ІНФОРМАТИКА

для загальноосвітніх навчальних закладів
з поглибленим вивченням інформатики

Підручник для 8 класу
загальноосвітніх навчальних закладів

Рекомендовано Міністерством освіти і науки України

Львів
Видавництво “Світ”
2016

УДК 004(075.3)
ББК 32.81я72
І-74

Авторський колектив:
А.М. Гуржій, Л.А. Карташова, В.В. Лапінський, В.Д. Руденко

Рекомендовано Міністерством освіти і науки України
(наказ МОН України від 10.05.2016 р. № 491)

Видано за рахунок державних коштів. Продаж заборонено

Експерти, які здійснили експертизу підручника під час проведення конкурсного відбору проектів підручників для учнів 8 класу загальноосвітніх навчальних закладів і дійшли висновку про доцільність надання підручнику грифа "Рекомендовано Міністерством освіти і науки України":

Михайловська Н.В., учитель Миколаївського муніципального колегіуму ім. В.Д. Чайки Миколаївської міської ради Миколаївської області, учитель-методист;
Фатюк Н.С., методист інформатики Гусятинського районного методичного кабінету Тернопільської області;
Дудка О.М., доцент кафедри інформатики ДВНЗ "Прикарпатський національний університет імені В. Стефаника", кандидат педагогічних наук.

І-74 **Інформатика** для загальноосвітніх навчальних закладів з поглибленим вивченням інформатики : підруч. для 8 кл. загальноосвіт. навч. закл. / А.М. Гуржій, Л.А. Карташова, В.В. Лапінський, В.Д. Руденко. – Львів : Світ, 2016. – 296 с. : іл., табл.

ISBN 978-966-914-001-2

Підручник призначений для навчання інформатики у 8 класі загальноосвітніх навчальних закладів. Зміст підручника відповідає навчальній програмі "Інформатика. 8–9 класи з поглибленим вивченням інформатики", затвердженій Міністерством освіти і науки України (рішення Колегії МОН України від 27.06.2013 р., протокол № 2/4-2).

УДК 004(075.3)
ББК 32.81я72

ISBN 978-966-914-001-2

Дорогі діти !

Більшість людей перестають помічати комп'ютери, приймаючи як належне їх непомітну присутність. Але комп'ютери дедалі частіше починають керувати пристроями, що є необхідною частиною середовища, в якому живе і навчається, працює людина. Звичайно, можна не знати як відбувається керування пранням у сучасній пральній машині – треба лише натиснути кнопку вибору відповідної програми. Але людина стала людиною розумною тому, що не лише користувалася знаряддями, а намагалася їх удосконалити, придумати нові.

У галузі інформаційних технологій, як і в усіх галузях науки, техніки та мистецтва, є знання, які не застарівають. У інформатиці це знання про принципи програмного управління, на основі яких побудована робота всіх комп'ютерів. Вивчаючи інформатику, ви докладніше ознайомитеся з конструкцією комп'ютера, почнете освоювати створення програм у середовищі, наближеному до того, яким користуються професійні програмісти. І зрозумієте, що це легко!

Отже, для того щоб ефективно використовувати блага цивілізації, придумані й створені для нас, потрібно хоча б у загальних рисах знати, як вони працюють. Ці знання допоможуть з'ясувати, що можна вимагати від певного гаджета, а що – ні, визначити, чи правильно він працює, тощо. А в майбутньому ви зможете усвідомлено обрати для себе вид діяльності, пов'язаний зі створенням нових засобів інформаційних технологій, або просто дуже ефективного їх використання.

Ви продовжуватимете вивчати правила безпеки життєдіяльності під час роботи з комп'ютером та іншими засобами інформаційних технологій.

Успіхів вам у навчанні й житті!

Автори

ЯК ПРАЦЮВАТИ З ЦІЄЮ КНИГОЮ

Для користування цим підручником обов'язковою є наявність у вашому розпорядженні персонального комп'ютера. Бажано підключити до комп'ютера описані в підручнику пристрої. На ньому має бути встановлено комплект програмних засобів, які описано в підручнику.

Навчальний матеріал поділено на дві частини – перша присвячена поглибленому вивченню сучасних інформаційних технологій, а друга – вивченню основ алгоритмізації та програмування. Нові терміни надруковані **жирним шрифтом.**

У підручнику використано такі підзаголовки та позначення:

	"Це ви вже знаєте" – короткий виклад знань, необхідних для засвоєння матеріалу розділу або підрозділу
	"Що вивчатимемо" – короткий виклад змісту розділу (підрозділу)
	"Важливе положення. Бажано запам'ятати"
	"Зверніть особливу увагу"
	"Для допитливих" – додаткові відомості
	"Словничок" – трактування термінів, які використовуються в розділі
	Рекомендується виконати (обговорювати) в колективі
	Рекомендується виконати вдома

Рівні складності завдань і запитань позначено так:

перший

другий

третій

РОЗДІЛ 1. МАТЕМАТИЧНІ ОСНОВИ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Загальні відомості про системи числення; арифметичні операції в позиційних системах числення; перетворення чисел із одної системи числення в іншу; способи подання чисел у комп'ютерах.

1.1. Поняття про системи числення. Позиційні та непозиційні системи числення

Системою числення називається сукупність правил запису (зображення, подання) чисел за допомогою символів (цифр) і виконання операцій над ними.

Розрізняють **непозиційні** та **позиційні** системи числення.

*У **непозиційних** системах числення значення цифри не залежить від її місця розташування в зображенні числа.* Такі системи досить складні для запису чисел. Нині використовується тільки римська система числення, в якій числа записуються за допомогою таких цифр: I (один), V (п'ять), X (десять), L (п'ятдесят), C (сто), D (п'ятсот), M (тисяча). Якщо кілька однакових цифр записано поряд, то їх значення додаються, наприклад, III (один + один + один = три), CC (двісті), XXX (тридцять). Якщо цифра, яка позначає менше число (молодша цифра), стоїть праворуч від цифри, яка позначає більше число (старшої), то її значення додається до старшої, наприклад VII (сім), XIII (тринадцять). Якщо молодша цифра стоїть ліворуч від старшої, то вона віднімається від неї, наприклад, IX (десять відняти один = дев'ять), XL (сорок).

*У **позиційних** системах числення значення цифри залежить не тільки від самої цифри, але й від її місця (позиції) у записі числа.*

Наприклад, у числі 64 значення цифри 6 дорівнює 60, а в числі 40,6–0,6 (шість десятих). Кожна позиція цифри називається **розрядом**. Для цілих чисел використовується така нумерація розрядів: молодший (із найменшими значеннями цифр) розряд цілого числа має номер нуль, а кожний наступний номер збільшується на одиницю. У дробовій частині, вираженій десятковим дробом, старший розряд має номер –1, а кожний наступний номер зменшується на одиницю (табл. 1). Таким чином, якщо ціла частина числа має n розрядів, а дробова – m розрядів, то старший розряд цілої частини має номер $<n-1>$, а молодший розряд дробової частини – номер $<-m>$.

Таблиця 1.1

Десяткове число	7	1	4	5	3	2	6	4	9
Номери розрядів	4	3	2	1	0	–1	–2	–3	–4

Основними характеристиками позиційних систем числення є: основа системи числення; вага розрядів; значення цифр, які використовуються в системі числення.

Основою системи числення (q) називають **кількість** цифр, які можуть використовуватися в кожному розряді числа. Основа системи числення визначає її назву (десяткова, вісімкова, двійкова тощо).

Вагою розряду називають число, на яке треба помножити значення розміщеної в ньому цифри.

Вага розрядів у позиційних системах зазвичай приймається рівною основі системи числення у степені, що дорівнює номеру розряду.

Наприклад, у десятковому числі 725 вага другого розряду дорівнює $10^2 = 100$, а в числі 0,4563 вага розряду –3 дорівнює $10^{-3} = 0{,}001$. Десяткова система числення є прикладом системи з **природним порядком** ваги розрядів, тобто такою, в якій у цілому числі вага кожного наступного розряду більша ніж вага попереднього розряду в кількість разів, що дорівнює основі системи числення, а у дробовій – менша у стільки ж разів. Застосовуються й системи числення з вагою розрядів, відмінною від природної. Таку вагу розрядів називають **штучною.**

Перевіряємо себе

1. Що таке система числення?
2. Що називається основою системи числення?
3. Які системи числення називають позиційними?
4. Чим відрізняється цифра від числа?
5. Що називається розрядом числа?
6. Поясніть, чому вираз “цифра відмінників у класі зросла” та інші подібні є неправильними.
7. Назвіть основні ознаки позиційних систем числення.
8. Поясніть, як визначається вага розрядів у позиційних системах числення.
9. Скільки цифр потрібно для подання чисел у вісімковій системі числення?

Виконуємо

1. Запишіть позначки годин на циферблаті римськими цифрами. Зробіть це для годинника, годинна стрілка якого робить один оберт за добу.
2. Для числа $144_{(10)}$ визначте значення кожного розряду.

3. Подайте числа $8_{(10)}$, $32_{(10)}$, $64_{(10)}$, $128_{(10)}$, $256_{(10)}$, $512_{(10)}$, $1024_{(10)}$ у двійковій системі числення. Поясніть, у чому особливість цих чисел. ★

4. Запишіть дату, коли ви виконуєте завдання, в форматі: місяць, число, рік – римськими цифрами, у двійковій, вісімковій, десятковій, шістнадцятковій системах числення. ★

1.2. Системи числення, що використовуються в обчислювальній техніці

✓ *У обчислювальній техніці основною системою числення є двійкова система зі символами 0 і 1.*

Крім двійкової застосовуються також десяткова, двійково-десяткова і шістнадцяткова системи числення.

Будь-який тип даних (текстовий, графічний, відео тощо) у комп'ютерних системах подається символами 0 і 1, бо подання й опрацювання даних здійснюється електронними пристроями, в яких електричні сигнали набувають лише двох значень **(бістабільними пристроями)**. Ці значення позначаються символами 0 і 1.

Наприклад, сигналу з напругою 3В відповідає значення 1, а сигналу напругою 0В – 0. Такі пристрої дуже надійні, позаяк встановлено однозначну відповідність між фізичними та математичними величинами.

✓ *Двійкова система числення порівняно з іншими системами найпростіше й надійніше реалізується технічно.*

✓ *У двійковій системі числення найпростіше виконуються арифметичні операції.*

Двійкова система має досить прості правила додавання:

0 + 0 = 0, 0 + 1=1, 1 + 0 = 1, 1 + 1 = (1) 0, (у дужках – одиниця, перенесена в старший розряд).

Таблиця множення двійкових чисел також проста:

$0 \cdot 0 = 0$, $0 \cdot 1 = 0$, $1 \cdot 0 = 0$, $1 \cdot 1 = 1$.

Простота виконання арифметичних операцій на папері означає, що й електронні схеми, які реалізують ці операції, також досить прості.

Електронні схеми з двома стійкими станами мають малий час переходу з одного стану в інший, і навпаки. Це означає, що в таких системах швидкість виконання арифметичних операцій висока.

Двійкова система (як і десяткова) є позиційною системою. Нагадаємо, що в таких системах кількісне значення цифри залежить як від значення цифри, так і від її розташування (позиції, розряду) в записі числа.

Наприклад, у десятковому числі 326,75 цифра 3 розміщена в другому розряді й позначає число з $3 \cdot 10^2 = 300$ одиниць, цифра 2 розміщена в першому розряді й позначає число з $2 \cdot 10^1 = 20$ одиниць, цифра 6 розмі-

щена в нульовому розряді й позначає число з $6 \cdot 10^0 = 6$ одиниць, цифра 7 розміщена в мінус першому розряді й позначає число з $7 \cdot 10^{-1} = 0{,}7$ одиниць, цифра 5 розміщена в мінус другому розряді й має $5*10^{-2} = 0{,}05$ одиниць. Подане десяткове число може бути записане у вигляді суми: $326{,}75 = 3 \cdot 10^2 + 2 \cdot 10^1 + 6 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$.

✓ *Для того щоб визначити кількісне значення цифри в позиційній системі числення, треба помножити цю цифру на основу системи в степені того номера розряду, в якому розміщена ця цифра.*

Наприклад, двійкове число 1011,01 можна записати у вигляді суми: $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 11{,}25$.

Таким чином, двійковий запис 1011,01 є десятковим числом 11,25. У табл. 1.2 наведені десяткові цифри, двійкове подання їх значень і двійковий запис чисел у вигляді сум.

Таблиця 1.2

Двійкове число	Запис двійкового числа у вигляді суми	Десяткове значення
0	$0 \cdot 2^0$	0
1	$1 \cdot 2^0$	1
10	$1 \cdot 2^1 + 0 \cdot 2^0$	2
11	$1 \cdot 2^1 + 1 \cdot 2^0$	3
100	$1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$	4
101	$1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	5
110	$1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$	6
111	$1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$	7
1000	$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$	8
1001	$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	9

У комп'ютерах крім двійкової і десяткової систем числення використовуються також двійково-десяткова та шіснадцяткова системи числення. У двійково-десятковій системі числення кожний розряд десяткового подання подається чотирма двійковими розрядами – тетрадою. Наприклад, десяткове число 618,74 має такий запис у двійково-десятковому поданні:

0110 0001 1000, 0111 0100 – двійково-десятковий запис,

6 1 8 , 7 4 – запис у десятковій системі числення.

Крайні ліві нулі в цілій частині та крайні праві нулі у дробовій частині можна на папері не писати, тобто те ж саме десяткове число у двійково-десятковій системі числення можна записати так: 11000011000,011101.

✓ *Чотири двійкових розряди називають* ***тетрадою.***

Щоб знайти десятковий еквівалент двійково-десяткового числа, потрібно в цілій частині ліворуч від коми й у дробовій частині праворуч від коми виокремити тетради і знайти їх десяткові значення. Неповні тетради доповнюються нулями.

Приклад:

0100 1000 1001, 0101 0010 – двійково-десяткове подання

Подане двійково-десяткове число дорівнює десятковому числу 489,52.

Щоб розрізняти, в якій системі подано число, праворуч від нього в дужках (або в нижньому індексі) інколи записують систему числення.

Наприклад: $101001_{(2)}$; $1011001{,}011001_{(2\text{-}10)}$. Перше число записане в двійковій системі числення, друге – у двійково-десятковій системі.

Таблиця 1.3

Шістнадцяткові символи	Двійкові значення	Десяткові значення
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Перших десять символів (цифр) шістнадцяткової системи числення збігаються з цифрами десяткової системи, а інші символи позначаються латинськими літерами A, B, C, D, E, F. У табл. 1.3 подано ці символи й відповідні їм двійкові та десяткові значення. Наведено кодування десяткових і шістнадцяткових символів двійковими значеннями, у яких розряди мають значення (вагу) 8, 4, 2, 1. Такий код (8421) називають **кодом прямого заміщення**. На практиці інколи використовуються інші коди, застосування яких визначається специфікою задач.

Перевіряємо себе

1. Чому двійкова система є основною в обчислювальній техніці?
2. Як нумеруються розряди числа?
3. Поясніть, як отримується значення числа в непозиційних системах числення.
4. Як відображаються значення десяткових цифр у двійковій системі числення?
5. Чи потрібна цифра 8 для відображення чисел у вісімковій системі числення?
6. Як записуються значення шістнадцяткових цифр у двійковій системі?

7. Як записуються десяткові числа у двійково-десятковій системі?

8. У якій системі числення подано число 1024,75? Чому не можна дати однозначну відповідь на це запитання?

Виконуємо

1. Запишіть десяткове число 93,64 у вигляді суми значень його розрядів.

2. Запишіть двійкове число 110111,011 у вигляді суми значень його розрядів (у десятковій системі числення).

3. Запишіть десяткове число 170,25 у двійково-десятковій системі числення.

4. Доведіть, чому запис 1011110111,011110 не може бути двійково-десятковим поданням числа.

5. Результат обчислень $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$ запишіть у двійковій системі числення.

6. Двійково-десяткове число 1011000,00100101 запишіть у десятковій системі числення, а після цього – у вигляді суми значень його розрядів.

7. Шістнадцяткове число A0E,B запишіть у двійковій системі числення.

1.3. Арифметичні дії в позиційних системах числення

Арифметичні операції в позиційних системах числення, в тому числі у двійковій системі, виконуються за тими ж правилами, за якими вони виконуються в десятковій системі числення.

Додавання чисел здійснюється послідовно, розряд за розрядом, починаючи з молодшого розряду, з перенесенням із розряду в розряд одиниць старшого розряду числа, що є сумою значень розрядів, які додаються.

Приклад. Додати двійкові числа 10111,01 і 110,111.

$$\begin{array}{r} 10111{,}01 \\ +\ 110{,}111 \\ \hline 11110{,}001 \end{array}$$

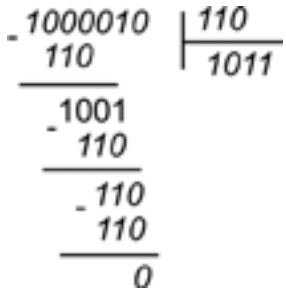
Приклад. Помножити двійкові числа 101,01 на 11,01.

$$\begin{array}{r} 101{,}01 \\ *\ 11{,}01 \\ \hline 10101 \\ +\ 10101 \\ \quad 00000 \\ \quad\ 10101 \\ \hline 10001{,}0001 \end{array}$$

У наведеному прикладі множення почалося з молодшого розряду множника, а часткові добутки зсувалися ліворуч. У комп'ютерах мно-

ження інколи виконується починаючи зі старшого розряду множника, при цьому часткові добутки зсуваються праворуч.

Приклад. Розділити двійкове число 1000010 на двійкове число 110.



Перевіряємо себе

1. За якими правилами виконуються арифметичні операції у двійковій системі числення?
2. Наведіть приклад додавання двох чисел у двійковій системі числення.
3. Наведіть приклад множення двох чисел у двійковій системі числення.
4. Наведіть приклад ділення двох чисел у двійковій системі числення.

Виконуємо

1. Додайте двійкові числа 100111,011 і 1111,001.
2. Від двійкового числа 100001,01 відніміть двійкове число 1101,011.
3. Помножте двійкове число 1101,01 на двійкове число 101,11.
4. Поділіть двійкове число 1110101 на двійкове число 1101.
5. Додайте вісімкові числа 57,3 і 26,1 і відніміть від першого друге.
6. Поділіть шістнадцяткове число $A0_{(16)}$ на число $14_{(10)}$, відніміть від числа $A0_{(16)}$ число $14_{(10)}$.
7. Додайте й помножте двійкові числа 110,1 і 1011,01.

1.4. Перетворення подання чисел із однієї системи числення в іншу

Існують різні методи перетворення подання чисел.

Перший метод можна вважати **універсальним**, тобто придатним для цілих, дробових і мішаних чисел.

Нехай число A, у якому в цілій частині міститься n-розрядів, а в дробовій – m-розрядів, подано в системі числення з основою q.

Алгоритм перетворення подання числа в системі числення з основою q ($A_{(q)}$) на подання цього числа в системі числення з основою p ($A_{(p)}$).

1. Число $A_{(q)}$ записати у вигляді суми значень його розрядів:
$A_{(q)} = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \ldots + a_1 \cdot q + a_0 + a_{-1} \cdot q^{-1} + \ldots + a_{-m} \cdot q^{-m}$.

2. Основу системи q і всі числа a_i записати в системі числення з основою p.

3. Виконати арифметичні операції кроку 2 в системі числення з основою p.

Приклад. Двійкове число 11011,01 подати в десятковій системі числення.

1. $1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$.
2. $1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$;
3. $16 + 8 + 4 + 1 + ¼ = 27{,}25$.

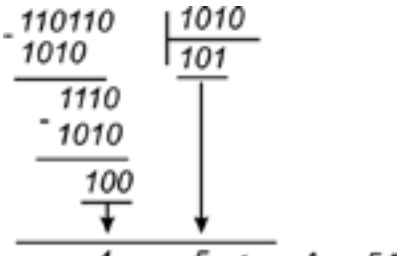
Приклад. Десяткове число 21,5 подати у двійковій системі числення.

1. $2 \cdot 10^1 + 1 \cdot 10^0 + 5 \cdot 10^{-1}$.
2. $010 \cdot 1010^1 + 001 \cdot 1010^0 + 101 \cdot 1010^{-1}$.
3. 10100 + 1+ 0,1 = 10101,1.

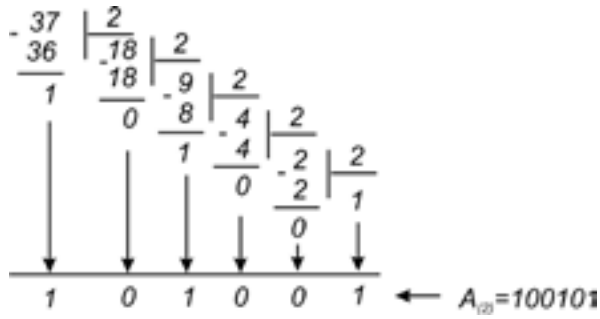
Другий метод перетворення подання чисел із однієї системи числення в іншу називається **методом ділення на основу**. Він придатний тільки для цілих чисел. Нехай ціле число $A_{(q)}$ потрібно перевести в систему з основою p. Перетворення здійснюється за наведеним нижче правилом.

Число $A_{(q)}$ ділиться на нову основу (p), яка представлена в системі q. Отримаємо остачу і цілу частину. Якщо ціла частина не менша за p, вона також ділиться на p. Цей процес здійснюється доти, доки остання остача стане менше p. Остачі від ділення – це і є цифри числа A за основою p. При цьому молодшою цифрою є перша остача, а старшою – остання.

Приклад. Число $A_{(2)}$ = 110110 подати в десятковій системі числення.



Приклад. Число $A_{(10)}$ = 37 подати у двійковій системі числення.

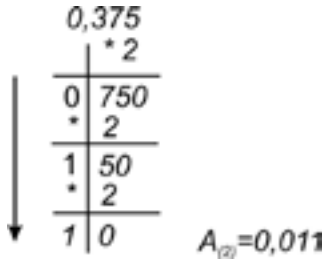


Третій метод називається методом множення на основу. Він використовується лише для дробових чисел. Перетворення здійснюється за наведеним нижче правилом.

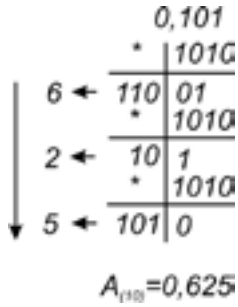
Помножити число A на основу p, яка подана в системі числення з основою q. Отримаємо цілу частину й дріб. Дріб також множиться на

основу p. Знову отримаємо цілу частину й дріб. Процес множення продовжується доти, доки дріб стане рівним нулю або меншим від числа, заданого як точність подання. Множення виконується в системі числення з основою q. Цифри, що відповідають цілим частинам чисел, які отримуються в процесі множення, і є цифрами числа A в системі з основою p.

Приклад. Число $A_{(10)} = 0{,}375$ подати у двійковій системі числення.



Приклад. Число $A_{(2)} = 0{,}101$ подати в десятковій системі числення.



Якщо основа першої системи числення є цілим степенем основи другої системи числення, то перетворення подання чисел між системами суттєво спрощується. Зокрема, такими системами є двійкова та шістнадцяткова системи числення ($2^4 = 16$). Степінь при основі два визначає кількість двійкових розрядів, необхідних для запису шістнадцяткових символів.

Приклад. Число $A_{(16)} = 4E5{,}CA$ у двійковій системі має такий запис: $A_{(2)} = 10011100101{,}1100101$.

Щоб записати двійкове число в шістнадцятковій системі числення, треба в його лівій частині ліворуч від коми й у дробовій частині праворуч від коми виокремити по 4 двійкових розряди і знайти їх подання в шістнадцятковій системі.

Приклад. Число $A_{(2)} = 101010000101{,}11011$ у шістнадцятковій системі має запис $A_{(16)} = A85{,}D8$.

Перевіряємо себе

1. Сформулюйте правило перетворення подання чисел із однієї системи числення в іншу методом ділення на основу.
2. Сформулюйте правило перетворення подання чисел із однієї системи числення в іншу методом множення на основу.
3. Сформулюйте правило перетворення подання чисел із однієї системи числення в іншу універсальним методом.
4. У чому полягає особливість перетворення подання чисел із однієї системи числення в іншу для систем, у яких основа однієї є цілим степенем основи іншої системи числення?

Виконуємо

1. Подання числа $A_{(2)} = 101011{,}1$ надайте в десятковій системі числення.
2. Подання числа $A_{(10)} = 75{,}375$ надайте в двійковій системі числення.
3. Подання числа $A_{(16)} =$ B 1,7 надайте в двійковій системі числення, а потім – у десятковій.
4. Подання числа $A_{(8)} = 57{,}625$ надайте в десятковій системі числення, а потім – у двійковій.
5.  Подання числа $A_{(2-10)} = 10010001{,}011000100101$ надайте в двійковій системі числення, а потім – у шістнадцятковій.
6.  Подання числа $A_{(10)} = 140{,}75$ надайте в шістнадцятковій системі числення, потім – у двійковій, а відтак – у десятковій.
7.  Знайдіть десяткове значення суми чисел $A_{(2)} = 111001{,}001$ і $B_{(2)} = 11101{,}01$.
8.  Знайдіть двійкове значення суми чисел $A_{(10)} = 28,\ 375$ і $B_{(8)} = 44{,}3$.

1.5. Способи подання чисел у обчислювальній техніці

Поширені три основних способи (коди) подання двійкових чисел. Далі користуватимемося такими назвами: прямий код (або значення зі знаком); зворотний код (або доповнення до одиниці); додатковий код (або додавання до двох).

Для цих способів крайній лівий розряд (старший розряд) у цілій частині числа дорівнює нулю для додатних чисел і одиниці – для від'ємних чисел. Надалі під час запису на папері знаковий розряд від інформативних відокремлюватимемо крапкою, а коди позначатимемо так: прямий – індексом пк, зворотний – індексом зк, додатковий – індексом дк, наприклад, $A_{пк}$, $A_{зк}$, $A_{дк}$. Додатні числа в усіх трьох кодах мають однакове зображення, від'ємні числа – різне.

Прямий код зазвичай використовується в процесі введення та виведення числових даних із комп'ютера, а також під час зберігання чисел у пам'яті комп'ютера.

У прямому коді числа подаються так само, як і в звичайному (на папері) запису, лише замість знака "+" у знаковому розряді записується нуль, а замість знака "–" у цей розряд записується одиниця. Наприклад, двійкове число 10110,01 у прямому коді записується так: $A_{пк} = 0.10110{,}01$, а від'ємне число $-100011{,}101$ – у такому вигляді: $A_{пк} = 1.100011{,}101$. Число нуль у прямому коді має два зображення: 0.00...0,0...0 і 1.00...0,0...0.

Зауважмо, що знаковий розряд у прямому коді несе інформацію тільки про знак числа і не має ніякого кількісного значення, тобто його вага дорівнює нулю.

Зворотний і додатковий коди позбавлені цього недоліку, тобто додавання та віднімання реалізується однією електронною схемою. Для цього перед виконанням операції віднімання знак числа, що віднімається, змінюється на протилежний. Наприклад, $(A) - (B) = (A) + (-B)$; $(A) - (-B) = (A) + (B)$.

Зворотний код. Знаковий розряд від'ємного числа має від'ємну вагу – $(2n-2-m)$, де n – кількість розрядів у цілій частині, m – кількість розрядів у дробовій частині.

Таке значення ваги обрано не випадково. З одного боку, воно забезпечує дуже простий перехід від прямого коду до зворотного й навпаки, а з іншого – дає змогу виконувати операцію додавання над кодами чисел за звичайними правилами. При цьому знакові розряди беруть участь у додаванні так само, як і інформаційні розряди.

Для перетворення подання від'ємного числа із прямого коду в зворотний необхідно знаковий розряд залишити без зміни, а в кожному інформаційному розряді замінити одиниці на нулі, а нулі – на одиниці.

Приклад. Число, зображене в прямому коді як $A_{пк} = 1.10010{,}011$, у зворотному коді зображається так: $A_{зк} = 1.01101{,}100$. Для перетворення зворотного коду від'ємного числа у прямий код виконуються аналогічні дії. Наприклад, число в зворотному коді $A_{зк} = 1.01001{,}010$ у прямому має значення $A_{пк} = 1.10110{,}101$.

Додавання чисел у зворотному коді виконується за звичайним правилом додавання, а саме: додавання починається з молодших розрядів із врахуванням одиниць перенесення з розряду в розряд. Одиниця перенесення, яка виникає зі знакового розряду, переноситься в наймолодший розряд раніше отриманої суми і додається до неї.

Кількість розрядів чисел, які додаються, має бути однаковою в обох числах. Вирівнювання кількості розрядів необхідно виконувати в прямому коді.

$$\begin{array}{r} A_{зк} = 1.101000{,}10 \\ B_{зк} = 0.110001{,}11 \\ \hline +\ 10.011010{,}01 \\ \hline 0.011010{,}10 \end{array} \quad \overset{\smile}{\longrightarrow} 1$$

Приклад. Додати в зворотному коді числа $A_{(2)} = -10111{,}01$ і $B_{(2)} = 110001{,}11$. Урівнюємо кількість розрядів чисел у прямому коді: $A_{пк} = 1.010111{,}01$ і $B_{пк} = 0.110001{,}11$. Перетворюємо числа у зворотний код і виконуємо їх додавання:

Додатковий код. Вага знакового розряду від'ємних чисел додаткового коду від'ємна й дорівнює $-2n$.

Для перетворення прямого коду від'ємного числа на додатковий код необхідно отримати його зворотний код, до молодшого розряду якого додати одиницю.

Приклад. $A_{(2)} = -10010{,}01$ записати в додатковому коді: $A_{пр} = 1.10010{,}01 \rightarrow A_{зк} = 1.01101{,}10 \rightarrow A_{дк} = 1.01101{,}11$.

Для отримання прямого коду з додаткового необхідно отримати його зворотний код (для цього слід відняти одиницю з молодшого розряду додаткового коду) й зі зворотного коду отримати прямий код. Приклад: $A_{дк} = 1.001010{,}011 \rightarrow A_{зк} = 1.001010{,}010 \rightarrow A_{пк} = 1.110101{,}101$.

Недоліком розглянутого методу є те, що для отримання прямого коду потрібно виконувати операцію віднімання. На практиці використовується інший метод, який полягає у створенні додаткового коду від додаткового. Тобто спочатку створюється зворотний код від додаткового, а потім до його молодшого розряду додається одиниця.

Приклад. $A_{дк} = 1.01100{,}010 \rightarrow [A_{дк}]_{зк} = 1.10011{,}101 \rightarrow A_{пк} = 1.10011{,}110$.

Додавання чисел у додатковому коді виконується так само, як і в зворотному коді, однак одиниця перенесення, що виникає зі знакового розряду, відкидається.

Зазначмо, що в комп'ютерах кількість розрядів у схемах для додавання фіксована. Тому під час додавання двох чисел однакових знаків результат може бути більшим від максимального числа, яке можна записати в розрядну сітку. У цьому випадку результат буде неправильним. Таке явище називається *переповненням розрядної сітки*.

Приклад. Додати числа $A_{(2)} = 110{,}01$ і $B_{(2)} = 11{,}111$.

$$\begin{array}{r} + A_{зк} = 0.110{,}010 \\ B_{зк} = 0.011{,}111 \\ \hline 1.010{,}001 \end{array}$$

У наведеному прикладі додавалися два додатні числа, а результат – від'ємний. Це означає, що відбулося переповнення розрядної сітки. Для виявлення переповнення розрядної сітки використовується спеціальна схема – індикатор, що аналізує знаки чисел і знак суми. Якщо обидва числа мають однаковий знак, а знак суми інший, то ця схема видає сигнал переповнення розрядної сітки й обчислення припиняється.

Перевіряємо себе

1. Сформулюйте правило запису чисел у прямому коді.
2. Наведіть правило перетворення чисел із прямого коду на зворотний і навпаки.
3. Сформулюйте правило додавання чисел у зворотному коді.
4. Наведіть правило перетворення чисел з прямого коду на додатковий і навпаки.
5. Сформулюйте правило додавання чисел у додатковому коді.

Виконуємо

1. Числа $A_{(2)} = 101{,}1$ і $B_{(2)} = -1001{,}01$ запишіть у зворотному й додатковому кодах.

2. Числа $A_{зк} = 0.10011{,}01$ і $B_{дк} = 1.10000.111$ запишіть у прямому коді.

3. Числа $A_{(2)} = -10{,}01$ і $B_{(2)} = 1100{,}011$ додайте в зворотному коді.

4. Числа $A_{(2)} = 11011{,}0011$ і $B_{(2)} = -1010{,}011$ додайте в додатковому коді.

5. Числа $A_{(2\text{-}10)} = 100001{,}01$ і $B_{(2\text{-}10)} = -11001{,}101$ додайте у двійковій системі числення в зворотному коді. Доведіть, що отримано правильний результат.

6. Числа $A_{(16)} = -A{,}3$ і $B_{(16)} = 1B{,}5$ додайте у двійковій системі числення в додатковому коді. Доведіть, що отримано правильний результат.

7. Числа $A_{дк} = 1.00011{,}101$ і $B_{дк} = 0.110{,}01$ додайте у зворотному коді.

8. Числа $A_{зк} = 0.01011{,}001$ і $B_{зк} = 1.1000{,}01$ додайте в додатковому коді.

ДЛЯ ДОПИТЛИВИХ

У математиці використовуються природна (з фіксованою комою) і нормальна (з плаваючою комою) форми подання чисел. У природній формі число подається у вигляді цілої і дробової частин, між якими ставиться кома.

У нормальній формі число зображується у вигляді **мантиси** й **порядку**. Наприклад, число A = 101011 може бути записане так: $0{,}00101011 \cdot 2^{1000}$, $0{,}101011 \cdot 2^{110}$ або $1{,}01011 \cdot 2^{101}$ та ін. Останній запис означає, що число нормалізоване.

Нормалізованими називаються числа, в яких кома стоїть праворуч після першої значущої цифри, тобто у двійковому записі завжди після одиниці.

Перевага подання чисел із фіксованою комою, порівняно з плаваючою комою, полягає в тому, що в ньому значно простіше виконуються арифметичні операції. Нормалізована форма дає змогу за однієї й тієї ж кількості розрядів представити значно більший діапазон чисел.

У сучасних комп'ютерах числа з фіксованою і плаваючою комами найчастіше подаються в форматі стандарту IEEE (англ.: Institute of Electrical and Electronics Engineers). Цей стандарт визначає не лише форми подання чисел, а й правила виконання арифметичних операцій, правила перетворення подання чисел із десяткового у двійкове і навпаки та інші дії над числами. Стандарт IEEE визначає дві форми подання чисел із плаваючою комою: з 32 розрядами – її називають формою з одинарною точністю, і з 64 розрядами – її називають формою з подвійною точністю.

Порядок у форматі 64 розрядів має надлишок 1023. 11-розрядний порядок відповідає десятковому порядку приблизно $10^{\pm 308}$, а 53-розрядна мантиса забезпечує майже таку ж точність, як і 16 десяткових цифр.

Практична робота № 1	**Тема:**	Перетворення чисел із десяткової системи числення в іншу і навпаки, з двійкової в шістнадцяткову і навпаки. Операції над числами в двійковій і шістнадцятковій системах числення
	Мета:	Набути практичних навичок подання й порівняння чисел у системах з різними основами

Завдання

1. Число $A_{(10)} = 107{,}625$ перетворити у двійкову, шістнадцяткову та двійково-десяткову системи числення.

2. Число $A_{(2)} = 101011101{,}0011$ перетворити в десяткову, шістнадцяткову та двійково-десяткову системи числення.

3. У режимі on-line перевірити правильність виконання операцій над числами, наведеними у п. 2 і п. 3. Скористайтеся для цього одним із сайтів, наприклад, http://math-ua.semestr.ru/inf/index.php.

4. Над числами $A_{(2)} = 1011011{,}101$ і $B_{(2)} = 11101{,}01$ виконати операції додавання і множення. Від числа A відняти число B.

5. Розділити число $A_{(2)} = 100011000$ на число $B_{(2)} = 1110$.

6. У режимі on-line перевірити правильність виконання операцій над числами, наведеними в п. 4 і п. 5. Використайте для цього, наприклад, сайт https://www.numsys.ru/.

7. Виконати операцію додавання в додатковому та зворотному кодах над числами $A_{(2)} = 10001010{,}001$ і $B_{(2)} = -1011{,}01$.

СЛОВНИЧОК

Вага розряду – основа системи числення в степені номера розряду.

Двійкова система числення – позиційна система числення зі символами 0 і 1.

Кількісне значення цифри – значення цифри, помножене на вагу розряду.

Основа системи числення – кількість цифр, які можуть використовуватися у кожному розряді числа.

Розряд числа – номер позиції цифри в записі числа.

Система числення – сукупність правил запису (зображення) чисел за допомогою символів (цифрових знаків) і виконання операцій над ними.

Тетрада – чотири двійкових розряди.

РОЗДІЛ 2. КОДУВАННЯ ДАНИХ

Будь-який реальний або уявний об'єкт і дії (процеси), що виконуються ним або над ним, можна описати, створивши повідомлення. Повідомлення може складатися з тексту і числових даних. Для того щоб із повідомлення можна було отримати інформацію, воно має створюватися за певними правилами.

Отримання й опрацювання даних як інформаційний процес. Кодування та декодування повідомлень. Двійкове кодування. Одиниці вимірювання довжини двійкового коду. Кодування символів.

2.1. Отримання, кодування і декодування даних

Для того щоб описати об'єкт або процес, передати відомості про нього таким чином, щоб створене повідомлення було зрозумілим для того, хто його отримуватиме, слід спочатку визначити певні правила. Ці правила мають описувати процес отримання відомостей про об'єкт, їх запису та відтворення. Щоб подати значення властивості об'єкта у вигляді числа (даних), її потрібно виміряти.

✓ *Вимірюваність властивостей об'єктів визначається через досвід людини і суспільства, через поняття «більше» і «менше».*

Для прямої властивість вимірюваності полягає в тому, що обмежені двома точками частини різних прямих (відрізки) можна порівнювати між собою, визначивши для них властивість "довжина". Для цього слід уважати за можливе порівняння відрізків, які належать різним прямим.

Для інших властивостей об'єктів також створюються засоби вимірювання, що відтворюють значення властивостей об'єктів.

Після цього маємо зберегти отримані дані, але таким чином, щоб вони були надалі доступними як для нас, так і для інших людей. Отже, отримані числа слід записати словами, подати у вигляді чисел, креслення тощо – створити повідомлення.

Повідомлення має бути передане або збережене як певний фізичний об'єкт зі зміненими властивостями. Наприклад, це може бути аркуш паперу з написаними на ньому цифрами і літерами, ієрогліфами, оптичний диск із дзеркальними та недзеркальними мікроскопічними ділянками.

✓ *Створення повідомлення за певними, наперед визначеними правилами, називається* ***кодуванням****.*

Властивості повідомлення як об'єкта, що містить інформацію, теж потрібно вимірювати. Для визначення способу і засобів вимірювання

властивостей повідомлення слід спочатку проаналізувати, як створюється й прочитується повідомлення.

Нехай у повідомленні описуємо колір деякого об'єкта. Використовуватимемо лише три фарби: червону (R – red), зелену (G – green) і синю (B – blue), які змішуватимемо для отримання потрібного кольору. Кожну фарбу братимемо лише в певній кількості або не братимемо взагалі. Тоді колір об'єкта можна описати значеннями трьох змінних – R, G, B, кожна з яких може набувати значення 0 – фарби цього кольору немає, 1 – взяти 1 одиницю фарби. Тоді повідомлення, в яких буде **закодований** колір, можуть мати такий вигляд: 111 – білий; 100 – червоний; 010 – зелений; 001 – синій. Можливими будуть і повідомлення 110; 011; 101; 000 – разом вісім різних повідомлень і відповідно вісім різних кольорів, отриманих трьома фарбами.

Зауважмо, що для запису цих повідомлень ми взяли лише дві цифри – 0 і 1. Отже, використовуючи для повідомлення **двійковий запис** числа довжиною три цифри (розряди), можна записати вісім значень кольорів – $2 \cdot 2 \cdot 2 = 2^3 = 8$.

Нехай для позначення кількості кожної фарби ми можемо використовувати не один, а два розряди. За такого кодування код 00 означає відсутність фарби, коди 01, 10 і 11 – 1/3, 2/3 і 3/3 одиниці кількості відповідної фарби. Двійковий запис матиме вже шість розрядів, а кількість кольорів, які можна буде описати, становитиме $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^6 = 64$.

Для того щоб ще точніше поінформувати про колір об'єкта, можна використати ще більше розрядів, тобто повідомлення більшої довжини. Приблизно так само можна подати й інші повідомлення – більшої точності подання властивості об'єкта можна досягти, збільшуючи кількість символів (розрядів числового подання) у повідомленні.

У обчислювальній техніці повідомлення передають і зберігають у пристроях, які можна уявити як набори вимикачів, кожен з яких може бути замкненим або розімкненим. Один вимикач відповідає одному розряду двійкового числа. Вимикачі реалізуються як фізичні об'єкти зі зміненими властивостями.

На лазерних дисках нулям і одиницям відповідають дзеркальні й недзеркальні **піти** (дуже маленькі ділянки диска), як показано на рис. 2.1.

Рис. 2.1. Піти на компакт-диску – об'єкти з двома можливими станами

✓ *Об'єкт, у якому зберігається один розряд двійкового числа, називається* ***бістабільним****; цей об'єкт (пристрій) може перебувати в одному з двох можливих станів – "так" або "ні", 0 або 1.*

Форма запису повідомлення у вигляді послідовності нулів і одиниць називається ***двійковим кодом*** *(повідомлення, даних, команд).*

Очевидно, що найменшою можливою довжиною двійкового коду повідомлення є вміст одного бістабільного пристрою, або **1 біт** (англ. bit – binary digit – двійкове число). Внутрішній запам'ятовуючий пристрій (ВЗП) більшості сучасних комп'ютерів складається з комірок пам'яті, які мають по 32 або 64 бістабільних пристрої, тобто можуть зберігати 32-розрядне або 64-розрядне двійкове число.

Одиниці розміру файлів, із якими вам уже доводилося працювати, є похідними одиницями від **біта**.

Вісім бітів становлять 1 байт (1 Б).
1024 байти становлять 1 кілобайт (1 кБ).
1024 кілобайти становлять 1 Мегабайт (1 МБ).
1024 Мегабайти становлять 1 Гігабайт (1 ГБ).
1024 Гігабайти становлять 1 Терабайт (1 ТБ).

Кратність одиниць **байт**, **кілобайт**, **мегабайт** і т.д. прийнято не 1000, як у Міжнародній системі одиниць (СІ), а 1024, бо це основа двійкової системи числення в степені 10 (2^{10}).

Задля того щоб уникнути непорозумінь із префіксами кратності (кіло, Мега, Гіга і т. д.), Міжнародна електротехнічна комісія (МЕК, англ.: IEC – International Electrotechnical Commission) у 1999 році прийняла поправку до міжнародного стандарту, в якій префікси кратності для двійкового подання величин називаються так: kibi- (Ki), mebi- (Mi), gibi- (Gi), tebi- (Ti) і т. д.

Попри те що зазначений стандарт і його доповнення прийняті й погоджені досить давно (IEC 60027-2, ISO / IEC IEC 80000-13: 2008), використання спеціальних префіксів кратності для величин, що відрізняються в $2^{10} = 1024$ рази, досить повільно впроваджується в практику.

Повідомлення може бути виміряне довжиною двійкового коду, яким воно записане (рис. 2.2).

Читаючи повідомлення (лист, цей текст, телеграму, опис алгоритму тощо), ми здійснюємо процес, який називається декодуванням.

Декодування – процес відновлення інформації, закодованої в повідомленні.

Для людини, яка не знає певної мови, лист, стаття в газеті тощо, подані незнайомою мовою, можуть не містити жодної інформації, крім того, що це якийсь текст. Але це жодним чином не означає, що в повідомленні відсутня інформація.

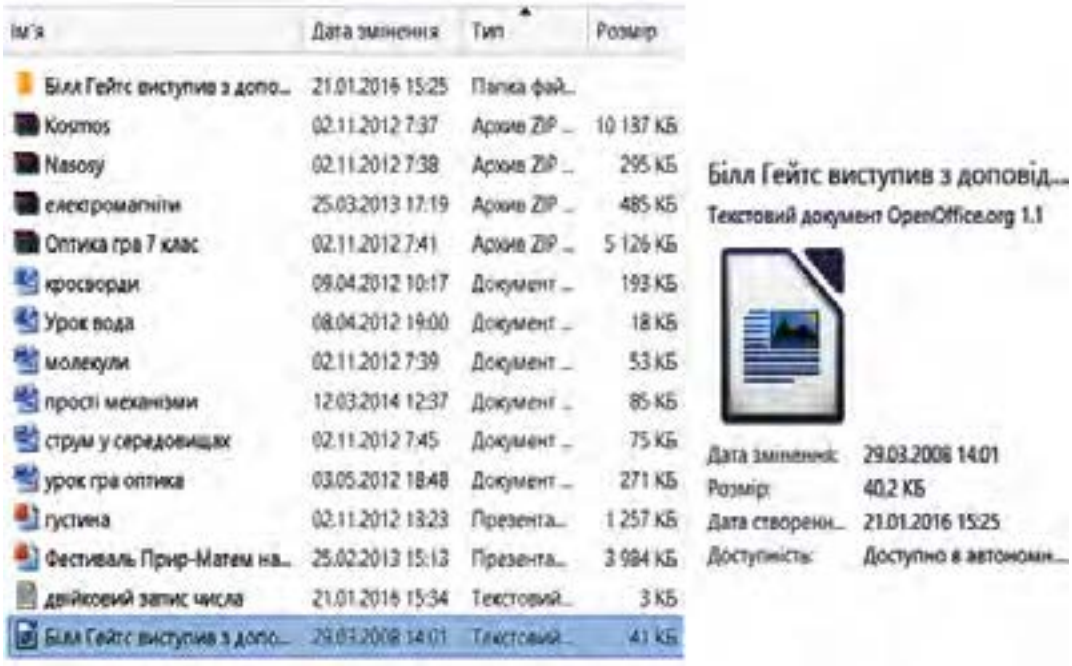


Рис. 2.2. Уміст папки з електронними документами

Зручно вважати, що отримувач повідомлення завжди знає, як його сприйняти, опрацювати, отримати з нього всю інформацію, яку воно містить. У такому випадку стає можливим порівняння кількості інформації в різних повідомленнях.

✓ *Для порівняння кількості інформації в повідомленнях використовують довжину їх подання у двійковому записі (двійковому коді), яка вимірюється в бітах, байтах і похідних від них одиницях.*

Зазначене трактування інформаційної місткості повідомлення використовується для визначення розмірів місця, необхідного для його розміщення на фізичному носії – магнітному або оптичному диску, флеш-накопичувачі.

Форми та способи подання інформації у повідомленнях, як і подання самих повідомлень, можуть бути різними (рисунки, ієрогліфи, піктограми, текст і дані, сформовані в документи на різних матеріальних носіях).

Окремим, нині дуже важливим, видом документа є електронний документ. Зважаючи на важливість для суспільства таких явищ, як електронний документ і електронний документообіг, їх означення та властивості, форми використання, основні інформаційні процеси, які відбуваються в процесах їх створення й застосування, означено нормативно (закон України "Про інформацію". – Відомості Верховної Ради України. – 1992. – № 48).

Перевіряємо себе

1. Що необхідно виконати для того, щоб отримати числове значення властивості об'єкта?
2. Як називається інформаційний процес, результатом якого є повідомлення зі значеннями температури в кількох різних приміщеннях?
3. У чому полягає сутність кодування даних у комп'ютері?
4. Назвіть основні особливості кодування даних у комп'ютерних системах.
5. Що називаться декодуванням даних у комп'ютерних системах?
6. Яким чином в Україні регламентується здійснення суспільно значущих інформаційних процесів?
7. Чому двійкова система є основною в обчислювальній техніці?
8. Як нумеруються розряди числа?
9. Що називається бістабільним пристроєм? Наведіть приклади.
10. Чому в одному кілобайті (кібібайті) не 1000, а 1024 байти?
11. У яких одиницях вимірюється ємність накопичувачів на магнітних дисках (вінчестерів)?
12. Скільки мегабайтів у одному гігабайті?
13. Скільки бістабільних пристроїв використовується для зберігання файла "двійковий запис числа.txt" (див. рис. 2.2)?
14. Чи завжди доцільно для порівняння кількості інформації у повідомленнях використовувати довжину їх подання у двійковому записі?

Виконуємо

1. Визначте ємність пристроїв зовнішньої пам'яті комп'ютера, за яким ви працюєте. Яка їх частина вже заповнена?
2. Визначте на рис. 2.2 найбільший і найменший файли серед файлів одного типу.
3. Запишіть довжини всіх файлів архівів у мегабайтах.
4. У скільки разів файл "двійковий запис числа.txt" менший за файл "прості механізми.docx" (рис. 2.2)?
5. Обчисліть, скільки потрібно двобайтових комірок пам'яті, щоб зберегти двійкове число 1010110101.
6. Обчисліть, скільки різних станів об'єкта можна закодувати, використовуючи три канали подання сигналів 0 або 1.
7. Латинський алфавіт має 26 літер. Обчисліть, скільки потрібно мати розрядів у комірці пам'яті, щоб закодувати ці літери (великі й маленькі), шість розділових знаків і цифри.

2.2. Текстові повідомлення та їх кодування

Тексти вводяться в комп'ютер зазвичай за допомогою клавіатури. На будь-якій сучасній клавіатурі комп'ютера розміщені символи англійського й національного алфавітів, синтаксичні та деякі спеціальні знаки, десяткові цифри, функціональні клавіші й клавіші керування (Ctrl, Enter та інші). Натиснення на клавішу або комбінацію клавіш передається для опрацювання в комп'ютер. Створений текстовий електронний документ зберігається спочатку в оперативному запам'ятовуючому пристрої (ОЗП) комп'ютера, а потім – на зовнішньому запам'ятовуючому пристрої (ЗЗП) або передається іншому користувачеві. Текст зберігається й передається у вигляді двійкового коду.

Відповідність між літерою та її числовим кодом має бути однакова для всіх комп'ютерів. Таке узгодження здійснюється через унормовування так званої таблиці кодування, тобто таблиці, в якій кожному символу (літері, цифрі, розділовому знаку, пробілу) і несимвольним командам (кінець рядка, кінець абзацу та іншим) поставлено у відповідність число.

Нині найпоширенішою є таблиця кодування ASCII (American Standard Code for Information Interchange – американський стандарт коду для обміну інформацією). У основному варіанті коду ASCII для будь-якого символу клавіатури використовується один байт.

Перша, або нижня, половина кодової таблиці, тобто коди молодших 7 бітів (коди 0–127), застосовується в усьому світі для кодування стандартного набору символів (символів латинського алфавіту, цифр, знаків арифметичних і логічних операцій та деяких інших). При цьому перші 32 комбінації (коди 0–31) призначені для команд керування комп'ютером і зовнішніми пристроями (наприклад, принтером). Другу половину таблиці (коди 128–255) використовують для кодування символів національних алфавітів, а також символів псевдографіки. У табл. 2.1 наведена нижня половина кодової таблиці ASCII за стандартом ANSI X3.4-1977, яку змінювати не можна.

✔ *Цю таблицю часто називають основною, або базовою, таблицею ASCII.*

Для того щоб не використовувати довгі двійкові коди, для нумерування символів застосовують шістнадцяткову систему числення, в якій до десяти арабських цифр 0, 2...9 додано ще шість A – 10, B – 11, C – 12, D – 13, E – 14, F – 15.

Як бачимо з табл. 2.1, наприклад, велика літера F має код 1000110 (у шістнадцятковій системі 46), символ 5 – код 0110101 (35), символ h – 1101000 (68). Великі та малі літери відрізняються лише значенням шостого розряду. Наприклад, символ A має код 1000001 (41), а символ a – 1100001 (61).

Таблиця 2.1

Основна, або базова, таблиця ASCII X3.4-1977

Номери розрядів 4↓ 3↓ 2↓ 1↓	7→	6→	5→	0	0	0	0	1	1	1	1
				0	0	1	1	0	0	1	1
				0	1	0	1	0	1	0	1
0	0	0	0	NUT	DLE	Пробіл	0	@	P	.	P
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	“	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	’	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	-
1	1	1	1	SI	US	/	?	O	C	o	DEL

Перші 32 кодові комбінації таблиці, тобто комбінації від $0000000_{(2)}$ ($00_{(16)}$) до $0011111_{(2)}$ ($1F_{(16)}$), відведені під символи керування. Вони не виводяться на екран, а використовуються для спеціальних цілей, зокрема, для передавання команд периферійним пристроям, наприклад, для управління друкарськими пристроями. Базова таблиця ASCII була застосована в комп’ютерах IBM PC для внутрішнього подання символів.

Позаяк семирозрядний код не забезпечував кодування національних алфавітів, стандартом ISO 646 уведено нову восьмирозрядну версію коду ASCII. Восьмий розряд надав ще 128 кодових комбінацій, які могли використовуватися з різною метою, в тому числі й для подання національного алфавіту.

Група комп’ютерних компаній розробила міжнародний стандарт 16-розрядного кодування ISO 10646 під назвою Unicode (Юнікод), який забезпечу 65536 кодових комбінацій.

Програми MS Windows Office підтримують це кодування починаючи з 1997 року. Юнікод має кілька версій. Найпоширенішими UTF (Unicode Transformation Format – формат перетворення Юнікоду), який застосовується для передавання символів через Інтернет, і UCS (Universal Character Set – універсальна таблиця символів).

Коди в системі Юнікод поділені на області. Коди від 0000 до 007F – це символи набору ASCII, а коди від 0400 до 052F відведені для символів кирилиці.

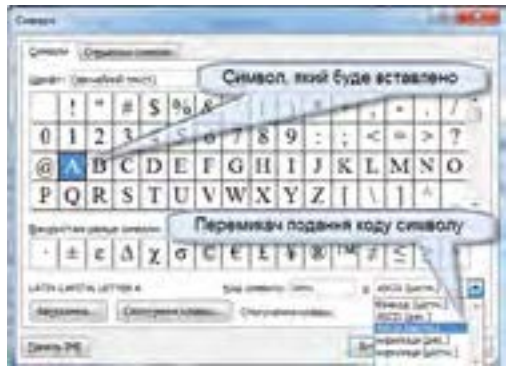


Рис. 2.3. Вікно вставлення символів редактора MS Word

Для визначення шістнадцяткового коду символу в системі Unicode (а також у інших системах кодування, наприклад, у ASCII (шістн.), кирилиця (шістн.) необхідно в MS Word на вкладці Вставлення натиснути кнопку **Символ**, потім – **Інші символи**. Відкриється вікно **Символ** (рис. 2.3), у нижній частині якого в шістнадцятковій системі числення виводиться код вибраного символу.

Перші коди кодової таблиці UNICODE (від 0000h до 00FF) відведені під код ASCII (базова таблиця і таблиця з англійськими літерами зі штрихами, так званий набір Latin-1). Ця таблиця поділена на блоки, кожний по 16 кодів. Кілька блоків утворюють зону. Зона може мати різну кількість блоків. Зона закріплюється за певною мовою. Наприклад, російська зона має 256 кодів.

Перевіряємо себе

1. Що називається кодовою таблицею?
2. Чому було прийняте кодування символів восьмирозрядними двійковими числами?
3. Де розташовані літери кириличних абеток у кодовій таблиці ASCII?
4. Скільки символів можна закодувати, використовуючи семирозрядне кодування?
5. Чому інколи замість літер кирилиці на деяких сайтах мережі Інтернет видно незрозумілі символи?
6. У якій кодовій таблиці можна закодувати більшу кількість символів – ASCII чи UTF?
7. Чому при впорядкуванні за абеткою прізвищ, поданих українською мовою, деякі програмні засоби виводять спочатку прізвища, які починаються не з А, а інших літер? Знайдіть з яких.

8. У списку є рядки, що починаються з символів кирилиці (великих і малих літер), латиниці (великих і малих літер), цифр. У якій послідовності програмний засіб, що використовує кодову таблицю ASCII, виводитиме рядки?

9. У комп'ютерах, які випускалися у 80-х роках у СРСР, використовувався стандарт кодування КОИ-7. Які проблеми з відтворенням кириличного тексту могли виникати?

Виконуємо

1. Заповніть таблицю шістнадцятковими кодами для наведених символів, для чого завантажте MS Word і відкрийте вікно **Символи** (рис. 2.3).

Символи	1	2	A	B	+	?	>	:
Юнікод								
ASCII								

Проаналізуйте отриману таблицю. Чим відрізняються коди наведених символів?

2. У редакторі Word відкрийте вікно **Символи**. У віконці **Шрифт** встановіть **звичайний шрифт** і виберіть таблицю **кирилиця** (шістн.). Використовуючи повзунок смуги прокручування, знайдіть символи кирилиці. Запишіть шістнадцяткові коди великих і малих літер В, Г, Д, Е, Ж.

3. Книжка має 450 сторінок. На кожній сторінці 30 рядків по 68 символів. Визначте обсяг пам'яті, потрібний для її збереження.

4. Знайдіть в Інтернеті таблицю кодування KOI8-U, яка є стандартом української Інтернет-спільноти, і таблицю KOI8-R. Визначте їх спільні риси та відмінності між ними.

5. Прийнято кодове повідомлення в системі кодування кирилиця (шістнадцяткове подання): 00C7 00E0 00E2 00F2 00F0 00F0 00E0 00F8 00F2 00EE 00F0 00E0. Розшифруйте прийнятий код.

6. Книжка зберігається в комп'ютері і займає 420 кБ. Кожна сторінка книжки містить 28 рядків по 75 символів у кодах ASCII. Визначте кількість сторінок у книжці.

7. Прочитайте в підрозділі "Для допитливих" описи кодування за методом Гауса і кодування за Бодо. Що спільного в цих методах? Чим вони відрізняються?

Підказка: зверніть увагу на кількість можливих комбінацій станів у повідомленні, яке відповідає одному символу.

2.3. Кодування графічних даних

У комп'ютерних системах графічні об'єкти створюються, редагуються, зберігаються й переглядаються за допомогою програмних засобів, які

називаються **графічними редакторами.** Плоска модель об'єкта, яка може бути відтворена на папері, називається **двовимірною (2D-графіка)** (рис. 2.4), а об'ємна, яка відтворюється спеціальними пристроями, – **тривимірною (3D-графіка).**

Рис. 2.4. Двовимірна графіка

Зображення графічних об'єктів (як і для інших типів даних) кодуються двійковими символами 0 і 1. У процесі кодування графічних даних та їх відтворення у вигляді зображень застосовуються два основних способи: **растровий** і **векторний.** Відповідно розрізняють растрові й векторні графічні редактори, а також тривимірні графічні редактори, які ґрунтуються на одночасному використанні растрового та векторного зображень.

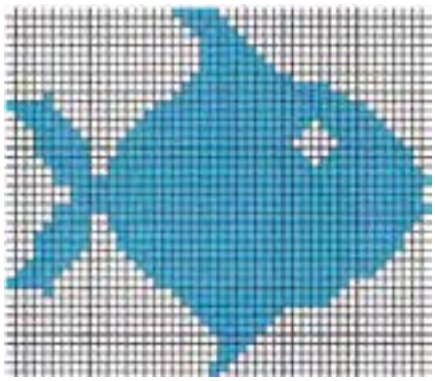


Рис. 2.5. Растрове зображення рибки

Растрове зображення – це зображення об'єкта окремими маленькими точками – пікселями (англ. picture cell – зображення, комірка). На рис. 2.5 подано приклад зображення об'єкта за допомогою темних і світлих комірок.

Для зображення об'єкта на екрані монітора за допомогою растрового способу екран поділяється на рядки та стовпці. Кількість рядків і стовпців у сучасних моніторах різна, наприклад, 800*600, 1024*768, 1240*1024, 1600*1200.

Екран засобу відтворення зображень, поділений на пікселі, називають **растром.**

Зображення об'єкта на екрані монітора створене точками.

Що менший розмір точки і що більше їх по горизонталі й вертикалі екрана, то краща якість зображення.

Важливою характеристикою монітора, яка впливає на якість зображення, є **роздільна здатність екрана.** Вона вимірюється кількістю пікселів на одиницю довжини, найчастіше на дюйм (1 дюйм = 2,54 см). Така одиниця позначається dpi (англ. dot per inch – точок на дюйм). У сучасних моніторах вона становить 72 або 96 dpi.

Описуючи двійкове кодування (підрозділ 2.1), ми як приклад розглядали отримання кольорів трьома фарбами з кількістю двійкових розрядів, виділених для кожної фарби, від одного до трьох. Нині для кодування кольору пікселя застосовуються 8 біт ($2^8 = 256$), 16 біт ($2^{16} = 65536$),

24 біти (2^{24} = 16777216). У деяких моніторах для кодування кольору відводиться до 48 біт.

Кількість бітів, що використовується для кодування пікселів, називається глибиною кольору.

Растрове зображення застосовується в поліграфії, медицині, рекламній діяльності, фотографії. Цей тип зображення створюється сканерами, медичною апаратурою, цифровими фотоапаратами і відеокамерами.

Векторне зображення об'єктів складається з ліній, прямокутників, еліпсів, багатокутників тощо, які називають **графічними примітивами**. Базовим елементом графічних примітивів є лінія. Лінія, як реальний об'єкт, має властивості: товщину, колір, спосіб накреслення (пунктир, суцільна тощо). Графічні примітиви описуються математичними формулами. Наприклад, об'єкт квадрат можна описати так: центр – 80, 60 (координати x, y), сторона – 20, лінія – суцільна, товщина – 0,50, заливка – відсутня. Розмір файла графічного об'єкта не залежить від розміру об'єкта, а залежить від кількості графічних примітивів, за допомогою яких описується об'єкт.

Векторне подання графічних об'єктів зазвичай потребує меншого обсягу пам'яті для їх збереження, ніж растрове.

У цьому полягає його основна перевага порівняно з растровим.

Перевага векторного подання зображення полягає в простоті його масштабування, тобто зміні розміру об'єкта без втрати якості його зображення. Наприклад, для зміни розміру квадрата досить вказати новий розмір його сторони. Для растрового зображення змінити розмір будь-якого об'єкта можна лише на піксельному рівні, тобто змінивши кількість пікселів на одиниці розміру.

Растрове зображення доцільно використовувати для об'єктів зі складними гамами кольорів, відтінків і форм. Такими об'єктами, наприклад, є світлини, малюнки, відсканові дані. Растрове зображення вже давно використовується в поліграфії. Переглянувши, наприклад, за допомогою оптичної лінзи текст газети, ви побачите, що літери складаються з окремих точок. Векторне зображення краще застосовувати для креслень і зображень із простими формами, тінями та кольором.

Векторні й растрові зображення відтворюються на моніторах, лазерних і струменевих принтерах, які за принципом дії є растровими. Для відтворення на растрових пристроях векторних зображень вони перетворюються на набори пікселів. Процес перетворення здійснюється з урахуванням масштабу зображення і роздільної здатності пристрою.

Зображення, яке відтворюється на екрані монітора, зберігається у відеопам'яті. Кожній точці екрана відповідає певна комірка цієї пам'яті.

Для того щоб змінити зображення на екрані монітора, потрібно змінити вміст відеопам'яті. Ця функція програмно виконується процесором відеокарти.

Уміст комірок відеопам'яті, які призначаються для кожного пікселя, визначає його властивості: колір, яскравість. Якщо пікселю надається один біт (одна комірка), то можливі два її стани: одиниця – піксель висвічується, нуль – не висвічується.

На рис. 2.6 подано схему, яка пояснює виведення інформації для випадку, якщо дані про піксель займають 1 біт. Якщо стан комірки дорівнює 1, то піксель світиться, якщо біт дорівнює 0, то він не світиться.

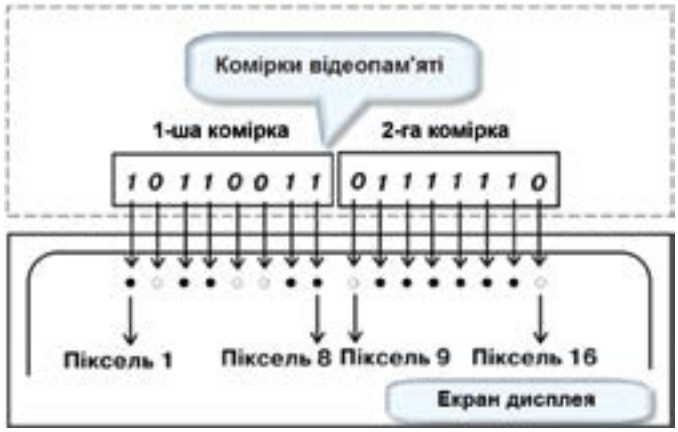


Рис. 2.6. Схема подання зображення у відеопам'яті

Розглянутий спосіб відображення називають лінійним, тому що лінійній послідовності пікселів відповідає лінійна послідовність бітів.

На рис. 2.7 подано приклад зображення стрілки лінійним способом.

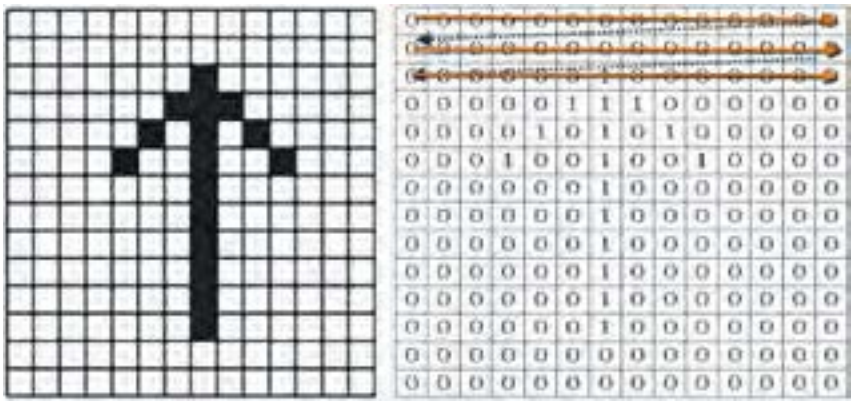


Рис. 2.7. Растровий спосіб утворення зображення стрілки

Нехай растр екрана складається з 800*600 = 480000 пікселів і на кожний піксель виділяється 1 біт (піксель світиться або ні). Тоді мінімальний обсяг відеопам'яті для збереження одного кадру має бути

480000* /8 = 60000 байт ≈ 58,6 кБ. У сучасних комп'ютерах на кожний піксель надається 24 біти (по 8 біт на кожний базисний колір), що забезпечує 2^{24} = 16,7 млн кольорів (True Color – істинний колір). У цьому випадку для моніторів із растром 800×600 пікселів мінімальний обсяг відеопам'яті має бути: 800×600×24 біт = 1406,25 кБ. У більшості сучасних комп'ютерів глибина кольору становить 32 або 48 біт. Фізичний обсяг відеопам'яті сягає від 0,5 до 4 ГБ і більше. У відеопам'яті зазвичай створюється кілька сторінок. При цьому в кожний окремий момент часу актуальною (такою, вміст якої відображається на екрані) є лише одна сторінка. Сторінкова організація відеопам'яті дає змогу більш ефективно організувати її взаємодію з процесором.

Рис. 2.8. Кольорова модель RGB

Кольорове зображення будується на основі певної кольорової моделі, тобто способу відтворення кольорів. У електронних пристроях, зокрема моніторах, найчастіше застосовується модель RGB (Red – червоний, Green – зелений, Blue – синій), а в поліграфії – модель CMYK (Cyan – блакитний, Magenta – пурпуровий, Yellow – жовтий, Black – чорний).

Модель RGB. Ми розглядали її в підрозділі 2.1. На рис. 2.8 зображено результати змішування в рівних пропорціях базових кольорів.

✓ *Від змішування в однакових кількостях* ***двох*** *базових кольорів моделі RGB отримані такі кольори: Yellow – жовтий; Magenta – пурпуровий; Cyan – блакитний, а* ***трьох*** *– White – білий.*

Рис. 2.9. Кольорова модель CMYK

У табл. 2.2 подано двійкові коди базових кольорів і результатів їх змішування.

Модель CMYK. Базовими кольорами цієї моделі є блакитний, пурпуровий, жовтий і чорний. Колір у цій моделі задається чотирма числами від 0 до 100, кожне з яких визначає інтенсивність базового кольору, наприклад: (40, 52, 60, 5). Перше число визначає відсоток блакитного кольору (Cyan), друге – пурпурового (Magenta), третє – жовтого (Yellow), четверте – чорного (Black). На рис. 2.9 показано базові кольори моделі CMYK і результати їх змішування.

Таблиця 2.2

Результуючий колір	Базові кольори		
	Червоний	Зелений	Синій
Червоний	11111111	00000000	00000000
Зелений	00000000	11111111	00000000
Синій	00000000	00000000	11111111
Блакитний	00000000	11111111	11111111
Пурпуровий	11111111	00000000	11111111
Жовтий	11111111	11111111	00000000
Білий	11111111	11111111	11111111
Чорний	00000000	00000000	00000000

Зазначмо, що в кольоровій моделі CMYK можуть бути подані не всі кольори, доступні в моделі RGB, і навпаки. Наприклад, найяскравіші кольори моделі RGB не можна відтворити в моделі CMYK. Водночас найтемніші кольори моделі CMYK не можна відтворити на папері за допомогою моделі RGB.

Файли, в яких зберігаються дані графічних об'єктів, мають великий обсяг. Особливо це стосується растрових зображень. Для зменшення обсягу таких файлів застосовуються спеціальні методи стиснення даних. Способи стиснення і збереження даних визначають **формати графічних файлів**. Найпоширеніші формати графічних файлів – BMP, JPEG, TIFF, GIF, PNG.

BMP (Bitmap). Формат растрових графічних зображень в ОС Windows. Дані зберігаються без стиснення. Кодування кольору пікселів виконується з глибиною 1, 4, 8 або 24 біти.

GIF (Graphics Interchange Format). Формат растрових зображень із глибиною кольору тільки 8 біт, тому файли цього формату досить компактні.

PNG (Portable Network Graphics). Використовується в основному для зображень, які розміщують в Інтернеті. Стиснення не призводить до втрати якості.

JPEG (Joint Photographic Expert Group). Доцільно застосовувати для збереження світлин. Інколи стиснений файл досягає 10 % необхідної пам'яті початкового. Під час запису зображення в форматі JPEG користувач може визначати показник стиснення. Зауважмо, що більший показник стиснення, то нижча якість зображень. Зазначмо, що JPEG, GIF і PNG є основними графічними растровими форматами Інтернету.

TIFF (Tagged Image File Format). Формат призначений для збереження растрових зображень високої якості. Це один із основних стандартів професійної поліграфії. Формат дає змогу кодувати будь-яке зображення без втрати якості.

Формат **PSD** (PhotoShop Document) – один із найпотужніших форматів растрової графіки. Глибина кольору може становити 48 біт. Недолік формату полягає в тому, що існуючі алгоритми не забезпечують високий ступінь стиснення.

Формат **WMF** (Windows Metafile) підтримує векторну і растрову графіку додатками середовища Windows. Глибина кольору 8 або 24 біти.

Популярні також універсальні комбіновані формати, в яких зберігають одночасно зображення і текст. Прикладом такого формату є **PDF** (Portable Document Format), який широко застосовується в поліграфії.

Перевіряємо себе

1. Які методи використовуються для кодування графічних даних?
2. Поясніть сутність растрового зображення об'єктів.
3. Що таке піксель?
4. Що таке растр?
5. Поясніть зміст поняття «роздільна здатність екрана».
6. Скільки бітів може використовуватися для кодування кольору?
7. Що називається глибиною кольору?
8. Поясніть сутність векторного зображення.
9. Які переваги та недоліки має векторне зображення порівняно з растровим?
10. У яких випадках доцільно застосовувати растрове зображення?
11. Поясніть сутність кольорової моделі RGB.
12. Який обсяг має мати відеопам'ять, якщо растр дорівнює 1280*1024, а для збереження кольору пікселя відведено 16 біт?
13. Поясніть сутність кольорової моделі CMYK.
14. Яким чином пов'язані кольорові моделі RGB і CMYK (через які кольори)?

Виконуємо

1. На екрані відображається малюнок розміром 5*6 см. Один квадратний сантиметр екрана містить 24*24 пікселів, а глибина кольору дорівнює 16 біт. Визначте теоретичний обсяг пам'яті, потрібний для збереження малюнка.

2. Довжина екрана монітора по горизонталі дорівнює 32 см, а кількість пікселів – 1024. Визначте роздільну здатність екрана в dpi.

3. Завантажте графічний редактор Paint і відкрийте будь-який графічний файл. Відкрийте вікно **Властивості** цього файла. Зверніть увагу, що роздільна здатність екрана дорівнює 96 точок на дюйм, увімкнутий перемикач **пікселі**, ширина зображення на екрані – 500 пікселів, висота – 444.

Обчисліть розмір зображення в сантиметрах і дюймах, потім увімкніть перемикачі **дюйми** і **сантиметри** й переконайтеся, що обчислення виконано правильно, в іншому випадку знайдіть свої помилки.

4. Повноекранний растр 17-дюймового екрана монітора має 1280*1024 пікселів. Визначте його роздільну здатність у dpi.

Підказка. Для отримання ширини і висоти екрана скористайтеся теоремою Піфагора.

5. Запустіть графічний редактор Paint і відкрийте один із малюнків. Обчисліть теоретичний обсяг необхідної для нього пам'яті й порівняйте його з реальним обсягом. Обґрунтуйте різницю між ними.

6. Малюнок по горизонталі та вертикалі має розмір 600*400 пікселів. Цей малюнок виводиться на принтер, роздільна здатність якого дорівнює 600 dpi. Який розмір він матиме на папері (у міліметрах)?

Підказка. Роздільна здатність принтера 600 dpi означає, що на кожному дюймі паперу розміщується 600 точок. На кожну точку відводиться один піксель малюнка. Тому розмір малюнка по горизонталі (без масштабування) дорівнює (600/600)*2,54=2,54 см, а по вертикалі (400/600)*2,5=1,27 см. Спроба збільшити розмір малюнка в процесі його друкування призводить до погіршення якості зображення, тому що кількість пікселів на малюнку при цьому не збільшується. Наприклад, якщо описаний малюнок збільшити вдвічі, тобто надрукувати його розміром 5*2,5 см, то кількість пікселів у ньому не зміниться, а їх розмір збільшиться, а отже, його якість буде гірша. Це пояснює те, що друкування фотографій або малюнків із веб-сторінки часто має низьку якість.

7. Зображення має розмір 400*200 пікселів. Який він матиме розмір на екрані монітора, роздільна здатність якого 72 ppi?

Підказка. Значення роздільної здатності 72 ppi визначає, що на кожний його дюйм (2,54 см) припадає 72 фізичних пікселі (світлі точки). На кожний піксель малюнка відводиться один фізичний піксель екрана. Тому розмір малюнка на екрані по горизонталі визначається за формулою (кількість пікселів у малюнку по горизонталі/72)*2,54 см.

8. Визначте мінімально необхідний обсяг відеопам'яті для монітора з растром 800*600 і глибиною кольору 16 біт, якщо вважати, що у ній розміщується повне зображення екрана.

2.4. Кодування звукових даних

Звуки, які ми чуємо, є коливаннями повітря або іншого пружного середовища. Змінний тиск повітря у вусі людини діє на нерви, які передають сигнал до мозку. Для опрацювання звуків електронними пристроями (у тому числі комп'ютером) змінний тиск повітря перетворюється на електричний сигнал пристроєм, який називається **мікрофоном**. Звук передається в середовищі у вигляді механічної хвилі.

Щоб розуміти, як звук опрацьовується в комп'ютері, потрібно знати, які величини описують коливання. Залежність миттєвого значення тиску від часу можна зобразити на графіку. Для простого звукового коливання (наприклад, від камертона) цей графік має вигляд, показаний на рис. 2.10. Такі коливання називають **гармонічними**, або **синусоїдальними**.

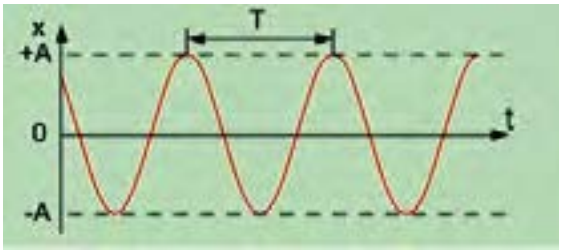


Рис. 2.10. Синусоїдальне коливання

Гармонічні коливання характеризуються такими параметрами: **амплітуда, період, швидкість, частота.**

Амплітуда (A) – максимальне значення величини тиску, що змінюється (див. рис. 2.10). Що більша амплітуда коливань, то більша гучність звуку.

Період (T) – проміжок часу, за який здійснюється одне повне коливання (або час, за який хвиля пробігає шлях, що дорівнює її довжині). Вимірюється в секундах (с).

Частота (f – англ. *frequency*, частота) – кількість повних коливань, що здійснюються за одиницю часу. Кількість таких коливань за одну секунду вимірюється в герцах (Гц). Один герц – це одне повне коливання за одну секунду.

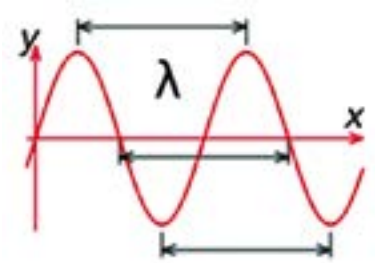


Рис. 2.11. Характеристики хвилі

Звук, як уже зазначалося, поширюється у вигляді хвилі (рис. 2.11). Хвиля характеризується швидкістю поширення і довжиною.

Швидкість (v – англ. *velocity*) – переміщення хвилі в напрямку її поширення (вимірюється в метрах за секунду, м/с). Швидкість звуку в повітрі у середньому становить 340 м/с (1200 км/год). У повітрі швидкість звуку менша, ніж у рідині, а в рідині менша, ніж у твердому середовищі.

Довжина хвилі (λ) – відстань, яку *проходить* хвиля за час, що дорівнює періоду коливань T (вимірюється в метрах).

Звуковими **називають такі хвилі у пружних середовищах, що мають частоти від 16 Гц до 20000 Гц.**

Саме такий діапазон частот сприймає вухо людини. Природні звуки (шум моря, щебет пташок, шелест листя тощо), а також музичні звуки, звуки мови є складними і часто виникають унаслідок накладання різних

хвиль. Звукові сигнали сприймає й опрацьовує мозок людини, на основі чого розпізнаються повідомлення.

✓ *Основні співвідношення між характеристиками хвилі:*

$$f = 1/T;\ \lambda = \upsilon \cdot T = \upsilon/f;\ \upsilon = \lambda/T;\ \upsilon = \lambda \cdot \upsilon.$$

2.5. Оцифровування звуку

Для збереження й опрацювання звукових даних у комп'ютері необхідно насамперед перетворити звукові хвилі на електричні сигнали. Пристроєм, що здійснює таке перетворення, є мікрофон. На рис. 2.12 показано приклад зміни напруги на виході мікрофона. Як бачимо, напруга змінюється в часі плавно, безперервно. Таку форму подання сигналу називають **аналоговою.**

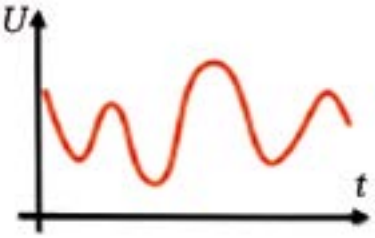


Рис. 2.12. Напруга на виході мікрофона

Оскільки комп'ютер опрацьовує тільки двійкові сигнали, то необхідно перетворити сигнал з аналогової форми на дискретну, тобто виконати **аналого-цифрове перетворення** (або оцифровування звуку).

Оцифровування виконується в три етапи: дискретизація в часі; квантування за значенням (вимірювання миттєвого значення); кодування сигналу.

Дискретизація – процес вимірювання значень аналогового сигналу через рівні проміжки часу, який називається **кроком дискретизації**. Сутність процесу дискретизації ілюструється рис. 2.13, де подано результати вимірювання миттєвих значень сигналу.

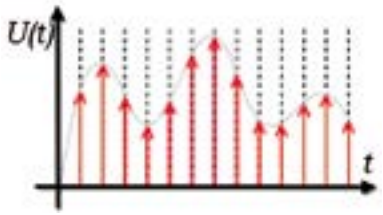


Рис. 2.13. Дискретизація аналогового сигналу

✓ *Кількість вимірювань величини сигналу за одну секунду називають* ***частотою дискретизації.***

Що менший крок дискретизації, то вища її частота і точніше буде представлена аналогова хвиля дискретною. Для звукових хвиль поширеною частотою дискретизації в комп'ютерах є 44 кГц. Однак у деяких сучасних комп'ютерах вона досягає 192 кГц і навіть 384 кГц.

Квантування величини сигналу – процес заміни реального значення сигналу, отри-

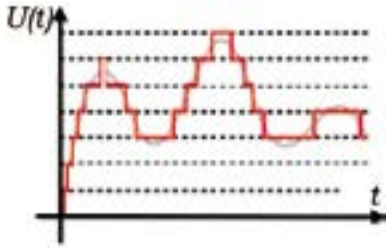


Рис. 2.14. Квантування сигналу

маного в результаті вимірювання, найближчим значенням із набору фіксованих значень, які називають **рівнями квантування**. Його сутність полягає в тому, що значення напруги визначаються через рівні проміжки часу. Процес квантування пояснюється прикладом, поданим на рис. 2.14.

Як бачимо з рис. 2.14, реально виміряні значення сигналу, отримані в процесі дискретизації, замінюються найближчими фіксованими значеннями.

Кодування – процес присвоєння кожному фіксованому значенню сигналу двійкового коду. Приклад кодування подано на рис. 2.15.

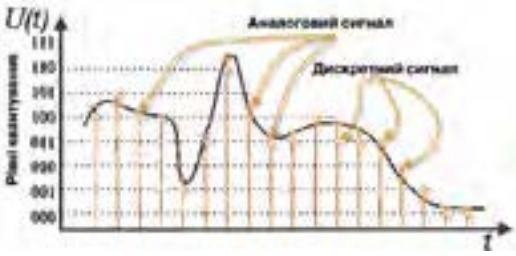


Рис. 2.15. Кодування сигналу

Точність кодування залежить від кількості двійкових розрядів, відведених для запису значень сигналу. У прикладі (див. рис. 2.15) фіксованих значень сигналу 7, тому для кодування обрано 3 двійкових розряди ($7<23$). Кількість двійкових розрядів і кількість рівнів сигналу перебувають у співвідношенні $K \le 2n$, де K – кількість вимірювань; n – кількість двійкових розрядів. Що більше використовується двійкових розрядів, то вища точність квантування. Найчастіше в сучасних комп'ютерах для цього використовується 16 двійкових розрядів (інколи 24 і 32 розряди).

✓ *Кількість двійкових розрядів, відведених для кодування звукових даних, називається* ***глибиною квантування.***

Людське вухо – дуже складний орган, який перетворює коливання повітря, тобто звук, на нервові сигнали. Так само діє й мікрофон, який перетворює звукові коливання на електричний сигнал – зміни електричного струму.

Людина, як і більшість тварин, має два вуха. **Навіщо?**

Поширюючись від джерела, звук змінюється, зменшується його гучність, інколи звуки від кількох різних джерел послаблюються по-різному. Два вуха допомагають людині визначити напрям на джерело звуку. Слухаючи "наживо" у залі великий оркестр, можемо, навіть заплющивши очі, розпізнати, як на сцені розташовані інструменти, як рухається по сцені соліст. **Для того** щоб **створити** у слухача **відчуття присутності** в залі, під час запису музики використовують щонайменше два мікрофони, сигнали від яких записуються окремо й відтворюються двома пристроями – навушниками, акустичними колонками. Такий запис звуку називають **стереозаписом.**

ДЛЯ ДОПИТЛИВИХ

Апаратна підтримка звуку

Збереження й опрацювання звуку в комп'ютері здійснюється за допомогою спеціального апаратного та програмного забезпечення. Апаратне забезпечення реалізоване у вигляді звукової плати (звукової карти, звукового адаптера), що встановлюється на материнській платі.

