулу **=ROUND(E3/\$B\$5;2)**.
9. Виділити клітинки діапазону **F3:H3**.
10. Скопіювати вміст клітинок виділеного діапазону у відповідні клітинки стовпців **F:H**.

V етап. Отримання комп'ютерної моделі

У результаті ми одержимо таблицю, фрагмент якої ми бачимо у таблиці 2.2.

Таблиця 2.2. Таблиця розв'язування задачі в табличному процесорі Excel 2007

1	A	B	C	D	E	F	G	H
2				Товар	Ціна в гривнях	Ціна в доларах	Ціна в євро	Ціна в фунтах стерлінгів
3	Курс долара	8,02		Міксер	115,22	14,37	9,98	8,93
4	Курс євро	11,55		Торшер	216,57	27	18,75	16,79
5	Курс фунта стерлінгів	12,9		Телевізор	6615	824,81	572,73	512,79
6				Пилосос	723,25	90,18	62,62	56,07

VI етап. Дослідження отриманих результатів на реальність і на відповідність умові

Після одержання підсумкової таблиці ми повинні проаналізувати її реальність. Наприклад, якщо курс долара до гривні 8,02, то ціна в дола-

рах має бути приблизно в 8 разів менша, ніж ціна у гривнях. Можна також перевірити кілька значень, використавши інші обчислювальні засоби. Можна ввести до таблиці такі контрольні вхідні значення, для яких результат заздалегідь відомий, наприклад 802 грн. = \$100. Аналогічно потрібно перевірити роботу формул і в інших стовпцях.

Якщо таке дослідження виявляє помилки, потрібно шукати їхні причини, виправляти їх і знову проходити етапи, наступні за тим, на якому виявилися помилки.

Якщо б на III етапі було вибрано як засіб розв'язування цієї задачі складання спеціальної програми, то далі етапи розв'язування цієї задачі виглядали б так:

IV етап. Складання алгоритму розв'язування задачі

Алгоритм розв'язування цієї задачі, на основі якого складалася б комп'ютерна програма, може бути, наприклад, таким, як показано на рисунку 2.6.

Можна скласти й інші алгоритми розв'язування цієї задачі. Наприклад, спочатку ввести ціни всіх товарів у гривнях, а потім їх по черзі переводити у відповідні валюти.

V етап. Складання комп'ютерної програми

Після складання алгоритму потрібно вибрати мову програмування, розробити інтерфейс для роботи користувача з програмою та скласти програму. З цим етапом ми докладно ознайомимося у наступних пунктах.

VI етап. Налагодження програми, дослідження отриманих результатів на реальність і на відповідність умові

Після створення комп'ютерної програми потрібно її виконати на кількох наборах тестових прикладів. Якщо виявилися результати, що не відповідають очікуваням, то це означає, що програма містить помилки. Потрібно виявити ці помилки та виправити їх.

Підбір цих тестових прикладів є досить непростою та важливою справою і повинен відповідати цілому ряду вимог. Найголовніші з них такі:

1. Тестові приклади повинні бути такими, щоб можна було легко визначити кінцеві результати і порівняти їх з тими, що видає комп'ютерна програма.

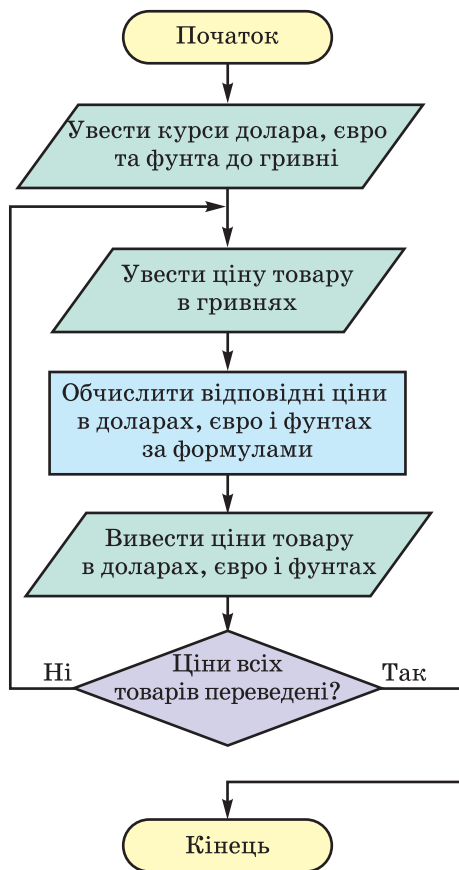



Рис. 2.6. Блок-схема алгоритму розв'язування задачі 2

2. Набір цих прикладів має бути таким, щоб він охоплював усі випадки, які можуть трапитися під час використання програми для реальних даних.




VII етап. Використання програми

Після того як ми впевнімося, що для всіх тестових прикладів програма працює правильно, можна виконати її для всіх заданих значень вхідних даних і отримати необхідні результати.

Перевірте себе

- 1°. Що таке комп'ютерна програма?
- 2°. Що таке програмна логіка й інтерфейс користувача?
- 3°. Які дані називають вхідними, вихідними, проміжними?
- 4°. Що таке мова програмування?
- 5°. Назвіть компоненти, з яких складається мова програмування.
- 6°. Опишіть кожний компонент мови програмування.
- 7°. Що таке синтаксична помилка?
- 8°. Що таке машинна мова програмування? Який вигляд мають команди в цій мові програмування?
- 9°. Які програми називаються компіляторами? Опишіть загальний принцип їх роботи.
-  10*. Які програми називаються інтерпретаторами? Опишіть загальний принцип їх роботи.
- 11°. Назвіть і опишіть кожний етап розв'язування задач з використанням ЕОМ, якщо буде використано прикладне програмне забезпечення.
- 12°. Назвіть і опишіть кожний етап розв'язування задач з використанням ЕОМ, якщо буде використано спеціально складене програмне забезпечення.

Виконайте завдання

-  1°. Знайдіть інформацію про роки створення перших версій мов програмування **Basic, PL/1, Cobol, ALGOL, Fortran, Prolog, Lisp, Java, C, C++, C#, Smalltalk**. Коротко запишіть основні характерні риси кожної з них.
-  2*. Підготуйте повідомлення про мови програмування, якими склалися програми для комп'ютерів українського виробництва у 50–60 роках ХХ ст.
3. Для кожної з наведених задач деталізуйте кожний з етапів її розв'язування з використанням комп'ютера:
 - а°) Визначити вартість фарби для фарбування підлоги в класній кімнаті.
 - б°) Визначити вартість шпалер для кабінету інформатики.
 - в°) Два потяги вийшли одночасно з пунктів *A* і *B* назустріч один одному. Відомі відстань між пунктами та швидкості потягів. Визначити відстань між ними через 2 год після виходу; через 3 год. Через скільки годин вони зустрінуться?
 - г°) Два потяги вийшли одночасно з пунктів *A* і *B* у протилежних напрямках. Відомі відстань між пунктами та швидкості потягів. Визначити відстань між ними через 1 год після виходу; через 3 год.
 - д°) У трикутнику відомі довжини двох сторін і міра кута між ними. Знайти довжину третьої сторони і його площу.
 -  е°) У паралелограмі відомі довжини двох суміжних сторін і міра кута між ними. Знайти його площу і довжини його діагоналей.

2.4. Мова програмування Delphi та середовище розробки Turbo Delphi 2006



1. Що таке комп'ютерна програма?
2. Що таке програмна логіка та інтерфейс користувача?
3. Що таке мова програмування? Назвіть основні компоненти мови програмування. Поясніть призначення кожної з них.
4. Назвіть етапи розв'язування задач з використанням комп'ютера. Поясніть, у чому полягає кожний з них.

Мова програмування Delphi та інтегроване середовище візуальної розробки проектів Turbo Delphi 2006

Ми з вами вивчатимемо об'єктно-орієнтовану мову програмування **Delphi**. Цю мову використовують як учні та студенти для навчання основних принципів об'єктно-орієнтованого програмування, так і професійні програмісти для створення великих і складних програм.

Для створення, редагування та налагодження програм мовою програмування **Delphi** використовуватимемо **інтегроване середовище візуальної розробки Turbo Delphi 2006**. Це середовище складається з текстового редактора, компілятора для перекладу програми з мови програмування **Delphi** машинною мовою, засобів налагодження програми, довідкової системи та інших компонентів. Тому таке середовище розробки називається **інтегрованим**.

Це середовище розробки дає змогу використовувати велику **бібліотеку візуальних компонентів** (англ. *Visual Component Library (VCL)*). Це вже знайомі нам елементи керування: кнопки, поля, написи, прапорці, перемикачі, лічильники, списки, смуги прокручування, а також багато ін. Під час розміщення компонентів на формі автоматично генерується відповідний фрагмент програми мовою **Delphi**. Це значно зменшує і спрощує роботу зі створення програм. Такий спосіб розробки називають **візуальним**.

Програми, створені в таких середовищах, часто називають **проектами**. Найчастіше вони складаються з кількох файлів.



Мова програмування **Delphi** – це результат розвитку й удосконалення мови програмування **Pascal**. Мова програмування **Pascal** була створена в 1970 році групою співробітників Інституту інформатики Швейцарської вищої політехнічної школи під керівництвом професора **Ніклауса Вірта** (нар. у 1934 р., рис. 2.7).



Рис. 2.7. Ніклаус Вірт


Спочатку мова програмування **Pascal** створювалася спеціально для навчання студентів основних принципів і методів процедурного програмування. Але згодом з'ясувалося, що ця мова програмування настільки зручна, що виникла ідея застосовувати її не тільки для навчання, а й для практичної роботи з ЕОМ. Так з'явилися перші компілятори з мови програмування **Pascal** і середовище розробки **Turbo Pascal**, які з часом здобули широку популярність.

З роками мова програмування **Pascal** розвивалася й удосконалювалася. Нові версії цієї мови, які вже мали елементи об'єктно-орієнтованої мови програмування, з'явилися у другій половині 80-х років ХХ ст. Деякі версії цієї мови

одержали назву **Object Pascal**. На початку 2000-х років чергова нова версія **Object Pascal** одержала назву **Delphi**.

Початкове вікно середовища Turbo Delphi 2006

Запустити **Turbo Delphi 2006** можна традиційними способами:

- виконати **Пуск** \Rightarrow **Усі програми** \Rightarrow **Borland Developer Studio 2006** \Rightarrow **Turbo Delphi**;
- використати ярлик  на **Робочому столі**;
- двічі клацнути на значку файлу з розширенням імені **bdsproj**.

У перших двох випадках відкривається початкове вікно **Turbo Delphi 2006** (рис. 2.8). У верхній частині цього вікна розміщені традиційні **Рядок меню** та **Панель інструментів**. У лівій і правій частинах цього вікна розташовані підлеглі вікна:

- **Structure** (англ. *structure* – структура) – у ньому відображається структура компонентів (елементів керування) проекту;
- **Object Inspector** (англ. *object inspector* – інспектор об'єкта) – у ньому відображаються властивості та методи поточного компонента проекту;
- **Project Manager** (англ. *project manager* – управляючий проектом) – у ньому відображається структура файлів проекту;
- **Tool Palette** (англ. *tool palette* – палітра інструментів) – у ньому відображається список об'єктів, доступних для використання в поточному режимі.

Поруч з кнопками закриття цих вікон розташовані кнопки їхнього згортання \downarrow . Вони схожі на звичайні канцелярські кнопки. За вибору

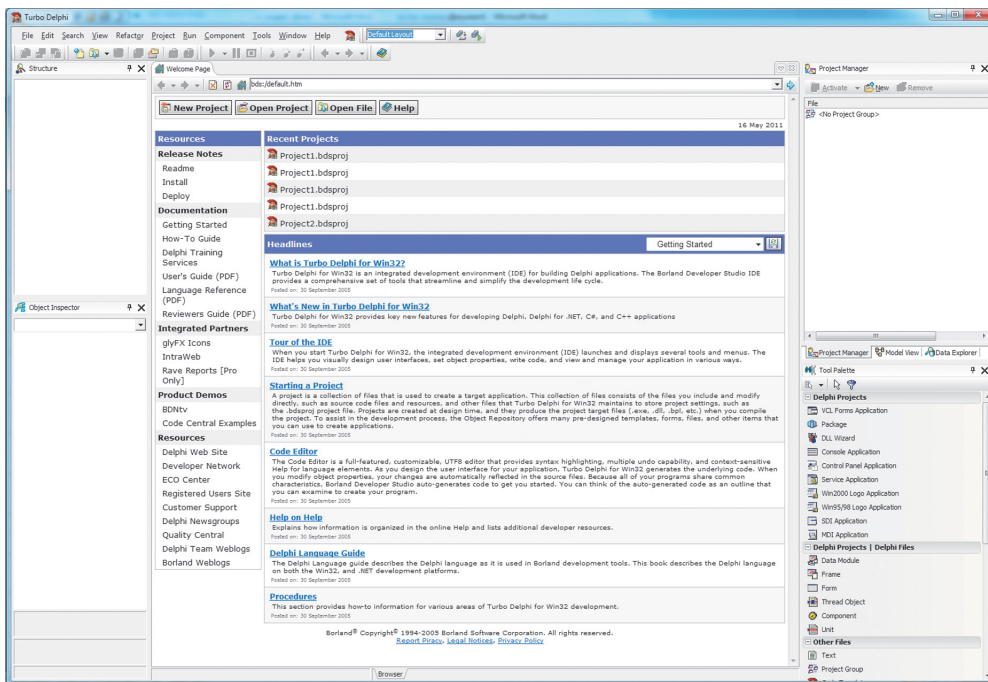



Рис. 2.8. Початкове вікно середовища **Turbo Delphi 2006**

цих кнопок вони змінюють своє положення на таке ➔, а самі вікна згортаються. Якщо ці вікна закрити, то повторно відкрити їх можна, виконавши відповідні команди меню **View** (англ. *view* – вигляд).

Інколи у початковому вікні **Turbo Delphi 2006** можуть бути відкриті інші підлеглі вікна. Це залежить від режиму показу цього вікна, який встановлюється в **Рядку меню** в полі зі списком, що відкривається. Для нашої подальшої роботи потрібно, щоб був встановлений режим **Default Layout** (англ. *default layout* – розташування за замовчуванням) або **Classic Undocked** (англ. *classic undocked* – класичний відстикований).

У центральній частині відкрито вкладки **Welcome Page** (англ. *welcome page* – сторінка запрошення або стартова сторінка). На цій вкладці розташовані:

- кнопка **New Project** (англ. *new project* – новий проект) – для відкриття у середовищі заготовки нового проекту;
- кнопка **Open Project** (англ. *open project* – відкрити проект) – для відкриття у середовищі проекту, записаного на носії;
- кнопка **Help** (англ. *help* – допомога) – для відкриття системи допомоги та ін.

Кнопка закриття вкладки знаходиться в правій частині рядка з іменем вкладки. Поруч з нею знаходиться кнопка  відкриття списку відкритих вкладок.

Якщо вибрати третій з перерахованих вище способів запуску **Turbo Delphi 2006**, то початкове вікно середовища не відкривається, а одразу відкривається **головне вікно середовища** і в ньому відкривається відповідний проект.

Створення, виконання та збереження нового проекту в середовищі Turbo Delphi 2006

Для створення нового проекту мовою програмування **Delphi** в середовищі **Turbo Delphi 2006** потрібно в початковому вікні вибрати кнопку **New Project**. У результаті цього вибору відкриється вікно **New Items** (англ. *new items* – нові пункти, рис. 2.9).

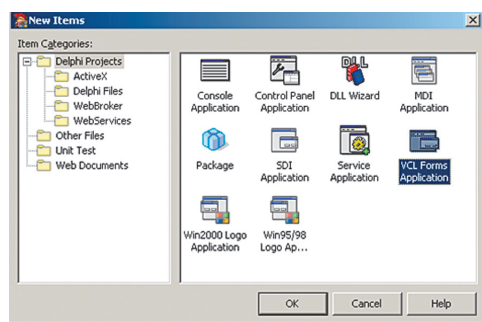


Рис. 2.9. Вікно **New Items**

Середовище **Turbo Delphi 2006** дає змогу створювати проекти різних видів. Ми будемо вчитися створювати проекти з вікнами, кнопками, меню, полями та іншими компонентами (елементами керування), які утворюватимуть інтерфейс користувача. Такі проекти в **Turbo Delphi 2006** називаються **VCL Forms Application** (англ. *forms application* – додаток (програма) з формами). Для створення таких проектів потрібно у вікні **New Items**:

1. Вибрати у лівій частині вікна елемент списку **Delphi Projects**.
2. Вибрати у правій частині вікна значок **VCL Forms Application**.
3. Вибрати кнопку **OK**.

Після вибору кнопки **OK** відкривається головне вікно середовища **Turbo Delphi 2006** (рис. 2.10).

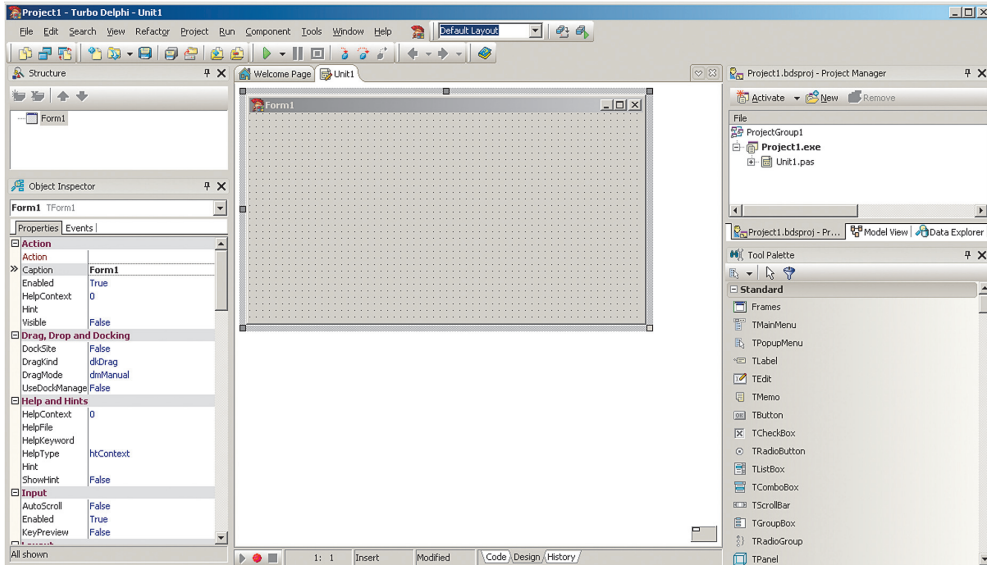


Рис. 2.10. Головне вікно середовища **Turbo Delphi 2006**

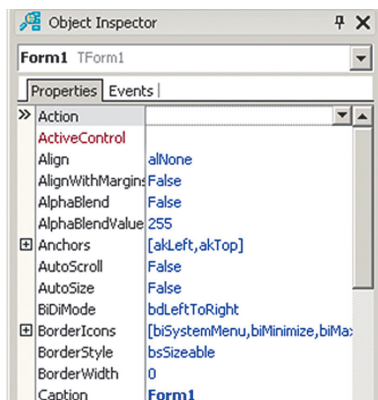
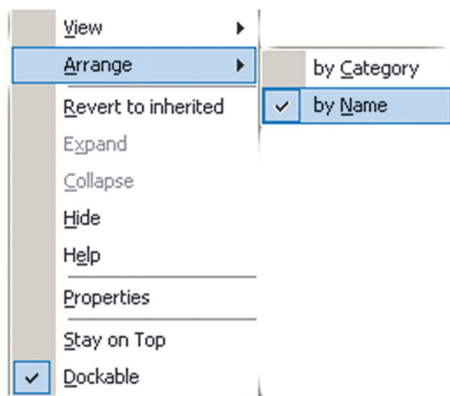
У центральній частині цього вікна знаходиться відкрита вкладка **Unit1** (англ. *unit* – частина), яка має три внутрішні вкладки: **Code** (англ. *code* – код), **Design** (англ. *design* – дизайн, проект, план), **History** (англ. *history* – історія). За замовчуванням відкривається вкладка **Design**, на якій розташована **форма**.



Форма – це об’єкт, на якому можна розмістити кнопки, поля, написи, меню та інші компоненти для створення інтерфейсу користувача. Під час виконання проекту форма відкривається як вікно **Windows**, на якому відображатимуться компоненти, розміщені на формі, якщо вони не є прихованими або невізуальними.

Форма, як і будь-який інший об’єкт, має властивості. Якщо форма є поточним об’єктом, то його властивості та їхні значення відображаються на вкладці **Properties** (англ. *properties* – властивості) вікна **Object Inspector**. Наведемо приклади деяких з цих властивостей:

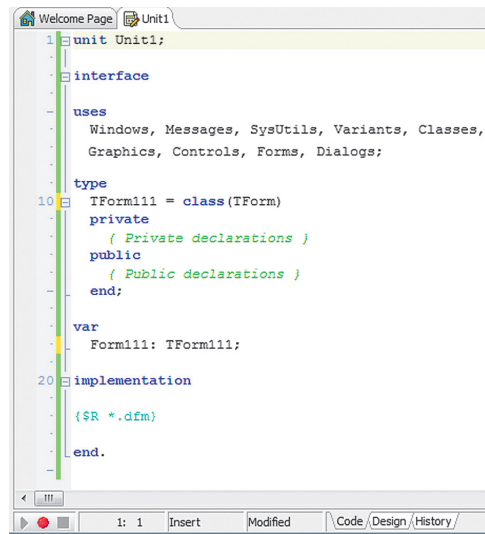
- **Caption** (англ. *caption* – заголовок) – її значенням є текст у рядку заголовка форми;
- **Color** (амер. *color* – колір) – її значенням є колір фону форми;
- **Height** (англ. *height* – висота) – її значенням є висота форми в пікселях; нагадаємо, **піксел** – це мінімальний елемент графічного зображення на екрані;
- **Width** (англ. *width* – ширина) – її значенням є ширина форми в пікселях;
- **Top** (англ. *top* – верх) – її значенням є відступ у пікселях верхньої границі вікна, що відкриється під час виконання проекту, від верхньої межі екрана;
- **Left** (англ. *left* – лівий) – її значенням є відступ у пікселях лівої границі вікна, що відкриється під час виконання проекту, від лівої межі екрана;
- **Name** (англ. *name* – ім’я) – її значенням є ім’я форми як об’єкта.

Рис. 2.11. Вкладка **Properties** вікна **Object Inspector**Рис. 2.12. Контекстне меню вкладки **Properties**

Значення більшості властивостей встановлюються за замовчуванням. Так, наприклад, значення властивостей **Name** і **Caption** за замовчуванням встановлюються *Form1*, значення властивості **Height** – 338, значення властивості **Width** – 651. Ці значення, як і значення інших властивостей, можна змінити або встановити на вкладці **Properties**.

Список властивостей може бути впорядкований за категоріями (рис. 2.10) або за іменами (рис. 2.11). Змінити порядок розташування властивостей у списку можна, використавши команду **Arrange** контекстного меню вкладки **Properties** (англ. *arrange* – розставляти в певному порядку, *by category* – за категоріями, *by name* – за іменами) (рис. 2.12). Радимо вам для зручності пошуку необхідних властивостей впорядковувати список властивостей за іменами.

Відкривши вкладку **Code**, можна продивитися текст програми, який створюється за замовчуванням у середовищі **Turbo Delphi 2006** під час створення нового проекту (рис. 2.13).

Рис. 2.13. Вкладка **Code** з текстом програми за замовчуванням

Навіть цього тексту програми достатньо, щоб виконати створений за замовчуванням проект. Для виконання проекту можна використати команду **Run** (англ. *run* – виконання, запуск) меню **Run** або кнопку **Run** панелі інструментів.

Під час виконання проекту, який створюється за замовчуванням, на екрані відкривається порожнє вікно, значення властивостей якого (колір, розміри, положення на екрані, текст у рядку заголовка та ін.) встановлені за замовчуванням (рис. 2.14).

Для завершення виконання проекту потрібно закрити його вікно.

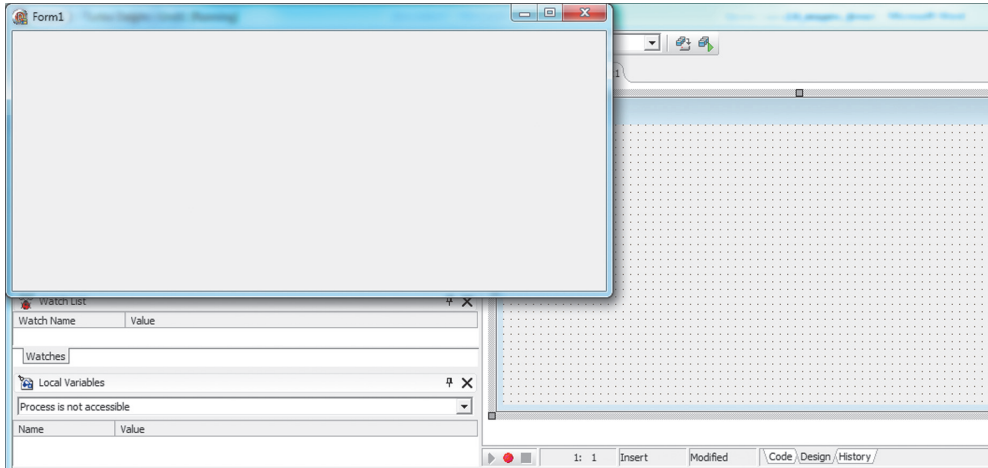


Рис. 2.14. Виконання проекту, створеного за замовчуванням

У вікні **Structure** головного вікна середовища **Turbo Delphi 2006** відображається єдиний об'єкт, який на даний момент використовується в проекті – форма з іменем *Form1*, у вікні **Tool Palette** – список усіх доступних компонентів, які можна використати в проекті, а вікно **Project Manager** містить список деяких файлів, які входять до складу проекту (див. рис. 2.10).

Для збереження всіх файлів проекту потрібно використати команду **Save All** (англ. *save all* – зберегти все) меню **File** (англ. *file* – файл) або кнопку **Save All** панелі інструментів.

У вікні, що відкриється, доцільно перш за все для кожного проекту створити окрему папку, у якій і зберегти файли проекту.

Turbo Delphi 2006 послідовно пропонує записати два файли проекту:

- файл з іменем **Unit1** і розширенням імені **pas**;
- файл з іменем **Project1** і з розширенням імені **bdsproj**.

Розширення імен змінювати не варто, а імена файлів можна змінити.

*Звертаємо вашу увагу, що в **Turbo Delphi 2006** ім'я файлу може містити тільки літери англійського алфавіту, цифри та знак підкреслення, і цифра не повинна бути першим символом імені.*

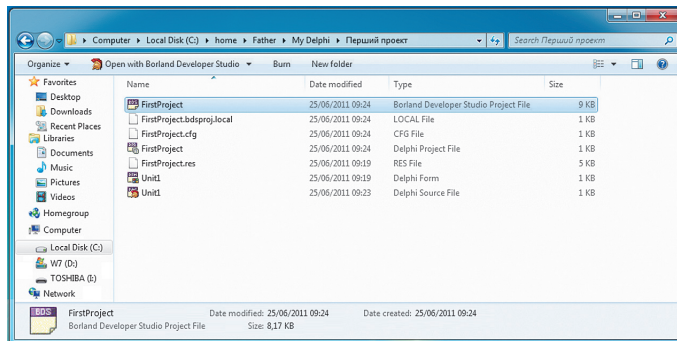


Рис. 2.15. Папка **Перший проект** з файлами проекту

Крім тих двох файлів, які нам пропонується записати на зовнішній носій, **Turbo Delphi 2006** автоматично створює та зберігає ще кілька файлів, які входять до даного проекту (рис. 2.15).

Щоб завершити роботу над проектом, потрібно виконати команду **Close All** (англ. *close* – закрити, *all* – усі) меню **File**.

Звертаємо вашу увагу, що одночасно в середовищі може бути відкритим тільки один проект.

Редагування проекту

До початкового проекту, а також до будь-якого іншого, можна вносити зміни. Наприклад, можна на вкладці **Properties** вікна **Object Inspector** змінити значення деяких властивостей форми. Для зміни значення певної властивості треба ввести потрібне значення в поле справа від назви властивості або вибрати його зі списку.

Так, наприклад, якщо встановити такі значення властивостей:

- **Caption** = *Моє перше вікно*
- **Color** = *clRed* (англ. *colour* – колір, *red* – червоний)
- **Height** = *450*
- **Width** = *700*
- **Left** = *600*
- **Top** = *350*

і виконати змінений проект, то відкриється вікно іншого кольору, інших розмірів, розташоване в іншому місці екрана, з іншим текстом у рядку заголовка (рис. 2.16).

Звертаємо вашу увагу, що змінити значення властивостей **Height** і **Width** форми можна перетягуванням маркерів на її межі.

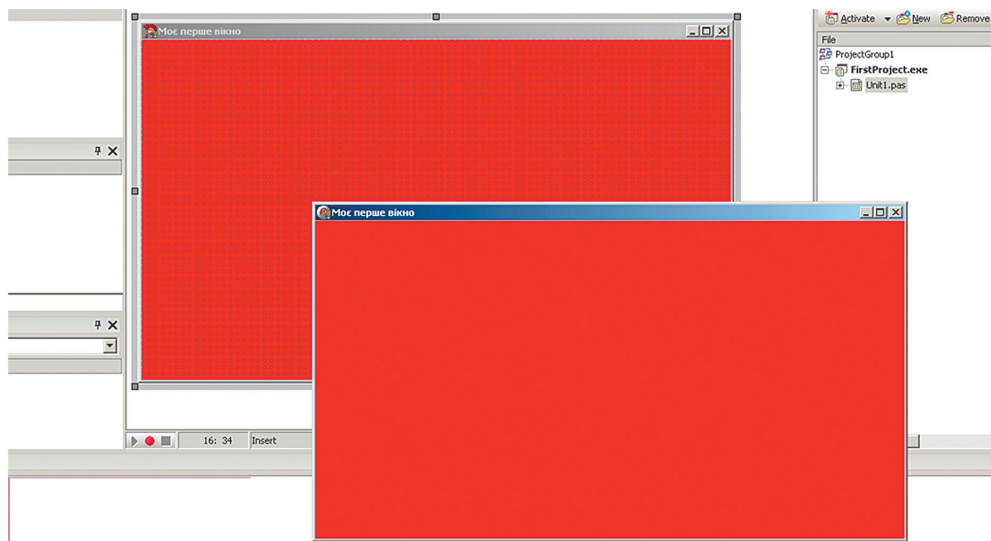


Рис. 2.16. Виконання зміненого проекту

Розміщування елементів керування на формі

Працюючи з різноманітними програмами, ви використовували різні елементи керування:

- кнопки – для виконання певних операцій;
- поля – для введення даних;
- написи – для виведення текстів і чисел;
- перемикачі та прапорці – для встановлення певних режимів роботи програми та ін.

Для того щоб під час виконання програми елементи керування утворили інтерфейс користувача, потрібно перед запуском програми на виконання розмістити їх на формі. Розглянемо, як розмістити на формі кнопку, поле, напис і область зображення (останній елемент керування призначений для розміщення на формі графічного зображення). Для цього потрібно:

1. Зробити форму поточною.
2. Знайти у вікні **Tool Palette** рядок:
 - для кнопки – **TButton** (англ. *type* – тип; *button* – кнопка) у групі **Standard** (англ. *standard* – стандарт);
 - для поля – **TEdit** (англ. *edit* – редагувати) у групі **Standard**;
 - для напису – **TLabel** (англ. *label* – мітка, напис) у групі **Standard**;
 - для області зображення – **TImage** (англ. *image* – зображення) у групі **Additional** (англ. *additional* – додатковий).
3. Двічі клацнути на відповідному рядку.

Після виконання цього алгоритму вибраний об'єкт з'являється в центральній частині форми. Далі можна перетягуванням розмістити об'єкт в іншому місці форми (рис. 2.17). При розміщенні нового елемента керування на формі у вікні **Structure** змінюється структура елементів керування, які використовуються в проекті, – значок цього об'єкта та його

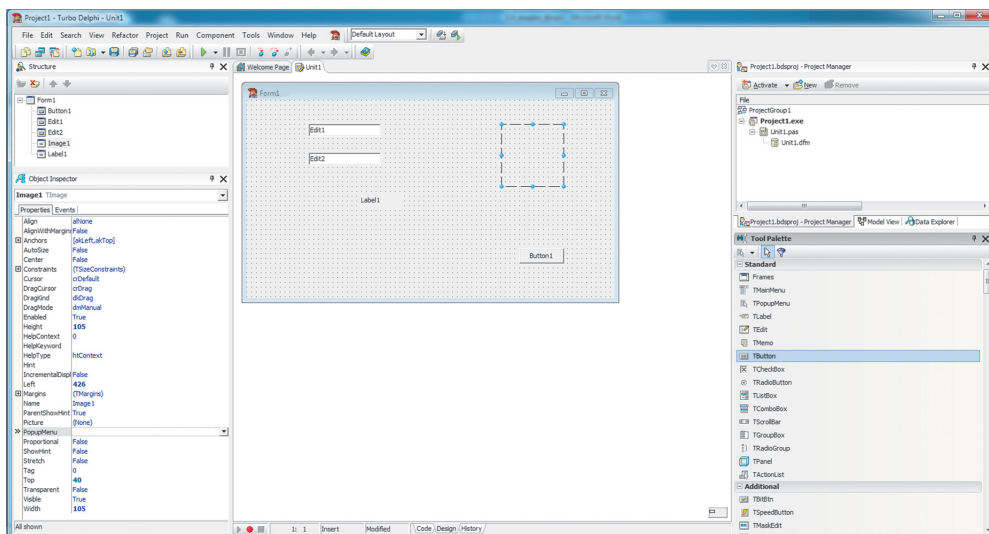


Рис. 2.17. Форма з розміщеними на ній двома полями, написом, кнопкою і областю зображення

ім'я додаються до структури, у вікні **Object Inspector** на вкладці **Properties** відображаються властивості цього об'єкта та їхні значення, а у полі, що знаходиться над вкладками цього вікна, з'являється його ім'я. У списку цього поля вказуються імена всіх елементів керування, що використовуються в проекті. Використовуючи цей список, можна зробити відповідний елемент поточним.


Кнопка, поле, напис і область зображення, як і форма, мають властивості **Name, Height, Width, Left, Top**, а також цілий ряд інших властивостей. Значення властивостей цих об'єктів під час розміщення на формі встановлюються за замовчуванням. Їх можна змінити, увівши нові значення на вкладці **Properties** вікна **Object Inspector**. Значення деяких з них можна змінити перетягуванням меж об'єкта або самого об'єкта по формі.

Звертаємо вашу увагу:

1. Для кнопки, поля, напису і для області зображення, на відміну від форми, значення властивостей **Left** і **Top** визначають положення об'єкта не відносно лівої та верхньої меж екрана, а відносно лівої та верхньої меж форми, на якій вони розміщені.
2. Поле і напис мають властивість **Color**, а кнопка і область зображення цієї властивості не мають.
3. Кнопка і напис мають властивість **Caption**, а поле і область зображення цієї властивості не мають.
4. Замість властивості **Caption** поле має властивість **Text** (англ. *text* – текст), значення якої визначає, який текст знаходиться в полі.

Розглянемо ще дві властивості, які мають усі елементи керування, розглянуті вище:

- **Enabled** (англ. *enabled* – доступний) – її значення визначає, чи доступний об'єкт для операцій над ним. Ця властивість може набувати лише одне з двох значень: *True* (англ. *true* – істина, правда) – об'єкт доступний, *False* (англ. *false* – хибність, неправда) – об'єкт недоступний;
- **Visible** (англ. *visible* – видимий) – її значення визначає, чи видимий даний об'єкт у вікні. Ця властивість також може набувати лише одне з двох значень: *True* – об'єкт видимий, *False* – об'єкт невидимий.

Усі елементи керування, розглянуті вище, крім області зображення, мають властивість **Font** (англ. *font* – шрифт). Її значення визначають властивості символів, якими буде виводитися текст на об'єкті. На відміну від усіх попередніх властивостей, ця властивість є **комплексною**. Комплексною вона називається тому, що складається з кількох властивостей символів: **Color, Height, Name, Size** та ін. Значення кожної з них можна змінити, якщо відкрити їх список вибором кнопки  зліва від назви властивості **Font**. А якщо вибрати кнопку з трьома точками, то відкриється знайоме діалогове вікно для встановлення значень властивостей символів.

Розглянемо детальніше область зображення. За замовчуванням цей компонент не пов'язаний ні з яким зображенням. Щоб розмістити в ньому зображення, необхідно змінити значення його властивості **Picture** (англ. *picture* – малюнок): увести у відповідне поле повне ім'я потрібного графічного файлу або вибрати файл із зовнішнього носія. Для вибору файлу із зовнішнього носія треба:

1. Вибрати кнопку в правому кінці поля **Picture**.

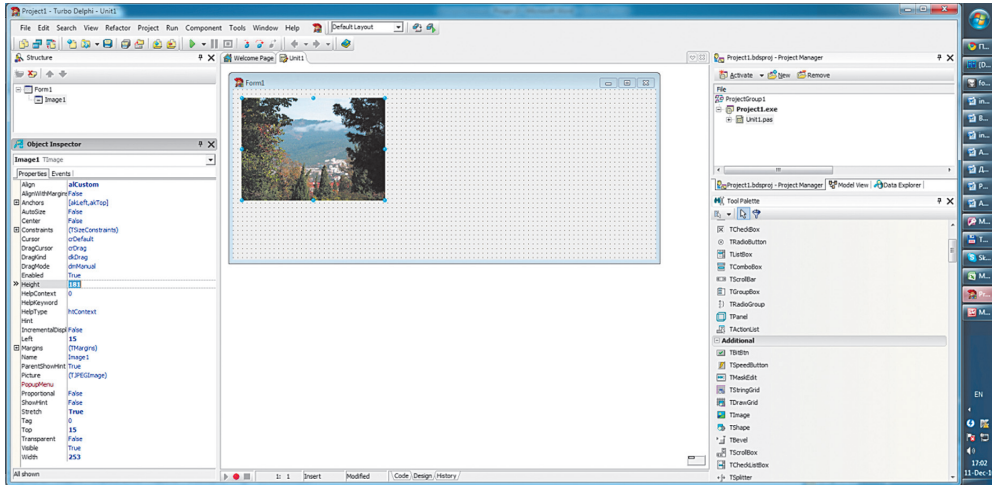


Рис. 2.18. Зображення, розміщене на формі, зі зміненими значеннями деяких його властивостей

2. Вибрати кнопку **Load** (англ. *load* – завантажити) у вікні **Picture Editor** (англ. *editor* – редактор).
3. Відкрити у вікні **Load Picture** необхідну папку, вибрати потрібний графічний файл і кнопку **Open** (англ. *open* – відкрити).
4. Вибрати кнопку **OK**.

Відкрите зображення буде розміщено в області зображення, причому, залежно від розмірів компонента і зображення, може бути видимою лише частина зображення. Після цього доцільно змінити значення таких властивостей області зображення (рис. 2.18):

- установити значення властивості **Stretch** = *True* (англ. *stretch* – розтягнути), щоб зображення зайняло всю область компонента;
- установити значення властивості **Proportional** = *True* (англ. *proportional* – пропорційний), щоб зображення не спотворювалося під час розтягування;
- установити значення властивості **Align** = *alCustom* (англ. *align* – вирівняти, *custom* – налаштувати), щоб можна було налаштувати розташування компонента на формі (якщо вибрати значення цієї властивості *alClient* (англ. *client* – клієнт), то зображення покриватиме всю поверхню форми);
- установити такі значення властивостей **Top**, **Left**, **Width**, **Height**, щоб компонент мав необхідні розміри та розташовувався в потрібному місці форми.





Перевірте себе

- 1°. Назвіть складові частини інтегрованого середовища візуальної розробки **Turbo Delphi 2006**.
- 2°. Що таке бібліотека візуальних компонентів? Для чого вона використовується?
- 3°. Назвіть способи запуску **Turbo Delphi 2006**.
- 4°. Назвіть і опишіть призначення та вміст підлеглих вікон початкового вікна **Turbo Delphi 2006**.

- 5°. Як створити новий проект у **Turbo Delphi 2006**?
- 6°. Що таке форма?
- 7°. Назвіть та опишіть, що визначають значення властивостей форми. Як змінити значення властивостей?
- 8°. Як зберегти проект на зовнішньому носії?
- 9°. Які вимоги до імені файлу в **Turbo Delphi 2006**?
- 10°. Як виконати проект?
- 11°. Як розмістити на формі елементи керування?
- 12°. Назвіть приклади властивостей відомих вам елементів керування. Що визначають їхні значення?
- 13°. Як змінити значення властивостей елементів керування?



Виконайте завдання

- 1°. Відкрийте середовище розробки **Turbo Delphi 2006**. Виконайте проект, створений за замовчуванням. Створіть у власній папці папку **Проект 2.4.1** і збережіть у ній проект, змінивши ім'я файлу **Unit1** на **task_2_4_1**, а **Project1** на **project_2_4_1**. Установіть розміри форми: **Height = 500** і **Width = 100**, колір **Color = clGreen** і знову виконайте проект. Збережіть відредагований проект.
-  2°. Відкрийте **Turbo Delphi 2006**. Збільшіть розміри форми. Змініть напис у рядку заголовка форми на *Я вчуся програмувати*. Змініть значення властивостей форми **Top** і **Left**, щоб під час виконання проекту вікно відкрилося в правій нижній частині екрана. Створіть у власній папці папку **Проект 2.4.2** і збережіть у ній проект.
- 3°. Відкрийте **Turbo Delphi 2006**. Створіть проект, виконання якого відкриє вікно синього кольору, розмірами 300 на 300 пікселів, ліва межа якого знаходиться на відстані 300 пікселів від лівої межі екрана, а верхня – на відстані 400 пікселів від верхньої межі екрана. Створіть у власній папці папку **Проект 2.4.3** і збережіть у ній проект.
- 4°. Відкрийте **Turbo Delphi 2006**. Розмістіть на формі область зображення. Задайте значення властивостей цього компонента, щоб його розміри були 150 на 200 пікселів і щоб його ліва межа знаходилася на відстані 30 пікселів від лівої межі форми, а верхня – на відстані 20 пікселів від верхньої межі форми. Зв'яжіть із цим компонентом графічне зображення, наприклад з файлу **pic 2.4.4**. Установіть такі значення властивостей області зображення, щоб воно займало всю область. Виконайте проект. Створіть у власній папці папку **Проект 2.4.4** і збережіть у ній проект.
-  5*. Відкрийте **Turbo Delphi 2006**. Розмістіть на формі область зображення. Розмістіть у ньому довільне зображення. Дослідіть експериментально, що визначають інші значення властивості **Align**. Запишіть висновки в зошит.
- 6*. Відкрийте **Turbo Delphi 2006**. У меню **File** є команди **Save**, **Save As**, **Save Project As**. З'ясуйте призначення кожної з них та їхню відмінність від команди **Save All**. Запишіть висновки в зошит.
-  7*. Відкрийте **Turbo Delphi 2006**. З'ясуйте відмінність команд **Close** і **Close All** меню **File**. Запишіть висновки в зошит.
- 8°. Відкрийте готовий проект, наприклад з папки **Проект 2.4.8**. З'ясуйте значення п'яти властивостей форми та чотирьох властивостей кнопки. Запишіть їх у зошит.
-  9°. Відкрийте готовий проект, наприклад з папки **Проект 2.4.9**. Порівняйте значення властивостей поля та напису. Зробіть у зошиті порівняльну таблицю.
- 10*. У **Довідці** знайдіть пояснення про 5–6 властивостей форми, кнопки, поля, напису, які не були розглянуті в даному пункті. Зробіть відповідні записи в зошит.



Практична робота № 5. Створення та виконання проекту. Використання елементів керування

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте **Turbo Delphi 2006**.
2. Змініть значення властивостей форми на такі:
 - **Caption** = *Практична 5*
 - **Color** = *clGreen* (англ. *green* – зелений)
 - **Height** = *580*
 - **Width** = *580*
 - **Left** = *500*
 - **Top** = *150*
3. Розмістіть на формі два поля, напис і кнопку відповідно до рисунка 2.19.
4. Розмістіть на формі область зображення. Змініть значення її властивостей на такі:
 - **Stretch** = *True*
 - **Align** = *alCustom*
 - **Left** = *10*
 - **Top** = *30*
 - **Height** = *150*
 - **Width** = *130*

Розташуйте у цій області зображення, наприклад з файлу **практична 5.4.jpg**. Установіть значення властивості області, щоб зображення займало всю область.

5. Розмістіть на формі ще одну область зображень. Змініть значення її властивостей на такі, щоб вона мала розміри 60 на 60 пікселів і розташовувалася у правому верхньому куті вікна. Розташуйте в області зображення, наприклад з файлу **практична 5.5.jpg**.
6. Створіть у власній папці папку **Практична 5** і збережіть у ній проект.
7. Виконайте проект.

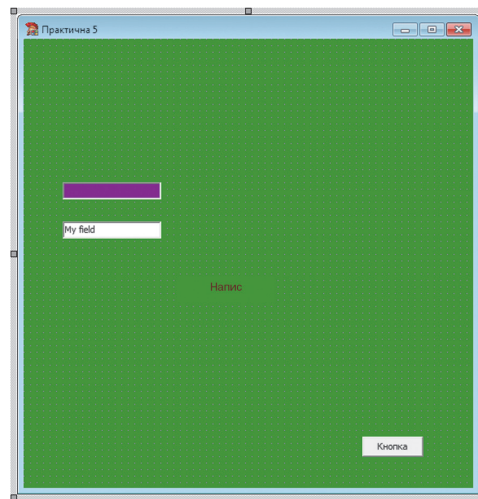


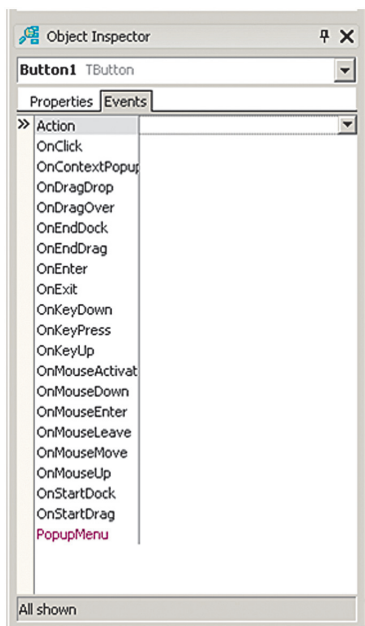
Рис. 2.19

2.5. Події та їх обробники



1. Назвіть вікна **Turbo Delphi 2006** і поясніть їх призначення.
2. Що таке форма? Назвіть кілька її властивостей. Як змінити їх значення?
3. Які спільні властивості мають кнопка, поле, напис і область зображення? Які властивості мають одні з указаних об'єктів і не мають інші?
4. Сформулюйте правила запису виразів у формулах в **Excel**.
5. Що таке алгоритм? Який алгоритм називається лінійним? Який вигляд має блок-схема лінійного алгоритму?

Події та їх обробники

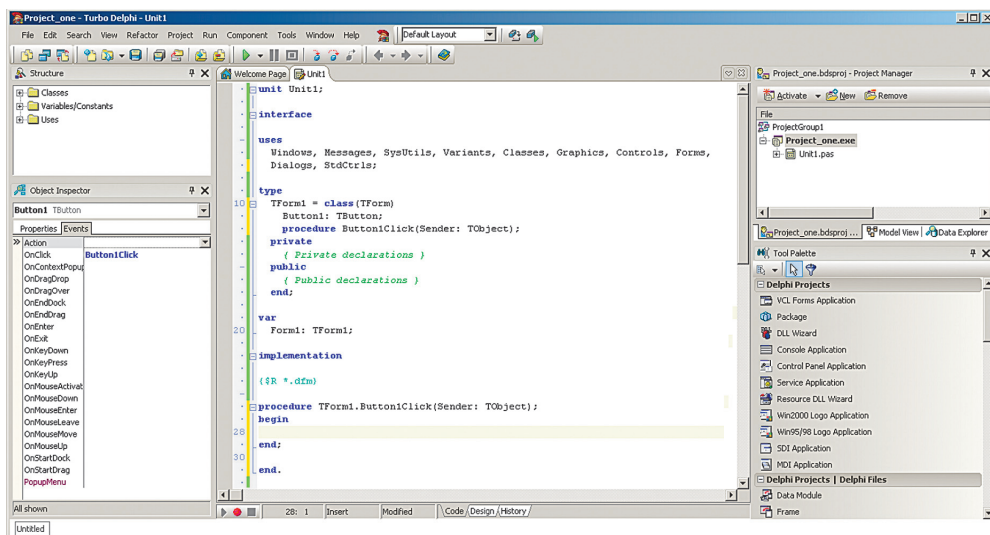
Рис. 2.20. Вкладка **Events** для вибраної кнопки **Button1**

Працюючи з різноманітними програмами, ви звикли до того, що вибір кнопки завжди приводив до настання певної події: відкриття або закриття певного вікна, змінення значень властивостей певних об'єктів, переміщення певного об'єкта по екрану тощо. Але якщо вибрати кнопку в будь-якому з проектів, які ви створювали під час вивчення попереднього пункту, то нічого не відбудеться.

Причиною цієї ситуації є те, що лише самого розміщення на формі кнопки недостатньо, щоб за її вибору щось відбулося. Програмі потрібно «повідомити», яка нова подія повинна відбутися за вибору кнопки.

Якщо виділити кнопку, то у вікні **Object Inspector** на вкладці **Events** (англ. *events* – події) можна вказати, яка подія повинна відбутися як реакція на настання іншої події, наприклад вибір кнопки (рис. 2.20). Для цього потрібно двічі клацнути в полі справа від напису **OnClick** (англ. *on click* – на клацання кнопкою миші). У результаті цих дій у полі **OnClick** на вкладці **Events**

з'являється текст **Button1Click** (**Button1** – це ім'я виділеної кнопки), а у центральній частині вікна середовища розробки відкривається вкладка **Code**, на якій створюється заготовка процедури **TForm1.Button1Click**, команди якої й будуть виконуватися після вибору кнопки **Button1** (рис. 2.21).

Рис. 2.21. Вкладка **Code** із заготовкою процедури **TForm1.Button1Click**



Процедура (лат. *procedere* – просуватися, йти вперед) – це частина програми, яка має ім'я та яку можна за цим іменем викликати на виконання в різних частинах програми.

Отже, після вибору кнопки **Button1** (подія **Click**) настає подія **OnClick**, яка полягає у виконанні команд процедури **TForm1.Button1Click**.

Процедури в **Delphi** є одним з видів підпрограм. Іншим видом підпрограм у **Delphi** є функції. Їх розглянемо детальніше в наступних пунктах.



Процедура, яка виконується при настанні певної події, називається обробником цієї події.

Процедура, яка пов'язана з певним об'єктом, називається методом цього об'єкта.

Так, процедура **TForm1.Button1Click** є обробником події **OnClick**, яка настає після вибору кнопки **Button1**, а також є методом об'єкта «Кнопка **Button1**».

Аналогічно викладеному вище можна створювати обробники інших подій, наприклад:

- **OnMouseMove** (англ. *on mouse move* – на переміщення миші) – ця подія настає після наведення вказівника на кнопку;
- **OnKeyPress** (англ. *on key press* – на натиснення клавіші) – ця подія настає після натиснення клавіші клавіатури;
- **OnStartDrag** (англ. *on start drag* – на початок перетягування) – ця подія настає після початку перетягування об'єкта та ін.

Аналогічно можна створювати методи й інших об'єктів, наприклад форми. Список подій на вкладці **Events** для форми містить події, які ми вже бачили на вкладці для кнопки, а також деякі інші події. Наприклад:

- **OnCreate** (англ. *on create* – на створення) – ця подія настає після початку створення форми; команди обробника цієї події виконуються під час створення форми, перед її відкриттям;
- **OnDblClick** (англ. *on double click* – на подвійне клацання) – ця подія настає після подвійного клацання на формі.

Створення процедур-обробників подій у Delphi

Розглянемо детальніше структуру процедури **TForm1.Button1Click** – обробника події **OnClick**:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    end;
```

Перший рядок процедури – **рядок заголовка**. Він складається зі стандартного слова **procedure** (англ. *procedure* – процедура), імені процедури **TForm1.Button1Click** і круглих дужок, у яких вказується, що дана процедура виконуватиметься в результаті настання певної події з певним об'єктом, у даному випадку події **Click** з кнопкою **Button1**. Закінчується рядок заголовка **крапкою з комою (;)**.

Команди процедури, які будуть виконуватися при настанні події **OnClick**, записуються між двома стандартними словами **begin** (англ. *begin* – початок) і **end** (англ. *end* – кінець). Команди процедури утворюють **тіло процедури**. Слова **begin** і **end** визначають, де починається і де закінчується тіло процедури. Можна сказати, що вони відіграють роль відкриваючої та закриваючої дужки, між якими записується тіло процедури, тому їх називають **операторними дужками**. Після слова **end** повинна стояти крапка з комою (;). Кожна команда процедури також повинна закінчуватися крапкою з комою (;).

У попередньому пункті ми встановлювали значення властивостей елементів керування перед запуском проекту на виконання. Тепер же можемо змінювати значення цих властивостей під час виконання проекту. Для цього потрібно додати до тексту програми обробник певної події.

Якщо потрібно, щоб у результаті виконання процедури змінилися значення властивостей одного з об'єктів, процедура повинна містити **команди встановлення значення властивостей об'єкта**. Загальний вигляд цих команд такий:

<ім'я об'єкта>. <ім'я властивості> := <значення або вираз>;

Наприклад,

Form1.Color := clGreen; – встановити колір фону вікна – *зелений*;

Form1.Left := 300; – встановити відступ лівої межі вікна – *300 пікселів*;

Form1.Width := Form1.Width*2; – збільшити поточну ширину вікна вдвічі;

Button1.Caption := 'Збільшити'; – встановити текст заголовка на кнопці *Збільшити* (якщо значенням властивості є текст, то його потрібно брати в одинарні лапки);

Label1.Font.Color := clRed; – встановити колір символів, яким відображатиметься текст у написі, *червоний*.

Звертаємо вашу увагу, що ці команди є різновидами команд призначення, які ми вже використовували під час складання алгоритмів.

Наведемо приклад процедури **TForm1.Button1Click**, виконання якої змінить колір вікна на синій, змінить текст у рядку заголовка, зменшить його ширину на 150 пікселів, змінить положення кнопки і текст на ній:

```
procedure TForm1.Button1Click (Sender: TObject);
begin
  Form1.Color := clBlue;
  Form1.Caption := 'Значення властивостей змінилися';
  Form1.Width := Form1.Width - 150;
  Button1.Left := 60;
  Button1.Top := 100;
  Button1.Caption := 'Змінити!';
end;
```

На рисунку 2.22 зліва наведено вигляд вікна після запуску проекту, а справа – після вибору кнопки **Button1**.

Середовище розробки **Turbo Delphi 2006** має зручні засоби допомоги під час введення тексту проекту:

- після введення першої літери імені об'єкта можна натиснути сполучення клавіш **Ctrl+Пропуск** і відкриється список імен об'єктів, влас-

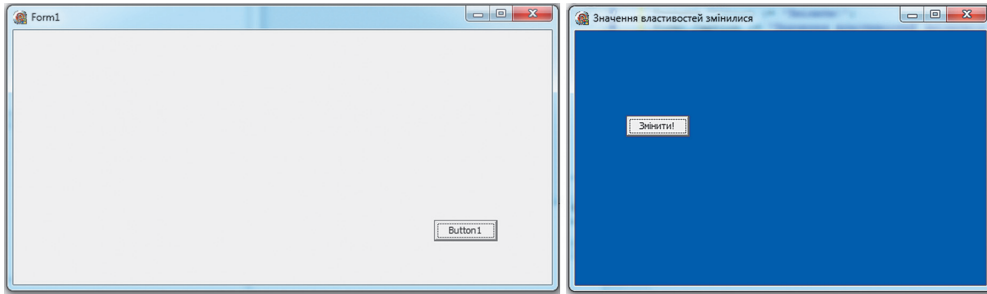


Рис. 2.22. Вікно виконання проекту до вибору кнопки **Button1** і після її вибору

тивостей, процедур, які починаються з цієї літери (рис. 2.23); з цього списку можна вибрати потрібне і цей текст буде вставлено в текст проекту;

- у ході написання тексту проекту середовище розробки відслідковує синтаксичні правила написання команд; якщо ці правила порушуються, то текст з помилками або текст одразу після нього підкреслюється червоною хвилястою лінією (рис. 2.23, підкреслюється літера **L**; рис. 2.24, слово **Label1** написано правильно, але після крапки не вказана властивість цього об'єкта, тому підкреслюється наступне слово); потрібно уважно слідкувати за цим і вчасно виправляти всі синтаксичні помилки;
- **Turbo Delphi 2006** не розрізняє великі та малі літери; але якщо слово утворено з кількох слів, прийнято (і зручно) писати першу літеру кожного слова великою, наприклад **TForm1.Button1Click**;
- після введення крапки, що розділяє ім'я об'єкта та ім'я властивості, автоматично відкривається список властивостей даного об'єкта (рис. 2.24); далі ім'я властивості можна не вводити з клавіатури, а знайти в списку та натиснути клавішу **Enter**; такий спосіб введення запобігає допущенню синтаксичних помилок, особливо тими, хто не знає англійської мови;
- після запуску проекту на виконання компілятор аналізує текст на наявність синтаксичних помилок; якщо такі помилки будуть знайдені, то після аналізу всіх помилок компіляція проекту переривається та червоним кольором виділяється фон першого з тих рядків тексту проекту, який містить синтаксичні помилки, або наступного за ним рядка (на рис. 2.25 неправильно написано слово **Label1**, тому червоним

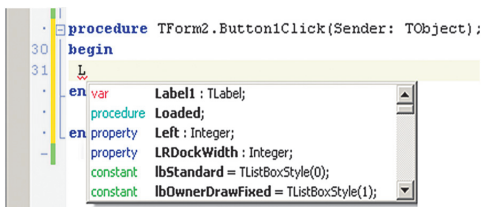


Рис. 2.23. Список, що відкривається після натиснення **Ctrl+Пропуск**

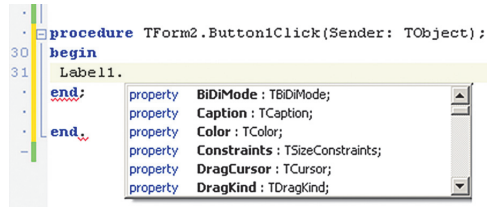


Рис. 2.24. Список властивостей об'єкта **Label1**, що відкривається після введення крапки


```

1 procedure TForm2.Button1Click(Sender: TObject);
2 begin
3   Label1.Color := clRed;
4   Edit1.Color := clRed;
5 end;
6
7 end.

```

Рис. 2.25. Позначення синтаксичної помилки під час уведення слова **Label1**

```

1 procedure TForm2.Button1Click(Sender: TObject);
2 begin
3   Label1.Color := clRed
4   Edit1.Color := clRed
5 end
6
7 end.

```

Рис. 2.26. Позначення синтаксичної помилки, коли пропущена крапка з комою в кінці команди

фоном позначено саме цей рядок, а на рис. 2.26 пропущено крапку з комою в кінці першої команди, у таких випадках червоним фоном позначається наступний рядок); усі інші помилки підкреслюються червоною хвилястою лінією; крім того, у вікні **Structure** з'являються повідомлення-підказки про вид помилки; потрібно виправити в тексті програми всі синтаксичні помилки і знову запустити проект на виконання.

Ми розглянули, як створити і використати процедуру **TForm1.Button1Click** – обробник події **OnClick**. Аналогічно можна створити інші процедури – обробники інших подій.

Уведення та виведення даних під час виконання проекту

Ми розглянули, як змінити значення властивостей елементів керування під час виконання проекту, використовуючи процедуру **TForm1.Button1Click** – обробник події **OnClick**. Але в розглянутому прикладі за кожного вибору кнопки виконуватимуться одні й ті самі команди процедури, а отже, кожного разу однаково змінюватимуться значення вказаних у командах процедури властивостей об'єктів. Якщо необхідно під час виконання проекту кожного разу по-іншому змінювати значення властивостей вказаних об'єктів, то потрібно значення цих властивостей якимось чином уводити в програму під час її виконання.

Для введення даних під час виконання проекту можна використати поля. Ви вже знаєте, що текст, який знаходиться в полі, є значенням його властивості **Text**. Скористаємося цим.

Створимо таку процедуру **TForm1.Button1Click** – обробник події **OnClick**, виконання якої дає змогу встановити довільний колір вікна та збільшити його висоту на довільну цілу кількість пікселів.

Для цього потрібно:

1. Розмістити на формі 2 поля – перше для введення значення кольору вікна, а друге для введення кількості пікселів.
2. Змінити значення властивості **Text** обох цих полів на порожнє.
3. Розмістити зліва від кожного поля напис, змінити значення властивості **Caption** першого на *Колір*, а другого на *Розмір*.
4. Розмістити на формі у правому нижньому куті кнопку, змінити значення її властивості **Caption** на *Змінити*.
5. Створити заготовку процедури **TForm1.Button1Click**, яка виконуватиметься після вибору кнопки.
6. Увести текст процедури **TForm1.Button1Click**:

```

procedure TForm1.Button1Click (Sender: TObject);
begin
    Form1.Color := Edit1.Text;
    Form1.Hight := Form1.Hight + StrToInt (Edit2.Text);
end;

```

Після запуску проекту відкриється вікно його виконання (рис. 2.27). У відповідні поля потрібно ввести потрібні значення кольору вікна, наприклад *clRed*, і кількості пікселів для збільшення висоти вікна, наприклад 300, після чого вибрати кнопку.

Після вибору кнопки настане подія **OnClick**, а отже, запуститься на виконання процедура **TForm1.Button1Click** – обробник події **OnClick**. При виконанні першої команди процедури текст, який буде введено в перше поле і стане значенням властивості **Text** цього поля, буде присвоєний властивості **Color** форми, у результаті чого колір форми стане заданим.

Незважаючи на те, що в друге поле буде введено число, програма сприйматиме його як текст, тобто не зможе виконувати з ним математичних операцій. Тому потрібно спочатку перевести цей текст у ціле число. Для цього використовується функція **StrToInt** (англ. *string to integer* – рядок у ціле). Після цього це число додається до поточного значення висоти вікна.

Таким чином, при кожному запуску проекту на виконання або перед кожним вибором кнопки можна вводити в поля різні значення, досягаючи тим самим установлення потрібного кольору форми і збільшення його висоти.

Написи можна використовувати не тільки для оформлення підписів об'єктів, а й для виведення результатів виконання проекту.

Створимо простий калькулятор, який буде додавати два довільні дійсні числа. Під час виконання проекту користувач вводитиме по одному довільному дійсному числу в кожне поле і вибиратиме кнопку із заголовком *Додати*. За вибору кнопки настане подія **OnClick**, в результаті чого виконуватиметься процедура **TForm1.Button1Click**, яка додаватиме ці числа і виводитиме суму в напис.

Для створення такого проекту потрібно (рис. 2.28):

1. Розмістити на формі два поля, напис для виведення результату, три написи для оформлення та кнопку.
2. Установити порожні значення властивості **Text** двох полів і властивості **Caption** напису для виведення суми.
3. Установити відповідні значення властивості **Caption** для інших трьох написів і кнопки.
4. Створити заготовку процедури **TForm1.Button1Click**, яка виконуватиметься після вибору кнопки.
5. Увести текст процедури **TForm1.Button1Click** (рис. 2.29).

Розглянемо призначення кожної команди наведеної процедури.

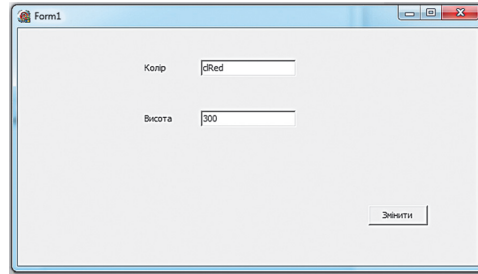


Рис. 2.27. Вікно виконання проекту для встановлення кольору вікна і збільшення його висоти

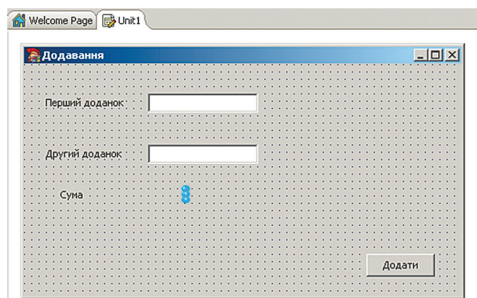


Рис. 2.28. Форма з розміщеними на ній елементами керування для додавання двох дійсних чисел

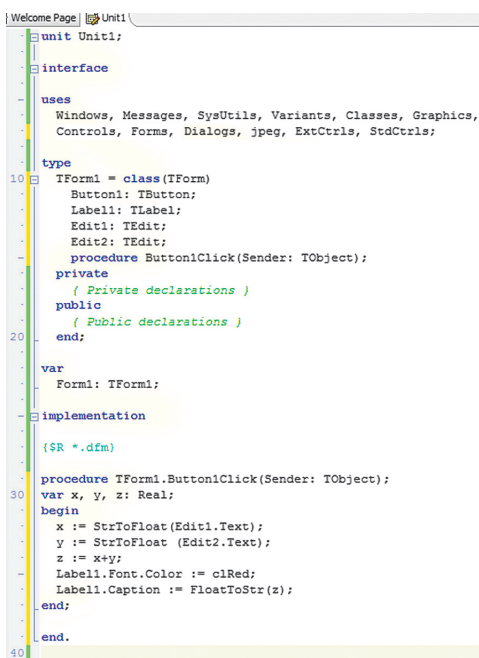


Рис. 2.29. Текст проекту на вкладці **Code**

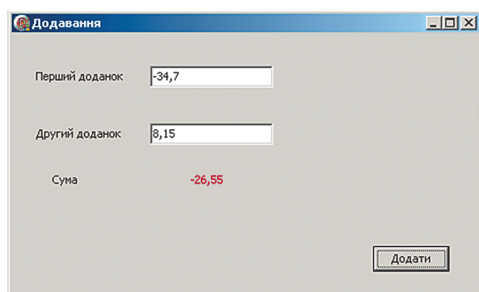


Рис. 2.30. Виконання проекту додавання двох дійсних чисел

Як ми вже зазначали вище, число, яке вводиться в поле, стає значенням його властивості **Text** і сприймається програмою як текст. Тому перші дві команди процедури є командами присвоювання, які призначені для переведення текстового представлення дійсних чисел у самі дійсні числа (для цього використана функція **StrToFloat** (англ. *string to float* – рядок у число з плаваючою десятковою крапкою, тобто у дійсне число) і присвоювання цих чисел змінним *x* та *y* відповідно.

Третя команда додає ці числа і присвоює результат змінній *z*.

Четверта команда встановлює червоним колір символів, яким виводитиметься результат у напис.

П'ята команда переводить результат додавання (значення змінної *z*) із числа в текст, використовуючи функцію **FloatToStr** (англ. *float to string* – число з плаваючою десятковою крапкою у рядок) і присвоює цей текст значенню властивості **Caption** напису з іменем **Label1** для виведення результату.

Під час виконання створеного проекту користувач уводить у поля числа (наприклад, -34,7 і 8,15), вибирає кнопку **Додати** і одержує результат -26,55 (рис. 2.30).

Не закриваючи проект, користувач може ввести в поля інші числа, знову вибрати кнопку **Додати** й одержати новий результат. І так можна повторювати довільну кількість разів. Це демонструє таку властивість алгоритму як масовість.

Іноді зручно виводити повідомлення не в написи на формі, а в спеціальне окреме вікно. Це можна здійснити, включивши до тексту програми команду **ShowMessage(<текст>)** (англ. *show message* – показати повідомлення). За цієї команди відкривається спеціальне вікно виведення повідом-

лення, в якому виводиться текст, указаний у команді. Якщо таку команду включити до процедури, що розглядалася вище, для виведення суми двох уведених дійсних чисел замість останньої команди виведення результату в напис або після цієї команди, то вона виглядатиме так:

ShowMessage('Сума '+FloatToStr(z));

Знак *плюс* у дужках означає з'єднання двох указаних текстів. Пропуск після слова **Сума** використовують для того, щоб у вікні відокремити число від тексту. Результат виконання цієї команди подано на рисунку 2.31.

При використанні вікна повідомлення для завершення виконання проекту спочатку слід закрити вікно повідомлення, а потім – вікно проекту.

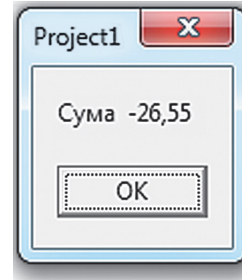


Рис. 2.31. Вікно виведення повідомлень

Змінні та їх властивості. Правила запису математичних виразів у Delphi

В останній розглянутій процедурі були використані три змінні: x , y , z . У мові програмування **Delphi** кожна змінна має ім'я, значення та тип.



Тип змінної визначає, які значення може набувати ця змінна, які операції над нею можна виконувати і ділянку пам'яті якого розміру програма виділить для зберігання значення цієї змінної.

Змінні x , y , z , які використали в останній процедурі, за умовою задачі можуть набувати довільні дійсні значення. Таким значенням відповідає тип **Real** (англ. *real* – дійсний). Змінні цього типу можуть набувати цілі й дробові значення від $-5 \cdot 10^{-324}$ до $1,7 \cdot 10^{308}$, для кожного значення змінних цього типу виділяється **8 байт** пам'яті.

Типи змінних, які використовуються в процедурі, вказуються після ключового слова **var** (англ. *variable* – змінна) у рядку, що знаходиться між рядком заголовка процедури і рядком зі словом **begin** (див. рис. 2.29). Цей рядок називається **рядком оголошення змінних**. Загальний вигляд рядка оголошення змінних такий:

var <список імен змінних одного типу, розділених комою>: <ім'я типу>;
<список імен змінних іншого типу, розділених комою>: <ім'я типу>; і т. д.

Якщо змінні набуватимуть тільки цілі значення, то для них доцільно використовувати тип **Integer** (англ. *integer* – цілий). Змінні цього типу можуть набувати цілі значення від $-2\,147\,483\,648$ (-2^{32}) до $2\,147\,483\,647$ ($2^{32} - 1$), для кожного значення змінних цього типу виділяється **4 байти** пам'яті.

Для перетворення текстового представлення цілого числа у саме ціле число використовується функція **StrToInt** (англ. *string to integer* – рядок у ціле число), а для зворотного перетворення – функція **IntToStr**.

Вище тексту складеної процедури знаходиться ще один рядок **var**. На рисунку 2.29 у цьому рядку вказано ім'я однієї змінної **Form1** типу **TForm**. Ця змінна відповідає формі, що використовується у цьому проєкті. У цьому рядку вказуються імена і типи тих змінних, які є доступ-

ними в будь-якій частині програми. Такі змінні називаються **глобальними**. На відміну від глобальних змінних, ті змінні, імена і типи яких описані в рядку заголовка процедури і в рядку **var** у тексті певної процедури, доступні тільки командам цієї процедури. Такі змінні називаються **локальними**. Про локальні та глобальні змінні поговоримо детальніше в наступних пунктах.

Об'єктам, які ми розташовуємо на формі, також відповідають змінні, імена яких збігаються з іменами відповідних об'єктів, наприклад **Button1**, **Edit1**. Кожна з цих змінних теж має свій тип, наприклад змінна **Button1** має тип **TButton**, а змінна **Edit1** – тип **TEdit**. Імена і типи цих змінних середовище розробки вносить до тексту проекту автоматично в розділ **type** (англ. *type* – тип) (рис. 2.29).

Змінна може отримати значення в результаті виконання команди присвоювання. Загальний вигляд команди присвоювання в **Delphi** такий: **<ім'я змінної> := <вираз>**. Вираз може складатися із значення (**константи**: число, текст в одинарних лапках та ін.), з імені змінної або містити кілька значень та імен змінних, поєднаних знаками арифметичних операцій, функції, дужки.

Позначення арифметичних дій і правила запису виразів у **Delphi** схожі з аналогічними правилами, які ви розглядали під час вивчення табличного процесора **Excel**. Але є й певні особливості:

- 1) під час введення в текст проекту десяткових дробів цілу та дробову частини потрібно розділяти крапкою;
- 2) під час введення при виконанні проекту в поле десяткового дробу розділювач цілої та дробової частин залежить від налаштувань **Windows** (в українській версії **Windows XP** за замовчуванням встановлена кома);
- 3) ні в тексті проекту, ні під час введення в поле не можна використовувати звичайні дробі;
- 4) у **Delphi** немає операції піднесення до степеня; для піднесення до степеня використовується спеціальна функція, яку ми розглянемо пізніше, а поки що степінь з натуральним показником можна замінювати добутком;
- 5) пріоритет операцій в усіх випадках визначається правилами математики.



Перевірте себе

- 1°. Що таке обробник події? Як пов'язати з певною подією її обробник?
- 2°. Що таке процедура? Опишіть структуру процедури – обробника події.
- 3°. Наведіть загальний вигляд команди, якою можна змінити значення властивості елемента керування. Наведіть приклади таких команд.
- 4°. Як позначаються синтаксичні помилки в середовищі розробки? Наведіть приклади.
- 5°. Які елементи керування можна використати для введення та виведення даних?
- 6°. Як вивести повідомлення в окреме спеціальне вікно?
- 7°. Для чого призначені функції **StrToFloat**, **FloatToStr**, **StrToInt**, **IntToStr**?
- 8°. Що визначає тип змінної? Які типи змінних ви знаєте?
- 9°. Які значення можуть набувати змінні типу **Real** і типу **Integer**? Який розмір ділянки пам'яті виділяється для зберігання змінних кожного з цих типів?

- 10°. Як описуються типи змінних, які використовуються в процедурі; в усій програмі?
- 11°. Які змінні називаються локальними, а які – глобальними?
- 12°. Який загальний вигляд команди присвоювання в **Delphi**?



Виконайте завдання

- 1°. Запишіть команди присвоювання змінній *s* значень наведених виразів:

а) $3x^2 - 5y^3 + 7z$; г) $\frac{34 - 5x^3 + yz}{-24 - 4,76y^2}$;

б) $(x + y + z) : 3$; д) $\frac{2x^2 - 5}{3 - 4y^2} - 7zy$;

в) $\frac{7x - 12y^3}{45 - 2z^2}$; е) $7,23xyz - \frac{7x^2 - 15y}{4z - 8,23y^3} - x$.

- 2°. Відкрийте новий проект. Змініть заголовок форми на *Я зміню значення властивостей форми*. Розмістіть на формі кнопку. Змініть заголовки кнопки *Змінити*. Змініть ім'я форми на **MyForm1**. Створіть для події **OnClick** кнопки заготовку обробника цієї події. Уведіть до процедури команди, які змінять колір вікна на **clPurple**, зменшать ширину вікна на 100 пікселів і збільшать його висоту на 200 пікселів. Виконайте проект. Виберіть кнопку. Переконайтеся, що змінилися розміри і колір вікна. Створіть у власній папці папку **Проект 2.5.2** і збережіть у ній проект.

- 3°. Створіть проект для множення двох дійсних чисел. Установіть жовтий колір фону і синій колір тексту для виведення результату. Передбачте виведення результату, ще й у спеціальне вікно для повідомлень. Виконайте проект кілька разів для різних чисел, стежте за правильністю результату. Створіть у власній папці папку **Проект 2.5.3** і збережіть у ній проект.



- 4°. Відкрийте новий проект. Установіть заголовок форми *Доступний/Недоступний*. Розмістіть на формі дві кнопки. Для першої встановіть такі значення властивостей: **Caption** – *Кнопка 1*, **Enabled** – *True*. Для другої відповідно: **Caption** – *Кнопка 2*, **Enabled** – *False*. Створіть для події **OnClick** першої кнопки заготовку обробника цієї події та введіть до нього такі команди:

```
Button2.Enabled := True;
Button1.Enabled := False;
```

Створіть для події **OnClick** другої кнопки заготовку обробника цієї події та введіть до нього такі команди:

```
Button1.Enabled := True;
Button2.Enabled := False;
```

Проаналізуйте тексти цих процедур і зробіть висновок, що відбуватиметься під час виконання проекту. Виконайте проект. По черзі вибирайте доступну кнопку та спостерігайте, що відбуватиметься. Створіть у власній папці папку **Проект 2.5.4** і збережіть у ній проект.



- 5°. Створіть проект, у якому подія **OnMouseMove** кнопки викличе зміну її розмірів, заголовка і положення у вікні. Створіть у власній папці папку **Проект 2.5.5** і збережіть у ній проект.
- 6°. Створіть проект, у якому подія **OnClick** поля викличе зміну його розмірів, кольору його фону і появу в полі тексту з вашим прізвищем та іменем символами червоного кольору. Створіть у власній папці папку **Проект 2.5.6** і збережіть у ній проект.

- 7*. Створіть проект, який за довжиною, шириною і висотою вашої кімнати обчислюватиме площу підлоги і площу поверхні стін. Створіть у власній папці папку **Проект 2.5.7** і збережіть у ній проект.
- 8*. Створіть проект для знаходження суми, різниці та добутку двох довільних дійсних чисел. Створіть у власній папці папку **Проект 2.5.8** і збережіть у ній проект.
- 9*. Створіть проект, у якому розташуйте на формі три кнопки. Установіть такі значення їхніх властивостей, щоб на початку виконання проекту перша кнопка була видима і доступна, друга – видима і недоступна, третя – невидима і недоступна. Для кожної кнопки створіть процедуру для події **OnClick**, щоб:
- 1) вибір першої кнопки робив її невидимою і недоступною, другу кнопку – видимою і доступною, третю кнопку – видимою і недоступною;
 - 2) вибір другої кнопки робив її невидимою і недоступною, третю кнопку – видимою і доступною, першу кнопку – видимою і недоступною;
 - 3) вибір третьої кнопки робив її невидимою і недоступною, першу кнопку – видимою і доступною, другу кнопку – видимою і недоступною.
- Створіть у власній папці папку **Проект 2.5.9** і збережіть у ній проект.



Практична робота № 6. Введення та виведення даних. Робота зі змінними

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте середовище **Turbo Delphi 2006** і створіть новий проект.
2. Установіть для форми колір фону *жовтий*, розміри *1000 на 600 пікселів*, текст заголовка *Практична 6*.
3. Розмістіть на формі дві кнопки, три поля та два написи.
4. Задайте на першій кнопці текст *Переміщення*, на другій – *Обчислення*, у полів і написів – *порожній текст*.
5. Розмістіть на формі біля кожного поля та кожного напису додаткові написи з текстами, що будуть пояснювати їхні призначення.
6. Складіть обробник події **OnClick** першої кнопки, виконання якого приведе до переміщення форми на *50 пікселів управо* та на *30 пікселів униз*, кожного поля – на *20 пікселів уліво* та на *30 пікселів уверх* і зробить їхній фон *червоним*.
7. Виберіть першу кнопку та переконайтеся у правильності результатів.
8. Складіть обробник події **OnClick** другої кнопки, виконання якого приведе до знаходження об'єма та площі повної поверхні прямокутного паралелепіпеда, вважаючи, що у поля будуть введені довжини його ребер.
9. Виберіть другу кнопку та переконайтеся у правильності результатів.
10. Додайте до другого обробника події команду виведення повідомлення про об'єм паралелепіпеда у вікно повідомлень. Повторно виберіть другу кнопку і переконайтеся у правильності результатів.
11. Створіть у власній папці папку **Практична 6** і збережіть у ній проект.

2.6. Налаштування програм



1. Яке призначення компіляторів? Які особливості їхньої роботи?
2. Які помилки у програмному коді виявляються на етапі компіляції? У чому вони полягають? Як вони позначаються в **Turbo Delphi 2006**? Де відображається їх опис?
3. Які помилки називаються логічними? Як можна виявити наявні логічні помилки?
4. Як відкрити раніше створений проект?

Тестування програми. Коментарі

Ви вже знаєте, що під час складання проекту можуть виникати синтаксичні та логічні помилки. Синтаксичні помилки, які не помітив автор проекту, знаходить компілятор на етапі компіляції під час запуску проекту на виконання. Потрібно виправити всі такі помилки і знову запустити на виконання нову версію проекту.

Логічні помилки в проекті виявити значно важче, ніж синтаксичні. Адже проект виконується, видає результати, але вони не відповідають умові задачі, для розв'язування якої створювався даний проект.

Під час розгляду етапів розв'язування задач з використанням комп'ютера вже пояснювалося, що одним зі способів виявлення логічної помилки є створення тестових наборів вхідних даних з наперед відомими результатами і виконання створеного проекту для цих наборів даних. Після чого потрібно порівняти одержані результати з очікуваними і знайти команди в програмному коді, виконання яких є причиною появи логічних помилок. Процес пошуку логічних помилок у тексті програми з використанням тестових наборів початкових даних називають **тестуванням програми**.

Пояснимо цей спосіб на такому прикладі. Нехай потрібно скласти проект для обчислення значень функції $y = \frac{2x - 5}{x^2 + 1}$. Часто учні, записуючи команду присвоєння для обчислення значення відповідного виразу $y := (2*x - 5)/(x*x + 1)$, забувають узяти знаменник (а то й чисельник) у дужки, тобто вводять, наприклад, команду $y := (2*x - 5)/x*x + 1$ або $y := 2*x - 5/x*x + 1$. Таку логічну помилку легко помітити, застосувавши вдалу добірку тестів. Так, під час тестування для $x = 0$ стане очевидним, що проект не видає очікуваного правильного результату -5 , а перериває своє виконання, відкриваючи вікно з повідомленням **division by zero** (англ. *division by zero* – ділення на нуль). Після цього потрібно знайти команду, яка містить дію ділення та може стати причиною такої помилки.

Звертаємо вашу увагу, що для нескладних програм, які ви поки що вчитеся складати, підібрати систему тестових прикладів не дуже важко. Ця справа ускладнюється, якщо потрібно протестувати більшу за кількістю команд і складнішу за своєю логічною структурою програму, а тим більше, професійні проекти, які містять сотні, а то й тисячі різноманітних команд.

Допомагають у пошуку логічних помилок **коментарі**, які можна включати до тексту програми.



Коментар – це текст, який ігнорується компілятором і включається до тексту програми з метою полегшення розуміння структури програмного коду.

Щоб компілятор відрізняв команди від коментарів, у **Turbo Delphi 2006** коментарі потрібно брати у фігурні дужки, наприклад **{Це команда знаходження частки}**. Якщо текст коментаря короткий і не займає більше ніж один рядок, то можна замість дужок поставити перед ним дві похилі риски, наприклад **// Це команда знаходження частки**. Інколи коментарі замість фігурних дужок беруть у круглі дужки із зірочками, наприклад **(*Це команда знаходження частки*)**.

Якщо коментар займає не більше ніж один рядок, то він може знаходитися після команди, а може розпочинатися з початку рядка. Будь-який коментар не може знаходитися всередині команди програми.

На рисунку 2.32 наведені приклади використання коментарів у тексті програми.

