

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**Курс лекцій з дисципліни
«Обчислювальна математика та
програмування»**

**для студентів напрямку підготовки
«Хімічна технологія»**

Донецьк 2011

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**Курс лекцій з дисципліни
«Обчислювальна математика та
програмування»**

**для студентів напрямку підготовки
«Хімічна технологія»**

Розглянуто
на засіданні кафедри ОМіП
протокол № 7 от 8.02.11

Затверджено
навчально-видавничою
Радою ДонНТУ
протокол № 2 від 21.03.11

Донецьк 2011

УДК 681.3.06(071)

Курс лекцій з дисципліни "Обчислювальна математика та програмування" для студентів напрямку підготовки «Хімічна технологія»/ Лазебна Л.О. – Донецьк: ДонНТУ, 2011. – 199 с.

Курс лекцій з дисципліни «Обчислювальна математика та програмування» для студентів напрямку підготовки «Хімічна технологія» складається з 6 розділів. Перший розділ присвячений питанням вивчення операційної системи Windows, другий розділ - основам алгоритмізації, третій розділ – програмуванню на мові Паскаль. Четвертий розділ містить завдання для самостійного рішення. У подальших розділах розглядаються складові пакету Microsoft Office: Word, Excel.

Кожен розділ містить теоретичний матеріал, приклади, які спрямовані на розвиток алгоритмічного мислення і навичок конструювання алгоритмів, а також розробки програм на мові Паскаль, навичок роботи з текстом і таблицями.

Курс лекцій відповідає робочій програмі з дисципліни «Обчислювальна математика та програмування» для студентів напрямку підготовки «Хімічна технологія».

Автор: Лазебна Л.О., ст. викладач

Рецензенти: І.Ю. Анохіна., к.т.н, доц.

Відп. за випуск: Павлиш В.М., д.т.н., проф.

З М І С Т

Вступ	7
1 Операційна система WINDOWS	8
1.1 Завантаження ОС WINDOWS. Робочий стіл	8
1.2 Типи вікон. Структура вікна	11
1.3 Операції з вікнами	15
1.4 Файлова система	16
1.5 Основні операції з об'єктами	19
1.5.1 Створення об'єктів	19
1.5.2 Виділення групи об'єктів	21
1.5.3 Видалення об'єктів	21
1.5.4 Перейменування об'єктів	22
1.5.5 Копіювання і переміщення об'єктів	22
1.6 Пошук об'єктів	23
1.7 Властивості об'єктів	24
1.8 Створення ярлика	25
2. Основи алгоритмізації	28
2.1 Етапи підготовки і рішення задачі	28
2.2 Алгоритм і форми його зображення	28
2.3 Типові структури алгоритмів	31
2.3.1 Лінійні і розгалужені алгоритми	31
2.3.2 Алгоритми циклічної структури	36
2.4 Алгоритми із структурою вкладених циклів	41
2.5 Алгоритми визначення суми, кількості і добутку	44
2.6 Алгоритми ітераційних циклів	46
2.7 Алгоритми формування і обробки одномірних масивів	47
2.8 Алгоритми визначення найбільшого і найменшого значення	54
2.9 Алгоритми формування і обробки двомірних масивів	59
2.10 Алгоритми, які містять допоміжні підзадачі	65
3. Основи програмування на мові Pascal	69
3.1 Середовище програмування Turbo Pascal	69
3.2 Алфавіт мови Pascal	73

3.3 Типи даних	74
3.4 Операції і стандартні функції мови Pascal. Вирази.	76
3.5 Структура програми	80
3.6 Оператори	81
3.6.1 Оператор привласнення	81
3.6.2 Введення-виведення даних	82
3.6.3 Умовний оператор If	85
3.6.4 Оператор безумовного переходу	87
3.6.5 Оператор вибору	90
3.6.6 Оператори циклу	93
3.7 Приклади програмування обчислювальних процесів	95
3.8 Обробка та формування масивів	104
3.9 Загальні відомості про підпрограми	115
3.9.1 Процедури	116
3.9.2 Функції	119
3.9.3 Приклад програмування завдання з використанням підпрограм	120
4. Завдання для самостійного рішення	125
5. Текстовий редактор Microsoft Word	130
5.1 Екранний інтерфейс редактора	130
5.2 Початок роботи у текстовому редакторові	130
5.3 Створення нового документа, введення тексту і його збереження	133
5.4 Основи редагування документа	135
5.5 Основи форматування документа	138
5.5.1 Форматування символів	139
5.5.2 Форматування абзацу	141
5.5.3 Створення рамок і заливок	142
5.5.4 Створення колонок тексту	144
5.5.5 Створення нумерованих і маркірованих списків	145
5.6 Створення колонтитулів	146
5.7 Робота з таблицями	148
5.7.1 Створення таблиці	148
5.7.2 Форматування таблиці	149

5.7.3 Робота з даними в таблиці	151
5.8 Графіка в документі WORD	153
5.8.1 Вставка картинки з колекції	153
5.8.2 Вставка діаграми	154
5.8.3 Використання автофігур і напису	155
5.8.4 Використання об'єкту WordArt	157
5.8.5 Набір математичних формул	158
6. Електронна таблиця MICROSOFT EXCEL	160
6.1 Інтерфейс електронної таблиці EXCEL	160
6.2 Основні принципи роботи в EXCEL	162
6.3 Створення робочої книги і її збереження	163
6.4 Дії з робочими листами	164
6.5 Основи форматування робочого листа	166
6.5.1 Виділення групи осередків	166
6.5.2 Зміна зовнішнього вигляду інформації осередків	167
6.5.3 Форматування чисел	169
6.5.4 Обрамлення і заливка осередків	170
6.6 Зміна структури робочої таблиці	172
6.6.1 Додавання і видалення рядків і стовпців	172
6.6.2 Копіювання і переміщення даних	174
6.7 Створення формул	176
6.7.1 Створення простих формул	176
6.7.2 Створення формул з використанням Майстра функцій	178
6.7.3 Приклади створення формул	179
6.7.4 Копіювання формул	187
6.8 Автоматичне введення даних	188
6.9 Створення й редагування діаграм	190
Список літератури	198

ВСТУП

Впровадження комп'ютерної техніки у всі сфери людської діяльності послужило поштовхом до зародження нової наукової і прикладної дисципліни – інформатики. Вперше цей термін став використовуватися у Франції в 60-роках. У англійських країнах йому відповідає синонім computer science (наука про комп'ютерні технології або комп'ютерна наука). Інформатика визначає сферу людської діяльності, пов'язану з процесами зберігання, перетворення і передачі інформації за допомогою комп'ютера.

Але могутність комп'ютера визначається людиною і тими знаннями, якими він володіє. Це зумовлює необхідність, з одного боку, підвищення рівня підготовки фахівців в області проектування комп'ютерів і їх програмного забезпечення, а з іншого боку, підвищення рівня комп'ютерної письменності користувачів, тобто наукових, інженерно-технічних і адміністративних працівників, що не є фахівцями в області програмного забезпечення.

Для успішного використання комп'ютерів, грамотної постановки завдань на розробку програмних комплексів інженерові будь-якої спеціальності необхідне знання алгоритмізації, процесів підготовки і рішення задач на ЕОМ, етапів і принципів розробки програм.

В процесі вивчення інформатики треба не тільки навчитися працювати на комп'ютері, але і уміти цілеспрямовано його використовувати для пізнання і творення навколишнього нас світу. Можна навчитися працювати з персональним комп'ютером так, що рішення будь-яких математичних задач, створення і друк всіляких текстів, побудова графіків доставить задоволення, а можна з його допомогою просто слухати музику, дивитися фільми.

1 ОПЕРАЦІЙНА СИСТЕМА WINDOWS

1.1 Завантаження ОС WINDOWS. Робочий стіл

Найпоширенішою в світі операційною системою (ОС) тривалий час був виріб фірми Microsoft - ОС MS DOS або просто DOS. Фактично DOS був загальним стандартом для персонального комп'ютера. Новий стандарт – ОС Microsoft Windows - продовжує славні традиції MS DOS.

Операційна система - це програма, яка завантажується при включенні комп'ютера. Вона проводить діалог з користувачем, здійснює управління комп'ютером, його ресурсами (оперативною пам'яттю, місцем на дисках і т.п.), запускає інші (прикладні) програми на виконання.

WINDOWS - є багатозадачною ОС, що має графічний інтерфейс, який заснований на зображеннях у вигляді картинок. *Інтерфейс* - це логіка побудови вікон, меню, команд, настройок. *Багатозадачність* ОС означає, що при роботі з нею можна запустити відразу декілька програм і працювати з ними паралельно. Наприклад, поки одна програма виводить результати на принтер, можна перемкнутися на іншу програму, виконувати обчислення, працювати з текстом і т.п.

У ОС Windows спілкування відбувається в основному через меню і вікна. Для роботи з екраном, що містить графічні елементи використовувати клавіатуру незручно, тому основним інструментом є *миша*.

Для запуску ОС Windows не треба вводити ніяких команд. Просто необхідно включити комп'ютер і через декілька митей опиняєшся в системі.

Після завантаження системи на екрані з'явиться *робочий стіл* – це графічне середовище, на якому відображаються об'єкти Windows і елементи управління (рис.1.1).

Набір елементів, що з'являється на робочому столі при запуску Windows, залежить від настройки комп'ютера.

Найважливіші елементи робочого столу:

- системна папка **Мой компьютер** дозволяє отримати доступ до ресурсів: дискам, папкам, файлам, принтерам;

- системна папка **Сетевое окружение** дозволяє отримати доступ до локальної мережі, до якої підключений даний комп'ютер;
- системна папка **Корзина** використовується для тимчасового зберігання видалених об'єктів;
- кнопка **Пуск** і панель завдань, яка розташована в нижній частині робочого столу. На панелі відображається список виконуваних в даний момент застосунків і область індикаторів.

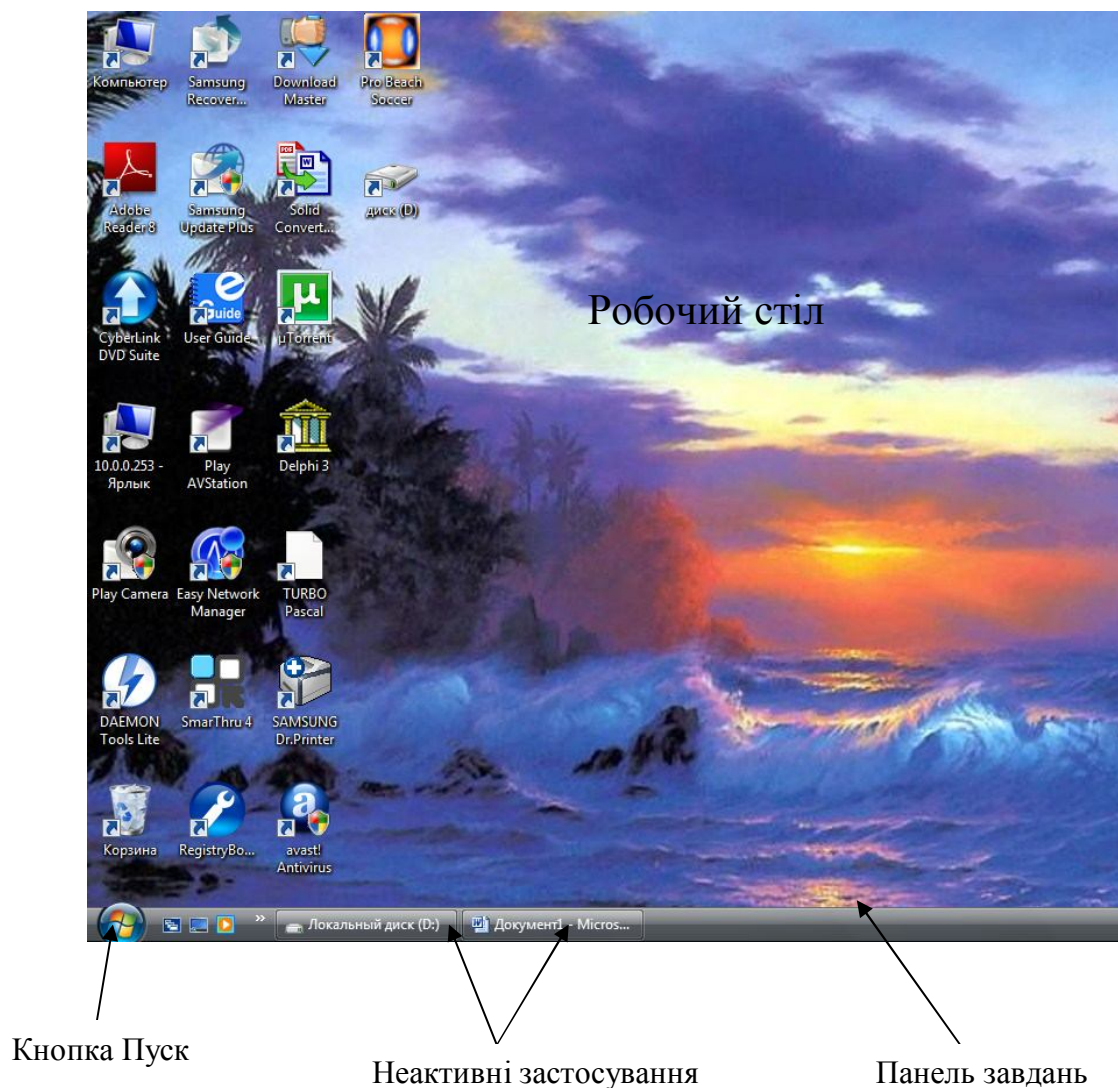


Рис.1.1 – Робочий стіл Windows

За допомогою кнопки **Пуск** відкривається головне меню Windows, яке містить все, що потрібно для роботи з ОС Windows (рис.1.2). Пункт із

стрілкою відкриває підміню. Щоб відкрити підміню досить підвести до нього покажчик миші. Пункт меню, який не містить стрілки, є командою, і для її запуску необхідно клацнути лівою кнопкою миші. Короткий опис пунктів головного меню Windows представлений в таблиці 1.1. Залежно від настройки комп'ютера і набору встановлених компонентів головне меню може містити додаткові команди.

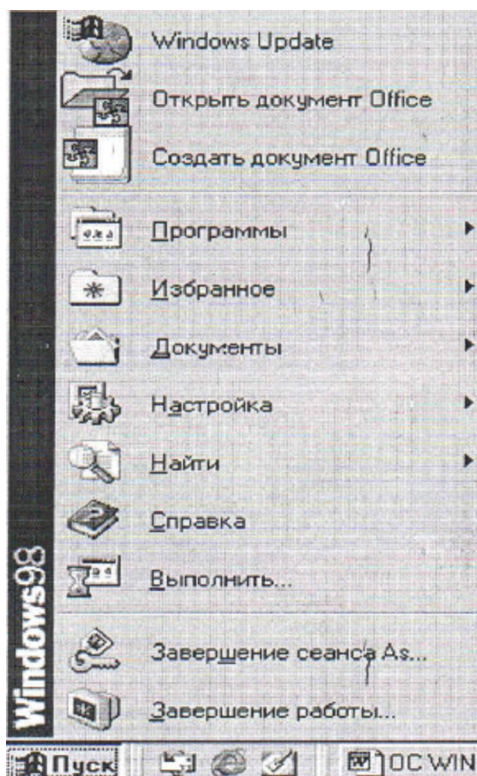


Рис.1.2 – Головне меню Windows

Управління в ОС Windows здійснюється за допомогою миші, з якою пов'язаний активний елемент управління – її покажчик.

За допомогою миші можна виконати наступні дії:

- **вказати**, це означає - перемістити покажчик миші на один з елементів екрану;

- **клацнути** - короткочасно натиснути на одну з кнопок миші. За допомогою клацання лівої кнопки миші зазвичай проводиться вибір одного з елементів. За допомогою клацання правої кнопки миші на виділеному об'єкті проводиться виведення *контекстного меню*. Це меню, яке містить всі команди, які можна виконати для даного об'єкту;

- **двічі клацнути**, це означає - швидко натиснути ліву кнопку миші для запуску на виконання команди або програми;

- **перетягнути** - натиснути кнопку миші (зазвичай ліву) і перемістити покажчик миші на нове місце робочого столу.

Таблиця 1.1 – Основні пункти головного меню Windows

Пункт меню	Призначення
Программы	Містить перелік всіх встановлених програм
Документы	Містить перелік документів, з якими працював користувач останнім часом
Настройка	Дає можливість налаштувати пристрої комп'ютера
Найти (Поиск)	Зручний і практичний засіб для пошуку папки, файлу і інших об'єктів
Справка	Виклик довідкової системи
Выполнить	Містить командний рядок, який містить назву програми, що запускається
Завершение сеанса	Завершити сеанс користувача, перемкнутися на роботу нового користувача
Завершение работы	Дозволяє перевести комп'ютер в режим чекання, вимкнути або перезавантажити його

1.2 Типи вікон. Структура вікна

Кожна програма в Windows працює в своєму вікні. *Вікно* - це деяка частина екрану, в якій може міститися список файлів і папок. У середовищі Windows передбачені вікна двох типів: прикладні і підлеглі, а також вікно діалогу.

У *прикладних* вікнах виконуються застосування – це програми, призначені для роботи з документами користувача, наприклад, WORD, EXCEL. *Документом* називається частина вікна застосування, в якому користувач може виконувати передбачені даним застосуванням дії: вводити

текст або дані, редагувати малюнки, проводити розрахунки і т.п. Вікно документа розташовується завжди усередині прикладного вікна.

Підлеглі вікна містять документи або використовуються середовищем для спілкування з користувачем, наприклад, вікно «Мой компьютер».

Вікна, в яких поміщаються запити до користувача для уточнення його дій, називаються *діалоговими вікнами*. Їх легко розпізнати, оскільки вони містять наступні елементи: розділи, перемикачі, списки, які розкриваються, текстові поля, а також кнопки «ОК», «Отмена» (рис.1.3).

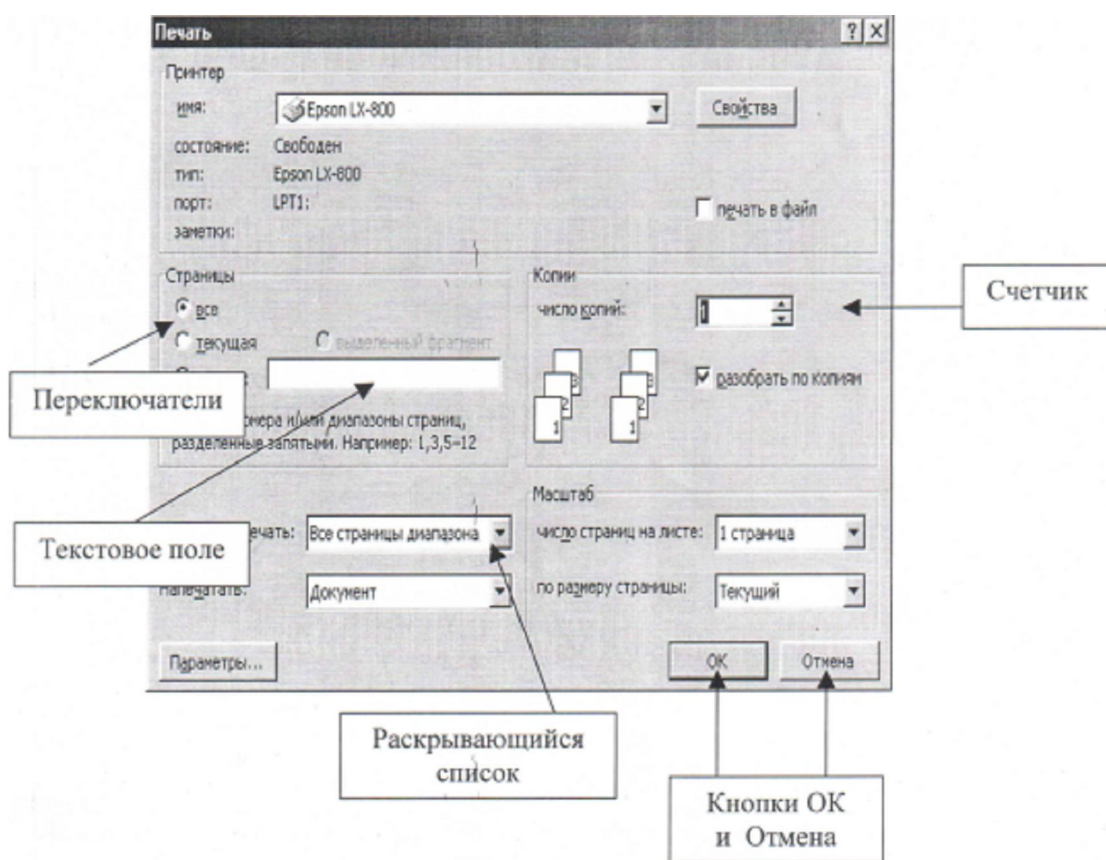




Рис.1.3 – Елементи діалогового вікна друку

У викликаному вікні будь-якого програмного забезпечення обов'язково справа присутні три кнопки управління вікном на екрані:

- кнопка **Свернуть**  використовується для того, щоб помістити вікна на панель завдань;

- кнопка **Развернуть**  використовується для того, щоб розвернути вікно на весь екран або вернути «розкрите» вікно в колишній стан;

- кнопка **Закрити**  використовується для того, щоб закрити вікно.

Структура будь-якого вікна в ОС Windows однакова, відрізняється тільки складністю його оформлення (рис.1.4).



Рис.1.4 – Структура вікна

Верхній рядок будь-якого вікна – це **рядок заголовка**, який містить кнопку системного меню, назву вікна і кнопки управління вікном.

Рядок меню знаходиться під заголовком і містить основні пункти меню: Файл, Правка і т.д. Щоб побачити команди даного пункту меню, необхідно клацнути на ньому мишею. Доступні для виконання команди будуть виділені чорним кольором, а не доступні - бляклого сірого кольору. Біля деяких команд вказані швидкі клавіші, наприклад, для виконання копіювання об'єкту замість команди **Правка/Копировать** досить натиснути

послідовно комбінацію клавіш **Ctrl** і **C**. Для закриття списку меню досить клацнути в будь-якому місці робочої області вікна.

Панель інструментів використовується для прискорення і спрощення виконання основних завдань. Як правило, натиснення кнопки на панелі інструментів еквівалентно вибору певної команди в пункті меню. Щоб визначити призначення кнопки, слід встановити на неї покажчик миші і трохи почекати. Через деякий час на екрані з'явиться короткий опис цієї кнопки.

Робоча область використовується для розміщення об'єктів.

Рядок стану містить корисну інформацію про об'єкти вікна.

Панель інструментів і рядок стану можуть бути відсутніми у вікні. Для їх підключення використовується пункт меню **Вид**.

У нижній і/або правій частині вікна, яке недостатньо велике для відображення всіх даних в ньому, автоматично додається смуга прокрутки (рис.1.5). Для прокрутки зображення в поточному вікні слід перетягнути бігунок або скористатися кнопками прокрутки.

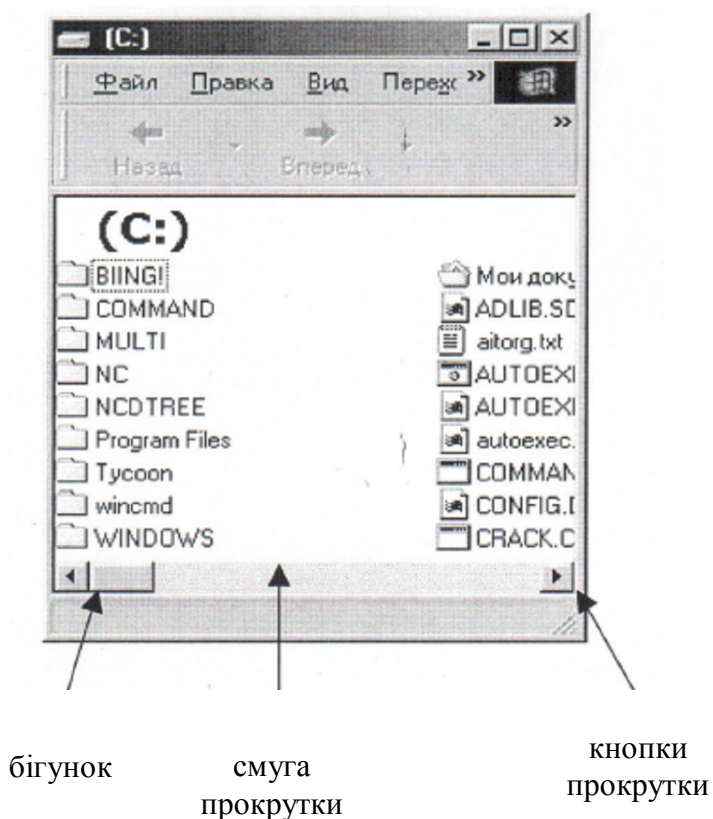
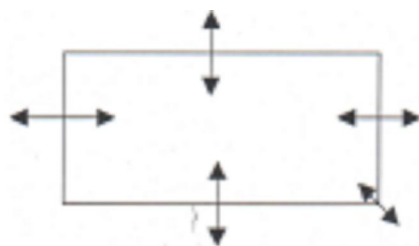


Рис.1.5 – Вікно, яке містить смуги прокрутки

1.3 Операції з вікнами

Вікно можна переміщати в будь-яке місце по екрану дисплея. Для переміщення вікна необхідно встановити покажчик миші на рядок заголовка, натиснути ліву кнопку миші і, утримуючи її, перемістити вікно в інше місце.