Основне призначення звукової плати – забезпечити введення в комп'ютер аналогового звукового сигналу, перетворити його на двійковий код, і навпаки – вивести аналоговий сигнал на звукові колонки. Звукова плата має щонайменше три роз'єми: вихід для звукових колонок (найчастіше роз'єм синього кольору), для мікрофона (рожевий) і лінійний вихід (зелений). Лінійний вихід використовується для підключення таких пристроїв, як магнітофон, програвач компакт-дисків. Значна кількість звукових плат має лінійний вхід (блакитний колір), що використовується для запису звуку в комп'ютер із зовнішнього пристрою, наприклад, музичного центру. Сучасні звукові плати мають також ігровий порт (жовтий колір) для підключення джойстика або MIDI-пристрою.

Значна кількість сучасних звукових плат мають роз'єми цифрового виходу (Digital Out) і роз'єми для підключення аудіосистеми. Структура звукової плати і зв'язки між її елементами зображені на рис. 2.16.

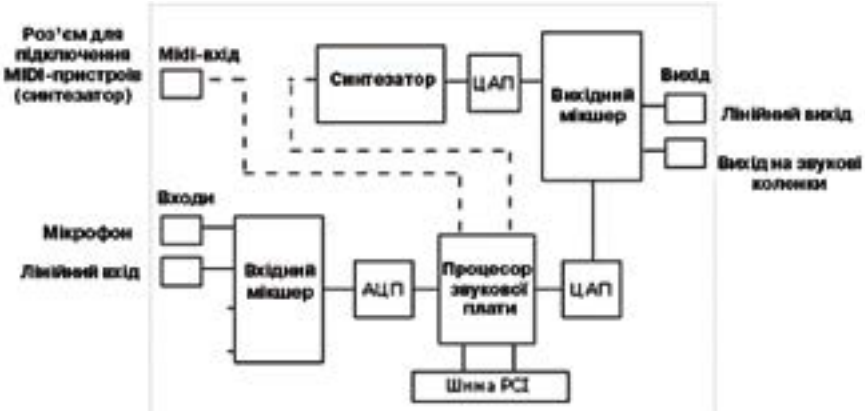


Рис. 2.16. Структура звукової плати

Сигнал із мікрофона або плеєра подається через вхідні роз'єми на пристрій (мікшер), призначений для змішування сигналів, що надходять на кілька входів. Із мікшера аналоговий сигнал подається на аналого-цифровий перетворювач (АЦП). Із його виходу двійковий код надходить у процесор звукової плати, який через шину PCI материнської плати здійснює обмін даними з комп'ютером. Керує процесом обміну звуковими даними центральний процесор.

Під час відтворення звукового файла дані з жорсткого диска через шину PCI надходять у процесор звукової плати і далі на цифро-аналоговий перетворювач (ЦАП), який перетворює двійковий код на аналоговий електричний сигнал. Далі цей сигнал із вихідного мікшера надходить на лінійний вихід, а також на вихід звукових колонок звукової плати (рис. 2.17).

Рис. 2.17. Звукова плата

Якщо в комп'ютері зберігається MIDI-файл, то його команди з вінчестера через шину PCI і процесор звукової плати надходять у синтезатор і далі в ЦАП, вихідний мікшер і на лінійний вихід.

До роз'ємів MIDI можуть підключатися MIDI-клавіатура, автономний синтезатор та інші електричні музичні інструменти.

2.6. Формати аудіофайлів

Із появою компакт-дисків був розроблений спеціальний формат запису звукових даних, який отримав назву Audio CD. Звукові дані записуються на спіральній доріжці окремими треками. Кожний трек містить один звуковий фрагмент, яким може бути, наприклад, одна пісня. Для відтворення звуку з компакт-дисків у комп'ютері використовуються спеціальні програми – програвачі. З появою оптичних дисків DVD був розроблений формат запису DVD-Audio.

Нині існує величезна кількість форматів для роботи зі звуковими файлами. Але не всі файли, що зберігаються на CD-дисках і відтворюються на комп'ютерах, можуть зберігатися на вінчестерах. Наприклад, файли типу Audio можуть відтворюватися на персональному комп'ютері (ПК), оснащеному пристроєм CD-ROM, однак для збереження на жорстких магнітних дисках без додаткового їх опрацювання вони не придатні.

Надалі під терміном "аудіофайли" розумітимемо два типи файлів: музичні (ті, що створюються за допомогою спеціальних синтезаторів) і звукові (моноаудіофайли і стереоаудіофайли).

Для збереження звукових файлів без їх стиснення потрібен значний обсяг пам'яті. Орієнтовно його можна обчислити за формулою: $V = (f \cdot t \cdot k)/8$, де V – обсяг пам'яті (в байтах); f – частота дискретизації (в одиницях за секунду); t – тривалість запису звуку (у секундах); k – глибина квантування (у бітах). Наприклад, для запису звуку протягом 5 хв (300 с) із частотою дискретизації 44 кГц (44000 1/с) і глибиною квантування 16 біт (2 байти) потрібно $V = 44000*16*300/8 = 26400000$ байт / (1024*1024) ≈ 25,2 МБ.

З метою зменшення обсягу звукових файлів здійснюється стиснення (компресія) звукових даних. У результаті стиснення даних досягається економія пам'яті до 12 разів. Однак надмірно високий коефіцієнт стиснення призводить до втрати якості звуку. Метод стиснення даних визначає **формат звукового файла**. Залежно від цього розрізняють два основних способи запису.

1. **Формати без стиснення даних**. До цієї групи належать формати файлів WAV (для ОС Windows) і AIFF (для ОС Mac), а також внутрішні формати звукових редакторів, наприклад, Audacity. Ця група форматів забезпечує високу якість звуку, але потребує великих обсягів пам'яті. Вони незручні для передавання даних через Інтернет.

2. **Формати зі стисненням даних**. Ця група форматів використовується найчастіше. Значення коефіцієнта стиснення файлів цього формату міститься в межах від 2 до 12. Стискаються насамперед ті компоненти звуку, що нечітко сприймаються вухом людини. Такими є звуки з частотою більше 10 кГц. Для таких частот глибина квантування може бути, наприклад, 4 біти, а для інших частот – 16 біт. У цьому випадку для всього діапазону частот глибина квантування в середньому може становити 8 біт. За рахунок цього досягається економія пам'яті. Популярними файлами цього формату є файли типів MP3 і WMA (Windows Media Audio).

Особливим типом є формат MIDI. Абревіатура MIDI (англ. Musical Instrument Digital Interface) в перекладі означає "цифровий інтерфейс музичних інструментів". Файли формату MIDI – певна сукупність команд для відтворення синтезованих звуків. Кожній команді MIDI-файла в синтезаторі звукової плати відповідає конкретна нота того чи іншого музичного інструмента, її тембр і гучність. Файли MIDI невеликі за обсягом і відтворюються за допомогою спеціальних програм – MIDI-плеєрів. Найчастіше використовуються програвачі Windows Media і Winamp.

Зазначмо, що один і той самий MIDI-файл може з різною якістю відтворюватися на різних ПК. Це пояснюється різною якістю звукових плат, які відрізняються відтінками синтезованих звуків – імітуванням певного музичного інструмента.

Існує велика кількість програм для роботи зі звуковими даними. За призначенням їх можна поділити на такі групи: програми для прослуховування (програвачі); програми для запису (захоплення) звуку (грабери); звукові редактори; конвертери.

Програвачі, що забезпечують прослуховування звукових даних, називають **аудіоплеєрами**. Для прослуховування звукових файлів в ОС Windows використовується **Медіяпрогравач Windows**.

Програми для запису звуку призначені для запису звуку з мікрофона, DVD-програвача, телевізійного тюнера та інших пристроїв. Звукові редактори не лише забезпечують запис і прослуховування звукових даних,

але й дають змогу виконувати операції із записами та їх фрагментами: виділення фрагментів, їх вставлення, заміну частин записів, змішування записів, а також змішування стереоканалів, фільтрування небажаних компонентів, застосування спеціальних ефектів, що збагачують звук, тощо.

Сучасні програмні засоби роботи зі звуком забезпечують роботу не з одним, а з кількома форматами звукових файлів.

Перевіряємо себе

1. Які хвилі називають гармонічними?
2. Якими параметрами характеризуються гармонічні коливання?
3. На що впливає значення амплітуди звукової хвилі?
4. Що називається періодом коливання?
5. Як зв'язані швидкість і довжина хвилі?
6. Що таке частота коливання?
7. Наведіть основні співвідношення між параметрами хвилі.
8. Назвіть основні етапи оцифровування звуку.
9. Поясніть сутність дискретизації аналогового сигналу.
10. Що таке частота дискретизації?
11. Поясніть сутність квантування амплітуди сигналу.
12. Що таке глибина квантування?
13. Назвіть основні типи форматів звукових файлів.
14. Яка програма використовується для роботи зі звуком в ОС Windows 7?

Виконуємо

1. Період коливань хвилі дорівнює 0,005 с, а її довжина – 1,5 м. Обчисліть швидкість хвилі.

2. Обчисліть кількість двійкових розрядів, необхідних для кодування 1000 рівнів квантування.

3.  Запустіть Медіяпрогравач Windows. Відкрийте меню **Довідка**, потім пункт **Довідка медіяпрогравача Windows**. Ознайомтеся з початком роботи з програвачем і закрийте вікно довідки. Прослухайте і перегляньте деякі записи, наприклад, **Записані телепрограми**.

4.  Обчисліть час звучання стереоаудіофайла, якщо частота дискретизації дорівнює 32 КГц, глибина кодування звуку – 16 біт, а інформаційний обсяг файла – 5 МБ.

5. Обчисліть інформаційний обсяг моноаудіофайла, якщо частота дискретизації дорівнює 60 КГц, глибина квантування – 16 біт, а тривалість звуку – 30 с.

6.  Глибина квантування для лінійного методу дорівнює 16 біт. Визначити крок квантування за рівнем, якщо максимальне значення напруги дорівнює 200 МБ, а мінімальне – 30 МБ. ★

Підказка. Величина кроку квантування за рівнем визначається за формулою: $h = (U_{max} - U_{min})/2^n$.

2.7. Кодування відеоданих

Відео – це послідовність нерухомих зображень – кадрів, швидка зміна яких створює ілюзію руху об'єкта.

Для людського зору така ілюзія виникає при зміні 16 і більше кадрів за секунду. У кінотеатрах використовується зміна 24 кадри/с, у телебаченні – від 25 кадрів/с. Що більша швидкість зміни кадрів, то якісніше зображення.

Кодек – це програма, що перетворює потік даних або сигналів на цифрові коди, або навпаки.

Звукові та візуальні дані потребують різних методів стиснення, а тому для них розроблені окремі кодеки. Через велику кількість різноманітних форматів (табл. 2.3) аудіо- та відеофайлів часто виникає потреба перекодовувати ці файли з одного формату в інший. Процес перекодування файла з одного формату в інший називається **конвертацією** файлів.

Є багато спеціалізованих програм для здійснення конвертації різноманітних мультимедійних даних. Однією з таких програм є Total Video Converter, який здійснює конвертацію аудіо- та відеофайлів більшості форматів. Кодек може складатися із двох компонентів: кодувальника та декодера.

***Кодувальник** виконує функцію стискання (кодування).*

***Декодер** виконує функцію розпакування (декодування). Деякі кодеки мають обидва ці компоненти, а деякі – лише один із них.*

***Медіяпрогравач** – пристрій або програмний засіб, за допомогою якого відтворюється відео-, фото- і аудіоконтент.*

Найпростішим у керуванні та найпоширенішим є медіяпрогравач Windows Media Player. Якщо для стиснення мультимедійних даних застосувати різні алгоритми, то ці дані будуть записані у файлах різних форматів. Оскільки рухоме зображення має супроводжуватися звуком, для зберігання відеофільмів розроблено спеціальні формати, які називаються медіяконтейнерами.

Медіяконтейнер – формат, що дає змогу розміщувати в одному файлі мультимедійні дані різних типів і синхронізувати звук; відеозображення й текстову інформацію.

Медіяконтейнери:

– WAV (Waveform Audio Format) – звук у форматі WAV зберігається без втрати якості, але відсутність стиснення призводить до того, що обсяги wav-файлів дуже великі;

– AVI (Audio and Video Interleaved) – надає можливість об'єднувати нестиснені або закодовані різноманітними кодеками аудіо- та відеодані;

– MOV (QuickTime Movie) – як і формат AVI, дає змогу поєднувати аудіо- та відеопотоки, закодовані в різний спосіб, розроблений для програвача QuickTime.

Утиліти-**конвертери мультимедія – це програми, що виконують перетворення** файлів, які належать до одного типу даних, але в різних форматах.

Таблиця 2.3

Популярні формати відео- та аудіофайлів (звукових файлів)

Формат файла	Розширення	Додаткові відомості
Відео Flash – Adobe Flash Media	.swf	Формат файла, який переважно використовується для передавання відео через Інтернет за допомогою програвача Adobe Flash
Advanced Streaming Format – медіяфайл Windows	.asf	Файл зберігає синхронізовані мультимедійні дані та може використовуватися для потокового відтворення через мережу аудіо- та відеовмісту, зображень і команд сценаріїв
Audio Video Interleave – відеофайл Windows	.avi	Мультимедійний формат файла, що використовується для збереження звуку та відеозображення в форматі Microsoft Resource Interchange File Format (RIFF). Один із найуживаніших форматів, оскільки звук і відеовміст, які стискаються за допомогою різноманітних кодеків, можуть зберігатись у файлі з розширенням .avi
Moving Picture Experts Group – файл фільму	.mpg або .mpeg	Стандарт відео- й аудіокомпресії, який розвивається, створено Moving Picture Experts Group. Цей формат файла створено спеціально для використання на мультимедійних пристроях, наприклад, на відеокомпакт-дисках і дисках CD
Windows Media Video – відеофайл Windows Media	.wmv	Формат зі стисненням звуку та відео за допомогою кодека Windows Media Video. Дуже стиснутий формат, який потребує мінімального обсягу дискового простору на жорсткому диску комп'ютера

Програми, що забезпечують перетворення одного формату файлів на інший, можна поділити на чотири умовні категорії:

– програми, призначені для конвертації аудіо та відео для різних пристроїв (програвачів MP3, мобільних телефонів, плеєрів, ігрових приставок тощо);

– програми, орієнтовані на користувацький режим конвертації;

– гібридні програми, у яких поєдналися характеристики обох згаданих вище категорій;

– вузькоспеціалізовані програми, зорієнтовані на виконанні конвертації аудіофайлів.

Перевіряємо себе

1. Поясніть, що таке відео.

2. Як називається програма, що перетворює потік даних чи сигналів на цифрові коди, або навпаки?

3. Який пристрій називають медіяпрогравачем?

4. Назвіть найпоширеніші аудіоформати.

5. Назвіть найпоширеніші відеоформати.

6. Поясніть призначення медіяконтейнера.

7. Який процес називається конвертацією файлів?

8. З чого складається кодек?

9. Як називаються програми, що виконують перетворення файлів?

10. Які програми називаються аудіокодеками?

Виконуємо

1. Знайдіть мультимедійні документи на вашому комп'ютері.

2. Поясніть, чим вони відрізняються від текстових і графічних документів.

3. Скопіюйте в окрему папку файли, які належать до **Аудіоформатів**.

4. Скопіюйте в окрему папку файли, які належать до **Відеоформатів**.

Практична робота № 2

Тема:	Розв'язування задач на визначення довжини двійкового коду даних різних типів
Мета:	Набути практичних навичок визначення довжини двійкового коду даних різних типів

1. У цифровому пристрої необхідно зберігати повідомлення ЛАБОРАТОРНА РОБОТА. Обчислити кількість бітів, необхідних для кодування символів цього повідомлення, і потрібний обсяг пам'яті для його збереження.

2. У цифровому пристрої з обсягом пам'яті 512 байт зберігається текстове повідомлення довжиною 1200 символів. Визначити максимально можливу кількість різних символів у цьому повідомленні.

3. Символами кодової таблиці ASCII передано повідомлення: МАТЧ ШАХТАР – ДИНАМО ЗАВЕРШИВСЯ З РАХУНКОМ 2:2. Обчислити обсяг пам'яті, необхідний для його збереження.

4. Монітор має повноекранний растр 1024*768 і глибину кольору 16 біт. Довести, що відеопам'ять обсягом 2 МБ достатня для відтворення графічних зображень.

5. Графічне зображення на екрані має 50*100 пікселів і займає 15 МБ. Визначити глибину кольору.

6. Визначити роздільну здатність екрана Вашого монітора в dpi.

7. Запустити програму Paint, відкрити графічний файл і вікно його властивостей. Обчислити теоретичний обсяг пам'яті, потрібний для файла, і порівняти його з реальним. Обґрунтуйте різницю.

8. Обчислити глибину кодування звуку, якщо інформаційний обсяг моноаудіофайла дорівнює 2,2 МБ, частота дискретизації – 44,1 кГц, тривалість звучання – 30 с.

9. Визначити інформаційний обсяг стереоаудіофайла, в якому зберігається звук тривалістю 45 с, глибина кодування звуку 16 біт і частота дискретизації 45 КГц.

ДЛЯ ДОПИТЛИВИХ

Розвиток систем кодування символів фактично **розпочався зі спроб використати електрику** для передавання повідомлень.

У 1833 році математик Фредерік Гаус запропонував метод кодування 25 символів (букви I та J були об'єднані) за допомогою матриці 5*5. Ідея полягала в такому: по одному дроту передавався електричний сигнал п'яти рівнів, від якого стрілка відхилялася на певне значення праворуч, а потім – електричний сигнал також п'яти рівнів, від якого стрілка відхилялася ліворуч.

Перше значення визначало номер рядка в таблиці, а друге – номер стовпця. Наприклад, якщо перше відхилення стрілки було зафіксоване на значенні 3, а друге – на значенні 2, то це означало, що була передана літера M (рис. 2.18).

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Рис. 2.18. Кодування символів за методом Гауса

У XIX столітті американський художник і винахідник Семюел Морзе запропонував систему кодування символів за допомогою коротких і довгих сигналів (крапка, тире). Літера А кодується коротким і довгим сигналом (• –), літера S – трьома короткими сигналами (• • •), цифра 1 коротким і чотирма довгими сигналами (• – – – –). Насправді код Морзе не є двійковим, оскільки для кодування використовується тривалість не тільки натиснення ключа (тире і точка), але й тривалість пауз між натисненнями. Тепер код Морзе використовують переважно радіоаматори.

Спосіб кодування для телеграфії запропонував у 1870 році французький інженер Еміль Бодо, який обмежив кількість сигналів у кодовій комбінації п'ятьма. Сигнал мав два стани: увімкнено–вимкнено, тобто кодування здійснювалося справді на основі двійкових сигналів. Це давало змогу мати 32 кодові комбінації (2^5=32), що недостатньо для 26 літер, 10 десяткових цифр і синтаксичних знаків. Тому Е. Бодо використав 26 комбінацій для літер і 6 комбінацій для керуючих символів. Телеграфні апарати на основі коду Бодо розвивалися і використовувалися понад 50 років.

СЛОВНИЧОК

Векторне зображення – зображення об'єкта за допомогою графічних примітивів.

Глибина квантування – кількість двійкових розрядів для кодування звукових даних.

Глибина кольору – кількість бітів, що застосовуються для кодування одного пікселя.

Графічний редактор – програма, за допомогою якої створюються, редагуються, зберігаються й переглядаються графічні об'єкти.

Двійкова система числення – система числення з основою два.

Двійковий код – подання повідомлення або даних у вигляді послідовності чисел у двійковій системі числення.

Декодування – процес перетворення даних із коду на форму, яку сприймає людина або може опрацювати певна програма.

Десяткова система числення – система числення з основою десять (прийнята для повсякденного застосування система запису чисел).

Дискретизація – процес вимірювання значень аналогового сигналу через однакові проміжки часу.

Звукова хвиля – хвиля у пружному середовищі в діапазоні частот від 16 до 20000 Гц.

Квантування сигналу – процес заміни реального значення сигналу найближчим фіксованим з набору можливих значень.

Код – подання повідомлення (даних) після кодування.

Кодек – програма, що здійснює кодування і декодування звукових, або відеофайлів, поданих у певному форматі (кількох форматах).

Кодова таблиця – таблиця, в якій кожному можливому значенню в одному поданні поставлено у відповідність це ж значення в іншому поданні.

Кодування – процес створення на основі повідомлення або даних послідовності (упорядкованої сукупності) символів (знаків або сигналів), який виконується за певним правилом (алгоритмом кодування).

Кодування (у обчислювальній техніці) – процес перетворення даних (повідомлення) на подання, доступне для опрацювання комп'ютером, зазвичай двійковий код.

Кодування сигналу – процес присвоєння кожному фіксованому значенню сигналу двійкового коду.

Медіяпрогравач – програма для прослуховування звукових файлів

Модель RGB – відображення кольору за допомогою трьох базових кольорів (червоного, зеленого та синього).

Растрове зображення – зображення об'єкта сукупністю точок.

Роздільна здатність екрана – кількість пікселів на одиницю довжини (або характерний розмір) екрана.

Частота дискретизації – кількість вимірювань амплітуди сигналу за одиницю часу.

Шістнадцяткова система числення – система числення з основою шістнадцять.

РОЗДІЛ 3. КОМП'ЮТЕР ЯК УНІВЕРСАЛЬНИЙ ПРИСТРІЙ ДЛЯ ОПРАЦЮВАННЯ ДАНИХ

Комп'ютер – пристрій для автоматизованого опрацювання даних, управління різним устаткуванням, приладами, виробничими процесами. Комп'ютер складається з апаратного і програмного забезпечення. Подання даних і команд у комп'ютері здійснюється у двійковому коді. Бістабільний пристрій – об'єкт, який може перебувати в одному з двох станів.

3.1. Архітектура комп'ютера

Основними і обов'язковими частинами кожного комп'ютера є запам'ятовуючий пристрій (пам'ять), процесор і пристрій управління. Пам'ять, у якій дані й команди можуть зберігатися без споживання електроенергії називається постійним запам'ятовуючим пристроєм. Пам'ять, у якій дані та команди можуть зберігатися за умови постійного живлення, називається тимчасовою пам'яттю, або оперативним запам'ятовуючим пристроєм. Програма – послідовність команд, яка описує процес опрацювання даних, частина з яких може міститись у самій програмі, а частина – надходити з пристроїв уведення, або зовнішніх запам'ятовуючих пристроїв.

Будова і алгоритм роботи ЕОМ. Процесор, його будова та призначення. Пам'ять комп'ютера, її види. Зовнішні та внутрішні запам'ятовуючі пристрої. Визначення властивостей комп'ютера.

Комп'ютер – пристрій, створений як універсальний інструмент опрацювання даних через послідовне виконання команд із певного набору. Комп'ютер є **програмно керованим автоматом**, тобто все, що він виконує, має бути наперед описане в програмі.

Загальний опис будови комп'ютера як складної системи, в якій поєднано **апаратну** (англ. *hardware*) і **програмну** (англ. *software*) складові, називають **архітектурою** комп'ютера.

Наприкінці 40-х років XX століття американський учений угорського походження Джон фон Нейман уперше в загальному вигляді описав функціонування та будову (архітектуру) універсальної обчислювальної системи (комп'ютера). Він запропонував правила побудови комп'ютерів, які є основою вже для чотирьох поколінь комп'ютерів.

Ці правила коротко можна викласти так:

– Програма і дані зберігаються в одній загальній пам'яті (внутрішньому запам'ятовуючому пристрої).

– Кожна комірка внутрішньої пам'яті позначається номером, який називається адресою.

– Вміст комірок пам'яті розрізняється (команди або дані) за способом використання, а не за способом подання в пам'яті.

– Команди програми виконуються послідовно, починаючи з першої команди. Для зміни цієї послідовності використовуються команди передавання управління.

Внутрішня пам'ять комп'ютера (внутрішній запам'ятовуючий пристрій) є єдиною і лінійною:

– "*єдина*" означає, що програма й дані зберігаються в одній пам'яті, одна й та сама комірка пам'яті для однієї задачі може зберігати команду, а для іншої – дані;

– "*лінійна*" означає, що всі комірки пам'яті послідовно пронумеровані, номер комірки є її адресою.

Апаратна частина кожного комп'ютера містить такі складові (рис. 3.1):

1) центральний процесор (ЦП), який виконує арифметичні операції та операції порівняння (логічні операції);

2) пристрій управління (ПУ), який забезпечує управління виконанням програм;

3) внутрішній запам'ятовуючий пристрій (ВЗП), призначений для збереження програм і даних у процесі виконання програми (внутрішня пам'ять);

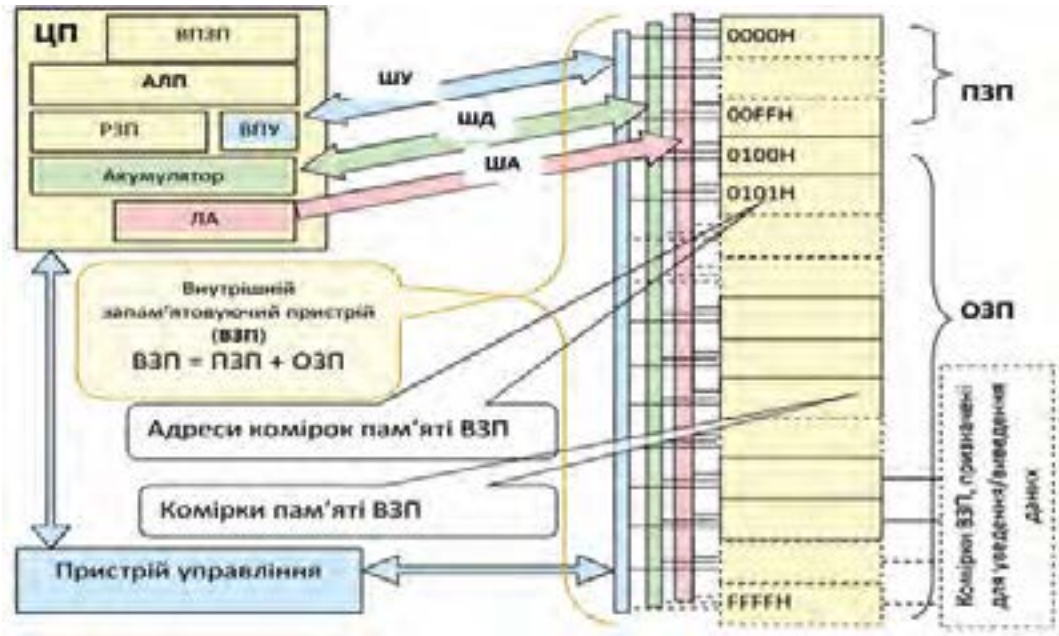


Рис. 3.1. Спрощена схема комп'ютера

4) пристрої для забезпечення уведення та виведення команд і даних (У/В).

Центральний процесор сучасного комп'ютера виконаний у вигляді великої мікросхеми, яка сама є маленьким комп'ютером.

Програмна частина, залежно від призначення комп'ютера, міститься в зовнішньому постійному запам'ятовуючому пристрої (ЗПЗП) або у постійному запам'ятовуючому пристрої (ПЗП), який є частиною внутрішньої пам'яті.

На рис. 3.1 позначено:

ЦП – центральний процесор (CPU – Central Processing Unit);

ОЗП – оперативний запам'ятовуючий пристрій – (RAM – Random Access Memory – пам'ять із довільним доступом);

ПЗП – постійний запам'ятовуючий пристрій (ROM – Read Only Memory – пам'ять тільки для читання);

Пристрої В/В – пристрої, призначені для введення/виведення даних (I/O – Input/Output – уведення/виведення).

ВЗП – внутрішній запам'ятовуючий пристрій (внутрішня пам'ять) комп'ютера (ВП) складається із постійного запам'ятовуючого пристрою (ПЗП) і оперативного запам'ятовуючого пристрою (ОЗП).

✓ *Внутрішній запам'ятовуючий пристрій з'єднаний з центральним процесором трьома системами провідників, які називаються* ***шинами.***

Шина даних (ШД) служить для передавання даних між центральним процесором і внутрішньою пам'яттю.

Шина управління (ШУ) слугує для передавання команд від процесора до внутрішньої пам'яті. Ці команди подаються тоді, коли процесор приймає дані з певної комірки пам'яті (кажуть: "зчитує дані"), або коли записує їх у пам'ять.

Шина адреси (ША) слугує для того, щоб виокремити певну комірку. На провідники ША подаються сигнали, які у двійковому поданні відповідають номеру комірки.

Така будова комп'ютера отримала назву **магістрально-модульна** архітектура. Завдяки застосуванню такої архітектури стало можливим побудувати дуже складні пристрої з більш простих "цеглинок" – модулів.

✓ ***Магістрально-модульна архітектура*** *– будова комп'ютера, за якої окремі частини комп'ютера (****модулі****) з'єднуються між собою провідниками, призначення кожного з яких визначене для всіх комп'ютерів певного типу.*

Комірки пам'яті сучасних запам'ятовуючих пристроїв складаються з бістабільних пристроїв, тобто пристроїв, які можуть перебувати в одному з двох станів. Ці стани відповідають нулю або одиниці двійкового подання числа. Стани m бістабільних пристроїв можуть бути записані

як m-розрядне двійкове число. Наприклад, якщо комірка пам'яті складається з 32 бістабільних пристроїв, які перебувають у таких станах 10101010101010101010101010101010, що відповідає АААААААА в шістнадцятковому поданні, це означає, що в ній записане число 2863311530 = $= 16^7 \cdot 10 + 16^6 \cdot 10 + 16^5 \cdot 10 + 16^4 \cdot 10 + 16^3 \cdot 10 + 16^2 \cdot 10 + 16^1 \cdot 10 + + 16^0 \cdot 10 =$ $= 2684354560 + 167772160 + 10485760 + 655360 + 40960 + 2560 + 160 +$ $+ 10$, якому може відповідати й певна команда.

Перші персональні комп'ютери мали пам'ять, кожна комірка якої складалась із восьми бістабільних пристроїв. Комірки пам'яті сучасних комп'ютерів складаються зі 64 бістабільних пристроїв.

✓ *Кількість бістабільних пристроїв у одній комірці пам'яті називається* ***розрядністю пам'яті.***

Центральний процесор складається з таких основних частин.

ВПЗП – внутрішній постійний запам'ятовуючий пристрій, пристрій пам'яті, в якому зберігаються програми, що описують алгоритми виконання ЦП окремих команд (мікропрограми).

АЛП – арифметично-логічний пристрій, у якому відбуваються операції порівняння двійкових чисел, арифметичні та інші дії над ними.

Акумулятор – спеціалізований ОЗП, в якому дані та команди зберігаються до і після виконання дій в АЛП.

РЗП – регістри загального призначення, ОЗП, в якому містяться дані під час їх опрацювання в АЛП.

ЛА – лічильник адреси – ОЗП, у якому зберігається число, що відповідає адресі комірки ВЗП, з якою в конкретний момент часу з'єднано ЦП.

ВПУ – внутрішній пристрій управління, який забезпечує керування частинами ЦП.

Зазвичай АЛП, пристрій управління та інші частини центрального процесора не розглядаються окремо, а об'єднуються під назвою "центральний процесор" (ЦП).

✓ ***Процесор*** *– пристрій для опрацювання даних, у якому виконуються всі дії, необхідні для виконання команд програми, яка міститься у ВЗП.*

✓ ***Мікропрограма*** *– послідовність команд і даних, яка міститься у ВПЗП процесора й описує виконання команди, що надходить у ЦП із ВЗП.*

✓ ***Команда*** *– вказівка, яку саме операцію і над якими даними потрібно виконати (наприклад: "додати два числа 2 і 3"). Команди, які виконує центральний процесор, кодуються числами, кожному з яких відповідає певна мікропрограма, розміщена у ВПЗП процесора. Ці числа (коди) називаються* ***машинними командами.***

Оскільки команди й дані розміщуються у ВП і розрізняються лише тоді, коли подаються шиною даних до ЦП, вміст комірок ВП прийнято називати ***кодом.***

Для того щоб бути виконаною центральним процесором, програма, тобто послідовність команд і даних, має бути розміщена в комірках внутрішньої пам'яті. Як бачимо з рис. 3.1, до складу внутрішньої пам'яті входять і постійні запам'ятовуючі пристрої, тобто такі, що не потребують живлення для збереження даних. У цих комірках зберігаються команди й дані, потрібні одразу після ввімкнення комп'ютера. Інші програми для виконання завантажуються до оперативного запам'ятовуючого пристрою внутрішнього запам'ятовуючого пристрою із зовнішніх запам'ятовуючих пристроїв.

Стан ЛА (число, яке міститься в ньому) відображається на ША і, після подання на ШУ відповідного сигналу, комірка з відповідною адресою з'єднується з ЦП через ШД.

Дуже спрощено алгоритм роботи програмно-керованого автомата (комп'ютера) фоннеймановського типу можна описати таким чином.

1. Лічильник адреси (ЛА) встановлюється у стан, який відповідає адресі першої команди програми, що має виконуватись, цей же код подається на шину адреси (ША).

2. На шину управління (ШУ) подається сигнал "читаю". З комірки ВЗП, адреса якої подана на ША, через шину даних (ШД) код (команда або дані) подається до ЦП.

3. У ЦП прийнятий код порівнюється з кодами, що містяться у внутрішньому постійному запам'ятовуючому пристрої (ВПЗП) ЦП, запускається на виконання відповідна мікропрограма. Якщо це не код зупинення, то ЛА встановлюється в стан, що відповідає адресі комірки ВЗП, в якій містяться необхідні дані, інакше – виконання програми зупиняється.

4. На шину управління (ШУ) подається сигнал "читаю", дані через ШД передається у ЦП й опрацьовуються.

5. ЛА встановлюється в стан, що відповідає адресі комірки ВЗП, у яку має бути записано результат., На шину управління (ШУ) подається сигнал "записую", дані розміщуються у комірку пам'яті.

6. ЛА встановлюється в стан, що відповідає адресі комірки ВЗП, у якій розміщено код наступної команди, цей же код подається на ША.

7. Здійснюється перехід до кроку 2.

Спільне використання апаратного обладнання і програмного забезпечення вимагає застосування певної системи управління програмними й апаратними засобами. Слід мати на увазі, що принтер, процесор та інші пристрої не можуть одночасно виконувати більш ніж одну команду.

Одночасне звернення кількох програм, або кількох команд, що належать одній програмі, до одного пристрою, називають **конфліктом.**

Тому обчислювальна система розглядається як сукупність пристроїв, кожен із яких володіє певним **ресурсом**.

Ресурсами вважають:

– *час процесора*, який розподіляється між програмами, якщо виконується більш ніж одна;

– *оперативну пам'ять*, яку керуюча система виділяє кожній програмі у вигляді обмеженої початковою і кінцевою адресами ділянки внутрішнього запам'ятовуючого пристрою;

– *час периферійних пристроїв*, який керуюча система також розподіляє так, щоб уникнути конфліктів доступу;

– *файли*, зовнішню пам'ять, тобто дані, які зберігаються в зовнішній пам'яті.

Використанням ресурсів керує сукупність програм, яка називається ***операційною системою***. Для універсальних комп'ютерів обов'язкова наявність операційної системи, оскільки її складові забезпечують виконання операцій із завантаження до ОЗП програм і даних, роботу зі зовнішніми пристроями пам'яті, управління апаратними складниками комп'ютера.

Без операційної системи програмно-керований автомат зможе виконувати лише одну програму. Такі програмно-керовані автомати використовують, наприклад, для опрацювання даних у простому телефонному апараті, телевізорі, мікрохвилівці.

Одним із основних способів забезпечення злагодженої роботи програм і зовнішніх пристроїв, уникнення конфліктів є використання ***переривань***, які викликаються ***подіями***.

Подія *– поява сигналу (повідомлення) від зовнішнього пристрою (рух миші, натиснення кнопки на миші або клавіші на клавіатурі, вставлення флеш-картки у відповідний роз'єм тощо), або отримання в процесі виконання програми певного значення.*

Переривання *– спеціальний сигнал, або команда, що подається для перемикання процесора на опрацювання події (переривання) за певною програмою, на виконання якої переходить процесор.*

Опрацювання події (часто кажуть – опрацювання переривання) означає, що деякі дані, які виникли внаслідок обчислень або отримані від апаратного забезпечення, мають бути опрацьовані задля подальшого виконання необхідних для отримання результату дій. Наприклад, після вставлення флеш-карти в роз'єм спостерігаємо на екрані відповідне повідомлення, рух миші по поверхні килимка супроводжується переміщенням курсора по екрану тощо.

Властивості комп'ютера, зокрема його швидкодія, залежать від властивостей його модулів. Ці властивості – розрядність шини адреси, розрядність шини даних (процесора), розрядність комірок внутрішньої пам'яті, частоти роботи центрального процесора і внутрішньої пам'яті – можна

отримати як з технічної документації, так і засобами операційних систем (ОС). Для ОС Windows 7 це можна зробити так, як показано на рис. 3.2.

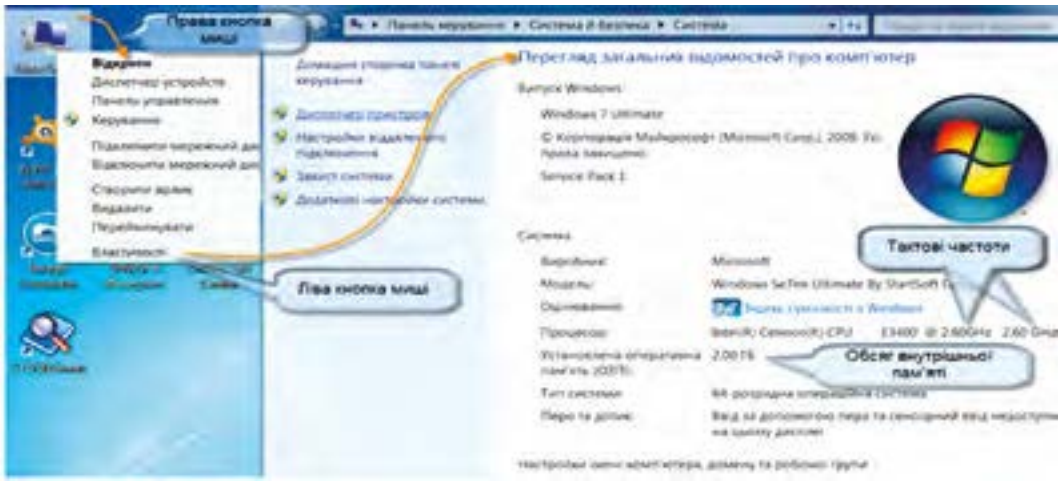


Рис. 3.2. Визначення властивостей апаратної частини комп'ютера

Знаючи розрядність адресної шини і розрядність внутрішньої пам'яті, обчислюємо максимально можливий обсяг внутрішньої пам'яті. Розрядність адресної шини визначає максимально можливу кількість комірок пам'яті, а розрядність комірок пам'яті – її загальний обсяг.

Якщо ША 32 розрядна, то для адресування пам'яті використовується 2^{32} біт, тобто кількість комірок не може перевищувати 4 294 967 296. Кожна комірка має ємність 8 біт = 1 байт. 4294967296 комірок мають ємність 4 ГБ.

Шинно-модульна архітектура комп'ютера дає можливість виробникам апаратних засобів розроблювати окремі частини **(модулі)** – пристрої пам'яті, процесори, пристрої управління тощо, не маючи всіх технічних характеристик інших частин комп'ютера, а лише знаючи, які сигнали приходитимуть на пристрій із шини і які він має подавати на шину. Таким чином досягається взаємозамінність пристроїв, що виробляються на різних заводах.

Для користувача шинно-модульна архітектура спрощує ремонт – досить визначити, який модуль комп'ютера несправний, і замінити його.

Перевіряємо себе

1. Які переваги має шинно-модульна архітектура комп'ютера?
2. Що таке ША, ШД, ШУ?
3. Оцініть кількість можливих команд процесора й відповідно мікропрограм процесора за 8-розрядного і 16-розрядного їх кодування.
4. Наведіть приклади подій, які викликають переривання виконання програми комп'ютером.

5. Опишіть, що відбувається після натиснення клавіші PrintScreen на клавіатурі.

6. Від чого залежить швидкість опрацювання даних комп'ютером? Порівняйте дані, отримані для кількох комп'ютерів.

7. Визначте властивості апаратного забезпечення комп'ютера робочого місця учня. Порівняйте з властивостями комп'ютера, показаними на рис. 3.2.

8. Перші процесори мали 8-розрядну шину даних і 16-розрядну шину адреси. З яким обсягом пам'яті вони могли працювати?

9. Мова програмування Макроасемблер, яку використовували для перших мікропроцесорів, забезпечувала пряме адресування тільки 64 КБ 8-розрядної пам'яті. Якою була розрядність ША цих процесорів?

10. Сучасні операційні системи забезпечують або 32-розрядне, або 64-розрядне адресування команд і даних. Під управлінням якої ОС комп'ютер працюватиме швидше? Чому?

11. Для яких операційних систем доцільно використовувати обсяги ОЗП більше 4 ГБ? Чому?

12. Уміст якої комірки ВЗП передається до ЦП?

13. Усередині системного блока комп'ютера розміщено пристрій пам'яті, який називається "вінчестер". Чи можна назвати вінчестер внутрішнім запам'ятовуючим пристроєм? Поясніть чому.

ДЛЯ ДОПИТЛИВИХ

Готова до виконання програма (програма у машинних кодах) розміщується на ЗЗП, звідки завантажується до ВЗП.

Рис. 3.3. Програма, готова до завантаження до ВЗП комп'ютера

Існують спеціальні програми, за допомогою яких можна переглядати вміст внутрішньої пам'яті комп'ютера, спостерігати за процесами виконання програм, "розшифровувати" програми, подані в машинних кодах. На рис. 3.3 подано результат роботи однієї з таких програм.

3.2. Мультимедійні пристрої введення та виведення даних

Комп'ютером управляють через пристрої введення/виведення, які забезпечують відображення повідомлень від комп'ютера у зрозумілій користувачеві формі (текстовій, графічній, звуковій) і введення до комп'ютера даних і команд. Під час виконання програми процесор періодично опитує пристрої введення–виведення. Виконання програми можуть переривати сигнали пристроїв керування, які генеруються як реакція на події – запити зовнішніх пристроїв на обслуговування, команди користувача тощо.

Пристрої, що входять до складу мультимедійного обладнання. Технічні характеристики мультимедійного обладнання.

Мультимедія, з одного боку, – особливий тип цифрових (електронних) документів, а з іншого – особливий клас програмного і апаратного забезпечення.

Щоб правильно використовувати пристрої введення–виведення, потрібно знати одиниці фізичних величин, які застосовують для порівняння якості різних засобів відтворення зображень і звуку.

Загальною назвою засобу, на якому відтворюється зображення, є назва "екран".

Одним із основних методів утворення зображення, яке може спостерігати велика група людей, є світлова проекція.

Світлова проекція *– одержання на екрані зображення деякого об'єкта з використанням потужного джерела світла й оптичної системи.*

Засоби світлової проекції характеризуються такими основними показниками якості: світловий потік і яскравість.

Світловий потік *характеризує потужність світлового випромінювання проектора і вимірюється в люменах (лм).*

Знаючи величину отримуваного від проектора світлового потоку, можна визначити, чи можна використовувати засіб відтворення зображення без затемнення у певному приміщенні.

✔ *Зображення, незалежно від способу його відтворення, характеризується такими основними показниками:* ***яскравість, контрастність, чіткість, розміри.***

Яскравість характеризує зображення, отримане на екрані. Для проекційних засобів вона залежить від світлового потоку джерела освітлення й якості поверхні екрана. Яскравість вимірюється в канделах на квадратний метр (кд/м^2). Орієнтовно можна вважати, що яскравість зображення понад 120 кд/м^2 достатня для спостереження за умов нормальної освітленості.

Яскравість зображення, створюваного сучасними проекційними засобами зі світловим потоком 1000...1600 лм на сучасних екранах із діагоналлю 100...170 см, становить 120...180 кд/м^2. Яскравість зображення, створюваного сучасними рідинно-кристалічними моніторами з діагоналлю екрана 19" (46 см), становить близько 300 кд/м^2.

Контрастність зображення визначається відношенням яскравості найбільш світлих ділянок зображення на екрані до яскравості найбільш темних ділянок. Досить якісне за контрастністю зображення характеризується значенням 100:1. Реальні об'єкти мають контрастність від 10:1 до 10000:1. Сучасні проекційні засоби відтворення зображення забезпечують контрастність до 2000:1 і більше.

Роздільна здатність зображення є його об'єктивною характеристикою, яка визначається кількістю точкових елементів (пікселів, англ. *picture cell*, комірка зображення), що утворюють це зображення. Найменшою роздільною здатністю засобу відтворення зображення, за якої зображення розмірами приблизно 16*24 см може вважатись досить якісним для спостереження на віддалі найкращого зору (20...30 см), є така, що забезпечується кількістю елементів зображення 640*480 = 307200. Більш якісне зображення створюється за роздільної здатності 800*600 = = 480000 елементів. Більшість сучасних цифрових засобів відтворення зображення забезпечує роздільну здатність 1024×768 = 786432 елементів та 1152×864, 1280×720, 1280×768, 1280×1024 і більшу.

Відомості про яскравість і колір кожного елемента (пікселя) зображення зберігаються в одному або кількох бістабільних пристроях пам'яті. Кількість таких пристроїв, виділених для зберігання даних про один піксель, називається **глибиною кольору**. **Глибина кольору** вимірюється в бітах (Б). Для цифрових засобів відтворення зображення роздільна здатність і глибина кольору визначаються обсягом пам'яті, призначеної для зберігання зображення.

Отже, для того щоб оцінити розмір файла, в якому зберігається зображення без стиснення, у байтах, потрібно: перемножити розміри зображення у пікселях (визначити кількість пікселів у зображенні), отримане число помножити на глибину кольору і поділити на вісім.

Наприклад, зображення розміром 640*480 пікселів і глибиною кольору 8 біт, матиме розмір 640*480*8/8 = 307200(Б) = 307200/1024 = = 300 (кБ).

Зображення розміром 1280*1024 пікселів, з глибиною кольору 24б зберігатиметься у файлі з розширенням *.bmp розміром 1280*1024*24/1024/1024 = = 3,75 (МБ).

Чіткість зображення визначається можливістю відтворення дрібних деталей зображення на екрані.

Зображення на папері, світлині, вважається досить чітким, якщо можна відрізнити 50 ліній на 1 мм зображення. Така чіткість вимагається від зображень, які людина з нормальним зором спостерігає з віддалі найкращого зору. Для зображень, призначених для спостереження на іншій віддалі, враховуються кутові розміри найменшого елемента зображення, тобто кут при вершині рівнобедреного трикутника, основою якого є елемент зображення, а у вершині розташоване око спостерігача.

Око людини не розрізняє кольорів дрібних деталей, тому чіткість зображення зазвичай визначається для одноколірного (монохромного) зображення. Чіткість зображення, таким чином, є частково суб'єктивною його характеристикою.

Сучасним пристроєм уведення, який використовується в багатьох пристроях, є **сенсорний екран.**

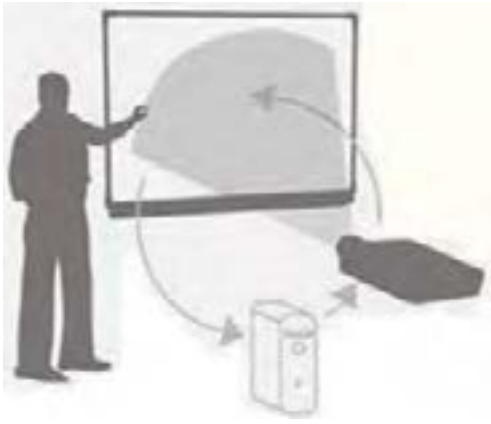


Рис. 3.4. Комплекс апаратних засобів для відтворення електронних презентацій (мультимедійна презентаційна система)

Сенсорний екран – поверхня, на якій розташовані чутливі елементи, дотик до яких пальцем або спеціальним пристроєм перетворюється на сигнал, що передається в комп'ютер.

Одним із перших застосувань подібних засобів уведення команд у комп'ютер була система "електронне перо", яка використовувалась на вітчизняних ЕОМ "МІР" (Машина Інженерних Розрахунків), що серійно випускались в Україні в 60–70-х роках минулого сторіччя.

За технічними засобами, які поєднують проектор, екран, систему відтворення звуку та систему управління комп'ютером безпосередньо з екрана, закріпилась назва "мультимедійна система" (рис. 3.4).

Перевіряємо себе

1. Які апаратні засоби і пристосування потрібні для відтворення зображення методом світлової проекції?
2. Чому для екранів використовують матеріали з найбільшим можливим коефіцієнтом відбивання світла?

3. Які фізичні величини характеризують зображення?

4. Коли доцільно використовувати управління демонстрацією з клавіатури, а коли – з сенсорного екрана? Наведіть приклади.

5. Обчисліть розмір файла, в якому розміщено (без стиснення) копію зображення, створеного на екрані розміром 800*600 пікселів із роздільною здатністю 800*600 пікселів і глибиною кольору 24 біти.

6. Яким чином обчислити розмір файла, в якому розміщено зображення, яке займає прямокутну область певного розміру на екрані, якщо задано роздільну здатність і розміри (у пікселях) зображення?

7. У яких сучасних пристроях використовуються сенсорні поверхні? Наведіть приклади.

8. У якому випадку дисплей може бути тільки пристроєм виведення, а в якому забезпечуватиме й уведення команд?

3.3. Програмне забезпечення

Програмне забезпечення сучасного комп'ютера складається з програм, які належать до операційної системи, програм, за допомогою яких створюються нові програми, і програм, які забезпечують виконання завдань користувача.

Класифікація, основні функції та склад операційних систем. Поняття про ядро операційної системи, інтерфейс користувача, драйвери та утиліти. Поняття сумісності програмного забезпечення.

Важливою частиною обчислювальної системи є програмне забезпечення. Між апаратними засобами і більшістю користувачів є три рівні програмних засобів (рис. 3.5).

Рис. 3.5. Загальна структура програмного забезпечення обчислювальної системи

Системне програмне забезпечення містить дві основні частини: операційну систему (ОС) й утиліти.

Операційна система забезпечує такі головні функції:

– здійснює керування усіма пристроями комп'ютера та виконанням програм;

– забезпечує обмін даними й командами користувача з комп'ютером.

Нині найпоширенішими операційними системами для персональних комп'ютерів є ОС Windows (версії XP, 7 і 8), досить поширені Mac OS, Linux. Для портативних пристроїв (смартфонів, планшетних комп'ютерів, електронних книг, цифрових програвачів, наручних годинників, ігрових приставок, нетбуків, смартбуків) використовуються ОС Андроїд (створена на основі Linux), Windows CE і iPhone OS.

Операційні системи поділяються на **однозадачні** та **багатозадачні**.

Однозадачні операційні системи використовуються тоді, коли комп'ютер призначений для одночасного виконання тільки однієї задачі – керування деяким пристроєм (двигуном, автоматичною лінією на виробництві, атомним реактором), в ігрових приставках.

Багатозадачні ОС використовуються здебільшого у випадках, коли виникає необхідність паралельного виконання кількох задач – створювати текстовий документ і шукати для нього відомості в Інтернеті, виконувати обчислення в електронних таблицях. Багатозадачні ОС застосовуються й у багатофункціональних пристроях – телевізорах, медіяцентрах тощо.

*ОС Windows, Mac OS, Linux є **багатозадачними**.*

Більшість ОС призначені для завантаження із ЗЗП комп'ютера, в який вони завантажуються, й управління ним. Набувають популярності ОС із віддаленим завантаженням і так звані хмарні ОС, наприклад, GoogleChrome OS, iCloud. Особливостями цих ОС є забезпечення доступу користувача до ресурсів незалежно від його місця перебування.

*Значна частина ОС передбачає можливість авторизації користувача. Такі ОС називаються **багатокористувацькими**.*

Створення для користувача власного **реєстраційного запису** (англ. *account* – рахунок) передбачає зберігання для нього налаштувань програмних і апаратних засобів, створення простору для зберігання даних, встановлення певних прав на керування обчислювальною системою. Авторизація, тобто вхід до системи, передбачає зазвичай два кроки – введення імені користувача (login) і пароля (password). Процедура авторизації обов'язкова для роботи в усіх системах з великою кількістю користувачів – поштових сервісах, хмарних сервісах тощо.

*Кожна ОС складається з **ядра**, програмної частини, що забезпечує **інтерфейс користувача**, і програм, які забезпечують роботу в мережі та обслуговування пристроїв.*

Ядро операційної системи забезпечує розподіл і управління ресурсами обчислювальної системи.

Ядро системи – це набір даних і окремих програм, які завантажуються в пам'ять комп'ютера при завантаженні операційної системи та забезпечують три типи системних сервісів:

– управління введенням – виведенням даних (підсистема введення – виведення ядра ОС);

– управління оперативною пам'яттю (підсистема управління оперативною пам'яттю ядра ОС);

– управління процесами (підсистема управління процесами ядра ОС).

Багатозадачні ОС включають ще одну обов'язкову складову – механізм підтримки багатозадачності, або ***планувальник процесів.***

Планувальник ОС в процесі завантаження коду програми до ВЗП розподіляє його на ***процеси***, тобто частини програми, які допускають самостійне виконання.

Існує три основних способи забезпечення багатозадачності (планування процесів):

– надання процесора окремій задачі (процесу) на певний час (квант часу), який визначається самою задачею;

– надання процесора окремій задачі (процесу) на квант часу, який визначається обладнанням обчислювальної системи – інтервальним таймером;

– виділення під окрему задачу окремого процесора в багатопроцесорних системах.

Після початку виконання програм планувальник процесів послідовно перемикає процесор на виконання команд процесів, які можуть належати різним програмам. Таким чином досягається ефект одночасного виконання кількох програм, хоча в кожний момент часу процесор виконує тільки одну команду.

Кожному зі створених процесів ОС надає ідентифікатор і пріоритет, завдяки чому здійснюється керування виконанням процесів. Найвищий пріоритет надається процесам, невиконання яких може спричинити помилки або втрату керування користувачем. До таких процесів належать процеси ядра системи, процеси, які виконують опрацювання переривань пристроїв уведення. Найнижчий пріоритет надається процесам, які забезпечують виконання програм користувача.

На рис. 3.6 показано одночасне виконання комп'ютером кількох програм: процесів ОС Windows XP, текстового редактора Microsoft Word, антивірусного програмного засобу Avast! та інших.

Графічний інтерфейс користувача може бути частиною ОС (як у ОС Windows) або забезпечуватись окремим програмним засобом, який входить до комплекту ОС (як у ОС Linux).

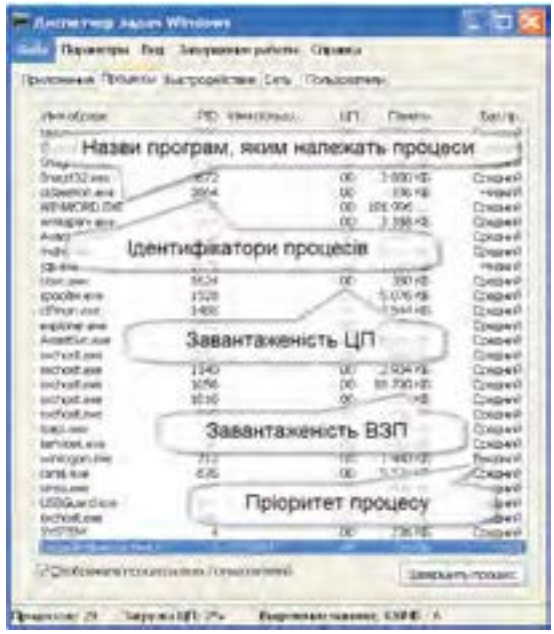


Рис. 3.6. Виконання програм обчислювальною системою під управлінням ОС Windows XP

✓ *У найпростішому випадку ОС має інтерфейс користувача, який працює в алфавітно-цифровому режимі, так званому режимі командного рядка.*

Рис. 3.7. Графічний інтерфейс ОС Windows 7

Такий режим має більшість ОС. Він використовується для налагодження системи. Усі сучасні ОС мають графічний інтерфейс користувача, який відображає команди керування обчислювальною системою у вигляді меню, піктограм, лінійок, кнопок тощо. Для роботи з графічним інтерфейсом використовують засоби введення: клавіатуру, мишу, джойстик, сенсорний екран. На рис. 3.7 показано деякі дії, що можуть виконуватись з використанням графічного інтерфейсу користувача ОС Windows 7.

Сучасні ОС надають користувачеві можливість подавати голосові команди, які сприймаються мікрофоном.

Режим голосового керування (голосовий інтерфейс або акустичний) використовують люди з вадами зору. Голосове керування використовують і тоді, коли користувач керує транспортним засобом (системи free hands – вільні руки).

Для ОС Windows 8 розроблено графічний інтерфейс користувача, подібний до інтерфейсу мобільних пристроїв, що працюють під управлінням ОС Android. Особливістю інтерфейсу стала його пристосованість до роботи з сенсорним екраном (рис. 3.8). Кожна кнопка, крім кнопки "Робочий стіл", викликає певний програмний засіб або ресурс. Деякі з них можуть бути хмарними, тобто розташовуватись не на комп'ютері користувача, а на інших комп'ютерах мережі Інтернет.

Рис. 3.8. Графічний інтерфейс ОС Windows 8

Для поліпшення роботи ОС до її складу можуть бути додані програми, призначені для виконання певних дій з обслуговування пристроїв комп'ютера. Ці програми називають *утилітами*.

***Утиліти** – окремі програми, які підвищують ефективність використання комп'ютера.*

Утилітами є програми для перевірки працездатності комп'ютера, програми очищення жорсткого магнітного диска, програми для роботи з архівами та інші. Утиліти можуть бути розроблені постачальником ОС і постачатись разом з нею або іншими постачальниками програмного забезпечення (навіть самим користувачем).

Для того щоб ОС могла надавати програмам користувача доступ до ресурсів апаратних засобів, у тому числі пристроїв уведення–виведення, мережевих пристроїв, вона має мати в своєму складі відповідні програми. Очевидно, що на етапі створення ОС неможливо створити такі програми для всіх можливих зовнішніх пристроїв, оскільки вони створюються різними виробниками, постійно вдосконалюються. Для ефективної роботи ОС необхідно для кожної материнської плати додавати до ОС певне програмне забезпечення. Тому виробники апаратного забезпечення розробляють і спеціальні програми, що постачаються разом із пристроями.

✓ *Програми, призначені для обслуговування певних пристроїв, називаються* ***драйверами****.*

Ці програми надаються на компакт-диску або розташовуються на сайті виробника. Більшість сучасних апаратних засобів мають властивість ідентифікування засобами ОС, тобто є пристроями типу plug and play, але це не означає, що драйвери, вбудовані в ОС, забезпечуватимуть їх повноцінне функціонування, тобто виконання всіх описаних у документації функцій. Навіть для таких пристроїв здебільшого потрібне встановлення драйверів.

✓ *Пристроями типу plug and play (підключи і грай) називають пристрої, які готові для використання після підключення до комп'ютера.*

Разом із драйверами на компакт-диску зазвичай розміщено й утиліти, які забезпечують більш ефективне, ніж із використанням засобів ОС, керування пристроєм. Наприклад, на диску 1 як утиліта розташоване програмне забезпечення, призначене для досить ефективного опрацювання растрових зображень, їх друкування (рис. 3.9).

Інколи на компакт-диску розміщують ОС, разом із усіма необхідними програмами та відповідними драйверами (диск 4 на рис. 3.9).

Перевіряємо себе

1. Чи можна створити комп'ютер, який би не потребував ОС?
2. Які дії виконує ядро ОС для запуску на виконання програми користувача?
3. Які програмні засоби використовуються для розроблення програм користувача?
4. Скільки може існувати варіантів подання команди копіювання файла в ОС Windows 7?

5. Чим відрізняється вміст дисків 3 і 4 (рис. 3.9) від вмісту інших?

Рис. 3.9. Компакт-диски з програмним забезпеченням, призначеним для обслуговування зовнішніх пристроїв, та іншим програмним забезпеченням

6. Опишіть призначення і орієнтовний вміст кожного з компакт-дисків, зображених на рис. 3.9. Можна використати результати пошуку необхідних даних у мережі Інтернет. Обговоріть у класі результати ваших пошуків.

7. Викличте (командою Ctrl+Alt+Del) Диспетчер задач Windows. Який із процесів виконується завжди? Поясніть чому.

8. Викличте (командою Ctrl+Alt+Del) Диспетчер задач Windows. Запишіть кількість процесів, які виконуються. Запустіть на виконання кілька програмних засобів. Поясніть, чому збільшилась кількість виконуваних процесів.

9. Визначте, які процеси належать запущеним вами на виконання програмам.

10. Запустіть на виконання програму EVEREST або подібну, визначте значення основних властивостей комп'ютера. Порівняйте отримані дані для кількох комп'ютерів. Отримайте відомості щодо апаратного і програмного забезпечення комп'ютера засобами операційної системи. Порівняйте з отриманими від програми EVEREST. Зробіть висновок. Коли і навіщо такі дії доцільно виконувати?

3.4. Ліцензії на програмне забезпечення

Програмні засоби створюються людьми, вони є продуктами їхньої суспільно корисної праці. Їх використання регламентоване законодавством.

Поняття про авторське право на програмні засоби. Правила використання програмного забезпечення. Типи ліцензій на програмні засоби.

Програмне забезпечення, як і будь-який результат продуктивної діяльності людини, може надавати користувачеві виробник за певні кошти або безкоштовно. Виробник програмного забезпечення має право визначати правила його використання, а користувач має їх дотримуватись.

На законодавчому рівні в усіх країнах запроваджена відповідальність користувачів за недотримання зазначених правил. В Україні діє закон "Про авторське право і суміжні права" – вітчизняний закон, який охороняє: авторське право – особисті немайнові та майнові права авторів і їхніх правонаступників, пов'язані зі створенням та використанням творів науки, літератури і мистецтва; суміжні права – права виконавців, виробників фонограм і відеограм та організацій мовлення.

✓ *Документ, у якому автор, розробник або виробник виклали правила використання програмного засобу, називається* ***ліцензійною угодою****, або* ***ліцензією****.*

Користувач, встановлюючи на комп'ютер певний програмний засіб або їх комплект (у тому числі й операційну систему), укладає з його виробником угоду щодо дотримання правил використання цього програмного засобу. Це відбувається тоді, коли він, прочитавши ліцензійну угоду, погоджується з викладеними в ній правилами, натискаючи у відповідній екранній формі кнопку або встановлюючи прапорець.

На авторство програмного засобу насамперед вказує знак "копірайт" ©, біля якого подано ідентифікатор фізичної або юридичної особи (прізвище людини або назва фірми).

У світі використовуються такі основні типи ліцензійних угод.

Commercial (Комерційна) – Закрите, Власницьке або **Пропрієтарне** програмне забезпечення. Користувач має право використовувати одну або декілька придбаних ним версій програмного забезпечення без права передавати її третім особам, продавати, вносити зміни в програмний код. Це найбільш "жорстка" ліцензія, яка передбачає безумовну цивільну відповідальність за її порушення.

Trial Software – ліцензія на програмне забезпечення, яке надається для ознайомлення, працює лише певний час, після завершення якого потрібно його придбати, або має певні функціональні обмеження. Зазвичай така ліцензія надається модифікованим версіям програмних засобів, які поширюються як пропрієтарні.

Non-Commercial Use – для некомерційного використання. Користувач може використовувати програмне забезпечення для навчання, виконання інших видів діяльності, не пов'язаних з отриманням прибутку (зокрема, не дозволяється створення з його використанням програмних засобів, які згодом продають іншим особам).

Shareware – програмне забезпечення, придбання якого не є обов'язковим для використання, у додаткових умовах можуть бути вказані можливості добровільної сплати деяких коштів у формі благочинних внесків.

Royalty-free binaries, "**Freeware**" – безкоштовне програмне забезпечення, подане у вигляді двійкового коду, для використання as is ("як воно є").

Open Source, BSD-Style – відкрите програмне забезпечення за ліцензією, запропонованою розробниками BSD (Університет Берклі, США, штат Каліфорнія). Найбільш "вільна" ліцензія, за якою дозволено створювати будь-яку кількість копій, поширювати їх як у вигляді двійкового коду, так і у вигляді програм мовами високого рівня.

Open Source, Apache Style – відкрите програмне забезпечення за ліцензією, запропонованою розробниками фірми Apache. Ліцензія надає користувачеві право використовувати програмне забезпечення для будь-яких цілей, вільно поширювати, змінювати й поширювати змінені копії. Обов'язковою умовою є інформування одержувача про факт використання початкового коду, ліцензованого під ліцензією Apache.

Open Source, Linux/GNU style – відкрите програмне забезпечення за ліцензією Linux і GNU. Ліцензія **GNU General Public License** (Загальна публічна ліцензія GNU) – одна з найпопулярніших ліцензій на вільне програмне забезпечення, створена для проекту GNU. Часто її скорочено називають **GNU GPL**. Ця ліцензія надає користувачеві право копіювати, змінювати та розповсюджувати програми. Ліцензія зобов'язує передавати це право користувачам усіх похідних програм. Принцип "спадковості" таких прав називають "копілефт" (транслітерація англійського *copyleft*).

Перевіряємо себе

1. Під якою ліцензією розповсюджується ОС Windows?
2. Під якою ліцензією розповсюджується ОС Linux?
3. Хто вважається автором програмного продукту?

4. Чи існують ліцензії, які дозволяють декомпозицію програмного коду?

5. Дослідіть програмне забезпечення, встановлене на вашому комп'ютері. Які види ліцензій використано і для яких програм?

6. У яких випадках і яке програмне забезпечення можна передавати в користування третім особам? Можна використати результати пошуку необхідних даних у мережі Інтернет.

7. Чому разом із програмним забезпеченням інколи передається його вихідний код?

8. У яких випадках необхідно звертатись до виробника програмного засобу? Поясніть чому.

9. Яка різниця між тріал-версією програмного продукту і програмним забезпеченням вільного поширення?

10. Яка версія ліцензійної угоди вважається юридично чинною? Чому?

11. Хто вважається ліцензіатом? Яким документом (документами) визначено його права та обов'язки?

3.5. Інсталяція програмного забезпечення

Інсталяція та деінсталяція програм і компонентів операційної системи.

Програмне забезпечення може постачатись разом із апаратним, у складі комп'ютера, або встановлюватись із окремих носіїв (компакт-дисків, флеш-накопичувачів або мережі Інтернет). Здебільшого для того щоб програмний засіб можна було використовувати, необхідно виконати низку операцій.

Для інших програмних засобів потрібно виконати певні налаштування. Це зумовлено тим, що складні програмні засоби використовують не лише той код, який безпосередньо належить їм, але й код так званих системних бібліотек – частин ОС, що містять код, призначений для виконання операцій, подібних у багатьох задачах. Наприклад, більшістю програм користувач керує з використанням миші або сенсорного екрана, тому недоцільно дублювати програмні засоби опрацювання переривань від миші, іншого маніпулятора в кожній програмі.

Для введення й виведення даних використовується Базова система введення – виведення (Base Input Output System – BIOS), програми якої спільні для всіх програм користувача.

Обслуговування пристроїв комп'ютера виконують програми – драйвери, які можуть бути різними на різних комп'ютерах. Тому для того щоб програмний засіб працював правильно, необхідно створити для нього

певні умови, надати адреси, за якими розташовані необхідні програми, до яких він звертатиметься.

Дуже важливим є й те, що користувач обов'язково має ознайомитись із ліцензійною угодою, погодитися з умовами, на яких він користуватиметься певним програмним засобом.

У ОС родини Windows дані щодо встановлених програм і їх налаштувань зберігаються у спеціальному текстовому файлі – **реєстрі**.

Реєстр Windows має чітко визначену структуру. Будь-яке її порушення призводить до порушення роботи ОС.

Основна частина реєстру – це ключі (або параметри), в яких зберігається вся інформація щодо налагодження ОС та програм користувача. Кожен параметр реєстру Windows відповідає за певну властивість системи. Ключі з даними про налаштування комп'ютера і програм об'єднані в розділи, які своєю чергою є підрозділами більших розділів, і т.д. У реєстрі, зокрема, зберігаються дані, які вказують, файли яких типів опрацьовуються певною програмою, інші налаштування.

У файл реєстру під час встановлення записуються налаштування для кожної програми.

Усі налаштування виконуються спеціальними програмами, які постачаються разом із програмним забезпеченням. Програми, призначені для встановлення й налаштування програмних засобів, називаються **інсталяторами**, а процес установлення називається **інсталяцією програм**.

Операційні системи також установлюються з використанням спеціального програмного забезпечення.

Оскільки ОС є дуже складними програмними комплексами, які постійно вдосконалюються, більшість із них мають у своєму складі засоби налаштування й оновлення.

Панель керування показана на рис. 3.10. Основні налаштування, які доводиться виконувати користувачеві: створення облікових записів (account) для користувачів; оформлення робочого стола і встановлення роздільної здатності екрана, розмірів шрифту у вікнах; встановлення часу, дати, мов уведення і мови графічного інтерфейсу; вилучення програм; оновлення ОС.

На рис. 3.11 показано вихід на налаштування годинника, календаря, регіональних стандартів і мовних параметрів інтерфейсу для ОС Windows 7.

***Оновлення** сучасних ОС (Windows 7, 8, 10, Linux Ubuntu та інших) відбувається автоматично.*

Необхідною умовою для цього є наявність підключення комп'ютера до мережі Інтернет.

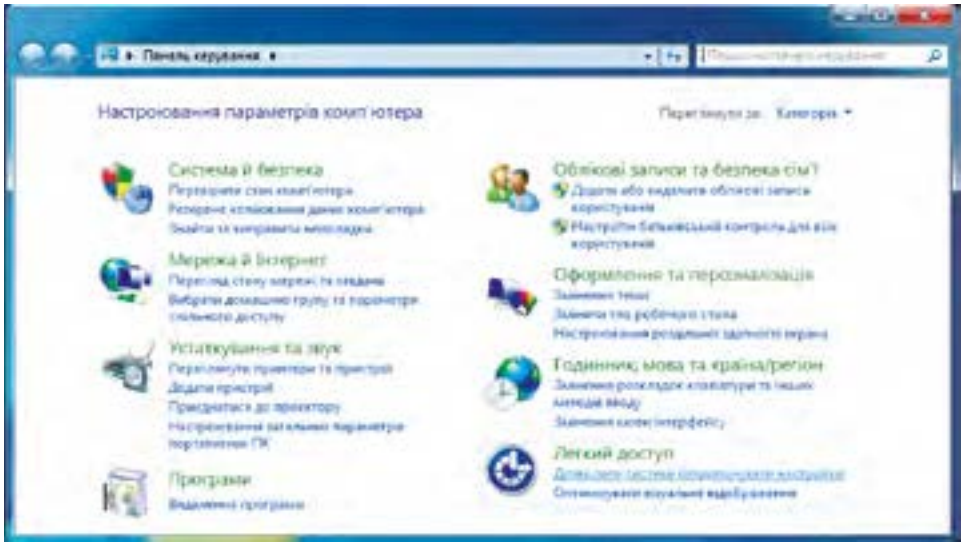


Рис. 3.10. Панель керування ОС Windows 7

Рис. 3.11. Вихід на налаштування регіональних і мовних параметрів інтерфейсу ОС Windows 10

Інсталяцію програм дозволяється виконувати тільки користувачеві з правами адміністратора. Як уже зазначалося, цей процес відбувається за кілька етапів, протягом яких необхідно виконувати певні дії, відповідати на запитання програми, вказувати при потребі особливості встановлення програм (рис. 3.12–3.15).

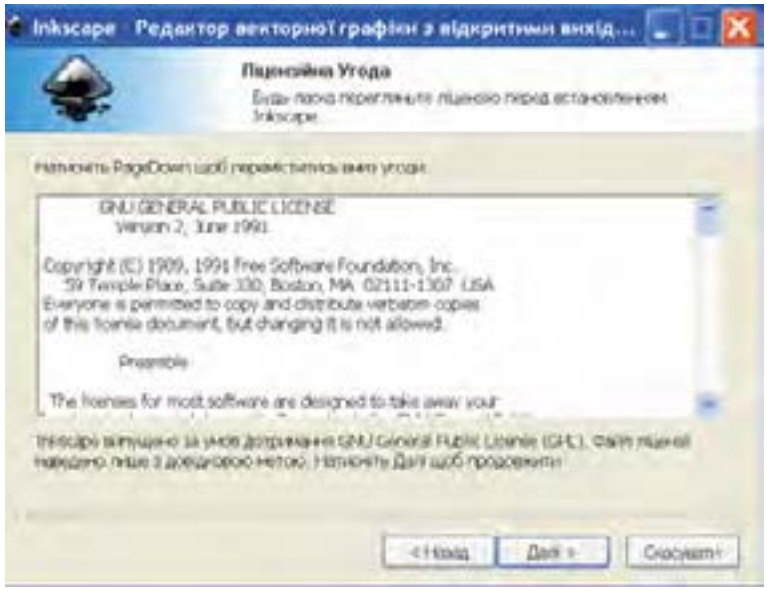


Рис. 3.12. Ознайомлення з ліцензійною угодою

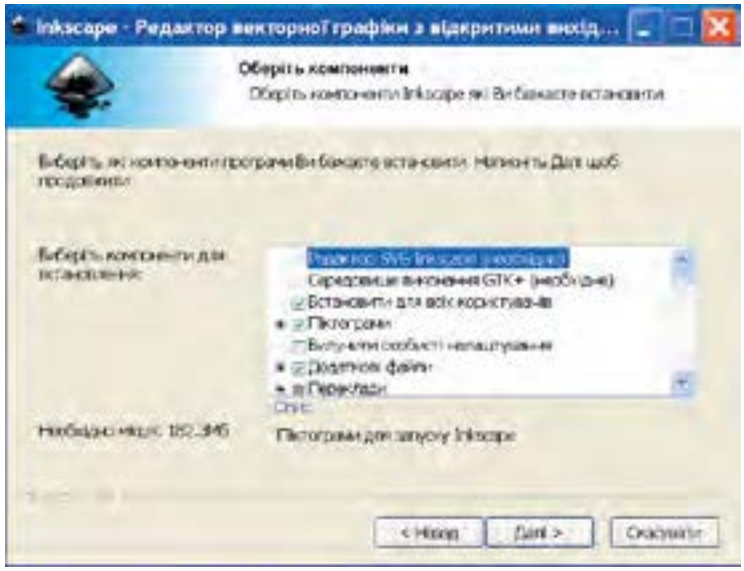


Рис. 3.13. Налаштування складу програмного засобу та мови його інтерфейсу

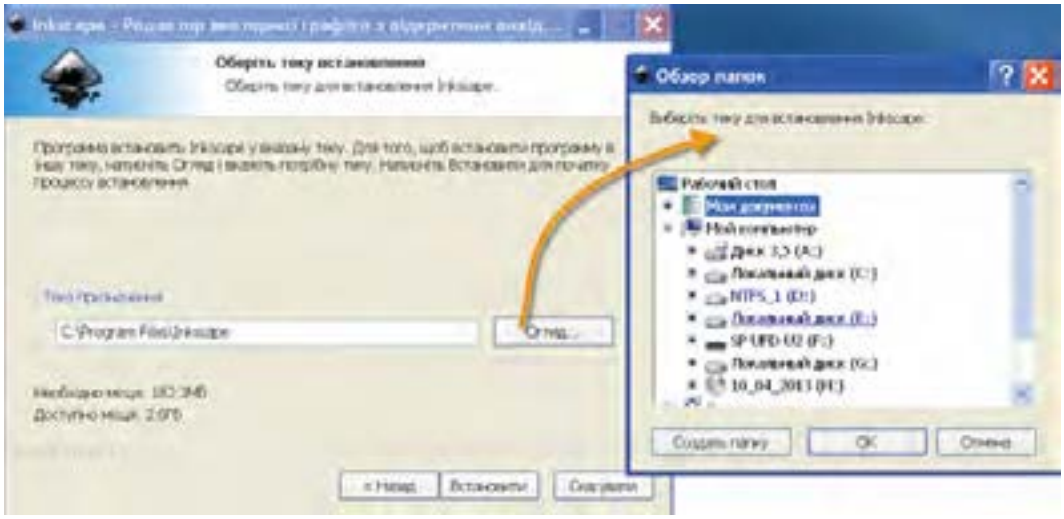


Рис. 3.14. Налаштування місця встановлення програмного засобу

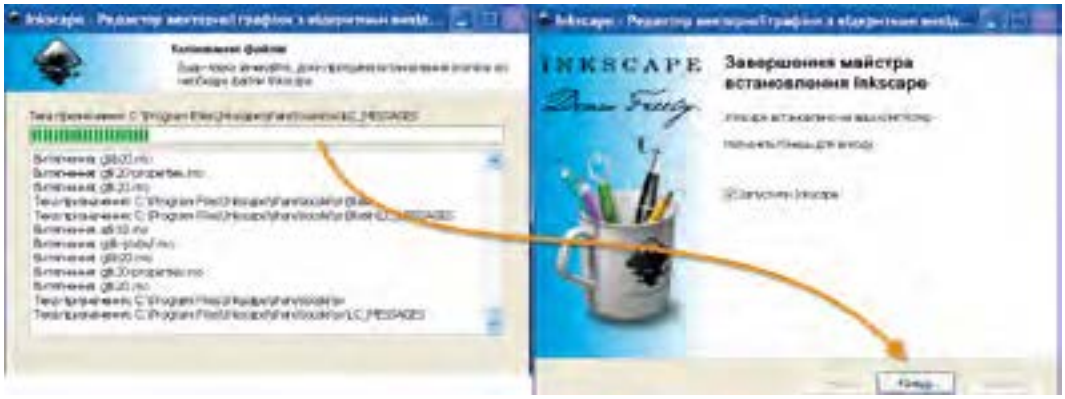


Рис. 3.15. Перебіг встановлення програмного засобу та його завершення

Перевіряємо себе

1. Які параметри ОС користувач налагоджує після її встановлення?
2. Навіщо і які дані зберігає ОС для кожної програми користувача?
3. Чому не можна вилучати файли програм, які інстальовано?
4. Як можна переглянути й відкоригувати перелік програм, що завантажуються автоматично після увімкнення живлення комп'ютера?
5. Якою є ліцензія програмного засобу Inkscape?
6. Яким чином можна додати мови введення і розкладки клавіатури?

Виконуємо

1. Зайдіть у панель налагодження часу, дати. Яким чином можна змінити її налаштування? Виконайте коригування дати й часу.

2.   Визначте мови введення, які можна використовувати, і мови, для яких здійснюватиметься контроль правильності введення тексту, наявність розкладок клавіатури для них. Запишіть, обговоріть ознаки, на підставі яких зроблено висновки.

3.  Як змінити роздільну здатність зображення на екрані монітора? Визначте, яку роздільну здатність встановлено (на комп'ютері, за яким Ви працюєте). Чи відрізняється вона від рекомендованої для наявного монітора?

3.6. Архівування даних

Архівування даних. Стиснення даних, види стиснення даних. Архіватори. Типи архівів. Операції над архівами.

У сучасних персональних комп'ютерах можуть зберігатися різноманітні типи даних (графічні, звукові та інші), які потребують величезних обсягів пам'яті. Задля економії обсягу пам'яті зовнішніх пристроїв, на яких зберігаються такі дані, здійснюється їх стиснення.

Стиснення даних – процес перекодування початкового файла даних у новий файл меншого розміру, із якого можна відновити початковий.

Є різні алгоритми стиснення даних.

Один із найпростіших полягає в тому, що на кодування символів, які використовуються часто (наприклад, *а, о, к*), відводиться менше бітів, ніж для кодування символів, частота використання яких невелика.

Інший алгоритм передбачає пошук у вихідному файлі ланцюжків символів, які повторюються, створення "бібліотеки" таких ланцюжків і кодування кожного з них коротким (зазвичай – дво- або трибайтним) номером. Алгоритми стиснення даних поділяються на алгоритми з **втратами** і **без втрат**.

Стиснення без втрат *– таке стиснення, за якого файл, відновлений зі стиснутого, повністю відповідає оригіналу.*

Стиснення з втратами *– таке стиснення, коли у відновленому файлі є спотворення, але для людини вони непомітні.*

Стиснення **"без втрат"** застосовується для текстових і числових даних, програм, тому що відсутність навіть одного символу може призвести до неправильного тлумачення слова, збою виконання програми, а стис-

нення “**з втратами**” – для графічних (зокрема, формат *.jpg), звукових і відеоданих.

✓ *Алгоритм кодування «**з втратами**» полягає у поділі зображення (світлини, відеокадру) на ділянки з кількох пікселів і надання цій ділянці кольору, середнього для цих пікселів.*

Стиснення даних характеризується двома основними параметрами:

1. Коефіцієнтом стиснення, який визначається відношенням обсягу стиснутого файла до обсягу початкового файла. Значення цього коефіцієнта міститься в межах від 1 (стиснення вже стиснутих даних) до 10 і більше.

2. Швидкістю стиснення, яка визначає тривалість процесу стиснення і наступного відновлення даних з архіву.

Стиснення даних здійснюється за допомогою спеціальних програм. У ОС Windows XP і старших версіях для цього існують вбудовані програмні засоби. Вони виконують стиснення на тих зовнішніх пристроях, які підтримуються файловою системою NTFS.

✓ *Стиснення даних в ОС Windows 7 може бути виконане так.*

1. У програмі **Провідник** виокремити об’єкти, які потрібно стиснути. Виокремимо, наприклад, на диску **F:** папку СТАТТЯ_ОС і файл Стаття_БЕЗПЕКА. Потім слід відкрити контекстне меню виділених об’єктів і виконати команду **Властивості**.

2. Відкриється вікно **Властивості**, зміст якого на вкладці **Загальні** подано на рис. 3.16.

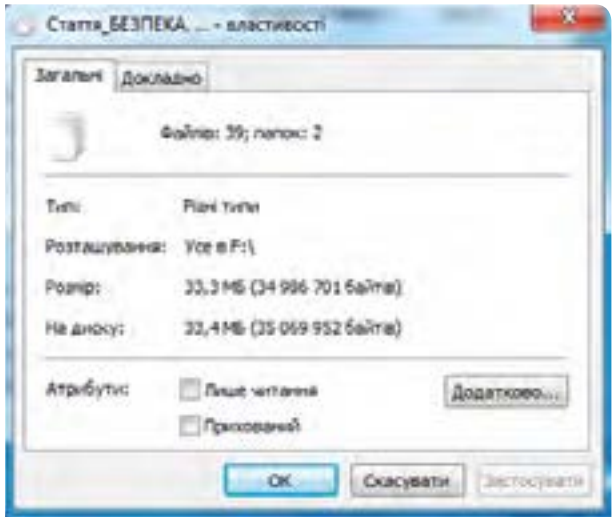


Рис. 3.16. Вікно властивостей виокремлених об’єктів

Як бачимо з рис. 3.16, виокремлені об’єкти мають 2 папки і 39 файлів різних типів. На диску вони займають 33,4 МБ. У цьому вікні слід

натиснути кнопку **Додатково...**. Відкриється вікно **Додаткові атрибути**, зображене на рис. 3.17.

У цьому вікні увімкнемо прапорці **Дозволити індексування вмісту цих файлів...** і **Стискати вміст для заощадження місця на диску** і натиснемо кнопку **ОК**. У результаті відкриється вікно **Підтвердження змінення атрибутів** (якщо воно не відкриється, то у вікні, зображеному на рис. 3.17, слід натиснути кнопку **ОК**, після чого воно має відкритися).

3. Увімкнемо у цьому вікні перемикач для вкладених папок і файлів і натиснемо кнопку **ОК**. Після цього починається процес стиснення, який залежно від обсягу виділених об'єктів може виконуватися від кількох секунд до десятків хвилин.

Зверніть увагу, що стиснуті об'єкти у вікні програми **Провідник** помічені іншим кольором. Для перегляду результатів стиснення слід виділити відповідні об'єкти, відкрити їх контекстне меню і виконати команду **Властивості**.

*Для скасування стиснення папок і файлів необхідно виокремити їх, відкрити вікно **Додаткові атрибути** (рис. 3.17), вимкнути у ньому прапорець **Стискати вміст для заощадження місця на диску** і натиснути кнопку ОК.*

Окремим випадком застосовування стиснення файлів є їх архівування.