```

30 procedure TForm1.Button1Click(Sender: TObject);
   var x, y, z: Real;
   begin
       // Команди введення
       x := StrToFloat(Edit1.Text);
       y := StrToFloat(Edit2.Text);
       z := x+y;      // Команда обчислення результату
       // Команди для виведення
       Label1.Font.Color := clRed;
       Label1.Caption := FloatToStr(z);
   40   ( Далі йде команда виведення результату в
       спеціальне вікно повідомлень)
       ShowMessage('Сума ' + FloatToStr(z));
   43   end;
   end.

```

Рис. 2.32. Текст програми з коментарями

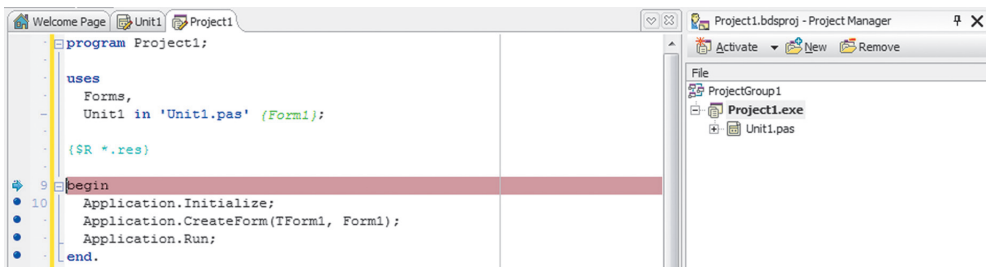
Якщо текст програми великий, то для пошуку місця логічної помилки інколи зручно тестувати програму не повністю, а частинами. Якщо правильність роботи деяких частин програми не викликає сумнівів і виконання цих частин не є необхідним для виконання інших, то ці частини програми закоментовують, тобто беруть у фігурні дужки. Компілятор пропускає при компіляції закоментовані фрагменти програми і компілює тільки всі інші фрагменти. Це дає змогу користувачу зосередити свою увагу на пошуку логічних помилок у тих частинах, де вони можливі, а також зекономити час на тестуванні програми.

Покрокове виконання програм

Ще одним способом виявлення в програмі логічних помилок є її покрокове виконання, або **трасування**. **Крок виконання програми** – це **рядок програми**. У рядку можуть бути записані одна або кілька команд програми. Підпрограму, як окрему частину програми, можна виконати як один крок, а можна виконати по рядках – це залежить від режиму покрокового виконання.

Для виконання одного кроку програми, якщо необхідно і підпрограму виконати по рядках, потрібно натиснути клавішу **F7** або виконати **Run ⇒ Trace Into** (англ. *trace into* – трасувати всередину). А для виконання одного кроку програми, якщо можна підпрограму виконати повністю за один крок, потрібно натиснути клавішу **F8** або виконати **Run ⇒ Step Over** (англ. *step over* – крок поверх). Ще раз наголосимо, що підпрограму можна виконати як один крок тільки в тому випадку, коли ми впевнені, що вона не містить логічних помилок.

Розглянемо, як відбувається покрокове виконання програми на прикладі програми знаходження суми двох довільних дійсних чисел, розглянутої в попередньому підпункті (див. рис. 2.32).

Рис. 2.33. Текст програми, що увійшов до файлу **Project1.dpr**

Після першого натиснення клавіші **F7** відбувається компіляція проекту, і якщо проект не містить синтаксичних помилок, то починається його виконання. Спочатку відкривається вкладка з текстом програми, який увійшов до файлу з розширенням імені **dpr** (англ. *delphi project* – проект Делфі). Для проекту, що розглядається, це файл **Project1.dpr**. На рисунку 2.33 наведено команди цієї частини програми та її початок підсвічується кольоровою смужкою. За кожного наступного натиснення клавіші **F7** виконуватимуться команди наступного рядка. У результаті виконання наведених на рисунку 2.33 трьох команд відкриється вікно проекту з розміщеними на ньому елементами керування (якщо вікно не відкрилося, то потрібно вибрати його кнопку, яка з'явилася на **Панелі завдань**).

Після цього потрібно увести в поля відповідні числа і вибрати кнопку **Додати**. У результаті відкривається вкладка з текстом програми, який увійшов до файлу з розширенням імені **pas** (англ. *pascal* – Паскаль). Для проекту, що розглядається, це файл **Unit1.pas**. На рисунку 2.34 наведено команди цієї частини програми, а її початок підсвічується кольоровою смужкою. За кожного наступного натиснення клавіші **F7** виконуватимуться команди наступного рядка. У результаті послідовного натискання клавіші **F7** виконуються усі наведені команди, у результаті чого в напис і у вікно повідомлень виведеться результат. Зрозуміло, що всі коментарі при виконанні будуть пропущені.

Для того щоб перервати режим покрокового виконання, потрібно натиснути сполучення клавіш **Ctrl+F2**.

Ясно, що розглянуте вище покрокове виконання програми лише продемонструвало сам процес і нічого нам не дало для з'ясування того місця програми, яке призводить до логічної помилки, якщо така існує. Адже ми не бачили, як змінюються значення змінних у ході такого виконання. А саме це може нам підказати, які команди є причиною логічної помилки.

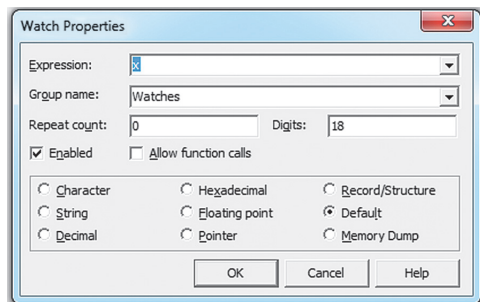
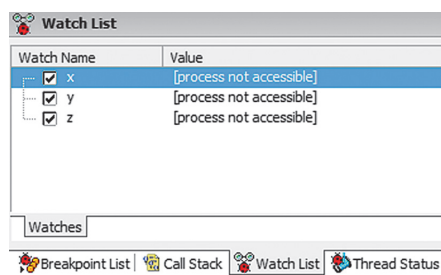
Щоб під час покрокового виконання програми слідкувати за зміненням значень змінних, потрібно відкрити вікно **Watch List** (англ. *watch list* – список перегляду) і

```

30 procedure TForm1.Button1Click(Sender: TObject);
31   var x, y, z: Real;
32   begin
33     // Команди введення
34     x := StrToFloat(Edit1.Text);
35     y := StrToFloat(Edit2.Text);
36     z := x*y; // Команда обчислення результату
37     // Команди для виведення
38     Label1.Font.Color := clRed;
39     Label1.Caption := FloatToStr(z);
40     ( Далі йде команда виведення результату в
      спеціальне вікно повідомлень )
41     ShowMessage('Сума ' + FloatToStr(z));
42   end;
43 end.

```

Рис. 2.34. Текст процедури, що увійшов до файлу **Unit1.pas**

Рис. 2.35. Вікно **Watch Properties**Рис. 2.36. Вікно **Watch List**

помістити в це вікно імена змінних, значення яких ми хочемо спостерігати в ході покрокового виконання програми.

Для того щоб помістити ім'я змінної до вікна **Watch List**, потрібно натиснути сполучення клавіш **Ctrl+F5** або виконати **Run** \Rightarrow **Add Watch** (англ. *add watch* – додати перегляд). У результаті спочатку відкривається вікно **Watch Properties** (англ. *watch properties* – властивості перегляду, рис. 2.35). У перше поле цього вікна потрібно ввести ім'я змінної та вибрати кнопку **OK**.

Після цього відкривається вікно **Watch List** (рис. 2.36) із внесеним до нього іменем змінної. Таким чином можна до цього вікна внести імена всіх тих змінних, за значеннями яких доцільно спостерігати під час покрокового виконання програми.

Під час покрокового виконання програми значення цих змінних змінюватимуться і будуть відображатися у вікні **Watch List**. Уважно слідкуючи за цими зміненнями і порівнюючи ці значення з очікуваними, можна помітити, в який момент виконання програми значення певної змінної стає неочікуваним. Саме ця команда й може бути причиною логічної помилки.



Перевірте себе

- 1°. У чому полягає процес тестування програми?
- 2°. Поясніть, для чого виконується тестування програми.
- 3°. Що таке коментар у програмі? Яким чином можна коментар включити до тексту програми?
- 4°. Поясніть, для чого в програму включають коментарі.
- 5°. Для чого програми виконують покроково?
- 6°. Поясніть, у чому полягає процес покрокового виконання програми.
- 7°. Прокоментуйте хід покрокового виконання програми.
- 8°. Чим відрізняється покрокове виконання програм з використанням клавіш **F7** і **F8**?
- 9°. Як переглянути значення змінних під час покрокового виконання програми?



Виконайте завдання

- 1°. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.1**. Завданням проекту є знаходження розв'язку задачі: *Два потяги виїхали одночасно назустріч один одному. Знайти, через який час вони зустрінуться, якщо відомі відстань між ними у момент початку руху*

та швидкість кожного потяга. Виконання обробника події **OnClick** кнопки повинно привести до виведення шуканого часу, якщо значення відстані та швидкостей вводяться у поля. Для заданих початкових даних обчисліть очікувані результати:

- а) відстань – 1000, швидкості – 120 та 130;
- б) відстань – 600, швидкості – 80 та 120;
- в) відстань – 1200, швидкості – 115 та 85.

Перевірте, чи збігаються результати, отримані під час виконання проекту, з обчисленими вами результатами. Поясніть, у чому може бути помилка.



2*. Створіть три тестові набори вхідних даних для тестування проекту, призначеного для розв'язування задачі: *Два потяги вийшли одночасно з однієї станції у протилежних напрямках. Знайти відстань, на якій вони будуть один від одного через заданий час, якщо відома швидкість кожного потяга.* Обчисліть очікувані результати за складеними тестами. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.2**. Протестуйте його на ваших тестових даних. Зробіть висновок, чи містить програма логічні помилки. Якщо так, то у чому може бути помилка?

3*. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.3**. Перегляньте програмний код і на основі аналізу коментарів зробіть припущення, яку задачу розв'язує цей проект. Спираючись на ваше припущення, складіть три тестові набори вхідних даних для тестування проекту. Обчисліть очікувані результати. Протестуйте проект на ваших тестових даних. Зробіть висновок, чи правильним було ваше припущення.

4*. Створіть проект для обчислення довжини кола та площі круга за значенням радіуса, що вводиться у поле користувач. Поставте коментарі для позначення рядків, у яких відбувається обчислення шуканих значень. Створіть у власній папці папку **Проект 2.6.4** і збережіть у ній проект.

5*. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.5**. Виконайте програму в покроковому режимі. Зверніть увагу на відображення ходу виконання програми.



6*. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.6**. У ході покрокового виконання програми прослідкуйте за значеннями змінних **S1** та **S2**.

7*. Учні для свята купили цукерки та печиво. Відомі ціна цукерок і печива та кількість куплених солодощів кожного виду. Створіть проект для знаходження повної вартості покупки. Створіть у власній папці папку **Проект 2.6.7** і збережіть у ній проект. Виконайте програму в покроковому режимі, прослідкуйте за значеннями змінних, що містять значення вартості кожного товару.

8*. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.8**. У ході покрокового виконання програми прослідкуйте за значеннями змінних x , a та S . З'ясуйте, як значення, які послідовно набуває змінна a , пов'язані зі значенням змінної x . Змініть проект так, щоб у результаті виконання змінна S містила суму $x + x^2 + x^4$ для введеного значення x . Додайте до тексту програми коментар, у якому вкажіть, які зміни були внесені у програмний код. Створіть у власній папці папку **Проект 2.6.8** і збережіть у ній змінений проект.

9*. Відкрийте проект, наприклад з папки **Тема 2\Завдання 2.6\Проект 2.6.9**. Цей проект призначений для знаходження, на скільки більше часу затратив один з потягів на шлях між містами, ніж другий, за даними відстанню між містами і швидкостями кожного з потягів. З'ясуйте, чи містить проект логічні помилки. Якщо так, визначте, які команди є їх причинами, і виправте їх. Запишіть набори тестових даних, на яких ви виявили логічні помилки. Створіть у власній папці папку **Проект 2.6.9** і збережіть у ній змінений проект.



Практична робота № 7. Налаштування програм

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте **Turbo Delphi 2006**.
2. Створіть проект для визначення, у скільки разів добуток двох заданих цілих чисел більший за суму цих чисел.
3. Створіть три тестові набори вхідних даних для тестування проекту. Обчисліть очікувані результати. Протестуйте проект на ваших тестових даних.
4. Поставте коментарі для позначення рядків, у яких відбувається обчислення суми та добутку.
5. Виконайте проект у покроковому режимі. У ході покрокового виконання проекту прослідкуйте за значеннями суми та добутку. Запишіть, які повідомлення відображаються у стовпці **Value** для вказаних змінних перед початком покрокового виконання проекту, в ході виконання для одного з наборів початкових даних і після завершення виконання.
6. Створіть у власній папці папку **Практична 7** і збережіть у ній проект.

2.7. Основні поняття математичної логіки



1. Що таке логічний вираз? Які значення він може набувати?
2. Опишіть загальний вигляд, кількість аргументів і результат функцій **AND**, **OR**, **NOT** табличного процесора.
3. Які властивості змінної ви знаєте?
4. Що визначає тип змінної? Які вам відомі числові типи?

У попередніх пунктах ми розглядали проекти, які реалізовували лінійні алгоритми. Але для розв'язування більшості задач лінійних алгоритмів недостатньо. У проектах, що реалізують інші види алгоритмів, наприклад алгоритми з розгалуженнями, використовують логічні вирази. Розглянемо логічні вирази і операції над ними.

Висловлення. Логічні константи. Логічні операції

Логіка (грец. λόγος – слово, смисл, думка, мова) – наука про форми, методи і закони інтелектуальної пізнавальної діяльності, правильного мислення, про способи міркування. Логіка як наука вивчає способи отримання правильних висновків з висновків, отриманих раніше.

Один з розділів логіки – **математична логіка** – є наукою про закони математичного мислення. До основних понять математичної логіки входить поняття **висловлення**. Під висловленням розуміють розповідне речення, про яке можна однозначно сказати, правильне (істинне) воно чи неправильне (хибне). Висловлення позначають великими літерами англійського алфавіту, наприклад $A = \text{«Київ – столиця України»}$, $B = \text{«}2 + 2 = 5\text{»}$.

Основною властивістю висловлення є його **істинність**, інші властивості вважаються несуттєвими. Значення істинності висловлення позначають **1** або **true**, якщо висловлення істинне, **0** або **false**, якщо висловлення хибне. Наприклад, висловлення $A = \text{«Київ – столиця України»}$ є **істинним**, тобто значення його властивості **істинність** дорівнює **true**. Висловлення $C = \text{«}3 > 5\text{»}$ є **хибним**, тобто значення його властивості **істинність** дорівнює **false**.



Значення true та false називаються логічними константами.

Із заданих висловлень можна отримати нові – складені, – використовуючи логічні операції: заперечення, кон'юнкцію (лат. *conjunctio* – об'єднання), диз'юнкцію (лат. *disjunctio* – роз'єднання) та ін.



Запереченням називається операція утворення з висловлення A такого складеного висловлення \bar{A} (читається «не A »), яке істинне тоді, коли A хибне, і хибне тоді, коли A істинне.

Використанню операції заперечення в українській мові відповідає вживання частки «не» перед дієсловом. Наприклад, запереченням висловлення A = «Існує найбільше просте число» є висловлення \bar{A} = «Не існує найбільшого простого числа».

Висловлення та його заперечення не можуть бути одночасно істинними або одночасно хибними.

Подібно до того, як у математиці використовують таблиці додавання та множення для визначення результатів виконання цих арифметичних операцій, у математичній логіці використовують **таблиці істинності**. У таблиці істинності для кожного можливого набору значень властивості **істинності** висловлень наводиться відповідне значення цієї властивості для висловлення, яке є результатом виконання логічної операції над ними.

За означенням, заперечення виконується над одним висловленням, яке може бути або істинним, або хибним. Тому таблиця істинності операції заперечення має такий вигляд:

A	\bar{A}
0	1
1	0



Кон'юнкцією двох висловлень A та B називається операція утворення такого складеного висловлення $A \wedge B$ (читається « A і B »), яке істинне тоді й тільки тоді, коли обидва висловлення A і B істинні.

Використанню операції кон'юнкція в українській мові відповідає вживання сполучника «і». Наприклад, висловлення B = «Число 27 кратне 3 і число 27 кратне 9» є кон'юнкцією двох висловлень: «Число 27 кратне 3» і «Число 27 кратне 9».

За означенням, кон'юнкція виконується над двома висловленнями. Кожне з них може бути або істинним, або хибним. Тому таблиця істинності операції кон'юнкція має такий вигляд:

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Операцію кон'юнкція також називають **логічним множенням**.



Диз'юнкцією двох висловлень A та B називається операція утворення такого складеного висловлення $A \vee B$ (читається « A або B »), яке істинне тоді й тільки тоді, коли істинне хоча б одне з висловлень A або B .

Використанню операції диз'юнкція в українській мові відповідає вживання сполучника «або». Наприклад, висловлення $C = «21 \leq 21»$ є диз'юнкцією двох висловлень: « $21 = 21$ » або « $21 < 21$ ».

За означенням, диз'юнкція виконується над двома висловленнями. Кожне з них може бути або істинним, або хибним. Тому таблиця істинності операції диз'юнкція має такий вигляд:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Операцію диз'юнкція також називають **логічним додаванням**.

Під час розв'язування багатьох задач трапляються висловлення зі змінними, в яких використовуються знаки порівняння: $>$ (більше), $<$ (менше), $=$ (дорівнює), $<>$ (не дорівнює), $>=$ (більше або дорівнює), $<=$ (менше або дорівнює), наприклад $x > 5$, $y < z$. Такі висловлення можуть бути істинними при одних значеннях змінних і хибними при інших.

Висловлення $x < 0$ є простим, а висловлення $3 < x < 5$ – складеним. Останнє можна представити як кон'юнкцію двох простих висловлень: $(x > 3) \wedge (x < 5)$.

Наприклад, щоб визначити, чи лежить точка з координатами $(x; y)$ у першій координатній чверті, потрібно визначити істинність складеного висловлення $(x > 0)$ і $(y > 0)$. Для точки з координатами $(3; -5)$ матимемо $(3 > 0)$ і $(-5 > 0)$. Оскільки перше твердження істинне, а друге – хибне, то їхня кон'юнкція хибна. Тобто ця точка не лежить у першій чверті.

Логічні змінні. Логічні вирази та їхні таблиці істинності. Логічні формули

Алгеброю логіки (булевою логікою, алгеброю висловлень) називають розділ математичної логіки, в якому розглядаються загальні властивості виразів, складених з висловлень з використанням логічних операцій.



Основні положення алгебри логіки були сформульовані англійським математиком і філософом **Джорджем Булем** (1815–1864) (рис. 2.37). У 1854 році вийшла його основна праця «Дослідження законів думки, на яких засновані математичні теорії логіки й імовірності». Ця книжка нині зраховується до математичної класики. У ній досліджується система алгебри, яку сьогодні називають «алгеброю висловлень» або «булевою логікою». Булева логіка стала основним математичним інструментом для створення комп'ютерів.

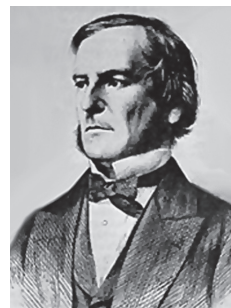


Рис. 2.37. Джордж Буль

В алгебрі логіки використовується поняття **логічна змінна**.



Логічна змінна – це змінна, яка може набувати лише значень **true** або **false**.

Завданням алгебри логіки є визначення істинності **логічних виразів** – виразів, що складаються з логічних констант, логічних змінних, логічних операцій, дужок, наприклад $(\bar{A} \vee B) \wedge (A \vee B)$.

Зрозуміло, що значенням логічного виразу може бути лише **true** або **false**.

Для логічних операцій, розглянутих вище, визначено такий **пріоритет операцій**: заперечення, кон'юнкція, диз'юнкція. Для змінення цього порядку виконання логічних операцій використовують дужки.

Для обчислення значення істинності логічного виразу можна використати таблиці істинності.

Для складання та заповнення таблиці істинності потрібно:

1. Обчислити кількість можливих наборів значень логічних змінних. Якщо формула містить n різних логічних змінних, то можливих наборів значень цих змінних буде 2^n . Це число визначає кількість рядків у таблиці істинності.
2. Обчислити кількість логічних операцій у логічному виразі. Ця кількість плюс кількість логічних змінних визначає кількість стовпців таблиці.
3. Заповнити перші n стовпців усіма можливими наборами значень логічних змінних.
4. Заповнити кожний наступний стовпець значеннями, отриманими під час виконання чергової логічної операції. Черговість встановлюється згідно з названим пріоритетом операцій.

В останньому стовпці таблиці будуть отримані всі можливі значення істинності заданого логічного виразу.

Наприклад, вираз $\bar{A} \vee B \wedge A$ містить дві логічні змінні, тобто $n = 2$. Отже, усього існує 4 набори можливих значень цих змінних ($2^n = 2^2 = 4$). Вираз містить 3 логічні операції: заперечення, диз'юнкція та кон'юнкція. Таким чином, таблиця істинності складатиметься із 4 рядків і 5 стовпців. Першою, згідно з пріоритетом, виконуватиметься операція заперечення, другою – кон'юнкція, останньою – диз'юнкція, в якій потрібно використовувати результати перших двох операцій.

Отримаємо таку таблицю істинності:

A	B	1	2	3
		\bar{A}	$B \wedge A$	$\bar{A} \vee (B \wedge A)$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	0
1	1	0	1	1

Аналізуючи останній стовпець таблиці істинності, робимо висновок, що вираз матиме значення **false** лише у випадку, коли логічні змінні мають такі значення: $A = \text{true}$, $B = \text{false}$. У всіх інших випадках значенням логічного виразу буде **true**.



Два логічні вирази називаються *рівносильними*, якщо вони набувають однакових значень при одних і тих самих наборах значень логічних змінних, що входять до цих виразів.

Рівносильність двох логічних виразів утворює *логічну формулу*.

Наведемо кілька цікавих і корисних логічних формул:

1. $\overline{\overline{A}} = A$ – формула подвійного заперечення;
2. $A \wedge (A \vee B) = A$ – формула поглинання;
3. $\overline{A \vee B} = \overline{A} \wedge \overline{B}$ – формула заперечення диз'юнкції (закон де Моргана).

Для доведення цих рівносильностей можна скласти і порівняти таблиці істинності логічних виразів у правій і лівій частинах. Пропонуємо вам зробити це самостійно.



Август де Морган (1806–1871) (рис. 2.38) – шотландський математик і логік, професор математики Лондонського університетського коледжу, перший президент Лондонського математичного товариства. Результати своїх досліджень з логіки одержав незалежно від Джорджа Буля та виклав у 1847 році.



Рис. 2.38. Август де Морган



Перевірте себе

- 1°. Що вивчає наука логіка? Що вивчає математична логіка?
- 2°. Що таке висловлення? Наведіть приклади істинних і хибних висловлень.
- 3*. Які речення не є висловленнями?
- 4°. Що називається логічною константою?
- 5°. Як утворюються складені висловлення? Наведіть приклади.
- 6°. Що таке заперечення? Наведіть приклади.
- 7°. Для чого використовують таблиці істинності?
- 8°. Наведіть таблицю істинності заперечення.
- 9°. Що таке кон'юнкція? Наведіть приклади. Наведіть таблицю істинності.
- 10°. Що таке диз'юнкція? Наведіть приклади. Наведіть таблицю істинності.
- 11°. Яка змінна називається логічною?
- 12*. Чим логічна змінна відрізняється від логічного висловлення?
- 13°. З чого складаються логічні вирази? Які значення вони можуть набувати?
- 14°. Назвіть пріоритет логічних операцій.
- 15°. Наведіть алгоритм складання та заповнення таблиці істинності логічного виразу.
- 16°. Які логічні вирази називаються рівносильними? Що таке логічна формула?



Виконайте завдання

- 1°. Побудуйте заперечення висловлень і з'ясуйте їх істинність:
 - а) Число 3 є дільником числа 545.
 - б) Автомобіль не має права їхати на червоне світло.
 - в) Існують паралелограми з прямими кутами.
 - г) Рівняння $2x^2 - 3x + 1 = 0$ має цілий корінь.
 - д) Не існує натурального числа, що ділиться на 2.
 - е) Існує ціле число, яке ділиться на всі цілі числа.

- 2*. Серед наведених складених висловлень знайдіть кон'юнкції та диз'юнкції та визначте їх істинність:
- Число 27 кратне 3 і кратне 9.
 - $17 < 42 < 18$.
 - Число 2 просте або парне.
 - $\triangle ABC$ є гострокутним, прямокутним або тупокутним.
 - Діагоналі будь-якого паралелограма перпендикулярні та точкою перетину поділяються навпіл.
 - $7^2 = 49$ і $(-7)^2 = 49$.
 - Якщо трикутник рівнобедрений, то він рівносторонній.
 - $21 \leq 21$.
 - $21 \leq 18$.

Кон'юнкція	
Диз'юнкція	
Ні кон'юнкція, ні диз'юнкція	

- 3*. Визначте істинність або хибність складених висловлень для наведених значень змінних:

- $x > 0$ і $y > 0$ або $x < 0$ і $y < 0$;
- $x > 0$ і не $y < 0$ або $x < 0$ і $y > 0$.

Значення змінних:

- $x = 5, y = 8$;
- $x = 5, y = -8$;
- $x = -5, y = 8$;
- $x = -5, y = -8$.



- 4*. Нехай $A = \text{«Іванов вивчає англійську мову»}$, $B = \text{«Іванов має оцінку 8 з інформатики»}$. Сформулюйте висловлення:

- $A \wedge B$;
- $A \vee B$;
- $A \vee \overline{A \wedge B}$;
- $\overline{A \vee B} \wedge A$.

- 5°. Укажіть порядок виконання операції в логічних виразах:

- $\overline{A \vee B} \wedge A$;
- $\overline{A \vee B} \wedge (A \vee B)$;
- $A \vee B \wedge A$;
- $A \wedge B \vee C \wedge D$.

- 6*. Побудуйте таблиці істинності логічних виразів:

- $A \vee A \wedge B$;
- $A \vee \overline{A \wedge B}$;
- $A \wedge B \vee \overline{B}$;
- $(A \vee \overline{B}) \wedge (\overline{A \vee B})$;
- $\overline{A \vee B}$;
- $A \wedge \overline{B} \vee \overline{A \wedge B}$.

- 7*. Доведіть логічні формули:

- $A \wedge A = A$;
- $\overline{A \wedge B} = \overline{A} \vee \overline{B}$;
- $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$;
- $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$;
- $A \wedge (A \vee B) = A$.

2.8. Розгалуження в алгоритмах і програмах



- Що таке алгоритм? Назвіть основні блоки блок-схеми алгоритму та поясніть їх призначення.
- Які алгоритми (фрагменти алгоритмів) називаються лінійними? У чому полягає їхня характерна особливість?
- Які вирази називаються логічними? Які значення вони можуть набувати?
- Назвіть логічні операції, наведіть означення кожної з них.
- Назвіть логічні функції табличного процесора **Excel 2007**. Чому дорівнюють їхні значення залежно від значень аргументів?

Алгоритми з розгалуженням

У попередніх пунктах було розглянуто кілька лінійних алгоритмів, зокрема, алгоритми для розв'язування задач на обчислення значень

арифметичного виразу для виконавця, який вміє виконувати арифметичні операції. Розглянемо приклад задачі, алгоритм розв'язування якої не є лінійним.

Задача 1. Обчислити значення виразу $(a - b) / (c - d)$, де a, b, c, d – дійсні числа.

Звернемо увагу на те, що значення цього виразу можна обчислити не для будь-якого набору значень змінних a, b, c, d . Адже цей вираз містить дію ділення на вираз зі змінними, значення якого може дорівнювати нулю. Тобто якщо значення різниці $c - d$ дорівнює нулю, то значення виразу $(a - b) / (c - d)$ обчислити не можна, а якщо не дорівнює, то можна.

Це означає, що система команд виконавця повинна містити команду порівняння двох чисел, наприклад таку: « $s = t?$ », де s і t – або числа, або змінні чи вирази, які мають певні числові значення. Така команда є прикладом **команди перевірки умови**. Результатом виконання команди перевірки умови може бути або **істина (умова виконується)**, або **хиба (умова не виконується)**.

З іншого боку, ви вже знаєте, що порівняння « $s = t?$ » можна розглядати як висловлення або як простий логічний вираз, який набуватиме

значення **true** або **false** залежно від конкретних значень змінних s і t . І тоді команду перевірки умови можна інтерпретувати як **команду обчислення значення логічного виразу**.

Алгоритм розв'язування задачі 1 виглядатиме так:

1. Увести значення змінних a, b, c, d .
2. $x := c - d$.
3. Обчислити значення логічного виразу $x = 0$.
4. Якщо обчислене значення логічного виразу **true**, то вивести повідомлення «Вираз значення не має: ділення на нуль», після чого виконати команду 8, якщо **false**, то виконати команду 5.
5. $y := a - b$.
6. $z := y/x$.
7. Вивести значення змінної z .
8. Закінчити виконання алгоритму.

Команди 1–3 наведеного алгоритму виконуватимуться при будь-якому наборі значень змінних a, b, c, d . Подальше виконання цього алгоритму залежатиме від значення логічного виразу, обчисленого в

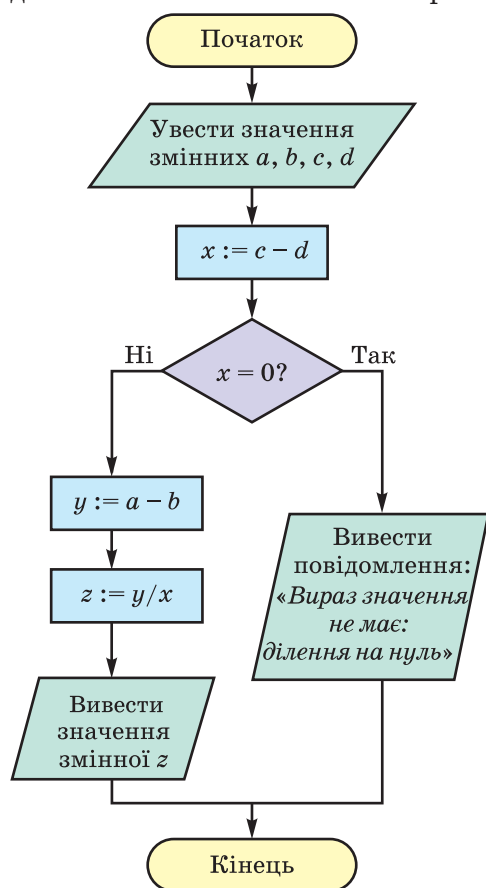



Рис. 2.39. Блок-схема алгоритму обчислення значення виразу $(a - b) / (c - d)$

команді 3. Якщо це значення **false**, то виконуватимуться команди 5–8, а якщо **true**, то виконуватимуться команда виведення повідомлення: «Вироз значення не має: ділення на нуль» і команда 8.

У блок-схемі алгоритму команди перевірки умови або обчислення значення логічного виразу позначаються блоком **Рішення** . Оскільки результатом виконання цих команд може бути або **true**, або **false**, то з цього блока є два виходи. Вихід **Так** означає, що результатом перевірки умови є **true**, а вихід **Ні** – результатом перевірки умови є **false**.

Наведемо блок-схему розглянутого вище алгоритму розв’язування задачі 1 (рис. 2.39).

Розглянемо фрагмент алгоритму на рисунку 2.39 від блока **Рішення** до блока **Термінатор** (не включаючи цей блок).



Характерною рисою цього фрагмента алгоритму є те, що при кожному його виконанні деякі команди будуть виконуватися, причому кожна по одному разу, а деякі – виконуватися не будуть. Це залежить від результату виконання команди перевірки умови (команди обчислення значення логічного виразу).

Такий фрагмент алгоритму називається **розгалуженням**.

Виконаємо наведений алгоритм для двох наборів даних. Ці набори підберемо так, щоб проілюструвати хід виконання алгоритму для кожного з двох можливих результатів обчислення значення логічного виразу.

Команда	Результат виконання
Виконання для першого набору даних	
Увести значення змінних a, b, c, d	$a = 5, b = 6, c = -3, d = 5$
$x := c - d$	$x = -3 - 5 = -8$
Обчислити значення логічного виразу $x = 0$	$(-8 = 0) = \text{false}$
$y := a - b$	$y = 5 - 6 = -1$
$z := y/x$	$z = -1/(-8) = 0,125$
Вивести значення змінної z	$z = 0,125$
Виконання для другого набору даних	
Увести значення змінних a, b, c, d	$a = 12,3, b = -1, c = 8,2, d = 8,2$
$x := c - d$	$x = 8,2 - 8,2 = 0$
Обчислити значення логічного виразу $x = 0$	$(0 = 0) = \text{true}$
Вивести повідомлення: «Вироз значення не має: ділення на нуль»	Повідомлення: «Вироз значення не має: ділення на нуль»

Звертаємо вашу увагу:

- наведений алгоритм містить як розгалуження, так і лінійні фрагменти;
- у розгалуженнях можна використовувати як прості логічні вирази, так і складені.

В алгоритмах використовують розгалуження двох видів: **повне розгалуження** (рис. 2.40) і **неповне розгалуження** (рис. 2.41).

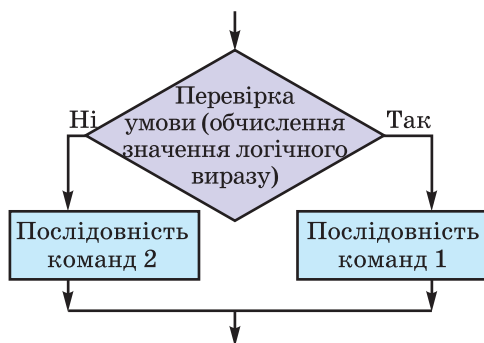


Рис. 2.40. Повне розгалуження

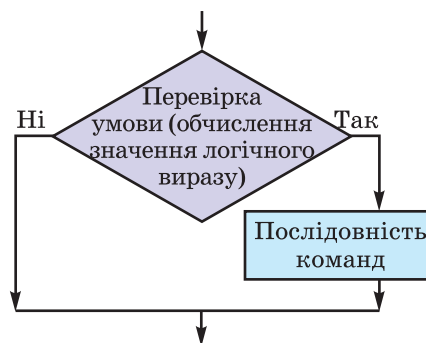


Рис. 2.41. Неповне розгалуження

Виконання повного розгалуження відбувається так: виконавець виконує команду перевірки умови (команду обчислення значення логічного виразу); якщо результат виконання цієї команди **true**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; якщо ж результат виконання цієї команди **false**, то виконавець виконує **послідовність команд 2**, після чого також переходить до виконання першої команди наступного фрагмента алгоритму.

Виконання неповного розгалуження відрізняється від виконання повного розгалуження тим, що при результаті виконання команди перевірки умови **false** виконавець одразу переходить до виконання першої команди наступного фрагмента алгоритму.

Усередині розгалуження можуть знаходитися як лінійні фрагменти алгоритму, так і інші розгалуження. Наведемо приклад алгоритму із розгалуженням у розгалуженні.

Задача 2. Дано два числа. Визначити, чи рівні вони. Якщо ні, то яке з них більше?

Блок-схема алгоритму розв'язування цієї задачі представлена на рисунку 2.42.

Наведемо приклад ще однієї задачі, алгоритм розв'язування якої містить розгалуження.

Задача 3. Є дев'ять однакових на вигляд монет, одна з яких фальшива і легша за інші. Двома зважуваннями на терезах без важків визначити фальшиву монету.

Складемо алгоритм для виконавця з такою системою команд:

1. Узяти вказану купку монет.
2. Розділити вказану купку монет на три рівні купки.
3. Покласти на терези вказані купки монет.
4. Перевірити умову «Терези в рівновазі?».
5. Визначити при зважуванні, яка з купок монет легша.
6. Повідомити результат.

Для виконавця з такою системою команд алгоритм розв'язування задачі 3 такий:

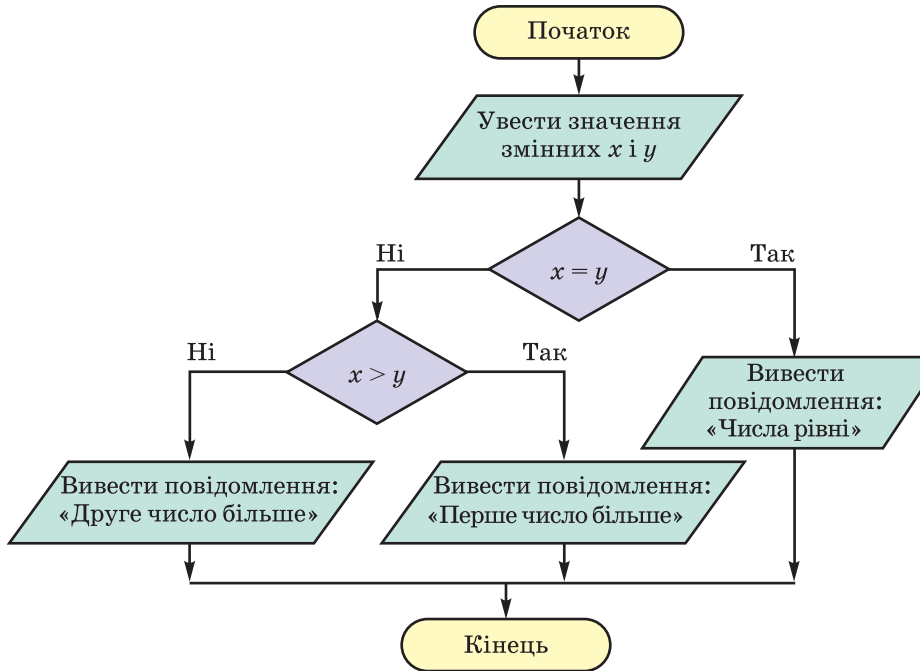


Рис. 2.42. Блок-схема алгоритму розв'язування задачі 2

1. Узяти дану купку з дев'яти монет.
2. Розділити взяту купку монет на три рівні купки.
3. Покласти на терези першу і другу купки монет.
4. Перевірити умову «Терези в рівновазі?».
5. Якщо **істина**, то взяти третю купку монет, якщо **хиба**, то взяти легшу купку.
6. Розділити взяту купку монет на три рівні купки.
7. Покласти на терези першу і другу купки монет.
8. Перевірити умову «Терези в рівновазі?».
9. Якщо **істина**, то повідомити: «Фальшивою є монета, що залишилася не покладеною на терези», якщо **хиба**, повідомити: «Фальшивою є легша монета».

Звертаємо вашу увагу: в алгоритмі розв'язування задачі 3 виконання або невиконання тих чи інших команд алгоритму залежить від того, в яку купку попаде фальшива монета. Наприклад, якщо під час першого зважування терези будуть не в рівновазі, то команда «Узяти третю купку монет» не буде виконана. Але якщо монети розподілити на купки по-іншому, то може статися так, що при першому зважуванні терези будуть у рівновазі, і тоді не буде виконана команда «Узяти легшу купку монет».

Розгалуження в Delphi

Команда повного розгалуження в **Delphi** має такий загальний вигляд:

If <логічний вираз>

Then begin

< послідовність команд 1 >

```

    end
Else begin
    < послідовність команд 2 >
end;

```

(англ. *if* – якщо, *then* – тоді, *else* – інакше).

Є в Delphi й команда неповного розгалуження:

```

If <логічний вираз>
Then begin
    <послідовність команд>
end;

```

Виконання команди повного розгалуження відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за командою розгалуження; якщо це значення **false**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за командою розгалуження.

Виконання команди неповного розгалуження відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд** і після цього виконується команда, наступна за командою розгалуження; якщо це значення **false**, одразу виконується команда, наступна за командою розгалуження.

Звертаємо вашу увагу:

- якщо після ключових слів **Then** і **Else** слідує тільки по одній команді, то операторні дужки **begin** і **end** можна не ставити;
- перед **Else** ставити крапку з комою не можна.

Якщо записати в Delphi фрагмент програми розв'язування задачі 1 (блок-схема відповідного алгоритму наведена на рис. 2.39), то він виглядатиме так:

```

a := StrToFloat (Edit1.Text);
b := StrToFloat (Edit2.Text);
c := StrToFloat (Edit3.Text);
d := StrToFloat (Edit4.Text);
x := c-d;
If x = 0
Then Label1.Caption := 'Вираз значення не має: ділення на нуль'
Else begin
    y := a-b;
    z := y/x;
    Label1.Caption := FloatToStr (z);
end;

```

Звертаємо вашу увагу: обчислення значення виразу можна було б записати не двома командами, а однією: $z := (a-b)/x$.

Логічний вираз у команді розгалуження може бути простим або складеним. Складені логічні вирази утворюються з простих шляхом використання розглянутих у попередньому пункті операцій кон'юнкції (**and**), диз'юнкції (**or**), заперечення (**not**). Порядок виконання цих операцій в Delphi такий самий, як і в математичній логіці.

Наведемо приклад задачі з використанням складених логічних виразів.

Задача 4. Обчислити значення заданої функції

$$y = \begin{cases} 2x - 12, & \text{якщо } -3 < x < 5, \\ 7 - 8x & \text{для всіх інших } x. \end{cases}$$



Фрагмент програми для розв'язування цієї задачі виглядатиме так:

```
x := StrToFloat (Edit1.Text);
If (x > -3) and (x < 5)
Then y := 2*x - 12
Else y := 7 - 8*x;
Label1.Caption := FloatToStr (y);
```

Команда розгалуження в **Delphi** може містити в собі іншу команду розгалуження. Ось як виглядатиме команда розгалуження для алгоритму, наведеного на рисунку 2.42:

```
If x = y
Then Label1.Caption := 'Числа рівні'
Else If x > y
Then Label1.Caption := 'Перше число більше'
Else Label1.Caption := 'Друге число більше';
```

Перемикачі та прапорці

Для реалізації розгалуження в проекті можна використати такі елементи керування як **перемикачі** та **прапорці**. Працюючи з текстовими і графічними редакторами, редактором презентацій, іншими програмами, ви неодноразово бачили, що за вибору перемикача всередині з'являється позначка , а за вибору прапорця – позначка .

Нагадаємо, що з групи перемикачів може бути вибраний тільки один, а в групі прапорців позначка може бути встановлена або на одному з них, або на кількох, або на жодному. Перемикачі призначені для того, щоб у ході виконання проекту вибрати один з можливих варіантів. Наприклад, установити для певної властивості об'єкта одне значення з деякого визначеного набору значень. Прапорці призначені для того, щоб у ході виконання проекту вибрати або один з можливих варіантів, або кілька. Наприклад, установити значення однієї або кількох властивостей об'єкта з деякого визначеного набору властивостей.

Для розміщення на формі перемикача потрібно встановити вказівник на рядок **TRadioButton** (англ. *radio button* – кругла кнопка) вікна **Tool Palette** та двічі клацнути ліву кнопку миші, а для розміщення прапорця – встановити вказівник на рядок **TCheckBox** (англ. *check box* – коробка для перевірки) і також двічі клацнути ліву кнопку миші.

Перемикач і прапорець мають більшість властивостей, які ми розглядали для форми, поля, кнопки і напису, та ще деякі, серед яких розглянемо властивість **Checked** (англ. *checked* – вибраний), яка може набувати одне з двох значень *True* або *False* і яка визначає, встановлена позначка в даному об'єкті чи ні (вибраний цей об'єкт чи ні). Тому значення цієї властивості можна використовувати в команді розгалуження замість логічного виразу.

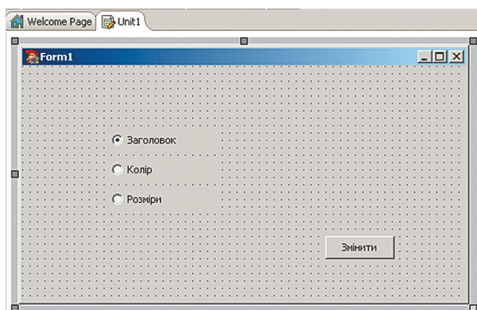


Рис. 2.43. Форма з кнопкою і трьома перемикачами

властивості **Checked** – *False*. До запуску проекту змінимо у першого перемикача значення цієї властивості на *True*. Після цього на формі він виглядатиме вибраним (рис. 2.43). Таким він буде і після запуску проекту на виконання.

Створимо процедуру для опрацювання події **OnClick** кнопки, і до її тексту помістимо таку команду:

```

If RadioButton1.Checked
Then Form1.Caption := 'Заголовок змінений'
Else If RadioButton2.Checked
Then Form1.Color := clRed
Else begin
Form1.Width := 600;
Form1.Height := 300;
end;

```

Розпочинається виконання цієї команди буде з визначення значення властивості **RadioButton1.Checked**.

Якщо при настанні події **OnClick** кнопки вибраним буде перший перемикач, то **RadioButton1.Checked = true**. Тому виконається команда змінення заголовка форми, а інші частини всієї цієї команди виконуватися не будуть.

Якщо при настанні події **OnClick** кнопки вибраним буде другий перемикач, то **RadioButton1.Checked = false**. Тому будуть виконуватися команди після першого ключового слова **Else**, тобто внутрішня команда розгалуження. Під час її виконання визначається значення властивості **RadioButton2.Checked**. Оскільки вибрано другий перемикач, то це значення дорівнює **true**. Тому виконається команда змінення кольору форми, а інші частини команди виконуватися не будуть.

Якщо ж при настанні події **OnClick** кнопки вибраним буде третій перемикач, то **RadioButton1.Checked = false**. Тому будуть виконуватися команди після першого ключового слова **Else**, тобто внутрішня команда розгалуження. При її виконанні визначається значення властивості **RadioButton2.Checked**. Оскільки другий перемикач не вибрано, то її значення **false**. Тоді будуть виконуватися команди після другого ключового слова **Else**, тобто дві команди змінення розмірів форми.

Якщо перед запуском проекту на виконання у жодного перемикача не змінювати значення властивості **Checked**, встановлені за замовчуванням, то при запуску проекту жоден з них не буде вибраним. Якщо

Розглянемо, як можна використати перемикачі для реалізації розгалуження в проекті.

Нехай, наприклад, потрібно в ході виконання проекту змінювати значення тільки однієї з таких властивостей форми: або заголовок, або колір, або розміри.

Для цього розмістимо на формі кнопку і три перемикачі (рис. 2.43). При розміщенні перемикачів на формі для кожного з них за замовчуванням встановлюється значення

і під час виконання проекту не вибрати жоден з перемикачів і вибрати кнопку, то зміняться розміри форми (подумайте самостійно, чому відбудуться саме такі зміни). Щоб так не сталося, потрібно або обов'язково вибирати один з перемикачів, або після другого ключового слова **Else** вставити ще одну команду розгалуження:

```
If RadioButton1.Checked
Then Form1.Caption := 'Заголовок змінений'
Else If RadioButton2.Checked
      Then Form1.Color := clRed
      Else If RadioButton2.Checked
            Then
              begin
                Form1.Width := 600;
                Form1.Height := 300;
              end;
```

Пояснить, як виконуватиметься ця команда, якщо жоден з перемикачів не буде вибраний, і чому.

Розглянемо тепер, як можна використати прапорці для реалізації розгалуження.

Нехай, наприклад, нам потрібно в ході виконання проекту змінювати значення або тільки однієї з властивостей форми (або заголовок, або колір, або розміри), або кількох із них, або не змінювати значення жодної з них.

Для цього розмістимо на формі кнопку і три прапорці (рис. 2.44). При розміщенні прапорців на формі для кожного з них за замовчуванням автоматично встановлюється значення властивості **Checked** – **False**. Оскільки при виконанні проекту позначка жодного прапорця може бути не встановлена, то доцільно залишити такі початкові значення властивості **Checked** усіх прапорців.

Створимо процедуру для опрацювання події **OnClick** кнопки і до її тексту помістимо такі три послідовні (не вкладені одна в одну) команди розгалуження:

```
If CheckBox.Checked
Then Form1.Caption := 'Заголовок змінений';
If CheckBox.Checked
Then Form1.Color := clRed;
If CheckBox.Checked
Then begin
      Form1.Width := 500;
      Form1.Height := 200;
    end;
```

Кожна з цих команд незалежно одна від одної буде визначати, чи вибраний відповідний прапорець, і залежно від цього кожна з них буде чи не буде виконувати відповідні команди зміни значень властивостей форми.

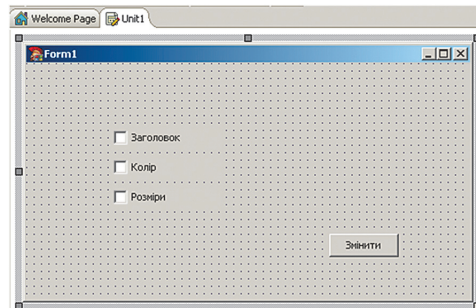


Рис. 2.44. Форма з кнопкою і трьома прапорцями

З розглянутих вище прикладів можна зробити такі висновки:

- перемикачі використовуються тоді, коли потрібно вибрати один і тільки один варіант дій з кількох можливих;
- для опрацювання стану перемикачів у процедуру включають команди розгалуження, які вкладені одна в одну;
- прапорці використовуються тоді, коли може бути вибраний або один, або кілька, або не вибраний жоден варіант дій з кількох можливих;
- для опрацювання стану прапорців у процедуру включають послідовні команди розгалуження, які незалежні одна від одної.



Перевірте себе

- 1°. Які значення можуть бути результатом виконання команди перевірки умови (обчислення значення логічного виразу)?
- 2°. Як позначається команда перевірки умови в блок-схемі алгоритму?
- 3°. Назвіть характерні особливості розгалуження.
- 4°. Зобразіть блок-схеми повного і неповного розгалуження. Опишіть особливості їх виконання.
- 5°. Поясніть відмінності повного і неповного розгалуження.
- 6*. Наведіть приклади правил з української мови, математики, інших предметів, що містять розгалуження.
- 7*. Наведіть приклади життєвих ситуацій, які можна описати алгоритмом з розгалуженням.
- 8°. Наведіть загальний вигляд команд повного і неповного розгалуження в Delphi.
- 9°. Поясніть виконання команд повного і неповного розгалуження в Delphi.
- 10°. Які операції використовують для утворення складених логічних виразів у Delphi?
- 11°. Для чого використовують прапорці та перемикачі?
- 12°. Що визначає властивість **Checked** для перемикачів і прапорців?
- 13°. Поясніть відмінності використання команд розгалуження при складанні проектів з прапорцями та перемикачами.
- 14°. Поясніть хід виконання фрагментів програм:

a) **If** <логічний вираз 1>
Then begin
 <команди 1>
end
Else If <логічний вираз 2>
 Then begin
 <команди 2>
 end
 Else begin
 <команди 3>
 end;

б) **If** <логічний вираз 1>
Then begin
 <команди 1>
end;
If <логічний вираз 2>
 Then begin
 <команди 2>
 end;



Виконайте завдання

- 1°. Складіть блок-схему алгоритму обчислення значення виразу $(a + b) - c : a$. Виконайте його при різних значеннях a, b, c . Підберіть числа так, щоб кожного разу виконання алгоритму відбувалося по-різному.
- 2°. Складіть блок-схему алгоритму обчислення значення виразу $(a + b) - c : (a - 2b)$. Виконайте його при різних значеннях a, b, c . Підберіть числа так, щоб кожного разу виконання алгоритму відбувалося по-різному.

- 3*. Складіть блок-схему алгоритму знаходження x з рівняння $ax = b$. Виконайте його при: а) $a = 2, b = -8$; б) $a = 20, b = 0$; в) $a = 0, b = 12$; г) $a = 0, b = 0$.
- 4*. Складіть блок-схему алгоритму розв'язування квадратного рівняння. Виконайте його при різних значеннях коефіцієнтів рівняння.
- 5*. Є деяка кількість однакових на вигляд монет. Відомо, що серед них є одна фальшива і вона важча за справжню. Складіть алгоритм виявлення фальшивої монети найменшою кількістю зважувань на шалькових терезах без важків, якщо кількість монет дорівнює: а) 3; б) 9; в) 27; г) 30.
- 6*. Створіть проект, який визначає менше з двох довільних дійсних чисел або виводить повідомлення, що числа рівні. Створіть у власній папці папку **Проект 2.8.6** і збережіть у ній проект.
- 7*. Створіть проект, який визначає найменше число серед трьох різних чисел. Створіть у власній папці папку **Проект 2.8.7** і збережіть у ній проект.
- 8*. Створіть проект, який за градусною мірою двох кутів трикутника обчислює градусну міру третього кута і визначає вид цього трикутника за його кутами (гострокутний, прямокутний або тупокутний). Створіть у власній папці папку **Проект 2.8.8** і збережіть у ній проект.
- 9*. Створіть проект, в якому кнопка переміщуватиметься на 20 пікселів вправо при наведенні на неї вказівника. Після того як кнопка зникне за правою межею вікна, вона повинна з'явитися біля лівої його межі. Створіть у власній папці папку **Проект 2.8.9** і збережіть у ній проект.
- 10*. Створіть проект, в якому можна обчислити значення функції
- $$y = \begin{cases} 15 - 3x, & \text{якщо } -1 < x < 4, \\ 6 + 4x & \text{для всіх інших } x. \end{cases}$$
- Створіть у власній папці папку **Проект 2.8.10** і збережіть у ній проект.
- 11*. Створіть проект, у якому можна обчислити або суму, або різницю, або добуток, або частку двох довільних дійсних чисел залежно від вибраного перемикача. Створіть у власній папці папку **Проект 2.8.11** і збережіть у ній проект.
- 12*. Створіть проект, у якому можна обчислити суму, різницю, добуток, частку двох довільних дійсних чисел залежно від установлених позначок прапорців. Створіть у власній папці папку **Проект 2.8.12** і збережіть у ній проект.

Практична робота № 8. Складання програм з розгалуженнями

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

- Відкрийте середовище розробки **Turbo Delphi 2006** і створіть новий проект.
- Розмістіть на формі елементи керування для розв'язування задачі: *Відомі довжини сторін двох прямокутників. Обчислити площі прямокутників і порівняти їх. Визначити, чи є прямокутники квадратами. З'ясувати, чи вміститься один прямокутник у середину другого. Повідомити отримані результати.* У проекті:
 - складіть обробник події **OnClick** першої кнопки для обчислення площ прямокутників і виведення повідомлення про те, у якого прямокутника площа більша;
 - складіть обробник події **OnClick** другої кнопки, виконання якого залежно від установлених прапорців приведе до визначення того, чи є відповідний прямокутник квадратом;
 - складіть обробник події **OnClick** третьої кнопки, виконання якого приведе до визначення, чи вміститься перший прямокутник усередину другого або другий прямокутник усередину першого залежно від вибраного перемикача;

- 4) виконайте складені процедури і переконайтеся, що результати їхньої роботи правильні.
3. Створіть у власній папці папку **Практична 8** і збережіть у ній проект.

2.9. Цикли в алгоритмах і в програмах



1. Які процеси називаються циклічними? Наведіть приклади.
2. Опишіть характерні властивості лінійних алгоритмів (фрагментів алгоритмів), алгоритмів з розгалуженням.
3. Як виглядає та як виконується команда повного розгалуження в **Delphi**?
4. Як виглядає та як виконується команда неповного розгалуження в **Delphi**?
5. Як і для чого використовуються перемикачі та прапорці?

Цикли в алгоритмах

У лінійних алгоритмах і в алгоритмах із розгалуженням кожна команда алгоритму могла бути виконана не більше одного разу.

Але для розв'язування багатьох задач потрібно складати алгоритми, команди яких можуть бути виконані більше одного разу. Розглянемо приклади таких задач.

Задача 1. Є діжка і відро. Використовуючи відро, наповнити діжку водою з колодязя.

Розглянемо виконавця з такою системою команд:

1. Наповнити відро водою.
2. Вилити воду з відра в діжку.
3. Перевірити умову «Діжка неповна?».

Оскільки з умови задачі невідомо, чи є в діжці вода, виконавець повинен спочатку перевірити умову «Діжка неповна?». Якщо результат цієї перевірки **true**, то він повинен наповнити відро водою, вилити її з відра в діжку і знову перевірити умову «Діжка неповна?». І так до тих пір, поки результат перевірки цієї умови стане **false**. Після цього можна припинити виконання алгоритму.

Подамо алгоритм розв'язування цієї задачі для розглянутого виконавця у словесній формі та у вигляді блок-схеми (рис. 2.45).

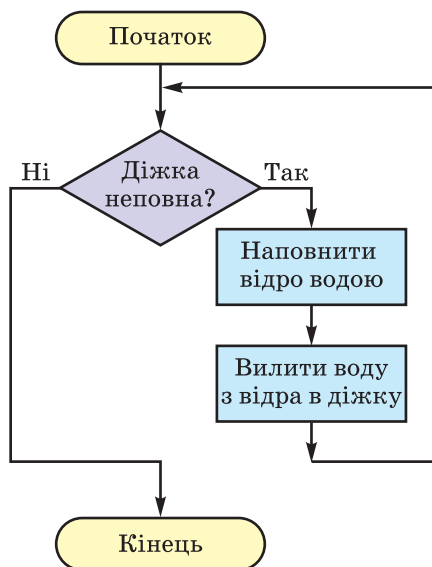


Рис. 2.45. Блок-схема алгоритму розв'язування задачі 1

1. Перевірити умову «Діжка неповна?».
2. Якщо результат виконання попередньої команди **true**, то виконати команду 3, якщо **false**, то закінчити виконання алгоритму.
3. Наповнити відро водою.
4. Вилити воду з відра в діжку.
5. Перейти до виконання команди 1.

У цьому алгоритмі команди 3–5 можуть бути виконані більше одного разу.

Чергове виконання цих команд залежить від результату перевірки умови в команді 1. Якщо цей результат **true**, то команди 3–5 виконуються ще раз, якщо ж **false**, то ці команди більше не виконуватимуться.

Звертаємо вашу увагу: команди 3–5 саме можуть бути виконані більше одного разу, а не обов'язково виконуються більше одного разу. Адже можливо, що після першого ж виливання води з відра в діжку вона наповниться і виконання алгоритму закінчиться. Крім того, якщо діжка із самого початку є повною, то ці команди не виконуються жодного разу.

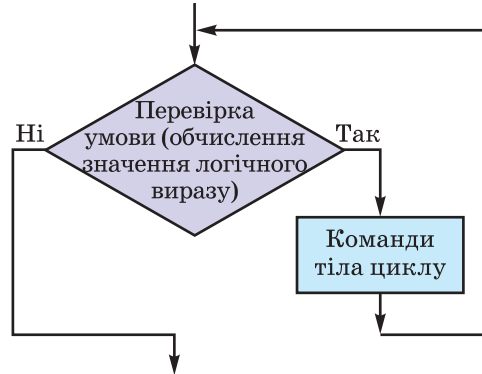


Рис. 2.46. Блок-схема циклу з передумовою



Фрагмент алгоритму, що складається з команд, які можуть бути виконані більше одного разу, називається *циклом*. Алгоритми, які містять цикли, називаються *алгоритмами з циклами*.

У наведеному алгоритмі цикл складається з трьох команд: команди перевірки умови і двох команд, які утворюють **тіло циклу**.

Розглянутий вище цикл називається **циклом з передумовою**, тому що умова перевіряється до початку виконання команд тіла циклу.

Загальний вигляд циклу з передумовою наведено на рисунку 2.46. Виконання такого циклу відбувається так: виконавець виконує **команду перевірки умови (обчислення значення логічного виразу)**; якщо результат виконання цієї команди **true**, то виконавець виконує **команди тіла циклу**, після чого знову виконує **команду перевірки умови (обчислення значення логічного виразу)**; якщо ж результат виконання команди перевірки умови (обчислення значення логічного виразу) **false**, то виконавець переходить до виконання першої команди наступного фрагмента алгоритму.

Якщо б в умові задачі 1 було відомо, що діжка порожня, то виконавцю не потрібно було б одразу перевіряти умову «Діжка неповна?». Він мав би хоча б один раз наповнити відро водою, перелити воду з відра в діжку і лише після цього перевірити умову «Діжка неповна?» (або умову «Діжка повна?», якщо вона входить до системи його команд).

Блок-схема алгоритму розв'язування такої задачі з використанням умови «Діжка повна?» представлена на рисунку 2.47.

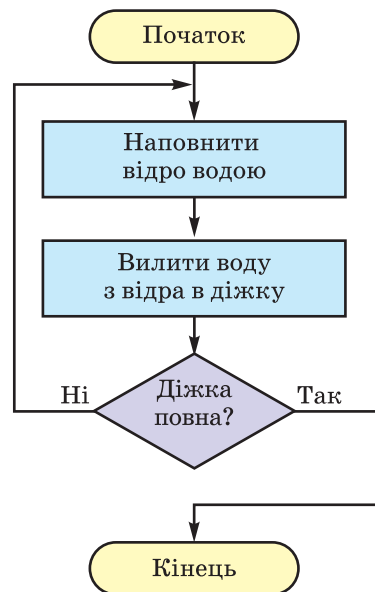


Рис. 2.47. Блок-схема алгоритму розв'язування модифікованої задачі 1

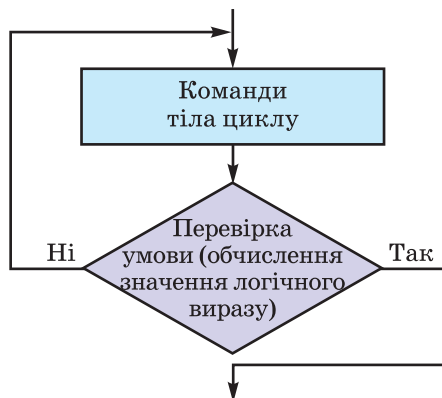


Рис. 2.48. Блок-схема циклу з післяумовою

При виконанні наведеного алгоритму команди тіла циклу обов'язково виконуватимуться хоча б один раз, тому що команда перевірки умови виконується після виконання команд тіла циклу. Такий цикл називається **циклом з післяумовою**.

Загальний вигляд блок-схеми циклу з післяумовою наведено на рисунку 2.48.

Виконання такого циклу відбувається так: виконавець виконує **команди тіла циклу**, після чого виконує **команду перевірки умови (обчислення значення логічного виразу)**; якщо результат виконання цієї команди **false**,

то виконавець знову виконує **команди тіла циклу**; якщо ж **true**, то виконавець переходить до виконання першої команди наступного фрагмента алгоритму.

Звертаємо вашу увагу: якщо в алгоритмі, блок-схема якого наведена на рисунку 2.47, використати умову «Діжка неповна?», то виконання циклу продовжувалося б при результаті **true** виконання команди перевірки умови і припинялося б при результаті **false** виконання цієї команди.

Таким чином, ми розглянули **три базові структури алгоритмів: лінійні фрагменти (слідування), розгалуження та цикли (повторення)**. Доведено, що використовуючи тільки ці три структури, можна скласти алгоритм розв'язування будь-якої задачі, якщо він існує.

Зауважимо, що більшість алгоритмів містять і лінійні фрагменти, і розгалуження, і цикли.

Команда циклу з лічильником у Delphi

У мові програмування **Delphi** є кілька команд, які можуть реалізувати цикл. Одна з них – **команда циклу з лічильником**. Її доцільно використовувати в тих випадках, коли кількість повторень команд тіла циклу відома до початку виконання команди циклу. Загальний вигляд цієї команди такий:

```

for <ім'я змінної> := <вираз 1> to <вираз 2> do
begin
  <команди тіла циклу>
end;
  
```

Рядок **for ... to ... do** (англ. *for* – для, *to* – до, *do* – робити, виконувати) називається **рядком заголовка команди циклу з лічильником**. Змінна в рядку заголовка команди циклу з лічильником, що стоїть перед знаком присвоювання, називається **лічильником циклу**.

Лічильник циклу, вираз 1 і вираз 2 мають набувати тільки цілих значень. Якщо тіло циклу складається лише з однієї команди, операторні дужки **begin і end** можна не ставити.

Виконується команда циклу з лічильником так:

1. Надати лічильнику циклу (наприклад, змінній **i**) значення **<вираз 1>**.

2. Обчислити значення логічного виразу $i \leq \text{вираз 2}$.
 3. Якщо значення логічного виразу, яке обчислено в команді 2, **true**, то виконати команди тіла циклу і перейти до виконання команди 4, інакше виконати команду, наступну за командою циклу.
 4. Збільшити значення лічильника циклу на 1.
 5. Перейти до виконання команди 2.
- Блок-схема виконання команди циклу з лічильником наведена на рисунку 2.49.

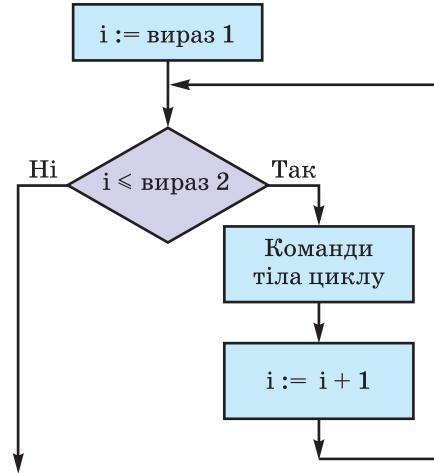


Рис. 2.49. Блок-схема виконання команди циклу з лічильником

Для ілюстрації виконання команди циклу з лічильником розглянемо фрагмент програми, в якому обчислюється сума $1! + 2! + 3! + 4!$ (нагадаємо, що $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$):

```

var a, s, i: Integer;
begin
  s := 0; a := 1;
  For i := 1 to 4 do
  begin
    a := a*i;
    s := s + a;
  end;
  Label1.Caption := IntToStr (s);
end;
  
```

Виконаємо цей фрагмент програми.

Команда	Результат виконання
$s := 0$	$s = 0$
$a := 1$	$a = 1$
$i := 1$	$i = 1$
$i \leq 4$	$(1 \leq 4) = \text{true}$
$a := a*i$	$a = 1*1 = 1$
$s := s + a$	$s = 0 + 1 = 1$
$i := i + 1$	$i = 2$
$i \leq 4$	$(2 \leq 4) = \text{true}$
$a := a*i$	$a = 1*2 = 2$
$s := s + a$	$s = 1 + 2 = 3$
$i := i + 1$	$i = 3$
$i \leq 4$	$(3 \leq 4) = \text{true}$
$a := a*i$	$a = 2*3 = 6$
$s := s + a$	$s = 3 + 6 = 9$
$i := i + 1$	$i = 4$

Команда	Результат виконання
$i \leq 4$	$(4 \leq 4) = \text{true}$
$a := a * i$	$a = 6 * 4 = 24$
$s := s + a$	$s = 9 + 24 = 33$
$i := i + 1$	$i = 5$
$i \leq 4$	$(5 \leq 4) = \text{false}$
$\text{Label1.Caption} := \text{IntToStr}(s)$	$\text{Label1.Caption} = 33$

Звертаємо вашу увагу:

- Після закінчення виконання команди циклу з лічильником лічильник циклу має значення на 1 більше, ніж значення вираз 2, і це значення, за потреби, можна використовувати в наступних командах.
- Існує різновид команди циклу з лічильником, який змінюється в зворотному порядку:

```
for <ім'я змінної> := <вираз 1> downto <вираз 2> do
begin
    <команди тіла циклу>
end;
```

(англ. *down* – униз), який відрізняється від попереднього лише тим, що після кожного виконання команд тіла циклу значення лічильника циклу не збільшується на 1, а зменшується на 1.

Команди циклу з передумовою та циклу з післяумовою в Delphi

Якщо кількість повторень команд тіла циклу до початку виконання команди циклу невідома, потрібно використовувати команду циклу з передумовою або команду циклу з післяумовою.

Загальний вигляд команди циклу з передумовою такий:

```
while <логічний вираз> do
begin
    <команди тіла циклу>
end;
```

(англ. *while* – поки). Якщо тіло циклу складається лише з однієї команди, операторні дужки **begin** і **end** можна не ставити.

Виконується команда циклу з передумовою так:

- Обчислити значення логічного виразу.
- Якщо це значення **true**, то виконати команди тіла циклу і перейти до команди 1, а якщо **false**, то виконати команду, наступну за командою циклу.

Для ілюстрації виконання команди циклу з передумовою розглянемо фрагмент програми для розв'язування наступної задачі.

Задача 2. Обчислити суму додатних членів арифметичної прогресії з додатним першим членом і від'ємною різницею.

Домовимося вводити перший член прогресії в поле **Edit1**, а її різницю – в поле **Edit2**.

```

var a, d, s: Real;
begin
  a := StrToFloat (Edit1.Text);
  d := StrToFloat (Edit2.Text);
  s := 0;
  while a > 0 do
    begin
      s := s + a;
      a := a + d;
    end;
  Label1.Caption := FloatToStr (s);
end;

```

Виконаємо цей фрагмент програми для деякого набору значень a і d .

Команда	Результат виконання
$a := \text{StrToFloat}(\text{Edit1.Text})$	$a = 7$
$d := \text{StrToFloat}(\text{Edit2.Text})$	$d = -3$
$s := 0$	$s = 0$
$a > 0$	$(7 > 0) = \text{true}$
$s := s + a$	$s = 0 + 7 = 7$
$a := a + d$	$a = 7 + (-3) = 4$
$a > 0$	$(4 > 0) = \text{true}$
$s := s + a$	$s = 7 + 4 = 11$
$a := a + d$	$a = 4 + (-3) = 1$
$a > 0$	$(1 > 0) = \text{true}$
$s := s + a$	$s = 11 + 1 = 12$
$a := a + d$	$a = 1 + (-3) = -2$
$a > 0$	$(-2 > 0) = \text{false}$
$\text{Label1.Caption} := \text{FloatToStr}(s)$	$\text{Label1.Caption} = 12$

Загальний вигляд команди циклу з післяумовою такий:

```

repeat
  <команди тіла циклу>
until <логічний вираз>;

```

(англ. *repeat* – повторити, *until* – доки, не раніше як).

Виконується команда циклу з передумовою так:

1. Виконати команди тіла циклу.
2. Обчислити значення логічного виразу.
3. Якщо це значення **false**, то виконати п.1, а якщо **true**, то виконати команду, наступну за командою циклу.

Команду циклу з післяумовою доцільно використовувати в тих випадках, коли команди тіла циклу повинні виконатися хоча б один раз.

Проілюструємо, як виглядатиме фрагмент програми для розв’язування попередньої задачі з використанням циклу з післяумовою:

```

var a, d, s: Real;
begin
  a := StrToFloat (Edit1.Text);
  d := StrToFloat (Edit2.Text);

```

```

s := 0;
repeat
  s := s + a;
  a := a + d;
until a <= 0;
Label1.Caption := FloatToStr (s);
end;

```

Виконаємо цей фрагмент програми для деякого набору значень a і d .

Команда	Результат виконання
$a := \text{StrToFloat}(\text{Edit1.Text})$	$a = 7$
$d := \text{StrToFloat}(\text{Edit2.Text})$	$d = -3$
$s := 0$	$s = 0$
$s := s + a$	$s = 0 + 7 = 7$
$a := a + d$	$a = 7 + (-3) = 4$
$a \leq 0$	$(4 \leq 0) = \text{false}$
$s := s + a$	$s = 7 + 4 = 11$
$a := a + d$	$a = 4 + (-3) = 1$
$a \leq 0$	$(1 \leq 0) = \text{false}$
$s := s + a$	$s = 11 + 1 = 12$
$a := a + d$	$a = 1 + (-3) = -2$
$a \leq 0$	$(-2 \leq 0) = \text{true}$
$\text{Label1.Caption} := \text{FloatToStr}(s)$	$\text{Label1.Caption} = 12$

Звертаємо вашу увагу:

1. У команді циклу з післяумовою операторні дужки не використовують незалежно від кількості команд у тілі циклу.
2. Якщо цикл з передумовою замінити на цикл з післяумовою або навпаки, то логічний вираз одного є запереченням логічного виразу іншого.

Приклади програм з використанням циклів

Розглянемо кілька цікавих і корисних для подальшого вивчення теми задач, в алгоритмах розв'язування яких використовуються цикли.

Під час розв'язування багатьох задач доцільно використовувати ще дві арифметичні операції: знаходження неповної частки та остачі при діленні цілого числа на натуральне. Нагадаємо, що для будь-якого цілого числа m і натурального числа n існує єдина пара цілих чисел q і r ($0 \leq r < n$), таких що $m = nq + r$. Число q називається **неповною часткою**, а число r – **остачею**. Для знаходження неповної частки в **Delphi** використовується операція **div** (англ. *divide* – поділити), а для знаходження остачі – **mod** (англ. *modulo* – остача від ділення). Наприклад,

$$\begin{aligned}
23 \text{ div } 5 &= 4, & 23 \text{ mod } 5 &= 3, \\
28 \text{ div } 4 &= 7, & 28 \text{ mod } 4 &= 0, \\
2 \text{ div } 3 &= 0, & 2 \text{ mod } 3 &= 2.
\end{aligned}$$

Задача 3. Дано натуральне число n , більше за 1. З'ясувати, чи є це число простим.

Нагадаємо, що простим називається натуральне число, яке має рівно два дільники. Тому можна перебрати всі натуральні числа від 1 до дано-

го числа і підрахувати кількість дільників даного числа. Якщо ця кількість дорівнює двом, то це число просте, якщо більше – не просте. Відповідний фрагмент програми виглядатиме так:

```
var i, k, n: Integer;
begin
  n := StrToInt (Edit1.Text);
  k := 0; // Кількість дільників числа n
  for i := 1 to n do
    if n mod i = 0 //Перевірка, чи є число i дільником числа n
    then k := k + 1; {Збільшення на 1 кількості дільників числа n, якщо число i є його дільником}
    if k = 2
    then Label1.Caption := 'просте'
    else Label1.Caption := 'не просте';
  end;
```

Але час виконання програми для розв'язування цієї задачі можна суттєво зменшити, якщо врахувати такі властивості натуральних чисел:

1. Будь-яке натуральне число, більше за 1, завжди має два дільники (одиницю і саме це число). Тому простим буде таке натуральне число, яке не матиме інших дільників.
2. Серед натуральних чисел тільки одне парне число є простим (2), всі інші прості числа – непарні.
3. Якщо не враховувати саме число, то у натурального числа немає дільників, які перевищують арифметичний квадратний корінь з цього числа.

Якщо використати вказані властивості, то відповідний фрагмент програми може бути таким:

```
var i, k, n: Integer; f: Boolean;
begin
  n := StrToInt (Edit1.Text);
  f := true; {Будемо поки що вважати число n простим, адже дільників у нього поки що не знайшлося}
  if (n mod 2 = 0) and (n <> 2)
  then f := false // Якщо число n парне і не дорівнює 2, то воно не просте
  else
    begin
      k := 3; {Якщо число непарне, то будемо шукати його дільники, починаючи з числа 3}
      while (k <= sqrt (n)) and f do {Шукати дільники числа будемо серед чисел, які не перевищують арифметичний квадратний корінь з числа n, і поки такий дільник не знайшовся}
        if n mod k = 0 // Перевірка, чи є число k дільником числа n
        then f := false
        else k := k + 2; {Якщо k не є дільником n, то наступний можливий дільник – наступне непарне число}
      end;
    end
  if f
  then Label1.Caption := 'просте'
  else Label1.Caption := 'не просте';
end;
```

У наведеному фрагменті програми використана логічна змінна **f**. Її значення визначатиме, чи є число **n** простим чи ні: **true** – просте, **false** –

не просте. Тип логічної змінної в **Delphi** позначається **Boolean**, на честь Джорджа Буля. Для обчислення арифметичного квадратного кореня використана стандартна функція **sqrt** (англ. *square root* – квадратний корінь).

Задача 4. Знайти найбільший спільний дільник (НСД) двох даних натуральних чисел a і b ($a > b$).

У курсі математики 6-го класу ви навчилися знаходити НСД чисел, розкладаючи їх на прості множники. Можна скласти програму, в якій реалізується цей метод знаходження НСД.

Але більш простою виявляється програма, яка реалізує інший метод знаходження НСД, що базується на такому математичному твердженні: **якщо $a > b$, то НСД $(a, b) = \text{НСД}(b, r)$, де r – остача від ділення a на b .** Ідея цього методу полягає в тому, що послідовно замінюються числа, для яких потрібно знайти НСД: більше з них замінюється на менше, а менше – на остачу від ділення більшого числа на менше. Закінчується цей процес замінювання тоді, коли остача від ділення дорівнює нулю. Тоді НСД дорівнює останній відмінній від 0 остачі від ділення.

Наприклад,

$$\begin{aligned} \text{НСД}(80, 12) &= \text{НСД}(12, 8) = \text{НСД}(8, 4) = \text{НСД}(4, 0) = 4, \\ \text{НСД}(125, 54) &= \text{НСД}(54, 17) = \text{НСД}(17, 3) = \text{НСД}(3, 2) = \text{НСД}(2, 1) = \\ &= \text{НСД}(1, 0) = 1. \end{aligned}$$

Цей метод знаходження НСД називається **алгоритмом Евкліда**. Перевірте цей метод знаходження НСД для різних пар натуральних чисел.

Нижче наведено фрагмент програми, в якому знаходиться НСД двох чисел за алгоритмом Евкліда.

```
var a, b, r: Integer;
begin
  a := StrToInt (Edit1.Text);
  b := StrToInt (Edit2.Text);
  r := a mod b;
  while r <> 0 do
    begin
      a := b;
      b := r;
      r := a mod b;
    end;
  Label1.Caption := IntToStr (b);
end;
```

Звертаємо вашу увагу, що наведений фрагмент програми працює правильно і в тих випадках, коли $a \leq b$. Спробуйте самостійно з'ясувати, чому.



Перевірте себе

- 1°. Що таке цикл в алгоритмі?
- 2°. Наведіть блок-схему циклу з передумовою. Поясніть виконання цього циклу. Наведіть приклади циклів з передумовою.
- 3°. Наведіть блок-схему циклу з післяумовою. Поясніть виконання цього циклу. Наведіть приклади циклів з післяумовою.

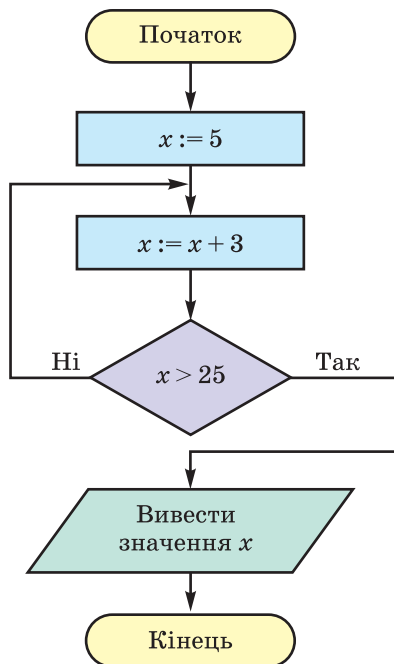
- 4*. Чи можуть команди тіла циклу не виконуватися жодного разу? Поясніть свою відповідь. Наведіть приклади.
- 5*. Чи може виконання циклу ніколи не закінчитися? Поясніть свою відповідь. Наведіть приклади.
- 6*. Поясніть відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, цикл.
- 7*. Наведіть загальний вигляд команди циклу з лічильником у **Delphi**.
- 8*. Поясніть виконання команди циклу з лічильником у **Delphi**.
- 9*. Наведіть загальний вигляд команди циклу з передумовою і команди циклу з післяумовою в **Delphi**.
- 10*. Поясніть виконання команди циклу з передумовою і команди циклу з післяумовою в **Delphi**.
- 11*. Поясніть відмінності у використанні команд циклу з параметром і циклу з передумовою в **Delphi**.
- 12*. Поясніть відмінності у використанні команд циклу з передумовою та циклу з післяумовою в **Delphi**.



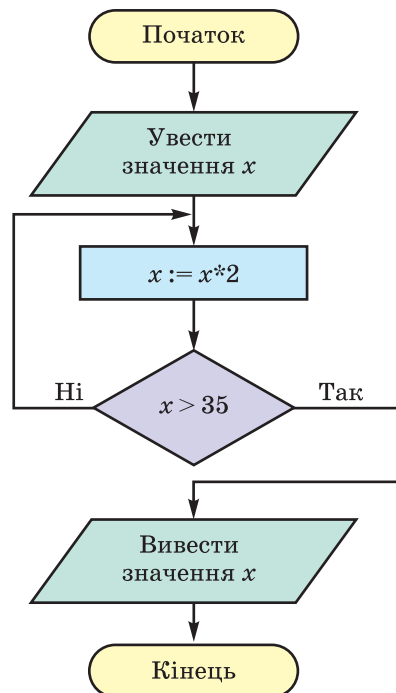
Виконайте завдання

1. Виконайте алгоритм:

а°)



б°)



- 2*. Виконайте в таблиці фрагмент програми та з'ясуйте, якими будуть значення змінних після його завершення:

а) `p := 1; a := 2;`
`for i := 1 to 5 do`
`begin`
`a := 3*a + 1;`
`p := p*a;`
`end;`

б) `k := 1; a := 12;`
`while a < 100 do`
`begin`
`a := 2*a - 4;`
`k := k + 1;`
`end;`

в) `s := 0; a := 1; k := 1;`
`repeat`
`s := s + a;`
`k := k + 1;`
`a := k*k;`
`until a > 50;`



3°. Виконайте фрагмент програми та з'ясуйте, якими будуть значення змінних після його завершення:

а) $p := 1$; $a := 8$;
for $i := 1$ to 4 do
begin
 $a := 2 * a - 5$;
 $p := p * a$;
end;

б) $k := 1$; $a := 100$;
while $a > 10$ do
begin
 $a := a \div 2$;
 $k := k + 1$;
end;

в) $s := 0$; $a := 5428$;
repeat
 $c := a \bmod 10$;
 $s := s + c$;
 $a := a \div 10$;
until $a < 1$;

4°. Виконайте в таблиці фрагмент програми для задачі 3, наведеної в тексті пункту.



5°. Виконайте в таблиці фрагмент програми для задачі 4, наведеної в тексті пункту.

6°. Складіть блок-схему алгоритму знаходження суми перших п'яти членів послідовності, перший член якої $x = 7$, а кожне наступний обчислюється на основі попереднього значення за формулою $x := 2 * x + 3$. Виконайте алгоритм.



7°. Складіть блок-схему алгоритму знаходження кількості додатних членів арифметичної прогресії з відомим першим членом та різницею -5 . Виконайте алгоритм для трьох різних значень першого члену прогресії. Підберіть ці значення так, щоб команди тіла циклу виконалися кілька разів; один раз; жодного разу.

8°. Створіть проект, у якому можна обчислити добуток перших N членів послідовності чисел, перше з яких дорівнює x , а кожне наступне вдвічі менше за попереднє. Створіть у власній папці папку **Проект 2.9.8** і збережіть у ній проект.

9°. Створіть проект, в якому можна обчислити кількість двоцифрових чисел у арифметичній прогресії. Перший член прогресії більший за 10, різниця більша за 0. Створіть у власній папці папку **Проект 2.9.9** і збережіть у ній проект.

10°. Створіть проект, у якому можна знайти кількість дільників заданого натурального числа. Створіть у власній папці папку **Проект 2.9.10** і збережіть у ній проект.

11°. Створіть проект, у якому можна обчислити кількість простих чисел серед перших 100 натуральних чисел. Створіть у власній папці папку **Проект 2.9.11** і збережіть у ній проект.

12°. Створіть проект для розв'язування задачі: *Людина поклала у банк певну суму грошей. Щорічно банк додає до суми наперед визначений відсоток від суми, що зберігається на рахунку після попереднього року. Яка сума буде на рахунку через N років?* Створіть у власній папці папку **Проект 2.9.12** і збережіть у ній проект.



13°. Створіть проект для розв'язування задачі: *Людина поклала у банк певну суму грошей. Щорічно банк додає до суми наперед визначений відсоток від суми, що зберігається на рахунку після попереднього року. Через скільки років сума на рахунку стане не меншою за S грн.?* Створіть у папці **Мої документи** папку **Проект 2.9.13** і збережіть у ній проект.

14°. Створіть проект для обчислення суми цифр заданого натурального числа, кількість цифр якого не більша за 9. Створіть у власній папці папку **Проект 2.9.14** і збережіть у ній проект.



Практична робота № 9. Програмування циклічних обчислень

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте середовище візуального проектування Turbo Delphi 2006.

2. Створіть проект для розв'язування задачі: *Перед початком повені рівень води у річці становив H м. Під час повені кожну годину рівень води зростає на P % від рівня попередньої години. Яким буде рівень води через N годин після початку повені? Через скільки годин після початку повені рівень води буде не менше, ніж K м?*
 1. Розмістіть на формі поля для введення початкових даних, написи з текстами, що будуть пояснювати їхні призначення, та три кнопки.
 2. Установіть на першій кнопці напис *Питання 1*, на другій – *Питання 2*, на третій – *Спочатку*, у полів – *порожній текст*.
 3. Складіть обробник події **OnClick** першої кнопки, виконання якого приведе до виведення у вікно повідомлення відповіді на перше запитання задачі.
 4. Виконайте складену процедуру і переконайтеся, що результати її роботи правильні.
 5. Складіть обробник події **OnClick** другої кнопки, виконання якого приведе до знаходження відповіді на друге запитання задачі та виведення його в окремий напис.
 6. Виконайте складену процедуру та переконайтеся, що результати її роботи правильні.
 7. Складіть обробник події **OnClick** третьої кнопки, виконання якого приведе до очищення тексту в полях і напису з відповіддю на друге запитання задачі.
3. Створіть у власній папці папку **Практична 9** і збережіть у ній проект.

2.10. Процедури і функції користувача та стандартні процедури і функції в Delphi



1. Що таке процедура?
2. Що таке обробник події? Як створити обробник події?
3. Який загальний вигляд процедур-обробників подій, які ви створювали в **Delphi**?
4. Які змінні називаються локальними, а які – глобальними? Поясніть різницю між ними.
5. Як у **Delphi** отримати текстове представлення значення числової змінної і навпаки?

Процедури користувача в Delphi

У попередніх пунктах ви створювали і використовували процедури, які були пов'язані з певними подіями і були методами певних об'єктів. Заготовки цих процедур середовище розробки створювало автоматично, а ви доповнювали ці заготовки командами і рядком **var** оголошення змінних.

Під час створення програм можна включати до проекту процедури, не пов'язані з подіями. Такі процедури називаються **процедурами користувача**. Їх найчастіше використовують у випадках, коли потрібно кілька разів при різних значеннях змінних виконувати одні й ті самі операції: обчислювати площу і периметр трикутників або інших геометричних фігур, вибирати більше з двох чисел, визначати, просте число чи ні, та ін. Або коли задачу можна поділити на окремі підзадачі, кожна з яких розв'язується за окремим алгоритмом, тобто при використанні методу покрокової деталізації.

Процедура користувача, як й інші процедури, – це відокремлена частина програми, яка має своє ім'я та яка може бути викликана для виконання в будь-якій іншій частині програми.

Процедури користувача можуть бути:

- без аргументів і без результатів;
- з аргументами і без результатів;
- без аргументів і з результатами;
- з аргументами і з результатами.



Аргументи процедури – це змінні, які отримують свої значення в команді виклику процедури перед початком виконання команд процедури.

Результати процедури – це змінні, які передають свої значення з процедури певним змінним у команді виклику процедури.

Найчастіше використовують процедури останнього типу. Загальний вигляд цих процедур у Delphi такий:

```
procedure <ім'я> (<списки аргументів з їхніми типами>; var <списки результатів з їхніми типами>);
var (<списки локальних змінних процедури з їхніми типами>);
begin
    <команди>
end;
```

Загальний вигляд команди виклику процедури з аргументами і результатами такий:

```
<ім'я> (<списки виразів, значення яких передаються аргументам процедури>, <списки змінних, яким передають свої значення результати процедури>);
```

Звертаємо вашу увагу: у команді виклику процедури порядок виразів, значення яких передаються аргументам процедури, і змінних, яким передають свої значення результати процедури, повинен строго відповідати порядку запису аргументів і результатів процедури в рядку її заголовка.

Оформимо як процедуру фрагмент програми, який обчислює НСД двох натуральних чисел. У цієї процедури два аргументи – два натуральні числа, НСД яких потрібно знайти, і один результат – їхній НСД. Виглядатиме ця процедура так:

```
procedure MCD (a, b: Integer; var c: Integer);
var r: Integer;
begin
    r := a mod b;
    while r <> 0 do
        begin
            a := b;
            b := r;
            r := a mod b;
        end;
    c := b;
end;
```

Викликати на виконання цю процедуру для одержання значення НСД двох конкретних натуральних чисел можна, наприклад, такою командою: **MCD (35, 77, x)**. Виконуватися ця команда буде так:

1. Змінні *a* і *b* (**аргументи процедури**) одержують значення 35 і 77 відповідно.
2. Виконуються команди процедури.
3. Змінна *c* (**результат процедури**) передає своє значення змінній *x*, вказаній у команді виклику процедури.

Отже, в результаті виконання вказаної команди виклику процедури змінна *x* одержить значення 7.

У команді виклику цієї процедури в дужках на перших двох місцях, що відповідають аргументам процедури, можуть стояти не тільки константи, а й змінні, які на момент виконання команди виклику мають певні натуральні значення. Тобто при виконанні команди **MCD (к, р, x)** аргументи процедури *a* і *b* одержать значення змінних *к* і *р* відповідно. На цих місцях можуть стояти також арифметичні вирази, значення яких будуть обчислені на початку виконання команди виклику процедури і ці значення будуть передані аргументам процедури, наприклад **MCD (2*к+3, 3*р-1, x)**.

Звертаємо вашу увагу:

1. Аргументи та результати процедури можуть бути одними й тими самими змінними, тобто допустима процедура із заголовком **Procedure One (var a, b: Integer)**. У такої процедури змінні *a* та *b* є одночасно і аргументами, і результатами. Команда виклику цієї процедури виглядатиме так: **One (x, y)**.
2. Процедура може серед своїх команд мати команду виклику іншої процедури.



Змінні, які вказані в заголовку процедури, називаються формальними параметрами процедури.

Змінні, які вказуються в команді виклику процедури, називаються фактичними параметрами процедури.

Проілюструємо зручність використання процедур користувача (зокрема, процедури знаходження НСД двох натуральних чисел) на прикладі такої задачі.

Задача 1. Знайти суму двох звичайних дробів.

У наведеному тексті проекту (рис. 2.50) процедура викликається двічі. Перший раз для знаходження НСД знаменників дробів. Цей НСД у наступній команді використовується для знаходження найменшого спільного знаменника дробів ($НСК(x, y) = x*y/НСД(x, y)$). Другий – для знаходження НСД чисельника і знаменника дробу-суми з метою його подальшого скорочення. Без використання процедури текст цього проекту збільшився б, погіршилася б його структурованість. Результат виконання додавання двох звичайних дробів можна побачити на рисунку 2.51.

```

{$R *.dfm}

procedure MCD (a, b: Integer; var c: Integer);
var r: Integer;
begin
  r := a mod b;
  while r <> 0 do
  begin
    a := b;
    b := r;
    r := a mod b;
  end;
  c := b;
end;

procedure TForm1.Button1Click(Sender: TObject);
var nom1, nom2, den1, den2, den, nom, k, z: Integer;
begin
  nom1 := StrToInt (Edit1.Text);
  den1 := StrToInt (Edit2.Text);
  nom2 := StrToInt (Edit3.Text);
  den2 := StrToInt (Edit4.Text);
  MCD(den1, den2, z);
  den := den1*den2 div z;
  nom := nom1*den div den1 + nom2*den div den2;
  MCD(den,nom, k);
  if k <> 1
  then
  begin
    den := den div k;
    nom := nom div k;
  end;
  Label5.Caption := IntToStr(nom) + '/' + IntToStr(den);
end;
end.

```

Рис. 2.50. Програма додавання двох звичайних дробів з використанням процедури MCD

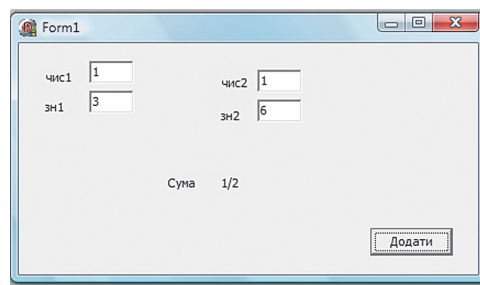


Рис. 2.51. Виконання програми додавання двох звичайних дробів

Функції користувача в Delphi

Як зазначалося вище, процедура може не мати результатів, а може мати один або більше результатів. Якщо процедура має один результат, то зручніше її оформити і викликати як **функцію користувача**.

Функція користувача, як і процедура користувача – це відокремлена частина програми, яка має своє ім'я та яка може бути викликана для виконання в будь-якій іншій частині програми.

Відрізняється функція від процедури таким:

1. Рядок заголовка функції виглядає так:

function <ім'я> (<списки аргументів з їхніми типами>): <тип результату>;

2. Значення, яке функція повинна повернути (передати) команді, яка її викликає, у тексті самої функції має бути присвоєно змінній, ім'я якої збігається з іменем функції.

3. Функція може бути викликана або в правій частині команди присвоєння, або в команді виведення.

```

{$R *.dfm}

function MCD (a, b: Integer): Integer;
var r: Integer;
begin
  r := a mod b;
  while r <> 0 do
  begin
    a := b;
    b := r;
    r := a mod b;
  end;
  MCD := b;
end;

procedure TForm1.Button1Click(Sender: TObject);
var nom1, nom2, den1, den2, den, nom, k: Integer;
begin
  nom1 := StrToInt (Edit1.Text);
  den1 := StrToInt (Edit2.Text);
  nom2 := StrToInt (Edit3.Text);
  den2 := StrToInt (Edit4.Text);
  den := den1*den2 div MCD(den1, den2);
  nom := nom1*den div den1 + nom2*den div den2;
  k := MCD(den,nom);
  if k <> 1
  then
  begin
    den := den div k;
    nom := nom div k;
  end;
  Label5.Caption := IntToStr(nom) + '/' + IntToStr(den);
end;
end.

```

Рис. 2.52. Програма додавання двох звичайних дробів з використанням функції MCD

Якщо обчислення НСД оформити як функцію, то програма виглядатиме так, як показано на рисунку 2.52.

Зверніть увагу не тільки на відмінності в оформленні процедури і функції, а й на відмінності в їх виклику та використанні в програмах, наведених на рисунках 2.50 і 2.52 (табл. 2.3):

Таблиця 2.3. Відмінності у виклику процедури і функції

Використання процедури MCD	Використання функції MCD
MCD (den1, den2, z); den := den1*den2/z;	den := den1*den2/MCD (den1, den2);
MCD (den, nom, k);	k := MCD (den, nom);

Процедури користувача і функції користувача називаються **підпрограмами користувача**.

Стандартні процедури та функції в Delphi

У мові програмування **Delphi** можна використовувати великий набір стандартних процедур і функцій, призначених для виконання різноманітних операцій над числами, текстами, елементами керування, графічними об'єктами, файлами та ін.

Деякі з них ви вже використовували, наприклад функцію **StrToInt** для переведення тексту в ціле число, функцію **FloatToStr** для переведення дійсного числа в текст, функцію **sqr** для обчислення арифметичного квадратного кореня з дійсного числа.

Розглянемо ще кілька стандартних математичних функцій та їхнє призначення (табл. 2.4).

Таблиця 2.4. Приклади стандартних математичних функцій у Delphi

Функція	Кількість аргументів та їх типи	Результат	Тип результату
Abs	1, цілий або дійсний	Модуль числа	Той самий, що й в аргументу
Arctan	1, дійсний	Арктангенс числа (в радіанах)	Дійсний
Cos	1, дійсний	Косинус числа	Дійсний
Exp	1, дійсний	Експонента числа (e^x)	Дійсний
Frac	1, дійсний	Дробова частина числа	Дійсний
Int	1, дійсний	Ціла частина числа	Дійсний
Ln	1, дійсний	Натуральний логарифм числа	Дійсний
Pi	0	Число π	Дійсний
Power	2, дійсні	I аргумент у степені II аргумент	Дійсний
Random	0	Випадкове число від 0 до 1	Дійсний
Random	1, цілий	Випадкове ціле число від 0 до значення аргументу – 1	Цілий
Sin	1, дійсний	Синус числа	Дійсний
Sqr	1, дійсний	Квадрат числа	Дійсний

Звертаємо вашу увагу:

1. При виклику функції без аргументів потрібно вказувати лише її ім'я, наприклад $x := \text{Pi}$.
2. Перед першим використанням функції **Random** потрібно включити до тексту проекту команду виклику процедури **Randomize** без аргументів і без результатів, яка запустить генератор випадкових чисел.
3. Для використання функції **Power** потрібно в рядок **Uses** тексту проекту додати текст **Math**. Після цього до проекту підключиться додаткова бібліотека стандартних процедур та функцій і стане можливим використання ще цілого ряду математичних та інших процедур і функцій, з якими ви можете ознайомитися в **Довідці**.






Перевірте себе

- 1°. Що таке процедури користувача? Якими вони можуть бути?
- 2°. У яких випадках доцільно використовувати процедури користувача? У чому полягають переваги їхнього використання?
- 3°. Який загальний вигляд процедур користувача з аргументами та з результатами?
- 4°. Що таке аргументи процедури; результати процедури?
- 5°. Який вигляд має команда виклику процедури з аргументами і результатами?
- 6°. Що таке формальні параметри процедури і фактичні параметри процедури? Поясніть відмінність між ними.
- 7°. Що таке функція користувача?
- 8°. Чим функція користувача відрізняється від процедури користувача?
- 9*. Який загальний вигляд функції?
- 10°. Які ви знаєте стандартні процедури і функції в **Delphi**? Охарактеризуйте їх.
- 11*. Знайдіть у **Довідці** інформацію про процедури без аргументів і без результатів; з аргументами і без результатів. Наведіть приклади їхнього використання.



Виконайте завдання

- 1°. Запишіть заголовок процедури з іменем **stepin**, аргументами якої є змінна a дійсного типу і змінна b цілого типу, а результатом – змінна c дійсного типу, і приклад команди її виклику.
- 2°. Запишіть процедуру з іменем **simp**, яка здійснює перевірку, чи є дане натуральне число простим і повертає як результат значення змінної f логічного типу ознаку того, чи є число простим, і значення змінної k цілого типу – кількість дільників даного числа.
- 3°. Запишіть процедуру з іменем **square**, яка виконує обчислення площі трикутника за відомими довжинами його сторін, значення яких передаються до процедури як аргументи. Для обчислення площі використовуйте формулу Герона.
- 4°. Створіть проект, у якому для двох заданих на площині точок з цілими координатами визначається, чи лежать вони в одній координатній чверті. Використайте процедуру для визначення координатної чверті, у якій розміщена точка із заданими координатами. Створіть у власній папці папку **Проект 2.10.4** і збережіть у ній проект.
- 5°. Запишіть заголовок функції з іменем **length**, аргументами якої є чотири змінні цілого типу, а результат має дійсний тип, і приклад команди її виклику.

- 6°. Запишіть функцію, яка виконує обчислення довжини відрізка за відомими координатами його кінців. Запишіть команду виклику цієї функції для обчислення довжини відрізка a з координатами кінців (0; 0) та (20; 30).
-  7°. Запишіть функцію, в якій обчислюється значення факторіалу заданого натурального числа n ($n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$). Запишіть команду використання цієї функції для обчислення суми факторіалів перших п'яти натуральних чисел.
- 8°. Створіть проект, у якому можна знайти максимальне із чотирьох заданих різних цілих чисел. Використайте функцію для знаходження більшого з двох заданих цілих чисел. Створіть у власній папці папку **Проект 2.10.8** і збережіть у ній проект.
- 9°. Створіть проект, у якому для двох заданих цілих чисел визначається, яке з них має більшу суму цифр. Використайте функцію для знаходження суми цифр заданого цілого числа. Створіть у власній папці папку **Проект 2.10.9** і збережіть у ній проект.
-  10°. Створіть проект, у якому для двох заданих цілих чисел визначається, яке з них має більшу суму дільників. Використайте функцію для знаходження суми дільників заданого цілого числа. Створіть у папці **Мої документи** папку **Проект 2.10.10** і збережіть у ній проект.
- 11*. Створіть проект, у якому для заданого цілого числа визначається, чи є одночасно простим саме число і число, записане тими самими цифрами, взятими у зворотному порядку. Використайте функцію для визначення того, чи є число простим. Створіть у власній папці папку **Проект 2.10.11** і збережіть у ній проект.
-  12°. Створіть проект, у якому обчислюється сума $\sqrt{1} + \sqrt{2} + \sqrt{3} + \sqrt{4} + \sqrt{5} + \dots + \sqrt{n}$. Створіть у власній папці папку **Проект 2.10.12** і збережіть у ній проект.

Практична робота № 10. Використання підпрограм

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте середовище розробки **Turbo Delphi 2006**.
2. Створіть проект, у якому двома способами обчислюється площа опуклого чотирикутника:
 - 1) за відомими довжинами сторін AB , BC , CD , DA та довжиною діагоналі AC . Використайте процедуру обчислення площі трикутника за формулою Герона. Складіть обробник події **OnClick** першої кнопки, виконання якого приведе до виведення розв'язку задачі у вікно повідомлення;
 - 2) за відомими координатами вершин чотирикутника, поданими у порядку обходу. Використайте функцію обчислення довжини відрізка за координатами його кінців. Складіть обробник події **OnClick** другої кнопки, виконання якого приведе до виведення розв'язку задачі в окремий напис.
3. Створіть у власній папці папку **Практична 10** і збережіть у ній проект.

2.11. Одновимірні масиви



1. Що визначає тип змінної? Які типи змінних вам відомі?
2. Який загальний вигляд команд розгалуження в **Delphi**? Як вони виконуються?
3. Який загальний вигляд команди циклу з лічильником у **Delphi**? Як вона виконується?
4. Який загальний вигляд команди циклу з передумовою в **Delphi**? Як вона виконується?
5. Де трапляються впорядковані набори даних? У чому зручність їхнього використання?

Поняття одновимірного масиву

У попередніх пунктах розглядали задачі, в яких використовувалася невелика кількість змінних. Але існує багато задач, у яких потрібно опрацьовувати значення досить великої кількості змінних.

Наприклад, якщо потрібно скласти програму для знаходження середньої маси 100 учнів 11-х класів, то очевидно потрібно використати 100 змінних, значенням кожної з яких буде маса відповідного одинадцятикласника, знайти суму значень цих змінних і поділити її на 100.

Якщо використати 100 змінних з довільними іменами, наприклад *a*, *b*, *c*, ..., *z*, *aa*, *ab*, *ac*, ..., *az*, *ba*, *bb*, *bc*, ..., *bz*, *ca*, *cb*, *cc*, ..., *cv* (як в електронних таблицях), то команда знаходження суми їхніх значень міститиме в правій своїй частині 100 доданків, або потрібно включити до тексту програми 100 команд, які по черзі додають до значення змінної *summa* значення маси наступного одинадцятикласника:

```
summa := a;
summa := summa + b;
summa := summa + c;
...
summa := summa + cv;
```

Усе це робить текст програми громіздким і незручним, зважаючи на те, що кількість учнів може бути ще більшою або взагалі довільною.

Для розв'язування задач, у яких опрацьовуються значення великої кількості змінних, можна піти таким шляхом: **розглядати всі ці змінні як елементи деякого впорядкованого набору змінних**. Цей набір змінних має ім'я, а ім'я кожної змінної набору складається з імені набору та порядкового номера цієї змінної в наборі. Такому набору даних можна надати загальне ім'я, а кожна окрема змінна визначатиметься наданим загальним іменем і своїм порядковим номером у цьому наборі.



Впорядкований набір змінних одного типу називається масивом. Кожна змінна, що входить до масиву, називається елементом масиву.

Масив, у якому кожний елемент визначається тільки одним порядковим номером, називається **одновимірним**.

Ім'я кожного елемента одновимірного масиву складається з імені масиву й узятото у квадратні дужки порядкового номера цього елемента в масиві.

Так, для прикладу, розглянутого вище, замість 100 окремих змінних варто розглядати масив, що складається зі 100 елементів. Якщо надати ім'я масиву *a*, то імена змінних – елементів цього масиву будуть такі: *a[1]*, *a[2]*, *a[3]*, ..., *a[100]*. Кожна з цих змінних матиме тип **Real**.

Тоді, щоб знайти суму значень усіх цих 100 змінних, достатньо записати такі команди:

```
summa := 0;
for i := 1 to 100 do
  summa := summa + a[i];
```

Звертаємо вашу увагу, що номер елемента масиву може задаватися як числом, так і значенням виразу, що набуває цілочисельного значення.

Виконання цього фрагмента програми відбуватиметься так:

Команда	Результат виконання
<i>summa</i> := 0	<i>summa</i> = 0
<i>i</i> := 1	<i>i</i> = 1
<i>i</i> ≤ 100	(1 ≤ 100) = <i>true</i>
<i>summa</i> := <i>summa</i> + <i>a[i]</i>	<i>summa</i> = 0 + <i>a[1]</i> = <i>a[1]</i>
<i>i</i> := <i>i</i> + 1	<i>i</i> = 2
<i>i</i> ≤ 100	(2 ≤ 100) = <i>true</i>
<i>summa</i> := <i>summa</i> + <i>a[i]</i>	<i>summa</i> = <i>a[1]</i> + <i>a[2]</i>
<i>i</i> := <i>i</i> + 1	<i>i</i> = 3
<i>i</i> ≤ 100	(3 ≤ 100) = <i>true</i>
<i>summa</i> := <i>summa</i> + <i>a[i]</i>	<i>summa</i> = <i>a[1]</i> + <i>a[2]</i> + <i>a[3]</i>
...	...
<i>i</i> := <i>i</i> + 1	<i>i</i> = 100
<i>i</i> ≤ 100	(100 ≤ 100) = <i>true</i>
<i>summa</i> := <i>summa</i> + <i>a[i]</i>	<i>summa</i> = <i>a[1]</i> + <i>a[2]</i> + <i>a[3]</i> + <i>a[4]</i> + ... + <i>a[100]</i>
<i>i</i> := <i>i</i> + 1	<i>i</i> = 101
<i>i</i> ≤ 100	(101 ≤ 100) = <i>false</i>

У результаті виконання наведеного фрагмента програми значення змінної ***summa*** дорівнюватиме сумі значень усіх 100 змінних – елементів одновимірного масиву, тобто сумарній масі даних 100 учнів.

У тексті програми мовою **Delphi** у рядку **var** змінна *a* типу **одновимірний масив** оголошується так: **var a: array [1..100] of Real** (англ. *array* – масив). У квадратних дужках указується **діапазон номерів елементів масиву**: від 1 до 100. За цим записом компілятор виділяє в пам'яті комп'ютера 100 ділянок під кожен зі 100 змінних – елементів масиву. Розмір кожної ділянки залежить від типу елементів масиву.

Звертаємо вашу увагу, що інколи зручно нумерувати елементи масиву, починаючи не з першого номера. Так, наприклад, для змінних, значення яких – кількість населення України в період з 1990 по 2011 роки можна використати масив **a: array [1990..2011] of Integer**. Взагалі, номери елементів одновимірного масиву можуть буди елементами довільного діапазону цілих чисел.

Введення та виведення значень елементів одновимірного масиву

Для введення і виведення значень елементів одновимірного масиву можна скористатися **багаторядковим полем** (компонент типу **TMemo**), який можна розмістити на формі так само, як й інші компоненти (рис. 2.53).

Серед властивостей цього компонента є багато таких, які мали раніше розглянуті нами компоненти: **Name**, **Color**, **Font**, **Enabled**, **Visible**, **Height**, **Width**, **Top**, **Left** та ін. Серед властивостей, яких не мали попередні компоненти, розглянемо такі:

- **ScrollBars** (англ. *scroll* – прокручування, *bars* – прямокутні смуги) – значення цієї властивості визначає наявність чи відсутність на цьому компоненті **смуг прокручування** (горизонтальної – *ssHorizontal*, вертикальної – *ssVertical*, обох – *ssBoth*) для перегляду тексту, який виходить за границі компонента;
- **ReadOnly** (англ. *read* – читати, *only* – тільки) – значення цієї властивості визначає можливість редагування рядків компонента: *False* – можна редагувати, *True* – не можна (тільки для читання);
- **Lines** (англ. *lines* – лінії, рядки) – значенням цієї властивості є набір значень, розташованих у рядках цього компонента; кожний рядок має номер, нумерація рядків починається з 0;
- **Lines.Count** (англ. *count* – підрахунок) – значення цієї властивості дорівнює кількості заповнених даними рядків компонента; *зазначимо*, що ця властивість відсутня у списку властивостей вікна **Object Inspector**.

Розглянемо задачу, в якій проілюструємо, як увести елементи одновимірного масиву з багаторядкового поля і як вивести елементи одновимірного масиву в багаторядкове поле.

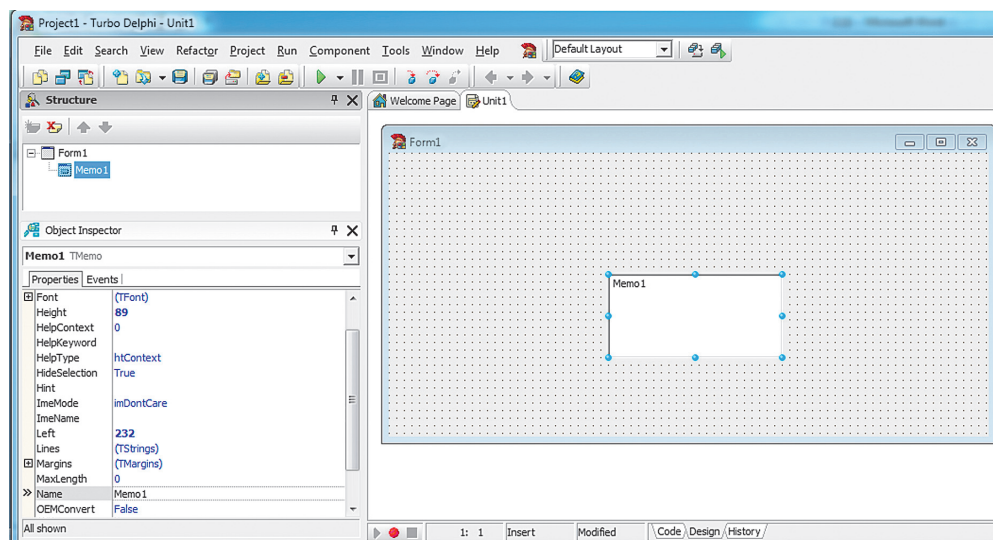


Рис. 2.53. Багаторядкове поле (компонент типу **TMemo**) і його властивості

Задача 1. Сформувати одновимірний масив з десяти дійсних чисел, які вводяться в рядки багаторядкового поля, піднести ці числа до квадрата і вивести ці квадрати в інше багаторядкове поле.

Розмістимо на формі два багаторядкових поля з іменами *Memo1* і *Memo2*, розташуємо їх на формі одне поруч з іншим та встановимо для них значення властивостей: **Height** = 150, **ScrollBars** = *ssVertical*, **ReadOnly** = *False*. Розмістимо на формі також кнопку, встановимо для неї **Caption** = 'До квадрата' і створимо процедуру, пов'язану з подією **OnClick** для цієї кнопки.

На рисунку 2.54 наведено текст цієї процедури і вікно її виконання. У тексті процедури використано два стандартні методи, що пов'язані з багаторядковим полем: процедура **Clear** (англ. *clear* – очищувати), що очищує всі рядки цього компонента, і процедура **Append** (англ. *append* – додавати), що додає до багаторядкового поля новий рядок і виводить у нього значення параметра, вказаного в дужках.

Звертаємо вашу увагу: в наведеному на рисунку 2.54 прикладі процедури розв'язування задачі 1 уведення значень елементів, піднесення їх до квадрата і виведення нових значень елементів масиву організовано трьома окремими циклами лише для кращого розуміння. Очевидно, що усі ці операції можна було б організувати в одному циклі.

Після запуску проекту на виконання у вікні, що відкрилося, у першому багаторядковому полі (на рис. 2.54 зліва) потрібно видалити текст *Memo1*, який знаходився там за замовчуванням, і ввести в кожний рядок по одному числу, які й будуть значеннями відповідних елементів масиву. Після цього потрібно вибрати кнопку і в результаті виконання процедури з чисел у рядках першого багаторядкового поля буде сформований одновимірний масив, потім значення елементів цього масиву будуть піднесені до квадрата і нові значення будуть виведені в рядки другого багаторядкового поля.

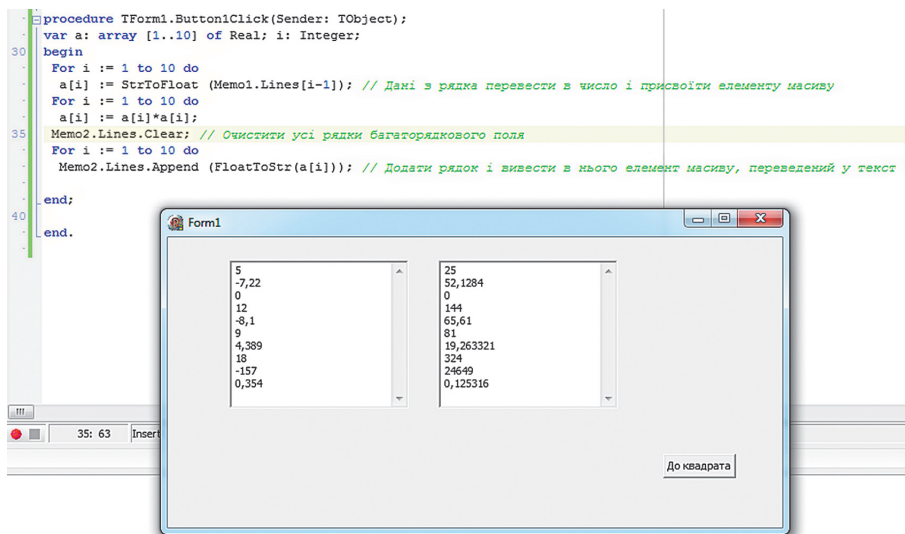


Рис. 2.54. Процедура введення значень елементів масиву, піднесення їх до квадрата і виведення нових значень елементів масиву

Можна увести числа в багаторядкове поле до запуску проекту, змінюючи значення властивості **Lines** цього поля. Це зручно робити, наприклад, на етапі налагодження програми.

Для цього потрібно:

1. Вибрати багаторядкове поле.
2. Вибрати кнопку в кінці рядка **Lines** вікна **Object Inspector**.
3. У вікні **String List Editor** (англ. *string* – рядок тексту, *list* – список, *editor* – редактор), що відкрилося, видалити текст першого рядка і ввести дані десяти чисел, натискаючи після кожного клавішу **Enter**.
4. Вибрати кнопку **OK**.

Тоді при збереженні проекту ці значення також будуть збережені й автоматично з'являться в цьому полі після запуску проекту на виконання.

Деякі задачі опрацювання одновимірних масивів

Розглянемо кілька задач на опрацювання значень елементів одновимірного масиву. В усіх задачах, розглянутих нижче, будемо вводити значення елементів масиву з багаторядкового поля, а виводити результат у багаторядкове поле, якщо результатом є значення елементів одновимірного масиву, або в напис; створюватимемо процедури, пов'язані з подією **OnClick** для кнопки. За потреби ви зможете аналогічно створити процедури, пов'язані з іншими подіями або з іншими об'єктами.

Будемо розглядати всі масиви з десяти чисел. Якщо кількість елементів масиву інша, то потрібно буде внести до текстів процедур відповідні зміни.

Задача 2. Визначити, скільки разів трапляється дане дійсне число серед значень елементів даного масиву з десяти дійсних чисел.

Очевидно, що для розв'язування цієї задачі потрібно переглянути послідовно значення всіх елементів масиву, порівняти кожне з них з даним числом, і якщо деяке з них дорівнює цьому числу, то збільшити лічильник таких елементів масиву на 1.

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i, k: Integer; x: Real;
begin
  x := StrToFloat (Edit1.Text); {Уведення числа, з яким потрібно порівнювати
  значення елементів масиву}
  k := 0; {Лічильник кількості елементів масиву, що дорівнюють даному числу
  x – їх поки що не траплялося жодного}
  for i := 1 to 10 do
    begin
      a[i] := StrToFloat (Memo1.Lines[i-1]); {Уведення значень елементів
      одновимірного масиву з багаторядкового поля}
      if a[i] = x then k := k + 1; {Якщо значення чергового елемента масиву
      дорівнює заданому числу, то значення лічильника збільшується на 1}
    end;
  Label1.Caption := IntToStr (k) + ' разів' {Виведення результату в напис з
  додаванням пояснювального тексту}
end;
```

Звертаємо вашу увагу: якщо не включити перед початком циклу команду $k := 0$, то перше виконання команди $k := k + 1$ може виявитися некоректним.