Для зміни розмірів вікна необхідно встановити покажчик миші на межу вікна. При цьому його форма зміниться залежно від того, на якій стороні вікна знаходиться покажчик. Натиснувши у цей момент ліву кнопку миші і, утримуючи її, перемістити межу вікна.



Windows дозволяє працювати з декількома вікнами. При цьому одне з вікон є активним. Розташовується воно на передньому плані, і рядок його заголовка підсвічується. Перемикається між вікнами можна двома способами:

- клацанням миші в області вікна (не обов'язково на рядку заголовка).

При цьому пасивне вікно стає активним і відображається на передньому плані;

- за допомогою клацання на кнопці відкритого вікна в панелі завдань.

На екрані може бути декілька відкритих вікон. Одні вікна можуть перекриватися іншими, а деякі можуть бути зовсім закриті рештою вікон. Існує можливість автоматично упорядкувати розташування відкритих вікон на моніторі:

- каскадом;
- зверху вниз;
- зліва направо.

Для цього необхідно на вільному місці в панелі завдань клацнути правою кнопкою миші, тобто викликати контекстне меню, в якому вибрати відповідну команду.

1.4 Файлова система

На робочому столі розташована системна папка «Мой компьютер», яка містить список дисків: Диск 3.5' (A:), (C:), (D:), CD-дисковод (E:) і інші папки, що мають спеціальне призначення (рис.1.6).

На будь-якому іншому комп'ютері можуть бути відсутніми деякі диски. Кількість локальних дисків, на які розбитий жорсткий диск, так само може розрізнятися.

Необхідно пам'ятати, що перші дві букви англійського алфавіту використовуються для позначення дисководів, які використовуються для роботи з дискетами. Наступні букви алфавіту визначають локальні диски. Потім позначаються CD-дисководи.

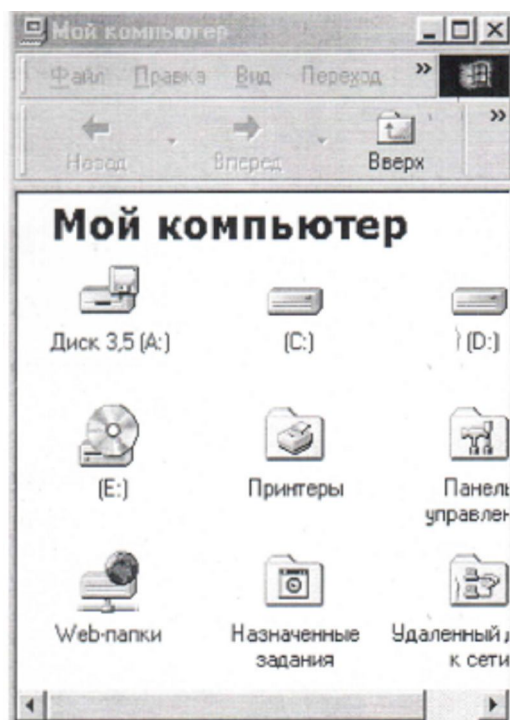


Рис. 1.6 – Вікно папки Мій комп'ютер

Для проглядання змісту будь-якого диска необхідно зробити подвійне клацання на імені диска, наприклад, диск C:. У вікні диска C:, що з'явиться, можна побачити його структуру: які об'єкти зберігаються на цьому диску (рис.1.7).

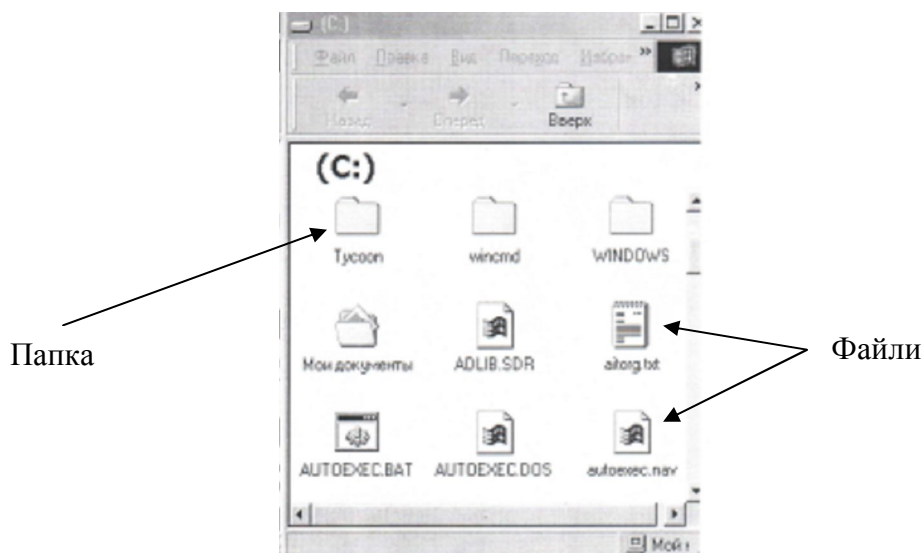


Рис. 1.7 – Вікно диска C:

Об'єкт – це, те чим оперує ОС Windows: папка, файл, програма, диск, ярлик і т.д. Кожен об'єкт має ім'я і графічне позначення (піктограма, значок) і властивості. Властивості визначають характеристики об'єкту.

У вигляді папок жовтого кольору зображуються папки. **Папка** призначена для угруповання об'єктів Windows. Вона нагадує звичайну папку, в яку можна складати папери, керуючись певними принципами. Наприклад, в одну папку можна складати ділові папери, в іншу – кулінарні рецепти і т.д. Папка може містити файли, інші папки або бути порожньою. Назва папки може бути будь-яка. Папка, з якою працює в даний момент користувач, називається поточною. Папка, яка знаходиться усередині іншої папки, називається вкладеною.

У кожного файлу своя піктограма, вона визначається програмою, в якій він був створений. **Файл** – це поіменована область пам'яті на диску або іншому носіїв інформації. Назва файлу складається з двох частин: імені і розширення. Ім'я визначає користувач. Розширення указує на тип програми, за допомогою якої він був створений і може містити не більше трьох - чотирьох символів (букви англійського алфавіту). Для завантаження файлу ОС Windows спочатку завантажує програму, в якій був створений файл,

використовуючи розширення, а потім завантажується інформація файлу.

Наприклад, часто використовувані розширення:

com, exe - програмні файли;

txt, doc, rtf – файли з текстовою інформацією;

xls – файли електронних таблиць;

mp3, wav – музичні файли;

avi – файли з відеоінформацією;

bmp, jpg – графічні файли.

У імені і розширенні файлу не можна використовувати наступні символи: / \ , : ? < > “ .

Ярлик – це спеціальний файл, що є посиланням на інший об'єкт, наприклад файл. Посилання на об'єкт – це шлях до об'єкту. Відмітна ознака ярлика - чорна стрілка в нижньому лівому кутку значка. Адреса файлу складається з імені диска, імен папок, в які він входить, і власного імені. Наприклад, D:\ Отчеты\ Поточний год\ налоги.xls. Похила межа, використовувана для розділення папок, називається слешем.

ОС Windows забезпечує декілька варіантів для зміни зовнішнього вигляду представлення інформації у вікні: «Крупные значки», «Мелкие значки», «Список», «Таблица». Для цього використовуються команди меню **Вид** (рис.1.8).

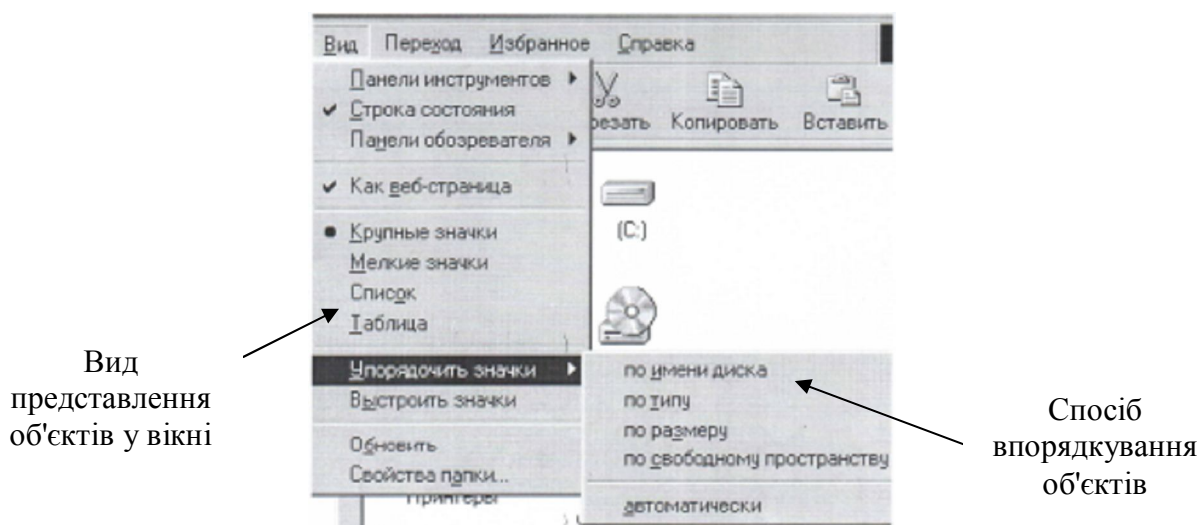


Рис. 1.8 – Вибір способу уявлення і впорядкування об'єктів

1.5 Основні операції з об'єктами

Основними операціями з об'єктами є:

- створення;
- видалення;
- перейменування;
- копіювання і переміщення.

1.5.1 Створення об'єктів

Для **створення папки** необхідно виконати наступні дії:

- відкрити диск і папку, в якій потрібно створити нову папку;
- вибрати команду **Файл/Создать/Папку** або в контекстному меню - команду **Создать/Папку**;
- ввести з клавіатури ім'я нової папки в рамочці зі значком нової папки і натиснути клавішу **Enter**.

ОС Windows дозволяє створити файл в будь-якому місці без явного відкриття застосування.

Для **створення файлу** необхідно виконати наступні дії:

- відкрити диск і папку, в якій потрібно створити файл;
- вибрати команду **Файл/Создать** і в підміню вибрати тип документа, наприклад, Текстовий документ. Створивши файл «Текстовий документ.txt», ОС Windows пропонує це ім'я змінити, але розширення не чіпати, інакше Windows «не дізнається» про файл і натиснути клавішу **Enter**.

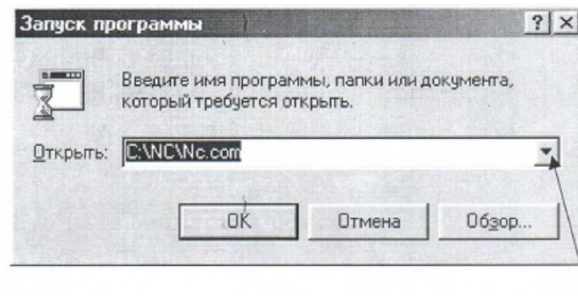
Для **створення файлу з використанням застосування** необхідно виконати наступні дії:

- запустити застосування, за допомогою якого буде створюватись файл (наприклад, Word, Блокнот, Excel). Запустити застосування або програму можна різним способом:

1) у головному меню Windows в пункті **Программы** вибрати папку, яка містить потрібну програму;

2) використати ярлик програми на робочому столі (якщо є);

3) якщо відоме ім'я програми, яку слід запустити, то найпростіше запустити її за допомогою команди **Виполнить**. У головному меню Windows вибрати команду **Виполнить** і в діалоговому вікні, що з'явиться, ввести ім'я програми або скористатися для пошуку імені програми кнопкою **Обзор**.



Розкриваючий список містить команди, які були введені раніше

- ввести необхідну інформацію;
- виконати збереження результатів роботи. Щоб зберегти новий документ необхідно вибрати команду **Файл/Сохранить как**; у вікні, яке з'явиться, заповнити текстове поле **Имя файла**, а із списку **Папка** вибрати диск і/або папку, в яку необхідно зберегти файл (рис.1.9). Потім натиснути кнопку **Сохранить**.

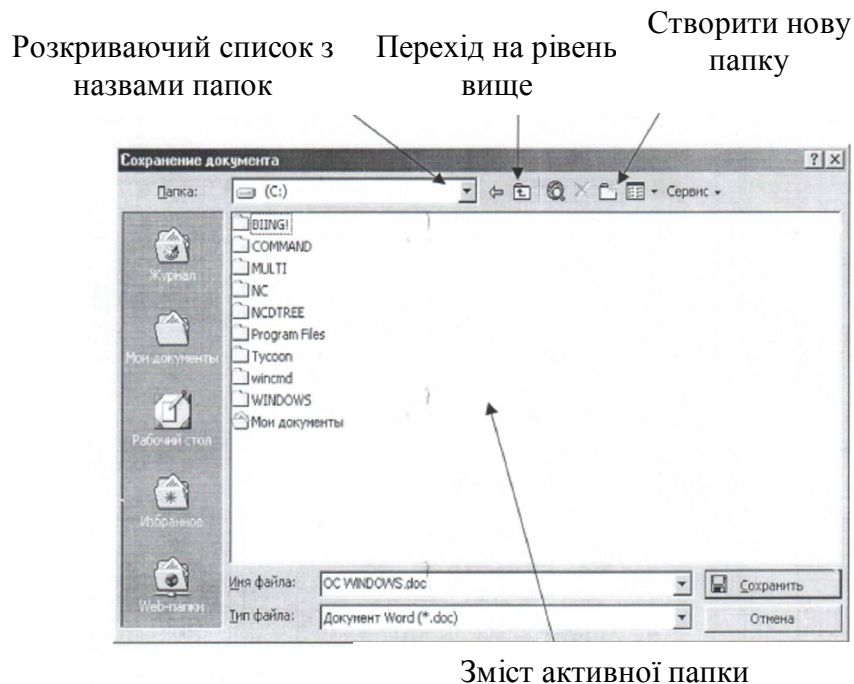


Рис.1.9 – Діалогове вікно збереження документа

Щоб зберегти зміни, внесені до існуючого файлу необхідно вибрати команду **Файл/Сохранить**.

1.5.2 Виділення групи об'єктів

Всі операції можна виконувати з одним об'єктом або ж з декількома, тобто з групою об'єктів. Якщо об'єктів декілька їх необхідно виділити.

Для **виділення групи несуміжних об'єктів** необхідно виконати наступні дії:

- виділити перший об'єкт;
- натиснути клавішу **Ctrl** і, утримуючи її, виділити інші об'єкти;
- відпустити клавішу **Ctrl**.

Для **виділення групи суміжних об'єктів** необхідно виконати наступні дії:

- встановити покажчик миші перед першим об'єктом;
- натиснути ліву кнопку миші і, утримуючи її, розтягнути прямокутну область, укладаючи в неї всі об'єкти групи.

Існує також інший спосіб для виділення групи суміжних об'єктів:

- виділити перший об'єкт;
- натиснути клавішу **Shift** і, утримуючи її, клацнути по останньому об'єкту групи.

Для виділення всіх елементів в поточному вікні служить команда **Правка/Выделить все**. Щоб зняти виділення з об'єктів необхідно клацнути на вільному місці у вікні.

1.5.3 Видалення об'єктів

Для **видалення об'єктів** необхідно виконати наступні дії:

- виділити об'єкт або групу об'єктів;
- натиснути клавішу **Delete** або вибрати команду **Файл/Удалить**.

Об'єкти, що видаляються, потрапляють в **Корзину** - це папка, яка представлена на екрані значком «**Корзина**». На кожному з дисків

створюються спеціальні директорії Recycled (помічені як приховані і системні), де видалені файли знаходяться до тих пір, поки користувач не видалить їх звідти. Не виключається можливість випадкового видалення об'єктів в корзину. Щоб відновити об'єкт необхідно вибрати команду **Файл/Восстановить**. Після цього об'єкт повинен зникнути з корзини і з'явитися в тому місці, звідки він був видалений. Якщо установка *«Уничтожатъ файлы сразу после удаления, не помещая их в корзину»* включена, то корзина завжди буде порожньою. За умовчанням корзина займає 10% від місткості диска.

Заповнення корзини уповільнює роботу ОС і може бути причиною браку пам'яті при копіюванні, збереженні файлів, може бути припинене автоматичне збереження файлу, особливо, якщо файл зберігає графічні об'єкти. Щоб очистити корзину необхідно вибрати команду **Файл/Очистить корзину**.

1.5.4 Перейменування об'єктів

Для **перейменування об'єктів** необхідно виконати наступні дії:

- виділити об'єкт;
- вибрати команду **Файл/Переименовать** або на виділеному об'єкті клацнути правою кнопкою миші і в контекстному меню вибрати команду **Переименовать**;
- ввести з клавіатури нове ім'я або клацнути мишею в полі імені, щоб з'явився текстовий курсор і відредагувати старе ім'я об'єкту;
- натиснути клавішу **Enter**.

1.5.5 Копіювання і переміщення об'єктів

При копіюванні об'єкту копія розміщується в тому місці, на яке було проведено копіювання, а оригінал об'єкту залишається на колишньому місці.

При переміщенні об'єкту копія розміщується в тому місці, на яке було проведено переміщення, а оригінал віддаляється із старого місця.

Для переміщення і копіювання об'єктів існує декілька способів.

Перший спосіб - за допомогою буфера обміну.

Для виконання цього способу необхідно виконати наступні дії:

- виділити об'єкт, який необхідно скопіювати або перемістити;
- вибрати команду **Правка/Копировать** для копіювання або **Правка/Вырезать** для переміщення;

Вырезать для переміщення;

- відкрити диск або папку, куди необхідно скопіювати або перемістити об'єкт;

- вибрати команду **Правка/Вставить**. В результаті, з буфера обміну копія об'єкту переміститься у вказане місце.

Другий спосіб - за допомогою миші.

Для виконання цього способу необхідно виконати наступні дії:

- виділити об'єкт, який необхідно скопіювати або перемістити;
- натиснути на об'єкті праву кнопку миші і, утримуючи її, перемістити об'єкт у відкриту папку (диск), куди необхідно перемістити або скопіювати об'єкт;

об'єкт;

- відпустити праву кнопку миші і в меню, що з'явиться, вибрати команду **Копировать** або **Переместить**.

1.6 Пошук об'єктів

Іноді потрібно знайти файл або папку, яка знаходиться на комп'ютері, але Ви не пам'ятаєте, де вона розміщена. А іноді необхідно знайти інформацію на чужому комп'ютері. Для пошуку інформації використовується пункт меню кнопки Пуск – **Найти/Файлы и папки**.

Пошук проводиться по критеріях, які встановлюються в діалоговому вікні **Найти** (рис.1.10). За допомогою пошуку файлів можна шукати інформацію по наступних критеріях: по імені, даті створення, типу і розміру.

На вкладці **«Имя и местоположение»** необхідно виконати наступні дії:

1) у полі **Имя** потрібно ввести ім'я об'єкту або шаблон імені. *Шаблоном* називається ім'я файлу, що містить спеціальні символи (*, ?), де * - зірочка

замінює будь-яку кількість будь-яких символів в імені файлу; ? - знак питання замінює лише один символ.

Наприклад,

*.doc – знайти файли з будь-яким ім'ям і вказаним розширенням;

text.* - знайти файли з вказаним ім'ям і будь-яким розширенням;

S??.txt - знайти файли, ім'я яких складається з трьох символів, причому перший з них буква S і з вказаним розширенням.

2) у полі **Где искать** необхідно вказати район пошуку (диск, всі диски або конкретна папка);

3) для початку пошуку натиснути кнопку **Найти**.

З'явиться список знайдених об'єктів, які можна переглядати, перейменувати, копіювати, видаляти.

Поле **Искать текст** дозволяє шукати файли не тільки по імені, але і по текстовому фрагменту.

Вкладка «**Дата**» дозволяє здійснювати пошук відповідно до дати зміни файлу.

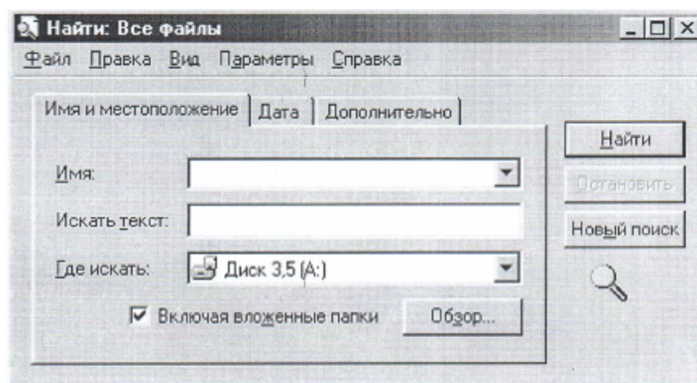


Рис.1.10 – Діалогове вікно для пошуку об'єктів

1.7 Властивості об'єктів

З властивостями будь-якого об'єкту в ОС Windows можна ознайомитися, використовуючи команду контекстного меню **Свойства** або команду **Файл/Свойства**. Залежно від об'єкту діалогове вікно **Свойства** може містити різну кількість вкладок. Вкладка **Общие** присутня завжди. Дана команда дозволяє багато що дізнатися про виділений об'єкт.

Наприклад, про папку можна дізнатися: скільки в ній файлів і вкладених папок; скільки місця вони займають на диску; атрибути. Приблизно теж і про файл (рис.1.11).

Атрибути – це позначки, які використовуються для позначення деяких характеристик файлів і папок. При бажанні атрибути можна включити або відключити. Існують наступні атрибути:

архівний - привласнюється файлу автоматично при його створенні, не накладає ніяких обмежень;

тільки для читання - встановлюється для файлів, які необхідно захистити від редагування;

прихований – привласнюється файлам, які не повинні відображатися у вікні;

системний - привласнюється файлам, які використовуються ОС.