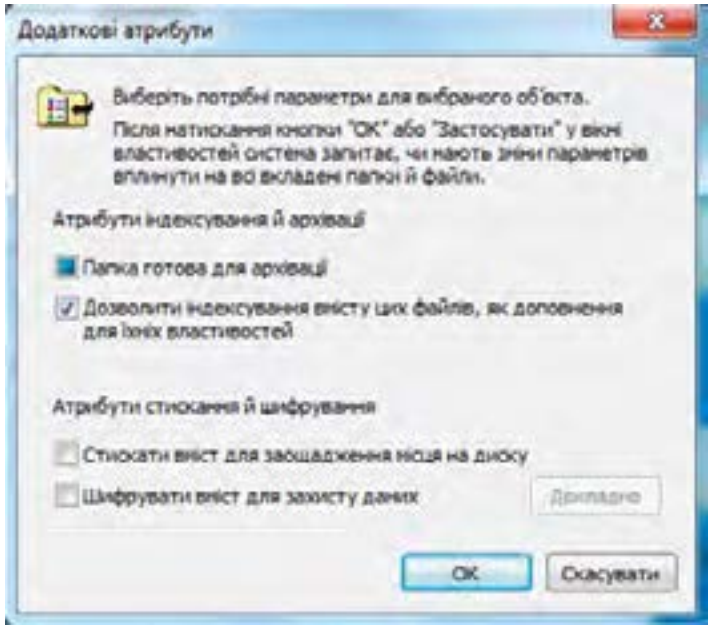


Рис. 3.17. Вікно **Додаткові атрибути**

Архівування – це стиснення файлів і, якщо їх два і більше, об'єднання їх в один архівний файл з метою довготривалого його зберігання, а також для передавання.

Програми-архіватори виконують як стиснення, так і відновлення оригіналу файла з архіву. Останній процес називається **розархівуванням**. Сьогодні популярні архіватори WinRAR, WinZip, 7-Zip. Архіватори працюють не лише з файлами власного формату, але й з файлами інших форматів. Наприклад, архіватор WinRAR працює з файлами власного формату, з файлами формату ZIP та іншими.

✓ *Можливе створення архівів, що саморозпаковуються.*

Для розпакування таких файлів не потрібно жодних додаткових програм. Ці архіви мають розширення **exe**. Їх доцільно застосовувати для пересилання мережею Інтернет, якщо невідомо, чи має кореспондент відповідну програму-архіватор.

✓ *У ОС Windows 7 є вбудовані засоби архівування папок і файлів формату ZIP.*

Один із найпростіших методів створення архіву цими засобами наведений нижче:

1. У вікні програми **Провідник** виокремити папки і файли, які слід архівувати. Виокремимо, наприклад, на диску **F:** папку СТАТТЯ_ОС і файл Стаття_Безпека.

2. Відкрити контекстне меню виокремлених об'єктів, встановити курсор на пункт **Надіслати** й у підменю, що відкриється, виконати команду **Стиснута папка** (рис. 3.18). На диску **F:** з'явиться папка, назва якої збігається з назвою одного з об'єктів, виділених для архівування (для прикладу, що розглядається, ця папка має назву Стаття_БЕЗПЕКА).

У створену архівну папку можна додавати папки і файли, які під час копіювання стискаються. З архівної папки об'єкти можна копіювати у звичайну папку. Під час копіювання здійснюється їх розпакування.

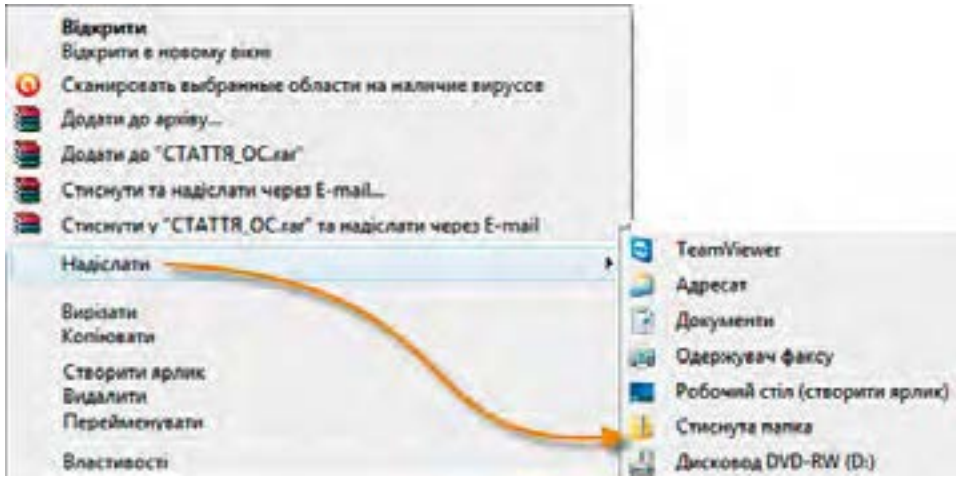


Рис. 3.18. Контекстне меню виокремлених об'єктів

Обмеження розмірів носіїв даних і обмеження розмірів файлів, які можуть бути приєднані до електронних листів, спричинили потребу в створенні **багатотомних архівів.** Багатотомну технологію підтримують формат *.rar і деякі інші.

Архіватор WinRAR має зручний інтерфейс, забезпечує високий ступінь стиснення і має добрі функціональні можливості. Для запуску програми WinRAR слід натиснути кнопку **Запустити** і виконати команди **Усі програми → WinRAR.** Відкриється вікно програми, фрагмент якого зображено на рис. 3.19.

Рис. 3.19. Вікно архіватора WinRAR

Для створення нового архіву потрібно у вікні програми WinRAR за допомогою команди **Змінити диск** меню **Файл** перейти до диска, в якому містяться об'єкти для архівування. Перейдемо, наприклад, до диска **F:**, у якому виокремимо папку СТАТТЯ_ОС і файл Стаття_БЕЗПЕКА, які будемо архівувати. Після цього слід натиснути кнопку **Додати** або в меню **Команди** виконати команду **Додати файли до архіву**. Відкриється вікно **Ім'я та параметри архіву** (див. рис. 3.19).

У цьому вікні можна ввести нове ім'я архіву або погодитися із запропонованим, вибрати формат архіву, метод стиснення (звичайний, найкращий тощо) та інші параметри. Уведемо, наприклад, ім'я *Перший*, а інші параметри залишимо за замовчуванням. Після встановлення параметрів слід натиснути кнопку **ОК**, у результаті почнеться процес архівування. Після його завершення на диску **F:** з'явиться архівна папка *Перший.rar*.

Видобути файли з архіву можна різними способами. Наприклад, у вікні програми WinRAR знайти і відкрити архівну папку, для чого слід

двічі клацнути її ім'я. Якщо відкрити папку *Перший.rar*, відкриється вікно, фрагмент якого зображено на рис. 3.20.

Рис. 3.20. Вікно архівної папки Перший.rar

У цьому вікні виокремимо папки і файли, які необхідно видобути (виокремимо, наприклад, папку СТАТТЯ_ОС) і натиснемо кнопку **Видобути до...** або в меню **Команди** виконаємо команду **Видобути файли до зазначеної папки.**

У результаті відкриється вікно **Шлях та параметри видобування**. У цьому вікні вводимо ім'я папки, до якої слід записати об'єкти з архіву, і натискаємо кнопку **ОК** (уведемо, наприклад, ім'я *Початкова* у кореневій папці диска **F:**).

Якщо тепер відкрити папку *Початкова*, побачимо в ній розархівовану папку СТАТТЯ_ОС. Якщо видобування не виконається, то буде видане відповідне діагностичне повідомлення.

Для створення саморозпаковуваного архіву необхідно у вікні **Ім'я та параметри архіву** програми WinRAR (див. рис. 3.19) увімкнути прапорець **Створити SFX-архів** (англ.: Self-extract).

Для створення багатотомного архіву слід у вікні **Ім'я та параметри архіву** вказати обсяг тому в полі **Розбити на томи розміром.** За замовчуванням кожний том отримує ім'я **ім'я_тому.partNNN**, де NNN – номер тому. Для розпаковування архіву всі томи мають зберігатися в одній папці, а розпаковувати їх слід починаючи з першого.

Перевіряємо себе

1. Як можна перевірити результат стиснення файлів засобами ОС Windows 7?
2. З якою метою виконується стиснення даних?
3. Як можна скасувати стиснення файлів у ОС Windows 7?
4. З якою метою виконується архівування даних?
5. Які архіви називають багатотомними?

6. Які існують типи алгоритмів стиснення даних?

7. Поясніть методику стиснення даних засобами Windows 7.

8. Коли доцільно застосовувати архіви, що саморозпаковуються?

9. Поясніть порядок створення архіву в ОС Windows 7.

10. Поясніть сутність стиснення даних.

11. Назвіть основні параметри алгоритмів стиснення даних.

12. Поясніть методику створення архіву засобами WinRAR.

13. Як можна видобути файли з архіву WinRAR?

Виконуємо

У особистій папці створіть папку “Резервні копії”. Скористайтеся послугою **Пошук** і знайдіть на комп’ютері 3 довільних малюнки формату *.jpg й 3 довільних малюнки формату *.bmp. Скопіюйте їх із вікна пошуку в папку “Малюнки” (ця папка має бути вже створена у вашій особистій папці). Заархівуйте малюнки формату *.jpg і малюнки формату *.bmp в окремі архіви (назвіть їх jpg і bmp). Запишіть у зошит розміри архівів та сумарні розміри вихідних файлів. Обчисліть коефіцієнт стиснення для кожного архіву. Зробіть висновки.

3.7. Запис даних на оптичні носії. Форматування та копіювання дисків

Оптичний диск – носій даних у вигляді полікарбонатного диска, призначеного для запису й відтворення даних за допомогою лазерного променя. Перші оптичні диски для комп’ютерів почали випускати на початку 1980-х років. Основними типами дисків нині є “звичайні” компакт-диски CD діаметром 12 см, ємність яких становить близько 700 МБ (вважаються застарілими), і диски DVD (англ. Digital Versatile Disc – цифровий багатоцільовий диск, або англ. Digital Video Disc – цифровий відеодиск) ємністю від 4,7 ГБ і більше, залежно від кількості сторін і шарів запису. Нині поширені диски Blu-ray Disc, або скорочено BD (англ. blue ray – блакитний промінь і англ. disc – диск) ємністю від 25 ГБ і більше, залежно від кількості сторін і шарів запису.

Для кожного типу оптичних дисків є види, які позначаються символами:

ROM (англ. *read only memory*) – диски, запис на які здійснюється під час виготовлення, використовуються тільки для читання;

R (англ. *read*) – диски одноразового запису;

RW (англ. *read write*) – диски багаторазового запису, на яких можна неодноразово видаляти й записувати нові дані.

Таких же видів існують і дисководи. Дисководи RW забезпечують читання і запис, дисководи R забезпечують лише читання.

Нові диски типів R і RW перед використанням потребують обов'язкового форматування, тобто нанесення початкової розмітки.

Для роботи з оптичними дисками розроблені спеціальні програмні засоби. Водночас ОС Windows, починаючи з версії XP, і сучасні версії ОС Linux, мають вбудовані програмні засоби, які забезпечують усі дії з дисками (форматування, копіювання, вилучення та інші операції). Тому нині немає особливої потреби у використанні програми Nero або іншої.

Оптичні диски можуть бути відформатовані або для запису файлів даних, або для запису відео та звуку. Після встановлення нового диска в дисковод має відкритися вікно **Автовідтворення**, зображене на рис. 3.21. Якщо воно не відкриється, то у вікні програми **Провідник** необхідно знайти піктограму диска й двічі клацнути на ній лівою кнопкою миші.

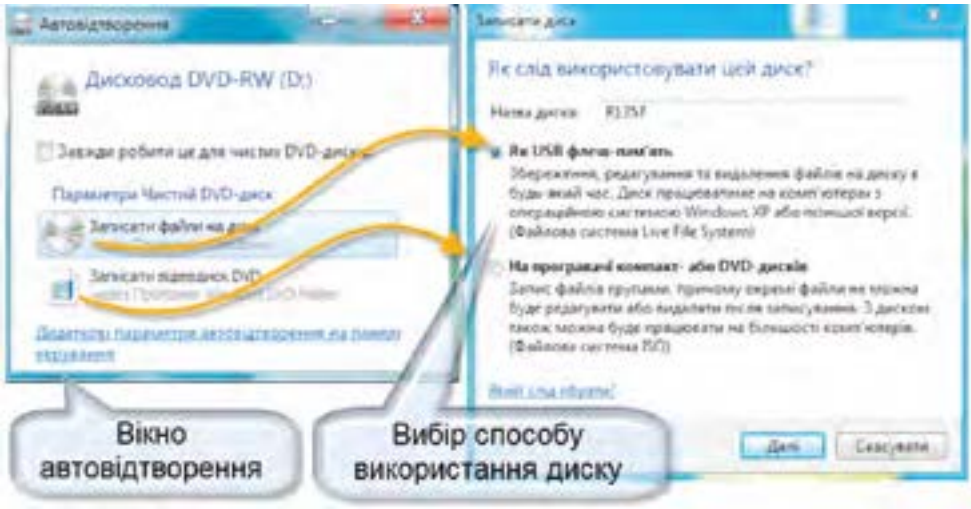


Рис. 3.21. Вікно для початку форматування диска

У цьому вікні натиснемо кнопку **Записати файли на диск**. Відкриється вікно **Записати диск**. У цьому вікні вмикаємо прапорець **Як USB флеш-пам'ять**, уводимо ім'я диска, наприклад R1357, і натискаємо кнопку **Далі**. Почнеться форматування диска, яке може тривати більше 10 хв. Під час форматування не можна виймати диск і виконувати над ним інші операції. Динаміка форматування відображається на екрані монітора.

Якщо потрібно одразу після форматування записати на цей диск папки і файли, слід відкрити вікно програми **Провідник**, для чого можна натиснути кнопку **Відкрити папку для перегляду файлів**. Відкриється вікно, зображене на рис. 3.22.

Рис. 3.22. Вікно для копіювання або перенесення об'єктів на диск

З цього моменту виконання операцій копіювання, переміщення, перейменування та вилучення файлів і папок з диска виконується за допомогою програми **Провідник** і принципово не відрізняється від таких самих операцій на жорсткому магнітному диску або флеш-пам'яті.

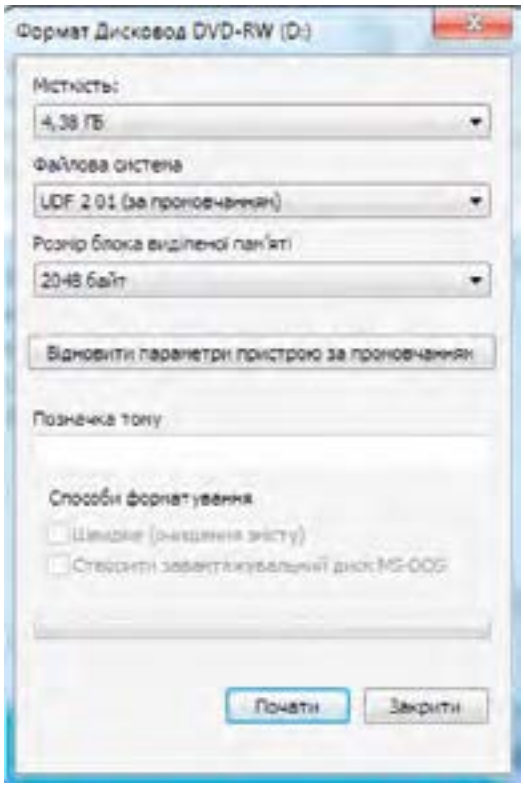


Рис. 3.23. Вікно для початку форматування диска

Якщо в попередньому сеансі роботи диск був відформатований, але запису файлів на нього не було, то після його вставлення у дисковод відкриється вікно, зображене на рис. 3.22. Після цього можна записувати на диск дані й виконувати над ними операції.

Форматування нового диска для запису на нього файлів даних можна виконати й таким способом. Після вставлення диска в дисковод у вікні програми **Провідник** відкрити контекстне меню диска, в якому виконати команду **Форматувати...**. Відкриється вікно, зображене на рис. 3.23.

У цьому вікні вибираємо файлову систему за замовчуванням

(UDF 201), розмір блока (кластера) – стандартний (2048 байт), уводимо назву диска, наприклад R1357, і натискаємо кнопку **Почати**.

Після завершення форматування з'явиться вікно відповідного повідомлення, у якому слід натиснути кнопку **ОК**.

Для очищення оптичного диска слід у вікні програми **Провідник** відкрити його контекстне меню і виконати команду **Стерти диск**. Відкриється вікно **Майстра очищення диска**, в якому необхідно натиснути кнопку **Далі**. На екрані буде відображатися динаміка очищення диска. Для повторного використання цього диска його слід форматувати ще раз.

Перевіряємо себе

1. Які існують типи оптичних дисків?
2. Чи можна на дисководі типу R працювати з дисками RW?
3. Які обсяги пам'яті мають оптичні диски?
4. Які операції можна виконувати на оптичних дисках?
5. З якими дисками можуть працювати дисководи типу R?
6. Поясніть методику форматування оптичних дисків засобами Windows 7.
7. Опишіть властивості диска DVD-R. Чим вони відрізняються від властивостей диска DVD-RW?

Виконуємо

Підготуйте до запису диск DVD-RW. Запишіть на нього декілька файлів.

3.8. Дефрагментація пристроїв пам'яті з файловими системами, визначення розкладу її проведення

У процесі роботи комп'ютер постійно зберігає різноманітні дані на зовнішньому запам'ятовуючому пристрої (вінчестері). При цьому файли на ньому не завжди розташовуються найкращим чином. Нераціональне розміщення файлів на диску з часом призводить до зниження швидкості доступу до даних і негативно впливає на довговічність запам'ятовуючого пристрою на магнітних дисках.

Нераціональне розміщення файлів на диску виникає здебільшого тому, що під час запису великого файла на диску може не бути великої безперервної вільної області, в яку цей файл помістився б повністю. Якщо сумарний розмір усіх вільних областей запам'ятовуючого пристрою перевищує розмір файла, що зберігається, він буде поділений на декілька фрагментів, які збережуться в цих вільних областях.

Процес запису файлів на запам'ятовуючий пристрій у вигляді окремих фрагментів називається ***фрагментацією****, а такий файл –* ***фрагментованим****.*

Під час роботи з такими файлами витрачається додатковий час на пошук необхідних фрагментів файлів, а зчитуюча магнітна головка жорсткого диска постійно переміщується між ними. Унаслідок цього збільшується час доступу до даних і виникає підвищене зношування обладнання. Що більше на запам'ятовуючому пристрої фрагментованих файлів, то відчутніший негативний вплив фрагментації на його роботу.

Але якщо на диску є ще досить місця, можна здійснити "збирання" фрагментів фрагментованих файлів, тобто здійснити його **дефрагментацію**, усунувши таким чином вказані проблеми.

Дефрагментація *– процес, зворотний до фрагментації. Він передбачає перерозподіл файлів на диску і розташування усіх їх фрагментів в одній безперервній області.*

Увага! Усе, що написано далі щодо ***дефрагментації,*** не стосується твердотільних накопичувачів – дисків SSD, флеш-пам'яті тощо. Застосування дефрагментації до них не збільшує швидкість передавання даних, але може суттєво скоротити час життя носія.

Що активніше використовується комп'ютер, то частіше потрібно контролювати ступінь фрагментації його дисків. Рівень фрагментації значною мірою залежить також від кількості вільного простору на диску (що його більше, то менше фрагментуються файли) та файлової системи носія. Дефрагментацію логічних розділів жорсткого диска необхідно проводити частіше в тому випадку, якщо вони відформатовані у файловій системі FAT. Файли під час запису на такі запам'ятовуючі пристрої часто фрагментуються навіть за наявності досить великих безперервних вільних областей на диску. NTFS є досконалішою файловою системою. Ступінь дефрагментації NTFS-носіїв зазвичай менший. Дефрагментація потрібна їм значно рідше.

Розпочинаючи з Windows Vista, дефрагментація вмикається автоматично, за розкладом у фоновому режимі, який встановлюється або за замовчуванням самою системою, або налагоджується користувачем.

На рис. 3.24 показано, як у Windows 7 запустити дефрагментацію або налаштувати її періодичний запуск.

Ефективне розміщення даних на диску забезпечить максимальну швидкість системи, але для цього потрібно надати системі можливість виконати заплановану дефрагментацію.

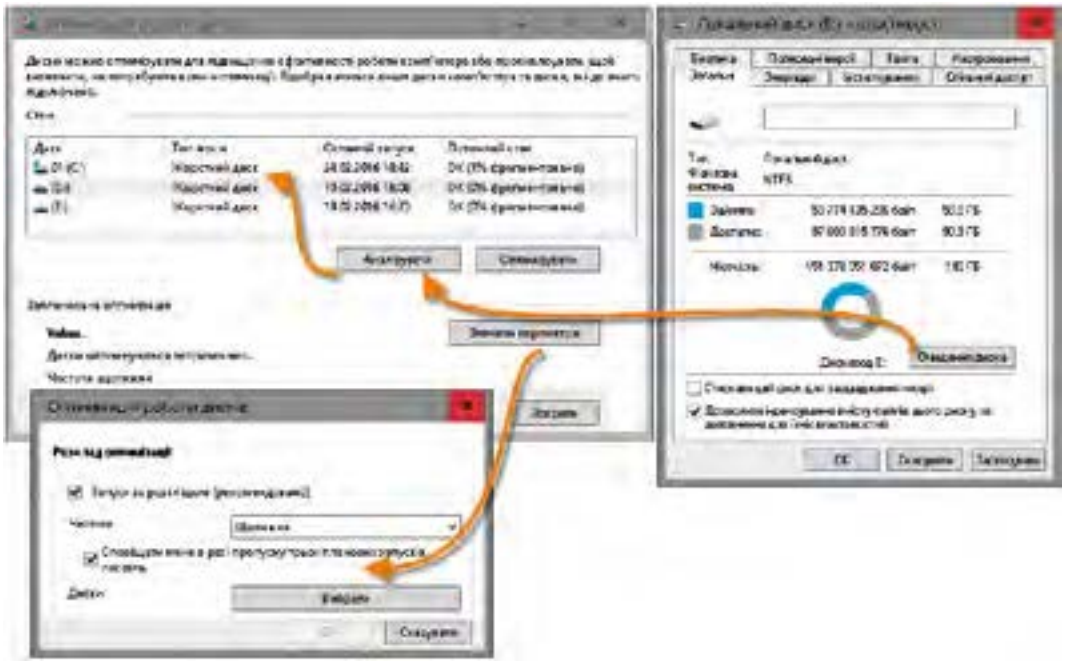


Рис. 3.24. Вікно для початку дефрагментації диска

Практична робота № 3

Тема:	Конфігурування комп'ютера під потреби користувача
Мета:	Набути практичних навичок конфігурування комп'ютера

Вказівки. Конкретні рекомендації можна навести лише щодо вибору монітора. Нині це має бути монітор рідинно-кристалічного типу з діагоналлю не менше 19 дюймів. Інші параметри залежать від наявності коштів. Щодо інших пристроїв та їх характеристик можна навести лише загальні рекомендації. Насамперед слід визначити мету придбання комп'ютера.

1. У офісі комп'ютер застосовується зазвичай для роботи з текстом, таблицями, для бухгалтерської справи, електронного листування та перегляду сторінок в Інтернеті. Для таких робіт вистачає бюджетного варіанта ПК із двохядерним процесором частотою близько 1,5 ГГц, оперативною пам'яттю 2 ГБ, жорстким диском ємністю 200–440 ГБ. Потужність блока живлення 300 Вт. Влаштовує вбудована відеоплата. Для офісних робіт часто використовують принтер і сканер.

2. Удома комп'ютер використовується для роботи в Інтернеті, для простих комп'ютерних ігор, перегляду телевізійних передач і фільмів, для роботи з мультимедія. Тому потрібен потужніший комп'ютер з 2–4-ядерним процесором частотою близько 2 ГГц, оперативною пам'яттю 2–4 ГБ,

жорстким диском 500–700 ГБ. Обов'язково потрібен пристрій CD- або DVD-типу із можливістю запису, блок живлення не менше 400 Вт.

3. Для ігор бажано мати комп'ютер із 4-ядерним процесором частотою близько 3 ГГц, оперативною пам'яттю 4 ГБ, жорстким диском ємністю 1 ТБ. Комп'ютер має мати добру відеокарту і звукову систему. Блок живлення потужністю не менше 450 Вт. Обов'язково потрібен пристрій DVD або Bly-Ray із можливістю запису.

4. Дизайнерські роботи вимагають найпотужнішого комп'ютера. Він має бути орієнтований на використання 64-бітних ОС. Процесор бажано мати частотою 3,3 Гц, а кількість ядер – не менше 4, оперативна пам'ять – не менше 8 ГБ, вінчестер – 1 ТБ.

Виконання роботи

1. Із учнів класу формується 3–4 команди, а також обирається журі у складі 3–4 учнів. Кожна команда повинна обговорити й прийняти рішення: з якою метою можуть використовуватися комп'ютери, основні дані яких наведені в таблиці. За 5 хв представник від кожної команди обґрунтовує рішення, а члени журі оцінюють якість обґрунтування.

2. Команди і журі залишаються в тому самому складі. Учитель пропонує кожній команді визначити склад і параметри комп'ютера учня і комп'ютера вчителя. За 5 хв представник від кожної команди обґрунтовує свій варіант. Журі оцінює доповіді й за підсумками двох виступів оголошує кращу команду. Остаточні підсумки підводить учитель.

Пристрої	**Комп'ютери**			
	№ 1	**№ 2**	**№ 3**	**№ 4**
Процесор	2,0 ГГц, 2 Cores, Cashe 1 МБ	2,4 ГГц, 2 Cores, Cashe 2 МБ	3,0 ГГц, 4 Cores, Cashe 4 МБ	3,6 ГГц, 4 Cores, Cashe 8 МБ
ОП	1 ГБ	2 ГБ	4 ГБ	16 ГБ
Вінчестер	300 ГБ	500 ГБ	1 ТБ	1 ТБ
Відеокарта	Вбудована		1 ГБ	2 ГБ
Привід	–	DVD+DW	DVD+RW	DVD+RW

Практична робота № 4

Тема: Створення архівів та операції над ними

Мета: Набути практичних навичок використання архівів

1. У особистій папці створити папку "Резервні копії". Скористайтеся послугою **Пошук** і знайдіть на комп'ютері три довільних малюнки формату *.jpg й три довільних малюнки формату *.bmp. Скопіюйте їх із вікна пошуку в папку "Малюнки" (ця папка має бути вже створена у вашій

особистій папці). Заархівуйте папку “Малюнки” в папку “Резервні копії” (архів назвіть pictures), застосовуючи метод максимального стиснення. Запишіть у зошит розмір архіву та сумарний розмір вихідних файлів. За допомогою калькулятора знайдіть різницю і запишіть її в зошит. Очистіть папку “Малюнки”. Розархівуйте з архіву pictures будь-який малюнок у Папку Резерв2. Додайте до архіву pictures коментар “Це архів малюнків… (ваше прізвище, ім’я)”. Перевірте архів на помилки. Заархівуйте всю свою папку в архів під назвою “Резервна папка Прізвище ім’я” на диск D: (або флеш-пам’ять).

2. Знайти на комп’ютері та заархівувати до архіву з назвою “Мікст” один файл формату *.mp3, два файли формату *.doc, три файли формату *.gif так, щоб: а) архів не потребував для розкриття наявності архіватора; б) була додана інформація для відновлення; в) архів можна було розкрити, лише вказавши пароль (пароль обов’язково запишіть у зошит). Архів має бути розміщений у папці “Резервні копії”.

СЛОВНИЧОК

Архівування – процес стиснення даних, що належать одному або кільком файлам.

Архітектура комп’ютера – спосіб побудови комп’ютера, з’єднання його окремих частин.

Бістабільний пристрій– матеріальний об’єкт, який може перебувати в одному з двох станів – нуль або одиниця, вимкнено або увімкнено.

Блочно-модульна архітектура – спосіб побудови комп’ютера з виокремленням його частин таким чином, щоб можна було їх замінювати.

Мікропрограма – програма, записана в постійному. запам’ятовуючому пристрої процесора.

Операційна система – комплекс взаємозв’язаних програм, призначений для управління обчислювальними процесами (програмами), апаратними засобами, ефективного розподілу ресурсів комп’ютера між обчислювальними процесами, а також для організації взаємодії користувача з комп’ютером.

Стиснення даних – процес перетворення даних зі зменшенням довжини коду.

Шина – набір провідників, об’єднаних певним призначенням.

Шина адреси – шина, подання двійкового коду на провідники якої зумовлює приєднання до шини даних комірки пам’яті з адресою, що відповідає двійковому коду.

Шина даних – шина, по якій із комірок внутрішньої пам’яті передаються коди даних і команд до центрального процесора і навпаки – від процесора до комірок внутрішньої пам’яті.

Шина управління – шина, по якій до внутрішньої пам’яті передаються команди від процесора і пристрою управління.

РОЗДІЛ 4. ОПРАЦЮВАННЯ ТЕКСТОВИХ ДОКУМЕНТІВ

Текстові документи, як і будь-які інші, зберігаються в електронному вигляді у файлах. Спосіб відтворення відомостей, що містяться в файлі, залежить від того, якими програмними засобами це робитиметься.

Створення, редагування та форматування списків, таблиць, колонок, символів, формул у текстовому документі. Недруковані знаки. Створення, редагування та форматування графічних об'єктів у текстовому документі. Стильове оформлення абзаців. Шаблони документів. Структура документа.

4.1. Формати файлів текстових документів. Форматування символів і абзаців (повторення)

Сучасні текстові процесори, зокрема, LibreOffice Writer і Microsoft Word, дають змогу відкривати й зберігати документи в кількох форматах. У діалоговому вікні **Збереження документа** є розкривний список **Тип файла**, де можна вибрати відповідний формат.

Існують універсальні формати текстових файлів, які можуть бути прочитані більшістю текстових редакторів, і оригінальні формати, які використовуються окремими текстовими редакторами.

✓ *Шаблон імені файла прийнято позначати так:* ***.***, **де першою зірочкою позначається** *ім'я файла, присвоєне користувачем, друга зірочка вказує на тип файла.*

✓ *Найпоширеніші формати файлів текстових документів:*

Документ Word (*.doc) – основний формат текстового процесора Word (до версії 2003 включно). У цьому форматі документи зберігаються за замовчуванням.

Word (*.docx) – основний формат текстового процесора Word (починаючи з версії 2007). У цьому форматі документи зберігаються за замовчуванням.

Шаблон документа (*.dot) – формат шаблону, на якому можуть базуватись інші документи.

Текст у форматі RTF (*.rtf) – формат RTF (Rich Text Format – розширений текстовий формат), універсальний формат, який зберігає все форматування, перетворює коди керування на команди, які можуть бути прочитані та інтерпретовані у багатьох застосунках, водночас обсяг файла суттєво зростає.

Звичайний текст (*.txt) – найбільш універсальний формат, що зберігає текст без форматування; у текст вставляються тільки символи кінця абзацу. Цей формат використовують для збереження документів, які можуть бути прочитані в застосунках, що працюють у різних операційних системах.

Веб-сторінка (*.htm; *.html) (англ. *Hyper Text Markup Language* – гіпертекстова мова розмічання) – формат Веб-сторінки; його використовують у разі розміщення документа в Інтернеті.

Формат PDF (*.pdf) (англ. *Portable Document Format*) – електронний формат файла, створений на основі формату Post Script (формату, призначеного для виведення документів на папір), який зберігає форматування документа. Використання формату PDF забезпечує незмінність вигляду документа під час перегляду файла в мережі або виведення на друк. Тобто дані файла не можуть бути випадково змінені. Аналогічним є і формат XPS (англ. *XML Paper Specification*).

Для перетворення файла текстового документа з одного формату в інший використовують спеціальні програми – програми-конвертери. У текстових процесорах такі конвертери входять до складу програмного засобу. Для надання текстовому документу вигляду, який поліпшує його сприйняття, використовують **форматування**.

Форматування тексту передбачає вибір виду, розміру та кольору символів, розміру абзацного відступу, відстані між рядками й абзацами, положення та вирівнювання абзаців відносно меж; нумерацію та маркірування абзаців і рядків; встановлення чи заборону переносу слів; встановлення параметрів розрідження (ущільнення) та зміщення елементів тексту; встановлення параметрів колонок та ін. Усі основні засоби форматування розміщені в групах **Шрифт** і **Абзац** меню **Основне**.

Для редагування формату символів потрібно:

виокремити фрагмент тексту (чи весь текст документа) й почергово (не знімаючи виділення) в меню **Основне** обрати за вимогами всі параметри форматування (розмір, тип, вид, колір, абзацний відступ тощо).

Форматування за зразком.

1. Виокремити один із фрагментів тексту, який вважатиметься зразком.
2. Двічі клацнути кнопку **Формат за зразком** на панелі **Основне**.
3. Не клацаючи кнопкою миші, перевести курсор на текст документа (він набуде вигляду щіточки, що означає запам'ятовування параметрів форматування).
4. Знайти в тексті документа фрагмент, до якого мають бути застосовані такі ж параметри форматування, як і до першого зразка.
5. Встановити (не притискаючи) курсор на початок (чи кінець) цього фрагмента.

6. Утримуючи натиснутою ліву кнопку миші, провести курсором по тексту фрагмента.

7. Відпустити ліву кнопку миші – фрагмент набуде такого вигляду, як перший відформатований фрагмент. Після виконання цих дій курсор все ще матиме вигляд щіточки – це означає, що засоби форматування можна застосувати й до інших фрагментів.

Якщо необхідно продовжити форматування фрагментів за зразком першого, курсор миші слід, не клацаючи, переміщати між фрагментами!

Форматування інших фрагментів виконується за схемою, описаною вище, але з тією різницею, що вже немає необхідності щоразу натискати кнопку **Формат за зразком**.

Після форматування фрагментів тексту за зразком першого команду **Формат за зразком** слід відмінити натисненням кнопки **Формат за зразком** або клавіші **Esc** на клавіатурі.

Одночасне форматування виділених фрагментів.

1. Виокремити один із фрагментів тексту.

2. Притримуючи клавішу клавіатури Ctrl, виокремити інший фрагмент.

3. Відпустити клавішу Ctrl та знайти наступний фрагмент, виокремити його, як і попередній.

4. Подібним чином виокремити всі фрагменти тексту документа, які мають мати одні й ті ж параметри форматування.

5. Перейти в меню **Основне** і обрати потрібні параметри.

Перевіряємо себе

1. Назвіть найпростіший формат текстового файла.
2. Що називають **Шаблоном документа**?
3. Які кнопки стрічкового меню **Основне** або опції контекстного меню використовуються для швидкого надання значення міжрядковому інтервалу; абзацному відступу?
4. Як називаються програми, що використовуються для перетворення файла текстового документа з одного формату на інший?
5. У чому полягає форматування символів тексту?
6. Назвіть способи форматування символів тексту.
7. Як можна виконати форматування фрагментів тексту за зразком?
8. Які формати текстового файла доцільно використовувати для передавання документів, якщо невідомо, яким текстовим редактором користуватимуться для його розкриття і наступної роботи?
9. Що називають **Шаблоном** документа?

Виконуємо

1. Створіть текстовий документ.
2. Збережіть створений документ у форматі Документ Word (*.docx).
3. Збережіть створений документ у форматі Документ Word 97–2003 (*.doc).
4. Збережіть створений документ у форматі RTF (*.rtf).
5. Збережіть створений документ у форматі Звичайний текст (*.txt).
6. Порівняйте розміри отриманих файлів і вигляд документів, прочитаних із них. Зробіть висновки.
7. Збережіть документ у вигляді шаблону. Які частини документа доцільно зберігати повністю?

4.2. Списки і таблиці

Створення, редагування та форматування списків, таблиць, колонок, символів, формул у текстовому документі. Недруковані знаки.

Особливим видом форматування абзаців є оформлення їх у вигляді списків.

Списком називають сукупність взаємозв'язаних абзаців (елементів списку), які позначені спеціальними **маркерами** або **номерами**.

Списки бувають **маркіровані**, **нумеровані** та **багаторівневі**.

Для швидкого нумерування або маркірування користуйтеся кнопками **Маркери**, **Нумерація**, **Багаторівневий список** вкладки **Абзац** стрічкового меню **Основне** або контекстного меню, яке виводиться після виокремлення тексту й натиснення правої кнопки миші (рис. 4.1).

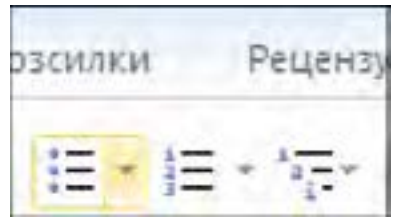


Рис. 4.1. Кнопки **Маркери**, **Нумерація**, **Багаторівневий список**

Маркіровані списки (кожен абзац на початку помічається спеціальним символом-маркером (♣ ♦ ● ▪ ☺ -)) застосовують для опису основних положень доповіді, дій користувача, переліку подій, властивостей об'єкта тощо (рис. 4.2).

У нумерованому списку на початку кожного абзацу зазначений його номер. Порядковий номер абзацу в списку може задаватися числом (наприклад, 1, 2, 3) або літерою (наприклад, а, в, с) (рис. 4.3). Такий список використовують для представлення інформації, якщо важливим є поря-

Рис. 4.2. Вікно **Бібліотека маркерів**

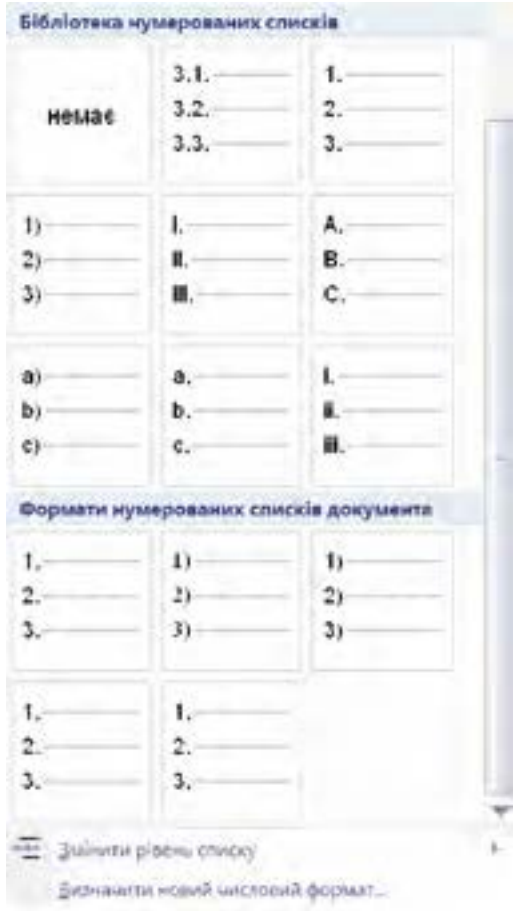


Рис. 4.3. Вікно **Бібліотека нумерованих списків**

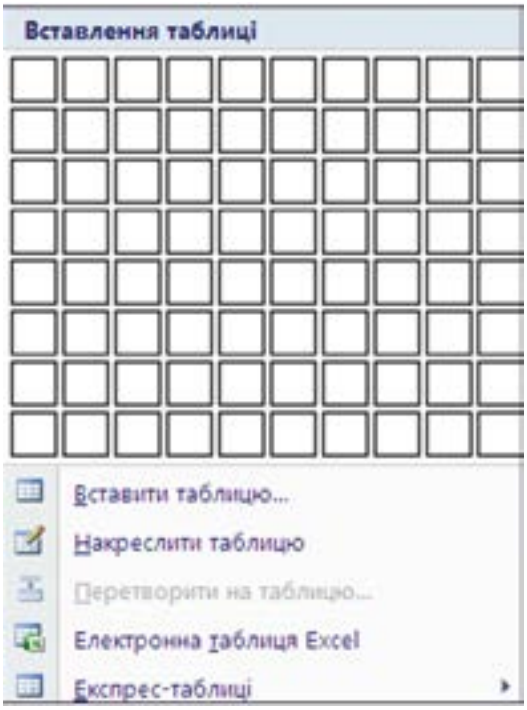


Рис. 4.4. **Вставлення таблиці**

док елементів, наприклад, опис покрокових процедур або список придбаних товарів у накладній. У багаторівневому списку абзаци пронумеровані за їх ієрархічною структурою.

Максимальна кількість вкладень елементів списку – дев'ять рівнів.

Для створення списку необхідно виокремити абзаци, які потрібно зробити елементами списку, або зробити поточним той абзац, із якого починається список.

Таблиці зазвичай використовують для упорядкування та унаочнення даних.

Дані в таблиці мають компактний вигляд і зручні для сприйняття. Таблиця складається зі стовпців і рядків, на перетині яких містяться комірки. Комірки можуть містити текст, графічні зображення, числові значення, посилання на дані з інших документів.

Нестандартне застосування таблиць. За допомогою таблиці можна форматувати документи, наприклад, розташовувати абзаци в кількох рядках, суміщати рисунок із текстовим підписом. Таблиці використовують з метою специфічного розташування текстових блоків деяких документів, наприклад, під час створення бланків, буклетів, Веб-сайтів. Межі такої таблиці роблять невидимими, тому про табличну організацію відповідних даних можна й не здогадатися.

У текстовому процесорі можна: створити порожню таблицю, а потім заповнити комірки; перетворити наявний текст на таблицю.

Для розташування таблиці в тексті потрібно вибрати меню **Вставлення**, де обрати групу **Таблиці**. У списку, який відкрився, вибрати потрібну команду (рис. 4.4).

Виконання команди **Вставити таблицю** розпочинається з діалогового вікна, в якому можна вказати кількість стовпців і рядків майбутньої таблиці або вибрати інший варіант додавання таблиці в документ.

Електронна таблиця Excel.Команда передбачає вставлення таблиці як об'єкта застосунку Excel.

Експрес-таблиці. Передбачається використання шаблону таблиці з колекції програми. Дані шаблону надалі можна замінити на потрібні (рис. 4.5).

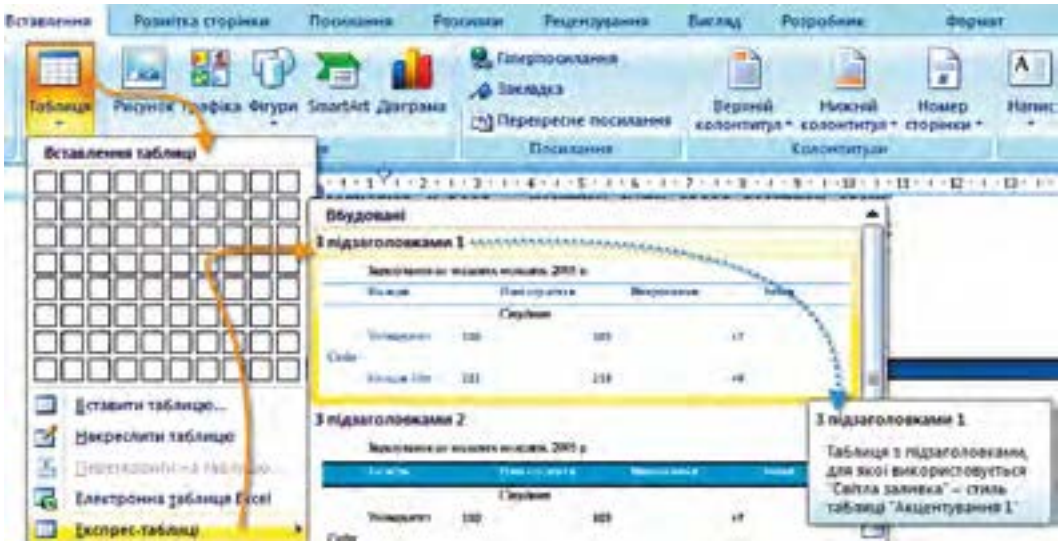


Рис. 4.5. Вставлення Експрес-таблиці

Засоби переміщення в таблиці та введення даних

Результат	Дія
Переміщення у наступну комірку	Натисніть клавішу TAB. Якщо курсор міститься в останній комірці таблиці, натискання цієї клавіші додає до таблиці новий рядок
Переміщення в попередню комірку	Натисніть клавіші SHIFT+TAB
Переміщення в попередній або наступний рядок	Натисніть клавішу СТРІЛКА УГОРУ або СТРІЛКА УНИЗ
Початок нового абзацу	Натисніть клавішу ENTER
Додавання нового рядка в кінці таблиці	Натисніть клавішу TAB у кінці останнього рядка

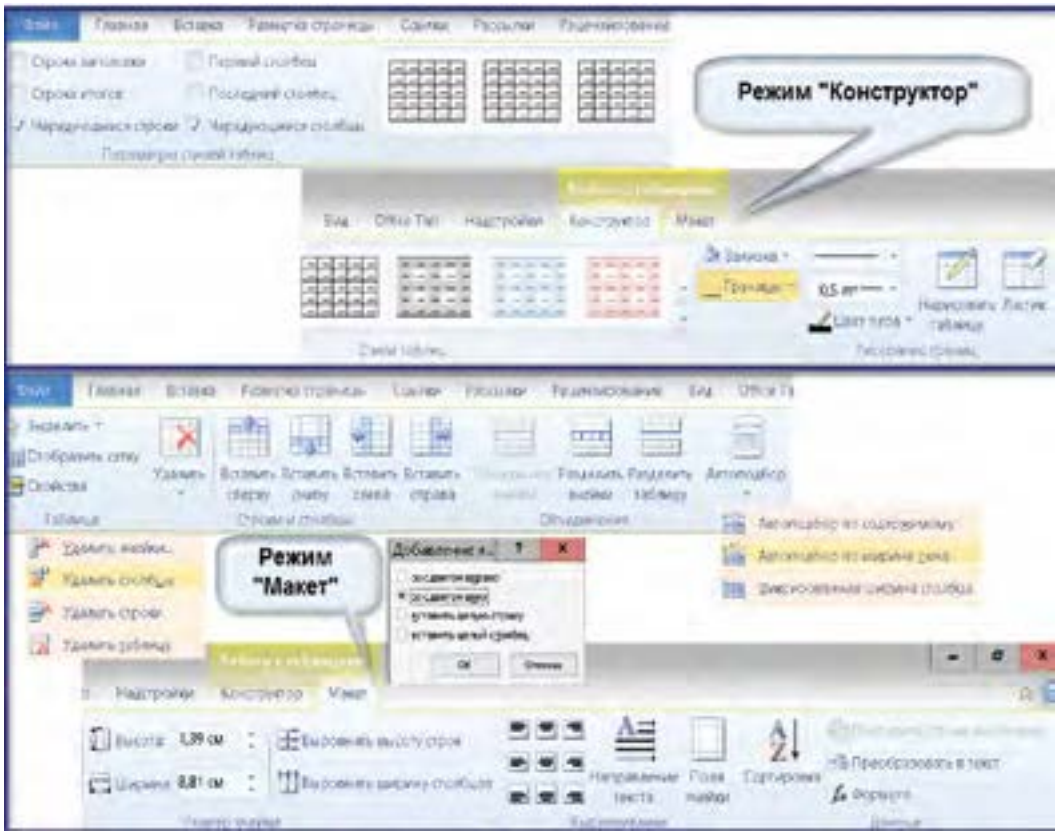


Рис. 4.6. Контекстні вкладки **Табличні знаряддя**

Засоби переміщення в таблиці та введення даних наведені в таблиці. Після вставлення таблиці (або при актуалізації таблиці, що вже існує) на екрані з'явиться меню **Табличні знаряддя**, де розташовані дві вкладки **Конструктор** і **Макет** (рис. 4.6).

За допомогою засобів вкладок **Конструктор** і **Макет** можна змінювати вигляд таблиці. **Конструктор** містить параметри стилів таблиці, олівець та ґумку для накреслення таблиці. Такий спосіб зручний для створення складних таблиць.

Макет містить засоби редагування таблиці (розташування на сторінці, обтікання текстом, встановлення меж та заливки), роботи зі стовпцями та рядками (об'єднання, розділення, розмір комірок або автодобір за змістом, напрямок на вирівнювання тексту, сортування тексту за алфавітом або сортування числових даних).

✓ *Набраний текст можна перетворити на таблицю, якщо виконати таке: виокремити ту частину тексту, яку передбачається перетворити на таблицю → відкрити меню* ***Вставлення*** *→ перейти до групи* **Таблиця** *→ в меню вибрати опцію* ***Перетворити на таблицю.***

Після цього в діалоговому вікні вказати, які символи є розділювачами комірок (знак абзацу, знак табуляції, крапка з комою або інший).

✓ *У деяких випадках текст документа необхідно розмістити в декілька колонок. Для цього слід виконати такі дії: виокремити необхідні абзаци документа → в меню* ***Розмітка*** *сторінки клацнути* ***Стовпці*** *→ у списку, що відкриється, вказати кількість стовпців.*

Якщо потрібно змінити параметри колонок тексту, у цьому ж списку обрати позицію **Додаткові стовпці.** У діалоговому вікні, що відкриється, можна вибрати одну з п'яти стандартних конфігурацій стовпців, задати кількість стовпців, їх ширину і відстань між ними.

✓ *Уведення стандартних математичних формул або побудову власних формул можна здійснити за допомогою бібліотеки математичних символів з меню* ***Вставка*** *в групі* ***Символи.***

Для використання найбільш уживаних формул потрібно клацнути стрілку поряд із написом Формула й вибрати потрібну формулу. У документі з'явиться поле введення формули – блок "Місце для формули" і відповідне меню. Для створення формули, відсутньої серед найбільш уживаних, слід перейти на пункт **Формула (Вставити нову формулу)**, після чого відкриється меню **Конструктор (Робота з формулами)** (див рис. 4.7). Для редагування формули досить двічі клацнути її в тексті документа, відкриється стрічкове меню Конструктор (Робота з формулами), за допомогою якого можна внести зміни в формулу.

Рис. 4.7. Уведення формули

Для зміни типу, розміру, кольору та інших параметрів символів формули у MS Word 2007–2010 можна використовувати засоби форматування та редагування текстового редактора.

У режимі **Розмітка сторінки** ми не бачимо символів: пробіл, перенесення, варіант перенесення, розрив рядка, кінець абзацу та інших, коди яких містяться на початку кодової таблиці ASCII.

Рис. 4.8. Група Абзац

*Щоб побачити всі приховані символи, слід здійснити налаштування: в меню **Основне** у групі **Абзац** натиснути **Показати всі знаки** (рис. 4.8).*

Перевіряємо себе

1. Що називають **Списком**?
2. Які кнопки стрічкового меню **Основне** або опції контекстного меню використовуються для швидкого нумерування або маркірування?
3. Що потрібно зробити для розташування таблиці у тексті?
4. Коли з'являється меню **Табличні знаряддя**?
5. Як перетворити готовий текст на колонки?
6. Як перетворити частину тексту на таблицю?
7. Назвіть три різні способи розташування таблиці на сторінці. Як їх застосувати до вже створеної й заповненої таблиці?

Виконуємо

1. Створіть документ "Що я знаю про комп'ютер?"
2. Перетворіть частину тексту на таблицю.
3. Частину тексту документа розмістіть у три колонки.
4. Частину тексту документа розмістіть у дві колонки.
5. Одну з колонок перетворіть на список.
6. Виконайте пошук у Інтернеті матеріалів за запитом "Створення маркірованого або нумерованого списку – Word". Знайдений текст скопіюйте в новий документ. Виконайте його форматування (Основний текст – шрифт Times New Roman, 14, чорний, заголовки не форматувати). Збережіть текст у файлі для наступного редагування.

4.3. Створення та редагування графічних об'єктів у текстовому документі

Текстовий процесор Word може використовувати графічні файли, створені різними прикладними програмами. Крім стандартного набору картинок, які можна вставити в документ, можна створювати власні графічні об'єкти: рисунки, окремі лінії, фігури тощо.

Створення графічного об'єкта розпочинається з меню **Вставлення,** підменю **Фігури,** де представлені кнопки для графічних об'єктів: ліній, стрілок, просторових об'єктів тощо. Після вставлення об'єкт можна залити кольором або візерунком, змінити колір і тип ліній, збільшити або зменшити, перемістити, повернути або дзеркально відобразити об'єкти.

✓ *Можна* ***згрупувати*** *кілька об'єктів, щоб утворити новий і задати для нього режими обтікання текстом та інші властивості. Для цього потрібно виокремити кілька об'єктів, утримуючи затисненою клавішу Ctrl, клацнути правою кнопкою миші виділені об'єкти і вибрати в контекстному меню команду* ***Групування****, а потім* ***Групувати****.*

✓ *Імпорт графічних об'єктів здійснюється у двох варіантах: у вигляді рисунка з файла або частини зображення, збереженого як фрагмент у іншому файлі.*

✓ *Щоб вставити рисунок із файла в документ, треба виконати таке: установити курсор в місце документа, куди потрібно помістити рисунок → вибрати меню* ***Вставлення*** *→ натиснути* ***Рисунок*** *→ вибрати папку з потрібним рисунком → двічі клацнути лівою кнопкою миші на імені файла в списку чи ввести ім'я файла в текстовому полі* ***Ім'я файла****.*

Розміри рисунка, імпортованого в документ, можна змінити після його вставлення в документ. При зміні розміру рисунка можна зберігати його пропорції або, навпаки, змінити їх, збільшуючи або зменшуючи тільки висоту або ширину. Перш ніж працювати з рисунком у документі, його слід актуалізувати. Для цього треба клацнути на рисунку лівою кнопкою миші. Навколо виділеної картинки з'являться вісім маленьких чорних квадратів, що називаються **маркерами розмірів.**

✓ *Для того щоб змінити розміри рисунка, потрібно виокремити (актуалізувати) його → встановити курсор миші на одному з маркерів розмірів → натиснути ліву кнопку миші й перетягнути маркер розміру.*

Перевіряємо себе

1. Що називають графічним об'єктом?
2. Для чого використовується меню **Фігури**?

3. Назвіть основні дії створення власного графічного об'єкта.

4. Як виконати редагування графічних об'єктів?

5. Як виконати додавання графічного об'єкта з зовнішнього файла?

6. Як можна змінити розміри рисунка, імпортованого в документ?

Виконуємо

1. З'ясуйте, які засоби меню **Вставлення** призначені для вставлення рисунків та фігур, які дії можна виконати над уже вставленим із файла рисунком.

2. Вставте у текстовий документ рисунок, який міститься в файлі.

3. Створіть власноруч графічний об'єкт, який складається з трьох об'єктів.

4. Згрупуйте створені об'єкти так, щоб Word розглядав їх як одне ціле. Які властивості створеного об'єкта доступні для редагування?

4.4. Стильове оформлення абзаців. Структура об'єкта

Під час підготовки великих за обсягом документів виникає низка завдань технічного характеру (форматування тексту, створення змісту, літературного покажчика тощо), на розв'язання яких доводиться витрачати багато часу.

Для швидкого оформлення абзаців рекомендується використовувати стилі тексту.

Стиль тексту – це іменований набір значень властивостей фрагмента тексту як виокремленого об'єкта.

*Властивість елементів документа (**Рівень контуру**) використовується для автоматичної побудови його структури.*

Текстовий процесор Word має певний набір стилів, який називають **бібліотекою стилів**.

Переглянути стилі, наявні в бібліотеці, можна на вкладці Основне → група **Стилі** (рис. 4.9). Розрізняють такі групи стилів:

– стиль символів задає формат символів (шрифт, розмір символів, накреслення, ефекти, колір тощо);

– стиль абзацу задає формат абзацу (спосіб вирівнювання тексту, позиції табуляції, міжрядковий інтервал, інтервал перед і після, може містити формат символів тощо);

– стиль таблиці задає формат таблиці (вигляд меж, заливки, вирівнювання тексту, шрифти тощо);

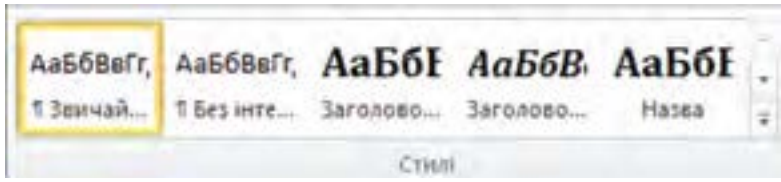


Рис. 4.9. Група **Стилі**

– стиль списку задає формат списків (спосіб вирівнювання, знаки нумерації або маркери, шрифти).

✓ *Використання стилів дає змогу однією дією змінити значення кількох властивостей тексту.* За збереження документа з ним автоматично зберігаються й застосовані стилі, тобто за подальших відкривань документа навіть на інших комп'ютерах вигляд документа не змінюватиметься.

Стилі можна вилучати, перейменовувати, модифікувати. Зауважмо, що досить внести зміни до одного з використаних у документі стилів – весь документ буде автоматично змінено.

Для застосування стилю символу, таблиці, списку потрібно виокремити цей текстовий об'єкт, а для застосування стилю абзацу досить встановити курсор у потрібний абзац тексту, переглянути список запропонованих експрес-стилів і вибрати потрібний.

Поряд із назвою стилів є значок, який показує призначення цього стилю (стиль абзацу, стиль символів, стиль таблиці, стиль списків, пов'язаний стиль абзац і символ).

✓ *Для того щоб побачити, як відображатиметься текст після застосування до нього конкретного стилю, затримайте курсор на кнопці з зображенням цього стилю.*

Бібліотеку стилів можна доповнювати власними стилями, створюючи їх на основі наявних. Для цього в меню **Основне** слід клацнути групу **Стилі**, вибрати кнопку із зображенням стрілки, яка викличе діалогове вікно **Стилі** (рис. 4.10). У вікні, яке відкриється, клацнути **Створити стиль**, з'явиться діалогове вікно **Створення стилю** за допомогою форматування (рис. 4.11).

У відповідних полях діалогового вікна **Створення стилю за допомогою форматування** потрібно ввести назву нового стилю і вибрати

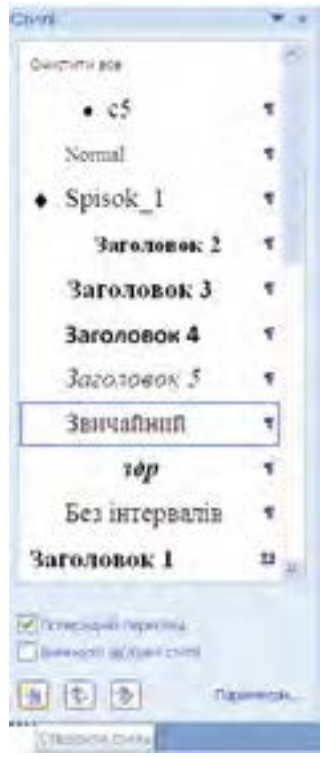


Рис. 4.10. Діалогове вікно **Стилі**

його тип. У полі **Оснований на стилі** вибрати наявний стиль, на основі якого буде створено новий. Для встановлення формату нового стилю потрібно розкрити список **Формат** та вибрати об'єкти, властивості яких потрібно змінити.

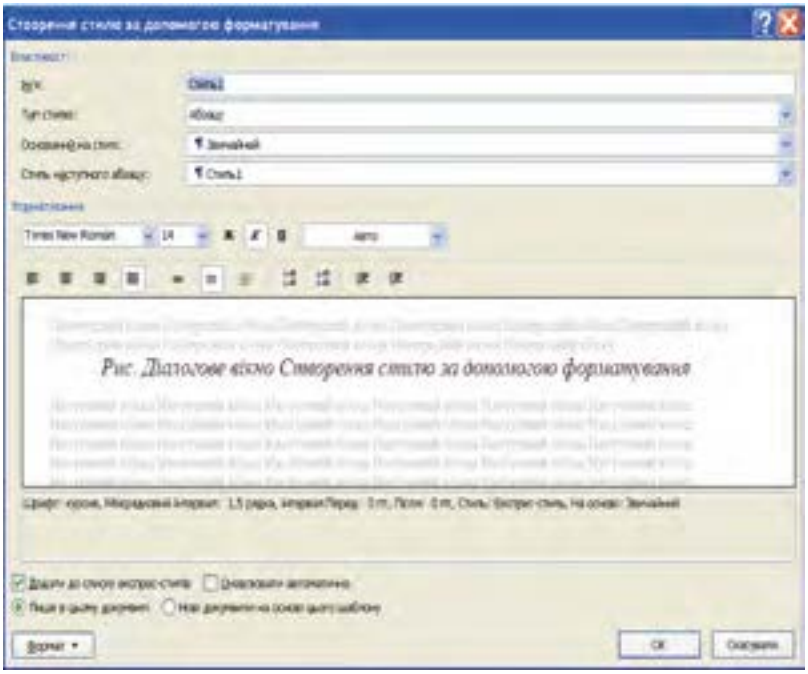


Рис. 4.11. Діалогове вікно **Створення стилю**

4.5. Розділи. Колонтитули

Навчаючись працювати в середовищі текстового процесора, ви зрозуміли, що будь-який текстовий документ складається з відповідних частин – розділів, які своєю чергою можуть бути поділені на менші частини – підрозділи або параграфи. Таких поділів може бути безліч. Найпростіший документ складається з розділів.

Розділом називаємо частину документа, об'єднану за функціональним змістом.

Розуміння структури документа дає можливість грамотно його оформити й переформатувати. Оскільки одні розділи є частинами інших (розділ складається з параграфів, параграфи складаються з пунктів), то розділи розрізняють за рівнями. Розділ, що входить до складу іншого, є нижчим від нього на рівень. Узагальнюючи, можна сказати, що весь текстовий документ – це розділ 1-го рівня, розділи, з яких він складається, – розділи 2-го рівня і так далі.

Назви розділів прийнято називати заголовками відповідно до їх рівня: **Заголовок 1 рівня**, **Заголовок 2 рівня** і так далі.

Побудова структури спрощує подальшу роботу з документом – від створення змісту до підготовки його до будь-якої публікації.

Зміст документа складається із назв розділів різного рівня – заголовків.

Структура текстового документа формується не тільки за розділами. Кожен окремий абзац можна класифікувати за функціональним змістом, який він несе. Розрізняють три типи функціональних одиниць, або структурних елементів, текстового документа: **розділи**, **абзацні** та **символьні** структурні елементи.

Символьними структурними елементами називають речення, які не є абзацами, і навіть окремі слова.

Виокремлення й іменування структурних елементів тексту забезпечує структуризацію документа. Для правильної структуризації всі структурні елементи тексту мають бути чітко визначені. Кожен із них має власне форматування (стиль), яке описується лише один раз.

Формування структури допомагає легше орієнтуватись у великому документі. У структурованому документі можна досить швидко пересуватися між розділами. Це називається **навігацією структурою**.

Для позначення структурних елементів у списку стилів є спеціальні стилі, які мають назву структурних елементів.

Наприклад, стилі:

– Заголовок 1 – позначення заголовків до основних розділів;
– Заголовок 2 – позначення заголовків другого рівня;
– Заголовок 3 – позначення заголовків третього рівня;
– Звичайний – позначення основного тексту документа.

Правильне структурування документів важливе не тільки для користувачів, які безпосередньо працюють з ними, а й для людей, що користуються пошуковими системами та системами автоматичного обліку електронної документації.

Якщо структурований документ буде розташований у мережі Інтернет, то пошуковий робот може точніше його класифікувати і внести до каталогу обліку ресурсів коректну інформацію.

Важливим елементом структури документа є колонтитул – текст і/або зображення, розташоване внизу або вгорі кожної сторінки. Залежно від місця розташування (на верхньому або на нижньому полі сторінки) **колонтитули** бувають верхніми та нижніми. Верхня або нижня частина сторінки називається зоною колонтитула і використовується для введення однакового тексту (назви книжки, імені автора, номера сторінки,

логотипа тощо) на кожній сторінці (або парних/непарних сторінках) документа.

Допускається створювати унікальний колонтитул або взагалі його не створювати для першої сторінки документа або певного розділу. Можна також створювати різні колонтитули для парних і непарних сторінок розділів або документа.

✓ *Для створення колонтитула потрібно двічі натиснути ліву кнопку миші у верхньому або нижньому полі сторінки, тобто в місці, де має бути розташований колонтитул → набрати текст колонтитула або вставити рисунок → для повернення до роботи з основним текстом документа двічі клацнути лівою кнопкою миші на робочому полі документа.*

Поки на екрані видно зону колонтитула, основний текст документа редагувати неможливо. Колонтитул може містити номер сторінки.

✓ *Вставлення номерів сторінок здійснюється зі стрічкового меню: клацнути вкладку* ***Вставлення*** *→ у групі* ***Колонтитули****; клацнути команду (кнопку)* ***Номер сторінки*** *→ у списку, що відкрився, обрати вигляд номера сторінки.*

Із цієї самої групи можна також вибрати готові колонтитули (кнопки **Верхній колонтитул** і **Нижній колонтитул**) і виконати їх налагодження.

✓ *Для вилучення колонтитулів потрібно подвійним клацанням лівої кнопки миші перейти в область колонтитула → виконати видалення тексту колонтитула.*

Перевіряємо себе

1. Що ви розумієте під “структурою” текстового документа?
2. Що називають розділом?
3. Як прийнято називати назви розділів?
4. Із чого складається зміст документа?
5.  Які розрізняють типи функціональних одиниць або структурних елементів текстового документа?
6. Яку частину тексту називають “символьними структурними елементами”?
7. Який засіб називається навігацією структурою?
8. Як називають текст та/або зображення, що розташовується внизу або вгорі кожної сторінки документа?
9. Як можна створити колонтитул?
10. Які дії потрібно виконати для видалення колонтитулів?

Виконуємо

1. Відкрийте раніше створений документ “Що я знаю про комп’ютер?” або інший, створений для роботи зі списками (п. 4.3).
2. Виокремте частини, які вважатимуться назвами розділів різних рівнів.
3. Позначте їх відповідними стилями з меню Стилі.
4. Перегляньте структуру документа, використовуючи **Область навігації**.
5. Вставте нижній колонтитул – своє ім’я та прізвище.
6. Вставте верхній колонтитул – номер школи та населений пункт.
7. Збережіть документ у Документ Word (*.doc).

4.6. Посилання. Автоматизоване створення змісту і покажчиків

Кожний багатосторінковий документ обов’язково має спеціальний список, який містить назви структурних елементів – зміст.

Зміст документа – це впорядкований перелік заголовків та підзаголовків. Для автоматичного створення змісту документа слід попередньо застосувати вбудовані стилі під час форматування до заголовків усіх рівнів.

Для автоматизованого створення змісту слід у меню **Посилання** клацнути **Зміст** й вибрати потрібний стиль змісту, після чого можна вказати потрібні значення властивостей (кількість рівнів, заповнювач тощо) (рис. 4.12).

Рис. 4.12. Вбудовані стилі змісту

Якщо заголовки в тексті не були визначені за рівнями, то слід спочатку здійснити структуризацію в меню **Основне** в групі **Стилі**. Інакше зміст не буде побудовано.

Предметний покажчик – це список слів (термінів), використаних у текстовому документі. Такий список, для зручності пошуку слів, вбудовується зазвичай у алфавітному порядку в кінці чи на початку документа. Покажчик містить номери сторінок, на яких ці слова згадуються в тексті документа.

*Для побудови **Предметного покажчика** потрібно у стрічковому меню **Посилання** клацнути **Покажчик** → натиснути **Позначити** та, не закриваючи це вікно, почергово позначити всі ключові й додаткові фрагменти тексту → по закінченню натиснути **Закрити** (з'явиться замість **Скасувати**).*

Усі позначені елементи **Предметного покажчика** вставляються до документа як елементи предметного покажчика, вони оформляються як приховані знаки. Для того щоб їх побачити/приховати, слід натиснути кнопку **Недруковані знаки ¶** в меню **Основне**.

Посилання – це об'єкт у документі, за допомогою якого можна перейти до іншого фрагмента цього ж документа або до іншого документа.

Для створення посилання слід виокремити (актуалізувати) місце, з якого буде можливим перехід, натисненням правої клавіші миші викликати меню **Гіперпосилання** і, керуючись підказками, вказати об'єкт переходу. Ним може бути, зокрема, позначений при створенні предметного покажчика об'єкт документа.

Особливим варіантом посилання є перехресне посилання, яке вставляється в документ як гіперпосилання, але з підменю **Перехресне посилання**. Об'єктом переходу для нього можуть бути частини документа, такі, як елементи списків, рисунки, фрагменти тексту тощо. При реструкторизації документа встановлені зв'язки зберігаються.

Перевіряємо себе

1. Що таке **Посилання**?
2. Що називають **Змістом документа**?
3. Які дії слід виконати для автоматичного створення змісту документа?
4. Що називають **Предметним покажчиком**?
5. Послідовність яких дій потрібно виконати, щоб побудувати **Предметний покажчик**?
6. Як створити гіперпосилання на місце в документі?

Виконуємо

1. Відкрийте документ “Що я знаю про комп’ютер?”.

2. У меню **Посилання** клацніть **Зміст** і виберіть стиль змісту через засіб **Вставити таблицю змісту.**

3. У вікні, що відкрилося, виберіть параметри змісту.

4. Перегляньте структуру документа, використовуючи **Область навігації.**

5. Наприкінці документа, користуючись вказівками, які є в параграфі, створіть предметний покажчик із 10 слів.

6. Збережіть документ.

4.7. Опрацювання складного текстового документа

Документ, який поділено на кілька розділів або параграфів, можна вважати складним.

Існує багато прийомів, використання яких спростить його форматування та редагування. Наприклад, для переходу до того чи іншого заголовка в документі досить встановити курсор миші на відповідний заголовок у його змісті й натиснути клавішу Ctrl на клавіатурі. Після цього клацнути лівою кнопкою миші заголовок – виконається перехід на відповідний заголовок у документі.

Якщо в текст документа внесено зміни (вставка/вилучення сторінок, зміна заголовків (їх кількість та текст)), їх досить легко відобразити в змісті.

1. Внести зміни до заголовків та (чи) зміни кількості сторінок, тексту, назв тощо.

2. Перейти на поле змісту й клацнути правою кнопкою миші – відкриється контекстне меню.

3. У контекстному меню клацнути **Оновити зміст** та обрати **Оновити цілком** – ОК.

У заголовку **Область завдань** є кнопки **Вставити все** та **Очистити все,** а також список скопійованих фрагментів. Усі вони можуть бути вставлені в будь-який відкритий на цей час документ.

Для вставлення всіх фрагментів у актуальний документ потрібно встановити курсор миші (клацнути) у відповідне місце документа й натиснути кнопку **Вставити все.**

У Microsoft Word є кілька засобів, які дозволяють швидко переглянути документ та внести до нього зміни. Один з них:

1. Клацнути меню **Вигляд.**

2. Встановити позначку в полі **Область навігації** – відкриється вікно **Навігація.**

У цьому вікні доступні три режими:

– Перегляд заголовків у документі.
– Перегляд сторінок документів.
– Перегляд результатів поточного пошуку.

Після встановлення позначки в полі **Область навігації** й обрання режиму "Перегляд заголовків у документі" вздовж лівого поля робочої частини вікна документа відкриється вертикальна область, у якій відображатиметься структура документа (заголовки, підзаголовки). Для перегляду тексту документа досить виконати перехід, клацаючи відповідні заголовки.

У пакеті Microsoft Office є можливість почергового копіювання кількох фрагментів документа (в останніх версіях – 24) для їх подальшого використання в інших місцях документа або в інших документах.

Це виконується одночасним розміщенням у **Буфері обміну** кількох фрагментів тексту (і не тільки тексту). **Буфер обміну** Microsoft Office є спільним для програмних продуктів Microsoft. Для роботи ним в додатках пропонується спеціальна **Область завдань,** яка теж має назву **Буфер обміну**.

Для обміну інформацією між кількома документами або переміщенням чи копіюванням фрагментів у межах одного документа в меню **Основне** клацнути **Буфер обміну** – зліва від тестового поля (робочого) відкриється область завдань **Буфер обміну** (рис. 4.13).

Буфер обміну заповнюється скопійованими фрагментами, які показані умовними значками. Значки відповідають різним типам даних фрагментів.

Для вставлення окремого фрагмента потрібно встановити курсор миші (клацнути) у відповідне місце документа, помістити покажчик миші на потрібний фрагмент і клацнути на ньому. Він буде вставлений у документ у поточну позицію курсора.

Для вилучення фрагмента з **Буфера обміну** потрібно клацнути на стрілці поряд із фрагментом. Після цього розкриється список команд, які можна застосувати до фрагментів, із них обрати й натиснути **Видалити**.

Однією з особливостей застосування засобів інформаційних технологій є те, що людина вивільняється від виконання дій, які можна автоматизувати. Така автоматизація відбувається через створення програм.

Рис. 4.13. Область завдань **Буфер обміну**

Макрос – це команди та інструкції, згруповані в послідовність, яка запускається на виконання однією командою та призначена для автоматичного виконання певного завдання.

Макроси застосовують для прискорення часто використовуваних операцій редагування або форматування; для об'єднання кількох команд.

Макрос можна описати мовою програмування (для застосунків Microsoft Office – мовою Visual Basic) або записом послідовності команд, які подаються користувачем. У останньому випадку ці команди описуються мовою Visual Basic автоматично. Макроси записують застосунком Microsoft Office до файла, в якому зберігається електронний документ, або до файла, в якому зберігається шаблон, або до файла загальних налаштувань.

Наприклад, якщо ви перенесли з Веб-сторінки в документ Microsoft Word певний текст, він буде відформатований певним чином – кожний рядок абзацу закінчується командою переходу на новий рядок. Для того щоб надати документу вигляд, прийнятий у текстовому документі, ці команди треба вилучити. Вам доведеться виконати досить велику кількість одноманітних дій. Якщо таких рядків три–п'ять, то дії можна виконати і без використання макросів. Але якщо їх набагато більше, то їх можна об'єднати в одне натиснення комбінації клавіш.

Макрос спочатку записують, а потім використовують. Макросу ставлять у відповідність комбінацію клавіш на клавіатурі або кнопку на панелі швидкого доступу. Щоб виконати макрос, потрібно натиснути на відповідну комбінацію клавіш (або створену у вигляді піктограми кнопку на екрані).

Для роботи з макросами спершу потрібно відобразити на стрічці вкладку **Розробник**. Для цього слід виконати послідовність команд:

1) у меню **Файл** в пункті натиснути **Параметри**;

2) у діалоговому вікні Параметри вибрати команду **Настроїти стрічку**;

3) у полі **Настроїти стрічку** установити прапорець поряд із пунктом **Розробник**.

Надалі дозвіл на відображення вкладки **Розробник** зберігатиметься.

Для запису макросу:

1) на вкладці **Розробник** у групі **Код** натиснути **Записати макрос**;

2) у діалоговому вікні **Запис макросу** дати назву макросу в полі **Ім'я макросу**;

3) у полі **Зберегти макрос у** вказати документ, у якому потрібно зберегти макрос;

4) у полі **Опис** вказати призначення цього макросу;

5) щоб поставити у відповідність макросу кнопку на панелі швидкого доступу, вибрати пункт **Кнопці – ОК**.

6) Після зазначених дій запис розпочався: виконати дії, які має містити макрос, і натиснути кнопку **Зупинити макрос**. Зверніть увагу на маркер макросу, який з'явиться на панелі швидкого доступу.

*Під час запису макросу можна вибирати команди та параметри за допомогою миші, але виділяти текст нею не можна. Для цього слід використовувати клавіатуру (утримуючи натиснутою клавішу **Shift**, перемістити курсор уздовж тексту, який потрібно виділити).*

Перевіряємо себе

1. Який документ можна вважати складним?
2. Які прийоми використовують, щоб спростити форматування та редагування?
3. Якщо в текст документа внесено зміни (вставлення/вилучення сторінок, зміна заголовків (їх кількість і текст)), то яким чином можна відобразити їх у змісті?
4. Які засоби можна використати для швидкого перегляду документа та внесення змін у нього?
5. Назвіть засіб, який використовується для обміну інформацією між кількома документами або переміщення чи копіювання фрагментів у межах одного документа.
6. Як можна вставити одночасно кілька скопійованих фрагментів тексту?
7. Як вставити окремий фрагмент тексту?
8. Що слід зробити для вилучення фрагмента з **Буфера обміну**?

Виконуємо

1. Відкрийте документ “Що я знаю про комп’ютер?”
2. Додайте розділ “Як я користуюсь Інтернетом?” Для назви розділу виберіть стиль Заголовок 1.
3. Виокремте кілька частин документа з назвами, яким дайте імена Заголовок 2 та Заголовок 4.
4. Використайте **Буфер обміну** для переміщення частин тексту: Заголовок 3 на місце Заголовок 2.
5. Створіть новий документ. Скопіюйте кілька фрагментів у документі “Що я знаю про комп’ютер?” і вставте одночасно в створений документ.
6. Очистіть **Буфер обміну**.
7. Збережіть обидва документи.

Практична робота № 5

Тема:	Створення текстового документа, що містить об’єкти різних типів
Мета:	Набути практичних навичок створення текстових документів з упровадженими та зв’язаними об’єктами

Завдання

1. Наберіть текст відповідно до обраної теми.
2. Продумайте та порадьтесь із учителем щодо необхідності та місця вставлення зображень у текст.
3. Продумайте та порадьтесь із учителем щодо необхідності та місця вставлення таблиці в текст.
4. Виокремте в документі фрагмент тексту, який можна відформатувати в дві колонки.

5. Відформатуйте зазначений фрагмент у дві колонки. Додайте до цього фрагмента заголовок.

6. Збережіть створений документ.

Рекомендовані теми для Практичної роботи.

1. Предмети, які вивчають учні восьмого класу.
2. Що я знаю про Україну.
3. Найбільші річки України.
4. Що я знаю про Сполучені Штати Америки.
5. Мій особистий розклад на тиждень – мої улюблені справи.
6. Місто, де я народився (народилася).
7. Тварини, які живуть у лісах України.

Практична робота № 6		
	Тема:	Використання стилів для оформлення текстових документів
	Мета:	Набути практичних навичок створення складних текстових документів

1. У тексті документа, який створено раніше, виокремте заголовки першого та другого рівнів.

2. Застосуйте до них відповідні стилі (використовуйте при цьому список **Стилів**).

3. До основного тексту документа застосуйте стиль **Звичайний**.

4. Отримайте від учителя вимоги до оформлення документа (передбачається застосування засобів **Абзац, Шрифт** тощо).

5. Перегляньте структуру документа.

6. Для пояснення особливих для тексту документа термінів, власних імен, назв тощо створіть **Предметний покажчик**.

7. Збережіть документ.

Примітка. Додаткові вимоги до оформлення тексту надає вчитель.

Практична робота № 7		
	Тема:	Структура документа. Автоматизоване створення змісту й покажчиків. Макроси
	Мета:	Набути практичних навичок створення складних текстових документів

1. У тексті документа, який створено в школі, виокремте, додатково до заголовків першого та другого рівнів, заголовки третього і четвертого рівнів.

2. Застосуйте до них відповідні стилі (використовуйте при цьому список **Стилів**).

3. Перегляньте структуру документа.

4. На останній сторінці документа вставте його зміст.

5. Додайте до тексту рисунки з підрисунковими підписами. Підписам під рисунками надайте такий рівень, який би не відображався (або відображався) у Змісті.

6. Оновіть Зміст.

7. Збережіть документ.

СЛОВНИЧОК

Бланк – аркуш паперу з розміщеними на ньому постійними реквізитами.

Веб-сторінка (*.htm; *.html) – формат Веб-сторінки; його використовують у разі розміщення документа в Інтернеті.

Гіперпосилання – виділений кольором, підкреслений текст або графічний об'єкт, під час активізації якого (при натисканні на кнопку миші на ньому) виконується перехід до файла, фрагмента файла або Веб-сторінки в мережі Інтернет, який описано в посиланні.

Гіпертекст – електронний документ (переважно текстовий), який містить гіперпосилання.

Документ Word (*.doc) – власний формат текстового процесора Word, у цьому форматі документи зберігаються замовчуванням.

Звичайний текст (*.txt) – найбільш універсальний формат, що зберігає текст без форматування; у текст вставляються тільки символи кінця абзацу.

Зміст документа – упорядкований перелік заголовків та підзаголовків.

Колонтитул – текст та/або зображення, що розташовується внизу або вгорі сторінки документа. Залежно від місця розташування (на верхньому або на нижньому полі сторінки) колонтитули бувають верхніми та нижніми

Макрос – команди та інструкції, згруповані в послідовність, яка запускається на виконання однією командою.

Розділ у тексті документа – частина документа, яка містить певний функціональний зміст.

Стиль абзацу – задає формат абзацу (спосіб вирівнювання тексту, позиції табуляції, міжрядковий інтервал, інтервал перед і після, може містити формат символів тощо).

Стиль символів – задає формат символів (шрифт, розмір символів, накреслення, ефекти, колір тощо).

Стиль списку – задає формат списків (спосіб вирівнювання, знаки нумерації або маркери, шрифти тощо.

Стиль таблиці – задає формат таблиці (вигляд меж, заливки, вирівнювання тексту, шрифти тощо).

Стиль тексту – набір значень властивостей тексту (у тому числі рівень контуру, що використовується для автоматичної побудови документа), який має певну назву.

Формуляр – сукупність реквізитів.

Шаблон документа (*.dot) – формат шаблону, на якому можуть базуватись інші документи.

Шаблон електронного документа – аналогічну роль відіграє електронна форма, яка містить постійні реквізити й структуру документа і в термінології, поширеній у більшості офісних систем (зокрема й у Microsoft Office).

РОЗДІЛ 5. КОМП'ЮТЕРНА ГРАФІКА. ВЕКТОРНИЙ ГРАФІЧНИЙ РЕДАКТОР

Для зберігання й відтворення зображення здійснюється його оцифровування, яке полягає в поділі зображення на прості об'єкти і визначення властивостей цих об'єктів. Найпростішим об'єктом, із сукупності яких можна утворити зображення, є піксель. Властивостями цього об'єкта є колір і яскравість. Зображення, утворене з сукупності пікселів, називається растровим зображенням. Відомості про значення властивостей пікселів зберігаються в цифровій формі у файлах певних типів: .bmp, .jpg,.png,.gif,.pcx, .tif та інших. Зображення в растровому поданні опрацьовується програмними засобами, які називаються редакторами растрової графіки (Paint, Adobe Photoshop, GIMP та інші). Растрове зображення – об'єкт, властивостями якого є роздільна здатність зображення (у пікселях на сантиметр зображення), глибина кольору (кількість бітів, якими закодовано кольори пікселів) і сукупність значень властивостей усіх його пікселів.

Комп'ютерна графіка; векторні зображення; властивості векторних і растрових зображень. Формати файлів векторних зображень. Формати файлів векторної графіки та графічні редактори векторних зображень. Способи створення зображень із базових графічних примітивів; обертання, відображення і масштабування об'єктів; зафарбовування об'єктів із використанням однорідних, градієнтних, візерункових і текстурних заливок; додавання до графічних зображень тексту та його форматування; групування і вирівнювання об'єктів; використання шарів для створення зображень; інструменти, призначені для креслення ліній, стрілок, основних геометричних фігур; лінійки, сітка.

5.1. Основні поняття комп'ютерної графіки

Растрові та векторні зображення; векторна графіка як спосіб описання й подання зображень; основні поняття векторної графіки; редактори векторної графіки; операції над об'єктами векторних зображень.

Комп'ютерна графіка з'явилась досить давно – вже у 1960-х роках існували програми, призначені для роботи із зображеннями.

✓ *Нині прийнято користуватися термінами "комп'ютерна графіка" і "комп'ютерна анімація".*

Поняття "комп'ютерна графіка" об'єднує всі види робіт із нерухомими зображеннями, "комп'ютерна анімація" вказує на те, що зображення змінюються.

За структурою зображення можуть бути **растровими** і **векторними**.

Растрові зображення – комп'ютерні зображення, на яких рисунок подано як сукупність точок (пікселів – найменших елементів зображення). Під час редагування растрових об'єктів користувач змінює колір точок – пікселів, навіть у тому випадку, якщо використовує інструменти "Лінія", "Дуга", "Прямокутник", "Напис" тощо. А отже, після того як деякий об'єкт створено, редагувати його не можна інакше, як сукупність точок. Це ми бачимо, використовуючи растрові редактори Paint, Paint.NET та інші.

Векторні зображення складаються з груп пікселів, які утворюють окремі зображення – лінії, геометричні фігури тощо.

Кожна з фігур є об'єктом, який має певні властивості: колір, товщину ліній і їх накреслення, орієнтацію відносно умовного аркуша паперу, колір внутрішньої частини та спосіб її зафарбовування тощо.