Задача 3. Визначити, чи трапляється дане дійсне число серед значень елементів даного масиву з десяти дійсних чисел.

Можна було б розв'язати цю задачу аналогічно попередній: визначити, скільки елементів масиву дорівнюють даному числу, і якщо ця кількість 0, то даного числа в масиві немає, а якщо більше 0, то є. Але такий метод розв'язування є нераціональним. Уявіть собі, що дане число є значенням уже першого елемента масиву. Тоді всі інші елементи масиву переглядати не потрібно, а попередній метод розв'язування передбачає обов'язковий перегляд усіх елементів масиву.

Для реалізації раціонального методу розв'язування цієї задачі використаємо змінну логічного типу f і присвоїмо їй початкове значення *false* (дане число в масиві поки ще не траплялося). Порівнюватимемо послідовно значення елементів масиву з даним числом і якщо один з них дорівнює даному числу, то змінимо значення змінної f на *true* і перервемо виконання циклу командою **break** (англ. *break* – переривати). Виведення результату реалізуємо залежно від значення змінної f .

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i: Integer; x: Real; f: Boolean;
begin
    for i := 1 to 10 do
        a[i] := StrToFloat (Memo1.Lines[i-1]);
        x := StrToFloat (Edit1.Text);
        f := false; // Дане число в масиві поки що не траплялося
    for i := 1 to 10 do
        if a[i] = x
        then begin
            f := true; // Дане число в масиві трапилося
            break; {Перериваємо виконання циклу, бо число трапилося в
масиві}
        end;
    if f
    then Label1.Caption := 'трапляється'
    else Label1.Caption := 'не трапляється';
end;

```

Задача 4. У даному масиві з десяти дійсних чисел визначити найбільше значення.

Спочатку вважатимемо, що значення першого елемента масиву і є найбільшим серед усіх значень елементів масиву. Після чого послідовно переглядатимемо всі інші елементи масиву і якщо трапиться значення, яке більше за те, яке ми на даний момент вважаємо найбільшим, то зробимо це значення найбільшим.

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i: Integer; max: Real;
begin
    for i := 1 to 10 do
        a[i] := StrToFloat (Memo1.Lines[i-1]);
        max := a[1]; // Вважаємо, що найбільший елемент – перший
    for i := 2 to 10 do
        if a[i] > max then max := a[i]; {Замінюємо значення найбільшого еле-
мента}

```

```
Label1.Caption := FloatToStr (max) + ' – найбільше значення';
end;
```

Задача 5. У даному масиві з десяти різних дійсних чисел визначити найбільше та найменше значення й поміняти їх місцями.

Оскільки потрібно поміняти місцями найбільше і найменше значення серед значень елементів масиву, то потрібно визначити не тільки самі ці значення, а ще й порядковий номер елемента масиву, значення якого найбільше, і порядковий номер елемента масиву, значення якого найменше. Спробуємо визначити все це, переглядаючи елементи масиву лише один раз.

Спочатку вважатимемо, що значення першого елемента масиву – і найбільше, і найменше, а номер найбільшого і найменшого елемента масиву 1. Після чого послідовно переглядатимемо всі інші елементи масиву. Якщо значення чергового елемента масиву більше за те значення, яке ми на той момент вважаємо найбільшим, то замінюємо найбільше значення серед уже переглянутих елементів масиву і номер найбільшого елемента. Якщо ні, то може це значення менше ніж те, яке ми поки що вважаємо найменшим. І якщо це так, то замінюємо найменше значення серед уже переглянутих елементів масиву і номер найменшого елемента.

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i, nmax, nmin: Integer; max, min: Real;
begin
  for i := 1 to 10 do
    a[i] := StrToFloat (Memo1.Lines[i-1]);
    max := a[1]; nmax := 1; {Значення першого елемента приймається за
найбільше}
    min := a[1]; nmin := 1; {Значення першого елемента приймається за
найменше}
    for i := 2 to 10 do
      if a[i] > max {Порівняння значення чергового елемента з найбільшим на
цей момент}
      then
        begin
          max := a[i]; // Замінюємо значення найбільшого елемента
          nmax := i; // Замінюємо номер найбільшого елемента
        end
      else
        if a[i] < min {Порівняння значення чергового елемента з найменшим
на цей момент}
        then
          begin
            min := a[i]; // Замінюємо значення найменшого елемента
            nmin := i; // Замінюємо номер найменшого елемента
          end;
        a[nmax] := min; a[nmin] := max; {Обмін значеннями найбільшого і най-
меншого елементів}
      for i := 1 to 10 do
        Memo1.Lines.Append (FloatToStr(a[i])); {Виведення значень елемен-
тів масиву з переставленими найбільшим і найменшим значеннями}
      end;
```

Задача 6. Визначити елементи даного масиву з десяти дійсних чисел, значення яких більші за дане дійсне число.

Очевидно, що як і в задачі 2, потрібно переглянути послідовно значення всіх елементів масиву і з'ясувати, які з них більші, ніж задане число. Виникає питання, де зберігати значення цих елементів масиву. Адже залежно від значень елементів масиву і від даного числа, кількість таких елементів може бути різною: від 0 до 10.

Було б доцільно заносити значення шуканих елементів до іншого масиву, але нам заздалегідь невідома кількість його елементів.

У мові програмування **Delphi** є можливість використовувати масиви, кількість елементів яких заздалегідь невідома й яка може змінюватися в ході виконання проекту. Такі масиви називаються **динамічними**. На відміну від них, масиви, кількість елементів яких відома заздалегідь і яка не змінюється в ході виконання проекту, називаються **статичними**.

Звертаємо вашу увагу, що нумерація елементів динамічного масиву завжди розпочинається з 0.

У рядку **var** динамічний масив **b**, елементами якого будуть дійсні числа, описується так: **var b: array of Real;**

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; b: array of Real; i, k: Integer; x: Real;
begin
    for i := 1 to 10 do
        a[i] := StrToFloat (Memo1.Lines[i-1]);
        x := StrToFloat (Edit1.Text);
        k := 0; {Лічильник кількості елементів масиву b (спочатку в масиві b немає елементів)}
        for i := 1 to 10 do
            if a[i] > x
            then begin
                k := k+1; {Збільшення на 1 значення лічильника кількості елементів масиву b}
                SetLength (b, k); {Установлення нового розміру (кількості елементів) динамічного масиву b}
                b[k-1] := a[i]; {Занесення відповідного елемента масиву a до масиву b}
            end;
        Memo2.Lines.Clear;
        if k = 0
        then Memo2.Lines[0] := 'Таких чисел не існує'
        else
            for i := 1 to k do
                Memo2.Lines.Append (FloatToStr(b[i-1]));
        end;

```

SetLength (англ. *set* – встановлювати, *length* – довжина).

Використовуючи динамічні масиви, можна створювати програми для довільної кількості чисел, уведених у рядки багаторядкового поля. У таких випадках потрібно ввести дані з цього багаторядкового поля в динамічний масив **a**, наприклад так:

```

k := Memo1.Lines.Count; // Визначається кількість заповнених рядків поля
SetLength (a, k); {Установлюється відповідна кількість елементів динамічного масиву a і виділяється для його елементів відповідна ділянка пам'яті}

```

For i := 0 to k-1 do a[i] := StrToFloat (Memo1.Lines[i]); {Уведення даних з рядків поля до масиву}

Далі дані з динамічного масиву можна опрацьовувати різноманітними методами.

Сортування одновимірного масиву та пошук даного числа у впорядкованому масиві

Багато програм на опрацювання значень елементів одновимірного масиву працюватимуть значно швидше, якщо значення його елементів будуть упорядковані за зростанням або за спаданням. Це стосується перш за все задач пошуку тих значень елементів масиву, які відповідають певним умовам.

Існує понад десять різноманітних методів сортування одновимірного масиву. Одні з них виконуються швидше, інші – повільніше, одні – більш складні за своєю логічною структурою, інші – простіші. Ми розглянемо один з методів сортування одновимірного масиву – **метод вибору**.

Пояснимо сутність цього методу на прикладі. Нехай нам потрібно впорядкувати за зростанням такий одновимірний масив із шести елементів (табл. 2.5, рядок 0):

Таблиця 2.5. Упорядкування масиву методом вибору

№	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
0	23	15	7	2	6	4
1	2	15	7	23	6	4
2	2	4	7	23	6	15
3	2	4	6	23	7	15
4	2	4	6	7	23	15
5	2	4	6	7	15	23

На першому кроці визначимо значення найменшого елемента в усьому масиві ($a[4] = 2$) й обміняємо його зі значенням першого елемента. Одержуємо масив у рядку 1, в якому найменший елемент зайняв своє місце. На другому кроці визначимо значення найменшого елемента серед усіх елементів масиву, крім першого, ($a[6] = 4$) й обміняємо його зі значенням другого елемента. Одержуємо масив у рядку 2, в якому перші 2 елементи зайняли своє місце. На третьому кроці визначимо значення найменшого елемента серед усіх елементів масиву, крім перших двох, ($a[5] = 6$) та обміняємо його зі значенням третього елемента. Одержуємо масив у рядку 3, в якому перші 3 елементи зайняли своє місце. Повторивши аналогічні дії ще 2 рази, одержуємо масив, впорядкований за зростанням.

Звертаємо вашу увагу: хоча масив має шість елементів, достатньо 5 разів знайти найменше значення елементів з ще неупорядкованої частини масиву й обміняти його місцями зі значенням першого з ще неупорядкованої частини масиву елемента. На останньому кроці не лише 5-й, а й 6-й елемент масиву займає своє місце у впорядкованій частині масиву, і таким чином увесь масив стає впорядкованим.

З пояснень суті цього методу з'ясовується зміст його назви: адже на кожному кроці вибирається найменший елемент серед ще неупорядкова-

ної частини масиву, і його значення обмінюється зі значенням елемента, першого серед ще невпорядкованої частини масиву.

Відповідна процедура, в якій реалізується цей метод, виглядає так:

```
procedure TForm1.Button2Click (Sender: TObject);
var a: array [1..10] of integer; i, j, min, nmin: integer;
begin
  for i := 1 to 10 do
    a[i] := StrToInt (Memo1.Lines[i-1]);
  for i := 1 to 9 do
    begin
      min := a[i]; nmin := i;
      for j := i+1 to 10 do
        if a[j] < min then
          begin
            min := a[j]; nmin := j;
          end;
      a[nmin] := a[i];
      a[i] := min;
    end;
  Memo2.Lines.Clear;
  for i := 1 to 10 do
    Memo2.Lines.Append (IntToStr (a[i]))
  end;
```

Продемонструємо той факт, що пошук потрібного значення серед значень елементів масиву (задача 3) відбуватиметься значно швидше, якщо масив упорядкований.

Якщо масив невпорядкований, то задачу можна розв'язати методом, розглянутим вище. При реалізації цього методу, якщо даного числа в масиві немає, то для з'ясування цього факту потрібно буде порівняти дане число зі значеннями **усіх елементів масиву**. І якщо кількість елементів масиву велика, наприклад 1000, то й число порівнянь (а отже і час виконання проекту) буде відповідним.

Якщо ж масив впорядкований, то можна з'ясувати, чи є дане число в масиві, іншим способом, значно ефективнішим. Пояснимо його на прикладі. Нехай маємо впорядкований за зростанням масив з десяти чисел: 2, 5, 8, 12, 13, 16, 17, 20, 22, 30 і деяке дане число x . Порівняємо це число із значенням елемента масиву, який знаходиться посередині масиву (з числом 13). Якщо дане число x дорівнює 13, то воно в масиві є, якщо ні, то з'ясуємо, чи більше дане число за 13 ($x > 13$). Якщо так, то його потрібно шукати тільки серед правої половини масиву, якщо ні, то тільки серед лівої половини масиву. Таким чином область пошуку звужується вдвічі. На наступному кроці робимо так само: порівнюємо дане число із значенням елемента масиву, який знаходиться посередині тієї частини масиву, яка залишилася для пошуку. Знову або дане число дорівнює значенню цього елемента масиву, або залишаємо для пошуку або ліву, або праву половину тієї частини масиву, що залишилася, тобто область пошуку знову зменшується вдвічі.

Такий метод пошуку є ефективнішим, ніж попередній, бо значно швидше приводить до результату, особливо для великих N (максимум за $\lceil \log_2 N \rceil + 1$ кроків, де N – кількість елементів у масиві, а квадратними дужками тут позначена ціла частина числа).

Такий метод пошуку заданого числа в одновимірному масиві називається **методом половинного (бінарного) пошуку**.

Відповідна процедура, в якій реалізується цей метод для впорядкованого масиву з десяти цілих чисел, виглядає так:

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Integer; i, x, left, right, m: Integer; f: Boolean;
begin
    for i := 1 to 10 do
        a[i] := StrToInt (Memo1.Lines[i-1]);
        x := StrToInt (Edit1.Text);
        left := 1; {Початковий номер елемента тієї частини масиву, де відбува-
        тиметься пошук}
        right := 10; {Кінцевий номер елемента тієї частини масиву, де відбува-
        тиметься пошук}
        f := false; // Задане число в масиві поки що не знайдене
        while (left <= right) and not f do
            begin
                m := (left + right) div 2; {Номер елемента посередині тієї частини
                масиву, де далі продовжуватиметься пошук}
                if x > a[m]
                    then left := m+1 {Змінюється початковий номер елемента тієї
                    частини масиву, де відбуватиметься пошук}
                else if x < a[m]
                    then right := m-1 {Змінюється останній номер елемента
                    тієї частини масиву, де відбуватиметься пошук}
                else f := true; // Число в масиві знайшлося
            end;
        if f
            then Edit1.Text := 'Число в масиві є'
            else Edit1.Text := 'Числа в масиві немає';
    end;

```

Метод половинного поділу використовували в своїх логічних міркуваннях, класифікаціях, методах розв'язування задач ще давні греки. Вони називали цей метод **дихотомією** (грец. διχῆ – навпіл, τομή – поділ). Може й ви використовували цей метод у грі «Відгадай задумане число».



Перевірте себе

- 1°. Що таке одновимірний масив?
- 2°. З чого складається ім'я елемента масиву?
- 3°. Що може використовуватись як номер елемента масиву?
- 4*. При складанні проектів для розв'язування яких задач зручно використовувати масиви?
- 5°. Як записати змінну типу *одновимірний масив* у рядку **var**?
- 6°. Яким може бути діапазон номерів елементів масиву?
- 7°. Назвіть та охарактеризуйте відомі вам властивості багаторядкового поля.
- 8°. Назвіть і поясніть відомі вам стандартні методи багаторядкового поля.
- 9°. Як увести числа в багаторядкове поле до запуску проекту?
- 10°. Опишіть різні способи визначення, чи трапляється задане число серед значень елементів масиву.
- 11°. Опишіть послідовність дій для визначення найбільшого значення серед значень елементів масиву.

- 12°. Який масив називається динамічним? Як описується динамічний масив у рядку **var**?
- 13°. Поясніть, у чому полягають відмінності між статичним і динамічним масивами.
- 14°. Поясніть сутність методу вибору при сортуванні масиву.
- 15°. Поясніть сутність методу половинного (бінарного) пошуку даного числа в одновимірному масиві.
- 16*. Поясніть, у чому перевага пошуку заданого елемента у впорядкованому масиві порівняно з невпорядкованим. Наведіть приклади такого пошуку у вашій навчальній діяльності.



Виконайте завдання

- 1°. Складіть таблицю виконання фрагмента програми та визначте значення змінної **dob** після його завершення для масиву, значеннями елементів якого є числа 2; -3,5; 1,2; 0,25; -4:


```
dob := 1;
for i := 1 to 5 do
  dob := dob * a[i];
```
- 2°. Складіть таблицю виконання фрагмента програми та визначте значення змінної **sum** після його завершення для масиву, значеннями якого є числа 2; 0; 23; -14:


```
sum := 0;
for i := 1 to 4 do
  if a[i] mod 2 = 0 then sum := sum + a[i]*a[i];
```
- 3°. Запишіть рядок оголошення змінних типу одновимірний масив:
 - а) масив змінних цілого типу з іменем *x* і діапазоном номерів від 1 до 50;
 - б) масив змінних дійсного типу з іменем *mas* і діапазоном номерів від 10 до 35;
 - в) масив змінних цілого типу з іменем *tab* і діапазоном номерів від -7 до 12.
- 4°. Створіть проект, у якому потрібно сформувати масив із десяти цілих чисел, що знаходяться в рядках багаторядкового поля, знайдіть середнє арифметичне значення елементів масиву та виведіть результат у напис. Створіть у власній папці папку **Проект 2.11.4** і збережіть у ній проект.
- 5°. Створіть проект, у якому потрібно сформувати масив із дванадцяти дійсних чисел, що знаходяться в рядках багаторядкового поля, збільшіть значення кожного елемента масиву в 3 рази і виведіть нові значення в інше багаторядкове поле. Створіть у власній папці папку **Проект 2.11.5** і збережіть у ній проект.
- 6°. Створіть проект, у якому потрібно сформувати масив із дев'яти цілих чисел, що знаходяться в рядках багаторядкового поля, визначте, чи трапляються серед значень елементів цього масиву числа, кратні числу 3, і виведіть результат у напис. Створіть у власній папці папку **Проект 2.11.6** і збережіть у ній проект.
- 7°. Створіть проект, у якому потрібно сформувати масив із десяти дійсних чисел, що знаходяться в рядках багаторядкового поля, визначте найменше серед значень елементів цього масиву та обміняйте його місцями з першим елементом масиву. Отриманий масив виведіть в інше багаторядкове поле. Створіть у власній папці папку **Проект 2.11.7** і збережіть у ній проект.
- 8°. Створіть проект, у якому потрібно сформувати масив із дев'яти дійсних чисел, що знаходяться в рядках багаторядкового поля, визначте найбільше серед значень елементів цього масиву і обміняйте його місцями із значенням елемента масиву, що знаходиться посередині масиву.

- Отриманий масив виведіть в інше багаторядкове поле. Створіть у папці **Мої документи** папку **Проект 2.11.8** і збережіть у ній проект.
- 9*. Створіть проект, у якому потрібно сформуванати динамічний масив з дійсних чисел, що знаходяться в рядках багаторядкового поля, визначте середнє арифметичне значення елементів цього масиву і знайдіть елементи, які більші за середнє арифметичне значення. Для зберігання результатів використайте динамічний масив. Створіть у власній папці папку **Проект 2.11.9** і збережіть у ній проект.
- 10*. Створіть проект, у якому потрібно сформуванати масив з дійсних чисел, що знаходяться в рядках багаторядкового поля, відсортуйте його і знайдіть кількість різних значень елементів масиву. Створіть у власній папці папку **Проект 2.11.10** і збережіть у ній проект.



Практична робота № 11. Опрацювання одновимірних масивів

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте середовище розробки **Turbo Delphi 2006**.
2. Створіть проект для розв'язування задачі: *Відома середня температура кожного дня тижня. Визначте середню температуру за весь тиждень. З'ясуйте, скільки разів на тиждень температура була вище нуля. Відсортуйте значення температур за спаданням та з'ясуйте, чи була однаковою температура кілька днів на тиждень. У проекті:*
 - а) розмістіть на формі потрібні елементи керування: багаторядкові поля, написи, кнопки;
 - б) складіть обробник події **OnClick** першої кнопки, виконання якого приведе до виведення у напис середньої температури за тиждень;
 - в) складіть обробник події **OnClick** другої кнопки, виконання якого приведе до визначення, скільки разів на тиждень температура була вище нуля, та виведення результату у напис;
 - г) складіть обробник події **OnClick** третьої кнопки, виконання якого приведе до сортування значень температур і визначення, чи є в масиві однакові значення, виведення відсортованого масиву у багаторядкове поле, повідомлення про наявність однакових елементів у напис;
 - д) виконайте складені процедури і переконайтеся, що результати їх роботи правильні.
3. Створіть у власній папці папку **Практична 11** і збережіть у ній проект.

Розділ 3

БАЗИ ДАНИХ.

СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ

У цьому розділі ви дізнаєтеся про:

- бази даних, комп'ютерні бази даних, їх класифікацію та призначення;
- моделі даних, їх класифікацію;
- реляційну модель даних, її особливості та основні об'єкти;
- зв'язки в базах даних, їх класифікацію;
- модель «сутність–зв'язок», правила її побудови та відображення на базу даних;
- системи управління базами даних, їх призначення;
- систему управління базами даних **Access 2007** і можливості, які вона надає користувачу під час роботи з базами даних;
- створення, редагування та форматування таблиць, форм, запитів і звітів бази даних у середовищі СУБД **Access 2007**;
- пошук, сортування та фільтрування даних засобами СУБД **Access 2007**.

3.1. Бази даних. Модель «сутність–зв'язок»



1. Що таке інформаційна система? Назвіть складові інформаційної системи та їх призначення.
2. Що таке модель об'єкта? Які різновиди моделей ви знаєте?
3. Що таке предметна область? Наведіть приклади предметних областей.

Поняття про бази даних

Необхідність пошуку потрібної інформації у людини виникає повсякчас, незалежно від сфери її професійних інтересів: з якої платформи відправляється потяг на Хмельницький, як приготувати вареники з вишнями, яку будову має молекула води, скільки днів тривала Друга світова війна, чи справджується прикмета про чорну кішку, яка частота змінного електричного струму в побутовій електричній мережі, яке закінчення мають іменники третьої відміни в родовому відмінку однини та ін. Відповіді на частину з цих запитань людина може отримати зі своєї пам'яті, для отримання інших необхідно звернутися до інформаційної системи залізничного вокзалу, переглянути кулінарну книжку, довідник з хімії, фізики чи електротехніки, посібник з правопису тощо. Для полегшення пошуку потрібної інформації людство придумало багато засобів – універсальні енциклопедії та енциклопедії з предметних галузей, довідники й словники, довідкові бюро та інформаційні табло та ін.

Обсяги повідомлень, які накопичило людство, невпинно зростають. Так, при розкопках стародавнього міста шумерів Ур було знайдено понад 20 тисяч глиняних табличок з відомостями про звичаї давнього народу, його легенди та історичні події, що відбувалися понад 5 тисяч років тому. Знаменита Александрійська бібліотека, яка була заснована в Єгипті у III ст. до нової ери, за різними джерелами містила від 100 до 700 тисяч

рукописів. Сьогоднішні бібліотеки вражають обсягами різноманітних даних. Найбільшою у світі вважається Британська бібліотека в Лондоні, яка нараховує понад 150 млн одиниць зберігання, а найбільша бібліотека нашої країни – Національна бібліотека України імені В.І. Вернадського в Києві нараховує понад 15 млн одиниць зберігання.

Учені запевняють, що зберігання великих обсягів даних виправдано тільки за умови, якщо пошук потрібних даних здійснюється швидко і подаються вони в доступній для розуміння формі. Ці умови забезпечують сучасні технології зберігання даних. Основою цих технологій є комп'ютеризовані бази даних (БД).



База даних – це впорядкований за певними правилами набір взаємопов'язаних даних.



Перша в Україні комп'ютерна база даних була розроблена в ході робіт з проектування і експлуатації електронної обчислювальної машини «Київ» (1959 р.). ЕОМ була розроблена для обчислювального центру Академії наук УРСР Л.Н. Дашевським, К.Л. Ющенко, К.О. Шкарабарою, С.Б. Погребинським під науковим керівництвом Б.В. Гніденка та В.М. Глушкова.

Віктору Михайловичу Глушкову (1923–1982) (рис. 3.1) належить цикл робіт з використання ЕОМ в управлінні не тільки в межах одного підприємства чи організації, а всієї країни. Це він розробив проект **Загальнодержавної автоматизованої системи збирання та опрацювання інформації** для обліку, планування й управління економікою країни. Основу цієї системи повинні були складати бази даних або, як їх називав В.М. Глушков, «банки даних». У 1975 р. він казав: «Уже в найближчі десятиліття в ряді високорозвинених країн будуть створені так звані *національні банки даних*». Створення подібних банків даних, як і прогнозував В.М. Глушков, докорінно змінило роботу у науковця, інженера, журналіста.



Рис. 3.1. Віктор Михайлович Глушков

Поняття сутності, модель «сутність–зв'язок»

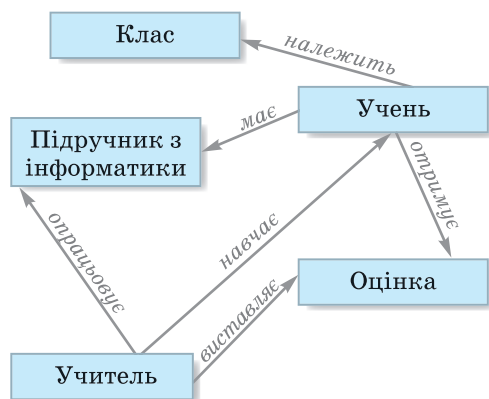


Рис. 3.2. Модель предметної області Інформатика в школі

На початку проектування баз даних, як правило, розробляється модель предметної області, для якої створюється ця БД. У ній указуються типи об'єктів, що будуть включені до бази даних, і зв'язки між ними. Для наочності таку модель можна подати у графічному вигляді.

Розглянемо приклад моделі предметної області **Інформатика в школі** (рис. 3.2).

У прямокутниках записані назви типів об'єктів (груп об'єктів) предметної області, стрілки вказу-

ють на зв'язки між ними, а написи біля стрілок описують ці зв'язки. **Учитель** заздалегідь *опрацьовує* **Підручник з інформатики** і організовує процес навчання учнів (*навчає*). Він же оцінює рівень навчальних досягнень учнів і виставляє **Оцінки**. **Учні** навчаються під керівництвом учителя та самостійно, *маючи* **Підручник з інформатики**. Вони також *належать* до певного **Класу** і *отримують* **Оцінки**.

Тип об'єкта предметної області називають **сутністю**. Сутностями розглянутої вище предметної області є **Учитель**, **Оцінка**, **Учень**, **Підручник з інформатики**, **Клас**. Сутності, по суті, є сукупностями однотипних об'єктів. Наприклад, сутність **Клас** може складатися з об'єктів **11-А**, **11-Б**, **11-В** тощо. Окремі об'єкти сутності називають **екземплярами сутності**. Сутність має як мінімум один екземпляр. Наприклад, сутність **Підручник з інформатики** має тільки один екземпляр сутності, якщо всі учні школи навчаються за одним і тим самим підручником з інформатики.



Сутність предметної області – це тип реального або уявного об'єкта предметної області.

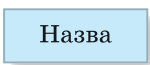




При подальшій формалізації моделі словесний опис зв'язків між сутностями замінюють на їх умовні позначення відповідного типу. За **множинністю** виділяють такі типи зв'язків:

- **один до одного** (позначається як **1:1**), коли одному екземпляру однієї сутності відповідає один екземпляр іншої сутності. Для наведеного прикладу такий зв'язок є між сутностями **Учитель** і **Підручник з інформатики**, якщо кожен учитель використовує тільки один підручник, який не використовують інші вчителі;
- **один до багатьох** (позначається як **1:∞** або **1:M**, де **M** – від англ. *Many* – багато), коли одному екземпляру однієї сутності може відповідати кілька екземплярів іншої сутності. Для наведеного прикладу такий зв'язок можливий між сутностями **Учитель** і **Учень**, коли кожен учитель інформатики здійснює навчання сукупності учнів (одного або кількох класів) і при цьому немає вчителів, які здійснюють навчання одних і тих самих учнів;
- **багато до одного** (позначається як **∞:1** або **M:1**), коли кільком екземплярам однієї сутності відповідає один екземпляр іншої сутності. Цей тип зв'язку є протилежним до зв'язку один до багатьох. Наприклад, якщо розглянути зв'язок між сутністю **Учень** і сутністю **Клас**, то тип зв'язку буде *багато до одного*, а якщо розглядати зв'язок між сутностями **Клас** і **Учень**, то тип зв'язку буде *один до багатьох*;
- **багато до багатьох** (позначається як **∞:∞** або **M:M**), коли кільком екземплярам однієї сутності можуть відповідати кілька екземплярів іншої сутності. У розглянутому прикладі такий зв'язок є між сутностями **Учень** та **Оцінки** – багато учнів отримують багато різних оцінок. Зв'язки між сутностями класифікують також *за повнотою*. За цією класифікацією виділяють зв'язки, в яких:
 - кожний екземпляр однієї сутності **обов'язково пов'язаний** з одним чи кількома екземплярами іншої сутності. Наприклад, зв'язок між сутностями **Клас** і **Учень** передбачає, що кожен учень належить до певного класу і кожний клас складається з певної групи учнів;
 - кожний екземпляр однієї сутності **не обов'язково пов'язаний** хоча б з одним екземпляром іншої сутності. Наприклад, між сутностями

Учень і Комп'ютер може бути встановлений зв'язок, який передбачає, що не кожен учень має власний комп'ютер.

Розглянуту нами модель предметної області називають моделлю «сутність–зв'язок», або **ER-моделлю**, або **ER-діаграмою** (англ. *Entity-Relationship* – сутність (об'єкт)–зв'язок (відношення)). Під час створення ER-моделі використовуються спеціальні позначення типів сутності, властивостей екземплярів сутностей, зв'язків та ін. Набір таких умовних позначень називають **нотацією** (англ. *notation* – позначення, зображення умовними знаками). У таблиці 3.1 подано основні елементи однієї з нотацій.

Таблиця 3.1. Основні елементи графічного подання моделі «сутність–зв'язок»

Умовне позначення	Назва фігури	Що позначає
	Прямокутник	Сутність предметної області
	Овал	Властивість екземплярів сутності
	Ромб	Зв'язок
	Суцільна лінія	Лінія обов'язкового зв'язку
	Штрихова лінія	Лінія необов'язкового зв'язку

Побудуємо модель «сутність–зв'язок» для предметної області **Кінотеатр**, у якій розглянемо дві сутності: **Глядач** і **Квиток**. Кожен глядач, який прийшов у кінотеатр, має квиток. Двох однакових квитків, як і двох однакових глядачів, не існує (рис. 3.3).

У сутності **Глядач** для даної моделі розглянемо дві властивості: *Прізвище* та *Ім'я*, а у сутності **Квиток** – п'ять: *Дата*, *Час*, *Ряд*, *Місце* та *Номер квитка*. Між сутностями **Глядач** і **Квиток** існує зв'язок – **Має**. За множинністю це зв'язок **один до одного**. На це вказують позначення над лініями зв'язку біля прямокутників сутностей. За повнотою цей зв'язок є

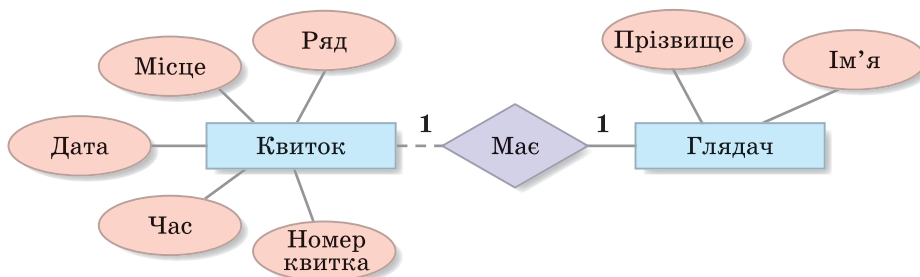


Рис. 3.3. Модель «сутність–зв'язок» предметної області **Кінотеатр**

обов'язковим для сутності **Глядач** (для того щоб бути присутнім на сеансі, глядач повинен мати квиток) і не обов'язковим для сутності **Квиток** (оскільки не всі квитки можуть бути розпродані, і тоді вони залишаться в касі та не потраплять до глядача).



Перевірте себе

- 1°. Для чого потрібне упорядковане зберігання даних?
- 2°. Які вимоги до зберігання даних висуваються сучасним інформатизованим суспільством?
- 3°. Що таке база даних? Де використовуються бази даних?
- 4°. Поясніть, що таке сутність.
- 5°. Що таке екземпляр сутності? Наведіть приклади.
- 6°. Назвіть типи зв'язків між сутностями предметної області.
- 7°. Опишіть особливості зв'язку «один до одного» між сутностями предметної області.
- 8°. Опишіть особливості зв'язку «один до багатьох» між сутностями предметної області, наведіть приклади таких зв'язків.
- 9°. Опишіть особливості зв'язку «багато до багатьох» між сутностями предметної області.
- 10°. На що вказує тип зв'язку при їх класифікації за повнотою?
- 11°. Що таке модель «сутність–зв'язок»? Що вона описує?
- 12°. Як називається набір умовних позначень під час графічного подання моделі «сутність–зв'язок»?
- 13°. Опишіть позначення, що використовуються під час графічного подання моделі «сутність–зв'язок».
- 14°. Яка роль В.М. Глушкова у розвитку напрямків інформатики, пов'язаних з використанням баз даних?



Виконайте завдання

- 1°. Розгляньте діаграму «сутність–зв'язок» (рис. 3.4). Визначте:
 - а) кількість і назву сутностей, зображених на діаграмі;
 - б) властивості кожної сутності;
 - в) види зв'язків для випадків:

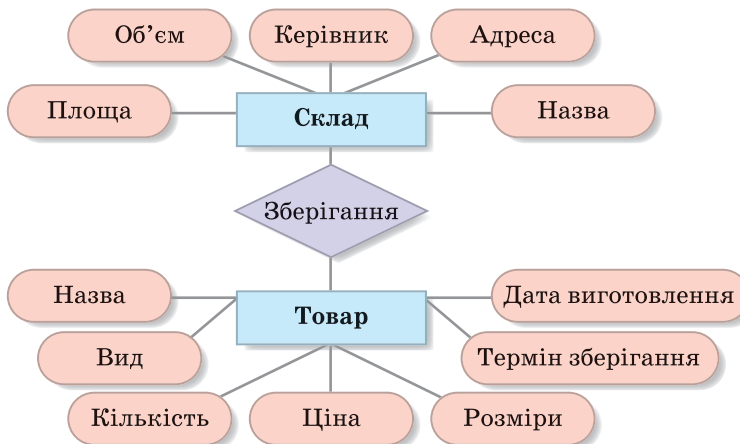


Рис. 3.4

- є кілька видів товару та є кілька складів, кожний товар зберігається на одному складі і на кожному складі зберігається тільки один вид товару;
 - є кілька видів товару та є кілька складів, на кожному складі є всі види товарів;
 - є тільки один вид товару і він зберігається на одному складі;
 - є кілька видів товару і всі вони зберігаються на одному складі;
 - є один вид товару і він зберігається на кількох складах.
- 2*. Розгляньте діаграму «сутність–зв’язок» (рис. 3.5). Визначте:
- кількість і назву сутностей, зображених на діаграмі;
 - властивості кожної сутності;
 - види зв’язків для випадків:
 - є кілька магазинів і є кілька моделей автомобілів, у кожному магазині є автомобілі тільки однієї моделі і кожна модель автомобіля продається тільки в одному магазині;
 - є тільки один магазин і він продає автомобілі однієї моделі;
 - є кілька магазинів і всі вони продають автомобілі однієї моделі;
 - є кілька магазинів і є кілька моделей автомобілів, у кожному магазині є автомобілі всіх моделей;
 - є один магазин і він продає автомобілі кількох моделей.

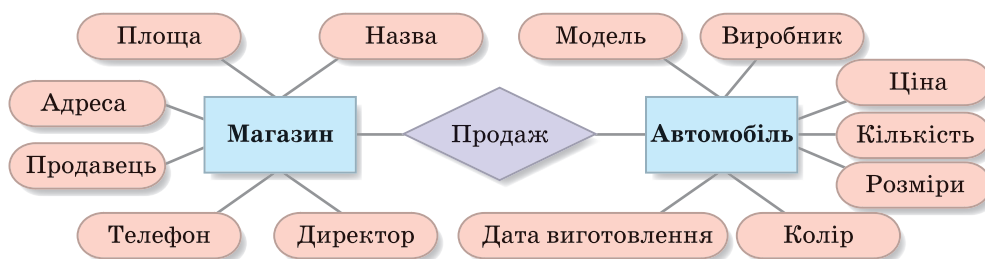


Рис. 3.5

- 3*. Розробіть модель «сутність–зв’язок» для створення таких баз даних:
- а) телефонний довідник учнів класу, який повинен містити дані про номери стаціонарного і мобільного телефонів учня, його прізвище та ім’я, адресу.
 - б) домашня аудіотека, яка повинна містити дані про номер диска у домашній аудіотеці, назву диска, виконавця, дату випуску, загальну тривалість музичних творів на диску.
- 4*. Підготуйте повідомлення за такими темами:
- а) Використання баз даних у перших українських ЕОМ.
 - б) Використання баз даних у перших автоматизованих системах управління (АСУ) підприємством, розроблених Інститутом кібернетики НАН України.

3.2. Моделі даних. Основні поняття реляційної бази даних



1. Що таке база даних? Для чого створюються бази даних?
2. Опишіть структуру файлової системи операційної системи **Windows XP**.
3. Як здійснюється пошук потрібних даних в операційній системі **Windows XP**, в Інтернеті?
4. Що таке модель «сутність–зв’язок»? Для чого вона використовується?

Поняття про моделі даних. Види моделей даних

Основним призначенням бази даних є гарантоване систематизоване збереження значних обсягів даних і надання доступу до них користувачеві або комп'ютерній програмі. Оскільки база даних є досить складним за своєю структурою об'єктом, то перед створенням вона заздалегідь проектується – створюється **модель бази даних**. Під час створення моделі бази даних використовують ту чи іншу модель даних. **Модель даних** має три складові:

1. **Структура даних** визначає спосіб організації даних, множину можливих типів даних (цілі числа, дійсні числа, текстові дані, мультимедійні дані, масиви чисел або текстів та ін.) і набір операцій, які можна виконати над даними конкретного типу.
2. **Засоби опрацювання даних** визначають набір команд (аналогічно до системи команд виконавця алгоритму), які забезпечують опрацювання даних залежно від способу їх організації.
3. **Обмеження цілісності** визначає вимоги для забезпечення правильності даних у будь-який момент часу. Розрізняють вимоги, що накладаються відповідним видом моделі даних, і вимоги, що встановлює користувач.

Найпоширенішими є такі види моделей даних: **ієрархічна, мережна, реляційна й об'єктно-реляційна**.

Ієрархічна (грец. *ιєραρχία* – священна влада) **модель даних** визначає організацію даних про об'єкти у вигляді дерева. В ієрархічній моделі структура даних передбачає, що у кожного об'єкта є тільки один об'єкт вищого рівня, якому він підпорядкований (батьківський), і може бути кілька підпорядкованих (нащадків). Виключення складає тільки найвищий за ієрархією об'єкт – у нього немає батьківського об'єкта.

Прикладом ієрархічної організації даних є файлова структура, що використовується під час розміщення даних на дисках, наприклад **FAT32** (ви ознайомилися з нею в курсі інформатики 9-го класу під час вивчення розділу «Системне програмне забезпечення»). У зазначеній файловій структурі на диску є один основний батьківський об'єкт (коренева папка), який містить дані про підпорядковані об'єкти – папки і файли. У свою чергу ці об'єкти можуть містити різноманітні дані, у тому числі і про підпорядковані їм папки і файли.

Ще один приклад ієрархічної організації даних подано на рисунку 3.6.

Звертаємо вашу увагу, що кожний батьківський об'єкт у сукупності з підпорядкованими об'єктами (нащадками) можна розглядати як окреме дерево.

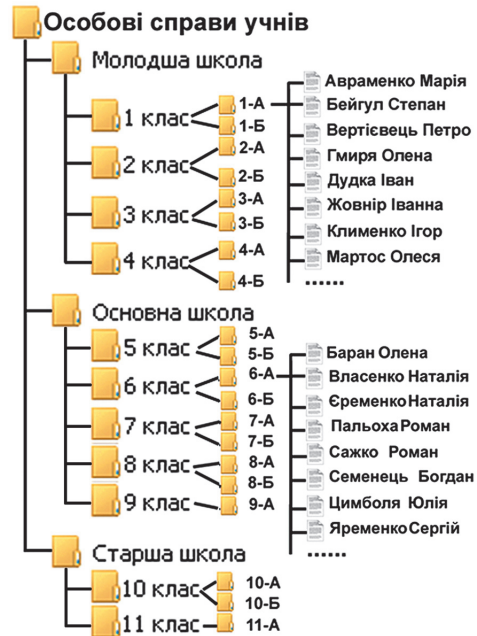


Рис. 3.6. Приклад ієрархічного розміщення даних про учнів школи

Для *опрацювання даних* в ієрархічній моделі даних використовується такий набір команд:

- знайти вказане дерево, наприклад дерево **8 клас**;
- перейти від одного дерева до іншого, наприклад від дерева **Молодша школа** до дерева **Старша школа**;
- перейти від батьківського об'єкта до об'єкта-нащадка всередині дерева, наприклад від об'єкта **6-А** до об'єкта **Семенець Богдан**;
- перейти від одного об'єкта до іншого в порядку, передбаченому ієрархічною структурою, наприклад від об'єкта **7-Б** до об'єкта **1-А**;
- вставити новий об'єкт в указаному місці;
- видалити поточний об'єкт

та ін.

Обмеження цілісності в ієрархічній моделі даних зокрема передбачає збереження зв'язків між батьківськими об'єктами і нащадками. Основне правило обмеження цілісності – жоден підпорядкований об'єкт (нащадок) не може існувати без батьківського об'єкта, за виключенням одного основного батьківського об'єкта. Тобто, об'єкт **9-Б** не може існувати без батьківського об'єкта **9 клас**, бо якщо немає дев'ятих класів у школі, то не може існувати список учнів одного з дев'ятих класів.

Ієрархічну модель даних зручно використовувати, коли потрібно створити базу даних предметної області, об'єкти якої також мають між собою ієрархічну залежність.

При значній кількості даних у базах даних, побудованих на ієрархічній моделі даних, на пошук потрібних даних можна витратити занадто багато часу. Наприклад, спробуйте знайти серед усіх файлів жорстких дисків вашого комп'ютера файл, що містить певний фрагмент тексту. Пошук може тривати кілька хвилин, а то і кілька десятків хвилин. У ході пошуку на відповідній панелі ви можете спостерігати як послідовно гілка за гілкою переглядається ієрархічна структура даних. За такий самий час як в Інтернеті подібний пошук триватиме максимум кілька секунд, при цьому опрацьовуються значно більші обсяги даних. У ході стандартного пошуку засобами операційної системи відбувається перегляд вмісту файлів по всіх гілках ієрархічного дерева файлової структури. Під час пошуку даних в Інтернеті використовуються створені пошуковими машинами бази даних, у яких вміст різноманітних сайтів проаналізовано і класифіковано, і для збереження цих даних використовують інші моделі даних, наприклад реляційну. У сучасних операційних системах, таких як **Windows 7**, для зменшення часу пошуку використовують індексацію даних – створення додаткової бази даних про об'єкти файлової системи та їх вміст.



Рис. 3.7. Чарльз Бахман

Ієрархічну модель даних важко використовувати для предметної області, в якій об'єкти пов'язані між собою більш складною залежністю, ніж ієрархія. Розширені можливості для опису такої предметної області надає *мережна модель даних*, що була розроблена наприкінці 60-х років XX ст. Автором концепції мережної моделі даних є американський учений **Чарльз Бахман** (нар. 1924 р.) (рис. 3.7). Мережна структура даних передбачає, що у кожного об'єкта може бути як кілька батьківських об'єктів, так і кілька об'єктів-нащадків.

Приклад зв'язків між об'єктами при використанні мережної моделі даних зображено на рисунку 3.8.

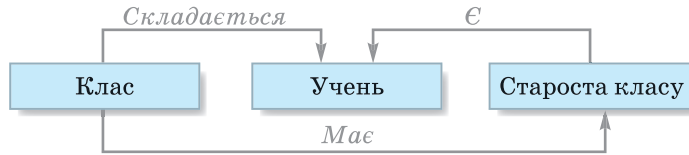


Рис. 3.8. Приклад застосування мережної моделі даних

Для **опрацювання даних** у мережній моделі даних використовується такий набір команд:

- знайти вказаний об'єкт серед однотипних об'єктів, наприклад об'єкт з даними про учня *Степаненка*;
- перейти від батьківського об'єкта до першого нащадка, використовуючи певний зв'язок;
- вставити новий об'єкт у вказаному місці;
- видалити поточний об'єкт;
- змінити об'єкт;
- включити об'єкт до певного зв'язку;

та ін.

Обмеження цілісності в мережній моделі даних передбачає збереження зв'язків між об'єктами.

Разом з тим використання мережної моделі даних ускладнюється при значному збільшенні кількості об'єктів предметної області й ускладненні зв'язків між цими об'єктами. Проблеми виникають і під час модифікації бази даних: додаванні нових зв'язків, заміні об'єктів, що зв'язані між собою, тощо.

Для спрощення опису об'єктів і зв'язків між ними в 1970 році американським ученим **Едгаром Франком Коддом** (1923–2003) (рис. 3.9) була запропонована **реляційна модель даних**. Математик за освітою, він увів у теорію баз даних математичний підхід, що базується на теорії множин.

Основою структури даних цієї моделі є таблиця. У таблицях кожний рядок містить набір значень властивостей одного з об'єктів предметної області. Кожний стовпець таблиці містить набір значень певної властивості об'єктів предметної області.

Така таблиця з набором стовпців, кожний з яких містить значення з певної скінченної множини, з точки зору математики задає відношення між множинами. Тому для опису структури даних Кодд використав термін «**relation**» (англ. *relation* – відношення), а модель даних стали називати **реляційною**.

Враховуючи, що таблиця реляційної бази даних складається з елементів певних множин, то для **опрацювання даних** цієї таблиці використовуються операції над множинами.

Обмеження цілісності в реляційній моделі бази даних передбачає дотримання двох принципів: обов'язкова можливість ідентифікації об'єкта бази даних за рахунок унікальності набору значень його властивостей,

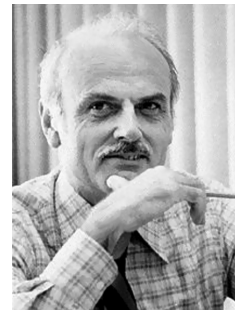


Рис. 3.9. Едгар Франк Кодд

указаних у рядку реляційної таблиці, та обов'язкова коректність зв'язків між таблицями бази даних.

Більш детально основні поняття баз даних, побудованих на реляційній моделі даних, розглянемо в наступних пунктах.

В останні роки ряд учених почали вказувати на недосконалість реляційної моделі даних, її обмеженість під час роботи з мультимедійними даними та даними про складені об'єкти. Для розширення можливостей реляційної моделі даних з кінця 90-х років XX ст. почала використовуватися **об'єктно-реляційна модель даних**.

Існують й інші моделі даних, такі як багатовимірні, комбіновані, однак поки що вони не набули широкого розповсюдження.

Основні поняття реляційної бази даних

Вище були розглянуті моделі даних, на основі яких створюються бази даних. Відповідно до моделі даних, яка лежить в основі бази даних, розрізняють **ієрархічні, мережні, реляційні та об'єктно-реляційні бази даних**. У подальшому розглядатимемо реляційні бази даних.

Ознайомимося з основними поняттями реляційної бази даних на прикладі таблиці, поданої на рисунку 3.10. Як ви вже знаєте, основним об'єктом реляційної бази даних є таблиця. Таблиця реляційної бази даних складається з рядків і стовпців. Для зберігання даних про сутності деякої предметної області може бути використано кілька таблиць, які можуть бути пов'язані між собою.

Рядок таблиці реляційної бази даних називають **записом**, або **кортежем**. Запис містить значення властивостей одного екземпляра сутності. Наприклад, у таблиці **Країни світу** (рис. 3.10) п'ятий зверху запис містить такий набір даних: *Австралія, Австралія і Океанія, 7636, 19138, 01.01.1901, Конституційна монархія*. Ці дані є значеннями відповідних властивостей одного екземпляра сутності **Країна світу: назва, частина**

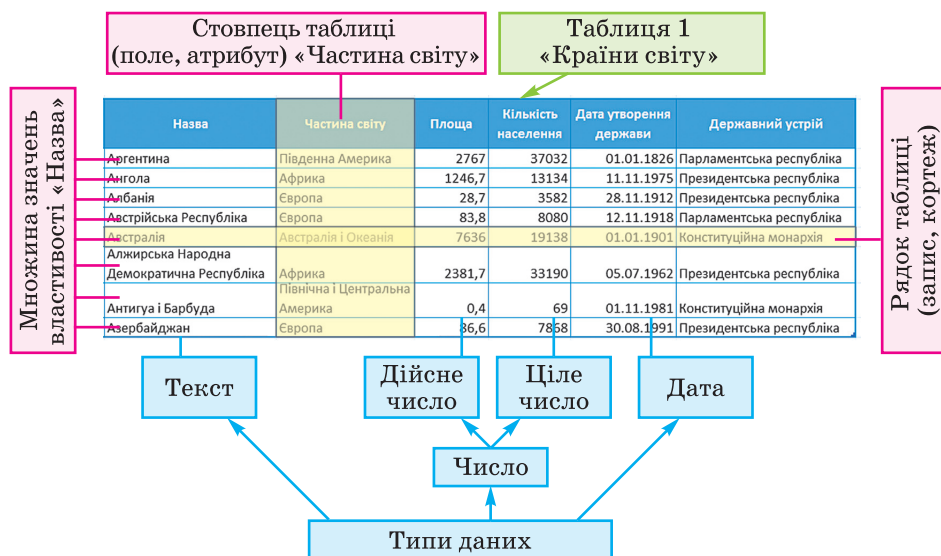


Рис. 3.10. Таблиця реляційної бази даних

світу, площа, кількість населення, дата утворення держави та державний устрій. Інший запис містить набір значень тих самих властивостей, але вже іншого екземпляра сутності – іншої країни.

Стовпець таблиці реляційної бази даних називають **полем**, або **атрибутом**. Поле має ім'я, яке відображає назву властивості. Поле містить множину значень однієї властивості всіх екземплярів сутності. Дані, що містяться в кожному полі таблиці, є однотипними. Для кожного поля під час проектування таблиці бази даних встановлюється **тип даних**.

Як уже зазначалося, у реляційній моделі даних передбачена обов'язкова вимога ідентифікації екземплярів сутності за рахунок унікальності набору значень властивостей. Для таблиці реляційної бази даних це означає, що значення в одному з полів не повинні повторюватися. Таке поле називають **ключовим**, або **ключем** таблиці. Ключові поля ще називають **ідентифікаторами**. В окремих реалізаціях реляційних баз даних допускається ідентифікація об'єкта бази даних за набором значень кількох властивостей, тобто ключем буде набір даних з кількох полів, наприклад трьох полів: *Прізвище, Ім'я, По батькові*.

У таблиці бази даних, поданої на рисунку 3.10, як ключове можна використати поле **Назва**, оскільки не існує однакових назв країн і тому значення в цьому полі є унікальним для кожного запису.



Перевірте себе

- 1°. Поясніть, що таке модель даних. Для чого створюється модель даних?
- 2°. Назвіть складові моделі даних.
- 3°. Поясніть, що визначає структура даних у моделі даних.
- 4°. Які типи даних можуть використовуватись у базах даних?
- 5°. Що таке обмеження цілісності в моделі даних?
- 6°. Назвіть найпоширеніші види моделей даних.
- 7°. Опишіть ієрархічну модель даних, вимоги до її структури, засоби опрацювання даних та обмеження цілісності.
- 8°. Опишіть мережну модель даних, вимоги до її структури даних, засоби опрацювання даних та обмеження цілісності.
- 9°. Коли і ким було розроблено теорію реляційної моделі даних?
- 10°. Опишіть реляційну модель даних, вимоги до її структури даних, засоби опрацювання даних та обмеження цілісності.
- 11°. Чим відрізняється об'єктно-реляційна модель даних від реляційної?
- 12°. За значенням якої властивості визначаються типи баз даних? Які ви знаєте типи баз даних?
- 13°. Що є основним об'єктом реляційної бази даних?
- 14°. Опишіть складові таблиці реляційної бази даних.
- 15°. Що таке ключове поле? Які вимоги до ключового поля?



Виконайте завдання

- 1°. Визначте, яку модель даних ілюструє кожне зображення (рис. 3.11).

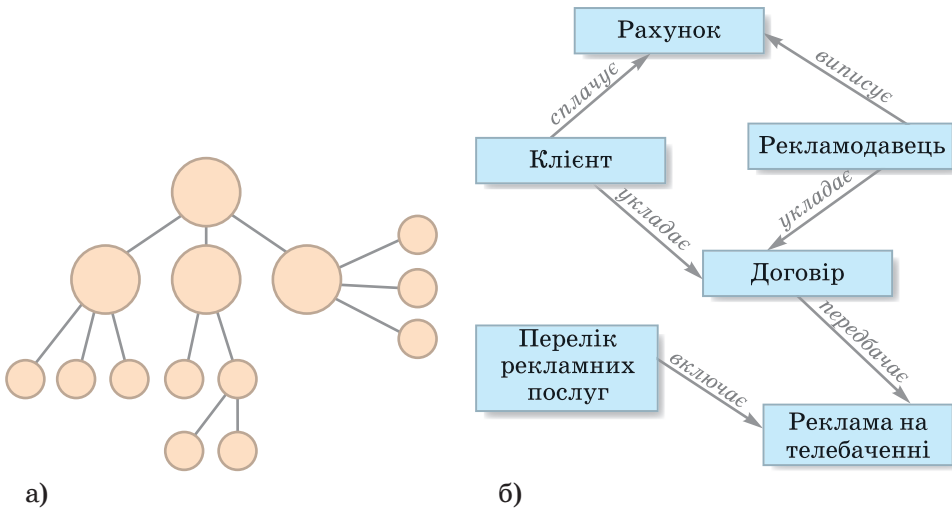


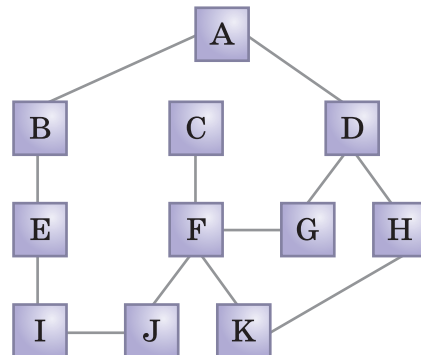
Рис. 3.11



2°. Визначте, яку модель даних ілюструє кожне зображення (рис. 3.12).

Властивість A	Властивість B	Властивість C	Властивість D
A ₁	B ₁	C ₁	D ₁
A ₂	B ₂	C ₂	D ₂
A ₃	B ₃	C ₃	D ₃
...
A _n	B _n	C _n	D _n

а)



б)

Рис. 3.12

3°. У фірмі «Біт і Байт» керівництво здійснює генеральний директор, якому підпорядкована рада директорів із чотирьох осіб – директора з маркетингу, директора з виробництва, директора з персоналу та директора з економічних питань. Директору з маркетингу підпорядкований відділ маркетингу, а директору з виробництва – три цехи зі складання комп'ютерів. У підпорядкуванні директора з економічних питань знаходиться відділ економічного прогнозування та бухгалтерія. Використовуючи одну з моделей даних, розробіть графічну модель управління фірмою.



4°. Необхідно створити базу даних бібліотеки, в якій повинні зберігатися такі відомості про книжки: *автор* (прізвище, ім'я, по батькові кожного з авторів); *назва*; *тип видання*; *місце видання*; *назва видавництва*; *рік видання*; *кількість сторінок*; *кількість томів*; *номер тома*; *бібліотечний шифр*. Запропонуйте модель даних для цієї бази даних.

5°. Розгляньте рисунок 3.13. Запропонуйте варіанти проведення ліній зв'язків між об'єктами для випадку, коли малюнок буде ілюструвати модель бази даних, яка побудована:

а) на ієрархічній моделі даних; б) на мережній моделі даних.



6°. Розгляньте фрагмент таблиці реляційної бази даних, поданий на рисунку 3.14. Визначте:

- назви об'єктів, виділених на рисунку іншим кольором і позначених цифрами 1 і 2;
- яке поле можна використати як ключове для цієї таблиці, поясніть чому;
- як можна назвати сутність, дані про яку представлені в таблиці;
- скільки екземплярів сутності подано на рисунку;
- які властивості екземплярів сутності описані в даній таблиці.

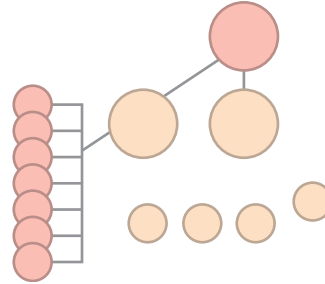


Рис. 3.13

№ особової справи ▾	Прізвище ▾	Ім'я ▾	По батькові ▾	День ▾	Місяць ▾	Рік народження ▾
Ф-09	Рябуха	Олександр	Сергійович	14	10	1991
Ф-13	Руденко	Сергій	Олександрович	28	12	1991
Ф-08	Орловський	Богдан	Віталійович	15	10	1990
Ф-07	Карлаш	Марина	Анатоліївна	16	10	1990
Ф-06	Михно	Олексій	Іванович	2	4	1992
Ф-26	Пальонка	Олег	Олександрович	30	5	1992
Ф-33	Костюк	Сергій	Володимирович	31	12	1991
Ф-17	Булаєв	Сергій	Юрійович	2	6	1990

Рис. 3.14

3.3. Етапи створення бази даних.

Поняття про системи управління базами даних



- Що таке база даних? Для чого створюються бази даних?
- Опишіть складові моделі даних.
- Які особливості реляційної моделі даних?
- Назвіть і опишіть етапи розв'язування задач з використанням комп'ютера.

Етапи створення бази даних

Процес створення бази даних є одним з прикладів розв'язування задач з використанням комп'ютера. До цього процесу потрібно ставитися відповідально, особливо на етапі проектування. Помилки в проектуванні можуть з'ясуватися після виконання значного обсягу робіт з введення даних. Це призводить до невиправданих втрат часу і коштів.

Створення баз даних передбачає такі етапи:

- Постановка завдання.** На цьому етапі визначається мета створення бази даних, окреслюється предметна область, для якої буде створюватися база даних, визначаються потенційні користувачі базою даних.
- Створення описової інформаційної моделі бази даних,** у якій виділяються сутності майбутньої бази даних, описуються зв'язки між ними.
- Створення моделі «сутність—зв'язок».**
- Визначення моделі даних.**
- Створення структури бази даних з використанням однієї із систем управління базами даних або однієї з мов програмування.**

6. Введення даних.

7. Тестування бази даних, її корекція.

Після тестування та корекції робота з базами даних не завершується. У ході експлуатації бази даних, як правило, дані регулярно оновлюються, можуть змінюватися зв'язки між сутностями тощо.

Розглянемо процес створення бази даних на прикладі бази даних **Країни світу**, яка може бути використана під час вивчення курсу географії в загальноосвітніх навчальних закладах.

Як було зазначено, на першому етапі в ході постановки завдання слід дати відповіді на такі питання:

- **мета створення бази даних:** база даних **Країни світу** створюється з метою використання під час вивчення географії в загальноосвітніх навчальних закладах для узагальнення відомостей про країни світу;
- **предметна область:** країни світу;
- **потенційні користувачі:** учні 9–10 класів загальноосвітніх навчальних закладів, учителі.

На другому етапі створення бази даних створюється її **описова інформаційна модель**: база даних повинна містити дані про назву країни, її площу, кількість і густоту населення, державні символи: прапор і гімн, національну валюту, дату утворення або завоювання незалежності, наявність дипломатичних відносин з Україною, адресу сайту уряду, державний устрій та економічний стан країни на день занесення даних. У базі даних повинна бути передбачена можливість підготовки узагальнених даних за країнами певної частини світу, а також за країнами, що мають певний державний устрій.

На третьому етапі створення бази даних визначається:

- **модель «сутність–зв'язок»:** модель «сутність–зв'язок» предметної області **Країни світу** зображена на рисунку 3.15. У ній три сутності: **Країна**, **Частина світу** та **Державний устрій**. Між сутністю **Частина світу** і сутністю **Країна** встановлюється зв'язок один до багатьох (1:∞), тобто одному екземпляру сутності **Частина світу** можуть відповідати багато екземплярів сутності **Країна**. Кожна країна належить до однієї із частин світу (зв'язок обов'язковий), однак існує частина світу (Антарктида), на якій немає жодної країни (зв'язок необов'яз-



Рис. 3.15. Модель «сутність–зв'язок» предметної області **Країни світу**

ковий). Між сутностями **Державний устрій** та **Країна** також встановлюється зв'язок один до багатьох (1:∞). Зв'язок для обох сутностей обов'язковий – кожна країна має один з видів державного устрою і кожна з форм державного устрою існує як мінімум в одній країні. Сутність **Країна** має такі властивості – *назва, площа, дипломатичні відносини, кількість населення, прапор, гімн, сайт уряду, дата утворення, економічний стан*. Сутність **Державний устрій** має дві властивості – *назва і глава уряду*, а сутність **Частина світу** одну – *назва*.

Далі визначається **модель даних**: реляційна.

Наступні етапи створення бази даних передбачають використання спеціального програмного забезпечення для створення та опрацювання баз даних і тому розглянемо їх пізніше.

Поняття про системи управління базами даних

Для роботи з базами даних використовують спеціальні програми – **системи управління базами даних (СУБД)**. Використовуючи їх, користувачі створюють бази даних, виконують пошук, оновлення даних тощо.



Системи управління базами даних – це прикладні комп'ютерні програми, призначені для створення, збереження та використання баз даних.

Залежно від моделі даних, яка використовується в СУБД, їх поділяють на **ієрархічні, мережеві, реляційні, об'єктно-реляційні** тощо.

Залежно від розміщення баз даних і компонентів, які забезпечують їхнє опрацювання, СУБД поділяють на **локальні** та **розподілені**. Локальні СУБД забезпечують зберігання й опрацювання даних на локальному комп'ютері, а у розподілених – дані можуть зберігатися та опрацьовуватися на різних комп'ютерах у локальній або глобальній мережі.

Залежно від способу доступу до бази даних СУБД розподіляють на **файл-серверні, клієнт-серверні та інтегровані (вбудовані)**. У файл-серверних СУБД файли з даними розміщуються на сервері, а на кожному клієнтському комп'ютері встановлено повну версію СУБД. Доступ з клієнтського комп'ютера до даних на файл-сервері здійснюється з використанням мережі. Недоліком таких СУБД є необхідність встановлювати інколи доволі дорогі повні версії програм на кожний комп'ютер, висока завантаженість мережі під час передавання даних із клієнтського комп'ютера на сервер і навпаки, необхідність у досить потужних

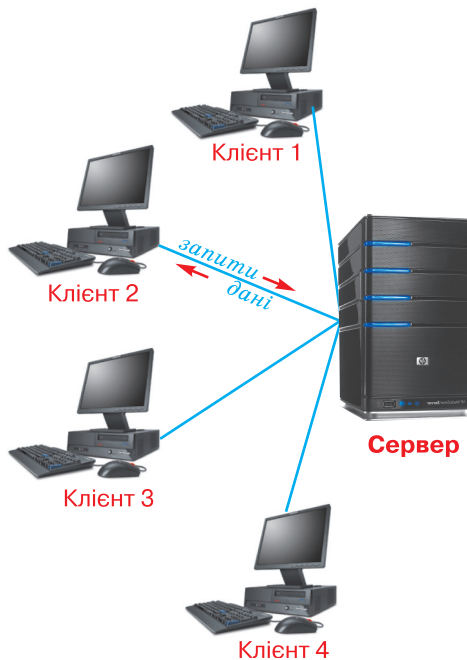


Рис. 3.16. Схема використання бази даних за клієнт-серверною технологією

комп'ютерах на робочих місцях клієнтів. Позитивним є невисокі вимоги до продуктивності комп'ютера, який є файл-сервером. Одним із варіантів використання файл-серверних СУБД є розміщення і СУБД, і даних на одному клієнтському комп'ютері.

Найбільш широко в останні роки розвивається клієнт-серверна технологія використання баз даних (рис. 3.16). За цією технологією на сервері встановлюється серверна версія СУБД, на ньому ж, як правило, розміщена і база даних. На клієнтських комп'ютерах встановлюються тільки невеликі за обсягом і функціональністю клієнтські версії СУБД, у завдання яких входить формування запитів на опрацювання даних і виведення результату опрацювання, надісланого з сервера. Усі операції з даними здійснюються на сервері. Такі СУБД не потребують потужних клієнтських комп'ютерів і не перевантажують мережу, простіше реалізується система збереження цілісності даних, однак значно підвищуються вимоги до технічних характеристик сервера. За клієнт-серверною технологією організована робота залізничних кас продажу квитків, робота банкоматів тощо.

Інтегровані (вбудовані) СУБД використовуються як складові інших програмних продуктів, наприклад електронних енциклопедій, словників, пошукових систем тощо. Ці системи не потребують окремої інсталяції, можуть мати обмежений набір функцій з управління базою даних, наприклад не забезпечувати оновлення даних. Доступ до даних здійснюється з використанням засобів прикладної програми, в яку інтегрована СУБД.

Схему основних класифікацій СУБД наведено на рисунку 3.17.

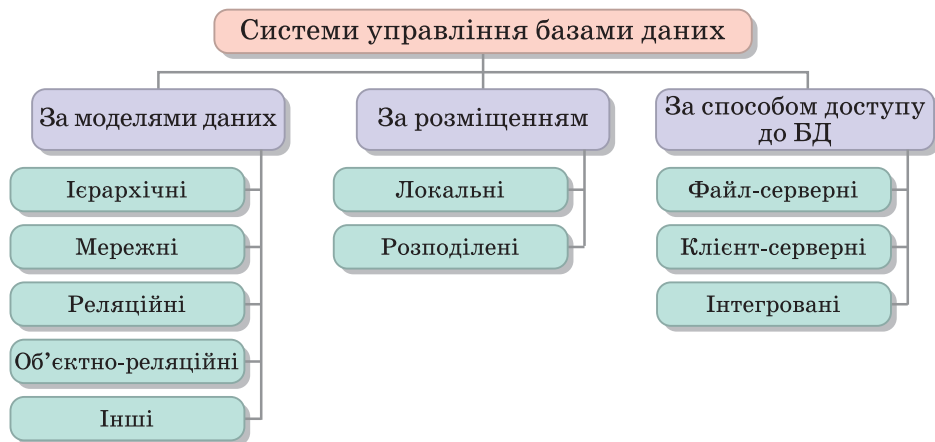


Рис. 3.17. Схема основних класифікацій систем управління базами даних



Перевірте себе

- 1°. Назвіть етапи створення бази даних.
- 2°. Опишіть процес постановки завдання як одного з етапів створення бази даних.
- 3°. Які завдання розв'язуються на другому етапі створення бази даних?
- 4°. Поясніть, які завдання розв'язуються в процесі створення моделі «сутність–зв'язок» і визначення моделі даних як етапу створення бази даних.
- 5°. Що таке системи управління базами даних?
- 6°. Які типи СУБД виділяють за моделями даних, що вони використовують?
- 7°. Чим відрізняються локальні та розподілені СУБД?

- 8*. У чому відмінність між файл-серверними і клієнт-серверними системами управління базами даних?
- 9*. Опишіть схему основних класифікацій СУБД за рисунком 3.17.



Виконайте завдання

- 1*. Розробіть перші чотири етапи створення:
 - а) бази даних учнів випускних класів школи для зберігання даних про їхні підсумкові оцінки з предметів і подальшого друку додатків до атестатів;
 - б) бази даних обліку обладнання одного з кабінетів школи;
 - в) бази даних обліку результатів шкільних спортивних змагань з легкої атлетики.
- 2*. Є шість комп'ютерів, об'єднаних у локальну мережу. Один з комп'ютерів – сервер бази даних. Побудуйте схему зв'язків між комп'ютерами при використанні файл-серверної технології роботи бази даних. Укажіть на схемі, де опрацьовуються дані з бази даних.
- 3*. Побудуйте схему розподіленої СУБД. Використайте під час побудови матеріали Інтернету.
- 4*. Підготуйте повідомлення за такими темами:
 - а) Використання інтегрованих СУБД в мобільних пристроях.
 - б) Використання об'єктно-орієнтованих СУБД.

3.4. Система управління базами даних Microsoft Office Access 2007



1. Опишіть етапи створення бази даних.
2. Опишіть основні поняття реляційної бази даних.
3. Що таке система управління базами даних? Для чого вона призначена?
4. Поясніть, як створити новий документ у **Microsoft Office Word**; нову електронну таблицю в **Microsoft Office Excel**.

У попередньому пункті були детально розглянуті перші етапи створення бази даних – постановка завдання, розробка інформаційної моделі, моделі «сутність–зв'язок» та визначення моделі даних. Наступні етапи неможливо розглядати без ознайомлення з однією із систем управління базами даних. Ми з вами вивчатимемо СУБД **Microsoft Office Access 2007** (далі **Access 2007** (англ. *Access* – доступ, вибірка інформації)) з пакета програм **Microsoft Office 2007**.

Загальна характеристика Access 2007

Система управління базами даних **Access 2007** – це реляційна СУБД. **Access 2007** належить також до файл-серверних СУБД і передбачає роботу з невеликими за обсягом базами даних для особистого користування та для невеликих організацій.

СУБД **Access 2007** призначена для створення та редагування:

- **таблиць**, що використовуються для зберігання даних і встановлення зв'язків між сутностями;
- **форм**, які використовуються для введення і перегляду даних;
- **запитів**, які забезпечують пошук, фільтрування й опрацювання даних з використанням спеціальної мови запитів – **SQL** (англ. *Structured Query Language* – мова структурованих запитів);




- **звітів** на основі наявних даних та ін.

З переліком властивостей цих об'єктів і можливими значеннями цих властивостей ознайомимося в ході створення і роботи з базами даних.

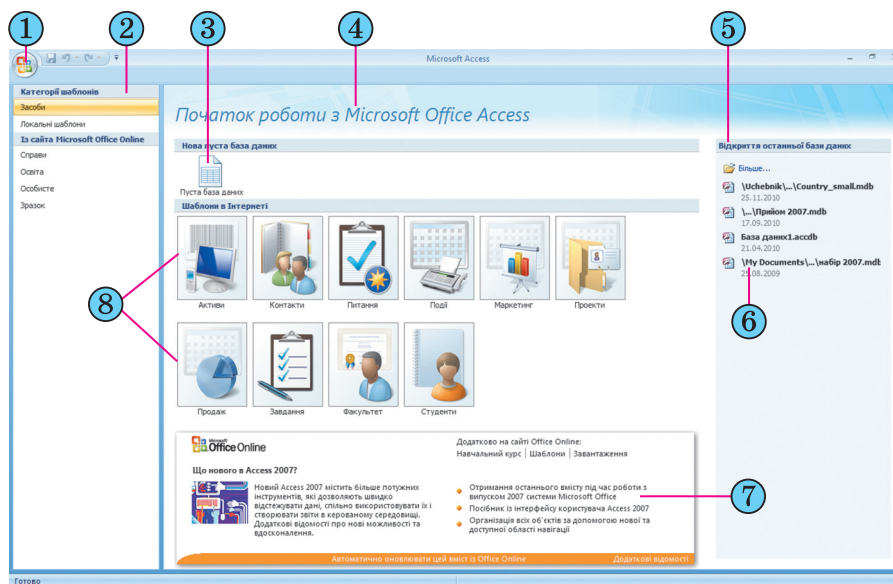
Для збереження баз даних **Access 2007** використовує як основний формат файлів **ACCDB** (англ. *Access DataBase* – база даних **Access**), забезпечуючи роботу і з форматами файлів попередніх версій програми – **MDB** (англ. *Microsoft DataBase* – база даних **Microsoft**).

Запуск Access 2007. Вікно програми

Access 2007 можна запустити на виконання кількома способами. Наведемо найпоширеніші з них:

- виконати **Пуск** ⇒ **Усі програми** ⇒ **Microsoft Office** ⇒ **Microsoft Office Access 2007**;
- використати ярлик програми на **Робочому столі**  або на **Панелі швидкого запуску** ;
- двічі клацнути на значку вже існуючого файлу бази даних формату, який визначено в операційній системі для відкриття з використанням **СУБД Access 2007** .


У першому та другому випадках після запуску програми **Access 2007** на екрані відкривається початкове вікно програми (рис. 3.18).



- | | |
|---|---|
| ① Кнопка Office | ⑤ Область Відкриття останньої бази даних |
| ② Список Категорії шаблонів | ⑥ Список файлів, що відкривалися останніми |
| ③ Кнопка Пуста база даних | ⑦ Область Microsoft Office Online |
| ④ Область Початок роботи з Microsoft Office Access | ⑧ Кнопки вибору шаблонів для завантаження |

Рис. 3.18. Початкове вікно програми **Access 2007**

У рядку заголовка вікна розміщена кнопка **Office**, кнопки керування вікном і недоступна для використання в цьому режимі роботи **Панель швидкого доступу**. Основна частина вікна розділена на три області. У лівій знаходиться список **Категорії шаблонів** з іменами вкладок **Засоби**, **Локальні шаблони** та іменами вкладок груп шаблонів, розміщеними на сайті **Microsoft Office Online (Справи, Освіта, Особисте, Зразок)**.



Вміст центральної області змінюється залежно від обраної в лівій частині категорії шаблонів. За замовчуванням після запуску програми в центральній області вікна **Access 2007** відображаються елементи керування, зображені на рисунку 3.18. Ця область горизонтально розділена на три частини. Угорі у розділі **Нова пуста база даних** є тільки одна кнопка – **Пуста база даних** . Середню частину займають кнопки вибору шаблонів бази даних, а нижню – посилання на довідкову інформацію, розміщену на сайті **Microsoft Office – Що нового в Access 2007?, Навчальний курс, Посібник з інтерфейсу користувача Access 2007** та ін.

У правій частині початкового вікна розміщено область **Відкриття останньої бази даних** зі списком файлів баз даних, з якими користувач працював раніше. Над списком файлів розміщено посилання **Більше**, використовуючи яке користувач може відкрити файл, розміщений у зовнішній пам'яті комп'ютера або в мережі.

Створення нової бази даних

На відміну від більшості прикладних програм робота з новою базою даних в **Access 2007** починається обов'язково зі створення файлу цієї бази даних.

Для створення файлу **нової пустої бази даних** потрібно:

1. Вибрати кнопку **Пуста база даних**  в області **Початок роботи з Microsoft Office Access**.
2. Увести у правій частині вікна в області **Пуста база даних** ім'я файлу майбутньої бази даних у поле **Ім'я файлу** або використати вказане за замовчуванням (останнє не рекомендується).
3. Перевірити, чи влаштовує вас запропоноване програмою місце збереження файлу, вказане під полем **Ім'я файлу**.
4. Указати за потреби нове розміщення файлу бази даних, для чого вибрати кнопку **Пошук розташування бази даних** , яка розташована справа від поля **Ім'я файлу**.
5. Вибрати кнопку **Створити**.

Для створення нової бази даних користувач також може використати вже розроблені шаблони. Частина шаблонів розміщується на локальному комп'ютері під час інсталяції програми **Access 2007**, більша ж частина доступна для копіювання із сайту **Microsoft Office**.

Для створення бази даних на основі інстальованого шаблону необхідно:

1. Запустити програму **Access 2007**.
2. Вибрати **Локальні шаблони** у списку **Категорії шаблонів**.
3. Вибрати в області **Локальні шаблони**, що відкриється, кнопку з назвою потрібного шаблону (наприклад, **Контакти**).
4. Увести у правій частині вікна в області **Справи-Контакти** (рис. 3.19) ім'я файлу майбутньої бази даних у поле **Ім'я файлу**

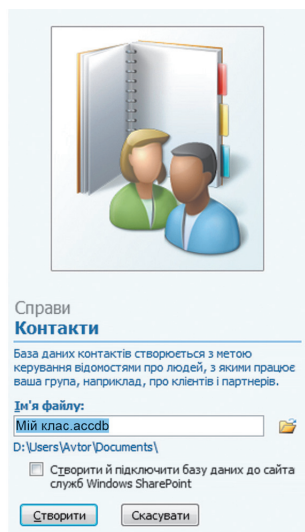


Рис. 3.19. Область
Справи–Контакти

(наприклад, **Мій клас**) або використати вказане за замовчуванням (останнє не рекомендується).

5. Вибрати за потреби нове місце збереження файлу.

6. Вибрати кнопку **Створити**.

Програма створить нову базу даних із вказаним іменем і обумовленою шаблоном структурою.

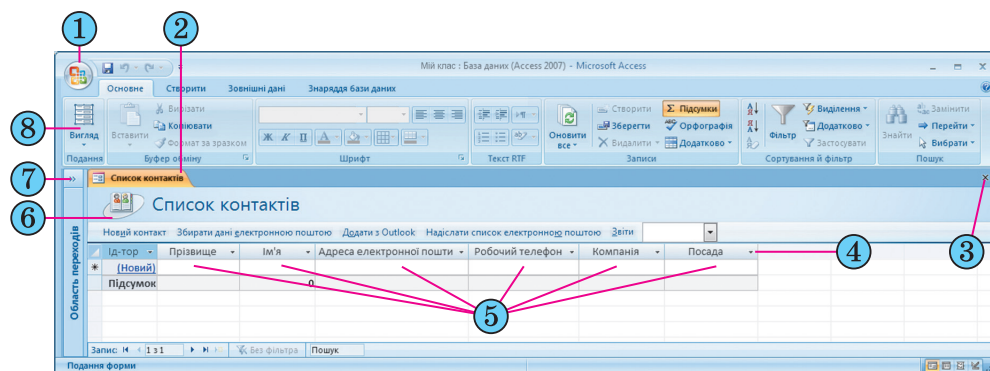
При відкритті створеної бази даних під **Стрічкою**, як правило, з'являється рядок (рис. 3.20) з попередженням системи безпеки про те, що певний вміст бази даних вимкнено. Для ввімкнення повного вмісту бази даних необхідно вибрати кнопку **Параметри** та у вікні, що відкриється, вибрати перемикач **Дозволити цей вміст**. Блокування окремих елементів бази даних здійснюється з метою запобігання ураженню вірусами.

На прикладі бази даних, створеної на основі шаблону **Контакти**, ознайомимося з інтерфейсом та основними операціями з об'єктами бази даних.





Рис. 3.20. Рядок попередження системи безпеки при відкритті файлів

Під час створення нової бази даних на основі шаблону **Контакти** за замовчуванням на екран виводиться форма **Список контактів** (рис. 3.21) для введення даних про осіб, з якими ви плануєте підтримувати ділові чи дружні контакти.

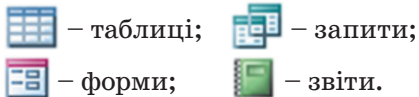


- | | |
|------------------------------------|---|
| ① Кнопка Office | ⑤ Поля |
| ② Вкладка відкритого об'єкта БД | ⑥ Заголовок форми |
| ③ Кнопка закриття поточної вкладки | ⑦ Кнопка відкриття/закриття області переходів |
| ④ Імена полів | ⑧ Кнопка Вигляд |


Рис. 3.21. Вікно СУБД з формою бази даних, створеної на основі шаблону **Контакти**

Щоб переглянути список інших об'єктів бази даних, потрібно відкрити **Область переходів**, вибравши кнопку  (рис. 3.21, 7). Вигляд списку користувач може змінити. Для змінення вигляду списку об'єктів необхідно вибрати кнопку  справа від заголовка області. Відкриється список, що складається з двох частин (рис. 3.22). У верхній частині можна вибрати одну з категорій, у які будуть групуватися об'єкти, а у нижній – фільтр, за яким у цій категорії будуть відбиратися об'єкти: *Усі об'єкти Access*, *Таблиці*, *Форми*, *Запити* чи *Звіти*.

За вибору категорії відображення **Тип об'єкта** в **Області переходів** з'явиться список об'єктів бази даних, згрупований за типами: **Таблиці**, **Форми**, **Запити**, **Звіти** (рис. 3.23). Кожний з об'єктів має умовне позначення у вигляді значка:



Для відкриття потрібного об'єкта необхідно двічі клацнути на його імені в **Області переходів**. Вміст даного об'єкта буде виведено на екран, його вкладка (рис. 3.21, 2) з'явиться у центральній частині вікна програми. Такі дії відбудуться для всіх названих об'єктів, крім деяких форм, які відкриваються в окремому вікні.

Для закриття поточного об'єкта необхідно вибрати кнопку закриття поточної вкладки . Для закриття поточної бази даних потрібно виконати *Office* ⇒ *Закрити базу даних*. Для закриття *Access 2007* треба вибрати кнопку закриття програми у **Рядку заголовка** або виконати *Office* ⇒ *Вийти з Access*. Якщо не передбачено змінення формату файлу бази даних або місця його збереження, то додатково зберігати файл БД перед закриттям не потрібно – дані в базі даних зберігаються автоматично після завершення їхнього введення.

Введення даних у форми в Access 2007

У базах даних введення даних можна здійснювати в таблицях або формах. Частіше користувач здійснює введення даних у формах.

Розглянемо введення даних на прикладі вже створеної на основі шаблону **Контакти** бази даних **Мій клас**. Відкриємо у цій базі форму **Відомості про контакт**. Для цього потрібно двічі клацнути в **Області переходів** на імені форми. Зазначена форма відкриється в окремому вікні (рис. 3.24). Такий тип форми називають **формою одного елемента**, тому що на екран виводяться дані тільки про один екземпляр сутності.

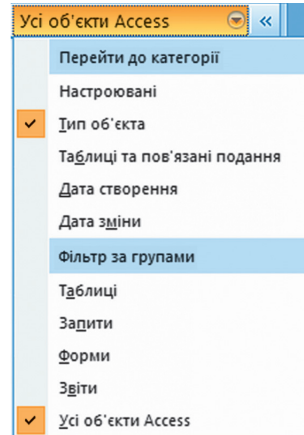


Рис. 3.22. Список вибору категорії та фільтра відображення об'єктів в **Області переходів**

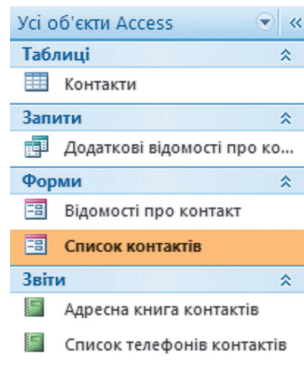
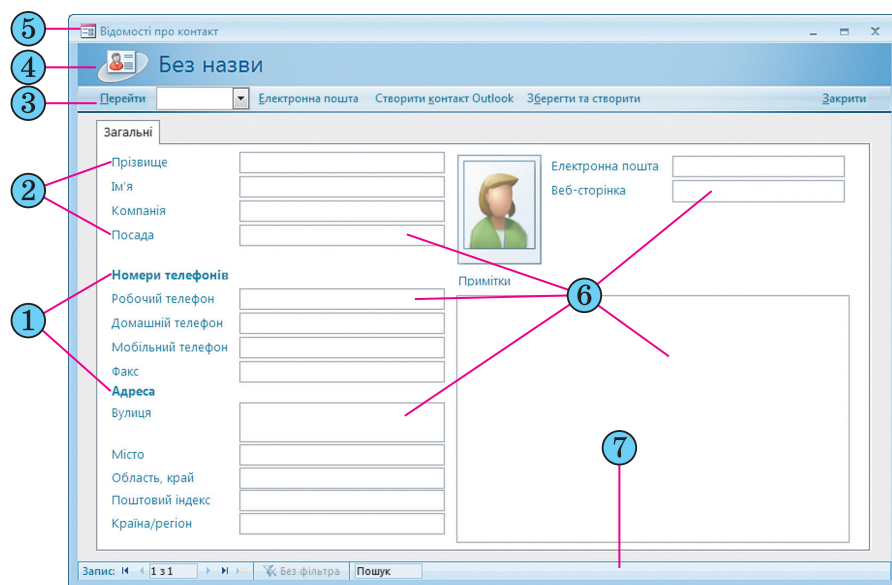
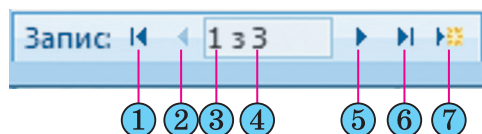


Рис. 3.23. Список об'єктів в **Області переходів**



- | | |
|--------------------------|---------------------------|
| ① Заголовки блоків даних | ⑤ Рядок заголовка вікна |
| ② Підписи полів | ⑥ Поля для введення даних |
| ③ Панель інструментів | ⑦ Рядок стану |
| ④ Рядок заголовка форми | |

Рис. 3.24. Вікно форми **Відомості про контакт**

- | | |
|----------------------------------|--------------------------------|
| ① Перехід до першого запису | ⑤ Перехід до наступного запису |
| ② Перехід до попереднього запису | ⑥ Перехід до останнього запису |
| ③ Номер поточного запису | ⑦ Створити запис |
| ④ Загальна кількість записів | |

Рис. 3.25. Елементи керування для переходу між записами

Для введення даних у поля форми необхідно встановити курсор у потрібне поле і ввести з клавіатури або вставити з **Буфера обміну** потрібні дані. Для переходу до наступного поля, якщо введення здійснюється з клавіатури, можна використовувати клавішу **Tab**.

Як уже було зазначено раніше, сукупність значень властивостей екземпляра сутності є **записом** бази даних. Завершивши введення даних першого запису, треба перейти до введення даних наступного запису. Для цього використовуються елементи керування у лівій частині **Рядка стану** (рис. 3.25).

Введені дані автоматично зберігаються в базі даних.

Типи даних

Під час уведення даних у поля бази даних слід дотримуватися не тільки певного шаблону, а й уводити дані відповідно до того типу, який був

запроектований на етапі створення моделі даних. Як ви знаєте, в одному полі бази даних можуть зберігатися дані тільки одного типу.

Для зберігання даних певного типу у пам'яті комп'ютера використовується різна довжина двійкового коду. Так, для зберігання даних логічного типу потрібен всього 1 біт пам'яті, а для текстових даних – до 256 байт (або до 512 байт при використанні Unicode). Чим менше обсяг даних, тим швидше відбувається опрацювання даних. Тому під час проектування баз даних намагаються мінімізувати обсяги даних за рахунок раціонального використання даних різних типів.

Основні типи даних, що використовуються у **Access 2007**, наведено у таблиці 3.2.

Таблиця 3.2. Основні типи даних, що використовуються в Access 2007

№	Тип даних	Довжина двійкового коду	Застосування
1	Так/Ні (логічний)	1 біт	Для зберігання значення логічного виразу (істина або хиб)
2	Текст	256 байт (512 байт при використанні таблиці кодування Unicode)	Для зберігання тексту завдовжки від 0 до 255 символів
3	Дата й час	8 байт	Для зберігання значення дати між роками 100 і 9999 та часу між 0:00:00 і 23:59:59
4	Примітка	До 2,14 Гбайт	Для зберігання тексту, довжина двійкового коду якого не перевищує вказаного значення
5	Об'єкт OLE	До 2,14 Гбайт	Для зберігання об'єктів довільного типу (наприклад, графічні або відео-файли), довжина двійкового коду яких не перевищує вказаного значення
6	Гіпер-посилання	До 1 Гбайт	Для зберігання посилання на веб-сторінки, файли в локальній мережі або на комп'ютері користувача
7	Авто-нумерація	4 байт	Для зберігання згенерованих програмою унікальних даних (цілих чисел від 0 до 4 294 967 295) і подальшого визначення поля як первинного ключа
8	Число	Залежно від значення	Див. табл. 3.3

Таблиця 3.3. Розміри числового типу даних, що використовуються в Access 2007

№	Розмір	Довжина двійкового коду	Застосування
1	Байт	1 байт	Для зберігання значень цілих чисел у діапазоні від 0 до 255
2	Ціле число	2 байт	Для зберігання значень цілих чисел у діапазоні від -32 768 до 32 767

№	Розмір	Довжина двійкового коду	Застосування
3	Довге ціле число	4 байт	Для зберігання значень цілих чисел у діапазоні від -2 147 483 648 до 2 147 483 647
4	Одинарне значення	4 байт	Для зберігання значень дійсних чисел у діапазонах: <ul style="list-style-type: none"> • від $-3,402823 \cdot 10^{38}$ до $-1,401298 \cdot 10^{-45}$ для від'ємних значень • від $1,401298 \cdot 10^{-45}$ до $3,402823 \cdot 10^{38}$ для додатних значень • та числа 0
5	Подвійне значення	8 байт	Для зберігання значень дійсних чисел у діапазонах: <ul style="list-style-type: none"> • від $-1,79769313486232 \cdot 10^{308}$ до $-4,94065645841247 \cdot 10^{-324}$ для від'ємних значень • від $4,94065645841247 \cdot 10^{-324}$ до $1,79769313486232 \cdot 10^{308}$ для додатних значень • та числа 0

Крім типу даних, **Access 2007** надає можливість установити обмеження на форму подання або на значення, яких можуть набувати дані в полях. Так, для типу даних **Текст** можна встановити обмеження на кількість символів, для полів з типом даних **Дата й час** – на форму подання, наприклад довгий формат часу, короткий формат дати тощо. Ці обмеження називають **форматами даних**. Види форматів числових даних у багатьох випадках схожі з форматами числових даних в **Excel**.

Перевірте себе

- 1°. До якого типу систем управління базами даних належить **Access 2007**?
- 2°. Поясніть призначення СУБД **Access 2007**.
- 3°. Який основний формат файлів використовується для зберігання баз даних в **Access 2007**?
- 4°. Опишіть можливі способи запуску на виконання програми **Access 2007**.
- 5°. Опишіть структуру початкового вікна програми **Access 2007** за рисунком 3.18.
- 6°. Поясніть послідовність створення нової пустої бази даних.
- 7°. Поясніть послідовність створення нової бази даних на основі інсталюваного шаблону, на основі шаблону з Інтернету.
- 8°. Для чого виконується блокування певного вмісту баз даних? Що треба зробити для вимкнення цього блокування для відкритої бази даних?
- 9°. Опишіть вікно **Access 2007** за рисунком 3.21.
- 10°. Поясніть, для чого використовується **Область переходів**.
- 11°. Як відкрити об'єкт бази даних, використовуючи **Область переходів**?
- 12°. Як закрити базу даних? Як закрити програму **Access 2007**?
- 13°. Опишіть послідовність введення даних у форму бази даних. На що слід звертати особливу увагу під час введення даних?
- 14°. Які типи даних використовуються в **Access 2007**?
- 15*. Поясніть, чим відрізняється тип даних **Текст** від типу даних **Примітка**. Поясніть потребу в кожному із цих типів.

- 16*. Поясніть, чим відрізняється розмір числового типу даних **Ціле число** від розміру числового типу даних **Одинарне значення**. Поясніть потребу в кожному з цих типів.
- 17*. Для чого використовується в базах даних тип даних **OLE**? Які дані можна помістити в поле з цим типом даних?
- 18*. Чому в базах даних використовуються різні типи даних? Чи не можна всі типи даних звести до одного, якогось універсального?
- 19*. Які розміри числового типу даних використовуються в **Access 2007**?
- 20*. Що таке формат даних? Для чого він використовується?



Виконайте завдання

- 1°. Укажіть назви об'єктів, зображених на рисунку 3.26.

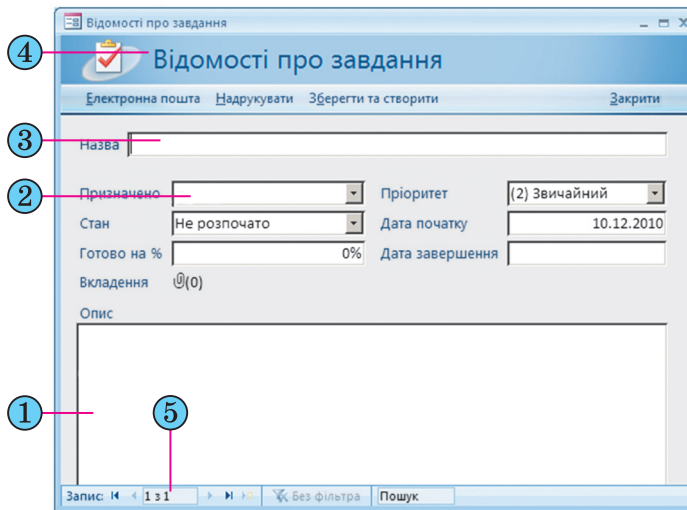


Рис. 3.26



- 2°. Укажіть назви об'єктів, зображених на рисунку 3.27. Поясніть їх призначення.

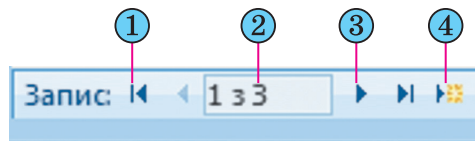


Рис. 3.27

- 3*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.4\зразок 3.4.3.accdb**) та:
 1. Увімкніть повний вміст бази даних.
 2. Відкрийте **Область переходів** та установіть такі її налаштування: **Перейти до категорії** – *Дата зміни*; **Фільтр за групами** – *Усі дати*.
 3. Визначте, який з об'єктів редагувався останнім, який – найдавніше.
 4. Змініть налаштування **Області переходів** на такі: **Перейти до категорії** – *Тип об'єкта*; **Фільтр за групами** – *Усі об'єкти Access*.
 5. Визначте, скільки й яких об'єктів включено до бази даних.
 6. Закрийте **Область переходів** і базу даних.



4*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.4\зразок 3.4.4.accdb**) та:

1. Перегляньте початкову сторінку бази даних.
2. Увімкніть повний вміст бази даних.
3. Закрийте початкову сторінку бази даних вибором кнопки
4. Виберіть у **Діалоговому вікні входу** довільне ім'я замовника, наприклад *Sergienko Mariya*, та кнопку **Вхід**.
5. Відкрийте **Область переходів** та установіть такі її налаштування: **Перейти до категорії** – *Тип об'єкта*; **Фільтр за групами** – *Усі об'єкти Access*.
6. Відкрийте групу **Таблиці** та визначте, скільки таблиць є в цій базі даних.
7. Перегляньте одну з них, наприклад **Клієнти**. Визначте основні об'єкти реляційної бази даних.
8. Закрийте відкриту таблицю, а потім і базу даних.

5*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.4\зразок 3.4.5.accdb**) та:

1. Установіть такі налаштування **Області переходів**: **Перейти до категорії** – *Тип об'єкта*; **Фільтр за групами** – *Усі об'єкти Access*. Відкрийте по черзі всі об'єкти бази даних, перегляньте їх та впевніться, що вони не містять даних. Закрийте всі об'єкти.
2. Відкрийте форму **Відомості про контакт**. Уведіть у форму дані про п'ятьох ваших однокласників. Якщо вам невідомі окремі дані – залиште ці поля порожніми. У полі **Компанія** вкажіть назву школи, а в полі **Посада** – учень.
3. Закрийте форму **Відомості про контакт** і відкрийте по черзі всі об'єкти бази даних. Переконайтеся, що тепер ці об'єкти містять введені вами дані.
4. Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.4.5. accdb**.



6*. За формою, поданою на рисунку 3.26, визначте типи даних, які слід використати для таких полів: **Назва, Дата початку, Опис**.

3.5. Відображення моделі «сутність–зв'язок» на базу даних. Робота з таблицями бази даних



1. Опишіть модель «сутність–зв'язок» для деякої предметної області, наприклад «Шкільна бібліотека».
2. Поясніть етапи створення нової бази даних.
3. Які типи даних використовуються в **Access 2007**? Для чого використовуються різні типи даних?
4. Як створити нову пусту базу даних в **Access 2007**?
5. Опишіть послідовність введення даних у форму бази даних.

Відображення моделі «сутність–зв'язок» на базу даних

У пункті 3.3 було розглянуто етапи створення бази даних і визначено зміст кількох перших етапів на прикладі створення бази даних **Країни світу**. Було визначено мету створення бази даних, предметну область і потенційних користувачів, створено описову модель БД і модель «сутність–зв'язок», вибрано реляційну модель для цієї бази даних.

Наступним етапом є **створення структури бази даних з використанням однієї із систем управління базами даних або однієї з мов програмування**.

У ході створення структури бази даних потрібно визначити:

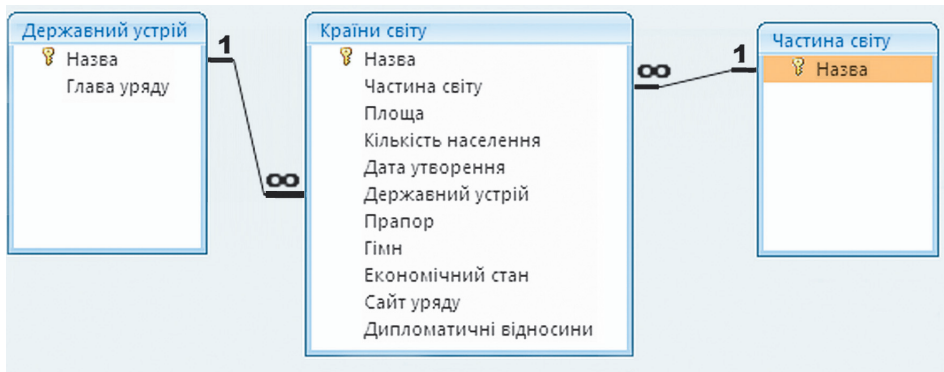
1. Кількість таблиць бази даних.
2. Перелік полів для кожної таблиці, які з цих полів будуть ключовими.
3. Типи даних для кожного поля кожної з таблиць з урахуванням можливостей конкретної СУБД.
4. Типи зв'язків між об'єктами таблиць, якщо використовується кілька таблиць.
5. Види і кількість інших об'єктів бази даних: форм, запитів, звітів.



Створення структури бази даних на основі її моделі «сутність–зв'язок» називається відображенням моделі на базу даних.

Розглянемо один з варіантів відображення моделі «сутність–зв'язок» **Країни світу** на базу даних, що буде створена в СУБД Access 2007. Структура цієї бази даних буде такою:

1. Кількість таблиць відповідає кількості сутностей (рис. 3.15) моделі «сутність–зв'язок». Їх три: **Країни світу**, **Державний устрій**, **Частина світу**.
2. Перелік полів для кожної з таблиць відповідає переліку властивостей для кожної із сутностей. Крім того, таблиця **Країни світу** повинна мати ще два поля *Державний устрій* і *Частина світу*, які будуть забезпечувати зв'язок між відповідними таблицями. Наведемо перелік полів для зазначених таблиць:
 - таблиця **Країни світу**: *Назва*, *Частина світу*, *Площа*, *Кількість населення*, *Дата утворення*, *Державний устрій*, *Прапор*, *Гімн*, *Економічний стан*, *Сайт уряду*, *Дипломатичні відносини*;
 - таблиця **Державний устрій**: *Назва*, *Глава уряду*;
 - таблиця **Частина світу**: *Назва*.
3. Типи даних для кожного з полів таблиць:
 - таблиця **Країни світу**:
 - **Назва** – тип даних – *текст*, розмір – 70 символів; **ключове поле**
 - **Частина світу** – тип даних – *майстер підстановок*. Цей тип даних передбачає використання даних з іншої таблиці;
 - **Площа** – тип даних – *число*, розмір – *одинарне значення*;
 - **Кількість населення** – тип даних – *число*, розмір – *довге ціле число*;
 - **Дата утворення** – тип даних – *дата й час*, формат *Short Data* (англ. *Short Data* – коротка дата);
 - **Державний устрій** – тип даних – *майстер підстановок*;
 - таблиця **Державний устрій**:
 - **Назва** – тип даних – *текст*, розмір – 50 символів; **ключове поле**;
 - **Глава уряду** – тип даних – *текст*, розмір – 30 символів;
 - таблиця **Частина світу**:
 - **Назва** – тип даних – *текст*, розмір – 50 символів; **ключове поле**.
4. Типи зв'язків (рис. 3.28):
 - таблиця **Державний устрій** поле **Назва** і таблиця **Країни світу** поле **Державний устрій** – *зв'язок один до багатьох*;
 - таблиця **Частина світу** поле **Назва** і таблиця **Країни світу** поле **Частина світу** – *зв'язок один до багатьох*.

Рис. 3.28. Зв'язки в базі даних **Країни світу**

5. Інші об'єкти бази даних:

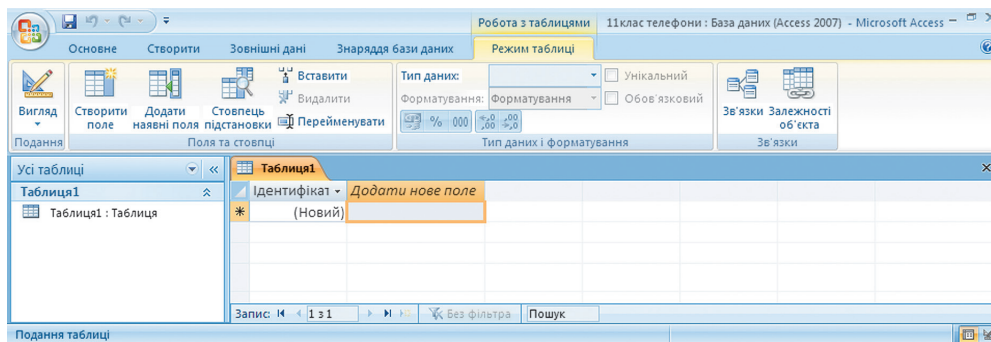
- форма **Країни світу** з полями: *Назва, Частина світу, Площа, Кількість населення, Дата утворення, Державний устрій, Прапор, Гімн, Економічний стан, Сайт уряду, Дипломатичні відносини*;
- запити:
 - **Площа країн світу** з полями: *Назва країни, Частина світу, Площа (з підрахунком загальної площі по певній частині світу)*;
 - **Державний устрій країн світу** з даними з полів: *Назва країни, Державний устрій*;
- звіт **Площа і кількість населення країн світу** з даними з полів: *Назва країни, Частина світу, Площа, Кількість населення з групуванням по Частині світу*.

Створення таблиць бази даних з використанням Подання таблиці

Перейдемо до створення бази даних. Для цього створимо нову порожню базу даних (рис. 3.29), як це було описано в попередньому пункті.

У подальшому можливі два основних способи створення полів таблиці та визначення типів даних у них:

- 1) з використанням **Подання таблиці**;
- 2) з використанням подання **Конструктор**.

Рис. 3.29. Вікно СУБД **Access 2007** з відкритою новою таблицею

Розглянемо перший спосіб. Особливістю його є те, що користувачу не обов'язково безпосередньо вказувати тип даних певного поля. СУБД Access 2007 автоматично визначає тип даних під час введення першого значення в поле і за замовчуванням встановлює певні значення форматів даних. Необхідно тільки створити поля з відповідними іменами. Одне з полів – **Ідентифікатор**, створюється програмою за замовчуванням під час створення нової порожньої бази даних, для нього встановлюється тип даних – *Автонумерація*. Це поле за замовчуванням визначається як ключове.

Для створення всіх інших полів з певними іменами слід виконати таку послідовність дій:

1. Двічі клацнути на заголовку **Додати нове поле**.
2. Увести з клавіатури або з **Буфера обміну** ім'я поля.
3. Натиснути клавішу **Enter** або клавішу →.
4. Повторити виконання пунктів 2 і 3 потрібну кількість разів для введення імен усіх необхідних полів.
5. Завершити процес введення імен полів вибором рядка під іменем поля або натисненням клавіші ↓.

Слід зважати на те, що імена полів не повинні містити крапок, знаків оклику, квадратних дужок, прихованих символів (наприклад, символа кінця абзацу).

Для встановлення типів даних та їхніх форматів використовуються елементи керування групи **Тип даних і форматування** вкладки **Режим таблиці** тимчасової вкладки **Робота з таблицями** (рис. 3.30). Для встановлення типу даних і формату необхідно виконати таку послідовність дій:

1. Зробити поточним потрібне поле.
2. Вибрати у списку **Тип даних** зазначеної групи (рис. 3.30) потрібний тип поля, наприклад для поля **Назва** таблиці **Державний устрій** – *Текст*.
3. Установити вибором зі списку **Форматування** потрібний формат подання даних (можливо лише для типів даних *Число*, *Дата й час*, *Грошова одиниця*, *Так/Ні*).

Установити обов'язковість заповнення певного поля або відсутність повторення значення у певному полі можна, встановивши позначку відповідних прапорців **Обов'язковий** або **Унікальний**. Використання інших кнопок цієї групи елементів керування аналогічно до розглянутого в Excel 2007.

Наступним етапом створення бази даних є *введення даних*. Для цього потрібно встановити курсор у поле і ввести дані з клавіатури або з **Буфера обміну**. Для переходу до наступного поля цього самого запису необхідно натиснути клавішу **Enter** або →. Після введення даних в останнє поле запису і натиснення клавіші **Enter** курсор автоматично переходить до першого поля наступного запису.

Так для таблиці **Державний устрій** уводимо такі значення у відповідні поля (рис. 3.31):

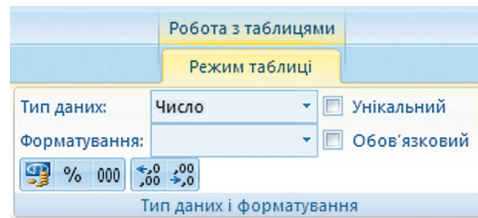


Рис. 3.30. Групи елементів керування **Тип даних і форматування**

Назва	Глава уряду
Президентська республіка	Президент
Парламентсько-президентська республіка	Прем'єр-міністр
Парламентська республіка	Прем'єр-міністр
Конституційна монархія	Прем'єр-міністр
Абсолютна монархія	Монарх
Однопартійна система	Глава партії
Військова диктатура	Диктатор (Військовий керівник)

Рис. 3.31. Таблиця **Державний устрій**



Подібним способом можна створити також і таблицю **Частина світу** з одним полем **Назва** та значеннями: *Азія, Європа, Америка, Африка, Австралія і Океанія, Антарктида*.

Для збереження створених таблиць слід виконати:

1. **Панель швидкого доступу** \Rightarrow **Зберегти**.
2. Увести в поле **Ім'я таблиці** вікна **Зберегти як** нове ім'я таблиці, наприклад **Державний устрій**.
3. Вибрати кнопку **ОК**.

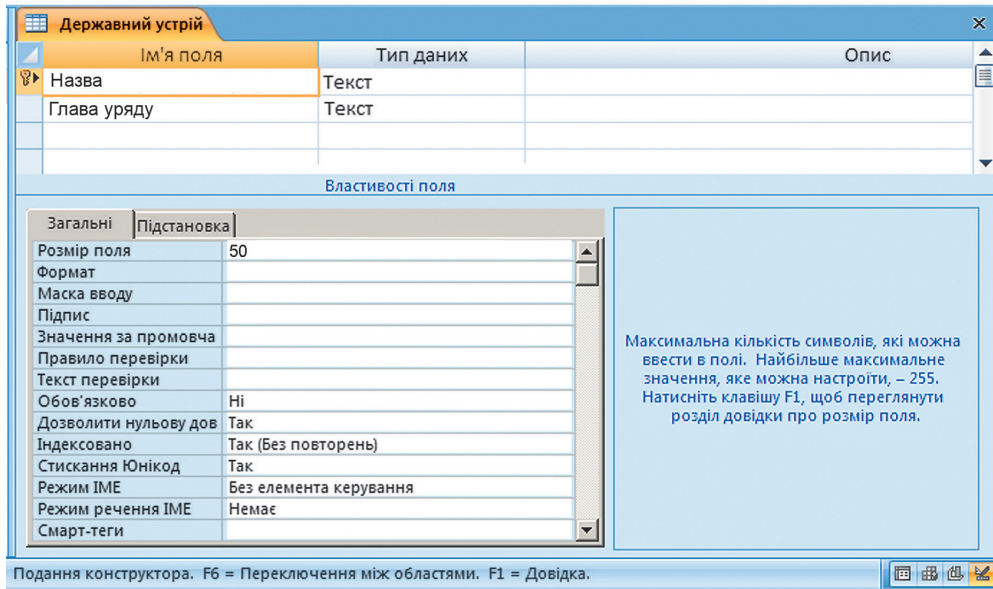
Створення таблиць бази даних у поданні Конструктор

Для створення нової таблиці бази даних зі складнішою структурою та для встановлення значень інших властивостей полів, як правило, використовують другий із вказаних способів створення таблиць – у поданні **Конструктор**. Для цього потрібно виконати **Створити** \Rightarrow **Таблиці** \Rightarrow **Конструктор таблиць**.

Якщо ж таблиця була створена раніше, то для переходу від **Подання таблиці** до подання **Конструктор** слід вибрати кнопку **Вигляд**  групи **Подання** вкладки **Основне**. У списку, що відкриється, слід вибрати кнопку **Конструктор** .

В обох випадках відкриється вікно таблиці у поданні **Конструктор**, аналогічне до наведеного на рисунку 3.32. У верхній частині вікна є три стовпці: **Ім'я поля**, **Тип даних** і **Опис**. Лівий стовпець використовується для введення імен полів. Для встановлення типу даних слід вибрати відповідне поле справа від імені поля та вибрати кнопку відкриття списку. Відкриється список типів даних (рис. 3.33). Серед них необхідно вибрати потрібний.

Після вибору типу даних поля можна змінити встановлені за замовчуванням і встановити не встановлені значення властивостей цього типу даних у нижній частині вікна на вкладці **Загальні**. Вміст цієї вкладки змінюється залежно від обраного типу даних. Так, для текстових даних основною властивістю, значення якої слід встановити в полі **Розмір поля**, є максимальна кількість символів, яку можна буде вводити в поле. Для типу даних **Число** у списку **Розмір поля** обирається один з типів числових даних, розглянутих вище. Крім того, для більшості числових типів даних на вкладці **Загальні** можна обрати у списку **Формат** один з форматів відображення числа (загальний числовий, грошова одиниця, євро, фіксований, стандартний, відсоток, експоненційний). У списку **Кількість знаків після коми** можна вибрати потрібне значення цієї властивості числа.

Рис. 3.32. Вікно таблиці **Державний устрій** у поданні **Конструктор**

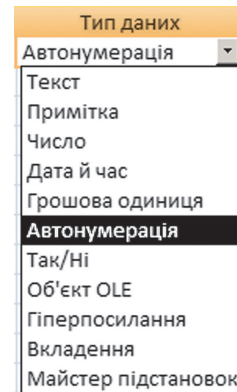
Для типів даних **Дата й час**, **Так/Ні** та **Грошова одиниця** можна обрати вид форматування. Ці види мають багато спільного з аналогічними форматами в **Excel 2007**. Для типу даних **Об'єкт OLE** можна встановити тільки підпис.

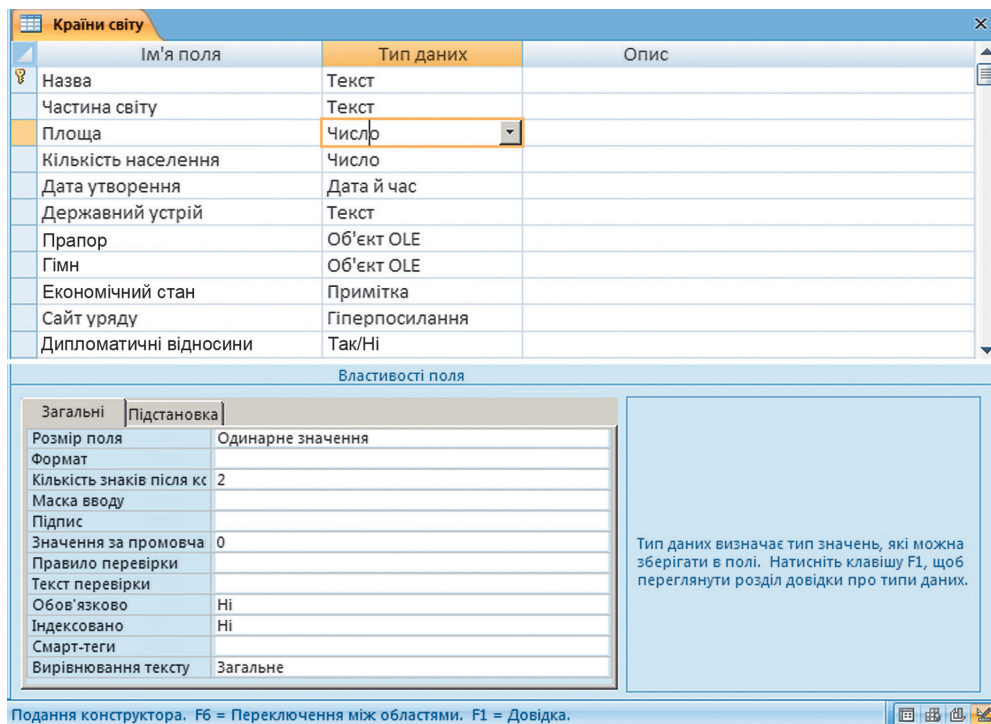
Серед інших обмежень на значення, яких можуть набувати дані, важливим є обмеження на обов'язковість заповнення певного поля під час введення даних. Для цього у нижній частині вікна на вкладці **Загальні** слід у списку **Обов'язково** вибрати **Так** або **Ні**. Якщо встановлено **Так**, то в подальшому дані з усього запису не будуть зберігатися, поки не буде заповнено зазначене поле. Така позначка є обов'язковою для ключових полів.

Для встановлення ключового поля слід зробити поточним відповідне поле і виконати **Конструктор** ⇒ **Знаряддя** ⇒ **Ключове поле** або відкрити контекстне меню поля та у списку команд вибрати **Ключове поле**. Для відміни встановлення ключового поля слід ще раз повторити описані вище операції.

Продовжимо роботу бази даних **Країни світу**. Створюємо і зберігаємо під іменем **Країни світу** третю таблицю. Створюємо поля та встановлюємо типи даних, їхні розміри та формати відповідно до визначених раніше значень (рис. 3.34).

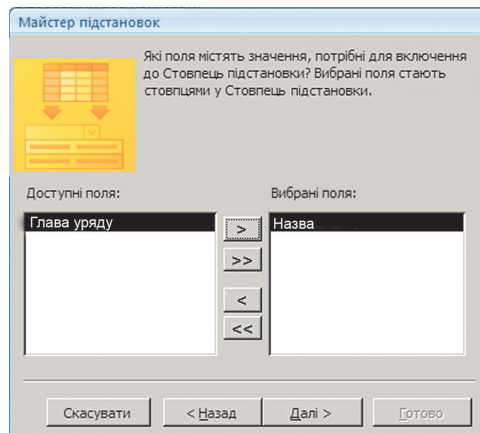
При визначенні типу даних для полів **Державний устрій** та **Частина світу** одночасно здійснюється зв'язування таблиць. Для вказаних полів у списку типів даних обирається **Майстер підстановок**. Після цього запускається названий майстер – невелика спеціальна програма, яка допомагає користувачу задати список для підстановки в поле та встановити зв'язки між полями таблиць.

Рис. 3.33. Список **Тип даних**

Рис. 3.34. Вікно таблиці **Країни світу** у поданні **Конструктор**

На першому кроці роботи майстра пропонується обрати один з двох видів джерел даних – поле з уже існуючої таблиці, або запиту бази даних, або з того переліку значень, які введе користувач на наступних кроках роботи майстра. Для нашого випадку ми повинні вибрати перший варіант.

На другому кроці роботи майстра обирається таблиця або запит, які містять необхідні дані. Наприклад, для поля **Державний устрій** вибирається таблиця **Державний устрій**, а для поля **Частина світу** – таблиця **Частина світу**.

Рис. 3.35. Вибір полів, що містять потрібні значення, у вікні **Майстер підстановок**

На третьому кроці роботи майстра (рис. 3.35) обирається поле або поля, які містять необхідні дані. Для вибору слід у списку доступних полів вибрати ім'я поля та вибрати кнопку . Для полів **Державний устрій** і **Частина світу** вибираються поля **Назва** тільки з різних таблиць.


Наступні кроки надають можливість користувачу обрати вид сортування даних, установити ширину стовпця. На останньому кроці ко-

ристувач може встановити можливість вибирати кілька значень з таблиці для підстановки у поле. Для цього треба встановити позначку прапорця **Дозволити кілька значень**.

Після такого зв'язування у полях **Державний устрій** і **Частина світу** таблиці **Країни світу** введення даних може відбуватися шляхом вибору значення зі списку. За потреби, для доповнення цих списків достатньо ввести нові значення в поля таблиці **Державний устрій** чи **Частина світу**.

Зв'язки в СУБД Access 2007

Ви ознайомилися з одним із способів створення зв'язків між таблицями – з використанням **Майстра підстановок**.

Для перегляду вже створених зв'язків у базі даних слід виконати **Знаряддя бази даних** ⇒ **Відобразити або приховати** ⇒ **Зв'язки**. Відкриється вікно **Зв'язки**, для відображення у ньому всіх зв'язків таблиці слід вибрати кнопку **Усі зв'язки**  групи **Зв'язок** вкладки **Конструктор** тимчасової вкладки **Знаряддя для зв'язків** (рис. 3.36). Зображення вікна **Зв'язок** зі встановленими зв'язками між полями відповідних таблиць подано на рисунку 3.28.

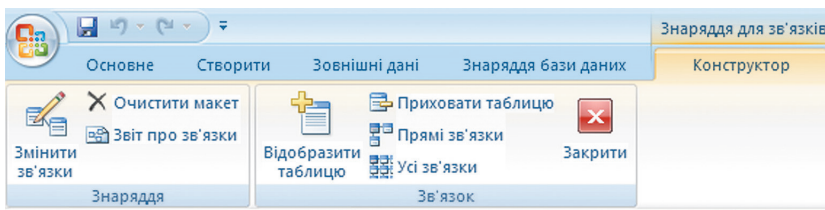
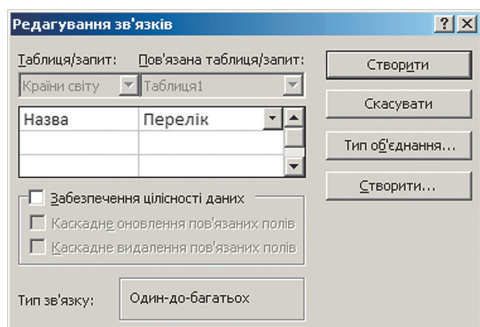


Рис. 3.36. Елементи керування вкладки **Конструктор** тимчасової вкладки **Знаряддя для зв'язків**

У СУБД **Access 2007** встановлюються два типи зв'язків – *один до одного* та *один до багатьох*. Зв'язок *один до одного* встановлюється між полями, в яких властивість **Індексовано** має значення **Так (Без повторень)**. Тобто ці поля є ключовими і не допускають однакових значень. Зв'язок *один до багатьох* встановлюється між полями, одне з яких не допускає повторів значень (значення властивості **Індексовано** – **Так (Без повторень)**), а поле іншої таблиці допускає повторення значень (значення властивості **Індексовано** – **Ні**).

У більшості випадків для створення, редагування та видалення зв'язків між таблицями використовують вікно **Зв'язок**. Для встановлення нових зв'язків необхідно у вікні **Зв'язок** виконати таку послідовність дій:

1. Виконати **Знаряддя для зв'язків** ⇒ **Конструктор** ⇒ **Зв'язок** ⇒ **Відобразити таблицю**.
2. У вікні **Відображення таблиці** вибрати назву таблиці для створення зв'язку і вибрати кнопку **Додати**.
3. Повторити дії для відображення інших таблиць, з якими планується зв'язати першу таблицю.
4. Закрити вікно **Відображення таблиці**.
5. Вибрати у списку імен полів першої таблиці ім'я поля, яке буде зв'язано, і перетягнути його на ім'я поля пов'язаної таблиці.

Рис. 3.37. Вікно **Редагування зв'язків**

об'єднання відкривається вікно **Параметри об'єднання**, в якому вибором відповідного перемикача можна встановити обов'язковість входження у зв'язок усіх чи окремих записів з відповідних таблиць.

При встановленні у вікні **Редагування зв'язків** позначки прапорця **Забезпечення цілісності даних** програма автоматично відслідковуватиме, щоб:

- у полі підпорядкованої таблиці не були введені дані, які відсутні у полі головної таблиці, наприклад, у таблиці **Крайни світу** у полі **Частина світу** не може бути введена інша частина світу, крім перерахованих у таблиці **Частина світу** в полі **Назва**;
- не видалялися або не змінювалися записи з головної таблиці, якщо в підпорядкованій таблиці є пов'язані з ними записи, тобто з поля **Назва** таблиці **Частина світу** не може бути видалено запис **Америка**, якщо у пов'язаній з нею таблиці **Крайни світу** є запис, що містить у полі **Частина світу** значення **Америка**, наприклад запис для країни **Канада**;
- поле головної таблиці було ключовим;
- зв'язані поля мали однаковий тип даних.

Для того щоб зміни значень у зв'язаному полі головної таблиці приводили до автоматичної зміни в усіх записах відповідного поля підпорядкованої таблиці, необхідно встановити позначку прапорця **Каскадне оновлення пов'язаних полів** (рис. 3.37). Наприклад, якщо у таблиці **Державний устрій** у полі **Назва** замість значення **Військова диктатура** запишемо значення **Диктатура**, то в підпорядкованій таблиці **Крайни світу** для всіх записів у полі **Державний устрій** значення **Військова диктатура** зміниться на **Диктатура**.

Якщо встановити позначку прапорця **Каскадне видалення пов'язаних полів**, то під час видалення значення з пов'язаного поля головної таблиці у підпорядкованій таблиці будуть видалені всі записи, що містять це значення. Наприклад, якщо у таблиці **Державний устрій** у полі **Назва** видалимо значення **Військова диктатура**, то з таблиці **Крайни світу** будуть видалені всі записи про країни, в яких у полі **Державний устрій** було значення **Військова диктатура**.

Для видалення зв'язку між таблицями достатньо відкрити контекстне меню лінії зв'язку між таблицями і вибрати команду **Видалити**.

6. Уточнити у вікні **Редагування зв'язків** (рис. 3.37) у списках **Таблиця/запит** та **Пов'язана таблиця/запит** назви полів, між якими встановлюється зв'язок.

7. Установити тип зв'язку (кнопка **Тип об'єднання...**).

8. Установити за потреби позначку прапорця **Забезпечення цілісності даних**.

9. Вибрати кнопку **Створити**.

У ході встановлення властивостей зв'язку за вибору кнопки **Тип**

Редагування та форматування таблиці бази даних

У процесі створення та використання таблиць бази даних може виникнути потреба у редагуванні самої таблиці. До основних операцій редагування структури таблиці слід віднести: вставлення та видалення полів (стовпців), перейменування полів (стовпців), вставлення та видалення записів (рядків), переміщення полів (стовпців), зміна ключового поля тощо.

Для редагування структури таблиці використовують елементи керування групи **Записи** вкладки **Основна** та групи **Поля та стовпці** вкладки **Режим таблиці** тимчасової вкладки **Робота з таблицями** (рис. 3.38).

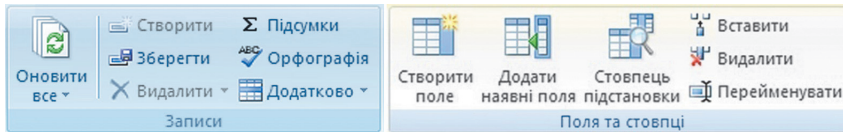


Рис. 3.38. Елементи керування вкладок **Записи** і **Поля та стовпці**

Для вставлення поля (стовпця) в **Поданні таблиці** необхідно зробити поточним поле (стовпець), перед яким ви плануєте вставити поле, і виконати **Робота з таблицями** \Rightarrow **Режим таблиці** \Rightarrow **Поля та стовпці** \Rightarrow **Вставити**. Для нового поля слід задати ім'я розглянутим вище способом. Вибір кнопки **Створити поле** групи **Поля та стовпці** приводить до створення нового поля в кінці списку полів.

Для видалення поля (стовпця) потрібно:

1. Зробити поточним поле (стовпець), яке планується видалити.
2. Виконати **Робота з таблицями** \Rightarrow **Режим таблиці** \Rightarrow **Поля та стовпці** \Rightarrow **Видалити**.
3. Підтвердити видалення поля та всіх даних у ньому у відповідному вікні.

Інший спосіб видалення полів:

1. Виділити поле вибором його імені.
2. Виконати **Основне** \Rightarrow **Записи** \Rightarrow **Видалити**.
3. Підтвердити видалення стовпця у відповідному вікні.

Можна також скористатися командою **Видалити стовець** контекстного меню імені поля.

Для видалення кількох полів потрібно попередньо виділити ці поля, використовуючи заголовки стовпців.

Під час видалення полів слід зважати на те, що в таблиці повинно бути мінімум одне поле. Також слід враховувати те, що змінити або видалити ключове поле в **Поданні таблиці** не можна. Це можна зробити тільки у поданні **Конструктор**.

Вставлення нових записів відбувається шляхом заповнення нижнього порожнього рядка.

Звертаємо вашу увагу. У таблиці бази даних завжди є нижній порожній рядок. Як тільки його починають заповнювати, автоматично створюється новий запис, а нижче з'являється новий порожній рядок.

Для вставлення нового запису можна виконати **Основне** \Rightarrow **Записи** \Rightarrow **Створити**. Цей спосіб зручно використовувати під час створення нового запису у великому масиві записів.

Для видалення запису потрібно зробити поточним запис і виконати **Оснoвнe** ⇒ **Записи** ⇒ **Видалити** ⇒ **Видалити запис**.

Для редагування імені поля необхідно двічі клацнути на імені поля, яке ви збираєтеся редагувати. Вміст полів редагується аналогічно до редагування таблиці у текстовому процесорі за винятком зв'язаних полів, у яких дані можуть вводитися шляхом вибору зі списку та полів з логічним типом даних, у яких редагування вмісту зводиться до встановлення/зняття позначки прапорця.

Для переміщення поля необхідно виділити поле і перетягнути його за заголовок у потрібне місце.

Форматування таблиці бази даних здійснюється шляхом зміни висоти рядків і ширини стовпців, формату шрифтів, оформлення ліній сітки та заливки полів, приховування окремих полів, закріплення елементів таблиці аналогічно до дій в **Excel 2007**.



Рис. 3.39. Елементи керування групи **Шрифт**

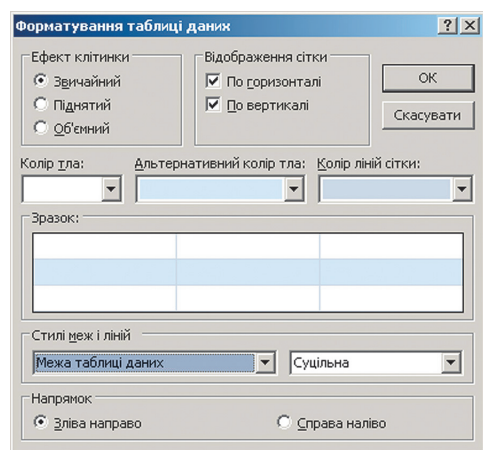


Рис. 3.40. Вікно **Форматування таблиці даних**

Для змінення формату шрифтів використовують елементи керування групи **Шрифт** вкладки **Оснoвнe** (рис. 3.39).

Розширені можливості щодо форматування таблиці бази даних надають користувачу елементи керування вікна **Форматування таблиці даних** (рис. 3.40). Це вікно можна відкрити вибором кнопки відкриття діалогового вікна групи **Шрифт**. Додатково можна встановити один з трьох ефектів оформлення таблиці (*звичайний, піднятий, об'ємний*), напрям розміщення полів (*зліва направо* чи *справа наліво*), стиль ліній меж таблиці та стиль лінії підкреслення заголовка (*суцільна, пунктир, короткі штрихи, крапки, розріжені крапки, штрихпунктир*).

Звертаємо вашу увагу. Форматування в таблиці бази даних не можна здійснювати для окремих полів, записів, символів. Встановлені значення форматування застосовуються для всієї таблиці.



Перевірте себе

- 1°. Поясніть, у чому полягає відображення моделі «сутність–зв'язок» на базу даних. Наведіть приклади.
- 2°. Опишіть перелік операцій, що виконуються при відображенні моделі «сутність–зв'язок» на базу даних.
- 3°. Назвіть два основних способи створення таблиць бази даних.
- 4°. Яке поле створюється автоматично під час створення таблиці, який тип даних цього поля?

- 5°. Опишіть послідовність створення нових полів у **Поданні таблиці**.
- 6°. Які існують обмеження на використання символів в іменах полів?
- 7°. Як зберегти таблицю з певним іменем?
- 8°. Як змінити вид подання для таблиці? Опишіть послідовність відкриття таблиці в поданні **Конструктор**.
- 9°. Поясніть призначення об'єктів таблиці в поданні **Конструктор**.
- 10°. Як створити нове поле таблиці в поданні **Конструктор**?
- 11°. Як установити тип поля таблиці в поданні **Конструктор**?
- 12°. Що означає встановлення значення *Так* властивості поля **Обов'язково**? Для чого воно використовується?
- 13°. Як встановити ключове поле таблиці?
- 14°. Що означає значення 25 у рядку **Розмір** поля з типом даних **Текст**?
- 15°. Що таке **Майстер підстановок**? Для чого він використовується?