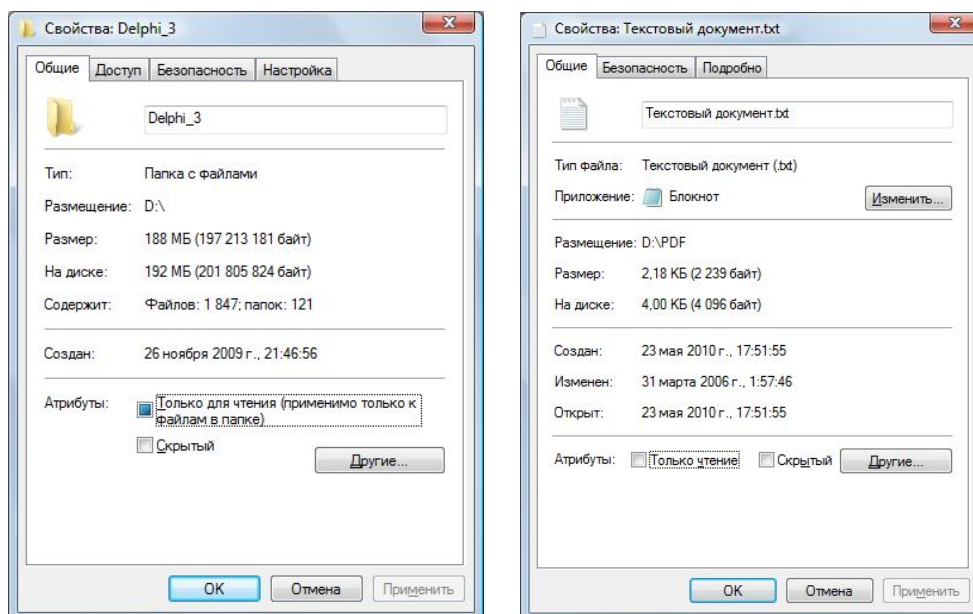


Рис.1.11 – Діалогове вікно властивостей об'єктів: папки та файлу

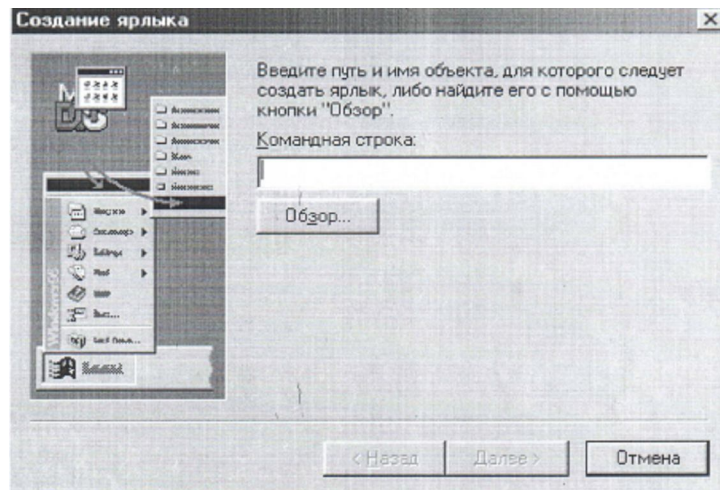
1.8 Створення ярлика

Використання ярлика спрощує доступ до найчастіше використовуваних документів і програм. Ярлик може бути створений для будь-якого об'єкту, зокрема папки, диска, іншого комп'ютера або принтера. Досить буде вказати

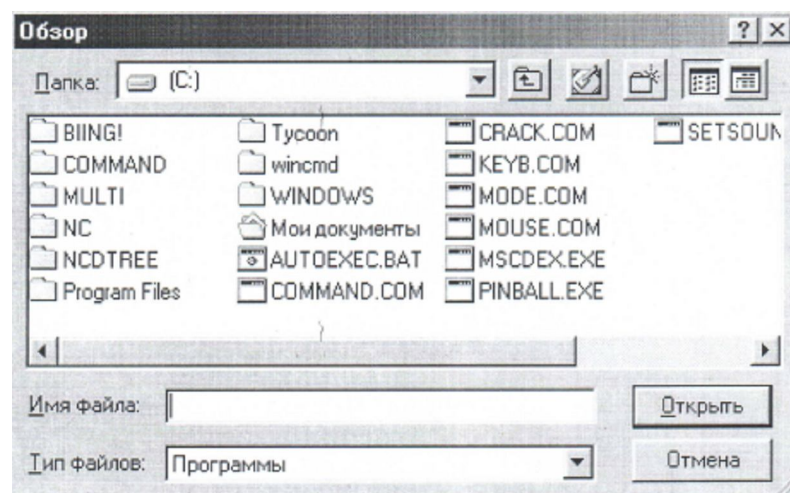
на ярлик і двічі клацнути на нім - і це дозволить відкрити застосування, що створило цей файл, або організувати доступ до папки або пристрою. Наявність ярлика не змінює місцеположення об'єкту, воно просто спрощує його відкриття, тому видалення ярлика не приводить до видалення файлу або програми.

Для створення ярлика необхідно виконати наступні дії:

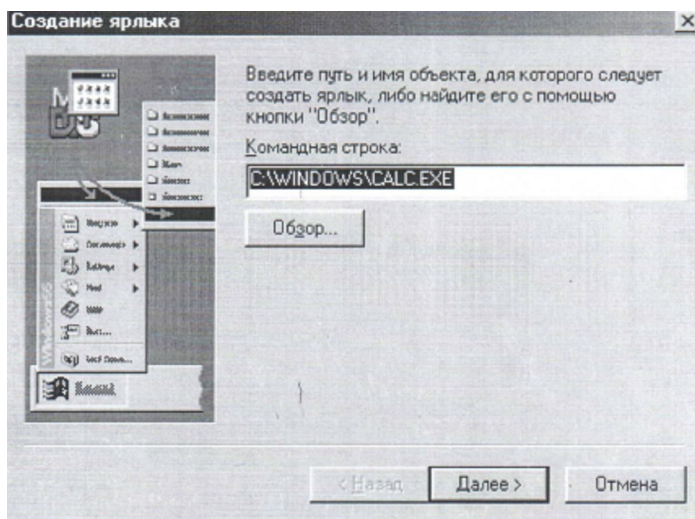
- на вільному місці робочого столу або папки натиснути праву кнопку миші і в контекстному меню вибрати команду **Создать/Ярлык** або вибрати команду **Файл/Создать/Ярлык** (у відкритому вікні папки). З'явиться діалогове вікно **Создание ярлика**



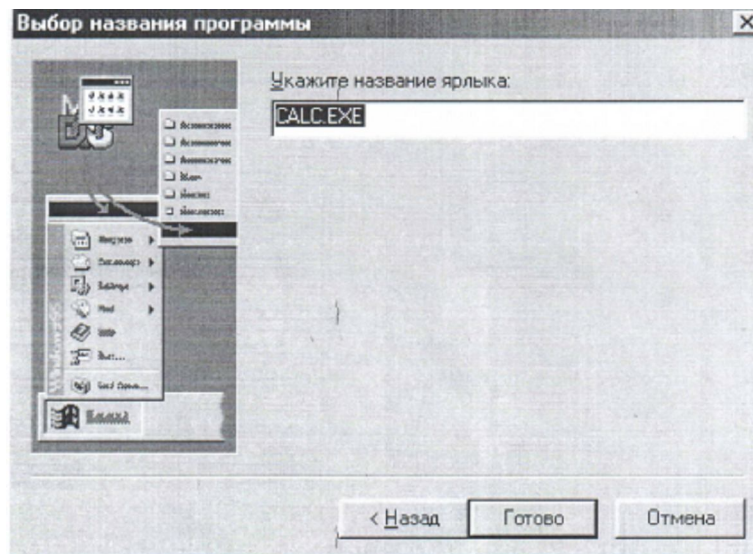
- для того, щоб вказати в текстовому полі **Командная строка** ім'я об'єкту і його місцеположення необхідно скористатися кнопкою **Обзор**. У вікні, що з'явиться, необхідно виділити ім'я потрібного файлу та натиснути кнопку **Открыть**.



В результаті відбудеться повернення в діалогове вікно **Создание ярлыка**, але вже із заповненим командним рядком.



- для перехода на наступний крок необхідно натиснути кнопку **Далее**.
У діалоговому вікні **Выбор названия программы**, що з'явиться, ввести назву ярлика, якщо Ви не згодні з тим ім'ям яке пропонує ОС Windows, та натиснути кнопку **Готово**.



Створити ярлик можна і іншим способом: виділити об'єкт, для якого необхідно створити ярлик і при натиснутої правої кнопки миші перетягнути цей об'єкт на робочий стіл або в іншу відкриту папку, потім відпустити кнопку миші і вибрати в меню команду **Создать ярлык(и)**.

2. ОСНОВИ АЛГОРИТМІЗАЦІЇ

2.1 Етапи підготовки і рішення задачі

Процедура підготовки і рішення задач з використанням комп'ютера - достатньо складний і трудомісткий процес, що складається з наступних етапів:

1. постановка завдання (завдання, яке належить вирішувати, формулюється користувачем або надходить у вигляді завдання);
2. математичне формулювання завдання;
3. розробка алгоритму рішення задачі;
4. написання програми на мові програмування;
5. підготовка початкових даних;
6. введення програми і початкових даних у комп'ютер;
7. відлагодження програми;
8. тестування програми;
9. рішення задачі і обробка результатів.

Оскільки умови більшості завдань дані в математичному формулюванні, тому необхідність у виконанні етапів 1 і 2 відпадає і можна відразу приступити до розробки алгоритму рішення задачі.

2.2 Алгоритм і форми його зображення

Під **алгоритмом** розуміється опис послідовності арифметичних і логічних дій над числовими значеннями змінних, що приводять до обчислення результату рішення задачі при зміні початкових даних в достатньо широких межах.

Таким чином, при розробці алгоритму рішення задачі математичне формулювання перетвориться в процедуру рішення, що є послідовністю арифметичних дій і логічних зв'язків між ними.

При цьому алгоритм володіє наступними властивостями: детермінірованістю, що означає, що застосування алгоритму до одних і тих

же початкових даних повинне приводити до одного і того ж результату; масовістю, що дозволяє отримувати результат при різних початкових даних; результативністю, що забезпечує отримання результату через кінцеве число кроків.

Основні форми представлення алгоритмів: словесна, операторна і графічна. Для запису алгоритмів при словесній формі використовуються засоби звичайної мови, але з ретельно підібраним набором слів і фраз мови. Операторна форма запису алгоритмів – це використання інструкцій мови програмування.

Найбільш наочним і зручним способом уявлення, є графічна форма запису алгоритму у вигляді блок-схеми. При цьому алгоритм представляється послідовністю блоків, що виконують певні функції (див. табл. 2.1), які об'єднуються між собою лініями потоку інформації. У середині блоків вказується інформація, що характеризує виконуваними ними функції. Блоки алгоритму мають крізну нумерацію.

На етапі 4 складається програма на одній з мов програмування. При описі програм необхідно використовувати характерні прийоми програмування і зважати на специфіку конкретної мови.




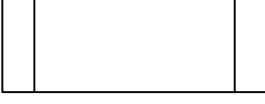
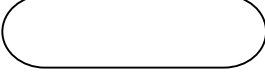
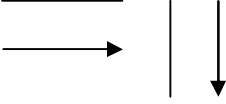

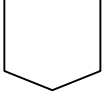
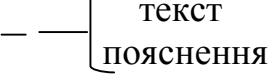
Етапи алгоритмізації і програмування є найбільш трудомісткими, тому їм приділяється велика увага.

Відлагодження програм полягає у виявленні і виправленні помилок, допущених на всіх етапах підготовки завдань до рішення на комп'ютері.

Синтаксичні помилки виявляються транслятором, який видає повідомлення, вказуючи місце і тип помилки. Виявлення семантичних помилок здійснюється на етапі тестування програми, в якому перевіряється правильність виконання програми на спрощеному варіанті початкових даних або за допомогою контрольного розрахунку.

Обробка результатів рішення задачі здійснюється або за допомогою ЕОМ (побудова таблиць, графіків), або уручну. Результати, що виводяться, оформляються у вигляді, зручному для сприйняття людиною.

Таблиця 2.1 - Графічні символи, які вживаються при складанні блок-схем

№ п/п	Найменування	Позначення	Функція
1	Введення, виведення		Введення, виведення даних
2	Документ		Друк результатів на папір
3	Процес		Обчислення арифметичних операцій
4	Рішення		Перевірка умов
5	Модифікація		Початок циклу
6	Зумовлений процес		Обчислення по підпрограмах
7	Початок, кінець		Початок, кінець процесу обробки даних
8	Лінії потоку		Зображення зв'язків між блоками
9	З'єднувач		Вказівка зв'язку між перерваними лініями потоку
10	Міжсторінковий з'єднувач		Вказівка зв'язку між частинами блок-схеми, розташованих на різних листах
11	Коментар		Запис пояснення до блоку або до лінії потоку

2.3 Типові структури алгоритмів

По характеру зв'язків між блоками, тобто по структурі, алгоритми діляться на лінійні, розгалужені і циклічні.

2.3.1 Лінійні і розгалужені алгоритми

Простим прикладом алгоритму є алгоритм лінійної структури.

Алгоритм лінійної структури – це алгоритм, в якому блоки виконуються послідовно один за одним, в порядку, заданому схемою. Такий порядок виконання називається **природним**.

Приклад 1.

Обчислити висоти трикутника із сторонами a , b , c , використовуючи формули:

$$h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}$$

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)}$$

$$h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)}$$

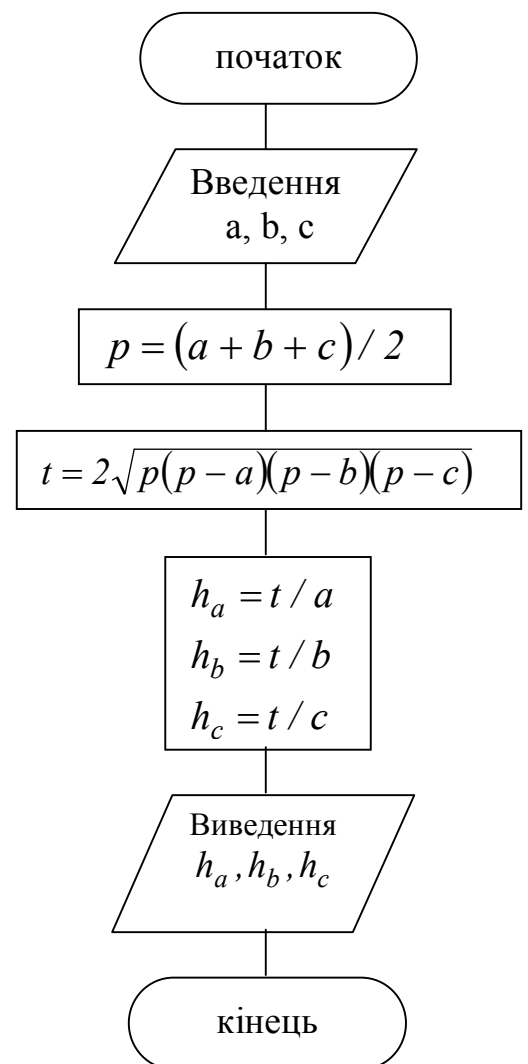
$$\text{де } p = (a + b + c) / 2$$

При рішенні даної задачі для виключення повторень слід обчислювати висоти не за приведеними вище формулами безпосередньо, а використовуючи проміжну змінну:

$$t = 2\sqrt{p(p-a)(p-b)(p-c)},$$

$$\text{тоді } h_a = t/a, \quad h_b = t/b, \quad h_c = t/c$$

Схема алгоритму рішення задачі має вигляд:



Приклад 2.

Обчислити значення z використовуючи формулу:

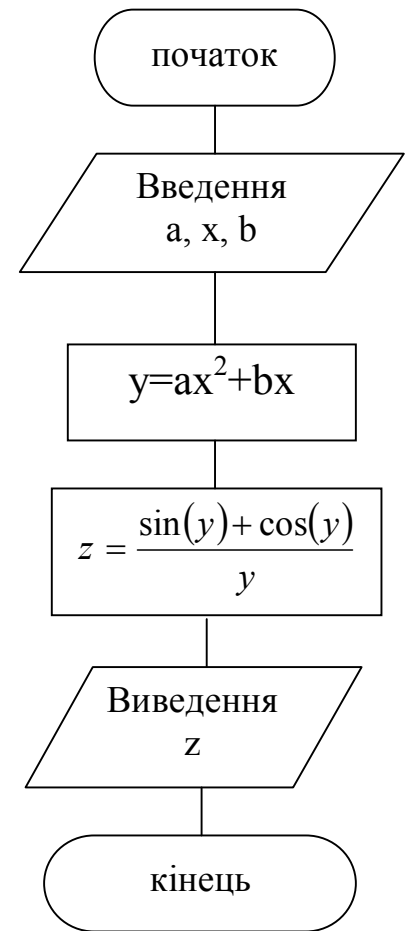
$$z = \frac{\sin(ax^2 + bx) + \cos(ax^2 + bx)}{ax^2 + bx}$$

Початкові дані: $a=2$; $x=0.5$; $b=1.3$.

При рішенні даної задачі для виключення повторень слід використовувати проміжну змінну:

$$y = ax^2 + bx$$
$$\text{modi } z = \frac{\sin(y) + \cos(y)}{y}$$

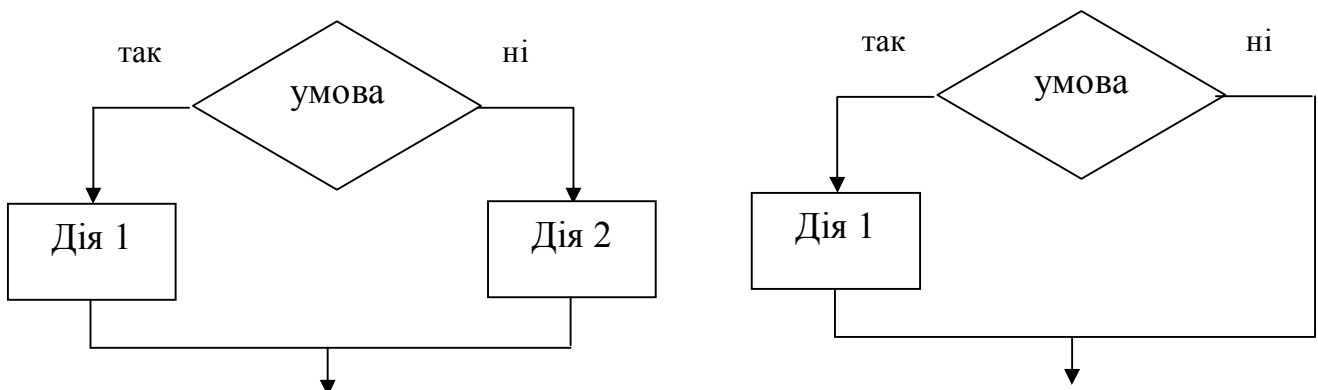
Схема алгоритму рішення задачі має вигляд:



Проте рішення абсолютної більшості інженерних задач неможливо звести до алгоритмів лінійної структури. Часто залежно від яких-небудь проміжних результатів обчислення здійснюється або по одних, або по інших формулах, тобто залежно від виконання деякої логічної умови обчислювальний процес здійснюється по одній або по іншій гілці.

Алгоритм такого обчислювального процесу називається **алгоритмом розгалуженої структури**.

У блок-схемах розгалужений обчислювальний процес зображується наступним чином:

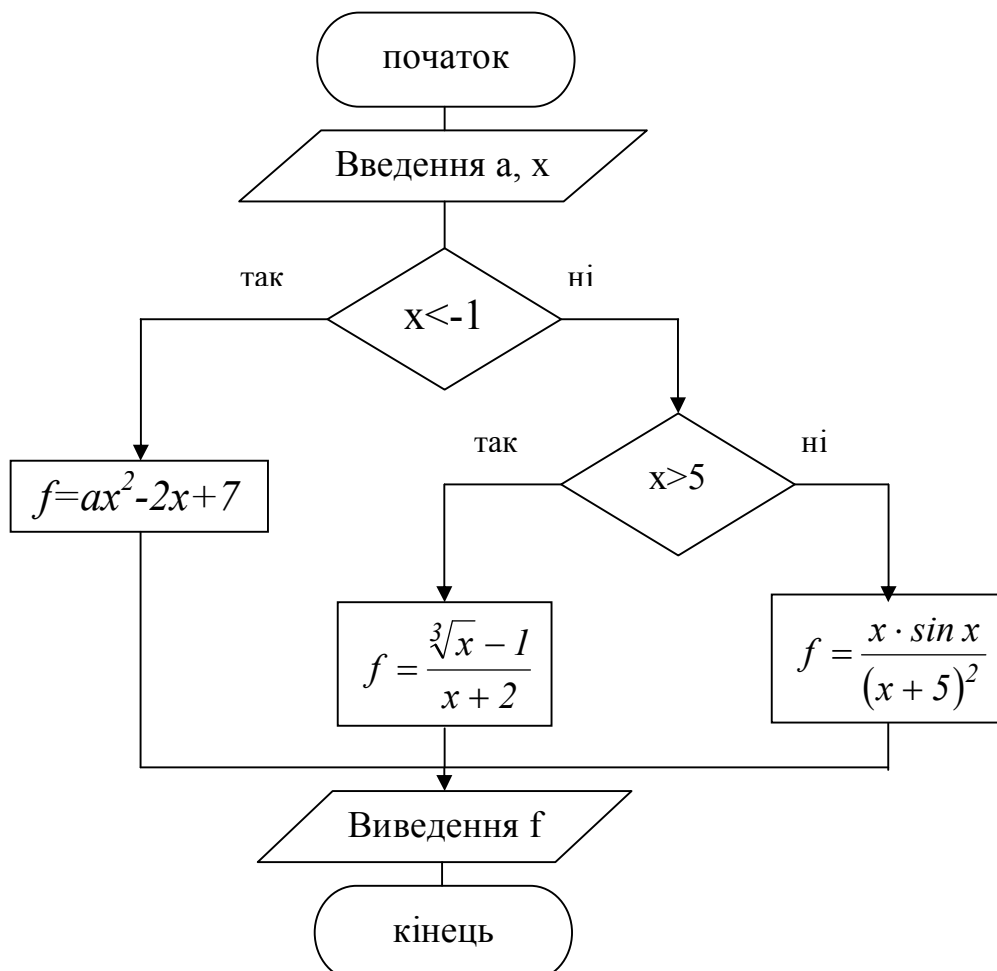


Приклад 3.

Обчислити значення f для початкових значень $a=2.8$; $x=1.5$:

$$f = \begin{cases} ax^2 - 2x + 7, & \text{якщо } x < -1 \\ \frac{x \cdot \sin x}{(x+5)^2}, & \text{якщо } -1 \leq x \leq 5 \\ \frac{\sqrt[3]{x} - 1}{x+2}, & \text{якщо } x > 5 \end{cases}$$

З математичного формулювання задачі можна побачити, що обчислювальний процес має три гілки. За допомогою умовного блоку перевіряється виконання тільки однієї умови, по якій вибирається вираз для реалізації однієї гілки. Тому для визначення того, по якій з двох гілок, що залишилися, повинен йти обчислювальний процес після невиконання першої умови, необхідно використовувати ще один умовний блок. Після обчислення по будь-якій з формул здійснюється перехід в загальну гілку до блоку виведення результату. Схема алгоритму рішення задачі має вигляд:



Приклад 4.

Обчислити значення z використовуючи формулу:

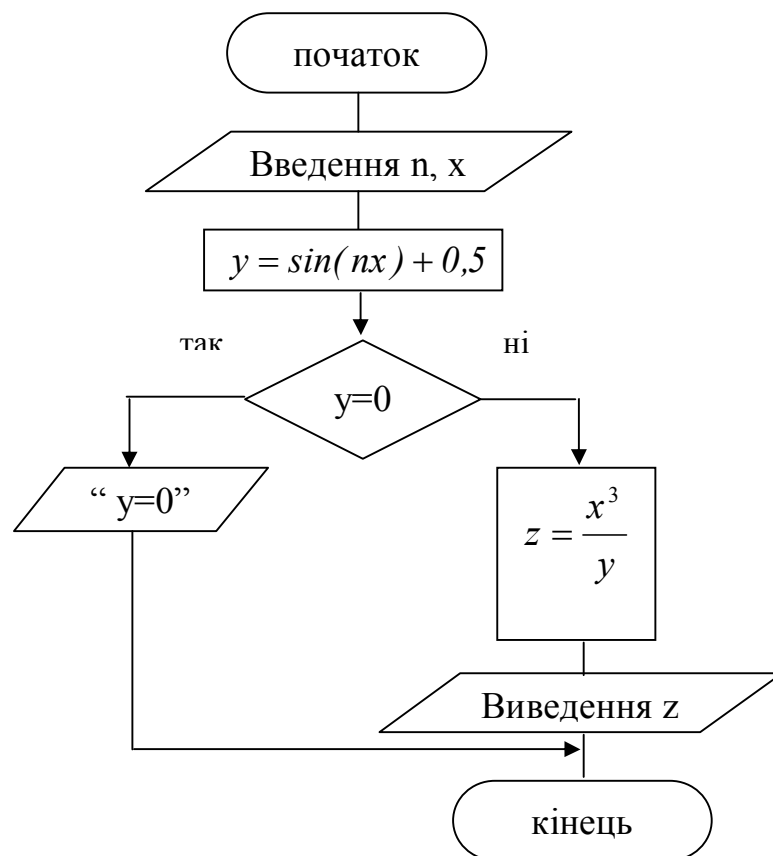
$$z = \frac{x^3}{y}, \quad \text{де } y = \sin(nx) + 0,5.$$

Здавалося б, що рішення цієї задачі описує алгоритм лінійної структури.

Проте для задоволення властивості масовості і результативності алгоритму необхідно, щоб при будь-яких початкових даних був отриманий результат або повідомлення про те, що задача не може бути вирішена при заданих даних.