Ці властивості можна редагувати, змінювати й після того, як об'єкт створено. Тобто об'єкти векторного рисунка залишаються об'єктами і після розміщення їх на рисунку. Таким чином, створення зображення векторним способом подібне на створення аплікації з готових елементів, які можна деформувати, розфарбовувати, робити прозорими тощо. Основними елементами зображення є відрізок прямої, прямокутник, коло (еліпс), точка.

*Ці елементи векторного рисунка називаються **графічними примітивами**.*

Як і растрові, векторні зображення зберігаються в файлах різних типів. Із деякими ви вже працювали, використовуючи готові рисунки з бібліотеки Microsoft Office (рис. 5.1).

Найчастіше в цих бібліотеках використовуються рисунки формату *.wmf. Засобами вбудованого в усі додатки Microsoft Office графічного векторного редактора їх можна розібрати на складові, які є графічними примітивами. Ця операція називається **декомпозицією**, або **розгруповуванням** (рис. 5.2).

Для роботи з векторною графікою використовують наведені нижче програмне забезпечення і формати збереження.

Формат *.svg (англ. Scalable Vector Graphics – масштабована векторна графіка) може описувати як статичні (нерухомі), так і анімовані зображення, в тому числі й керовані користувачем. Файли цього формату насправді є текстовими. Кожний примітив, його розмір, положення описано послідовністю команд. Це зручно для систем автоматизованого проектування, використання для створення зображень, які відтворюються Інтернет-браузерами.

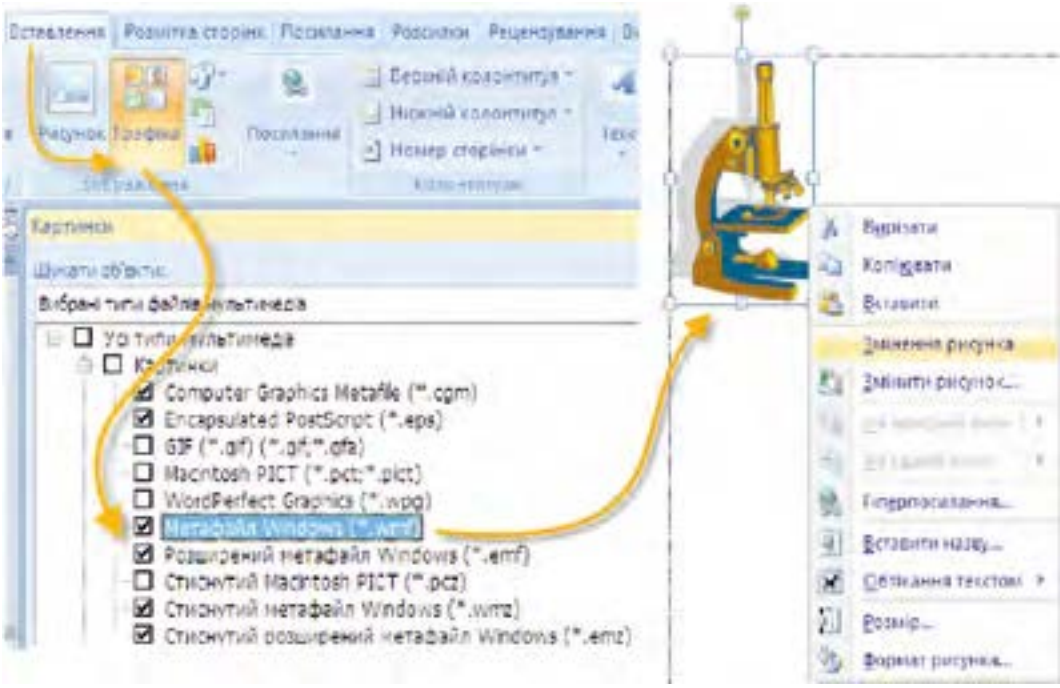


Рис. 5.1. Вставлення векторного рисунка в текстовий документ і виконання його декомпозиції (розгруповування)

Рис. 5.2. Декомпозиція векторного рисунка

Для створення високоякісних складних зображень, призначених для відтворення на великих і дуже великих (десятки метрів) плакатах, використовуються спеціалізовані редактори Adobe Illustrator і Adobe Photoshop. Застосовуються формати *.ai – (Adobe Illustrator), *.eps – (Encapsulated PostScript).

Популярний програмний комплекс CorelDraw має в своєму складі растровий і векторний графічні редактори, інші програмні засоби, призначені для захоплення зображень, у тому числі рухомих, з екрана комп'ютера, створення буклетів, виконання інших робіт. Для цього використовуються формати *.cdr, *.cmx та інші.

Векторні зображення широко застосовуються в системах автоматизованого проектування (САПР). Такі системи зараз є основним інструментом роботи сучасного інженера.

Уміст файла, в якому зберігаються відомості про векторне зображення, може бути текстом, який складається з команд, подібних до таких: “накреслити коло радіусом 56 пікселів лінією чорного кольору товщиною 5 пікселів, зафарбувати його внутрішню частину зеленим кольором”, “накреслити відрізок прямої від точки 10,36 довжиною 100 шириною 5 червоним кольором”. Подібні описи траплялися у програмуванні в системі Скретч. Ці команди, звичайно, будуть певним чином закодовані. Їх можуть виконати спеціальні програми, відтворюючи зображення.

Зображення водія в автомобілі на рис. 5.3 створене приблизно так, як спрощені рисунки в дитячих книжечках для розфарбовування.

Спрощення зображень при їх векторному поданні є обов’язковою умовою, інакше втрачаються його переваги над растровим поданням.

При векторному поданні зображень описують не властивості кожного пікселя, а властивості їх груп – об’єктів. Тому при масштабуванні зображення, тобто збільшенні або зменшенні його розмірів, програмно виконується обчислення параметрів усіх об’єктів, що утворюють зображення, додається (при потребі) деяка кількість пікселів, обчислюються їх властивості. Для векторного подання зображення властивості й положення кожного пікселя можуть бути обчислені точно, тому й об’єкт відтворюється з тією точністю, яку забезпечує засіб відтворення зображення.

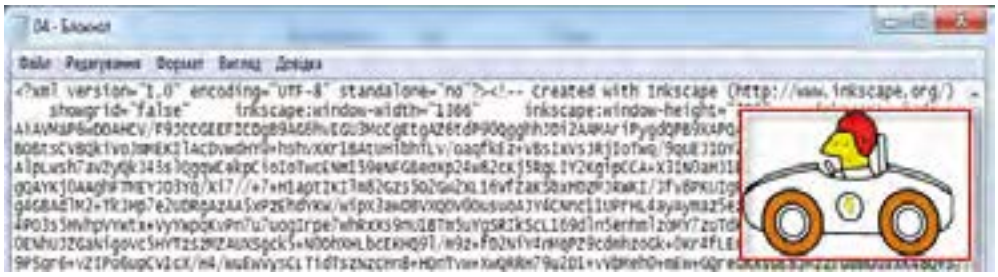


Рис. 5.3. Векторне зображення, яке зберігається в файлі з розширенням svg (створеному в графічному редакторі Inkscape), і частина цього файла, відтворена в редакторі **Блокнот** як текст

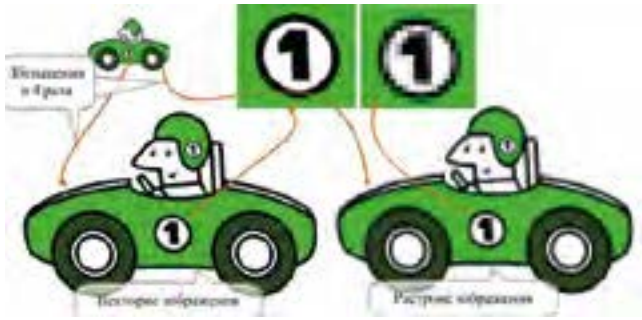


Рис. 5.4. Вплив збільшення на якість векторного і растрового зображень

На рис. 5.4 показано відмінності між растровим і векторним способами подання зображень на прикладі збільшення зображення. Як бачимо, для векторного зображення, побудованого з об'єктів, не важлива роздільна здатність первинного зображення. Форма об'єктів збільшеного зображення точно відтворює форму первинного зображення.

✓ *Растрове зображення, створене цифровим фотоапаратом, навіть із найвищою роздільною здатністю, має певні межі збільшення, при досягненні яких кількість дрібних деталей, що розрізнятимуться на ньому, не збільшуватиметься, а зображення стане нечітким.*

Для растрового зображення, в якому найменшим об'єктом є піксель, при певному збільшенні рисунка доводиться його властивості передавати декільком пікселям (у випадку, показаному на рис. 5.4, 16 пікселям).

✓ *Векторне зображення має обмежену кількість об'єктів, деталей вже від створення. Тому його збільшення до будь-яких розмірів не призводить до втрат чіткості, але й не додає відомостей для спостерігача.*

Для утворення **векторного зображення** використовуються наперед визначені фігури, кількість яких обмежена.

Зазвичай це: лінії, многокутники, стандартні елементи креслень і блок-схем (стрілки, виноски, позначення частин машин), об'ємні фігури тощо (рис. 5.5). Більшість редакторів векторної графіки мають і спеціальні інструменти, за допомогою яких створюються графічні об'єкти, подібні до намальованих художником.

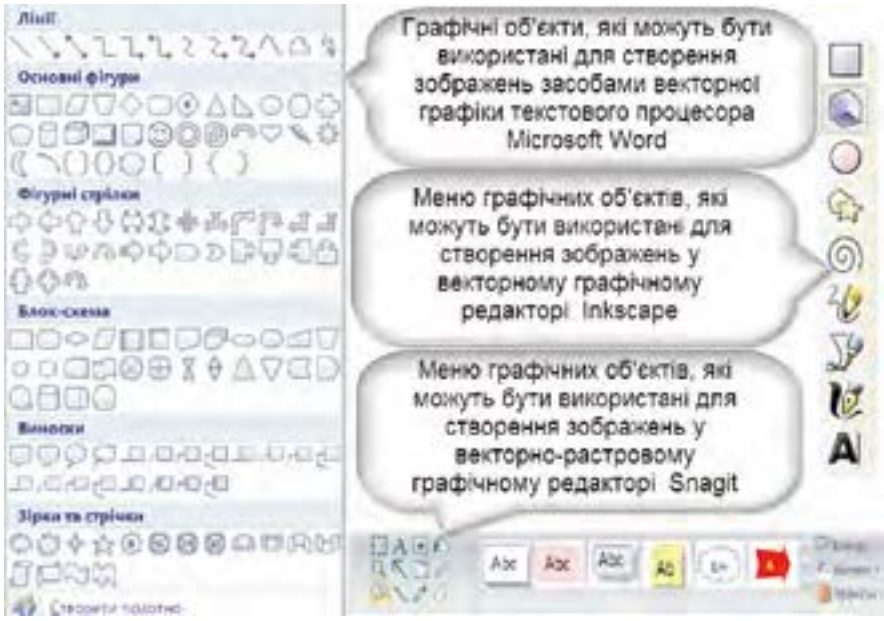


Рис. 5.5. Меню графічних об'єктів, які можуть використовуватися для створення зображень у різних векторних графічних редакторах

Для всіх редакторів векторної графіки спільним є те, що об'єкти, з яких вибудовується зображення, є простими геометричними фігурами або їх комбінаціями.

Фігури, з яких утворюється векторне зображення, мають назву **"графічні примітиви"** і є об'єктами, властивості яких після вставлення у рисунок можна змінювати.

У векторному графічному редакторі зображення створюється у вигляді рисунка, який є комбінацією графічних примітивів.

Графічні примітиви (фігури) можуть розташовуватись ніби вирізані з паперу фігури, що частково накладаються одна на одну, утворюючи ***шари.***

Шаром називають умовну площину, в якій розташовані об'єкти.

Шар разом із **примітивами** можна переміщувати – ближче до спостерігача або далі від нього. Фігури першого від спостерігача **шару**, як і в реальному житті, закривають фігури віддалених **шарів**. Якщо зробити фігури першого **шару** прозорими, крізь них можна бачити фігури інших **шарів**.

Основні властивості графічних об'єктів подано на рис. 5.6, створеному у векторному редакторі. На ньому ж можна побачити й розташування примітивів у шарах. У найближчому до нас шарі розташовано стрілки, в наступному – виноски з написами. Виноски зроблено напівпрозорими, щоб вони не дуже заважали бачити автомобіль, розташований у найдальшому від спостерігача шарі.

Рис. 5.6. Основні властивості графічних об'єктів

Кожний графічний об'єкт має точки, рухаючи які можна змінювати його форму. Ці точки називають ***вузлами.***

Об'єкти, що утворюють зображення, можуть об'єднуватись, або **групуватись**, утворюючи **групу** об'єктів.

Об'єкт, утворений після групування кількох інших, матиме всі властивості об'єктів, що входять до групи, але змінювати можна лише властивості, спільні для всіх об'єктів.

Властивості ***групи*** *об'єктів можна змінювати як властивості одного об'єкта.*

Сукупність ліній об'єктів, згрупованих для утворення зображення, інколи називають "скелетом зображення" (рис. 5.7, 5.8).

Рис. 5.7. Векторне зображення і його “скелет”, на якому відображено вузли

Рис. 5.8. Схематичне зображення глядача в кріслі (растрове зображення) і його “скелет” для створення векторного зображення

Зазвичай векторне зображення створюється як відображення реального об’єкта. Цей процес можна виконувати подумки, можна обводити олівцем на малюнку контури елементів зображення, креслити геометричні фігури, намагаючись створити з них певне зображення, а можна робити це, наносячи на растрове зображення фігури, які зберігаються у векторному редакторі як примітиви. Ці перетворення можуть виконати й деякі векторні редактори. Процес, який називається **векторизацією** зображення, виконується приблизно так, як показано на рис. 5.8.

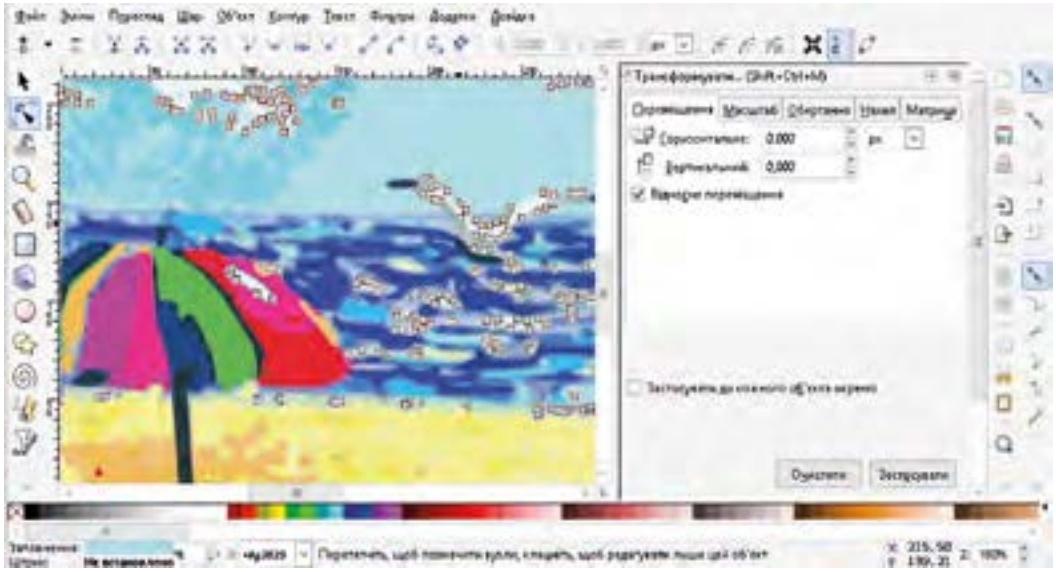


Рис. 5.9. Результат оцифровування растрового рисунка (показано вузли об’єктів одного з шарів. Редактор Inkscape)

Отже, якщо рисунок може бути побудований із простих геометричних фігур, над ним потрібно виконати такі перетворення:

1. Виокремити геометричні фігури, розподілити зображення на прості фігури та їх групи.

2. Визначити параметри кожної геометричної фігури (до якого класу фігур належить виокремлений об'єкт), її колір, прозорість.

3. Послідовно викликати необхідні примітиви й нанести їх на рисунок, зафарбувати, надати значення параметру "прозорість" і розташувати шари таким чином, щоб отримати потрібне зображення.

Векторизація може бути здійснена і програмним засобом, але для цього потрібно вказати певні передумови (рис. 5.9).

Перевіряємо себе

1. Чим різняться растровий і векторний способи подання зображень?

2. Скільки примітивів використано для побудови скелета векторного зображення на рис. 5.8?

3. Чи можна вважати групу об'єктів новим об'єктом? Якщо так, то як передаватимуться властивості кожного об'єкта групі?

4. Які властивості може мати об'єкт "лінія"?

5. Скільки шарів (щонайменше) мають зображення на рис. 5.4 і 5.6? Як це визначити?

6. У групу об'єднано коло, відрізок прямої і трикутник. Назвіть властивості, які може мати новий об'єкт. Перевірте, виконавши побудову.

7. Спостерігаючи за збільшенням цифрової світлини, помічаємо, що за певного збільшення зображення на ній стає неначе побудованим із цеглинок. Чи зможете ви отримати більше відомостей про об'єкт, зображений на світлині, ще більше збільшуючи її зображення? Обговоріть і обґрунтуйте висновок.

8. Збільшуючи зображення, отримане у векторній формі, щоразу бачимо бездоганні лінії, плавні переходи яскравостей і кольорів. Чи означає це, що й об'єкт, який зображено на векторному рисунку, має такі ж властивості? Якщо ні, то коли втрачено відомості щодо точних значень його властивостей? Чи можна їх відновити, маючи лише векторне зображення?

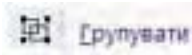
9. Які примітиви використано для побудови векторного зображення на рис. 5.8? Чи можна було використати меншу їх кількість? Запропонуйте варіанти, спробуйте побудувати зображення.

10. Чому в деяких випадках (у яких саме) не всі команди в меню груп стрічкового меню **Засоби малювання** доступні? Перевірте.

Виконуємо (використовуючи MS Word)

1. Побудуйте контури зображення сніговика, використовуючи тільки примітиви (фігури)  овали  . Застосуйте копіювання, вставлення і обертання. Для прискорення роботи користуйтесь командами Ctrl+c і Ctrl+v.

2. Побудуйте контур ялинки, використовуючи примітиви (фігури) "Трикутник" і "Трапеція"  . Для виокремлення фігур на рисунку застосуйте команди групи **Редагування** стрічкового меню **Основне**:  і  . Для вирівнювання примітивів використайте послуги стрічкового меню **Засоби малювання** стрічкового меню  групи **Упорядкування** і меню  **Вирівняти.**

3.  Об'єднайте об'єкти кожного рисунка в групи, використовуючи команду  з меню **Згрупувати**  групи **Упорядкування**. Створіть кілька копій згрупованих зображень, переконайтесь у тому, що всі об'єкти згруповано. Змініть розміри копій зображень. Використовуючи стрічкове меню **Засоби малювання**, зверніть увагу, що не завжди всі позиції меню є активними. Визначте, зокрема, коли й які позиції меню **Згрупувати** є активними. Поясніть чому.

5.2. Векторний графічний редактор. Особливості побудови й опрацювання векторних зображень

Для створення векторних зображень використовують спеціальні програмні засоби – векторні графічні редактори. Ці зображення зберігаються в файлах, з іменами розширення яких вказують, з використанням якого програмного засобу зображення створені й з використанням якого можуть бути відтворені.

Вивчаючи текстовий процесор, ви бачили, що в текстовий документ можна вставити растрові рисунки й прості фігури, виконати над ними деякі дії.

Рис. 5.10. Векторні зображення, побудовані засобами редактора векторних зображень, вбудованого у текстовий процесор Microsoft Word

Сучасні текстові процесори (Microsoft Word, Libre Office Writer) мають досить потужні редактори векторних зображень, які дають змогу

створювати складні й досконалі зображення (рис. 5.10 – із конкурсу учнівських рисунків).

✓ *Кожний векторний графічний редактор надає користувачеві такі основні можливості:*

– вставляти у створюваний рисунок графічний примітив із бібліотеки примітивів;

– установлювати значення властивостей об'єкта (розміри, форма; спосіб накреслення ліній, їх колір; способи заливки, прозорість; розташування в певному шарі);

– групувати й розгруповувати об'єкти.

Для того щоб створити векторне зображення, потрібно накреслити одну або декілька простих фігур, надати цим фігурам певних властивостей і об'єднати їх.

✓ *Зазвичай це робиться таким чином:*

1. Із бібліотеки примітивів викликається потрібна фігура (у редакторі Microsoft Office Word – об'єкт **автофігура**).
2. Розміщується в потрібному місці.
3. Її властивості змінюються так, щоб отримати необхідне зображення.
4. Пункти 1–3 повторюються для інших фігур доти, доки "скелет" зображення не буде створено.
5. Фігури (об'єкти) розташовуються в шарах так, щоб зображення мало потрібний вигляд.
6. Коригуються їх розміри і форма.
7. Виконується розфарбовування (заливка) об'єктів.
8. Об'єкти об'єднуються в групу.

У Microsoft Office Word це робиться так, як показано на рис. 5.11.

Для того щоб створений рисунок можна було вставити у текстовий документ як одне ціле, не виконуючи додаткових дій (встановлення його властивості **Розташування**), можна ще до початку формування скелета рисунка з примітивів виконати команду **Створити полотно** з набору команд **Фігури**. Рисунок створюватиметься на "полотні" й переміщуватиметься разом із ним.

Виокремлення фігур (графічних примітивів) виконується в особливому режимі редактора – **Вибрати об'єкти**. Цей режим вмикається з вкладки стрічкового меню **Основне**, групи **Редагування** (зазвичай крайня праворуч у стрічковому меню), підгрупи **Виділити**. Курсор набуває вигляду похилої стрілки. Вихід із цього режиму здійснюється натисненням клавіші Esc.

Товщина контурних ліній фігур і їх колір спочатку встановлюються однаковими для всього рисунка. Для цього для першого ж перенесеного на рисунок примітива викликається натисненням правої кнопки миші меню **Формат рисунка**, у якому виконуються відповідні встановлення.

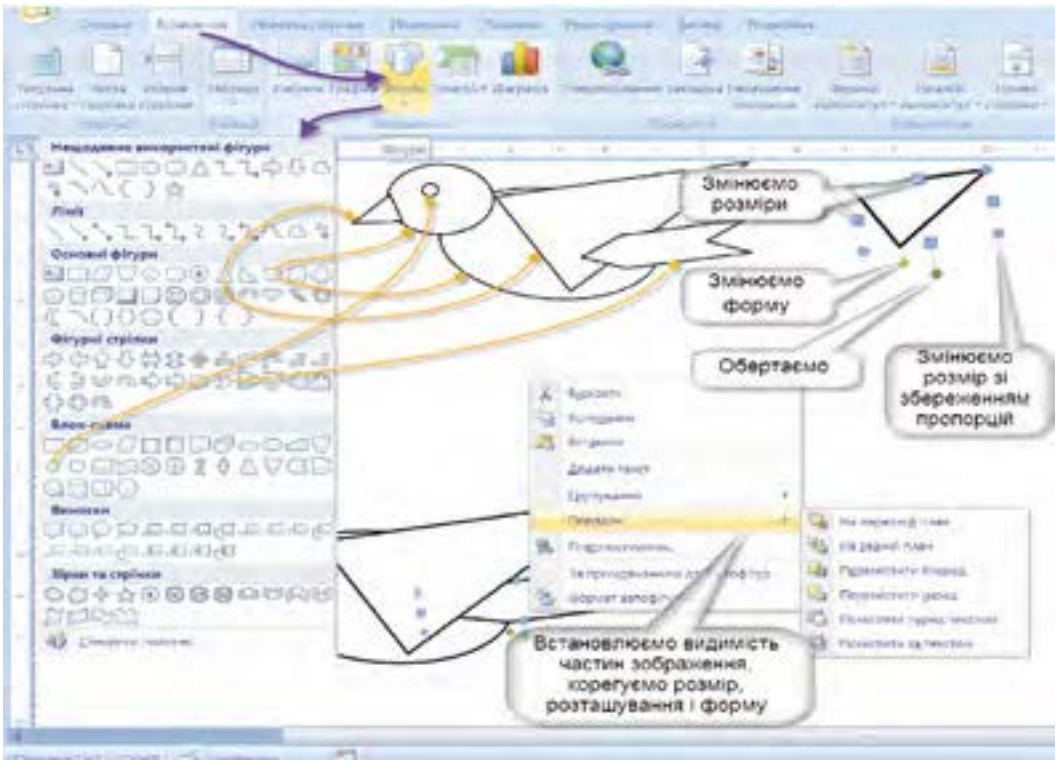


Рис. 5.11. Послідовність створення "скелета" зображення пташки засобами векторного редактора, вбудованого у текстовий процесор Microsoft Word

Розміри та, для деяких примітивів (фігур), форма встановлюються переміщенням **маркерів** зображень (маленькі квадратики і круги синього й зеленого кольорів, ромби жовтого кольору), які з'являються при натисненні кнопок миші на фігурі. Так само виконується і обертання фігур.

✓ *До більшості фігур можна додати текст, викликавши відповідну опцію контекстного меню.*

✓ *Властивості фігур можна змінювати, використовуючи меню* ***Формат автофігури****, яке викликається з контекстного меню або з груп стрічкового меню вкладки* ***Засоби малювання – Формат****.*

Після розміщення всіх необхідних примітивів на рисунку необхідно їх "наближати" або "віддаляти" таким чином, щоб забезпечити правильну видимість частин зображення, розміщуючи відповідним чином шари зображення. Команди переміщення шарів об'єктів **(Перемістити наперед, Перемістити назад, На передній план, На задній план)** містяться в меню групи **Упорядкування** вкладки **Формат** стрічкового меню або в контекстному меню, яке з'являється після виділення об'єкта або групи.

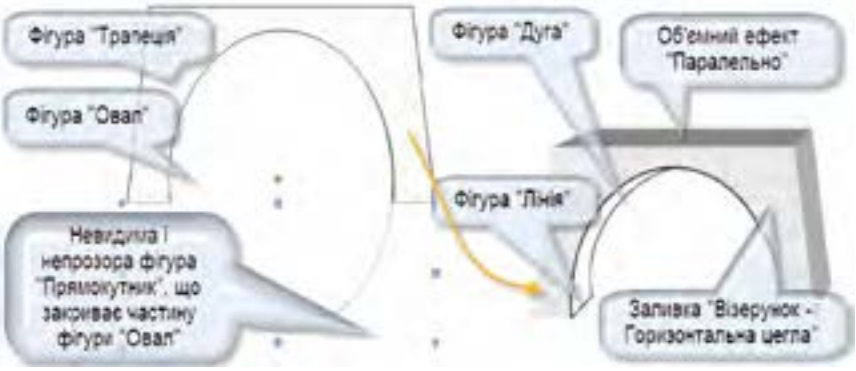


Рис. 5.12. Послідовність створення зображення арки засобами векторного редактора

Інколи на векторному рисунку потрібно перекрити частину зображення таким чином, щоб створити контур фігури, якої немає серед примітивів. Наприклад, арку воріт можна нарисувати, використавши примітиви "трапеція", "овал" і "прямокутник", перекривши непрозорим, але невидимим прямокутником частину овалу (рис. 5.12).

Створеним зображенням можна надати об'ємного вигляду (команди стрічкового меню **Засоби малювання**, група **Об'ємні ефекти**), зафарбувати (команди стрічкового меню **Засоби малювання**, група **Стилі фігур**, команда **Заливка фігури**).

Перевіряємо себе

1. Які властивості має об'єкт "лінія"? Об'єкт "стрілка"?
2. Які властивості має об'єкт "прямокутник"?
3. Чому в наборах примітивів більшості векторних редакторів відсутня фігура "коло"?
4. Яку фігуру потрібно використовувати для побудови кола?
5. Що називаємо "шаром зображення"?
6. Які дії можна виконати над будь-яким примітивом?
7. Серед маркерів яких об'єктів є жовті квадратики? Що вони позначають? Чому їх немає серед маркерів, наприклад, трикутника?
8. Що потрібно зробити, щоб із фігури (примітива) "куб" утворити паралелепіпед?
9. Для яких фігур не можна викликати опцію "додати текст"? Чому? Перевірте.
10. Визначте, з яких об'єктів складається зображення пташки.
11. У який шар слід помістити зображення хвостика пташки (рис. 5.11)?

Виконуємо

1. Побудуйте контур дерева, використовуючи примітиви (автофігури) “Виноска-хмарка” (крона), “Трапеціям” (стовбур) і “Пляма 1” (трава під деревом). Стики зображення крони і стовбура, стовбура і трави перекрийте непрозорими овалами.

2. Побудуйте зображення контуру арки (враховуючи відомості, подані написами на рис. 5.12), використовуючи прийоми перекривання непотрібних частин зображення непрозорими примітивами.

3. Спробуйте побудувати схематичні зображення різних об’єктів, використовуючи вже відомі вам прийоми, знаходячи нові примітиви й інструменти в групах вкладки **Засоби малювання**.

5.3. Способи зафарбовування об’єктів

Способи зафарбовування об’єктів, вибір кольору та способу зафарбовування об’єктів. Використання й змінювання готових векторних зображень.

Зафарбовування частин рисунка здійснюється з використанням відповідних меню. Цей процес називається **Заливка**. Заливка може бути суцільною, змінного кольору (градієнтна), візерунком, рисунком (штрихуванням).

Рис. 5.13. Основне меню властивостей автофігури (примітива)

Найпростішим є варіант суцільної заливки (досить вказати колір і прозорість). Колір заливки визначають на вкладці Кольори та лінії меню **Формат автофігури**. Заливку можна зробити повністю або частково прозорою, пересуваючи повзунок шкали “прозорість” (рис. 5.13).

Можна використовувати різні види заливки, у меню управління якими переходять із вкладки Кольори та лінії меню **Формат** автофігури натисненням кнопки Ефекти заливки... (див. рис. 5.13).

Якщо потрібно заповнити однаковою заливкою декілька примітивів на рисунку, їх необхідно виокремити, клацаючи на них лівою кнопкою миші й утримуючи при цьому клавішу Ctrl на клавіатурі.

На рис. 5.14 показано, як здійснити перехід до вкладки **Градієнтна заливка** й виконати заливку лівого крильця пташки двома кольорами (помаранчеві стрілки).

На рис. 5.15 показано, яким чином зафарбовано інші елементи зображення пташки.

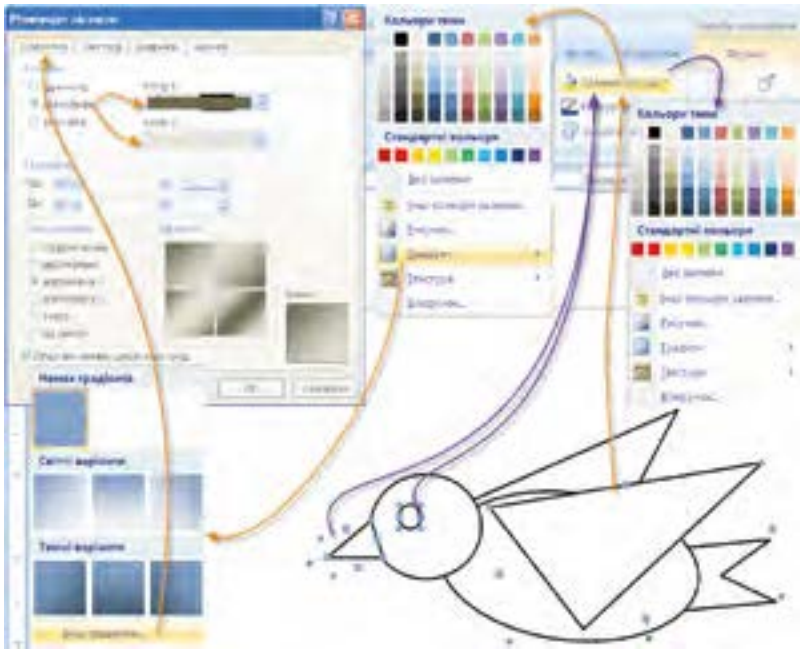


Рис. 5.14. Послідовність створення зображення пташки

Рис. 5.15. Завершення створення зображення пташки засобами векторного редактора

5.4. Вставлення і редагування векторних рисунків

У комплексі програмних засобів Microsoft Office є набір готових векторних зображень. Ці зображення зберігаються в декількох форматах векторної графіки, основними з яких є *.wmf і *.emf. Їх можна вставляти в документи так само, як і растрові рисунки з файлів, змінювати розміри, повертати.

Насамперед слід вибрати рисунок, який потрібно вставити. Якщо рисунок вже є в окремому файлі, то з вкладки **Вставлення** переходимо в групу **Зображення** (рис. 5.16), вибираємо в ній **Рисунок** і далі переходимо до вибору файла, розташованого на зовнішньому пристрої пам'яті (вінчестер, оптичний диск, флеш-накопичувач).

Якщо потрібно знайти рисунок у галереї, натисненням кнопки **Графіка** переходимо до меню **Картинки**. Зазвичай потрібно знайти файли певного типу (наприклад, із розширенням .wmf), тому позначаємо тільки цей тип файлів і натискаємо кнопку **Почати**.

Серед зображень, які з'являться, вибираємо потрібне. Вставляємо його в документ, двічі натиснувши на ньому лівою кнопкою миші.

Інколи потрібно щось змінити у вже готовому рисунку. Наприклад, забрати зайві написи, тіні, тло тощо, скомбінувати одне зображення з елементів кількох рисунків.

Нехай потрібно змінити колір тла, на якому зображено трактор, із жовтого з коричневим на блакитний, змінити форму тла (рис. 5.17).

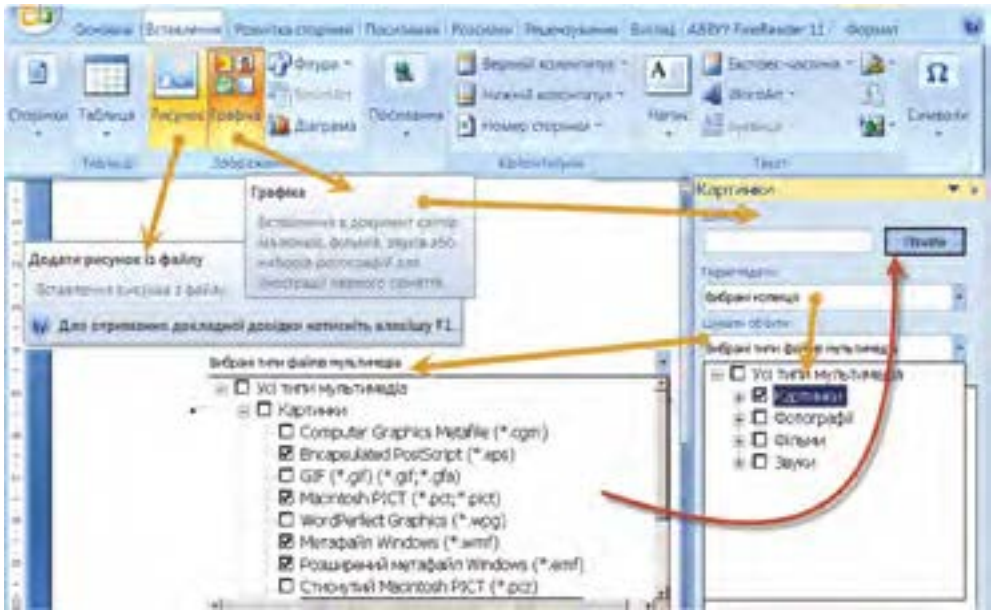


Рис. 5.16. Вставлення векторних рисунків (вибір і вставлення рисунка)

Рис. 5.17. Використання векторних рисунків (пошук і початок вставлення рисунка)

✓ *Спочатку слід розгрупувати рисунок.*

Для розгруповування в групі **Редагування** стрічкового меню **Основне** слід перейти в режим **Вибрати об'єкти** і вибрати рисунок. Потім у контекстному меню, яке з'явиться, вибрати опцію **Змінення рисунка**. Після цього можна вибрати потрібну фігуру і, рухаючи вузли, змінювати її форму. При потребі можна вилучити будь-яку фігуру, виокремивши її й натиснувши клавішу Delete.

Використавши меню **Формат автофігури**, фігуру можна зафарбувати, виконати інші дії (рис. 5.18).

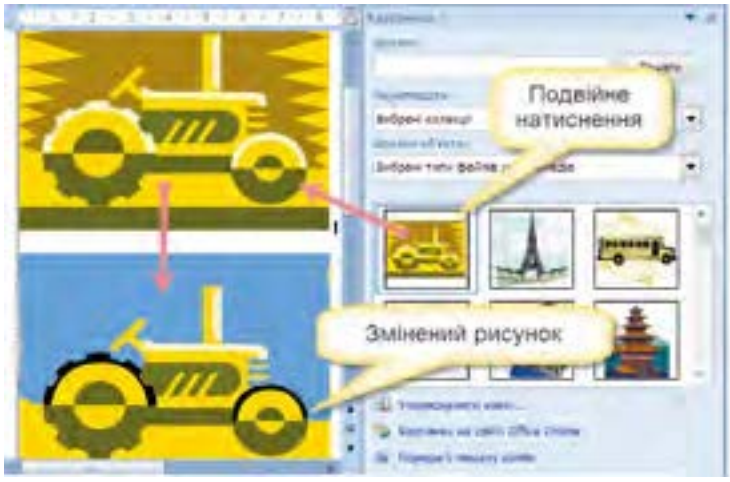


Рис. 5.18. Змінювання форми об'єкта (використання переміщення вузлів), змінювання кольору (заливка виокремленого об'єкта)

Перевіряємо себе

1. Як і навіщо встановлюється прозорість заливки фігури?
2. Чим відрізняються заливки Текстура і Візерунок?
3. Як встановити колір контурів цеглин заливки “Візерунок – Цеглина горизонтальна” (див. рис. 5.13)? Колір цеглин? Який з них буде кольором тла?
4. Що потрібно зробити, щоб розфарбувати фігуру двома кольорами? Перевірте на простих зображеннях.
5.  Що означає перемикач “Обертати заливку разом з фігурою”? Перевірте, заливши фігуру заливкою “Візерунок – Цеглина горизонтальна” і спробувавши обертати фігуру.
6. Збережене в яких форматах зображення можна змінити у текстовому процесорі? Як саме?
7. Скільки способів заливки використано для побудови зображення пташки?
8. Чому не всі зображення, отримані з колекції рисунків, вдається розгрупувати?
9. Чим відрізняється зображення трактора, взяте з колекції рисунків, від створеного зображення пташки?

Виконуємо

1. Розфарбуйте створеного раніше сніговика: морквяний ніс – червоним, очі (з вуглинок) – чорним, рот – буряковим кольорами. Збережіть для подальшої роботи.
2. Намалюйте кульку з градієнтною заливкою, центр градієнта якої не збігається з центром кульки (**підказка** – намалюйте два овали, для одного використайте градієнтну заливку “від центра” зі світлим центром, а для другого – “діагональну”, накладіть один на другий, ближчий шар зробіть напівпрозорим). Де можна використати таку заготовку?
3. Зафарбуйте ялинку, використовуючи градієнтну заливку.
4. Зафарбуйте створене дерево, використовуючи суцільні заливки (**підказка** – овали, які прикривають стики крони і стовбура, стовбура і трави, збільшити до повного перекриття стиків, їх лініям надати такого ж кольору, як і заливка).

5.5. Засоби векторного графічного редактора Inkscape

Inkscape – професійний інструмент для роботи з векторною графікою для Windows, Mac OS X і Linux. Його широко використовують аматори й професіонали по всьому світу для створення ілюстрацій, іконок, логотипів, діаграм, карт, а також веб-графіки.

Редактор Inkscape використовує відкритий стандарт SVG (Scalable Vector Graphics) як формат за замовчуванням, а також сам є вільним і відкритим програмним забезпеченням (рис. 5.19, 5.20).

Рис. 5.19. Основні меню графічного редактора

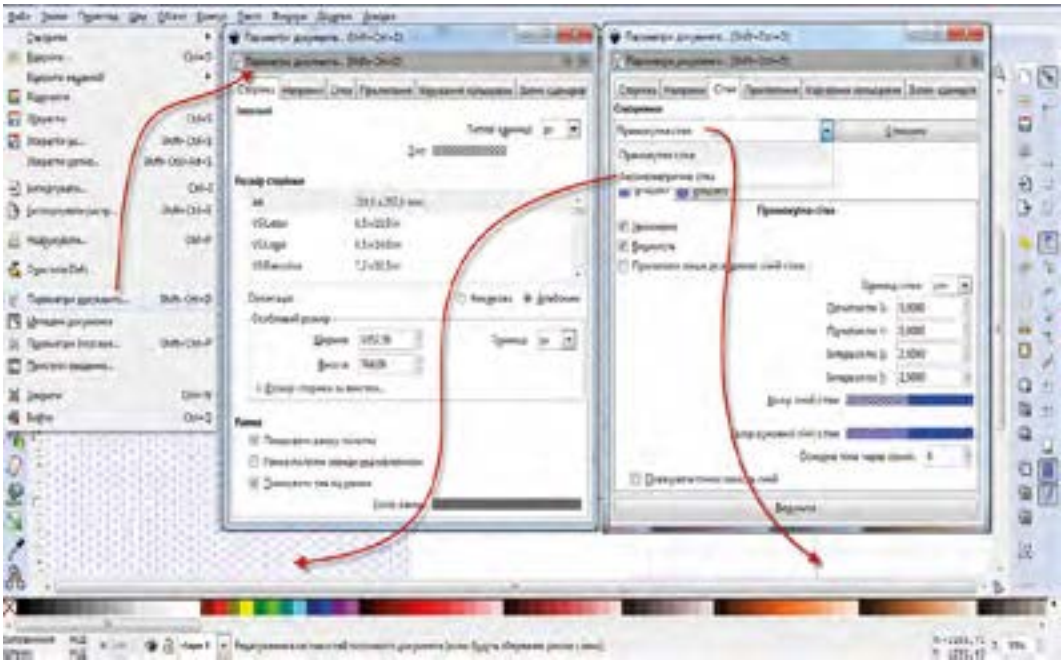


Рис. 5.20. Встановлення параметрів документа

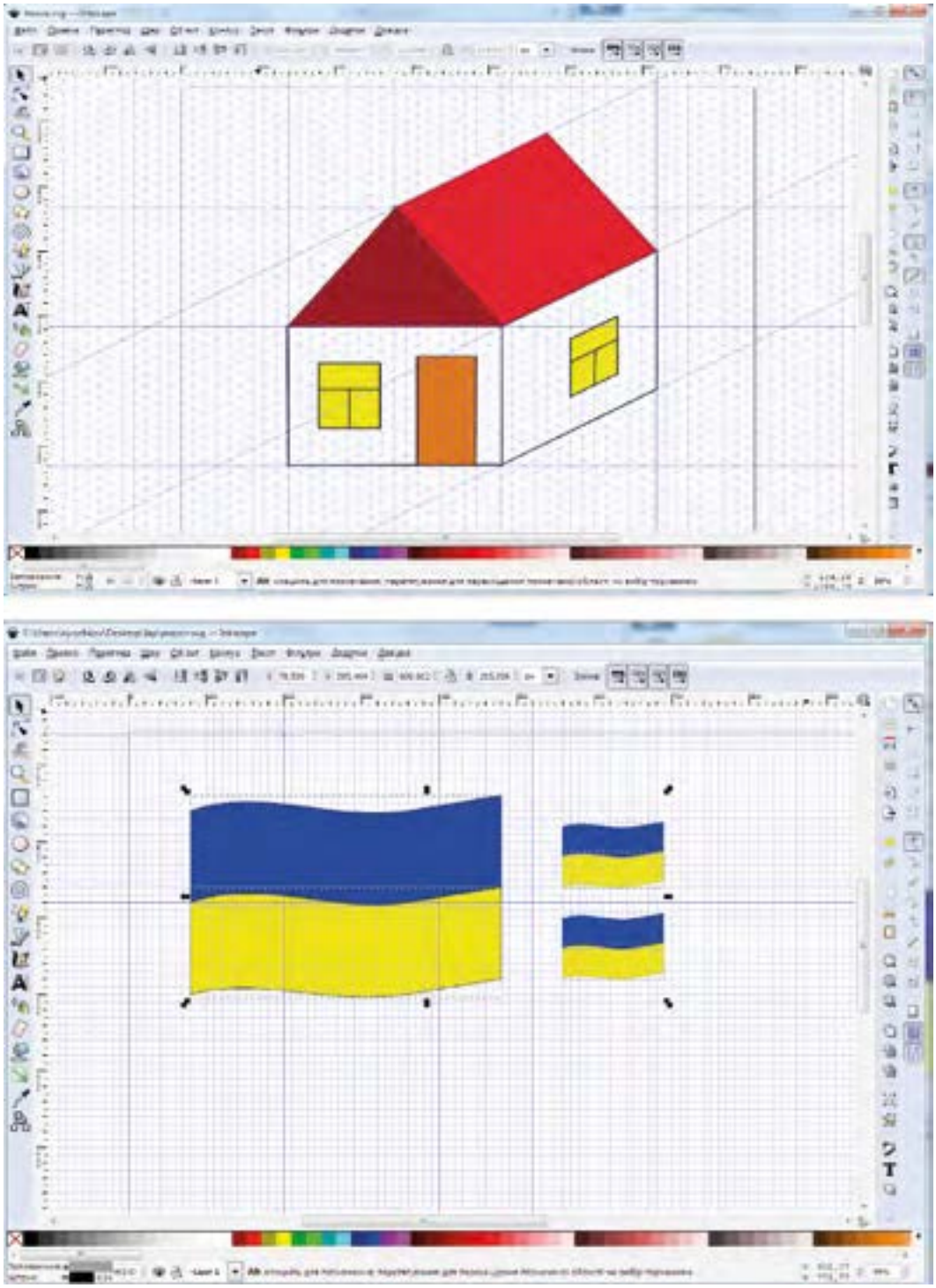


Рис. 5.21. Зображення, побудовані засобами редактора

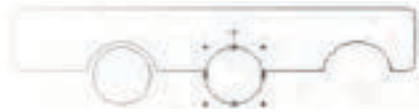
Практична робота № 8	Тема:	Створення простих векторних зображень
	Мета:	Навчитися створювати прості векторні зображення

1. Покроково виконайте побудову обрисів автомобіля (у MS Word або Inkscape).

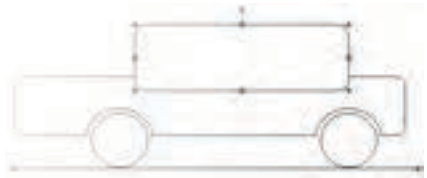
Насамперед рисуємо кузов (заокруглений прямокутник) і на ньому вирізаємо дві ніші для коліс. Для того щоб ніші були однакові, вставимо на місце першої овал, скопіюємо його, копію мишею перемістимо на початкове зображення, точно сумістимо. Далі переміщатимемо його праворуч, на те місце, де на кузові має бути ніша заднього колеса, але вже не мишею, а клавішею “стрілка вправо” на клавіатурі (щоб не змістити його по вертикалі).

Після цього рисуємо прямокутник, який прилаштовуємо так, щоб він перекрив нижні частини овалів, які будуть колісними нішами.

Робимо цей прямокутник невидимим, але й непрозорим, він закриє нижні частини овалів. Потім рисуємо коло, яке буде переднім колесом, ставимо його на місце. Копіюємо його, суміщаємо копію колеса з переднім колесом (щоб точно зафіксувати його положення). Переміщуємо копію, використовуючи клавішу “стрілка вправо” на клавіатурі.



Треба ще нарисувати дорогу та кабіну, сховати (закрити) нижню частину кабіни (так само, як ми закривали непрозорим прямокутником колісні ніші).



І, нарешті, об’єднуємо всі примітиви в зображення автомобіля. Дорисуємо вікна, бампери, інші деталі. Розфарбуємо створене зображення.

2. Перенесіть створене зображення у редактор растрової графіки. Спробуйте виконати дії над зображенням.

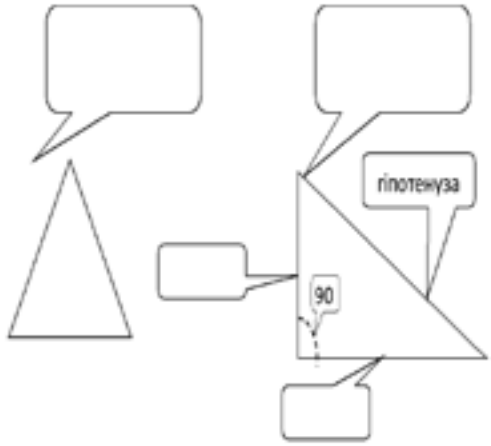
3. Дайте відповіді на такі запитання. Які дії над зображенням перестали бути можливими? Чому? Які дії над зображенням, неможливі у векторному редакторі, можна виконати у редакторі Paint? Чому? ★

Практична робота № 9	Тема:	Створення складених векторних зображень
	Мета:	Навчитися використовувати векторні зображення в документах із текстовими доповненнями

1. Створити зображення відомих вам геометричних фігур, додавши у виносках необхідні пояснення (назви фігур та їх елементів) за зразком. ✦

2. Створити афішу-запрошення на святкування Нового року (або іншого свята) за зразком. Можна використовувати всі зображення, які вже створені, та зображення з Microsoft ClipGallery. ✦

3. Створити у векторному графічному редакторі Inkscape одне із зображень, поданих на рис. 5.21. ★



СЛОВНИЧОК

Векторне зображення – зображення, найменшими елементами якого є примітиви (прості фігури).

Виноска – фігура (примітив), яка використовується для того, щоб зв'язати напис із зображенням або його частиною.

Вирівнювання – операція над кількома об'єктами, внаслідок якої вони розташовуються в певному порядку (команди меню: вирівняти по центру, ліворуч, праворуч тощо; розподілити по вертикалі, по горизонталі).

Глибина кольору – кількість бітів, якими кодується значення властивості "колір".

Градієнтна заливка – зафарбовування внутрішньої частини фігури кольором (або двома), які змінюються за певним законом.

Групування – операція над кількома об'єктами, внаслідок виконання якої створюється новий об'єкт.

Заливка – надання деякій частині зображення кольору, однакового для всіх точок, або такого, що змінюється певним чином, заповнення частини зображення штрихуванням або рисунком.

Примітив – об'єкт (відрізок, геометрична фігура, інше просте зображення), який вважається найпростішим у векторному зображенні (у графічному редакторі, вбудованому в програми Microsoft Office, називається "фігура" або "автофігура").

Розгруповування – операція над об'єктом, унаслідок виконання якої його частини (примітиви) стають окремими об'єктами.

Шар зображення – умовна площина, в якій розташовано примітив або групу примітивів (графічних об'єктів).

РОЗДІЛ 6. СТВОРЕННЯ ТА ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІЯ

Формати аудіо- та відеофайлів. Програмне забезпечення для опрацювання об'єктів мультимедія. Засоби перетворення аудіо- та відеоформатів. Захоплення аудіо та відео, створення аудіо-, відеофрагментів. Побудова аудіо- та відеоряду. Додавання до відеокліпу відеоефектів та налаштування переходів між його фрагментами. Налаштування часових параметрів аудіо- та відеоряду. Сервіси розміщення аудіо- та відеофайлів у Інтернеті.

6.1. Мультимедія

Мультимедія – цифровий продукт, що містить зображення й текст, які супроводжуються звуком, відео, анімацією, та оснащений інтерактивним інтерфейсом з елементами керування.

Мультимедійні документи відрізняються від звичайних тим, що крім традиційних текстових і графічних даних можуть містити звукові та музичні об'єкти, анімовану графіку (мультиплікацію), відеофрагменти.

Мультимедійне програмне забезпечення – програмні засоби, призначені для створення й відтворення мультимедійних об'єктів.

Найпоширеніші формати мультимедія.

Аудіоформати:

MP3 (Motion Picture Experts Group Layer 3) – використовує стиснення з втратами, підтримує стерео і потокове передавання.

WMA (Windows Media Audio) – ліцензований формат, розроблений корпорацією Майкрософт як альтернатива MP3.

Відеоформати:

WMV (Windows Media Video) – дуже стиснутий формат, який потребує мінімального обсягу дискового простору на жорсткому диску комп'ютера.

MPEG (Moving Pictures Experts Group) – забезпечує високу якість, підтримує додаткові можливості (захист від несанкціонованого копіювання, використання інтерактивних елементів) і потокове передавання відео.

.SWF (Відео Flash) – використовується для передавання відео через Інтернет за допомогою програвача Adobe Flash.

Для відтворення (прослуховування чи перегляду) аудіо- та відеофайлів використовують спеціальні програми – програвачі – плеєри.

RealPlayer – програвач відтворює звук та відео більшості відомих форматів, зокрема потокових, дає змогу записувати аудіофайли на компакт-диски в аналоговому вигляді.

QuickTime Player – надає змогу створювати й редагувати звук та відео, перекодовувати й зберігати їх у різних форматах;

WinAmp – універсальний програвач мультимедійних файлів, який відтворює файли практично всіх поширених аудіо- та відеоформатів;

Media Player Classic – безкоштовний програвач із простим і зручним інтерфейсом;

Windows Media Player – відтворює аудіо- та відеофайли більшості популярних форматів, входить до складу операційної системи Windows.

Через велику кількість різноманітних форматів аудіо- та відеофайлів у користувачів часто виникає необхідність перекодовувати ці файли з одного формату в інший.

Процес перекодування файла з одного формату в інший називається **конвертуванням** файлів.

Найпростіше конвертування можна виконати через медіяпрогравач Windows Media Player.

Існує також багато спеціалізованих програм для здійснення конвертування різноманітних мультимедійних даних.

Медіяпрогравач Windows та інші програми використовують кодеки для створення й відтворення цифрових медіяфайлів. Кодеки взаємодіють із певними прикладними програмами, зокрема з медіяпрогравачами, і допомагають їм відтворювати медіядані. Кодек може складатися з двох компонентів: кодувальника та декодера.

***Кодувальник** виконує функцію стискання (кодування).*

***Декодер** виконує функцію розпаковування (декодування). Деякі кодеки мають обидва ці компоненти, а деякі – лише один із них.*

Залежно від вимог, можна обрати кодек як із попередньо встановленими режимами конвертування аудіо і відео, так і з широкими можливостями налаштування процесу.

Утиліти-**конвертери мультимедія – програми, що виконують перетворення** файлів, які містять однотипні дані, але в різних форматах. Універсальні конвертери дають змогу змінювати параметри формату або формат файлів різних типів. Прикладом конвертера відео та аудіоформатів є Any Video Converter Freeware. Програми для запису звуку та відео (програми захоплення звуку, відео) називають програмами-**граберами**. Такими програмами є Exact Audio Copy, Wondershare Streaming Video Recorder та інші.

Перевіряємо себе

1. Що називають мультимедія?
2. Що таке мультимедійне програмне забезпечення?
3. Поясніть, що таке відео.
4. Як називається програма, що перетворює потік даних або сигналів на цифрові коди або навпаки?

5. Який пристрій називають медіяпрогравачем?
6. Назвіть найпоширеніші аудіоформати.
7. Назвіть найпоширеніші відеоформати.
8. Поясніть призначення медіяконтейнера.
9. Який процес називається конвертуванням файлів?
10. З чого складається кодек?
11. Як називаються програми, що виконують перетворення файлів?

Виконуємо

1. Знайдіть мультимедійні документи на вашому комп'ютері.
2. Чим мультимедійні документи відрізняються від текстових і графічних документів?
3. Скопіюйте в окрему папку файли, які належать до **Аудіоформатів.**
4. Скопіюйте в окрему папку файли, які належать до **Відеоформатів.**
5. Дослідіть і розкажіть про особливості роботи з утилітами-конвертерами.

6.2. Програмне забезпечення для опрацювання об'єктів мультимедія

Нині відомо багато програм, які можна використовувати для створення відеокліпів із зображень або створених раніше відео- та аудіофайлів.

Розглянемо стандартну програму операційної системи Windows – **Windows Movie Maker**. Цю програму можна знайти в меню: **Запустити – Усі програми – Windows Movie Maker.**

Вікно програми Windows Movie Maker складається з чотирьох основних частин (рис. 6.1): **Панель операцій з відео; Панель вмісту (контенту); Вікно попереднього перегляду** та **Область монтажу (Аркуш розкадрування/Шкала часу).**

Панель операцій з відео. Ця панель поділена на розділи, що відповідають покроковому процесу створення відео:

– ***Запис відео (Capture Video)*** – інструменти, які використовуються для початку створення відео: запис відео та імпорт існуючого відео, зображення чи звуку.

– ***Монтаж фільму (Edit Video)*** – інструменти перегляду існуючого відео, зображення, відеоефектів, прослуховування звуку або додавання у відео назв і титрів.

– ***Завершення створення відео (Finish Movie)*** – інструменти, призначені для збереження підготовленого відео на комп'ютері, компакт-диску чи відправлення засобами Інтернет.

– ***Поради щодо створення фільмів (Movie Making Tips)*** – довідковий матеріал про виконання операцій у програмі Windows Movie Maker.

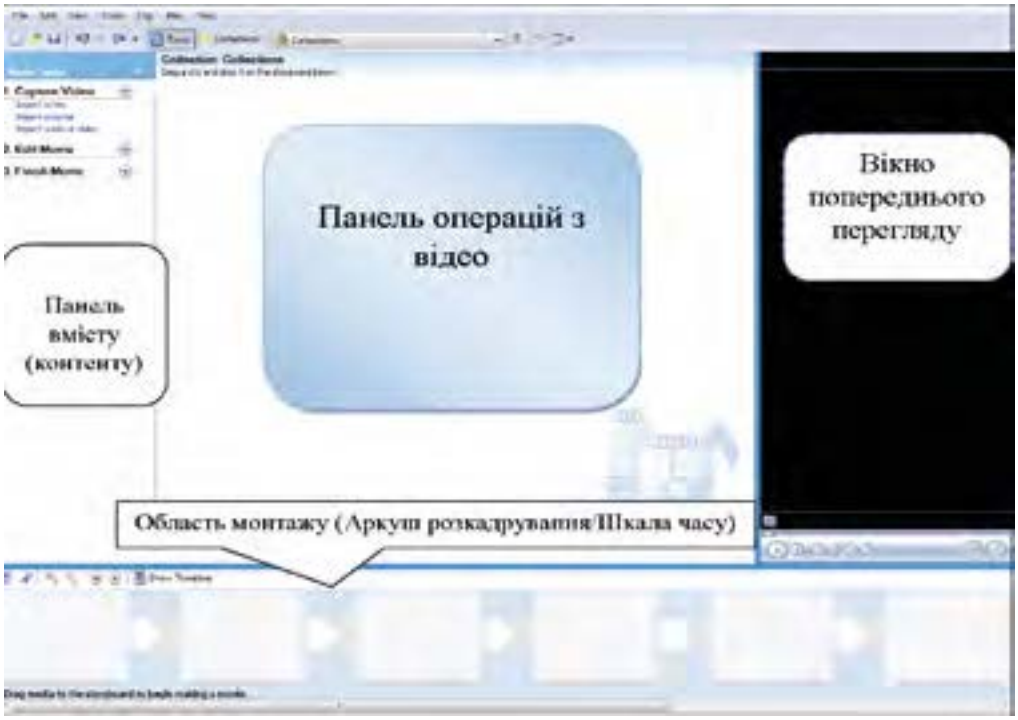


Рис. 6.1. Будова вікна програми Windows Movie Maker

Панель вмісту (контенту). На цій панелі відображаються кліпи, які містяться у рубриці, вибраній на **Панелі операцій.** Тут відображаються всі файли відео, аудіо, зображення, відеопереходи та візуальні ефекти, які можна додати на шкалу розкадрування або часу.

Вікно попереднього перегляду. Це вікно можна використовувати для перегляду як окремих кліпів, так і всього проекту, перш ніж його зберегти.

Область монтажу. Тут створюються й монтуються проекти. Вона відображається у двох режимах: **Аркуш розкадрування** і **Шкала часу.** Під час створення фільму можна переключатися між цими двома режимами.

У режимі **Аркуш розкадрування** відображаються картинки всіх вибраних фрагментів і налаштовані переходи між ними.

У режимі **Шкала часу** відображаються відеофрагменти, довжини яких пропорційні часу їх відтворення, переходи між ними, звуковий супровід і титри.

Перевіряємо себе

1. Які програми можна використовувати для створення відеокліпів із зображень або створених раніше відео- та аудіофайлів?
2. Із яких основних частин складається вікно програми Windows Movie?

3. Розкажіть про призначення кожного розділу **Панелі операцій з відео.**
4. Для чого призначена **Панель вмісту (контенту)?**
5. Для чого призначено **Вікна попереднього перегляду?**
6. Розкажіть про призначення **Області монтажу.**

Виконуємо

1. Завантажте програму Windows Movie Maker.
2. Знайдіть на вашому комп'ютері відеофайл.
3. Завантажте його на **Панель вмісту (контенту).**
4. Додайте назву й титри, використовуючи **Область монтажу.**
5. Перегляньте результат вашої роботи через **Вікно попереднього перегляду.**
6. Збережіть створений файл.

6.3. Створення та опублікування мультимедія

Захоплення аудіо та відео, створення аудіо-, відеофрагментів із використанням цифрових відео або фотокамер. Побудова аудіо- та відеоряду. Додавання до відеокліпу відеоефектів та налаштування переходів між його фрагментами. Налаштування часових параметрів аудіо- та відеоряду. Збереження створених відеофільмів на носіях даних. Сервіси публікування відеофайлів. Подка́стинг.

Додавання зображень і аудіо- й відеофайлів можна виконати за допомогою **Панелі операції:**

1. Для імпортування фотографій натисніть **Import Pictures.**
2. Щоб додати до проекту аудіофайл, натисніть **Import audio or music.**
3. Для додавання існуючого відео натисніть **Import video.**

Усі вибрані файли будуть додані на **Панель вмісту,** як показано на рис. 6.2.

Для монтажу фільму потрібно.

1. За допомогою мишки перетягнути фотографії та аудіофайл із Панелі операції в **Область монтажу** на **Шкалу часу.**
2. Потім відсортувати фотографії, перетягуючи їх мишкою по **Шкалі часу,** в такому порядку, в якому вони будуть відображатись у створеному фільмі.
3. Беручи до уваги, що стандартна тривалість показу фотографії 5 секунд, тривалість показу конкретного зображення можна змінити, розтягуючи його, притиснувши ліву межу зображення лівою кнопкою миші.

Рис. 6.2. Вікно програми Windows Movie Maker, в якому на **Панель вмісту** додано фотографії та аудіофайл

4. В **Області монтажу** перейти зі **Шкали часу** на **Аркуш розкадрування.** Для переходу між цими двома режимами використовується кнопка перемикання між режимами **Show Storyboard** (рис. 6.3).

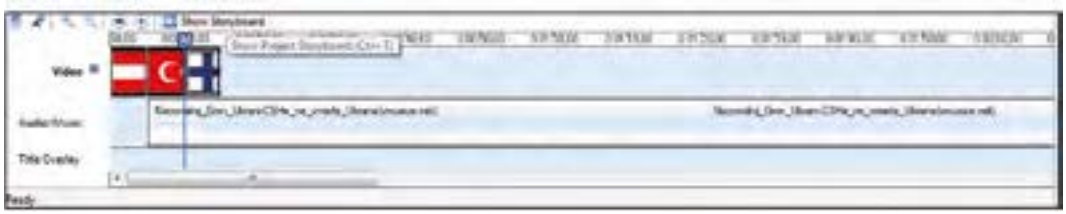


Рис. 6.3. Область монтажу, де виділено кнопку перемикання між режимами **Show Storyboard**

5. Для того щоб фото перегорталися з особливим ефектом, можна використати спеціальні переходи. Для їх перегляду необхідно клацнути на меню **Show video transitions** (рис. 6.4).

6. Для перегляду ефекту клацнути лівою кнопкою миші на іконці ефекту – він відобразиться у **Вікні перегляду.**

7. Вибравши ефект, його слід перетягнути мишкою на **Шкалу розкадрування** між двома фотографіями, до яких цей ефект буде застосований.

Відеопереходи скорочують тривалість показу зображень на шкалі, оскільки відбувається накладання кадрів, тобто після застосування відеопереходів на весь час програвання аудіо буде зайняте. У цьому випадку рекомендується додати ще декілька зображень.

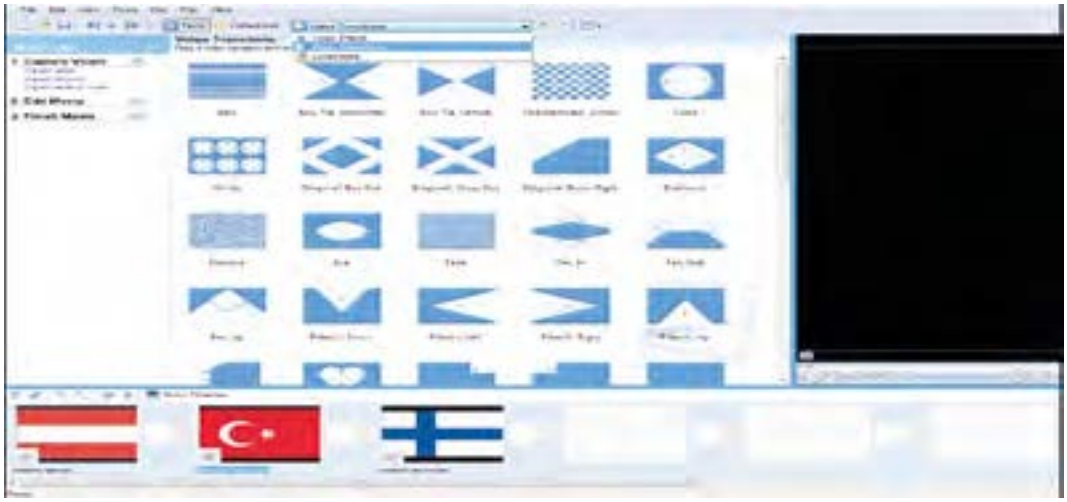


Рис. 6.4. Перегляд колекції переходів

Для видалення ефекту слід вибрати кнопку  *на потрібному кадрі й натиснути клавішу* ***Delete****.*

Збереження створеного відеофрагмента виконується так:

1. На **Панелі операцій** натиснути на посилання **Save to my computer**, дати ім'я створеному відео та обрати папку, де воно буде збережене.

2. Файл буде збережений у форматі *.WMV, і надалі його можна використовувати як звичайний відеофайл: копіювати, відтворювати з використанням відеопрогравачів, конвертувати, вставляти в слайдові презентації тощо.

Після завершення процедури збереження рекомендується зачекати 5–10 хв, допоки програма перетворить створений файл проекту на фільм. Відтак його можна опублікувати на Youtube або записати на будь-який носій збереження інформації.

Невеликі за розмірами відеофільми називаються ***кліпами****.*

Текст, що виводиться наприкінці фільму, називають **титрами**, а будь-який інший текст у фільмі – **назвами**.

Мультимедійні файли, які розповсюджуються через Інтернет, називаються **подкастами** (англ. Podcast). Подкасти можуть містити інтерв'ю, лекції чи будь-що інше, що належить до усного жанру.

Термін "**podcast**" є поєднанням назви портативного програвача музики iPod і слова **broadcast** (трансляція).

У Windows Movie Maker можна захопити зображення окремого кадру з імпортованого відео- чи аудіофайла, а потім використати це зображення як статичне у створеному фільмі. Зображення, які захоплюються з відео-

кліпів у Windows Movie Maker, автоматично зберігаються як файли в форматі JPEG з розширенням .jpg.

1. У області **Вміст** клацніть відеокліп, з якого потрібно захопити зображення.

2. Знайдіть кадр у відеокліпі, який потрібно зберегти як окреме зображення, пересуваючи індикатор відтворення під **Вікном попереднього перегляду**. Для пошуку потрібного кадру можна скористатися кнопками **Наступний кадр** і **Попередній кадр**.

3. Виберіть у меню **Знаряддя** пункт **Зробити знімок із попереднього перегляду**.

4. Уведіть ім'я для файла зображення та натисніть кнопку **Зберегти**.

Можна також захопити окреме зображення з відеокліпу на **Аркуші розкадрування/шкалі часу**, але якщо захоплювати його з кліпу в області **Вміст**, на виході часто можна отримати зображення вищої якості. Можна провести захоплення (запис) аудіо(відео) за допомогою спеціальних програм (граберів).

Наприклад, програма Windows **Звукозапис**. Її можна запустити натисканням меню **Пуск** – **Усі програми** – **Стандартні** – **Звукозаписувач**. Ця програма записує звуки в форматі WAV тривалістю 60 с (за допомогою мікрофона).

Перевіряємо себе

1. За допомогою чого можна виконати додавання зображень та аудіо- і відеофайлів?
2. Які дії потрібно виконати для монтажу фільму?
3. Що потрібно зробити для видалення ефекту?
4. Які відеофільми називаються кліпами?
5. Які мультимедійні файли називаються подкастами?

Виконуємо

1. Зробіть кілька фотографій робочого дня вашого класу.
2. Відберіть фотографії, які найчастіше відтворюють особливості навчання у вашій школі.
3. Підберіть аудіофайли, які можуть бути супровідними для фотографій.
4. Відберіть програму, за допомогою якої можна створити відеокліп. Поясніть свій вибір.
5. За допомогою обраної програми створіть відеокліп на 20 с.

Практична робота № 10	**Тема:**	Захоплення та конвертування аудіо- та відеоданих
	Мета:	Набути практичних навичок опрацювання об'єктів мультимедія

Завдання. Користуючись Windows Movie Maker, за матеріалом, наданим учителем, створити мультимедійний файл.

Примітка. *Файл має містити аудіо та відеодані.*

Практична робота № 11	**Тема:**	Створення відеокліпу. Додавання відеоефектів, налаштування часових параметрів аудіо- та відеоряду
	Мета:	Набути практичних навичок опрацювання об'єктів мультимедія

Завдання. Створити відеокліп **"Мої друзі".**

Дотримуйтесь таких етапів створення.

1. На першому етапі відеомонтажу кліпу створюєте новий проект і перетягуванням встановлюєте мультимедійні об'єкти зі збірника в **Область монтажу.**

2. На другому етапі створюєте переходи між зображеннями, додаєте відеоефекти, титри та голосовий супровід.

3. На третьому етапі відеофільму вставляєте титри або текстовий супровід.

Рекомендації

1. Для оформлення відеокліпу використовуйте власну фотогалерею.

2. Для супроводу показу вставте улюблену музику.

3. Додайте текстові написи до відеофільму. Для цього виконайте таке:

– У меню **Tools** клацніть **Titles and Credits** (рис. 6.5).

– Відкриється вікно, в якому оберіть: вставлення титрів, назви перед обраним зображенням чи після нього тощо.

– Вставте необхідний текст.

– Збережіть виконане.

Рис. 6.5. Меню **Tools** з виділеною командою вставлення назв і титрів

Тривалість демонстрації вставлених титрів за замовчуванням 3,5 с. Її можна змінити так само, як і тривалість демонстрації зображень.

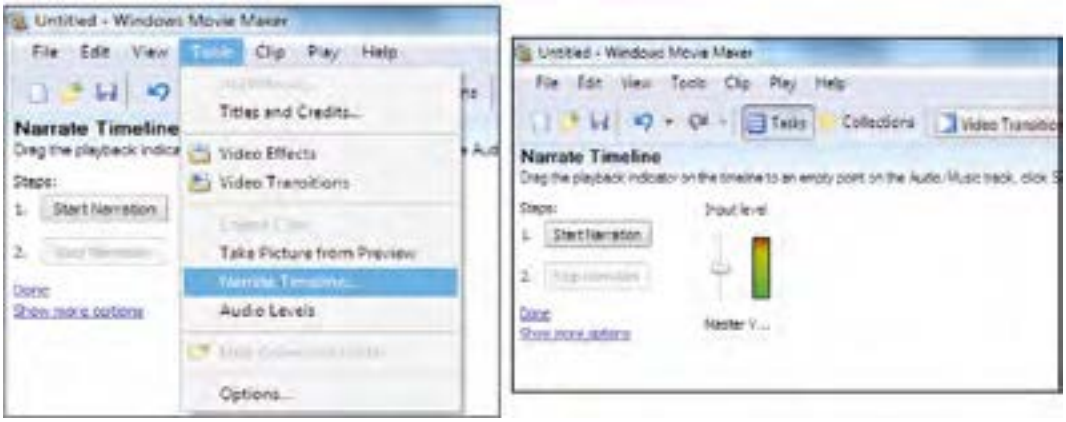


Рис. 6.6. Засоби запису дикторського тексту

Для титрів, що розміщуються всередині кліпу, можна перетягувати як ліву, так і праву межу.

4. Додайте дикторський звуковий супровід. Для цього виберіть точку на **Шкалі часу**. У меню **Tools** клацніть **Narrate Timeline** або виберіть кнопку у лівій частині **Області монтажу** (рис. 6.6). Відкриється вікно запису. Для початку запису виберіть кнопку **Start**, а для призупинення чи завершення запису – кнопку **Done**.

СЛОВНИЧОК

Відео – послідовність зображень – кадрів, які створюють рухоме зображення.

Декодер – програма, що виконує функцію розпаковування (декодування).

Кліп – невеликий за розмірами відеофільм.

Кодек – програма, що перетворює потік даних або сигналів на цифрові коди.

Кодувальник – програма, що виконує функцію стискання (кодування).

Конвертування файлів – процес перекодування файла з одного формату в інший.

Медіяконтейнер – формат, що дає змогу розміщувати в одному файлі мультимедійні дані різних типів і синхронізувати звук; відеозображення й текстову інформацію.

Медіяпрогравач – пристрій, за допомогою якого відтворюється відео-, фото- й аудіоконтент.

Мультимедія – програмний продукт, що містить зображення й текст, які супроводжуються звуком, відео, анімацією, та оснащений інтерактивним інтерфейсом з елементами керування.

Подкасти – мультимедійні файли, які розповсюджуються через Інтернет.

Титри – текст, що виводиться на початку і наприкінці фільму.

РОЗДІЛ 7. МУЛЬТИМЕДІЙНІ ПРЕЗЕНТАЦІЇ

Растрове зображення – зображення, найменшим елементом якого є піксель. Векторне зображення – зображення, найменшим елементом якого є примітив. Анімоване зображення – зображення, що рухається по екрану або змінює форму. Відеозображення – рухоме зображення (або зображення зі звуком), створене відеозніманням реальних об'єктів.

Етапи розроблення презентації. Критерії оцінювання презентації. Макети слайдів. Стильове оформлення слайдів презентації. Елементи дизайну презентацій. Використання організаційних діаграм у презентаціях. Проектування та розроблення розгалужених презентацій. Гіперпосилання та елементи керування в презентаціях. Додавання відеокліпів, звукових ефектів і мовного супроводу до слайдової презентації. Елементи анімації. Вбудовані та зв'язані об'єкти в презентаціях. Керування показом презентації. Друкування презентації.

7.1. Розроблення презентації

У загальному випадку презентацією називають захід, який проводиться з метою поширення інформації про певну організацію, подію, товар, послугу. За допомогою презентації бажають привернути увагу до певного об'єкта, події (організації, фірми, яка продає товар, надає послуги тощо, власне товару) або до певної особи.

Сукупність зображень (нерухомих, анімованих та відеозаписів), звукових записів і екранних засобів керування, певним чином упорядковану, називають **"комп'ютерною презентацією"**, або **"мультимедійною презентацією"**.

Презентації Microsoft Office PowerPoint 2007 можна перетворити на файли формату *.xps (англ. XML Paper Specification) і *.pdf (англ. Portable Document Format) для їх передавання користувачам, що працюють у будь-якій операційній системі.

Комп'ютерні презентації можна класифікувати за різними ознаками.

За структурою і способом подання матеріалу.

1. Комп'ютерні презентації, які складаються з окремих зображень (слайдів). Такі презентації створюють за допомогою програми PowerPoint або подібних. Відтворення таких презентацій відбувається у вигляді послідовності зображень, відеофрагментів як неперервна в часі послідовність

або з використанням засобів управління. Управління відтворенням таких комп'ютерних презентацій може здійснювати доповідач (на лекції, уроці, презентаційному заході) або особа, що ознайомлюється з інформацією (у мережі Інтернет, локальній мережі, окремому комп'ютері).

2. Комп'ютерні презентації у вигляді відеоролика. Зазвичай у них використовуються зняті на відеокамеру невеликі сюжети або комп'ютерна графіка (дво- або тривимірна, анімована, записані у вигляді неперервної послідовності зображення нерухомих або рухомих об'єктів екрана тощо).

✓ *За призначенням.*

Виокремлюють дві групи комп'ютерних презентацій: для аудиторного або для індивідуального використання. Ці дві групи мають як спільні ознаки, так і істотні відмінності, що необхідно брати до уваги під час їх створення.

Презентації можуть бути **лінійними**, коли переходи можливі від попереднього зображення до наступного, і навпаки. Лінійні презентації досить поширені, просто й швидко створюються. Зазвичай їх використовують для супроводу лекцій, доповідей.

Інтерактивні презентації відрізняються розгалуженою структурою, в них можливі переходи від однієї групи зображень до іншої. Переходами керує доповідач або користувач. Такі презентації можуть бути використані індивідуально, інколи їх використовують у мережі Інтернет або локальній мережі (після збереження в форматі веб-сторінки).

✓ *У процесі створення презентації можна виокремити декілька етапів:*

1) розроблення змісту і структури комп'ютерної презентації;

2) створення окремих слайдів;

3) формування послідовності показу слайдів, або створення набору слайдів, у яких передбачені переходи зі слайда на слайд за командою користувача ("дерева" слайдів);

4) створення допоміжної підтримки презентації (відеозаписів, програмних засобів тощо);

5) планування презентації (визначення часу демонстрування окремих слайдів, необхідності повернення до початкових слайдів тощо).

Насамперед слід визначити мету створення презентації та спосіб її відтворення в процесі демонстрування. Від цього залежатимуть як продуктивність роботи над її створенням, так і успішність застосування.

Залежно від мети створення презентації її слайди можуть відтворюватись або послідовно один за одним, або в іншій послідовності, можуть бути використані зображення, текстові документи, програмні засоби, звукові та відеозаписи, які зберігаються за межами файла комп'ютерної презентації.

Для створення презентації використовується команда *Створити*, яка подається зі стрічки команд (кнопка *Office*). Відкриється вікно (рис. 7.1), в якому можна вибрати шаблон презентації.

У наборі шаблонів PowerPoint є значна кількість уже готових шаблонів, які можна вибрати, скориставшись опцією “*Інстальовані шаблони*”.

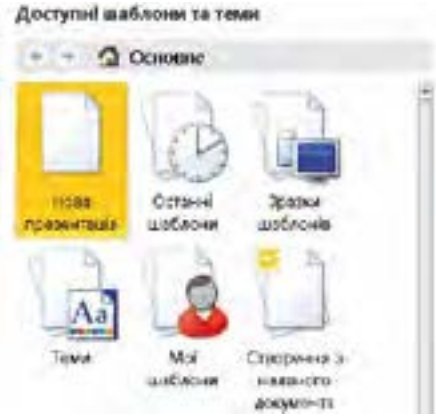


Рис. 7.1. Вікно з шаблонами презентацій

✓ *Найпростіший і найбільш поширений спосіб визначення зовнішнього вигляду презентації – використати шаблон презентації: кнопка Offise → Створення презентації → Інстальовані шаблони → Створити* (див. рис. 7.1).

Вибір шаблону забезпечує визначення зовнішнього вигляду презентації. Шаблон є найбільш потужним засобом управління дизайном презентації.

✓ *Зміна шаблону дає змогу:*

– змінювати схему кольорів, від якої залежать кольори складових слайдів презентації;

– змінювати зразок заголовка і зразок слайда, від яких своєю чергою залежать вигляд слайдів і параметри форматування тексту, що використовуватимуться за замовчуванням.

Змінюючи шаблон, можна надати презентації абсолютно нового зовнішнього вигляду: всі слайди отримають однакове тло, колірну схему і набір шрифтів тексту. Проте кожним із цих елементів дизайну можна управляти окремо, не застосовуючи новий шаблон. Є можливість змінити вигляд окремого слайда, не поширюючи його на інші слайди, тобто залишити їх оформлення у вигляді, визначеному шаблоном.

Користувачі мають змогу створити власний шаблон через розроблення базових елементів презентації та збереження їх у вигляді файла-шаблону.

Перевіряємо себе

1. Що називають комп’ютерною презентацією?
2. Якими можуть бути комп’ютерні презентації за структурою і за призначенням?
3. У якій послідовності доцільно створювати презентацію?

4. Із якою метою можна використати певні презентації?

5.  У яких файлах можуть зберігатись шаблони презентацій, створені користувачем?

6.  Із якою метою використовують колір тла?

Виконуємо

1. Завантажте з пристрою пам'яті готову презентацію, визначте, в якій послідовності відтворюються слайди.

2. Завантажте з підменю **Інстальовані шаблони** шаблон презентації "Знакомство с Power Point 2007", перейдіть у режим перегляду, перегляньте презентацію. Збережіть файл у форматах презентації та демонстрації. Перегляньте файли. Зробіть висновки.

7.2. Елементи дизайну презентацій

Найменшим елементом презентації, яким можна оперувати під час її відтворення, є слайд, названий за аналогією з кадром фотографічної плівки, призначеним для використання у звичайному проекторі.

Слайдом називають інформаційну структуру, яка містить різні об'єкти, що подаються на екрані монітора, аркуші паперу або прозорої плівки у вигляді єдиної композиції.

Термін **"слайд"** використовується для позначення одиниці матеріалів презентації незалежно від того, чи буде ця сторінка демонструватися на екрані дисплея, роздруковуватись на принтері або виводитися на прозору плівку.

До слайда можуть належати такі об'єкти: заголовок і підзаголовок, зображення, таблиці, діаграми, організаційні діаграми, тексти, фрагменти мультимедія, маркіровані списки, тло, колонтитул, номер слайда, дата, різні зовнішні об'єкти.

Презентація – це набір кадрів (слайдів), які зберігаються зазвичай у загальному файлі, об'єднаних можливістю переходу від одного слайда до іншого.

Одним із компонентів шаблону є схема кольорів, яка визначає кольори окремих елементів оформлення слайдів.

Особливим елементом слайда, за допомогою якого здійснюється структурування презентації, є гіперпосилання.

Гіперпосилання – виокремлений текст або графічний об'єкт, під час активізації якого (клацанням кнопкою миші на ньому) виконується

перехід до файла, фрагмента файла або веб-сторінки в мережі Інтернет, який описано в посиланні.

Одночасно в слайді може міститися декілька гіперпосилань, кожне з яких визначає маршрут до певного об'єкта.

Нижче подано перелік об'єктів слайда, колір яких формує схему кольорів і визначає колірне оформлення елементів слайда.

Тло – колір, використовуваний для оформлення всього слайда (розміщується як найдальший від спостерігача шар зображення). На окремих слайдах колір тла може перекриватись іншим кольором, указаним користувачем.

Текст і лінії – колір тексту в маркірованих списках і текстових блоках. Цей же параметр використовується для ліній і стрілок, нарисованих за допомогою інструментів **Лінія** (Line) і **Стрілка** (Arrow), для контурів автофігур та інших об'єктів, нарисованих за допомогою інструментів малювання.

Тіні – колір тіней, якими будуть оформлені графічні об'єкти і текст.

Заголовок – напис на слайді, який, разом із номером слайда, ідентифікує слайд у презентації. Може бути визначений колір заголовків і підзаголовків слайдів. Заголовок використовується у відтворенні схеми презентації.

Заливка – зафарбовування автофігур та об'єктів, створених за допомогою інструментів малювання.

Акцент – колір, використовуваний як другий колір на діаграмах та інших екранних об'єктах.

Акцент із гіперпосиланням – колір, використовуваний як третій колір на діаграмах та інших упроваджених об'єктах. Цей же колір застосовується для об'єктів – носіїв гіперпосилань.

Акцент із подальшим гіперпосиланням – колір, використовуваний як четвертий колір на діаграмах та інших вставлених об'єктах. Він також застосовується для об'єктів – носіїв гіперпосилань.

Гіперпосилання на об'єкти, що зберігаються не в файлі презентації, потрібно здійснювати з вказівкою шляху до них. Наприклад, якщо програма, яку необхідно запустити на виконання, розташована на пристрої пам'яті в каталозі, в якому зберігається презентація, досить вказати назву файла, інакше – слід вказати повний шлях.