- 16°. Опишіть послідовність зв'язування таблиць з використанням **Майстра підстановок**.
- 17°. Опишіть послідовність зв'язування таблиць у вікні **Зв'язки**.
- 18°. Опишіть основні операції редагування таблиць бази даних.
- 19°. Як здійснюється форматування таблиці бази даних? Поясніть основні операції із форматування таблиць.



Виконайте завдання



- 1°. Створіть у середовищі СУБД **Access 2007** таблицю бази даних **Телефонний довідник**, у яку включіть поля для зберігання даних про прізвище та ім'я власника телефону, прототип (стаціонарний чи мобільний) і номер телефону. Уведіть телефонні номери та інші дані п'яťох учнів вашого класу. Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.5.1.accdb**.
- 2°. Створіть у середовищі СУБД **Access 2007** базу даних **Країни світу** за описом, наведеним у пункті підручника. Уведіть дані про п'ять країн. Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.5.2.accdb**.
- 3°. Створіть базу даних **Домашня аудіотека**, яка повинна містити дані про номер диска в домашній аудіотеці, назву диска, виконавця, дату випуску, загальну тривалість музичних творів на диску. Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.5.3.accdb**.
- 4*. За моделлю «сутність–зв'язок», поданою на рисунку 3.4, створіть базу даних **Зберігання товарів**. Попередньо визначте для цієї бази даних: кількість таблиць, перелік полів для кожної таблиці, які з цих полів будуть ключовими, типи даних для кожного поля кожної з таблиць, типи зв'язків між об'єктами таблиць для випадку, коли є п'ять видів товару, що зберігаються на двох складах. До сутності **Товар** додайте властивість **Місце зберігання**. Усі таблиці створіть, використовуючи подання **Конструктор**. Заповніть таблиці даними із файлу, вказаного вчителем, наприклад **Тема 3\Завдання 3.5\зразок 3.5.4.docx**, відповідно до моделі БД і завдання. Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.5.4.accdb**.



Практична робота № 12. Розробка моделі «сутність–зв'язок» заданої предметної області. Створення бази даних у середовищі СУБД

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. За завданням на створення бази даних розробіть модель «сутність–зв’язок» і визначте модель даних.
Завдання. Створити базу даних **Бібліотека**, яка складається з двох сутностей – сутності **Об’єкт зберігання бібліотеки**, що має такі властивості: *код, найменування, автор, кількість, ціна, рік видання, зображення титульної сторінки*, і сутності **Категорія**, яка має одну властивість – *назву*, що може набувати тільки такі значення: *книга, аудіокнига, газета, журнал*. Між сутністю **Одиниця зберігання бібліотеки** і сутністю **Категорія** є зв’язок: одне значення властивості **Назва** сутності **Категорія**, наприклад *аудіокнига*, може мати багато екземплярів сутності **Об’єкт зберігання бібліотеки**.
2. Створіть необхідні таблиці бази даних, визначте імена полів і типи даних цих полів.
3. Установіть зв’язок між таблицями або з використанням **Майстра підстановок**, або з використанням інструмента **Зв’язки**.
4. Уведіть дані в таблицю **Фонд бібліотеки**, використовуючи таблицю (файл **таблиця 12.docx**) та графічні файли з папки **Тема 3\Практична 12**.
5. Установіть для таблиці сутності **Об’єкт зберігання бібліотеки** колір сітки – *темно-зелений*, колір тла – *блідо-жовтий*, альтернативний колір тла – *світло-голубий*.
6. Збережіть базу даних у своїй папці у файлі з іменем **практична робота 12.accdb**.

3.6. Форми. Створення, редагування та форматування форм



1. Назвіть основні об’єкти бази даних та їхнє призначення.
2. Опишіть послідовність створення нової таблиці бази даних.
3. Як вводяться дані в таблицях бази даних?
4. Які види подання таблиць бази даних ви знаєте? Для чого вони використовуються?

Види форм у базі даних

Як зазначалося, форма є одним з основних об’єктів бази даних. Форма забезпечує зручний для користувача інтерфейс під час введення, перегляду чи редагування даних у базі даних. Форма створюється на основі однієї або кількох таблиць (запитів) і містить вибрані користувачем поля з цих таблиць (запитів). Крім полів форма може мати й інші об’єкти: малюнки, текстові написи, діаграми, елементи керування – кнопки, прапорці, перемикачі тощо. Дані, введені користувачем у певні поля форми, зберігаються в тих таблицях, на основі яких була створена форма. Крім того, у формах можна створити поля, в яких буде проведено обчислення нового значення на основі значень з інших полів. У СУБД **Access 2007** виділяють кілька видів форм:

- **звичайна (стовпцева або одна форма)** – форма, в якій поля розміщуються у кілька стовпців з підписами полів зліва під поля. На екран виводяться дані з одного запису (рис. 3.24);
- **таблиця (таблиця даних)** – форма, аналогічна об’єкту бази даних **таблиця** з тією тільки різницею, що до форми можна включати не всі поля таблиці, а також можна створювати поля, в яких виконувати обчислення на основі даних з інших полів. Графічні дані у формі цього виду не відображаються. На екран виводяться дані з кількох записів;

- **кілька елементів (таблична)** – поля у цій формі розміщуються у вигляді таблиці, однак, на відміну від попереднього виду, ця форма має заголовок, у який, як і в основне поле, можна помістити текстові та графічні об'єкти, елементи керування. Графічні дані, що містяться в полях, відображаються. На екран виводяться дані з кількох записів (рис. 3.21);
- **за шириною** – форма, в якій поля, як і в звичайній формі, розміщені в стовпчик, тільки підписи полів розміщуються над полем. На екран виводяться дані з одного запису;
- **розділена форма** – форма, що містить і звичайну форму, і таблицю. Переглядати і редагувати дані можна і в формі, і в таблиці (рис. 3.42);
- **форма, що містить таблицю**, – також містить два об'єкти – форму і таблицю, однак, на відміну від попереднього виду, таблиця може бути не пов'язана із формою, наприклад, у формі можуть відображатися дані про працівника, а в таблиці – дані про завдання, які він виконує;
- **кнопкова (модальна) форма** – форма, що не містить даних, а містить елементи керування для виконання певних дій або ініціалізації виконання макросів.

Як і таблиці, форми можуть бути відкриті у різних поданнях:

- **Режим форми;**
- **Режим розмічування;**
- **Конструктор;**
- **Подання таблиці**

тощо.

Для переходу від одного режиму до іншого використовується список кнопки **Вигляд групи Подання** вкладки **Основне**.

Створення нової форми на основі таблиці бази даних

Як уже зазначалося, нові форми бази даних можна створювати тільки на основі вже створених таблиць або запитів. Є кілька способів створення форм з використанням елементів керування групи **Форми** вкладки **Створити** (рис. 3.41):

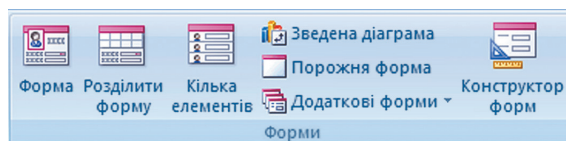






Рис. 3.41. Елементи керування групи **Форми**

- вибором кнопки **Форма**  створюється *звичайна*, або *стовпцева, форма*;
- вибором кнопки **Розділити форму**  створюється *розділена форма* (рис. 3.42);
- вибором кнопки **Кілька елементів**  створюється форма *кілька елементів (таблична)*;
- вибором кнопки **Порожня форма**  створюється *звичайна*, або *стовпцева, форма* шляхом перетягування полів з вікна **Список полів** у область форми;

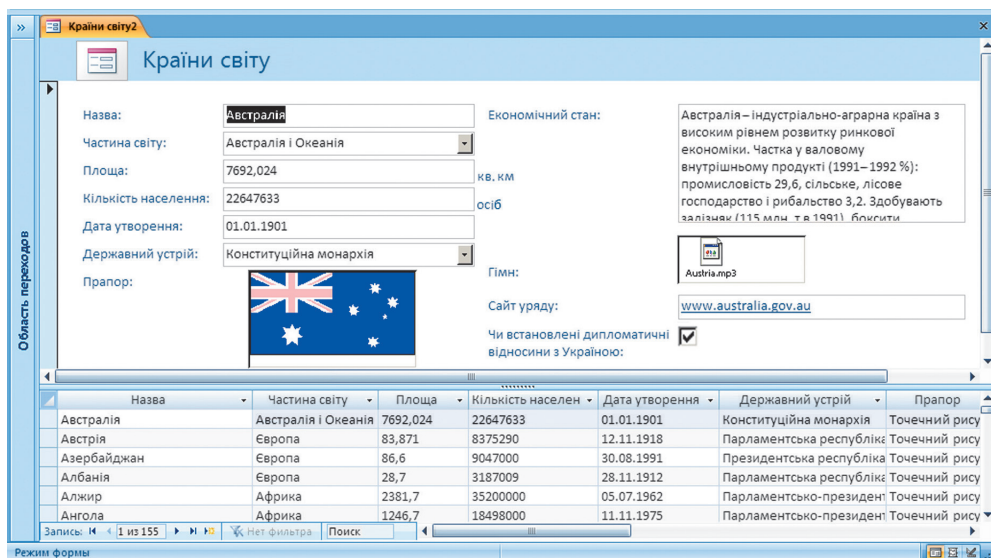


Рис. 3.42. Розділена форма **Країни світу**, створена на основі однойменної таблиці

- з використанням списку **Додаткові форми** можна створити:
 - різні види форм, використовуючи **Майстер форм**;
 - форму **Таблиця**, використовуючи однойменну команду;
 - кнопкову (модальну)** форму, використовуючи команду **Модальний діалог**;
- вибором кнопки **Конструктор форм** можна перейти до створення форми довільного виду, використовуючи елементи керування вкладки **Конструктор** тимчасової вкладки **Знаряддя конструктора форм**.

Під час використання кнопок керування групи **Форми** – **Форма**, **Розділити форму**, **Кілька елементів**, а також команди **Таблиця** списку **Додаткові форми** форма створюється автоматично на основі відкритої в поточний момент таблиці або на основі об'єкта, виділеного в **Області переходів**. Як правило, після створення форма відкривається в поданні **Режим розмічування**. У цьому поданні користувач має можливість переглянути вигляд форми, дані, які вона містить, і внести зміни в оформлення, але вносити нові дані в форму і редагувати вже існуючі дані користувач не може. Для цього слід відкрити форму в поданні **Режим форма**.

Перед внесенням змін у дані необхідно зберегти форму. Для цього слід вибрати кнопку **Зберегти** на **Панелі швидкого доступу** і ввести ім'я форми у відповідне поле вікна **Зберегти як**.

Створення нової форми з використанням Майстра форм

Розглянуті вище способи створення форм передбачають, що до нової форми будуть включені всі поля обраної таблиці, їх розміщення та розміри визначаються програмою. Більше можливостей для добору полів та їхнього розміщення надає користувачу **Майстер форм**. Для його запуску слід виконати **Створити** ⇒ **Форми** ⇒ **Додаткові форми** ⇒ **Майстер форм**. Після цього необхідно виконати таку послідовність дій:

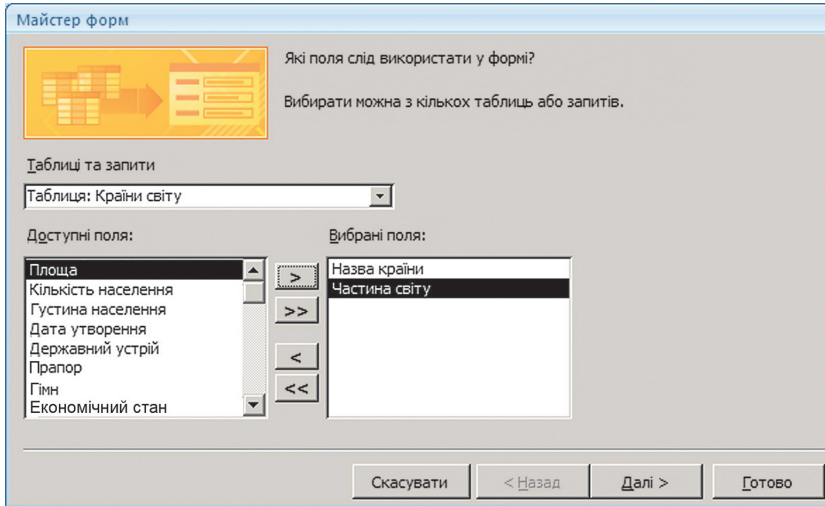



Рис. 3.43. Вікно **Майстра форм** на першому кроці його роботи

1. Вибрати у списку **Таблиці та запити** (рис. 3.43) ім'я таблиці або запити, поля з яких будуть включені до форми.
2. Вибрати у списку **Доступні поля** ім'я поля, яке планується включити до форми.
3. Вибрати кнопку  для переміщення імені поля до списку **Вибрані поля**.
4. Повторити дії, описані в пунктах 2 і 3, потрібну кількість разів для включення до списку **Вибрані поля** всіх потрібних імен полів з обраної таблиці або запити.
5. Вибрати за потреби в списку **Таблиці та запити** ім'я іншої таблиці або запити та перемістити описаним вище способом імена потрібних полів у список **Вибрані поля**.
6. Виконати дії, описані в п. 5, потрібну кількість разів.
7. Вибрати кнопку **Далі**.
8. Вибрати один з видів макета форми (*Стовпцева, Табличний, Таблиця даних, За шириною*), використовуючи відповідний перемикач.
9. Вибрати кнопку **Далі**.
10. Вибрати у списку стиль оформлення форми (наприклад, *Потік*).
11. Вибрати кнопку **Далі**.
12. Увести ім'я форми у поле **Виберіть назву для форми** (наприклад, *Країни світу*).
13. Вибрати дію, що буде виконуватися після завершення роботи майстра, використовуючи відповідний перемикач: *Відкрити форму для перегляду або вводу даних* чи *Змінити макет форми*.
14. Вибрати кнопку **Готово**.

Редагування та форматування форми у Режимі розмічування

Після завершення роботи **Майстра форм** або після створення нової форми іншими способами створені форми, як правило, потребують реда-

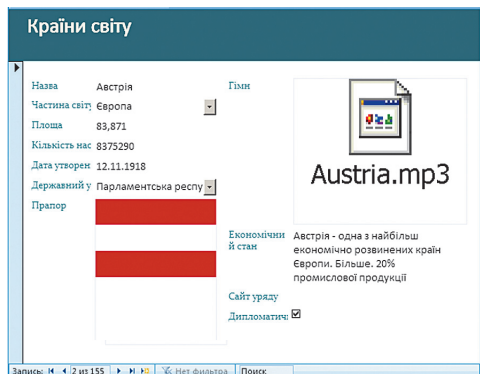



Рис. 3.44. Звичайна (стовпцева) форма
Країни світу

взаємопов'язаних елементів – напису, який містить ім'я поля, і самого поля (рис. 3.45). У свою чергу кожна пара цих елементів розміщена в таблиці. Кожний стовпець такої форми є таблицею. Користувач може змінити розміри цієї таблиці шляхом перетягування меж. Для виділення одного з об'єктів форми необхідно вибрати цей об'єкт. Для виділення всієї таблиці об'єктів слід вибрати один з об'єктів таблиці, а потім – маркер  у лівому верхньому куті таблиці.

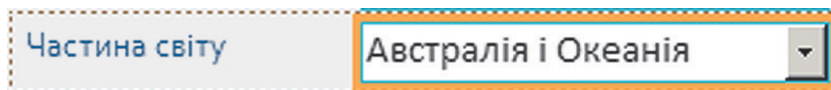


Рис. 3.45. Об'єкт форми з двох елементів – напис (зліва) і поле

У **Режимі розмічування** для виділених об'єктів форми, використовуючи елементи керування групи **Шрифт** вкладки **Формат** тимчасової вкладки **Знаряддя для макетів форм**, можна змінити:

- шрифт символів, їх розмір, колір, стиль і вирівнювання;
- колір заливки тла;
- умовне форматування вмісту полів.

Елементи керування групи **Форматування** тієї самої вкладки надають користувачу можливість установити різні формати числових даних, а групи **Сітка** – стиль, ширину, колір і вид сітки таблиці подання даних у формі.

Об'єкти групи **Елементи керування** вкладки **Формат** тимчасової вкладки **Знаряддя для макетів форм** (рис. 3.46) призначені для:

- кнопка **Логотип** – вставлення зображення у заголовок форми;
- кнопка **Назва** – вставлення назви форми в її заголовок;
- кнопка **Дата й час** – вставлення поточної дати й часу в заголовок форми;
- кнопки **Товщина лінії**, **Тип лінії**, **Колір лінії** – встановлення формату відображення ліній контуру поточного об'єкта форми;
- кнопка **Додати наявні поля** – відкриття вікна зі списком полів таблиці бази даних для їх вставлення у форму. Вставлення відбу-

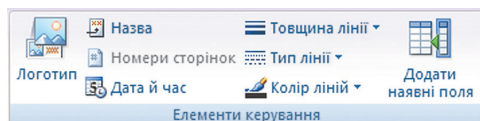


Рис. 3.46. Група **Елементи керування**

вається шляхом перетягування імені поля з вікна зі списком на макет форми.

Використовуючи елементи керування групи **Автоформат** (рис. 3.47), можна змінити тему оформлення форми. Зміна відбувається аналогічно до подібних операцій, розглянутих у **Word 2007, PowerPoint 2007**.

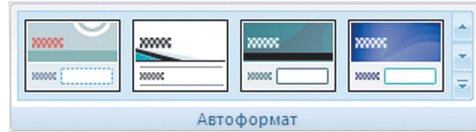


Рис. 3.47. Група **Автоформат**

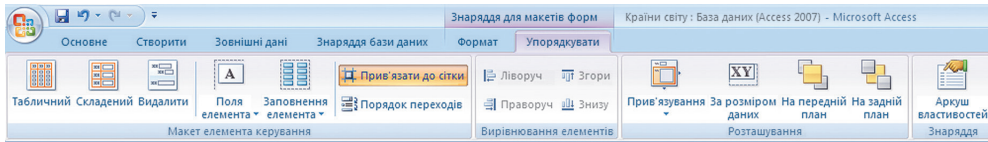





Рис. 3.48. Вкладка **Упорядкувати** тимчасової вкладки **Знаряддя для макетів форм**

Інструменти для змінення розміщення об'єктів форми знаходяться на вкладці **Упорядкувати** тимчасової вкладки **Знаряддя для макетів форм** (рис. 3.48). Група **Макет елемента керування** містить елементи керування для змінення положення окремого об'єкта або виділеної групи об'єктів. Вибір кнопки **Табличний**  приводить до виділення вибраного об'єкта зі стовпця таблиці в окрему таблицю з розміщенням напису над полем у заголовку форми (рис. 3.49). За вибору кнопки **Складений**  об'єкт також виділяється в окрему таблицю з розміщенням напису зліва від поля (рис. 3.50). Цей об'єкт можна розмістити в будь-якому місці форми, використовуючи для перетягування маркер таблиці .

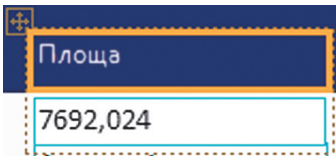


Рис. 3.49

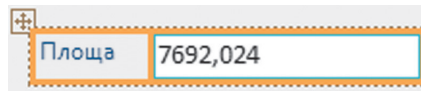





Рис. 3.50

Кнопка **Видалити**  використовується для видалення об'єкта із застосованого макета форми. Об'єкт видаляється з таблиці та розміщується в довільному місці форми. Положення об'єкта можна змінити перетягуванням його меж або переміщенням. При цьому можна окремо змінювати розміри напису і поля.

За вибору кнопки **Поля елемента**  відкривається список, у якому користувач може встановити відступи тексту, чисел чи малюнків від краю напису чи поля. Можливі варіанти: *Немає, Вузьке, Середнє, Широке*. Використовуючи список кнопки **Заповнення елемента** , можна змінити відстані між елементами таблиці у формі.

Редагування форми з використанням Конструктора

Для редагування форми доволі часто використовують **Конструктор**, який надає ширші можливості щодо змінення взаємного розміщення об'єктів, вставки елементів керування тощо. Перехід до подання **Конструктор** здійснюється описаним вище способом. Вигляд форми в поданні **Конструктор** представлено на рисунку 3.51.

Однією з основних операцій редагування в поданні **Конструктор** є зміна положення і розмірів об'єктів – полів і підписів до них. Здійснюється вона аналогічно до операцій у поданні **Режим розмічування**.

Також доволі часто у формах змінюються підписи до полів. При включенні поля до форми його підпис утворюється з імені поля з додаванням двокрапки. Однак якщо для імені поля в таблиці одна з головних умов – стислість, то у формі навпаки – підпис поля повинен максимально пояснювати дані, що є вмістом поля. Так замість підпису **Назва** бажано ввести **Назва країни**, замість **Дата утворення** бажано увести більш точний підпис **Дата утворення або проголошення незалежності** і т. д. Для редагування вмісту підпису необхідно встановити курсор всередину підпису і змінити текст з використанням стандартних операцій редагування тексту. При цьому слід зауважити, що якщо текст підпису довгий, то його можна розмістити у кілька рядків. Для цього слід перетягуванням межі збільшити висоту рамки підпису.

- | | |
|--------------------------------|---------------------------------|
| ① Область нижнього колонтитула | ④ Поля |
| ② Область Подобиці | ⑤ Область верхнього колонтитула |
| ③ Підписи полів | |

Рис. 3.51. Форма у поданні **Конструктор**

Для пояснення вмісту полів використовують не тільки підписи полів, а й окремі написи. Наприклад для пояснення, в яких одиницях подана площа країн у цій базі даних, бажано після поля вставити напис з одиницями вимірювання. Для цього слід виконати таку послідовність дій:

1. Вибрати кнопку **Напис** **Aa** групи **Елементи керування** вкладки **Конструктор** тимчасової вкладки **Знаряддя конструктора форм**.
2. Виділити в області **Подробиці** місце розміщення напису.
3. Увести у напис текст *кв. км.*
4. Уточнити розміщення напису шляхом перетягування об'єкта та його меж.
5. Переглянути результати у поданні **Режим форми**.

Як уже згадувалося, форми можуть містити поля, в яких здійснюється обчислення на основі даних, що містяться в інших полях. Розглянемо послідовність створення такого поля на прикладі створення поля **Густота населення**, значення якого обчислюється шляхом ділення значення поля **Кількість населення** на значення поля **Площа**. Для цього потрібно виконати таку послідовність дій:

1. Вибрати кнопку **Текстове поле** **ab** групи **Елементи керування** вкладки **Конструктор** тимчасової вкладки **Знаряддя конструктора форм**.
2. Виділити в області **Подробиці** місце розміщення текстового поля.
3. Уточнити розміщення обох частин текстового поля шляхом перетягування об'єкта та меж.
4. Увести підпис поля **Густота населення** замість стандартного **Текст №**.
5. Увести в поле формулу для підрахунку густоти населення країни у вигляді $=[\text{Кількість населення}]/[\text{Площа}]$.

Країни

Назва країни: Японія

Частина світу: Азія

Площа: 377944 кв. км

Кількість населення: 127767994 осіб

Державний устрій: Конституційна монархія

Державний гімн: Японія.mid

Сайт уряду: http://www.kantei.go.jp

Чи встановлено дипломатичні відносини з Україною: ☒

Густота населення: 338,061 осіб на кв. км

Дата утворення держави або здобуття незалежності: 12.02.1603

Короткий опис економіки країни: Економіка Японії є однією з найпотужніших у світі. Вона займає друге місце після США за показниками ВВП (\$4.5 трильйони в 2005) і третє місце після США і КНР за показниками купівельної спроможності. Японська економіка є найбільшою в Азії. Вона сильно залежить від поставок сировинних матеріалів з інших країн через нестачу природних ресурсів. Співпраця уряду і промисловців, розвинена культура бізнесової етики, розробки високих технологій та відносно малі витрати на оборонні потреби забезпечили стрибкоподібне зростання японської економіки після Другої світової війни. З 1960-х по

Записи: 155 из 155

Нет фильтра Поиск




Рис. 3.52. Форма **Країни** в поданні **Режим форми**

6. Переглянути результати в поданні **Режим форми**.

Для форматування об'єктів форми в поданні **Конструктор** використовуються операції, аналогічні до описаних вище під час роботи в поданні **Режим розмічування**. Вигляд форми в поданні **Режим форми** зображено на рисунку 3.52.



Перевірте себе

- 1°. Для чого використовуються форми в базах даних?
- 2°. Укажіть основні види форм, що використовуються в СУБД Access 2007.
- 3°. Поясніть, чим відрізняється таблиця бази даних від форми виду *таблиця*.
- 4°. Поясніть, чим відрізняється вид форми *таблиця* від виду форми *таблична*.
- 5°. Поясніть, чим відрізняється вид форми *стовпцева* від виду форми *за шириною*.
- 6°. З якою метою створюється форма виду *розділена*? Опишіть загальний вигляд такої форми.
- 7°. Назвіть види подання, у яких можуть відкриватися форми бази даних. Опишіть послідовність переходу від одного подання до іншого.
- 8°. Опишіть послідовність створення форми з використанням кнопки **Форма**  групи **Форми** вкладки **Створити**.
- 9°. Опишіть послідовність збереження створеної форми.
10. Поясніть, як створити стовпцеву форму на основі однієї з таблиць бази даних, використовуючи **Майстер форм**.
- 11°. Які види форм можна створити, використовуючи **Майстер форм**?
- 12°. У яких поданнях можна редагувати форму?
- 13°. Опишіть, як змінити положення окремих об'єктів форми в поданні **Режим розмічування**.
- 14°. Для чого використовується кнопка **Табличний** , а для чого – кнопка **Складений**  групи **Макет елемента керування** вкладки **Упорядкування** тимчасової вкладки **Знаряддя для макетів форм**?
- 15°. Поясніть призначення об'єктів групи **Елементи керування** вкладки **Формат** тимчасової вкладки **Знаряддя для макетів форм**.
- 16°. Для чого використовуються елементи керування групи **Шрифт** вкладки **Формат** тимчасової вкладки **Знаряддя для макетів форм**?
- 17°. Опишіть послідовність дій для зміни теми оформлення форми.
- 18°. Опишіть вікно форми в поданні **Конструктор** за рисунком 3.51.
- 19°. Які операції редагування форми можна виконати, використовуючи подання **Конструктор**?
- 20°. Опишіть послідовність включення напису до форми в поданні **Конструктор**.
- 21*. Поясніть, як створити поле, у якому здійснюється опрацювання даних, що містяться в інших полях.



Виконайте завдання


- 1°. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.6\зразок 3.6.1.accdb**) та:
 1. Увімкніть повний вміст бази даних.
 2. Відкрийте **Область переходів** і перегляньте таблиці бази даних.
 3. Використовуючи кнопку **Форма** , створіть форму бази даних.

Рис. 3.53



4. Відредагуйте та відформатуйте її відповідно до рисунка 3.53 у поданні **Режим розмічування** чи **Конструктор**.
 5. Уведіть, використовуючи форму, дані до п'яти записів на власний розсуд.
 6. Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.6.1.accdb**.
-  **2***. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.6\зразок 3.6.2.accdb**) та:
1. Увімкніть повний вміст бази даних.
 2. Відкрийте **Область переходів** і перегляньте таблиці бази даних.
 3. Використовуючи кнопку **Розділити форму** , створіть форму бази даних.
 4. Відредагуйте і відформатуйте її відповідно до рисунка 3.54 у поданні **Режим розмічування** чи **Конструктор**.
 5. Уведіть, використовуючи форму, дані до п'яти записів на власний розсуд.
 6. Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.6.2.accdb**.

Рис. 3.54

Рис. 3.55

- 3*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.6\зразок 3.6.3.accdb**) та:
1. Увімкніть повний вміст бази даних.
 2. Відкрийте **Область переходів** і перегляньте таблиці бази даних.
 3. Перегляньте зв'язки в базі даних і визначте, який вид зв'язку встановлено і між якими полями таблиць.
 4. Використовуючи кнопку **Майстер форм**, створіть форму бази даних.
 5. Відредагуйте і відформатуйте її відповідно до рисунка 3.55 у поданні **Конструктор**.
 6. Створіть на формі поле для підрахунку суми наявного товару, враховуючи кількість і вартість відповідного товару.
 7. Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.6.3.accdb**.

3.7. Сортювання, пошук і фільтрування даних у базі даних



1. Що таке сортювання? Як здійснюється сортювання в таблицях **Word 2007**, **Excel 2007**? Які особливості сортювання різних типів даних?
2. Пошук яких даних можна здійснювати в **Word 2007**, **Excel 2007**?
3. Що таке фільтр в **Excel 2007**? Як здійснюється фільтрування в **Excel 2007**?
4. Як в **Excel 2007** провести фільтрування даних, що мають значення більше за 0, але менше за 100?


Сортювання даних

Як і в таблицях **Word 2007** та **Excel 2007**, дані в таблицях баз даних можна сортювати. За замовчуванням дані в **Access 2007** сортюються за зростанням значень по ключовому полю під час відкриття таблиці. Для зміни порядку сортювання слід виконати таку послідовність дій:

1. Відкрити таблицю бази даних, дані в якій потрібно відсортювати.
2. Установити курсор у межах поля, за даними якого буде виконано сортювання записів.
3. Виконати **Основне** ⇒ **Сортювання й фільтр** ⇒ **За зростанням (За спаданням)**.

Назва ▾	Частина світу ▾	Площа ▾	Кількість населення ▾
Катар	Азія	11	565
Ліван	Азія	10,4	3496
Бахрейн	Азія	0,6	640
Мальдівська Республіка	Азія	0,3	291
Австралія	Австралія і Океанія	7636	19138
Нова Зеландія	Австралія і Океанія	268,7	3778
Соломонові Острови	Австралія і Океанія	29	447
Тонга	Австралія і Океанія	0,7	100

Рис. 3.56. Фрагмент таблиці бази даних **Країни світу**, відсортованої за даними полів **Частина світу** та **Площа**

Біля імені поля, за даними якого здійснено сортування, з'являється стрілочка, яка вказує на вид сортування: ↑ – за зростанням, ↓ – за спаданням. Для того щоб відмінити сортування, необхідно вибрати кнопку **Очистити все сортування**  групи **Сортування й фільтр** вкладки **Основне**. Для сортування за даними кількох полів з однаковими значеннями параметрів сортування слід виділити ці поля (виділити можна лише сусідні поля) і виконати **Основне** ⇒ **Сортування й фільтр** ⇒ **За зростанням (За спаданням)**. При сортуванні за даними кількох полів спочатку сортування відбувається за даними полів, розміщених ліворуч. Фрагмент таблиці **Країни світу**, відсортований за спаданням за даними полів **Частина світу** та **Площа**, наведено на рисунку 3.56.

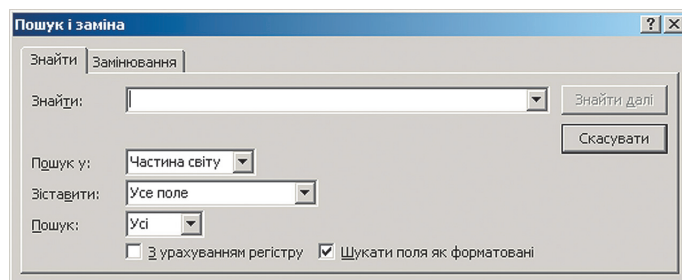
Можна виконати сортування за даними кількох полів, довільно розміщених у таблиці, послідовно виконавши сортування для кожного з них. При цьому можна використати різні значення параметрів сортування.

Сортування даних у формі виконується аналогічно сортуванню в таблиці, тільки у формі не можна виділити одночасно кілька полів для встановлення однакових параметрів сортування.

Пошук даних

Як було зазначено раніше, бази даних призначено для забезпечення швидкого доступу до потрібних даних. Досить часто користувачі здійснюють пошук необхідних даних у базах даних: залізниці, бібліотеки, пошукового сервера Інтернету та ін. У СУБД **Access 2007** пошук у базі даних здійснюється подібно до пошуку в **Excel 2007**. Для пошуку потрібних даних необхідно:

1. Відкрити файл бази даних, у якій потрібно здійснити пошук.
2. Відкрити таблицю або форму, у яких здійснюватиметься пошук.
3. Виконати **Основне** ⇒ **Пошук** ⇒ **Знайти**.
4. У вікні **Пошук і заміна** (рис. 3.57) на вкладці **Знайти** у полі **Знайти** ввести зразок даних, за яким здійснюватиметься пошук.
5. Вибрати у списку **Пошук у ім'я поля таблиці або форми**, у якому буде здійснено пошук.
6. Установити у списку **Зіставити одне із значень: Усе поле, Будь-яка частина поля, Початок поля**.
7. Установити у списку **Пошук один з напрямів пошуку: Усі, Угору, Вниз**.

Рис. 3.57. Вікно **Пошук і заміна**

8. Установити за потреби відповідні позначки прапорців для пошуку з урахуванням регістру та пошуку з використанням шаблонів введення (**Шукати поля як форматовані**).

9. Вибрати кнопку **Знайти далі**.

Для переходу до наступних записів, значення яких відповідають пошуковому запиту або параметрам пошуку, слід послідовно натискати кнопку **Знайти далі**.

Значення у списку **Пошук** встановлюють напрям пошуку від поточного запису: *Угору* – до першого запису, *Вниз* – до останнього запису і *Усі* – по всій таблиці (формі).

Заміна даних

Заміна даних у полях таблиці чи форми виконується з використанням елементів керування вкладки **Замінювання** вікна **Пошук і заміна**. Крім описаних параметрів для пошуку, під час заміни в поле **Замінити на** вводяться дані, які повинні замінити знайдені. Заміну можна здійснювати по кроках (кнопки **Знайти далі** і **Замінити**) або одразу всіх знайдених даних (кнопка **Замінити все**).



Використання символів узагальнення для пошуку даних

Під час пошуку даних можна використати спеціальні символи, які ще в **Access 2007** називають **символами узагальнення**. У базах даних, створених у **Access 2007** і попередніх версіях **Access**, використовувалися символи узагальнення за стандартом **ANSI-89** (один зі стандартів мови запитів **SQL**):

- * – будь-яка кількість будь-яких символів, у тому числі й їх відсутність;
- ? – будь-який один символ;
- [] – будь-які символи, вказані у квадратних дужках;
- [!] – будь-які символи, крім тих, що вказані у квадратних дужках перед знаком оклику;
- - – будь-який символ з послідовності символів, наприклад *д-к* відповідає символам від *д* до *к* в алфавіті;
- # – будь-яка одна цифра.

Фільтрування даних

Для відбору записів, дані в яких відповідають певним умовам, використовують фільтри. Для фільтрування даних використовують елементи керування групи **Сортування й фільтр** вкладки **Основне**. Наприклад, для знаходження записів про країни, у яких державний устрій *монархія* або

конституційна монархія, з використанням кнопки **Фільтр** потрібно виконати таку послідовність дій:

1. Зробити поточним поле, за даними якого здійснюватиметься фільтрування, наприклад **Державний устрій**.
2. Вибрати кнопку **Фільтр**.
3. Установити у списку (рис. 3.58) позначки прапорців біля одного або кількох значень, які відповідають умові фільтрування, наприклад *Монархія* і *Конституційна монархія*.
4. Вибрати кнопку **ОК**.

Можна здійснювати фільтрування за кількома полями. Наприклад, для визначення країн *Африки*, у яких державний устрій *монархія* або *конституційна монархія*, слід після проведення фільтрування за полем **Державний устрій** ще провести фільтрування для поля **Частина світу** зі значенням параметра фільтрування *Африка*.

Для відміни фільтрування необхідно виконати **Оснoвне** ⇒ **Сортування й фільтр** ⇒ **Відмінити фільтр**. Для повторного застосування уже визначених значень параметрів фільтрування слід знову виконати **Оснoвне** ⇒ **Сортування й фільтр** ⇒ **Застосувати**.

Під час проведення фільтрування з використанням кнопки **Фільтр** є можливість задати більш складні умови фільтрування. Так, у списку фільтрування будь-якого текстового поля під час наведення вказівника на напис **Текстові фільтри** (рис. 3.58) відкривається список умов фільтрування (рис. 3.59). Вибір будь-якого елемента цього списку відкриває додаткове вікно з полем для введення фрагмента тексту, що стане складовою відповідної умови: *Дорівнює*, *Не містить*, *Закінчується* тощо.

Якщо поле містить числові дані, то список умов буде іншим: *Дорівнює*, *Не дорівнює*, *Більше*, *Менше*, *Між*, а для даних дати й часу – *Дорівнює*, *Не дорівнює*, *Перед*, *Після*, *Між*.

Для застосування **фільтра по виділеному** необхідно виконати таку послідовність дій:

1. Виділити фрагмент даних в одному з полів, за яким буде здійснено фільтрування, наприклад *Африка*.
2. Відкрити список кнопки **Виділення**

Виділення (рис. 3.60).

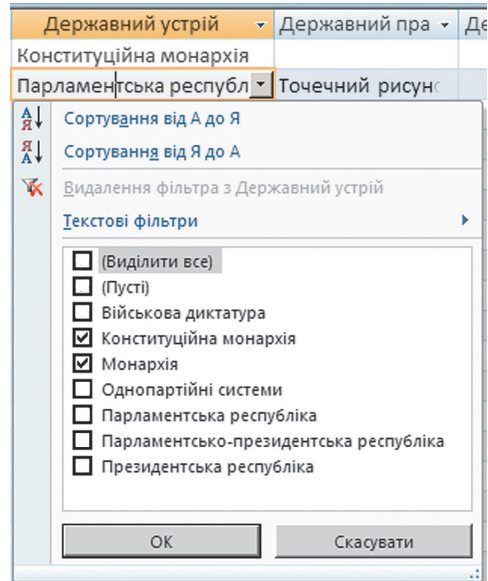


Рис. 3.58. Список фільтрування поля **Державний устрій**

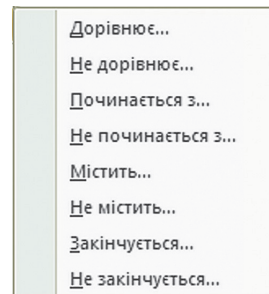


Рис. 3.59. Список вибору додаткових умов фільтрування

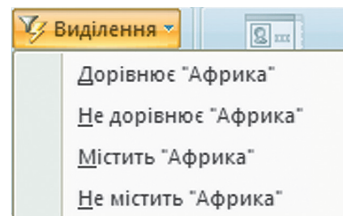



Рис. 3.60. Список кнопки **Виділення**

3. Вибрати у списку одну з умов застосування виділеного фрагмента для фільтрування даних, наприклад **Дорівнює "Африка"**.



Для видалення фільтра певного поля слід виконати таку послідовність дій:

1. Вибрати поле, за даними якого було проведено фільтрування, наприклад **Державний устрій**.
2. Вибрати кнопку **Фільтр**.
3. Вибрати кнопку **Видалити фільтр з Державний устрій** .

Для видалення всіх фільтрів необхідно виконати **Основне** \Rightarrow **Сортування й фільтр** \Rightarrow **Додатково** \Rightarrow **Очистити всі фільтри**.



Перевірте себе

- 1°. Що таке сортування? Як виконати сортування за даними одного з полів?
- 2°. Опишіть послідовність сортування за значеннями полів, що розміщені поруч, з однаковими значеннями параметрів сортування.
- 3°. Опишіть послідовність сортування за значеннями полів, розміщених довільно в таблиці, з різними значеннями параметрів сортування.
- 4°. Як відмінити сортування?
- 5*. Порівняйте здійснення сортування в таблицях **Word 2007**, **Excel 2007** та **Access 2007**. Що спільного і чим відрізняються ці операції в зазначених прикладних програмах **Microsoft Office 2007**?
- 6°. Опишіть послідовність виконання пошуку, наприклад слова *інформатика*, що міститься на початку поля; у будь-якій частині вмісту поля.
- 7°. Чи можна здійснити пошук тільки в межах одного поля; одного запису?
- 8*. Як переглянути всі знайдені в результаті пошуку записи?
- 9*. Опишіть послідовність здійснення заміни вмісту всього поля.
- 10*. Як замінити фрагмент тексту *метр* на *кілометр*, враховуючи, що цей фрагмент може трапитися в різних частинах вмісту полів? Як провести цю заміну максимально швидко?
-  11°. Для чого використовуються символи узагальнення під час пошуку даних?
-  12°. Опишіть використання символів узагальнення [] та [-] у ході пошуку даних.
- 13*. Порівняйте здійснення пошуку і заміни в таблицях **Word 2007**, **Excel 2007** та **Access 2007**. Що спільного і чим відрізняються ці операції в зазначених прикладних програмах **Microsoft Office 2007**?
- 14°. Що таке фільтрування?
- 15°. Поясніть послідовність фільтрування за значенням одного із полів таблиці.
- 16°. Поясніть, як вивести записи, що містять хоча б одне поле, в яке не введено значень.
- 17°. Як визначити кількість записів, що задовольняють умови фільтрування?
- 18°. Як здійснити фільтрування за значеннями трьох полів?
- 19°. Чим відрізняються умови фільтрування, які можна встановити для полів з даними типу **Текст** від полів з даними **Число**, **Дата й час**?
- 20°. Як відмінити фільтрування, як його застосувати знову з тими самими значеннями параметрів?

- 21°. Порівняйте результати виконання операцій пошуку та фільтрування. Що у них спільного та у чому відмінність?



Виконайте завдання

- 1°. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.7\зразок 3.7.1.accdb**). Увімкніть повний вміст бази даних. Відкрийте таблицю **Країни** та виконайте сортування даних:
 - 1) Відсортуйте записи за даними поля **Кількість населення за спаданням**. Визначте п'ять країн з найбільшою кількістю населення. Визначте, яке місце займає **Україна** в світі за кількістю населення.
 - 2) Виконайте сортування за даними полів **Кількість населення** та **Частина світу** для визначення країн, що мають найбільшу кількість населення в певній частині світу. Визначте, у якій послідовності слід виконувати сортування для швидкого знаходження вказаних значень. Для якого поля (полів) слід виконати сортування за зростанням, а для якого (яких) – за спаданням? Збережіть базу даних у своїй папці у файлі з іменем **вправа 3.7.1.2.accdb**.
 - 3) Відсортуйте записи за даними поля **Державний устрій** і визначте, скільки країн мають державний устрій *монархія* і скільки – *конституційна монархія*.
 - 4) Проведіть сортування записів за даними полів **Державний устрій** і **Частина світу** для визначення кількості країн, що мають державний устрій *президентська республіка* в кожній частині світу. Визначте, у якій послідовності слід виконувати сортування для швидкого знаходження вказаних значень. Для якого поля (полів) слід виконати сортування за зростанням, а для якого (яких) – за спаданням? Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.7.1.4.accdb**.
- 2°. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.7\зразок 3.7.2.accdb**). Увімкніть повний вміст бази даних. Відкрийте таблицю **Країни** та виконайте пошук і заміну даних:
 - 1) Знайдіть країни, у яких грошовою одиницею є: *Крузадо, Кетсаль, Золота кордоба, долар*. Для останньої грошової одиниці здійсніть пошук у будь-якій частині поля.
 - 2) Виконайте заміну значення *Північна Америка* на *Америка*. Збережіть базу даних у власній папці у файлі з іменем **вправа 3.7.2.2.accdb**. Складіть алгоритм здійснення такої заміни.
 - 3) Знайдіть країни, у яких у значенні площі трапляються цифри *1* або *2*. Знайдіть країни, в яких дата утворення *1960* рік.
 - 4) Виконайте заміну значення *Центральна і Південна Америка* на *Америка*. Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.7.2.4.accdb**.
- 3°. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.7\зразок 3.7.3.accdb**). Увімкніть повний вміст бази даних. Відкрийте таблицю **Географічні та фізичні карти** та виконайте фільтрування даних:
 - 1) Виберіть *адміністративні* карти, що виготовлені *на картоні* або *на картоні з планками*. Збережіть базу даних у власній папці у файлі з іменем **вправа 3.7.3.1.accdb**.
 - 2) Відмініть усі фільтри. Виберіть карти *України* з масштабом перегляду *M1:2 100 000* і ціною понад *20 грн*. Скільки таких карт знайдено? Збережіть базу даних у власній папці у файлі з іменем **вправа 3.7.3.2.accdb**.

- 3) Відмініть усі фільтри. Виберіть усі карти, крім карт з даними про Україну і ціною від 5 до 40 грн. Скільки таких карт знайдено? Збережіть базу даних у власній папці у файлі з іменем **вправа 3.7.3.3.accdb**.



Практична робота № 13. Форми. Введення даних. Сортування, фільтрування та пошук даних

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте файл бази даних (наприклад, **Тема 3\Практична 13\Зразок 13**).
2. Доповніть таблицю **Країни** новим полем **Союзи країн**.
3. На основі таблиці **Країни** створіть форму **Країни світу** за зразком, наведеним на рисунку до цієї практичної роботи. Установіть тему оформлення **Строго**.
4. Використайте текстове поле для створення у формі поля з обчисленням густоти населення.
5. Використовуючи форму і пошук, введіть дані про входження певних країн до об'єднання країн: СНГ та ОРЕС (англ. *The Organization of the Petroleum Exporting Countries* – Організація країн-експортерів нафти). Дані про входження країн у ці об'єднання візьміть із файлу **Союзи країн.doc** з папки **Тема 3\Практична 13**.
6. Відкрийте таблицю **Країни** та відсортуйте дані за зростанням по полю **Союзи країн**. Перевірте, чи правильно ви ввели дані про членство країн у названих об'єднаннях.
7. Установіть фільтрування даних: для поля **Площа** – менше ніж 1 тис. кв. км, а для поля **Кількість населення** – понад 1 млн осіб.
8. Збережіть базу даних у своїй папці у файлі з іменем **практична робота 13.accdb**.

3.8. Запити в Access 2007



1. Що таке формула в **Excel 2007**? Які об'єкти може містити формула в **Excel 2007**?
2. Для чого використовуються фільтри в базах даних? Як застосувати фільтр до таблиці бази даних в **Access 2007**?
3. Які оператори використовуються в формулах **Excel 2007**? Як вони позначаються?
4. Опишіть послідовність дій для створення фільтрів для кількох полів.

Поняття про запити в базі даних

Запити є одним з основних інструментів опрацювання даних у базах даних. Запити можуть забезпечувати не тільки пошук даних, які відповідають певним критеріям, як це здійснюється під час фільтрування, але й одночасне виконання операцій над даними та збереження результатів пошуку.

Основними операціями, які може здійснити користувач з використанням запитів, є:

- створення нових таблиць на основі аналізу даних у вже існуючих таблицях бази даних;

- обчислення узагальнених даних (суми, максимального чи мінімального значення тощо) для заданих полів;
 - знаходження значень нових властивостей (проведення обчислень), використовуючи дані з різних таблиць або запитів, наприклад знаходження густоти населення певної країни на основі площі та кількості населення;
 - внесення змін у вже існуючі таблиці (оновлення даних, вставлення і видалення записів тощо).
- Залежно від призначення запитів їх поділяють на:
- **запити на вибірку даних** – запити, з використанням яких на основі існуючих таблиць (рис. 3.61) створюється таблиця з даними, які відповідають певним умовам (рис. 3.62);
 - **перехресні запити** – запити, у яких на першому етапі здійснюється узагальнення даних (знаходиться сума, середнє, максимальне значення тощо), а на другому – групування цих даних за двома наборами даних, один з яких визначає заголовки стовпців таблиці, а другий – заголовки рядків. Наприклад, у таблиці, створеній запитом на вибірку найкращих результатів кожного зі спортсменів на трьох дистанціях (рис. 3.63), заголовками стовпців стали дані з поля **Дистанція**, а заголовками рядків – дані з поля **Спортсмен**. У клітинках на перетині стовпців і рядків указано найкращий (мінімальний) результат певного спортсмена на певній дистанції;
 - **запити на внесення змін** – запити, використовуючи які користувач може змінювати значення у полях певних записів, створювати нові записи або видаляти існуючі записи тощо;

Результати змагань			
Ид	Спортсмен	Дистанція	Результат
2	Голованов Василь	100 м	11,4
3	Петренко Степан	100 м	10,5
4	Федченко Микола	100 м	10,8
5	Хмиз Дмитро	100 м	11,1
6	Суботів Віктор	100 м	10,9
7	Голованов Василь	1500 м	4:01,23
8	Петренко Степан	1500 м	4:02,00
9	Федченко Микола	1500 м	4:03,11
10	Хмиз Дмитро	1500 м	4:02,45
11	Петренко Степан	400 м	54,65
12	Хмиз Дмитро	400 м	51,4
13	Суботів Віктор	1500 м	4:03,22
14	Суботів Віктор	400 м	52,34
15	Голованов Василь	1500 м	4:01,54
16	Голованов Василь	100 м	11,6
17	Федченко Микола	400 м	55,7
18	Федченко Микола	1500 м	4:03,7
19	Голованов Василь	400 м	52,6

Рис. 3.61. Таблиця з результатами виступів спортсменів у різних змаганнях на трьох дистанціях

Результати змагань_перекресний			
Спортсмен	Дистанція	Min-Результат	
Голованов Василь	100 м	11,4	
Петренко Степан	100 м	10,5	
Суботів Віктор	100 м	10,9	
Федченко Микола	100 м	10,8	
Хмиз Дмитро	100 м	11,1	

Рис. 3.62. Таблиця, утворена як результат запити на вибірку найкращих результатів спортсменів на дистанції 100 м

Результати змагань_перехресний				
Спортсмен	100 м	1500 м	400 м	
Голованов Василь	11,4	4:01,23	52,6	
Петренко Степан	10,5	4:02,00	54,65	
Суботів Віктор	10,9	4:03,22	52,34	
Федченко Микола	10,8	4:03,11	55,7	
Хмиз Дмитро	11,1	4:02,45	51,4	

Рис. 3.63. Таблиця, утворена як результат запити на вибірку найкращих результатів спортсменів на трьох дистанціях

- **запити з параметрами** – запити, після запуску на виконання яких користувач має ввести значення певних параметрів, за якими буде здійснено опрацювання даних. Наприклад, можна створити запит, що

буде створювати таблицю – список учнів того з класів певної школи, дані про який потрібні користувачу в даний момент. Після запуску такого запиту на виконання буде відкрито вікно з полем, у яке користувач повинен ввести назву потрібного класу. Залежно від уведеного значення буде побудована таблиця – список учнів указанного класу.


Запит зберігається у базі даних під певним іменем. Під час відкриття запиту відбувається опрацювання даних відповідно до умов запиту і результати запиту виводяться на екран у вигляді таблиці.

Вирази та оператори в запитах

У запитах для запису умов відбору даних або для виконання дій над даними використовують **вирази**. Вираз в **Access 2007** схожий на формулу в **Excel 2007**. Вирази можуть містити:

- **ідентифікатори** – імена полів таблиць та елементів керування у формах і звітах;
- **оператори** – послідовність символів для позначення операцій. Розрізняють оператори: *арифметичні, порівняння, логічні, об'єднання, специфічні*;
- **функції** – аналогічні за призначенням і використанням функціям в **Excel 2007**, однак в **Access 2007** трохи інший набір функцій і вони згруповані за іншими категоріями (*масиви, перетворення, база даних, математичні, фінансові, текст та ін.*);
- **константи** – незмінні значення, наприклад фрагмент тексту або число;
- **круглі дужки** – для встановлення пріоритету операцій у виразі. Під час створення виразів слід дотримуватися певних правил:
- імена полів та інші ідентифікатори записуються в квадратних дужках;
- при посиланні на поле певної таблиці використовується знак оклику, наприклад вираз **[Товари]![Код товару]** містить посилання на поле **Код товару** таблиці **Товари**;
- текст записується в лапках **""**.

Запити на вибірку

Розглянемо створення запиту на вибірку з використанням **Майстра запитів**. Для бази даних **Країни світу** створимо запит, який би містив дані про площу країн. Для створення запиту необхідно відкрити базу даних **Країни світу** та вибрати кнопку **Майстер запиту**  групи **Додатково** вкладки **Створити**. У подальшому слід виконати таку послідовність дій:

1. Вибрати у вікні **Новий запит** у списку видів майстрів запитів **Майстер простих запитів**, після чого вибрати кнопку **ОК**.
2. Вибрати у списку **Таблиці і запити** вікна **Майстер простих запитів** таблицю, на основі якої буде створено запит, наприклад таблицю **Країни світу**.
3. Перемістити зі списку **Доступні поля** у список **Вибрані поля** потрібні імена полів, наприклад **Назва**, **Частина світу**, **Площа**.
4. Вибрати один з варіантів відображення записів із даними: *докладно* (без виконання операцій зведення) або з використанням *зведення* (знаходження *суми, середнього значення, мінімуму, максимуму* для певних полів) вибором відповідного перемикача. (Цей і наступ-

- ний крок виконується тільки для записів, що містять числові дані.)
5. Вибрати за потреби кнопку **Параметри зведення** та встановити для кожного з полів потрібні операції зведення вибором відповідних позначок прапорців. Вибрати кнопку **ОК**.
 6. Увести назву запиту в поле **Виберіть назву для запиту**, наприклад **Площа країн світу**.
 7. Вибрати варіант продовження роботи із запитом – *відкрити запит для перегляду даних* чи *перейти до зміни макета запиту* вибором відповідного перемикача.
 8. Вибрати кнопку **Готово**.

Якщо ви вибрали варіант *відкрити запит для перегляду даних*, то на екран буде виведено таблицю результатів запиту. Це нова таблиця, що містить поля, які були вибрані під час конструювання запиту.

Якщо ж обрано другий варіант – *перейти до зміни макета запиту*, то новий запит відкриється у поданні **Конструктор** (рис. 3.64). У цьому поданні у верхній частині робочого поля розміщено список полів таблиці, на основі якої було створено запит. Кількість списків з переліком полів таблиць чи запитів можна змінити, виконавши *Знаряддя для запиту* ⇒ *Конструктор* ⇒ *Настроювання запиту* ⇒ *Відобразити таблицю*.

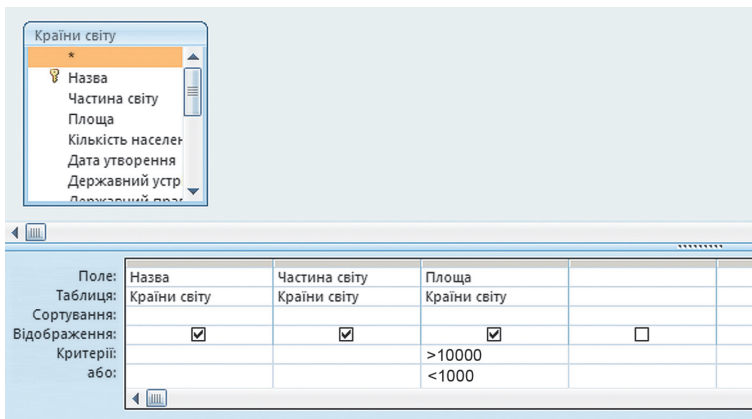


Рис. 3.64. Фрагмент вікна запиту **Площа країн світу** у поданні **Конструктор**

У вікні **Відображення таблиці** слід вибрати назву потрібної таблиці та вибрати **Додати**. Для закриття списку полів таблиці слід у контекстному меню відповідного списку вибрати **Видалити таблицю**.

У нижній частині робочого поля вікна запиту розміщено таблицю конструктора. Вона містить такі рядки:

- **Поле**, в якому відображаються імена обраних полів;
- **Таблиця**, в якому відображається ім'я таблиці, до складу якої входить відповідне поле;
- **Сортування**, в якому може відображатися вид сортування, що буде застосоване до записів даного поля у таблиці запиту;
- **Відображення**, який може містити прапорець, встановлена позначка якого свідчить про те, що дане поле буде відображатися у запиті;
- **Критерії**, який може містити вираз-умову для вибірки даних з відповідного поля;

- **або** – для запису ще одного виразу, який є частиною складеної умови та який поєднується з виразом у рядку **Критерії** логічною операцією **АБО**. Наступні рядки можуть містити інші вирази, які пов'язані з виразами у рядках **Критерії** та **або** логічною операцією **АБО**.

Якщо у рядку **Критерії** записані вирази для кількох стовпчиків (полів), то вони об'єднуються в один вираз логічною операцією **І**.

Наприклад, для того щоб у запиті відображалися лише записи для країн **Австралії і Океанії**, необхідно у рядку **Критерії** у стовпці **Частина світу** записати «**Австралія і Океанія**». Для застосування обраних значень параметрів запиту необхідно перейти до **Подання таблиці**. Для цього потрібно виконати **Знаряддя для запиту** ⇒ **Конструктор** ⇒ **Результати** ⇒ **Запуск**.

Назва	Частина світу	Площа
Австралія	Австралія і Океанія	7636
Нова Зеландія	Австралія і Океанія	268,7
Соломонові Острови	Австралія і Океанія	29
Тонга	Австралія і Океанія	0,7
Підсумок		7934,4

Рис. 3.65. Результат виконання запиту **Площа країн світу** для частини світу **Австралія і Океанія**

До отриманого запиту можна додати рядок з підсумками. Для цього слід виконати **Основне** ⇒ **Записи** ⇒ **Підсумки**, в рядку **Підсумок** у полі **Площа** відкрити список можливих варіантів узагальнення даних і вибрати, наприклад **Сума**. Створену таблицю запиту подано на рисунку 3.65.

Запити з полями, що містять дані, обчислені за даними з інших полів

Досить часто виникає потреба використати дані з двох чи більше полів для отримання нового значення. Наприклад, за даними полів **Кількість населення** і **Площа** таблиці **Країни світу** визначити густоту населення для кожної з країн. Для цього створимо простий запит на вибірку, в який з таблиці **Країни світу** включимо поля **Назва**, **Частина світу**, **Площа** і **Кількість населення**. Назвемо запит **Густота населення**. Для створення поля **Густота населення** необхідно відкрити створений запит у поданні **Конструктор**.

У поданні **Конструктор** у рядку **Поле** слід у стовпці, наступному за стовпцями з даними, увести **Густота населення: [Кількість населення] \ [Площа]**, де:

- **Густота населення**: – ім'я нового поля (так званого обчислювального поля);
- **Кількість населення** і **Площа** – імена полів, значення яких використовуються у виразі;
- **** – оператор ділення та округлення до цілого.

Густота населення					
	Назва	Частина світу	Площа	Кількість населен	Густота населення
	Австралія	Австралія і Океанія	7692,024	22510123	2926
	Австрія	Європа	83,871	8356707	99484
	Азербайджан	Європа	86,6	9047000	103988
	Албанія	Європа	28,7	3187009	109896
	Алжир	Африка	2381,7	35200000	14777
	Ангола	Африка	2381,74	13134	5

Рис. 3.66. Таблиця, що містить результат запиту – поле **Густота населення**, дані якого обчислені за даними з інших полів

Результат запиту з полем, що містить дані, обчислені за даними з інших полів, подано на рисунку 3.66.



Запити з параметрами

Уведення користувачем значення параметра запиту – значення, що буде використано в одній або кількох умовах запиту, може бути здійснено і для запитів на вибірку, і для перехресних запитів, і для запитів на внесення змін.

При використанні запиту, створення якого розглянуто вище, може статися, що нам необхідно буде переглядати площі країн то однієї, то іншої частини світу. І тоді доведеться або постійно переходити до подання **Конструктор** і змінювати умови запиту, або створити окремі запити для кожної частини світу. Цього можна уникнути, якщо використати запит з параметром. Для його створення слід відкрити запит у поданні **Конструктор** і виконати таку послідовність дій:

1. Зробити поточним рядок **Критерії** стовпця з іменем поля, для якого буде вводитися значення параметра.
2. Увести у квадратних дужках текст, що буде відображатися в діалоговому вікні, яке відкривається на початку виконання запиту, наприклад *[Назва частини світу]*.
3. Увести за потреби текст іншого повідомлення для введення значення іншого параметра цього самого поля або інших полів.
4. Вибрати кнопку **Запуск**



Перевірте себе

- 1°. Для чого призначені запити?
- 2°. Назвіть основні операції, що виконуються з використанням запитів у базах даних.
- 3°. Опишіть призначення основних видів запитів.
- 4°. Наведіть приклади використання запитів на вибірку.
- 5°. Чим відрізняється запит на вибірку від перехресного запиту?
- 6*. Порівняйте запит з параметрами у базі даних і запит на пошук даних у пошуковій системі Інтернету. Що в них спільного і що відмінного?
- 7°. Для чого у запитах використовують вирази?
- 8°. З чого можуть складатися вирази?
- 9°. Назвіть відомі вам оператори, що використовуються у виразах для створення запитів.
- 10°. Опишіть послідовність створення запиту на вибірку з використанням **Майстра запитів**.
- 11°. Що є результатом виконання запиту на вибірку?
- 12°. Опишіть структуру вікна запиту у поданні **Конструктор**. Поясніть призначення його об'єктів.
- 13°. Поясніть послідовність створення запиту з використанням полів, що містять дані, обчислені за даними з інших полів.
- 14°. Для чого використовуються запити з параметрами?



Виконайте завдання

- 1°. Складіть схему класифікації запитів.
- 2°. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.8\зразок 3.8.2.accdb**). Створіть з використанням **Майстра простих запитів** запит на вибірку даних з таблиці **Географічні та історичні карти**. Запит повинен включати дані з полів **Країна**, **Регіон**, **Вид карти**, **Розмір**, **Ціна**. У поданні **Конструктор** установіть такі умови відбору даних: за полем **Вид карти** – *Політична*; за полем **Ціна** – *менше ніж 45 грн*. Збережіть базу даних у власній папці у файлі з іменем **вправа 3.8.2.accdb**.

- 3*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.8\зразок 3.8.3.accdb**). Створіть з використанням **Майстра простих запитів** запит на вибірку даних з таблиці **Географічні та історичні карти**. Запит повинен включати дані з полів **Країна, Регіон, Вид карти, Вид оформлення, Ціна**. У поданні **Конструктор** установіть такі умови відбору даних: за полем **Вид карти** – *Ілюстрована або Оглядова*; за полем **Ціна** – *від 10 до 35 грн*. Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.8.3.accdb**.
- 4*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.8\зразок 3.8.4.accdb**). Створіть з використанням **Майстра простих запитів** запит на вибірку даних з таблиці **Країни**. Запит повинен включати дані з полів **Назва, Частина світу, Дата утворення, Державний устрій, Державний прапор, Державний гімн**. У поданні **Конструктор** установіть такі умови відбору даних: за полем **Дата утворення** – *після 1990 року*; за полем **Державний устрій** – *Парламентська республіка*. Збережіть базу даних у власній папці у файлі з іменем **вправа 3.8.4.accdb**.
- 5*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.8\зразок 3.8.5.accdb**). Створіть з використанням **Майстра простих запитів** запит на вибірку даних з таблиці **Отримання товару**. Запит повинен включати дані з полів **Тип товару, Назва товару, Ціна, Кількість**. У поданні **Конструктор** доповніть запит новим полем **Сума**, обчисливши в ньому дані на основі значення полів **Ціна** і **Кількість**. Збережіть базу даних у власній папці у файлі з іменем **вправа 3.8.5.accdb**.
- 6*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.8\зразок 3.8.6.accdb**). Створіть з використанням **Майстра простих запитів** запит на вибірку даних з таблиці **Електронні засоби для навчання в школі**. Запит повинен включати дані з полів **Назва, Тип програми, Ціна, Кількість, Гриф МОН**. У поданні **Конструктор** доповніть запит новим полем **Сума**, обчисливши в ньому дані на основі значення полів **Ціна** і **Кількість**. Передбачте виведення у результаті запиту даних тільки про ті засоби, що мають гриф МОН. Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.8.6.accdb**.

3.9. Звіти. Створення звітів в Access 2007



1. Як надрукувати документ в **Word 2007, Excel 2007**?
2. Опишіть призначення і послідовність створення запитів у **Access 2007**.
3. Як у запиті створити поле з даними, що обчислюються за даними з інших полів? У яких випадках такі поля використовуються?
4. Опишіть можливості редагування форм у поданні **Конструктор СУБД Access 2007**.

Поняття про звіти в базі даних

Звіти в БД призначені, як правило, для виведення на друк результатів опрацювання даних. Звіт – це готовий до друку електронний документ. Звіти можуть бути використані для заповнення бланків документів, наприклад додатків до атестатів про повну загальну середню освіту, сертифікатів про проходження зовнішнього незалежного оцінювання тощо. Під час створення звіту можуть бути використані дані з різних таблиць і запитів, засоби узагальнення, сортування і групування даних, можна провести операції над даними полів, використовуючи дані з інших полів і вбудовані функції. Також до звіту можуть бути включені написи, що пояснюють наведені дані, діаграми і графіки, малюнки тощо.

Площа і кількість населення країн світу

Частина світу	Назва країни	Площа	Кількість населення
Австралія і Океанія	Австралія	7692,024	22510123
	Нова Зеландія	268,7	3778
	Соломонові Острови	29	447
	Тонга	0,7	100
Зведення для "Частина світу" = Австралія і Океанія (4 записи з відомостями)			
Сума		7990,4	
Азія	Іран	1650	70330
	Таїланд	514	62806
	Словаччина	49	5399
	Словенія	20,3	1988
Нідерланди	Нідерланди	41,5	15864
Зведення для "Частина світу" = Європа (42 записи з відомостями)			
Сума		23128	
Загальний підсумок		128299	

5 грудня 2011 р. Сторінка 4 з 4

① Нижній колонтитул сторінки ⑤ Верхній колонтитул групи ④ Відомості
 ② Нижній колонтитул звіту ⑥ Верхній колонтитул сторінки
 ③ Нижній колонтитул групи ⑦ Верхній колонтитул звіту

Рис. 3.67. Основні об'єкти звіту **Access 2007**

На першому етапі створення звіту слід розробити його проект як документа, який буде надруковано або передано в електронному вигляді, визначити джерела даних, що будуть включені до звіту.

Розглянемо структуру звіту на прикладі звіту **Площа і кількість населення країн світу** (рис. 3.67), який створений на основі таблиці **Країни світу**. Цей звіт містить:

- **верхній колонтитул звіту** (рис. 3.67, 7), в якому розміщено назву звіту. Він може містити графічні об'єкти (логотип, емблема, ілюстрація тощо), поточні дату та час, елементи керування та ін. Верхній колонтитул звіту розміщується тільки на початку звіту перед колонтитулом сторінки і більше не повторюється;
- **верхній колонтитул сторінки** (рис. 3.67, 6), в якому розміщено заголовки стовпців звіту. Цей об'єкт повторюється у верхній частині кожної наступної сторінки звіту аналогічно до колонтитулів сторінки у **Word 2007**;
- **верхній колонтитул групи** (рис. 3.67, 5), в якому розміщено назву групи об'єктів. Розміщується перед початком даних кожної групи. Наприклад, для звіту, наведеного на рисунку 3.67, на початку списку з даними про країни певної частини світу виведено заголовок – назва цієї частини світу. Може містити елементи керування для узагальнення даних для відповідної групи об'єктів;
- **відомості** (рис. 3.67, 4) – в цій частині розміщено дані про кожний об'єкт відповідно до полів, що були включені до звіту;
- **нижній колонтитул групи** (рис. 3.67, 3), що розміщується після даних про всі об'єкти групи і може містити узагальнені дані про групу об'єктів: кількість об'єктів, суму по певному полю тощо;

- **нижній колонтитул звіту** (рис. 3.67, 2), що розміщується на останній сторінці звіту після всіх даних звіту і використовується для виведення узагальнених даних (кількість, сума, найбільший або найменший за значенням об'єкт групи тощо);
- **нижній колонтитул сторінки** (рис. 3.67, 1), що розміщується внизу кожної сторінки звіту, містить дані про номер поточної сторінки звіту та загальну кількість сторінок звіту, поточні дату і час та інші службові дані.

Створення звітів

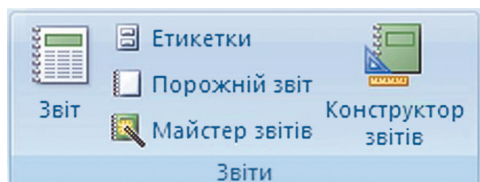


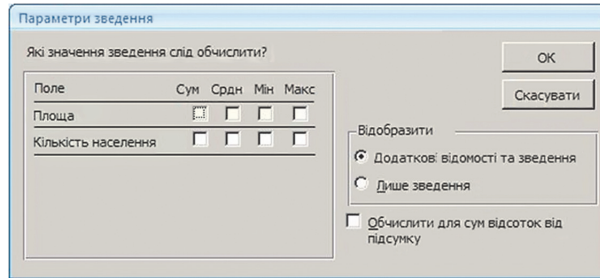
Рис. 3.68. Елементи керування групи **Звіти**

Access 2007 надає користувачу кілька засобів для створення звітів. Доступ до цих засобів забезпечується з використанням елементів керування групи **Звіти** вкладки **Створити** (рис. 3.68). Найпростіший засіб створення звітів – із використанням кнопки **Звіт**

Для створення звіту достатньо відкрити або зробити поточною таблицю чи запит, на основі якої (якого) буде створено звіт, і вибрати кнопку **Звіт** . Буде створено звіт у вигляді таблиці з усіма даними, що містить таблиця (запит). Ця таблиця буде відформатована для друку з тими значеннями параметрів аркуша, які встановлені за замовчуванням.

Більш широкі можливості зі створення звітів надає користувачу **Майстер звітів**. Розглянемо послідовність створення звіту на прикладі, наведеному на рисунку 3.67. Для запуску **Майстра звітів** слід виконати **Створити** \Rightarrow **Звіти** \Rightarrow **Майстер звітів**. У подальшому слід дотримуватися вказівок програми і виконати таку послідовність дій:

1. Вибрати у вікні **Майстер звітів** у списку **Таблиці і запити** потрібний об'єкт, на основі якого буде створено звіт, наприклад таблицю **Країни світу**. Вибрати, перемістивши у список **Вибрані поля**, потрібні для створення звіту поля, наприклад поля **Назва**, **Частина світу**, **Площа**, **Кількість населення**.
2. Вибрати за потреби іншу таблицю або запит і вибрати потрібні поля. Вибрати кнопку **Далі**.
3. Вибрати за потреби поле, за яким буде здійснено групування об'єктів у звіті, наприклад **Частина світу**, та вибрати кнопку **Далі**.
4. Установити за потреби порядок сортування об'єктів у звіті (у межах груп).
5. Установити за потреби вибором кнопки **Параметри зведення** та встановленням позначок прапорців види зведення для відповідних полів. Наприклад, установивши параметр **Сум** для поля **Площа**. *Звертаємо вашу увагу:* за вибору перемикача **Додаткові відомості та зведення** у групі **Відобразити вікна** **Параметри зведення** (рис. 3.69) звіт міститиме дані про всі об'єкти обраної таблиці чи запиту та узагальнені дані за певним полем (полями), наприклад за полем **Площа**, а за вибору перемикача **Лише зведення** – тільки

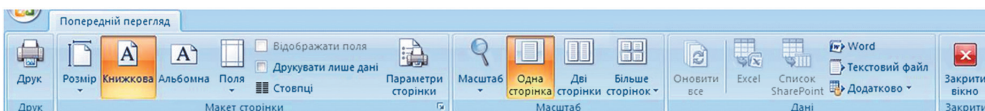
Рис. 3.69. Вікно **Параметри зведення**

- узагальнені дані для певної групи, наприклад тільки суму площ країн частин світу.
6. Установити за потреби позначку прапорця **Обчислити для сум відсоток від підсумку** та вибрати кнопку **ОК**.
 7. Вибрати кнопку **Далі**.
 8. Обрати макет майбутнього звіту вибором відповідного перемикача у групі **Макет**, наприклад **Східчатий**. Установити вибором відповідного перемикача в групі **Орієнтація** вид орієнтації сторінки. Установити за потреби позначку прапорця **Настроїти ширину поля, щоб усі поля поміщалися на сторінці**. Вибрати кнопку **Далі**.
 9. Вибрати у відповідному списку стиль оформлення звіту і вибрати кнопку **Далі**.
 10. Увести назву звіту в поле **Виберіть назву для звіту**, наприклад **Площа і кількість населення країн світу**. Обрати вибором відповідного перемикача наступну дію після створення звіту, наприклад **Попередній перегляд звіту**.
 11. Вибрати кнопку **Готово**.

Після завершення створення звіту за описаним алгоритмом на екран буде виведено звіт у режимі попереднього перегляду перед друком. А на **Стрічці** з'являться елементи керування тимчасової вкладки **Попередній перегляд** (рис. 3.70). Ці елементи керування надають користувачу можливість:

- змінити макет звіту, тобто змінити розміри сторінки, її орієнтацію, розміри полів сторінки, кількість стовпців тощо;
- змінити масштаб перегляду звіту;
- експортувати звіт до іншого документа – документа **Word**, текстового документа тощо.

Якщо у звіт не потрібно вносити ніяких додаткових змін, крім тих, які можна зробити з використанням елементів керування вкладки **Попередній перегляд**, то можна одразу надрукувати документ. Для цього слід вибрати кнопку **Друк** . Однак, як правило, звіт необхідно відредагувати і для цього слід спочатку закрити вікно **Попереднього перегляду** вибором відповідної кнопки .

Рис. 3.70. Елементи керування тимчасової вкладки **Попередній перегляд**

Редагування та форматування звітів

Для редагування звітів в **Access 2007** потрібно використати один із двох видів подання – **Режим розмічування** або **Конструктор**.

Для відкриття звіту в **Режимі розмічування** виконаємо **Основне** ⇒ **Подання** ⇒ **Вигляд** ⇒ **Режим розмічування**.

Для зміни назви стовпця необхідно двічі клацнути на відповідній клітинці та відредагувати текст, наприклад увести **Назва країни**. Таким самим шляхом можна додати до назв стовпців **Площа** і **Кількість населення** одиниці вимірювання, наприклад **Площа, кв. км** і **Кількість населення, осіб**. Але виведення найменувань у такий спосіб не зовсім зручне під час перегляду довгого звіту. Бажано розмістити позначення одиниць вимірювання біля числа. Однак під час роботи в **Режимі розмічування** це зробити неможливо. Також складно виконати редагування повідомлення у нижньому колонтитулі групи. У цих випадках редагування краще здійснити у поданні **Конструктор**.

Для відкриття звіту в поданні **Конструктор** слід виконати **Основне** ⇒ **Подання** ⇒ **Вигляд** ⇒ **Конструктор**. Вигляд вікна звіту у поданні **Конструктор** наведено на рисунку 3.71.

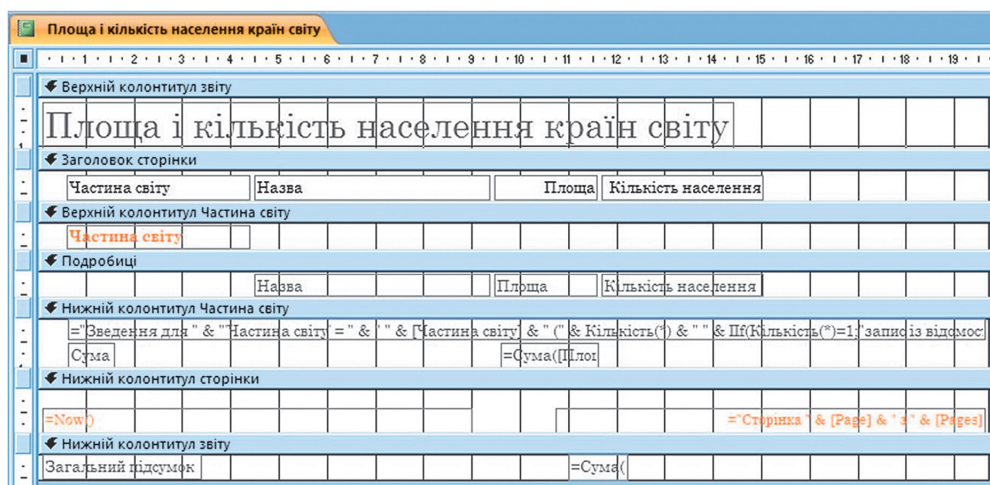


Рис. 3.71. Звіт **Площа і кількість населення країн світу** в поданні **Конструктор**

Під час редагування колонтитула звіту можна доповнити його додатковими написами або малюнками. Наприклад, заголовок звіту *Площа і кількість населення країн світу* можна доповнити додатковим написом з текстом: *Розроблено на основі даних комп'ютерної енциклопедії <http://uk.wikipedia.org>*. Для вставлення цього напису з текстом як підзаголовка звіту необхідно виконати такі дії:

1. Збільшити розміри поля верхнього колонтитула звіту шляхом перетягування його нижньої межі.
2. Виконати **Знаряддя конструктора звітів** ⇒ **Конструктор** ⇒ **Елементи керування** ⇒ **Напис**.
3. Вибрати на полі верхнього колонтитула звіту місце розміщення напису.

4. Увести текст.

5. Переглянути отриманий результат у **Поданні звіту**.

Додати графічний об'єкт до верхнього колонтитула звіту можна кількома способами, використовуючи різні елементи керування однойменної вкладки: **Логотип**, **Зображення**, **Вільна рамка об'єкта**, **Приєднана рамка об'єкта**. Так, для додавання графічного об'єкта з використанням елемента керування **Логотип** слід:


1. Вибрати кнопку **Логотип** .

2. Вибрати у вікні **Вставлення зображення** файл малюнка, що буде вставлено до верхнього колонтитула звіту, наприклад **GLOBE.WMF**, та вибрати кнопку **ОК**.

3. Розмістити вставлене зображення у потрібному місці, використовуючи перетягування самого малюнка і написів заголовка та підзаголовка або їх меж.

4. Переглянути отриманий результат у режимі **Подання звіту**.

Вставлення малюнка з використанням інших інструментів майже не відрізняється від описаного, за виключенням того, що при вставленні потрібно вказати місце розміщення малюнка, а при вставленні як об'єкта – ще й вказати вид об'єкта. Можна також відокремити заголовок – тобто верхній колонтитул звіту від інших об'єктів звіту горизонтальною лінією.

Для цього необхідно вибрати кнопку **Лінія**  і намалювати лінію низу верхнього колонтитула звіту. Колір, тип і товщину ліній можна встановити відповідними елементами керування групи **Елементи керування**. Результат редагування верхнього колонтитула подано на рисунку 3.72.

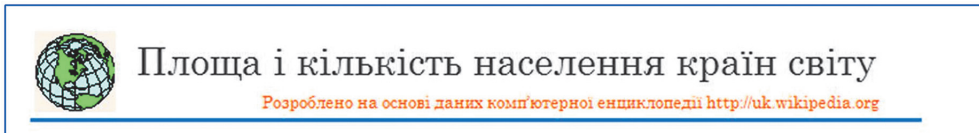


Рис. 3.72. Заголовок звіту (верхній колонтитул звіту) після редагування

Вставлення текстових і графічних об'єктів у нижній колонтитул звіту відбувається за аналогічними алгоритмами. При вставленні в колонтитул номери сторінки і назви групи ці об'єкти будуть повторюватися відповідно на кожній сторінці або на початку чи в кінці даних кожної з груп.

Частина звіту **Відомості** (рис. 3.67, 4) в поданні **Конструктор** називається **Подробиці** (рис. 3.71). Під час вставлення об'єктів у цю частину звіту вони будуть повторені для кожного рядка даних. Наприклад, якщо вставити біля полів **Площа** і **Кількість населення** написи з назвами одиниць вимірювання цих величин, (*кв. км* і *осіб* відповідно) (рис. 3.73, а), то ці написи повторяться в кожному рядку (рис. 3.73, б).

Як ви вже знаєте, для введення формул, що підраховують певні значення, використовують елементи керування, які називають **текстові поля**. Розглянемо використання формули для знаходження кількості об'єктів у кожній групі країн. Для цього слід відкрити звіт у поданні **Конструктор**, видалити з нижнього колонтитула групи вказаний об'єкт і виконати таку послідовність дій:

◆ Подобиці									
				Назва країни			Площа		кв. км
								Кількість населення	осіб

а)

Частина світу	Назва країни	Площа	Кількість населення
Австралія і Океанія	Австралія	7692,024 кв. км	22510123 осіб
	Нова Зеландія	268680 кв. км	4377000 осіб
	Соломонові Острови	28896 кв. км	478000 осіб
	Тонга	747 кв. км	120898 осіб

б)

Рис. 3.73. Фрагмент звіту з написами біля числових даних

1. Виконати *Знаряддя конструктора звітів* ⇒ *Конструктор* ⇒ *Елементи керування* ⇒ *Текстове поле*.
2. Вибрати на полі нижнього колонтитула місце для текстового поля.
3. Установити шляхом перетягування меж довжину і ширину лівої та правої частин текстового поля.
4. Увести в ліву частину текстового поля текст, наприклад *Кількість країн*.
5. Увести в праву частину поля вираз $\text{=Кількість}([Назва])$.
6. Відформатувати за потреби тексти відповідно до стилю оформлення звіту.
7. Переглянути отриманий результат у режимі *Подання звіту*.

Відповідно до введеного виразу буде здійснено підрахунок кількості об'єктів у стовпці **Назва** для кожної групи країн, тобто для кожної частини світу. У цьому самому колонтитулі можна замінити підпис даних узагальнення з *Сума*, запропонований програмою автоматично, на *Загальна площа країн*. Після редагування нижній колонтитул групи матиме вигляд, поданий на рисунку 3.74.

	Камбоджа	181040 кв. км	14805000 осіб
	Узбекистан	447400 кв. км	28095900 осіб
Кількість країн	35		
Загальна площа країн		27706588	

Рис. 3.74. Вигляд нижнього колонтитула групи після виконання редагування

Форматування звітів, як правило, здійснюється в **Режимі розмічування**. Для цього використовуються елементи керування вкладок **Формат**, **Упорядкувати** та **Параметри сторінки** тимчасової вкладки **Знаряддя для макетів звітів**. Операції з використанням цих інструментів аналогічні описаним раніше.

Зберігаються звіти автоматично під час створення. Після редагування звітів їх можна зберегти стандартним способом збереження об'єктів СУБД Access 2007. Зміни, що вносяться у поля таблиць або у запити, на основі яких створені звіти, автоматично враховуються під час відкриття звітів.



Перевірте себе

- 1°. Поясніть призначення звітів бази даних.
- 2°. Наведіть приклади використання звітів бази даних.
- 3°. Опишіть структуру звіту і призначення його окремих складових.
- 4°. Опишіть створення нового звіту з використанням кнопки **Звіт**.
- 5°. Які можливості зі створення звітів надає користувачу **Майстер звітів**?
- 6°. Поясніть етапи створення звіту з використанням **Майстра звіту**.
- 7°. Для чого використовується під час створення звітів **Параметри зведення**?
- 8°. Опишіть можливості, які надає користувачу з підготовки до друку режим **Попередній перегляд**.
- 9°. Коли виникає необхідність у редагуванні та форматуванні звітів?
- 10°. Які об'єкти можна відредагувати та відформатувати у верхньому колонтитулі звіту? Як це виконати?
- 11°. Опишіть особливості редагування та форматування верхнього колонтитула сторінки і верхнього колонтитула групи.
- 12°. Опишіть особливості редагування та форматування нижнього колонтитула сторінки і нижнього колонтитула групи. Які об'єкти, як правило, розміщуються в цих частинах звіту?
- 13°. Як включити до звіту графічні об'єкти, де вони можуть розміщуватися?
- 14°. Які об'єкти, як правило, розміщуються в області **Подробиці подання Конструктор**? Чим це зумовлено?
- 15°. З якою метою використовуються вирази у звітах бази даних? У яких областях звіту (подання **Конструктор**) вони розміщуються?



Виконайте завдання

- 1°. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.9\зразок 3.9.1.accdb**). Створіть, використовуючи кнопку **Звіт**, звіт за даними таблиці **Географічні та історичні карти**. Відредагуйте і відформатуйте його відповідно до рисунка 3.75. Логотип вставте з файлу **Тема 3\Завдання 3.9\optdisk.wmf**. Збережіть базу даних у власній папці у файлі з іменем **права 3.9.1.accdb**.

Географічні та історичні карти				
Вид карти	Розмір	Масштаб	Вид оформлення	Ціна
Автономна Республіка Крим				
Адміністративна	42 x 30 см	M1:850 000	ламінація	15,00 грн.
Сума				15,00 грн.
Європа				
Політична	160 x110 см	M1:3 750 000	ламінація	90,00 грн.
Політична	160 x110 см	M1:3 750 000	папір	24,00 грн.
Політична	110x77 см	M1:5 400 000	ламінація з планками	72,00 грн.
Політична	110x77 см	M1:5 400 000	ламінація	60,00 грн.
Місцевість				
Фізична	65 x 45 см	M1:11 000 000	папір	4,50 грн.
Політична	160 x110 см	M1:3 750 000	ламінація з планками	120,00 грн.
Сума				505,50 грн.
Молдова				
Політична	100 x 68 см	M1:35 000	папір	1,00 грн.

Рис. 3.75



- 2*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.9\зразок 3.9.2.accdb**). Створіть, використовуючи **Майстер звітів**, звіт **Площа і кількість населення країн світу** відповідно до опису, поданого в тексті пункту. Виконайте відповідне редагування та форматування звіту в поданнях **Конструктор** і **Режим розмічування**. Логотип вставте з файлу **Тема 3\Завдання 3.9\globe1.wmf**. Збережіть базу даних у папці **Мої документи** у файлі з іменем **вправа 3.9.2.accdb**.
- 3*. Відкрийте файл бази даних (наприклад, **Тема 3\Завдання 3.9\зразок 3.9.3.accdb**). Створіть з використанням **Майстра звітів** звіт за даними таблиці **Отримання товару**. Відредагуйте і відформатуйте його відповідно до рисунка 3.76. Логотип вставте з файлу **Тема 3\Завдання 3.9\worldmap.wmf**. Збережіть базу даних у власній папці у файлі з іменем **вправа 3.9.3.accdb**.

Отримання товару <i>Магазин "Bim"</i>						12 грудня 2010 р. 23:27:28
Номер накладної	Дата отримання	Назва товару	Тип товару	Кількість	Ціна	
123	11.11.2011	Комп'ютер	Настільний	4	4 600,00 грн.	
123	11.11.2011	Монітор	19"	6	1 567,00 грн.	
123	11.11.2011	Сканер	Планшетний	3	437,00 грн.	
124	21.11.2011	Комп'ютер	Ноутбук	10	3 656,00 грн.	
124	21.11.2011	Фотоапарат	Цифровий	15	1 273,00 грн.	
125	23.11.2021	Комп'ютер	Настільний	12	4 398,00 грн.	
125	23.11.2011	Принтер	Лазерний	13	954,00 грн.	
125	23.11.2011	Принтер	Струменевий	5	673,00 грн.	
126	27.11.2011	Комп'ютер	Нетбук	14	3 100,00 грн.	
126	27.11.2011	Монітор	22"	7	1 895,00 грн.	
123	11.11.2011	Комп'ютер	Ноутбук	5	2 910,00 грн.	

Сторінка 1 з 1

Рис. 3.76



Практична робота № 14. Створення запитів і звітів

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте файл бази даних (наприклад, **Тема 3\Практична 14\зразок 14.accdb**).
2. На основі даних таблиці **Калорійність страв** створіть запит, включивши в нього поля: **Страви**, **Калорійність**, **Включення до меню**.
3. У поданні **Конструктор** установіть для запиту умову відбору за полем **Включено до меню** – не включати записи, що мають в цьому полі значення **Не включено**. Вираз повинен мати такий вигляд <> «**Не включено**».
4. Створіть на основі створеного вами запиту звіт згідно зі зразком, поданим на рисунку 3.77.
5. У колонтитул звіту включіть виведення поточної дати у повному форматі.
6. Групування здійсніть за полем **Включення до меню**.
7. Суму значень знайдіть для кожної групи по полю **Калорійність**.
8. Відредагуйте і відформатуйте звіт.
9. Закрийте створені запит і звіт.
10. Збережіть базу даних у своїй папці у файлі з іменем **практична робота 14. accdb**.

Меню їдальні на 13 грудня 2011 р.		
Страви	Калорійність	Маса
Сніданок		
Кава з молоком	150 ккал	200 г
Млинці із сиром і сметаною	650 ккал	209 г
Вінегрет	60 ккал	100 г
Загальна калорійність страв	860 ккал	
Обід		
Соки ягідні, фруктові, томатні	160 ккал	200 г
Картопля варена	250 ккал	209 г
Риба тушкована з овочами	220 ккал	200 г
Борщ, овочевий суп	46 ккал	100 г
Салат зелений с огірками, помідорами	100 ккал	150 г
Загальна калорійність страв	776 ккал	
Вечеря		
Макарони відварені	330 ккал	209 г
Курка відварена	130 ккал	100 г
Ікра баклажанна	175 ккал	100 г
Чай з цукром	50 ккал	200 г
Загальна калорійність страв	685 ккал	

13 грудня 2011 р.

Сторінка 1 з 1

Рис. 3.77

Розділ 4

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПЕРСОНАЛЬНОЇ ТА КОЛЕКТИВНОЇ КОМУНІКАЦІЇ

У цьому розділі ви дізнаєтеся про:

- мультимедійні дані та програмне забезпечення для їх обробки;
- створення аудіофрагментів і відеокліпів;
- публікації, комп'ютерні публікації, їх класифікацію та призначення;
- засоби створення комп'ютерних публікацій;
- особливості роботи з об'єктами під час створення комп'ютерних публікацій;
- різновиди веб-сторінок і веб-сайтів, їх структуру, етапи створення;
- засоби автоматизованої розробки веб-сайтів і створення сайтів з їх використанням;
- основи веб-дизайну;
- технології та сервіси **Веб 2.0**, їх призначення та види;
- створення блогів, їх оформлення та публікацію повідомлень на блогах;
- призначення та особливості функціонування веб-спільнот;
- інтегроване використання засобів опрацювання електронних документів;
- спільну роботу з документами.

4.1. Поняття про мультимедійні дані та їхнє опрацювання



1. Які пристрої для відтворення мультимедійних даних ви знаєте?
2. Що таке тип (формат) файлу? Які формати файлів ви знаєте?
3. Що таке стиснення даних? У яких форматах файлів використовується стиснення?
4. Які типи оптичних дисків ви знаєте? Чим вони відрізняються один від одного?
5. Що таке конвертація форматів файлів?

Поняття про мультимедійні дані

Як ви вже знаєте з курсу 9-го класу, для відтворення мультимедійних даних використовуються спеціальні пристрої комп'ютера. До цих пристроїв належать звукові колонки, мікрофони, головні телефони, мультимедійні проектори, мультимедійні дошки тощо. Однак нам слід уточнити, які дані називаються **мультимедійними**.



Мультимедійними даними називають дані, для подання яких використовують різні способи та які людина сприймає одночасно кількома органами чуття.

Прикладом об'єкта, що містить мультимедійні дані, є комп'ютерна презентація. Презентація може містити текст, графічні зображення,

звукові та відеооб'єкти. Тому дуже часто такі презентації називають мультимедійними. Мультимедійні дані ми отримуємо, коли переглядаємо відеофільми, спектаклі, відвідуємо концерти.

Програмне забезпечення для опрацювання мультимедійних даних

Комп'ютерних програм, які опрацьовують мультимедійні дані, досить багато. З деякими з них ви вже ознайомилися в 9-х та 10-х класах. Наприклад, текстовий процесор **Word 2007** опрацьовує текстові та графічні дані, документи, створені з його використанням, можуть містити аудіо- та віододані. Разом з тим коли говорять про програми опрацювання мультимедійних даних, то частіше розуміють програми, що опрацьовують звукові та віододані. Класифікацію програм для опрацювання звукових і віододаних за їхнім призначенням подано на рисунку 4.1.

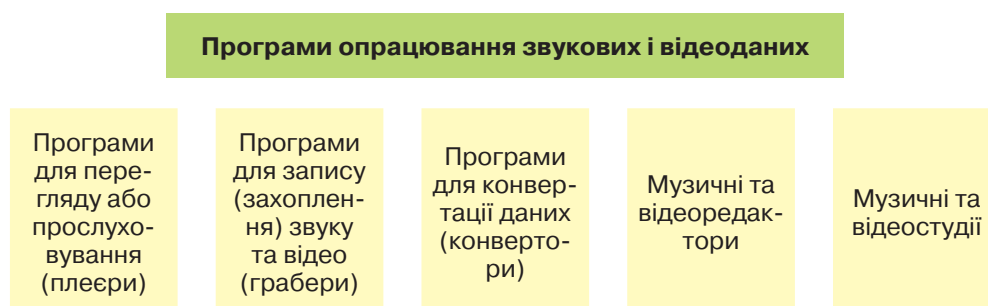


Рис. 4.1. Схема класифікації програм для опрацювання звукових і віододаних

Програми для перегляду або прослуховування мультимедійних даних призначені для відтворення відео та звуку. Ці програми ще називають **мультимедійними програвачами** або **медіаплеєрами**. Існує багато різноманітних мультимедійних програвачів. Більшість з них може відтворювати і відео, і звукові дані, окремі спеціалізуються тільки на одному з видів даних, і тоді їх називають **аудіо-** або **віодоплеєрами**. Розповсюдженими є мультимедійні плеєри **Winamp** (англ. *Windows Advanced Media Player* – розширений програвач мультимедіа для **Windows**), **Media Player Classic**, **QuickTime Player**, **Flv Player** та ін.

Часто до комп'ютера під'єднуються додаткові пристрої для введення звуку і відео. До таких пристроїв належать: мікрофон, відеокамера, веб-камера, телевізійний тюнер, DVD-програвач, віеомагнітофон тощо. Як правило, ці пристрої забезпечують уведення звуку та відео, але не мають функцій з їх збереження у вигляді звукових і віодофайлів певних форматів.

Для збереження звукових і віододаних, що вводяться з наведених пристроїв, призначені спеціальні **програми запису (захоплення) звуку та відео**, які ще називають **граберами** (англ. *grabber* – той, хто захоплює, хапуга). Ці програми під час відтворення, наприклад, відеофільму з DVD-програвача одночасно здійснюють перекодування даних і їх запис у віодофайл обраного користувачем формату. Прикладами програм для захоплення звуку є стандартні програми ОС **Windows Звукозапис**, **Exact Audio Copy**, **Audio Grabber 3.0**, **Streamripper**, **AML Easy Audio Recorder**

та ін. Програмами для захоплення відео є **Extra Screen Capture Pro**, **Wondershare Streaming Video Recorder**, **Wondershare P2P TV Recorder** та ін.

У зв'язку з великою кількістю різноманітних форматів аудіо- і відео-файлів користувачам часто необхідно перекодувати ці файли у формати, які можуть бути відтворені на тих чи інших технічних пристроях або тими чи іншими програмами. Для перекодування файлів з одного формату в інший призначені *програми конвертації файлів*, такі як: **Easy Video Converter**, **MediaCoder**, **Xilisoft Video Converter**, **Flash to Video Encoder**, **Movavi VideoSuite**, **Direct MIDI to MP3 Converter**, **AVS Audio Converter** та ін.

Програми *музичних і відеоредакторів* призначені для створення та редагування музичних творів, аудіокниг, звукових і відеоефектів, звукового супроводу, відеофільмів тощо. Вони забезпечують виконання таких функцій: вставлення, видалення, копіювання, зміна тривалості аудіо- чи відеофрагментів, об'єднання кількох фрагментів, вставлення текстових і графічних об'єктів, синхронізація звуку та відео тощо. Музичними редакторами є такі програми: **Anvil Studio**, **ALO Audio Editor**, **AKRAM Audio Editor** та ін. Прикладами відеоредакторів є стандартні програми **Windows Movie Maker**, **Xilisoft Video Editor**, **VideoMach** та ін.

Музичні та відеостудії мають приблизно таке саме призначення, що й редактори, однак мають розширений набір функцій і використовуються в професійній діяльності на студіях звукозапису, кіно- й анімаційних студіях для створення звукозаписів, аудіо- і відеокomпозицій, анімації, кіно- і відеофільмів, рекламних роликів, відеокліпів тощо. Прикладами програм-музичних студій є: **Steinberg Cubase**, **Sound Forge**, **Cakewalk Sonar**, **Adobe Audition** та ін. Програмами-відеостудіями є **Pinnacle Systems STUDIO**, **Apple Final Cut**, **Adobe Premiere Pro**, **Sony Vegas Movie Studio** та ін.

Формати аудіо- та відеофайлів

У 9-му класі ви розглядали поняття типу (формату) файлу. Тип (формат) файлу визначає структуру даних у файлі. Ви вже знаєте різні формати текстових і графічних файлів. Мультимедійні дані можуть бути збережені у файлах різних форматів.














Приклади форматів мультимедійних файлів подані в таблиці 4.1.



У 9-му класі ви ознайомилися з оптичними дисками, які використовуються в комп'ютерах для збереження даних. Однак спочатку оптичні диски (компакт-диски) були розроблені корпораціями **Philips** і **Sony** для зберігання аудіозаписів. У 1979 р. був створений спеціальний формат запису цих даних, відомий як «червона книга» (англ. *Red Book* – червона книга), а самі диски, записані в цьому форматі, стали називатися **Audio CD** (англ. *Audio Compact Disk* – звуковий компакт-диск). Звукові дані записуються на дисках у вигляді окремих ділянок на спіральній доріжці – треків (англ. *track* – слід, дорога, звукова доріжка). Один трек містить один звуковий фрагмент, наприклад одну пісню.

Для відтворення звукових даних використовувалися спеціальні CD-програвачі. У подальшому були розроблені формати і пристрої для запису і відтворення відеодисків – **Video CD** (англ. *Video Compact*

Disk – відеокомпакт-диск), а з появою оптичних дисків **DVD** були розроблені спеціальні формати цих дисків для збереження аудіо- і відеоданих.





Таблиця 4.1. Типи (формати) мультимедійних файлів

Тип (формат) файлу, його опис	Розширення імені	Значок Windows	
		XP	Vista
Файли, у яких не використовується стиснення або використовується стиснення без втрати даних			
WAV або WAVE (англ. <i>waveform audio format</i> – аудіо-формат типу <i>хвиля</i>) – формат звукових файлів, що використовується як основний для нестиснутих аудіо-даних в операційних системах сімейства Windows	wav		
MIDI (англ. <i>Musical Instrument Digital Interface</i> – цифровий інтерфейс музичних інструментів) – формат файлів, які містять команди для відтворення звуку спеціальним пристроєм або програмою – синтезатором (сенквенсером) (лат. <i>sequentia</i> – послідовність)	mid, midi		
Файли, у яких використовується стиснення із частковою втратою даних			
MP3, точніше MPEG-1/2/2.5 Layer 3 (англ. <i>Motion Picture Experts Group</i> – експертна група з питань рухомих зображень, <i>Layer 3</i> – третій рівень), – один із найрозповсюдженіших форматів аудіофайлів. Використовується для передавання звукових даних у глобальних мережах, у файлообмінних системах	mp3		
AVI (англ. <i>Audio Video Interleave</i> – звук і зображення, розміщені в шарах (каналах)) – формат файлів, що частіше використовується для збереження відеоданих. В одному файлі цього формату можуть бути використані кілька різних алгоритмів стиснення	avi		