Дійсно, якщо $y=0$, то задача не може бути вирішена, оскільки ділити на нуль не можна. Тому в алгоритмі необхідно передбачити таку умову і видати як результат інформацію про те, що $y=0$.

Схема алгоритму рішення задачі має вигляд:



Приклад 5.

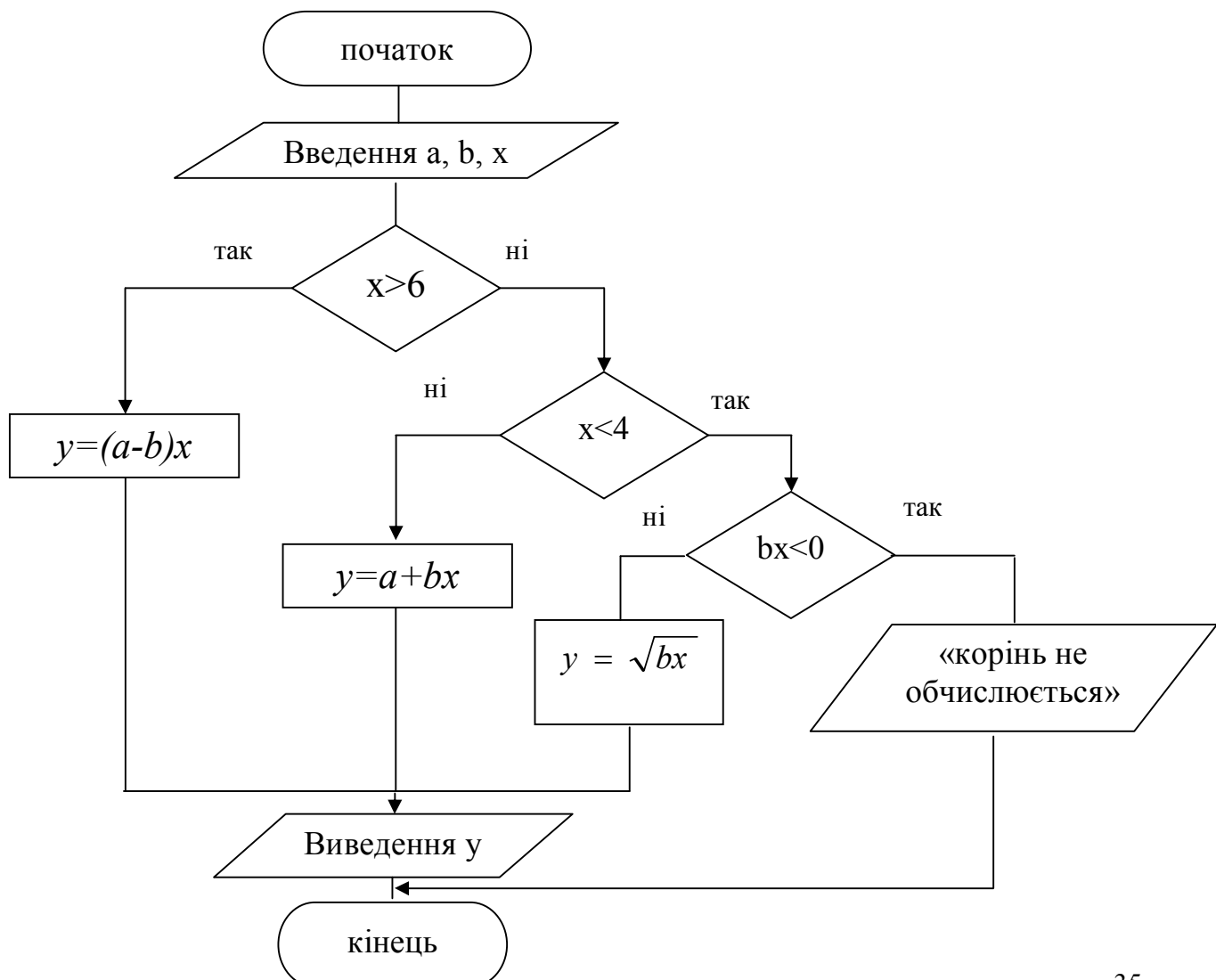
Обчислити значення y :

$$y = \begin{cases} \sqrt{bx}, & x < 4 \\ a + bx, & 4 \leq x \leq 6 \\ (a - b)x, & x > 6 \end{cases}$$

Початкові данні: a, b, x .

Для задоволення властивості масовості і результативності алгоритму необхідно, щоб при будь-яких початкових даних був отриманий результат або повідомлення про те, що завдання не може бути вирішене при заданих даних. Тому в алгоритмі необхідно при перевірці умови $x < 4$, ще перевірити умову $bx < 0$ і видати як результат інформацію про те, що корінь з від'ємного числа не обчислюється.

Схема алгоритму рішення задачі має вигляд:



2.3.2 Алгоритми циклічної структури

Часто при рішенні задач виникає необхідність багато разів обчислювати значення по одних і тих же математичних залежностях для різних вхідних величин. Багато разів повторюваний етап обчислювального процесу називається **циклом**, а обчислювальний процес, що містить такі етапи, – **циклічним**.

Для організації циклічних обчислень необхідно виконати наступні дії:

- задати перед циклом початкове значення змінної, яка буде змінюватися у циклі;
- змінювати змінну перед кожним новим повторенням циклу;
- перевіряти умову закінчення або повторення циклу, тобто переходити до його початку, якщо він не закінчений, або виходити з нього після закінчення.

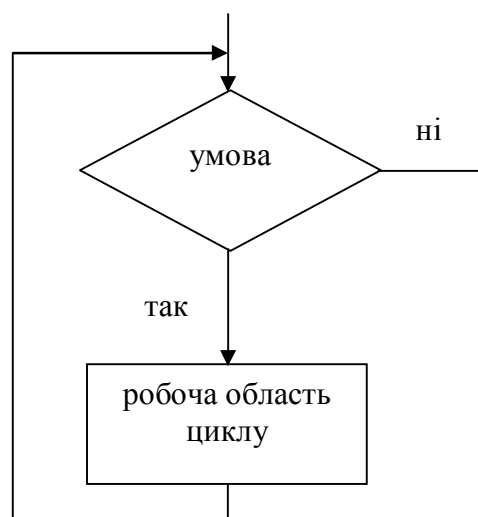
Змінна, що змінюється у циклі і від якої залежить умова виходу з циклу, називається **параметром циклу**.

Можливі три способи організації циклічних структур алгоритмів:

- 1) цикл з передумовою або цикл «ПОКИ»;
- 2) цикл з постумовою або цикл «ПОВТОРЮВАТИ ... ДО»;
- 3) цикл з відомим числом повторень.

Цикл з передумовою використовується, як правило, в тих випадках, коли наперед невідоме число повторень.

Структура циклу з передумовою:



Якщо умова задовольняється виконується робоча область циклу, потім знову перевіряється умова. Може трапитися, що робоча область циклу не буде виконуватись жодного разу, якщо умова не задовольняється із самого початку.

Приклад 6.

Обчислити значення y використовуючи формулу:

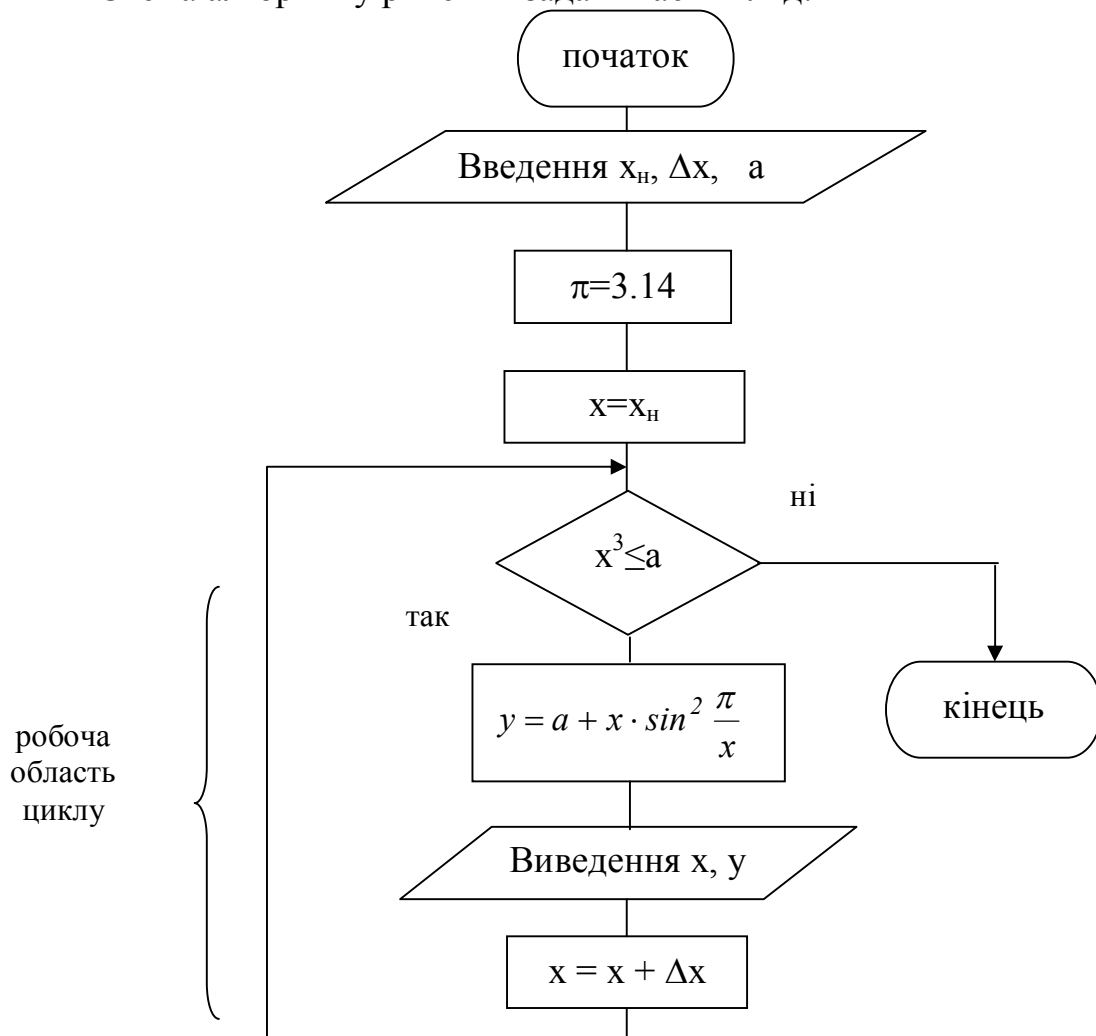
$$y = a + x \cdot \sin^2 \frac{\pi}{x}, \quad \text{для } x \geq 1 \quad \Delta x = 0.8$$

Обчислювати y до тих пір, поки значення x^3 стане більше a .

Початкові дані: $x_H=1$; $\Delta x=0.8$; a .

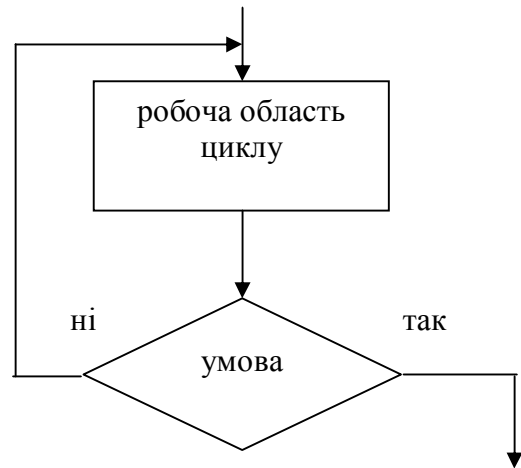
Перед першим виконанням циклу необхідно задати початкове значення змінної x , рівне 1. Щоб обчислювальний процес здійснився, необхідно задати умову повторення циклу, а потім виконувати обчислення і виведення значень x і y . При кожному новому виконанні циклу необхідно змінювати параметр циклу x на величину кроку, рівного 0.8.

Схема алгоритму рішення задачі має вигляд:



Цикл з постумовою використовується також, в тих випадках, коли наперед невідоме число повторень.

Структура циклу з постумовою:



У даному циклі виконується робоча область циклу, а потім перевіряється істинність умови. Якщо умова не задовольняється, то знову виконується робоча область циклу. Якщо ж умова задовольняється, то виконання робочої області циклу закінчується.

При такій організації циклу робоча область виконуватиметься принаймні один раз.

Приклад 7.

Обчислити значення t використовуючи формулу:

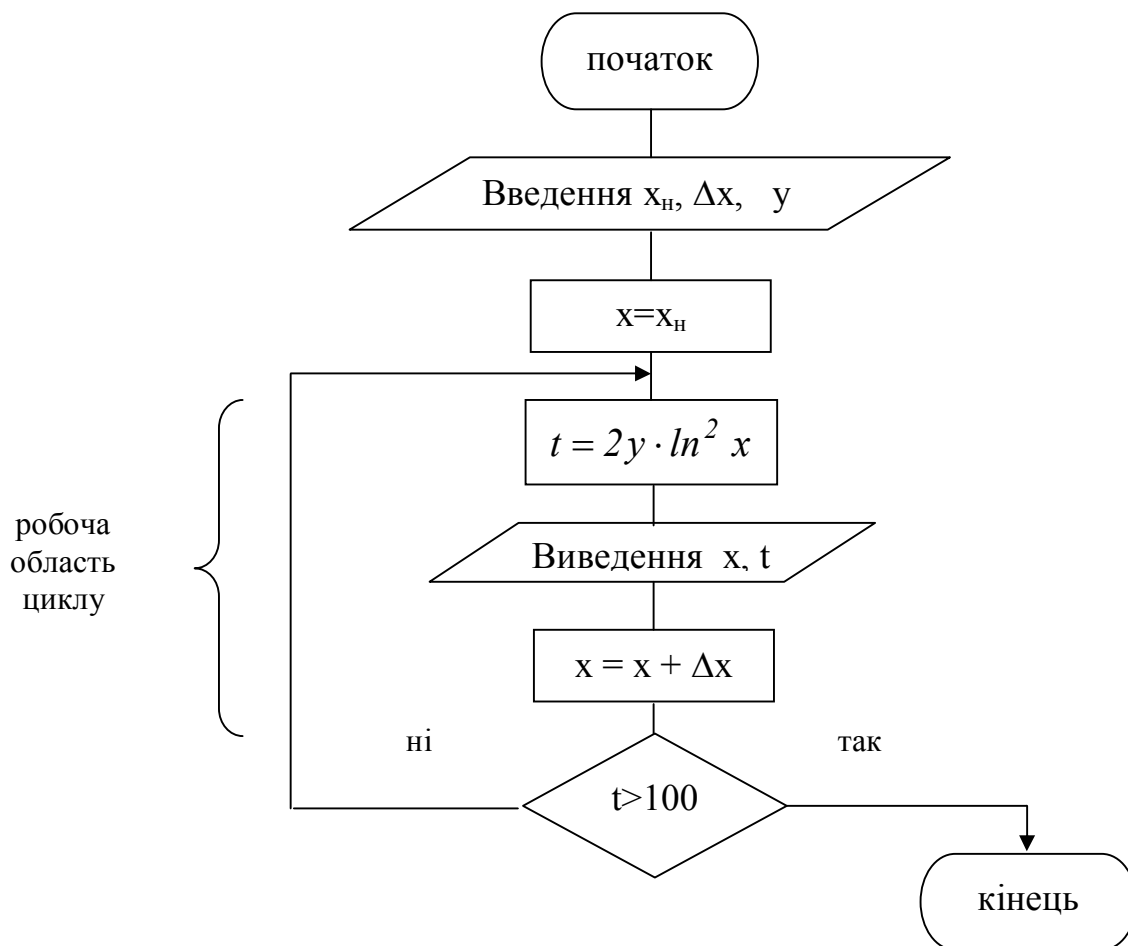
$$t = 2y \cdot \ln^2 x, \quad \text{для } x \geq 0.5 \quad \Delta x = 0.2$$

Рахувати t до тих пір, поки його значення стане більше 100.

Початкові дані: $x_n=0.5$; $\Delta x=0.2$; y .

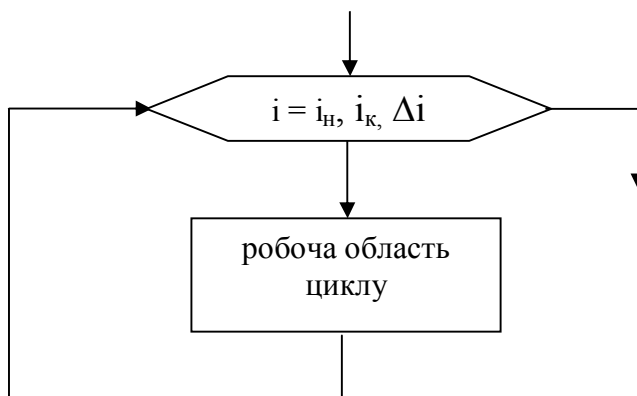
Перед першим виконанням циклу необхідно задати початкове значення змінної x , рівне 0.5, а потім виконувати обчислення і виведення значень x і t . При кожному новому виконанні циклу необхідно змінювати параметр циклу x на величину кроку, рівного 0.2. Щоб обчислювальний процес не був нескінченним, необхідно задати умову закінчення циклу.

Схема алгоритму рішення задачі має вигляд:



Для організації **циклу з відомим числом повторень** використовується блок модифікації. У цьому блоці об'єднано 3 дії: установка початкового значення параметра циклу, його зміна в процесі виконання циклу і перевірка умови закінчення циклу.

Структура циклу з відомим числом повторень:



Параметром циклу є змінна i , яка змінюється від свого початкового значення i_n до кінцевого значення i_k з кроком Δi . Якщо $\Delta i = 1$, то в блоці модифікації значення кроку не вказується.

Робоча область циклу виконується для кожного значення параметра циклу i від його початкового значення i_n до кінцевого значення i_k включно.

Приклад 8.

Обчислити значення y використовуючи формулу:

$$y = \frac{\ln ax + 2}{\sqrt[3]{a + x}},$$

$$\text{для } 2 \leq x \leq 3 \quad \Delta x = 0.2 \quad a = 1.67$$

Початкові дані: $x_n=2$; $x_k=3$; $\Delta x=0.2$; $a=1.67$.

Для організації циклу з відомим числом повторень необхідно обчислити кількість повторень циклу за формулою:

$$m = \left\lceil \frac{x_k - x_n}{\Delta x} \right\rceil + 1,$$

де x_k и x_n - кінцеве і початкове значення змінної x , Δx - крок зміни змінної x , дужки $\lceil \rceil$ означають, що береться тільки ціла частина від ділення.

Перед першим виконанням циклу необхідно задати початкове значення змінної x , рівне 2, а потім 6 разів (кількість повторень циклу) виконувати обчислення і виведення значень x і y . При кожному новому виконанні циклу необхідно змінювати значення змінної x на величину кроку, рівного 0,2.

Параметром циклу є змінна i , яка змінюється від 1 до m , де m – число повторень циклу, тому що змінна i змінюється у циклі і від неї залежить умова виходу з циклу.

Схема алгоритму рішення задачі зображена на рис.2.1.

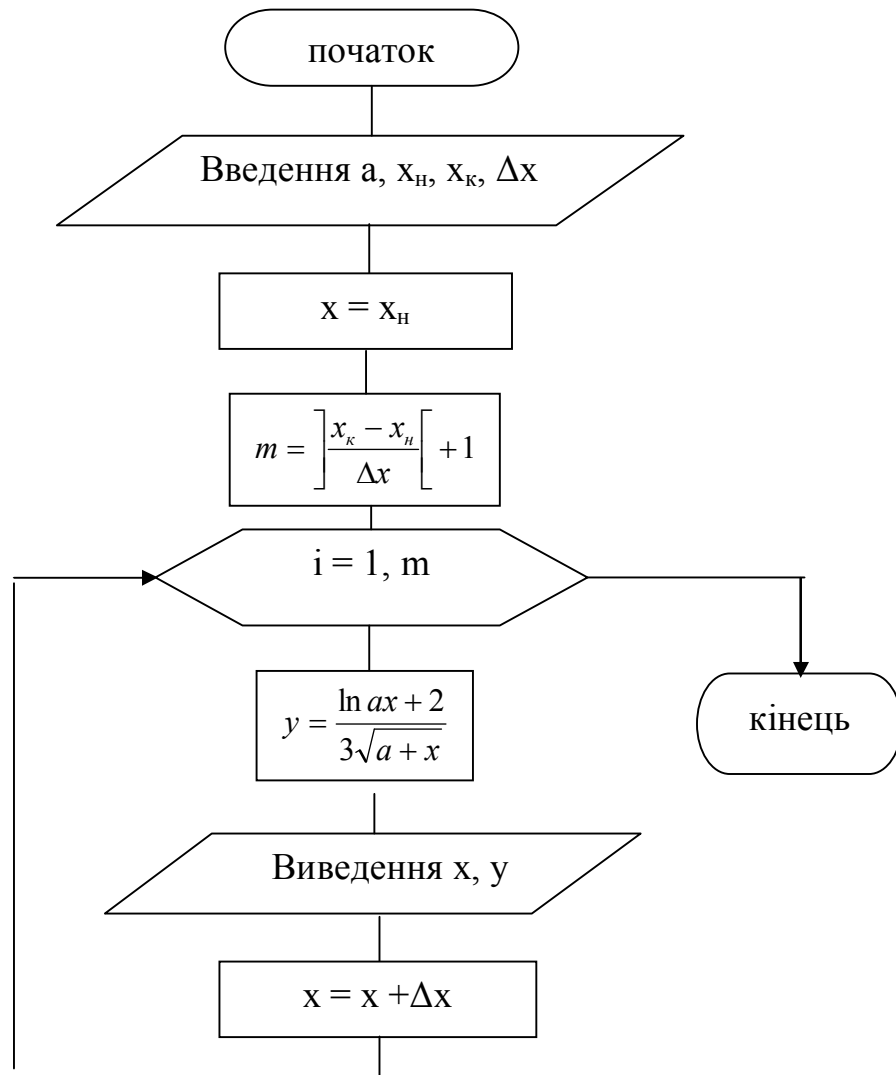


Рис.2.1 - Організація алгоритму циклу з відомим числом повторень.

2.4 Алгоритми із структурою вкладених циклів

На практиці часто зустрічаються завдання, в яких одночасно змінюється декілька параметрів.

В цьому випадку з'являються структури з вкладеними циклами. У середині простого циклу може знаходитися ще один або декілька циклів. При цьому перший цикл називається зовнішнім, а вкладені в нього – внутрішніми.

Правила їх організації нічим не відрізняються від правил організації простого циклу, причому один і той же цикл може бути зовнішнім по відношенню до одного і внутрішнім по відношенню до іншого циклу.

Зовнішній і внутрішній цикл мають свої параметри.

Для кожного значення параметра зовнішнього циклу параметр внутрішнього циклу приймає послідовно всі свої значення. Іншими словами, завжди виконується в першу чергу самий внутрішній цикл.

Межі внутрішнього циклу не можуть виходити за межі зовнішнього по відношенню до нього циклу, але можуть співпадати з ними.

Приклад 9.

Обчислити значення x та y використовуючи формули та вивести результати у формі таблиці:

$$x = \frac{2,5a - c}{a^2 - 2ac} \quad y = \frac{|x + 2c|}{c^2 + 1}$$

$$-3 \leq a \leq 3; \quad \Delta a = 1,5;$$

$$-1 \leq c \leq 1; \quad \Delta c = 0,2$$

Початкові дані: $a_{\text{н}}, a_{\text{к}}, \Delta a, c_{\text{н}}, c_{\text{к}}, \Delta c$.

Для організації зовнішнього і внутрішнього циклів використовуватиме цикли з постумовою.

Таблиця результатів повинна мати наступний вигляд:

$c \backslash a$	-3	-1,5	0	1,5	3
-1	2,085	0,835			
-0,8	1,95				
-1,8					
...					
1					

У випадку коли $a^2 - 2ac = 0$ в таблицю результатів виводитимемо текстову константу «***» замість повідомлення “Рішень не має”.

Схема алгоритму рішення задачі зображена на рис.2.2.