Наявність складових презентації, які зберігаються в зовнішніх файлах, слід брати до уваги при перенесенні презентації.

Якщо здійснюється посилання на сайт мережі Інтернет, потрібно вказати шлях до нього таким же чином, яким він вказується для браузера.

Для створення слайда використовується команда **Створити слайд**, яка подається зі стрічки команд (закладка **Основне**). При цьому відкри-

ється вікно, в якому можна вибрати авторозмітку для створюваного слайда, тобто розташування заголовка, тексту і зображень, які будуть складовими слайда.

Рис. 7.2. Створення слайда певного призначення за авторозміткою

Авторозміткою слайда називають структуру та оформлення написів і зображень на слайді, які розроблені для слайдів різного призначення і зберігаються в програмі PowerPoint. Як бачимо з переліку авторозміток (рис. 7.2) та прикладів структур слайдів, до складу слайда можуть бути включені: зображення, графіки, таблиці, маркіровані списки, організаційні діаграми, а також їх поєднання.

Можна розпочати і з порожнього слайда (авторозмітка "Порожній слайд") і самостійно сформувати потрібну структуру слайда. Важливим елементом слайда є заголовок, оскільки він використовуватиметься надалі як назва слайда і разом із номером слайда відіграватиме роль ідентифікатора (імені) слайда. Наявність заголовка на слайді необхідна для формування структури документа у разі збереження презентації у форматах *.xml та *.html.

Колонтитул (не обов'язкова, але важлива частина слайда) призначений для розміщення написів, які розміщуються автоматично на всіх слайдах. Це можуть бути назва презентації, відомості про автора, номер слайда, дата створення тощо.

✓ *У презентації може існувати спеціальний слайд, який використовується не в демонстрації, а для визначення та збереження зразків об'єктів для інших слайдів. Цей слайд називається* ***майстер-слайдом.*** Його можна приховати від перегляду в режимі демонстрації з використанням кнопки **Приховати слайд** стрічкового меню **Показ слайдів**.

Розміщені на майстер-слайді об'єкти містяться на решті слайдів. Це, з одного боку, дає змогу автоматизувати такі рутинні роботи, як розміщення колонтитулу, дати, загального тла тощо, а з іншого – унеможливлює внесення будь-яких змін у об'єкти, які прийшли на слайди

з майстер-слайда. До об'єктів слайда, запозичених із майстер-слайда, не можна застосовувати ефекти анімації.

Під час створення презентації дуже важливо вибрати спільне для всіх слайдів оформлення, яке б відображало певним чином зміст презентації.

Оформлення презентації, яке відповідає змісту презентації, підкреслює спрямованість оповіді, об'єднує слайди в єдиний твір, називають **стилем презентації.**

Оформлення презентації не має бути занадто яскравим, не відволікати увагу від змісту презентації та змісту кожного слайда.

Крім суто естетичних й інформативних вимог до стилю комп'ютерної презентації необхідно враховувати й особливості способів її відтворення, що різні для аудиторного або індивідуального застосування презентації.

✓ *Більшість систем відтворення комп'ютерних презентацій забезпечує такі режими:*

– відтворення на повному екрані (для одного користувача, або для групи) у формі послідовності слайдів, що відтворюється без переходів на слайди, які не йдуть за попереднім, повернень тощо, тобто у формі "потокової презентації";

– відтворення слайдів у екранному вікні з можливістю залишати нотатки як на поверхні слайда, так і в спеціальному вікні, накладання інших зображень, переміщення складових слайда користувачем (забезпечують системи відтворення презентацій, що входять до комплектів більшості інтерактивних дощок);

– відтворення презентацій у формі гіпермедійного документа, призначеного для індивідуального перегляду або для демонстрування групі осіб, які здійснюють аналіз її змісту та обговорення.

Різні способи відтворення презентації передбачають відповідно різні вимоги до стилю презентації. Комп'ютерні презентації, призначені для супроводу лекцій, мають містити якнайменше тексту, адже лектор може розповісти набагато більше. І навпаки, комп'ютерні презентації, призначені для відтворення у формі гіпермедійного документа, можуть містити стільки тексту, скільки потрібно для вичерпного викладу повідомлень.

Тексти слайдів мають бути якомога лаконічнішими, оскільки швидкість читання тексту з екрана в більшості людей невелика. Загальне правило для текстових частин презентації звучить так: "тексту має бути стільки, щоб людина встигла його прочитати і змогла запам'ятати".

Створюючи комп'ютерні презентації, призначені для відтворення на великому екрані, доцільно використовувати шрифти, розміри символів (знаків) яких відповідатимуть умовам видимості для спостерігачів, що перебувають у приміщенні на найбільшій віддалі від екрана (рис. 7.3).

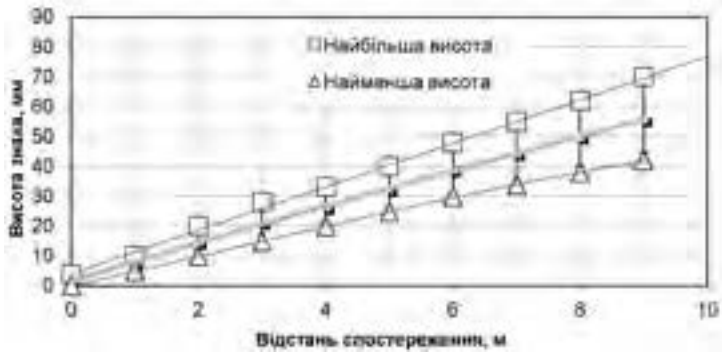


Рис. 7.3. Залежність необхідної висоти найменшого знака від відстані між спостерігачем і екраном.

Перевіряємо себе

1. Чому комп'ютерні презентації, створені з використанням сучасного апаратно-програмного забезпечення, називають мультимедійними?
2. Що таке слайд і презентація?
3. Які складові слайда можуть зберігатися окремо від презентації?
4. Що таке шаблон?
5. Яку роль відіграє кольорова схема презентації?

Виконуємо

Відкрийте набір шаблонів, обговоріть, які з них доцільно застосовувати для презентації роботи краєзнавчого гуртка. Що можна змінити в цих шаблонах? Які елементи додати?

7.3. Використання організаційних діаграм у презентаціях

Організаційна діаграма є графічним відображенням структури керівництва установи, наприклад керівників відділів і працівників у межах установи, структури понять, процесів тощо.

У програмах Microsoft Office 2007–2010 організаційну діаграму можна створити за допомогою графічного об'єкта SmartArt і додати її до презентації або документа.

Щоб вставити організаційну діаграму (схему), потрібно виконати такі дії: *Вставка* → *SmartArt*, після чого вибрати потрібну діаграму.

До кожної діаграми, після її актуалізації, виводиться текст-пояснення.

Після розміщення діаграми на слайді можна вводити текст, змінювати кольори елементів діаграми. Щоб швидко надати рисунку SmartArt ефектного вигляду, можна змінити кольори організаційної діаграми або застосувати до неї стиль SmartArt.

✓ *У презентаціях PowerPoint 2010–2013 до організаційної діаграми можна застосовувати анімацію.*

Щоб додати анімацію до організаційної діаграми, слід виокремити поле, гілку або рівень структури, вибрати організаційну діаграму рисунка SmartArt, до якої потрібно додати анімацію, на вкладці **Анімація** у групі **Додаткові параметри анімації** натиснути кнопку **Додати анімацію** та вибрати потрібний анімаційний ефект.

Щоб змінити анімацію організаційної діаграми так, щоб вона відбувалася за рівнями (наприклад, зробити так, щоб усі фігури на одному рівні з'являлись одночасно, але до фігур нижчого рівня), слід вибрати організаційну діаграму, на вкладці **Анімація** у групі **Анімація** натиснути кнопку **Параметри ефектів** і вибрати пункт **Послідовно за рівнями**.

Перевіряємо себе

1. Що таке структура презентації?
2. Яким чином можна вставити організаційну діаграму?
3. Дайте визначення організаційної діаграми.
4. Які типи макетів можна використовувати для створення організаційної діаграми?
5. З якою метою можна використати макети діаграм групи "**Процеси**"?
6. Чи можна сказати, що організаційна діаграма є моделлю явища, об'єкта, процесу? Обґрунтуйте.

Виконуємо

1. Для поняття "мультимедійна презентація" відобразіть у вигляді ієрархічної діаграми поняття: слайд, об'єкти слайда (підрозділ 7.2), рисунок, текст, відеофрагмент, діаграма.
2. Відобразіть основні поняття молекулярно-кінетичної теорії у вигляді організаційної діаграми. Спробуйте відобразити зв'язки між кожним положенням МКТ і його експериментальним підтвердженням.

7.4. Вбудовані та зв'язані об'єкти в презентаціях

Основу слайда становить набір двовимірних площин (шарів, за термінологією, прийнятою в комп'ютерній графіці, підрозділ 5.1), обмежених розміром екрана або аркуша паперу (вибір робиться через налагодження

параметрів сторінки). Зображення на слайдах може складатися з кількох шарів, об'єкти яких (рисунки і текст) можуть "перекриватись", тобто наступний шар може закривати попередній.

"Ближчий" до спостерігача шар можна зробити **частково прозорим**. Таким чином можна "накладати" на графічне зображення написи, інші рисунки.

✓ *Рисунки (як растрові, так і векторні), мультимедійні об'єкти можуть зберігатись або в файлі презентації, або окремо – на зовнішньому запам'ятовуючому пристрої, ресурсі локальної мережі (сервері), на серверах мережі Інтернет.*

Для того щоб рисунок відтворювався на слайді, необхідно виконати його **вставлення** або **зв'язування** (або вставлення і зв'язування одночасно). Послідовність дій подано на рис. 7.4.

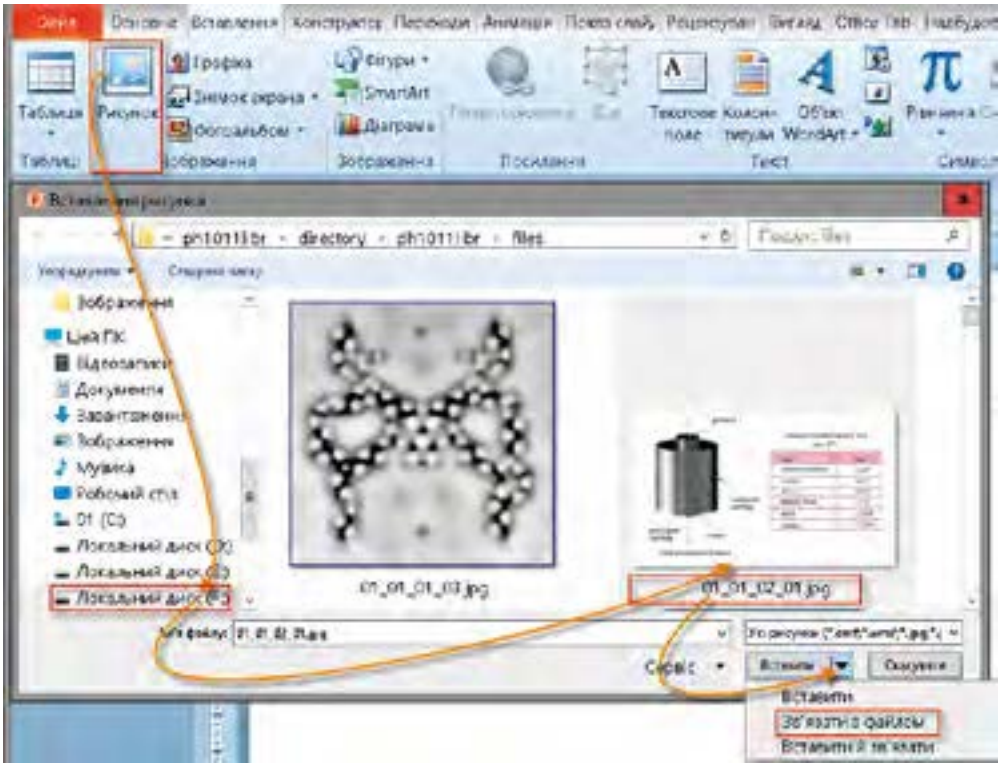


Рис. 7.4. Вставлення в презентацію або зв'язування з нею рисунка

Вставлення використовується тоді, коли рисунок (об'єкт мультимедія) передбачається зберігати в файлі комп'ютерної презентації, після вставлення він стає її частиною.

Якщо потрібно, щоб об'єкт можна було редагувати й відтворювати засобами, незалежними від програми Power Point, або зберігати на сервері мережі, здійснюється його **зв'язування** з файлом презентації.

✓ *Після зв'язування об'єкта з презентацією у файл, у якому зберігається комп'ютерна презентація, вставляється не сам об'єкт, а посилання на нього (ім'я відповідного файла і шлях до нього).*

7.5. Додавання відеокліпів, звукових ефектів і мовного супроводу до слайдової презентації. Елементи анімації

Додавання і зв'язування з презентацією мультимедія (рис. 7.5) виконується так само, як і статичних зображень.

Рис. 7.5. Вставлення у презентацію або зв'язування з нею мультимедія

У процесі демонстрації слайда графічні й текстові елементи можуть з'являтися всі одразу або послідовно. Це залежить від того, чи надано об'єкту ефект анімації. Всі неанімовані об'єкти з'являються на слайді одночасно під час його відкриття. Потім послідовно з'являються об'єкти, для яких встановлено ефекти анімації. Поява цих об'єктів може розпочинатись після певної дії користувача або після того, як мине певний час. Параметри події, яка викликає ефект анімації, встановлюються вибором опцій анімації **(Настроювання анімації)**. Використання великої кількості анімаційних ефектів потребує попереднього ознайомлення з ними, їх перегляду **(Попередній перегляд)**.

Виконуємо

1. Розмістіть на слайді кілька зображень, засобами анімації створіть демонстрацію, на якій вони замінювалися б одне на одне. Пригадайте, де ви вже використовували такий прийом. ★

2. Розмістіть на слайді текст, абзаци якого пронумеровані. Засобами анімації забезпечте послідовне виведення абзаців на екран.

3. Знайдіть, використовуючи *Довідку Power Point*, яким чином здійснити запис звуку з мікрофона. Де та яким чином можна розмістити цей запис?

4. Проаналізуйте рис. 7.6. Що ми бачимо завдяки анімації?

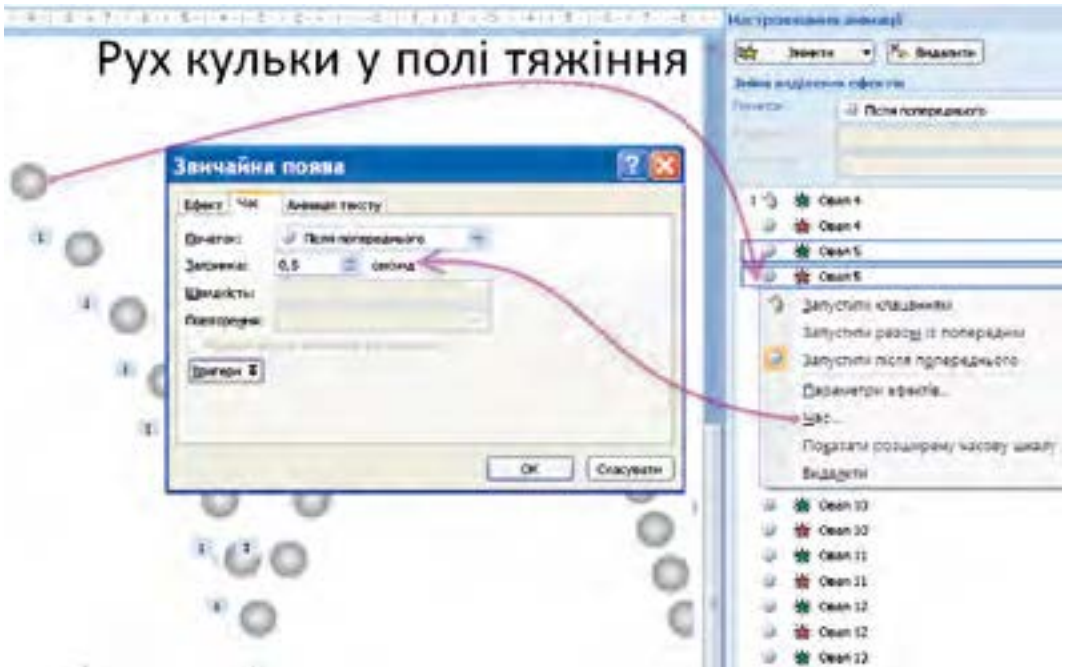


Рис. 7.6. Налагодження анімації для створення ілюзії руху

7.6. Гіперпосилання та елементи керування в презентаціях

Коли презентація демонструється не в автоматичному режимі, а під керуванням людини (доповідач або користувач), послідовність відтворення слайдів може відрізнятися від тієї, в якій слайди зберігаються у файлі презентації.

Система гіперпосилань забезпечує перехід із будь-якого слайда на: попередній слайд; наступний слайд; будь-який слайд презентації; слайд іншої презентації; web-сторінку; вихід на кінець презентації; повернення на початок презентації.

Крім того що з рисунка за подією "*клацання мишею*" здійснюється перехід по гіперпосиланню, для об'єктів, для яких створені гіперпосилання, можливе встановлення "*підказки для гіперпосилання*".

Підказка – текстове повідомлення, яке виникає при наведенні курсора на об'єкт (частина тексту, рисунок, окрема фігура, "кнопка дії" тощо), із якого здійснюються гіперпосилання.

Застосовується і прийом використання гіперпосилань, який називають “гарячі зони на бітовій мапі”. Об’єктом, із яким зв’язане гіперпосилання, оголошується певна область на растровому зображенні (об’єкт на мапі, елемент зображення технічного пристрою тощо). Щоб застосувати цей прийом, на растрове зображення накладають прозорі векторні зображення, які й використовують як об’єкти гіперпосилань. Якщо з цими прозорими зображеннями пов’язати не тільки гіперпосилання, але й підказки для гіперпосилання, то поява підказки вказуватиме на те, що курсор міститься в зоні бітової мапи, з якою пов’язане гіперпосилання.

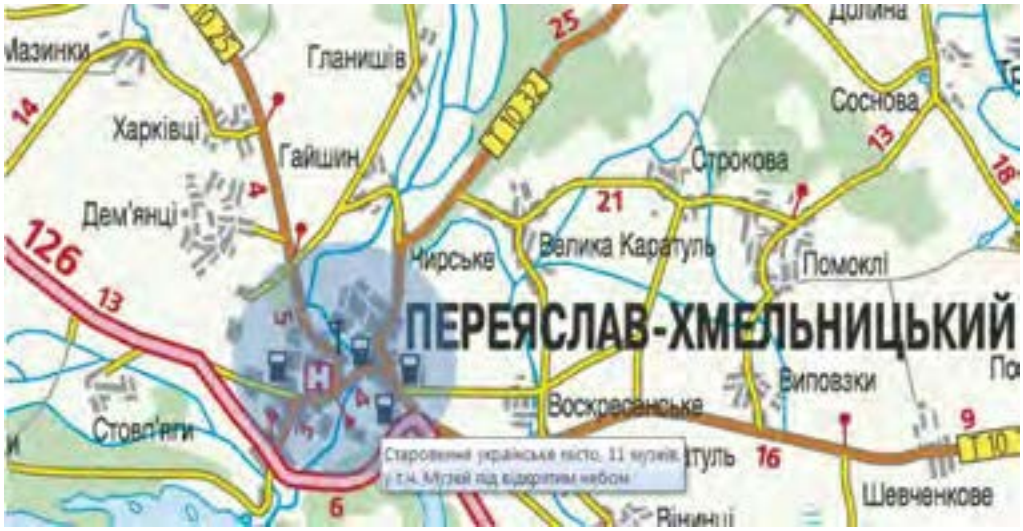


Рис. 7.7. Відображення підказки гіперпосилання, зв’язаного з векторним об’єктом (овал сірого кольору) , що розташований перед растровим (мапою Київської області)

На рис. 7.7 показано, як на растровому зображенні можна створити зони, з якими пов’язані гіперпосилання. Об’єктом, з яким зв’язано гіперпосилання, є *Фігура Овал*, заливка якої навмисне зроблена напівпрозорою. Гіперпосилання здійснюється на слайд, на якому містяться назви всіх 11 музеїв міста. З кожної назви можливий перехід на короткі відомості про музей, кілька світлин. На кожному зі слайдів розміщено кнопки дій, які забезпечують перехід до наступного слайда або повернення на початковий слайд.

Перевіряємо себе

1. Визначте, для яких випадків доцільно використовувати кнопки дій.
2. У яких випадках необхідне покрокове фіксування часу і дій?

3. Чи можна призначити гіперпосилання на електронний документ? Що для цього потрібно?

4. Що потрібно зробити для того, щоб створити гіперпосилання на об'єкти, які зберігаються не у файлі презентації?

Виконуємо

1. Створіть презентацію, яка б забезпечувала виведення на екран зображення за його назвою.

2. Створіть презентацію, яка б забезпечувала виведення на екран тексту, який описує певне зображення (наприклад, виведення відомостей про обласний центр України при натисненні на кнопку миші на його позначенні на мапі).

3. Виконайте роботу з тлом (заливку певним кольором, плавний перехід одного кольору в інший, заливку тла зображеннями-примітивами, завантаженням зовнішнього зображення).

4. Створіть шаблон презентації та майстер-слайд, призначені для створення презентації "Як я відпочивав улітку".

5. Створіть чотири слайди зі структурами: титульного слайда; слайдів для порівняння двох текстів, ілюстративного; з кількома растровими рисунками (використати структури слайдів, подібні до поданих на рис. 7.2), один з рисунків використайте як тло з відповідною його підготовкою).

6. Доповніть слайди графічними елементами (векторними, використовуючи зображення з колекції Microsoft Office і графічні примітиви – Фігури).

Практична робота № 12		
	Тема:	Проектування та розробка розгалужених презентацій за визначеними критеріями. Використання вбудованих і зв'язаних об'єктів у презентації
	Мета:	Набути практичних навичок розроблення розгалужених презентацій

Завдання. Створити презентацію класу.

1. Створіть титульний слайд із позначенням теми та виконавця і керівника роботи.

2. Додайте на титульний слайд зображення (фото) вашого класу.

3. Створіть 10 нових слайдів.

4. Збережіть презентацію та продовжуйте її оформляти.

5. Виберіть оформлення слайдів презентації: розміри, колір та види шрифтів, тло слайдів та інші засоби.

6. На другому слайді створіть зміст презентації. Послідовність виконання:

– наберіть назви розділів (як зразок: Розділ 1. Мої однокласники: ми навчаємось! Розділ 2. Мої однокласники: ми відпочиваємо! Розділ 3. Мої однокласники: мої друзі! Розділ 4. Наш вільний час! Розділ 5. Ми – творчі особистості);

– пронумеруйте їх як список;

– додайте в кінець кожного рядка списку позначку (зображення), яке надалі використовуватимете для переходу на слайд розділу.

7. На третьому слайді наберіть назву першого розділу.

8. Додайте текст, який відповідає його темі.

9. Вставте фотографії та малюнки, що відповідають темі.

10. У папку з файлом презентації помістіть файл із дикторським текстом (до 60 с), який має супроводжувати перший слайд.

11. Розмістіть матеріал, який стосується розділу, на кілька слайдів.

12. Оформіть їх за власним бажанням і баченням.

13. На останньому слайді цього розділу вставте позначку, яка слугуватиме засобом переходу на Слайд 2.

14. Створіть, використовуючи цю позначку, гіперпосилання на Слайд 2.

15. На слайді 2 (у Змісті), навпроти рядка з назвою Розділ 1, використовуючи вставлену раніше позначку, створіть гіперпосилання на слайд, з якого починається Розділ 1 (в нашому випадку це Слайд 3).

16. За цим алгоритмом оформіть декілька слайдів для інших розділів.

17. Продумайте наповнення та створіть слайд, яким завершите презентацію.

Практична робота № 13	**Тема:**	Проектування та розробка розгалужених презентацій за визначеними критеріями. Використання вбудованих і зв'язаних об'єктів у презентації
	Мета:	Набути практичних навичок розроблення розгалужених презентацій

Завдання. Створити презентацію "Працівники цих професій є фахівцями з інформаційних технологій".

1. Створіть перший слайд – основний титульний.

2 На другому слайді наведіть перелік назв професій. Підберіть до нього тло.

3. Обґрунтуйте такий вибір професій.

4. Підготуйте і запишіть власне пояснення як мовний супровід до другого слайда.

5. Розпочинаючи з третього слайда, створіть титульні слайди для кожної професії.

6. Таких слайдів має бути стільки, скільки ви описуєте професій.

7. Слайди мають містити зображення, короткий опис; галузі, де професія використовується.

8. Перехід від слайда до слайда зробіть за допомогою автоматичного зсуву вправо.

9. Тло слайдів виконайте з урахуванням особливостей професії.

10. Після групи титульних слайдів додайте слайди, які ширше показують кожну професію (до кожної професії їх має бути три).

11. Продумайте та оберіть мелодії, які, на вашу думку, можуть супроводжувати кожну професію.

12. Додайте мелодії як звуковий ефект до групи слайдів, які ширше показують кожну професію.

13. На титульні слайди додайте гіперпосилання на перший із трьох слайдів, які стосуються професії.

14. На третьому з цих слайдів розмістіть і налагодьте кнопку повернення на титульний слайд професії.

15. Виконайте такі дії до кожної професії.

16. Створіть слайд, яким завершуватимете екскурсію по професіях.

17. Вставте назву професій. Застосуйте анімацію для появи кожної з них.

18. Додайте декілька зображень, які зроблять наголос на особливостях професій.

19. Застосуйте анімацію до кожного зображення (поява, виділення тощо).

20. Збережіть презентацію.

СЛОВНИЧОК

Анімація – процес переміщення об'єктів на екрані.

Гіперпосилання – виокремлений кольором підкреслений текст або графічний об'єкт, під час активізації якого (при натисканні на кнопку миші на ньому) виконується перехід до файла, фрагмента файла або веб-сторінки в мережі Інтернет.

Майстер-слайд – спеціальний слайд, який використовується не в демонстрації, а для визначення та збереження зразків об'єктів для інших слайдів.

Презентація – захід, який проводиться з метою поширення інформації про певну організацію, подію, товар, послугу.

Слайд – інформаційна структура, яка містить різні об'єкти, які подаються на екрані монітора, аркуші паперу або прозорої плівки у вигляді єдиної композиції;

– використовується для позначення одиниці матеріалів презентації; найменший елемент презентації, яким можна оперувати під час її відтворення.

Стиль презентації – оформлення презентації, яке відповідає змісту презентації, підкреслює спрямованість оповіді, об'єднує слайди у єдиний твір.

РОЗДІЛ 8. ТЕХНОЛОГІЇ ОПРАЦЮВАННЯ ЧИСЛОВИХ ДАНИХ У СЕРЕДОВИЩІ ТАБЛИЧНОГО ПРОЦЕСОРА

Електронні таблиці створюються у спеціальному програмному середовищі, яке також називається електронною таблицею, або табличним процесором. У вільно поширюваному офісному пакеті Libre Office – це електронні таблиці Libre Office Calc (рис. 8.1), а в офісному пакеті Microsoft Office – Microsoft Office Excel (рис. 8.2). Електронний документ, який створюється табличним процесором, називають Книгою. Книга складається з аркушів, вміст одного з яких виводиться на екран. Аркуш відповідно складається з комірок (клітинок). Форматом напису називають описання вигляду та розміру символів, розташування напису відносно поля.

Обчислювальні алгоритми в середовищі табличного процесора; абсолютні, відносні й мішані посилання на комірки та діапазони комірок; формули з використанням посилань на комірки та діапазони; умовне форматування; автофільтр і розширений фільтр; проміжні підсумки; типи діаграм відповідно до мети їх застосування; математичні, статистичні, логічні функції табличного процесора; упорядкування даних за значеннями одного чи кількох полів; шаблони електронних таблиць; підготовка таблиці до друкування.

✓ ***Увага!*** *У цьому розділі деякі команди й назви функцій подаються англійською мовою для того, щоб при переході до нелокалізованих (або частково локалізованих) версій табличних процесорів і їх програмних надбудов не виникали труднощі.*

8.1. Обчислення в середовищі табличного процесора

Властивості об'єктів відображаються у вигляді даних, зв'язки між властивостями об'єктів описуються у вигляді математичних моделей. У електронних таблицях ці моделі подаються через запис у комірки формул.

Дії (арифметичні, порівняння або впорядкування – для числових даних, тільки порівняння або впорядкування – для текстових) виконують над вмістом комірки або вмістом діапазону комірок, а звернення до даних виконують за адресою комірки або діапазону комірок, іменем, тобто за посиланнями на дані.

✓ *Для зручності можна включити режим, у якому ЕТ відображатимуть не результат обчислення формули, а власне формулу.*

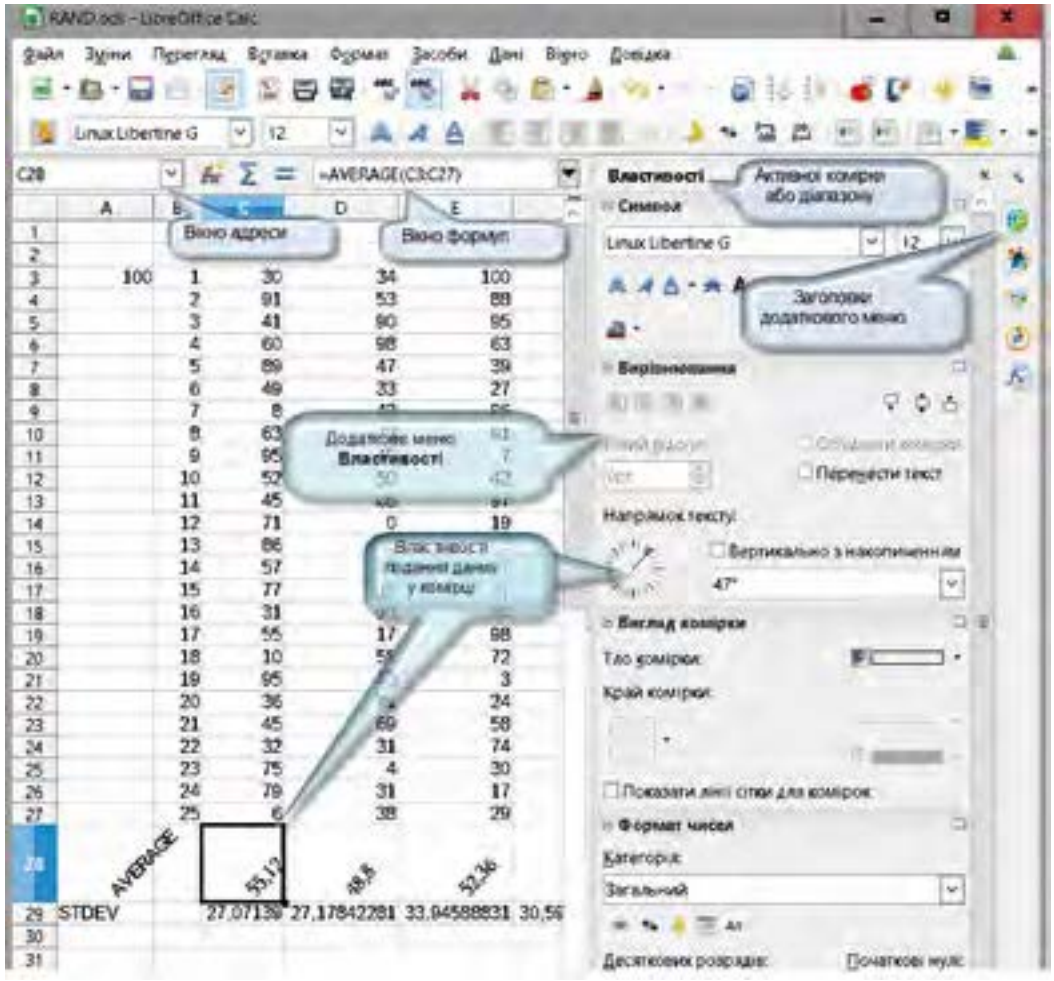


Рис. 8.1. Будова вікна Libre Office Calc

Нехай у комірці **A1** таблиці міститься число 100, а в комірці **B1** – число 20. Щоб поділити перше число на друге й результат помістити в комірку **B1**, у комірку **B1** слід увести формулу **=A1/B1** і натиснути **Enter.**

Уведення формули можна виконати інакше, не вводячи посилань на комірки, а вказуючи на них мишею (під час роботи зі сенсорним екраном – виконуючи одинарний дотик):

*до комірки, у якій має бути розміщено результат обчислень за формулою, ввести знак рівності (=) → клацнути лівою кнопкою миші комірку **A1** → у комірці з'явиться адреса першої комірки формули → ввести знак операції ділення "/" → клацнути на комірці **B1** → у комірці з'явиться адреса другої комірки → натиснути **Enter**.*

У всіх табличних процесорах можливе подання адрес комірок у форматі з нумеруванням як рядків (англ. *row*), так і стовпців (англ. *column*)

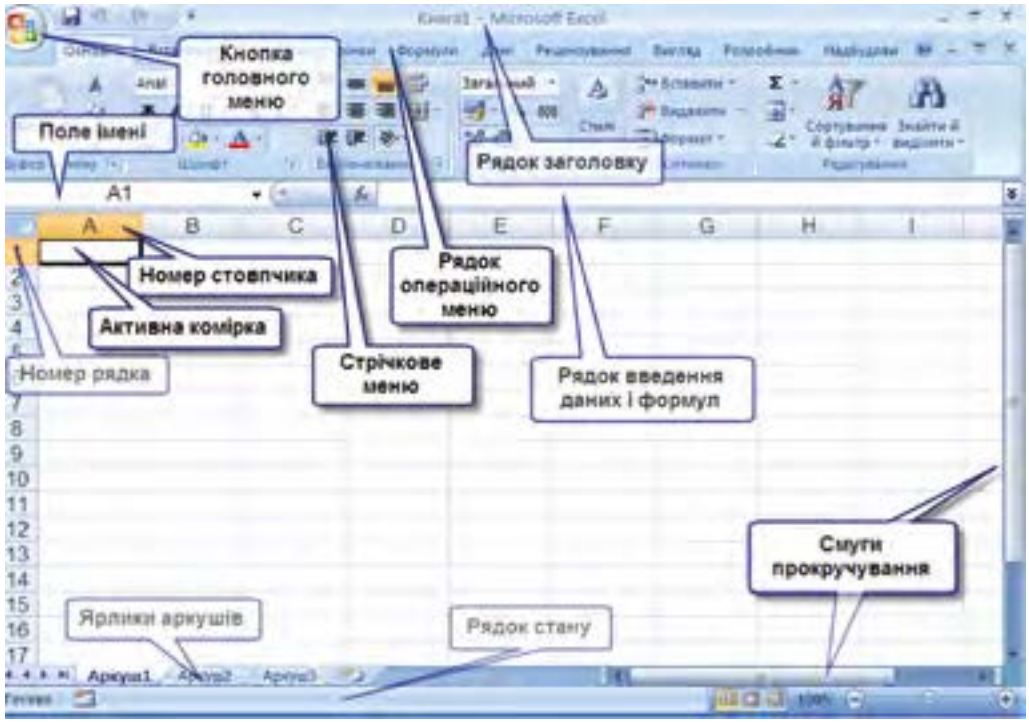


Рис. 8.2. Будова вікна Microsoft Excel 2007–2010

(наприклад, формули матимуть вигляд = RC[-2]-R[-1]C[-1]). Перехід до такого подання показано на рис. 8.3.

Застосування складних формул продемонструємо на прикладі (рис. 8.4). Нехай потрібно обчислити вартість виконання певних робіт.

У стовпці **D** таблиці зазначено час (у годинах), витрачений на виконання роботи, у стовпці **E** – вартість однієї години роботи, а в стовпці **F** – проміжну суму, яку треба сплатити за роботу. У комірці **F6** потрібно показати загальну вартість усіх робіт. Для цього в комірку слід записати таку формулу: **= F3+F4+F5.**

Для обчислення податку на додану вартість отриману суму слід помножити на 0,15 і результат помістити в комірку: **F7: =F6*0,15.**

Для обчислення кінцевої суми, яка підлягає оплаті (наприклад, у комірці **F8**), треба спочатку отримати проміжні суми, а потім результат помножити на 1,15. Формула матиме такий вигляд: **=(F3+F4+F5)*1,15.**

Звичайно, можна було б додати вміст комірок **F6 і F8.** Для додавання кількох чисел можна також використовувати функцію суми **SUM( )**, тоді формула матиме такий вигляд: **=Sum(F3:F5)*1,15.**

Для редагування вмісту комірки (комірок) їх потрібно спочатку активізувати. Далі слід включити режим редагування, натиснувши на клавіатурі клавішу **F2,** або подвійним клацанням лівої кнопки миші. Редагування можна виконувати й у вікні верхньої частини екрана (під

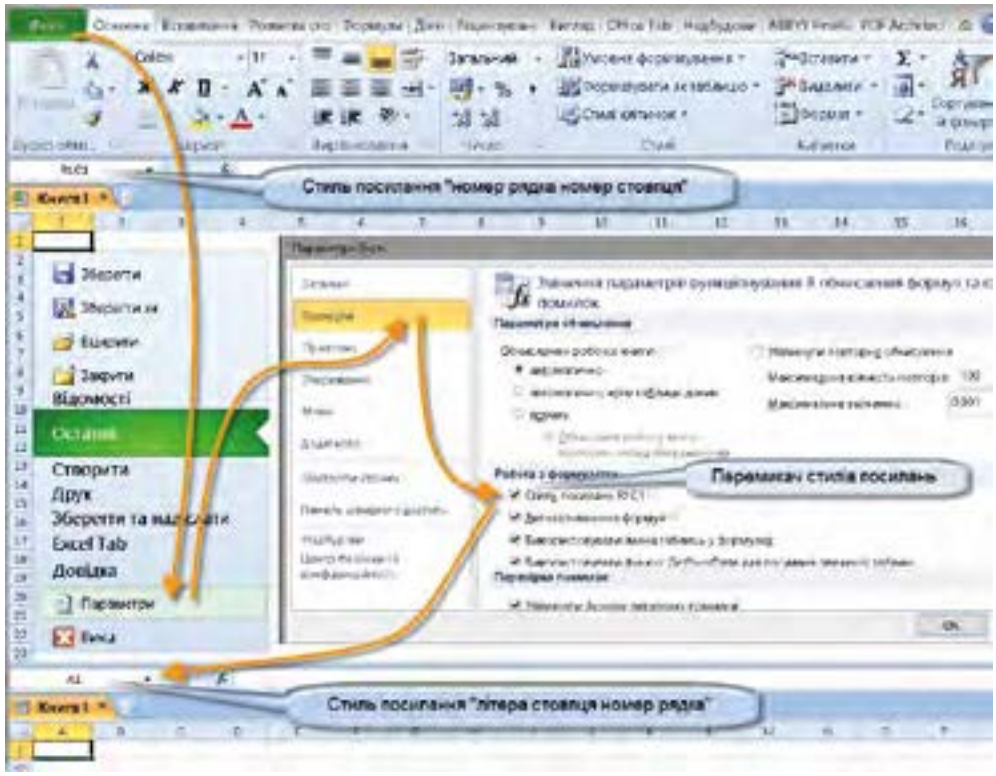
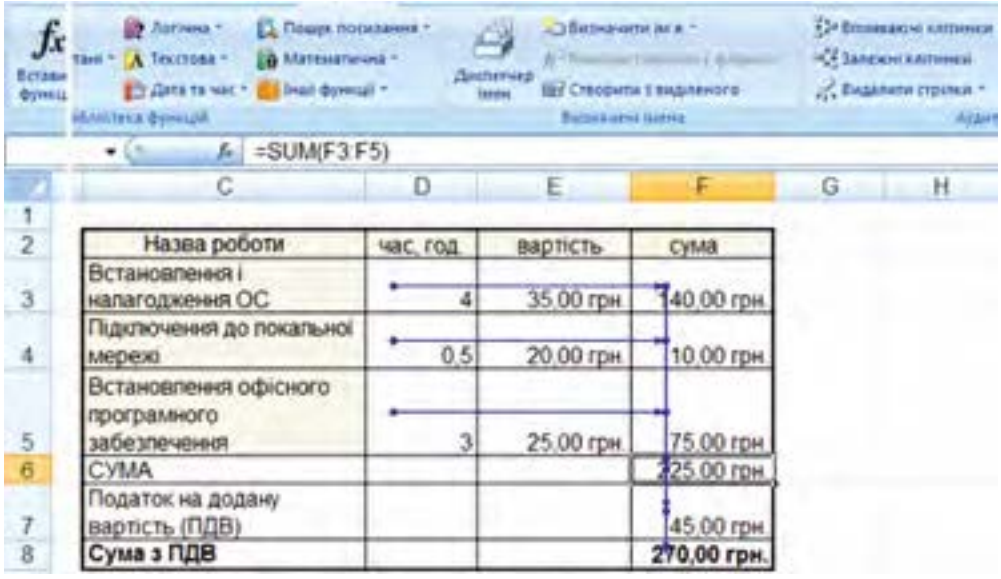


Рис. 8.3. Перемикання форми подання адрес комірок у Microsoft Excel 2010



	C	D	E	F	G	H
1						
2	Назва роботи	час, год	вартість	сума		
3	Встановлення і налагодження ОС	4	35,00 грн	140,00 грн		
4	Підключення до покальної мережі	0,5	20,00 грн	10,00 грн		
5	Встановлення офісного програмного забезпечення	3	25,00 грн	75,00 грн		
6	СУМА			225,00 грн		
7	Податок на додану вартість (ПДВ)			45,00 грн		
8	**Сума з ПДВ**			**270,00 грн.**		

Рис. 8.4. Застосування складних формул для обчислення загальної вартості виконаних робіт

панелями інструментів), у якому подається формула для редагування, а не результат її обчислення.

Для звернення до значення, що міститься в комірці, розташованій на іншому робочому аркуші, потрібно вказати ім'я цього аркуша разом із адресою відповідної комірки. Наприклад, для звертання до комірки **D6** на робочому аркуші **Sheet3** потрібно ввести формулу: **=Sheet3!D6.**

✓ *Для обчислення суми вмісту групи комірок не потрібно називати (перелічувати) в формулі окремі комірки або вводити діапазон у форматі F3:F5. Досить виокремити всю групу та надати їй ім'я.*

Надалі це ім'я можна буде використовувати в формулах. Щоб надати ім'я групі комірок, виконуємо таке (для MS Office 2003):

*меню **Вставка (Insert)** → відкрити підменю **Ім'я (Name)** → викликати директиву **Надати** → у полі введення діалогового вікна **Надання імені (Define Name)** вказати ім'я цієї групи (у це вікно виведено список уже наданих групових імен, які розташовані на аркуші) → натиснути **Додати (Адд)** → **ОК**.*

✓ *Якщо в назві аркуша є пробіли, то назва подається в лапках. Адреси комірок мають бути зазначені латинськими літерами.*

Інформаційне зв'язування двох комірок можна спростити, якщо скопіювати значення вихідної комірки в буфер (за допомогою комбінації клавіш **Ctrl+C**) й актуалізувати комірку, в якій має з'явитися результат. Потім потрібно викликати з меню **Правка (Edit)** директиву **Спеціальна вставка (PasteSpecial),** а відтак у діалоговому вікні натиснути на кнопку **Вставити зв'язок (PasteLink)** (рис. 8.5).

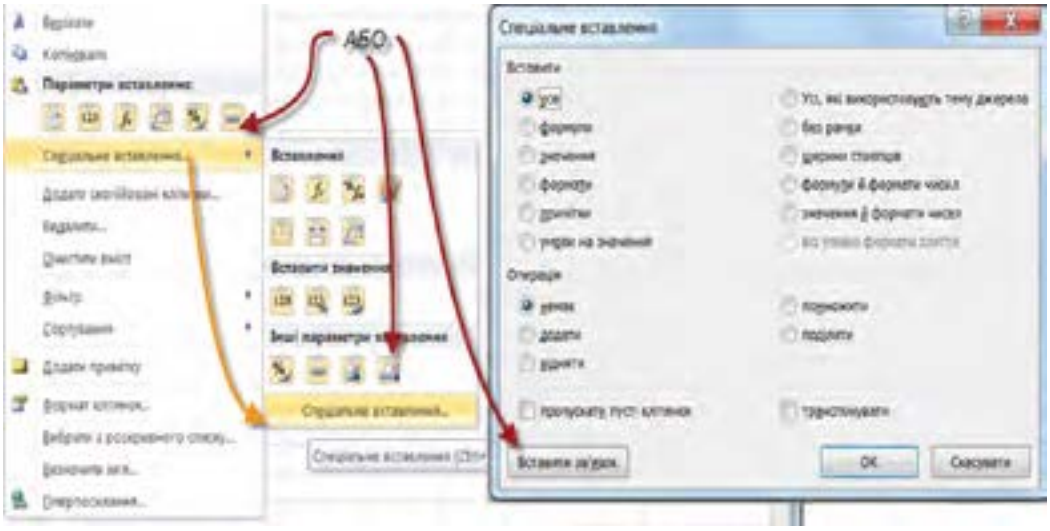


Рис. 8.5. Спеціальне вставлення (MS Office 2010)

✓ *У програмах Excel і Libre Office Calc можна ввести посилання і на комірку, яка розташована в іншій таблиці (іншому файлі).*

Після вставлення посилання значення, які містяться в комірках, будуть автоматично поновлюватись при кожному завантаженні.

Процедура надання імен групам комірок у MS Office 2007–2010 та у Libre Office Calc простіша – досить виділити діапазон і у вікні адреси комірки розкрити список імен, ввести у вікно адреси ім'я групи (рис. 8.6).

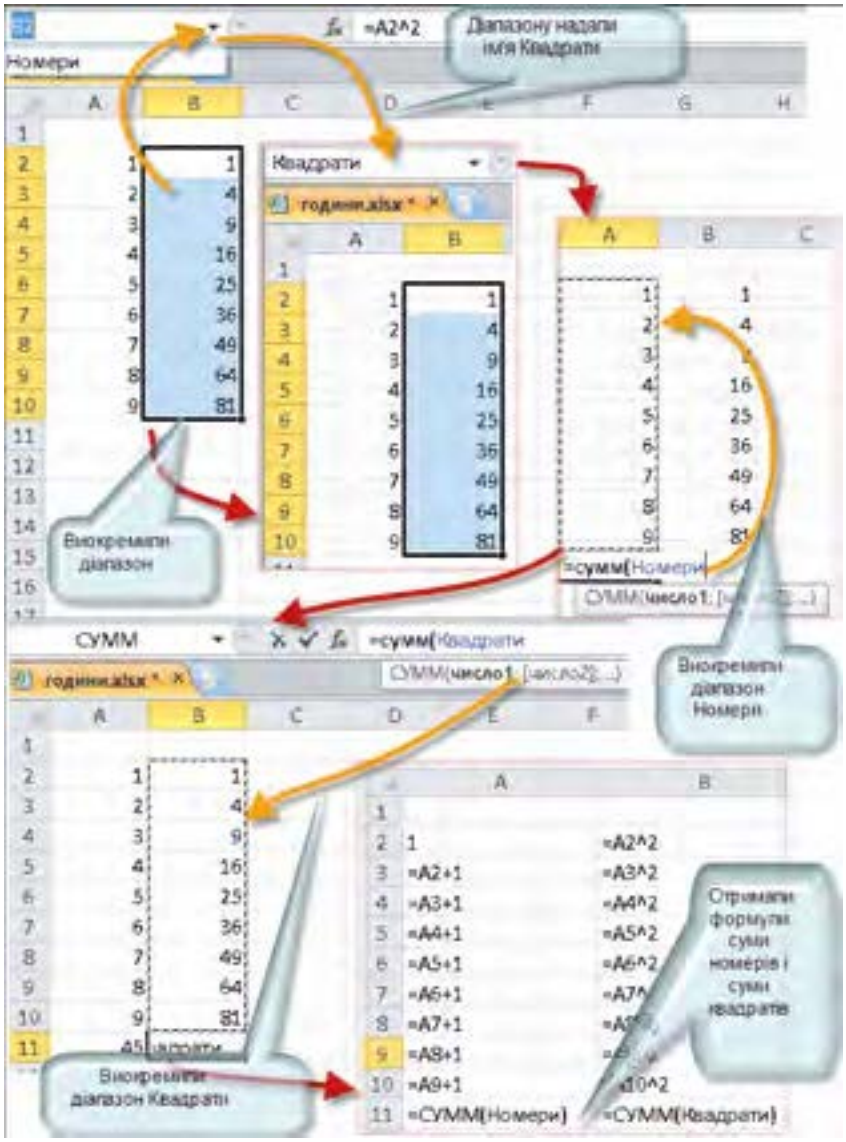


Рис. 8.6. Процедури надання імені групі комірок і використання імен у формулах (MS Office 2010)

У Libre Office Calc ім'я може бути набране тільки латиницею. Ім'я групи має починатися з літери та містити не більше 255 символів.

Не допускається використання пробілів. Ім'я групи не має збігатися з адресами комірок (А1, В6 тощо).

Якщо таблиця містить заголовки рядків і стовпців, то їх також можна використовувати як імена цих діапазонів. Для цього потрібно:

виокремити сусідні рядки (стовпці), включаючи перші комірки, де розташовані імена (заголовки) → меню ***Вставка (Insert)*** *→ відкрити підменю* ***Ім'я (Name)*** *→ викликати директиву* ***Створити (Create)*** *→ у діалоговому вікні, яке відкрилося, вказати місце розташування імен (у першій чи останній комірці рядка чи стовпця) →* ***OK.***

Для редагування функцій потрібно:

двічі клацнути лівою кнопкою миші комірку, в якій міститься функція → в комірці, яка містить результат обчислення, або в ***Рядку формул*** *→ відредагувати →* ***Enter.***

Звернення до вмісту комірок (посилання) можуть бути **відносними, абсолютними** і **мішаними.**

Якщо адресу комірки записати у формулу, наприклад так: А4, або діапазону комірок так: А8:В20, то при копіюванні або перенесенні формули в іншу комірку цю адресу буде змінено – номер стовпців збільшено або зменшено на стільки, на скільки стовпців перенесено формулу, і так само буде змінено номери рядків.

Адреси комірок (і діапазонів комірок), які змінюються при копіюванні (перенесенні) формул, називаються **відносними.**

Якщо потрібно, щоб при копіюванні формул посилання не змінювалися, їх записують так: $A$4, $A$8:$B$20.

Адреси комірок (і діапазонів комірок), які не змінюються при копіюванні формул, називаються **абсолютними.**

Абсолютним є й адресування з використанням імен.

Іноді потрібно, щоб при копіюванні формули змінювалися тільки значення стовпця або рядка. У такому випадку символ "$" розташовують лише перед номером рядка або літерою стовпця, наприклад так: А$29 або $А29. Такі записи адрес є мішаними.

Адреси комірок (і діапазонів комірок), які тільки частково змінюються при копіюванні формул, називаються **мішаними.**

Інколи потрібно в комірки певного діапазону (наприклад, А5:А 25) ввести значення, кожне наступне з яких більше за попереднє на певне значення ("крок", або різниця).

Для цього слід у комірку A5 ввести перше значення, а в комірку A6 – формулу =A5+a, де a – значення різниці. Після цього досить скопіювати вміст комірки A6 у комірки A7:A25, і вони заповняться необхідною послідовністю чисел.

Перевіряємо себе

1. Назвіть основні формати збереження ЕТ табличного процесора Microsoft Excel 2010.
2. Для чого призначено формат *.xml? Знайдіть у Довідці Excel 2010 необхідні відомості.
3. Для чого призначено формати *.xlt та *.xls?
4. До яких програмних засобів можливий експорт електронних документів, створених у Excel 2010?
5. Які відмінності між **Шаблонами** текстового процесора, презентаційної системи і табличного процесора? Чим вони зумовлені?
6. Знайдіть у Довідці Excel 2010 відомості щодо форматування та властивостей ЕТ, створених у застосунку Excel 2010, які не зберігаються у файлах інших форматів.
7. У яких випадках доцільно використовувати різні види посилань у формулах? Наведіть приклади й перевірте їх у таблиці.
8. Яким чином можна надати ім'я несуміжним коміркам? Знайдіть у Довідці Excel 2010 необхідні відомості.
9. Проаналізуйте рис. 8.5 і його опис у тексті. Коли доцільно використовувати кожен із видів вставлення?

Виконуємо

1. Проаналізуйте рис. 8.4 і його опис у тексті. Створіть ЕТ. Які посилання доцільно використовувати в формулах, записаних у комірках F3 – F5? Чому?
2. Модифікуйте створену за попереднім завданням ЕТ таким чином, щоб значення податку на додану вартість можна було оперативно змінювати, вводячи з клавіатури.
3. Модифікуйте створену за попередніми завданнями ЕТ таким чином, щоб значення податку на додану вартість можна було оперативно змінювати і розраховувати для кожного виду роботи окремо.

Підказка: додати стовпчик (комірки G2–G5) із назвою "ПДВ робіт", для зручності використайте абсолютне посилання на комірку зі значенням коефіцієнта для ПДВ.

4. Проаналізуйте рис. 8.5 і його опис у тексті. Як називається група комірок, що відповідає діапазону A2–A10? Яким чином

ці комірки були заповнені значеннями? Як можна назвати числа, розміщені в комірках A11 та B11?

5. Створіть ЕТ, подану на рис. 8.6. Модифікуйте таблицю таким чином: збільшіть кількість рядків до 20; змініть формули у стовпчику A так, щоб можна було отримувати послідовності чисел, які відрізняються не на 1, а на довільну величину (крок).

6. Знайдіть у Інтернеті пояснення особливостей форматів файлів *.ods і *.csv. Занотуйте основні відмінності між форматами файлів. Збережіть створену електронну таблицю в форматах, що відрізняються від того, в якому її подано.

7. Виконайте, якщо можливо, пересилання телефонної книги з мобільного телефону на комп'ютер (у файл *.csv) і відкрийте цей файл текстовим редактором Блокнот, текстовим процесором і табличним процесором. Зробіть висновки.

8.2. Призначення й використання основних математичних функцій табличного процесора

Одиницею зберігання даних у ЕТ є вміст комірки. Дії (арифметичні, порівняння або впорядкування – для числових даних, тільки порівняння або впорядкування – для текстових) виконують над вмістом комірки або вмістом діапазону комірок. Звернення до даних виконують за адресою комірки або діапазону комірок, іменем, тобто за посиланням на дані. Для того щоб ЕТ виконала опрацювання формули, запис формули, який уводиться в комірку таблиці, має починатися зі знака рівності (=).

Оскільки деякі формули та їх комбінації трапляються дуже часто, то табличні процесори містять понад 200 заздалегідь запрограмованих формул, які називаються **функціями**.

Функції *– це наперед визначені формули, за якими ЕТ виконують обчислення в певній послідовності для величин, які називаються* ***операндами****, або* ***аргументами****.*

Аргумент функції може займати одну комірку чи розміщуватися в групі комірок. Є **аргументи** різних типів: число, текст, логічне значення (TRUE та FALSE), масиви, значення помилки (наприклад, #N/A) або посилання на комірку. У кожному окремому випадку потрібно використовувати відповідний тип аргументу. Константи, формули або функції також використовуються як аргументи.

Із функцій можна формувати вирази, в яких операндом однієї з функцій може бути значення, яке повертає інша (такий запис у програмуванні називають "вкладенням").

Для введення функцій використовується команда **Вставка функції**, яка розташована в **Рядку формул (рядку введення даних)**. У результаті натиснення кнопки **Вставка функції** відкривається вікно **Вставка функції**, яке містить упорядкований за алфавітом повний список усіх функцій. При наведенні курсора на ім'я функції внизу списку програма надає її короткий опис.

Якщо функцію не знайдено, її пошук необхідно виконувати за категоріями, клацнувши в рядку **Категорії** або застосувавши команду **Знайти** (рис. 8.7).

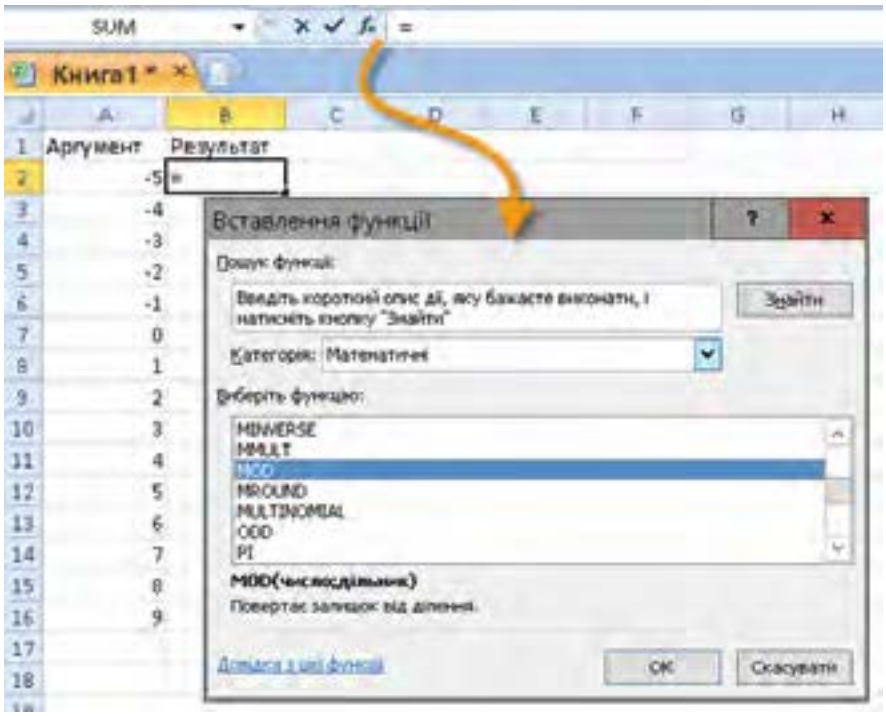


Рис. 8.7. Вікно введення функції

При введенні функції у формулу діалогове вікно **Вставка функції** відображає ім'я функції, всі її аргументи, опис функції та кожного аргументу, поточний результат функції та всієї формули.

Використання вікна **Вставка функції** полегшує введення функцій під час створення формул, які містять функції, та допомагає вставити правильну формулу й потрібні аргументи (рис. 8.8).

✓ *Для введення функції слід: клацнути ім'я функції (під полем* ***Виберіть функцію*** *буде показано синтаксис цієї функції та її короткий опис) → двічі клацнути ім'я функції → функцію та її аргументи буде відображено в вікні* ***Майстра Аргументи функції*** *→ вказати правильні аргументи, користуючись коротким описом та поясненнями, які наводяться під полями аргументів →* ***ОК****.*

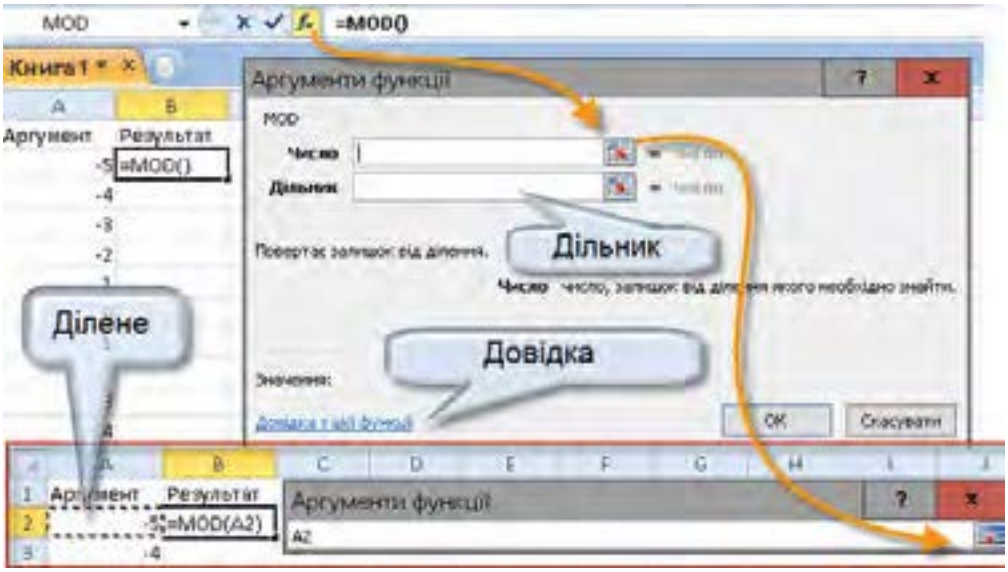


Рис. 8.8. Вікно **Майстра Аргументи функції**

✓ *Для редагування функцій: двічі клацнути лівою кнопкою миші на комірку, в якій міститься функція → в комірці, яка містить результат обчислення, та в* ***Рядку формул*** *текст функції, що містить значення аргументів (тому редагування можна виконати безпосередньо в комірці або в цьому рядку) → відредагувати →* ***Enter.***

Більш докладні відомості про застосування математичних функцій та їх синтаксис можна отримати з Довідки Microsoft Excel. Наприклад, правила заокруглення, які зазвичай застосовують при обчисленнях, у ЕТ Microsoft Excel 2007–2010 доповнено умовами, що можуть накладатися на результат, і створено відповідні формули.

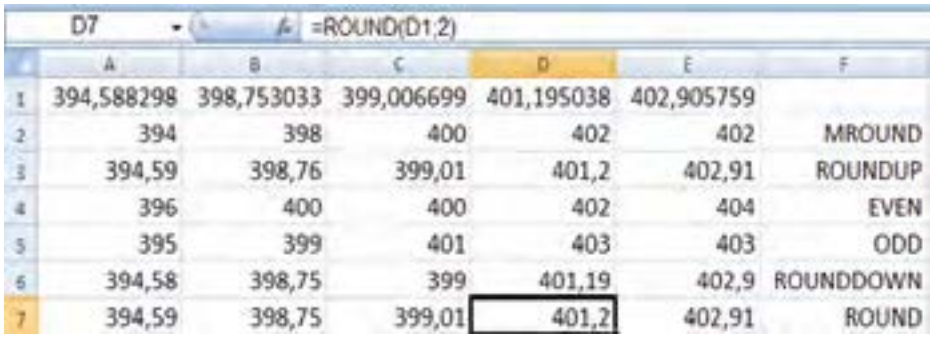
Досить часто доводиться розв'язувати задачі, подібні до такої.

Приклад. Доставили 105 л пального. Його потрібно розмістити у каністрах, ємність кожної з яких становить 20 л. Скільки потрібно каністр?

Відповідь. Для цього знадобиться 6 каністр, в одній із яких міститиметься лише 5 л пального.

Тому, якщо деякий вантаж можна розмістити порціями, не більшими за певну (одне вантажомісце, у задачі, наведеній як приклад – каністра для пального), то навіть для порції вантажу, маса якої значно менша від визначеної для вантажомісця, треба надати окреме вантажомісце, тобто для визначення кількості вантажомісць потрібно виконати "заокруглення вгору" (ROUNDUP).

Застосування деяких формул до чисел показано на рис. 8.9. До чисел першого рядка застосовано формули, назви яких подано в стовпці F. Формули застосовано в рядках 2...8.



D7 =ROUND(D1;2)

	A	B	C	D	E	F
1	394,588298	398,753033	399,006699	401,195038	402,905759	
2	394	398	400	402	402	MROUND
3	394,59	398,76	399,01	401,2	402,91	ROUNDUP
4	396	400	400	402	404	EVEN
5	395	399	401	403	403	ODD
6	394,58	398,75	399	401,19	402,9	ROUNDDOWN
7	394,59	398,75	399,01	401,2	402,91	ROUND

Рис. 8.9. Приклад застосування формул, призначених для заокруглення чисел

Аргументи функцій, так само як і простих формул, можуть розташовуватися на різних аркушах книги. Наприклад, кілька постачальників постачають однаковий товар, але для кожного з них облік постачання ведеться на окремому аркуші, на якому також зберігається ціна товару від постачальника. Потрібно знайти загальну кількість товару і його вартість, якщо облік постачання товару в натуральному вимірі ведеться на різних аркушах: **Пост_1**, комірки **B2:E9; Пост_2**, комірки **B2:E9; Пост_3**, комірки **B4:E12**, а ціни зберігаються в комірках **Пост_1!A10, Пост_2!A10, Пост_3!A10**.

Вираз:

=SUM(SUM(Пост_1!B2:E9)*Пост_1!A10;SUM(Пост_2!B2:E9)*Пост_2!A10;SUM(Пост_3!B4:E12)*Пост_3!A10) поверне значення загальної вартості товару від трьох постачальників.

Особливим випадком аргументу є масив даних.

Масивом називають сукупність однотипних величин, кожен елемент якої нумерований.

*Масив є структурою даних. Звернення до окремого значення (**елемента** масиву) здійснюється за його номером, в ЕТ його значення розміщують у блоці суміжних комірок.*

Формули, в яких аргументами є масиви, можуть описувати дії упорядкування елементів, арифметичні дії над елементами кількох масивів, пошуку елементів за певними ознаками.

Можливе використання формул, результатом обчислення яких є не одне значення, а масив (сукупність значень).

Можна написати й власні формули, що застосовуються до діапазонів комірок, результатом обчислення яких буде діапазон комірок. Наприклад, **=F4:F9−G4:G9**.

Для введення подібних формул (формул масиву): виокремте діапазон комірок, що мають містити результати обчислення формули масиву

*(розмір виокремленого діапазону має відповідати кількості значень, що створюються формулою) → натисніть клавішу **F2** → введіть потрібну формулу, вказуючи посилання на діапазони комірок, що мають використовуватися в обчисленнях → завершіть уведення формули натисканням сполучення клавіш Ctrl+Shift+Enter (а не натисканням кнопки ОК у підменю вибору даних!).*

Таблиця 8.1

Деякі математичні функції електронних таблиць

Назва функції	Опис
1	2
ABS()	Повертає абсолютне значення числа
CEILING() ОКРУГЛТ()	Заокруглює число до найближчого цілого і кратного вказаному значенню
COS()	Повертає косинус числа
DEGREES() ГРАДУСЫ()	Перетворює радіани на градуси
EXP()	Повертає число *e*, піднесене до вказаного степеня
FACT()	Повертає факторіал числа
FLOOR() ОКРУГЛВНИЗ()	Заокруглює число до меншого за модулем, у напрямку нуля
GCD() НОД()	Повертає найбільший спільний дільник
INT()	Заокруглює число до найближчого меншого цілого
LCM() НОК ()	Повертає найменше спільне кратне
MOD() ОСТАТ()	Повертає остачу від ділення
PI()	Повертає число π
POWER() СТЕПЕНЬ()	Повертає число, піднесене до степеня
RADIANS() РАДИАНЫ()	Перетворює градуси на радіани
RAND() СЛЧИС()	Повертає випадкове число в інтервалі від 0 до 1
RANDBETWEEN() СЛУЧМЕЖДУ()	Повертає випадкове число в указаному інтервалі
ROUND() ОКРУГЛ()	Заокруглює число до вказаної кількості знаків
ROUNDDOWN() ОКРУГЛВНИЗ()	Заокруглює число до меншого, у напрямку нуля

Закінчення табл. 8.1

1	2
CEILING() ОКРУГЛВВЕРХ()	Заокруглює число вгору, у напрямку від нуля
SIGN() ЗНАК()	Повертає знак числа (1, якщо число більше нуля, 0, якщо число 0, -1, якщо число менше нуля)
SIN()	Повертає синус кута (значення кута подається в радіанах)
SQRT() КОРЕНЬ()	Повертає невід'ємне значення квадратного кореня
SUM() СУММ()	Підсумовує аргументи
SUMIF() СУММЕСЛИ()	Підсумовує вміст комірок, визначених за вказаною умовою
SUMIFS()	Підсумовує вміст комірок у діапазоні, який відповідає кільком умовам
SUMPRODUCT() СУММПРОИЗВ()	Повертає суму добутків відповідних елементів масиву
SUMSQ() СУММКВ()	Повертає суму квадратів аргументів
TAN()	Повертає тангенс числа
TRUNC() ОТБР()	Видаляє дробову частину числа

Програма Excel помістить формулу в фігурні дужки, що є ознакою формули масиву. У комірках виокремленого діапазону будуть представлені результати обчислення формули.

Табличний процесор завжди інтерпретує масив як одне ціле та не дозволяє змінити окремі комірки масиву. Проте можна вказати для окремих комірок різні параметри форматування (в тому числі використати умовне форматування). Комірки не можуть бути переміщені з масиву, а нові комірки – додані в масив.

Перевіряємо себе

1. Які типи значень можуть бути операндами математичних функцій?
2. Які математичні функції можна застосувати до числа –1?
3. Чим відрізняються *Грошовий* і *Фінансовий* формати подання чисел?
4. Чи є масивом іменована група комірок, у яких містяться як числові значення, так і текст?
5. У яких випадках доцільно використовувати автоматичне заповнення комірок?

Виконуємо

1. Знайдіть (використовуючи Довідку Microsoft Excel або довідкову систему іншого табличного процесора, з яким ви працюєте) у табл. 8.1 усі функції, які не потребують операндів.
2. Знайдіть (використовуючи Довідку) в табл. 8.1 усі функції, які потребують тільки одного операнда.
3. Знайдіть (використовуючи Довідку) в табл. 8.1 усі функції, які потребують тільки двох операндів.
4. Знайдіть (використовуючи Довідку) в табл. 8.1 усі функції, які потребують не менше двох операндів.
5. Створіть ЕТ, за допомогою якої можна обчислити значення квадратного тричлена за його коефіцієнтами.
6. Створіть таблицю, яка обчислює арифметичні значення коренів квадратних чисел натурального ряду від n до m, подаючи їх із точністю до трьох десяткових знаків.
7. Створіть ЕТ для обчислень значень функції $y = 2x^2 - 5x + 10$ в інтервалі значень x від -10 до 10 з кроком 1. Для створення масиву значень x використайте автоматичне заповнення комірок. Як зробити так, щоб крок послідовності (різницю) можна було змінювати без редагування формули?

8.3. Призначення й використання основних логічних функцій табличного процесора

Для опрацювання даних із вибором способу опрацювання використовують логічні вирази, тобто формули, які описують перевірку умови (або кількох умов). Значень, яких можуть набувати такі вирази, лише два: істинне або хибне (англ. *true* або *false*), так або ні, 1 або 0. Наприклад, вираз C1>10 матиме значення *true*, якщо в комірці C1 міститиметься число більше 10, і *false* в усіх інших випадках.

Логічні вирази використовують для визначення дій, які виконуються після перевірки істинності певної умови, наприклад, формула **=IF(G7<>0;F7/G7;“на нуль ділити не можна”)**, яку записано у комірку H7, означає, що вміст комірки F7 буде поділено на вміст комірки G7, якщо та містить ненульове значення або не є порожньою, результат обчислення буде записаний в комірку H7, інакше в комірку H7 запишеться текст “на нуль ділити не можна”.

Подібні вирази ви вже використовували, записуючи алгоритмічні структури розгалуження у навчальному середовищі програмування Scratch і описуючи їх мовою програмування Паскаль.

Очевидно, якщо в функції IF (умова; результат1; результат2) заповнено крім умови обидва місця для результатів, отримуємо структуру розга-

луження першого типу (рис. 8.10), тобто повне розгалуження. Якщо заповнено тільки місце для умови й першого результату, то отримуємо структуру “розгалуження” другого типу – неповне розгалуження (рис. 8.11).

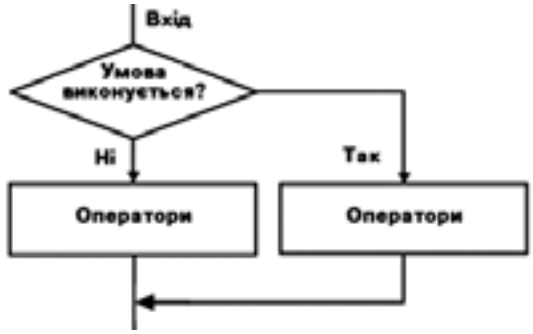


Рис. 8.10. Структура “розгалуження” першого типу (повне розгалуження)

Рис. 8.11. Структура “розгалуження” другого типу

Формули, які повертають логічні значення, також допускають використання вкладення.

Приклад 1. Якщо потрібно створити таблицю для відображення накопичення значень певного параметра для кількох об’єктів, а потім визначити відповідність накопичених сум певним критеріям, можна скористатись виразом: **=IF(SUM(D15:M15)<10;$E$17;IF(SUM(D15:M15)<50; $E$18;IF(SUM(D15:M15)<100;$E$20))).**

У комірках **D15:M15** містяться значення параметра (наприклад, оцінки, отримані протягом семестру), а в комірках **E17, E18, E20** – назви груп, до яких належать накопичені значення.

Створення таблиць, призначених для опрацювання масивів даних, отриманих для груп іменованих об’єктів (наприклад, розташованих у різних містах філій банку), може спростити застосування функції SUMIFS, яка виконує додавання значень із використанням певної умови.

Приклад 2. Нехай у стовпці С розташовані значення прибутку філій установи, а у стовпці В – назви міст, у яких вони розташовані. Формула **=SUMIFS (B4:B30;“Черкаси”;C4:C30)** поверне суму прибутків тільки тих філій, які розташовані в м. Черкаси.

Приклад 3. Нехай *n* учнів складали 10 тестів, відповідь на кожен з яких можна оцінити від 0 до 10 балів. Треба створити ЕТ для визначення результату тестування кожного учня з використанням оцінок “недостатньо”, “задовільно”, “добре”, “відмінно”, якщо відомо, що ці оцінки виставляють при сумах набраних балів: до 30, від 31 до 50, від 51 до 70, від 71 до 100 відповідно.

З цією метою доцільно застосувати математичні та логічні функції, як це показано на рис. 8.12.

Рис. 8.12. Електронна таблиця для підведення підсумків тестування

Формулу (вираз), яку використано для віднесення результатів тестування певної особи до однієї з чотирьох груп, подано в формі, яка передбачає можливість окремого введення значень критеріїв (меж інтервалів) і назв груп (з цією метою використано посилання на комірки N4:N7).

Формулу (рис. 8.12)

=IF(SUM(B4:K4)<30;$O$4;IF(SUM(B4:K4)<50;$O$5;IF(SUM(B4:K4)<70;$O$6;$O$7)))

можна подати у вигляді опису алгоритму :

якщо SUM(B4:K4)<30
    то оцінка = $O$4
    інакше якщо SUM(B4:K4)<50
        то оцінка = $O$5
        інакше якщо SUM(B4:K4)<70
            то оцінка = $O$6
            інакше назва інтервалу = $O$7

Перевіряємо себе

1. З якою метою замість явного подання (Приклад 3) назв інтервалів у формулі використано посилання? Чому ці посилання мають форму абсолютних посилань?
2. Яка частина (частини) формули описує критерії віднесення значення до певного інтервалу?
3. Як можна розширити список учнів (рис. 8.12)?
4. Які зміни потрібно внести до формули, для того щоб можна було оперативно змінювати критерії віднесення значень до певного інтервалу?

5. Чому у формулі жодного разу не використано значення нижньої межі інтервалів? ★

6. Яку оцінку “поставить” алгоритм, якщо значення у стовпчику L перевищуватимуть 100? Що треба зробити, щоб уникнути такої помилки? ★

Виконуємо

1. Спробуйте описати на словах і побудувати графічне подання алгоритму, описаного формулою **=IF(SUM(D15:M15)<10;$E$17;IF(SUM(D15:M15)<50;$E$18;IF(SUM(D15:M15)<100;$E$20))).** ✦

2. Перепишіть алгоритм, передбачивши надання назві інтервалу значення без використання посилання на комірку. Яка частина таблиці після цього перестане бути необхідною? ⅄

3. Створіть таблицю для обчислення найменшого спільного кратного двох чисел. ★

4. Створіть таблицю для обчислення найбільшого спільного дільника двох чисел. ✦

5. Створіть таблицю, щоб перевірити, чи можна побудувати трикутник, заданий значеннями довжин трьох сторін. Використайте функцію SUM і логічний вираз. ★

8.4. Призначення й використання основних статистичних функцій табличного процесора

Опрацювання великих наборів даних і отримання на основі їх аналізу відомостей щодо об’єкта або сукупності об’єктів – одна з основних задач науки статистики.

✓ *Потреба у статистичному аналізі даних виникає тоді, коли необхідно визначити, чи пов’язані між собою дві або кілька величин, подій (як впливає паління тютюну на виникнення певних хвороб у людини, яким буде врожай зернових, якщо протягом зими випала певна кількість снігу тощо), спрогнозувати певні явища в природі (передбачити погоду, знайти родовище корисних копалин тощо), спрогнозувати розвиток виробництва.*

З цією метою майже завжди потрібно досліджувати великі обсяги даних. Найпростішим прикладом є завдання: визначити межі змін розміру деталі, виробленої автоматичною лінією. Для цього відбирають досить велику (300...1000 штук) кількість готових деталей з партії, вимірюють їх розміри. Для розміру (або кількох розмірів), важливого

для наступних етапів процесу виготовлення та подальшої експлуатації виробу, частиною якого буде деталь, складають таблицю.

✓ *Статистичне опрацювання отриманих даних має відповісти на такі запитання:*

– "Яке середнє значення розміру деталі?"

– "У яких межах змінюється цей розмір?"

– "З якою надійністю можна гарантувати, що розмір деталі, взятої довільно з партії, міститься в певних межах?"

– "Скільки деталей певного розміру є в партії?"

Приблизно такі самі запитання можна поставити, досліджуючи дані соціологічних опитувань (скільки людей проголосують на виборах за певну партію), дані спостережень за популяціями тварин, навіть опрацьовуючи дані, отримані під час виконання лабораторних робіт з фізики, хімії тощо.

Тому для подібних досліджень використовують методи, які в ЕТ описано у вигляді статистичних функцій. Статистичні функції ЕТ забезпечують можливість опрацювання результатів фізичних, хімічних експериментів, соціальних досліджень. У результаті такого опрацювання великих масивів однотипних даних можна знайти взаємозв'язок між величинами, прогнозувати значення даних у залежностях, тобто визначити тенденцію розвитку певного процесу.

Статистичні функції ЕТ забезпечують можливість опрацювання результатів фізичних, хімічних експериментів, коли потрібно опрацювати великий масив однотипних даних, визначити взаємозв'язок між величинами, виконати статистичний аналіз даних, спрогнозувати наступні значення даних у деякому діапазоні величин, тобто визначити тенденцію розвитку певного процесу тощо.

Для обчислення середнього значення існує статистична функція **СРЗНАЧ.** (англ.: **AVERAGE**), яка має такий вигляд:

AVERAGE(число1; число2;...), де **число1, число2, ...** – це від 1 до 255 аргументів, для яких обчислюється середнє.

*А*ргументами можуть бути лише числа або імена, масиви або посилання, ***які містять числа,*** при цьому порожні комірки функція не враховує, але комірки з нульовими значеннями беруть участь у обчисленнях.

Якщо аргумент масиву чи посилання містить текст, логічні значення або порожні комірки, то ці значення не враховуються. Аргументи, які є значеннями помилки або текстом, який не можна перетворити на числа, призводять до помилки.

Для обчислення найбільшого (максимального) значення даних, що містяться в діапазоні комірок, слугує функція **МАКС** (англ. **MAX**).

Функція має такий вигляд:

МАХ(число1;число2;…) де число1, число2, … – від 1 до 30 чисел, серед яких визначається максимальне значення.

Функція може опрацьовувати не тільки дані, подані у вигляді чисел, можна задавати аргументи також у вигляді логічних значень або текстовими поданнями чисел.

Під час роботи з масивами функція опрацьовує тільки числа.

Приклад. У таблиці в діапазоні комірок **A1:A6** міститься послідовність чисел {4; 5; 2; 3; 4; 5; 2; 3}. Визначити максимальне значення числа в цій послідовності. Для цього створюємо формулу = **МАКС(A1:A6)**, яка повертає число 5.

Для обчислення мінімального значення з певного діапазону значень використовують функцію **МИН** (англ. **MIN**). Синтаксис функції аналогічний синтаксису функції **МАКС**.

Для визначення номера певного найбільшого значення на заданому діапазоні комірок зручно використовувати статистичну функцію **НАИБОЛЬШИЙ** (англ. **LARGE**). Функція визначає *i*-те найбільше значення з множини даних, наприклад, функцію можна застосувати для визначення першого, другого й третього місць серед учнів класу, які брали участь у змаганнях.

Функція має такий вигляд:

НАИБОЛЬШИЙ(масив; *i*), де **масив** – діапазон даних, серед яких потрібно визначити *i*-те найбільше значення; *i* – позиція в діапазоні даних.

Різницю між значеннями, що повертають функції **МАКС** *і* **МИН** *для вибірки, називають* ***“розмахом вибірки”***.

У вибірці (наборі значень) **мода** *– це значення, що найбільш часто трапляється;*

медіана *– це значення, рівновіддалене від максимального і мінімального значень;*

середнє *– це середнє арифметичне значення.*

Жодне з цих чисел не характеризує повною мірою те, як розташовані дані на числовій осі. Уявімо, що дані згруповані в трьох областях, одна половина даних близька до деякого малого значення, а друга половина – до двох інших великих значень. **Медіана** і **середнє** значення при цьому будуть близькі до порожньої середини, а мода, найімовірніше, дорівнюватиме найменшому значенню, що часто трапляється.

Для обчислення медіани використовують функцію **МЕДИАНА** (англ. **MEDIAN**).

Функція має вигляд **MEDIAN** (*число1;число2*;…), де ***число1; число2***; … – від 1 до 255 аргументів, для яких потрібно знайти медіану.

✓ *Якщо кількість чисел у масиві парна, функція **MEDIAN** обчислює середнє з двох чисел, які розташовані посередині.*

Приклади: МЕДИАНА(1; 2; 3; 4; 5) дорівнює 3, **МЕДИАНА**(1; 2; 3; 4; 5; 6) дорівнює 3,5, середнє арифметичне 3 і 4.

Мода обчислюється за допомогою функції **МОДА** (англ. **MODE**)

Якщо набір даних не містить однакових даних, то функція **МОДА** повертає значення помилки #Н/Д (немає даних).

Приклад. MODE ({5,6; 4; 4; 3; 2; 4}) дорівнює 4.

Подібним чином працює і функція **РАНГ** (англ. **RANK**), але за її допомогою визначають місце (ранг) значення у масиві значень. Повертає числове значення, яке є місцем аргументу в списку чисел. (Якщо впорядкувати список, то ранг числа дорівнюватиме його позиції.)

Функція має вигляд **RANK(***число; посилання; порядок***)**, де *число* – число, ранг якого потрібно визначити; *посилання* – посилання на список чисел. Нечислові значення в посиланні ігноруються; *порядок* – параметр, що визначає, як розподіляються порядкові номери.

Якщо порядок дорівнює 0 (нулю) або не вказаний, функція визначає ранг числа, спираючись на припущення, що посилання є списком, відсортованим за спаданням. Якщо порядок має будь-яке ненульове значення, функція визначає ранг числа на основі припущення, що посилання є списком, відсортованим за зростанням.

✓ *Функція **RANK** призначає числам, що повторюються, однаковий ранг. Це впливає на ранги наступних чисел. Наприклад, якщо у списку цілих чисел, відсортованих за зростанням, число 10 трапляється двічі й має ранг 5, то число 11 матиме ранг 7 і жодному числу не буде призначено ранг 6.*

Наприклад, після проведення змагань отримано список, який містить прізвища спортсменів і суми балів, здобуті ними з усіх вправ. Необхідно визначити місце, яке посів кожний спортсмен (рис. 8.13). Зверніть увагу, друге й сьоме місця присвоєні двом спортсменам, а третє й восьме місця – нікому.

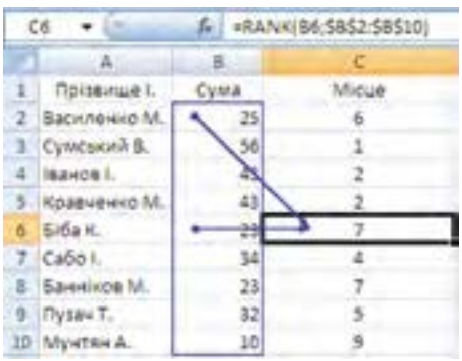


Рис. 8.13. Приклад застосування функції РАНГ

Уявімо, що деяка фізична величина змінюється так, що в кожний момент часу набуває певного значення. Дослідникові не відомо, чи є в появі певних значень величини якась закономірність. Отже, потрібно насамперед з'ясувати, які значення величини з'являються частіше за інші, від чого залежить їх поява.