WMA/WMV (англ. <i>Windows Media Audio/Video</i>) – формат файлів для зберігання і відтворення аудіо/відеоданих, який розроблений корпорацією Microsoft і стандартно використовується в ОС Windows	wma, wmv		
3GP (англ. <i>3rd Generation Phone</i> – третє покоління телефонів) – формат файлів для передачі мережею мобільних телефонів третього покоління. Високий ступінь стиснення передбачає малі розміри файлів і невисоку якість відтворення	3gp		
QuickTime (англ. <i>Quick Time</i> – швидкий час) – формат файлів, що базується на технології збереження та відтворення відео- та звукових даних, яка була розроблена корпорацією Apple	mov		

Тип (формат) файлу, його опис	Розширення імені	Значок Windows	
		XP	Vista
Flash video – формат файлів для зберігання та відтворення файлів, які створені програмою Adobe Flash або іншою, яка опрацьовує файли цього типу. Відзначається високим ступенем стиснення даних	flv		

Незважаючи на те, що диски зазначених форматів призначені для відтворення спеціальними програвачами, більшість сучасних комп'ютерних пристроїв для роботи з оптичними дисками можуть опрацьовувати аудіо- та відеодиски. Приклади форматів цих дисків подані в таблиці 4.2.

Таблиця 4.2. Типи (формати) мультимедійних дисків

Тип (формат) диска, його опис	Позначення
Audio CD – формат, що використовується для запису музичних творів на компакт-диски. Мінімальна тривалість звучання одного треку – 2 с, максимальна кількість треків – 99, максимальний час звучання всіх творів (треків) – 74 хв	
Video CD – формат, що використовується для запису відеофільмів на компакт-диски. Максимальний час програвання відеофільму – 74 хв низької якості	
DVD-Audio – формат, що використовується для запису музичних творів на DVD-диски. Звук, записаний у цьому форматі, має кращу якість за рахунок використання кількох каналів запису даних	
DVD-Video – формат запису відеофільмів на DVD-диски. Максимальний час програвання відеофільму – 120 хв високої якості	

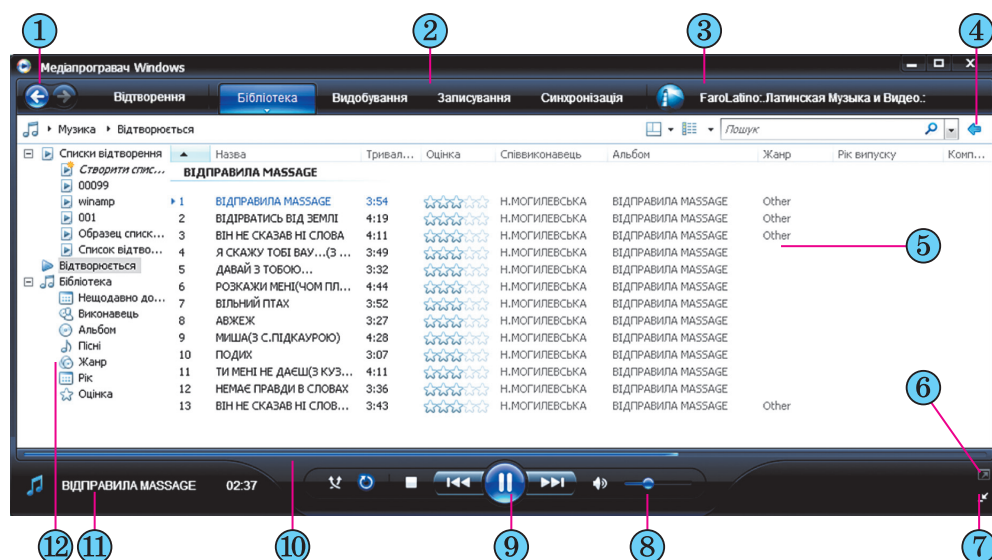
У зв'язку з тим, що більшість аудіо- та відеоданих стискаються з використанням різноманітних алгоритмів, для їхнього відтворення потрібно застосовувати спеціальні програми – кодувальники/декодувальники, або **кодеки**. Кодеки входять до складу програм-програвачів аудіо- чи відеофайлів, а також можуть випускатись як додатковий набір програм, наприклад **K-Lite Codec**.

Медіапрогравач Windows

Одним з мультимедійних програвачів є програма **Медіапрогравач Windows**, що входить до складу стандартних програм операційної системи **Windows**. На рисунку 4.2 зображено вікно цієї програми.

Медіапрогравач Windows надає користувачу такі можливості в опрацюванні мультимедійних (медіа-) файлів:

- здійснювати пошук і включення до бібліотеки (медіатеки) медіафайлів;
- прослуховувати та переглядати медіафайли різних форматів;



- | | |
|--|--|
| 1 Кнопки переходу Назад і Вперед | 7 Кнопка переходу до компактного режиму |
| 2 Рядок меню | 8 Регулятор гучності |
| 3 Кнопка відкриття списку команд роботи зі списками он-лайн сховищ | 9 Кнопки керування відтворенням |
| 4 Кнопка Показати область списку | 10 Індикатор ходу відтворення |
| 5 Область відомостей | 11 Область сповіщення про файл, що відтворюється |
| 6 Кнопка переходу до повноекранного режиму | 12 Область навігації |

Рис. 4.2. Вікно **Медіапрогравач Windows**

- здійснювати конвертацію мультимедійних даних з аудіо- та відео-оптичних дисків у різні формати мультимедіафайлів;
- створювати аудіо- та відеооптичні диски шляхом перекодування мультимедійних файлів і запису їх на диски відповідно до зазначених у таблиці 4.2 форматів;
- завантажувати та відтворювати мультимедійні дані із сайтів в Інтернеті (*он-лайн сховищ*), які надають послуги зі зберігання медіафайлів і доступу до них зареєстрованих користувачів;
- синхронізувати медіафайли на комп'ютері та на обраному портативному пристрої, наприклад флеш-плеєрі та ін.

Програма здійснює перегляд папок і внесення знайдених мультимедійних файлів до медіатеки. Для кожного з аудіофайлів у медіатеку програми заносяться дані про виконавця, назва альбому, назва пісні, жанр і рік створення. Для відеофайлів – дані про акторів і жанр відео.




Для зміни папок, у яких здійснюється пошук мультимедійних файлів, потрібно:

1. Вибрати кнопку **Бібліотека**.
2. Вибрати в списку кнопки **Бібліотека** команду **Додати до бібліотеки**.
3. Вибрати перемикач **Мої папки та доступні мені чужі папки**.
4. Вибрати кнопку **Додаткові параметри**.
5. Вибрати кнопку **Додати**.
6. Указати папку, в якій необхідно здійснити пошук мультимедійних файлів для додавання їх до бібліотеки програвача.

У подальшому програма буде автоматично відстежувати наявність файлів у вказаних папках і додавати нові до бібліотеки.




Користувач може створити свої **Списки відтворення** – добірки улюблених творів або творів, які необхідно відтворити під час проведення певного заходу, наприклад розповіді про певного виконавця, огляд його творів з різних альбомів. Для створення списку відтворення слід:

1. Вибрати над **Областю навігації** кнопку **Вибір категорії** .
2. У списку кнопки вибрати тип об'єктів, які будуть відображатися в **Області відомостей**: *Музика, Зображення, Відео, Записані телепрограми або Інші медіафайли*.
3. Вибрати кнопку **Показати область списку**  в правій верхній частині вікна.
4. Перетягнути з **Області відомостей** в **Область списку** необхідний мультимедійний об'єкт.
5. Виконати команду 4 потрібну кількість разів.
6. Вибрати кнопку **Зберегти список відтворення**  **Зберегти список відтворення**.
7. Увести у відповідне поле ім'я списку відтворення.
8. Вибрати кнопку **Зберегти список відтворення**.

Ім'я створеного списку відтворення з'являється у папці **Списки відтворення** **Області навігації**. Користувач може створити кілька списків відтворення.

Для відтворення аудіо- або відеофайлів потрібно:

1. Вибрати в **Області навігації** один з об'єктів:
 - у папці **Списки відтворення** один зі списків відтворення;
 - у папці **Бібліотека** одну з груп сортування занесених до бібліотеки мультимедійних об'єктів.
2. Вибрати в **Області відомостей** потрібний альбом або файл.
3. Вибрати кнопку **Відтворити** .

За вибору відеофайлу програма автоматично перейде до відображення вкладки **Відтворення** та в **Області перегляду** почнеться відтворення відео. Для керування процесом відтворення звукових чи відеоданих використовують кнопки керування (рис. 4.2, 9). Призначення цих кнопок описано в таблиці 4.3.









За потреби користувач може зменшити вікно програвача та перейти до так званого **компактного режиму**. Для цього слід вибрати кнопку **Перейти до компактного режиму**  (рис. 4.2, 7). Вигляд вікна **Медіапрогравач Windows** у компактному режимі подано на рисунку 4.3.



Рис. 4.3. Вікно **Медіапрогравач Windows** у компактному режимі

Таблиця 4.3. Кнопки керування відтворенням у програмі Медіапрогравач Windows

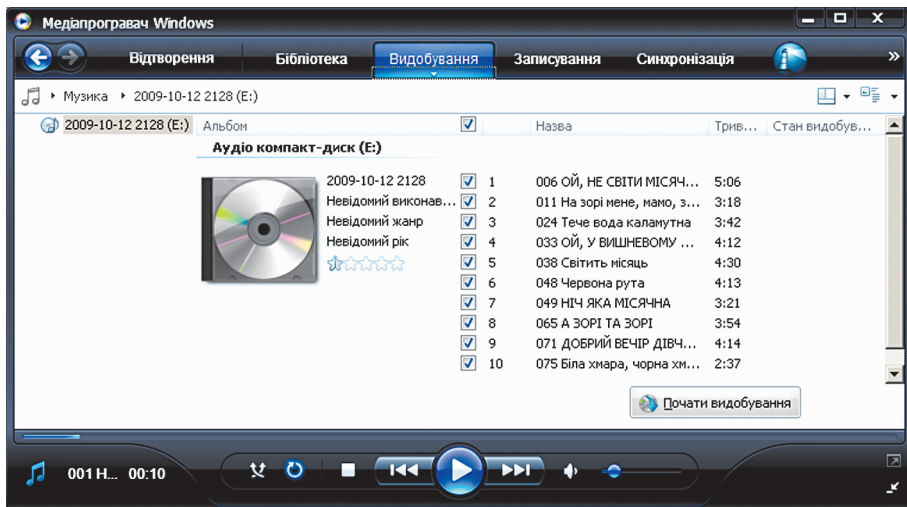
Кнопка керування		Призначення
Зображення	Назва	
	Увімкнути/ вимкнути перемішування	Для увімкнення/вимкнення режиму перемішування відтворюваних творів, тобто для відтворення їх у випадковому порядку
	Увімкнути/ вимкнути повторювання	Для увімкнення/вимкнення режиму безперервного відтворення файлів обраного списку
	Зупинити	Для зупинки процесу відтворення
	Попередній	Для переходу до відтворення попереднього твору
	Наступний	Для переходу до відтворення наступного твору
	Відтворити	Для початку або продовження відтворення (під час відтворення зображення кнопки змінюється на зображення кнопки Пауза)
	Пауза	Для тимчасової зупинки відтворення (під час зупинки зображення кнопки змінюється на зображення кнопки Відтворити)


Засоби перетворення аудіо- та відеофайлів з одного формату в інший

У зв'язку з великою кількістю різноманітних форматів аудіо- та відеофайлів у користувачів часто виникає необхідність перекодувати ці файли з одного формату в інший. Як ви знаєте з курсу інформатики 9-го класу, процес перекодування файлу одного формату в інший називається **конвертацією файлів**.

Конвертація може здійснюватися в процесі відкриття мультимедійного файлу або під час його збереження на диску в іншому форматі. Одним із способів конвертації в програмі **Медіапрогравач Windows** є процес видобування даних з оптичних дисків форматів **Audio CD**, **Video CD**, **DVD-Audio** та **DVD-Video** і збереження їх, наприклад, на жорсткому магнітному диску у файлах формату **WMA**, або **WMV**, або інших за вибором користувача. Для виконання цієї операції потрібно:

1. Вставити аудіо- чи відеодиск у пристрій для роботи з оптичними дисками.
2. Вибрати кнопку **Видобування** та відкрити відповідну вкладку (рис. 4.4).
3. Відкрити за потреби список команд кнопки **Видобування**, вибрати команду **Додаткові можливості** та на вкладці **Конвертування музики** встановити:

Рис. 4.4. Вкладка **Видобування** вікна **Медіапрогравач Windows**

- тип файлу, у який буде здійснено конвертування;
 - папку, у якій будуть зберігатися файли після конвертації (за замовчуванням **C:\Documents and Settings\user\Мої документи\Моя музика**, де **user** – назва поточного облікового запису);
 - перелік властивостей твору, які будуть відображатися в **Області відомостей**, наприклад ім'я виконавця та назва пісні.
4. Установити позначки прапорців біля тих творів, які передбачається конвертувати.
 5. Вибрати кнопку **Почати видобування** .
 6. Вибрати перемикач для встановлення захисту музичних творів від копіювання або відмовитися від захисту.
 7. Установити позначку прапорця про усвідомлення відповідальності за можливі порушення авторських прав.
 8. Дочекатися завершення процесу конвертації.
- Хід процесу конвертації відображається у стовпці **Стан видобування** (рис. 4.5). Користувач може в будь-який час припинити процес видобування, вибравши кнопку **Зупинити видобування** .

Зворотний процес конвертації медіафайлів для запису на диск і створення відповідного формату аудіо- чи відеодиска в програмі **Медіапрогравач Windows** називається **записуванням**. Створення подібних дисків необхідне, якщо передбачається прослуховування або перегляд медіаданих з використанням пристроїв, що не відтворюють дані, записані в комп'ютерних форматах, наприклад програвачі DVD-дисків застарілих моделей.

Для створення аудіодиска потрібно:

1. Уставити чистий диск у пристрій для роботи з оптичними дисками.
2. Відкрити вкладку **Записування** (рис. 4.6).
3. Перетягнути з **Області відомостей** в **Область записування** (права частина вікна) твори або альбоми, які планується записати на диск.

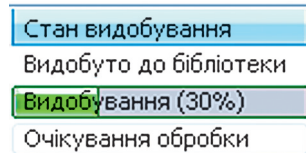
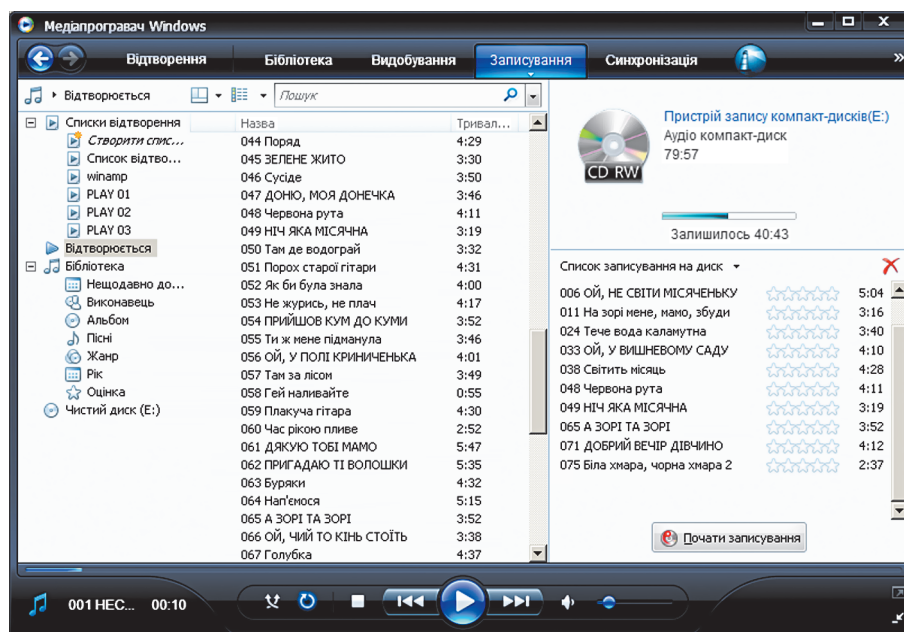




Рис. 4.5. Відображення ходу конвертації

Рис. 4.6. Вкладка **Записування** вікна **Медіапрогравач Windows**

4. Визначити за індикатором у верхній частині **Області записування** загальну тривалість відтворення даних з диска.
5. Вибрати кнопку відкриття списку **Список записування на диск** і вибрати команду **Перейменувати список відтворення**.
6. Увести в поле назву диска (якщо в **Області записування** з **Області відомостей** було перетягнуто альбом певного виконавця, то назву диска вводити не потрібно – вона буде введена автоматично за назвою альбому).
7. Відкрити за потреби список команд кнопки **Записування**, вибрати команду **Додаткові можливості** та на вкладці **Записування** встановити:
 - швидкість записування;
 - чи збільшувати тривалість звучання диска за рахунок зниження якості запису;
 - чи витягати диск після завершення записування.
8. Вибрати кнопку **Почати записування**  **Почати записування**.
9. Дочекатися завершення спочатку обробки (конвертації), а потім записування даних на диск (рис. 4.7).
10. Видалити список записування на диск вибором кнопки , якщо не плануєте записувати інші диски з тим самим вмістом.

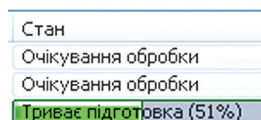


Рис. 4.7. Відображення ходу записування даних на диск



Крім медіаплеєрів, існує дуже багато спеціалізованих програм для здійснення конвертації різноманітних мультимедійних файлів. Прикладом такої програми є програма **Total Video Converter** (англ. **Total Video Converter** – загальний відеоконвертор). Завантажити її можна із сайту <http://www.effectmatrix.com>, а українізатор можна знайти за адресою <http://gettyfile.com.ua/397133>.

Незважаючи на назву, у якій ідеться про перетворення відео, зазначена програма здійснює конвертацію як відеофайлів більшості форматів MP4, AVI, 3GP, MPG, MPEG, WMV, FLV, DIVX тощо, так і звукових файлів форматів CDA, MP3, MP2, WAV, WMA та ін.

Програма **Total Video Converter** надає користувачу такі можливості:

- конвертувати медіадані з одного формату файлу в інший з використанням різних типів кодеків;
- створювати потокові презентації з використанням власних шаблонів зміни кадрів або переходу між відеофрагментами;
- конвертувати кілька медіафайлів одночасно;
- об'єднувати кілька відеороликів і аудіозаписів в одному файлі;
- розділяти відео- та звукові доріжки і записувати їх у різні файли та ін.

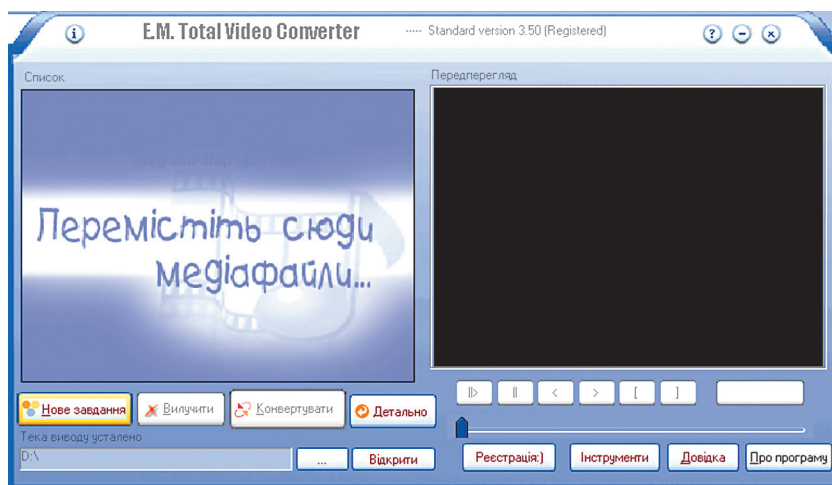
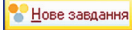

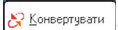


Рис. 4.8. Вікно програми **Total Video Converter**

Вікно програми **Total Video Converter** подано на рисунку 4.8. Розглянемо послідовність конвертування файлів на прикладі конвертації відеофайлу формату MOV у файл формату AVI. Для цього потрібно виконати такі дії:

1. Вибрати кнопку **Нове завдання**  і в списку вибрати команду **Імпорт медіафайлів**.
2. Вибрати файл, який потрібно конвертувати.
3. Вибрати у вікні **Вибір формату конвертування** вкладку відповідно до типу медіаданих (у нашому випадку – вкладку **Відеофайли**).
4. Вибрати кнопку з назвою формату, у який планується конвертувати вибраний файл – AVI.
5. Вибрати в списку, що відкрився, назву кодека, який буде використано під час конвертації, наприклад WMV AVI.
6. Змінити за потреби папку, у яку буде записано конвертований файл, використовуючи кнопку **Вибери теку для виведення файлів** .
7. Установити за потреби якість кодування даних, використовуючи список і гіперпосилання **Властивості**.
8. Вибрати кнопку **Конвертувати** .
9. Дочекатися завершення конвертації файлу.

Процес конвертування залежно від розмірів файлів, алгоритмів стиснення та потужності комп'ютера, на якому відбувається конвертація, може тривати досить

довго. Для виконання цієї операції у фоновому режимі необхідно встановити відповідну позначку прапорця у вікні **Converting** (англ. *converting* – перетворення) (рис. 4.9).

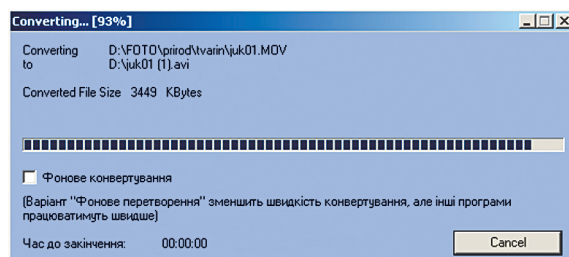


Рис. 4.9. Вікно відображення ступеня виконання завдання конвертації

Аналогічно можна конвертувати й інші аудіо- та відеофайли різних форматів.







Перевірте себе

- 1°. Які дані називаються мультимедійними? Наведіть приклади.
- 2°. Назвіть джерела мультимедійних даних.
- 3°. Які програми належать до програм опрацювання мультимедійних даних?
- 4°. Опишіть класифікацію програм опрацювання звукових і відеоданих за призначенням.
- 5°. Які програми називають мультимедіаплеєрами?
- 6°. Опишіть можливості **Медіапрогравача Windows**.
- 7°. Опишіть режими роботи **Медіапрогравача Windows**.
- 8°. Поясніть, як записати **Аудіо CD**, використовуючи **Медіапрогравач Windows**.
- 9°. Поясніть, як зберегти записи з **Аудіо CD** у звукових файлах на жорсткому диску вашого комп'ютера, використовуючи **Медіапрогравач Windows**.
- 10°. Опишіть, як створити **Список відтворення** у **Медіапрогравачі Windows**.
- 11°. Для чого використовується компактний режим **Медіапрогравача Windows**?
- 12°. Які програми називають граберами?
- 13°. Які програми називають конверторами?
- 14°. Поясніть, чим відрізняються музичні та відеоредактори від музичних та відеостудій.
- 15°. Назвіть відомі вам формати мультимедійних дисків і файлів.
- 16°. Поясніть, чим відрізняється файл формату WAV від файлу формату MP3.
- 17°. Що таке кодеки? Для чого вони призначені?
- 18°. Опишіть послідовність конвертації файлів з використанням однієї з програм конвертації, наприклад **Total Video Converter**.



Виконайте завдання

- 1°. Перегляньте список інсталюваних програм на вашому комп'ютері. Установіть, які програми для опрацювання мультимедійних даних є серед інсталюваних. Визначте, використовуючи довідку цих програм, який їх тип і для чого вони призначені.
- 2°. Проведіть пошук мультимедійних файлів на вашому комп'ютері, використовуючи стандартні засоби операційної системи. Визначте, файли яких форматів було знайдено.

- 3*. Запишіть, використовуючи **Медіапрогравач Windows, Audio CD**, включивши до нього звукові файли з папки, указаної вчителем, наприклад **Тема 4\Завдання 4.1\Музика**.
-  4*. Запишіть, використовуючи **Медіапрогравач Windows**, у файли формату WMA два довільні твори з **Audio CD**, що є у вашій домашній колекції.
-  5*. Створіть у **Медіапрогравачі Windows** список відтворення з іменем, що відповідає вашому прізвищу, включивши до нього музичні твори з папки, вказаної вчителем, наприклад **Тема 4\Завдання 4.1\Музика**.
-  6*. Проведіть конвертування двох перших файлів формату MP3 з папки, вказаної вчителем, наприклад **Тема 4\Завдання 4.1\Музика**, у формат WMA, використовуючи один з конверторів звукових файлів. Збережіть ці файли у папці **Вправа 4.1.6**, яку створіть у вашій папці.
-  7*. Проведіть конвертування двох перших файлів формату MOV з папки **Тема 4\Завдання 4.1\Відео** у формат WMV, використовуючи один з конверторів відеофайлів. Збережіть ці файли у папці **Вправа 4.1.7**, яку створіть у папці **Мої документи**.

4.2. Додавання звукових і відеооб'єктів до слайдових презентацій



1. Що таке слайдова презентація? Які об'єкти вона може містити?
2. Які формати звукових і відеофайлів ви знаєте?
3. Які програми опрацювання звукових і відеоданих ви знаєте?
4. Які програми називають граберами?
5. Що таке анімація? Як додати анімацію до об'єктів слайда?

Як ви вже знаєте, слайдові презентації, крім текстових і графічних об'єктів, можуть містити звукові та відеооб'єкти. Перед включенням до презентації в більшості випадків ці об'єкти заздалегідь створюються та редагуються. Це пов'язано з тим, що в програмі **PowerPoint 2007** до презентацій можна включити тільки файли певних типів: додавати звукові файли форматів WMA, WAV, MP3, MIDI, AU, AIFF та відеофайли форматів WMV, AVI, MPEG, ASF. Тому перед включенням до презентацій файли інших форматів слід конвертувати.




Ще одним способом підготовки звукових і відеооб'єктів є захоплення та запис у файли звуку та відео з використанням спеціальних програм – граберів.

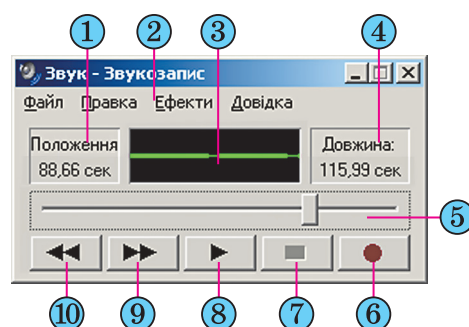
Захоплення аудіо

Однією з найпростіших у використанні програм для записування (захоплення) звуку є програма, що входить до стандартних програм операційної системи **Windows**, – програма **Звукозапис**. Для її запуску слід виконати **Пуск** ⇒ **Усі програми** ⇒ **Стандартні** ⇒ **Розваги** ⇒ **Звукозапис**.

Програма **Звукозапис** призначена для записування, відтворення та редагування звукозаписів. Вона забезпечує збереження звукозапису у файлах формату WAV без стиснення та відповідно без втрати даних. Тривалість одного сеансу запису не може перевищувати 60 с.

Для **створення нового звукозапису** з використанням мікрофона або іншого під'єданого до входу звукової карти пристрою введення звуку необхідно:


1. Виконати **Файл** ⇒ **Створити** (рис. 4.10).
2. Вибрати кнопку **Запис** .
3. Проговорити в мікрофон потрібне повідомлення (або відтворити на іншому пристрої потрібне звукове повідомлення).
4. Зупинити процес запису вибором кнопки **Стоп** .
5. Прослухати створений звукозапис, вибравши кнопку **Відтворити** .
6. Зберегти створений звукозапис у файлі, виконавши **Файл** ⇒ **Зберегти (Зберегти як)**.






- | | |
|--|------------------------------------|
| ① Індикатор поточного положення запису | ⑥ Кнопка Запис |
| ② Рядок меню | ⑦ Кнопка Стоп |
| ③ Вікно індикатора рівня запису | ⑧ Кнопка Відтворити |
| ④ Індикатор довжини (тривалості) всього запису | ⑨ Кнопка Перейти в кінець |
| ⑤ Повзунок ходу відтворення/запису | ⑩ Кнопка Перейти на початок |

Рис. 4.10. Вікно програми **Звукозапис**




Для **прослуховування** раніше створеного звукозапису необхідно:

1. Виконати **Файл** ⇒ **Відкрити**.
2. Вибрати файл, який потрібно прослухати.
3. Вибрати кнопку **Відтворити** .

Відтворення звукозапису можна зупиняти вибором кнопки **Стоп** , а потім продовжити відтворення з цієї самої позиції або з іншої, вибравши її повзунком ходу відтворення/запису та кнопку **Відтворити**. Для швидкого переходу до початку звукозапису або до його кінця слід вибрати кнопки **Перейти на початок**  або **Перейти в кінець**  відповідно.

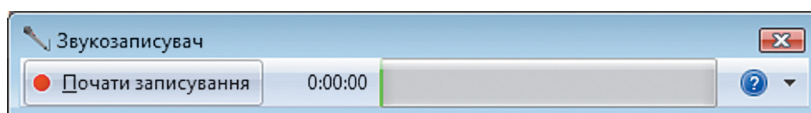
Програма **Звукозапис** надає користувачу також можливості в редагуванні та форматуванні звукозаписів. Для виконання операцій редагування або форматування звукозапису треба відкрити файл, що містить потрібний звукозапис, і виконати дії, описані в таблиці 4.4.

Таблиця 4.4. Операції редагування та форматування звукозаписів у програмі Звукозапис

Назва операції	Опис виконання операції
Додати до вже існуючого або замінити фрагмент звукозапису	Установити повзунок ходу відтворення/запису в кінець існуючого звукозапису (додавання) або в позицію, з якої розпочнеться новий запис (заміна), і вибрати кнопку Запис  . Для завершення запису вибрати кнопку Стоп 
Змінити гучність звучання	Виконати Ефекти \Rightarrow Збільшити гучність (на 25 %) або Ефекти \Rightarrow Зменшити гучність
Змінити швидкість відтворення звукозапису	Виконати Ефекти \Rightarrow Збільшити швидкість (на 100 %) або Ефекти \Rightarrow Зменшити швидкість (слід зважати на те, що зміна швидкості спотворює звук)
Відтворити звукозапис у зворотному порядку	Виконати Ефекти \Rightarrow Обернути і вибрати кнопку Відтворити 
Змінити якість звукозапису	Виконати Файл \Rightarrow Властивості , у вікні Властивості об'єкта «Звук» вибрати кнопку Перетворити і в наступному вікні вказати формат файлу для перетворення та атрибути
Змінити формат звукового файлу	Виконати Файл \Rightarrow Зберегти як , у вікні Зберегти як вибрати кнопку Змінити і в наступному вікні вказати формат файлу збереження та атрибути
Видалити частину звукозапису	Перемістити повзунок ходу відтворення/запису в позицію, починаючи з якої треба видалити звукозапис, і виконати Правка \Rightarrow Видалити до поточної позиції (Видалити після поточної позиції)
Додати до існуючого звукозапису аудіодані з іншого файлу	Перемістити повзунок ходу відтворення/запису в позицію, починаючи з якої потрібно додати звукозапис з файлу, виконати Правка \Rightarrow Вставити файл і вибрати файл, звукозапис з якого треба додати. У ході додавання старий звукозапис буде замінено звукозаписом з файлу з поточної позиції
Змішати звукозаписи (міксування) (англ. <i>mixing</i> – змішувати)	Перемістити повзунок ходу відтворення/запису в позицію, починаючи з якої потрібно змішати існуючий звукозапис з новим, виконати Правка \Rightarrow Змішати з файлом і вибрати файл, звукозапис з якого треба додати для змішування. Під час змішування один запис накладається на інший

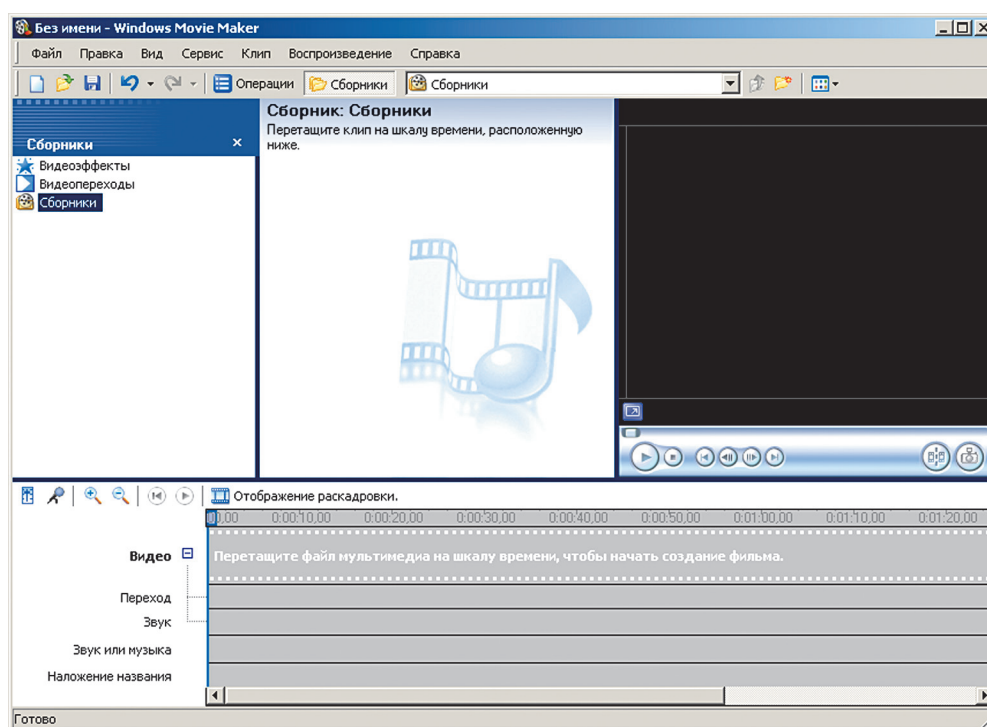


В операційній системі **Windows Vista** для запису та відтворення звукозаписів використовується дещо інша стандартна програма – **Звукозаписувач** (рис. 4.11). У ній відсутні можливості щодо редагування звукозаписів.

Рис. 4.11. Вікно програми **Звукозаписувач**

Захоплення відео

Для записування відео існує багато різноманітних програм. Розглянемо виконання записування (захоплення) відео на прикладі програми **Windows Movie Maker** (англ. *Movie Maker* – той, що створює кіно), яка входить до складу стандартних прикладних програм операційної системи **Windows**. Ця програма є відеоредактором. Для запуску програми слід виконати *Пуск* ⇒ *Усі програми* ⇒ *Windows Movie Maker*. Вікно програми подано на рисунку 4.12.

Рис. 4.12. Вікно програми **Windows Movie Maker**

Під час записування відео може здійснюватися запис і відео, і звуку або тільки відео. Відповідно до комп'ютера слід під'єднати пристрої введення відео- і звукових даних або тільки відеоданих.

Для записування відео необхідно виконати такі дії:

1. Виконати *Файл* ⇒ *Запис відео* (рос. *запись видео* – запис відео).
2. Вибрати пристрій, з якого буде здійснено записування відео, у полі *Доступные устройства* (рос. *доступные устройства* – доступні пристрої) вікна *Мастер видеозаписи* (рос. *мастер видеозаписи* – майстер з відеозапису) (рис. 4.13).

3. Установити значення параметрів пристроїв уведення звуку: тип пристрою для введення та для опрацювання звуку, рівень запису.
4. Вибрати кнопку **Далі** та перейти до наступного вікна **Мастер видеозаписи**.
5. Увести ім'я файлу відеозапису та шлях до папки, у яку він буде записаний.
6. Вибрати кнопку **Далі** та перейти до наступного вікна **Мастер видеозаписи**.
7. Установити вибором відповідного перемикача якості відтворення майбутнього відеозапису.
8. Вибрати за встановленого значення перемикача **Другие настройки** (рос. *другие настройки* – інші налаштування) у списку одне із значень налаштувань відео (у нижній частині вікна вказуються більш детальні відомості про обрані налаштування та розмір, який займе на диску файл із записом однієї хвилини відеозапису).
9. Вибрати кнопку **Далі** та перейти до наступного вікна **Мастер видеозаписи**.
10. Встановленням позначок прапорців вибрати:
 - чи створювати відеофайл після завершення роботи програми майстра з відеозапису;
 - чи відключати динаміки на час запису;
 - чи встановлювати обмеження на час запису.
11. Вибрати кнопку **Начать запись** (рос. *начать запись* – розпочати запис).
12. Вибрати кнопку **Остановить запись** (рос. *остановить запись* – зупинити запис) для завершення запису відеофрагмента.
13. Повторити кроки 10–11, якщо потрібно, записати кілька фрагментів.
14. Вибрати кнопку **Готово**.

Створені відеофрагменти будуть записані у файли у вибраній користувачем папці. У подальшому їх можна опрацьовувати, наприклад конвертувати у файли іншого формату, вставляти в документ тощо.

Додавання звукових об'єктів до слайдової презентації

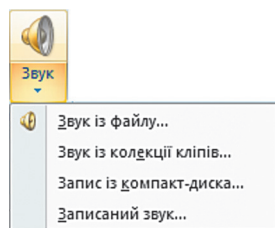


Рис. 4.14. Список команд кнопки **Звук**

Для додавання звукового об'єкта до слайдової презентації необхідно вибрати слайд, до якого він буде вставлений, і виконати **Вставлення** ⇒ **Медіа-кліпи** ⇒ **Звук**. Відкриється вікно для вставлення звуку із файлу. Якщо вибрати кнопку відкриття списку команд кнопки **Звук** (рис. 4.14), то можна вибрати один із чотирьох способів вставлення звукового об'єкта:

- **Звук із файлу** – вставлення звуку із файлу, що зберігається в зовнішній пам'яті або в мережі;

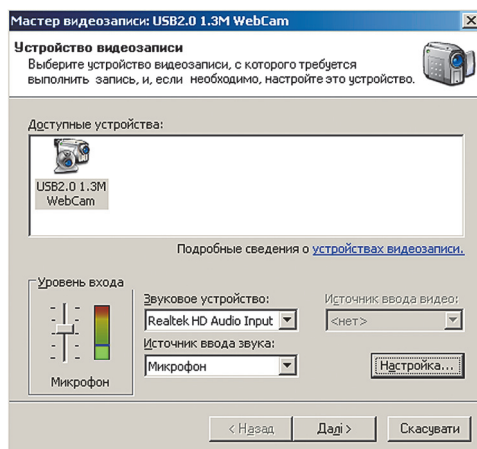


Рис. 4.13. Вікно майстра з відеозапису

- **Звук із колекції кліпів** – вставлення звуку із файлу, що включений до однієї з колекцій: колекції користувача, колекції **Microsoft Office** та **Веб-колекції**;
- **Запис із компакт-диска** – вставлення звуку з музичного компакт-диска, який знаходиться в пристрої для роботи з оптичними дисками;
- **Записаний звук** – запис звукового коментарю з мікрофона і вставлення його до слайда.

Розглянемо ці способи детальніше. Вставлення звукових об'єктів із файлів чи з колекції кліпів відбувається аналогічно відповідним операціям вставлення рисунків.

Під час вставлення звуку з музичного компакт-диска відкривається вікно **Вставлення аудіокомпакт-диска** (рис. 4.15). У цьому вікні користувач встановлює номер запису (звукового твору) на аудіодиску, з якого слід почати відтворення звуку, і номер запису, який буде останнім у відтворенні звуку. Час звучання кожного із цих записів можна задати відповідними лічильниками, а загальна тривалість звучання всіх записів автоматично зазначається в нижній частині вікна біля напису **Час відтворення**. Встановлення позначки прапорця **Безперервне відтворення** забезпечує безперервне відтворення обраних записів під час демонстрації презентації. Можна також установити гучність звучання записів (кнопка **Гучність звуку**) або приховати піктограму звукового об'єкта (прапорець **приховувати піктограму звуку під час показу**).

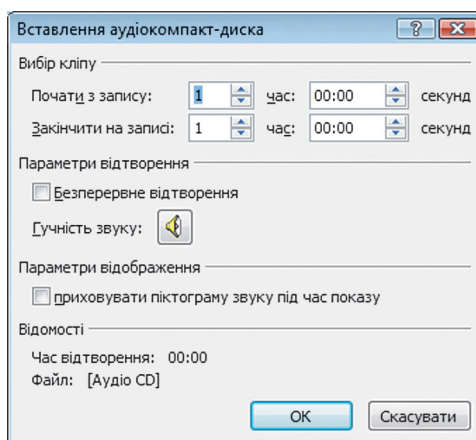


Рис. 4.15. Вікно **Вставлення аудіокомпакт-диска**

*Звертаємо увагу на те, що при заміні у пристрої аудіокомпакт-диска в ході демонстрації презентації буде відтворено ті записи, номери яких збігаються з встановленими у вікні **Вставлення аудіокомпакт-диска**. Якщо диск відсутній, то відтворення звуку не буде.*

Для вставлення до слайда звукового коментарю користувача після вибору команди **Записаний звук** необхідно у вікні **Запис звуку** (рис. 4.16) увести в поле назву звукового об'єкта та здійснити запис звуку за таким алгоритмом:

1. Підключити мікрофон до комп'ютера або скористатися вбудованим.
2. Вибрати кнопку **Запис**
3. Проговорити в мікрофон зміст коментарю.
4. Вибрати кнопку **Зупинити**
5. Прослухати записаний коментар, вибравши кнопку **Відтворити**

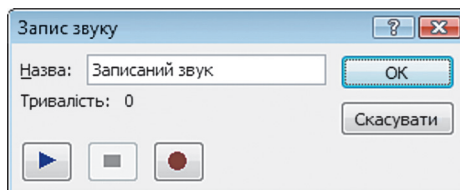


Рис. 4.16. Вікно **Запис звуку**

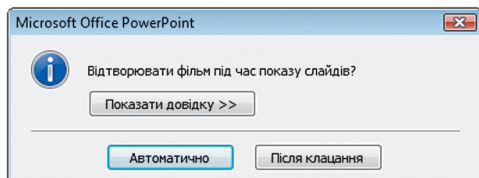



Рис. 4.17. Вікно встановлення налаштувань відтворення звуку

ти, як буде розпочинатися відтворення звуку на слайді: одразу, як тільки розпочнеться показ слайда (кнопка **Автоматично**), або після вибору піктограми звукового об'єкта на слайді під час демонстрації презентації (кнопка **Після клацання**).

Після вставлення звукового об'єкта на слайді з'являється його піктограма , якщо тільки не було обрано режим її приховування. За вибору піктограми звукового об'єкта на **Стрічці** з'являється додатковий розділ **Знаряддя для звуків** і тимчасова вкладка **Параметри**. Після вибору цієї вкладки для редагування та форматування звуку можна використати відповідні елементи керування.

За замовчуванням звукові файли типу wav, розмір яких більше 100 Кбайт, автоматично зв'язуються з файлом презентації, а не вбудовуються в нього. Для вбудованих файлів типу wav можна вказати максимальний розмір до 50 000 Кбайт, але це приведе до збільшення загального розміру файлу презентації. Якщо ж користувач під час вставлення вибере файл більшого розміру, ніж встановлено цим параметром, то файл не буде включено до презентації.

Відтворення звуку відбувається в режимі показу слайдів.

Вставлення і опрацювання відеооб'єктів

Для вставлення відеооб'єкта необхідно вибрати слайд, до якого він буде вставлений, і виконати **Вставлення** ⇒ **Медіакліпи** ⇒ **Фільм**. Відкриється вікно для вставлення фільму з файлу. Якщо вибрати кнопку відкриття списку команд кнопки **Фільм** (рис. 4.18), то можна вибрати один з двох способів вставлення відеооб'єкта:

- **Фільм із файлу** – вставлення відео із файлу, що зберігається в зовнішній пам'яті або в мережі;
- **Фільм із колекції кліпів** – вставлення відео із файлу, що включений до однієї з колекцій: колекції користувача, колекції **Microsoft Office** та **Веб-колекції**.

Аналогічно операції вставлення звуку під час вставлення відео користувачеві буде запропоновано встановити, яким чином буде розпочинатися відтворення відео: автоматично чи після вибору об'єкта на слайді у ході його демонстрації.

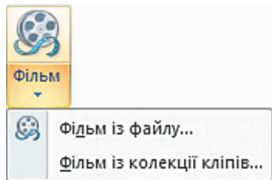


Рис. 4.18. Список команд кнопки **Фільм**

Після вставлення відеооб'єкта на слайді з'являється прямокутна область, у якій буде відтворюватися відеофільм. Розміри цієї області після її виділення можна змінити, використовуючи відповідні маркери або встановивши точні розміри лічильни-

6. Якщо коментар записано правильно, вибрати кнопку **ОК**, якщо ні – повторити команди 2–5 алгоритму.

На останніх кроках операції вставлення звукових об'єктів на екран виводиться вікно встановлення значень параметрів відтворення звуку (рис. 4.17). Необхідно вибрати

ками **Висота фігури** та **Ширина фігури** групи **Розмір** тимчасової вкладки **Формат** додаткового розділу **Знаряддя для зображення**. Можна також перемістити область відтворення відео стандартною операцією перетягування об'єкта.

Після виділення відеооб'єкта на **Стрічці** з'являється додатковий розділ **Знаряддя для фільмів** і тимчасова вкладка **Параметри**. Вибравши заголовок цієї вкладки, користувач отримує доступ до елементів керування, які встановлюють значення параметрів відеооб'єкта (рис. 4.19).

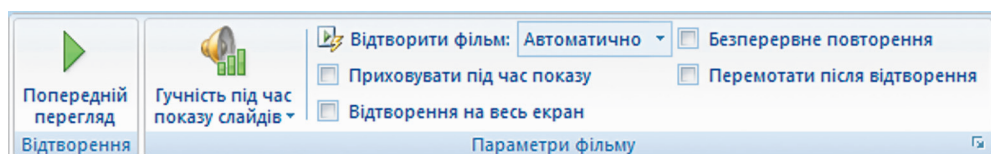


Рис. 4.19. Елементи керування тимчасової вкладки **Параметри** розділу **Знаряддя для фільмів**

Ці самі налаштування можна зробити, якщо вибрати кнопку відкриття діалогового вікна групи **Параметри фільму** і відкрити відповідне вікно.

Додавання анімаційних ефектів до звукових і відеооб'єктів слайда

Для додавання анімаційних ефектів до звукових і відеооб'єктів використовується та сама послідовність дій, як і для текстових або графічних. Тільки ефекти появи, виділення, виходу та шляхів переміщення застосовуються до

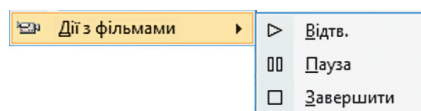



Рис. 4.20. Список команд **Дії з фільмами**

значка об'єкта  (звуковий об'єкт) або до вікна, у якому відтворюється відеооб'єкт. Список **Додати ефект** для цих об'єктів містить ще одну команду **Дії зі звуками** чи **Дії з фільмами** (рис. 4.20). В обох випадках у списку, що відкриється після вибору команди, можна вибрати одну з команд: **Відтв.** (відтворити), **Пауза** або **Закершити**.

Для встановлення значень параметрів відтворення ефектів анімації слід вибрати кнопку відкриття списку команд поточного ефекту і відкрити вікно, що має заголовок залежно від ефекту, параметри якого налаштовуються. Для відеооб'єкта, наприклад, це можуть бути такі вікна: **Фільм: відтворення**, **Фільм: пауза** або **Фільм: зупинка**.

На вкладці **Ефект** вікна **Фільм: відтворення** (рис. 4.21) користувач може встановити значення таких параметрів:

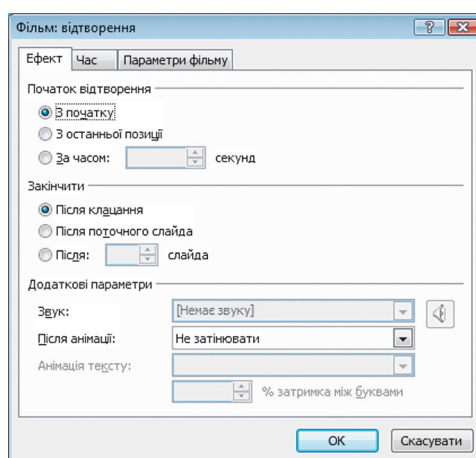


Рис. 4.21. Вкладка **Ефект** вікна **Фільм: відтворення**

- початок відтворення: з початку, з останньої позиції або за часом;
- завершення відтворення: після натиснення лівої кнопки миші (після клацання), після поточного або слайда із заданим номером;
- дії після анімації – аналогічно діям для текстових і графічних об'єктів.

На вкладці **Параметри фільму** вікна **Фільм: відтворення** користувач може встановити значення таких параметрів:

- гучність звуку відеофільму;
- приховування за відсутності відтворення;
- відображення відео на весь екран.

Налаштування ефектів звукових об'єктів аналогічне налаштуванню ефектів відео за виключенням відображення на весь екран.

Налаштувати параметри відтворення відео- (звукових) об'єктів можна також з використанням елементів керування груп **Параметри фільму** (**Параметри звуку**) та **Відтворення** вкладки **Параметри** додаткового розділу **Знаряддя для фільму** (рис. 4.22 і 4.23).

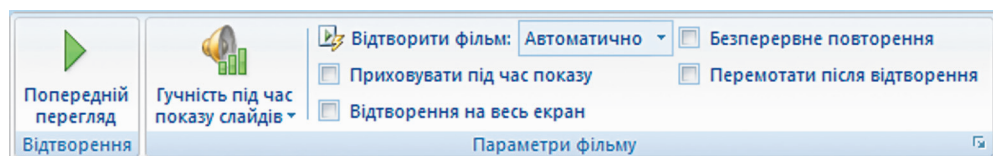


Рис. 4.22. Групи елементів керування **Параметри фільму** та **Відтворення** вкладки **Параметри** додаткового розділу **Знаряддя для фільму**

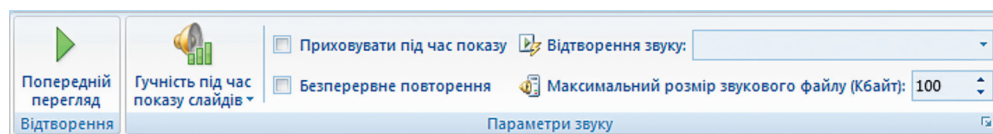


Рис. 4.23. Групи елементів керування **Параметри звуку** та **Відтворення** вкладки **Параметри** додаткового розділу **Знаряддя для звуку**






Перевірте себе

- 1°. Звукові файли яких форматів можна додавати до презентації в **PowerPoint 2007**?
- 2°. Відеофайли яких форматів можна додавати до презентації в **PowerPoint 2007**?
- 3°. Для чого призначена програма **Звукозапис**? Як її запустити?
- 4°. Поясніть операції, які можна виконати з використанням програми **Звукозапис**.
- 5°. Опишіть інтерфейс програми **Звукозапис**.
- 6°. Наведіть послідовність дій під час створення звукозапису в програмі **Звукозапис**.
- 7°. Наведіть послідовність дій під час прослуховування звукозапису в програмі **Звукозапис**.
- 8°. У якому форматі зберігає звукозаписи програма **Звукозапис**?
- 9*. Опишіть, як підготувати звукозапис тривалістю 3 хв у програмі **Звукозапис**.
- 10°. Опишіть, які можливості з редагування та форматування звукозапису надає користувачеві програма **Звукозапис**.

- 11*. Поясніть, як змішати один звукозапис з іншим. Для яких випадків використовують цю операцію?
- 12°. До якого виду програм належить програма **Windows Movie Maker**? Як її запустити?
- 13°. Опишіть виконання операції записування відео з використанням програми **Windows Movie Maker**.
- 14°. Як додати звуковий або відеооб'єкт до слайда презентації?
- 15°. Поясніть особливості вставлення звуку з компакт-диска. Які види оптичних дисків використовуються в цьому випадку?
- 16*. Порівняйте операції записування аудіо з використанням програми **Звукозапис** і вставлення до слайда звукового коментаря користувача після вибору команди **Записаний звук**. У яких випадках краще використовувати кожен із цих операцій?
- 17°. Опишіть, які параметри відтворення відеооб'єкта під час демонстрації презентації можна встановити та як це зробити.
- 18°. Опишіть відмінності у вставленні та встановленні параметрів відтворення звукового та відеооб'єктів, які вставляються з файлів до слайдової презентації.
- 19°. Поясніть, як установити безперервне відтворення звуку з однієї з доріжок компакт-диска протягом демонстрації всієї презентації.



Виконайте завдання


- 1°. Запишіть у власному виконанні декламацію вірша І. Франка «Сикстинська Мадонна» (**Тема 4\Завдання 4.2\Франко.docx**) з використанням програми **Звукозапис**. Збережіть документ у власній папці у файлі з іменем **вправа 4.2.1.wav**.
- 2°. Відкрийте в програмі **Звукозапис** указаний учителем файл, наприклад **Тема 4\Завдання 4.2\зразок 4.2.2.wma**. Допишіть до існуючого звукозапису декламацію вірша І. Франка «Котляревський» (**Тема 4\Завдання 4.2\Франко.docx**) у власному виконанні. Збережіть документ у власній папці у файлі з іменем **вправа 4.2.2.wav**.
-  3°. Запишіть у власному виконанні декламацію вірша І. Франка «Сонети – се раби. У форми пута...» (**Тема 4\Завдання 4.2\Франко.docx**) з використанням програми **Звукозапис**. Допишіть ще одну декламацію – вірша І. Франка «Колись в сонетах Данте і Петрарка...» (**Тема 4\Завдання 4.2\Франко.docx**). Збережіть документ у папці **Моя музика** у файлі з іменем **вправа 4.2.3.wav**.
- 4*. Відкрийте у програмі **Звукозапис** указаний учителем файл, наприклад **Тема 4\Завдання 4.2\зразок 4.2.4.wma**. Установіть повзунок ходу відтворення/запису в місце, де закінчується перший куплет вірша І. Франка «Сикстинська Мадонна». Видаліть частину звукозапису від цього місця до кінця звукозапису. Допишіть у цей звукозапис перший куплет вірша І. Франка «Котляревський» (**Тема 4\Завдання 4.2\Франко.docx**) у власному виконанні. Збережіть документ у власній папці у файлі з іменем **вправа 4.2.4.wav**.
-  5*. Використовуючи довідкову систему програми **Звукозапис**, визначте, що таке рівень запису. Як його змінити?
- 6°. Підключіть до комп'ютера веб-камеру та мікрофон. Виконайте з використанням програми **Windows Movie Maker** відеозапис декламації довільного фрагмента твору І. Франка «Сучасна приказка» (**Тема 4\Завдання 4.2\Франко.docx**). Збережіть відеозапис у форматі WMV у власній папці у файлі з іменем **вправа 4.2.6.wmv**.
-  7. Підготуйте разом з однокласниками групову декламацію одного з творів, що ви вивчаєте в курсі світової літератури. Виконайте з викорис-

танням програми **Windows Movie Maker** відеозапис підготовленої декламації. Збережіть відеозапис у форматі WMV у папці **Мої відеозаписи** у файлі з іменем **вправа 4.2.7.wmv**.

- 8°. Відкрийте вказаний учителем файл (наприклад, **Тема 4\Завдання 4.2\зразок 4.2.8.pptx**). Додайте до першого слайда звуковий об'єкт з файлу, наприклад **Тема 4\Завдання 4.2\звук 4.2.8.mid**. Установіть безперервне відтворення цього об'єкта під час всієї демонстрації презентації. Збережіть презентацію у власній папці у файлі з іменем **вправа 4.2.8.pptx**.


- 9°. Відкрийте вказаний учителем файл, наприклад **Тема 4\Завдання 4.2\зразок 4.2.9.pptx**. Додайте:

а) до слайда з розповіддю про білку замість її фотографії відеозапис з файлу, наприклад **Тема 4\Завдання 4.2\білка.avi**. Установіть приховування відеозапису до початку відтворення та відтворення автоматично після появи заголовка. Збережіть презентацію у власній папці у файлі з іменем **вправа 4.2.9.pptx**;

 б) до слайда з розповіддю про їжачка замість його фотографії відеозапис з файлу, наприклад **Тема 4\Завдання 4.2\їжак.avi**. Установіть відтворення цього об'єкта через 1 с після появи тексту та показ на весь екран. Збережіть презентацію в папці **Мої документи** у файлі з іменем **вправа 4.2.9_1.pptx**.

- 10°. Відкрийте вказаний учителем файл, наприклад **Тема 4\Завдання 4.2\зразок 4.2.10.pptx**. Перегляньте слайди презентації, присвяченої пристроям комп'ютера. Підготуйте короткі повідомлення про властивості сучасних пристроїв, які б супроводжували демонстрацію презентації. Запишіть мовленнєвий супровід для слайдів із зображеннями:

а) *монітора і планшетного сканера*. Установіть відтворення звукового супроводу одразу після переходу до слайдів, на яких розміщено об'єкти. Терміни показу слайдів не зберігати. Збережіть презентацію у власній папці у файлі з іменем **вправа 4.2.10.pptx**;

 б) *процесора*. Установіть відтворення звукового супроводу після натиснення лівої кнопки миші. Терміни показу слайда зберегти. Збережіть презентацію у папці **Мої документи** у файлі з іменем **вправа 4.2.10_1.pptx**.

4.3. Створення та редагування відеофільмів (потоківих презентацій)



1. Що, на вашу думку, відрізняє кінофільм від відеофільму? Що потрібно мати для перегляду відеофільму?
2. Що таке сценарій? Де він використовується?
3. Для чого можна використовувати програму **Windows Movie Maker**? Як її запустити?
4. Як виконати записування (захоплення) звуку, відео? Які програми здійснюють подібні операції? Як називається цей тип програм?

Загальні вимоги до створення відеофільму

Термін **відеофільм** походить від терміна **кінофільм** (англ. *cine-film* – кіноплівка, кінофільм), який у широкому розумінні означає твір кіномистецтва. Відеофільм відрізняє від кінофільму спосіб фіксації даних: кінофільм – на фотографічних плівках із використанням кінокамер, а відеофільм – на відеокасетах і відеодисках із використанням відеокамер. Для невеликих за розміром кіно- та відеофільмів використовують термін

кліп (англ. *clip* – стискати, обрізати), або **відеокліп**. Останнім часом відеокліпи широко використовуються у музичній індустрії. Їх називають **музичними відеокліпами**.

За своєю структурою і технологією створення **потоківі презентації** фактично є відеофільмами (відеокліпами), призначеними для представлення чого-небудь нового: ідей, проектів, продукції, товару тощо. Для створення поточкових презентацій використовують відеоредактори.

На першому етапі, перед початком створення відеофільму, необхідно розробити його сценарій. **Сценарій** – це детально розроблений план проведення якого-небудь заходу, здійснення яких-небудь дій. Сценарій навіть невеликого відеокліпу повинен мати внутрішню структуру (композицію). Сценарій повинен бути детальним, з посекундним плануванням усіх дій.

На другому етапі відбувається зйомка всіх епізодів згідно зі сценарієм, готуються фотографії, рисунки, підбирається музика, записується звуковий супровід. Часто відеофільм створюється на основі раніше відзнятого відеоматеріалу. Наприклад, фільм про певний період історії країни, фільм-розповідь про життя визначних людей, навіть фільм про ваш клас, мабуть, буде створюватися на основі вже існуючих документальних кадрів, фотографій, документів тощо.

На третьому етапі здійснюється **монтаж** – процес опрацювання та структурування окремих об'єктів (кадрів) для отримання цілісного твору – відеофільму. У процесі монтажу відзнятий матеріал редагується, уточнюється сценарій, накладається звук і титри – текстові коментарі до кадрів тощо.

Розробка сценарію відеофільму

Розглянемо процес створення сценарію відеофільму про птахів, що живуть біля річок. Його можна використати у ході вивчення теми **Види тварин** з курсу «Я і Україна» 3-го класу. Проаналізуємо наявний відеотата фотоматеріал (табл. 4.5).

Таблиця 4.5. Файли для включення до відеофільму

Ім'я файлу	Папка розміщення	Вміст	Тривалість, с
лисуха.avi	Тема 4\Відео	Відео	42
лелека.avi	Тема 4\Відео	Відео	31
лисуха01.jpg	Тема 4\Фото	Фотографії	–
лисуха02.jpg	Тема 4\Фото	Фотографії	–
лелека01.jpg	Тема 4\Фото	Фотографії	–
лелека02.jpg	Тема 4\Фото	Фотографії	–
лелека.wav	Тема 4\Звук	Звук	4
лисуха.wav	Тема 4\Звук	Звук	6,75

Відповідно до програми початкової школи та вимог до часу демонстрацій на уроці для учнів 3-го класу складемо сценарій нашого відеофільму. Сценарій складається зі **сценарних кадрів** – опису змістових фрагментів відеофільму та тих подій, які в них відбуваються.

1. Кадр 1. Титульна сторінка:

- заголовок – *Птахи України*, підзаголовок – *Для учнів 3-го класу*;
- колір тла – *зелений*, колір літер – *білий*;
- тривалість показу – *4 с*;
- звук – *відсутній*;
- відеоефект – *зернистість*;
- ефект переходу на наступний кадр – *око*.

2. Кадр 2. Фотографія лелеки:

- файл – *лелека02.jpg*;
- титри – *Лелека*; підзаголовок (*чорногуз, бусол, бузько*);
- тривалість показу – *4 с*;
- звук – файл *лелека.wav*; тривалість звучання – *4 с*;
- ефект переходу на наступний кадр – *вихор*.

3. Кадр 3. Фотографія лелеки:

- файл – *лелека01.jpg*;
- тривалість показу – *4 с*;
- звук – *відсутній*;
- ефект переходу на наступний кадр – *метелик, вертикальний*.

4. Кадр 4. Відео Лелеки:

- файл – *лелека.avi*;
- тривалість показу – *26,14 с* (загальна тривалість – *31 с*, необхідно видалити на початку *4,86 с*; оригінальний звук відеозапису треба видалити);
- звук – *дикторський текст*; тривалість – *22,14 с*;
- ефект переходу на наступний кадр – *вставити, зліва, знизу*;
- дикторський текст: *Ви побачили птаха та почули його голос, точніше стукіт дзьоба. Це лелека, або чорногуз, бусол, бузько. Має біле пір'я, а на крилах чорне. У лелеки довгі ноги та довгий дзьоб червоного або жовто-червоного кольору. В Україні лелека вважається вісником весни.*

5. Кадр 5. Фотографія качки:

- файл – *лисуха02.jpg*;
- титри – *Євразійська, або звичайна, лиска (лисуха)*;
- тривалість показу – *4 с*;
- звук – файл *лисуха.wav*; початок звучання через *2 с* після початку кадру, тривалість звучання – *6,75 с* (звук продовжується на наступному кадрі);
- ефект переходу на наступний кадр – *вставити, зліва, зверху*.

6. Кадр 6. Фотографія качки:

- файл – *лисуха01.jpg*;
- тривалість показу – *4 с*;
- звук – файл *лисуха.wav*; розпочалося звучання на попередньому кадрі, завершується разом із завершенням показу фотографії;
- ефект переходу на наступний кадр – *діафрагма*.

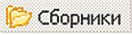
7. Кадр 7. Відео Лиски:



- файл – *лисуха.avi*;
- тривалість показу – 30 с (загальна тривалість – 42 с, необхідно видалити в кінці 12 с; оригінальний звук відеозапису зробити фоном для дикторського тексту);
- ефект переходу на наступний кадр – *діагональ, ніша*;
- звук – *дикторський текст*; тривалість – 30 с;
- дикторський текст: *Євразійська, або звичайна, лиска, яку ще називають лисуха, – невеликий водоплавний птах, що гніздиться біля річок і озер на всій території України. Її просто впізнати за білим дзьобом та білою плямою на лобі. Значну частину часу лиска проводить на воді, за що отримала ще одну народну назву: болотна, або водяна, курка.*

8. Кадр 8. Заклучні титри:

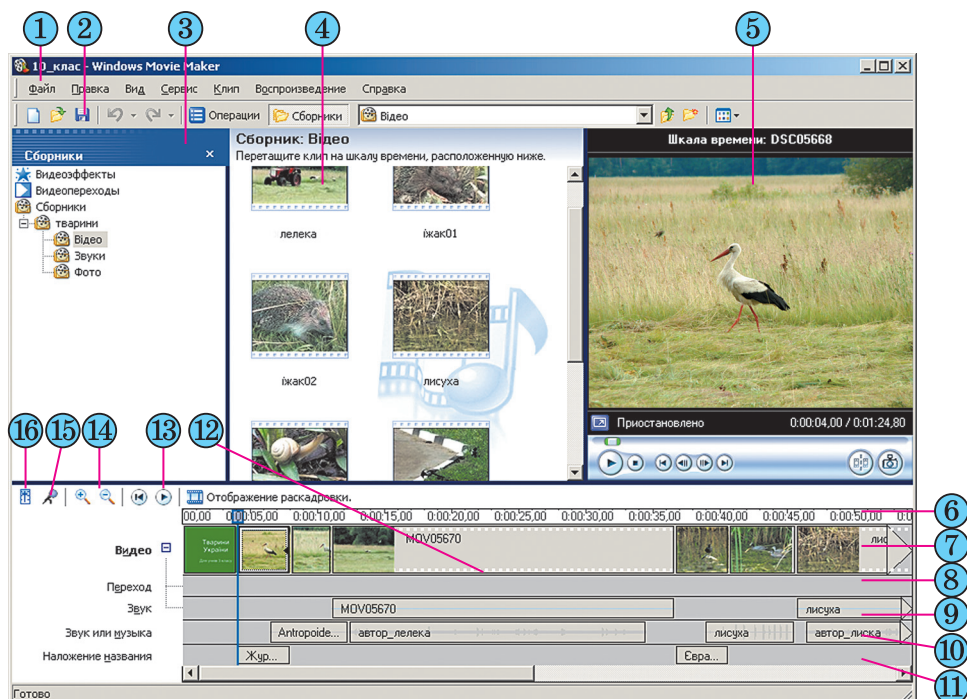
- титри – *Житомирський ліцей № 1, відеостудія «Майбутнє»*:
 - *Оператор – Михайленко Степан*;
 - *Автор тексту – Коваленко Оксана*;
 - *Звук – Іваненко Миколи*;
 - *Учні 10-А класу для учнів 3-х класів*;
- колір тла – *жовто-зелений*, колір літер – *білий*;
- тривалість показу – 10 с.

Створення відеофільму

Перейдемо до створення відеофільму в програмі **Windows Movie Maker**. Для цього розглянемо детальніше інтерфейс цієї програми (рис. 4.24). У лівій частині вікна після вибору кнопки **Сборники**  (рос. *сборники* – збірники) відображається однойменна панель (рис. 4.24, 3) з трьома стандартними папками **Видеоэффекты** (рос. *видеоэффекты* – відеоефекти), **Видеопереходы** (рос. *видеопереходы* – відеопереходи) і **Сборники**. За вибору папки **Видеоэффекты** в **Області відображення вмісту папок** (рис. 4.24, 4) відобразиться список відеоефектів (*відображення у відтінках сірого, відображення у вигляді акварельного малюнка, відображення зверху вниз тощо*), які можна застосувати до окремих кадрів або до фільму в цілому. Папка **Видеопереходы** містить список ефектів переходу від одного кадру до іншого. Папку **Сборники** користувач заповнює сам різними об'єктами – графічними, звуковими та відеофайлами, які потім зможе включити до своїх фільмів.

У правій частині вікна розміщено вікно програвача (рис. 4.24, 5) для попереднього перегляду результатів **монтажу**. Нижню частину вікна займає **Область монтажу** (рис. 4.24, 12). Вона відображається в одному з двох режимів: у **режимі розкадровки** або в **режимі відображення шкали часу**. Для перемикання режимів **Області монтажу** використовуються кнопки **Отображение шкалы времени**  (рос. *отображение шкалы времени* – відображення шкали часу) і **Отображение раскадровки**  (рос. *отображение раскадровки* – відображення розкадровки).

Область монтажу в режимі відображення шкали часу містить: звукову доріжку (рис. 4.24, 9), відеодоріжку (рис. 4.24, 7), звукову або музич-



- | | |
|--------------------------------------|---|
| 1) Рядок меню | 9) Звукова доріжка |
| 2) Панель інструментів | 10) Звукова або музична доріжка |
| 3) Панель Сборники | 11) Доріжка титрів |
| 4) Область відображення вмісту папок | 12) Область монтажу |
| 5) Вікно програвача | 13) Кнопки керування попереднім переглядом |
| 6) Шкала часу | 14) Кнопки керування масштабом шкали часу |
| 7) Відеодоріжка | 15) Кнопки ввімкнення режиму запису коментарю |
| 8) Доріжка переходів | 16) Кнопка ввімкнення режиму регулювання (встановлення рівня) звуку |


Рис. 4.24. Вікно програми **Windows Movie Maker**


ну доріжку (рис. 4.24, 10), доріжку титрів (рис. 4.24, 11). У верхній частині області розміщено шкалу часу (рис. 4.24, 6).

У режимі розкадровки в **Області монтажу** відображаються кадри – ті відео- або графічні фрагменти, з яких складається відеофільм.

Режим відображення шкали часу використовується в основному для вставлення мультимедійних об'єктів і визначення тривалості їхнього показу або звучання, а режим розкадровки – для застосування до кадрів відео ефектів та ефектів переходу між кадрами.

На *першому етапі відеомонтажу* необхідно створити новий проект. Для цього слід виконати **Файл** ⇒ **Создать проект** (рос. *Создать проект* –

створити проект) або вибрати кнопку **Создать проект**  на панелі інструментів. Проект містить посилання на мультимедійні об'єкти, що входять до складу фільму, значення їх властивостей у конкретному фільмі та інші дані, що потрібні для роботи над монтажем фільму. Для збереження проектів програма **Windows Movie Maker** використовує спеціальний формат MSWMM (англ. *Microsoft Windows Movie Maker*). Користувачу потрібно періодично зберігати файл проекту для того, щоб у разі збоїв у роботі комп'ютера результати монтажу не були втрачені.

Файли об'єктів, що будуть входити до фільму, слід помістити в один зі збірників. Для впорядкування об'єктів бажано заздалегідь створити в збірнику папки, наприклад як подано на рисунку 4.24: папка **тварини**, в якій створені папки **Відео**, **Звуки**, **Фото**. Для створення папки в обраній необхідно скористатися кнопкою **Создать папку сборников**  (рос. *создать папку сборников* – створити папку збірників) або відповідною командою контекстного меню папки **Сборники**.

На **другому етапі** слід вставити мультимедійні об'єкти зі збірника в область монтажу перетягуванням. Це можна виконати так:

1. Установити режим розкадровки вибором відповідної кнопки.
2. Перетягнути по черзі відповідно до розробленого сценарію з папок **Відео** та **Фото** відео- та графічні об'єкти у великі прямокутники (кадри) **Області монтажу** (рис. 4.25).
3. Перейти в режим відображення шкали часу.
4. Перетягнути по черзі відповідно до розробленого сценарію звукові об'єкти з папки **Звуки** в **Область монтажу** в ті точки на шкалі часу, з яких розпочнеться відтворення звуку.
5. Відредагувати тривалість показу вставлених об'єктів.

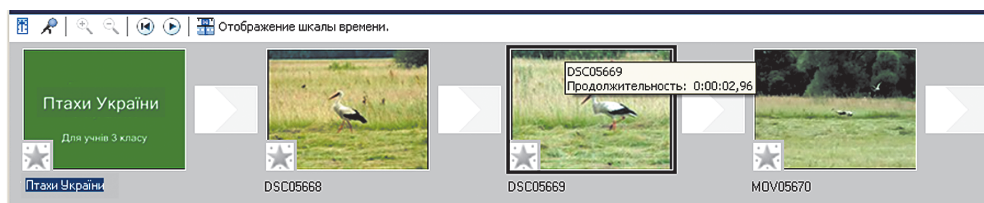


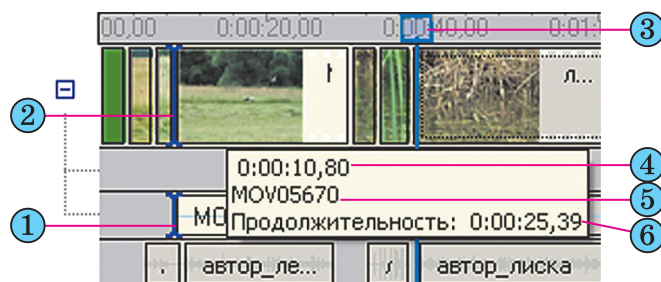


Рис. 4.25. **Область монтажу** в режимі розкадровки

Якщо вставляти відеооб'єкт у режимі відображення шкали часу, то під час перетягування біля вказівника з'являється позначка , а на відеодоріжці та на звуковій доріжці – позначка , яка вказує на місце вставлення об'єкта (рис. 4.26, 1–2). Біля вказівника також з'являється прямокутник з повідомленнями про момент часу в загальній тривалості фільму, з якого почнеться демонстрація об'єкта (наприклад, *0 год 0 хв 10,80 с*), про ім'я об'єкта, біля якого вставляється вибраний об'єкт (наприклад, *MOV05670*), і тривалість показу об'єкта (наприклад, *0 год 0 хв 25,39 с*).

Під час вставлення графічних об'єктів програмою встановлюється стандартна тривалість показу – 4 с. Для зміни цієї тривалості показу необхідно перетягнути праву межу об'єкта на відеодоріжці (рис. 4.27). Можна як збільшити, так і зменшити тривалість показу графічного об'єкта.



- ① Позначка місця вставлення об'єкта на звуковій доріжці
- ② Позначка місця вставлення об'єкта на відеодоріжці
- ③ Позначка місця початку поточного об'єкта на шкалі часу
- ④ Момент часу в загальній тривалості фільму, з якого почнеться демонстрація об'єкта
- ⑤ Ім'я об'єкта, поруч з яким вставляється даний об'єкт
- ⑥ Тривалість показу об'єкта, що вставляється

Рис. 4.26. Повідомлення і позначки під час вставлення об'єкта в **Область монтажу**

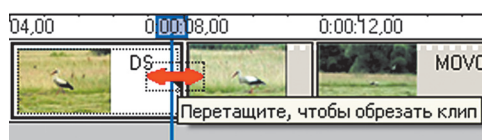


Рис. 4.27. Зміна тривалості показу графічного об'єкта шляхом перетягування його правої межі

Для відеооб'єктів тривалість їхнього показу дорівнюватиме встановленій тривалості у відповідному відеофайлі. Зміна тривалості показу відеооб'єкта здійснюється аналогічно зміні тривалості показу графічного об'єкта, тільки можна перетягувати як ліву, так і праву межу відеооб'єкта на відеодоріжці. При

цьому фрагмент відеооб'єкта не буде відтворюватися від його початку або кінця до моменту часу, визначеного положенням межі. Для більш точного редагування об'єкта в **Області монтажу** можна збільшити/зменшити масштаб відображення об'єктів. Для цього використовують кнопку **Увеличить/уменьшить шкалу времени** (рос. *увеличить/уменьшить шкалу времени* – збільшити/зменшити шкалу часу).

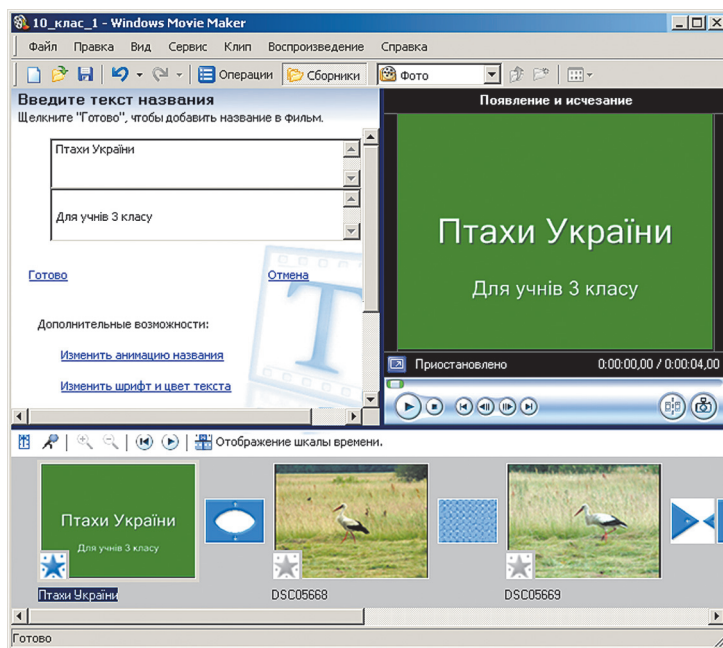
Під час перетягування меж у вікні програвача відображається поточний відеокадр, що дає змогу зорієнтуватися, до якого місця слід перетягнути межу. Це процес оборотний і можна в подальшому відновити тривалість показу відеооб'єкта, перетягнувши межу у зворотному напрямі. Разом з тим збільшити тривалість показу більше, ніж це передбачено властивостями файлу, неможливо.

Звукові об'єкти вставляються та редагуються аналогічно відеооб'єктам, тільки перетягуються вони на звукову або музичну доріжку.

У ході вставлення звукових і відеооб'єктів та під час їхнього редагування (зміни тривалості, зміни моменту часу початку відтворення) відбувається **синхронізація** звукового і відеоряду.

Результати монтажу слід періодично переглядати у вікні програвача.

На наступному **третьому етапі монтажу відеофільму** вставляються титри або текстовий супровід. Приклад початкових титрів подано на рисунку 4.28.

Рис. 4.28. Вікно **Windows Movie Maker** у режимі редагування титрів

Для додавання текстових написів до відеофільму необхідно виконати таку послідовність дій:

1. Виконати **Сервис** \Rightarrow **Названия и титры** (рос. *названия и титры* – назви і титри).
2. Вибрати на панелі **Где добавит название?** (рос. *где добавит название* – де додати назву) одне з гіперпосилань, що відповідає місцю вставлення напису. Розрізняють початкові та заключні титри, а також титри, що розміщуються на графічних чи відеооб'єктах або перед ними.
3. Увести текст написів у відповідні поля на панелі **Введите текст названия** (рос. *введите текст названия* – уведіть текст назви).
4. Змінити за потреби для написів, що розміщуються не на графічних чи відеооб'єктах, колір тла, колір літер, шрифт та накреслення, а також ефект анімації (гіперпосилання **Изменить шрифт и цвет текста** (рос. *изменить шрифт и цвет текста* – змінити шрифт і колір тексту) та **Изменить анимацию текста** (рос. *изменить анимацию текста* – змінити анімацію тексту)).
5. Переглянути, як буде виглядати напис у відеофільмі, для цього вибрати кнопку **Воспроизвести** (рос. *воспроизвести* – відтворити) у вікні програвача.
6. Вибрати гіперпосилання **Готово** (рос. *готово* – готово).

Створені написи будуть вставлені або на відеодоріжку, або на доріжку титрів. На відеодоріжку будуть вставлені всі написи, за виключенням тих, що вставляються на певному кадрі (графічному або відео).

Для вставлення титрів і написів усередині відеофільму та заключних титрів слід повторити наведений алгоритм потрібну кількість разів.


Тривалість демонстрації вставлених титрів за замовчуванням складає 3,5 с. Змінити її можна таким саме чином, як і тривалість демонстрації графічних об'єктів. Для титрів, що розміщуються всередині фільму, можна перетягувати як ліву, так і праву межу.

В **Області монтажу** для більшості об'єктів можна змінювати послідовність їх перегляду шляхом перетягування умовних позначень вставлених об'єктів (прямокутників) у межах відповідної доріжки.

Результати монтажу слід переглянути у вікні програвача.

На **четвертому етапі монтажу** відеофільму, коли розміщення та тривалість показу всіх об'єктів визначена, фільм доповнюється звуковим супроводом диктора (якщо це передбачено сценарієм) і музичним супроводом.

Звуковий супровід диктора може бути записано у відповідний файл з використанням програми **Звукозапис**, а потім вставлено до фільму описаним вище способом.

Для запису звукового супроводу диктора з використанням можливостей самої програми **Windows Movie Maker** необхідно вибрати точку на шкалі часу і виконати **Сервіс** ⇒ **Шкала времени комментария** (рос. *шкала времени комментария* – шкала часу коментарю) або вибрати кнопку  у лівій частині **Області монтажу**.

На панелі **Шкала времени комментария** для детальних налаштувань запису слід вибрати гіперпосилання **Показать дополнительные параметры** (рос. *показать дополнительные параметры* – показати додаткові параметри). На розширеній панелі (рис. 4.29) користувач може встановити позначку прапорця **Ограничить комментарий до размера свободного пространства на звуковой музыкальной дорожке** (рос. *ограничить комментарий до размера свободного пространства на звуковой музыкальной дорожке* – обмежити коментар до розміру вільного місця на звуковій музичній доріжці), якщо необхідно, щоб звуковий супровід не накладався на інший звуковий об'єкт, наприклад звук з відеофрагмента. При цьому програма автоматично припинить запис, якщо звукові фрагменти почнуть накладатися.

Встановлення позначки прапорця **Выключить динамики** (рос. *выключить динамики* – вимкнути динаміки) дає змогу уникнути можливого спотворення звуку від накладання звукових фрагментів. Слід також випробувати рівень сигналу від мікрофона та відрегулювати його, використовуючи повзунок **Уровень входа** (рос. *уровень входа* – рівень входу).

Для початку запису коментарю необхідно вибрати кнопку **Запустить** (рос. *запустить* – запустити), а для призупинення чи завершення запису – кнопку **Остановить** (рос. *остановить* – зупинити). Для включення записаного коментарю до фільму слід вибрати гіперпосилання **Готово**.

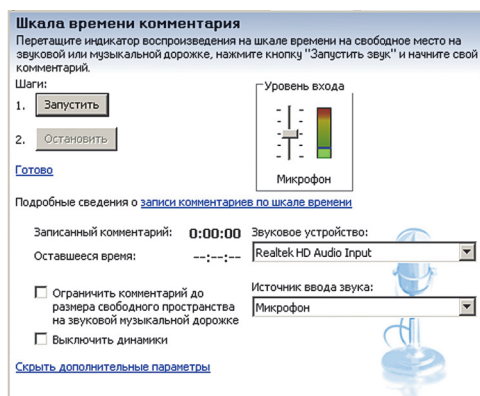


Рис. 4.29. Панель **Шкала времени комментария** з додатковими параметрами налаштування



Рис. 4.30. Вікно
Уровень звука

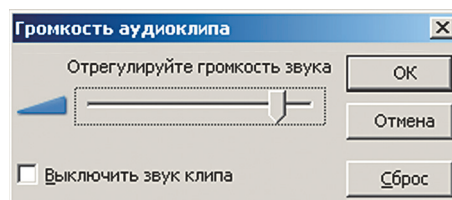


Рис. 4.31. Вікно
Громкость аудиоклипа

Створений коментар буде вставлено на звукову або музичну доріжку.



Далі користувачу слід відредагувати та вирівняти рівні звуку для різних фрагментів. Для цього в програмі є кілька інструментів. Зміна рівня звуку одночасно всіх звукових фрагментів фільму виконується з використанням діалогового вікна **Уровень звука** (рос. *уровень звука* – рівень звуку) (рис. 4.30). Для його відкриття слід виконати **Сервис** ⇒ **Уровень звука**.

Для зміни рівня звуку певного об'єкта слід вибрати його в **Області монтажу** і виконати **Клип** ⇒ **Аудио** ⇒ **Громкость** (рос. *громкость* – гучність) або вибрати команду **Громкость** у контекстному меню об'єкта. У вікні **Громкость аудиоклипа** (рис. 4.31) встановити повзунком потрібний рівень звуку об'єкта.

Для повного вимкнення звуку фрагмента (наприклад, звуку відеофрагмента) необхідно вибрати фрагмент в **Області монтажу** і виконати **Клип** ⇒ **Аудио** ⇒ **Выключить** (рос. *выключить* – вимкнути) або вибрати команду **Выключить** у контекстному меню об'єкта.

Після редагування та синхронізації звукового та відеоряду слід переглянути результати монтажу у вікні програвача.

П'ятий етап монтажу відеофільму полягає в налаштуванні відео-ефектів і ефектів переходу між кадрами фільму. Для цього краще перейти до режиму розкадровки в **Області монтажу**.

Для вставлення відео ефекту достатньо на панелі **Сборники** вибрати папку **Видеоэффекты** та перетягнути потрібний ефект на кнопку  в нижньому лівому куті потрібного кадру. До кожного кадру можна встановити кілька відео ефектів. Для попереднього перегляду результату застосування ефекту слід скористатися програвачем. Для видалення ефекту слід вибрати кнопку  на потрібному кадрі та натиснути клавішу **Delete**.

Для вставлення ефекту переходу між кадрами необхідно на панелі **Сборники** вибрати папку **Видеопереходы** та перетягнути потрібний ефект на прямокутну область між потрібними кадрами. Результат застосування ефекту слід переглянути з використанням програвача. Для видалення ефекту переходу слід вибрати прямокутну область між потрібними кадрами і натиснути клавішу **Delete**.

Після застосування всіх ефектів слід перевірити результати монтажу з використанням програвача програми.

Останній етап роботи над відеофільмом – **збереження відеофільму у файлі**. Для цього потрібно:

1. Виконати **Файл** ⇒ **Сохранить файл фильма** (рос. *сохранить файл фильма* – зберегти файл фільму);

2. Указати дію над файлом у першому вікні **Мастер сохранения фильма**:

- збереження в зовнішній пам'яті комп'ютера користувача;
- запис на оптичний диск;
- відправлення електронною поштою;
- розміщення на веб-сайті;
- запис у пам'ять цифрової відеокамери.

3. Вибрати кнопку **Далі**.

4. Якщо обрано збереження в зовнішній пам'яті комп'ютера, то слід у наступному вікні вказати ім'я файлу та папку, в яку він буде збережений, та вибрати кнопку **Далі**.

5. Установити значення параметрів якості відеофільму та вибрати кнопку **Далі**.

6. Дочекатися завершення кодування фільму з обраними параметрами якості та запису в файл.

Файл буде збережено у форматі WMV, і в подальшому його можна використовувати як звичайний відеофайл: копіювати, відтворювати з використанням відеопрогравачів, конвертувати, вставляти в слайдові презентації тощо.



Перевірте себе

- 1°. Що таке відеофільм? Чим він відрізняється від кінофільму?
- 2°. Поясніть, для чого розробляються сценарії.
- 3°. Опишіть етапи створення відеофільму. Які операції виконуються на кожному з них?
- 4°. Поясніть, які дії слід виконати в ході підготовки сценарію відеокліпу про вашу сім'ю.
- 5°. Що таке сценарний кадр? Які дані вказуються в сценарному кадрі?
- 6°. Що таке монтаж відеофільму?
- 7°. Опишіть, які підготовчі дії слід виконати в програмі **Windows Movie Maker** перед початком монтажу відеофільму.
- 8°. Для чого в програмі **Windows Movie Maker** використовуються папки **Сборники**, **Видеопереходы**, **Видеоэффекты**?
- 9°. Які режими відображення **Області монтажу** є в програмі **Windows Movie Maker**? Як їх перемикають?
- 10°. Поясніть призначення доріжок **Області монтажу** в режимі відображення шкали часу.
- 11°. Що таке файл проекту в програмі **Windows Movie Maker**? Які дані він містить?
- 12°. Опишіть послідовність дій для включення відео- і графічних об'єктів у відеофільм.
- 13°. Опишіть послідовність дій для включення звукових об'єктів у відеофільм.
- 14°. Як вимкнути відтворення звукового фрагмента?
- 15°. Поясніть способи зміни тривалості відтворення графічних, відео-, звукових і текстових об'єктів.
- 16°. Опишіть послідовність дій під час вставлення текстових об'єктів в **Область монтажу**. Які є варіанти їхнього розміщення відносно інших об'єктів відеофільму?
- 17°. Як змінити формат текстових об'єктів у програмі **Windows Movie Maker**?
- 18°. Опишіть порядок додавання дикторського звукового супроводу до фільму.

- 19*. Опишіть послідовність збереження файлу відеофільму у програмі **Windows Movie Maker**.
- 20*. Порівняйте можливості слайдових презентацій, створених у **PowerPoint 2007**, і потокових, створених у **Windows Movie Maker**, на прикладі роботи з титульними сторінками.
- 21*. Чи можна створити в **PowerPoint 2007** презентацію, за своїми властивостями схожу з відеофільмом (потоковою презентацією)? Відповідь обґрунтуйте.



Виконайте завдання



- 1°. Створіть відеофільм «Птахи України» за сценарієм, який описано в цьому пункті. Матеріали для включення до відеофільму розміщено в папках **Відео**, **Звук**, **Фото** папки **Тема 4\Завдання 4.3**. Збережіть відеофільм у власній папці у файлі з іменем **вправа 4.3.1.wmv**.
- 2°. Складіть сценарій і створіть за ним у **Windows Movie Maker** потокову презентацію про Лесю Українку до курсу української літератури 10-го класу. У презентації використайте фрагмент з відеофільму «Великі українці. Лесь Українка» і фотографії Лесі Українки та пам'ятників їй з папки **Тема 4\Завдання 4.3\Українка**. Підготуйте і включіть до відеофільму титульний слайд та титри супроводу. Самостійно доберіть музику, яка повинна звучати протягом усього відеофільму. Збережіть відеофільм у власній папці у файлі з іменем **вправа 4.3.2.wmv**.
- 3°. Підготуйте у **Windows Movie Maker** потокову презентацію про керівників української держави початку ХХ ст., використовуючи матеріал підручника з історії України 10-го класу та матеріали, розміщені в папках **Тема 4\Завдання 4.3\Скоропадський** та **Тема 4\Завдання 4.3\Петлюра**. Дикторський текст і музичний супровід доберіть самостійно. Збережіть відеофільм у власній папці у файлі з іменем **вправа 4.3.3.wmv**.



Практична робота № 15. Створення відеокліпу (потокової презентації)

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Створіть, використовуючи програму **Windows Movie Maker**, відеокліп (потокову презентацію) з питань охорони навколишнього середовища.
2. Розробіть сценарій презентації, передбачивши:
 - Кадр 1. Титульна сторінка:
 - заголовок – *Збережемо природу рідного краю*, підзаголовок – *Навчальна програма*;
 - колір тла – *жовтий*, колір літер – *темно-зелений*;
 - анімація – *відео, в тексті*;
 - тривалість показу – *4,40 с*;
 - звук – *відсутній*;
 - ефект переходу на наступний кадр – *круги*.
 - Кадр 2. Фотографія сміття:
 - файл – *сміття01.jpg*;
 - титри на слайді – *Що треба зробити, щоб такого не було?*
 - початок показу титрів – *через 2 с після початку показу фотографії*;
 - тривалість показу – *3 с*;
 - анімація – *Клас!*;
 - звук – файл *FINCL_02.MID*; тривалість звучання – *5 с*;
 - ефект переходу на наступний кадр – *вихор*.

- Відобразіть у наступних кадрах основні проблеми забруднення навколишнього середовища і шляхи боротьби за чисте довкілля. Загальна тривалість відеокліпу – від 3 до 5 хв.
- 3. Змонтуйте відеокліп, включивши до нього потрібні відео-, звукові, графічні та текстові об'єкти з папки **Тема 4\Завдання 4.3\Екологія** або з інших джерел.
- 4. За потреби відредагуйте об'єкти та синхронізуйте відео- й аудіоряди.
- 5. Збережіть презентацію у своїй папці у файл з іменем **Практична робота 15.wmv**.

4.4. Основи створення комп'ютерних публікацій. Програма Microsoft Office Publisher 2007



1. Що ви знаєте про поліграфію? Які види друкованих видань розробляють поліграфічні підприємства?
2. Коли і де з'явилися перші друковані видання?
3. Як класифікують програми опрацювання тексту? Що таке видавничі системи?
4. Що таке шаблон? Як використовуються шаблони у **Word 2007** та **PowerPoint 2007**?

Публікації та комп'ютерні публікації, їх призначення та види

Публікаціями (лат. *publico* – оголошувати публічно) називають процес видання якого-небудь твору. Сам виданий (оприлюднений) твір теж називають публікацією.

Залежно від способу оприлюднення документа розрізняють **друковані** та **електронні публікації**. До друкованих публікацій належать книжки, брошури, журнали, газети, бюлетені, буклети, листівки тощо. До електронних публікацій належать документи, розміщені в Інтернеті, як правило на веб-сайтах, передані по радіо або телебаченню тощо.

Тривалий час друковані публікації створювались виключно в типографії. І цим займалася ціла галузь виробництва – поліграфія (грец. *πολύ-* – багато і *γραφειν* – писати). А підготовкою матеріалів до друку займалися видавництва. При промисловому багатотиражному виробництві так є і сьогодні: видавництва готують макети книжок, журналів, газет і т. п., а підприємства поліграфії – друкують і за потреби зшивають їх.

Разом з тим із широким розповсюдженням персональних комп'ютерів і високоякісних лазерних і струменевих принтерів, копіювальних апаратів набули розповсюдження так звані **настільні видавничі системи**. Ці системи мають апаратну і програмну складові. До апаратної складової належать персональний комп'ютер, пристрої друку (принтер, копіювальний апарат) і пристрої введення даних (сканери, графічні планшети), а до програмної – спеціальна програма підготовки макета публікації. Усе це може розміститися на столі користувача, і тому ці системи отримали назву **desktop publishing** (англ. *desktop publishing* – настільні публікації) або настільні видавничі системи.

Як ви вже знаєте з курсу інформатики 9-го класу, програми, призначені для створення макетів поліграфічних видань і підготовки їх до друку, також називаються **видавничими системами**. До таких програм належать **QuarkXPress**, **Corel Ventura**, **Adobe PageMaker**, **Adobe InDesign**

тощо. Це професійні програми, які використовують, як правило, для підготовки до друку великих за обсягом публікацій масовим (від кількох сотень до кількох сотень тисяч) тиражем. Як видавничі системи для малих підприємств і організацій, навчальних закладів, а також для створення публікацій вдома використовують текстові процесори і спеціалізовані програми настільних видавничих систем зі спрощеним інтерфейсом і набором функцій – **Microsoft Publisher, Page Plus, Avery DesignPro, Paraben's Label Builder, Publish-It!** тощо.

На відміну від текстових процесорів видавничі системи орієнтовані не на введення та редагування тексту та інших об'єктів, а на верстку.



Верстка – це процес компонування текстових і графічних об'єктів для створення сторінок (полос) видання відповідно до принципів дизайну і технічних вимог.

За призначенням розрізняють *книжково-журнальну верстку, газетну і акцидентну* (лат. *accidentia* – випадок, випадковість). Акцидентна верстка застосовується для верстки окремих, як правило, невеликих за кількістю сторінок та тиражем об'єктів: оголошень, бланків, афіш, буклетів, проспектів, візитівок тощо.

Публікації, що здійснені з використанням комп'ютерних технологій, ще називають *комп'ютерними публікаціями*.

Основні можливості настільних видавничих систем

Настільні видавничі системи забезпечують:

- автоматизацію процесів розміщення блоків тексту та графічних зображень відповідно до типу публікації (*макетування*);
- імпортування фрагментів тексту та графічних зображень, створених у різних прикладних програмах;
- опрацювання фрагментів тексту, використовуючи стилі та розширені можливості зі встановлення відстані між символами та між рядками;
- автоматизацію створення списків змісту та посилань;
- використання розширеного набору шрифтів і їх використання відповідно до вимог поліграфії;
- відтворення публікації на екрані монітора з урахуванням особливостей пристрою, на якому буде здійснено друк публікації;
- збереження файлів публікації у форматах, що використовуються в поліграфії;
- можливість друку публікацій кожним кольором окремо (*кольороподіл*);
- забезпечення високоякісного друку на пристроях різного типу.

Звичайно, що набір можливостей може змінюватися залежно від призначення програми, поліграфічної бази, на яку зорієнтована програма.

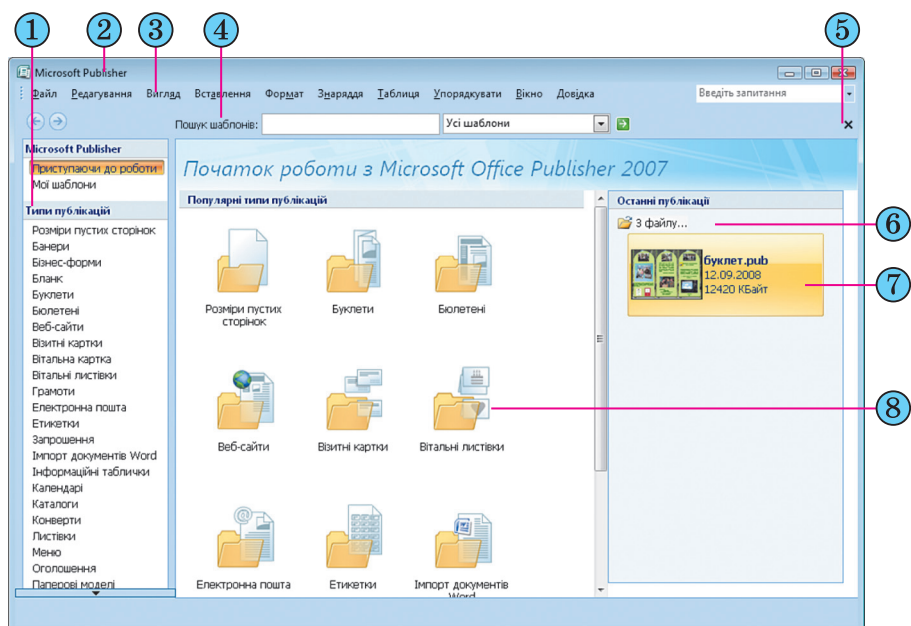
Загальна характеристика видавничої системи Microsoft Office Publisher 2007

Настільна видавнича система **Microsoft Office Publisher 2007** (далі **Publisher 2007**), як уже зазначалося, призначена для створення комп'ютерних публікацій малими тиражами на невеликих підприємствах та організаціях, а також для особистих потреб користувача.

Програма **Publisher 2007** має спрощену порівняно з професійними видавничими системами функціональність. Вона забезпечує акцидентну верстку публікацій на основі різноманітних шаблонів змісту і може працювати в двох основних режимах – створення та редагування **друкованої та електронної публікації**.

Стандартним форматом файлів для збереження публікацій, підготовлених з використанням **Publisher 2007**, є формат **PUB**. Слід зауважити, що збереження публікацій у файлах попередніх версій програми **Publisher** (1998 та 2000 року) може призвести до того, що ряд функцій, які працюють у версії програми 2007 року, можуть не працювати в програмах попередніх версій.

Основним об'єктом у **Publisher 2007** є **публікація** як сукупність сторінок. На сторінках друкованих публікацій можуть розміщуватися текстові та графічні об'єкти, а на сторінках електронних публікацій ще і звукові, відеооб'єкти та гіперпосилання. Властивості об'єктів публікацій практично не відрізняються від властивостей подібних об'єктів, розглянутих у **Word 2007** та **PowerPoint 2007**. Над кожним з об'єктів видавнича система може виконувати певний набір операцій з редагування та форматування.




- | | |
|---|---|
| ① Список Типи публікацій | ⑤ Кнопка закриття каталогу |
| ② Рядок заголовка вікна програми | ⑥ Кнопка 3 файли |
| ③ Рядок меню | ⑦ Піктограма останньої публікації |
| ④ Панель інструментів | ⑧ Панель відображення вмісту папок типів публікацій |

Рис. 4.32. Початкове вікно програми **Publisher 2007**

Запуск Publisher 2007 та інтерфейс вікна програми

Програму **Publisher 2007** можна запустити на виконання кількома способами. Наведемо найпоширеніші з них:

- виконати **Пуск** ⇒ **Усі програми** ⇒ **Microsoft Office** ⇒ **Microsoft Office Publisher 2007**;
- використати ярлик програми на **Робочому столі** або на **Панелі швидкого запуску**;
- двічі клацнути на значку файлу публікації **Publisher**, наприклад такому – .

У разі якщо вибрано один з перших двох варіантів запуску програми, на екрані з'явиться початкове вікно програми, розділене на три частини (рис. 4.32).

У лівій частині вікна розміщено список типів публікацій (рис. 4.32, 1), які можна створити в **Publisher 2007**. За вибору в цьому списку одного з об'єктів, наприклад **Буклети**, у центральній частині відображаються ескізи шаблонів публікації цього типу. При цьому в лівій частині замість панелі **Останні публікації** відобразиться панель з елементами керування, використовуючи які встановлюють значення властивостей обраного шаблону публікації.

Пошук потрібних шаблонів публікацій можна здійснити, використовуючи поле **Пошук шаблонів** і список, у якому можна вибрати місце пошуку: *на цьому комп'ютері, на сайті Microsoft Office Online* або в обох цих місцях (**Усі шаблони**).

Створення нової публікації

Створення нової публікації можна розпочати з початкового вікна програми, яке описано вище.

Після вибору типу публікації та одного із шаблонів на панелі настроювання значень властивостей (рис. 4.33) можна змінити колірну схему, схему шрифтів (або залишити ці властивості зі значеннями із шаблону за замовчуванням), вказати службові відомості (дані про організацію або особу), які будуть автоматично вставлені до шаблонів публікацій, та обрати форму для здійснення реєстрації, проведення опитування тощо. Після вибору кнопки **Створити** програма перейде в режим редагування та форматування публікації.

Можна пропустити ці дії, для чого в початковому вікні програми вибрати кнопку закриття каталогу (рис. 4.32, 5). Початкове вікно буде закрито, і відкриється основне вікно програми (рис. 4.34), яке має дещо інший інтерфейс, ніж розглянуті раніше програми пакета **Microsoft Office 2007**.

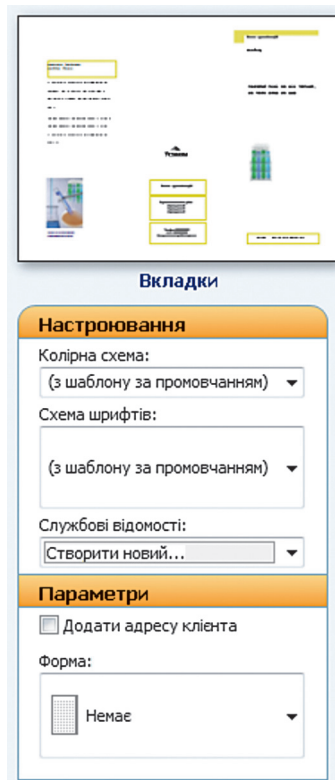
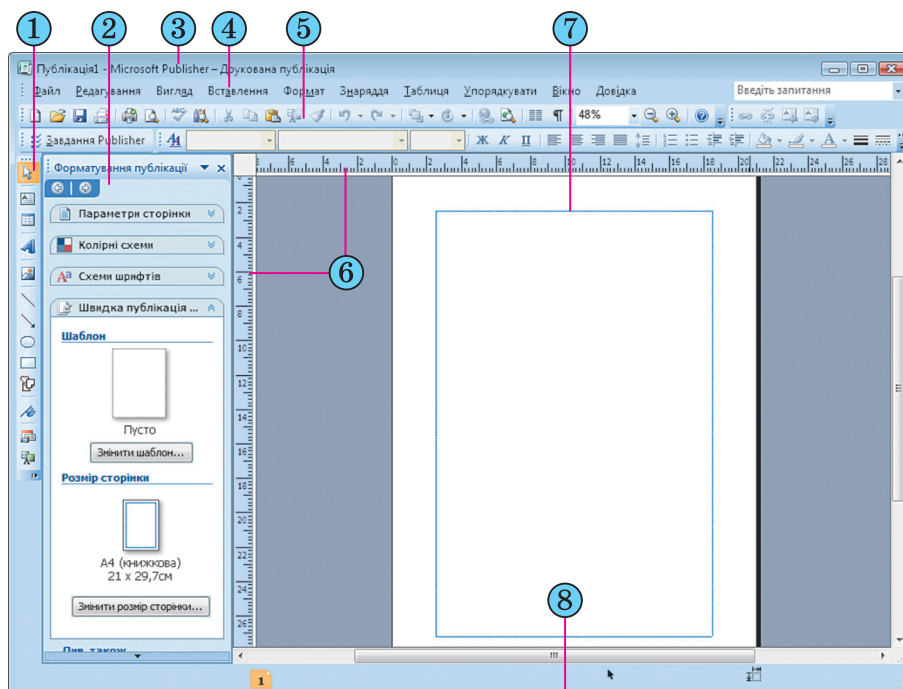


Рис. 4.33. Панель встановлення значень властивостей обраного шаблону



- | | |
|--------------------------------------|-----------------------|
| ① Панель інструментів Об'єкти | ⑤ Панелі інструментів |
| ② Область завдань | ⑥ Лінійки |
| ③ Рядок заголовка вікна | ⑦ Робоча область |
| ④ Рядок меню | ⑧ Рядок стану |

Рис. 4.34. Основне вікно програми **Publisher 2007**

У основному вікні програми, використовуючи елементи керування панелі інструментів **Об'єкти**, панелі **Форматування публікації** **Області задач** та інших панелей, користувач може самостійно розробити макет майбутньої публікації. Він також може повернутися до створення публікації на основі шаблонів, виконавши **Файл** ⇒ **Створити** або вибравши у списку на панелі **Форматування публікації** **Області задач** кнопку **Змінити шаблон**.

Як і в **PowerPoint 2007**, у **Publisher 2007** можна створювати публікації на основі вже існуючих. Для цього в початковому вікні програми слід вибрати кнопку **З файлу** і вказати файл, на основі якого буде створено нову публікацію. Цю ж дію можна виконати в основному вікні: **Файл** ⇒ **Відкрити** або **Файл** ⇒ **Створити** ⇒ **З файлу**.

Вставлення та редагування об'єктів


Вставлення більшості об'єктів здійснюється з використанням кнопок панелі інструментів **Об'єкти** (табл. 4.6). Операції з більшістю об'єктів аналогічні подібним операціям у **PowerPoint 2007**. Як і в **PowerPoint 2007**, текстові об'єкти вставляються тільки у вигляді написів.

Таблиця 4.6. Кнопки панелі інструментів Об'єкти та їх призначення

Кнопка	Назва	Призначення	Кнопка	Назва	Призначення
	Вибір об'єктів	Для вибору об'єктів публікації		Напис	Для вставлення текстового об'єкта (напису)
	Додати таблицю	Для вставлення таблиці		Додати об'єкт WordArt	Для вставлення об'єкта WordArt
	Рисунок	Для вставлення графічного об'єкта		Лінія	Для вставлення лінії
	Стрілка	Для вставлення стрілки		Овал	Для вставлення овалу
	Об'єкт бібліотеки макетів	Для відкриття вікна бібліотеки макетів публікацій		Автофігури	Для вставлення однієї з фігур бібліотеки автофігур
	Елемент з бібліотеки вмісту	Для відкриття в Області завдань панелі Бібліотека вмісту		Закладка	Для вставлення закладки для швидкого пошуку певного місця в публікації
	Прямокутник	Для вставлення прямокутника			

Розглянемо послідовність використання шаблонів і вставлення об'єктів на прикладі створення листа молодіжної екологічної організації. Для початку роботи слід виконати такий алгоритм:

1. Запустити на виконання програму **Publisher 2007**.
2. Вибрати в списку типів шаблонів **Бланк**.
3. Вибрати потрібний шаблон (наприклад, шаблон **Кольорові смуги** з групи **Сучасні макети**).
4. Вибрати кнопку **Створити**.

Буде створено макет публікації з місцями для вставлення тексту і графічних об'єктів. У верхній правій частині цього макета публікації розміщено поруч кілька об'єктів (рис. 4.35). Слід навести вказівник на один з об'єктів і вибрати кнопку , що з'являється біля об'єкта. У списку, що відкрився, слід вибрати команду **Редагувати** і у вікні **Редагування набору службових відомостей** увести дані організації, наприклад такі:

- Назва організації: *Екологічна молодіжна організація «Чисті роси»*;
- Адреса: *вул. Київська, 24, м. Житомир*;


Адреса основного місця роботи.	Телефон: 555-555-5555.	 Установа
Адреса_рядок 2.	Факс: 555-555-5555.	
Адреса_рядок 3.	Електронна пошта: xyz@example.com	
Адреса_рядок 4.		

Рис. 4.35. Об'єкти шаблону бланка організації

Рис. 4.36. Вікно **Редагування набору службових відомостей**

подальшому будуть автоматично вставлятися в шаблонах інших публікацій.

Для введення тексту звертання до адресата потрібно в правій верхній частині публікації, під емблемою, вставити напис і ввести текст, наприклад такого змісту: *Директору малого підприємства «Автохім»*. На основному полі листа слід також вставити напис і ввести текст, наприклад такого змісту:

Шановний пане директоре!

Протягом останнього місяця працівники малого підприємства «Автохім» постійно викидають сміття безпосередньо в зеленій зоні біля огорожі підприємства. Це погіршує екологічний стан нашого мікрорайону і є неприпустимим. Просимо негайно вжити заходів і прибрати це звалище.

З надією на розуміння та спільну роботу зі збереження природи рідного краю члени молодіжної екологічної організації «Чисті роси».

Фотографію звалища додаємо.

Голова організації

Петренко Микола

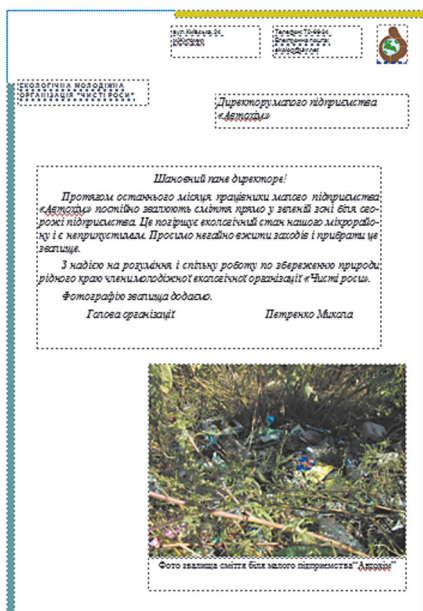



Рис. 4.37. Макет листа

- Телефон, факс і електронна пошта: *Телефон: 72-66-34, Електронна пошта: ekolog@ukr.net;*

- Емблема: файл *Чисті роси.WMF*.
Для зміни малюнка емблеми слід вибрати кнопку **Змінити** під полем **Емблема** і вибрати потрібний файл, наприклад **Тема 4\Завдання 4.4\Чисті роси.WMF**.

Після завершення редагування даних вікно повинно виглядати так, як це подано на рисунку 4.36. Далі слід вибрати кнопку **Зберегти** і в наступному вікні вибрати кнопку **Оновити публікацію**.

Службові відомості збережуться і в

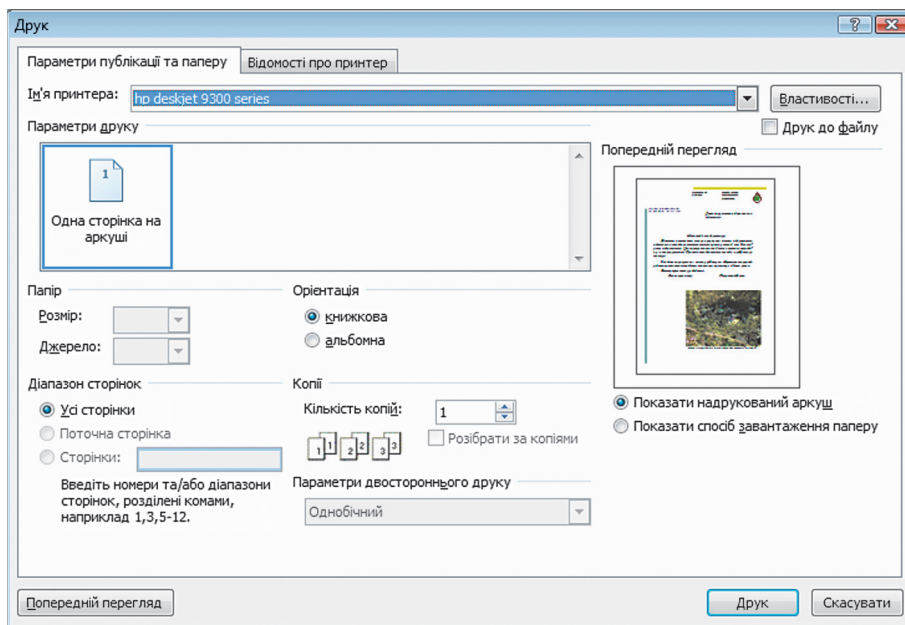
Нижче тексту листа слід вставити фотографію звалища сміття, наприклад з файлу **Тема 4\Завдання 4.4\Сміття.jpg**. Для цього потрібно на панелі інструментів **Об'єкти** вибрати кнопку **Рисунки** , у списку, що відкриється, вибрати команду **Вставити із файлу** і виділити прямокутну область на полі публікації – місце розміщення

Групування , що з'являється біля виділених об'єктів.

Збереження та друк публікації

Налаштування параметрів і сам процес друку практично не відрізняються від аналогічних операцій у **Word 2007**. Для встановлення значень властивостей друку можна попередньо відкрити вікно **Налаштування друку** (**Файл** \Rightarrow **Налаштування друку**). Такі самі налаштування можна встановити й у вікні **Друк** (**Файл** \Rightarrow **Друк**) (рис. 4.38).

Враховуючи, що, як і **PowerPoint 2007**, програма **Publisher 2007** має засоби зв'язування об'єктів, які розміщені в окремих файлах, при перенесенні на інші комп'ютери ці об'єкти можуть не відтворюватися. Для коректного перенесення публікації на інші комп'ютери для подальшого перегляду або друку використовується засіб упаковки публікації. Для




225

його використання слід виконати **Файл** ⇒ **Запакувати** ⇒ **Для іншого комп'ютера/Для друку**. В обох випадках відкриється вікно **Майстер пакування**. Послідовно вказуючи значення властивостей пакування (місце збереження, об'єкти, які слід включити і зв'язати) на кожному кроці роботи майстра для пакування, користувач створює архівний файл формату **ZIP**. Відмінність у пакуванні для іншого комп'ютера і для друку полягає в тому, що під час пакування для іншого комп'ютера не пропонується включати до архівного файлу шрифти і файли зв'язаних об'єктів.





Перевірте себе

- 1°. Що таке публікація?
- 2°. Що таке друкована публікація? Наведіть приклади друкованих публікацій.
- 3°. Що таке електронна публікація? Наведіть приклади електронних публікацій.
- 4°. Яка роль у створенні публікацій відводиться підприємствам поліграфії; видавництвам?
- 5°. Що таке настільна видавнича система?
- 6°. Що входить до складу настільної видавничої системи? Для чого призначена кожна складова?
- 7°. Що таке верстка?
- 8°. Які існують види верстки за призначенням? Де вони використовуються?
- 9°. Що таке комп'ютерна публікація? Наведіть приклади комп'ютерних публікацій.
- 10°. Які існують види публікацій за конструкцією? Де вони застосовуються?
- 11°. Опишіть можливості настільних видавничих систем.
- 12°. Поясніть призначення програми **Publisher 2007**.