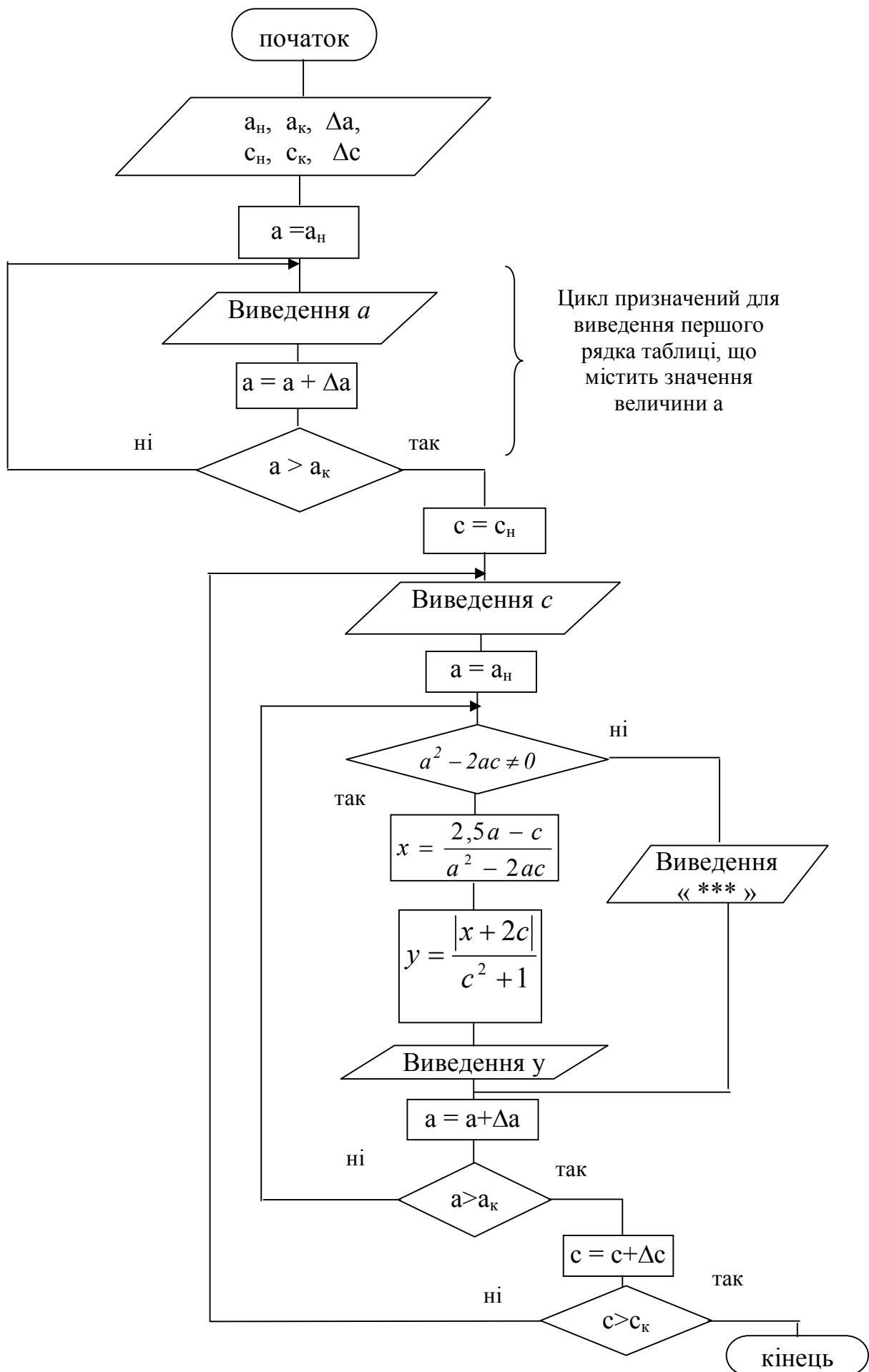


Рис.2.2 - Організація алгоритму із структурою вкладених циклів.

2.5 Алгоритми визначення суми, кількості і добутку

Часто разом з обчисленням значень величини або сукупності величин необхідно визначити суму, кількість або добуток значень.

При обчисленні кількості, суми або добутку доцільно використовувати принцип поступового накопичення.

Формула, яка використовується для накопичення суми, має вигляд: $z_i = z_{i-1} + y_i$. При першому виконанні циклу обчислюється значення $z_1 = z_0 + y_1$, яке повинне бути рівне y_1 . Тому початковому значенню суми перед циклом слід привласнити значення нуль.

Аналогічно накопичується і добуток, з тією лише різницею, що для його накопичення використовується формула $z_i = z_{i-1} \cdot y_i$, а початкове значення добутку повинне бути рівне одиниці.

Для накопичення кількості використовується формула: $k_i = k_{i-1} + 1$. При першому виконанні циклу обчислюється значення $k_1 = k_0 + 1$, яке повинне бути рівне 1. Тому початковому значенню кількості перед циклом слід привласнити значення нуль.

Приклад 10.

Обчислити значення у використовуючи формулу:

$$y = \frac{a^2 + x^2}{\ln 3 - |x|} \quad -0.5 \leq x \leq 2.5 \quad \Delta x = 0.1 \quad a = 3.8$$

Визначити: кількість (k) $y < 0$, $P = \prod_{y < 0} y$, $S = \sum y$.

Початкові дані: x_n , x_k , Δx , а.

Схема алгоритму рішення задачі зображена на рис.2.3.

Перед організацією циклу задаються початкові значення для величин k, S, P. У циклі відбувається накопичення добутку від'ємних значень у і підрахунок їх кількості, а також накопичення суми всіх значень у. Виведення значень k, S, P здійснюється після закінчення циклу, тобто значення необхідно вивести лише один раз, коли вони будуть остаточно обчислені.

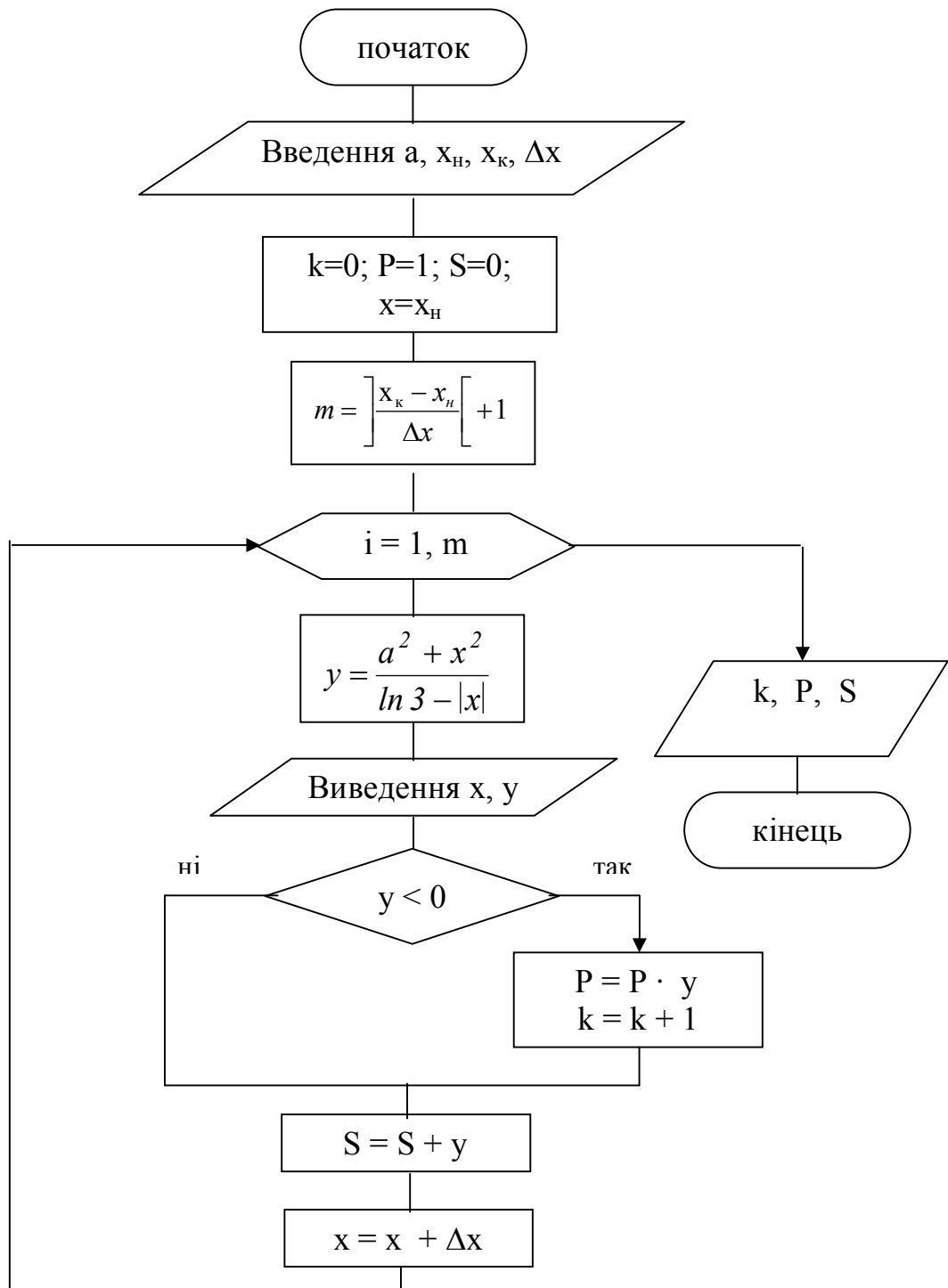


Рис. 2.3 - Організація алгоритму визначення суми, кількості і добутку.

2.6 Алгоритми ітераційних циклів

До циклів з невідомим числом повторень відносяться **ітераційні цикли**, які характеризуються послідовним наближенням до шуканого значення із заданою точністю. Кількість повторень визначається в процесі обчислень і залежить від того, наскільки вдале початкове наближення шуканого результату, з якою точністю необхідно знайти результат, наскільки вдало підібраний набір операцій для того, щоб отримати наступне наближення до рішення, виходячи з наявного наближення.

Ітераційним називається такий обчислювальний процес, в якому для визначення подальшого значення змінної використовується її попереднє значення. Вихід з циклу організовується по умові досягнення заданої точності.

Ітераційні цикли використовуються при конструюванні алгоритмів обчислення значення функції, заданої розкладанням в ряд, з наперед заданою точністю; алгоритмів обчислення суми членів нескінченного ряду; алгоритмів знаходження наближеного коріння рівнянь методами: ділення відрізка навпіл, хорд, ітерацій і т.д.

Приклад 11.

Обчислити значення $\sin(x)$ з точністю ε , використовуючи розкладання функції $\sin(x)$ в ряд по ступенях x :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{x^{2n-1}}{(2n-1)!}$$

Визначити кількість (k) ітерацій.

Початкові дані: x ; $\varepsilon=0.01$.

Це знакозмінний ряд. Обчислення слід продовжувати, накопичуючи суму членів ряду, до тих пір, поки буде отриманий черговий член ряду, по абсолютній величині, який не перевершує заданої точності обчислення суми ряду ε .

Якщо доданок позначимо через y , то умова виходу з циклу буде мати вигляд: $|y_{n+1}| < \varepsilon$.

Для обчислення суми n членів ряду буде використовуватися формула:

$$S_{n+1} = S_n + y_{n+1},$$

де S_n – сума n членів ряду, $n = 0, 1, 2, \dots$

Проаналізувавши початковий ряд, отримуємо початкові дані:

$$S_0 = 0 \quad y_1 = x \quad y_n = \frac{x^{2n-1}}{(2n-1)!} \cdot (-1)^{n+1}$$

Перевіримо, наприклад, для $n=1$ і $n=2$:

$$n = 1 \quad y = \frac{x}{1!} \cdot (-1)^2$$

$$n = 2 \quad y = \frac{x^3}{3!} \cdot (-1)^3$$

Члени нескінченного ряду містять факторіал – це є добуток n чисел натурального ряду: $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$.

Наприклад, якщо $n = 3$, то $3! = 1 * 2 * 3 = 6$.

Схема алгоритму рішення задачі зображена на рис.2.4.

2.7 Алгоритми формування і обробки одномірних масивів

У математиці, економіці, інформатиці часто використовуються впорядковані набори даних, наприклад послідовності чисел, таблиці, списки прізвищ. Для обробки даних одного типу вводиться поняття масиву.

Масив – це впорядкована сукупність змінних одного типу, позначених одним ім'ям.

Наприклад, $X=(1.6; 14.9; -0.5; 8.5; 0.46)$.

Кожен елемент масиву позначається ім'ям масиву з індексом. Індекс визначає місцеположення елемента у масиві, наприклад X_1 – це перший елемент масиву X , рівний 1.6.

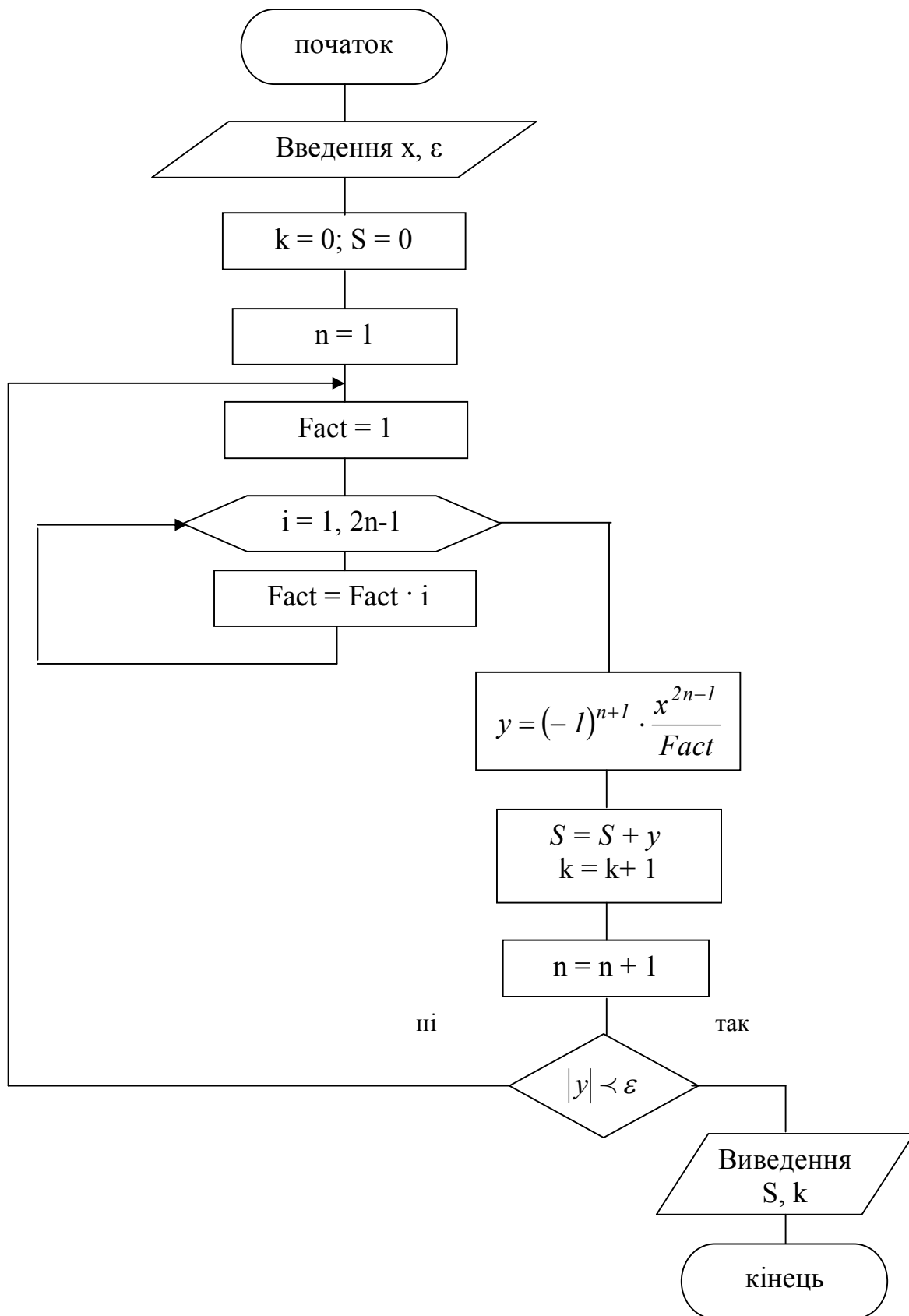


Рис. 2.4 - Організація ітераційного циклу.

Елементи масиву впорядковані по значенню індексу: $X=(X_1, X_2, \dots, X_n)$, тобто індекс елементів масиву змінюється від 1 до n . Змінна n указує на кількість елементів в масиві X , тобто на розмірність масиву.

Якщо кожен елемент масиву містить тільки один індекс, то такий масив називається **одномірним**.

Приклад 12.

Обчислити значення елементів масиву Y використовуючи формулу:

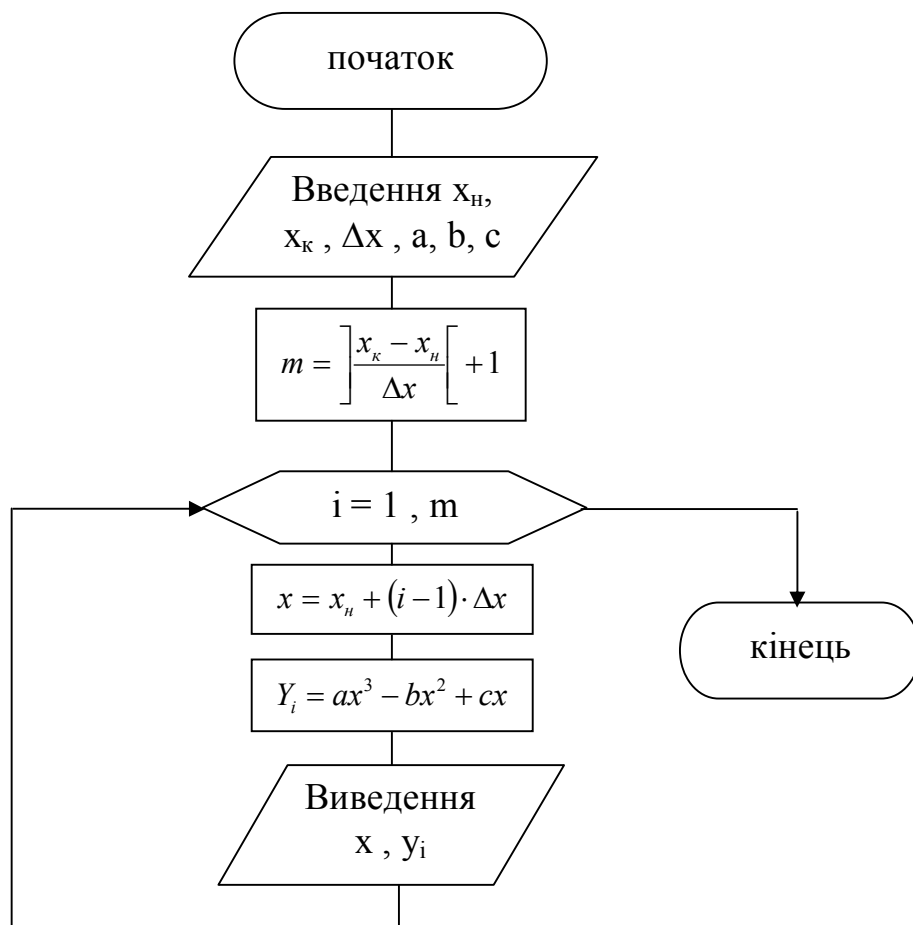
$$y_i = ax^3 - bx^2 + cx$$

$$0 \leq x \leq 10 \quad \Delta x = 2.5$$

$$a = 6.3 \quad b = 2.9 \quad c = -4.5$$

Початкові дані: $x_n, x_k, \Delta x, a, b, c$.

Схема алгоритму рішення задачі має вигляд:



У даному завданні значення y_i є елементами масиву (y_1, y_2, \dots, y_m) , де m – кількість елементів в масиві Y , яка обчислена за формулою

$$m = \left\lceil \frac{x_k - x_n}{\Delta x} \right\rceil + 1.$$

У циклі, який виконуватиметься m разів, змінюються два параметри: проста змінна x від початкового значення $x_n=0$ до кінцевого значення $x_k=10$ з кроком $\Delta x=2.5$ за формулою $x = x_n + (i-1) \cdot \Delta x$ і індекс i змінної y_i у блоці модифікації від 1 до m з кроком 1.

Приклад 13.

Є масив $X = (X_1, X_2, \dots, X_{10})$.

Обчислити значення елементів масивів t, z використовуючи формули:

$$t_i = x_i^3 + \frac{\pi \cdot x_i}{2}$$

$$z_i = \frac{\pi}{4} \cdot \cos t_i \cdot \sin(2 + 0.5 t_i)$$

Початкові дані: масив X .

Як його ввести?

Для введення елементів масиву необхідно спочатку ввести розмірність масиву, а потім організувати цикл з відомим числом повторень для зміни індексу елементу масиву.

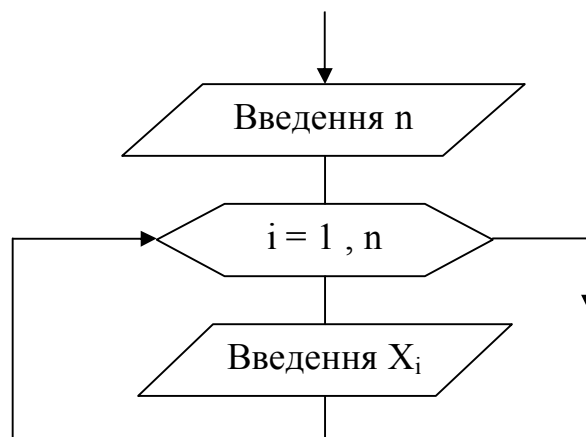
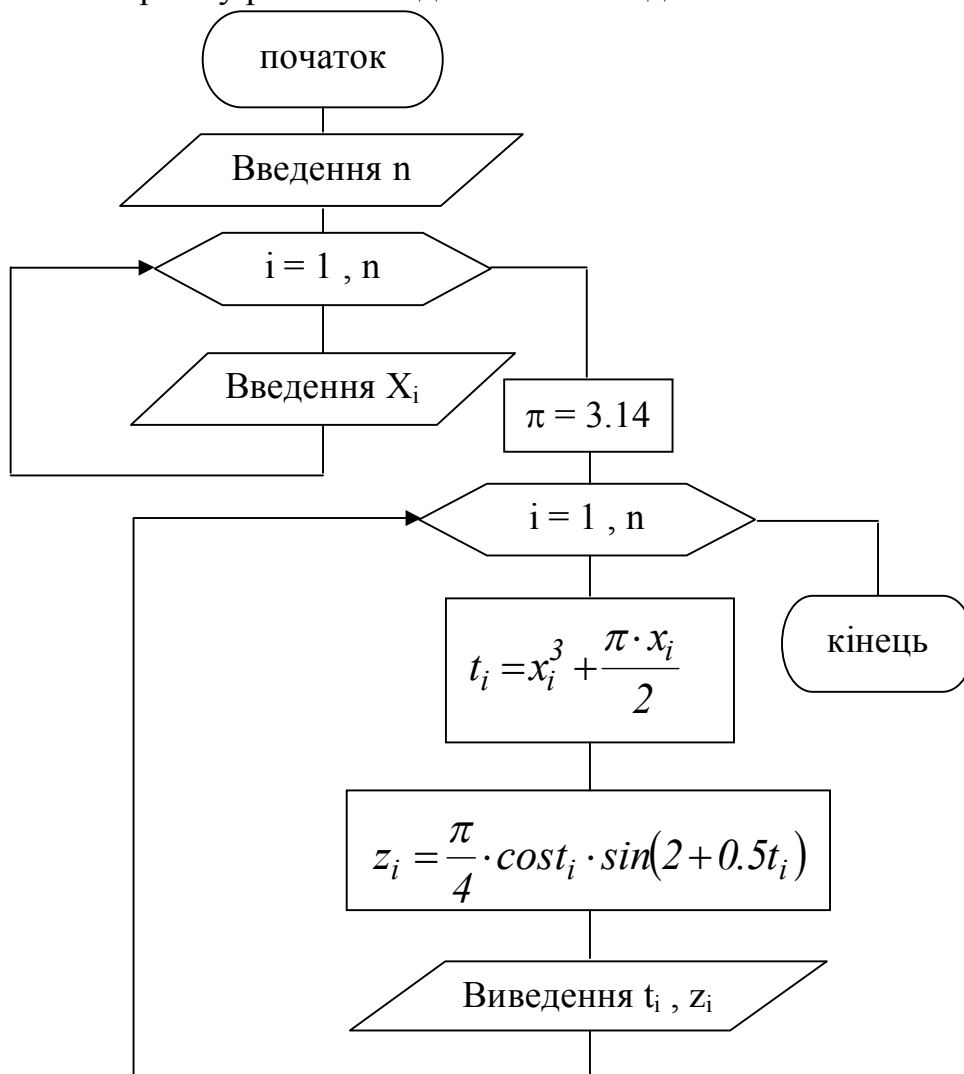


Схема алгоритму рішення задачі має вигляд:



Приклад 14.