Подібні задачі виникають і тоді, коли потрібно оцінити якість партії

приладів, призначених для вимірювань певної величини. Найпростіше це зробити, якщо виміряти одну й ту саму (відому наперед з високою точністю) величину кілька разів різними екземплярами цього приладу.

Для опису подібних задач та їх дослідження засобами електронних таблиць використовують функцію **ЧАСТОТА** (англ. **FREQUENCY**).

Функція має такий вигляд: **FREQUENCY**(*масив_даних; масив_інтервалів*), де *масив_даних* – масив або посилання на набір значень, для яких потрібно обчислити частоти. Якщо *масив_даних* не містить жодних значень, формула **FREQUENCY** повертає масив із нулів; *масив_інтервалів* – масив або посилання на сукупність інтервалів, за якими потрібно згрупувати значення масиву даних. Якщо *масив_інтервалів* не містить жодних даних, функція **FREQUENCY** повертає кількість елементів масиву даних.

Функція повертає масив, елементи якого дорівнюють кількості елементів даних, значення яких потрапляють у визначені інтервали – частоти. Кількість елементів у масиві частот на один більша за кількість елементів у масиві інтервалів. Додатковий елемент масиву містить кількість значень, які перевищують верхню межу інтервалів. Наприклад, під час обчислення трьох діапазонів (інтервалів) значень, уведених у три комірки, для відображення результатів потрібно ввести функцію **FREQUENCY** в чотири комірки. У додаткову комірку функція **FREQUENCY** повертає кількість значень у масиві даних, які перевищують значення верхньої межі третього інтервалу, якщо такі існують.

Формули, які повертають значення у вигляді масивів, слід уводити як формули масивів.

Відношення частоти певної події до всієї кількості подій називають **відносною частотою події**. Якщо подію визначити як "знаходження значення з масиву даних у певному діапазоні", то сума всіх відносних частот подій завжди дорівнюватиме 1, або 100 %.

На рис. 8.14 подано приклад аналізу даних, які розміщені в комірках **A2:W2**. Треба було визначити, як ці значення розподіляються в інтервалах 0...20; 21...40; 41...60; 61...80; 81...100, значення верхніх меж яких занесене в комірки **A4:A8**. Оскільки наперед було відомо, що значення, менші за 0, і значення, більші за 100, неможливі, то для масиву результатів виділено п'ять комірок – **B4:B8**. Для комірки **B4** засобами ЕТ позначено комірки, що впливають на її значення: це діапазон даних **A2:W2** і комірка **A4**, в якій міститься значення верхньої межі першого інтервалу. Для контролю та наступного обчислення відносних частот у комірці **B9** уведено формулу **=SUM(B4:B8)**. Значення, яке повертатиме ця формула, має дорівнювати кількості елементів масиву даних, розміщеного в комірках **A2:W2**.

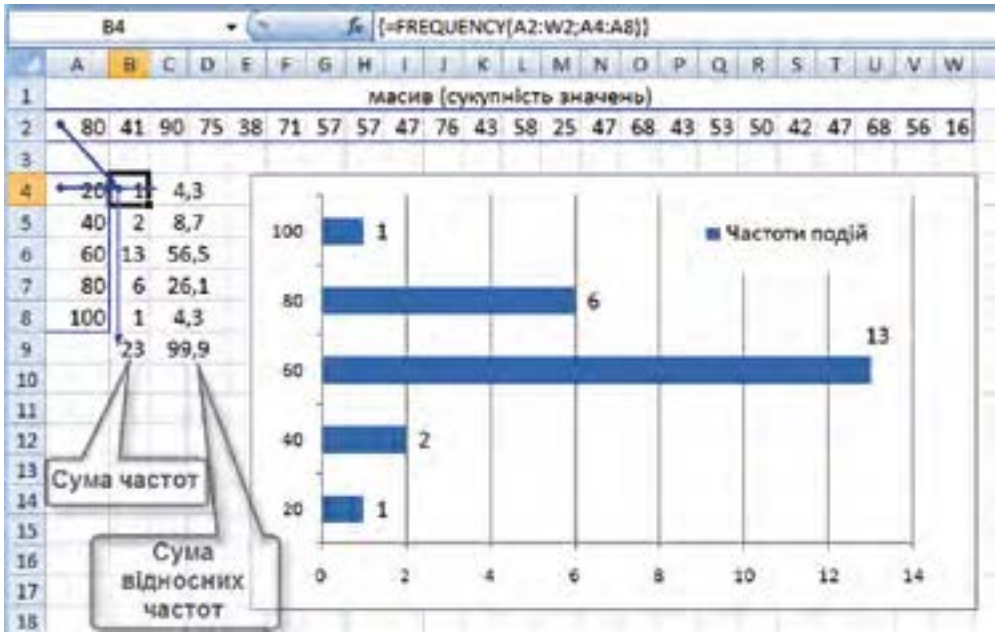


Рис. 8.14. Приклад аналізу даних з використанням функції **ЧАСТОТА** і побудови гістограми

Значення в комірці **B4**, як показано стрілками, впливає на значення в комірках **C4**=ROUND(B4/$B$9*100;1) і **B9**=SUM(B4:B8).

*Застосування функції **ЧАСТОТА** дає можливість побудувати особливий вид діаграми – гістограму, на якій показано, як часто трапляються події, описані у вигляді масиву даних і масиву інтервалів.*

Смужки, що відображають елементи ряду, підписані значеннями частот подій (цифрове значення біля маркера даних певного інтервалу), тому подібна діаграма може бути використана не тільки для якісного, а й кількісного аналізу.

Зокрема, її (якщо на ній відобразити значення відносних частот) можна використати для попереднього оцінювання **ймовірності** певної події, яка полягатиме в тому, що випадковим чином вибране з масиву даних значення потрапить у наперед визначений інтервал значень.

Перевіряємо себе

1. Що є об'єктом дослідження статистики?

2. У процесі розв'язування яких задач виникає потреба застосувати статистичні формули?

3. Що таке розмах вибірки? Чому не можна однозначно судити про вибірку, знаючи тільки середнє арифметичне значення і розмах?

4. Який параметр масиву чисел називають частотою? Відносною частотою?

5. Чому дорівнює сума частот значень для вибірки? Сума відносних частот?

6. Які параметри значень, отриманих у результаті вимірювань, є суттєвими для визначення якості процесу вимірювання?

7. Як можна використати ЕТ для опрацювання результатів соціологічного опитування, для якого використано анкету, що містила п'ять запитань, на кожне з яких можна було відповісти, увівши число 0...10? Запропонуйте варіант, який дасть можливість отримати дані не тільки щодо всього опитування, а й за кожним запитанням окремо.

Виконуємо

1. Відомо, що деякі деталі (наприклад, поршневі пальці та поршні автомобільних двигунів) добирають за групами, які позначають певним кольором. Створіть ЕТ, вхідними даними для якої будуть масиви значень діаметрів поршневих пальців, значення граничних розмірів для груп, кольори груп. Кількість груп – не більше чотирьох. У результаті маємо отримати для кожної деталі (поршневого пальця) номер групи та колір, яким деталь слід позначити.

Порада! Дані (начебто результати вимірювання з використанням мікрометра) можна згенерувати, застосувавши формулу =20+ROUND(RAND()*0,1;3), а умови записати як =IF(A3<=20,025;1;IF(A3<=20,05;2;IF(A3<=20,075;3; IF(A3<=20,1;4)))). Можна використати умовне форматування.

2. Визначте, як розташовано дані в ЕТ, представленій на рис. 8.14. Відтворіть електронну таблицю.

3. Визначте, як розташовано дані в ЕТ, представленій на рис. 8.13. Відтворіть електронну таблицю.

8.5. Умовне форматування

Форматування комірок і написів у комірках полягає у наданні певного кольору, накреслення літер, заливки тла, кольору й товщини меж комірок.

Особливим варіантом форматування, який не використовується у текстових процесорах, є **умовне форматування.**

Умовне форматування – це визначення певного формату подання вмісту комірок залежно від значень числових величин, які містяться в цих комірках, значень логічних виразів, для обчислення яких використовуються дані інших комірок.

Умовне форматування використовують, коли потрібно певним чином виокремити комірку залежно від значення даних, які зберігаються в ній або в іншій комірці.

Табличний процесор Excel 2007 і його наступні версії забезпечують, порівняно з попередніми версіями, ширші можливості *Умовного форматування* та більш зручні засоби його налаштування (рис. 8.15).

✓ *Для визначення умов форматування вмісту комірки або блоку комірок потрібно: виокремити комірки, для формату яких обчислюватиметься значення логічного виразу → викликати зі стрічкового меню* ***Головне*** *групи* ***Стилі*** *→ натиснути кнопку* ***Умовне форматування*** *вибрати кнопку необхідного логічного виразу → у відповідне поле ввести значення, адресу комірки або вираз, складений із адрес комірок та значень.*

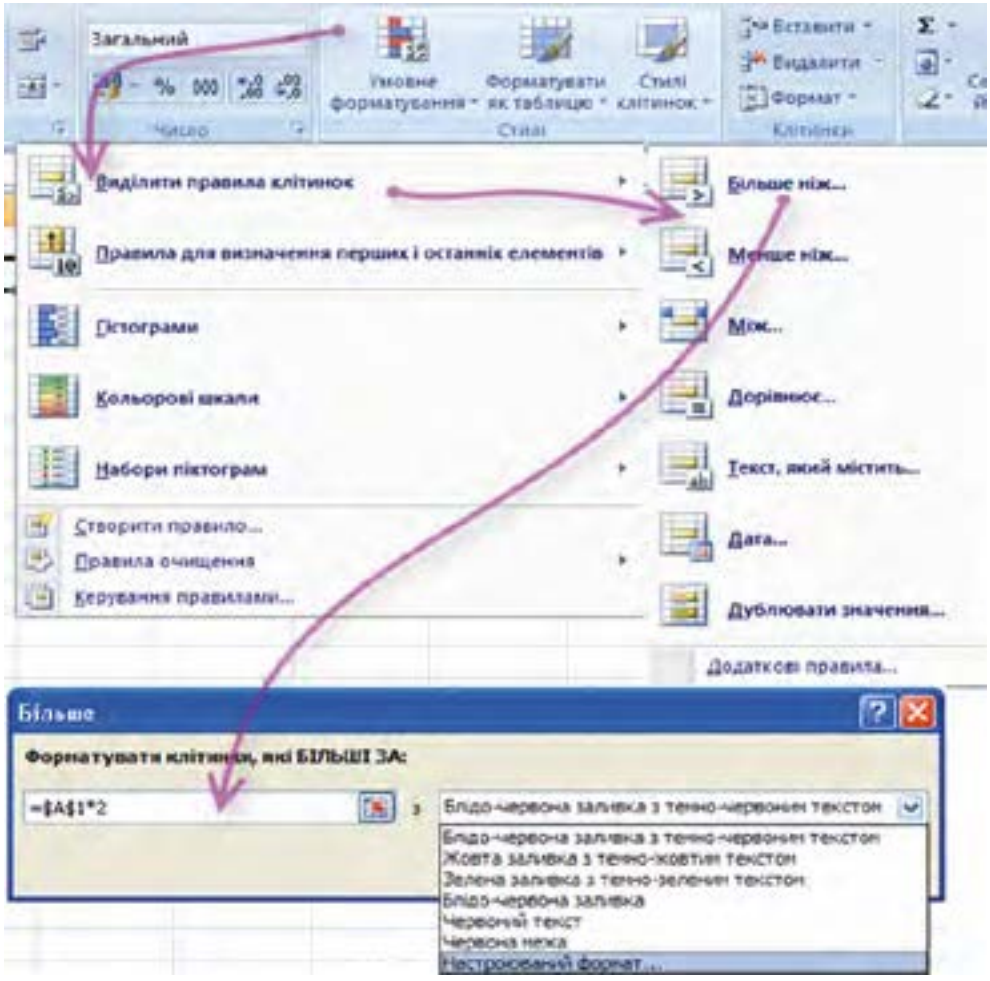


Рис. 8.15. Меню налаштування **Умовного форматування** комірок

Позиції меню: *Створити правило...*, *Правила очищення...*, *Керування правилами...* призначені для виклику відповідних підменю.

Слід зазначити, що скасувати дію *Умовного форматування* для блоків комірок або аркуша можна з використанням підменю *Правила очищення*.

Щоб позначити червоним кольором тексту значення температур, що зберігаються у комірках B2...B10, залежно від того, чи перебуває за цієї температури і нормального тиску деяка речовина в рідкому стані, необхідно в певну комірку (наприклад, A2) ввести значення температури плавлення (наприклад, для води 0 °C, бензолу 5,45 °C, камфори 178,5 °C). Потім виконати дії в певній послідовності (рис. 8.15).

Можливі й інші варіанти умов і форматів, які реалізуються при набутті логічним виразом значення “істина” (true).

Перевіряємо себе

1. Що таке умовне форматування?
2. Як накласти дві-три умови в процесі умовного форматування?
3. Що таке стиль?
4. Як створити новий стиль?
5. Як відбувається імпортування стилів?
6. Що таке автоформат?
7. Які параметри автоформату можна виставляти додатково?
8. Які типи значень можна використовувати при описанні правил (логічних виразів) умовного форматування?
9. Які з перших шести правил (умов) можна застосувати до значень текстового типу (рис. 8.15)?
10. Яке правило можна записати так: $a = x$, якщо a – стала, а x – значення в комірці? До яких типів даних можна застосовувати це правило?
11. Яке правило можна записати так: $a < x$, якщо a – стала, а x – значення в комірці? До яких типів даних можна застосовувати це правило?
12. Яке правило можна записати так: $x < b$, якщо b – стала, а x – значення в комірці? До яких типів даних можна застосовувати це правило?
13. Яке правило можна записати так: $a < x < b$, якщо a і b – сталі, а x – значення в комірці? До яких типів даних можна застосовувати це правило?
14. Чому не завжди доцільно застосовувати правило “Дорівнює...” до значень типу числового типу (дійсних чисел)? Що потрібно зробити, щоб, наприклад, порівняти правильність розв’язку задачі, отриманого у вигляді числа, з еталоннним?

Виконуємо

1. Наведіть приклади доцільності застосування умовного форматування.

2. Для діапазону комірок A3:G8 визначте правило умовного форматування: “якщо значення, введене в комірку, менше за значення, яке зберігається в комірці A2, залити комірку блідо-червоним кольором”.

3. Нехай у комірках стовпця з одинадцятої по двадцяту містяться числа від 1 до 10. Створіть умови форматування, за якими уведення певного числа в п’яту комірку цього стовпця викликало б забарвлення певним кольором потрібної кількості комірок.

4. Таблиця містить такі дані про учнів школи: прізвище, зріст і вік учня. До баскетбольної секції приймають дітей зростом не менше 160 см. Вік не може перевищувати 13 років. Вихідні дані для заповнення таблиці підберіть самостійно (не менше 10 рядків). Таблиця має позначати зеленим кольором прізвища дітей, які можуть бути прийняті до секції.

Порада: в окремому стовпці сформуйте логічний вираз, значення якого використайте для умовного форматування.

8.6. Створення та налагодження діаграм

Діаграми – це засоби наочного подання даних, які полегшують порівняння, виявлення закономірностей і тенденцій змін даних. Усі сучасні табличні процесори надають користувачеві можливість побудувати діаграму. Діаграми можна будувати для одного або кількох рядів даних. Діаграми типу Графік можуть унаочнювати зв’язок між двома величинами, відображати графік функції.

Спеціальні види діаграм, тривимірні діаграми. Налагодження діаграм.

При розгляді застосування формул масиву використано новий тип діаграм – **гістограму**, який відрізняється тим, що для стовпчикової діаграми вісь категорій розташована вертикально (рис. 8.14). Таке подання розподілу частот подій використовується досить часто для ілюстрування результатів економічних і соціометричних досліджень, фізичних експериментів.

Для підприємства, яке має кілька філій (видів діяльності), унаочненням внеску кожної філії (виду діяльності) у загальний прибуток (загальні витрати) може бути стовпчикова діаграма, в якій висота стовпчика, який відповідає часовому інтервалу (рік, місяць, тиждень) пропорційна сумі

коштів, отриманих (витрачених) підприємством, а кожна філія (вид діяльності) позначена іншим кольором.

Для подання даних щодо змін температури повітря (день – ніч), мінімального і максимального значення досягнень групи спортсменів, вартості цінних паперів на біржі використовують так звану **біржову діаграму**, на якій відображаються одночасно мінімальне, максимальне і середнє значення деякої величини.

Для того щоб порівняти й унаочнити внески кількох напрямів діяльності в загальну діяльність, використовують "пелюсткову" діаграму.

Діаграму можна створити на окремому аркуші або розташувати як впроваджений об'єкт на аркуші даних. Крім цього, діаграму можна опублікувати на веб-сторінці. Щоб створити діаграму, потрібно спочатку ввести для неї дані на аркуші. За допомогою панелі інструментів **Діаграма** створити діаграму.

Діаграма зв'язана з даними аркуша, на основі яких вона створена, і в разі зміни даних автоматично оновлюється.

Нижче наведено назви та описання елементів діаграми.

Маркер даних. Смуга, область, точка, сектор або інший об'єкт на діаграмі, який представляє одну точку даних або значення клітинки аркуша. Пов'язані один з одним маркери даних на діаграмі утворюють ряд даних. Послідовність маркерів даних представляє один ряд даних.

Ряд даних. Пов'язані одна з одною точки даних, нанесені на діаграму. Кожний ряд даних на діаграмі має власний колір або інший спосіб позначення та представлений на легенді діаграми. Діаграми всіх типів, за винятком кругової, можуть містити кілька рядів даних.

Основні лінії сітки. Лінії, які можна додати до діаграми для поліпшення сприйняття й оцінювання відображуваних даних. Лінії сітки починаються від поділок на осі й перетинають область побудови. На діаграмі можна також вивести на екран проміжні лінії, які позначають інтервали в межах основних інтервалів.

Імена категорій. Microsoft Excel використовує заголовки стовпчиків або рядків як імена інтервалів осі категорій. При налагодженні діаграми їх можна замінити іншими.

Імена рядів даних діаграми. Microsoft Excel також використовує заголовки стовпчиків або рядків як імена рядів даних. Імена рядів даних показані в легенді.

Легенда. Область, у якій подано кольори або інші способи позначення, що відповідають рядам даних або категоріям на діаграмі.

Наприклад, якщо навести курсор на легенду, з'явиться підказка, яка містить слово "Легенда".

Аркуш діаграми. Аркуш книги, який містить лише діаграму.

Відмінність від попередніх версій полягає в тому, що не обов'язково дотримуватися послідовності дій.

Упроваджена діаграма. Діаграма, розташована на аркуші даних, а не на окремому аркуші діаграми. Впроваджені діаграми зручні, коли потрібно переглянути або надрукувати діаграму чи звіт зведеної діаграми разом із вихідними даними й іншими відомостями, які містяться на аркуші.

✓ *Якщо затримати курсор миші на елементі діаграми, з'явиться підказка з назвою цього елемента.*

✓ *Створення діаграми розпочинається з вибору її типу.* Слід пам'ятати, що до початку створення діаграми необхідно повністю закінчити формування структури таблиці, наповнити її даними.

Процедура створення діаграми для Microsoft Excel 2003 і попередніх могла бути виконана з використанням **Майстра діаграм (ChartWizard)** покроково, за п'ять кроків.

На рис. 8.16 показано послідовність побудови діаграми для Microsoft Excel 2007–2010.

Після кроку, на якому на аркуші з'являється порожнє полотно діаграми, у вікні Excel 2007–2010 з'являється контекстний інструмент "Робота з діаграмами", що містить три стрічки "Конструктор", "Макет", "Формат". Інструменти роботи з діаграмами в Excel 2007–2010 прості й зрозумілі (рис. 8.17).

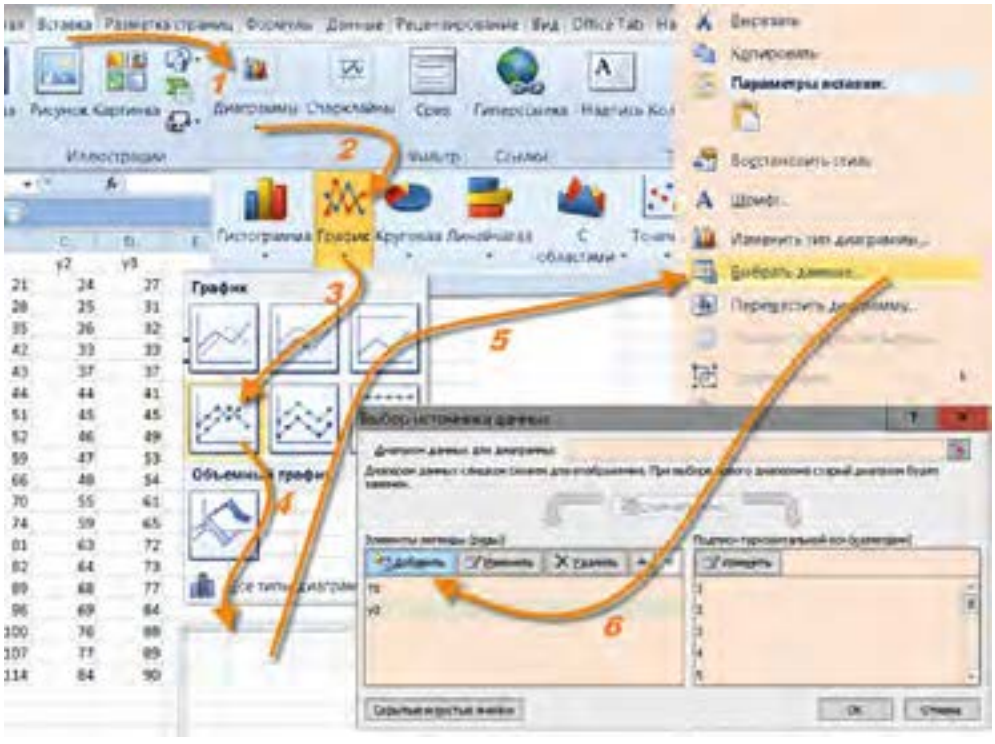


Рис. 8.16. Процедура створення діаграми для Microsoft Excel 2007–2010

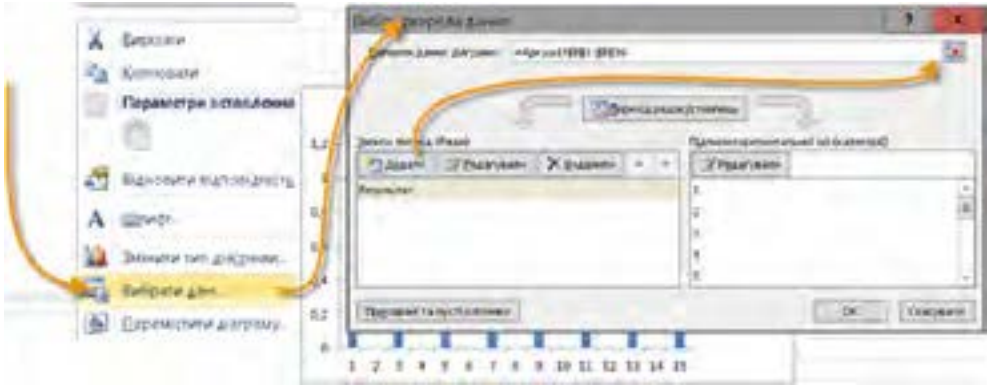


Рис. 8.17. Процедура вибору даних для діаграми у Microsoft Excel 2007–2010

Можливим є й інший спосіб створення діаграми, який показано на рис. 8.16 (кроки 5, 6 і наступні). Після того як на аркуші з'явилось полотно діаграми, можна, не користуючись стрічковими меню, викликати натисканням правої кнопки миші на полотні діаграми, через команду **Вибрати дані...** контекстного меню викликати вікно **Вибір джерела даних** (кроки 5 і 6 на рис. 8.16 і рис. 8.17).

✓ *У вікні **Вибір джерела даних** є такі об'єкти.*

Список **Елементи легенди (ряди).** Відображає список імен існуючих рядів даних.

Кнопка **Додати.** Натисніть цю кнопку, щоб додати до діаграми новий порожній ряд. Щоб додати ім'я та значення для нового ряду, клацніть поле **Ім'я** або **Значення** та виділіть діапазон на аркуші або введіть ім'я та значення в поля. Дані, введені в полях **Ім'я** та **Значення**, не додаються до аркуша.

Кнопка **Видалити.** Вилучає вибраний ряд даних із діаграми.

✓ *Додавання та вилучення рядів даних на діаграмі не впливає на дані на аркуші.*

Якщо в ту частину таблиці, за якою будувалася діаграма, буде внесено зміни, то програма Excel автоматично модифікує діаграму.

Для відображення на діаграмі нових даних виокремте їх у таблиці й перетягніть за допомогою миші на діаграму. Для цього поставте покажчик миші на межу маркірованої області, не відпускаючи лівої кнопки миші, перемістіть її на діаграму. Щойно відпустите кнопку миші, діаграма буде змінена (актуалізована).

Можна скористатися й кроками 5 і 6 на рис. 8.17.

Якщо діаграма міститься на окремому робочому аркуші, то для її актуалізації можна використовувати команду **Вибрати дані (New Data)** з меню **Вставлення (Insert).** У діалоговому вікні цієї команди потрібно

вказати область таблиці, яку додано. Для цього або виділіть цю область, або вкажіть її адресу, попередньо створивши новий ряд (кроки 5 і 6 на рис. 8.16).

Щоб додати текст у будь-який елемент діаграми, клацніть на ньому правою кнопкою миші, виберіть команду **Змінити текст**, а потім уведіть текст.

Розглянемо особливості деяких діаграм.

✓ *Точкова діаграма – єдина діаграма, для якої осі X і Y можна вважати осями декартової системи координат і будувати графіки функцій так само, як на папері.*

✓ *Для правильної побудови графіка функції як точкової діаграми не потрібно мати набір значень аргументу з постійним кроком, оскільки вісь X у цьому випадку є віссю значень, а не віссю категорій.*

На рис. 8.18 показано графік функції $y = 0.5x^2 - 10$, побудований як точкова діаграма.

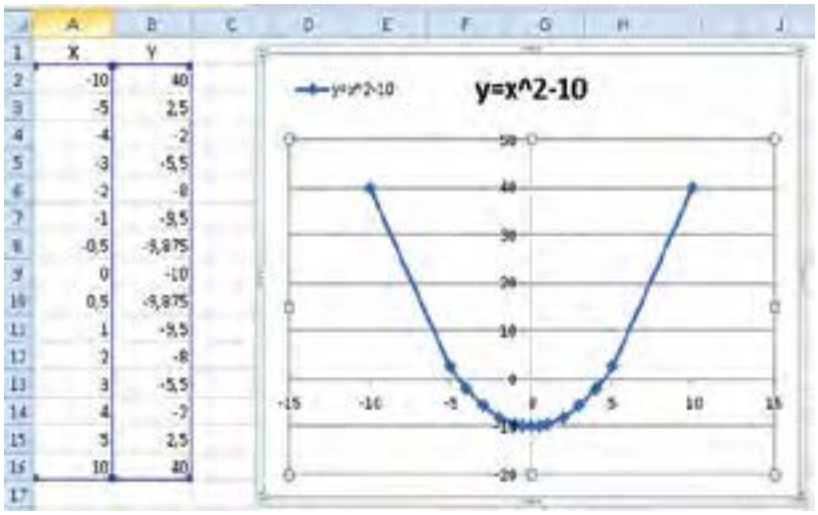


Рис. 8.18. Процедура створення діаграми для Microsoft Excel 2010

Нехай протягом 12 місяців три філії одного підприємства давали прибутки, показані у стовпчиках B, C, D. На рис. 8.19 подано зміни прибутків кожної філії та всього підприємства у вигляді двох стовпчикових діаграм – із накопиченням і в тривимірному вигляді.

На циклічних і цільових діаграмах текст можна вводити лише в призначені для цього рамки, які відображаються під час додавання діаграми або елемента діаграми. Для змін вигляду будь-якого елемента діаграми його слід актуалізувати натисненням на ньому лівої кнопки миші з наступним викликом меню натисненням правої кнопки. Послідовність дій подано на рис. 8.20.

Рис. 8.19. Два різних способи подання даних стовпчиковими діаграмами

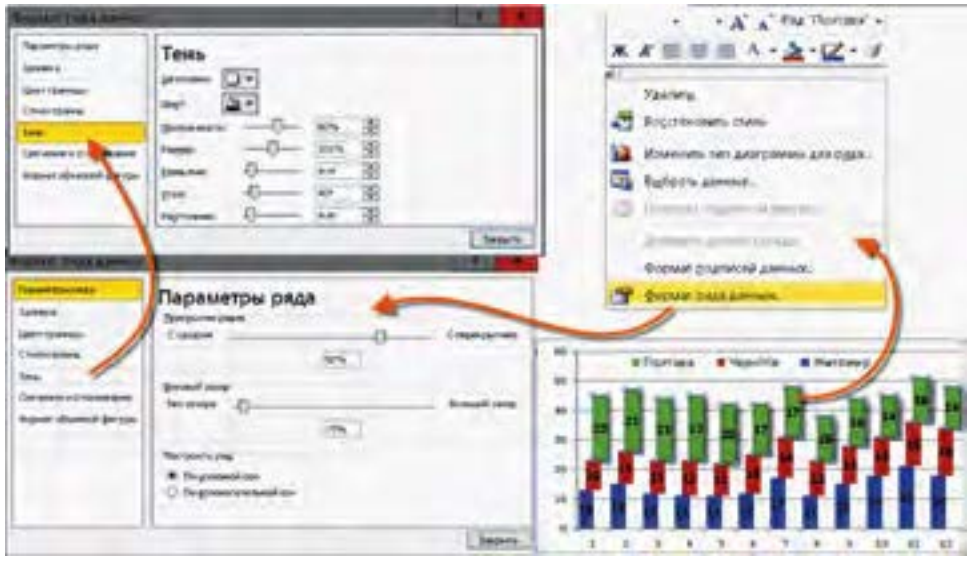


Рис. 8.20. Редагування елементів діаграми

Щоб додати елемент, який не передбачено стандартним макетом діаграми, натисніть кнопку **Додати фігуру** на панелі інструментів **Макет.**

Перевіряємо себе

1. Які основні елементи містить стандартний макет діаграми?
2. Як викликати певний тип діаграми?
3. Як викликати і застосувати стандартний макет діаграми?
4. Чи можна змінювати тип діаграми після зв'язування її з даними? Коли це можливо й доцільно?
5. Діаграма якого типу вимагає двох наборів даних?
6. Яким чином можна додати дані на діаграму?
7. У яких випадках потрібно користуватися контекстним меню редагування елементів діаграми?
8. Як додати підписи (значення) до зображень елементів ряду і коли це доцільно робити?
9. Як редагувати текстові елементи діаграми?

Виконуємо

1. Побудуйте графік функції $y = 2x^2 - 3x - 20$ для значень $-10 < x < 10$.
2. Визначте з графіка приблизні значення коренів рівняння $2x^2 - 3x - 20 - 0$. Перевірте обчисленням.
3. Створіть ЕТ і діаграми, подані на рис. 8.19.

8.7. Упорядкування даних у таблицях. Автоматичні та розширені фільтри

Пошук значень здійснюється поелементним порівнянням значень і зразка. Так само, як у текстовому редакторі та файловій системі, пошук можна виконувати з використанням шаблонів.

Упорядкування здійснюється з урахування розміщення літер у алфавіті мови, призначеної для тексту. Структури подання даних.

Табличні процесори надають можливості для пошуку будь-якого вмісту на робочому аркуші та, при потребі, заміни його новим вмістом.

Пошук провадиться у виокремленому діапазоні комірок, а за відсутності виокремлення – в усьому робочому аркуші.

Можна виокремити декілька робочих аркушів, тим самим задавши пошук потрібного вмісту відразу в кількох аркушах. При заданні зразка

шуканого вмісту можна використовувати будь-які літери, цифри та спеціальні символи. Крім цього, є такі символи підстановки:

✔ *Знак питання (?) застосовується для позначення будь-якого символу (але одного), зірочка (*) – для позначення будь-якої кількості будь-яких символів. Для задання пошуку самого символу підстановки (? або *) слід увести перед ним хвилясту риску (~).*

При використанні як пошуку, так і заміни можна зазначити, чи береться до уваги точне написання шуканого вмісту (малі або великі літери). Крім цього, можна зазначити, чи має провадитися пошук зразка тільки як окремого вмісту комірки чи як довільної частини вмісту комірки.

Можна також вказати напрямок пошуку: в рядках зліва направо, або в стовпцях зверху вниз. Клацання миші на вкладці **Замінити** дозволяє перейти з вікна **Знайти** у вікно **Замінити**.

*У меню **Правка** виконати команду **Замінити** або скористатися сполученням клавіш **Ctrl+H** → у поле **Знайти** ввести послідовність символів, які потрібно знайти (зразок) (рис. 8.21. А) → у полі **Замінити на** вказати послідовність символів для заміни.*

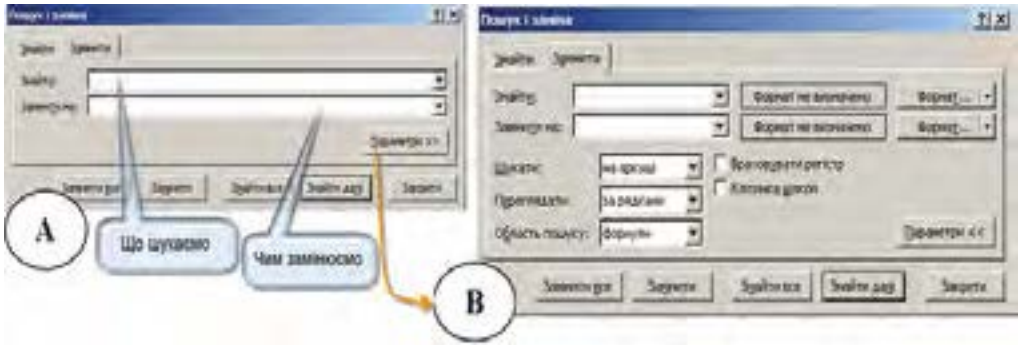


Рис. 8.21. Пошук вмісту для заміни: А – пошук; В – розширений пошук

*За потреби можна зазначити інші параметри пошуку – натиснути **Параметри** (рис. 8.21. В) → для продовження пошуку натиснути кнопку **Знайти далі** → для виконання заміни натиснути кнопку **Замінити**.*

Кнопка ***Замінити все*** дозволяє виконати заміну всіх знайдених послідовностей символів, які трапляються. При цьому потрібно впевнитися, що така заміна за замовчуванням усіх знайдених послідовностей дасть бажаний результат.

Усі табличні процесори мають зручний інструментарій для виконання сортування та фільтрації даних.

Під **сортуванням** розумітимемо **впорядковування** даних за одним або кількома заданими критеріями.

Програма виконує декілька варіантів сортування. Найпростішим варіантом сортування є впорядкування за зростанням. Цей варіант сортування встановлено за замовчуванням.

Упорядкування даних таблиці в алфавітно-цифровому порядку здійснюється за зростанням або спаданням значень. Числа сортуються від найменшого від'ємного до найбільшого додатного, а текст – у алфавітному порядку (за зростанням або спаданням).

✓ *Сортування даних за кількома полями.*

Засоби Excel дають змогу одночасно сортувати записи за трьома полями. Послідовність сортування полів вибирається в діалоговому вікні **Сортування діапазону** в списках, що розкриваються: **Сортувати за, Потім за, В останню чергу за.**

Розташовані поряд із кожним списком перемикачі за зростанням, за зменшенням дозволяють вказати напрямок сортування.

Перемикач **Ідентифікувати діапазон даних за** дає змогу ідентифікувати дані за підписами або позначеннями стовпців аркуша.

При потребі сортування за чотирма і більше полями варто виконати кілька послідовних сортувань. Щоб не втрачати результати попереднього сортування, потрібно спочатку виконати сортування за останніми трьома полями, а потім за найпершим.

Фільтрування використовується у роботі з великими таблицями і дає змогу бачити не всю таблицю, а лише ту її частину, яка висвітлюється за певними ознаками (критеріями).

✓ *Щоб вибрати критерії, за якими здійснюється фільтрування, потрібно виконати такі дії: виокремити діапазон (або всю таблицю), скористатись командою* ***Дані/Фільтр/Автофільтр****. Відразу в кожній комірці верхнього рядка з'явиться кнопка списку.*

Клацаючи кнопки списку, вибираємо відповідні критерії фільтрування, тобто відбору. Одержимо таблицю з відібраними даними. У цій таблиці можна отримувати потрібні суми, добутки, виконувати інші дії, як і в будь-якій таблиці.

✓ *Упорядкування за зростанням (за спаданням).*

Цей варіант сортування можна застосовувати до даних числових і текстових типів. Застосовуючи сортування за зростанням (у випадку сортування за спаданням порядок у списках буде протилежним), слід враховувати особливості роботи програми, а саме:

– порожні комірки завжди вміщуються у кінець відсортованого списку (як у варіанті сортування за зростанням, так і у варіанті за спаданням);

– числові типи даних сортуються від найменшого від'ємного до найбільшого додатного (у варіанті сортування за спаданням – навпаки);

– текстові типи даних сортуються познаково зліва направо;

– текстові дані сортуються за таким порядком: спочатку цифри, потім пробіл та символи цифрових клавіш верхнього регістра, і тільки після цього літери у алфавітному порядку;

– під час сортування логічних значень значення **ХИБНІСТЬ (False, 0)** ставиться перед значенням **ІСТИНА (True, 1)**.

Сортування за кількома критеріями відбору.

Розглянемо такий варіант:

– виокремити діапазон комірок, де слід виконати сортування;

– виконати команду меню **Дані** / **Сортування**;

– у діалоговому вікні **Сортування даних** у полях **Сортувати за** і **Потім за** вказати стовпці в такому порядку, який потрібен (спочатку за стовпцем "Вік", а потім за стовпцем "Особа"). Натиснути ОК.

Можна вказати три такі умови відбору, причому деякі стовпці можна сортувати за зростанням, а деякі за спаданням. Для проведення сортування за чотирма й більше критеріями відбору слід виконувати сортування в декілька етапів (починаючи з найменш важливого на першому етапі та з найбільш важливого на другому).

Для сортування даних у стовпцях за днями тижня, місяцями слід скористатися ключем (який викликають кнопкою **Параметри)** у діалоговому вікні **Сортування даних.**

Фільтр – швидкий і легкий спосіб пошуку та відображення даних, які задовольняють певні критерії. Після застосування фільтра в таблиці відображаються тільки ті дані, які задовольняють критерії відбору.

Програма Microsoft Excel має два різновиди фільтрування даних – **автофільтр** (для простих умов відбору) та **розширений фільтр** (для складених умов відбору.

Фільтрування від сортування відрізняється тим, що після фільтрування дані в списку не впорядковуються, з них лише тимчасово приховуються записи, які не задовольняють критерії відбору.

Але приховані записи в будь-який момент можна відобразити на екрані.

Найзручнішим інструментом фільтрування є **Автофільтр.** Цей інструмент містить такі фільтри:

– **Перші 10** – для відображення перших 10 записів даних;

– **Сортування за зростанням (сортування за спаданням)** – відображає всі записи, впорядковані за зростанням (за спаданням);

– **Умова ...** – інструмент для відображення на екрані записів, що задовольняють певні критерії, які можна задати в діалоговому вікні **Користувацький автофільтр.** Тут можна задавати як прості, так і складені умови фільтрування.

Таблиця 8.2

Організація структур даних

Особливості стовпця	Форма подання даних
1	**2**
Подібні елементи розташовувати в одному стовпці	Упорядковувати дані слід таким чином, щоб в усіх рядках подібні елементи містилися в одному й тому ж стовпці
Діапазон розташовується окремо	Слід залишати принаймні один порожній стовпець і один порожній рядок між діапазоном споріднених даних і рештою даних на аркуші. Тоді ЕТ ефективніше виявляє та виокремлює діапазон під час сортування, фільтрування або вставлення автоматичних проміжних підсумків
Важливі дані мають міститися над діапазоном або під ним	Слід уникати розташування важливих даних ліворуч або праворуч від діапазону, тому що при фільтруванні діапазону вони можуть стати прихованими
Рядки та стовпці мають бути відображені	Перш ніж вносити зміни до діапазону, слід переконатися, що приховані рядки та стовпці відображено. Якщо в діапазоні не відображаються деякі рядки або стовпці, можливе помилкове видалення даних
Формат даних	
Використовувати форматовані підписи стовпців	Рекомендується створювати в першому рядку діапазону даних підписи стовпців. Ці підписи використовуються в ЕТ для створення звітів, пошуку та впорядкування даних. Для підписів стовпців слід використовувати шрифт, вирівнювання, тло, межу або регістр, відмінні від параметрів форматування, які мають дані в діапазоні. Перш ніж уводити підписи стовпців, слід установити для комірок текстовий формат
Використовувати межі комірок	Якщо потрібно відокремити підписи від даних, бажано виконати це за допомогою меж, а не вставляти порожні рядки або риски між підписами та рядками даних
Уникати використання порожніх рядків і стовпців	Для того щоб ЕТ оптимально знаходила й виокремлювала діапазони споріднених даних, не слід вставляти в діапазон порожні рядки або стовпці
Не застосовувати пробіли перед даними або після них	Зайві пробіли на початку або в кінці комірки заважають правильному сортуванню або пошуку. Для зсування тексту в комірці слід застосовувати відступи

Закінчення табл. 8.2

1	2
Розширення форматів даних і формул	Якщо в кінець діапазону даних додаються нові рядки, на них автоматично поширюються наявне форматування та формули. Для того щоб відбувалося розширення формату, три з п'яти попередніх комірок мають мати однаковий формат. А щоб відбувалося розширення формул, усі попередні формули мають бути однакові

Розглянемо застосування автофільтра на прикладі. Нехай у списку треба показати всіх осіб віком до 12 років.

*Слід зробити таке: виділити стовпці з даними → виконати команду меню **Дані/Фільтр/Автофільтр** → після виконання команди праворуч у назвах виділених стовпців з'являться кнопки зі стрілками → натиснути на кнопку стовпця з назвою "Вік" і обрати з переліку можливих варіантів фільтрування **Умова** → у діалоговому вікні **Користувацький автофільтр** у полі **Вік** обрати умову **Менше або дорівнює**, у полі поряд вказати "12".*

Розширений фільтр. Якщо потрібно відібрати дані за складених умов відбору, зручніше застосувати інструмент для фільтрування "розширений фільтр".

Умови можуть бути такими:

1) декілька умов для одного стовпця даних;
2) одна умова для кількох стовпців даних;
3) одна умова для одного стовпця й інша для іншого;
4) набір умов для кількох стовпців;
5) пошук за умовою у вигляді формули.

Для використання функцій фільтрування і сортування таблиця має мати структуру, описану в табл. 8.2.

Перевіряємо себе

1. Чому при застосуванні фільтрування та сортування дані мають бути введені з суворим дотриманням певних правил?
2. Чим сортування відрізняється від фільтрування?
3.  У латинському та українському алфавітах є однакові на вигляд літери. Що буде зі списком адрес шкіл України після сортування, якщо у назві міста Львів літера "і" інколи набиратиметься як латинська, а інколи – як українська? Перевірте.
4. До яких наслідків може призвести порушення структури, рекомендованої в табл. 8.2, п. 1: "Подібні елементи розташовувати..."?

5. До яких наслідків може призвести порушення структури, рекомендованої в табл. 8.2, п. 8: “Не застосовувати пробіли перед даними...”?

6. Які інструменти для фільтрування даних вам відомі?

7. З яких етапів складається створення розширеного фільтра?

8. Які можливості надає розширений фільтр?

9. Наведіть приклади доцільності використання автофільтра, фільтра користувача, розширеного фільтра.

10.  Чим подібні розділи var і type програми мовою Паскаль і табл. 8.2?

Виконуємо

1.  З веб-порталу фірми КРІ-Сервіс завантажте прайс-лист http://web-portal.kpiservice.com.ua/kpi_new/price/price_opt.zip.

2. Відкрийте аркуш “Вспомогательное оборудование” і встановіть автофільтр (Дані/Фільтр/Автофільтр).

3. За допомогою встановленого фільтра виберіть із прайс-листа джерела безперебійного живлення back pro (кнопка списку поля Тип).

4. Упорядкуйте результат фільтрування щодо дрібногуртової ціни, а потім щодо роздрібної ціни в порядку зростання (Дані/Сортування).

8.8. Підсумки й проміжні підсумки

Сортування та фільтрування даних. Типізація даних у мові програмування і в табличному процесорі. Об’єкти і їх властивості. Формули в табличному процесорі, аргументами яких є масиви. Опрацювання даних як інформаційний процес.

База даних. Фільтри та формули, аргументом яких є масив як спосіб отримання нової інформації.

Якщо електронну таблицю вже створено з дотриманням правил, викладених у табл. 8.2, то це означає, що отримано новий об’єкт, який можна назвати **базою даних** (БД).

База даних – певним чином структурована сукупність даних, отриманих для певного класу об’єктів.

Найпростішим способом подання даних для описання одного об’єкта певного класу є **запис**, у якому для кожної властивості виокремлено **поле** з наперед визначеним форматом подання даних. У електронній таблиці таким **записом** може слугувати **рядок**, а **полем** – **комірка**.

Застосовуючи фільтр до даних, розміщених у комірках ЕТ, заповнених даними за правилами, викладеними в табл. 8.2, тобто до бази даних, здійснюємо **запит** до бази даних.

Запит до БД є інформаційним процесом, результатом якого є отримання нової інформації.

Це не означає, що інформація "виникає з нічого" – інформація з'являється як результат **спільного опрацювання запиту і даних.** Частина інформації, яку отримуємо в результаті запиту, вже міститься в БД, частину подаємо, коли формулюємо умови запиту.

Наприклад, у БД містяться дані щодо поживності основних продуктів, які ми будемо споживати, їх вартості. Якщо ми робимо запит: "*Показати, які й скільки продуктів мені потрібно взяти з собою в похід на 10 днів, якщо передбачаються витрати енергії 3500 ккал щодня. Загальна маса продуктів не більше 3 кг, мінімальна порція кожного продукту не менше 50 г, продукти мають бути досить різноманітними, а загальна вартість їх не має перевищувати 2000 грн*", то ми до інформації з БД застосовуємо досить складне правило, сформульоване з урахуванням декількох вимог. Отже, запит теж містить певну інформацію.

Запит до БД, створеної в ЕТ, можна сформулювати й застосувати, використовуючи фільтри та інші засоби (табл. 8.3).

Для числового наповнення БД можна створити список із проміжними підсумками, використовуючи команду **Підсумки** в меню **Дані**. Якщо список із проміжними підсумками вже створений, його можна модифікувати, редагуючи формулу з функцією **Проміжні підсумки.**

Синтаксис:

ПРОМІЖНІ.ПІДСУМКИ(номер_функції;посилання;посилання;...)

Номер_функції – це число від 1 до 11, яке вказує, яку функцію використовувати для обчислення підсумків у списку. Посилання 1; Посилання 2; – від 1 до 29 інтервалів або посилань, для яких підводяться підсумки.

Якщо в таблиці вже є формули підведення підсумків для аргументів посилання 1; посилання 2; ... (так звані вкладені підсумки), то ці вкладені підсумки ігноруються, щоб уникнути подвійного підсумовування.

Функція ПРОМІЖНІ.ПІДСУМКИ ігнорує всі приховані рядки, які стають такими в результаті фільтрування списку. Це важливо, оскільки стає можливим підвести підсумки тільки для тих даних, що містяться в рядках, які відтворюються після фільтрування списку.

Те, що ПРОМІЖНІ.ПІДСУМКИ, призначені для діапазону комірок, за наявності сформованого для цього ж діапазону комірок фільтра, обчислюються тільки для тих комірок, значення яких "пройшли фільтр", дуже важливо. Ми отримуємо змогу сформувати запит до БД, який можна описати так: "Покажи результат опрацювання значень у

комірках стовпця, розташованих у рядках, дані в яких відповідають певним об'єктам".

Таблиця 8.3

Деякі функції, що застосовуються для статистичного опрацювання даних

Номер функції	Функція	Опис функції
1	СРЗНАЧ (AVERAGE)	Повертає середнє (арифметичне) своїх аргументів
2	СЧЁТ (COUNT)	Підраховує кількість чисел у списку аргументів. Використовується для отримання кількості числових значень у інтервалах або масивах значень
3	СЧЁТЗ	Підраховує кількість непорожніх значень у списку аргументів. Використовується для підрахунку кількості комірок з даними в інтервалі або масиві
4	МАКС	Повертає найбільше значення з набору значень
5	МИН	Повертає найменше значення з набору значень
6	ПРОИЗВЕД	Перемножує числа, вказані як аргументи, і повертає їх добуток
9	СУММ	Додає всі числа в інтервалі комірок

Приклад. Нехай дані щодо об'єктів класу "зернові культури" розташовані в комірках (полях) рядків, які відповідають стовпцям із назвами "Назва культури", "Урожайність, центнерів з гектара", "Країна", "Рік". Тоді за запитом "Покажи середню врожайність для "Назва культури" = "Кукурудза" і "Країна" = "США" отримаємо певне значення. Для такого ж запиту, але з "Назва культури" = "ячмінь" отримаємо інше значення. Змінюючи фільтр по країні, отримуємо інформацію щодо врожайності зернових у різних країнах протягом кількох років. Порівнюючи отримані значення між собою, доходимо висновку щодо доцільності вирощування певної культури в конкретній країні.

Подібним чином можна виконати й інші дослідження, застосовуючи спільно засоби фільтрування даних і опрацювання результатів з використанням функції ПРОМІЖНІ.ПІДСУМКИ.

Заздалегідь запланувавши важливі для аналізу даних запити, доцільно створити **Зведену таблицю.** У цій таблиці вже мають бути сформовані (копіюванням значень або зв'язуванням комірок) рядки, комірки (поля) яких містять дані, що описують певну властивість об'єкта. Дані мають бути занесені з дотриманням вимог, поданих у табл. 8.2.

За даними **Зведеної таблиці** можна будувати діаграми, які відображатимуть отримані дані. Ці діаграми називатимуться **Зведеними діаграмами.**

Перевіряємо себе

1. Як відображаються властивості об'єкта в електронній таблиці?
2. У яких випадках доцільно застосовувати попереднє сортування (фільтр) до даних, які є аргументами проміжних підсумків?
3. Яку інформацію дасть застосування до даних обчислення проміжних підсумків функції з номером 3 (табл. 8.3)? Чи можна такий запит використати задля контролю правильності формування зведеної таблиці?
4. Що потрібно зробити насамперед, щоб створити зведену таблицю?
5. Чому відповідають і що містять поля зведеної таблиці?
6. Чому потрібно суворо дотримуватися однаковості форматів даних, які є аргументами проміжних підсумків?

Виконуємо

Створіть таблицю, яку можна вважати базою даних щодо успішності в навчанні учнів класу.

Таблиця має забезпечувати виконання таких запитів: "Який середній бал учнів класу з інформатики, математики, фізики тощо?", "Скільки учнів мають середній бал з предмета, нижчий за середній з усіх предметів?" та подібних.

8.9. Надання значень параметрам книги й аркуша

У Microsoft Excel можна створити **Стандартний шаблон книги** і **Стандартний шаблон аркуша**.

Стандартний шаблон книги – Книга.xlt створюється для зміни використовуваного за замовчуванням формату новостворюваних порожніх книг. Надалі цей шаблон буде використовуватись у Microsoft Excel для створення порожньої книги при запуску або для створення книги без зазначення шаблону.

Стандартний шаблон аркуша – Аркуш.xlt створюється для зміни використовуваного за замовчуванням формату новостворюваних порожніх аркушів. Перед створенням шаблону слід обрати його тип.

Шаблон книги: створити книгу, яка містить аркуші, стандартний текст (наприклад, заголовки сторінок, підписи стовпців і рядків), формули, макроси та інші елементи форматування, які мають бути у створюваних на основі цього шаблону книгах.

Шаблон аркуша: створити книгу з одним аркушем та помістити на цей аркуш елементи форматування, стилі, текст та інші дані.

*Створення шаблону: у меню **Файл** вибрати команду **Зберегти як** → у списку **Тип файла** вибрати пункт **Шаблон** → у рядку **Ім'я файла** набрати **ім'я** (при створенні шаблону книги для використання за замов-*

чуванням ввести текст ***Книга****, при створенні спеціального шаблону книги ввести довільне припустиме ім'я файла; при створенні шаблону аркуша для використання за замовчуванням ввести текст* ***Аркуш****, при створенні спеціального шаблону аркуша ввести довільне припустиме ім'я файла → натиснути* ***Зберегти.***

У полі **Папка** можна вказати папку, в якій має бути збережений шаблон. Для створення спеціального шаблону книги або аркуша необхідно вибрати папку **Шаблони.**

Щоб зображення першої сторінки шаблону було показане в полі **Перегляд** діалогового вікна **Шаблони**, потрібно *вибрати в меню* ***Файл*** *команду* ***Властивості*** *→ відкрити вкладку* ***Документ*** *→ установити прапорець* ***Створити малюнок для попереднього перегляду***. У цьому вікні можна також вказати низку додаткових відомостей про документ (тема роботи, керівник роботи, назва установи, в якій виконується робота) і зв'язок з іншими документами (гіперпосилання).

Практична робота № 14

Тема:	Розв'язування задач на обчислення Обчислювальні алгоритми в середовищі табличного процесора
Мета:	Набути практичних навичок створення електронних таблиць

Завдання.

1. У таблиці подано дані про площу та населення деяких країн Європи. Ввести у відповідні комірки формули для обчислення:

1) загальної площі та загальної кількості населення;
2) густоти населення в кожній країні (осіб/км^2);
3) частки, яку становить населення кожної з цих країн щодо загальної кількості населення в усіх країнах.

Країна	Площа, км^2	Населення, млн осіб	Густота населення	Відсоток
Україна	603700	46,3		
Франція	547030	67,3		
Іспанія	504782	40,5		
Швеція	449964	9,1		

2. Створити таблицю для опрацювання результатів вимірювання густини речовини методом обмірювання і зважування.

3. Створити таблицю для опрацювання результатів вимірювання швидкості звуку в повітрі (на віддалі S від спостерігача стріляють з гармати, спостерігач фіксує час між спалахом і звуком).

4. Створити таблицю для опрацювання результатів лабораторної роботи "Визначення питомої теплоємності речовини".

5. Створити таблицю для опрацювання результатів лабораторної роботи “Вимірювання опору провідника за допомогою амперметра і вольтметра”. ✦

Практична робота № 15		
	Тема:	Використання математичних, логічних і статистичних функцій табличного процесора. Умовне форматування
	Мета:	Набути практичних навичок створення електронних таблиць із використанням математичних, логічних і статистичних функцій табличного процесора

Завдання. З використанням електронної таблиці здійснити опрацювання даних із застосуванням статистичних функцій.

1. Дано відомості про учнів класу, що включають усі оцінки, отримані кожним учнем протягом одного семестру. Підрахувати кількість кожної з оцінок (від 1 до 12 балів), знайти середній бал кожного учня і середній бал всієї групи. Створити діаграму, яка ілюструє відсоткове співвідношення оцінок в групі. ✦

2. Створити електронну таблицю для опрацювання результатів лабораторної роботи “Вивчення теплового балансу за умов змішування води різної температури”. Передбачити, що в таблиці опрацьовуватимуться результати, отримані в п’яти експериментах. Таблиця має обчислювати: теоретичне (за рівнянням теплового балансу) значення температури суміші, різницю між обчисленим і виміряним значеннями температури суміші, середнє значення абсолютних величин різниць обчисленої та виміряної температур. ★

3. Створити електронну таблицю для спостереження за температурою повітря протягом місяця. Якщо до комірки вводиться значення температури менше нуля, поряд (у сусідній комірці) має з’явитися синя сніжинка, більше нуля – жовте сонечко. Якщо температура дорівнює нулю, у комірці порожньо. ★

Підказка. Для зображень сніжинки і сонечка використайте шрифт Wingdings, шістнадцяткові коди 00AF або 0054 (сніжинка) і шрифт Wingdings 2, шістнадцятковий код 00F0 (сонечко). Колір надайте з використанням умовного форматування.

Практична робота № 16		
	Тема:	Упорядкування даних у таблицях
	Мета:	Набути практичних навичок створення електронних таблиць із використанням упорядкування даних, автоматичних та розширених фільтрів, проміжних висновків

Завдання. Створити таблицю “Річки Європи”, використовуючи такі дані: протяжність (км) і площа басейну (тис. кв. км): Волга – 3688 і 1350; Дніпро – 2285 і 504000; Дністер – 1 362 і 72 100; Дунай – 2850 і 817; Рейн – 1330 і 224; Ельба – 1150 і 148; Вісла – 1090 і 198; Луара – 1020 і 120; Урал – 2530 і 220; Дон – 1870 і 422; Сена – 780 і 79; Темза – 340 і 15. ✦ Упорядкувати таблицю за назвами річок. Знайти переклад назви кожної річки англійською, ввести в рядок із назвою річки додаткові комірки (додати стовпець “Англ. назва”), повторити впорядкування за алфавітом по цьому стовпцю. ★ Додати до рядка кожної річки країну (країни), через які вона протікає, вставити в таблицю додаткові стовпці.✦ Визначити найдовшу і найкоротшу річки. ⅄ Підрахувати сумарну й середню площу басейнів річок, сумарну та середню протяжність великих річок Європи (виконати з використанням проміжних висновків для сумарних значень). ✦ За допомогою умовного форматування виокремити назви річок, довжина яких більша за середню. ✦ Якщо для країни в таблиці більше ніж одна річка, підрахувати їх загальну протяжність з використанням фільтра і проміжних висновків. ★

Практична робота № 17	**Тема:**	Пошук даних у таблицях. Автоматичні та розширені фільтри
	Мета:	Набути практичних навичок створення електронних таблиць із використанням упорядкування даних, автоматичних і розширених фільтрів, проміжних висновків

Завдання.

1. Із веб-порталу фірми КРІ-Сервіс завантажити прайс-лист http://web-portal.kpiservice.com.ua/kpi_new/price/price_opt.zip (або інший подібний документ за вказівкою вчителя).

2. Відкрити аркуш “Вспомогательное оборудование” і встановити автофільтр (Дані/Фільтр/Автофільтр).

3. За допомогою встановленого фільтра вибрати з прайс-листа джерела безперебійного живлення back pro (кнопка списку поля Тип).

4. Відсортувати результат фільтрації за дрібногуртовою ціною, а потім за роздрібною ціною в порядку зростання (Дані/Сортування).

5. Створити новий аркуш, на який помістити отримані результати.

6. Надрукувати аркуш на мережевому принтері (або на віртуальному принтері, який створює документ формату *.pdf).

7. Закрити ЕТ.

СЛОВНИЧОК

Абсолютні адреси комірок – адреси комірок (і діапазонів комірок), які не змінюються при копіюванні формул.

Аргумент формули – змінна (числова або текстова), над значенням якої виконують певні дії для обчислення значення функції.

База даних – певним чином структурована сукупність даних, отриманих для певного класу об'єктів.

Відносна частота події – відношення частоти певної події до всієї кількості подій.

Відносні адреси комірок – адреси комірок (і діапазонів комірок), які змінюються при копіюванні (перенесенні) формул.

Запит – відомості щодо відбору об'єкта (об'єктів), містять значення властивостей, за якими об'єкт може бути зарахованим до певної групи.

Зведена таблиця – таблиця, в якій наведені підсумкові значення, одержані за спеціальними формулами з великих масивів даних.

Змішані адреси комірок – адреси комірок (і діапазонів комірок), які лише частково змінюються при копіюванні формул.

Масив – сукупність однотипних величин, кожен елемент якої нумерований.

Медіана – значення (у вибірці (наборі значень)), рівновіддалене від найбільшого та найменшого значень.

Мода – значення (у вибірці (наборі значень)), що трапляється найчастіше.

Розмах вибірки – різниця між значеннями, що повертають функції МАКС і МИН для вибірки.

Середнє – середнє арифметичне значення у вибірці (наборі значень).

Сортування – упорядковування даних за одним або кількома заданими критеріями.

Стандартний шаблон аркуша – Аркуш.xlt, який створюється для зміни використовуваного за замовчуванням формату новостворюваних порожніх аркушів.

Стандартний шаблон книги – Книга.xlt, яка створюється для зміни використовуваного за замовчуванням формату новостворюваних порожніх книг.

Умовне форматування – встановлення певного формату подання вмісту комірок залежно від значень числових величин, що містяться в цих комірках, значень логічних виразів, для обчислення яких використовуються дані інших комірок.

РОЗДІЛ 9. ОСНОВНІ ПОНЯТТЯ АЛГОРИТМІЗАЦІЇ

Алгоритми та їх виконавці; властивості алгоритмів; способи опису алгоритмів; базові алгоритмічні структури.

9.1. Поняття алгоритму

Людина постійно користується алгоритмами, часто не підозрюючи цього. Наприклад, є алгоритми розв'язування систем лінійних рівнянь, складання шкільного розкладу занять, поділу відрізка навпіл за допомогою циркуля та лінійки, запису заданих чисел у порядку спадання, додавання двох чисел у десятковій системі числення. Прикладами алгоритмів є також правила переходу перехрестя вулиць, порядок вимикання комп'ютера, послідовність дій у процесі приготування страви та багато інших.

Наведені алгоритми є простими, з ними стикається багато людей у повсякденній діяльності. Існують також складні алгоритми, що описують, наприклад, пошук пошкоджень у телевізорі; процес плавлення металу в доменній печі; посадку авіалайнера на заданий аеродром; управління стикуванням космічних кораблів.

Поняття алгоритму походить з математики. Здавна під час виконання обчислювальних робіт використовувалися різноманітні інструкції, таблиці, списки правил, за допомогою яких обчислювач міг, не вникаючи в сутність задачі, отримати правильний результат. Надалі списки формальних правил отримали назву алгоритмів.

✓ Слово **"алгоритм"** походить від імені видатного арабського математика IX століття аль-Хорезмі, який розробив правила виконання чотирьох арифметичних дій над десятковими числами.

Приклад. Опишемо алгоритм поділу відрізка AB навпіл за допомогою циркуля та лінійки.

1. Встановити ніжку циркуля в точці A.
2. Другу ніжку циркуля встановити у точці B.
3. Окреслити коло.
4. Встановити ніжку циркуля в точці B, не змінюючи його розхил.
5. Окреслити коло.
6. Через точки перетину кіл за допомогою лінійки провести пряму.
7. Позначити точку перетину проведеної прямої з цим відрізком літерою O.

Точка O і є серединою відрізка AB.

✓ *Алгоритми, під час виконання яких перетворюються лише числа, називають* ***чисельними.***

У багатьох алгоритмах об'єктом перетворення є літерна або літерно-цифрова інформація. Прикладами таких алгоритмів є алгоритм запису прізвищ в алфавітному порядку, алгоритм підрахунку в тексті кількості речень, що починаються зі слова МОРЕ, та ін.

Алгоритм – це послідовність інструкцій (вказівок), виконання яких у визначеному порядку за скінченну кількість кроків приводить до передбачуваного результату.

Живу істоту чи пристрій, що виконує алгоритм, називають **виконавцем** алгоритму. Сукупність команд, які може виконувати виконавець, називають **системою команд виконавця.**

Перевіряємо себе

1. Що називають алгоритмом?
2. Як і коли виникло поняття алгоритму?
3. Що є об'єктом перетворення в чисельних алгоритмах?
4. Сформулюйте поняття виконавця.
5. Що таке система команд виконавця?
6. Наведіть приклади виконавців алгоритмів.

Виконуємо

1. Складіть алгоритм побудови бісектриси кута за допомогою циркуля та лінійки.
2. Складіть алгоритм обчислення середнього арифметичного двох чисел.

9.2. Властивості алгоритму

До основних властивостей алгоритмів належать такі:

1. **Визначеність** (детермінованість) передбачає, що всі вказівки алгоритму мають бути чітко сформульовані так, щоб потенційні виконавці розуміли їх однозначно. Чітко визначеним має бути також порядок виконання вказівок.

2. **Дискретність** полягає в тому, що інструкції алгоритму виконуються покроково. Перехід до чергового кроку може відбуватися лише після виконання попереднього. Так, у алгоритмі поділу відрізка навпіл, який наведено вище, вказівка 3 може бути виконана тільки після завершення дій, визначених вказівкою 2.

3. **Результативність** означає, що коли початкові дані належать області допустимих значень, алгоритм має завершити роботу за скінченну кіль-

кість кроків і видати правильний результат. Якщо ж початкові дані не належать області допустимих значень, то алгоритм або не завершить свою роботу, або завершить, але задачу буде розв'язано неправильно. Результатом роботи алгоритму може стати також повідомлення про те, що задача не має розв'язання.

Зазначимо, що в разі повторного виконання алгоритму для одних і тих самих початкових даних послідовність виконання вказівок, а також результати не можуть змінюватися.

4. **Масовість** означає, що алгоритм призначений для розв'язування не однієї конкретної задачі, а певного класу однотипних задач. Початкові дані в однотипних задачах можуть бути різними, але вони не можуть виходити за межі допустимого діапазону.

5. **Формальність.** Людина, як виконавець алгоритму, часто діє неформально, по-своєму розуміючи команди. В інформатиці розглядають лише формальних виконавців, які виконують дії, не розуміючи (вони її не можуть їх розуміти) сутності задачі, що розв'язується. Якщо команда належить до системи команд виконавця і не має перешкод для її виконання, то вона буде виконана.

Команди зазвичай виконуються послідовно, в порядку їх запису. Якщо команди мають виконуватися непослідовно, про це зазначається у спеціальній вказівці.

Приклад 1. Опишемо алгоритм розв'язування квадратного рівняння.

1. Надати змінним a, b і c певні числові значення.
2. Обчислити значення дискримінанта d за формулою $d = b^2 - 4ac$.
3. Якщо $d < 0$, перейти до пункту 6, інакше виконати пункт 4.
4. Визначити x_1 і x_2 за формулами:

$$x_1 = \frac{-b - \sqrt{d}}{2a}; \quad x_2 = \frac{-b + \sqrt{d}}{2a}.$$

5. Перейти до пункту 7.
6. Дискримінант від'ємний. Рівняння розв'язків не має.
7. Кінець.

Приклад 2. Алгоритм обчислення найбільшого спільного дільника двох натуральних чисел A і B (алгоритм Евкліда).

1. Якщо $A = B$, то перейти до пункту 7, інакше виконати пункт 2.
2. Якщо $A > B$, то перейти до пункту 5, інакше виконати пункт 3.
3. Значення B зменшити на величину A.
4. Перейти до пункту 1.
5. Значення A зменшити на величину B.
6. Перейти до пункту 1.
7. Найбільший загальний дільник, який має значення A, присвоїти змінній D.
8. Кінець.

Неважко впевнитися, що алгоритм Евкліда дає правильний результат при різних значеннях натуральних чисел A і B. Наприклад, якщо $A = 23$ і $B = 65$, то $D = 1$, якщо $A = 119$ і $B = 51$, то $D = 17$.

Для розв'язання двох останніх завдань ми використовували різноманітні дані. Одні з них були початковими (a, в, c – у прикладі 1 та A, B – у прикладі 2), остаточні результати (x_1, x_2 – у прикладі 1 та D – у прикладі 2) отримали після опрацювання початкових даних. При цьому використовували допоміжні дані (наприклад, d – у прикладі 1).

Початкові (вхідні) дані називають ***аргументами****, остаточні (вихідні) дані –* ***результатами****, допоміжні дані –* ***проміжними****.*

Слід зазначити, що деталізація алгоритму залежить від того, на яких виконавців він розрахований. Наприклад, для того, хто знає таблицю множення, запис $A{\cdot}B$ однозначно визначає послідовність дій. Для того, хто знає правило додавання, але ще не знає таблиці множення, множення двох натуральних чисел A і B може бути описане у вигляді послідовності вказівок із виконання операції додавання.

Перевіряємо себе

1. Які основні властивості має алгоритм?
2. У якій послідовності виконуються вказівки алгоритму?
3. Чи різні виконавці можуть отримати різні результати роботи одного алгоритму?
4. Від чого залежить міра деталізації алгоритму?
5. Назвіть основні властивості алгоритмів.
6. Наведіть характеристики властивостей алгоритмів.
7. Що таке аргументи, результати та проміжні дані алгоритму?

Виконуємо

1. Опишіть алгоритм підрахунку кількості літер "а" у слові довжиною 10 символів.
2. Складіть алгоритм визначення центра кола, що описане навколо рівнобічного трикутника. Можна користуватися циркулем і лінійкою.

9.3. Способи описання алгоритмів

Існують різноманітні способи описання алгоритмів, наприклад, словесно-формульний, зображення у вигляді блок-схеми, запис мовою програмування. Розглянемо два найбільш поширених простих способи.

Словесно-формульний спосіб. Цей спосіб описання алгоритмів люди широко використовують у повсякденному житті як інструкції з експлуатації пристроїв, рецепти виготовлення ліків тощо. Перевага цього способу полягає в його простоті. Але такі описи алгоритмів часто є громіздкими, а їх вказівки різні виконавці можуть сприймати неоднозначно.

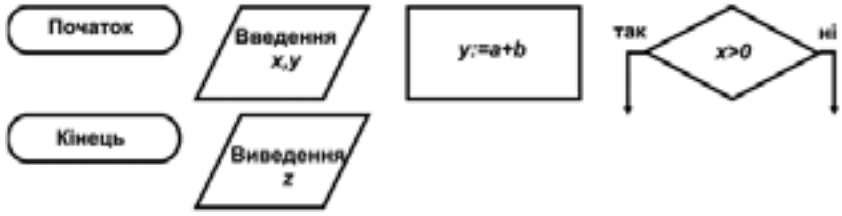
Для наочності й компактності запису алгоритмів у словесно-формульному вигляді використовується *оператор присвоювання*, що позначається символами ":=". Структура цього оператора така: **<змінна>:=< вираз>**

Під час виконання оператора присвоювання спочатку обчислюється значення виразу, розташованого праворуч від символів ":=", а потім отримане значення присвоюється змінній, записаній ліворуч.

Приклад. За допомогою оператора присвоювання алгоритм розв'язування квадратного рівняння (див. підрозділ 9.2) можна записати так:

1. Увести значення змінних a, b, c.
2. $d := b^2 - 4ac$.
3. Якщо $d < 0$, виконати пункт 6, інакше – пункт 4.
4. $x_1 = \dfrac{-b - \sqrt{d}}{2a}; \quad x_2 = \dfrac{-b + \sqrt{d}}{2a}$.
5. Перейти до пункту 7.
6. Дискримінант від'ємний. Рівняння розв'язків не має.
7. Кінець.

Зображення алгоритмів у вигляді блок-схеми. Переваги зображення алгоритму блок-схеми, порівняно зі словесно-формульним способом описання, полягають у наочності та простоті. Основні графічні позначення показані на рис. 9.1.



а *б* *в* *г*

Рис. 9.1. Основні графічні позначення на блок-схемах: термінатор (*а*); дані (*б*); процес (*в*); розгалуження (*г*)

1. *Термінатор* (рис. 9.1, *а*). Визначає початок або кінець алгоритму. У заокруглений прямокутник може записуватися слово "**Початок**" або "**Кінець**". Крім цього, разом зі словом "Початок" може бути вказане ім'я алгоритму або дія, яку він виконує.

2. *Дані* (рис. 9.1, *б*). Конструкція відображає дані, що можуть бути записані на різноманітних носіях. Може використовуватися для позначення операцій введення і виведення. У цьому випадку всередині пара-

лелограма слід записувати слово **“Введення” (“Виведення”)** і значення або імена змінних, що вводяться (виводяться).

3. *Процес* (рис. 9.1, *в*). Конструкція відображає процес опрацювання даних будь-якого типу (виконання певної операції або ж групи операцій). На кожний блок процесу можуть вказувати одна або декілька стрілок, а виходити з нього може лише одна стрілка.

4. *Розгалуження* (рис. 9.1, *г*). Конструкція розгалуження використовується з метою перевірки виконання певної умови та розгалуження обчислювального процесу залежно від результатів цієї перевірки. Всередині ромба записується умова, яка, власне, й перевіряється. Якщо така умова є істинною, то надалі обчислювальний процес здійснюється гілкою **“так”**, в іншому випадку – гілкою **“ні”**. Вказувати на конструкцію розгалуження можуть кілька стрілок, а виходити з неї можуть дві стрілки.

Всі конструкції блок-схеми алгоритму з’єднуються стрілками, які визначають послідовність виконання дій. Замість стрілок, що спрямовані згори вниз або зліва направо, можуть зображуватися лінії. Якщо кілька стрілок спрямовано в один блок, їх можна поєднати в одній точці.

Приклад. Сконструюємо блок-схему алгоритму обчислення значення виразу

$$y = \begin{cases} a^2 - bx, \text{ якщо } x > 0 \\ a^2 - \left(c + \dfrac{x}{b}\right), \text{ якщо } x \le 0. \end{cases}$$

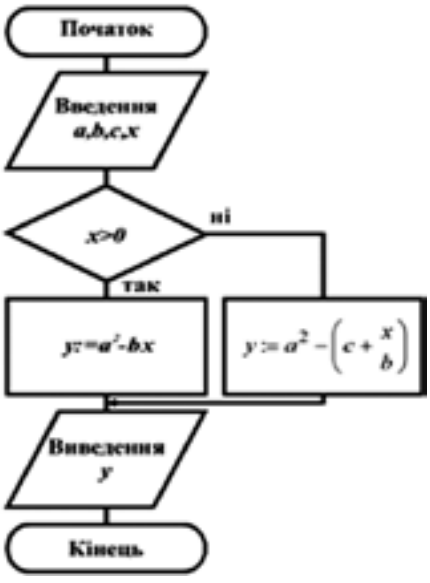


Рис. 9.2. Блок-схема алгоритму обчислення значення умовного виразу

Значення виразу залежить від чотирьох параметрів: a, b, c, x. Тому після термінатора, що позначає початок алгоритму, зобразимо блок даних, у якому надаватимуться значення цих параметрів. Вираз є умовним, адже те, за якою формулою обчислюватиметься значення y, залежить від виконання умови $x > 0$. Тому після блоку даних зобразимо розгалуження, на кожній із гілок якого розташуємо блоки процесу, де запишемо відповідні формули. За якою формулою не обчислювалося б значення y, його потрібно вивести. Отже, обидва блоки процесів з’єднуємо з блоком даних, яким позначимо виведення y. Завершуємо блок-схему термінатором “Кінець” (рис. 9.2).

Перевіряємо себе

1. Які основні переваги та недоліки має словесно-формульний спосіб описання алгоритмів?
2. У чому полягають основні переваги графічних способів описання алгоритмів?
3. Які позначення використовуються у блок-схемах алгоритмів?

Виконуємо

1. Складіть блок-схему алгоритму обчислення значення виразу

$$y = \begin{cases} kx + a, & \text{якщо } x > 0 \\ x, & \text{якщо } x \le 0 \end{cases}$$

2. Складіть блок-схему обчислення значення виразу

$$y = \begin{cases} \left(a^3 + b^2\right) \cdot c - 1, & \text{якщо } a = 0 \\ a^3 + b^2, & \text{якщо } a <> 0. \end{cases}$$

9.4. Базові алгоритмічні структури

Алгоритми довільного рівня складності можуть містити лише три основні структури: лінійну, розгалужену та циклічну.

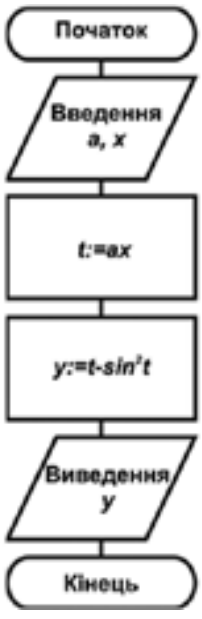


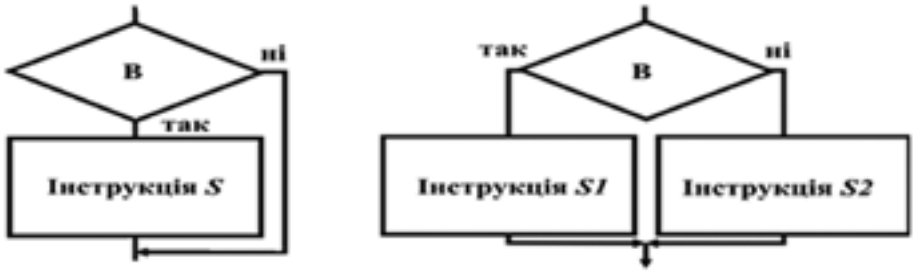
Рис. 9.3. Блок-схема лінійного алгоритму

Алгоритми з лінійною структурою. У таких алгоритмах вказівки виконуються послідовно в порядку їх запису. На рис. 9.3 подано приклад блок-схеми лінійного алгоритму обчислення значення виразу $y = ax - \sin^2 ax$.

Спочатку здійснюється введення значень a та x, які потім перемножуються, і одержаний результат присвоюється змінній t. Далі, від значення t віднімається значення $\sin^2 t$, і одержаний результат присвоюється змінній y. Завершується виконання алгоритму виведенням результату.

Алгоритми з розгалуженою структурою. У цих алгоритмах виконуються одні вказівки або інші, залежно від результату перевірки певної умови (або сукупності умов). Є два основних типи таких алгоритмів. *Одноальтернативне* розгалуження визначає таку послідовність дій: якщо умова B істинна,

виконується інструкція *S*, інакше вона не виконується (рис. 9.4, *а*). *Двохальтернативне* розгалуження задає таку послідовність дій: якщо умова *B* істинна, виконується інструкція *S*1, інакше – *S*2 (рис. 9.4, *б*).



а *б*

Рис. 9.4. Блок-схема алгоритмічної конструкції розгалуження:
а – одноальтернативне розгалуження; *б* – двохальтернативне розгалуження

Приклад. Побудуємо блок-схему алгоритму обчислення значення виразу

$$y = \begin{cases} ax, & \text{якщо } x > 0 \text{ і } a \geq 0; \\ a \cdot \sin x, & \text{якщо } x > 0 \text{ і } a < 0; \\ 0, & \text{якщо } x \leq 0. \end{cases}$$

Блок-схема алгоритму зображена на рис. 9.5. Цей алгоритм є прикладом алгоритму з перевіркою складної умови: якщо $x > 0$, здійснюється перевірка умови $a < 0$. Якщо умова $x > 0$ не виконується, змінній *y* присвоюється значення 0.

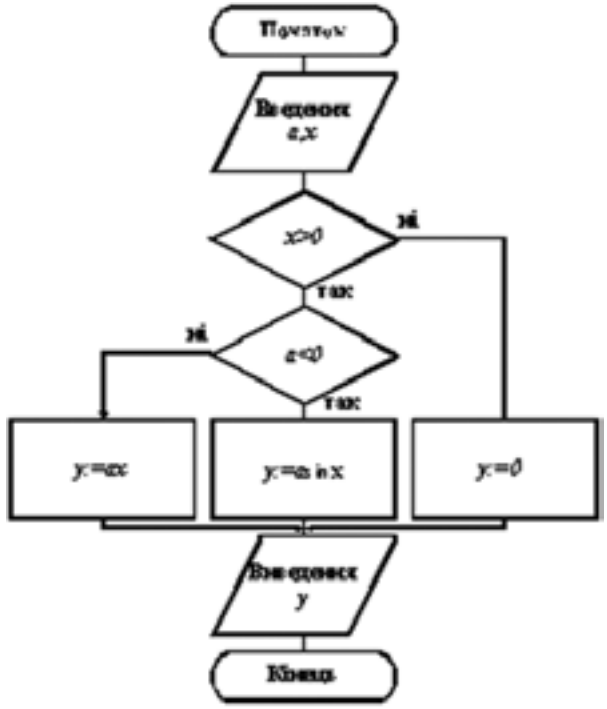


Рис. 9.5. Блок-схема алгоритму обчислення значення складеного умовного виразу

Алгоритми з повтореннями (з циклічною структурою). У таких алгоритмах одні й ті самі вказівки (команди) виконуються багаторазово для різних значень одних і тих самих змінних.

У циклічних структурах виділяють такі основні частини: *підготовча*, *робоча* та *керівна*. Підготовча частина містить вказівки (надалі будемо ототожнювати їх із операторами), які визначають початкові значення змінних.

Побудуємо блок-схему алгоритму обчислення значення функції $y = x^n$, де n – натуральне число (рис. 9.6). Праворуч від блок-схеми вказані значення, яких набувають змінні для $n = 4$. Найпростіший спосіб обчислення значення цієї функції полягає в послідовному множенні значення y на x, що має виконуватися n разів. Початковим значенням добутку буде одиниця, тобто $y_0 = 1$.

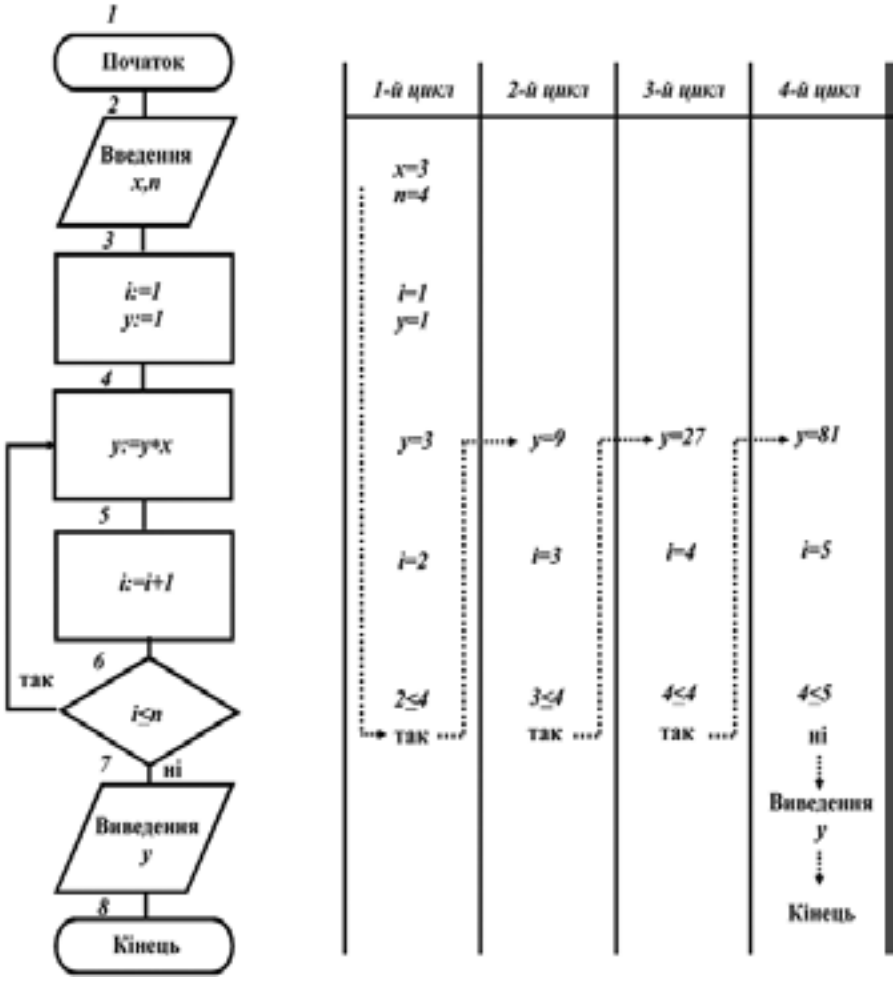


Рис. 9.6. Блок-схема та процес виконання циклічного алгоритму

Як бачимо з рис. 9.6, обчислення значення заданої функції здійснюється за такою схемою: $y_i = y_{i-1} \cdot x$, де $i = 1, 2, 3, \ldots, n$; $y_0 = 1$, тобто на першому кроці $y = 1^* x$, на другому – $y = x^* x$, на третьому – $y = x^{2*} x$ і т.д. На рис. 9.6 праворуч проілюстровано процес виконання алгоритму для $n = 4$.