- 13°. Опишіть формати файлів, у яких можуть зберігатися публікації. Чим вони відрізняються один від одного?
- 14°. Опишіть інтерфейс початкового вікна програми **Publisher 2007**.
- 15°. Опишіть інтерфейс основного вікна програми **PowerPoint 2007**.
- 16°. Опишіть послідовність створення нової публікації в початковому вікні **Publisher 2007**.
- 17*. Порівняйте операції вставлення текстових і графічних об'єктів у **PowerPoint 2007** та **Publisher 2007**. Що в них спільного і чим вони відрізняються?
- 18°. Для чого використовується кнопка  біля об'єктів шаблону?
- 19°. Наведіть послідовність дій для збереження публікації у стандартному для **Publisher 2007** форматі.
- 20°. Як роздрукувати публікацію?
- 21°. Для чого використовується пакування публікацій? Чим відрізняється пакування **Для іншого комп'ютера** і **Для друку**?



Виконайте завдання

- 1°. Запустіть на виконання програму **Publisher 2007**. Використовуючи початкове вікно програми, створіть нову публікацію на основі шаблону **Оголошення – Маркер**. Установіть колірну схему **Бузок**. Збережіть публікацію у власній папці у файлі з іменем **вправа 4.4.1.pub**.
- 2°. Створіть у програмі **Publisher 2007** на основі шаблону **Грамота** нову публікацію для нагородження переможців першості вашої школи з настільного тенісу. Доберіть самостійно колірну схему, введіть дані про

- вашу школу, підписи директора і вчителя фізичного виховання. Збережіть публікацію у власній папці у файлі з іменем **вправа 4.4.2.pub**.
-  3*. Створіть у програмі **Publisher 2007** на основі шаблону **Меню Особливі страви** нову публікацію – меню святкової вечери до дня народження. Доберіть самостійно колірну схему, уведіть назви 5–6 страв та їх короткий опис. Збережіть публікацію у папці **Мої документи** у файлі з іменем **вправа 4.4.3.pub**.
-  4*. Створіть у програмі **Publisher 2007** на основі одного із шаблонів власну візитівку. Включіть до картки власне фото і дані про місце навчання. Роздрукуйте візитівки, розмістивши на одному аркуші 10 копій. Збережіть публікацію в папці **Мої документи** у файлі з іменем **вправа 4.4.4.pub**.
- 5*. Розробіть фірмовий бланк вашої школи на основі шаблону **Бланк Геометрична**. Включіть до бланка назву школи, її адресу і телефони, емблему (герб). Кольорову гаму доберіть відповідно до кольорової гами прапора школи або прапора регіону. Збережіть публікацію у власній папці у файлі з іменем **вправа 4.4.5.pub**.

4.5. Особливості роботи з текстовими та графічними об'єктами в Publisher 2007. Створення інформаційного бюлетеня та буклета



1. Що є основою дизайну?
2. Які вимоги до кольорового оформлення ви знаєте?
3. Поясніть, що таке стилі.
4. Як створити нову публікацію в **Publisher 2007**?

Створення інформаційного бюлетеня

Як уже зазначалося, у публікаціях можуть розміщуватися два типи об'єктів – текстові написи та графічні зображення. Розглянемо особливості роботи з ними під час створення публікацій у **Publisher 2007**. Зробимо це на прикладі створення інформаційного бюлетеня.



Бюлетень (франц. *bulletin* – записка, листок, документ) – це коротке офіційне повідомлення, збірка офіційних актів або періодичне видання, яке містить відомості з певного кола питань.

Особливості структури бюлетеня

У **Publisher 2007** під бюлетенем розуміють публікацію з однієї чи кількох сторінок, яка містить відомості з певного кола питань: наприклад, аналіз економічного чи фінансового стану підприємства за певний період, добірка нормативних актів з певного питання або за певний період, аналітична записка з перспектив розвитку галузі, збірка статей з певного питання тощо.

Бюлетень має титульну і заключну сторінку, між якими може знаходитися кілька внутрішніх сторінок. На кожній сторінці може бути кілька статей. Матеріал статей розміщується в кілька колонок. Кожна колонка – це напис, який у **Publisher 2007** ще називають текстовим полем. Текстові поля однієї статті зв'язані між собою. Статті можуть починатися на одній сторінці та закінчуватися на іншій.

Один з головних об'єктів публікації – **заголовок бюлетеня**. За своєю структурою заголовок бюлетеня схожий на газетний. Він містить текст заголовка і назву видавця, може містити девіз, емблему, для періодичних видань – номер і дату випуску.

Основне місце на титульній сторінці займає перша або головна стаття. У бюлетені вона відіграє роль своєрідної програмної статті, тобто статті, у якій розкрито цілі цього видання або описано проблеми, які в наступних статтях будуть обговорюватися, деталізуватися й уточнюватися.

Остання сторінка бюлетеня містить дані про редакторів і дизайнерів бюлетеня та деякі службові дані: тираж публікації, реквізити редакції та типографії тощо.

Створення бюлетеня на основі шаблону

Процес створення бюлетеня, як і розглянутий раніше процес створення презентації, розпочинається з розробки структури (макета) і дизайну даного виду публікації. Перше за все слід ознайомитись із шаблонами публікацій цього типу, з'ясувати тематику та перелік питань, які будуть висвітлюватися в публікації, добрати матеріали до публікації: назву бюлетеня, заголовки і тексти статей, ілюстрації до них, графічні елементи тощо. Далі слід перейти до безпосереднього створення бюлетеня.

Для створення бюлетеня необхідно в початковому вікні **Publisher 2007** у списку типів публікацій обрати **Бюлетень**. Потім вибрати шаблон оформлення та на відповідній панелі (рис. 4.39) уточнити значення властивостей публікації: вибрати кольірну схему та схему шрифтів, указати службові відомості. На цій самій панелі в полі **Розмір сторінки** встановлюється кількість аркушів публікації. Слід враховувати, що передбачається двосторонній друк, і, відповідно, якщо обрати один аркуш публікації, то матимемо дві сторінки бюлетеня, а якщо два аркуші – то чотири.

Після завершення початкового встановлення значень властивостей публікації та друку слід вибрати кнопку **Створити** і перейти в режим редагування та форматування шаблону бюлетеня.

У **Рядку стану** вікна програми відображаються ярлики сторінок публікації (рис. 4.40). Для перегляду потрібної сторінки публікації слід вибрати ярлик з її но-

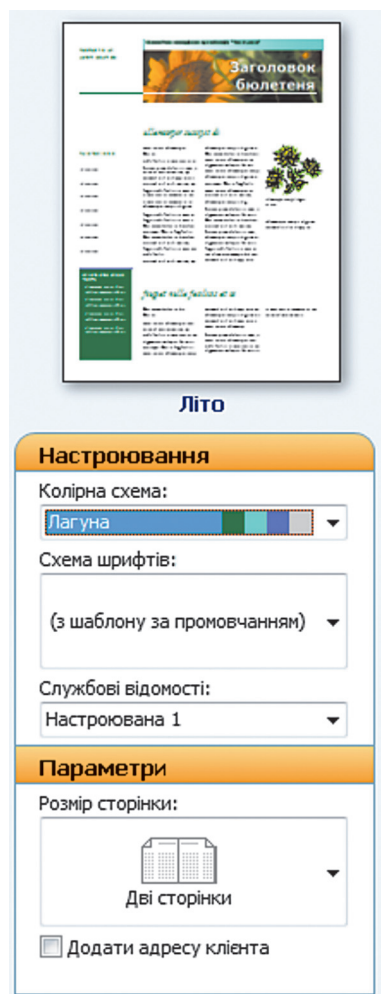




Рис. 4.39. Панель встановлення значень властивостей бюлетеня



Рис. 4.40. Ярлики сторінок публікації

мером. Внутрішні сторінки (наприклад, 2 і 3 на рис. 4.40) відображаються разом. Ярлик поточної сторінки має більш темний колір.

Налаштування параметрів сторінки

В області завдань **Форматування публікації** слід відкрити, вибравши кнопку  біля назви, список **Параметри сторінки** (рис. 4.41). У полі **Стовпці** можна встановити кількість стовпців тексту на поточній або на всіх сторінках тексту. Для цього слід навести вказівник на відповідну кнопку, вибрати кнопку відкриття списку (рис. 4.42) та вибрати команду, яка відповідає обраній області застосування. Вибір кнопки **Комбінація**  встановлює різну кількість стовпців для різних статей.

У списку **Колірні схеми** області завдань **Форматування публікації** можна змінити обрану колірну схему публікації, а у списку **Схема шрифтів** – вказати новий варіант комбінації шрифту для заголовків і основного шрифту.

У списку **Бюлетень** – параметри цієї самої області завдань, окрім зміни шаблону публікації, можна встановити і розміри сторінки публікації. Для цього слід вибрати кнопку **Змінити розмір сторінки** і у вікні **Параметри сторінки** обрати один із стандартних розмірів або встановити свій. У цьому самому вікні встановлюються і значення полів сторінки.

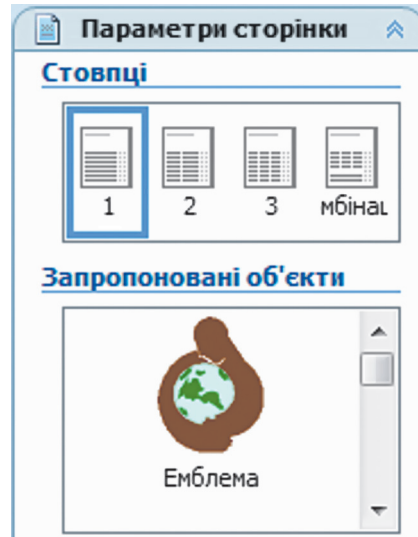


Рис. 4.41 Список **Параметри сторінки**

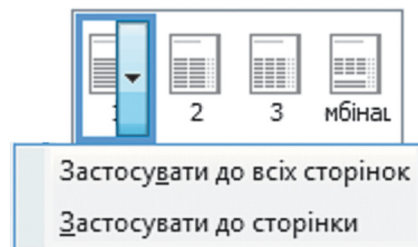


Рис. 4.42. Кнопки встановлення кількості стовпців тексту

Редагування заголовка, змісту і бічної панелі

Структура заголовка бюлетеня стандартна для всіх шаблонів. На рисунку 4.43 подано оформлення заголовка бюлетеня із шаблону *Літо*. А на рисунку 4.44 перероблений варіант заголовка цього шаблону для бюлетеня *Альманаху літературної творчості «Пролісок»*.



Рис. 4.43. Заголовок бюлетеня із шаблону **Літо**

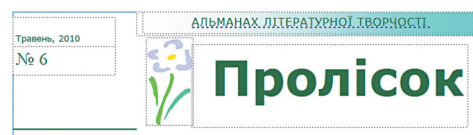


Рис. 4.44. Заголовок альманаху **Пролісок**

Для створення нового заголовка на основі існуючого шаблону слід увести у відповідні написи нові тексти, видалити існуючі малюнки та вставити нові.

На титульній сторінці бюлетеня, особливо якщо він має багатосторінкову структуру, вставляють заголовки внутрішніх статей, які складають **зміст публікації**. Шаблон для змісту (рис. 4.45), як правило, розміщується збоку, поряд з першою статтею бюлетеня, хоча може бути й інше розміщення залежно від структури шаблону бюлетеня. Користувач може змінити кількість рядків, видалити зайві або додати нові залежно від кількості статей в публікації. Цей об'єкт є таблицею, і його редагування та форматування здійснюються з використанням команд меню **Таблиця** або контекстного меню об'єкта. Редагування та форматування тексту в змісті відбувається відповідно до аналогічних операцій у **Word 2007**.

У цьому виданні:	
Внутрішня стаття	2
Внутрішня стаття	2
Внутрішня стаття	2


Рис. 4.45. Шаблон таблиці для розміщення заголовків статей

Наші автори:
• Учениця 10-Б класу Шаповал Ольга;
• Учениця 10-А класу Корольова Надія;
• Учениця 11-В класу

Рис. 4.46. Бічна панель з переліком авторів поезій




Під таблицею заголовків статей на титульній сторінці бюлетеня зі структурою шаблону *Літо* розміщено **Бічну панель** (рис. 4.46) зі списком – список анонсів, який має заголовок *Важливі деталі*. У цьому списку слід увести короткі фрази, які б вказували на своєрідність викладеного в бюлетені матеріалу, на цікаві факти, які б зацікавили і привернули увагу читача. Наприклад, для літературного альманаху це можуть бути повідомлення про літераторів, твори яких уперше з'явилися на його сторінках, або про твори вже відомих і популярних авторів. Для введення даних потрібно вибрати відповідні шаблонні написи в списку і ввести нові.


Редагування статей бюлетеня

Основною відмінністю розміщення тексту на сторінках публікації від розміщення на слайді презентації або в текстовому документі є можливість автоматичного продовження тексту з одного стовпця (напису) статті в інший. Цю особливість називають «перетіканням тексту». Перетікання можливо й у стовпці на іншій сторінці. Для підказки, у якому стовпці продовжується текст з вибраного стовпця, використовується спеціальна кнопка – **Перейти до наступного текстового поля** . Її вибір приводить до виділення напису, у якому продовжується або починається текст.

У шаблонах уже встановлено автоматичне перетікання тексту між стовпцями статті. Якщо ж користувач самостійно створює макет публі-

кації, то йому потрібно вміти самостійно встановлювати і відмінати перетікання тексту з одного текстового поля в інше. Ця операція називається **зв'язуванням текстових полів**.

Для зв'язування текстових полів необхідно вибрати початкове текстове поле і на панелі інструментів **Зв'язати написи** вибрати кнопку **Створити зв'язок із написом** . Вказівник набуде вигляду чашки зі стрілкою вниз – . Його слід підвести до текстового поля, з яким планується встановити зв'язок (курсор повинен набутися вигляду нахиленої чашки, з якої виливаються літери ). Поле, з яким створюється зв'язок, повинно бути порожнім.

Для розірвання зв'язку необхідно вибрати потрібне текстове поле і на панелі інструментів **Зв'язати написи** вибрати кнопку **Розірвати зв'язок з попереднім** . Незважаючи на таку назву кнопки, розрив відбувається з наступним текстовим полем.

Тексти статей, як правило, готуються заздалегідь в одному з текстових процесорів, наприклад **Word 2007**, і вставляються в текстові поля публікацій з використанням **Буфера обміну**. За вибору текстового поля шаблону виділяється увесь текст статті, розміщений в кількох стовпцях. І після виконання команди **Вставити новий текст** замінює шаблонний. Якщо текст не вміщується в одне чи кілька зв'язаних текстових полів, відведених для статті, то програма введе на екран діалогове вікно (рис. 4.47) з відповідним повідомленням і запитом на виконання **авторозливання**. Підтвердження виконання дії авторозливання приведе до зв'язування поточних текстових полів з наступними на сторінці текстовими полями і розміщення в них тексту, що вставляється.

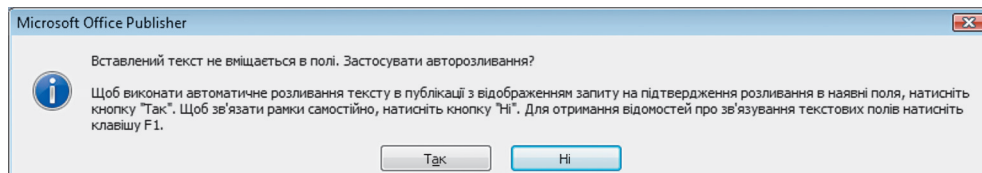





Рис. 4.47. Діалогове вікно, що сповіщає про переповнення текстового поля

Якщо ж користувач відмовиться від виконання операції авторозливання, то текст буде вставлено в зв'язані текстові поля поточної статті, але під останнім полем буде виведено позначку **Текст в області переповнення** . Це означає, що частина тексту не відображається на екрані, але залишається в пам'яті і її можна відтворити.

Для того щоб у текстових полях статті відображався весь текст, користувачу слід виконати одну з дій:

- відредагувати текст, видаливши несуттєві його фрагменти;
- відформатувати текст, змінивши значення властивостей символів або абзаців;
- змінити розміри текстових полів;
- зв'язати тестові поля статті з додатковими на цій або іншій сторінці.

Фотографії та малюнки в публікаціях підписуються. Підписи створюються з використанням кнопки **Напис**  панелі інструментів

Об'єкти. Після створення підпису його слід згрупувати з фотографією або малюнком: виділити ці об'єкти і вибрати кнопку **Групування** , що з'являється біля виділених об'єктів. Якщо використовуються шаблони графічних об'єктів, то вони вже мають підписи.

Редагування та форматування графічних об'єктів у публікаціях **Publisher 2007** здійснюються аналогічно до подібних операцій у **Word 2007** і **PowerPoint 2007**. Тільки якщо графічні об'єкти з файлів за замовчуванням у **Word 2007** вставляються з обтіканням **У тексті**, а в **PowerPoint 2007** – з обтіканням **Перед текстом**, то в **Publisher 2007** – з обтіканням **Навколо рамки**.

Під час створення публікацій слід дотримуватися вимог до збереження авторських прав. Усі статті та графічні матеріали повинні мати посилання на авторів. При використанні матеріалів з інших друкованих чи електронних публікацій слід отримати дозвіл на це їх авторів.

Створення буклета

Буклет (франц. *bouclette* – кільце) – публікація, виготовлена на одному аркуші та складена згинанням у кілька сторінок так, що її можна переглядати, не розрізаючи сторінок, а розкриваючи як ширму. У формі буклета виготовляються путівники, проспекти, програми концертів і вистав, реклами продукції та послуг тощо. Як правило, буклет складається з одного аркуша формату А4, розміщеного в альбомній орієнтації та розділеного на три рівні частини. По межах цих частин буклет згинається. Друк здійснюється, як правило, з обох сторін аркуша.

Створення буклета в **Publisher 2007**, вставлення текстових і графічних об'єктів не відрізняються від аналогічних операцій з бюлетенем. Для створення буклета слід в початковому вікні програми (або **Файл** ⇒ **Створити**) серед типів публікацій вибрати **Буклет**, а потім конкретний шаблон.

Основні відмінності у роботі з буклетом визначаються його призначенням і формою.

Основне призначення буклета – реклама навчального закладу, підприємства, організації, особи, нового товару, нової акції тощо. А враховуючи те, що всі відомості повинні бути розміщені на одному аркуші, зміст буклета повинен містити невеликі за обсягом фрагменти тексту, які точно передають основні відомості про об'єкт, а також фотографії та малюнки, що ілюструють ці відомості. Фотографії та малюнки повинні бути високої якості і містити незначну кількість об'єктів, які чітко розрізняються.

Як правило, буклет має титульну і заключну сторінки. На титульній



Рис. 4.48. Приклади титульних сторінок буклетів

сторінці, вказується заголовок буклета – його основна тема і фотографія або малюнок із цієї теми. Можуть вказуватися дані про того, хто випустив цей буклет. На рисунку 4.48 подано зразки оформлення титульної сторінки буклета.

На заключній сторінці зазначаються контактні дані тієї організації або особи, яка випустила цей буклет.

Приклади оформлення буклетів можна переглянути в Інтернеті. Для цього слід в рядку пошуку ввести ключове слово *буклет* і вибрати команду (або гіперпосилання, залежно від пошукової системи) **Зображення**.



Перевірте себе

- 1°. Що таке бюлетень?
- 2°. Опишіть послідовність створення бюлетеня в програмі **Publisher 2007**.
- 3°. Як встановити параметри сторінки в шаблоні бюлетеня?
- 4°. Як переглянути зміст певної сторінки бюлетеня?
- 5°. Опишіть послідовність встановлення різної кількості стовпців тексту на сторінці бюлетеня.
- 6°. Які дані, як правило, відображаються в заголовку бюлетеня? Як внести зміни до заголовка?
- 7°. Опишіть, як заповнити **Зміст** у бюлетені.
- 8°. Для чого в бюлетені використовують **Бічну панель**, що на ній розміщують?
- 9°. Поясніть, як змінити розміри або перемістити стовпець на інше місце в публікації.
- 10°. Як додати новий напис до публікації?
- 11°. Як визначити, чи продовжується текст статті в інших написах?
- 12°. Що таке «перетікання» та як установити або відмінити перетікання тексту з одного напису до іншого?
- 13°. Про що свідчить позначка **A ...** біля стовпця з текстом? Які дії повинен виконати користувач, щоб її прибрати?
- 14°. Поясніть, для чого виконується групування графічних об'єктів і підписів під ними.
- 15°. Що таке буклет? Для чого він призначений?
- 16*. Опишіть відмінності в створенні бюлетеня та буклета. Для яких цілей у вашій школі можуть виготовлятися ці публікації?



Виконайте завдання

- 1°. Запустіть на виконання програму **Publisher 2007**. Використовуючи початкове вікно програми, створіть новий бюлетень на основі шаблону *Фоторамка*. Установіть колірну схему *Море*, схему шрифтів *Архівна*, кількість сторінок – *одна*. Збережіть публікацію у власній папці у файлі з іменем **вправа 4.5.1.pub**.
- 2°. Створіть у програмі **Publisher 2007** буклет на основі шаблону *Геометрична*. Установіть колірну схему *Плюш*, схему шрифтів *Ділова 1*, кількість сторінок – *одна*. Додайте до буклета форму *Бланк реєстрації*. У заголовку бланка для реєстрації уведіть: *Укажіть ваші контактні дані, видаліть написи, що стосуються проведення заходів, ціни і способів оплати*. Перемістіть заголовок бланка і шаблон для заповнення контактних даних так, щоб вони займали нижню частину тієї самої сторінки. Збережіть публікацію у власній папці у файлі з іменем **вправа 4.5.2.pub**.
- 3°. Створіть у програмі **Publisher 2007** буклет, присвячений вашому класу. Шаблон, колірну схему і схему шрифтів доберіть самостійно. На ти-



тульній сторінці розмістіть фотографію школи та її назву, назву населеного пункту і рік розробки. Збережіть публікацію у папці **Мої документи** у файлі з іменем **вправа 4.5.3.pub**.



- 4*. Створіть у програмі **Publisher 2007** на основі одного із шаблонів бюлетень, присвячений одному з письменників, творчість якого ви вивчаєте з української або світової літератури. Кількість аркушів – *один*. Включіть до бюлетеня на титульній сторінці фото письменника і його коротку біографію, а в наступних статтях – уривки з віршованих або прозових творів, які вам найбільше сподобалися. На останній сторінці вкажіть власні дані як видавця. Збережіть публікацію у папці **Мої документи** у файлі з іменем **вправа 4.5.4.pub**.
- 5*. Відкрийте вказаний учителем файл, наприклад **Тема 4\Завдання 4.5\зразок 4.5.5.pub**. Використовуючи матеріали з файлу **Тема 4\Завдання 4.5\зразок 4.5.5.doc** та фотографії весняної природи з файлів папки **Тема4\Завдання 4.5**, завершіть створення бюлетеня *Альманах літературної творчості*.
- 6*. Створіть у програмі **Publisher 2007** бюлетень, присвячений навчанню у школі за вашим профілем. Підберіть текстовий та ілюстративний матеріали. Обґрунтуйте в статтях доцільність вивчення саме цього профілю. Збережіть публікацію у власній папці у файлі з іменем **вправа 4.5.6.pub**.



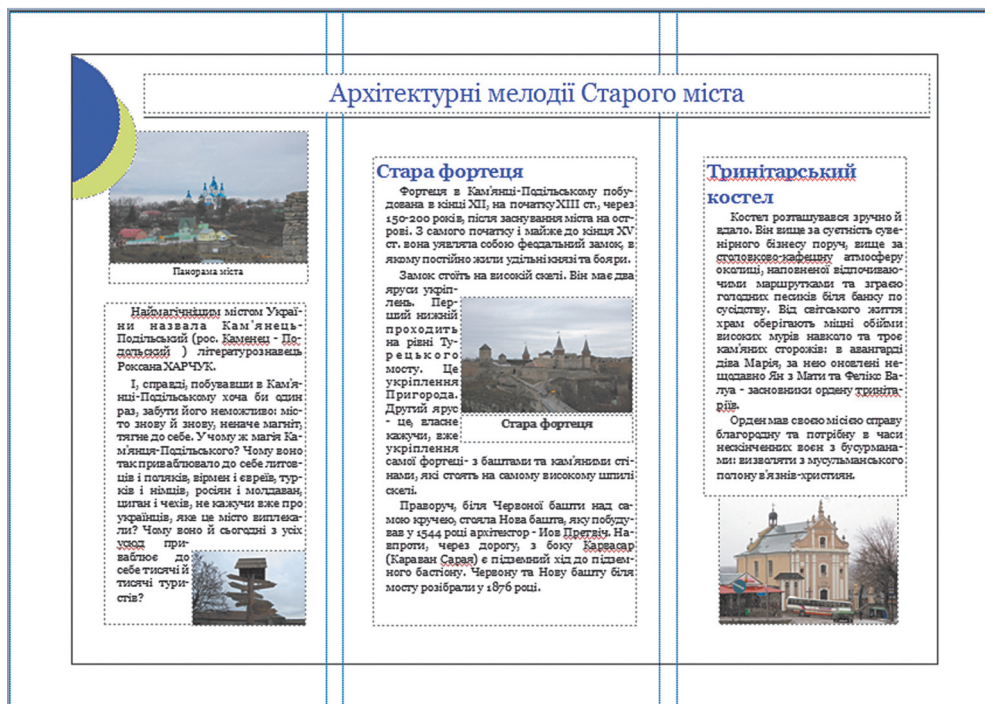
Практична робота № 16. Створення інформаційного бюлетеня та буклета

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Створіть, використовуючи програму **Publisher 2007**, бюлетень на тему «7 чудес України». Під час його створення можете використати шаблон *Осінь* або розробити власний макет. Зразок бюлетеня (рис. 4.49) – першої та другої сторінок розміщено в папці **Тема 4\Практична 16** у файлах **Практична 16_1.jpg** і **Практична 16_2.jpg** відповідно.
2. Для створення бюлетеня використовуйте текстові та графічні матеріали з файлів папки **Тема 4\Практична 16**.
3. Збережіть бюлетень у своїй папці у файлі з іменем **практична робота 16.1.pub**.
4. Створіть буклет про одну з історико-культурних або природних пам'яток України, що увійшли до переліку семи чудес України. Під час його створення можете використати шаблон *Затемнення* або розробити власний макет. Зразок буклета (рис. 4.50) – першої та другої сторінок розміщено в папці **Тема 4\Практична 16** у файлах **Практична 16_3.jpg** і **Практична 16_4.jpg** відповідно.
5. Для створення буклета використовуйте текстові та графічні матеріали з файлів папки **Тема 4\Практична 16**.
6. Збережіть буклет у своїй папці у файлі з іменем **практична робота 16.2.pub**.



a)



б)

Рис. 4.50. Зразок оформлення буклета

4.6. Інтегроване використання засобів опрацювання електронних документів



1. Що таке **Буфер обміну**? Які операції можна виконувати з його використанням? Чим **Буфер обміну** операційної системи відрізняється від **Буфера обміну Microsoft Office**?
2. Що таке конвертація файлів? У яких форматах можна зберегти текстовий документ; електронну таблицю; мультимедійну презентацію; базу даних?
3. Що таке ярлик об'єкта в операційній системі **Windows**?
4. Об'єкти яких типів можна вставити в текстовий документ? Як це можна зробити?
5. У яких форматах можна зберігати веб-сторінки?

Поняття складеного документа

Протягом вивчення курсу інформатики ви ознайомилися з різними прикладними програмами – графічним редактором **Paint**, текстовим процесором **Word 2007** і табличним процесором **Excel 2007**, програмами підготовки мультимедійних презентацій **Power Point 2007** і комп'ютерних публікацій **Publisher 2007**, системою управління базами даних **Access 2007** і навчалися створювати й опрацьовувати з їх використанням різноманітні електронні документи: текстові документи, електронні таблиці, комп'ютерні презентації, бази даних та ін.

Але досить часто під час створення електронного документа в певній програмі виникає потреба вставити до нього фрагменти з документів, які створені в інших програмах.

Наприклад, у текстовий документ, підготовлений у текстовому процесорі **Word 2007**, необхідно вставити растровий малюнок, створений у графічному редакторі **Paint**, а також фрагмент електронної таблиці, яка розроблена в табличному процесорі **Excel 2007**, та таблицю з бази даних **Access 2007**. Або під час заповнення бази даних у **СУБД Access 2007** потрібно використати вже підготовлені дані з електронної таблиці **Excel 2007**, або в мультимедійну презентацію потрібно вставити відеоролики та аудіозаписи, які були підготовлені заздалегідь. У результаті таких дій користувач отримує документ, що включає фрагменти, створені за допомогою різних програмних засобів. Подібний документ отримав назву *складеного*, або *інтегрованого*, документа.



Складений (інтегрований, лат. *integrum* – ціле) документ – це документ, у який вставлено фрагменти з даними різних типів, що створювалися в інших прикладних програмах.

Програма (документ, файл), з якої проводиться вставлення фрагмента у складений документ, називається *джерелом*, а програма (документ, файл), куди вставляються об'єкти з інших джерел, – *приймачем*.

Вставка фрагментів даних з різних програм

Операційна система **Windows** надає користувачу кілька способів вставки даних з різних програм до складеного документа. З деякими з них ви вже ознайомилися раніше: технологія **drag-and-drop** (англ. *перетягнути та кинути*) і використання **Буфера обміну**.

Так, у багатьох програмах, які працюють від керівництвом ОС **Windows**, для вставки об'єктів передбачена можливість перетягування потрібного фрагмента чи об'єкта з використанням миші з вікна однієї програми до вікна іншої (технологія **drag-and-drop**). При виконанні цієї операції об'єкт переміщується з документа-джерела у документ-приймач, виконання цієї операції за натиснутої клавіші **Ctrl** приводить до копіювання об'єкта в документ-приймач. Такий спосіб рекомендується використовувати, якщо відстань між вікнами програми-джерела і програми-приймача невелика. Для цього вікна цих програм доцільно розташувати поруч у віконному режимі.

Також вставити потрібний об'єкт у документ-приймач можна, використавши **Буфер обміну** операційної системи **Windows**. Для цього слід об'єкт з документа-джерела скопіювати в **Буфер обміну** (команда **Скопіювати**), а потім вставити його з **Буфера обміну** в документ-приймач (команда **Вставити**). Під час роботи з програмами **Microsoft Office 2007** використовується **Буфер обміну Microsoft Office 2007**, який, як ви знаєте, може одночасно зберігати 24 об'єкти. Це можна використати для швидкого збирання фрагментів з документів-джерел та одночасного їх вставлення в документ-приймач. При виконанні цієї операції вікно **Буфера обміну** в усіх програмах **Microsoft Office 2007**, які задіяні в цій роботі, повинно бути відкритим (для цього потрібно натиснути кнопку відкриття діалогового вікна відповідної групи на вкладці **Основне**).

Використовуючи ці способи, у документ-приймач вставляється копія об'єкта з документа-джерела, при цьому значення деяких властивостей вставлених об'єктів можуть змінитися.

За таких способів вставлення ніякого зв'язку між програмами, документами й об'єктами не встановлюється. І в подальшому опрацювання вставленого об'єкта забезпечується засобами програми, у якій здійснюється робота зі складеним документом. Наприклад, вставивши до текстового документа малюнок з програми **Paint**, користувач уже не зможе використовувати для зміни малюнка інструменти графічного редактора, такі як **Пензель**, **Ластик**, **Розпилювач**, не зможе здійснити інвертування кольорів, виділення довільної області тощо. Він може скористатися тільки засобами, які надає текстовий процесор для роботи з графічними зображеннями.

Відсутність зв'язку між документом-джерелом і документом-приймачем приводить до того, що ніякі зміни об'єкта-оригіналу не відображаються в документі-приймачі. Наприклад, якщо в текстовий документ вставлено діаграму з електронної книги **Excel**, то вона вставляється як рисунок, і при зміні числових даних у документі-джерелі це ніяк не вплине на зображення діаграми, яка вставлена в складений документ.

Технологія OLE

Крім перелічених вище способів, які не завжди є доцільними, вставити об'єкти до складеного документа можна і способами, які забезпечуються технологією **OLE**.

Технологія **OLE** (англ. *Object Linking and Embedding* – зв'язування та вбудовування об'єктів) дає змогу вставляти об'єкти в складений документ, забезпечуючи різні способи зв'язку цього об'єкта з документом-джерелом і з програмою-джерелом.



Технологія OLE – це технологія вставлення об'єктів різних типів у документи інших типів.

Технологія **OLE** передбачає два способи вставлення об'єктів:

- **вбудовування об'єкта** – забезпечує вставлення в документ-приймач копії об'єкта з автоматичним встановленням зв'язку з програмою-джерелом, але зв'язок з документом-джерелом втрачається. У такий спосіб ніякі зміни оригінального об'єкта не впливають на вигляд його копії у складеному документі, але для опрацювання вставленого об'єкта можна користуватися не тільки засобами програми-приймача, а й засобами програми-джерела;
- **зв'язування об'єкта** – у документ-приймач вставляється не сам об'єкт, а тільки посилання на нього у вигляді образу цього об'єкта з автоматичним встановленням зв'язку і з документом-джерелом, і з програмою-джерелом. За такого способу будь-які зміни об'єкта у документі-джерелі приводять до автоматичної зміни образу об'єкта у документі-приймачі. Для опрацювання об'єкта можна використовувати засоби програми-джерела.

Під час перегляду та друкування складеного документа відрізнити, у який спосіб вставлено об'єкти в документ, майже неможливо (крім встав-

лених у вигляді значка). Спосіб вставлення впливає тільки на спосіб редагування вставлених об'єктів (пояснено вище) і на розмір складеного документа. Оскільки вбудовані об'єкти розташовуються всередині документа-приймача, то розмір файлу складеного документа зростає за рахунок розміру цих об'єктів. При зв'язуванні об'єктів розмір складеного документа майже не збільшується, оскільки в документ вставляється не сам об'єкт, а посилання на нього, яке займає дуже мало місця.

Технологію **OLE** у повному обсязі підтримують не всі прикладні програми. Деякі з них можна використовувати тільки як приймач, інші – тільки як джерело, але є й ті, що можуть виступати в обох ролях. Наприклад, прикладні програми комплексу **Microsoft Office 2007** повністю забезпечують роботу за технологією **OLE**, а графічний редактор **Paint** можна використовувати за цією технологією тільки як джерело об'єктів.

Вбудовування та зв'язування об'єктів у програмах Microsoft Office 2007

Вбудовування або зв'язування об'єктів у програмах **Microsoft Office 2007** з відкритого документа-джерела здійснюється через **Буфер обміну** командою **Спеціальне вставлення** та виконується за таким алгоритмом:

1. Виділити в документі-джерелі об'єкт (фрагмент документа), який вставляється, і виконати **Основне** ⇒ **Буфер обміну** ⇒ **Копіювати**.
2. Розмістити курсор у потрібному місці документа-приймача і виконати **Основне** ⇒ **Буфер обміну** ⇒ **Вставити** ⇒ **Спеціальне вставлення**.
3. Указати в діалоговому вікні **Спеціальне вставлення** спосіб вставлення об'єкта, вибравши потрібний перемикач:
 - для вбудовування – перемикач **Вставити**;
 - для зв'язування – перемикач **Зв'язати** (рис. 4.51).
4. Вибрати у списку **Як** тип об'єкта, у форматі якого об'єкт-копію буде вставлено в документ-приймач.

Залежно від обраного типу в подальшому користувачу надаватимуться відповідні засоби опрацювання об'єкта-копії. Набір можливих варіантів для вибору залежить від типу об'єкта-оригіналу. Для вставлення

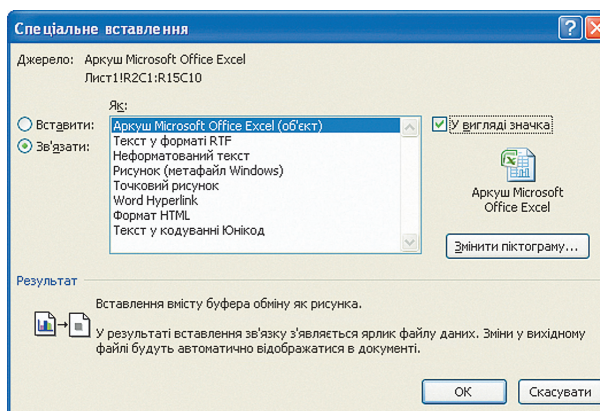


Рис. 4.51. Вікно **Спеціальне вставлення** для зв'язування об'єкта з відкритого документа

об'єкта з тими самими властивостями, які він має в документі-джерелі, рекомендується вибрати тип *Об'єкт*.

5. Вибрати спосіб відображення об'єкта-копії у документі-приймачі, встановивши за потреби позначку прапорця **У вигляді значка**:

- позначка прапорця не встановлена – об'єкт вставляється у такому самому вигляді, як і в документі-джерелі;
- позначка прапорця встановлена – у документ замість об'єкта вставляється значок даного типу об'єкта (його можна змінити, вибравши кнопку **Змінити піктограму**), який є посиланням на оригінальний об'єкт. Для відкриття об'єкта-оригіналу потрібно буде двічі клацнути на вставленому значку.

6. Вибрати кнопку **ОК**.

Використовуючи зв'язування об'єктів, слід пам'ятати:

- програма, яка була використана для створення об'єкта-оригіналу, повинна бути обов'язково встановлена на тому комп'ютері, де відбувається перегляд і опрацювання складеного документа;
- файли документа-джерела і документа-приймача потрібно заздалегідь зберегти;
- при зміні місця збереження складеного документа або документів-джерел може відбутися **розрив (втрата) зв'язків** і тоді у складеному документі вставлені об'єкти не відображатимуться.

Опрацювання вбудованих і зв'язаних об'єктів у програмах Microsoft Office 2007

Опрацювання вбудованих і зв'язаних об'єктів у складеному документі має свої відмінності.

За подвійного клацання на *вбудованому об'єкті* у вікні програми-приймача відкриється додаткове вікно із засобами програми-джерела, у якому відобразиться вбудований об'єкт, і засобами цієї програми можна здійснювати опрацювання об'єкта.

На рисунку 4.52 представлено зовнішній вигляд вікна програми **Word 2007**, у якому планується опрацювання діаграми табличного процесора **Excel 2007**, вбудованої в текстовий документ. Як видно, на **Стрічці** з'явилися нові вкладки **Формули**, **Дані**, **Знаряддя для діаграм**, за допомогою яких і здійснюватиметься редагування діаграми. Після завершення роботи для повернення у вікно складеного документа слід вибрати довільне місце поза вбудованим об'єктом.

Після подвійного клацання на *зв'язаному об'єкті* відкривається вікно програми-джерела, у якому буде відкрито весь документ-джерело для подальшого його опрацювання. Далі користувачу потрібно виконати необхідні операції над об'єктом, зберегти змінений документ і закрити вікно програми-джерела.

За замовчуванням зв'язані об'єкти у складеному документі оновлюються автоматично під час внесення змін до оригіналу об'єкта. Якщо документ-приймач закрито, то зміни будуть внесені при його відкритті. Якщо оновлення не відбувається автоматично, то це можна зробити вручну. Для цього в контекстному меню зв'язаного об'єкта потрібно вибрати команду **Оновити зв'язок**.

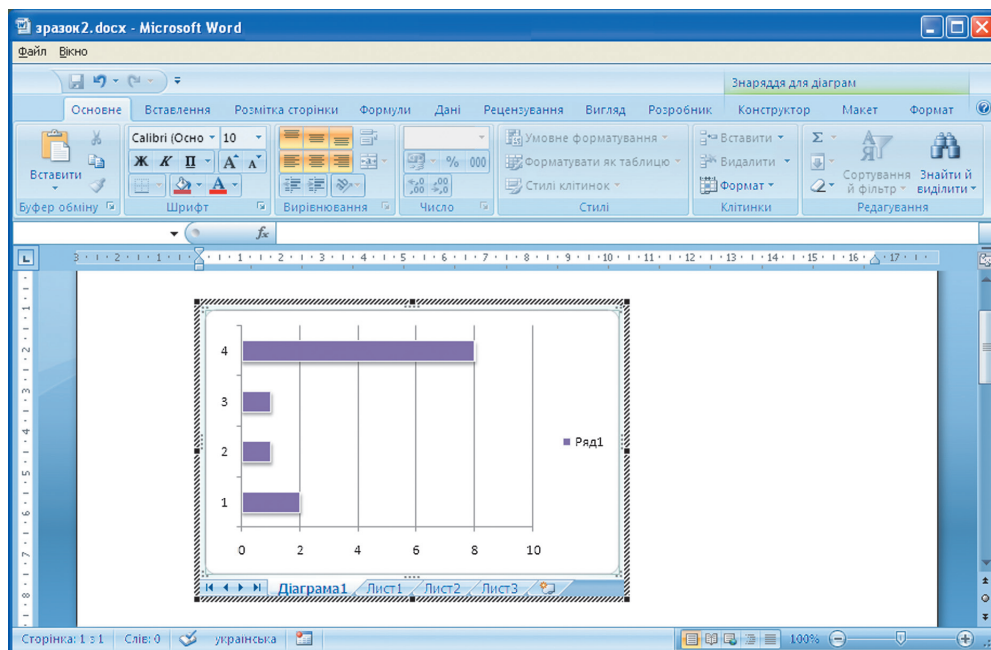


Рис. 4.52. Додаткове вікно програми-джерела під час опрацювання вбудованого об'єкта

Налаштування програмного середовища можна змінити так, щоб зв'язані об'єкти не оновлювалися взагалі або оновлювалися за запитом користувача під час відкриття документа-приймача (рис. 4.53). Також користувачу потрібно контролювати зв'язки документів на випадок їх розриву.

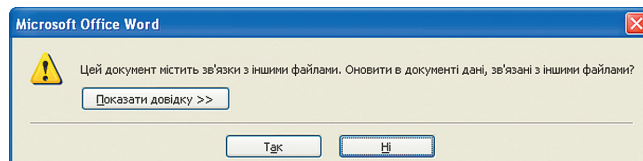


Рис. 4.53. Запит користувачу про оновлення зв'язаних об'єктів при відкритті складеного документа

Усі змінення зв'язків складеного документа, їх видалення, перегляд, відновлення чи налаштування можна здійснити в діалоговому вікні **Зв'язки** (рис. 4.54), яке можна відкрити двома способами:

- відкрити контекстне меню будь-якого зв'язаного об'єкта та виконати **Зв'язаний об'єкт** ⇒ **Зв'язки**;
- виконати **Office** ⇒ **Підготувати** ⇒ **Змінити посилання на файли**.

У вікні **Зв'язки** міститься перелік файлів, з якими встановлено зв'язки для даного складеного документа.

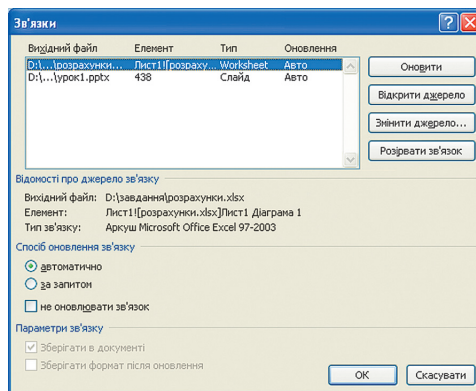


Рис. 4.54. Вікно **Зв'язки**

За вибору зі списку потрібного зв'язаного файлу можна дізнатися значення таких його властивостей:


- повне ім'я файлу-джерела;
- вид об'єкта в документі-джерелі;
- тип встановленого зв'язку;
- режим оновлення об'єкта в складеному документі – *автоматичне* або *за запитом*.

Змінення значень властивостей можна здійснити вибором відповідних кнопок і перемикачів у діалоговому вікні.



Технологія **OLE** дає змогу також вставити в документ вміст деякого файлу, створеного в іншій програмі, або створити у документі новий об'єкт засобами іншої прикладної програми. Таке вставлення можна виконати і як вбудовування, і як зв'язування файлів, залежно від обраного способу.

Наприклад, для вставлення в документ вмісту іншого файлу потрібно виконати такі дії:

1. Вибрати у документі-приймачі місце для вставлення вмісту файлу-джерела.
2. Виконати команду **Вставлення** ⇒ **Текст** ⇒ **Додати об'єкт** .
3. Вибрати у діалоговому вікні **Вставлення об'єкта** вкладку **Створення з файлу** (рис. 4.55).
4. Увести повне ім'я потрібного файлу в поле **Ім'я файлу** або вибрати його, скориставшись кнопкою **Огляд**.
5. Установити за потреби позначки прапорців **Зв'язок з файлом** (для зв'язування файлів) і **У вигляді значка**.
6. Вибрати кнопку **ОК**.

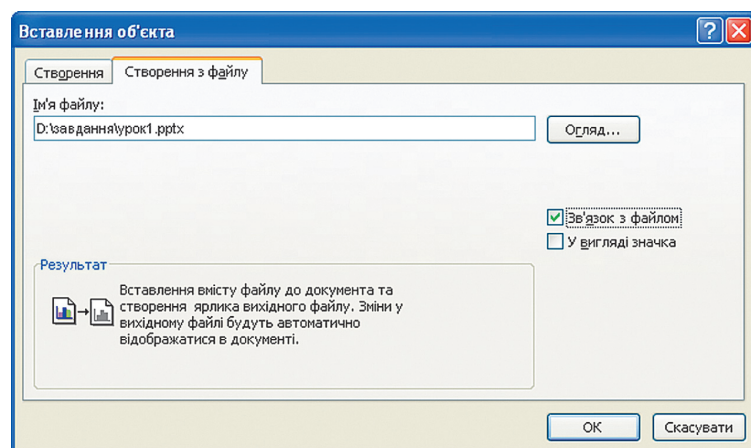


Рис. 4.55. Вкладка **Створення з файлу** вікна **Вставлення об'єкта**

У такий спосіб дуже зручно і швидко можна об'єднати кілька текстових документів в один або вставити слайди презентації в текстовий документ.

Імпорт і експорт даних із файлів

Часто обмін даними між різними програмами можна організувати з використанням операцій *імпорту* та *експорту*, якщо такі операції передбачені засобами цих програм.

Операція *імпорту* передбачає зчитування програмою даних із файлів, що підготовлені в інших програмах і формат яких не є основним для да-

ної програми. Після відкриття таких файлів (імпортування даних) програма опрацьовує отримані дані як стандартні для своєї роботи. **Експорт** полягає в збереженні даних у файлі, формат якого не є основним для даної програми. По суті, під час здійснення операцій імпорту та експорту відбувається *конвертація даних*, при цьому перетворення даних може відбуватися із частковою втратою даних, елементів форматування тощо.



Імпорт (лат. *importo* – ввезення) файлу – це відкриття в програмі файлу, створеного в іншому середовищі.

Експорт (лат. *export* – вивезення) даних – це збереження даних у файлі, призначеному для використання у якій-небудь іншій програмі.

Розглянемо як здійснюються деякі операції імпорту та експорту в прикладних програмах **Microsoft Office 2007**.

Приклад 1. Імпорт тестових файлів у програми **Microsoft Office 2007**.

Для імпорту текстових файлів у програму **Microsoft Excel 2007** потрібно виконати стандартну операцію відкриття файлу, вибравши у списку **Тип файлу** формат *Текстові файли*. Імпортуванню підлягають тільки текстові файли формату TXT.

Після цього запускається **Майстер імпорту текстів**, виконуючи кроки якого потрібно вказати, чи використовуються розділювачі в початковому документі, який саме символ-розділювач використано, яке кодування символів обрано, який тип і формат окремих стовпців даних потрібно встановити тощо. У результаті текст буде конвертовано і вставлено у вигляді таблиці в **Excel 2007**.

Приблизно так само працює **Майстер імпорту текстів** і в програмі **Access 2007**, тільки при його роботі додатково можна вказувати мову і спосіб кодування початкового тексту.

У програму **Power Point 2007** імпортувати можна текстові файли різних форматів (TXT, RTF, DOCX). У результаті буде створена презентація з деякої кількості слайдів, на кожному з яких як **заголовок** буде вставлено текст окремих рядків документа.

У **Publisher 2007** імпортувати текстові дані можна двома способами – відкрити текстовий файл вибраного формату (TXT, RTF, DOCX) або імпортувати документ **Word**.

Приклад 2. Імпорт і експорт файлів у **Access 2007**.

Система управління базами даних **Access 2007** надає особливі можливості для обміну даними з програмами **Microsoft Office 2007**. Усі необхідні для цього інструменти знаходяться на вкладці **Зовнішні дані** (рис. 4.56).

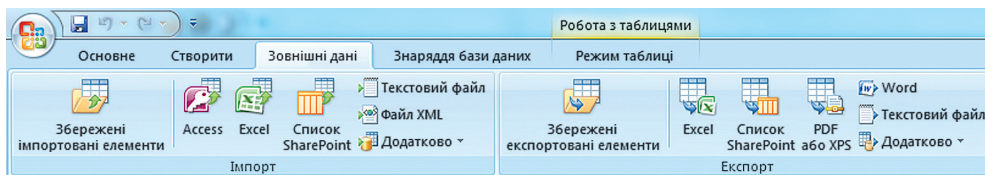







Рис. 4.56. Вкладка **Зовнішні дані**

Використовуючи групу елементів керування **Імпорт**, можна в поточну БД імпортувати дані з іншої бази даних **Access 2007** (кнопка ) , з електронної таблиці **Excel 2007** (кнопка ) , з текстового документа формату RTF (кнопка ) або експортувати (група **Експорт**) дані з поточної БД в електронну таблицю (кнопка ) або до текстового документа формату **Word 2007** (кнопка ) тощо.

Наприклад, щоб імпортувати в поточну БД **Access 2007** дані з книги **Excel 2007**, потрібно на вкладці **Зовнішні дані** в групі **Імпорт** вибрати кнопку **Excel**. Далі слідувати вказівкам майстра, який дає змогу імпортувати в **Access 2007** вміст вибраного робочого аркуша або іменованого діапазону клітинок.

Щоб експортувати дані з **Access 2007** в **Excel 2007**, у вікні бази даних необхідно вибрати таблицю, запит, форму або звіт, а потім виконати **Зовнішні дані** ⇒ **Експорт** ⇒ **Excel**. Програма збереже об'єкт бази даних у вигляді файлу електронної книги у вибраній папці. Ім'я файлу за замовчуванням буде збігатися з іменем збереженого об'єкта бази даних.

Аналогічно відбудеться експортування даних і в текстовий документ.

Приклад 3. Експорт даних у програмі **Power Point 2007**.


Створену презентацію можна експортувати в текстовий документ двома способами (**Office** ⇒ **Опублікувати** ⇒ **Створити супровідні матеріали в Microsoft Office Word**):

- у вигляді видач, коли кожний слайд презентації вставляється в текстовий документ у вигляді графічного ескізу слайда, поруч з яким вставляється додаткове місце для нотаток. Кількість слайдів на аркуші та місце розташування нотаток обираються додатково;
- у вигляді *структури* презентації, коли текстові дані з кожного слайда розміщуються в документі у вигляді послідовного тексту.

Також зручним видом експортування є збереження презентації у вигляді набору графічних зображень, де кожний слайд представлено окремим файлом заданого формату (**Office** ⇒ **Зберегти як** і вибрати тип файлу *Малюнок*). Усі зображення слайдів будуть зібрані в одну папку, ім'я якої збігатиметься з іменем файлу презентації.

Веб-публікація документів

Щоб швидко підготувати дані з програм **Microsoft Office** для публікації у Web, не обов'язково знати теги мови розмітки HTML, достатньо зберегти створений файл у форматі гіпертекстового документа. Для цього слід скористатися командою **Зберегти як** і вибрати тип файлу *Веб-сторінка*.

Під час збереження файлу у форматі веб-сторінки неминуче втрачаються деякі елементи форматування. Тому, перш ніж виконувати експорт даних, слід переглянути документ у режимі **Веб-документ** (кнопка ) у **Рядку стану**), щоб з'ясувати його майбутній вигляд.

Слід пам'ятати, що вставлені в початковий документ графічні зображення будуть збережені в окремій папці в окремих файлах, а при збереженні презентації у форматі веб-сторінки кожний слайд буде збережено в окремому файлі.

Отриману в результаті конвертації даних веб-сторінку можна переглядати, використовуючи програму-браузер, і в подальшому її можна буде розмістити в мережі Інтернет.



Перевірте себе

- 1°. Який документ називають складеним?
- 2°. Наведіть приклади завдань, які розв'язуються за допомогою кількох програм опрацювання електронних документів.
- 3°. Об'єкти яких типів можна вставити в текстовий документ; у мультимедійну презентацію; в електронну таблицю?
- 4°. Назвіть способи, якими можна вставити об'єкти в електронний документ. Поясніть їх сутність.
- 5°. Який документ під час вставлення об'єктів називають джерелом, а який – приймачем?
- 6°. Поясніть, як здійснюється вставка фрагментів документів та окремих об'єктів з використанням **Буфера обміну Microsoft Office 2007**.
- 7°. У чому полягає суть технології OLE? Яке її призначення?
- 8°. Поясніть, чим вбудовування об'єкта відрізняється від зв'язування об'єкта.
- 9°. Поясніть, чим вставка об'єктів через **Буфер обміну** відрізняється від вставки об'єкта за технологією OLE.
- 10°. Яка ситуація називається розривом (втратою) зв'язків? Коли це може статися? Як виправити таку ситуацію?
- 11*. У корпоративних комп'ютерних мережах створюють деякі стандартизовані об'єкти (бланки документів, логотипи, шаблони, файли-заготовки тощо), які використовуються як зв'язані об'єкти всіма співробітниками компанії. У чому перевага такого способу використання стандартизованих об'єктів? Які можливі недоліки?
- 12*. При вставленні діаграми в публікацію замість об'єкта в документі з'явилася тільки його піктограма. Поясніть, чому так сталося та як виправити цю ситуацію.
- 13*. Три фрагменти одного документа були збережені в різних файлах. Як їх швидко об'єднати?
- 14°. Чим відрізняється редагування зв'язаних об'єктів від вбудованих?
- 15°. У чому суть операцій імпортування та експортування даних?
- 16°. У чому полягає веб-публікація документа?



Виконайте завдання

1. Порівняйте два способи вставлення об'єктів в електронні документи, заповнивши таблицю.

Характеристика	Вбудовування об'єкта	Зв'язування об'єкта
Розмір файлу складеного документа		
Можливість втрати зв'язку		
Засоби для редагування		

- 2*. Відкрийте вказані вчителем файли (наприклад, з папки **Тема4\Завдання 4.6** файли **Зразок 4.6.2-1.docx**, **Зразок 4.6.2-2.docx**). Перегляньте та порівняйте їх вміст. Порівняйте розміри цих файлів. Поясніть відмінність.

Ім'я файлу	Розмір файлу	Спосіб вставлення об'єктів
Зразок 4.6.2-1.docx		
Зразок 4.6.2-2.docx		

- 3°. Укажіть переваги і недоліки вбудовування та зв'язування об'єктів.

Характеристика	Вбудовування об'єкта	Зв'язування об'єкта
Переваги		
Недоліки		

- 4°. Відкрийте по черзі вказані програми пакета **Microsoft Office** та визначте можливі формати електронних документів для експорту та імпорту даних.

Програми	Формати файлів для експорту даних	Формати файлів для імпорту даних
Word 2007		
Excel 2007		
Power Point 2007		
Access 2007		
Publisher 2007		

- 5°. Відкрийте вказані вчителем файли – текстовий документ і електронну книгу (наприклад, з папки **Тема4\Завдання 4.6** файли **текст.docx** і **таблиця.xlsx**). Вставте в текстовий документ діаграму з електронної книги з використанням **Буфера обміну**. Збережіть текстовий документ у власній папці у файлі з іменем **вправа 4.6.5.docx**. Відредагуйте довільним чином діаграму у файлі електронної книги і перегляньте текстовий документ. Чи відбулися якісь зміни в текстовому документі? Поясніть результат.
- 6°. Відкрийте вказані вчителем файли – мультимедійну презентацію і малюнок (наприклад, з папки **Тема4\Завдання 4.6** файли **малюнок.bmp** і **презентація.pptx**). На слайд № 1 вставте малюнок через **Буфер обміну**, на слайд № 2 – вбудуйте малюнок, на слайд № 3 – зв'яжіть малюнок. Збережіть презентацію у власній папці у файлі з іменем **вправа 4.6.6.pptx**. Відредагуйте малюнок у документі-джерелі. Перегляньте презентацію та поясніть отриманий результат.
- 7°. Відкрийте вказаний учителем файл текстового документа (наприклад, **Тема4\Завдання 4.6\текст 4.6.7.docx**). Спробуйте виконати редагування вставлених у нього об'єктів. Укажіть, у який спосіб були вставлені ці об'єкти. Поясніть, чому ви так вважаєте.
- 8°. Відкрийте мультимедійну презентацію, вказану вчителем (наприклад, **Тема4\Завдання 4.6\зразок 4.6.8.pptx**). Збережіть файл у форматах HTML, JPG, RTF у власній папці. Опублікуйте презентацію у вигляді файлу видач. Перегляньте створені файли.
- 9°. Відкрийте вказаний учителем файл текстового документа (наприклад, файл **Тема4\Завдання 4.6\текст 4.6.9.docx**). Перегляньте таблицю зв'язків даного документа. Видаліть усі зв'язки в цьому документі. Збережіть файл у форматах HTML, RTF, TXT у власній папці. Перегляньте створені файли.
- 10°. Створіть базу даних **Завдання 4.6.10** та імпортуйте в неї електронну таблицю (наприклад, **Тема4\Завдання 4.6\таблиця 4.6.10.xlsx**). Збережіть створений файл у власній папці у файлі з іменем **вправа 4.6.10.accdb**.

- 11*. Відкрийте вказаний учителем файл бази даних (наприклад, **Тема4\Завдання 4.6\база 4.6.11.accdb**). Експортуйте таблицю бази даних в електронну таблицю, а звіт – у текстовий файл. Збережіть створені файли у власній папці з іменами **вправа 4.6.11.xlsx** та **вправа 4.6.11.docx**.
- 12*. Імпортуйте вказаний текстовий документ (наприклад, **Тема4\Завдання 4.6\текст 4.6.12.rtf**) в базу даних і електронну таблицю. Збережіть створені файли у власній папці з іменами **вправа 4.6.12.xlsx** та **вправа 4.6.12.accdb**.



Практична робота № 17. Опрацювання даних у кількох програмних середовищах

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте у графічному редакторі **Paint** малюнок з файлу **Тема4\Завдання 4.6\Практична 17\Квітка.jpg**.
2. Відкрийте в табличному процесорі **Excel 2007** діаграму з файлу **Тема4\Завдання 4.6\Практична 17\Кількість.xlsx**.
3. Відкрийте в СУБД **Access 2007** таблицю бази даних з файлу **Тема4\Завдання 4.6\Практична 17\Олімпійці.accdb**.
4. Відкрийте програму **Power Point 2007** і створіть слайд, на який вставте об'єкти з відкритих документів: графічне зображення як вбудований об'єкт, діаграму як зв'язаний об'єкт, список учнів як об'єкт, вставлений через **Буфер обміну**.
5. Збережіть презентацію у власній папці у файлах з іменем **практична 17** у форматах **PPTX, RTF, JPG, HTML**.
6. Екпортуйте базу даних **Олімпійці.accdb** в електронну таблицю і текстовий документ.
7. Збережіть створену електронну таблицю і текстовий документ у власній папці у файлах з іменами **Олімпійці.xlsx** та **Олімпійці.txt**.
8. Закрийте вікна всіх програм.

4.7. Типи веб-сторінок. Класифікація веб-сайтів

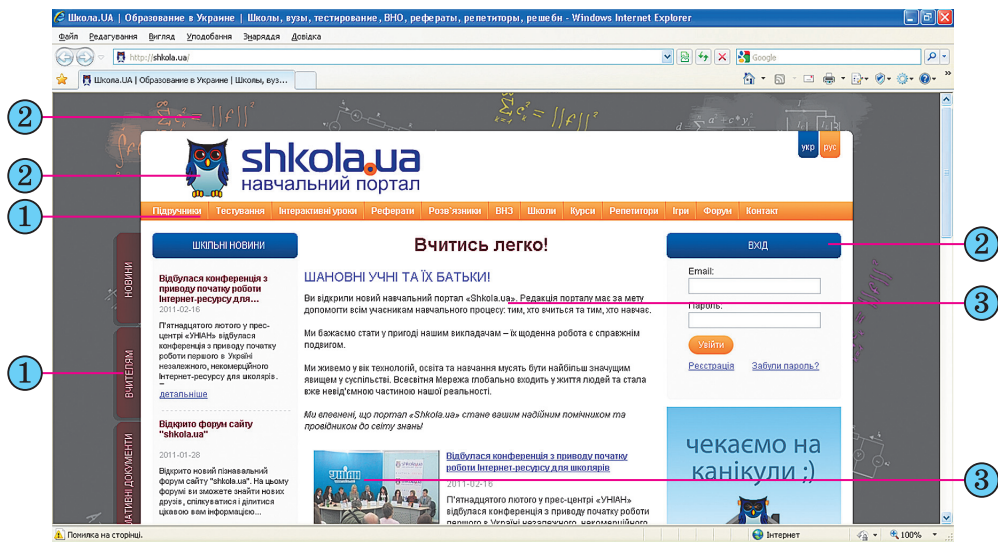


1. Який документ називають складеним?
2. Що таке веб-сторінка, веб-сайт, веб-портал? У чому полягає відмінність між цими поняттями?
3. Які мови використовуються для створення веб-сторінок?
4. Повідомлення яких видів за способом подання можуть міститися на веб-сторінках?
5. Із чого складається доменне ім'я Інтернет-ресурсу? Якою є структура URL-адреси?

Типи веб-сторінок

Прикладом складеного документа є веб-сторінка, оскільки вона може включати дані різних типів: графічні зображення, анімацію, відео та музичні фрагменти. Веб-сторінки є інформаційними ресурсами служби **World Wide Web**, і у 9-му класі ви ознайомилися з тим, як здійснювати пошук і перегляд цих ресурсів. Тепер ознайомимося зі способами їх створення. Під час створення веб-сторінок важливо враховувати їх структуру та типи, а також види сайтів, до складу яких будуть входити ці сторінки.

Розглянемо приклад веб-сторінки (рис. 4.57). Серед елементів, які на ній відображаються, можна виділити такі групи:



1 Елементи навігації 2 Елементи дизайну 3 Контент

Рис. 4.57. Елементи веб-сторінки

- **контент** (англ. *content* – зміст) – змістове наповнення веб-сторінки, доступне користувачу: тексти, зображення, відео, звукові дані та ін.;
 - **елементи навігації** – засоби для переходу до інших веб-сторінок;
 - **елементи дизайну** – елементи структурування контенту та його форматування, оформлення сторінки.
- Залежно від призначення можна виділити такі типи веб-сторінок:
- **головна (домашня) сторінка** – сторінка, з якої розпочинається перегляд веб-сайта при переході на сайт за його URL-адресою. На ній, як правило, розкривається тематика сайта, його призначення, наводяться дані про розробників, пояснюється, які матеріали можна знайти на інших сторінках сайта;
 - **інформаційні сторінки (сторінки тематичних розділів)** містять тексти, зображення та повідомлення інших видів, які розкривають тему сайта або деякого його розділу. Головна сторінка сайта є окремим випадком інформаційної сторінки;
 - **сторінки-контейнери** містять списки посилань на ресурси даного або інших сайтів:
 - **веб-каталоги** – посилання на веб-ресурси (веб-сторінки, веб-сайти);
 - **каталоги файлів** – посилання на файли, які можуть бути завантажені користувачем;
 - **комунікативні (інтерактивні) сторінки** призначені для надання користувачам сайта засобів спілкування та зворотного зв'язку з розробниками сайта:
 - **сторінки форуму** призначені для організації спілкування у форумі;
 - **сторінки чата** призначені для організації спілкування в чаті;
 - **сторінки гостьової книги** призначені для надання можливості відвідувачам сайта залишати свої коментарі, як правило, для авторів сайта;

- **сторінки форми** призначені для проведення опитування, з'ясування точки зору відвідувачів, здійснення вибору товарів або послуг на комерційних сайтах

та ін.

Залежно від технологій, використаних під час створення веб-сторінок, можна виділити такі типи сторінок:

- **статичні сторінки** створені, як правило, засобами мови розмітки гіпертексту **HTML**. Вміст сторінок залишається незмінним для всіх відвідувачів сайту. На таких сторінках не передбачається часта зміна даних та їх оформлення, вони не потребують спеціального програмного забезпечення для зберігання даних. Такими можуть бути сторінки з навчальними матеріалами, історичними відомостями, описами музейних експонатів та ін.;
- **динамічні сторінки** створені з використанням мов програмування, таких як **PHP** (англ. *Personal Home Page Tools* – інструменти розробки персональних домашніх сторінок), **ASP** (англ. *Active Server Pages* – активні серверні сторінки), **PERL** (англ. *Practical Extraction and Report Language* – практична мова для видобування даних і складання звітів) та ін. Такі сторінки призначені для відображення інформаційних матеріалів, що часто оновлюються. Такими можуть бути сторінки з новинами, з переліком товарів на сайті Інтернет-магазину, з добіркою популярних відеофрагментів та ін. При відвідуванні таких сторінок користувачі можуть бачити різний вміст;
- **флеш-сторінки** (англ. *flash* – спалах) створюються з використанням технології розробки анімаційних зображень **Adobe Flash**, а тому такі сторінки привабливі, яскраві, містять багато анімації та звукових ефектів. Вигляд веб-сторінки змінюється в результаті відтворення

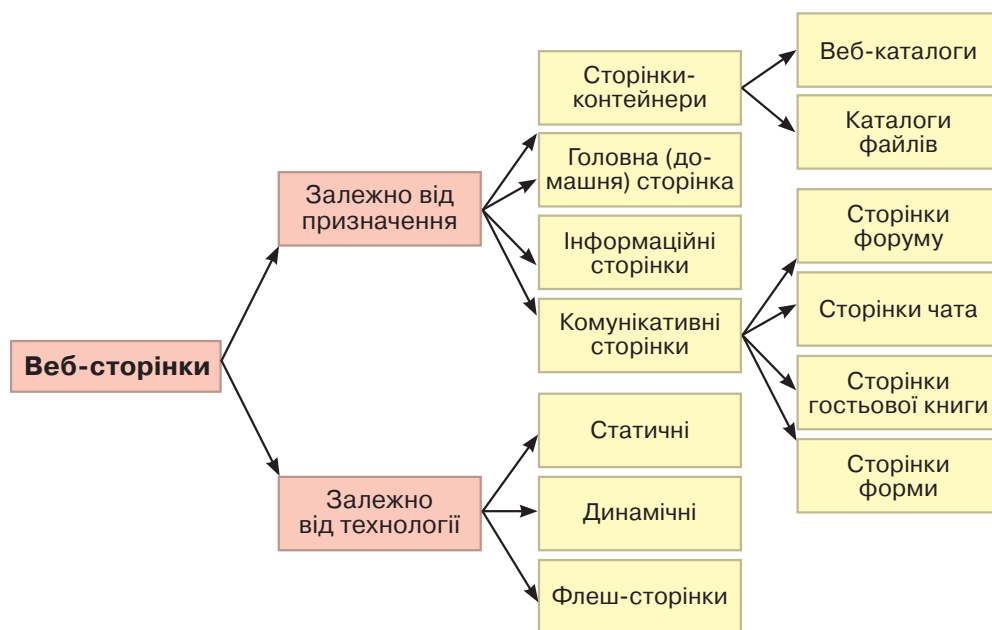


Рис. 4.58. Схема класифікації веб-сторінок

флеш-анімації та залежно від положення вказівника. Нетиповий дизайн зацікавлює відвідувачів, тому флеш-сторінки часто розміщуються на сайтах дизайнерів, художників, фотографів тощо, роботи яких ефектніше виглядають у динамічному оформленні. У той самий час флеш-сторінки важко змінювати, тому для них майже не передбачається оновлення даних та інших елементів. Обсяги файлів, у яких зберігаються флеш-сторінки, дуже великі, такі сторінки повільно завантажуються. Для їх перегляду повинні бути встановлені плагіни для відтворення флеш-анімації.

Класифікація веб-сторінок може бути представлена схемою, поданою на рисунку 4.58.

Класифікація веб-сайтів

Різні веб-сайти можуть мати різну **структуру** – кількість веб-сторінок та їх типи, внутрішню тематичну організацію, сукупність внутрішніх зв'язків. Спільним для всіх сайтів є наявність у структурі головної сторінки, яка пов'язана з усіма тематичними розділами сайту. Розділи можуть мати власні початкові сторінки, що пов'язані з головною сторінкою та інформаційними сторінками. Кількість веб-сторінок та їх типи визначаються обсягом і характером матеріалів сайту.

Класифікувати сайти можна за значеннями різних властивостей. Наведемо кілька можливих класифікацій.

Відповідно до основної технології, що була використана для створення веб-сторінок, веб-сайти поділяють на **статичні** (усі сторінки статичні), **динамічні** (усі сторінки динамічні), **флеш-сайти** (усі сторінки побудовані за флеш-технологією) та **мішані**.

За призначенням виділяють такі типи сайтів:

- **сайти, що надають інформаційні матеріали:** інформаційно-тематичні сайти, сайти новин, електронні бібліотеки, енциклопедії, словники, каталоги, сховища файлів різних видів, медіатеки та ін.;
- **сайти для он-лайн контактів і спілкування:** форуми та веб-чати, сайти соціальних мереж, блоги, сайти знайомств і мережних ігор та ін.;
- **сайти електронної комерції:** Інтернет-магазини та аукціони, системи електронних платежів, сайти банків, бірж, сайти, що пропонують різноманітні платні послуги – навчання іноземних мов, консультації психолога та ін.;
- **сайти он-лайн сервісів:** сайти електронної пошти, пошукових систем, сайти перекладу слів і текстів. Крім того, он-лайн сервісами, для яких створені відповідні сайти, є сервіси:
 - надання **хостингу** (англ. *hosting* – виявлення гостинності) – виділення апаратних і програмних ресурсів сервера для розміщення файлів користувача, забезпечення доступу до них, опрацювання запитів;
 - **он-лайн офіси** – сервіс створення он-лайн документів: текстових, презентацій, електронних таблиць та ін.;
 - сервіс автоматизованої розробки веб-сайтів. На сайтах, розроблених для підтримки цього сервісу, розміщують **системи керування контентом CMS** (англ. *Content Management System* – системи керування вмістом, контентом) – програми, що використовують для створення, редагування та керування вмістом веб-сайта

та ін.

Наведена вище класифікація сайтів представлена на рисунку 4.59.

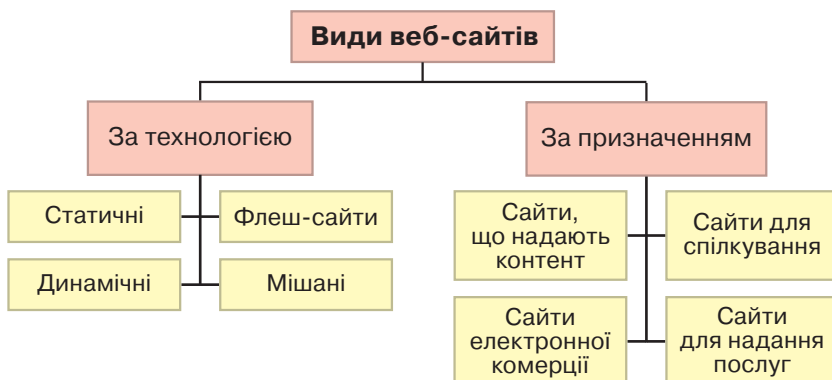


Рис. 4.59. Деякі класифікації веб-сайтів



Оцінювання веб-сайтів

Здійснюючи навігацію ресурсами Інтернету, ви, напевно, звертали увагу на те, як по-різному сприймаються сайти однакової тематики та спрямованості. На одних сайтах ви затримувалися надовго, з деяких відразу переходили на інший ресурс. Тривалість перегляду залежить від привабливості дизайну сайта, зручності пошуку потрібних матеріалів. Для відвідувача під час оцінювання веб-сайта важливими показниками є:

- ступінь інформативності веб-сайта, можливість отримати корисні матеріали, які не повторюють відомості з інших ресурсів;
- частота оновлення даних на сайті, що свідчить про увагу розробника до власного проекту;
- наявність інтерактивних засобів, можливість проведення обговорень, отримання відповідей, звертання до розробника за додатковими відомостями та коментарями;
- мови, якими доступні матеріали на веб-сайті;
- візуальна привабливість сайта, дизайн, стиль оформлення;
- **юзабіліті сайта** (англ. *usability* – зручність), тобто зрозумілість, зручність навігації, легкість знаходження потрібних даних, структурованість матеріалу – все те, що забезпечує простоту, комфортність та ефективність у процесі використання сайта

та ін.

Тривалість перебування відвідувачів на сайті, повторне звертання до його ресурсів визначають популярність сайта та його рейтинг у базах даних пошукових систем.



Перевірте себе

- 1°. Які групи елементів можна виділити на веб-сторінці? Охарактеризуйте кожну групу.
- 2°. Що таке контент веб-сторінки?
- 3°. Поясніть особливості різних видів веб-сторінок залежно від їх вмісту.
- 4°. Чим відрізняється домашня сторінка веб-сайта від інших інформаційних сторінок?
- 5°. Поясніть особливості різних видів веб-сторінок залежно від технологій, використаних при їх створенні.

- 6°. За якими ознаками можна класифікувати веб-сайти?
 7°. Поясніть особливості різних видів веб-сайтів за рисунком 4.59.
 8°. Який сервіс називають хостингом?
 9°. Що таке система керування контентом?
 10°. За якими критеріями оцінюють веб-сайти?
 11°. Що розуміють під юзабіліті сайта?



Виконайте завдання

- 1°. Установіть, до якої групи належать позначені елементи веб-сторінки (рис. 4.60).

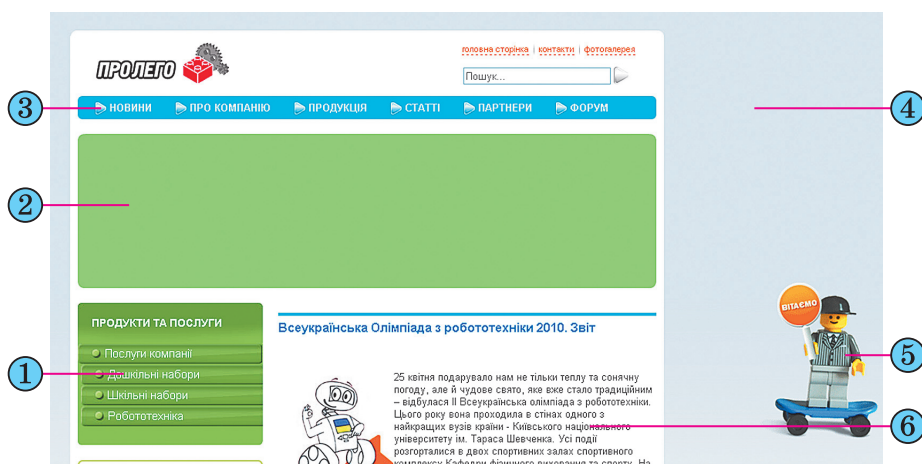


Рис. 4.60

Складові веб-сторінки	Контент	Елементи навігації	Елементи дизайну
Номери елементів			



- 2°. Відкрийте веб-сторінки з указаними URL-адресами та вкажіть їхній тип.

URL-адреса сторінки	Тип веб-сторінки	
	За призначенням	За технологією розробки
http://ukrainiancomputing.info/ada_u.html		
http://www.museum-ukraine.org.ua/index.php		
http://www.litforum.net.ua/showthread.php?t=144&page=70		
http://mncat.mikanewton.info		
http://children.kmu.gov.ua/history/persons/1358.html		

- 3°. Перегляньте веб-сайти з указаними адресами та встановіть відповідність між URL-адресами та типами веб-сайтів за їх призначенням.

	<i>URL-адреса сайта</i>
1	http://www.profeo.ua
2	http://www.freemail.ukr.net/ua
3	http://www.kancelarka.com.ua
4	http://formula.co.ua

	<i>Тип сайта</i>
А	Сайт, що надає інформаційні матеріали
Б	Сайт електронної комерції
В	Сайт он-лайн сервісу
Г	Сайт для он-лайн контактів і спілкування

- 4*. Перегляньте веб-сайти з указаними адресами та встановіть відповідність між URL-адресами веб-сайтів та їхніми типами відповідно до основної технології створення веб-сторінок.

	<i>URL-адреса сайта</i>
1	http://veselka-mebli.com
2	http://www.icfcst.kiev.ua/museum/museum-map_u.html
3	http://top.bigmir.net/show
4	http://www.brevis.kiev.ua

	<i>Тип сайта</i>
А	Мішаний
Б	Динамічний
В	Флеш-сайт
Г	Статичний

- 5*. Укажіть, яким типам сайтів за призначенням відповідають описи:
- сайт містить 1–5 сторінок, включає загальні відомості про власника сайта та його контактні дані;
 - сайт містить каталог товарів і засоби для їх замовлення та оплати через Інтернет;
 - сайт містить велику кількість відомостей з різною тематикою, сервісів, засобів для спілкування;
 - сайт містить новини автора, його ідеї, інші актуальні відомості та може супроводжуватися коментарями відвідувачів.

- 6*. Відкрийте веб-сайти з вказаними URL-адресами та вкажіть їхній тип.

<i>URL-адреса</i>	<i>Тип сайта</i>	
	<i>За технологією</i>	<i>За призначенням</i>
http://www.chl.kiev.ua		
http://www.onlandia.org.ua/ukr		
http://ticapac.pp.ua		
http://www.man.gov.ua		

- 7*. Висловіть свій погляд на веб-сайти з даними URL-адресами.

<i>Критерії</i>	<i>URL-адреси</i>			
	<i>http://www.nenc.gov.ua</i>	<i>http://www.Auth.gov.ua/index.htm</i>	<i>http://www.osvita.com</i>	<i>http://matajeva-sloboda.ua</i>
Наскільки інформативним для вас виявився веб-сайт?				
Коли востаннє було оновлено дані на сайті?				
Чи наявні на сайті інтерактивні засоби?				

Критерії	URL-адреси			
	<i>http:// www.nenc. gov.ua</i>	<i>http:// www.Auth.gov. ua/index.htm</i>	<i>http:// www. osvita.com</i>	<i>http:// mamajeva- sloboda.ua</i>
Якими мовами доступні матеріали веб-сайта?				
Чи вважаєте ви сайт візуально привабливим?				
Наскільки зручним і зрозумілим для вас є користування сайтом?				

4.8. Етапи створення веб-сайтів. Основи веб-дизайну



1. Які етапи розв'язування задач з використанням комп'ютера ви знаєте?
2. Що таке дизайн? Які основні принципи дизайну презентації та слайдів?
3. Що таке стиль?
4. У чому полягають закони композиції; колористики; ергономіки?
5. Що таке верстка? Яке її значення під час створення публікацій?

Етапи створення веб-сайтів

Наявність власного сайту підприємства, організації, навчального закладу тощо стає актуальним питанням сьогодення. Сайт може бути корисним у ході організації навчання, спілкування, бізнесу та ін. Розробку сайту можна замовити в дизайн-студіях або здійснити самостійно. У будь-якому разі розробка складається з кількох етапів. Ці етапи подібні до етапів розв'язування задач з використанням комп'ютера.

1. **Постановка завдання.** На цьому етапі визначається мета створення сайту, його основна тематика, обирається тип сайту, здійснюється аналіз існуючих сайтів такої самої або схожої тематики. У результаті розробник повинен знати:
 - мету, з якою створюється сайт;
 - тематику сайту;
 - тип сайту: домашня сторінка, форум, Інтернет-магазин, портал тощо;
 - відмінності сайту від інших сайтів такої самої тематики;
 - аудиторію потенційних відвідувачів сайту: вік відвідувачів, стать, коло інтересів тощо;
 - перелік сервісів для розміщення на сайті: форум, чат, пошукова система, веб-каталог, електронна пошта та ін.;
 - перспективи розвитку сайту.
2. **Визначення структури сайту та його окремих сторінок.** На цьому етапі важливо скласти перелік розділів сайту для формування системи навігації, список сторінок, визначити зв'язки між ними. Кількість сторінок залежатиме від того інформаційного наповнення, яке планується на ньому розмістити. Результатом повинна стати **мапа (карта) сайту** – діаграма, що візуально відображає ієрархію сторінок сайту, схему зв'язків і переходів між ними, тобто **внутрішню структуру сайту**.

Наприклад, для сайта-візитівки вашого класу, основними відвідувачами якого будуть учні класу та їхні батьки, мапа сайта може бути такою, як зображено на рисунку 4.61.

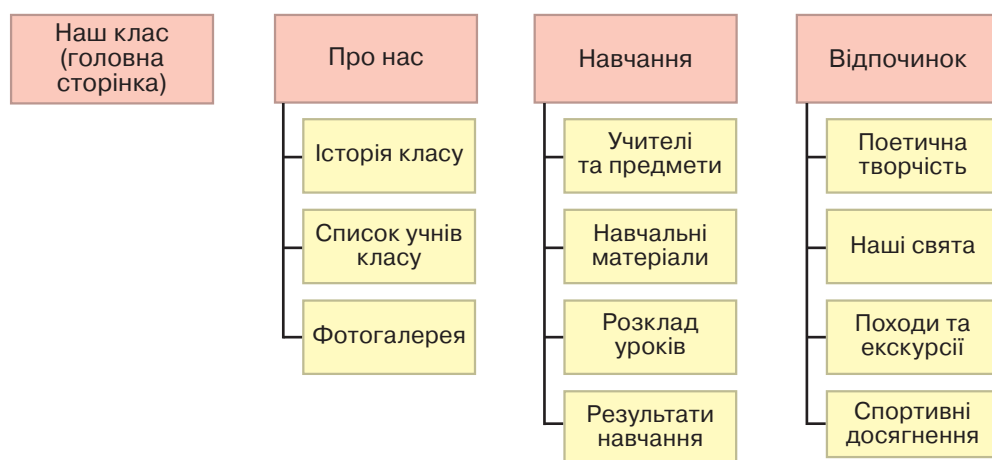


Рис. 4.61. Приклад мапи сайта

Другим завданням етапу є розробка так званої **зовнішньої структури** сайта, яка визначає зовнішній вигляд веб-сторінок. Оскільки для більшості сторінок сайта рекомендується застосовувати єдиний стиль оформлення, то потрібно визначити схему розташування на сторінках основних блоків: як буде розташований основний матеріал, додаткові інформаційні та рекламні блоки, анонси, меню, лічильник відвідувачів тощо.

Як правило, на веб-сторінках передбачено розміщення:

- верхнього блока – **заголовка**, у якому містяться логотип і назва сайта;
- **меню** для переходу до основних розділів сайта;
- **інформаційного блока** з основним матеріалом, що займає центральну частину сторінки;
- нижнього блока – **підвалу**, для розміщення контактних даних, повідомлення про авторські права тощо.

Схему зовнішньої структури представляють у вигляді **модульної сітки** (рис. 4.62). Якщо розробка сторінок сайта буде здійснюватись автоматизованими засобами, то зовнішня структура може бути запропонована в шаблоні сторінки.

3. Розробка дизайн-макета сторінок сайта. Дизайн-макет буде спиратися на попередньо розроблену зовнішню структуру сторінок сайта. Дизайн-макет сторінок включає набір значень властивостей текстових і графічних об'єктів сторінки: кольорової гами сторінок, елементів графічного оздоблення, набору шрифтів

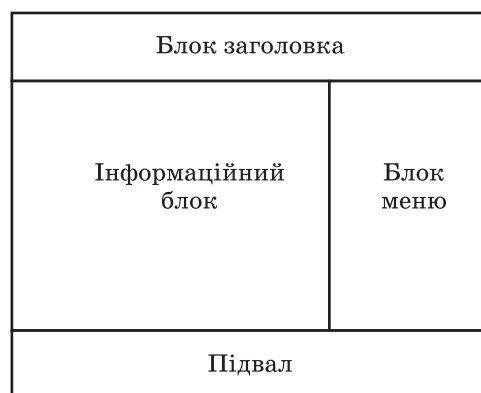


Рис. 4.62. Приклад схеми зовнішньої структури

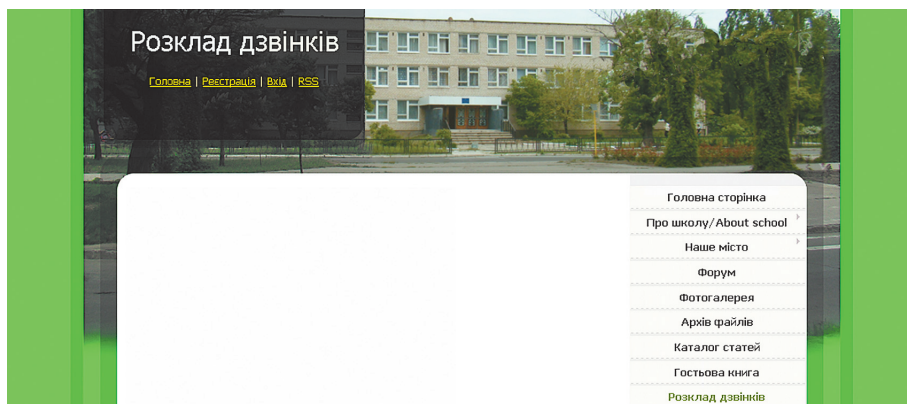


Рис. 4.63. Приклад дизайн-макета

та ін., тобто визначає **стиль сайту** (рис. 4.63). Важливо, щоб стиль відповідав призначенню сайту, особливостям основної аудиторії, на яку розрахований сайт, був орієнтований на надання найбільших зручностей для сприйняття основного матеріалу.

4. **Створення та верстка сторінок сайту.** Створюються сторінки, як правило, мовою розмітки гіпертексту **HTML**. У процесі створення відбувається верстка сторінок. Як ви вже знаєте, **верстка** – це процес розміщення на сторінці під час її створення текстових і графічних елементів таким чином, щоб сторінка отримала вигляд згідно з розробленим дизайн-макетом.

Правильно зверстана веб-сторінка однаково відображається різними браузерами, швидко завантажується для перегляду, може бути легко змінена, доповнена новими матеріалами. На даному етапі здійснюється інформаційне наповнення сайту.

5. Якщо на етапі постановки завдання передбачалося розміщення на сайті інтерактивних елементів, таких як системи пошуку, голосування, форуму та ін., то потрібен ще й **етап програмування сайту**.
6. **Розміщення (публікація) сайту в Інтернеті.** У ході попередніх етапів створені веб-сторінки могли зберігатися на локальному комп'ютері розробника. На даному етапі сайт розміщують на сервері, який надає послуги **хостингу**. Під час публікації в Інтернеті сайту надається доменне ім'я. Після цього сайт стає доступним для перегляду всіма бажаючими, якщо він або його частина не мають обмежень на доступ.

Організації, що надають послуги хостингу, називають **хостинг-провайдерами**. Існують сервери, які надають безкоштовний хостинг. При цьому, як правило, на вашому сайті буде розміщуватися стороння реклама та обмежуватися використання деяких інтерактивних засобів. Можна розмістити сайт на сервері платного хостингу без реклами та з усіма потрібними програмними засобами.

7. Після публікації сайту в Інтернеті його розробка не вважається завершеною. Певний час буде тривати **тестування сайту** для виявлення недоліків верстки.
8. **Популяризація та підтримка сайту.** Для того щоб ваш сайт почали відвідувати користувачі Інтернету, бажано зареєструвати його у пошукових системах і каталогах, розмістити посилання на нього

на інших сайтах. Цей процес називають **популяризацією, розкручуванням або просуванням сайта**. Для популяризації використовують й інші засоби, але важливо, щоб матеріали, доступні на вашому сайті, були варті того, щоб ними зацікавилися відвідувачі.

9. Для підтримки інтересу до вашого сайта важливо регулярно оновлювати його, доповнювати цікавими унікальними матеріалами. Можливо із часом стане бажаною зміна дизайну сайта – **редизайн**. При виконанні таких робіт кажуть про **супровід сайта**.

Основи веб-дизайну

Під час створення сайтів важливе значення відіграє дизайн. **Веб-дизайн** – це дизайн, об'єктами якого є сторінки веб-сайтів. Як і в кожному напрямку дизайну, для вдалої реалізації завдань веб-дизайну потрібно дотримуватися принципів композиції, колористики та ергономіки, з якими ви ознайомилися у 10 класі під час вивчення теми «Комп'ютерні презентації».

Як уже зазначалося, важливе значення у веб-дизайні має стиль. Дизайн сайта може бути спроектований в одному з наведених стилів (рис. 4.64).

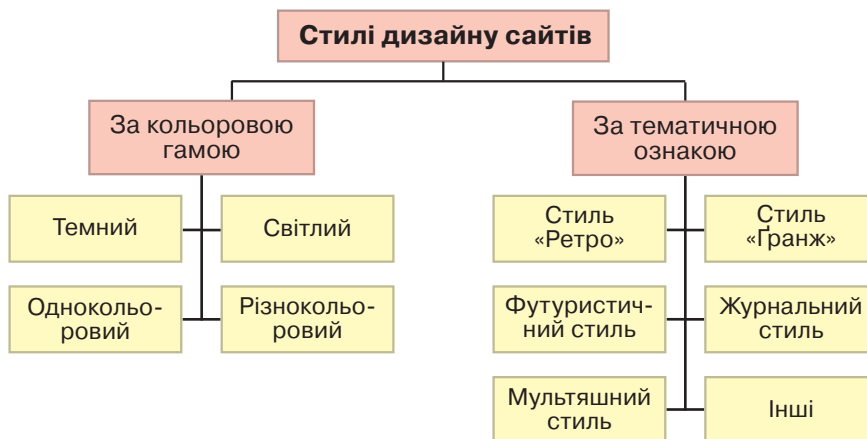


Рис. 4.64. Класифікація стилів дизайну сайта

Вибираючи стиль дизайну за **кольоровою гамою**, варто враховувати особливості сприйняття кольору відвідувачами різного віку та статі:

- **темний** стиль передбачає, що в оформленні сайта переважають темні кольори. Такі сайти добре сприймає молодь, але вони здаються занадто похмурими, тривожними для осіб старшого віку;
- **світлий** стиль з переважаючими світлими кольорами легко сприймається всіма категоріями відвідувачів, але дехто вважає світлі сайти занадто простими;
- **однокольоровий** стиль – це стиль оформлення, у якому переважає один колір, відмінний від чорного та білого. Часто обраний колір є елементом іміджу власника;
- **різнокольоровий** стиль передбачає використання двох або більше кольорів, що займають найбільшу площу на сайті. Різнокольоровий стиль часто обирають у ході створення сайтів для дітей.

Найбільшою є класифікація стилів дизайну за **тематичною ознакою**. Наведемо кілька прикладів стилів дизайну цієї класифікації:

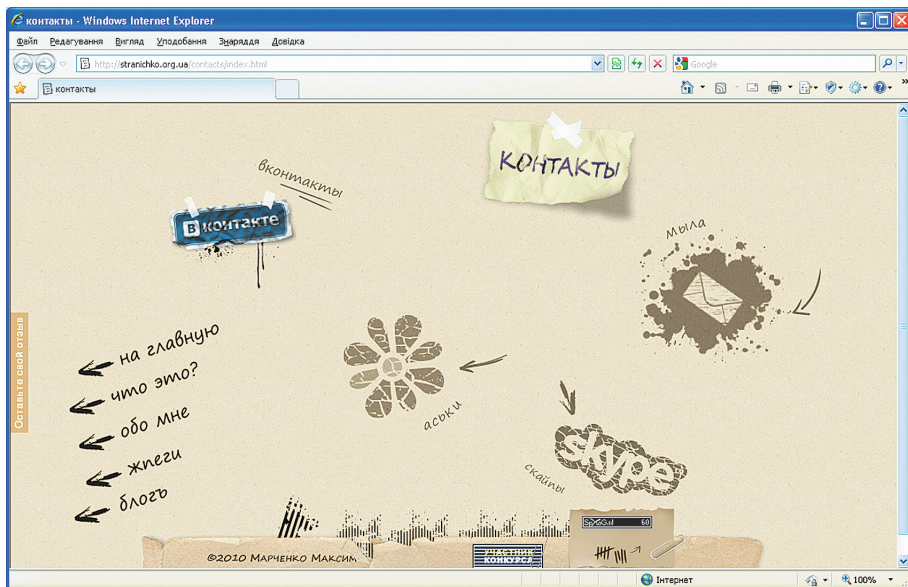


Рис. 4.65. Приклад сайта стилю «Гранж»

- **стиль «Ретро»** – в оздобленні сторінок використовують елементи декору, предмети інтер'єру та інші атрибути, притаманні минулим десятиліттям;
- **стиль «Гранж»** (амер. розмов. *grunge* – дещо неприємне) – асиметричне розміщення елементів на сторінці, яке складає враження хаотичної композиції, виконаної начебто недбало (рис. 4.65);
- **футуристичний стиль** – в оформленні сторінки використовують графічні елементи, що є атрибутами передбачуваного майбутнього: роботи, високотехнологічні механізми тощо;
- **журнальний стиль** – веб-сторінки оформлені за принципами дизайну глянцевого журналу: великі заголовки, крупні високоякісні фотографії та ін.;
- **мультяшний стиль** – оформлення сторінок стилізовано під мультфільми та комікси.

За тематичною ознакою стилів дизайну є набагато більше, оскільки різноманітною є тематика інформаційних матеріалів і кожна тема може знайти своє відображення в дизайні.

Особливе значення під час вибору дизайну відіграє призначення веб-сайта, а не лише його естетичне сприйняття. Будь-який відвідувач сайта завдяки дизайну повинен швидко зрозуміти, чому присвячено сайт, легко знайти корисні для себе інформаційні матеріали, інтуїтивно зорієнтуватися в навігації сторінками сайта. Аналіз найпопулярніших веб-сайтів доводить, що чим простіший дизайн, тим легше користувачу орієнтуватися на сайті. Велика кількість графічних елементів може відволікати увагу від дійсно корисних матеріалів, заради яких відвідувач потрапив на сайт.

З огляду на ці та інші фактори, можна запропонувати такі правила оформлення веб-сторінок:

- Оберіть кольорову гаму для вашого сайта, враховуючи ваші вподобання, тематику сайта та вподобання потенційних відвідувачів. Використайте для оформлення не більше ніж 2–3 кольори.
- У ході визначення кольорової гами сторінки вибирайте контрастні кольори для тексту та фону, щоб текст легше читався.
- Розбивайте текст на абзаци, між якими зробіть збільшені відступи.
- Виберіть розмір шрифту, при якому текст буде сприйматися комфортно – не занадто дрібний і в міру великий. Шрифт на заголовках зробіть більшим від шрифту основного тексту. Вид шрифту зробіть однаковим на всіх сторінках.
- Вирівняйте заголовки по центру, а основний текст – за шириною.
- Для структурування тексту використовуйте таблиці. Розміщуючи фрагменти тексту та графічні зображення у таблицях, можна створити цікаві композиційні рішення на веб-сторінках.
- Не зловживайте флеш-анімацією, відео, музичними та графічними елементами. Вони можуть відволікати увагу від корисних матеріалів та уповільнювати завантаження сторінок.
- Зробіть гіперпосилання для переходу між сторінками сайта, але не розміщуйте на сторінках занадто багато гіперпосилань. Виділяйте гіперпосилання кольором, щоб користувач бачив, що це посилання і які з них він уже відвідував.
- Зробіть логотип сайта, зображення або текст у заголовку гіперпосиланням на головну сторінку сайта.
- Створіть для відвідувачів мапу сайта для спрощення переходу на сторінки з потрібними матеріалами.



Перевірте себе


- 1°. Назвіть етапи розробки веб-сайта.
- 2°. У чому полягає етап постановки завдання під час розробки веб-сайта?
- 3°. Що таке внутрішня структура веб-сайта?
- 4°. Які складові має зовнішня структура сайта?
- 5°. Значення властивостей яких об'єктів вибирають на етапі розробки дизайн-макета сторінок сайта?
- 6°. Що таке верстка? Що є результатом верстки сайта?
- 7°. Що таке хостинг? Кого називають хостинг-провайдерами?
- 8°. У чому полягає етап популяризації та підтримки сайта?
- 9°. Поясніть схему класифікацій стилів дизайну (рис. 4.64).
- 10*. Поясніть, чому під час розробки дизайну сайта важливо враховувати основну аудиторію потенційних відвідувачів.
- 11°. Назвіть відомі вам правила оформлення веб-сторінок.




Виконайте завдання



- 1°. Перегляньте веб-сайти із зазначеними адресами: <http://pets.uz.ua>, <http://palace.dp.ua>, <http://www.odnoklasnyk.org.ua>, <http://smakuje.livejournal.com>. Дайте відповіді на запитання.
 - а) З якою метою створено сайт?
 - б) Яка тематика сайта?
 - в) Який тип сайта?
 - г) Що є оригінального в сайті?
 - д) Хто є потенційними відвідувачами сайта?
 - е) Які сервіси запропоновано на сайті?

- е) Які ідеї переглянутого сайту ви могли б використати під час розробки власного сайту?
- 2°. Перегляньте веб-сайти із зазначеними адресами: <http://www.kazka.in.ua>, <http://www.npblog.com.ua>, <http://maque.org.ua>, <http://neyrat.net>. Відкрийте мапи сайтів і порівняйте форми їхнього подання. Визначте, чи на кожному сайті представлена мапа сайту, чи надає вона додаткові зручності відвідувачам сайту.
- 3°. Перегляньте веб-сайти із зазначеними адресами: <http://platfor.ma>, <http://starodub.org.ua>, <http://molode.com.ua>, <http://www.4uth.gov.ua>, <http://osobystist.com>. Порівняйте схему зовнішньої структури сторінок сайтів. Визначте, чи на всіх сайтах використані блоки заголовка, інформаційний, підвал, як розміщений блок меню.
- 4*. Виберіть одну із запропонованих тем для веб-сайту: *Небезпечні тварини, Космічні дослідження, Колекція рекордів, Наукові відкриття, Художній салон*.
- а) Виберіть тип для сайту цієї тематики.
- б) Створіть у текстовому редакторі схему внутрішньої структури сайту. Збережіть її у власній папці у файлі з іменем **завдання 4.8.4.docx**.
- в) Розробіть у графічному редакторі схему зовнішньої структури сторінок сайту. Збережіть її у власній папці у файлі з іменем **завдання 4.8.4.bmp**.
- 5°. Перегляньте веб-сайт дизайн-студії **ArtPoint** (<http://artpoint.com.ua>). Проаналізуйте зразки робіт, розробки елементів корпоративного стилю. Зверніть увагу на дотримання тематики та кольорової гами. Визначте, використання яких елементів могло б підкреслити єдність стилю сторінок вашого сайту.
-  6°. Перегляньте веб-сторінку сайту **Файнохост** (<http://www.faynohost.com.ua/design-templates.html>) із зразками шаблонів оформлення сайтів. Зверніть увагу на схему зовнішньої структури шаблонів сторінок. Визначте, яку схему ви могли б використати для власного сайту.
- 7°. Перегляньте веб-сайти із зазначеними URL-адресами та визначте, до якої групи класифікації належить стиль дизайну кожного з них.

URL-адреса	Стиль дизайну сайту	
	За кольоровою гамою	За тематичною ознакою
http://www.nbg.kiev.ua		
http://1000plastinok.net		
http://www.vovazilvova.net		
http://fcdynamo.kiev.ua		
http://www.janda.ru		

-  8*. Знайдіть в Інтернеті 3–4 сайти, тематика яких збігається з вибраною вами темою у завданні 4. Оцініть, чи дотримані на цих сайтах правила оформлення веб-сторінок. Запишіть, які ідеї переглянутих сайтів ви могли б використати під час розробки вашого сайту.

4.9. Засоби автоматизованої розробки веб-сайтів



1. Що таке хостинг? Які організації називають хостинг-провайдерами?
2. За якими ознаками можна класифікувати веб-сторінки? Які вам відомі типи веб-сторінок?
3. З яких блоків складається зовнішня структура веб-сторінки?
4. Які вам відомі формати графічних, аудіо- та відеофайлів? Які із цих форматів передбачають стиснення даних?
5. Які засоби використовують для зміни формату файлів?

Засоби розробки веб-сайтів

З 9-го класу вам відомо, що веб-сторінки є текстовими файлами, розширення їх імен надається відповідно до мови розмітки гіпертексту, що використана під час розробки сторінки. Як ви вже знаєте, веб-сторінки часто створюють з використанням мови розмітки гіпертексту **HTML**. У файлів, створених мовою **HTML**, розширення імен файлів **htm** або **html**. У файлах такого формату міститься **HTML-код** сторінки, що складається з даних двох типів:

- тексту, який буде відображатися на сторінці;
- команд, що визначають **розмітку** тексту – його структуру, формат фрагментів тексту, забезпечують вставлення нетекстових об'єктів на сторінку та ін. Такі команди в мові **HTML** називають **тегами** (англ. *tag* – ярлик, ознака).

Переглянути **HTML-код** веб-сторінки, що відкрита у вікні браузера **Internet Explorer**, можна, виконавши **Вигляд** ⇒ **Перегляд HTML-кода**. В інших браузерах ця команда може бути **Вигляд** ⇒ **Початковий код сторінки** (**Mozilla Firefox**), **Меню** ⇒ **Сторінка** ⇒ **Інструменти розробника** ⇒ **Джерело** (**Opera**), **Інструменти** ⇒ **Див. джерело** (**Google Chrome**) та ін.

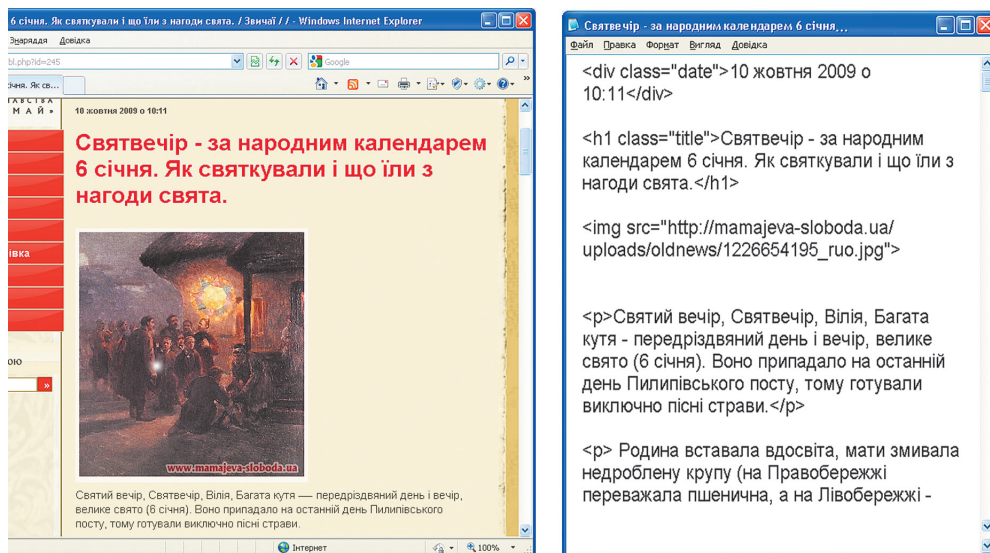


Рис. 4.66. Фрагмент веб-сторінки та відповідний фрагмент HTML-коду

На рисунку 4.66 наведено зображення частини веб-сторінки сайту та відповідний фрагмент HTML-коду.

Розробка веб-сторінок може здійснюватися з використанням різних засобів:

- **текстових редакторів**, у середовищі яких користувач може вводити текст і теги. Для цього можна використовувати текстові редактори **Блокнот**, **Edit Plus**, **Homesite**, **HTML Pad** та ін. Готову сторінку потрібно зберегти та надати розширення імені файлу **htm** або **html**. Створення веб-сторінки такими засобами передбачає обов'язкове знання мови розмітки гіпертекстів, наприклад **HTML**. Усю роботу з добору тегів розробник виконує власноруч. Розробка сторінки здійснюється повільно, але завдяки невеликому розміру файлу, у якому вона зберігається, така сторінка швидко завантажується та відкривається у вікні браузера;
- **прикладних програм загального призначення**, наприклад з пакета **Microsoft Office**, які можуть зберігати файли у форматі **HTML**. Як ви вже знаєте, документи, створені в програмах **Word 2007**, **PowerPoint 2007**, **Publisher 2007** та інших, можна зберігати, вибравши тип файлу *Веб-сторінка (*.htm, *.html)*. При цьому створюються веб-сторінки, у яких застосування тегів для розмітки відбувається автоматично. Розробнику не потрібно знати мову розмітки гіпертексту. Але файли, у яких зберігаються такі сторінки, мають набагато більший обсяг, ніж файли, створені в текстових редакторах. Причиною є велика кількість тегів, які описують структуру і форматування документа та дублюються під час застосування до кожного окремого елемента сторінки;
- **спеціалізованих веб-редакторів** – програм, що призначені для розробки веб-сайтів. Популярними веб-редакторами є **Adobe Dreamweaver**, **Microsoft FrontPage**, **SharePoint Designer**, **WYSIWYG Web Builder**, **KompoZer** та ін. Ці програми мають додаткові засоби для створення статичних і динамічних веб-сторінок, при цьому не вимагають від розробника знання мови **HTML**. Такі програми називають **WYSIWYG-редакторами** (англ. *What You See Is What You Get* – що ви бачите, то ви й отримуєте), створена їхніми засобами веб-сторінка виглядатиме так, як вона сконструйована в редакторі. **HTML**-код сторінки більш коректний, ніж під час використання, наприклад програм пакета **Microsoft Office**, але теж надлишковий;
- **систем управління веб-контентом WCMS** (англ. *Web Content Management System* – система управління веб-контентом), які надають користувачам зручні інструменти для керування текстовим і графічним наповненням веб-сайтів, додавання та видалення статей з інформаційними матеріалами, створення системи навігації веб-сайтів та ін. Популярними **WCMS** є системи **Joomla**, **Wordpress**, **Drupal**, **MediaWiki**, **Mambo**, **NUKE** та ін. Системи управління вмістом веб-сайтів пропонують набори шаблонів оформлення веб-сторінок і модулів, що роблять сайт динамічним: форумів, чатів, стрічок новин, каталогів файлів, контролю статистики тощо.

Названі засоби створення веб-сайтів можна встановити на локальному комп'ютері. Створені в їх середовищі веб-сторінки після завершення процесу розробки потрібно опублікувати в Інтернеті. У той самий час існують **WCMS**, які одночасно з послугами з розробки веб-сайтів надають по-

слуги безкоштовного хостингу. Такими є системи **uCoz**, **Google Sites**, **Prom.ua**, **Ua7.biz** та ін. У цих системах створення веб-сайта здійснюється в режимі он-лайн відразу на сервері хостингу.

Автоматизоване створення веб-сайта

Створення веб-сайта засобами он-лайн системи керування веб-контентом відбувається в кілька кроків:

1. Реєстрація облікового запису на сервері.
2. Вибір імені сайта та шаблону для його оформлення.
3. Створення сторінок сайта, системи навігації.
4. Заповнення сторінок контентом.

Розглянемо, як відбувається процес розробки веб-сайта засобами, що безкоштовно надає користувачам веб-сервер **Google**. Відповідний сервіс має назву **Сайти Google**.

Перш ніж розробляти сайт, потрібно створити акаунт **Google**. Із цією метою:

1. Відкрийте у вікні браузера головну сторінку сайта **Google** (<http://www.google.com.ua>).
2. Виберіть гіперпосилання **Увійти** ⇒ **Створити акаунт зараз**.
3. Заповніть поля форми **Створити акаунт** на сторінці **Облікові записи Google**. Виберіть кнопку **Я погоджуюся. Створіть мій акаунт**.
4. Активуйте ваш акаунт, використавши гіперпосилання у тексті листа, що надійде до електронної поштової скриньки, на яку ви зареєстрували ваш акаунт.
5. Виберіть гіперпосилання **Увійти** на веб-сторінці **Акаунт Google** у вікні браузера, що відкриється після вибору гіперпосилання з електронного листа.
6. Заповніть поля **Електронна пошта** та **Пароль** даними, які ви вводили під час створення облікового запису.
7. Виберіть кнопку **Увійти**.
8. Виберіть гіперпосилання **Домашня сторінка Google** у нижній частині веб-сторінки.

У результаті відкриється головна сторінка сервера **Google**, на якій ви матимете права користувача сервісів **Google**. Адреса, на яку зареєстровано ваш обліковий запис, відображатиметься у верхній частині веб-сторінки.

Для створення сайта потрібно:

1. Відкрити у вікні браузера головну сторінку сайта **Google** (<http://www.google.com.ua>).
2. Вибрати у верхній частині сторінки гіперпосилання **Ще** ⇒ **Сайти**.
3. Заповнити на сторінці **Google Sites** – **безкоштовні веб-сторінки та вікі** поля **Електронна пошта** та **Пароль** даними вашого облікового запису, якщо вони не заповнені. Вибрати кнопку **Увійти**.
4. Вибрати на сторінці **Сайти Google** кнопку **Створити сайт**.
5. Вибрати один з наведених шаблонів для створення сайта. За замовчуванням пропонується шаблон *Пустий шаблон*.
6. Заповнити поле **Дати назву сайта**. Наприклад, якщо ви створюєте сайт вашого класу, то назва може бути *Класне життя*. Одночасно автоматично буде запропонована URL-адреса головної сторінки сайта. Вона матиме вигляд: https://sites.google.com/site/<назва_сайта>. Частина **назва_сайта** не повинна містити літер кирилиці.

ці, а тому система пропонує запис українських слів літерами англійського алфавіту без пропусків. Для вказаної назви сайту системою буде запропонована адреса **https://sites.google.com/site/klasnezitta**, але за бажанням її можна змінити. URL-адреса головної сторінки сайту повинна бути унікальною.

7. Відкрити список **Вибрати тему** та вибрати тему оформлення сайту. За замовчуванням пропонується тема *Запустити за умовчанням*. Наприклад, виберемо тему *Бейсбол*.
8. Увести символи в поле **Введіть код, який Ви бачите на малюнку**.
9. Вибрати кнопку **Створити сайт**.

У вікні браузера відкриється домашня сторінка вашого сайту. У верхній частині сторінки буде відображатися вказана вами назва сайту, зліва – панель навігації, у центральній частині – область для інформаційного блока сторінки із заголовком **Домашня сторінка** (рис. 4.67).

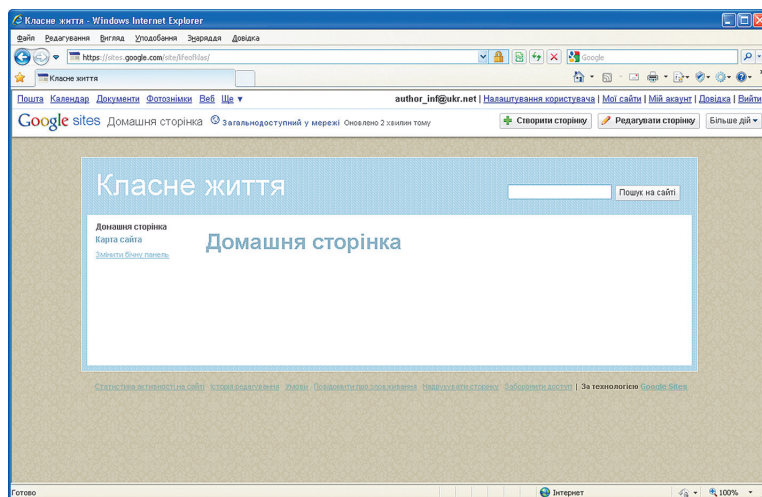


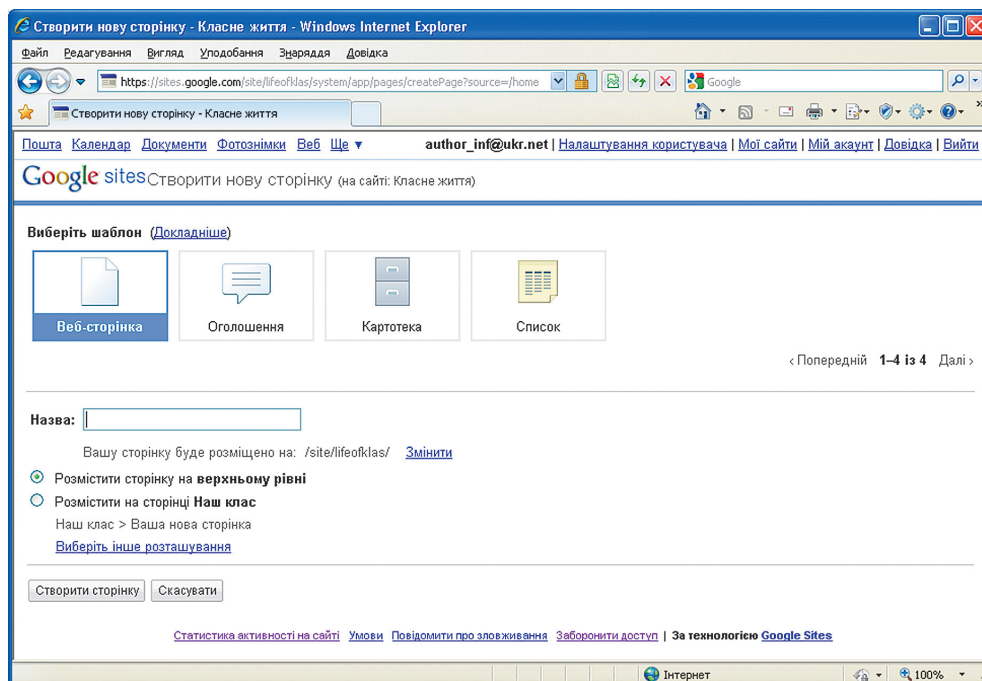
Рис. 4.67. Вигляд домашньої сторінки щойно створеного сайту

Панель навігації містить два гіперпосилання: **Домашня сторінка** та **Карта сайту**. Карта сайту на цей момент містить посилання лише на домашню сторінку.

Створення та налаштування веб-сторінок

Відразу після створення сайт містить лише одну домашню сторінку. Для створення нової сторінки на сайті потрібно:

1. Вибрати кнопку **Створити сторінку**  **Створити сторінку** у верхній частині вікна браузера.
2. Вибрати шаблон вмісту майбутньої сторінки (рис. 4.68):