Є масив $X(M)$. Визначити кількість ненульових елементів в масиві Y , елементи якого обчислюються за формулами:

$$y_i = \begin{cases} a \ln x_i, & \text{якщо } x_i \geq 1 \\ \sqrt{x_i^2 + 2,5}, & \text{якщо } x_i < 1 \end{cases} \quad \text{де } i=1, M$$

Схема алгоритму рішення задачі зображена на рис.2.5.

Приклад 15.

Записати елементи масиву $X = (X_1, X_2, \dots, X_{15})$, які менш за 1 в масив $Y = (Y_1, Y_2, \dots, Y_k)$. Визначити кількість і суму елементів масиву Y .

Схема алгоритму рішення задачі зображена на рис.2.6.

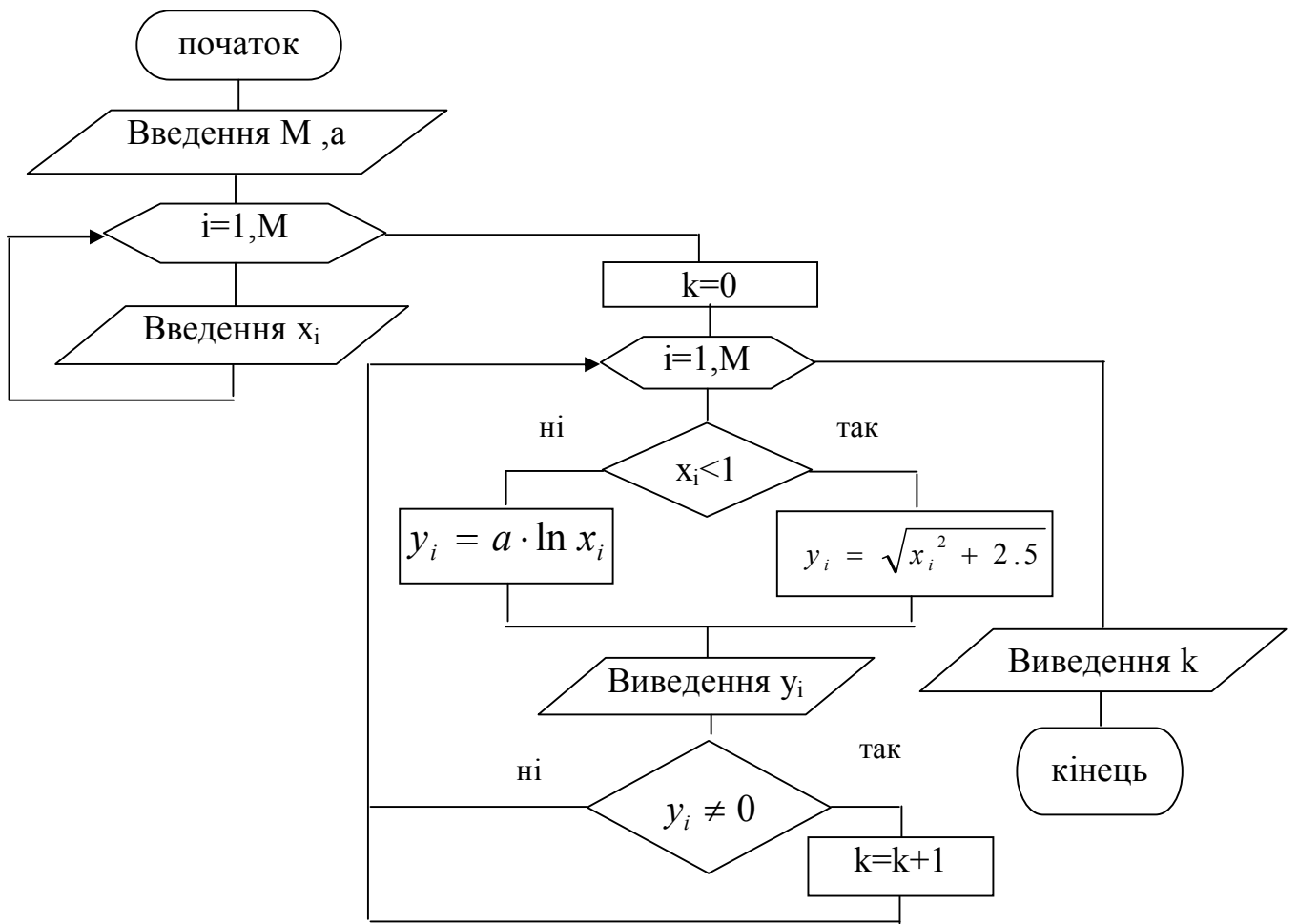


Рис. 2.5 - Алгоритм обчислення елементів масиву Y.

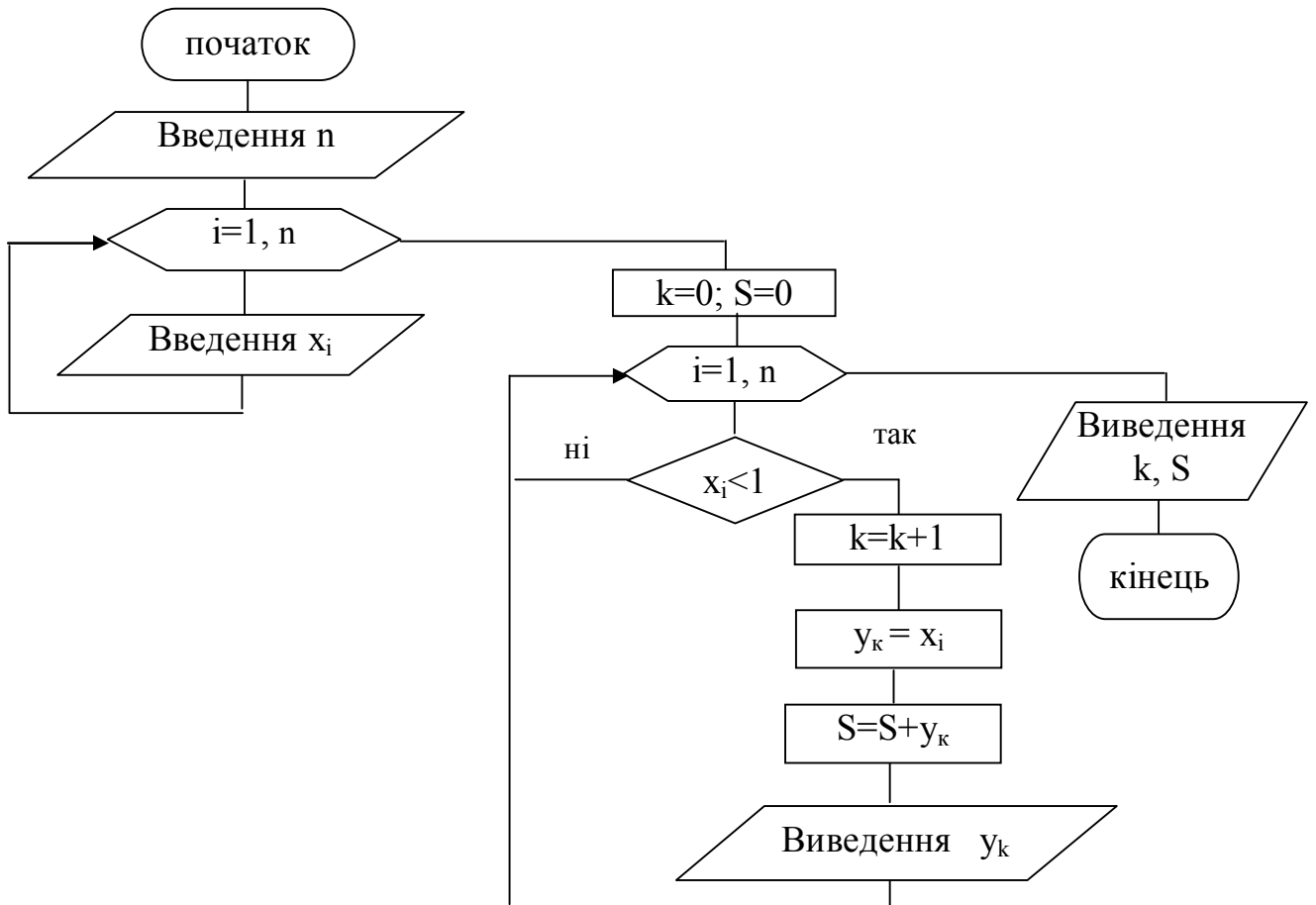


Рис.2.6 - Алгоритм формування елементів масиву Y.

Приклад 16.

У масиві $A(N)$ поміняти два елементи місцями: A_k та A_s .

Якщо елементу A_k привласнимо значення елемента A_s , то попереднє значення елемента A_k зітреться і не зможе бути переписане на місце A_s . І навпаки, якщо елементу A_s привласнимо значення елемента A_k , то тим самим знищимо попереднє значення елемента A_s . Таким чином, значення елемента A_s треба запам'ятати, тобто помістити заздалегідь в деякий допоміжний осередок R . Тепер елементу A_s можна привласнити значення елемента A_k , а елементу A_k привласнити значення елемента A_s , що зберігається в допоміжному осередку R .

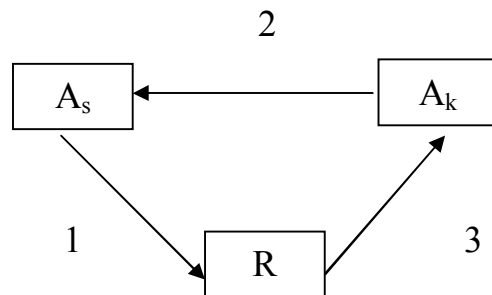
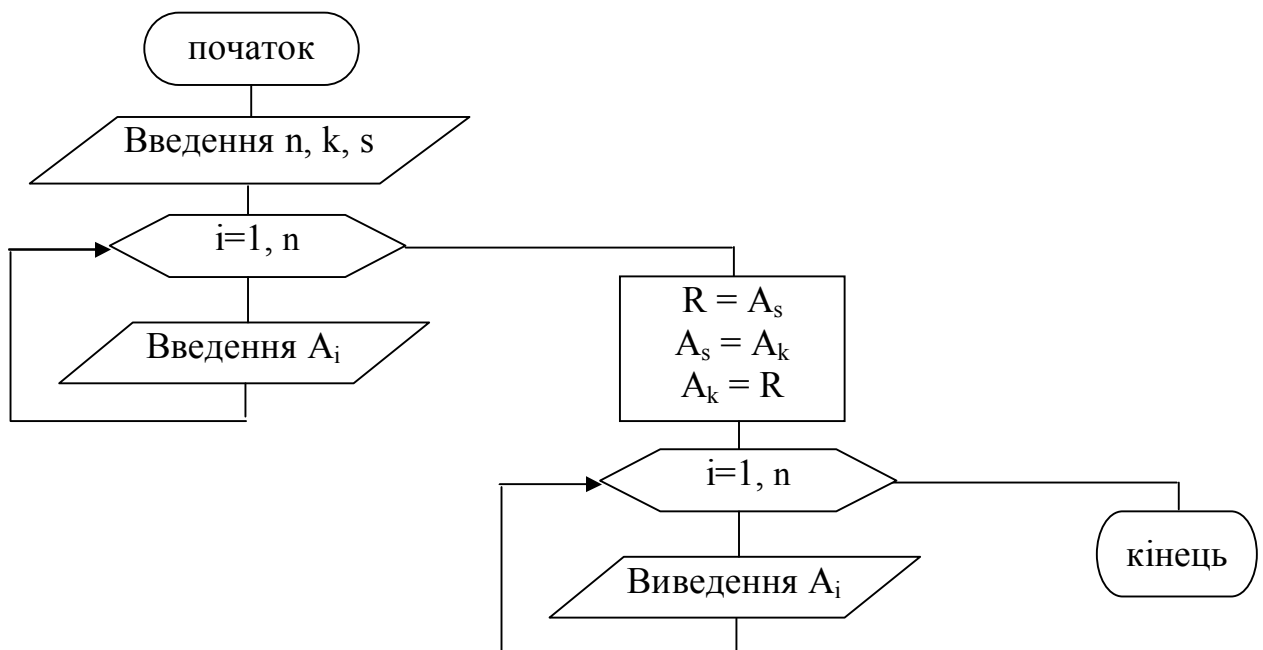


Схема алгоритму рішення задачі має вигляд:



2.8 Алгоритми визначення найбільшого і найменшого значення

Знаходження найбільшого або найменшого значення здійснюється в циклі в результаті порівняння деякого поточного значення з найбільшим (найменшим) зі всіх попередніх значень. При цьому якщо поточне значення більше найбільшого (менше найменшого) зі всіх попередніх, то найбільшому (найменшому) привласнюється значення поточного. Інакше найбільше (найменше) зберігає своє старе значення. Це можна описати умовною математичною формулою:

$$\max = \begin{cases} y_i, & \text{якщо } y_i > \max; \\ \max, & \text{якщо } y_i \leq \max. \end{cases}$$

Аналогічно, для найменшого значення:

$$\min = \begin{cases} y_i, & \text{якщо } y_i < \min; \\ \min, & \text{якщо } y_i \geq \min. \end{cases}$$

Після першого виконання циклу обчислюється перше значення y і порівнюється з початковим значенням \max або \min . При цьому значення \max або \min повинне бути рівне першому значенню y . Тоді після обчислення другого значення y найбільше або найменше знаходиться з перших двох значень. Для цього необхідно як початкове значення \max брати свідомо мале число, з тим щоб напевно виконати умову порівняння, а як початкове значення \min брати, навпаки, свідомо велике число.

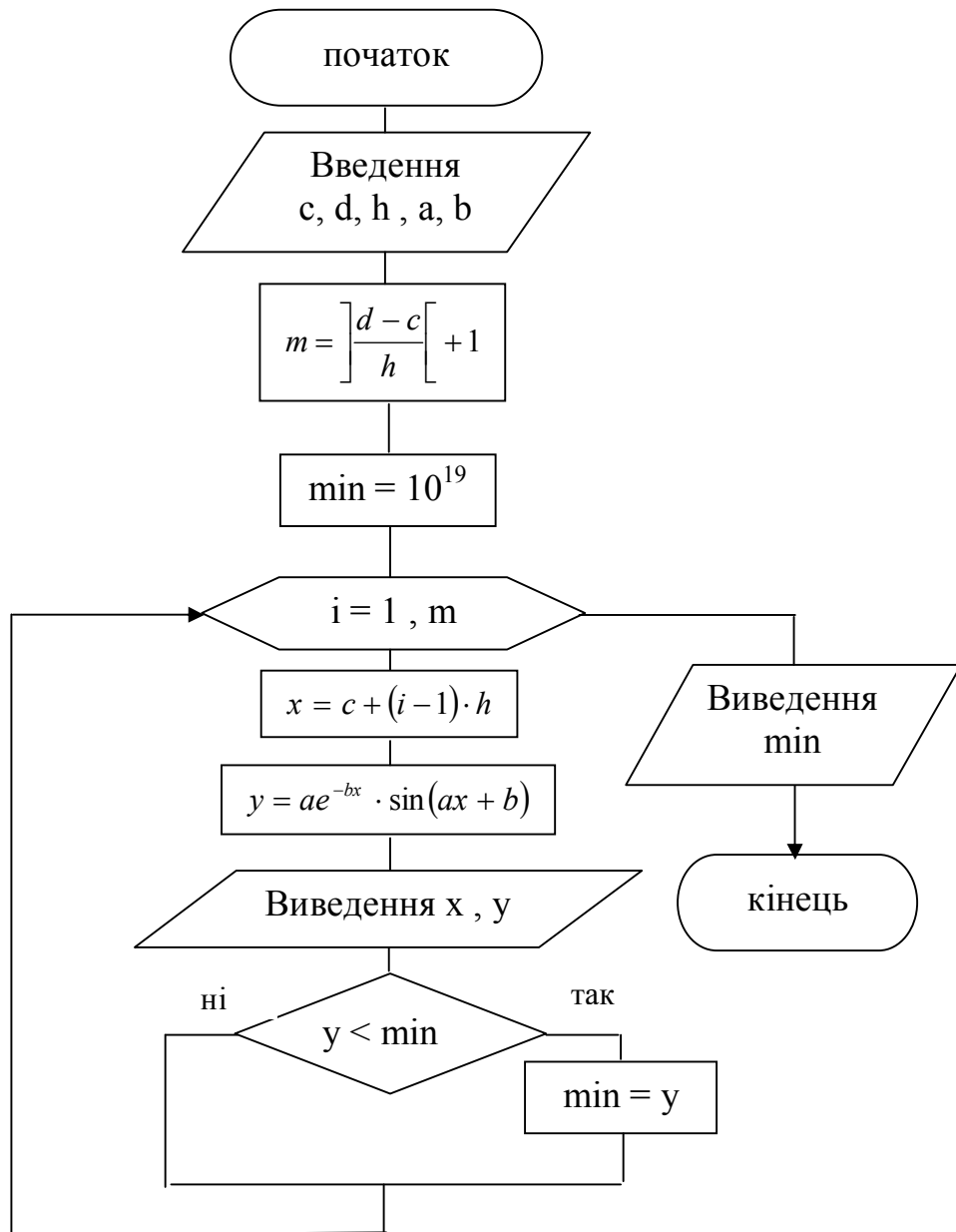
Приклад 17.

Визначити найменше значення функції:

$$y = ae^{-bx} \cdot \sin(ax + b)$$
$$c \leq x \leq d \quad \Delta x = h$$

Початкові дані: $x_n=c$, $x_k=d$, $\Delta x=h$, a , b .

Схема алгоритму рішення задачі має вигляд:



Приклад 18.

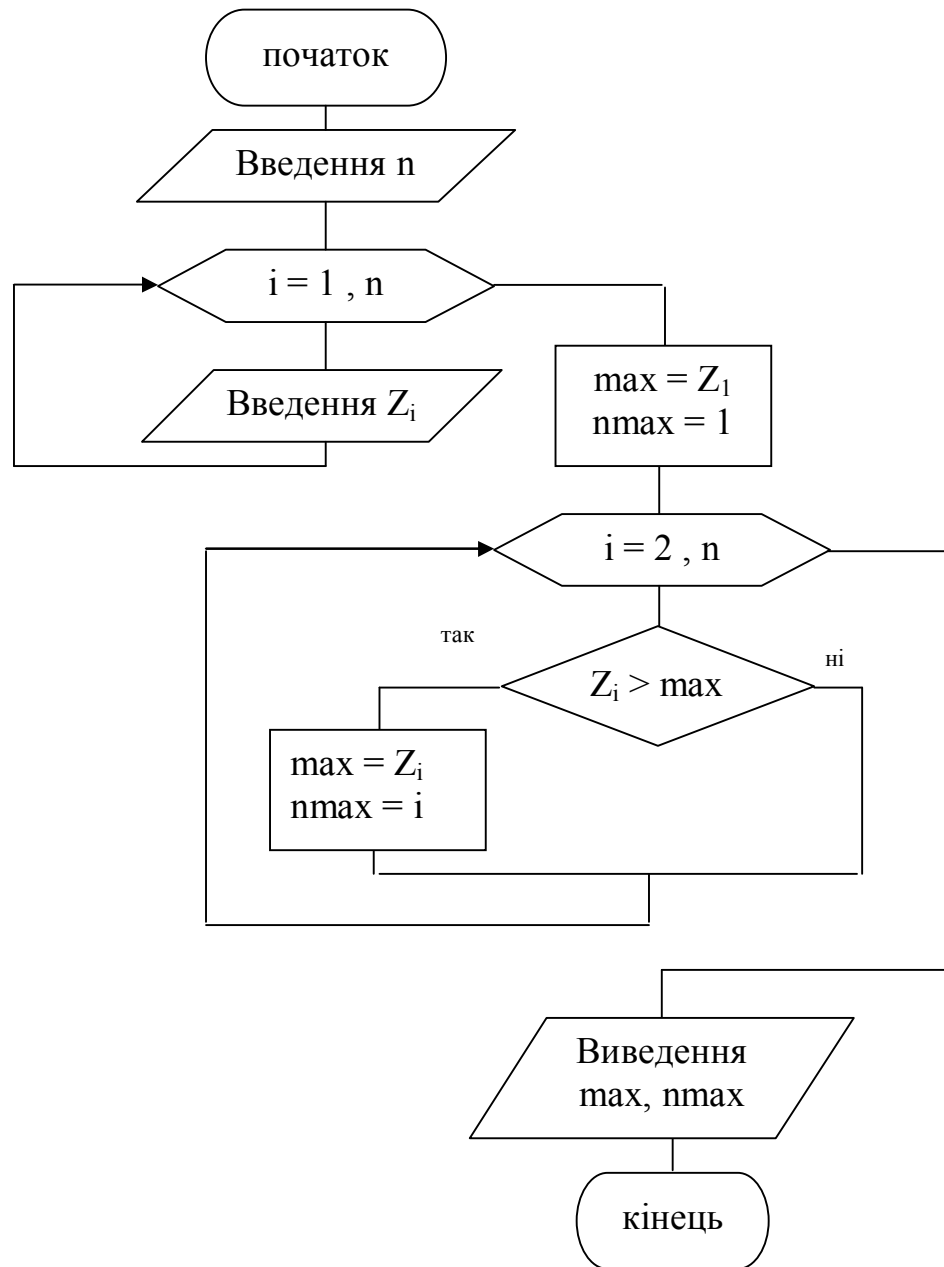
Визначити найбільший елемент масиву $Z(n)$ і його порядковий номер.

Початкові дані: Z – одномірний масив;

n – кількість елементів в масиві Z .

У даному завданні не має необхідності обчислювати порівнювані значення, оскільки вони є в масиві Z . Тому як початкове значення приймається перший елемент масиву Z . Оскільки порівняння першого елемента масиву з собою не має сенсу, цикл виконується починаючи з другого елемента.

Схема алгоритму рішення задачі має вигляд:



Приклад 19.