У розглянутому прикладі підготовчу частину зображає блок *3*, в якому змінним x та y присвоюються одиниці. Робочу частину (яку ще

називають *тілом циклу*) утворюють вказівки, що визначають основні обчислення. В алгоритмі, блок-схема якого зображена на рис. 9.6, тілом циклу є блок *4*, а блоки *5* і *6* здійснюють керування циклом. У керівній частині перевіряється *умова завершення циклу*. Якщо ця умова істинна, виконання циклу припиняється, інакше – триває далі. В алгоритмі (рис. 9.6) умовою завершення циклу є вираз $i < n$. У цій умові перевіряється значення лічильника, яким у даному випадку є змінна i.

Лічильник – це змінна, значення якої вказує на те, скільки разів виконано тіло циклу.

Поняття допоміжного алгоритму. Допоміжним алгоритмом називають алгоритм, який повністю використовується в складі інших алгоритмів.

Наприклад, при створенні алгоритму знаходження суми $\sin(x) + \cos(x)$ допоміжним алгоритмом може слугувати алгоритм обчислення окремо $\sin(x)s$ і окремо $\cos(x)$; у процесі створення алгоритму поділу кута на чотири рівні частини допоміжним алгоритмом слугуватиме алгоритм поділу кута навпіл.

Для того щоб виконати допоміжний алгоритм, в основному алгоритмі використовується **команда виклику допоміжного алгоритму**. В записі цієї команди вказують ім'я допоміжного алгоритму і, можливо, значення параметрів, при яких має виконуватись допоміжний алгоритм.

Перевіряємо себе

1. Із яких алгоритмічних структур складаються алгоритми?
2. Які є різновиди алгоритмів із розгалуженою структурою?
3. Із яких частин складаються циклічні алгоритми?
4. Для чого використовується лічильник циклу?
5. Чому послідовність, розгалуження і повторення називають базовими алгоритмічними конструкціями?
6. Що таке аргументи, результати і проміжні дані алгоритму?
7. Які алгоритми називають допоміжними?
8. Які переваги надає застосування допоміжних алгоритмів?

Виконуємо

1. Складіть блок-схему алгоритму обчислення значення функції $y = a^2 + kx$.
2. Складіть блок-схему алгоритму обчислення значення виразу

$$y = \begin{cases} 3a + x, & \text{якщо } x < 0; \\ x, & \text{якщо } x \geq 0. \end{cases}$$

3.  Задано послідовність натуральних чисел: 1, 2, 3, ..., n. Складіть блок-схему алгоритму, за яким обчислюється сума парних членів цієї послідовності. ★

4.  Задано послідовність натуральних чисел. Складіть блок-схему алгоритму обчислення добутку n перших членів цієї послідовності. ★

5.  Складіть блок-схему алгоритму обчислення суми

$$s = x + \frac{x^2}{2} + \frac{x^3}{3} + + \frac{x^n}{n}.$$ ★

Практична робота № 18	**Тема:**	Способи представлення алгоритмів. Базові алгоритмічні структури. Типи алгоритмів
	Мета:	Набути навичок подання алгоритмів у вигляді блок-схем

Завдання.

1. Розробити алгоритм переходу регульованого перехрестя вулиць у словесній формі.

2. Накреслити основні позначення на блок-схемах алгоритмів.

3. Накреслити блок-схему алгоритму обчислення суми членів арифметичної прогресії.

4. Накреслити й описати блок-схему алгоритму обчислення виразу $y = a^3 + (a \cdot x + 3)$, якщо $x > 2$, інакше $y = a/2 + 4$.

 СЛОВНИЧОК

Алгоритм – послідовність інструкцій, виконання яких у визначеному порядку приводить до передбачуваного результату за скінченну кількість кроків.

Алгоритм із розгалуженою структурою – алгоритм, у якому порядок виконання вказівок залежить від результату певної умови або сукупності умов.

Алгоритм з циклічною структурою – алгоритм, у якому одні й ті самі вказівки виконуються багаторазово для різних значень одних і тих самих змінних.

Чисельний алгоритм – алгоритм, у процесі виконання якого опрацьовуються лише числа.

РОЗДІЛ 10. МОВИ ПРОГРАМУВАННЯ

Підготовка задач для розв'язування на комп'ютері; етапи розв'язування на комп'ютері; еволюція та класифікація мов програмування; поняття середовища програмування; структура програми, символи, ключові слова та ідентифікатори.

10.1. Основні етапи розв'язування задач за допомогою комп'ютера

Фахівця, який розробляє для комп'ютера програму, називають програмістом. Робота програміста з розв'язування задач за допомогою комп'ютера поділяється на два основних етапи:

– *підготовка задач для розв'язування на комп'ютері;*

– *безпосередня робота на комп'ютері над розв'язуванням задачі.*

Роботи на першому етапі виконуються переважно без використання комп'ютера. Це дуже відповідальний і суто творчий процес. Від того, як буде виконана ця робота, залежать достовірність отриманих результатів і швидкість розв'язання задачі на комп'ютері. На другому етапі деякі процеси можна автоматизувати й виконувати безпосередньо на комп'ютері.

10.1.1. Підготовка задач для розв'язування на комп'ютері

Незважаючи на те що останнім часом у програмуванні відбулися принципові зміни, а сучасні мови програмування дуже різняться, процес підготовки задач для розв'язування на комп'ютері зазвичай виконується у послідовності, відображеній на рис. 10.1.

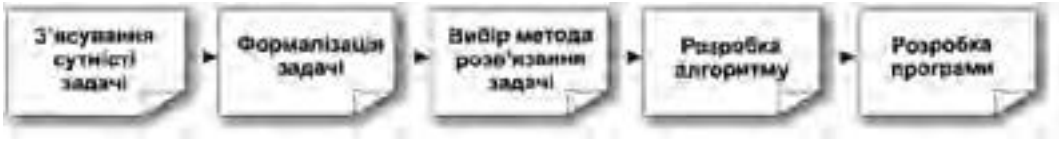


Рис. 10.1. Етапи підготовки задачі для розв'язування на комп'ютері

Розглянемо етапи підготовки задачі для розв'язування на комп'ютері детальніше.

1. *З'ясування сутності задачі.* Програмісту задача може бути поставлена в довільній, найчастіше словесній, формі. Насамперед необхідно зрозуміти умову задачі, визначити вхідні та вихідні дані, форму подання результатів тощо.

Наведемо приклади постановки задачі.

Задача 1. Визначити, чи є в журналі 8-а класу учень або учениця на прізвище Василенко.

Задача 2. На річці розміщені пункти А і В. Відстань між цими пунктами дорівнює S. З пункту А до пункту В за течією вирушає пліт, а з пункту В в пункт А одночасно з плотом вирушає катер. Через t годин вони зустрічаються і продовжують рух. Дійшовши до пункту А, катер, не зупиняючись, розвертається і прямує назад до пункту В, куди він прибуває разом із плотом. Визначити швидкість течії річки і швидкість руху катера в стоячій воді.

2. *Формалізація задачі*. Формалізувати задачу – означає побудувати математичну модель і описати її в математичних термінах. Описання основних властивостей об'єктів чи явищ за допомогою математичних виразів називають побудовою математичної моделі об'єктів чи явищ. Етап формалізації не завжди є обов'язковим. Формалізацію можна не виконувати для деяких обчислювальних задач із курсів шкільної математики, фізики, хімії, бо вони, власне, вже є формалізованими.

3. *Вибір методу розв'язування задачі*. Вибрати метод розв'язування задачі – означає знайти ефективні способи проведення обчислень за математичною моделлю. Так, для наведеної задачі про прибуток обрано рекурентну схему обчислення: при обчисленні прибутку на i-му році використовується значення прибутку на попередньому, $(i-1)$-му, році. За такої схеми обчислення скорочується кількість арифметичних операцій, порівняно з іншими методами розв'язування задачі.

4. *Розроблення алгоритму*. Розробити алгоритм – означає визначити послідовність інструкцій (вказівок), у результаті виконання яких задача буде розв'язана правильно.

5. *Розроблення програми*. Розробити програму – означає описати алгоритм мовою програмування (Pascal, Basic тощо). Нижче наведено приклад програми мовою Pascal, яка підраховує прибуток за кожний рік від початкової суми s, покладеної в банк під k відсотків річних. Спробуйте вручну виконати це завдання і ви переконаєтеся, які переваги надає комп'ютер.

```
program n10_01;                {заголовок програми з ім'ям n10_01}
var i: integer;                {оголошення змінної i цілого типу}
s, k, p: real;                 {оголошення змінних дійсного типу}
begin                          {початок операторної частини програми}
   writeln ('увести s, k');    {повідомлення про введення значень
                                змінних}
   readln (s, k);              {введення значень s, k}
   for i:=1 to 50 do           {оператор циклу}
   begin                       {початок тіла циклу}
      p:=s*k;                  {обчислення прибутку за поточний рік}
      s:=s+p;                  {обчислення накопиченої суми}
      writeln ('сума за ',i,' років=',s) {виведення прибутку за i років}
```

```
    end;                    {кінець тіла циклу}
    readln                  {призупинення виконання програми}
end.                        {кінець програми}
```

✓ *Запис алгоритмів мовою програмування високого рівня виконується точно і формально, тому їх подальша реалізація на комп'ютері легко автоматизується.*

Задачу автоматизованого перекладу алгоритму з мови програмування високого рівня мовою, вказівки якої може виконувати процесор (таку мову ще називають машинною), виконують спеціально призначені для цього програми – транслятори. Існують різні типи *трансляторів*, найпоширеніші з них – *компілятори*.

Для нескладних задач часто намагаються пропустити етап розробки алгоритму й відразу переходять до запису програм. Такий підхід є принципово неправильним, адже "чужу" програму, особливо складну, без алгоритму зрозуміти дуже важко. Тому потрібно виробити культуру оформлення алгоритмів на прикладах алгоритмізації простих задач. Записом алгоритму мовою програмування етап підготовки задачі до розв'язування на комп'ютері завершується.

Перевіряємо себе

1. Із яких етапів складається процес підготовки задач до розв'язування на комп'ютері?
2. Що означає "формалізувати задачу"?
3. Для яких задач етап формалізації не є обов'язковим?
4. Із чого складається мова програмування?
5. Яку функцію виконує транслятор?

10.1.2. Етапи розв'язування задач на комп'ютері

Процес розв'язування задач на комп'ютері зазвичай складається з кількох етапів (рис. 10.2).

Рис. 10.2. Етапи розв'язування задачі на комп'ютері

1. *Уведення програми*. Найчастіше текст програми вводиться до комп'ютера за допомогою клавіатури. Програма вводиться за допомогою текстового редактора, що вбудований у середовище програмування.

2. *Компіляція програми*. Компіляція – це автоматизований переклад програми з мови програмування високого рівня на машинну мову. Компіляцію доцільно здійснювати після того, як набрана програма записана в пам'ять комп'ютера. Програми, які виконують компіляцію, називають компіляторами. В результаті компіляції в найпростішому випадку буде отриманий один виконуваний файл, тобто файл із розширенням exe. Цей файл може одразу запускатися на виконання або бути збережений для виконання в майбутньому.

3. *Налагодження програми*. Налагодження програми полягає у виявленні та усуненні допущених у ній помилок. Деякі помилки (так звані синтаксичні помилки) знаходять компілятори. Повідомлення про такі помилки виводяться на екран. Синтаксичні помилки виникають, наприклад, у виразах, що містять недозволені знаки операції, зайві дужки тощо. Так, у виразі $y = a + (c/d - b))$ допущена синтаксична помилка, яка полягає в тому, що не вистачає однієї дужки, яка відкривається. Якщо під час уведення програми n10_01 буде відсутній оператор end;, ця синтаксична помилка також буде виявлена.

Інші помилки в програмі, які називають логічними, компілятори не виявляють. Наприклад, якщо у виразі $y = a + b$ замість знака "+" ввести знак "–", таку помилку компілятор не знайде. Логічні помилки повинен виявити й усунути програміст, виконуючи програму покроково та використовуючи інші засоби налагодження.

4. *Випробування програми*. Випробування проводиться для складних промислових і комерційних програм. Цей процес полягає в багаторазовому запуску програм з різними вхідними даними та в подальшому аналізі отриманих результатів, що проводиться з метою знайти помилки та перевірити ефективність роботи програми.

Для простих і навчальних програм на цьому етапі отримані результати аналізуються та порівнюються з контрольними результатами, які програміст обчислює вручну.

5. *Експлуатація програм*. На кожну реальну програму, призначену для масового використання, розробляється і затверджується необхідна документація. Програма разом з відповідною документацією передається для експлуатації замовникам або покупцям. Експлуатація таких програм здійснюється відповідно до інструкції, яка входить до комплекту документації.

Перевіряємо себе

1. Для чого компілюються програми?
2. Для чого здійснюється налагодження програми?
3. Які помилки в програмі виявляють компілятори?
4. З якою метою проводиться випробування програм?

10.2. Еволюція мов програмування

Початок розвитку мов програмування сягає XIX століття, коли англійський учений Чарльз Беббідж розробив механічну обчислювальну машину, програму для якої створила леді Ада Лавлейс, донька лорда Байрона (на честь цієї жінки названа мова програмування Ada). Однак мови програмування в сучасному розумінні фактично стали розвиватися з появою ЕОМ.

Мова програмування – це мова, призначена для опису алгоритмів і даних у вигляді, придатному для опрацювання комп'ютерами.

Нині існують сотні різних мов програмування та їх модифікацій, проте лише деякі здобули широке визнання. На різних етапах розвитку інформаційних технологій популярними були такі мови програмування: Fortran, Cobol, Algol-60, PL-1, Algol-68, Ada, C, C++, Basick, Pascal, Prolog, Delphi, Java.

Програми для перших ЕОМ розроблялися на так званих машинних мовах програмування. Програма, записана машинною мовою, виконується комп'ютером безпосередньо, без будь-яких її перетворень. Для запису машинних програм на папері використовувалися вісімкова та шістнадцяткова системи числення, а в комп'ютері такі програми зберігалися у двійковій системі числення. Програма цією мовою записується у вигляді послідовності машинних команд (див. таблицю).

Фрагмент програми машинною мовою

Номери комірок пам'яті, що містять команди	Команди		
	Код команди	Номери комірок, що містять дані	
00100	10	01051	02000
00101	15	06023	03002
00102	23	05012	07263
00103	15	02000	07005
...			
00164	37	–	–

Зауважмо, що для всіх обчислювальних машин першого покоління і для деяких машин другого покоління програми записувалися лише в машинних кодах. Програмістові доводилось деталізувати обчислювальний процес до рівня машинних команд і вручну розподіляти пам'ять комп'ютера між командами програм, вхідними даними, проміжними та кінцевими результатами. Реальні програми на машинних мовах містили десятки і сотні тисяч команд.

Із перших років використання обчислювальної техніки почався пошук мов програмування, доступних для широкого кола користувачів

і не пов'язаних із якоюсь конкретною обчислювальною машиною, які були названі мовами високого рівня. Першою такою мовою, що набула широкого визнання серед програмістів у всьому світі, стала розроблена у 1954 році в США мова Fortran (формульний транслятор).

У 1960 році група вчених із різних країн розробила мову програмування Алгол-60, яка орієнтувалася на розв'язування алгоритмічних задач і давала змогу обробляти ширший набір типів даних, ніж мова Fortran.

Із розвитком і вдосконаленням обчислювальних машин розширювалися й галузі їх застосування.

Потреба у розв'язанні найрізноманітніших задач, з одного боку, та поява й удосконалення нових обчислювальних машин – з іншого, поставили нові вимоги до мов програмування. Суть цих вимог полягає в тому, що мови мають забезпечити успішне розв'язування широкого кола задач, а також ефективне використання можливостей ЕОМ.

Мови програмування, які значною мірою відповідають таким вимогам, належать до класу універсальних. Найбільш відомими стали мова PL-1, розроблена в 1964 році, а також мова Algol-68. Згодом почала широко використовуватись універсальна мова Pascal, розроблена на початку 70-х років XX століття швейцарським математиком Ніклаусом Віртом.

Особливою популярністю в усьому світі користувалася мова Basic, проста й незамінна під час розв'язування порівняно нескладних задач. Однак вона мало придатна для розв'язування складних наукових, економічних та інших задач.

На початку 80-х років минулого століття, завдяки розробці мікропроцесорної техніки, з'являються та набувають поширення персональні комп'ютери, а на початку 90-х років створюється графічна технологія розробки програм. З цього часу почалася ера мов об'єктно-орієнтованого програмування.

Перевіряємо себе

1. Що називається машинною мовою програмування?
2. У чому полягають недоліки машинних мов програмування?
3. Які переваги мають мови символьного кодування порівняно з машинними?
4. Які основні відмінності між мовами програмування високого і низького рівнів?

10.3. Класифікація мов програмування

Існує велика кількість підходів до класифікації мов програмування. Розглянемо спрощену класифікацію, яка інтегрує кілька таких підходів і досить наочно відображає еволюцію мов програмування (рис. 10.3).

За ступенем залежності від апаратних засобів розрізняють мови низького, високого і надвисокого рівнів. Головна особливість мов програмування низького рівня полягає в тому, що програма розробляється за допомогою системи команд комп'ютера певного типу і може виконуватися на комп'ютерах саме цього типу. Найхарактерніша ознака мов високого рівня полягає в тому, що програмування ними абстраговане від особливостей архітектури комп'ютера.

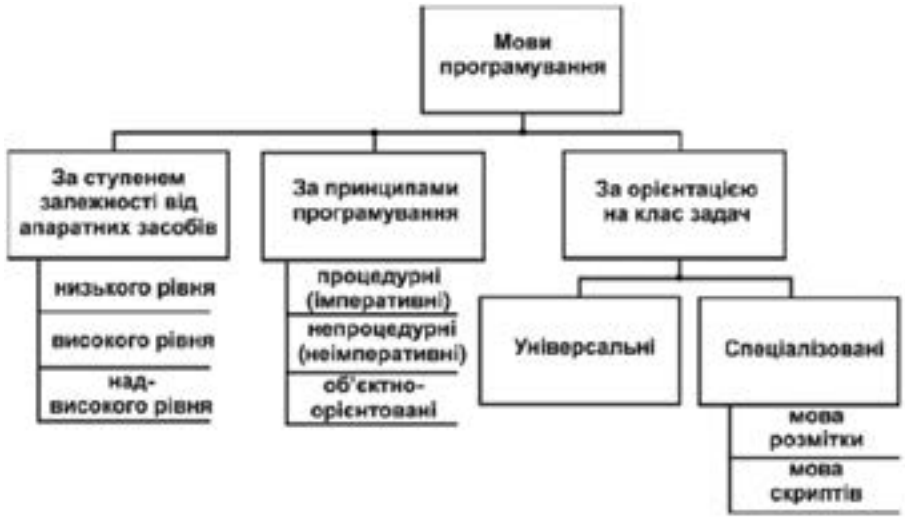


Рис. 10.3. Класифікація мов програмування

За принципами програмування мови поділяються на процедурні (імперативні), непроцедурні (неімперативні) та об'єктно-орієнтовані.

У програмах на **процедурних** мовах послідовно описуються зміни стану комп'ютера. Подібні програми обробляють дані в покроковому режимі, використовуючи інструкції, записані послідовно.

Непроцедурні мови програмування – це мови високого рівня абстракції. У цих мовах процедура пошуку розв'язку вбудована в інтерпретатор мови.

Об'єктно-орієнтовані мови програмування містять конструкції, що дають змогу оперувати такими поняттями, як **об'єкти** та **класи** об'єктів. Прикладами таких мов є C++, Object Pascal, Java, Delphi. При об'єктно-орієнтованому програмуванні програма розглядається як набір об'єктів, що взаємодіють між собою за допомогою обміну повідомленнями. В об'єкті поєднуються дані та засоби їх опрацювання. Кожен об'єкт характеризується набором властивостей і поведінкою. Поведінка об'єкта може змінюватись залежно від подій, які виникають унаслідок дій користувача (наприклад, переміщенням миші) чи роботи системи. Набір команд, які програмуються для виконання при настанні певної події, називають **обробником** події.

За орієнтацією на клас задач мови програмування поділяються на універсальні та спеціалізовані. **Універсальні** мови (до них належать PL-1, Algol, Pascal та ін.) призначені для розв'язування різноманітних задач. **Спеціалізовані** мови програмування враховують специфіку предметної галузі, тому область використання кожної з них дещо вужча. Сьогодні широко використовується декілька десятків спеціалізованих мов програмування.

Перевіряємо себе

1. За якими ознаками класифікуються мови програмування?
2. Які основні особливості мають мови програмування низького, високого та надвисокого рівнів?
3. Як мови поділяються за принципами програмування?
4. Які мови програмування належать до непроцедурних?
5. Що таке об'єктно-орієнтоване програмування?
6. Що таке об'єкт?
7. Що розуміють під подією?
8. Що таке обробник події?

10.4. Поняття середовища програмування

Досі ми розглядали мову програмування лише як засіб для запису алгоритму розв'язування задачі. Але сучасну мову програмування не можна відділяти від середовища розроблення програм, що містить компілятор мови й інструментальні засоби програмування. Інструментальні засоби забезпечують можливість виконання всього комплексу робіт із програмою на комп'ютері, а саме:

– введення і редагування тексту програми;

– компіляцію програми й автоматичний пошук помилок;

– налагодження програми: покрокове виконанням команд, перегляд значень змінних тощо;

– запуск програми на виконання;

– налаштування системи програмування з урахуванням потреб користувача.

Мова Pascal, як уже зазначалося, була розроблена на початку 70-х років минулого століття Н. Віртом і названа на честь видатного французького математика Блеза Паскаля (1623–1662). Хоча ця мова створювалася для навчання програмуванню, її можливості дають змогу розв'язувати набагато ширше коло задач.

Важливий внесок у розробку середовища для програмування мовою Pascal на персональних комп'ютерах зробила фірма Borland, створивши

низку компіляторів та інших інструментальних засобів. Окрім цього, вона випустила декілька версій інтегрованого середовища розробки програм мовою Turbo Pascal. Для Turbo Pascal 5.5 і старших версій були розроблені паралельні версії середовища – від Borland Pascal 5.5 до Borland Pascal 7.0. Подальше вдосконалення Borland Pascal 7.0 сприяло появі якісно нового середовища програмування, яке назвали Delphi. Мовою програмування в Delphi є об'єктно-орієнтована мова Object Pascal, в основу якої покладені конструкції Turbo Pascal 7.0. У Delphi широко використовуються візуальні засоби програмування.

Існує значна кількість середовищ програмування мовою Pascal. Серед них найбільшого поширення набули Turbo Pascal 7.0 та Free Pascal. Робота в середовищі Turbo Pascal починається з запуску файла bp.exe або turbo.exe. Після завантаження системи на екрані монітора відображається головне вікно Turbo Pascal (рис. 10.4), яке складається з трьох різних за своїм функціональним призначенням частин: рядка меню, робочої області та рядка стану.

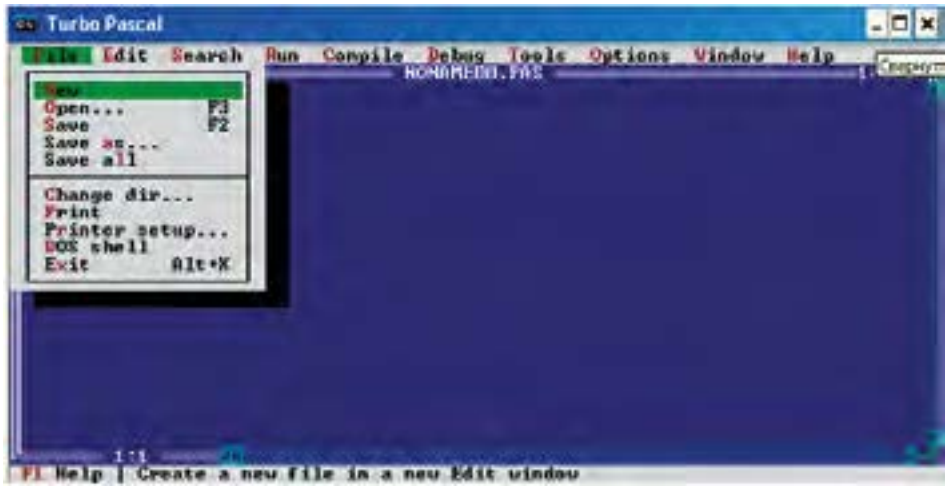


Рис. 10.4. Головне вікно Turbo Pascal 7.0 з відкритим меню File

Рядок меню, розташований у верхній частині головного вікна Turbo Pascal, містить десять меню (кожне з них складається з команд, призначених для виконання певного класу дій):

File – робота з файлами (створення, зберігання, друк файлів тощо);

Edit – редагування, копіювання, вставляння, вилучення окремих частин тексту;

Search – пошук тексту, заміна слів;

Run – виконання програми в звичайному та покроковому режимах;

Compile – компіляція програми;

Debug – налагодження програми;

Tools – використання допоміжних засобів;

Options – налаштування середовища програмування;

Windows – керування вікнами;

Help – отримання довідкової інформації.

Рядок стану, що розташований у нижній частині головного вікна програми, містить назви деяких доступних на поточний момент команд (тих, що застосовуються найчастіше) та контекстні підказки, якими можна скористатися під час роботи в середовищі програмування. Решта простору у вікні Turbo Pascal використовується для розміщення програм користувача.

Правила роботи з інтерфейсом системи Turbo Pascal 7.0 принципово не відрізняються від правил роботи з інтерфейсом системи Windows. Активація рядка меню може здійснюватися за допомогою миші або клавіші F10, а повернення до робочої області – за допомогою клавіші Esc. Вибрати той чи інший пункт меню можна одночасним натисканням "гарячої" клавіші (вона у назві пункту меню зображена напівжирним шрифтом) і клавіші Alt.

На рис. 10.4 показано вміст меню File. Команда New дає змогу створити новий файл. Якщо після створення файла виконати команду Save, буде запропоновано ввести його ім'я. Коли редагується створена раніше програма, під час запису на диск у файлі з розширенням bak зберігається її попередня версія.

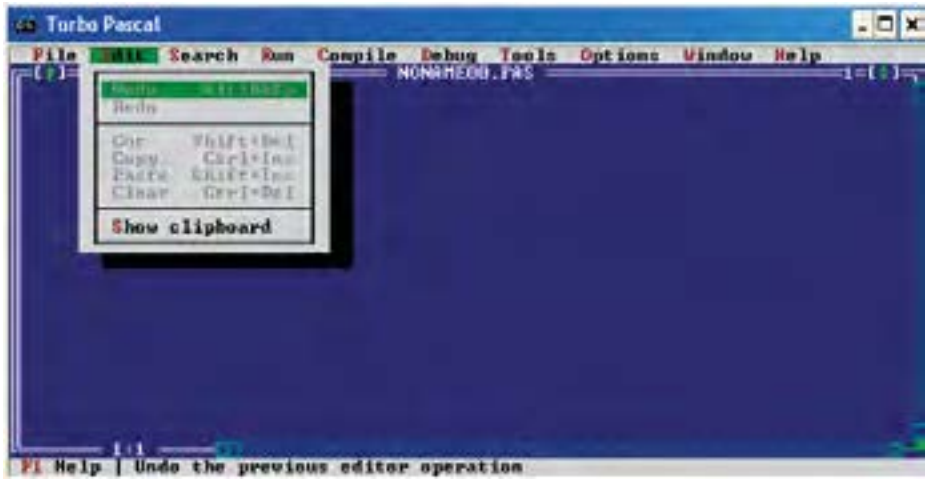


Рис. 10.5. Меню Edit середовища Turbo Pascal 7.0

Рис. 10.6. Меню Run середовища Turbo Pascal 7.0

Які команди містяться в меню Edit, показано на рис. 10.5, а які в меню Run – на рис. 10.6. Про призначення цих команд можна дізнатися з їх назв.

Перевіряємо себе

1. Які основні функції виконує середовище програмування?
2. Як запустити систему Turbo Pascal 7.0?
3. Із яких назв меню складається рядок головного вікна системи Turbo Pascal 7.0?
4. Які команди входять до складу меню File та меню Edit?
5. Яке призначення має меню Run?

Виконуємо

Завантажте систему програмування. Уведіть і виконайте програму n10_02. Проаналізуйте одержаний результат:

```
program n10_02;
var p, x, y, z: integer          {оголошення змінних цілочислового типу}
begin
   writeln (’уведіть x, y, z’); {повідомлення про введення}
   readln (x, y, z);             {введення значень змінних}
   p:=(x*x-z)*y;                 {обчислення значення виразу}
   writeln (’p=’, p);            {виведення значення виразу}
   readln                        {призупинення виконання програми}
end.
```

10.5. Робота в середовищі Pascal 7.0

Опишемо методику роботи в середовищі Turbo Pascal 7.0, якої має дотримуватись початківець. З цією метою навмисно введемо в програму n10_02 кілька помилок:

```
program n10_03
var p, x,, y, z: integer;        {оголошення змінних}
begin
   writeln (’увести x, y, z’)    {повідомлення про уведення значень
                                  змінних}
   redln (x, y, z);              {уведення значень змінних}
   p:=(x*x-z)*y;                 {обчислення виразу}
   writeln (‘p=’, p);            {виведення результату}
end.
```

Розглянемо процес налагодження цієї програми.

1. Після завантаження середовища Turbo Pascal 7.0 на екрані з'явиться вікно, зображене на рис. 10.5, і відкриється порожній файл Noname00.pas (якщо цей файл відсутній, його можна створити, скориставшись клавішею F10 та виконавши команду File→New). Уведемо програму n10_03, як це показано вище. У лівому нижньому куті вікна перше число вказує на номер рядка, друге – на номер стовпця, де розташований курсор.

2. Порівняємо програму на папері з програмою на екрані. Якщо є розбіжності, їх слід усунути, внісши зміни в програму на комп'ютері.

3. Збережемо файл. Для цього натиснемо клавішу F10 і виконаємо команду File→Save as. У вікні, що відкриється, слід набрати ім'я файла n10_03.pas і натиснути клавішу Enter. Цей файл буде збережений у каталозі <диск>:\TR\BIN\.

4. Компіляція програми може здійснюватися за допомогою команди Compile → Compile або Run → Run. Принципова відмінність між цими командами полягає в тому, що в першому випадку програма лише компілюється, а в другому – після завершення компіляції вона одразу виконується. Скористаємося командою Run→Run. На екран буде виведене повідомлення про помилку: Error 85:";" expected (очікується ";"), а курсор вказуватиме на другий рядок програми. Виправимо помилку, додавши до рядка n10_03 символ ";", після цього знову виконаємо команду Run→Run.

5. Уведемо початкові дані в тому порядку, в якому відповідні змінні перелічені в операторі readln, відокремлюючи їх одне від одного пробілами. Введемо, наприклад, 4 2 3. Після цього натиснемо клавішу Enter.

Щоб переглянути результат виконання програми, натиснемо клавіші Alt+F5. На екрані з'явиться повідомлення p=26. Для того щоб повернутися до програми, досить натиснути будь-яку клавішу.

6. Запишіть програму на диск. Нагадаємо, що під час виконання пункту 3 програму вже було збережено у файлі prog_03.pas, але тоді вона мала інший вигляд, тобто містила помилки. У програмі, яка відображається на екрані, помилок немає. Щоб зберегти цю програму з ім'ям n10_03.pas, досить виконати команду File→Save.

Для виходу із середовища Turbo Pascal потрібно виконати команду File→Exit.

Зазначимо, що навіть у тому випадку, якщо компіляція завершилась успішно, в програмі можуть бути помилки. Тому дуже важливим є етап налагодження програми, під час якого програміст має перевірити правильність її виконання. Для полегшення процесу налагодження в середовищі програмування передбачені спеціальні засоби. Зокрема, скомпільовану програму можна виконувати або окремими рядками (покроково), або до рядка, на якому встановлено курсор. У покроковому режимі перехід від рядка до рядка здійснюється за допомогою клавіші F7.

Перевіряємо себе

1. Які дії виконуються для збереження програми?
2. В який спосіб виконати компіляцію програми?
3. Як завершити роботу в середовищі TurboPascal?
4. Як визначити місце помилки в програмі, на яку вказує згенероване компілятором повідомлення про помилку?

Виконуємо

1. Уведіть програму n10_02.pas та виконайте її покроково.
2. Наберіть текст програми, що містить помилки (вона має назву n10_03). Відредагуйте її й виконайте в покроковому режимі.

10.6. Структура програми

У програмі мовою програмування високого рівня можна виокремити дві частини: розділ оголошень і основний блок. У деяких мовах програмування оголошення імен (ідентифікаторів) виконується неявно, тобто перша частина в програмі відсутня. На рис 10.7 зображена структура програми мовою Pascal.

У розділі оголошень означуються ідентифікатори даних, що використовуються в програмі. Нагадаємо, що змінні величини, хоча й у неявному вигляді, оголошуються також у звичайних математичних задачах. Так, в умові задачі “автомобіль рухався із пункту А в пункт В із середньою швидкістю 40,5 км/год протягом трьох годин” у неявному вигляді оголошено такі величини: час руху автомобіля – ціле число та швидкість руху – дійсне число.

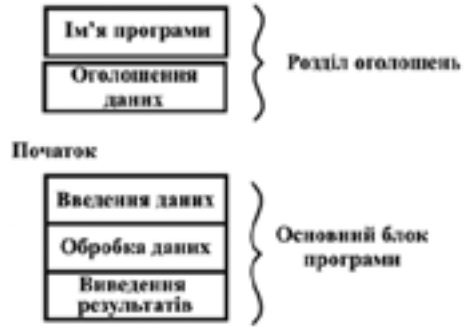


Рис. 10.7. Структура простої програми, записаної мовою Pascal

У описовому блоці програми описуються дії, пов’язані з уведенням даних, їх обробкою і виведенням результатів. Усі ці дії виконуються за допомогою спеціальних операторів. У будь-якій мові програмування високого рівня використовуються такі оператори: уведення даних; присвоювання; умовного переходу; циклу; виведення даних.

За допомогою операторів уведення вхідні дані присвоюються змінним, а за допомогою операторів виведення результати розв’язування задач виводяться на зовнішні пристрої, наприклад, на екран монітора. Опе-

ратор присвоювання використовується для оновлення значення змінної під час виконання програми, оператор умовного переходу – для опису розгалужених, а оператор циклу – циклічних обчислювальних процесів.

Наведемо загальну структуру програми, описаної мовою Pascal.

program<ім'я програми>;

{uses} <оголошення бібліотек та модулів, що підключаються до програми>;

{label} <оголошення міток>;

{const} <оголошення констант>;

{type} <оголошення типів даних>;

var<оголошення змінних>;

{procedure, function} <оголошення процедур та функцій>;

begin

<оператори>

end.

Розглянемо деякі елементи наведеної структури на прикладі найпростішої програми.

```
program n10_04;          {програма додавання двох чисел}
var a, b, c: integer;    {змінні цілочислового типу}
begin                    {початок основного блоку програми}
   readln (a,b);         {введення значень змінних a, b}
   c:=a+b;               {додавання двох чисел}
   writeln ('c=',c)      {виведення суми чисел}
end.                     {кінець програми}
```

Перший рядок програми – це її заголовок. Так зазвичай починаються програми, записані мовою Pascal. Після слова program вказується ім'я програми – воно може бути довільним. У даному прикладі програма має ім'я n10_04. Зверніть увагу, що заголовок, як і будь-який інший оператор, завершується символом крапка з комою. Хоча в останніх версіях мови Pascal заголовок програми не є обов'язковим, його варто використовувати, оскільки це полегшує роботу програміста.

У другому рядку оголошено, що в програмі будуть використовуватися три змінні з іменами a, b і c. Імена в оголошеннях розділяються комами. Ключове слово var є скороченням слова "variable" (змінна). Після імен a, b і c записано символ двокрапка і слово integer. У такий спосіб оголошується, що всі зазначені змінні мають один тип – цілочисловий. Про інші типи мова йтиме пізніше. Кожна змінна, що використовується в розділі основного блоку програми, має бути оголошена в розділі оголошень. У оголошенні змінної зазначаються її ім'я і тип.

У кожному рядку наведеного вище коду дається коментар, тобто інформація, яка полегшує читання та налагодження програми. Коментар обмежується фігурними дужками. Він не є обов'язковим. Третій рядок

містить слово begin, яким розпочинається основний блок програми, після нього крапка з комою не ставиться. Закінчується програма словом end з крапкою. У програмі може бути декілька слів begin, кожному з них відповідає слово end. Якщо слово end не останнє в програмі, то після нього записується крапка з комою (винятком є випадок, коли слово end розташоване перед словом else).

Четвертий рядок містить оператор уведення readln (a,b). Цей оператор зупиняє виконання програми і "очікує" на введення з клавіатури двох чисел, які відокремлюються пробілом. Перше число буде надане змінній a, а друге – змінній b.

У п'ятому рядку записано c:=a+b;. Цей оператор означає, що змінна c набуває значення, яке дорівнює сумі значень змінних a і b.

Шостий рядок містить оператор writeln (c), який здійснює виведення значення змінної c на екран. Оскільки він передостанній у коді, то крапка з комою після нього не ставиться.

Завантажимо та виконаємо програму для різних значень змінних a, b.

У мові програмування Pascal відсутні чіткі вимоги до зовнішнього вигляду програми. Але існують загальні методи й правила запису, які суттєво підвищують наочність програм і полегшують їх розуміння. Деякі з цих правил: якщо один (зовнішній) оператор містить декілька інших, внутрішні оператори записуються в окремих рядках і зсуваються вправо відносно початку зовнішнього; короткі оператори бажано розміщувати в одному рядку; суттєво полегшують розуміння програми коментарі, тому їх слід використовувати якомога частіше (коментарі записуються у фігурних дужках, вони не впливають на обчислювальний процес і призначені для пояснення дій, що виконуються); важливим є вибір імен змінної (бажано, щоб імена змінних відповідали їх змісту, а оголошення супроводжувались коментарями).

Перевіряємо себе

1. Які частини можна виділити в структурі Pascal-програми?
2. Які основні оператори входять до складу більшості мов програмування високого рівня?
3. Що записується в програмі на мові Pascal після слів label, var, program, const?

Виконуємо

1. Уведіть програму n10_04, відредагуйте її і запустіть на виконання.
2. Користуючись мовою Pascal, складіть програму для обчислення значення виразу: $y = a + b + c$.

10.7. Символи, ключові слова та ідентифікатори

Будь-яка мова програмування має набір допустимих символів (алфавіт). Алфавіт мови високого рівня містить літери, цифри, символи операцій, спеціальні символи тощо.

До алфавіту мови Pascal належать:

26 великих і маленьких латинських літер (компілятор не відрізняє великі літери від маленьких);

10 арабських цифр (від 0 до 9);

символ підкреслювання (_);

знаки арифметичних операцій: + (додавання чисел і об'єднання текстових рядків), – (віднімання чисел), * (множення чисел), / (ділення чисел);

знаки відношення: = (дорівнює), < (менше), >= (більше або дорівнює), > (більше), <= (менше або дорівнює), = (дорівнює), <> (не дорівнює);

синтаксичні знаки: . (крапка), , (кома), ; (крапка з комою), ” (подвійні лапки), ’ (апостроф), : (двокрапка), (символ пробілу), ? (знак питання), ! (знак оклику);

дужки: круглі ( ); квадратні [ ]; фігурні { };

спеціальні символи: $,%,#,@,& та ін.

У кожній мові програмування існує група службових слів (їх називають ще зарезервованими або ключовими), визначених синтаксисом мови винятково для створення мовних конструкцій. Таких слів у мові може бути від кількох десятків до кількох сотень. Службові слова не можна змінювати. Наведемо приклади службових слів мови Pascal: begin, var, const, end, for. Не можна використовувати звичні для математичних формул штрихи, індекси тощо. Наприклад, математичний вираз $y = 2a + \sin^2 x$ мовою Pascal записується так: y:=a*2+sqr (sin(x)).

В усіх мовах програмування для позначення різних об'єктів програм використовуються ідентифікатори, зокрема, ідентифікатори функцій, змінних, масивів, міток, процедур, файлів.

Ідентифікатор у мові Pascal – це послідовність латинських літер, цифр і символів підкреслювання, яка починається з літери або символу підкреслювання і не є ключовим словом.

Наприклад, слова a, z2, alfa, m21, dam_8 можуть використовуватися як ідентифікатори, а слова begin, 121, 5ab – ні.

Літерою в ідентифікаторі може бути будь-яка велика або маленька літера латинського алфавіту. В ідентифікаторі може міститися від 1 до 127 символів.

Перевіряємо себе

1. Які символи містить алфавіт мови Pascal?
2. Для чого використовуються ідентифікатори?

3. Як записуються ідентифікатори?

4. Які слова називаються ключовими?

5. Які ідентифікатори можна скласти із символів 7, A і B?

6. Запишіть всі ідентифікатори, які можна утворити з символів 7, _, A, в яких вони використовуються не більше одного разу.

Практична робота № 19	**Тема:**	Робота в середовищі програмування
	Мета:	Ознайомитись із середовищем програмування Turbo Pascal

Завдання. Створити код для обчислення виразу y = 3*(a + b)/2 й виконати його у середовищі Turbo Pascal 7.0.

1. Розробити код для обчислення наведеного виразу.
2. Використати змінні a і b типу integer, а змінну y – типу real.
3. Завантажити середовище Turbo Pascal 7.0.
4. Увести розроблений код.
5. Виконати компіляцію коду. Виправити всі синтаксичні помилки.
6. Виконати програму. Довести, що програма виконана правильно.
7. Зберегти програму.
8. Завантажити програму із файла. Виконати її в покроковому режимі.

СЛОВНИЧОК

Компіляція програми – автоматизоване перетворення програми з мови програмування високого рівня на машинну мову.

Мова програмування – сукупність символів, слів, команд і правил, за допомогою яких можна в формальному вигляді записати послідовність інструкцій для розв'язування на комп'ютері.

Розробка алгоритму – визначення послідовності інструкцій, у результаті виконання яких буде правильно розв'язана задача.

Формалізація задачі – побудова математичної моделі та її опис у математичних термінах.

РОЗДІЛ 11. ЛІНІЙНІ АЛГОРИТМИ

Типи даних та їх класифікація; поняття змінної, оператор присвоювання; стандартні типи даних і типи даних користувача; константи, арифметичні вирази; стандартні математичні функції; введення й виведення даних.

11.1. Дані й типи даних

Програмуючи у середовищі Scratch, навчаючись працювати з електронними таблицями, ви побачили, що різні дані мають різні властивості.

11.1.1. Класифікація типів даних

Тип даних – сукупність множини допустимих значень і множини допустимих операцій над цими значеннями. Тип даних визначає обсяг пам'яті, необхідної для зберігання його значень, а також структуру даних.

У мові програмування Pascal типи даних поділяються на статичні та динамічні. Для даних *статичних типів* пам'ять виокремлюється на етапі компіляції програми і використовується протягом усього часу її виконання. Для даних *динамічних типів* пам'ять виділяється і звільняється не лише на етапі компіляції програми, а й під час її виконання. Динамічні типи даних тут не розглядаються.

Рис. 11.1. Класифікація статичних типів даних у мові Pascal

Статичні типи даних поділяються на *прості* (скалярні) та *структуровані* (агрегатні). Прості типи даних своєю чергою поділяються на *стандартні* (базові) й такі, що визначаються програмістом – *типи даних користувача* (рис. 11.1).

У мові Pascal стандартні типи даних наперед означені в модулі System. Ця мова надає програмістові широкі можливості для створення власних типів даних.

Елементи даних структурованих типів об'єднуються в різні структури, зокрема, в масиви, записи, множини. Такі структури можуть бути однорідними або неоднорідними. В однорідних структурах усі елементи

Рис. 11.2. Класифікація структурованих типів даних у мові Pascal

можуть мати лише один тип, наприклад, бути числами або літерами. У неоднорідних структурах елементи можуть належати до різних типів. Наприклад, елементами даних у кожному рядку розкладу руху потягів є: номер потяга; час відправлення; кінцева станція; номер платформи.

На рис 11.2 подано класифікацію структурованих типів даних. Із зазначених структур розглядатимемо лише масиви.

Кожен тип даних позначається певним ідентифікатором. Ідентифікатори стандартних типів даних означені в мові програмування. Наприклад, у мові Pascal ідентифікатором стандартного символьного типу даних є слово char, цілочислового – integer, дійсного – real, логічного – boolean.

Перевіряємо себе

1. Які типи даних використовуються в мові Pascal?
2. У чому полягає основна відмінність статичних типів даних від динамічних?
3. Чим відрізняються прості типи даних від структурованих?
4. У якому модулі означуються стандартні типи даних?

11.1.2. Поняття змінної. Оператор присвоювання

Дані в комп'ютері зберігаються в комірках пам'яті, кожна з яких має свою адресу. Під час розроблення програми невідомо, в яких комірках зберігатимуться ті чи інші дані. Тому в мовах програмування використовуються змінні, що дає змогу програмістові оперувати не з адресами комірок пам'яті, а з ідентифікаторами.

Перш ніж користуватися змінною, слід оголосити її ім'я і тип. Ім'я змінної під час компіляції її оголошення зіставляється з певною областю пам'яті, й надалі програма посилатиметься саме на цю область.

Мова Pascal вимагає обов'язкового оголошення змінних.

У програмі мовою Pascal розділ оголошення змінних розміщується між заголовком програми і словом ***begin****.*

Оголошуючи змінну, її ім'я і тип, потрібно розділяти двокрапкою. Під час оголошення декількох змінних одного типу їх імена вказуються через кому:

<ім'я змінної 1>,...,<ім'я змінної N>: <тип змінних>;

В одній програмі можуть бути оголошені змінні різних типів. Наприклад:

```
program n11_1;
var
  a, b, c: real;
  j, n: integer;
  p, x: char;
begin
...
```

У наведеному фрагменті програми змінні *a, b* і *c* оголошені як дійсні, змінні *j* і *n* – як цілочислові, а змінні *p* та *x* – як символьні. Наведемо ще один приклад:

```
program n11_2;
var
  a1, a2: integer;
  a3, a4: boolean;
  a5, a6: real;
  a7, a8: char;
begin
```

У цьому прикладі *a*1 і *a*2 оголошені як цілочислові, *a*3 і *a*4 – як булеві, *a*5 і *a*6 – як дійсні, а змінні *a*7 і *a*8 – як символьні.

Для того щоб записати в область пам'яті, яка відповідає певній змінній, ті чи інші дані, їх слід присвоїти (надати) змінній. Зробити це можна за допомогою оператора присвоювання, що має такий вигляд:
<ім'я змінної> := <вираз>;

Результат обчислення виразу, розташованого праворуч від символів оператора присвоювання (:=), надається змінній, ім'я якої записане ліворуч від цих символів. Значення виразу має узгоджуватися з типом змінної, якій воно присвоюється. Наприклад, змінній дійсного типу можна присвоїти значення лише дійсного або цілочислового типу.

Приклад. Наведемо програму, що обчислює суму двох цілих чисел, уведених користувачем.

```
program n11.3;
var a, b, c: integer;          {оголошення змінних цілочислового типу}
begin
   writeln ('увести значення a, b');  {повідомлення про введення}
   readln (a, b);              {уведення значень змінних a, b}
   c:=a+b;                     {обчислення суми}
   writeln ('sum=', c);        {виведення результату}
   readln                      {призупинення виконання програми}
end.
```

Перевіряємо себе

1. Що означає термін “неявне оголошення змінних”?
2. В якому місці Pascal-програми оголошуються змінні?
3. Які дії виконує оператор присвоювання?
4. Що означає узгодженість типів даних під час присвоювання?

Виконуємо

1. Записати мовою Pascal програму обчислення середнього арифметичного двох чисел, уведених користувачем.
2. Записати мовою Pascal програму визначення довжини гіпотенузи прямокутного трикутника за умови, що відомі довжини обох катетів.
3. Увести й виконати програму n11.4. Проаналізувати отриманий результат.

```
program n11_4;
var a:integer;          {змінна цілочислового типу}
p: boolean;             {змінна логічного типу}
begin
  a:=3;                 {змінній a присвоюється значення 3}
  p:=a>2;               {вираз має значення true}
  writeln ('p=', p);    {виведення значення true}
  readln                {призупинення виконання програми}
end.
```

У цій програмі змінна **p**, що оголошена як булева змінна, набуде значення true, тому що $a=3$, а перевіряється умова $a>2$.

11.1.3. Стандартні типи даних

Нагадаємо, що імена стандартних типів даних означені в модулі **System**. На рис. 11.3 наведена класифікація стандартних типів даних мови Pascal. Розглянемо їх детальніше.

Цілочислові та дійсні типи – це числові типи даних. Цілочислові типи та їх характеристики наведені в табл. 11.1.

Рис. 11.3. Класифікація стандартних типів даних у мові Pascal

Таблиця 11.1

Цілочислові типи даних

Тип	Пояснення	Кількість байтів	Діапазон
Byte	ціле без знака	1	0...255
Shortint	ціле зі знаком	1	–128...127
Integer	ціле зі знаком	2	–32768...32768
Word	ціле без знака	2	0...65535
Longint	ціле зі знаком	4	–2147483648... 2147483647

У мові Pascal цілі числа можуть записуватись у десятковій і шістнадцятковій системах числення. Перед шістнадцятковими числами ставиться символ $, наприклад, $A5F.

Дійсні числа подаються в нормалізованій формі, у вигляді числа M – мантиси, помноженого на 10 у степені P – порядку. Отже, значення числа становитиме $M \cdot 10^P$. На записі мантиса відокремлюється від порядку літерою E. Для запису порядку використовується два розряди, а для запису його мантиси – від 8 до 20. У пам'яті комп'ютера дійсні числа зображаються в нормалізованій формі – таким чином, що мантиса M задовольняє нерівність $0 \le M < 1$. Будь-яке число можна подати в нормалізованій формі. Наприклад, десяткове число 0,0354 можна записати так: .354E-01, а число 852,4 – так: .8524E03.

Дійсні типи даних та їх характеристики наведені в табл. 11.2.

Таблиця 11.2

Дійсні типи даних

Тип	Кількість байтів	Цифр	Діапазон
Real	6	до 12	± (2.9E-39...1.7E38)
Single	4	до 8	± (1.5E-45...3.4E38)
Double	8	до 16	± (5.0E-324...1.7E308)
Extended	8	до 20	± (3.4E-4932...1.1E4932)
Comp	8	до 20	± (-9.2E+18)

Дані **логічного типу**, що в мові Pascal позначається ідентифікатором boolean, можуть мати лише два значення: true (істинне) або false (хибне). Змінним логічного типу часто присвоюють результати операцій порівняння або інших виразів, про які можна сказати істинні вони чи хибні.

Приклад. Розглянемо приклад використання логічної змінної.

```
var p: boolean: a, b: integer;
begin
  a:=3; b:=7;
  p:=a<b;
```

```
  writeln (p);
end.
```

Якщо a=3 і b=7, то буде виведено значення true, а якщо a=12 і b=4, то буде виведено значення false.

Символьний тип даних позначається ідентифікатором **char**. Дані символьного типу – це окремі символи, що записуються в одинарних лапках, наприклад, 'а', 'п'.

Цілочислові, символьні та логічні типи даних належать до так званих порядкових типів. Порядковими їх називають тому, що кожний їх елемент має свій порядковий номер – ціле число. Наприклад, літера "а" має в алфавіті перший номер, а літера "б" – другий (у повному наборі символів комп'ютера номери літер 'а' і 'б' будуть іншими). До будь-якого значення порядкового типу даних можна застосувати стандартну функцію ord, яка повертає порядковий номер цього значення. Для даних інших типів, скажімо для дійсних чисел, порядкових номерів не існує.

Рядковий тип даних, що позначається string, є структурованим. Значенням цього типу може бути послідовність символів довжиною до 255 елементів. Максимальна довжина рядка вказується в квадратних дужках. Наприклад, оголошення **var z: string**[12] означає, що значенням змінної z може бути рядок довжиною не більше 12 символів. Якщо максимальна довжина рядка не вказана, вважається, що вона дорівнює 255 символів. Рядкові дані беруться в одинарні лапки: 'Інформатика'. Доступ до окремих символів рядка здійснюється за допомогою операції індексування, яка позначається символами [ ]. Наприклад, якщо s – змінна рядкового типу, то значенням виразу s[i] буде i-й символ рядка s.

Приклад. Розглянемо фрагмент такої програми:

```
var s: string;
begin
  s:='Інформатика';
  writeln (s[3]);
end.
```

У результаті виконання цих операторів буде виведено символ "ф".

Стандартний **текстовий тип**, що позначається ідентифікатором **text**, використовується для оголошення текстових файлів.

Приклад. У наведеній нижче програмі оголошується змінна типу text, створюється текстовий файл і здійснюється запис і читання.

```
program f_1;
var f1:text;
a_1: string;
begin
assign (f1, 'file1.txt');   {зв'язування файла з файловою змінною}
rewrite (f1);               {створення нового файла}
```

```
    writeln (f1, учні 8-го класу вивчають Паскаль'); {запис у файл}
  close (f1);                {закриття файла для запису}
  reset (f1);                {відкриття файла для читання}
    readln (f1, a_1);        {читання файла}
    writeln (a_1);           {виведення на екран}
    readln;                  {призупинення виконання програми}
  close (f1);                {закриття файла для читання}
  end.
```

Стандартний **покажчиковий тип** позначається ідентифікатором pointer. Його значенням є адреси комірок оперативної пам'яті, зокрема, адреси змінних будь-яких типів. Користувач може оголошувати власні покажчикові типи даних.

Перевіряємо себе

1. Які числові типи даних використовуються в мові Pascal?
2. Які значення має логічний тип даних?
3. Як оголошуються змінні символьного типу?
4. У який спосіб оголошуються змінні рядкового типу?
5. Для чого використовується операція індексування?

11.1.4. Типи даних користувача

Типи даних користувача оголошує програміст. За структурою вони можуть бути простими або структурованими. У Pascal-програмах розділ оголошення типів даних користувача позначається ключовим словом type.

Перелічувальний тип даних оголошується користувачем шляхом перелічення у круглих дужках усіх його допустимих значень.

Наприклад:

```
type
  a1=(oct, nov, dec);
var
  x: a1;
```

Цей запис означає, що змінна *x* може набувати будь-якого з трьох значень: oct, nov, dec.

До значень перелічувального типу не можна застосовувати процедури введення–виведення (read, readln, write, writeln), а також арифметичні операції.

Для **інтервального типу** задається діапазон допустимих значень. Граничні значення відокремлюються двома крапками і мають мати однаковий порядковий тип (цілочисловий, символьний, логічний тощо).

Наприклад, після оголошень

```
type
  b1=1..30;
var
  x: b1;
```

змінна *x* може набути цілочислового значення з діапозону від 1 до 30.

Після оголошень

```
type
  letter='a'..'n';
var
  x: letter;
```

змінній *x* можна присвоїти символи від ‘a’ до ‘n’.

Як бачимо, значення інтервального типу в першому прикладі мають тип integer, а в другому – тип char.

Перевіряємо себе

1. Які існують типи даних користувача в мові Pascal?
2. Як оголошується перелічувальний тип даних?
3. Як оголошується інтервальний тип даних?
4. Чим відрізняються перелічувальний та інтервальний типи?

11.1.5. Константи

У мові Pascal використовуються прості й типізовані константи. Для кожної *простої константи* вказується тільки значення, а тип константи визначається компілятором за формою її запису. Як прості константи можуть використовуватися й константні вирази. Для *типізованих констант* вказуються не тільки значення, але й типи. Розглянемо спочатку прості константи.

У кожній мові програмування є правила зображення простих констант. У мові Pascal як прості константи можна використовувати десяткові цілі та дійсні числа, шістнадцяткові цілі числа, логічні, символьні та рядкові константи.

Цілі десяткові числа записуються в звичайному форматі й належать діапазону від –2147483648 до +2147483647, наприклад: –203; 25.

Дійсні десяткові числа записуються у такому вигляді:

<ціла частина>.<дробова частина> E <порядок>. Наприклад: 155E02; -31.4E-01. Так можна позначити числа 15500; –3,14 відповідно.

Шістнадцяткові цілі числа записуються шістнадцятковими цифрами, яким передує символ “$”, наприклад, $2AB, -$C2.

Логічна константа має значення true (істинне) або false (хибне).

Символьна константа – це будь-який символ у лапках: 'a', '+'. Для перетворення цілого числа на ASCII-символ використовується функція chr. Так, після виконання операторів

x1:=chr (97); x2:=chr(90);

змінна x1 набуде значення 'a', змінна x2 – значення 'Z'.

Рядкова константа – це будь-яка послідовність символів у лапках, наприклад: 'x+y-z='; 'Чемпіонат світу з футболу'. Максимальна довжина рядкової константи 255 символів.

У мові Pascal часто використовуються так звані *іменовані константи*, тобто константи, яким присвоюються імена (ідентифікатори), що надалі використовуватимуться замість констант. Тип константи визначається автоматично за її значенням. У оголошенні констант використовується службове слово const. Оголошення має таку структуру:

const <ідентифікатор> = <константа>;

Наведемо приклади оголошення констант:

```
const x=51;
  a='ліцей'; y=2.1E-02;
  z= 'ІДЕНТИФІКАТОР';
```

Типізовані константи можуть змінювати своє значення в процесі виконання програми. Однак формально типізовані константи відрізняються від іменованих лише тим, що в оголошенні після імен вказуються їх типи. Типизовані константи оголошуються в розділі оголошення констант:

const <ідентифікатор> : <тип>=<значення>;

Наведемо приклади оголошення типізованих констант:

const n:integer=7; s:string='файл'; sami:boolean=true;

Типізовані константи можуть мати будь-який тип (простий, рядковий, масив та ін.), крім файлового.

Приклад. У наведеній нижче програмі оголошується одна цілочислова константа і дві рядкові. Рядкові константи об'єднуються в один рядок.

```
program n11_5;
const n=20; x='іденти'; y='фікатор'; {оголошення констант}
var z:=string[n];                    {оголошення змінної типу string}
begin
  z:=x+y;                            {об'єднання рядків}
  writeln ('z=',z)                   {виведення слова ідентифікатор}
end.
```

Перевіряємо себе

1. Як мовою Pascal записуються дійсні десяткові числа?
2. В який спосіб записуються рядкові константи?

3. Як записуються іменовані константи?

4. Які переваги дає використання іменованої форми запису констант у порівнянні з неіменованою?

Виконуємо

1. Запишіть числа $125{,}4 \cdot 10^5$ і $-38{,}05 \cdot 10^5$ мовою Pascal.

2. Запишіть мовою Pascal оголошення рядкових констант СТОЛИЦЯ та x+y= .

3. Запишіть числа 25,4E-02 і 412,5E3 у звичайному десятковому форматі.

11.2. Оператори й вирази

11.2.1. Арифметичні вирази

Поняття виразу в мові програмування відповідає поняттю математичного виразу. Як і математичний вираз, вираз у тексті програми має бути побудований за певними синтаксичними правилами.

Вираз визначає порядок обчислення певного значення і складається з операндів, знаків операцій та круглих дужок. **Операндами** виразу в загальному випадку можуть бути константи, змінні та функції.

Найпростіший вираз складається з одного операнда. Прикладами такого виразу можуть бути: a – змінна, 5.3 – константа, sin(x) – функція. Приклад більш складного виразу: 3.5+a–sin(x).

Залежно від типу отриманого результату розрізняють **арифметичні** (результат арифметичного типу), **логічні** (результат логічного типу), **рядкові** (результат рядкового типу) та інші вирази .

У арифметичних виразах мови Pascal використовуються операції: додавання (+), віднімання (–), ділення (/), множення (*).

Записуючи арифметичні вирази, слід брати до уваги пріоритет операцій і дотримуватися певних правил. Перелічимо деякі з цих правил.

1. Вирази записуються в рядок без використання будь-яких підрядкових або надрядкових символів. Наприклад, запис $5*(a+b)^2$ є некоректним, його слід перетворити на такий: 5*sqr(a+b).

2. Операції мають бути задані в явному вигляді. Запис 5(a+b) вважається помилковим, а отже, його необхідно подати в такому вигляді: 5*(a+b).

3. Не дозволяється записувати дві операції безпосередньо одна за одною. Наприклад, вираз a*–b некоректний, його можна записати так: a*(–b) або –b*a.

4. Операції виконуються з урахуванням їх пріоритету, а операції, що мають однаковий пріоритет, – у послідовності їх запису (зліва направо). Правила пріоритету операцій такі: спочатку обчислюються значення функцій, потім виконуються операції множення та ділення, а після цього – операції додавання та віднімання. Якщо у виразі є круглі дужки, то насамперед виконуються операції, записані в них. Наприклад, значення виразу a+b*(c+d)*(c+a) обчислюється так: спочатку визначається сума c+d, потім – сума c+a, далі значення b множиться на c+d, одержаний результат – на c+a, а до нього додається a.

5. Кожній дужці, що відкривається, у виразі має відповідати дужка, що закривається. Отже, вираз a+b) є некоректним – його слід записати так: (a+b) або a+b.

Типи операндів виразів мають бути узгодженими. За цим правилом, не можна, скажімо, додавати символ до числа або віднімати від рядка булеве значення. У деяких випадках зведення різних типів операндів до одного типу виконується автоматично. Наприклад, якщо у виразі a+b число a є цілим, а число b – дійсним, то число a перетворюється на дійсне і виконується додавання дійсних чисел. Результатом такого виразу буде так само дійсне число.

У мові Pascal результатом операції додавання, віднімання та множення є ціле число, якщо обидва операнди також є цілими числами, в інших випадках – це дійсне число. Результатом операції ділення завжди є дійсне число.

Приклад. Запишемо мовою Pascal програму, що обчислює математичний вираз $\frac{b}{n(a+cm)}\sin(x-2z)$, де *a, n, m* – цілі числа, *b, c, x, z* – дійсні.

```
program n11_6;
var a, n, m: integer;          {оголошення змінних цілочислового типу}
   p, b, c, x, z: real;        {оголошення змінних дійсного типу}
begin
  writeln ('увести цілі значення a, n, m'); {повідомлення про введення
                                          значень a, n, m}
  readln (a, n, m);            {уведення значень змінних a, n, m}
  writeln ('увести дійсні b, c, x, z');     {повідомлення про уведення
                                          значень b, c, x, z}
  readln (b, c, x, z);         {уведення значень змінних b, c, x, z}
  p:=b*sin(x-2*z);             {обчислення виразу дійсного типу}
  writeln ('результат:', p/(n*(a+c*m)));  {виведення результату}
  readln
end.
```

Крім чотирьох стандартних арифметичних операцій у мові Pascal означені операції div і mod.

Результатом виконання операції div є ціла частина від ділення, а результатом операції mod – остача від ділення. Наприклад, якщо $x = 9$, а $y = 2$, то результатом операції x div y буде число 4, а результатом операції x mod y – число 1.

Перевіряємо себе

1. Із яких компонентів складається вираз у мові Pascal?
2. Назвіть різновиди операндів, що використовуються у виразах у мові Pascal.
3. Які основні правила застосовуються під час запису арифметичних виразів?
4. У яких випадках у мові Pascal результатом обчислення значення арифметичного виразу є дійсне число?

Виконуємо

1. Запишіть мовою Pascal такі вирази:

$$a=\pi r^2;\ S=4\pi r^2;\ A=\frac{1}{2}bh;\ y=c^2+a^2+b^2;$$

$$\frac{a}{b}+\frac{c}{d}=\frac{ad+bc}{bd};\ (x-y)(x+y)=x^2-y^2.$$

Обґрунтуйте, в яких виразах доцільно використовувати оператор присвоювання (:=), а в яких – ні.

2. Визначте, які вирази записані мовою Pascal коректно, а які ні.

а) (a+exp(x)*exp(x);
б) sinx+2*(a+b^2);
в) a^2+(3*a+b)/c;
г) sin(x)+a-*5*c.

3. Запишіть математичні вирази, які відповідають наведеним нижче виразам мовою Pascal.

а) sin(x)+sqr(x);
б) 3/sqr(1+x*x);
в) a+b/c+sqr(x);
г) (a+b)/(sin(x)-1).
д) sqr((p-a)*(p-b)*(p-c)*p);
е) 2*sin((a+b)/2)*cos((a-b)/2);
є) 1/2*a*b*sin(alfa).

11.2.2. Стандартні математичні функції

У математичних виразах часто використовуються різноманітні функції. Для обчислення часто вживаних функцій у кожній мові програмування високого рівня передбачені спеціальні програми. Ці програми є складовою мови програмування, а функції, які за їх використанням обчислюються, називаються стандартними.

Таблиця 11.3

Стандартні функції мови Pascal, призначені для оброблення числових значень

Позначення функції	Примітка
Pi	Повертає число 3.14
Random	Генерує випадкове число від 0 до 1
random (n)	Генерує випадкове число від 0 до n–1, де n – типу word
frac (x)	Повертає дробову частину числа x, аргумент – real, результат – real
int (x)	Повертає цілу частину числа x, аргумент – real, результат – real
round (x)	Заокруглює x до найближчого цілого, аргумент – real, результат – longint
trunc (x)	Повертає цілу частину числа x, аргумент – real, результат – longint

Для використання функцій у програмі необхідно записати її ім'я, а після нього в дужках вказати аргументи. Зрозуміло, що до моменту звернення до функції аргументу має бути присвоєне певне значення. Кількість стандартних функцій у більшості мов програмування сягає кількох десятків. Математичні функції мови Pascal, оголошені в модулі system і використовувані найчастіше, наведені в табл. 11.3.

Для всіх функцій, крім функцій $|x|$ і x^2, тип результату є дійсним. Для функцій $|x|$ і x^2 тип результату відповідає типу аргументу. У всіх стандартних функціях, наведених у табл. 11.3, аргумент може бути виразом, який набуває дійсних числових значень.

Перевіряємо себе

1. Які функції називають стандартними?
2. Яким є діапазон значень математичних функцій?
3. Як здійснюється переведення градусів у радіани?
4. Як за допомогою операції mod розпізнати парні та непарні числа?

11.3. Уведення даних з клавіатури

Вхідні дані зазвичай вводяться користувачем із клавіатури. Для цього в мовах програмування використовуються оператори введення даних. Ці оператори (в одній програмі їх може бути декілька) переривають виконання програми на той час, поки дані вводяться. Після їх уведення виконання програми буде продовжено.

У мові Pascal використовуються такі оператори (процедури) введення даних з клавіатури:

read (список імен змінних);
readln (список імен змінних);

Список імен змінних – це послідовність з одного або декількох розділених комами ідентифікаторів. Наприклад, read (a, b, c, d);. Числові дані під час їх уведення з клавіатури відокремлюються пробілами. Символьні дані пробілом не відокремлюються. Набір даних завершується натисканням клавіші Enter, і виконання програми продовжується.

Приклад. Нехай фрагмент програми має такий вигляд:

```
var a, b: real;
m, n: integer;
begin
read (a, b, m, n);
```

Припустімо, що під час виконання оператора read на клавіатурі було набрано: 2.5 5.2 7 12, а потім натиснуто клавішу Enter. У результаті змінні набудуть таких значень: a=2.5, b=5.2, m=7, n=12.

Процедура readln виконується так само, як і процедура read, з тією відмінністю, що після виконання процедури readln курсор буде переведено на початок наступного рядка, а після виконання процедури read курсор залишається на поточному рядку.

Приклад. Програма обчислення площі трикутника *s* за трьома його сторонами *a, b* і *c* може бути записана так:

```
program n11_8;
var a, b, c, s, p: real;          {оголошення змінних дійсного типу}
begin
  readln (a, b, c);               {уведення довжин сторін трикутника}
  p:=(a+b+c)/2;                   {обчислення напівпериметра трикутника}
  s:=sqrt(p*(p-a)*(p-b)*(p-c));   {обчислення площі трикутника}
  writeln ('s=', s);              {виведення площі трикутника}
  readln                          {призупинення виконання програми}
end.
```

Перевіряємо себе

1. Які процедури в мові Pascal використовуються для введення даних з клавіатури?
2. Чим відрізняється введення числових даних від уведення символьних даних під час виконання програми, записаної мовою Pascal?
3. Чим відрізняється процедура read від процедури readln?

Виконуємо

Визначити помилки в записі процедур уведення:

а) read a,b,c; б) read a;b;c;
в) read (a,b,c); г) readn (a;b;c);
д) readl (a,b,c); е) read (a b c);

11.4. Виведення даних

Результати розв'язання задачі виводяться на стандартні пристрої виведення. Такими пристроями на персональних комп'ютерах є екран монітора і принтер. Інколи доцільно виводити не тільки кінцеві, а й проміжні результати розв'язування задачі.

Для виведення даних на стандартні пристрої виведення у мовах програмування застосовують спеціальні оператори (або процедури). Дані, що виводяться, можуть бути числовими, текстовими, а також даними інших типів.

У мові Pascal використовуються дві процедури виведення даних на екран монітора:

write (<список виведення>);
writeln (<список виведення>);

У наведених операторах <список виведення> – це вирази, значення яких виводитимуться на екран монітора. Вирази відокремлюються комами. Виразами можуть бути змінні, константи, функції.

Відмінність між процедурами writeln і write полягає в тому, що після виконання першої курсор переводиться на початок наступного рядка, а після виконання другої – залишається на поточному.

Приклад. Наведемо фрагмент програми, де послідовно застосовуються процедури write.

```
var a, b, c: integer;
begin
...
  write (a, ' ', b, ' ');
  write (c);
```

Нехай змінні *a, b* і *c* набувають відповідно таких значень: 5, –72, 103. У результаті виконання указаних в програмі процедур ці числа будуть виведені в одному рядку.

```
5 –72 103
```

Зазначмо, що для відокремлення одних значень від інших слід виводити пробіли.

Приклад. Наведемо фрагмент програми, де застосовуються процедури write та writeln.

```
var a, b, c: integer;
```

```
begin
...
  writeln (a,' ',b,' ');
  write (c);
```

У результаті виконання цього коду значення змінних *a* і *b* будуть виведені в одному рядку, а значення змінної *c* буде виведене в наступному рядку:

5 –72
103

Приклад. Разом із даними бажано виводити інформацію, яка пояснює їх зміст.

```
var a, b, c: integer;
begin
...
  write ('a=', a, 'b=', b,'c=',c);
```

Числа будуть виведені в одному рядку: a=5 b=–72 c=103

Якщо дані вводяться з клавіатури, процедури writeln та readln бажано використовувати разом. Тоді у першій процедурі можна вказати, для яких змінних потрібно вводити дані.

Приклад.

```
program n11_9;
var a, b, c: integer;                {оголошення змінних типу integer}
begin
  writeln ('увести a, b');           {повідомлення про введення даних}
  readln (a, b);                     {уведення значень змінних}
  c:=a+b;                            {обчислення суми двох чисел}
  writeln ('перше число=', a);       {виведення першого числа}
  writeln ('друге число=', b);       {виведення другого числа}
  writeln ('сума=', c);              {виведення суми чисел}
  readln                             {призупинення виконання програми}
end.
```

У результаті виконання цієї програми буде виведена така інформація (за умови, що будуть уведені числа a=23 і b=46):

перше число=23
друге число=46
сума=69.

Перевіряємо себе

1. Які оператори мови Pascal використовуються для виведення інформації?
2. Чим оператор write відрізняється від оператора writeln?
3. У якому вигляді виводяться логічні значення в мові Pascal?

Виконуємо

1. Знайдіть помилки в операторах виведення:
   а) write (a; b); б) writeln (”a=”; a,,b)
   в) write a, b); г) write (a:b);

2.  Визначте, які повідомлення будуть виведені в результаті виконання таких операторів:
   а) writeln(7*2.1);
   б) writeln(’СЛАВА’);

11.5. Розробка програм для лінійних алгоритмів

Розглянемо декілька прикладів розроблення програм для лінійних алгоритмів.

1. Комп’ютер видає на екран запит учневі: “Ваше ім’я”. Після його введення висвітлюється новий запит – “який клас”. Після введення відповіді на друге запитання на екрані висвітлюється: “Ви навчаєтеся ...у класі”.

Позначимо ім’я учня змінною x, а клас – змінною y. Уважатимемо, що найбільша кількість символів у імені учня та назві класу не перевищує 10. Нижче наведена програма, що реалізує це завдання.

```
program n11_10;
var x, y: string [10];          {оголошення змінних рядкового типу}
begin
  writeln (‘як Вас звати?’);      {прохання ввести ім’я}
  readln (x);                     {уведення імені}
  writeln (’у якому ви класі?’); {прохання ввести клас}
  readln (y);                     {уведення класу}
  writeln (x, ’, Ви навчаєтеся у ’ , y, ’класі’); {виведення імені та класу}
  readln                          {призупинення виконання програми}
end.
```

2. Василь, Микола та Оксана збирали гриби. Потім вони зважили зібрані гриби й поділили порівну. Розробити програму, за допомогою якої визначається, скільки кілограмів грибів дісталося кожному з друзів.

Позначимо масу грибів, зібраних Василем змінною a, масу грибів Миколи – змінною b, а масу грибів Оксани – змінною c. Змінна s – загальна маса грибів. Програма, що реалізує це завдання, має ім’я n11.11. Виконайте програму для різних значень змінних і доведіть, що вона функціонує правильно.

```
program n11_11;
var a, b, c, s; real;           {оголошення змінних дійсного типу}
begin
  writeln (’увести a, b, c’);     {повідомлення про введення значень
                                   змінних}
```

```
  readln (a, b, c);                {уведення значень змінних}
  s:=(a+b+c)/3;                    {обчислення середнього значення}
  writeln ('середне=', s);         {виведення середнього значення}
  readln                           {призупинення виконання програми}
end.
```

3. У квадрат зі стороною a вписано коло. Розробити програму визначення різниці площ квадрата й утвореного круга.

Уважатимемо, що значення змінної a вводиться з клавіатури. Позначимо площу квадрата змінною s_1, площу круга змінною s_2, а різницю площ квадрата і круга – змінною r. Один із варіантів програми, що реалізує завдання, наведений нижче.

```
program n11_12;
var a, s1, s2, r: real;    {оголошення змінних дійсного типу}
begin
  writeln ('увести значення змінної a');  {повідомлення про введення
                                           значення змінної a}
  readln (a);                      {уведення значення змінної a}
  s1:=a*a;                         {обчислення значення площі квадрата}
  s2:=3.14*s1/4;                   {обчислення значення площі круга}
  r:=s1-s2;                        {обчислення різниці площ}
  writeln ('різниця=', r);         {виведення різниці площ}
  readln                           {призупинення виконання програми}
end.
```

Виконайте програму і переконайтеся, що для значення змінної a=4 результат різниці r=3.4400000000E+00.

4. Відстань між пунктами А і В дорівнює l км. Одночасно назустріч один одному з пункту А починає бігти Петро, а з пункту В їхати на велосипеді Марійка. Швидкість руху Петра дорівнює v_1 км/год, а Марійки – v_2 км/год. Потрібно розробити програму, яка визначає, через скільки годин вони зустрінуться.

Уважатимемо, що значення всіх змінних уводяться за допомогою клавіатури. Позначимо час до зустрічі змінною t. Програму обчислення часу до зустрічі можна подати в такому вигляді:

```
program n11_13;
var t, l, v1, v2: real;          {оголошення змінних дійсного типу}
begin
  writeln ('увести значення l, v1, v2');  {повідомлення про введення
                                           значень змінних}
  readln (l, v1, v2);              {уведення значень змінних}
  t:=1/(v1+v2);                    {обчислення часу зустрічі}
  writeln ('зустрінуться через ', t, ' годин');{виведення обчисленого
                                   результату}
  readln                           {призупинення виконання програми}
end.
```

Виконайте програму і перевірте, чи правильно вона функціонує.

5. У басейні ємністю 10000000 л залишилося p літрів води. До нього підведені два насоси, один з яких подає в басейн кожної хвилини a літрів води, а другий викачує кожної хвилини b літрів води ($a>b$). Розробити програму, що визначає, через скільки хвилин басейн наповнився, за умови, що насоси вмикаються одночасно.

Використаємо такі змінні: t – кількість хвилин, за які басейн наповниться повністю; m – кількість літрів води, яку потрібно накачати, щоб басейн наповнився. Програма, що реалізує це завдання:

```
program n11_14;
var p, a, b, m, t: real; {оголошення змінних дійсного типу}
begin
  writeln ('увести значення змінних p, a, b');      {повідомлення про
                                                     введення}
  readln (p, a, b);        {уведення значень змінних p, a, b}
  m:=10000000-p;           {кількість літрів, які необхідно накачати}
  t:=m/(a-b);              {обчислення часу наповнення басейну}
  writeln ('наповниться через ',t,' хвилин');  {виведення результату}
  readln                   {призупинення виконання програми}
end.
```

Виконуємо

1. Розробіть алгоритм і програму обчислення значення функції $y=e^x+(5x-1)(x+1)+x^2$ для будь-якого значення x. ★

2. Розробіть алгоритм і програму обчислення значення функції $y=(a^2+b+c)/(a^2+b+c-4ac)$ для додатних значень a, b, c за умови, що знаменник не дорівнює нулю. ★

3. Автомобіль і пішохід одночасно розпочали рух з пункту A в пункт B, які розташовані на відстані l км один від одного. Автомобіль, досягши пункту B, не затримуючись, повертає і рухається в пункт A. Через 4 год після початку руху він зустрічає пішохода на відстані $l/4$ від пункту A. Розробіть алгоритм і програму визначення, з якою швидкістю рухалися автомобіль і пішохід. ★

4. У посудині зберігалося n літрів спирту 96°. Потім у посудину додали m літрів води. Розробіть програму, щоб визначити, скільки градусів має суміш. ★

5. О 8 годині ранку з Києва до Харкова виїхав автобус. О 8.30 з Харкова до Києва виїхало маршрутне таксі, яке рухалося на 10 км швидше, ніж автобус. Об 11.30 вони зустрілися. Розробіть алгоритм і програму визначення швидкості руху автобуса і маршрутного таксі. Відстань від Києва до Харкова знайдіть в Інтернеті. ★

Практична робота № 20		
	Тема:	Розроблення лінійних алгоритмів та їх реалізація у вигляді програм
	Мета:	Навчитися програмувати лінійні алгоритми

Завдання

Є кімната довжиною m м, шириною n м і висотою h м. У одній стіні є двері висотою h_1 м і шириною n_1 м, а в другій – вікно шириною n_2 м і висотою h_2 м. Скільки потрібно повних рулонів шпалер шириною 1 м і довжиною 10 м, щоб оклеїти стіни кімнати?

1. Розробити програму обчислення потрібної кількості рулонів шпалер.
2. Увести програму. Виконати її компіляцію. Виправити всі синтаксичні помилки і запустити програму на виконання.
3. Довести, що програма виконується правильно.
4. Зберегти програму.

Практична робота № 21		
	Тема:	Розроблення лінійних алгоритмів та їх реалізація у вигляді програм з використанням текстових файлів
	Мета:	Ознайомитися з використанням об'єктів типу текстовий файл

Завдання

Розробити програму, за допомогою якої створюється текстовий файл, що містить такі дані учня: прізвище; клас; улюблений предмет.

За допомогою цієї програми здійснюється також виведення змісту файла на екран.

Приклад вмісту файла: Костенко, 8 клас, інформатика.

СЛОВНИЧОК

Дані логічного типу – дані, що набувають лише двох значень (true і false).
Операнди виразу – константи, змінні та функції.
Рядковий тип даних – послідовність символів довжиною до 255.
Символьний тип даних – окремі символи, що записуються в одинарних лапках.
Стандартні функції – найчастіше вживані функції, для яких розроблені спеціальні програми.
Структуровані типи даних – дані, що об'єднуються у структури (масиви, записи, множини та інші).

РОЗДІЛ 12. ЕЛЕМЕНТИ АЛГЕБРИ ЛОГІКИ

Історія виникнення математичної логіки; логічні операції й вирази; основні закони булевой алгебри; використання булевої алгебри для розв'язування логічних задач.

12.1. Предмет та історія виникнення математичної логіки

Математична логіка вивчає питання застосування математичних методів для розв'язування логічних задач. Вона також є теоретичною основою синтезу логічних схем, на базі яких побудовано комп'ютер і будь-який сучасний цифровий електронний пристрій.

Логіка – (від грец. logos – слово, думка, поняття, закон) – наука про загальні закони розвитку об'єктивного світу й пізнання їх. Вона має декілька напрямів: формальна логіка, імовірна логіка, математична логіка та інші. **Математична логіка** – складова формальної логіки. **Формальна логіка** – наука про форми й закони людського мислення. Основними формами мислення є **поняття, судження** й **твердження**.

Поняття *– така форма мислення, коли відокремлюються найваго-міші ознаки предмета або класу предметів. За їх допомогою відрізняють цей предмет від інших.*

Наприклад, за допомогою певних ознак відрізняються такі об'єкти: трактор, комп'ютер, літак, пароплав.

У процесі ***судження*** *встановлюється або заперечується зв'язок між предметом і його ознакою, відношення між предметами або сам факт існування предмета.* Судження може бути істинним або хибним. У мові воно висловлюється тільки розповідним реченням. Наприклад, річка Дніпро впадає в Чорне море.

Твердження (умовивід) *– форма мислення, яка дозволяє на основі одного або кількох тверджень-посилань отримати нове судження (значення або висновок).* Умовивід може бути істинним, якщо істинним є посилання, інакше він може бути хибним. Наприклад, якщо є такі посилання: "на диску є 2,3 МБ вільного простору" і "файл bosh.doc займає 1,9 МБ", то істинним є такий умовивід: "файл bosh.doc може бути записаний на диск".

Термін **судження** в математичній логіці називається **висловлюванням**, а термін *твердження* – **предикатом**. Висловлювання – окреме конкретне твердження, про яке можна однозначно сказати істинне воно чи хибне. Приклади висловлювань: 23 – просте число; пінгвіни – це птахи; 2 + 2 = 5; сума внутрішніх кутів трикутника дорівнює 180°; Туніс розташований у Європі.

При вивченні висловлювань повністю відволікаються від їх конкретного змісту, враховуючи лише їх істинність чи хибність. Тому висловлювання можна позначати, наприклад, великими латинськими літерами, скажімо, А – "Гривня дорівнює 100 копійкам" і надалі оперувати їх позначеннями.

Предикати *– це твердження, що містять змінні.*

Приклади предикатів: d – змінна дійсного типу; $a > b$; N – видатний письменник; x – корінь рівняння $5t = 10$.

Якщо замість змінних підставити їх конкретні значення, то предикати стануть висловленнями, які входять до складу предиката. Наприклад, якщо a=2, а b=3, то предикат "$a>b$" стає хибним висловленням, а якщо N дорівнює "Іван Франко", то відповідний предикат стає істинним висловленням.

Перевіряємо себе

1. Що є предметом науки логіка?
2. Що вивчає математична логіка?
3. У чому полягає сутність булевої алгебри?
4. Які є основні форми мислення?
5. Що називають висловлюванням?
6. Наведіть приклади істинних висловлювань.
7. Наведіть приклади хибних висловлювань.
8. Чи може висловлювання одночасно бути істинним і хибним?
9. Що називають предикатом?
10. Який зв'язок існує між предикатом і висловлюванням?
11. Чи є висловлюванням речення "Котра зараз година?"

12.2. Логічні операції та вирази

Булева алгебра ґрунтується на кількох логічних операціях, серед яких найчастіше використовуються такі: **заперечення** (НІ, NOT), **кон'юнкція** (І, AND), **диз'юнкція** (АБО, OR). Ці операції є базовими, тому що за допомогою цих операцій можна виразити будь-яку іншу.

Заперечення є унарною операцією, тобто вона застосовується для однієї змінної. Ця операція позначається рисочкою над змінною (наприклад, $\bar{a}$).

Операція ***кон'юнкція*** *позначається символом* ***&***. Наприклад, кон'юнкція над змінними x і y записується так: x&y. Часто символ & взагалі опускається, між змінними не вказується символ або ставиться символ крапки, наприклад, xy або $x{\cdot}y$.

Операція диз'юнкція позначається символом ∨, наприклад, x ∨ y.

У мові наведені операції та їх комбінації відповідають частці "ні" та сполучникам "і", "або", "якщо...то", "тоді й тільки тоді" та іншим. Висловлюванням у булевій алгебрі ставлять у відповідність логічні змінні, які можуть набувати одного з двох значень: істинно і хибно. Наприклад, висловлювання "швидкість автомобіля дорівнює 100 км" можна позначити змінною A. У мовах програмування істинність і хибність можуть позначатись іменованими константами true та false, конкретні значення яких залежать від обраного способу кодування розробниками мов і трансляторів. У математиці інколи значення *істинно* позначають одиницею, а значення *хибно* – нулем.