 - **Веб-сторінка** – сторінка для розміщення тексту, зображень тощо. Такі сторінки є інформаційними сторінками тематичних розділів.
 - **Оголошення** – сторінка для розміщення текстових повідомлень, впорядкованих у хронологічному порядку, починаючи з останніх введених. Кожне повідомлення може відображатися як окрема інформаційна сторінка.

Рис. 4.68. Сторінка **Створити нову сторінку**

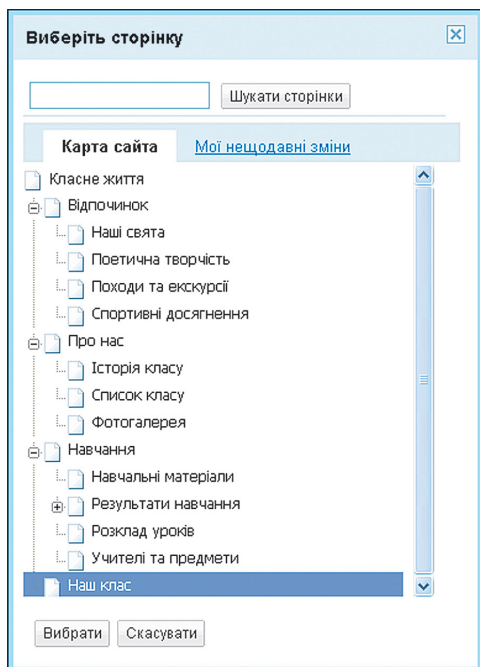
- **Картотека** – сторінка для збереження гіперпосилань на завантажені файли. Гіперпосилання згруповані відповідно до імен папок, у яких розміщені файли. Такі сторінки є сторінками-контейнерами, що містять каталоги файлів.
- **Список** – сторінка, на якій подаються структуровані дані як списки з кількох полів. Списки можна сортувати за даними в кожному полі.

Наприклад, для створення сторінки *Історія класу* сайту *Класне життя* (рис. 4.68) можна вибрати шаблон **Веб-сторінка**, для сторінки *Учителі та предмети* – шаблон **Список**, *Поетична творчість* – шаблон **Оголошення**, *Навчальні матеріали* – шаблон **Картотека**.

3. Увести назву сторінки в поле **Назва**.

4. Вибрати розміщення сторінки в структурі сайту. Можна вибрати варіанти **Розмістити сторінку на верхньому рівні**, **Розмістити на сторінці** <ім'я сторінки> або **Вибрати інше розташування**. У першому випадку гіперпосилання на сторінку буде розміщено в головному меню сайту, у другому – воно з'явиться на вибраній сторінці. За вибору гіперпосилання **Вибрати інше розташування** відкривається панель **Виберіть сторінку** (рис. 4.69), на якій слід вибрати сторінку, з якою буде пов'язана нова сторінка.

Наприклад, сторінки *Про нас*, *Навчання* та *Відпочинок* можна розмістити на верхньому рівні, сторінки *Історія класу*, *Список класу* та *Фотогалерея* пов'язати зі сторінкою *Про нас*; *Учителі та предмети*, *Розклад уроків*, *Результати навчання*, *Навчальні матеріали* – зі сторінкою *Навчання* тощо.

Рис. 4.69. Панель **Виберіть сторінку**

При цьому слід вибрати для завантаження на сайт файл з локального комп'ютера. Для впорядкованого зберігання файлів на сайті можна створювати папки. Для цього використовують кнопку **Перемістити до**



На сторінці із шаблоном **Список** потрібно вибрати один із запропонованих шаблонів списку або створити нетиповий список (рис. 4.70), вибравши відповідну кнопку та вказавши назву і тип даних кожного стовпця.

Команди зі списку кнопки **Більше дій** **Більше дій** ▾, що розміщена у верхній частині вікна браузера, призначені для додаткових налаштувань: зміни шаблону сторінки, видалення сторінок, керування доступом до матеріалів сайта та ін.

Панель навігації та карта веб-сайта формуються автоматично під час створення нових сторінок. Назви сторінок на панелі навігації розміщу-

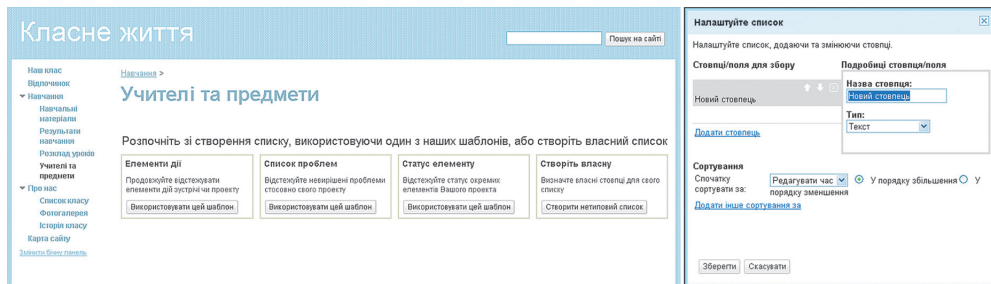
5. Вибрати кнопку **Створити сторінку**.

Після створення сторінки вона відкривається в режимі редагування, а панель навігації та мапа сайта автоматично доповнюються посиланнями на нову сторінку.

Під час створення сторінки на основі шаблону **Веб-сторінка** її потрібно заповнити матеріалами та зберегти. Це статична сторінка, на якій не передбачається часто змінювати наповнення.

Сторінка на основі шаблону **Оголошення** оновлюватиметься регулярно з появою деяких новин. Для додавання нового оголошення на сторінці слід вибрати кнопку **Новий запис** **Новий запис**, увести текст повідомлення та зберегти сторінку.

Сторінку на основі шаблону **Картотека** заповнюють, вибираючи кнопку **Додати файл** **Додати файл** **Додати файл**.

Рис. 4.70. Вигляд сторінки, створеної на основі шаблону **Список**, і панелі налаштування списку

ються в алфавітному порядку. Розміщення сторінок можна змінювати, виконавши *Карта сайта* \Rightarrow *Керування сторінками* та перетягнувши у схемі сайта заголовок однієї сторінки на заголовок іншої, з якою потрібно пов'язати сторінку.

Редагування веб-сторінок


Створену сторінку веб-сайта можна редагувати, наповнювати її інформаційними матеріалами, змінювати модульну сітку тощо. Для переходу до режиму редагування сторінок потрібно вибрати кнопку **Редагувати сторінку**  у верхній частині вікна браузера. Після цього у вікні браузера з'являється меню та панель інструментів (рис. 4.71).




Рис. 4.71. Меню та панель інструментів редактора веб-сторінок

Меню містить команди, призначені для виконання операцій з елементами веб-сторінки:

- **Вставити** – для вставлення об'єктів на сторінку;
- **Формат** – для форматування тексту на веб-сторінці;
- **Таблиця** – для вставлення та редагування таблиць;
- **Компонування** – для вибору модульної сітки інформаційного блока сторінки: в один, два, три стовпці, з лівою або правою бічною панеллю та ін. (рис. 4.72).

У режимі редагування робоча область сторінки містить поля заголовка та інформаційного блока, у які можна вводити текст з клавіатури або вставляти з **Буфера обміну**. Під час зміни заголовка сторінки його текст одночасно відобразиться на панелі навігації. Текст у інформаційному блоці сторінки можна формувати, використовуючи елементи керування панелі інструментів.

У ході редагування сторінки час від часу виконується автоматичне збереження чернетки. Після закінчення редагування зміни потрібно зберегти, вибравши кнопку **Зберегти**  у верхній частині сторінки.

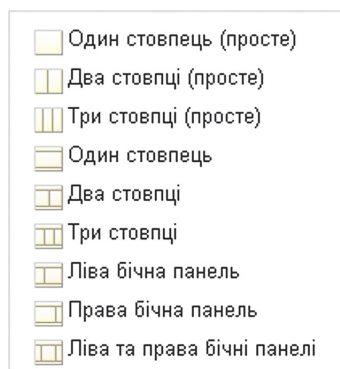


Рис. 4.72. Варіанти компонування веб-сторінки

Вставлення об'єктів на сторінку

На веб-сторінку можна вставити різні об'єкти: зображення, гіперпосилання, списки веб-сторінок, горизонтальні лінії, документи, що створені службами **Google**, та ін.

Під час вставлення зображення на веб-сторінку варто враховувати формати файлів зображень. З особливостями форматів графічних файлів ви ознайомилися в 9-му класі. Фотографії для розміщення на веб-сторінках найчастіше зберігають у файлах формату **JPG**. Анімовані зо-

браження, як правило, містяться у файлах формату **GIF**. Їх часто розміщують на веб-сайтах для надання емоційного забарвлення сторінці. Спеціально для розміщення растрових зображень у мережі був розроблений формат **PNG**. Якщо зображення, що потрібне вам для розміщення на веб-сторінці, зберігається у файлі іншого формату, то його варто конвертувати в один з названих, використовуючи засоби графічного редактора.

Для вставлення зображення на веб-сторінку потрібно:

1. Виконати **Вставити** \Rightarrow **Зображення**.
2. У вікні **Додати зображення** вибрати джерело зображення:
 - **Завантажені зображення** – вставити зображення, що вже розміщено на сайті, або вставити зображення з файлу, що міститься на локальному комп'ютері, вибравши кнопку **Вибрати файл**;
 - **Веб-адреса (URL)** – вставити зображення, що міститься за вказаною URL-адресою в Інтернеті.
3. Вибрати потрібний файл із зображенням або ввести його URL-адресу.
4. Вибрати кнопку **ОК**.

За вибору вставленого зображення під ним відкривається панель редагування, яка містить команди розміщення та встановлення розміру (рис. 4.73). Використовуючи гіперпосилання на панелі, можна розмістити зображення: **L** – за лівим краєм, **C** – по центру, **R** – за правим краєм. Можна встановити один з розмірів зображення: **S** – маленький, **M** – середній, **L** – великий або **Оригінальний**.

Вставлене на веб-сторінку зображення автоматично пов'язується гіперпосиланням з файлом, у якому міститься зображення. За вибору цього гіперпосилання відповідне зображення в повному розмірі відкривається у вікні браузера. Для зміни об'єкта для переходу слід використати гіперпосилання **Змінити** на панелі редагування в рядку **Перейти за посиланням**.

Використовуючи меню **Вставити** (рис. 4.74), на веб-сторінку можна вставити об'єкти, створені з використанням сервісів **Google** (карти, календарі, документи, презентації, електронні таблиці, форми для опитування)

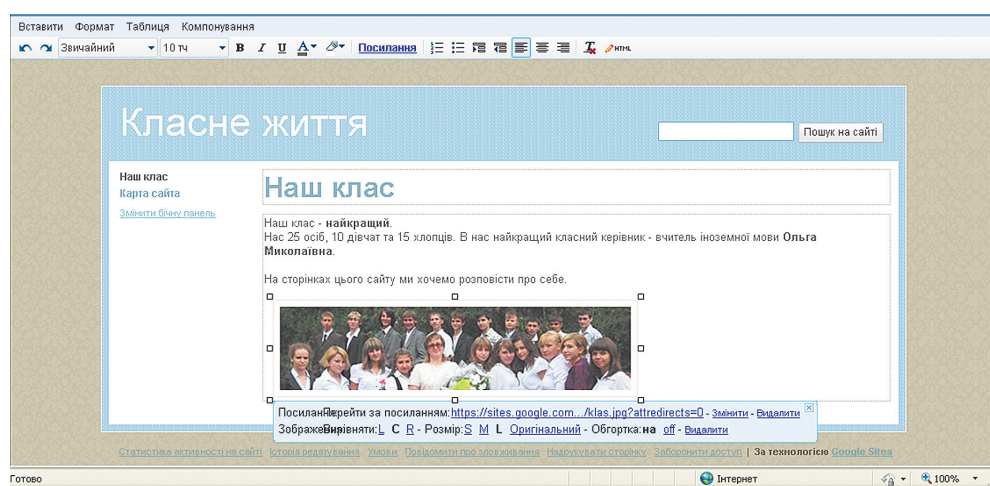


Рис. 4.73. Вставлення зображення на веб-сторінку

та фото сервісу **Picasa** (фотографії та слайд-шоу). Усі ці об'єкти вставляються на сторінку з використанням **гаджетів** (англ. *gadget* – засіб, пристосування) – невеликих програм, що розміщуються на веб-сторінках і призначені для відтворення деяких специфічних даних.

Відео можна вставляти на веб-сторінку з одного або з двох веб-ресурсів: **Відео Google** або **YouTube**. Для цього потрібно виконати **Вставити** ⇒ **Відео**, вибрати джерело відеоматеріалів, вставити URL-адресу відеофрагмента та вибрати кнопку **Зберегти**. На веб-сторінку буде вбудована панель гаджета із засобами відображення відео.

Гіперпосилання на веб-сторінку із цього сайту або на інший ресурс можна вставити, виконавши **Вставити** ⇒ **Посилання** і вибравши у вікні **Створити посилання**, що відкрилося, об'єкт, на який здійснюватиметься перехід за вибору гіперпосилання. При цьому створене посилання буде пов'язане з назвою сторінки або URL-адресою ресурсу. Гіперпосилання також можна пов'язати з будь-якими текстовими фрагментами або зображеннями, що містяться на веб-сторінці. Для цього потрібно виділити фрагмент, вибрати кнопку **Посилання** [Посилання](#) на панелі інструментів і вибрати об'єкт для переходу – існуючу сторінку цього сайту або URL-адресу іншого ресурсу.

Файли різних форматів (флеш-анімація, звук та ін.), для яких може бути недоступним вставлення на веб-сторінки, можна завантажити на сайт. Завантажені файли зберігаються на сервері, а на веб-сторінці в розділі **Вкладені файли** розміщуються гіперпосилання, вибравши які можна зберегти файл на локальному комп'ютері, видалити або переглянути, якщо формат файлу збігається з форматом документів **Google**. Для завантаження файлу на сервер потрібно вибрати в нижній частині сторінки гіперпосилання **Вкладені файли**, вибрати кнопку **Вибрати файл** і вибрати у вікні **Відкриття файлу** потрібний файл на локальному комп'ютері. Після вибору кнопки **Відкрити** файл автоматично завантажується на сервер, його ім'я та відповідні гіперпосилання **Видалити**, **Перегляд**, **Завантажити** відображаються в нижній частині веб-сторінки. Зауважимо, що на домашню сторінку завантаження файлів не можливе.

За вибору файлів для завантаження потрібно звертати увагу на їх розміри та перед завантаженням здійснювати конвертування аудіо- та відео-файлів в один з форматів, який передбачає стиснення даних, наприклад **MP3** – для аудіофайлів, **AVI**, **MP4** – для відеофайлів.

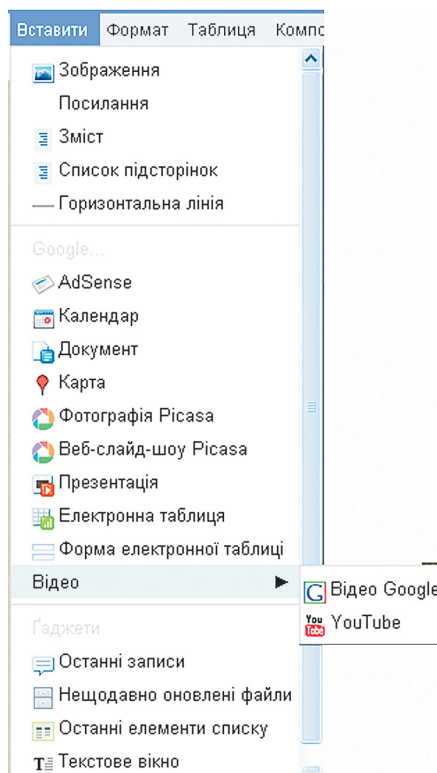


Рис. 4.74. Меню **Вставити**



Перевірте себе

- 1°. Які засоби можна використовувати для розробки веб-сторінок? Поясніть їх переваги та недоліки.
- 2°. Що таке HTML-код сторінки? Дані яких видів він містить? Як можна його переглянути?
- 3°. Назвіть відомі вам веб-редактори. У чому полягають особливості їх використання?
- 4°. Що таке система управління веб-контентом? Назвіть відомі вам системи управління вмістом веб-сайта.
- 5°. Які системи управління веб-контентом, що працюють у режимі он-лайн вам відомі? У чому їх особливості?
- 6°. Назвіть і поясніть етапи автоматизованого створення веб-сайтів засобами веб-серверів.
- 7°. Які шаблони веб-сторінок використовуються під час створення веб-сайтів засобами сервісу **Сайти Google**? Опишіть їх особливості.
- 8°. Як вставити зображення на веб-сторінку? Що може бути джерелом зображення під час вставлення на веб-сторінку?
- 9°. Як пов'язати гіперпосилання з текстовим фрагментом?
- 10°. Як вставити відеооб'єкт на веб-сторінку? Що може бути джерелом відео?
- 11°. Що таке гаджет?
- 12°. Які формати графічних, відео- та аудіофайлів рекомендується використовувати під час розміщення об'єктів на веб-сторінках? Чим це пояснюється?
- 13°. Чим відрізняються результати операцій вставлення графічних зображень та їх завантаження?



Виконайте завдання

- 1°. Перегляньте вказані учителем HTML-сторінки, наприклад з папки **Тема 4\Завдання 4.9\Вправа 1**, які створені різними засобами. Порівняйте значки файлів веб-сторінок, їх вміст у вікні браузера, HTML-код і розміри файлів. Заповніть таблицю.

<i>Засіб створення</i>	<i>Ім'я файлу</i>	<i>Розмір файлу, у байтах</i>
Блокнот	вправа 4.4.1.1.html	
Microsoft Word 2007	вправа 4.4.1.2.html	
KompoZer	вправа 4.4.1.3.html	
Сайти Google	вправа 4.4.1.4.html	

Зробіть висновок, який засіб створення веб-сторінок найекономічніший щодо трафіку, що витрачається під час завантаження сторінки.



- 2*. Відкрийте документи, створені з використанням різних програм пакета **Microsoft Office**, наприклад з папки **Тема 4\Завдання 4.9\Вправа 2**. Збережіть ці документи в папці **Мої документи** з тими самими іменами, вибравши тип файлу *Веб-сторінка (*.htm, *.html)*. Зверніть увагу на значки збережених файлів і створення додаткових папок. Відкрийте файли веб-сторінок, порівняйте їх вміст та оформлення з вмістом відповідних документів. Оцініть розміри файлів веб-сторінок і додаткових папок.
- 3*. Заповніть таблицю та порівняйте особливості створення веб-сайтів з використанням різних засобів.

Засіб створення	Потреба в знанні мови HTML	Наявність засобів для оформлення сторінок та наповнення їх вмістом	Зручність редагування	Потреба в розміщенні на сервері хостингу
Текстовий редактор				
Програма пакета Microsoft Office				
Веб-редактор				
Система управління вмістом веб-сайта				

- 4°. Зареєструйте свій акаунт на сервері **Google** відповідно до алгоритму, описаному в пункті.
- 5°. Створіть на сервері **Google** сайт відповідно до структури, наведеній на рисунку 4.61. Виберіть тему оформлення *Земля: вода*. Виберіть такі шаблони сторінок: *Історія класу* – шаблон *Веб-сторінка*, *Учителі та предмети* – шаблон *Список*, *Поетична творчість* – шаблон *Оголошення*, *Навчальні матеріали* – шаблон *Картоотека*. Шаблони інших сторінок виберіть самостійно. Повідомте вчителю URL-адресу вашого сайта.
- 6°. Створіть на сервері **Google** сайт з темою *Художній салон*:
 1. Доберіть URL-адресу для сайта, відповідну його назві.
 2. Виберіть тему оформлення *Медісон*.
 3. Перегляньте запропоновану схему внутрішньої структури сайта, наприклад з файлу **Тема 4\Завдання 4.9\Вправа 6\схема.docx**.
 4. Заповніть *Домашню сторінку* даними про призначення сайта, описом матеріалів, які будуть розміщені на сторінках, і даними про вас як розробника сайта.
 5. Створіть веб-сторінки на основі шаблону *Веб-сторінка*, розмістіть їх відповідно до наданої схеми внутрішньої структури.
 6. Розмістіть на сторінках текстові та графічні матеріали, наприклад з папки **Тема 4\Завдання 4.9\Вправа 6**, таким чином, щоб текст і відповідне зображення на сторінці були розміщені поруч. Використайте для компонування сторінок модульну сітку *Два стовпці (просте)*.
 7. Пов'яжіть гіперпосилання з текстовими та графічними об'єктами так, щоб, вибравши зображення, можна було відкрити одну з інформаційних сторінок сайта, а вибравши текстовий фрагмент – іншу сторінку.
 8. Завантажте на сторінки сайта файли, що містяться у папці **Тема 4\Завдання 4.9\Вправа 6\Файли**.
 9. Надішліть учителю інформатики електронного листа з адресою створеного сайта.
- 7°. Відкрийте свій сайт, створений під час виконання завдання 6, і відредагуйте його таким чином:
 1. Додайте сторінку *Події* на основі шаблону *Оголошення*. Додайте на сторінку запис з оголошенням про виставку сучасних художників, що відбудеться у художньому салоні.
 2. Вставте на домашню сторінку таблицю, заповніть її розкладом роботи художнього салону.
 3. Скомпонуйте вміст інформаційних сторінок, вибравши модульну сітку *Два стовпці*. У підвалі сторінок зазначте ваші дані як розробника сайта.



8*. Створіть на сервері **Google** сайт про свою майбутню професію.

9*. Використовуючи засоби одного із спеціалізованих веб-редакторів розробіть свій персональний веб-сайт, розмістіть його на сервері безкоштовного хостингу **TopUa** (<http://www.topua.net>).

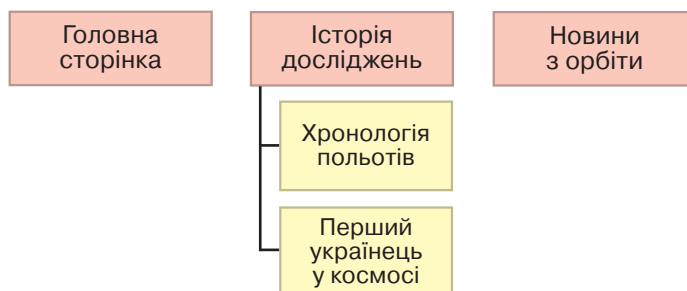


Практична робота № 18. Автоматизоване створення веб-сайта

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

Створіть на сервері **Google** сайт на тему *Українські дослідники космосу*:

1. Доберіть URL-адресу для сайта, відповідну його назві.
2. Виберіть тему оформлення *Мерехтіння*.
3. Заповніть *Головну сторінку* описом матеріалів, які будуть розміщені на сторінках сайта, та даними про вас як розробника сайта.
4. Створіть і розмістіть веб-сторінки на основі такої схеми внутрішньої структури:



Виберіть для сторінок такі шаблони: *Історія досліджень*, *Перший українець у космосі* – шаблон *Веб-сторінка*, *Новини з орбіти* – шаблон *Оголошення*, *Хронологія польотів* – шаблон *Картотека*.

5. Заповніть сторінку *Історія досліджень* текстовими матеріалами, наприклад з файлу **Тема 4\Практична 18\космічні дослідження.docx**.
6. Створіть на сторінці *Історія досліджень* гіперпосилання для переходу на сторінку *Новини з орбіти*, пов'язавши його з першим реченням на сторінці.
7. Розмістіть на сторінці *Перший українець у космосі* текстові та графічні матеріали, наприклад з папки **Тема 4\Практична 18\Попович**, таким чином, щоб фотографія розміщувалася по центру сторінки, а текст – ліворуч і праворуч від неї. Використайте для компонування сторінки модульну сітку *Три стовпці (просте)*.
8. Створіть на сторінці *Перший українець у космосі* гіперпосилання для переходу на сторінку з URL-адресою <http://h.ua/story/285307>, пов'язавши його з графічним зображенням.
9. Налаштуйте на сторінці *Хронологія польотів* нетиповий список так, щоб у ньому містилися про кожний політ такі дані: рік польоту, космонавт, космічний корабель, опис. Заповніть список даними про 3–4 польоти, наприклад з файлу **Тема 4\Практична 18\польоти.docx**.
10. Розмістіть на сторінці *Новини з орбіти* запис з останньою новиною із сайта <http://space.vn.ua>.
11. Завантажте на сторінку *Новини з орбіти* сайта файли, що містяться в папці **Тема 4\Практична 18\Файли**.
12. Повідомте учителю інформатики URL-адресу вашого сайта.

4.10. Технології та сервіси Веб 2.0. Веб-спільноти. Створення блогів



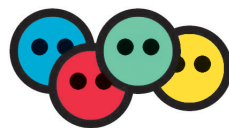
1. Що таке гаджети?
2. Як створюються та редагуються статті веб-енциклопедії **Вікіпедія**?
3. Як можна знайти потрібну статтю у веб-енциклопедії **Вікіпедія**?
4. Що таке форум і чат? Яку структуру мають форуми?
5. Як створити повідомлення на форумі? Повідомлення електронної пошти?

Огляд технологій і сервісів Веб 2.0

Останнім часом швидко розвиваються сервіси Інтернету, які отримали узагальнену назву сервіси **Веб 2.0**.

Терміном **Веб 2.0** позначають кілька технологій, на основі яких розробляються сучасні веб-ресурси. Такими технологіями є:

- **AJAX** (англ. *Asynchronous JavaScript and XML* – асинхронний JavaScript і XML, мови програмування та розмітки веб-сторінок) – технологія побудови веб-сторінок, під час використання якої можливе динамічне перезавантаження частини вмісту веб-сторінки у ході перегляду її у вікні браузера;
- **RSS** (англ. *Really Simple Syndication* – дійсно просте об'єднання) – технологія експорту гіпертексту з одних ресурсів на інші, що використовується для отримання узагальнених коротких повідомлень із сайтів, які цікавлять користувача – **стрічок новин**;
- **FOAF** (англ. *Friend of a Friend* – один одного) – технологія, що забезпечує користувачам можливість оформлювати підписку на отримання нових матеріалів від тих користувачів, дані яких занесені до так званих *списків друзів*;
- **трекбек** (англ. *track back* – шлях назад) – технологія надсилання повідомлення на сервер про встановлення гіперпосилання на деякий ресурс цього сервера та автоматичного формування зворотного гіперпосилання та ін.



Класичні веб-ресурси, які тепер називають **Веб 1.0** (статичні веб-сторінки, форуми, чати та ін.), функціонують на основі того, що розробляє ресурс одна особа – розробник, а користується інша – користувач, наповнює ресурс інформаційними матеріалами автор, а користувач є лише читачем цих матеріалів. На інформаційні матеріали розповсюджується авторське право. Єдиний спосіб отримання цих матеріалів – відкрити сторінки веб-сайта, на якому цей матеріал розміщено. Ресурси Інтернету використовуються в основному як джерело інформаційних матеріалів.

На відміну від ресурсів **Веб 1.0**, сервісам **Веб 2.0** притаманні такі особливості:

- **веб як платформа** – ресурси **Веб 2.0** використовуються для створення документів (презентацій, електронних таблиць, флеш-проектів тощо), які можна зберегти як на локальному комп'ютері, так і он-лайн. Багато ресурсів **Веб 2.0** розробляються як програми, що працюють на веб-сторінках. Вони отримали назву **веб-додатки**. Їх розміщення в мережі звільняє користувача від потреби встановлення відповідних програм на локальний комп'ютер. Прикладом є використання он-лайн словни-

ків і перекладачів, доступ до яких можна отримати не лише з комп'ютера, а й з мобільних пристроїв (комунікаторів, смартфонів та ін.), що мають доступ до мережі;

- **mash-up** (англ. *mash-up* – змішування) – функціональність ресурсів Інтернету розширюється за рахунок додавання нових сервісів до вже існуючих, попередні розробки знаходять застосування в нових ресурсах. Так, на багатьох веб-сторінках можна побачити гаджети для відтворення відео, при цьому самі відеофрагменти та механізми керування ними містяться на інших веб-ресурсах;
- **«колективний розум»** – користувачі беруть активну участь у створенні та оцінюванні контенту ресурсу, спільно реалізують певні проекти, а не лишаються пасивними споживачами чужого контенту. Прикладом є веб-енциклопедія **Вікіпедія**, вміст якої створюється та покращується спільними зусиллями всіх зацікавлених користувачів;
- **фолксономія** (англ. *folk* – народний, *taxonomy* – таксономія, принцип упорядкування) – народна класифікація, практика колективного розподілу даних за категоріями з використанням ключових слів – тегів, які користувачі ресурсу добирають самостійно для позначення вмісту контенту;
- **веб-синдикація** – ресурси Веб 2.0 динамічні, окремі елементи вмісту мають власні URL-адреси, а тому на них можна встановити гіперпосилання та написати коментар. За рахунок використання технологій **RSS** та **FOAF** для перегляду оновленого вмісту сайтів, на які оформлена підписка, або ресурсів користувачів зі списку друзів можна не відвідувати самі сайти, а слідкувати за їх змінами з використанням стрічок новин;
- **соціалізація** – використання сервісів Веб 2.0 сприяє створенню спільнот у мережі з метою обговорення деяких питань і спільної роботи над проектами, при цьому можливість виконання індивідуальних налаштувань, створення власного профілю користувача вирізняє кожну особистість поміж інших

та ін.

Як наслідок соціалізації сервіси Веб 2.0 отримали назву **соціальних**. Призначенням цих сервісів є створення умов для організації спільної діяльності та для спілкування користувачів Інтернету.

Соціальні сервіси можна класифікувати за видами діяльності користувачів у мережі. Назвемо окремі види діяльності та відповідні сервіси Веб 2.0 (табл. 4.7).

Таблиця 4.7. Приклади сервісів Веб 2.0 за видами діяльності користувачів

Вид діяльності	Сервіси Веб 2.0, приклади ресурсів	Опис сервісу
Пошук даних	Соціальні пошукові системи: Флексум (http://flexum.ru), Open Directory Project (http://www.dmoz.org), Квінтюра (http://quintura.ru)	Користувачі заповнюють пошукові бази даних адресами та описом цікавих ресурсів Інтернету, позначають їх тегами, створюючи перелік ресурсів, у межах якого можна здійснювати пошук. Сформовані списки ресурсів для пошуку можуть розміщуватися на веб-сторінках для здійснення пошуку потрібних користувачу даних лише з ресурсів запропонованого списку

Продовження таблиці 4.7

<i>Вид діяльності</i>	<i>Сервіси Веб 2.0, приклади ресурсів</i>	<i>Опис сервісу</i>
Створення колективних каталогів посилань	Народні класифікатори (служби соціальних закладок): БобрДобр (http://bobrdobr.ru), Делішес (http://delicio.us), Memori.ru (http://memori.ru)	Користувачі створюють закладки на різні ресурси Інтернету, позначаючи їх тегами. За потреби знайти ресурс, що містить матеріал з певної теми, достатньо на сайті сервісу вибрати відповідні теги з повного переліку, і з позначених ресурсів буде побудовано список посилань
Спільна розробка нових інформаційних матеріалів	Вікі-проекти: ВікіОсвіта (http://www.eduwiki.urau.net.ua), Веб-енциклопедія Києва (http://wek.kiev.ua), Вікісловник (http://uk.wiktionary.org)	Користувачі сайта після реєстрації мають можливість створювати нові веб-сторінки, розміщуючи на них інформаційні статті з текстовим і мультимедійним вмістом, редагувати власні матеріали та статті, створені іншими користувачами, переглядати історію редагувань, пов'язувати гіперпосиланнями сторінки цього сайта та встановлювати посилання на зовнішні ресурси, обговорювати розміщений матеріал
	Веб-хостинги з вбудованими системами керування контентом: uCoz (http://www.ucoz.ua), Сайти Google (https://sites.google.com), Блоги I.ua (http://blog.i.ua)	Користувачі можуть створювати веб-сайти та наповнювати їх контентом безпосередньо на серверах веб-хостингу. Веб-сторінки створюються на основі готових шаблонів. Відвідувачі сайтів, як правило, мають можливість залишати коментарі на веб-сторінках для висловлення власних думок щодо вмісту сайта
	Он-лайн офіси: Документи Google (https://docs.google.com), Toffix.Ru (http://toffix.ru), Feng Office (http://www.fengoffice.com)	Користувачі можуть створювати та зберігати електронні документи в Інтернеті. Документ одночасно можуть переглядати та редагувати кілька користувачів, для яких встановлені відповідні дозволи автором документа
Формування карт знань	Карти знань: Gliffy (http://www.gliffy.com), Bubbl.us (http://bubbl.us), Mapul (http://www.mapmyself.com)	Користувачі в режимі он-лайн можуть створювати та редагувати карти знань (когнітивні карти) – анімовані графічні зображення, що візуалізують зв'язки між об'єктами певної предметної області

Вид діяльності	Сервіси Веб 2.0, приклади ресурсів	Опис сервісу
Зберігання документів різних видів для спільного користування	Соціальні медіа-сховища: он-лайн фотоальбом Фламбер (http://flamber.ru), збірка відео Рутьюб (http://www.rutube.com), збірка документів різних форматів Scribd (http://www.scribd.com), збірка аудіоматеріалів Bee.fm (http://www.bee.fm)	Користувачі мають можливість зберігати на сервері власні документи, позначати їх тегами, відтворювати в режимі он-лайн документи, графічні зображення, презентації, відеоролики інших користувачів, що розміщені на сервері медіа-сховища, залишати власні коментарі. Забезпечується можливість доступу до них з різних місць і різними користувачами, зручного пошуку, спільного використання документів, включення документів до вмісту сайтів
	Файлообмінні системи (файловий хостинг): Shareua.Com (http://www.shareua.com), Upload.Com.Ua (http://upload.com.ua), Fileshare (http://fileshare.in.ua)	Користувачі мають можливість завантажувати на сервер для зберігання файли різних типів. Кожний файл отримує власну URL-адресу. За наявності адреси інші користувачі можуть завантажити відповідний файл із сервера на свій локальний комп'ютер. Завдяки цьому здійснюється обмін файлами
Організація та планування індивідуальної та спільної проектної діяльності	Он-лайн календарі: Календар Google (http://www.google.com/calendar), Яндекс.календарь (http://calendar.yandex.ru)	Користувачі мають можливість фіксувати в он-лайн календарі заплановані події для організації індивідуальної та спільної діяльності, отримувати повідомлення про наближення часу події, залишати коментарі щодо кожної події
	Органайзери: Organizeit.ru (http://organizeit.ru), Business it online (http://www.businessitonline.com)	Он-лайн органайзери об'єднують кілька функцій для організації діяльності: календар, адресну книгу, записник, програму для планування завдань, систему нагадування про події з використанням електронної пошти, ІМ-месенджера або SMS. Користувачам надається можливість спільно редагувати записи та використовувати названі засоби
	Он-лайн менеджери проектів: ZohoProject (http://projects.zoho.com), EasyProjects (http://www.easyprompts.net)	Веб-додатки призначені для надання групам користувачів послуг з управління проектами, їх планування, спілкування в процесі виконання робіт, спостереження за ходом проекту, підготовки звітів тощо

Продовження таблиці 4.7

Вид діяльності	Сервіси Веб 2.0, приклади ресурсів	Опис сервісу
Визначення географічного положення та прокладання маршрутів	Геосервіси: Карти Google (http://maps.google.com), Панораміо (http://www.panoramio.com), Google Планета Земля (http://earth.google.com)	Користувачі можуть знаходити об'єкти на мапі земної поверхні, визначати їх географічні координати, пов'язувати фотографії з об'єктами або координатами, коментувати їх, прокладати маршрути через визначені пункти, переглядати супутникові фотографії поверхні Землі тощо
Спілкування	Блоги (англ. <i>web log</i> – мережний журнал) – мережні щоденники: Українська блогосфера (http://blogosphere.com.ua), Blogoreader (http://blogoreader.org.ua), Блог газети «Українська правда» (http://blogs.pravda.com.ua)	Блоги – це веб-сайти, що створюються для розміщення повідомлень з висловленням своїх думок, міркувань щодо тих чи інших подій, презентації власної творчості тощо. Повідомлення в блозі мають власні URL-адреси, тому їх можна знайти засобами пошукових систем. Кожен користувач Інтернету може залишити свій коментар на повідомлення в блозі, якщо це дозволено блогером – автором блогу
	Ресурси соціальних мереж: Connect.ua (http://connect.ua), PROFEO (http://www.profeo.ua), ВКонтакте (http://vkontakte.ru), Мой Мир@mail.ru (http://my.mail.ru)	Сайти, що призначені створювати умови для спілкування групі користувачів, об'єднаних спільними інтересами, уподобаннями, місцем навчання, роботи, відпочинку тощо. У ході реєстрації та участі в обговореннях користувачі наповнюють сайт змістом. Сайти соціальних мереж мають засоби пошуку облікових записів учасників мережної спільноти та обміну повідомленнями між ними

За значної популярності сервісів **Веб 2.0** потрібно пам'ятати про проблеми, пов'язані з їх поширенням. Унаслідок доступності цих сервісів ресурси Інтернету заповнюються неякісними інформаційними матеріалами, що знижує довіру до електронних джерел. Масово порушуються авторські права на об'єкти інтелектуальної праці. Можливість висловлювати думки анонімно знімає моральну відповідальність за зміст повідомлень. Крім того, реєстрація на ресурсах соціальних мереж може призвести до незаконного використання приватних даних користувачів.

Вікі-технології

У числі сервісів **Веб 2.0** особливе місце займають вікі-проекти. Основне їх призначення – накопичення та структуроване зберігання інформаційних матеріалів з можливістю редагування та обговорення цих матеріалів групами користувачів. Найвідоміший з вікі-проектів – вільна багатомовна веб-енциклопедія **Вікіпедія**, з особливостями використання якої в навчальному процесі ви ознайомилися в 10-му класі.

Вікі-проекти розроблені на основі технології побудови веб-сайтів **Wiki** – системи зберігання та структурування бази знань, виведення їх у гіпертекстовому форматі, забезпечення навігації та пошуку. Вікі-технології надають користувачу такі можливості:

- створити нову сторінку з даними (статтю) може кожен відвідувач вікі-проекту, зареєстрований у системі, при цьому дозвіл на перегляд і редагування сторінки може бути наданий лише певній групі користувачів;
- кожна стаття має власне ім'я, унікальне в межах вікі-проекту, яке використовується для встановлення гіперпосилання на неї;
- можливе одночасне опрацювання статті у вікі-проекті кількома користувачами;
- можливе багаторазове редагування статті вікі-проекту зі збереженням історії редагування;
- відображення зміни в статтях відбувається відразу після збереження відредагованої сторінки;
- можливе повернення до попередніх версій статті, якщо зміни, на думку автора статті чи адміністратора проекту, погіршили її якість;
- користувачу надаються зручні засоби для форматування символів, вставлення таблиць, зображень та інших мультимедійних об'єктів, створення гіперпосилань;
- широко використовується механізм міток-тегів для об'єднання матеріалів веб-сторінок проекту за категоріями та легкого пошуку даних у базі знань;
- створення нової статті здійснюється шляхом створення гіперпосилання на неї в будь-якій існуючій статті

та ін.

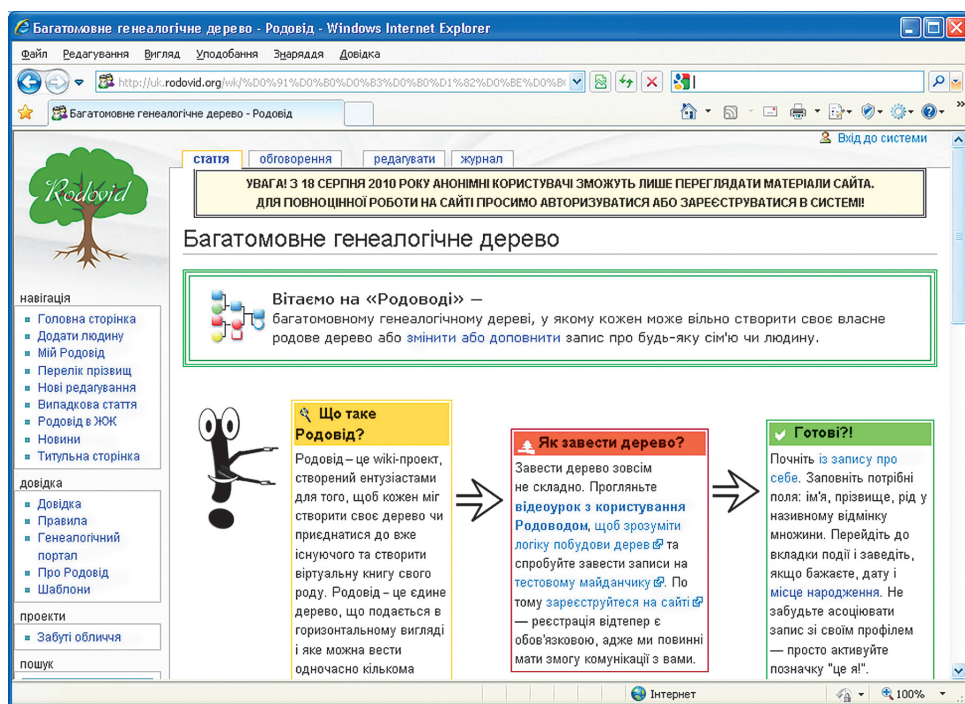


Рис. 4.75. Головна сторінка вікі-проекту **Родовід**

Наразі з використанням вікі-технологій розроблені енциклопедичні ресурси, наприклад **OpenWiki** (<http://www.openwiki.org.ua>) – проект вільної української енциклопедії, **Майдан Вікі** (<http://www.maidan.org.ua/wiki>) – політична енциклопедія, **Веб-енциклопедія Києва** (<http://wek.kiev.ua/uk>) тощо, та ресурси деяких громадських, освітніх, політичних організацій, наприклад **ЗапоВікі** (<http://wiki.ciit.zp.ua>) – мережне об'єднання учасників навчально-виховного процесу Запорізької області, **Родовід** (<http://uk.rodovid.org>) – багатомовне генеалогічне дерево (рис. 4.75), **Вікі Кіровоградського державного педагогічного університету ім. В. Винниченка** (<http://wiki.kspu.kr.ua>) та ін.

Веб-спільноти

Соціалізація сервісів **Веб 2.0** знаходить своє відображення в тому, як активно користувачі сервісів беруть участь у обговоренні різноманітних матеріалів, залишаючи свої коментарі, спілкуються один з одним, відслідковують новини, що з'являються на улюблених ресурсах. Як наслідок, з груп користувачів Інтернет-ресурсів утворюються **веб-спільноти**. Найчастіше – це групи людей, що не пов'язані офіційними стосунками і, можливо, не знайомі особисто один з одним. Їх об'єднує особистий чи професійний інтерес до теми, навколо якої проводиться обговорення. Метою веб-спільноти є обмін знаннями та досвідом, взаємонавчання, створення нових знань і знаходження шляхів розв'язування проблем.

Веб-спільноти можна класифікувати за різними властивостями. Наведено кілька класифікацій (рис. 4.76).

Кожний вид веб-спільнот має свої особливості. Вони відрізняються чисельністю, віком і професією учасників, специфікою інформаційного наповнення веб-ресурсу.

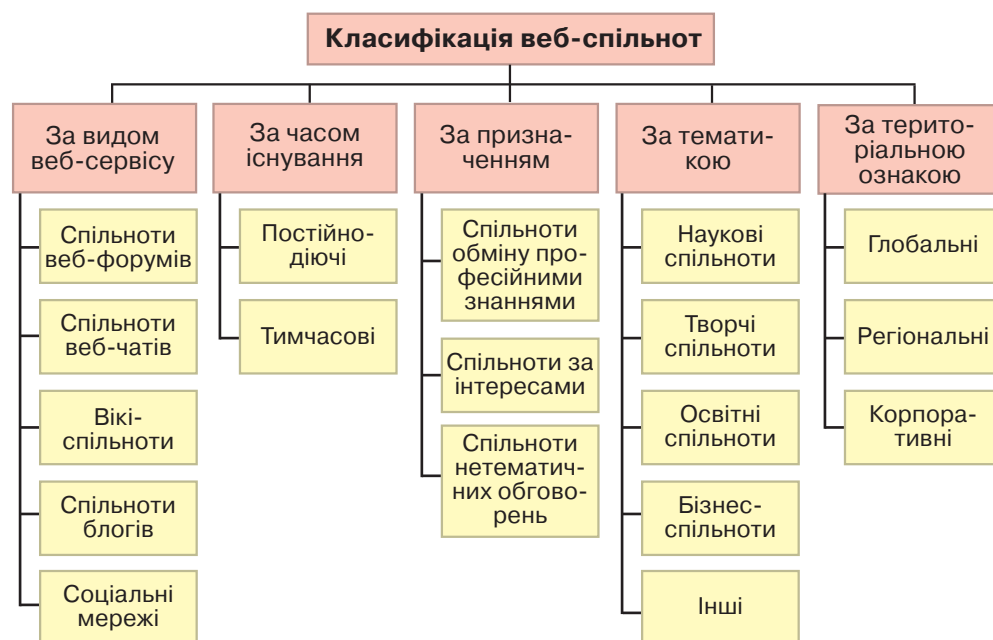


Рис. 4.76. Приклад класифікації веб-спільнот

Спільноти чатів, як правило, популярні серед молоді, що спілкується на довільні теми. Склад спільноти чата швидко змінюється. Обговорюються, як правило, короткочасні проблеми або спілкування зведено до обміну враженнями щодо останніх подій. Для інформаційного наповнення чатів характерні неструктурованість повідомлень, коротка тривалість збереження матеріалів обговорень, неможливість дізнатися про хід дискусій, що відбулися за відсутності користувача в чаті.

Спільноти форумів утворюються з метою обговорення питань найчастіше у межах визначеної теми, часто професійного спрямування. У ході обговорень, як правило, висловлюються думки щодо повідомлень автора теми та коментарів інших членів спільноти. Інформаційне наповнення форумів структуроване, зберігається тривалий час, подається в зручній формі. У спільноті реалізована система оцінювання активності учасників обговорень і можливість отримання автором теми повідомлень про нові публікації у форумі. Розроблені правила спілкування у форумах, за дотриманням яких слідкують модератори. Існує механізм обмеження доступу до обговорень. Для спільнот форумів характерне об'єднання під час обговорення певної проблеми і зміна складу у ході розгляду іншої проблеми.

Спільноти блогів утворюються для спілкування за темою, що запропонована блогером та оголошена в назві блогу. У ході обговорень, як правило, висловлюються думки щодо повідомлень автора блогу. Читачі блогу можуть оцінювати якість повідомлень і коментарів до них. Обговорення в блозі має найчастіше лінійну структуру. Повідомлення та коментарі до них зберігаються тривалий час. Реалізовані засоби отримання членами спільноти сповіщень про публікацію нових повідомлень у блозі. Правила спілкування та доступ до матеріалів блогу встановлює автор. Різні блоги можуть бути зв'язані гіперпосиланнями за технологією «трекбек». Служби, що надають хостинг для блогів, створюють власні внутрішні механізми, що сприяють поширенню зв'язків і утворенню спільнот серед блогерів. Сукупність блогів мережі називають **блогосферою**.

Вікі-спільноти утворюються, як правило, з метою розробки якісного інформаційного наповнення ресурсу, створення так званої «мережі знань». У ході обговорень уточнюються формулювання понять, формується стиль енциклопедичних і довідкових статей, доповнюються посилання на джерела даних і на зовнішні ресурси тощо. Вікі-сайти часто використовують для організації навчальних проєктів, дистанційного навчання, що сприяє утворенню освітніх спільнот. Інформаційні матеріали спільноти добре структуровані, пов'язані численними зв'язками. Правила спілкування та доступ до матеріалів вікі-спільноти встановлюють члени спільноти в ході обговорень.

Соціальні мережі, що утворені на основі веб-ресурсів, формуються, як правило, на основі особистого знайомства членів спільноти та подальшого розширення мережі знайомств. Об'єднання в спільноти здійснюється на основі особистих і професійних інтересів. Інформаційне наповнення слабо структуроване. Статус членів спільноти та склад адміністрації формуються на основі активності в обговореннях.

Блоги. Різновиди блогів

Сервіс мережних щоденників – **блогів** – став альтернативою персональним веб-сайтам. Як і персональний сайт, блог використовується для висловлення особистої позиції власника, але в блогах важливою частиною є коментарі відвідувачів, їх діалог з автором блогу.

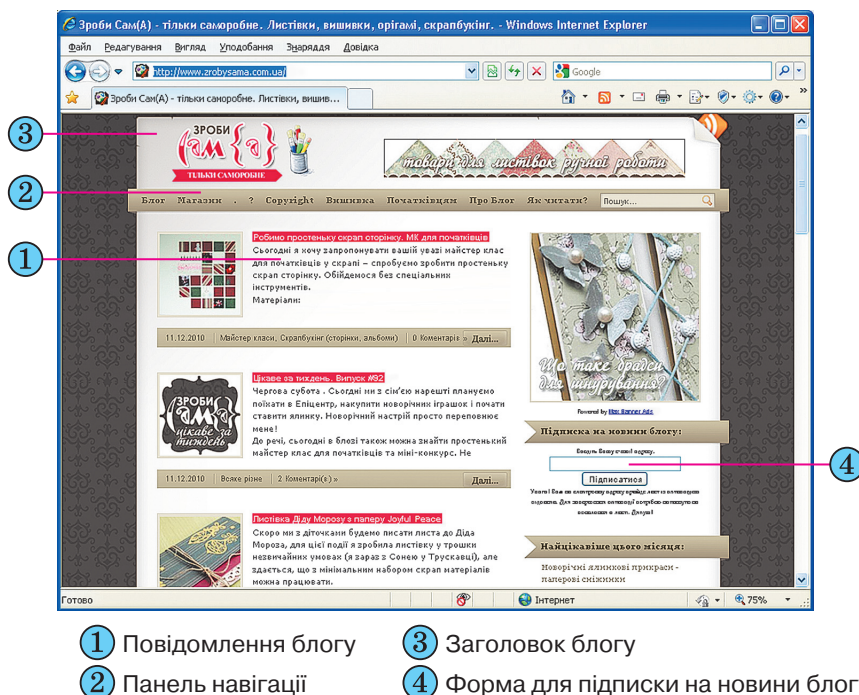
Блог побудований як стрічка повідомлень, розміщених на веб-сторінці у хронологічному порядку, до яких відвідувачі блогу можуть залишати свої коментарі. Останні повідомлення автора розташовуються, як правило, у верхній частині веб-сторінки, коментарі до повідомлень – під повідомленням (рис. 4.77). У тексті повідомлення можна розміщувати мультимедійні фрагменти. У кожного повідомлення блогу є заголовок і власна URL-адреса.

Для блогів характерні:

- короткі повідомлення;
- регулярне оновлення інформаційного наповнення;
- своєрідний діалог між автором і читачами;
- можливість отримання сповіщень про публікацію нових повідомлень;
- наявність єдиної для всіх повідомлень ідеї, що оголошена в назві блогу. Блоги можна класифікувати за різними ознаками.

Залежно від мети створення розрізняють блоги:

- **особистий** – аналог особистого щоденника, містить опис поточних подій і пов'язані з ними міркування автора;
- **тематичний** – містить повідомлення з однієї тематики;
- **бізнес-блог** – створюється для просування певного товару або послуги, містить рекламу, огляд продукції, коментарі тощо;



① Повідомлення блогу

③ Заголовок блогу

② Панель навігації

④ Форма для підписки на новини блогу

Рис. 4.77. Сторінка блогу

- **суспільний** – створюється з метою спілкування з колегами або клієнтами, громадою;
- **освітній** – створюється з метою спілкування учасників навчального процесу

та ін.

За типом повідомлень блоги поділяються на:

- **текстові** – блоги, основним видом повідомлень у яких є текстові повідомлення;
- **фотоблоги** – блоги, що містять добірки фотографій;
- **музичні** – блоги, основним вмістом яких є музичні записи;
- **блогкасти, або подкасти, (iPod – портативний програвач, англ. broadcast – передача, мовлення)** – блоги, вмістом яких є аудіофайли, надиктовані та опубліковані в мережі автором;
- **відеоблоги** – основний вміст блогів складають відеоматеріали.

Блоги можна створити на сайтах **Blog.net.ua** (<http://blog.net.ua>), **Блоги@Mail.Ru** (<http://blogs.mail.ru>), **Blogger** (<http://blogger.com>) та ін.

Створення блогу. Публікація повідомлень

Розглянемо процес створення блогу на прикладі сервісу **Blogger** (<http://blogger.com>). Для користування ним потрібно мати акаунт **Google**.

Для створення блогу потрібно:

1. Відкрити у вікні браузера головну сторінку сервісу **Blogger** (<http://blogger.com>).
2. Увести дані вашого облікового запису **Google**.
3. Вибрати на сторінці **Blogger: Панель інструментів** гіперпосилання **Створити блог**.
4. Увести в поле **Заголовок блогу** назву блогу (рис. 4.78).

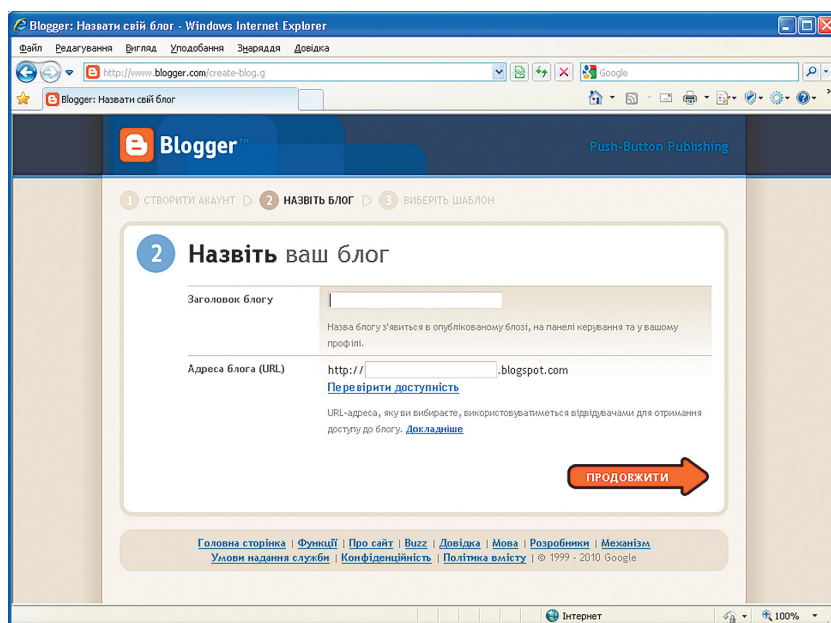


Рис. 4.78. Сторінка **Blogger: Назвіть ваш блог**

5. Увести в поле **Адреса блогу (URL)** частину URL-адреси блогу. Повна URL-адреса матиме вигляд `http://адреса.blogspot.com`, де *адреса* – уведений користувачем текст.
6. Вибрати гіперпосилання **Перевірити доступність** для перевірки адреси на унікальність. За потреби змінити адресу.
7. Вибрати гіперпосилання **Продовжити**.
8. Вибрати початковий шаблон оформлення блогу із запропонованих. Вибрати гіперпосилання **Продовжити**.
9. Вибрати на сторінці **Blogger: Готово** відповідне гіперпосилання для подальших дій:
 - **уже зараз розпочати створення дописів** – для створення та публікації першого повідомлення у блозі. Така сама дія буде виконана за вибору гіперпосилання **Початок ведення блогів**;
 - **налаштувати дизайн блогу** – для вибору шаблону оформлення, фону, розмітки сторінки, кольорової гами, шрифтів та ін.;
 - **розширені параметри налаштування** – для зміни домену та імпортування існуючого блогу в поточний.

За вибору першого із названих гіперпосилань відкриється сторінка **Blogger: Назва блогу – Створити публікацію**, де *Назва блогу* – назва, яку було введено користувачем (рис. 4.79). Для створення повідомлення слід ввести його заголовок у відповідне поле та текст у робочу область. Для редагування та форматування повідомлень використовується панель інструментів.

Нижче робочої області розміщено поле **Мітки** для цієї публікації. У цьому полі пропонується ввести ключові слова – мітки, що характеризують зміст повідомлення. За цими мітками повідомлення може бути знайдене та проіндексоване пошуковими машинами.

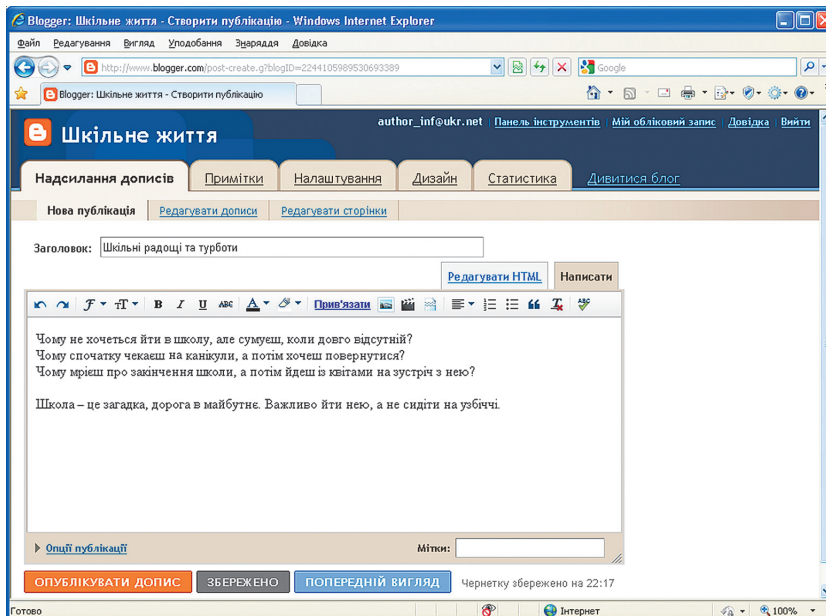


Рис. 4.79. Сторінка створення публікації блогу

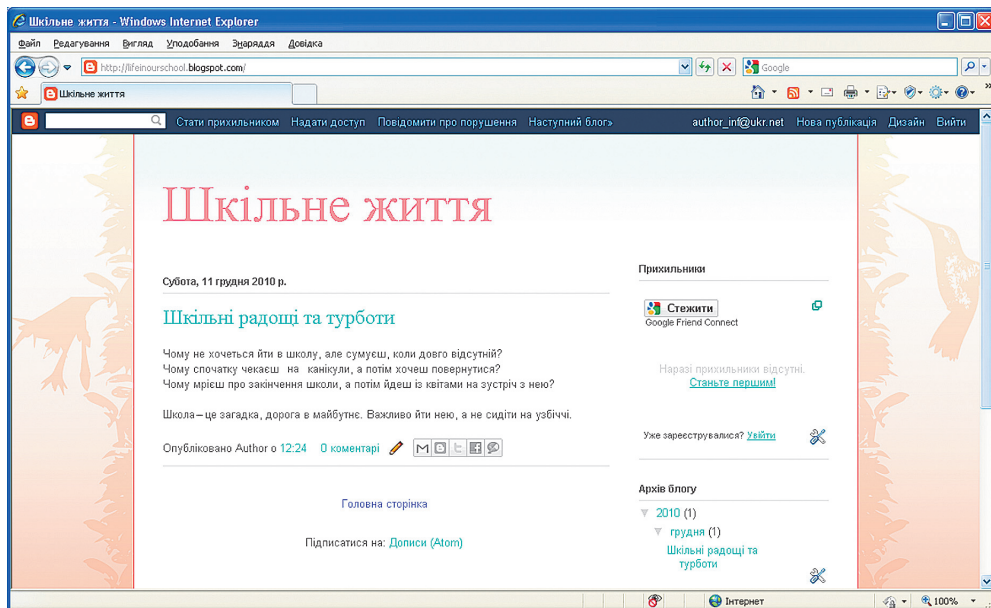


Рис. 4.80. Сторінка блогу

Після введення тексту повідомлення його слід опублікувати, вибравши гіперпосилання **Опублікувати допис**. За вибору гіперпосилання **Зберегти зараз** повідомлення буде збережено, але не відобразиться на сторінці блогу.

Автор може переглянути або відредагувати опублікований допис, вибравши відповідне гіперпосилання на сторінці, що відкриється. За вибору у верхній частині сторінки гіперпосилання **Дивитися блог** відкриється сторінка з повідомленнями блогу (рис. 4.80).

Налаштування та адміністрування блогу



Для створення нового повідомлення слід вибрати гіперпосилання **Нова публікація** у верхній частині веб-сторінки блогу, а для зміни оформлення – гіперпосилання **Дизайн**. В обох випадках здійснюється перехід на сторінку налаштувань блогу (див. рис. 4.79), що має кілька вкладок. Кожна з вкладок містить елементи керування для виконання окремих налаштувань (табл. 4.8).

Таблиця 4.8. Призначення вкладок налаштування блогу

Вкладка	Призначення
Надсилення дописів	Створення, редагування, збереження, попередній перегляд і публікації повідомлень автором блогу
Примітки	Опрацювання коментарів. Кожний коментар до повідомлень автор блогу може позначити як спам, видалити повністю або видалити лише зміст коментаря
Налаштування	Змінення автором заголовка блогу, додавання опису, включення до списків блогів, що переглядаються пошуковими системами, встановлення дозволів, форматування блогу та ін.

Продовження таблиці 4.8

Вкладка	Призначення
Дизайн	Додавання автором блогу гаджетів, змінення розташування окремих елементів сторінки, вставлення зображення в заголовок блогу, змінення шаблону оформлення блогу та ін.
Статистика	Отримання автором даних про кількість переглядів блогу та окремих публікацій, місцезнаходження відвідувачів тощо

Під час перегляду блогу можна виконати налаштування окремих об'єктів сторінки – гаджетів, окремих інформаційних блоків. Для переходу до режиму налаштувань потрібно вибрати кнопку . Виконати перехід на сторінку редагування текстів публікацій можна, вибравши кнопку .

Автор може здійснювати адміністрування блогу, встановлюючи користувачам права доступу до матеріалів блогу. За замовчуванням дозвіл на перегляд публікацій блогу надано усім користувачам Інтернету. Право публікувати нові повідомлення має лише автор, залишати коментарі на публікації – лише зареєстровані користувачі. Ці налаштування можна змінити. Для цього слід:

1. Відкрити сторінку налаштувань блогу.
2. Вибрати гіперпосилання **Примітки** на вкладці **Налаштування**.
3. Указати, які групи користувачів мають право залишати коментарі, вибором відповідного перемикача в групі **Хто може коментувати?**
4. Вибрати гіперпосилання **Дозволи** на вкладці **Налаштування** (рис. 4.81).
5. Вибрати кнопку **Додати авторів**, щоб надати право іншим користувачам створювати публікації у блозі, вказати адреси їхніх електронних поштових скриньок.

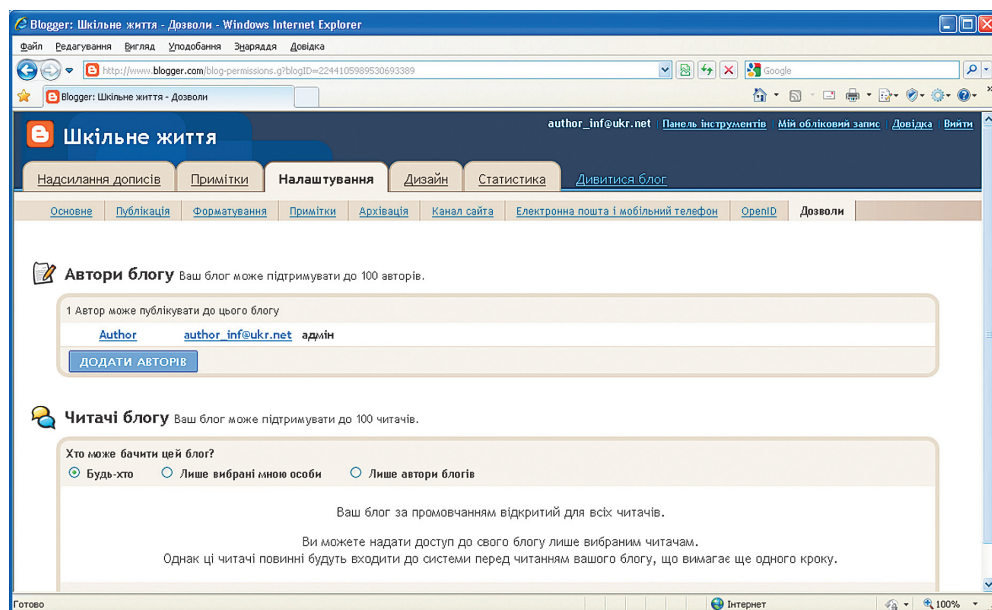


Рис. 4.81. Вкладка встановлення дозволів

6. Установити, яким групам користувачів буде дозволено читати публікації блогу, вибором відповідного перемикача в області **Читачі блогу**.



Перевірте себе

- 1°. Які технології покладено в основу розробки сервісів **Веб 2.0**? У чому полягають ці технології?
- 2°. Які особливості притаманні сервісам **Веб 2.0**? Поясніть їх.
- 3°. Які існують типи сервісів **Веб 2.0** за видами діяльності користувачів?
- 4*. Чим, на ваш погляд, пояснюється популярність сервісів **Веб 2.0**?
- 5°. Які проблеми пов'язані з поширенням сервісів **Веб 2.0**?
- 6°. У чому полягають можливості вікі-технології?
- 7°. За значеннями яких властивостей можна класифікувати веб-спільноти? Поясніть особливості кожного виду.
- 8°. Що таке блог?
- 9*. Чим, на ваш погляд, пояснюється популярність блогів у користувачів? Чому блоги вважаються альтернативою персональним веб-сторінкам?
- 10°. За значеннями яких властивостей можна класифікувати блоги? Поясніть особливості кожного виду.
- 11°. Які ви знаєте сайти, призначені для створення блогів?
- 12°. Наведіть алгоритм створення блогу з використанням сервісу **Blogger**.
- 13°. Як створити нову публікацію у блозі?
- 14°. Які налаштування можна виконувати у блогах?
- 15°. Які дозволи можна надавати користувачам при адмініструванні блогу?



Виконайте завдання

- 1*. Ознайомтеся з матеріалами сайтів **Вікіпедія** (<http://uk.wikipedia.org>), **Flickr** (<http://www.flickr.com>), **Google Планета Земля** (<http://earth.google.com>). Визначте, у чому полягає застосування названих технологій і заповніть порівняльну таблицю.

Технологія	Вікіпедія	Flickr	Google Планета Земля
Веб як платформа			
Mash-up			
Колективний розум			
Фолксономія			
Веб-синдикація			
Соціалізація			

- 2°. Установіть відповідність між видами діяльності людини та видами сервісів **Веб 2.0**.

	Вид діяльності		Сервіс Веб 2.0
1	Пошук даних	А	Народні класифікатори
2	Спільна розробка інформаційних матеріалів	Б	Менеджери проектів
3	Зберігання документів	В	Соціальні пошукові системи
4	Створення колективних каталогів посилань	Г	Соціальні медіа-сховища
5	Визначення географічного положення	Д	Вікі-проекти
6	Спілкування	Е	Геосервіси
7	Організація спільної проектної діяльності	Є	Блоги



- 3*. Використовуючи дані з таблиці 4.7, створіть ієрархічну діаграму – схему класифікації сервісів **Веб 2.0** за видами діяльності користувача.



4. Порівняйте особливості функціонування двох вікі-проектів **Родовід** (<http://uk.rodovid.org>) і **Веб-енциклопедії Києва** (<http://wek.kiev.ua/uk>) та заповніть таблицю 4.9.

Таблиця 4.9

<i>Запитання для порівняння</i>	<i>Сайт 1</i>	<i>Сайт 2</i>
Які матеріали представлено в проекті?		
Якими засобами повідомляється тематика сайта?		
Які засоби передбачено для переходу між сторінками сайта?		
Як отримати довідку про особливості роботи з матеріалами сайта?		
Як можна знайти потрібні матеріали на сайті?		
Які засоби пропонуються для редагування статей?		
Чи потрібна реєстрація у проекті для участі в редагуванні статей?		
Які засоби пропонуються для перегляду історії редагувань?		
Які засоби пропонуються для створення нової статті?		
Як організовано обговорення матеріалів?		
Чи потрібна реєстрація в проекті для участі в обговореннях матеріалів?		
Коли на сайті було внесено останні зміни?		



- 5*. Порівняйте особливості функціонування сайтів **ЕкоАгроВікі** (<http://agrowiki.nubip.edu.ua>) і **Фармацевтична енциклопедія** (<http://www.pharmencyclopedia.com.ua>) та заповніть таблицю 4.9.

- 6*. Заповніть порівняльну таблицю веб-спільнот.

<i>Властивість</i>	<i>Спільнота</i>				
	<i>Чат</i>	<i>Форум</i>	<i>Вікі</i>	<i>Блог</i>	<i>Соціальна мережа</i>
Мета утворення спільноти					
Тривалість збереження матеріалів					
Структурованість матеріалів					
Наявність системи оцінювання активності членів спільноти					
Хто встановлює правила спілкування					
Наявність механізму обмеження доступу до обговорень					
Наявність засобів отримання сповіщень про нові матеріали					
Можливість пошуку потрібних матеріалів					
Можливість редагування інформаційних матеріалів кожним членом спільноти					



- 7°. Створіть ієрархічну діаграму – схему класифікації блогів.
- 8°. Перегляньте блог **Українська блогосфера** (<http://blogosphere.com.ua>). Проаналізуйте дизайн блогу, його структуру, публікації та коментарі до них. Визначте, що з цього блогу вам було б цікаво використати у власному блозі.
- 9°. Створіть власний блог з використанням сервісу **Blogger** (<http://blogger.com>). Присвятіть ваш блог обговоренню особливостей утримання домашніх тварин. Виберіть шаблон оформлення з категорії *Водяний знак*. Розмістіть у блозі повідомлення про тварину, яку ви хотіли б утримувати вдома. Повідомте адресу вашого блогу вчителю інформатики.
- 10°. Додайте публікацію до вашого блогу про домашню тварину вашого друга.
- 11°. Змініть налаштування вашого блогу: у заголовку додайте опис блогу, розмістіть фотографію домашньої тварини, наприклад з папки **Тема 4\Завдання 4.10\Тварини**. Додайте гаджет *Статистика для блогу*. Змініть макет тіла сторінки блогу, розмістивши область гаджетів ліворуч від області повідомлень.
- 12°. Змініть дозволи на роботу користувачів з вашим блогом: дозвольте залишати коментарі в блозі будь-кому, включаючи анонімних користувачів, створювати повідомлення у блозі – учителю інформатики. Надішліть запрошення на адресу електронної пошти вашого вчителя інформатики бути одним з авторів вашого блогу.
- 13°. Перегляньте блог вашого однокласника. Залиште коментар до однієї з публікацій.



Практична робота № 19. Створення та ведення власного блогу

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

Створіть власний блог з використанням сервісу **Blogger** (<http://blogger.com>):

1. Присвятіть ваш блог обговоренню вашої майбутньої професії. Доберіть відповідний заголовок блогу.
2. Виберіть шаблон оформлення з категорії *Захоплення*.
3. Розмістіть у блозі повідомлення про професію, яка вам подобається.
4. Змініть налаштування вашого блогу:
 - а) у заголовку додайте опис блогу;
 - б) розмістіть у заголовку зображення атрибутів професії, наприклад з папки **Тема 4\Практична 19\Професії**;
 - в) додайте до блогу гаджет **Поле пошуку**, розмістивши його нижче від заголовка блогу;
 - г) змініть макет тіла сторінки блогу, розмістивши область гаджетів у дві колонки ліворуч і праворуч від області повідомлень.
5. Додайте публікацію до вашого блогу, розповівши про професію одного з ваших батьків або знайомих і ваше ставлення до цієї професії.
6. Змініть дозволи на роботу користувачів з вашим блогом:
 - а) дозвольте залишати коментарі у блозі лише користувачам облікових записів **Google**;
 - б) читачами блогу призначте лише вибраних вами осіб, запросіть трьох однокласників почитати ваш блог;
 - в) надішліть запрошення учителю інформатики бути одним з авторів вашого блогу.

4.11. Спільна робота з документами



1. Назвіть відомі вам сервіси **Веб 2.0**. Які з них призначені для організації спілкування користувачів?
2. Для чого призначені он-лайн офіси? Які їх можливості?
3. Які сервіси **Google** ви знаєте? Охарактеризуйте їх призначення.
4. Що таке облікові записи? У яких службах вони використовуються?
5. Які правила роботи на форумі та в чаті?

Системи колективної взаємодії

Із розповсюдженням Інтернету кардинально змінилося поняття робочого колективу. Люди, що живуть в різних країнах і працюють у різних організаціях, можуть брати участь у спільній розробці проектів чи опрацюванні одних і тих самих документів. Сьогодні вже нікого не здивуєш співробітниками, які зовсім ніколи не заглядають в офіс компанії, а виконують свою роботу, не залишаючи власного будинку. Інші – постійно переміщуються по світу і беруть участь у різних нарадах, семінарах, знаходячись у дорозі.

Доволі часто співробітники, що працюють над одним проектом, знаходяться в різних будівлях, кабінетах тощо. У таких випадках для організації документообігу між учасниками проекту необхідно організувати *віртуальний офіс*, де будуть зберігатися документи для спільного використання. Головна проблема такої організації діяльності полягає в доступі всіх учасників до загальних документів і в організації колективного спілкування віддалених один від одного учасників. Для цього можна використати відомі вам можливості – електронну пошту, чати, служби миттєвого обміну повідомленнями, різні сервіси **Веб 2.0**.

Також існує багато спеціально розроблених програм для організації групової роботи і використання спільних документів, наприклад **Google Wave**, **12Sprints**, **Sharepoint**, **Novell Pulse**, **IPI.Manager**, **Feng Office**, **Microsoft Office Groove 2007** та ін. Такий вид програмного забезпечення поки що не отримав однозначної назви, але найчастіше ці програми називають *системи колективної взаємодії*. Вони можуть використовуватися як у локальних, так і в глобальних мережах.

Основне призначення систем колективної взаємодії:

- доступ до спільних ресурсів деякої групи учасників;
- організація спілкування між учасниками групи;
- координація та документування ходу робіт.

Організація спільної роботи в Microsoft Office Groove 2007

Однією з програм для організації колективної роботи та віртуального офісу є програма **Microsoft Office Groove 2007**, яка входить в пакет прикладних програм **Microsoft Office 2007**.

Вона надає користувачам середовище для проведення онлайн-зустрічей, обговорення спільних проектів, передачі файлів безпосередньо з одного комп'ютера на інший, зберігання спільних документів в одному сховищі, доступ до якого можливий з різних комп'ютерів, автоматичне оновлення файлів у сховищі при їх зміні кимось із учасників групи, зберігання різних версій файлів тощо. Також програма дає змогу користувачам дізнатися про присутність учасників робочої групи в спільному

середовищі, проглядати відомості про них, направляти їм повідомлення тощо. Програма включає служби миттєвого обміну повідомленнями, чат, а також можливості спільної роботи над документами. Взаємодія учасників відбувається в так званих робочих областях, де всі учасники групи, програмні засоби та спільні документи концентруються в одній зоні.

Для роботи з програмою потрібно мати підключення до мережі Інтернет. Для запуску програми потрібно виконати **Пуск** \Rightarrow **Всі програми** \Rightarrow **Microsoft Office** \Rightarrow **Microsoft Office Groove 2007**.

Якщо програма запускається вперше, то на екрані відобразиться **Майстер налаштувань облікового запису**, у вікні якого треба виконати такі кроки:

1. Вибрати відповідь на запитання – чи хочете ви створити новий обліковий запис для реєстрації в цій програмі або у вас уже є такий?
2. Відповісти на запитання – чи маєте ви код конфігурації облікового запису **Groove**? Якщо ці дані невідомі, то слід вибрати перемикач **У мене немає коду конфігурації облікового запису**.
3. Указати ім'я користувача, адресу електронної пошти та пароль для доступу до облікового запису.
4. Вибрати вид реєстрації створеного облікового запису. Бажано вибрати режим в загальному каталозі **Groove** – це спростить іншим учасникам пошук відомостей про користувача і запрошення його в групу.

Якщо використання програми відбувається не вперше, то після запуску програми потрібно буде в текстове поле ввести пароль для доступу до облікового запису користувача.

Після того як вхід у програму буде здійснено, на екрані відобразиться основне вікно програми (рис. 4.82), у якому розташовані меню програми, дві вкладки **Робочі області** і **Контакти**, а також область **Загальні завдання**.

Робоча область – це створене одним з користувачів місце, яке всі учасники групи використовують для розміщення документів та обговорення того чи іншого питання. Її можна порівняти з окремою кімнатою в чаті. У **Groove 2007** можна створити робочі області трьох типів, які відрізняються один від одного початковим набором доступних інструментів:

- **Стандартна** – призначена для створення віртуального середовища для спілкування учасників і розміщення файлів для спільного користування.
- **Спільне використання файлів** – призначена для створення папки на локальному комп'ютері учасника групи, вміст якої буде надано у спільний доступ іншим учасникам робочої області.
- **Шаблон** – дає змогу створити робочу область, вибравши необхідні засоби, які використовуватимуться учасниками групи в спільній роботі:
 - **Альбом** – для спільного малювання;

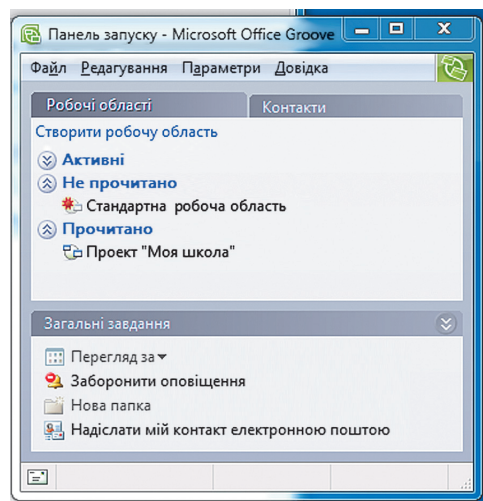


Рис. 4.82. Основне вікно програми **Microsoft Office Groove 2007**

- *Блокнот* – для створення коротких повідомлень;
- *Календар* – для створення колективного графіка роботи над проектом, власного розпорядку дня, призначення зустрічі;
- *Обговорення* – для ведення діалогів у стилі запитання-відповідь;
- *Зображення* – для додавання і перегляду зображень;
- *Наради* – для призначення порядку денного, доручень, ведення протоколу наради та ін.

Користувач, який створив робочу область, вважається її *Керівником*. Йому надані права на запрошення і виключення учасників групи, підключення і відключення інструментів, зміну ролі певного учасника робочої області та виконання інших операцій з управління робочою областю. Решта користувачів виступають у ролі *Учасника* або *Гостя*. Учасник може додавати нові панелі інструментів і запрошувати нових учасників до цієї групи. Гостю ж не надано ніяких дозволів, крім участі в обговореннях і роботі з документами.

Для створення робочої області потрібно:

1. Вибрати у вікні **Панель запуску** посилання **Створити робочу область**.
2. Вибрати тип створюваної робочої області (*Стандартна, Спільне використання файлів, Шаблон*).
3. Увести назву робочої області (наприклад, Проект «Моя школа»).
4. Вибрати кнопку **ОК**.

Щоб відкрити створену робочу область, потрібно двічі клацнути на її імені, після чого відкриється вікно цієї області. Наприклад, на рисунку 4.83 представлено вікно стандартної робочої області, у якій для користувачів надається два засоби – **Файли** та **Обговорення**, ярлики вкладок яких розташовані в нижній частині вікна (10, 9).

У верхній частині вікна при відкритій вкладці **Файли** розміщено меню (2) і панель інструментів (3), у лівій частині вікна – дерево папок області (4), в центральній – їх вміст (5). Використовуючи інструменти і команди робочої області, можна додати до неї файли або папки для спільного використання.

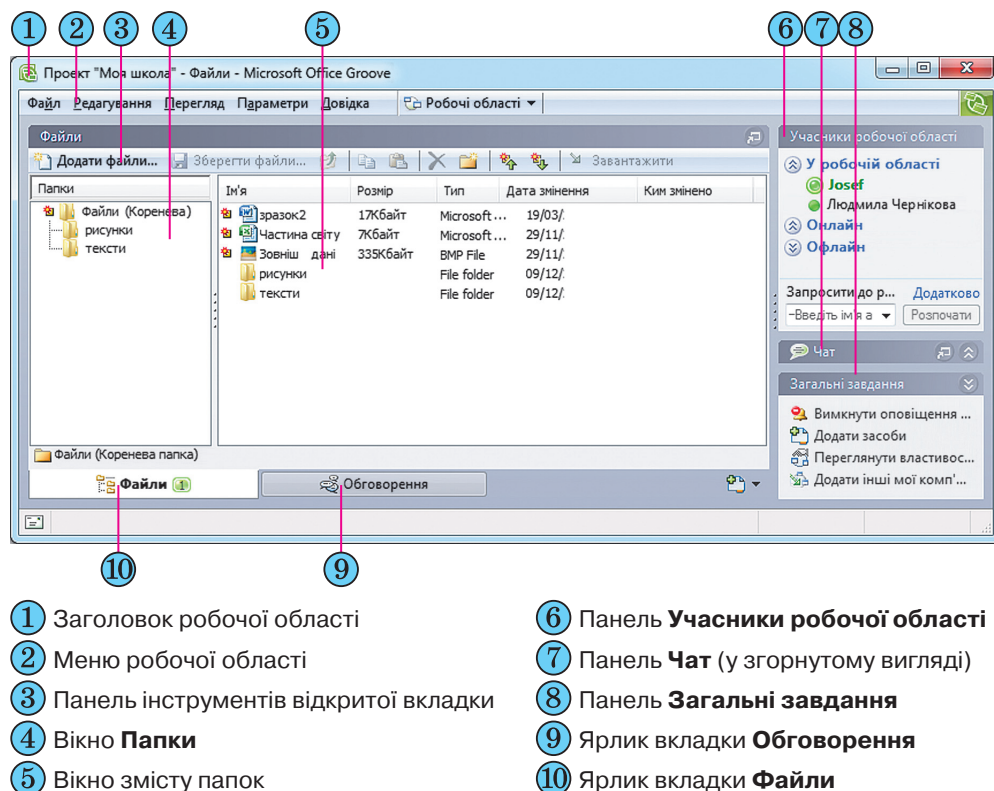
У правій частині екрана відображається список учасників групи (6). Усіх учасників розподіляють на три категорії:

- *У робочій області* – ті учасники, які зараз присутні в робочій області.
- *Онлайн* – користувачі, які зараз мають підключення до Інтернету та встановлену програму **Groove**, але не приєдналися до групи.
- *Офлайн* – користувачі, які приєдналися до групи, але на даний момент не мають підключення до Інтернету.

Панель **Чат** (7) використовується для організації текстового спілкування між учасниками, на панелі **Загальні завдання** (8) перераховано операції, які в даний момент доступні учасникові групи.

Після створення області потрібно організувати робочу групу, запросивши до неї учасників. *Звертаємо увагу*, що для цього в учасників на їх робочих комп'ютерах повинна бути встановлена програма **Groove** і настроений обліковий запис, також комп'ютери учасників повинні бути підключені до мережі Інтернет.

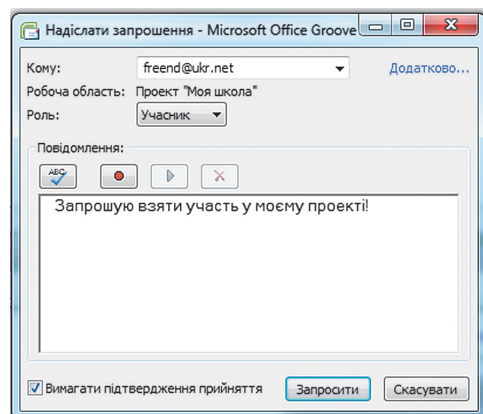
Для запрошення нових членів групи необхідно в нижній частині панелі **Учасники робочої групи** у текстовому полі ввести електронну адресу



- 1 Заголовок робочої області
- 2 Меню робочої області
- 3 Панель інструментів відкритої вкладки
- 4 Вікно **Папки**
- 5 Вікно змісту папок
- 6 Панель **Учасники робочої області**
- 7 Панель **Чат** (у згорнутому вигляді)
- 8 Панель **Загальні завдання**
- 9 Ярлик вкладки **Обговорення**
- 10 Ярлик вкладки **Файли**

Рис. 4.83. Вікно стандартної робочої області

людини, яку планується запросити до групи, і вибрати кнопку **Розпочати**. Після чого у вікні **Надіслати запрошення** (рис. 4.84) вибрати роль учасника, ввести текст повідомлення і вибрати кнопку **Запросити**. Якщо у вас підключено мікрофон, то запрошення можна записати як голосове повідомлення, скориставшись кнопкою , яка розміщена над вікном повідомлення.

Рис. 4.84. Вікно **Надіслати запрошення**

Тільки-но відправлене запрошення надійде користувачеві (тільки-но даний користувач запустить програму **Groove** та підключиться до Інтернету), його програма **Groove** сповістить про це відображенням тексту повідомлення в правому нижньому куті екрана. Коли запрошення буде прийнято, автору повідомлення буде надіслано підтвердження в аналогічний спосіб. А на моніторі у нового учасника групи відкриється вікно програми **Groove** з вмістом спільної робочої області. Імена всіх користувачів, які прийняли запрошення, з'являться у списку панелі **Учасники робочої групи**.

Надалі учасники робочої області можуть обмінюватися текстовими повідомленнями, створювати і редагувати спільні файли, залишати свої зауваження, змінювати інтерфейс і вміст робочої області. Усі зміни автоматично зберігатимуться в робочій області та будуть автоматично пересилатися всім учасникам групи – даний процес називається *синхронізація даних*. За відсутності доступу до мережі когось із членів групи, синхронізація даних відбудеться при першому ж сеансі доступу до Інтернету.


Розглянемо більш детально спільну роботу учасників стандартної робочої області.

Вкладка **Файли** призначена для роботи з документами. Використовуючи засоби панелі інструментів цієї вкладки, можна створювати нові папки, копіювати, видаляти, переміщувати об'єкти в робочій області, але це дозволяється робити тільки керівнику групи.

Усі учасники робочої групи мають дозвіл на розміщення нових документів у спільній області. Для цього слід виконати такі дії:

1. Відкрити вкладку **Файли**.
2. Відкрити потрібну папку.
3. Вибрати на панелі інструментів кнопку **Додати файли**.
4. Вибрати у вікні **Додавання файлів** потрібний файл на локальному комп'ютері учасника групи.
5. Вибрати кнопку **Відкрити**.

Розміщений у спільній папці документ з'явиться у вікні робочої області. Автоматично відбудеться синхронізація даних для всіх учасників групи. Кожен учасник групи може відкрити документ для перегляду і внесення змін. Після закриття відредагованого документа на екрані з'являється діалогове вікно, у якому потрібно підтвердити збереження файлу зі змінами та вказати за потреби нове ім'я файлу. Якщо кілька учасників будуть одночасно зберігати один і той самий документ, то програма повідомить про це і створить копію документа з іншим номером версії.

У списку біля імен змінених об'єктів буде розташовано значок . Це попередження учаснику групи, що він ще не переглянув нову версію документа. Після відкриття такого документа значок зникає.

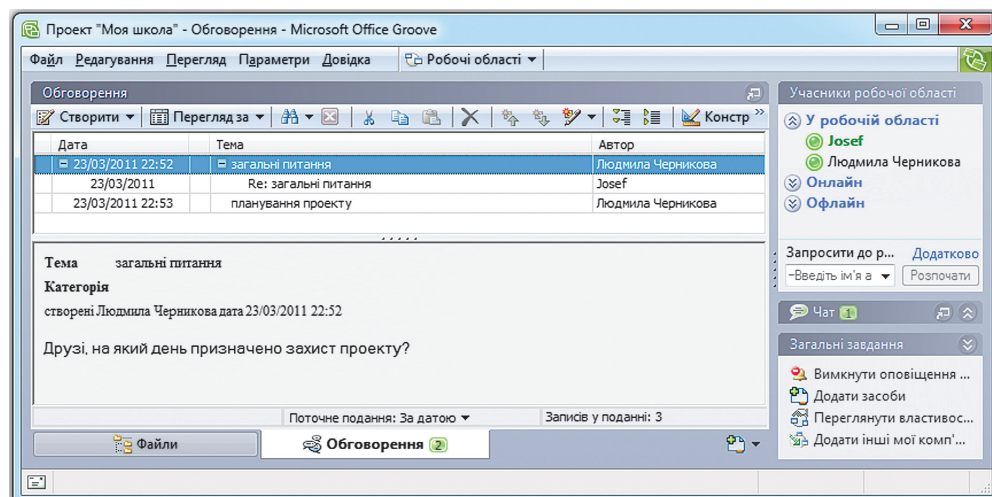


Рис. 4.85. Вкладка **Обговорення**

Інший засіб для спільної роботи учасників стандартної робочої області – це **Обговорення** (рис. 4.85). Використовуючи кнопки панелі інструментів цієї вкладки, можна створювати, видаляти, сортувати повідомлення тощо. Правила використання цього засобу, спосіб розміщення повідомлень на цій вкладці аналогічні роботі на форумах.

Для створення нової теми обговорення потрібно виконати такі дії:

1. Відкрити вкладку **Обговорення**.
2. Вибрати кнопку **Створити**, вибрати команду **Тема**.
3. Увести в діалогове вікно назву теми для обговорення і текст повідомлення.
4. Вибрати на панелі інструментів кнопку **Зберегти**.

Для написання відповіді на якесь повідомлення алгоритм дій такий:

1. Відкрити потрібне повідомлення.
2. Вибрати кнопку **Створити**, вибрати команду **Відповідь**.
3. Увести в діалогове вікно текст відповіді на повідомлення.
4. Вибрати на панелі інструментів кнопку **Зберегти**.

Також учасники групи можуть вести розмову у форматі чата. Для цього призначена панель **Чат**, яка розташована в правій частині вікна робочої області. Для розгортання чата в окреме вікно (рис. 4.86) потрібно в заголовку цієї області вибрати кнопку **Розгорнути** . Правила роботи та відображення повідомлень у чаті вам уже знайомі.

До робочої області можна додати й інші засоби (**Зображення**, **Альбом**, **Календар** тощо), використавши для цього команду **Додати засоби** панелі **Загальні завдання**.

Якщо на локальному комп'ютері учасника групи створено робочу область типу *Спільний доступ до файлів*, то керування такими папками здійснюється з програми **Провідник**. Програма **Groove** додає на панелі інструментів вікна програми **Провідник** нову кнопку **Синхронізація папки**, вибір якої відкриває у вікні відповідну панель **Завдання синхронізації** (рис. 4.87). Використання цієї панелі дає змогу спостерігати за тим, хто з користувачів знаходиться в робочій області, запрошувати нових користу-

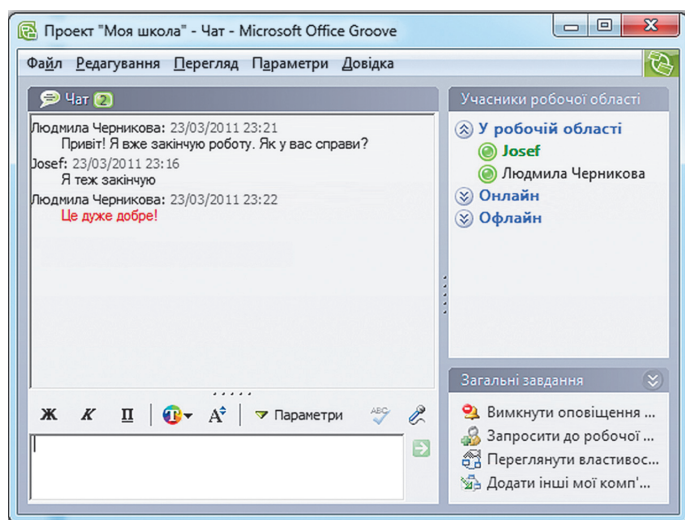
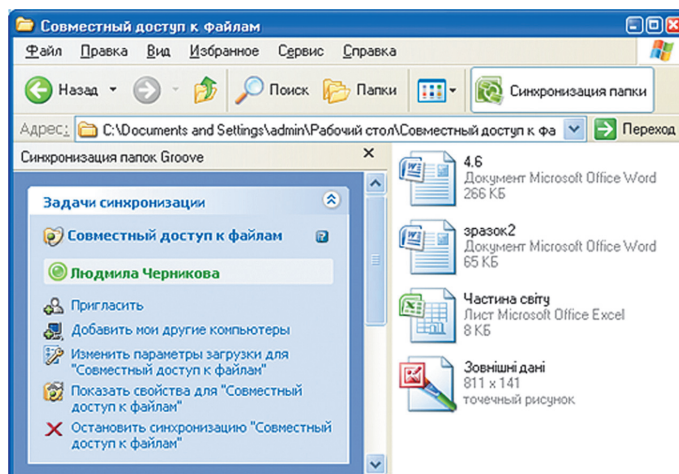


Рис. 4.86. Розгорнуте вікно **Чат**

Рис. 4.87. Вікно папки спільного доступу в програмі **Провідник**

вачів, додавати комп'ютери, з якими потрібно виконувати синхронізацію, синхронізувати файли робочої області з файлами на комп'ютері тощо.

Служби онлайнового документообігу

Організувати спільну роботу над документами учасників деякої групи можна і за допомогою он-лайн сервісів, які отримали назву «Он-лайн офіси». Їх використання дає змогу людям з усьому світу у зручний для них час працювати разом над спільними документами, зберігати їх не на локальному комп'ютері, а на відділеному комп'ютері в мережі і мати доступ до своїх матеріалів, де б ці особи не знаходилися.

Набір послуг, що надаються он-лайн сервісами, зазвичай включає всі основні засоби традиційних офісних пакетів, таких як текстовий редактор, табличний процесор, програми для створення презентацій, СУБД, органайзери, календарі тощо. Он-лайн офіс може бути доступний з будь-якого комп'ютера, у якого є підключення до Інтернету, незалежно від того, яку операційну систему він використовує.

Зазвичай основні сервіси он-лайн офісів пропонуються безкоштовно, а доступ до розширених можливостей потребує невеликої абонентської плати. На відміну від використання **Microsoft Office Groove 2007** користувачу не потрібно мати на своєму комп'ютері спеціального програмного забезпечення. Доступ до ресурсів здійснюється за допомогою браузерів.

Найпопулярнішою безкоштовною он-лайн службою, яка надає сервіси для колективної роботи з документами є на сьогодні служба **Документи Google** (англ. *Google docs* – документи Google). Використовуючи її, можна створювати електронні документи безпосередньо на порталі **www.docs.google.com** або завантажувати з локального комп'ютера користувача, зберігати документи в он-лайн сховищі **Google** і мати доступ до них з любого місця через мережу Інтернет, надавати за потреби доступ до цих ресурсів іншим користувачам.

Спільна робота з онлайновими документами в Google

Для роботи з електронними документами в **Документи Google** використовуються прості програмні засоби: текстовий он-лайн редактор **Google Documents**, табличний он-лайн процесор **Google Spreadsheets**, програма для роботи з он-лайн презентаціями **Google Presentations**, графічний он-лайн редактор **Google Drawings** і засіб створення форм для опитування.

Документи під час їх створення та опрацювання зберігаються автоматично через кожні 2–3 секунди, кожна правка фіксується, і можна скористатися функцією відміни та повернення змін так само, як і у звичайній локальній прикладній програмі. Робота користувачів відбувається в реальному часі, тобто якщо один з користувачів змінює вміст документа, то всі зміни одразу ж відображаються й у вікні інших користувачів. З одним і тим самим файлом одночасно може працювати до 200 користувачів, яких автор документа запросив до спільної роботи, вказавши його роль:

- **Співавтор** – може читати і редагувати документ, зберігати його копію на своєму комп'ютері, а також видаляти і додавати нових співавторів;
- **Читач** – може переглядати вміст останньої версії документа, а також зберігати її на своєму комп'ютері.

На роботу з електронними документами в **Документи Google** існує ряд обмежень, які стосуються розмірів і форматів файлів документів, що завантажуються, їх кількості тощо. Ознайомитися із цими обмеженнями можна на сайті **www.docs.google.com**, вибравши кнопку **Завантажити** та вибравши посилання **Типи файлів та обмеження розміру**.

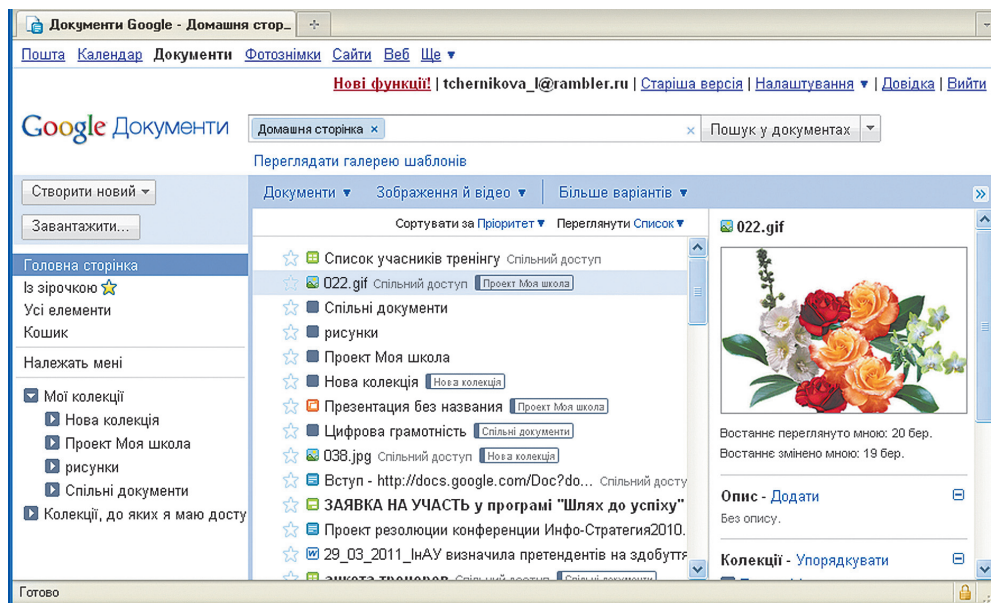
Для того щоб використовувати сервіс **Документи Google**, потрібно мати свій **Google**-акаунт. Як його створити, пояснювалось у попередніх пунктах підручника.

Щоб отримати доступ до он-лайн офісу потрібно:

1. Відкрити головну сторінку порталу **Google** **www.google.com.ua**.
2. Вибрати посилання **Увійти** в правому верхньому куті вікна.
3. Відкрити свій акаунт, увівши логін і пароль.
4. Вибрати посилання **Ще**.
5. Вибрати у списку команду **Документи**, що приведе до відкриття вікна **Документи Google**.

Вікно он-лайн офісу **Документи Google** (рис. 4.88) розділено вертикально на три частини. У лівій частині відображається область навігації зі списком доступних розділів сховища документів. Над областю навігації розміщено дві кнопки **Створити новий** (для створення нового документа засобами вбудованого програмного забезпечення) та **Завантажити** (для імпортування в он-лайн сховище раніше створених файлів, які збережені на локальному комп'ютері). У центральній частині вікна відображається вміст вибраного розділу сховища. Над ним розміщено кнопки для фільтрування файлів і сортування документів за типами, пріоритетами та іншими властивостями – **Документи**, **Зображення й відео**, **Більше варіантів** тощо. У правій частині вікна відображаються значення деяких властивостей виділених файлів і команди для їх налаштування.

Відкривши сховище, користувач потрапляє у розділ **Усі елементи**, тобто може побачити впорядкований список усіх об'єктів, які зберігаються в

Рис. 4.88. Вікно **Документи Google**

сховищі і до яких у нього є доступ. За наведення вказівника на будь-який документ біля його імені з'являється кнопка **Дії**, вибір якої відкриває список можливих операцій з файлом – видалення, перейменування, налаштування спільного доступу тощо.

Щоб створити новий файл засобами служби **Документи Google** потрібно:

1. Вибрати кнопку **Створити новий** в області навігації лівої частини вікна.
2. Вибрати у списку тип потрібного об'єкта – *документ, презентація, таблиця* тощо.
3. Створити вміст нового файлу засобами он-лайн програми.

Зберігати створений файл не потрібно, це буде зроблено автоматично, про що свідчитиме напис на кнопці **Збережено**.

Для завантаження в он-лайн сховище **Google** файлу з локального комп'ютера користувача потрібно:

1. Вибрати кнопку **Завантажити**.
2. Вибрати на сторінці посилання **Вибрати файли для завантаження**.
3. Вибрати потрібний файл у вікні **Загрузка файла**.
4. Вибрати послання **Вибрати більше файлів**, якщо потрібно завантажити кілька файлів.
5. Установити позначку прапорця для вибору способу конвертації файлів у формат документів **Google**.
6. Вибрати кнопку **Цільова колекція** для вибору папки, де буде розміщено файл.
7. Вибрати кнопку **Приватний** для вибору доступу користувачів до файлу.
8. Вибрати кнопку **Почати завантаження**.
9. Дочекатися завантаження файлу і вибрати посилання **Назад до Документи Google**.

Через деякий час після завершення завантаження файл відобразиться в списку документів.

Для впорядкування документів у сховищі, користувач може створювати нові папки (колекції):

1. Вибрати кнопку **Створити новий** в області навігації лівої частини вікна.
2. Вибрати у списку тип потрібного об'єкта – *Колекція*.
3. Увести в текстове поле ім'я нової папки.
4. Вибрати кнопку **ОК**.

Щоб перемістити об'єкти в деяку папку, слід перетягнути їх із центральної області вікна на назву потрібного розділу в області навігації. Для видалення файлу потрібно у списку кнопки **Дії** вибрати команду **Перемістити до кошика**.

Кожний документ, розміщений у сховищі **Документи Google**, можна відкрити для спільного читання та редагування кількома користувачами. Для цього потрібно:

1. Виділити файл і вибрати кнопку **Дії** біля документа.
2. Вибрати команду **Спільний доступ** ⇒ **Налаштування спільного доступу**, що відкриє відповідне вікно (рис. 4.89).
3. В області **Дозволи** ознайомитися із встановленими за замовчуванням правами доступу користувачів до файлу.
4. Вибрати за потреби посилання **Змінити**:
 - 1) Установити інші права доступу: *Загальнодоступний у мережі; Усі користувачі, які отримали посилання; Приватні.*
 - 2) Установити за потреби дозвіл на редагування файлу.
 - 3) Вибрати кнопку **Зберегти**.
5. Увести в текстове поле **Додати користувачів** електронні адреси тих, кому надається доступ до файлу.
6. Вибрати кнопку **Спільний доступ**.

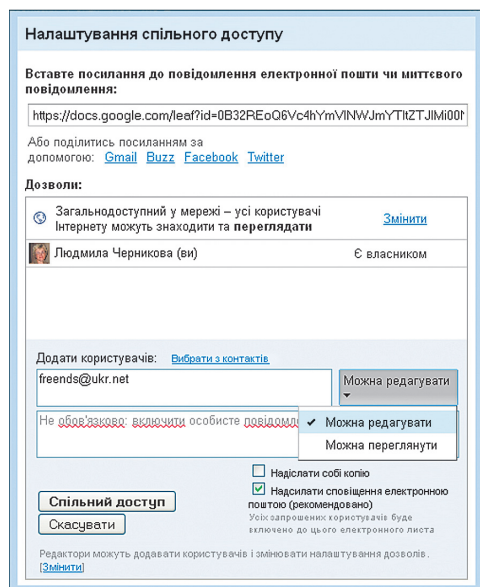


Рис. 4.89. Вікно **Налаштування спільного доступу**

Доступ до файлу може бути наданий й іншим користувачам, якщо ви повідомите їм адресу посилання на цей файл в Інтернеті (відображено у верхній частині вікна).

Відкривши спільний документ, співавтори засобами наданого програмного забезпечення можуть опрацьовувати документ у режимі реального часу. Кожен може додати в документ свої коментарі, виділяючи його кольором і додаючи своє ім'я (**Вставити** ⇒ **Коментарі**). За потреби, можна відмінити правки, які внесені співавторами. У процесі роботи з документом можна бачити, хто із співавторів одночасно з вами редагує даний документ. Повідомлення про це відображається над рядком меню.



Windows Live – набір безкоштовних он-лайн сервісів і локальних програм для обміну інформацією, організації спілкування між користувачами, використання різноманітних мультимедійних даних, пошуку та розміщення інформації в Інтернеті, спільної роботи з документами.

Він нараховує до 20 різних служб. Основний принцип, що проголошується творцями **Windows Live** (корпорація **Microsoft**): «Усі можливості Інтернету зібрано тут». Одні продукти та служби призначені для використання в Інтернеті, інші – для роботи на комп'ютері, куди їх необхідно завантажити з Інтернету.



Адреса сайту **Windows Live** – www.live.com.




Перевірте себе

- 1*. Поясніть, у чому суть спільного використання документів.
- 2*. Охарактеризуйте призначення та функції середовищ для спільної роботи з документами.
- 3*. Назвіть програмні засоби, які можна використати для колективної роботи над проектом. Охарактеризуйте їх можливості для комунікації учасників.
- 4°. Що таке віртуальний офіс? Яке його призначення?
- 5*. Опишіть можливості програми **Microsoft Office Groove 2007** для організації спільної роботи з документами.
- 6*. Опишіть можливості служби **Документи Google** для організації спільної роботи з документами.
- 7*. Поясніть, чим відрізняється використання програми **Microsoft Office Groove 2007** від середовища **Документи Google**.
- 8°. Що таке робоча область **Microsoft Office Groove 2007**? Які типи робочих областей можна використовувати?
- 9°. Назвіть ролі, у яких користувачі можуть отримувати доступ до спільних документів у **Microsoft Office Groove 2007**. Поясніть їх можливості.
- 10°. Які обмеження на файли існують у середовищі **Документи Google**?
- 11°. Файли яких типів можна створювати і розміщувати в **Документи Google**?
- 12°. У чому може полягати спільна робота над документом чи проектом?



Виконайте завдання

- 1*. Запустіть програму **Microsoft Office Groove 2007**:
 - 1) Налаштуйте свій обліковий запис.
 - 2) Створіть робочу область **Світ моїх захоплень** за типом **Стандартна**.
 - 3) Знайдіть в Інтернеті зображення за тематикою вашого хобі та збережіть їх у робочій області.
 - 4) Створіть в окремому документі список посилань на веб-ресурси, звідки були використані зображення.
 - 5) Розмістіть документ у робочій області.
 - 6) Установіть контакти з кількома вашими однокласниками і запросіть їх до робочої групи в ролі учасника.
 - 7) Надішліть учасникам групи текстове повідомлення в чаті з проханням переглянути матеріали у вашій папці спільного доступу та висловити свої думки щодо вашого захоплення на вкладці **Обговорення**.
 - 8) Приміть запрошення від двох ваших однокласників про участь у їх робочих групах.
 - 9) Відкрийте робочі області ваших однокласників і перегляньте їх матеріали.
 - 10) Залиште свої коментарі на вкладці **Обговорення**.
 - 11) Перегляньте коментарі ваших однокласників у вашій робочій області.
 - 12) Знайдіть серед однокласників однодумця по хобі.

- 13) Домовтеся з ним про створення спільної презентації про ваше хобі.
 - 14) Підготуйте три слайди презентації та збережіть її у своїй робочій області.
 - 15) Відправте повідомлення вашому співавтовру із запрошенням додати до вашої презентації його слайди.
 - 16) Додайте до списку посилань вашого співавтора свої посилання та збережіть оновлений документ у його робочій області.
 - 17) Перегляньте вашу спільну презентацію.
 - 18) Відправте повідомлення з подякою про дружню роботу.
 - 19) Закрийте вікно програми **Microsoft Office Groove 2007**.
-  20. Відкрийте свій акаунт на сайті **Google**:
- 1) Зайдіть на сервіс **Документи Google**.
 - 2) Створіть папку **Моя маленька батьківщина** та відкрийте до неї загальний доступ усім користувачам.
 - 3) Знайдіть в Інтернеті зображення цікавих місць вашої місцевості та завантажте їх у створену папку.
 - 4) Створіть в окремому документі список посилань на веб-ресурси, звідки було використано зображення, і розмістіть документ у спільній папці.
 - 5) Підготуйте три слайди презентації про ваш населений пункт і збережіть її в спільній папці.
 - 6) Надішліть учасникам групи електронною поштою запрошення для спільної роботи над презентацією, вказавши посилання на неї в Інтернеті.
 - 7) Приміть аналогічні запрошення від двох ваших однокласників.
 - 8) Перегляньте створені презентації ваших однокласників і додайте до них два своїх слайди.
 - 9) Додайте до списку посилань ваших однокласників свої посилання.
 - 10) Перегляньте спільну презентацію у вашому сховищі документів.
 - 11) Відправте подяку своїм однокласникам за дружню роботу, використавши служби миттєвого обміну повідомленнями.
 - 12) Закрийте свій акаунт і закрийте сайт **Google**.

Практична робота № 20. Розробка колективного проекту

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

Розробіть спільний проект на одну із запропонованих тем, присвячених закінченню школи: «Наша рідна школа», «Наш 11-й клас», «Наші вчителі», «Моя майбутня професія», «Куди піти вчитися?».

Для створення проекту:

1. Об'єднайтеся у групу з 3–4 осіб.
2. Обговоріть у групі тематику проекту, його структуру, складіть план розробки проекту, розподіліть ролі учасників проекту.
3. Підготуйте колективно різні складові проекту – плакати, буклети, календарі, текстові документи, презентації, відеоролики, звукові записи, публікації, стенди тощо, використавши для їх підготовки програмне забезпечення, що вивчалось в курсі інформатики.
4. Створіть засобами **Microsoft Office Groove 2007** або **Документи Google** спільний ресурс, де розмістіть складові проекту.
5. Перегляньте розроблені документи та внесіть у них свої корективи, обговоріть результати робіт, сплануйте колективний захист проекту, використавши засоби комунікацій для організації колективної роботи в групі.
6. Збережіть проект у спільній папці.
7. Презентуйте свій проект своїм однокласникам.

Зміст

Шановні одинадцятикласники!	3
Розділ 1. Інформаційні технології у навчанні	
1.1. Програмні засоби навчального призначення та особливості їх використання	4
<i>Практична робота № 1.</i> Використання програмних засобів під час вивчення фізики, хімії, біології	9
1.2. Математичні процесори. Програмно-методичний комплекс GRAN та його використання	10
<i>Практична робота № 2.</i> Автоматизація математичних обчислень	19
<i>Практична робота № 3.</i> Побудова графіків функції та їх аналіз	19
<i>Практична робота № 4.</i> Знаходження наближених значень розв'язків рівнянь та їх систем	19
Розділ 2. Основи алгоритмізації та програмування	
2.1. Моделі та моделювання. Типи моделей	20
2.2. Алгоритми та їх властивості. Форми подання алгоритмів	25
2.3. Комп'ютерні програми і мови програмування. Етапи розв'язування задач з використанням комп'ютера	31
2.4. Мова програмування Delphi та середовище розробки Turbo Delphi 2006	40
<i>Практична робота № 5.</i> Створення та виконання проекту. Використання елементів керування	51
2.5. Події та їх обробники	51
<i>Практична робота № 6.</i> Введення та виведення даних. Робота зі змінними	62
2.6. Налаштування програм	63
<i>Практична робота № 7.</i> Налаштування програм	68
2.7. Основні поняття математичної логіки	68
2.8. Розгалуження в алгоритмах і програмах	73
<i>Практична робота № 8.</i> Складання програм з розгалуженнями	83
2.9. Цикли в алгоритмах і в програмах	84
<i>Практична робота № 9.</i> Програмування циклічних обчислень	94
2.10. Процедури і функції користувача та стандартні процедури і функції в Delphi	95
<i>Практична робота № 10.</i> Використання підпрограм	101
2.11. Одновимірні масиви	102
<i>Практична робота № 11.</i> Опрацювання одновимірних масивів	114
Розділ 3. Бази даних. Системи управління базами даних	
3.1. Бази даних. Модель «сутність–зв'язок»	115
3.2. Моделі даних. Основні поняття реляційної бази даних	120

3.3. Етапи створення бази даних.	
Поняття про системи управління базами даних	127
3.4. Система управління базами даних	
Microsoft Office Access 2007	131
3.5. Відображення моделі «сутність–зв’язок»	
на базу даних. Робота з таблицями бази даних	140
<i>Практична робота № 12.</i> Розробка моделі «сутність–зв’язок»	
заданої предметної області. Створення бази даних	
у середовищі СУБД	151
3.6. Форми. Створення, редагування та форматування форм	152
3.7. Сортунання, пошук і фільтрування даних у базі даних	162
<i>Практична робота № 13.</i> Форми. Введення даних.	
Сортунання, фільтрування та пошук даних	168
3.8. Запити в Access 2007	168
3.9. Звіти. Створення звітів в Access 2007	174
<i>Практична робота № 14.</i> Створення запитів і звітів	182
 Розділ 4. Інформаційні технології персональної та колективної комунікації	
4.1. Поняття про мультимедійні дані та їхнє опрацювання	184
4.2. Додавання звукових і відеооб’єктів до слайдових презентацій	196
4.3. Створення та редагування відеофільмів (потоківих презентацій)	206
<i>Практична робота № 15.</i> Створення відеокліпу (потоківих презентацій)	217
4.4. Основи створення комп’ютерних публікацій. Програма Microsoft Office Publisher 2007	218
4.5. Особливості роботи з текстовими та графічними об’єктами в Publisher 2007 . Створення інформаційного бюлетеня та буклета	227
<i>Практична робота № 16.</i> Створення інформаційного бюлетеня та буклета	234
4.6. Інтегроване використання засобів опрацювання електронних документів	236
<i>Практична робота № 17.</i> Опрацювання даних у кількох програмних середовищах	247
4.7. Типи веб-сторінок. Класифікація веб-сайтів	247
4.8. Етапи створення веб-сайтів. Основи веб-дизайну	254
4.9. Засоби автоматизованої розробки веб-сайтів	261
<i>Практична робота № 18.</i> Автоматизоване створення веб-сайта	272
4.10. Технології та сервіси Веб 2.0 . Веб-спільноти. Створення блогів	273
<i>Практична робота № 19.</i> Створення та ведення власного блогу	288
4.11. Спільна робота з документами	289
<i>Практична робота № 20.</i> Розробка колективного проекту	300