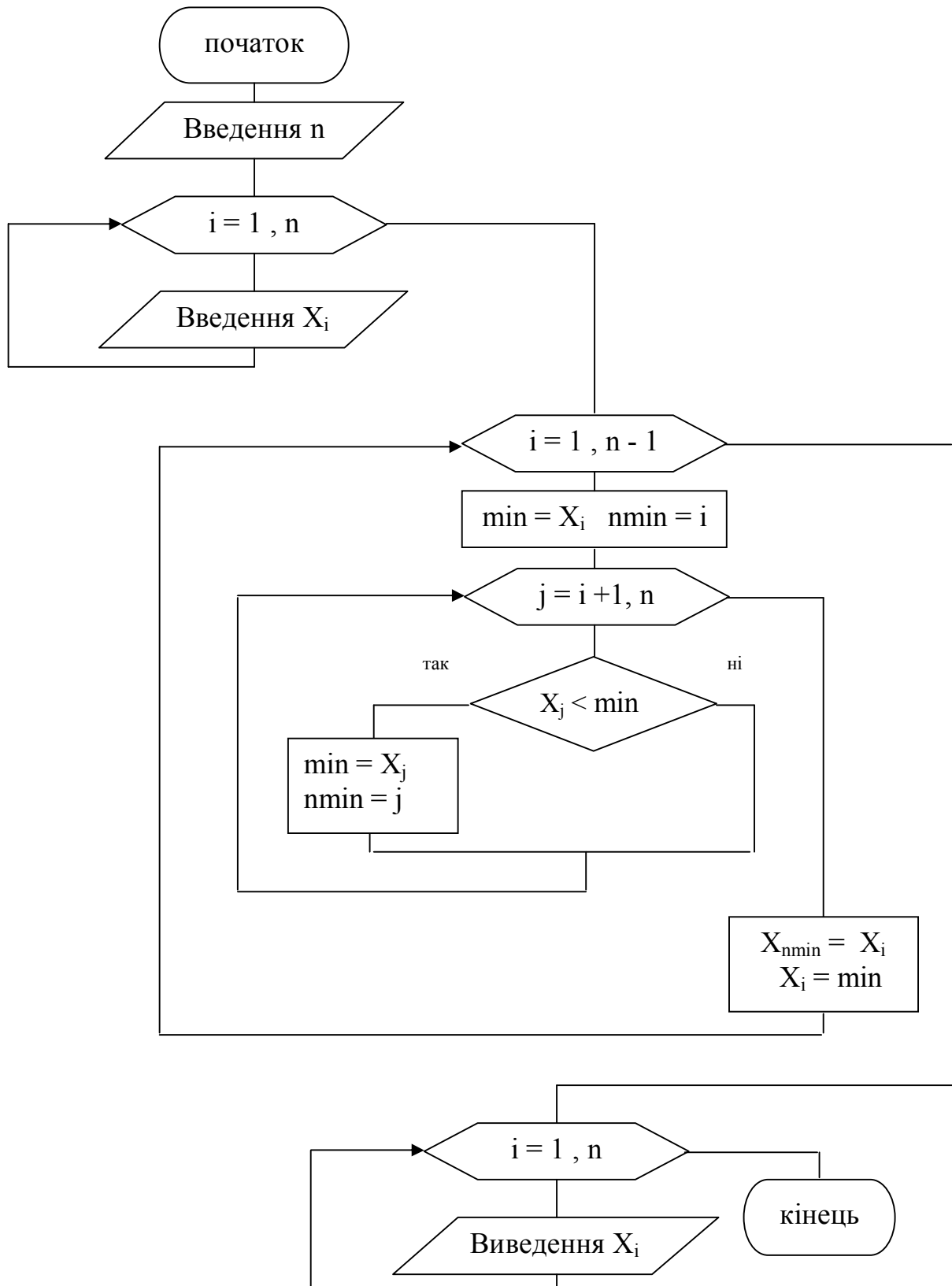
Упорядкувати елементи масиву $X(N)$, розташувавши їх в порядку зростання в тому ж масиві.

1 спосіб.

Рішення зводиться до багатократного знаходження найменшого елементу масиву. Тому перед внутрішнім циклом задається початкове значення найменшого, а усередині циклу відшукується найменший елемент і його порядковий номер. Після закінчення внутрішнього циклу найменший

елемент записується на місце першого, а перший – на місце найменшого. Повторюючи ці дії, починаючи з другого, потім третього елементу і т.д., можна добитися того, що всі елементи масиву будуть впорядковані за збільшенням.

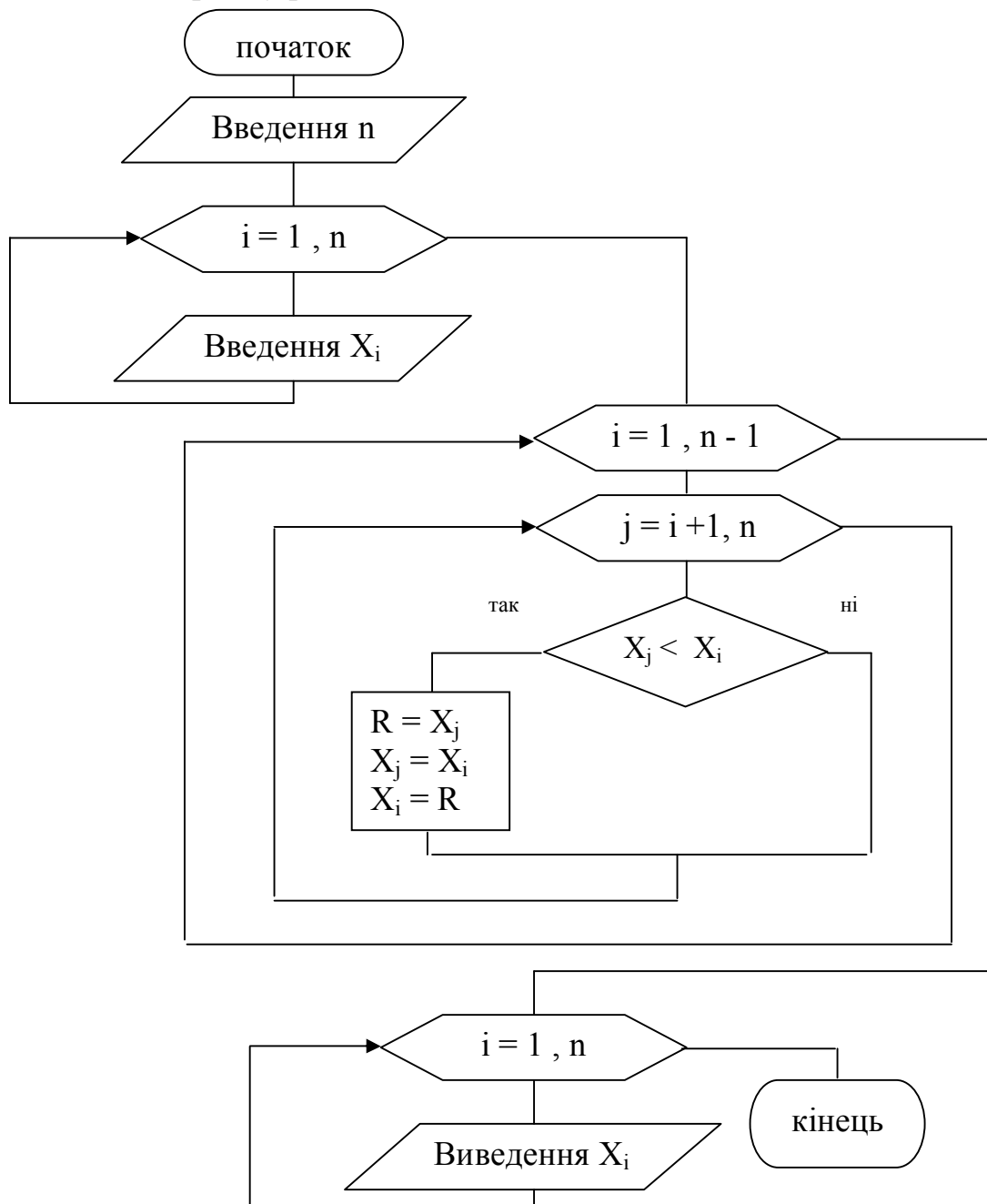
Схема алгоритму рішення задачі має вигляд:



2 спосіб.

Суть методу полягає в наступному: перший елемент масиву порівнюється зі всіма подальшими елементами. Якщо елемент буде більше якого-небудь подальшого елемента, необхідно здійснити їх перестановку. В результаті на місці першого елементу стоятиме найменший елемент. Потім порівнюється другий елемент з подальшими елементами. В результаті на місці другого елементу стоятиме число, більше першого, але менше за всіх інших. Далі виконуються аналогічні порівняння для решти всіх елементів до тих пір, поки не будуть порівняні два останніх. У результаті вийде зростаюча послідовність.

Схема алгоритму рішення задачі має вигляд:



2.9 Алгоритми формування і обробки двомірних масивів

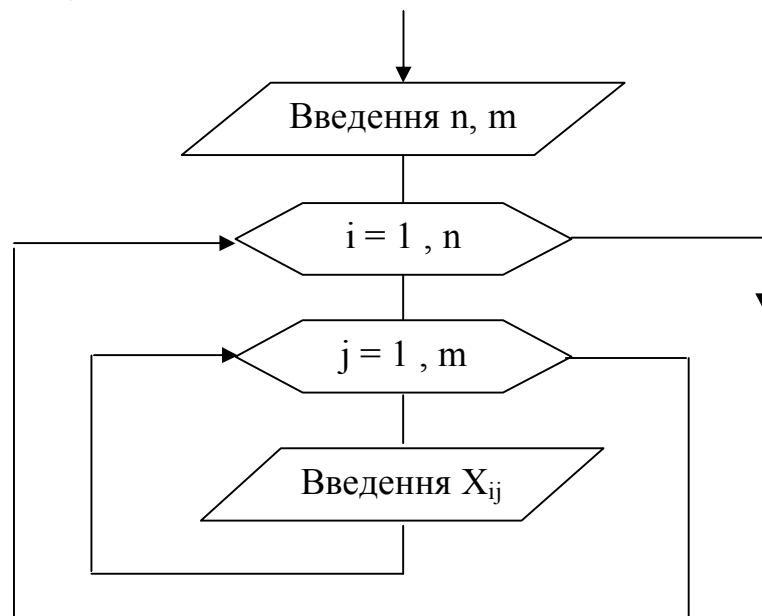
У математиці часто використовуються багатовимірні масиви, тобто масиви масивів. Особливо широкого поширення набули двомірні масиви, інакше звані матрицями, тобто двомірний масив - це деяка матриця з певним числом рядків і стовпців. Кількість рядків і стовпців визначають розмірність двомірного масиву. Наприклад, зображення цілих чисел послідовне в декількох рядках є матрицею:

```
5 4 3 6
2 8 1 7
4 3 9 5
```

Дана матриця має розмір 3 на 4, тобто вона складається з 3 рядків і 4 стовпців. Кожен елемент масиву позначається двома індексами, наприклад A_{ij} : перший з них, індекс i , указує номер рядка, а другий, індекс j , номер стовпця, на перетині яких розташовується даний елемент, наприклад A_{13} – це елемент масиву A , що розташовується на перетині першого рядка і третього стовпця.

Як ввести двомірний масив?

Для введення елементів масиву необхідно спочатку ввести розмірність масиву, тобто кількість рядків і кількість стовпців у матриці, а потім організувати цикли с відомим числом повторень для зміни індексів елементів масиву.



Приклад 20.

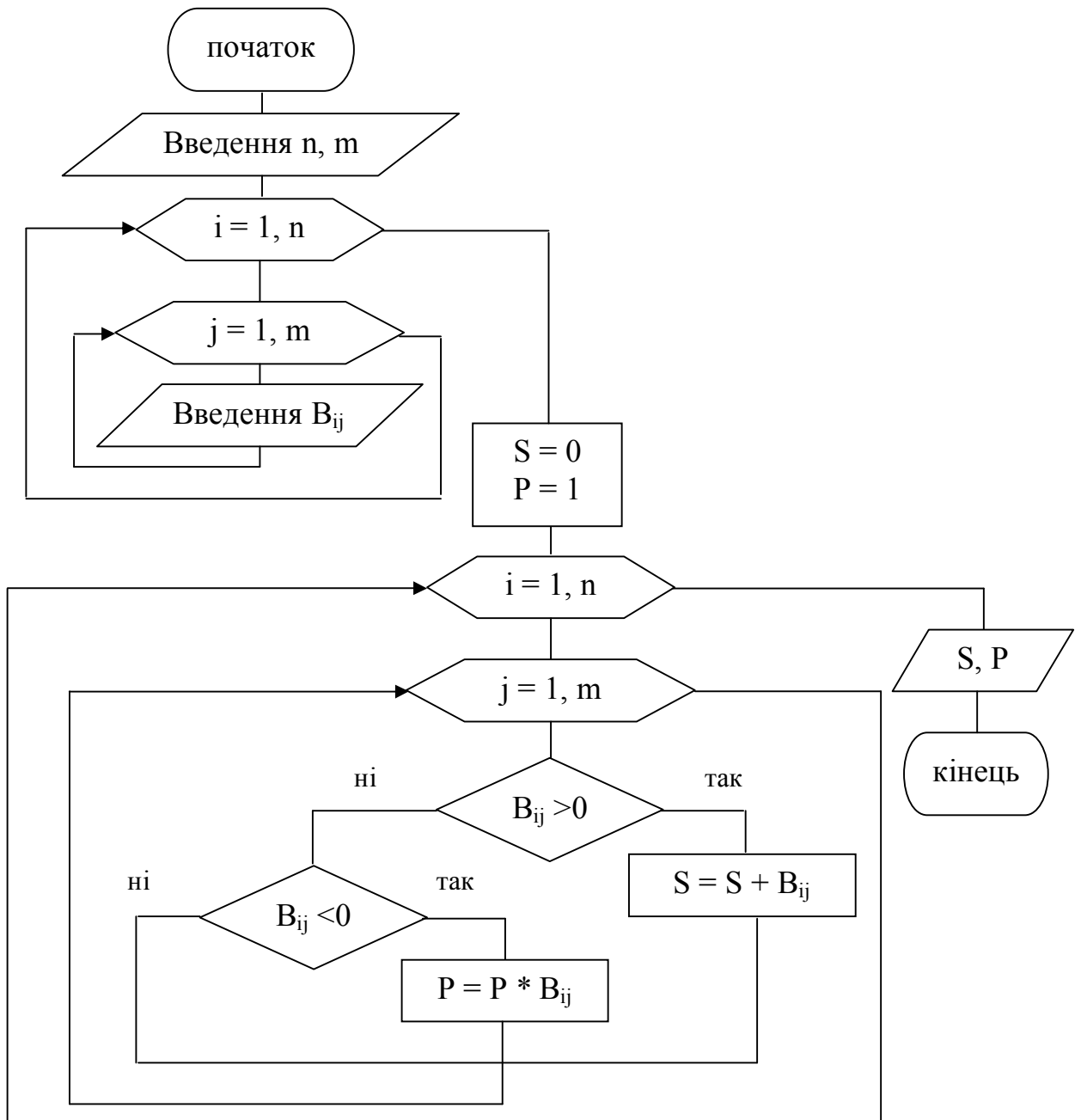
Для масиву $V(n,m)$ знайти добуток від'ємних (P) і суму додатних (S) елементів.

Початкові дані: V – двомірний масив;

n – число рядків в масиві V ;

m – число стовпців в масиві V .

Схема алгоритму рішення задачі має вигляд:



Приклад 21.

Для масиву $A(n,m)$ знайти відсоток нульових елементів.

Початкові дані: A – двомірний масив;

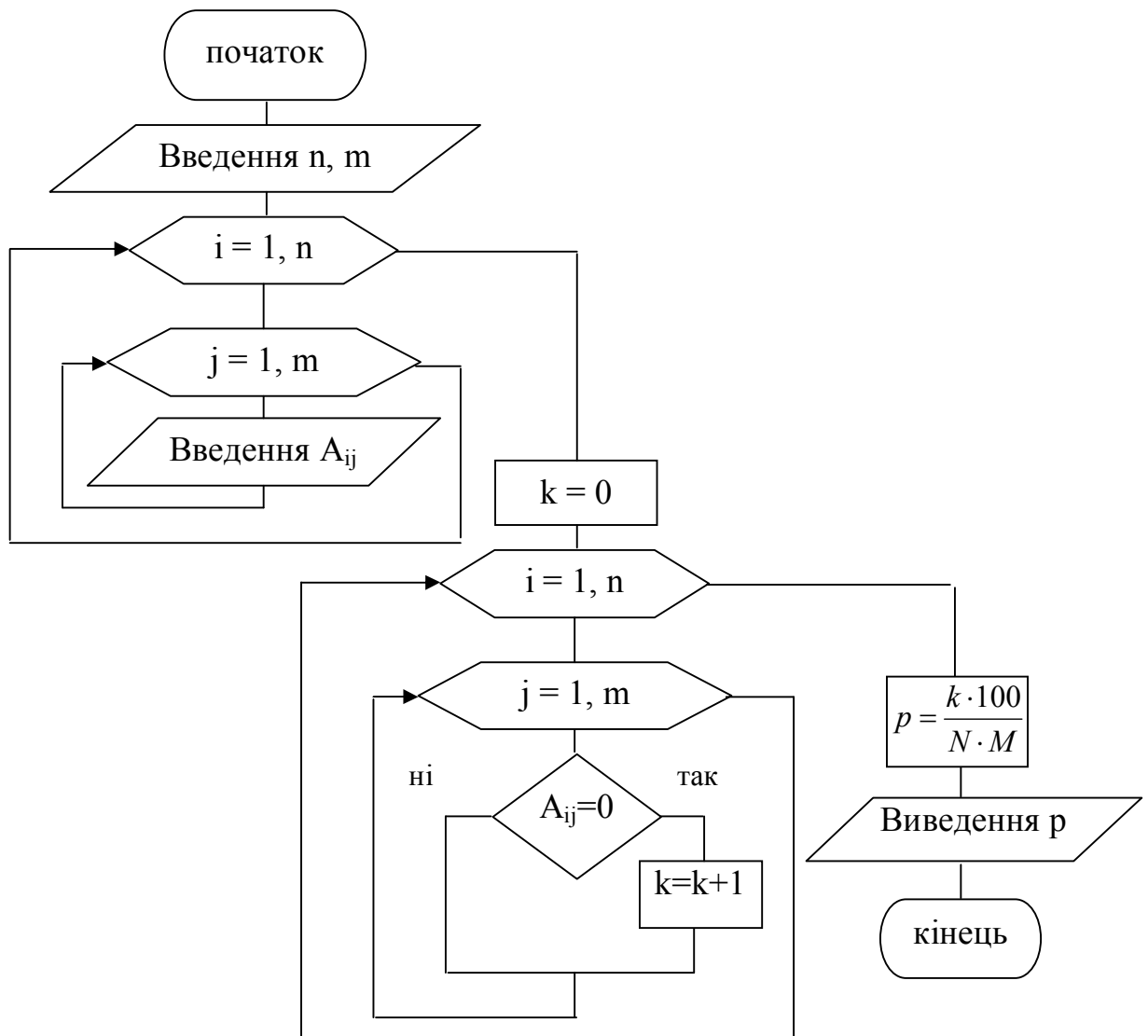
n – число рядків в масиві A ;

m – число стовпців в масиві A .

Формула визначення відсотка нульових елементів: $p = \frac{k}{n \cdot m} \cdot 100\%$,

де k – кількість нульових елементів в масиві A .

Схема алгоритму рішення задачі має вигляд:



Приклад 22.

Для масиву $T(n, m)$ знайти кількість і добуток додатних елементів, розташованих нижче за головну діагональ.

Початкові дані: T – двомірний масив;

n – число рядків в масиві T ;

m – число стовпців в масиві T .

Для вирішення завдання необхідно пригадати деякі властивості матриць:

- якщо номер рядка елемента співпадає з номером стовпця ($i = j$), то елемент знаходиться на головній діагоналі;

- якщо номер рядка перевищує номер стовпця ($i > j$), то елемент знаходиться нижче за головну діагональ;

- якщо номер стовпця перевищує номер рядка ($i < j$), то елемент знаходиться вище за головну діагональ;

- елемент знаходиться на побічній діагоналі, якщо його індекси задовольняють рівності $i + j - 1 = n$;

- нерівність $i + j - 1 < n$ характерна для елемента, що знаходиться вище за побічну діагональ;

- нерівність $i + j - 1 > n$ характерна для елемента, що знаходиться нижче за побічну діагональ.

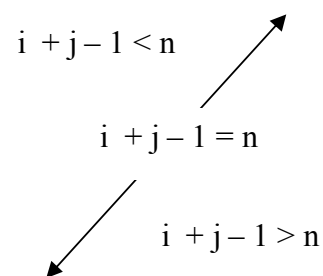
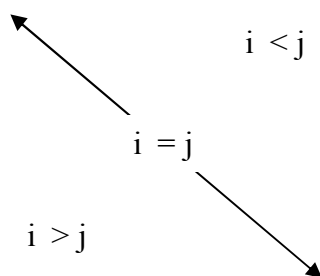
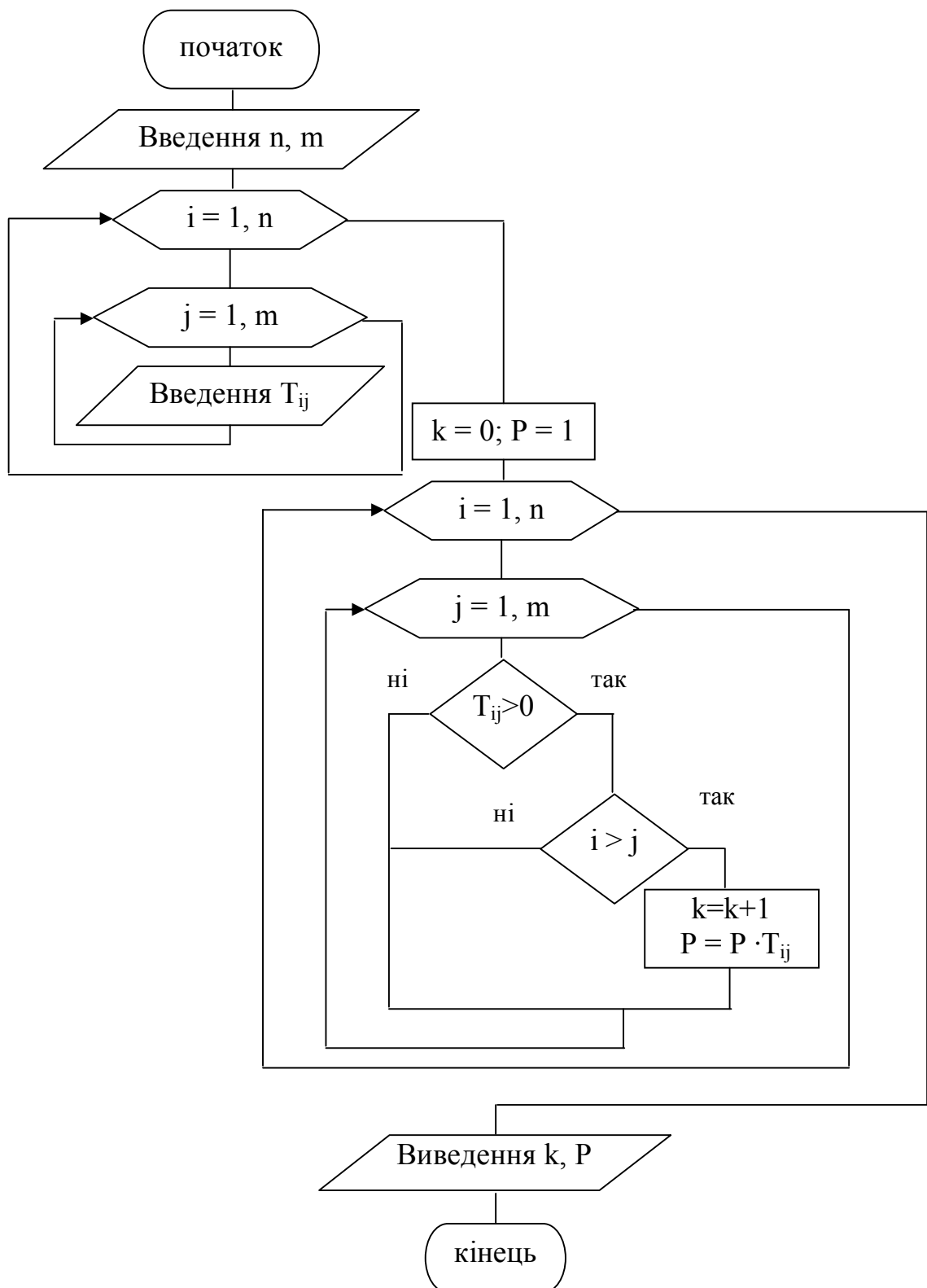


Схема алгоритму рішення задачі має вигляд:



Приклад 23.