Логічні вирази складаються з операндів, логічних операцій та круглих дужок. У мові Pascal означено шість операцій **порівняння**: > (більше), < (менше), = (дорівнює), <> (не дорівнює), >= (більше або дорівнює), <= (менше або дорівнює). Наприклад, логічний вираз $m>n$ – це вираз, який стверджує, що значення змінної m більше за значення змінної n. Тому, якщо $m=2$, а $n=3$, вираз $m>n$ хибний, а вираз $m*10>n$ – істинний. Таким чином, результатом обчислення логічного виразу може бути тільки одне з двох значень: true (істинно, або <так>) або false (хибно, або <ні>).

У розглянутих виразах використовувалася лише одна операція порівняння. Такі вирази називають *простими*. З простих логічних виразів за допомогою логічних операцій можуть створюватися складені.

Логічні операції

A	B	not A	A and B	A or B	A xor B
Істинно	Істинно	Хибно	Істинно	Істинно	Хибно
Істинно	Хибно	Хибно	Хибно	Істинно	Істинно
Хибно	Істинно	Істинно	Хибно	Істинно	Істинно
Хибно	Хибно	Істинно	Хибно	Хибно	Хибно

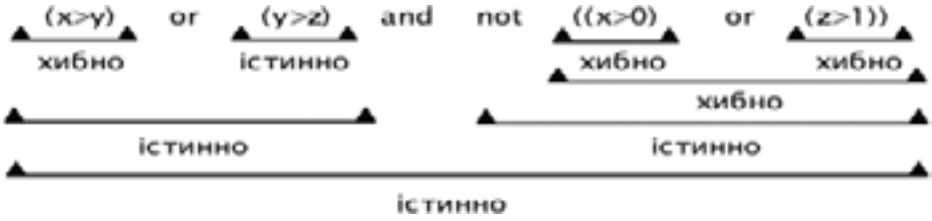
У всіх мовах програмування реалізовано чотири логічні операції: *not* (<не>), *and* (<і>), *or* (<або>) та *xor* (<виключаюче або>). Ці операції ще називають відповідно операціями заперечення, кон'юнкції, диз'юнкції та строгої диз'юнкції. Результати виконання цих операцій над логічними виразами A і B, які набувають значень "істинно" та "хибно", наведені в таблиці. Така таблиця називається *таблицею істинності*. Ця таблиця є строгим математичним означенням зазначених логічних операцій.

З цієї таблиці бачимо, що результат операції *and* є істинним лише за умови, що обидва висловлювання A і B є істинними одночасно. Результат операції *or* буде істинним, якщо істинним буде хоча б одне з висловлювань A та B. Результат операції *not* є істинним, якщо логічний вираз A є хибним, і, навпаки, хибним, якщо вираз A є істинним. Результат операції *xor* є істинним лише тоді, коли з двох висловлювань A та B буде істинним лише одне з них.

У складених логічних виразах установлено такий пріоритет операцій: *not, and, or, xor*. Першою з двох операцій одного пріоритету виконується та, яка у виразі розташована ліворуч. Операції, взяті в круглі дужки, виконуються першими. Наведемо приклад складеного логічного виразу:

(*y*>*x*) *or* (*y*>*z*) *and not* ((*x*>0) *or* (*z*>*x*)).

Нижче показано порядок виконання операцій у цьому виразі з припущенням, що змінні мають такі значення змінних: *x*=−1, *y*=−2, *z*=−3.



У виразах, що містять арифметичні та логічні операції, а також операції порівняння, порядок їх виконання такий: заперечення (not); множення, ділення і кон'юнкція (and); додавання, віднімання і диз'юнкція (or) та строга диз'юнкція (xor); операції відношення (=, <>, <, <=, >, >=).

✓ *Зауважмо, що вираз a>b or c>d є некоректним, оскільки згідно з правилами пріоритету першою має бути виконана операція b or c. Правильний запис цього виразу такий: (a>b) or (c>d).*

Приклад. Наведемо програму, яка дає змогу обчислювати значення представленого на рисунку логічного виразу, коли *x, y, z* мають різні значення.

```
program n12_01;
var x, y, z: integer;          {оголошення змінних цілочислового типу}
  p: boolean;                  {оголошення змінної логічного типу}
begin
  writeln ('увести x, y, z'); {повідомлення про введення}
  readln (x, y, z);            {уведення значень x, y, z}
  p:=(y>x) or (y>z) and not((x>0) or (z>x));     {обчислення виразу}
  writeln ('Результат=', p);{виведення результату}
  readln                       {призупинення виконання програми}
end.
```

Перетворення і спрощення логічних виразів здійснюється на основі тотожностей і законів булевої алгебри. Основні з них наведено далі.

$$x \vee 0 = x;\ \ x \vee 1 = 1;\ \ x \vee \overline{x} = 1;$$
$$x_1 \vee x_2 = x_2 \vee x_1;\ \ x \vee x \vee \ldots \vee x = x. \qquad (1)$$

$$x \cdot 0 = 0;\ \ x \cdot 1 = x;\ \ x \cdot \overline{x} = 0;$$
$$x_1 \cdot x_2 = x_2 \cdot x_1;\ \ \ x \cdot x \cdot \ldots \cdot x = x\ . \qquad (2)$$

$$x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3;$$
$$x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3. \quad (3)$$

$$x_1 \cdot (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3;$$
$$x_1 \vee x_2 \cdot x_3 = (x_1 \vee x_2) \cdot (x_1 \vee x_3). \quad (4)$$

$$\overline{x_1 \vee x_2} = \overline{x}_1 \cdot \overline{x}_2;$$
$$\overline{x_1 \cdot x_2} = \overline{x}_1 \vee \overline{x}_2. \quad \text{формули де Моргана} \quad (5)$$

$$x_1 x_2 \vee x_1 \overline{x}_2 = x_1; \quad (x_1 \vee x_2) \cdot (x_1 \vee \overline{x}_2) = x_1. \quad (6)$$

$$x_1 \vee x_1 x_2 = x_1; \quad x_1 (x_1 \vee x_2) = x_1. \quad (7)$$

$$\overline{\overline{x}} = x. \quad (8)$$

Усі наведені формули доводяться через визначення їх значень на основі прямого перебору всіх значень аргументів.

Перевіряємо себе

1. Які операції порівняння використовуються в логічних виразах?
2. Якого значення набуде логічний вираз *x*>=*y*, якщо *x*=3, *y*=5?
3. Які логічні операції застосовуються в логічних виразах?
4. Яким є порядок виконання операцій у виразі not *a*(*a* and *b*) or *c*?
5. Який пріоритет мають операції порівняння, логічні та арифметичні операції?

Виконуємо

1. Нехай A і B логічні вирази. Які з наведених нижче пар складених логічних виразів рівнозначні?

а) **not** (A **and** B)	A **or** B
б) **not** (A **and** B)	(**not** A) **or** (**not** B)
в) **not** (A **or** B)	A **and** B
г) **not** (A **or** B)	(**not** A) **and** (**not** B)
д) (A **and** B) **or** ((**not** A) **and** (**not** B))	A=B
е) A<>B	((**not** A) **and** B) **or** (A **and** (**not** B))
є) A=B=true	A **and** B
ж) A=B=true	A=B
з) A=false	false
и) (**not** (A **or** B)) **and** A	(A **or** B) **and** (**not** B) **and** (**not** A)
і) (A **or** B) **and** (**not** B)	A

2. Складіть програму для обчислення логічного виразу: ((not a) and b) or (a and(not b)).

12.3. Використання булевої алгебри для розв'язування логічних задач

Змістові логічні задачі можуть роз'язуватися різними способами, у тому числі засобами алгебри логіки. За цим способом вони розв'язуються в такій послідовності:

– детально вивчається умова задачі;

– логічні висловлювання позначаються логічними змінними;

– розробляється булевий вираз, у якому між усіма висловлюваннями встановлюються логічні зв'язки;

– визначається значення істинності булевого виразу;

– робиться висновок про результат розв'язання задачі.

Приклад. У фінальному забігу учнів школи на 100 м взяли участь Антонов, Васько, Соломко і Дахов. Три вболівальниці, які спостерігали за змаганнями, дали такі прогнози:

Ксеня: Першим буде Дахов, другим – Васько.

Марічка: Першим буде Соломко, четвертим – Антонов.

Світлана: Другим буде Дахов, третім – Васько.

Після змагань з'ясувалося, що тільки одне із тверджень у кожному прогнозі було правильним. Які місця посіли учасники змагань?

Позначимо кожне із тверджень учасників прогнозу такими змінними:

- Перший Дахов – D1;
- Перший Соломко – C1;
- Другий Васько – B2;
- Другий Дахов – D2;
- Третій Васько – B3;
- Четвертий Антонов – A4.

Оскільки одне з тверджень у кожному прогнозі правильне (істинне), можна записати:

D1 $\vee$ B2=1; C1 $\vee$ A4=1; D2 $\vee$ B3=1.

Очевидно, що (D1 $\vee$ B2)(C1 $\vee$ A4)(D2 $\vee$ B3)=1

Перетворимо цей вираз на такий: (D1C1 $\vee$ D1A4 $\vee$ B2C1 $\vee$ B2A4) (D2 $\vee$ b3)=1. Значення виразу D1C1 хибне, тому що за умовами змагань перше місце не можна присуджувати двом різним учасникам. Тому вираз можна записати так: (D1A4 $\vee$ B2C1 $\vee$ B2A4)(D2 $\vee$ B3)=1. Перетворимо останній вираз на такий:

D1A4D2 $\vee$ D1A4B3 $\vee$ **B2C1D2** $\vee$ **B2C1B3** $\vee$ **B2A4D2** $\vee$ **B2A4B3** = 1.

Усі виокремлені жирним шрифтом вирази хибні. Наприклад, вираз D1A4D2 хибні тому, що Дахов не може одночасно посісти перше й друге місце, а вираз B2C1D2 – тому, що Васько і Дахов не можуть разом бути на другому місці. Таким чином, істинним є вираз D1A4B3, тобто першим був Дахов, другим – Соломко, третім – Васько, четвертим – Антонов.

Перевіряємо себе

1. У якій послідовності розв'язуються логічні задачі засобами алгебри логіки?

2. На основі яких даних робиться висновок про результат розв'язання задачі?

Виконуємо

1. Сім'я, збираючись на прогулянку, ввечері обговорює почутий прогноз погоди на завтра.

Батько: Якщо не буде вітру, то буде хмарно і без дощу.

Мати: Якщо буде дощ, то буде хмарно і без вітру.

Син: Якщо буде хмарно, то буде дощ і не буде вітру.

То ж якою за прогнозом завтра буде погода?

2. Джону, Марго та Оліверу оголошено підозру в співучасті в пограбуванні банку. Викрадачі втекли на автомобілі, що чекав на них. На слідстві Джон дав свідчення, що злочинці зникли на синьому "Запорожці", Марго сказала, що це були чорні "Жигулі", а Олівер стверджував, що це була "Нива" і не синя. Щоб заплутати слідство, кожен із них вказав правильно або марку машини, або тільки її колір. Якого кольору і якої марки була машина?

3. Петрик, Василько і Марійка залишились удома самі. Хтось із них змінив пароль на комп'ютері. Коли мама запитала, хто це зробив, вони сказали:

Петрик: "Я не змінював. Марійка теж не змінювала".

Василько: "Марійка справді не змінювала. Це зробив Петрик".

Марійка: "Василько обманює. Це він змінив".

Після додаткових запитань з'ясувалося, що двоє дітей двічі сказали правду, а третя дитина – один раз обманула, а один раз сказала правду. Хто змінив пароль?

4.  Хтось із трьох – Ледачий, Байдужий чи Безвідповідальний забув вимкнути світло в офісі, внаслідок чого фірмі нарахували штраф.

У ході службового розслідування кожен із них зробив дві заяви.

Безвідповідальний: "Я цього не робив. Це зробив Байдужий". Ледачий: "Байдужий не винен. Це зробив Безвідповідальний". Байдужий: "Я цього не робив. Ледачий цього не робив". У результаті службового розслідування з'ясувалося, що один із них двічі збрехав, другий двічі сказав правду, а третій – один раз збрехав, а другий раз сказав правду. Хто повинен компенсувати фірмі збитки?

Практична робота № 22

Тема:	Побудова логічних виразів та їх обчислення
Мета:	Навчитися будувати логічні вирази на основі аналізу змісту висловлювань

Завдання

1. Скласти таблицю істинності для логічних виразів:

а) не А або В; б) А або не В і С

2. Побудувати логічний вираз для розв'язування задачі. Розв'язати задачу.

За підсумками першості школи з шахів кращими стали Нестор, Михайло, Люба та Руслан. Перед початком змагань палкі прихильники шахів висловили припущення щодо розподілу місць. Перший сказав, що переможе Нестор, а Михайло стане другим. Другий прихильник друге місце віддав Любі, а Русланові – четверте. Третій не погодився з двома попередніми. Він спрогнозував, що Руслан посяде третє місце, а Нестор буде четвертим. Після завершення змагань з'ясувалось, що кожен із прихильників був правий лише в одному зі своїх прогнозів. Як розподілились місця між чотирма учасниками?

ДЛЯ ДОПИТЛИВИХ

Корені математичної логіки сягають глибокої давнини. Першим до створення її наукових основ наблизився німецький вчений Г. В. Лейбніц (1646–1716), який указав шлях для перетворення логіки "зі словесного царства, повного невизначеностей, на царство математики, де відношення між об'єктами або висловлюваннями визначаються абсолютно точно".

Значного успіху в перетворенні логіки на математичну науку досяг у 1847 році англійський математик Джордж Буль, який розробив алгебру логіки, пізніше названу на його честь булевою алгеброю. Дж. Буль розробив систему позначень і правил, яку можна застосовувати до всіх об'єктів, у тому числі букв, цифр і речень. За допомогою цієї системи можна кодувати висловлювання символами мови, до яких можна застосовувати введені ним операції: кон'юнкції (І – AND), диз'юнкції (АБО – OR), заперечення (НІ – NOT).

Вагомий внесок у теорію застосування булевої алгебри для синтезу цифрових схем зробив відомий український учений В. М. Глушков.

СЛОВНИЧОК

Логіка – наука про загальні закони розвитку об'єктивного світу й пізнання їх.

Поняття – форма висловлювання, яким виокремлюються і означуються найвагоміші ознаки предмета або класу предметів.

Предикати – твердження, що містять змінні.

Твердження – форма мислення, яка дає змогу на основі одного або кількох тверджень-посилань отримати нове судження (значення або висновок).

РОЗДІЛ 13. АЛГОРИТМИ З РОЗГАЛУЖЕННЯМИ

Мітки та оператор безумовного переходу, оператори умовного переходу.

13.1. Мітки. Оператор безумовного переходу

Оператори програми виконуються послідовно в тому порядку, в якому вони записані в тексті програми. Але цей порядок може бути змінений різними методами, зокрема, за допомогою оператора *безумовного переходу*. У цьому операторі використовується мітка – ідентифікатор, який найчастіше записується перед іншим оператором і відокремлюється від нього символом двокрапки, наприклад: M: x:=x+1; {M – мітка}.

Оператор безумовного переходу має такий вигляд: goto M;.

У результаті виконання цього оператора керування передається оператору, позначеному міткою M.

Будь-яка мітка, що використовується в програмі, має бути описана в розділі описань міток. Цей розділ розпочинається зі службового слова label (мітка) і розміщується перед розділами описань констант і змінних.

Приклад. У наведеній нижче програмі оголошується мітка MA. Після виконання операторів x:=15, y:=4 і z:=2 керування передається оператору, позначеному міткою MA, тобто y:=y*x*z. Таким чином, у цій програмі оператор y:=y+x+z не виконається, а отже, буде виведено повідомлення y=120.

```
program n13_1;
label MA;                    {MA – мітка}
var x, y, z: integer;        {змінні цілочислового типу}
begin
  x:=15; y:=4; z:=2;         {оператори присвоювання}
  goto MA;                   {перехід на мітку MA}
  y:=y+x+z;                  {оператор не виконується}
MA : y:=y*x*z;               {на початку рядка розміщена мітка MA}
  writeln ('y=',y);          {виведення результату}
  readln                     {призупинення виконання програми}
end.
```

Використання команд безумовного переходу може зробити програму заплутаною й ускладнити її сприймання і налагодження, тому тепер намагаються їх не використовувати, надаючи перевагу структурному програмуванню.

Перевіряємо себе

1. З якою метою у програмах використовуються мітки?
2. Як записується оператор безумовного переходу в мові Pascal?
3. У якому місці оператора вказується мітка?
4. Чому намагаються уникати використання команди безумовного переходу?

13.2. Оператори умовного переходу

Оператор if...then...else

Раніше вже розглядалась алгоритмічна конструкція розгалуження, яка дає змогу виконавцеві алгоритму обрати один із двох шляхів подальших дій залежно від істинності певної умови. У багатьох мовах програмування високого рівня, зокрема в мові Pascal, для реалізації алгоритмів із розгалуженою структурою використовуються такі оператори умовного переходу:

if <умова> **then** <оператор>;

if <умова> **then** <оператор 1> **else** <оператор 2>;

Тут **if** (якщо), **then** (то), **else** (інакше) – зарезервовані слова, <умова> – довільний логічний вираз, <оператор 1> і <оператор 2> – довільні оператори.

Перший оператор реалізує одноальтернативне розгалуження, другий – двохальтернативне. Зазначимо, що перший тип оператора умовного переходу називають його *скороченою формою*. Блок-схеми цих операторів були розглянуті в попередніх розділах.

Виконання скороченої форми оператора умовного переходу починається з обчислення значення булевого виразу <умова>. Якщо умова істинна, то виконується <оператор>, якщо ж хибна, то виконання умовного оператора на цьому завершується.

Наприклад, розглянемо такий фрагмент програми тестування учня, в якій змінна vidpovid призначена для введення учнем відповіді на чергове запитання, змінна prav містить правильну відповідь, а змінна ocinka містить накопичену оцінку до відповіді на запитання:

if vidpovid = prav **then** ocinka := ocinka + 1;

При виконанні вказаної команди спочатку буде перевірена умова vidpovid = prav, яка буде істинною, якщо учень дасть правильну відповідь, і в цьому випадку виконається оператор ocinka := ocinka + 1, тобто оцінка учня зросте на 1 бал. Якщо ж учень дасть неправильну відповідь, то умова vidpovid = prav буде хибною, команда розгалуження завершить свою роботу і значення змінної оцінки не зміниться.

Оператор умовного переходу, записаний у повній формі, виконується так. Спочатку обчислюється значення булевого виразу <умова>. Якщо умова істинна, то виконується <оператор 1> і керування передається наступному за умовним оператору (<оператор 2> пропускається). Якщо ж умова хибна, то <оператор 1> пропускається, а виконується лише <оператор 2>, і на цьому дія умовного оператора завершується.

Звернімо увагу на таку синтаксичну деталь. Символ " ;" відокремлює оператори. Оскільки умовний оператор завершується діями, які записані після слова ***else****, то запис крапки з комою перед* ***else*** *є синтаксичною помилкою.*

Після ключових слів then та else можуть бути розташовані інші оператори умовного переходу. Така конструкція використовується, якщо існує не менше трьох можливих варіантів дій.

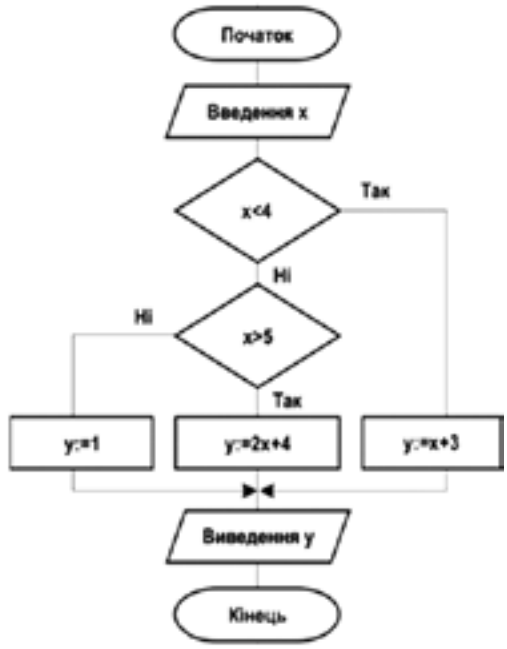


Рис. 13.1. Блок-схема алгоритму обчислення значення функції

Розглянемо програму, в якій один оператор умовного переходу вкладено в іншій.

Приклад. Необхідно обчислити значення такої функції:

$$y = \begin{cases} x + 3, & \text{якщо } x < 4; \\ 1, & \text{якщо } 4 \le x \le 5; \\ 2x + 4, & \text{якщо } x > 5. \end{cases}$$

Блок-схема алгоритму обчислення значення функції представлена на рис. 13.1.

За цією блок-схемою складемо програму. Гілка “ні” першого розгалуження, якій відповідає фраза else оператора if, містить друге розгалуження. Тому після слова else першого оператора if слід записати другий оператор умовного переходу.

```
program n13_2;
var x, y: real;              {оголошення змінних дійсного типу}
begin
  writeln ('увести x');      {повідомлення про введення значення змінної}
  readln (x);                {уведення значення змінної x}
  if x<4 then y:=x+3 else    {перевірка умови і обчислення значення
                              виразу}
  if x>5 then y:=2*x+4 else y:=1;
  writeln ('y=', y);         {виведення обчисленого значення}
  readln                     {призупинення виконання програми}
end.
```

Зазначимо, що в логічних виразах, які використовуються в операторах умовного переходу, інколи доцільно застосовувати логічні операції (not, or, and). Використання цих операцій дає змогу скоротити кількість операторів умовного переходу.

Приклад. Наведемо програму, що обчислює значення функції

$$y = \begin{cases} k + x, & \text{якщо } x < 0,5 \text{ і } k \geq 1; \\ 2k - x, & \text{інакше} \end{cases}$$

```
program n13_3;
var x, k, y: real;           {оголошення змінних дійсного типу}
begin
  writeln ('уведіть x, k');  {повідомлення про введення значень змінних}
  readln (x, k);             {уведення значень змінних x, k}
  {перевірка умови обчислення значення функції}
  if (x<0.5) and (k>=1) then y:=k+x else y:=2*k-x;
  writeln ('y=',y);          {виведення обчисленого значення}
  readln                     {призупинення виконання програми}
end.
```

*У мові Pascal дія ключових слів **then** та **else** поширюється лише на один, наступний, оператор.* Якщо в разі істинності або хибності певного логічного твердження потрібно виконати декілька операторів, то їх слід оточити *операторними дужками,* роль яких відіграють ключові слова begin та end:

```
if <умова> then
begin
  <оператор 1>; ...;<оператор N>
end
```

```
else
begin
  <оператор K>; ...;<оператор M>
end;
```

Якщо логічний вираз <умова> істинний, виконуватимуться оператори <оператор 1>, ..., <оператор N>, якщо хибний – оператори <оператор K>, ..., <оператор M>. Групу операторів, що розпочинається зі слова begin, а завершується словом end, можна розглядати як один *складений оператор*. Розглянемо приклад використання складеного оператора.

Приклад. Наведемо програму, яка обчислює значення функцій

$p = \sqrt{x} - y$, якщо $x > 1$ і $y < 4$;

$q = |x + y|$, якщо $x > 1$ і $y < 4$;

$t = x/y$ інакше

```
program n13_4;
var x, y, p, q, t: real;        {оголошення змінних дійсного типу}
begin
  writeln ('увести x, y');      {повідомлення про введення значень змінних}
  readln (x, y);                {уведення значень змінних x, y}
  if (x>1) and (y<4) then       {перевірка умови}
  begin                         {початок операторних дужок}
    p:=sqrt(x)-y;               {обчислення значення функції p}
    q:=abs(x+y);                {обчислення значення функції q}
    writeln (p, ' ', q)         {виведення значень функцій p та q}
  end                           {кінець операторних дужок}
  else
  begin                         {початок операторних дужок}
    t:=x/y;                     {обчислення значення функції t}
    writeln (t)                 {виведення значення функції t}
  end;                          {кінець операторних дужок}
readln                          {призупинення виконання програми}
end.
```

Оператор Case

Розглянуті вище оператори умовного переходу здійснюють розгалуження за двома гілками. В мовах програмування високого рівня часто використовують також оператор, який здійснює розгалуження за багатьма гілками. Зокрема, у мові Pascal таким є оператор багатоваріантного вибору, якому відповідає блок-схема, зображена на рис. 13.2.

Оператор багатоваріантного вибору має таку структуру:

```
case <вираз> of
<константа 1>: <оператор 1>;
...
<константа N>: <оператор N>;
```

[else <оператор K>]
end;

Тут <константа 1>,..., <константа N> – це константи або діапазони значень того ж типу, що й <вираз>.

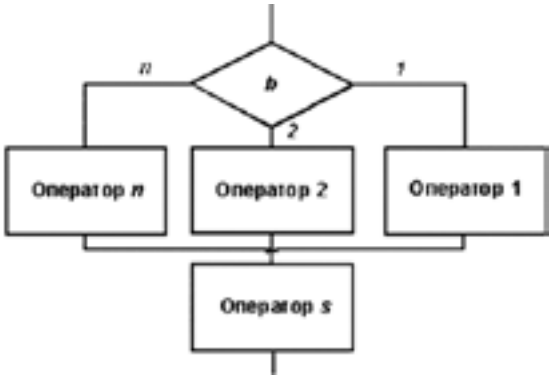


Рис. 13.2. Блок-схема оператора багатоваріантного вибору

Виконання оператора багатоваріантного вибору розпочинається з обчислення виразу, записаного після слова case. Якщо обчислене значення дорівнює одній із констант набору <константа 1>,...,<константа N>, то виконується оператор, який відповідає цій константі. Такий оператор може бути складеним. Ключове слово **else** використовувати не обов'язково. Оператор, який розташований після слова else, виконується тоді, коли значення виразу, записаного після слова **case**, не збігається з жодною з перелічених констант. Якщо ключове слово **else** відсутнє, а значення згаданого виразу не збігається з жодною константою, оператор **case** ніяких дій не виконує. У будь-якому випадку після виконання оператора **case** керування передається оператору, що розташований за словом **end**, яким завершується даний оператор.

Приклад. Учень уводить свою семестрову оцінку з інформатики. Вивести на екран повідомлення, в якому відображається рівень знань учня з цього предмета.

```
program n13_5;
var ocinka: integer;              {змінна типу integer}
begin
  writeln ('увести оцінку- '); {повідомлення про введення}
  readln (ocinka);              {уведення оцінки}
  case ocinka of                {початок оператора case}
1..3: writeln ('початковий рівень знань '); {якщо введено 1, 2 або 3}
4..6: writeln ('середній рівень знань');    {якщо введено 4, 5 або 6}
7..9: writeln ('достатній рівень знань');   {якщо введено 7, 8 або 9}
10..12: writeln ('високий рівень знань'); {якщо введено 10, 11 або 12}
```

```
  else
    writeln ('помилка уведення')        {якщо введено число менше
                                         1 або більше 12}
  end;                                   {кінець оператора case}
end.
```

Перевіряємо себе

1. Яку структуру має оператор умовного переходу?
2. Як виконуються оператори умовного переходу?
3. Яку структуру має оператор Case?
4. У яких із наведених нижче операторів умовного переходу є помилки?
   а) **if** x<>a **then** x:=x+1;
   б) **if** x=0 and y **then** x:=x+1 **else** x:=x-1;
   в) **if** x<y **then** x:=x*x **else** x:=x+1;
   г) **if** 23 **then** x:=x+2;

Виконуємо

1. Визначте, яких значень набудуть змінні a і b після виконання наведеного фрагмента програми за умови, що введено значення a=0,6 і b= −2,3.

```
var a, b: real;
begin
  read (a, b);
  if a<b then a:=b else b:=a;
end.
```

2. Складіть програму обчислення значення функції

$$y = \begin{cases} \sqrt{x} + a, & \text{якщо } x > 0; \\ \sqrt{1 - x} - a, & \text{якщо } x \le 0. \end{cases}$$

3. Цукор розфасували в два пакети. Маса першого – m кг, другого – n кг. Складіть програму, що визначає який пакет важчий – перший чи другий.

4. Тіло має масу M г і об'єм V см^3. Чи плаватиме воно в рідині, об'єм якої V_1 см^3, а маса M_1 г?

Підказка. Знайдіть густину тіла та рідини, а потім порівняєте їх.

13.3. Програми для алгоритмів із розгалуженням

І в науці, і в житті трапляється безліч задач, у яких застосовується розгалуження. Розглянемо найбільш типові з них.

1. Дано два числа. Розробити програму, що виводить на екран більше число.

Позначимо перше число змінною *a*, друге – змінною *b*. Першим із клавіатури вводиться значення змінної *a*. Програмна реалізації цього завдання може мати такий вигляд:

```
program n13_6;
var a, b: real;                    {оголошення змінний типу real}
begin
  writeln ('увести a, b');         {повідомлення про уведення}
  readln (a, b);                   {уведення значень змінних}
  if a>b then writeln ('a=', a) else writeln ('b=', b); {перевірка умови
                                   і виведення значення}
  readln                           {призупинення виконання програми}
end.
```

2. Розробити програму для визначення, чи можна в квадрат зі стороною *a* вписати коло радіусом *r*.

Значення сторони квадрата і радіуса кола вводяться з клавіатури. Програма з іменем n13_7, що виконує завдання визначення відповіді, наведена нижче.

```
program n13_7;
var a, r: integer;                 {змінні цілочислового типу}
y: string;                         {змінна рядкового типу}
begin
  writeln ('увести значення a, r'); {повідомлення про введення}
  readln (a, r);                   {уведення значень змінних}
  {перевірка умови і присвоювання значення змінної}
  if a=2*r then y:='можна вписати' else y:='не можна вписати ';
  writeln (y);                     {виведення результату}
  readln                           {призупинення виконання програми}
end.
```

3. Двоє учнів змагаються у влучності кидання м'яча. Розробити програму, яка виводить на екран результати лише тих кидків, у результаті яких м'яч падає на землю на віддалі 15–20 м від точки кидання.

Позначимо змінною *a* результат кидка першого учня, змінною *b* – другого учня. Уважатимемо, що у випадку, якщо м'яч потрапляє на лінію відмітки 15 або 20 м, то вважається, що учень потрапив у ціль. Один із можливих варіантів програми наведено нижче.

```
program n13_8;
var a, b: real;                    {оголошення змінних типу real}
begin
writeln ('увести a, b');           {повідомлення про введення}
readln (a, b);                     {уведення значень змінних}
{перевірка умови і виведення значень змінних}
if (f<=20) and (a>=15) then writeln ('кидок першого=', a);
if (b<=20) and (b>=15) then writeln ('=кидок другого=', b);
```

```
readln                          {призупинення виконання програми}
end.
```

Виконайте і доведіть, що програма функціонує правильно.

4. Батько і мати поклали в банк різні суми грошей. Якщо гроші зберігаються у банку 4 роки, то сума внеску збільшується вдвічі, інакше сума залишається незмінною. Розробити програму, що визначає суму внеску батька і матері.

Позначимо суму внеску батька змінною *a*, суму внеску матері змінною *b*. Початкові значення внесків уводяться з клавіатури. Нижче наведена програма визначення реального грошового стану внесків матері й батька.

```
program n13_9;
var a, b: real;                 {оголошення змінних дійсного типу}
begin
  writeln ('увести a, b');      {повідомлення про введення}
  readln (a, b);                {уведення значень змінних}
  {перевірка умови і надання значень змінним}
  if a>4 then a:=a+a;
  if b>4 then b:=b+b;
  writeln ('a=', a, ',b =', b); {виведення результату}
  readln                        {призупинення виконання програми}
end.
```

Виконайте програму і доведіть, що вона функціонує правильно.

5. Василь і Микола збирали гриби. Василь запропонував: "Якщо я назбираю грибів більше ніж ти, то віддаси мені половину своїх грибів, у іншому випадку я віддаю тобі половину своїх". Микола погодився з такою пропозицією. Необхідно розробити програму, за допомогою якої визначається, скільки грибів дістанеться кожному.

Позначимо змінною *a* масу грибів, зібраних Василем, а змінною *b* – масу грибів, зібраних Миколою. Ці самі змінні використовуються й для визначення маси грибів, що дістанеться кожному учню.

```
program n13_10;
var a, b: real;             {оголошення змінних дійсного типу}
begin
  writeln ('увести a, b'); {повідомлення про введення значень змінних}
  readln (a, b);            {уведення значень змінних}
  {перевірка умови і обчислення значення змінних}
  if a>b then begin a:=a+b/2; b:=b/2 end
  else begin b:=b+a/2; a:=a/2 end;
  writeln ('Василь ', a, ' (кг)',' ,Микола ',b,' (кг)'); {виведення результату}
  readln                    {призупинення виконання програми}
end.
```

Виконайте програму й доведіть, що вона функціонує правильно.

6. Єва, Аліна та Олеся кидають кубик, на гранях якого є цифри від 1 до 6. Кожна з них називає цифру від 1 до 6, а потім кидає кубик. Переможе та дівчинка, яка вгадає цифру, що випала.

Нехай змінна *p* – це реальне значення цифри, змінна *a* – значення цифри Єви, *b* – Аліни й *c* – Олесі. Необхідно послідовно порівнювати значення змінної *p* зі змінними *a*, *b* і *c*. Переможців може бути кілька. Ознакою того, що переможців нема, є нуль у змінній *к*. Програма, що реалізує завдання, може бути такою.

```
program n13_11;
var a, b, c, p, x, k: integer:                    {оголошення змінних}
begin
  writeln ('увести цифру Єви'); readln (a);       {уведення значення a}
  writeln ('увести цифру Аліни'); readln (b);     {уведення значення b}
  writeln ('увести цифру Олесі'); readln (c);     {уведення значення c}
  writeln ('увести цифру, що випала'); readln (p); {уведення значення p}
  x:='переможець '; k:=0;       {ознака наявності переможця}
  if a=p then begin writeln (x, ' Єва'); k:=k+1; end;
  if b=p then begin writeln (x,' Аліна'); k:=k+1 end;
  if c=p then begin writeln (x, ' Олеся'); k:=k+1 end;
  if k=0 then writeln ('переможців немає');
  readln                                          {призупинення виконання програми}
end.
```

Уведіть, виконайте програму і доведіть, що вона функціонує правильно.

Перевіряємо себе

1. Дано два цілих числа. Розробіть програму заміни додатніх чисел одиницею, а від'ємних – нулем.

2. Дано три дійсних числа *a*, *b*, *c*. Розробіть програму, що збільшує удвічі кожне з них, якщо $a>b>c$, у протилежному випадку – кожне число зменшується на одиницю.

3.  Розробіть програму для визначення, чи є трикутник зі сторонами *a*, *b*, *c* прямокутним трикутником.

4.  Дано три сторони трикутника. Розробіть програму для визначення, чи є цей трикутник рівнобедреним.

5.  Розробіть програму, що визначає, чи можна в рівносторонній трикутник зі стороною *a* вписати коло з радіусом *r*.

6.  Розробіть програму, що визначає, чи є в прямокутному трикутнику з катетами *a*, *b* і гіпотенузою *c* кут 30 градусів.

Виконуємо

1. Дано три дійсні числа a, b, c. Знайдіть найбільше з них.

2. Сторони одного прямокутника дорівнюють A і B. Сторони другого дорівнюють X і Y. Складіть алгоритм і програму перевірки прямокутників на рівність їх площ.

Підказка. Знайти різницю площ і порівняти її з наперед заданим значенням похибки.

3. Дано сторону одного квадрата і площу іншого. Чи рівні ці квадрати?

4. Дано сторону квадрата і дві суміжні сторони прямокутника. Чи є ці фігури рівновеликими?

5. Дано чотири натуральні числа. Чи можуть ці числа бути членами арифметичної прогресії?

6. Дано довжини відрізків AB, BC і AC. Чи можуть ці відрізки бути сторонами трикутника?

7. Дано координати трьох точок на числовій осі. Яка з цих точок найближча до початку відліку?

8. Дано градусні міри двох кутів. Чи можуть ці кути бути суміжними?

9. У стіні є квадратний отвір зі стороною a. Чи пройде в цей отвір труба радіусом R?

Практична робота № 23

Тема:	Розроблення алгоритмів з послідовними розгалуженнями та їх реалізація у вигляді програм
Мета:	Навчитися розробляти й описувати у вигляді програм алгоритми з послідовними розгалуженнями

Завдання

У табл. 13.1 і 13.2 вміщено варіанти завдань для виконання практичної роботи. У табл. 13.1 подано завдання з використанням стандартних математичних функцій. Складніші завдання з використанням логічних операцій наведено в табл. 13.2.

Для отриманого завдання треба розробити програму розв'язування задачі, розрахувати контрольний варіант і вивчити порядок роботи в середовищі програмування.

Таблиця 13.1

№ варіанта	Арифметичний вираз	Початкові дані	
		варіант 1	варіант 2
1	$y=\begin{cases} ax^2, \text{ якщо } x\le 6 \\ \sqrt{x}, \text{ якщо } x>0 \end{cases}$	$x=-2;\ a=3$	$x=3;\ a=0.4$
2	$y=\begin{cases} 1+x^3, \text{ якщо } x>4 \ i \ x\le 6 \\ 3.2-x^2, \text{ інакше} \end{cases}$	$x=5$	$x=7$
3	$y=\begin{cases} \lvert x\rvert, \text{ якщо } x<1 \\ 1+\sqrt{x}, \text{ якщо } x>4 \\ 3.5+2x, \text{ якщо } 0<x\le 4 \end{cases}$	$x=0.2$	$x=5.1$
4	$y=\begin{cases} (x+2)^2, \text{ якщо } x<-2 \\ x+2, \quad \text{якщо } -2\le x\le 0 \\ (x+2)^3, \text{ якщо } x\ge 0 \end{cases}$	$x=3$	$x=1.5$

Таблиця 13.2

№ варіанта	Завдання	Початкові дані	
		варіант 1	варіант 2
1	Відомі координати трьох точок на площині. Чи розміщені ці точки на одній прямій?	$x_1=2,\ y_1=3$ $x_2=4,\ y_2=6$ $x_3=8,\ y_3=12$	$x_1=1,\ x_2=2$ $x_2=2,\ y_2=5$ $x_3=4,\ y_3=8$
2	Відомі довжини трьох відрізків a, b, c. Чи можна з них побудувати рівносторонній трикутник?	$a=3,\ b=3,\ c=3$	$a=3,\ b=4,\ c=4$
3	Відомі довжини трьох відрізків a, b, c. Чи можна побудувати з них прямокутний трикутник?	$a=3,\ b=4,\ c=5$	$a=3,\ b=5,\ c=6$
4	Дано три дійсних числа a, b, c. Чи розміщені вони в порядку зростання?	$a=2,\ b=5,\ c=6$	$a=4,\ b=1,\ c=7$

Нижче подано зразок оформлення програми для першого варіанта табл. 13.1.

```
program n13_12;                    {програма з розгалуженням}
var a, x, y: real;                 {оголошення змінних}
```

```
begin                                   {початок виконавчої частини}
  writeln ('уведіть a, x');             {повідомлення про введення даних}
  readln (a, x);                        {уведення даних}
  if x<=0 then y:=a*x*x else y:=sqrt(x);   {перевірка умови}
  writeln (y)                           {виведення результату}
end.                                    {кінець програми}
```

1. Завантажити систему Turbo Pascal. Якщо не відкритий порожній файл Noname00.pas, відкрити його, виконавши команди F10→File→ →Enter→New→Enter.

2. Увести програму. Умисно зробити помилку в слові var.

3. Порівняти введену програму з розробленою. Виправити всі помилки, крім помилки у слові var.

4. Зберегти програму в робочому каталозі (F10→File→Enter→Save as <ім'я файла>→Enter).

5. Виконати програму.

6. Буде видано повідомлення про помилку: Error 36: BEGIN expexted. Виправити помилку й виконати програму.

7. За наявності в програмі інших синтаксичних помилок на екран видаватимуться певні повідомлення. Виправити помилку й виконати програму.

8. На екрані з'явиться повідомлення. Увести дані першого варіанта, наприклад, 3 4 2 – Enter.

9. Викликати результат на екран (Alt+F5).

10. Порівняти отриманий результат із контрольним. Якщо результат правильний, увести дані другого варіанта й виконати програму.

11. Якщо результат неправильний, проаналізувати програму, знайти помилку, виправити її і виконати програму. Ці дії виконувати доти, доки не буде отримано правильний результат.

12. Зберегти працездатну програму (F10→File→Enter→Save→Enter).

13. Для використання засобів налагодження програми необхідно включити опції компілятора Debug information і Local symbols в меню Option\Compiler. Для цього виконати команди: F10→Options→ Enter→Compiler→Enter. Якщо опції не включені, включити їх.

14. Виконати програму в покроковому режимі, послідовно натискаючи клавішу F7. Переконатися, що вона функціонує правильно.

15. Вийти зі середовища Turbo Pascal (F10→File→Enter→Exit→Enter).

Практична робота № 24

Тема:	Розробка алгоритмів із вкладеними розгалуженнями та їх реалізація у вигляді програм
Мета:	Навчитися розробляти алгоритми з вкладеними розгалуженнями

Завдання

1. Для оплати за користування електроенергією, що відпускається споживачам, які проживають у міській місцевості і мають однозонний лічильник, застосовуються такі тарифи (в копійках за 1 кВт·год)

За обсяг, спожитий до 100 кВт·год електроенергії на місяць (включно)	36,02
За обсяг, спожитий понад 100 кВт·год до 800 кВт·год електроенергії на місяць (включно)	75,48
За обсяг, спожитий понад 800 кВт·год електроенергії на місяць	145,76

Створити програму, за допомогою якої нараховується сума для сплати за користування електроенергією протягом місяця, якщо покази лічильника на початку і наприкінці місяця становлять Lstart і Lfinish відповідно.

2. Для оплати за користування електроенергією, що відпускається міським споживачам, які проживають у житлових будинках, обладнаних у встановленому порядку кухонними електроплитами і мають однозонний лічильник, застосовуються такі тарифи (в копійках за 1 кВт·год)

За обсяг, спожитий до 250 кВт·год електроенергії на місяць (включно)	21,54
За обсяг, спожитий понад 250 кВт·год до 800 кВт·год електроенергії на місяць (включно)	28,02
За обсяг, спожитий понад 800 кВт·год електроенергії на місяць	95,76

Створити програму, за допомогою якої нараховується сума для сплати за користування електроенергією протягом місяця, якщо покази лічильника на початку і наприкінці місяця становлять Lstart і Lfinish відповідно.

3. Створити програму, яка запитує про наявність електроплити в будинку і розраховує суму для сплати згідно з даними попередніх задач.

СЛОВНИЧОК

Ідентифікатор – мітка, яка записується перед оператором і відокремлюється від нього символом двокрапки.

Оператор Case – оператор, що здійснює розгалуження обчислювального процесу за багатьма гілками.

Оператор безумовного переходу – оператор, що змінює послідовний порядок виконання команд програми без будь-якої умови.

Оператор умовного переходу – алгоритмічна структура розгалуження, яка дає змогу виконавцеві алгоритму вибрати один із двох шляхів подальших дій.

РОЗДІЛ 14. АЛГОРИТМИ З ПОВТОРЕННЯМИ

Оператори циклу; особливості розв'язування обчислювальних задач; циклічні рекурентні алгоритми; циклічні обчислювальні алгоритми з невідомою кількістю ітерацій; обчислювальні алгоритми із вкладеними циклами.

14.1. Оператори циклу

Оператори циклу призначені для реалізації циклічних алгоритмічних структур. Можливі дві принципово різні ситуації: кількість повторень відома наперед; кількість повторень заздалегідь визначити не можна.

Зрозуміло, що управління циклом у цих випадках здійснюється по-різному. У першому випадку управління здійснюється за допомогою параметра циклу – змінної, яка послідовно змінює значення, а в другому випадку використовується умова – вираз логічного типу, від значення якого залежить виконання чи завершення циклу.

У мові Pascal реалізовано три різновиди операторів циклу: цикл із лічильником, цикл з передумовою, цикл з післямовою.

Будь-який оператор циклу складається з двох частин: **заголовка циклу** та **тіла циклу**. В заголовку циклу записуються умови, за яких виконання циклу триватиме або завершиться, а в тілі циклу містяться оператори, виконання яких потрібно повторювати.

Оператор циклу з лічильником

Оператор циклу з параметром (або, як його ще називають, циклу з лічильником) має такий синтаксис:

for <змінна> := <початкове значення> **to** <кінцеве значення> **do** <оператор>

або

for <змінна> := <початкове значення> **downto** <кінцеве значення> **do** <оператор>

Тут

for, to, downto, do – службові слова;

<змінна> – ідентифікатор деякої змінної порядкового типу даних, яка називається лічильником;

<початкове значення> – вираз, тип якого збігається з типом лічильника і його значення стає початковим значення лічильника;

<кінцеве значення> – вираз, тип якого збігається з типом лічильника і задає його кінцеве значення;

<оператор> – оператор тіла циклу.

Робота цього оператора здійснюється за таким алгоритмом:

1. Обчислюється початкове значення.
2. Це значення присвоюється параметру.
3. Обчислюється кінцеве значення.
4. Порівнюється значення параметра з кінцевим.
5. Якщо значення параметра перевищує (у випадку з використанням слова **to**) кінцеве значення або менше (у випадку з використанням слова **downto**) ніж кінцеве значення, то оператор циклу завершує роботу. В іншому випадку виконується тіло циклу, значення параметра змінюється на наступне (попереднє) і відбувається перехід до пункту 4.

Особливо слід підкреслити, що:

– і початкове, і кінцеве значення обчислюються до виконання оператора циклу;

– ідентифікатор змінної є параметром циклу й згідно зі стандартом не має змінюватися всередині оператора циклу. Нехтування цим правилом може призводити до непередбачуваних наслідків;

– після завершення циклу значення параметра циклу вважається не визначеним.

Приклад. Вивести таблицю квадратів натуральних чисел другого десятка.

```
for n := 11 to 20 do writeln(n, n * n: 5);
```

Приклад. Вивести великі латинські літери на екран у зворотному порядку.

```
for ch := 'Z' downto 'A' do write(ch: 2);
```

Якщо тіло циклу містить більше одного оператора, то ці оператори записуються в операторних дужках begin...end;.

Приклад. Використання складеного оператора в тілі циклу:

```
for i:=1 to n do
begin
  s:=s+a;
  y:=2*x
end;
```

Оператор while...do

Цикл із передумовою реалізується оператором while...do, синтаксис якого наведено нижче. У цьому операторі умова продовження циклу перевіряється перед початком кожного виконання операторів тіла циклу.

```
while <логічний вираз> do  {заголовок циклу}
<оператор>;                {тіло циклу}
```

Оператор тіла циклу виконуватиметься доти, доки істинним буде логічний вираз, який вказаний у заголовку циклу. Якщо в тілі циклу міститься декілька операторів, то вони беруться в операторні дужки begin...end;. Блок-схема оператора while...do зображена на рис. 14.1.

Приклад. Із клавіатури послідовно вводяться і додаються цілі числа, які не перевищують 10. Процес потрібно завершити при введенні числа, що більше 10.

Програма, що розв'язує цю задачу, наведена нижче.

```
program n14_1;
var x, s: integer;              {оголошення змінних типу integer}
begin
  s:=0; x:=0;                   {початкові значення змінних}
  while x<10 do                 {початок циклу}
  begin
    s:=s+x;                     {обчислення суми введених чисел}
    writeln ('увести x');       {повідомлення про введення}
    readln (x)                  {уведення числа}
 end;
writeln ('s=', s);              {виведення значення суми}
readln                          {призупинення виконання програми}
end.
```

Оператор repeat...until

Цикл із післяумовою реалізується оператором repeat...until, в якому умова завершення циклу перевіряється після кожного виконання операторів тіла циклу. Синтаксис цього оператора такий:

```
repeat
<оператори>                 {тіло циклу}
until <логічний вираз>      {заголовок циклу}
```

Оператори тіла циклу repeat...until виконуються доти, доки записаний після слова until логічний вираз є хибним.

Після останнього оператора в тілі циклу repeat...until символ ";" не записується. Оператори тіла циклу з післяумовою можна не брати в операторні дужки, навіть у тому випадку, якщо їх декілька. Блок-схему оператора repeat...until зображено на рис. 14.2.

Рис. 14.1. Блок-схема оператора циклу з передумовою

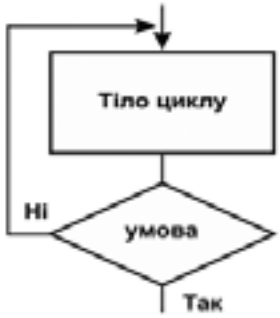


Рис. 14.2. Блок-схема оператора циклу з післяумовою

Приклад. Наведемо програму пошуку найбільшого спільного дільника цілих чисел, які вводяться з клавіатури. У програмі реалізовано **алгоритм Евкліда.**

```
program n14_2;
var a, b: integer;                {оголошення змінних цілочислового типу}
begin
  writeln ('увести значення a, b'); {повідомлення про введення}
  readln (a, b);                  {уведення значень a, b}
  repeat                          {початок циклу}
    if a>b then a:=a-b;           {оператор тіла циклу}
    if b>a then b:=b-a            {оператор тіла циклу}
  until a=b;                      {перевірка умови завершення циклу}
  writeln ('НСД=', a);            {виведення результату}
  readln                          {призупинення виконання програми}
end.
```

У деяких програмах усередині операторів циклу доцільно використовувати оператори break і continue. Перший із них перериває виконання циклу, а другий перериває виконання операторів тіла циклу і передає керування його заголовку.

Приклад. Розглянемо таку гру. Два гравці домовилися кидати гральні кості по черзі не більше 1000 разів. Виграє той, хто набере більшу суму балів. Однак гравці можуть кидати кості лише доти, доки суддя не підніме червоний прапорець. Нижче наведено фрагмент програми, яка моделює цю гру:

```
for i:=1 to 1000 do   {заголовок циклу}
begin
  <оператори тіла циклу>;
  if prapor='червоний' then
  break          {якщо підняти червоний прапорець, цикл завершити}
end;             {кінець циклу}
```

Перевіряємо себе

1. Для чого використовуються оператори циклу?
2. Яку структуру має оператор циклу repeat...until?
3. Які ключові слова використовуються в операторі циклу з параметром?
4. Як виконується оператор циклу while...do?

Виконуємо

1. Визначте, що буде виведено в результаті виконання фрагмента програми:

```
var i, y, b: integer;
```

```
begin
  y:=1; b:=4;
  for i:=2 to b do y:=y+1;
  y=y*y;
  writeln (y);
end.
```

2.  Розробіть програми для розв'язування таких задач:

а) вивести значення $\sqrt{x}$ для парних чисел x у діапазоні від 2 до 10;
б) визначити суму цілих додатних непарних чисел менше 30.

14.2. Розв'язування обчислювальних задач

14.2.1. Особливості розв'язування обчислювальних задач

 Найважливішою вимогою до методів та засобів розв'язування обчислювальних задач є забезпечення необхідної *точності обчислень*. Цим терміном позначається міра наближення результату, обчисленого комп'ютером, до точного математичного результату. Абсолютно точний розв'язок існує лише для деяких задач опрацювання цілих чисел. Для більшості інших задач розв'язки, отримані комп'ютером, відрізняються від точних унаслідок помилок і похибок. *Помилки* можуть виникати в постановці задачі, у процесі розробки алгоритму та під час програмування. *Похибки* виникають через наближений характер обчислювальних методів, неточність початкових даних, заокруглення значень у процесі виконання арифметичних операцій тощо.

Двома головними методами розв'язування обчислювальних задач є пряме обчислення та послідовне наближення. *Пряме обчислення* застосовується, зокрема, для розв'язування систем лінійних алгебраїчних рівнянь методом виключення невідомих, обчислення площі круга за заданим радіусом, обчислення значення певної функції в заданій точці тощо. Методи прямого обчислення характеризуються високою точністю, а коректність обчислювального процесу гарантується самим методом.

Проте для більшості задач методів прямого обчислення або взагалі не існує, або вони характеризуються низькою швидкодією, а тому слід застосовувати методи *послідовного наближення*, чи *ітераційні методи*. Вони дають змогу отримати наближений розв'язок задачі шляхом багаторазового виконання однотипної обчислювальної процедури (тобто певної сукупності операцій). У такому разі початковими даними для кожної наступної процедури є результати виконання попередніх. Для обчислення кожного наступного члена геометричної прогресії використовується значення її попереднього члена. Процес виконання обчислювальної

процедури в методі послідовного наближення називається *ітерацією*. На кожній ітерації визначається наближений розв'язок задачі. Зазвичай зі збільшенням кількості ітерацій розв'язок стає щораз точнішим.

Зазначимо, що інколи ітераційні методи застосовуються для отримання точного розв'язку задачі. У цьому випадку кількість ітерацій, як правило, відома наперед. Типовим прикладом задач, що розв'язуються зазначеними методами, є обчислення членів числових послідовностей, зокрема прогресій.

Для пошуку наближеного розв'язку обчислювальних задач застосовуються ітераційні методи, кількість обчислювальних операцій в яких наперед не відома. Вона може залежати, наприклад, від заданої точності обчислення певної змінної або виразу. Так, кількість ітерацій, необхідних для отримання наближеного значення функції $\sin(x)$, прямо залежить від заданої точності обчислення.

Зауважмо, що більшість обчислювальних задач розв'язується кількома методами. Для вибору доцільного методу, як правило, керуються трьома критеріями: точністю, часом розв'язання та обсягом необхідної пам'яті.

Ітераційні методи передбачають використання циклів. Способи застосування циклічних структур розглядаються в підрозділах 7.2–7.4.

Перевіряємо себе

1. Які особливості виникають у процесі розв'язування обчислювальних задач?
2. У чому полягає сутність ітераційного методу обчислення?
3. За яких причин результат розв'язання задачі за допомогою комп'ютера може відрізнятися від правильного?
4. Які існують основні вимоги до методів та засобів розв'язування задач?

14.2.2. Циклічні рекурентні алгоритми

Характерною ознакою багатьох обчислювальних алгоритмів із циклічною структурою є те, що значення змінних усередині циклу визначається *рекурентно*. Це означає, що значення змінної на кожній ітерації обчислюється за певною формулою за допомогою значення цієї ж змінної, отриманого під час виконання попередніх ітерацій. Згадана формула називається *рекурентним співвідношенням*, або рекурентною залежністю. Використання рекурентних обчислень дає змогу зменшити кількість операцій, що виконуються в циклі, а отже, підвищити швидкодію алгоритму. Розглянемо приклади використання рекурентних співвідношень.

Приклад. Потрібно розробити блок-схему алгоритму і програму обчислення n членів арифметичної прогресії з виведенням значення кожного члена.

Якби вимагалося обчислити будь-який, але лише один член прогресії, наприклад, i-й, то доцільно було б скористатися відомою формулою: $u_i = a + (i-1)d$. Звичайно, цю ж формулу можна було б використати й для обчислення кожного з n-членів прогресії, проте ефективніше буде використати рекурентну залежність між ними:

$u_i = u_{i-1} + d$, $i = 1, 2, ..., n$; $u_0 = a$.

Зазначимо, що під час обчислення кожного члена прогресії за формулою $u_i = a + (i-1)d$ необхідно було виконати по одній операції множення, віднімання та додавання. У разі застосування рекурентної схеми потрібно виконати лише одну операцію додавання.

Блок-схема алгоритму обчислення і виведення значень членів арифметичної прогресії зображена на рис. 14.3. Нижче наведена також програма реалізації алгоритму.

```
program n14_3;
var i, n: integer;                {оголошення змінних цілочислового типу}
a, d, u: real;                    {оголошення змінних дійсного типу}
begin
  writeln ('увести n, a, d'); {повідомлення про уведення}
  readln (n, a, d);               {уведення значень змінних}
  u:=a;                           {перший член прогресії}
  for i:=1 to n do                {початок циклу}
  begin                           {оператори тіла циклу}
    writeln ('u=', u);            {виведення значення члена прогресії}
    u:=u+d                        {обчислення члена прогресії}
  end;                            {закриття операторних дужок}
  readln                          {призупинення виконання програми}
end.
```

Приклад. Потрібно розробити блок-схему алгоритму і програму обчислення значення многочлена

$$P(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + ... + a_{n-1}x + a_n$$

за певних значень величин x, n та коефіцієнтів a_i ($i = 0, 1, 2, ..., n$).

Винісши x за дужки у формулі многочлена, отримаємо

$$P(x) = (a_0x^{n-1} + a_1x^{n-2} + ... + a_{n-1}) \cdot x + a_n.$$

Потім винесемо спільний множник x за дужки многочлена, записаного в дужках. Отримаємо

$$P(x) = ((a_0x^{n-1} + a_1x^{n-3} + ... + a_{n-1}) \cdot x + a_{n-1})x + a_n.$$

Продовжимо виконувати аналогічні дії, доки не отримаємо результат

$$P(x) = (...(a_0x + a_1)x + a_2)x + a_3)x + ... + a_{n-1})x + a_n.$$

Аналіз останнього виразу дає змогу отримати таку рекурентну залежність:

$$P_i = P_{i-1}x + a_i, \text{ де } P_0 = a_0, \ i = 1, 2, ..., n.$$

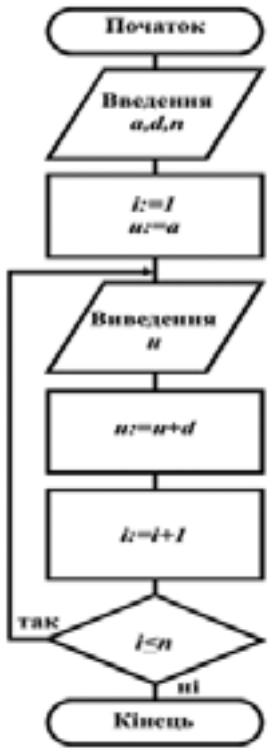


Рис. 14.3. Блок-схема алгоритму обчислення членів арифметичної прогресії

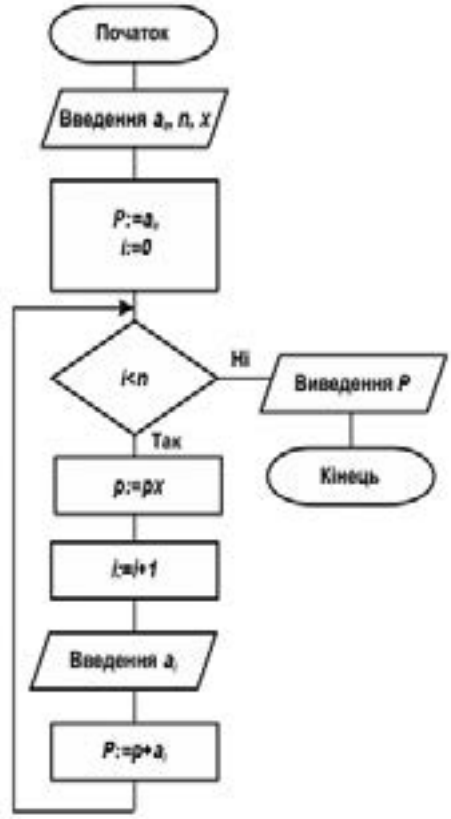


Рис. 14.4. Блок-схема обчислення многочлена за схемою Горнера

Описаний процес обчислень називається **схемою Горнера**. Блок-схема відповідного алгоритму зображена на рис. 14.4. Програма реалізації алгоритму наведена нижче.

```
program n14_4;
var x, a, p: real;                  {оголошення змінних}
i, n: integer;
begin
  writeln (’уведіть n, x, a[o]’);
  readln (n, x, a);                 {уведення початкових даних}
  p:=a;                             {початкове значення многочлена}
  for i:=0 to n do
  begin
    writeln ('увести a[',i,']=');
    readln (a);                     {уведення чергового коефіцієнта}
    p:=p*x+a;                       {застосування рекурентної формули}
  end;
  writeln (‘p= ’, p)                {виведення значення многочлена}
end.
```

Перевіряємо себе

1. Які залежності називають рекурентними?
2. Які переваги дає застосування рекурентних співвідношень у циклічних обчисленнях?
3. Як здійснюється обчислення многочлена за схемою Горнера?
4. Укажіть 4-й член послідовності, заданої рекурентним співвідношенням

$$x_{n+1} = 2 - \frac{1}{x_n},\ x_1 = 2.$$

5. Якими рекурентними співвідношеннями задаються послідовності
   а) 2, 4, 16, 256, ...
   б) 2, 0.5, 2, 0.5, 2, ...
   в) 2, 5, 8, 11, 14, ...
   г) 2, −4, 16, −256, ...?
6. Яку функцію змінної x обчислює програма? Вкажіть рекурентні співвідношення, за якими відбуваються обчислення.

А)
```
readln(x);
y:=0;
p:=1;
sgn:=1;
for i:=2 to 4 do
begin
p:=p*x;
y:=y+sgn*p;
sgn:=-sgn;
end;
writeln(y);
```

Б)
```
readln(x);
y:=0;
p:=1;
for i:=1 to 4 do
begin
p:=-p*x*x;
y:=y+p/i;
end;
writeln(y);
```

В)
```
readln(x);
y:=1;
p:=1;
for i:=1 to 4 do
begin
p:=p*(x-1);
y:=y+p;
end;
writeln(y);
```

Г)
```
readln(x);
y:=1;
p:=1;
for i:=1 to 4 do
begin
p:=-p/x;
y:=y+p;
end;
writeln(y);
```

Виконуємо

1. Задана послідовність натуральних чисел: 1, 2, 3, ..., n. Використовуючи рекурентні співвідношення, напишіть програму обчислення суми та добутку непарних і парних чисел.
2. Створіть блок-схему алгоритму обчислення значення виразу: $y = x + 2x + + 3x + ... + nx$. Розробіть для цього алгоритму програму.
3. Створіть блок-схему алгоритму обчислення значення виразу:

а) $y = 1 + a + a^2 + \ldots + a^n$;

б) $y = \frac{x}{2} + \frac{x^2}{4} + \frac{x^3}{6} + \ldots + \frac{x^n}{2n}$;

в) $y = 1 - a + a^2 - a^3 + \ldots - a^{2k-1}$;

г) $y = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^m}{m!}$;

д) $y = \sqrt{2 + \sqrt{4 + \sqrt{6 + \ldots + \sqrt{2n}}}}$.

14.2.3. Циклічні обчислювальні алгоритми з невідомою кількістю ітерацій

У розглянутих вище циклічних алгоритмах кількість ітерацій є наперед відомою. Проте на практиці часто зустрічаються алгоритми, кількість повторень циклу в яких можна визначити лише під час їхнього виконання. Кількість ітерацій у цих алгоритмах залежить від заданої точності обчислення математичного виразу.

Приклад. Задана спадна геометрична прогресія із першим членом a і знаменником q. Обчислити всі члени цієї прогресії, значення яких більші ε.

Члени геометричної прогресії пов'язані таким рекурентним співвідношенням:

$u_i = u_{i-1} \cdot q$, де $i = 2, 3, \ldots$;

$u_1 = a$; $|q| < 1$.

Нехай, наприклад, $a = 4$ і $q = 1/2$. Тоді членами прогресії будуть такі значення: 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, ...

Якщо ε = 1/15, буде обчислено 6 членів прогресії, і останнім у цьому випадку стане значення 1/8. Якщо ж ε = 0,02, то буде обчислено 8 членів. При цьому останній член прогресії дорівнюватиме 1/32.

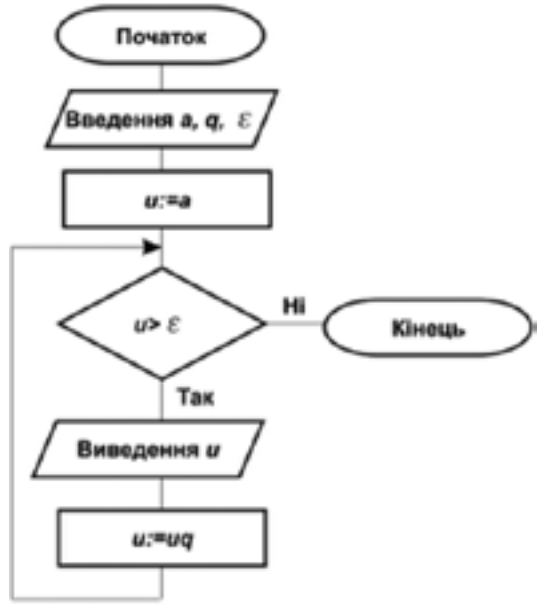


Рис. 14.5. Блок-схема алгоритму обчислення членів спадної геометричної прогресії

Блок-схему алгоритму обчислення членів спадної геометричної прогресії зображено на рис. 14.5. Програма, що розв'язує цю задачу, наведена нижче.

```
program n14_5;
var a ,u, g, e: real;          {оголошення змінних дійсного типу}
begin
  writeln ('увести a, g, e'); {повідомлення про введення значень змінних}
  readln (a, g, e);            {уведення значень змінних}
  u:=a;                        {перший член прогресії}
  while e<u do                 {початок циклу}
  begin                        {початок операторних дужок}
    writeln ('u=', u);         {виведення члена прогресії}
    u:=u*g                     {обчислення члена прогресії}
  end;                         {кінець операторних дужок}
  readln                       {призупинення виконання програми}
end.
```

Приклад. Із заданою точністю ε потрібно обчислити суму:

$$s = 1 + x + \frac{x^2}{2} + \frac{x^3}{2 \cdot 3} + \frac{x^4}{2 \cdot 3 \cdot 4} + \ldots + \frac{x^i}{2 \cdot 3 \cdot 4 \cdot \ldots \cdot i}.$$

Домовимося про такі позначення:

$$u_1 = x;\ u_2 = \frac{x^2}{2};\ u_3 = \frac{x^3}{2 \cdot 3};\ \ldots;\ u_i = \frac{x^i}{2 \cdot 3 \cdot \ldots \cdot i}.$$

Задана точність обчислення ε означає, що процес підсумовування триває, поки $u_i > \varepsilon$. Визначимо рекурентну залежність між доданками.

Оскільки $u_i = \frac{x^i}{2 \cdot 3 \cdot \ldots \cdot i}$ і $u_{i-1} = \frac{x^{i-1}}{2 \cdot 3 \cdot \ldots \cdot (i-1)}$ то $u_i = u_{i-1} \cdot \frac{x}{i}$.

Отже, обчислення суми із заданою точністю може виконуватися так:

$s_1 = s_{i-1} + u$, де $u_i = u_{i-1} \frac{x}{i}$; $i = 1, 2, 3$; $S_0 = 1$; $u_0 = 1$.

Ітераційний процес можна завершувати тоді, коли останній член стане меншим від заданої величини ε. Блок-схему алгоритму обчислення суми зображено на рис. 14.6. Програма, що розв'язує цю задачу, наведена нижче.

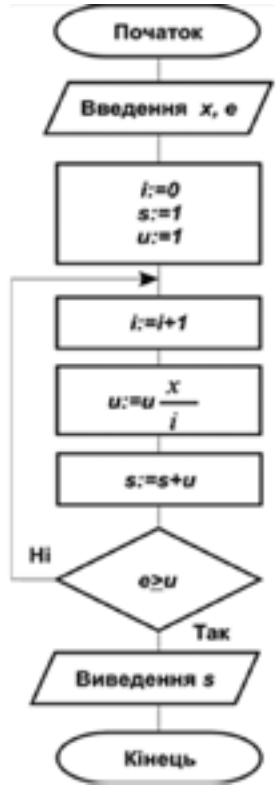


Рис. 14.6. Блок-схема алгоритму обчислення суми із заданою точністю

```
program n14_6;
var s, u, x, e: real;          {оголошення змінних дійсного типу}
i: integer;                    {оголошення цілочислової змінної}
begin
writeln ('увести x, e'); {повідомлення про введення значень змінних}
```

```
    read (x, e);          {введення значень змінних}
    u:=1; s:=1; i:=0;     {ініціалізація змінних}
    repeat                {початок циклу}
    i:=i+1;               {збільшення номера члена ряду}
    u:=u*x/i;             {обчислення значення чергового члена ряду}
    s:=s+u;               {обчислення поточної суми членів ряду}
    until e>=u;           {перевірка умови завершення циклу}
  writeln ('s=', s);      {виведення значення суми}
  readln                  {призупинення виконання програми}
  end.
```

Виконуємо

1. Розробіть блок-схему алгоритму і програму обчислення суми $s = x + \frac{x^2}{2} + \frac{x^3}{3} + ... + \frac{x^i}{i} + ...$, де $0 \le x < 1$.

Слід брати до уваги ті члени суми, що перевищують величину ε.

2. Складіть програму обчислення виразу 1+0,5+0,25+0,125+0,0625+... із заданою точністю ε.

14.2.4. Обчислювальні алгоритми із вкладеними циклами

Циклічний алгоритм, цикли якого не містять у собі інших циклів, називається *простим*. Отже, розглянуті вище алгоритми є простими. *Складним* називається циклічний алгоритм, принаймні один із циклів якого містить інший цикл. Цикл, що входить до іншого циклу, має назву *внутрішнього*, а цикл, який містить інший цикл, називається *зовнішнім*.

Приклад. Задано дві послідовності додатних чисел: {*a*}=5, 7, 9, 11, 13 та {*b*} = 3, 4, 5, 6, 7, 8. Потрібно розробити блок-схему алгоритму та програму обчислення площ усіх прямокутників, довжини сторін яких можна вибрати з послідовностей {*a*} і {*b*}.

Опишемо алгоритм. Нехай довжина однієї сторони (позначимо її літерою *a*) дорівнює 5. Обчислимо площі всіх прямокутників, другі сторони яких мають довжини, що дорівнюють значенням членів послідовності {*b*}. Змінивши довжину першої сторони (*a*=7), повторимо процедуру, і робитимемо так доти, доки *a* не перевищить 13. Блок-схему алгоритму зображено на рис. 14.7.

Нижче наведена програма, яка розв'язує цю задачу.

```
program n14_7;
var a, b: integer; {оголошення змінних цілочислового типу}
s: real;  {оголошення змінної дійсного типу}
begin
```

Рис. 14.7. Блок-схема алгоритму обчислення площ прямокутників

```
a:=5;  {початкове значення змінної a}
repeat  {зовнішній цикл}
for b:=3 to 8 do  {внутрішній цикл}
begin          {початок операторних дужок}
s:=a*b;        {обчислення площі поточного
               прямокутника}
writeln ('s=',s) {виведення площі поточного
               прямокутника}
end;           {кінець операторних дужок}
writeln;       {виведення порожнього рядка}
a:=a+2         {обчислення наступного значення
               змінної a}
until a>13; {перевірка умови завершення
               зовнішнього циклу}
readln  {призупинення виконання програми}
end.
```

Виконуємо

1. Розробіть блок-схему алгоритму і програму обчислення об'єму конусів $(V = \frac{1}{3}\pi r^2 h)$, якщо радіус основи (r) і висота конуса (h) набувають таких значень: r = 2,5; 3; 3,5; 4; 4,5; 5; 5,5; 6; h = 5; 6; 7; 8; 9; 10; 11; 12.

2. Складіть блок-схему алгоритму і програму перевірки заданого натурального числа на простоту. Число a називається простим, якщо його дільниками є лише 1 та a.

Підказка. Остачу від ділення числа a на $в$ можна обчислити за допомогою операції a *mod* b.

14.3. Розроблення програм для циклічних алгоритмів

Нижче наведені деякі типові програми для циклічних алгоритмів.

1. Дано послідовність чисел 1, 3, 5, 7, 9, 11, ..., n. Розробити програму обчислення їх добутку.

Використаємо такі змінні: z – поточне значення числа; i – лічильник кількості помножених чисел; p – поточне значення добутку чисел; k – кількість чисел у заданій послідовності.

Кількість чисел у заданій послідовності можна обчислити за формулою $k=(n+1)/2$. Зміст програми обчислення добутку може бути таким.

```
program n14_8;
var i, z, n, k, p: integer;{оголошення змінних цілочислового типу}
```

```
begin
  writeln (’увести n’);     {повідомлення про введення найбільшого
                             числа послідовності}
  readln (n);               {уведення найбільшого числа послідовності
  p:=1; z:=1;               {ініціалізація початкових значень змінних}
  k:=(n+1)/2;               {обчислення кількості чисел у послідовності}
  for i:=1 to n do          {початок циклу}
  begin                     {початок операторних дужок}
    p:=p*z;                 {обчислення поточного добутку чисел}
    writeln (’p(‘,i,’)=’,p); {виведення поточного добутку чисел}
    z:=z+2                  {вибір чергового значення числа}
    end;                    {кінець операторних дужок}
  readln                    {призупинення виконання програми}
end.
```

2. Дано додатне дійсне число a. Розробити програму визначення мінімального значення n, при якому виконується умова $a<2^n$.

Позначимо змінною p поточне значення обчислювального виразу. Значення змінної n уводитимемо за допомогою клавіатури. Нижче наведена програма обчислення значення змінної n.

```
program n14_9;
var a, n, p: integer;     {оголошення змінних цілочислового типу}
begin
  writeln (’увести a’);   {повідомлення про введення значення змінної a}
  readln (a);             {уведення значення змінної a}
  n:=0; p:=1;             {ініціалізація початкових значень змінних}
  while a>p do            {початок циклу}
  begin                   {початок операторних дужок}
    n:=n+1; p:=p*2        {обчислення виразу, збільшення показника
                           ступеня}
    end;                  {кінець операторних дужок}
    writeln (’n=’, n);    {виведення результату}
  readln                  {призупинення виконання програми}
end.
```

3. Розробити програму визначення суми цілих парних чисел, що більші 30, але менші 90.

Позначимо змінною x поточне значення парного числа, а змінною s – значення суми. Програма обчислення зазначеної суми чисел може бути такою:

```
program n14_10;
var s, x: integer;          {оголошення змінних цілочислового типу}
begin
  s:=0; x:=30;              {ініціалізація початкових значень змінних}
```

```
  while x<90 do              {початок циклу}
    begin                    {початок операторних дужок}
    x:=x+2; s:=s+x           {обчислення суми, вибір чергового числа}
    end;                     {кінець операторних дужок}
    writeln ('сума =', s);   {виведення значення суми}
  readln                     {призупинення виконання програми}
end.
```

4. Дано додатне натуральне число n. Розробити програму визначення в ньому кількості цифр і суми цих цифр.

Задачу можна розв'язати, наприклад, так. Спочатку виокремлюється крайня права цифра (молодший розряд), потім наступна цифра і т. д. Для виокремлення цифри молодшого розряду число ділиться повністю (цілком) на 10 і множиться на 10. Потім отримане значення віднімається від числа. Цифри, що виокремлюються, підсумовуються і обчислюється їх кількість. Використовуватимемо такі змінні: x – поточна цифра числа; s – сума цифр; k – кількість цифр у числі.

```
program n14_11;
var s, k, n, x: integer; {оголошення змінних цілочислового типу}
begin
  writeln ('увести число n');  {повідомлення про введення числа n}
  readln (n);              {уведення значення числа n}
  s:=0; k:=0;              {ініціалізація початкових значень змінних}
  while n>0 do             {початок циклу}
  begin                    {початок операторних дужок}
    x:=n mod 10;           {виокремлення поточної цифри у заданому числі}
    s:=s+x;                {обчислення суми цифр}
    k:=k+1;                {обчислення кількості цифр}
    n:=n div 10            {виокремлення чергового значення числа}
  end;                     {кінець операторних дужок}
  writeln ('сума цифр =',s,',кількість цифр =',k); {виведення
                           результатів}
  readln                   {призупинення виконання програми}
end.
```

Уведіть, виконайте програму і доведіть, що вона функціонує правильно.

5. Первинний внесок до банку становить 20000 грн. За кожний рік нараховується 15 % річних. За кожний рік знімається 500 грн. Розробити програму, що визначає суму внеску за перших n років.

```
program n14_12;
var i, n: integer;          {оголошення змінних цілочислового типу}
s: real;                    {оголошення змінної дійсного типу}
begin
```

```
  writeln ('увести значення n');      {повідомлення про введення
                                       значення змінної}
  readln (n);                         {уведення значення змінної}
  s:=20000;                           {первинний вклад}
  for i:=1 to n do                    {початок циклу}
    begin                             {початок операторних дужок}
    s:=s+0.15*s-500;                  {обчислення суми внеску за поточний рік}
    writeln ('за ', i ,'-ий рік=',s)  {виведення суми внеску за поточний
                                       рік}
    end;                              {кінець операторних дужок}
  readln                              {призупинення виконання програми}
end.
```

Виконайте програму й доведіть, що вона функціонує правильно.

Виконуємо

1. Перший член арифметичної прогресії дорівнює a, а різниця прогресії – d. Розробіть програму, що визначає, скільки необхідно підсумувати членів прогресії, щоб їх сума перевищила p.

2. На рахунок у банку поклали 3000 грн. За кожний повний рік збереження коштів нараховується 12 % річних. Розробіть програму, що визначає, через скільки років сума вкладу буде не менше 4500 грн.

3.  Зібрано a гарбузів. У кожний з n контейнерів покладено однакову кількість гарбузів. Розробіть програму, що визначає, скільки гарбузів було покладено в кожний контейнер і скільки гарбузів залишилося.

4.  Розробіть програму обчислення значення виразу:

$$5 + 4x + 3x^2 + 2x^3 + x^4.$$

Практична робота № 25

Тема:	Розробка алгоритмів з послідовними повтореннями та їх реалізація у вигляді програм
Мета:	Навчитися реалізовувати у вигляді програм алгоритми з кількома послідовними повтореннями

Завдання

У таблицях 14.1 і 14.2 наведені завдання для виконання практичної роботи. У табл. 14.1 наведені завдання, для розв'язування яких використовується оператор циклу **for…to…do**. У табл. 14.2 наведені складніші завдання. Для розв'язування цих завдань застосовуються різні типи операторів циклу. Для отриманого варіанта розробити програму розв'язання задачі та контрольний варіант розв'язку задачі. На практичній роботі учень уводить програму і налагоджує її. Необхідно добитися, щоб результат виконання програми збігся з контрольним варіантом.

Таблиця 14.1

Варіант	Завдання
1	Визначити площі 10 кругів. Радіус першого дорівнює 3 м, а радіус кожного наступного збільшується на 2 м
2	Радіус першого конуса дорівнює 2 м, а висота 10 м. Радіус кожного наступного конуса збільшується на 1 м, а висоти незмінні. Визначити об'єми 10 конусів
3	Визначити об'єми 10 куль. Радіус першої дорівнює 1 м, а кожної наступної збільшується на 2 м
4	Сторона a першого прямокутника дорівнює 4 м, а сторона b – 7 м. У кожному наступному прямокутнику сторона a збільшується на 1 м, а сторона b – незмінна. Визначити площі 10 прямокутників

Таблиця 14.2

Варіант	Завдання	Початкові дані
1	Задані два парних числа a_1 і a_2. Знайдіть суму всіх непарних чисел, більших a_2, але менших a_2.	a_1=20; a_2=60
2	Для дослідження нового автомобіля вирішено першого дня проїхати z км, а кожного наступного дня збільшувати пробіг на y відсотків порівняно з попереднім днем. Через скільки днів дистанція пробігу досягне s км?	z=100; y=3; s=500
3	Усі циліндри мають радіус основи r, висота першого дорівнює h, а висота кожного наступного збільшується на z. Висота останнього дорівнює H. Знайдіть об'єм кожного циліндра	r=5; h=3; z=0.2; H=5
4	З яблуні зняли m яблук і поклали порівну в n кошиків ($m>n$), але залишилося кілька яблук, менше ніж n. Скільки яблук поклали в кожний кошик і скільки залишилося?	m=37; n=7

Виконання роботи

1. Завантажити систему Turbo Pascal. Якщо не відкритий порожній файл, відкрити його (F10→File→Enter→New→Enter).

2. Увести програму. Умисно зробити помилку у слові for, наприклад, увести far.

3. Порівняти введену програму з розробленою. Виправити всі виявлені помилки. Помилку у слові far не виправляти.

4. Зберегти програму в робочому каталозі.

5. Виконати програму. Якщо в програмі до оператора циклу відсутні помилки, на екран буде видане повідомлення про помилку в операторі for. Виправити помилку та виконати програму.

6. Якщо в програмі є інші синтаксичні помилки, транслятор видаватиме на екран певні повідомлення.

7. Якщо програма виконана повністю, це означає, що синтаксичні помилки в ній відсутні, однак можуть бути логічні помилки. Викликати результат виконання програми на екран, для чого натиснути клавіші Alt+F5.

8. Порівняти отриманий результат із контрольним. Якщо результат правильний, зберегти програму (F10→File→Enter→Save→Enter). Перейти до пункту 10.

9. Якщо результат неправильний, знайти логічну помилку. Повторити дії, починаючи з пункту 6.

10. Виконати програму покроково, для чого послідовно натиснути клавішу F7. Переконатися, що програма функціонує правильно.

11. Установити режим для виконання програми з одночасним контролем поточних значень двох змінних.

12. Виконати програму в покроковому режимі. Проаналізувати значення змінних, які з'являються у вікні Watch, та переконатися, що програма функціонує правильно. Закрити вікно Watch (Alt+F3).

Практична робота № 26	**Тема:**	Розроблення рекурентних алгоритмів та їх реалізація у вигляді програм
	Мета:	Ознайомитися зі специфікою програмних описань рекурентних алгоритмів

Завдання

1. Скласти програму виведення послідовності факторіалів перших десяти натуральних чисел.

2. Скласти алгоритм виведення перших N (вводиться з клавіатури) чисел Фібоначчі:

1, 1, 2, 3, 5, 8, 13, 21, ...

Практична робота № 27	**Тема:**	Розроблення алгоритмів з вкладеними повтореннями та їх реалізація у вигляді програм
	Мета:	Ознайомитися з особливостями подання у вигляді програм алгоритмів із вкладеними повтореннями

Завдання

1. Із давніх часів математики намагалися вивести формулу, за допомогою якої можна обчислювати прості числа. Леонард Ейлер запропонував формулу $n^2 - n + 41$, де n – ціле невід'ємне число.

Перевірте справедливість цієї формули для всіх n від 0 до 50. Чи можна зробити висновок про правильність формули?

2. Дано текстовий файл, у якому спочатку записане прізвище учня, а потім усі його тематичні оцінки з інформатики за семестр, у наступному рядку – прізвище наступного учня, оцінки якого записані рядком нижче, і т. д. Учні можуть навчатися в різних класах за різними програмами, тому кількість тематичних оцінок може бути різна.

Створити програму, яка виводитиме на екран прізвища всіх учнів і їх семестрові оцінки (середнє арифметичне тематичних оцінок, заокруглене за загальноприйнятими правилами).

СЛОВНИЧОК

Метод ітерацій – багаторазове виконання однотипної обчислювальної процедури з метою наближення отриманого результату до істинного.

Оператор циклу з параметром – оператор, що завершує свою роботу, якщо поточне значення параметра перевищує кінцеве.

Оператор циклу з передумовою – оператор циклу, в якому умова продовження циклу перевіряється перед початком кожного виконання операторів тіла циклу.

Оператор циклу з післяумовою – оператор, у якому умова завершення циклу перевіряється після кожного виконання операторів тіла циклу.

Простий циклічний алгоритм – циклічний алгоритм, що не містить інших циклів.

Рекурентна залежність – залежність, у якій значення змінної на кожній ітерації обчислюється за допомогою значення цієї ж змінної, отриманого під час виконання попередніх ітерацій.

Точність обчислення – міра наближення результату, обчисленого комп'ютером, до точного значення.

ЗМІСТ

НАВЧАЛЬНЕ ВИДАННЯ

ГУРЖІЙ Андрій Миколайович
КАРТАШОВА Любов Андріївна
ЛАПІНСЬКИЙ Віталій Васильович
РУДЕНКО Віктор Дмитрович

ІНФОРМАТИКА

для загальноосвітніх навчальних закладів
з поглибленим вивченням інформатики

Підручник для 8 класу
загальноосвітніх навчальних закладів

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Редактор *Л.В. Дячишин*
Художній редактор *І.Б. Шутурма*
Коректор *О.А. Тростянчин*

Макет, підбір та колажування ілюстрацій *В.В. Лапінський*
Обкладинка *О.В. Шингур*

Формат $70\times100^{1}/_{16}$.
Ум. друк. арк. 23,98. Обл.-вид. арк. 22,20.
Тираж 4885 пр. Зам. №

Державне підприємство
"Всеукраїнське спеціалізоване видавництво "Світ"
79008 м. Львів, вул. Галицька, 21
Свідоцтво суб'єкта видавничої справи ДК № 4826 від 31.12.2014
www.svit.gov.ua
e-mail: office@svit.gov.ua
svit_vydav@ukr.net