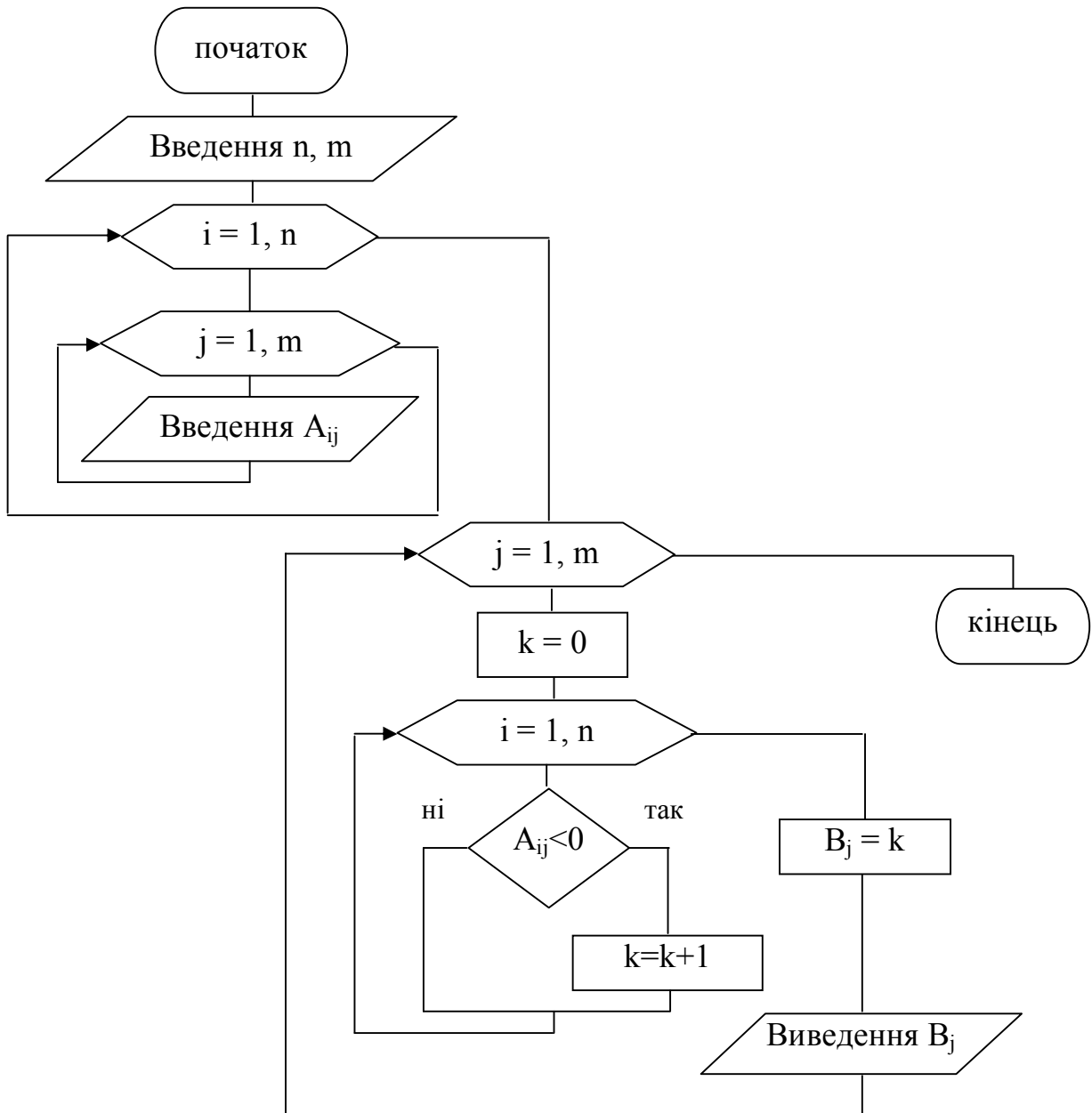
З масиву $A(n,m)$ сформувати масив $B(m)$, кожен елемент якого рівний кількості від'ємних елементів відповідного стовпця масиву A .

Початкові дані: A – двомірний масив;

n – число рядків в масиві A ;

m – число стовпців в масиві A .

Схема алгоритму рішення задачі має вигляд:



У даному прикладі зовнішній цикл повинен бути по індексу стовпця, оскільки інакше було б обчислено кількість від'ємних елементів кожного рядка.

2.10 Алгоритми, які містять допоміжні підзадачі

При складанні складніших програм неминуче виникають труднощі, обумовлені наявністю великої кількості об'єктів, безліч різних зв'язків між цими об'єктами. Тому краще розбивати програму на окремі смислові блоки.

У практиці часто зустрічаються випадки, коли по ходу виконання розрахунків в завданнях доводиться проводити одні і ті ж обчислення, але при різних початкових даних.

Щоб виключити повторення однакових дій і зробити тим самим алгоритм простіше і зрозуміліше, можна виділити дії, що повторюються, в самостійну частину у вигляді допоміжної підзадачі, яка може бути використана багато разів в міру необхідності. Така допоміжна підзадача, що реалізовує певний алгоритм і яка допускає звернення до неї з різних частин блок-схеми основної задачі, називається підпрограмою.

Підпрограми оформляються у вигляді окремих алгоритмів, що мають чітко позначений вхід і вихід.

При складанні блок-схеми основної задачі обов'язково передбачити звернення до підпрограми.

Підпрограми бувають двох видів: процедури і функції. Основна відмінність їх полягає в тому, що результат роботи процедури – це цілий набір значень, а результат виконання функції – це одне значення.

Приклад 24.

Обчислити різницю середніх арифметичних значень елементів масиву $X = (X_1, X_2 \dots, X_{50})$ з парними індексами і елементів з непарними індексами:

$$R = \frac{1}{25} \sum_{i=2,4,\dots}^{50} X_i - \frac{1}{25} \sum_{i=1,3,\dots}^{49} X_i$$

Для обчислення R необхідно двічі визначити середньо арифметичне значення 25 чисел. Тому в якості допоміжної підзадачі можна виділити обчислення середнього арифметичного значення 25 чисел:

$$S = \frac{1}{25} \sum_{j=k, k+2, \dots}^n X_j$$

Перший раз необхідно задати $k = 1$ і $n = 49$.

Другий раз необхідно задати $k = 2$ і $n = 50$.

Допоміжну підзадачу можна оформити у вигляді підпрограми функції.

Блок-схема алгоритму допоміжної підзадачі починається стрілкою «вхід» з вказівкою списку початкових даних, а закінчується стрілкою «вихід» із списком вихідних даних.

Результат функції привласнюється змінним $S1, S2$.

Схема алгоритму підпрограми функції SUM має вигляд:

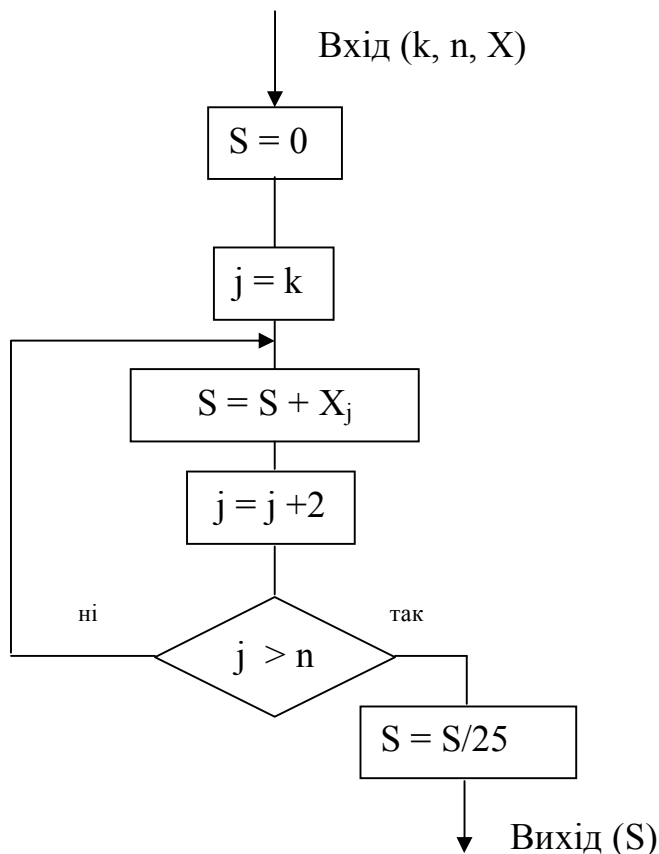
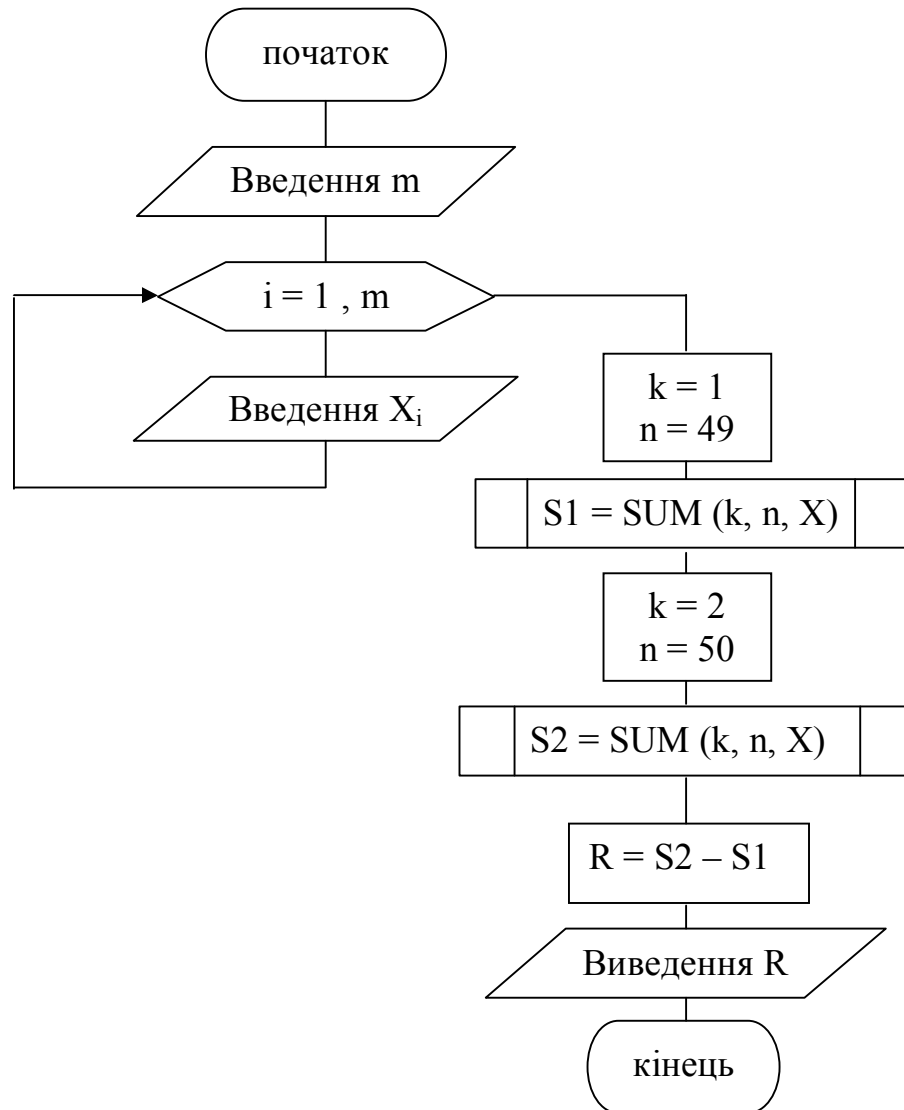


Схема алгоритму основної задачі має вигляд:



Приклад 25.

Сформувати масиви, елементи яких обчислюються за формулами:

$$a_k = 2.8(k - 3.6)^2 + \sqrt{k + 2},$$

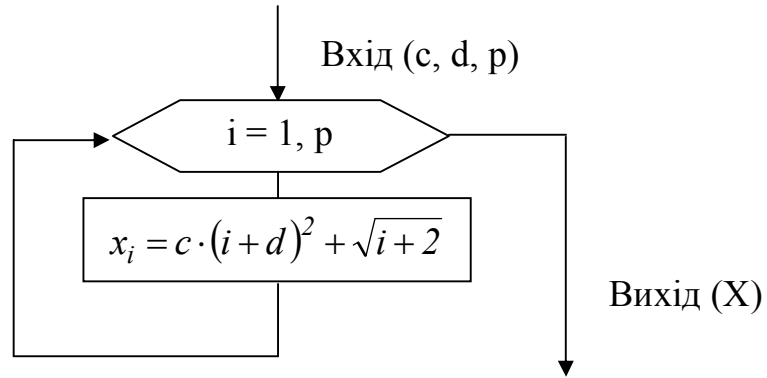
$$b_j = 5.4(j + 2.6)^2 + \sqrt{j + 2},$$

де

$$k = 1, m ; i = 1, n.$$

В якості допоміжної підзадачі можна виділити обчислення елементів масиву за формулою: $x_i = c \cdot (i + d)^2 + \sqrt{i + 2}$ і виведення елементів масиву. Допоміжну підзадачу можна оформити у вигляді підпрограми процедури.

Процедура розрахунку елементів масиву – FORM



Процедура виведення елементів масиву – VUVOD

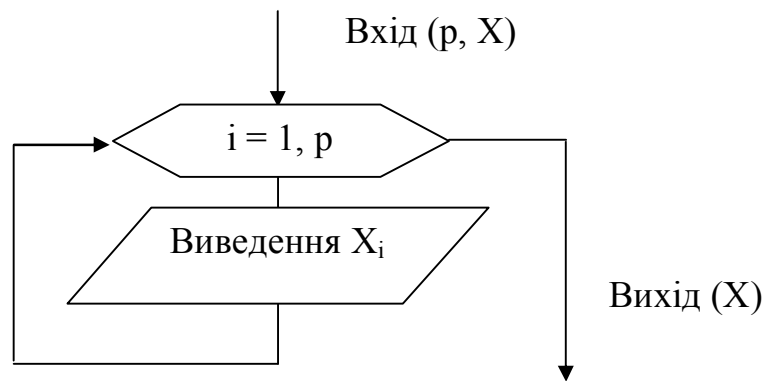
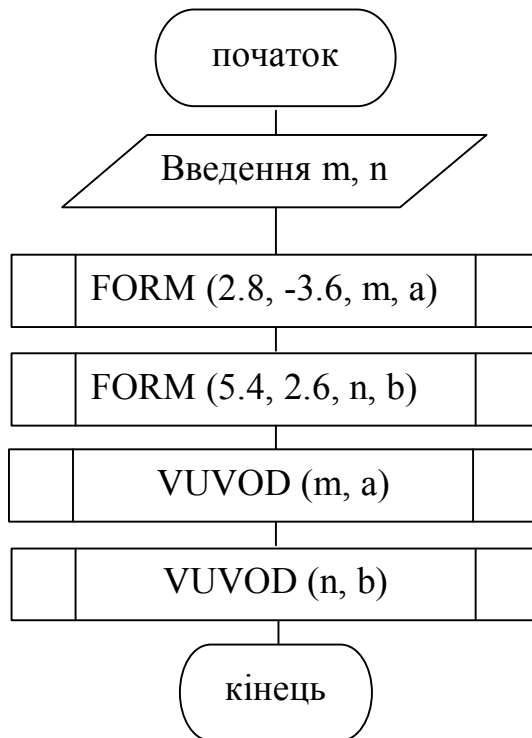


Схема алгоритму основної задачі має вигляд:



3. ОСНОВИ ПРОГРАМУВАННЯ НА MOBI PASCAL

3.1 Середовище програмування Turbo Pascal

При програмуванні важливо знати не тільки мову програмування, але і добре орієнтуватися в середовищі розробки. Turbo Pascal – це середовище програмування, орієнтоване на мову Pascal, що дозволяє створювати, редагувати, відладжувати і виконувати програми. Обов'язковим файлом при роботі з системою Turbo Pascal є файл turbo.exe.

Після завантаження файлу turbo.exe на екрані з'являється вікно середовища програмування Turbo Pascal (рис.3.1).

У верхньому рядку екрану відображається головне меню, потім вікно редагування Edit і в нижньому рядку – рядок підказки про призначення функціональних клавіш.

Всі дії, які необхідні для роботи в системі Turbo Pascal (виклик програми, компіляція, запуск, відлагодження і т.д.) доступні через головне меню, перехід в яке здійснюється клавішею **F10**.

Для завершення роботи з системою Turbo Pascal необхідно натиснути комбінацію клавіш **ALT+X**, що відповідає команді **Quit** (пункт меню File).

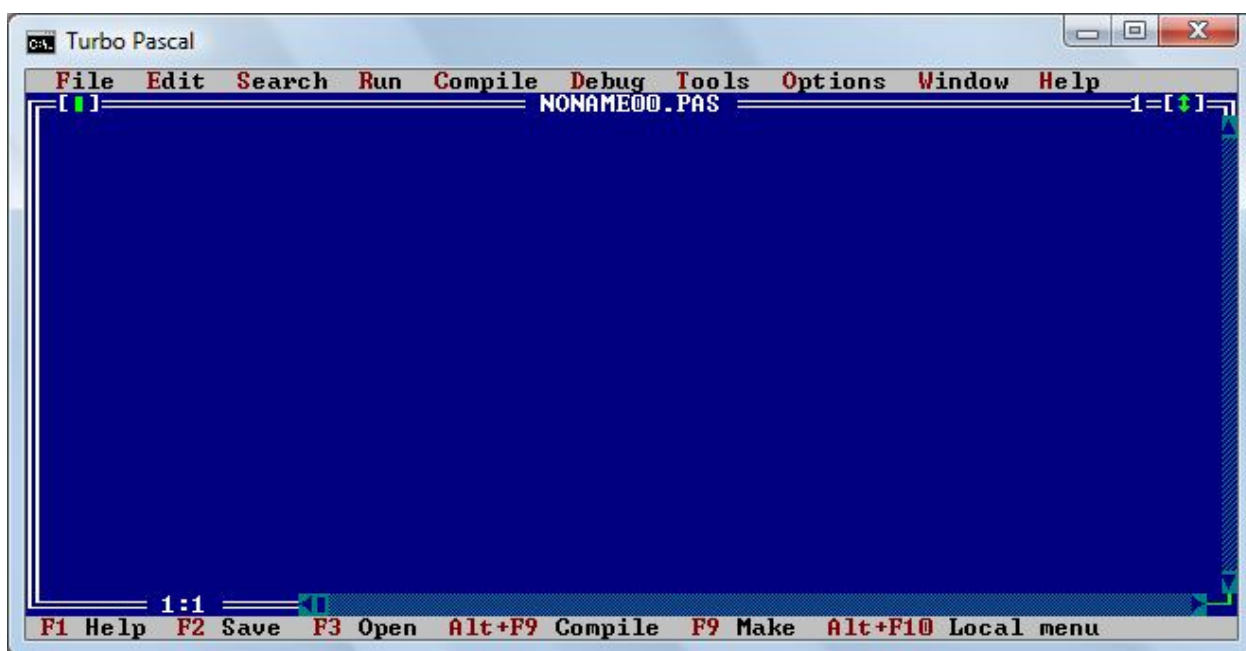


Рис.3.1 - Вид середовища програмування Turbo Pascal

На екрані одночасно може бути відкрито декілька вікон. Одне з видимих вікон завжди є поточним, що підтверджується подвійною лінією по його верхній межі і яскравішим кольором заголовка (рис.3.2).

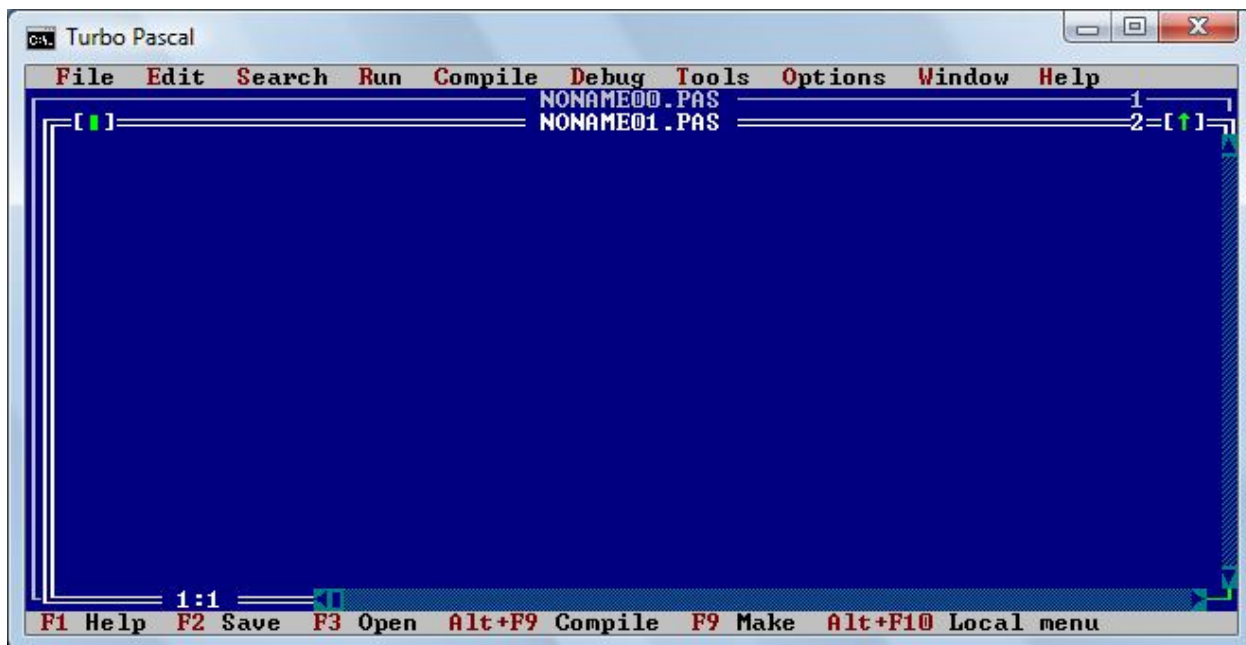
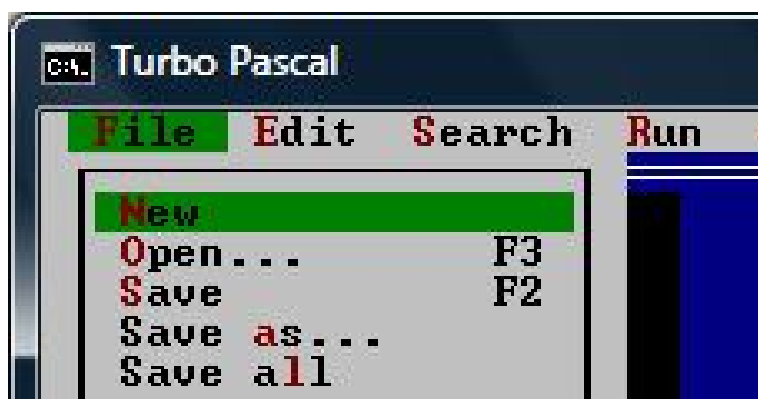


Рис.3.2 - Середовище програмування TP містить декілька відкритих вікон

Послідовність дій для створення файлу.

1. Запустити середовище Turbo Pascal.
2. Написати текст програми.
3. У головному меню вибрати пункт меню **File** і команду **Save as...** для збереження файлу.



З'явиться діалогове вікно, в якому необхідно ввести ім'я файлу і вказати його місце зберігання (рис.3.3).

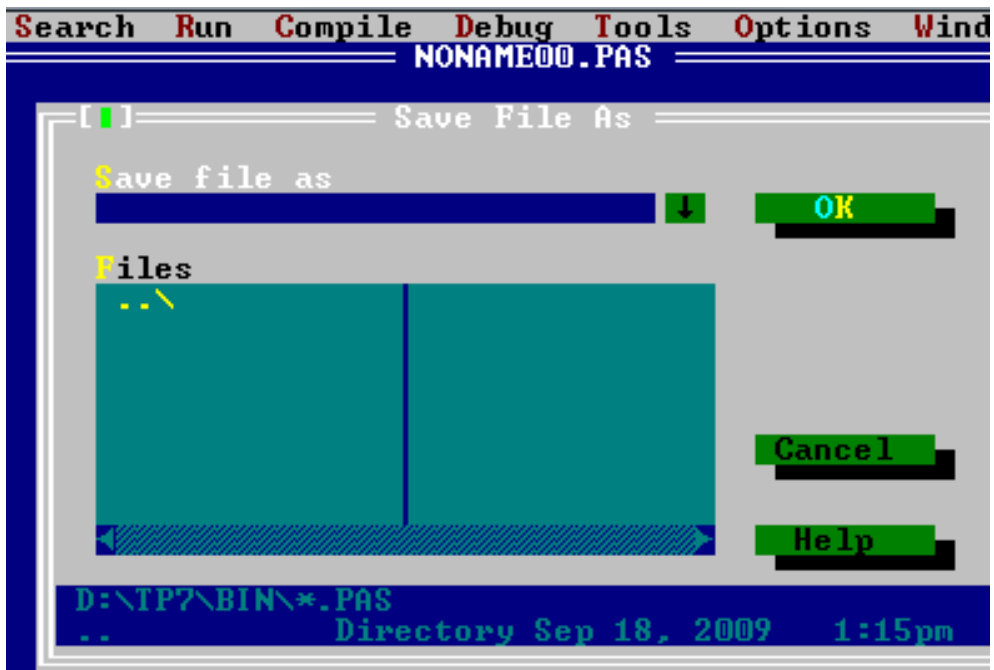
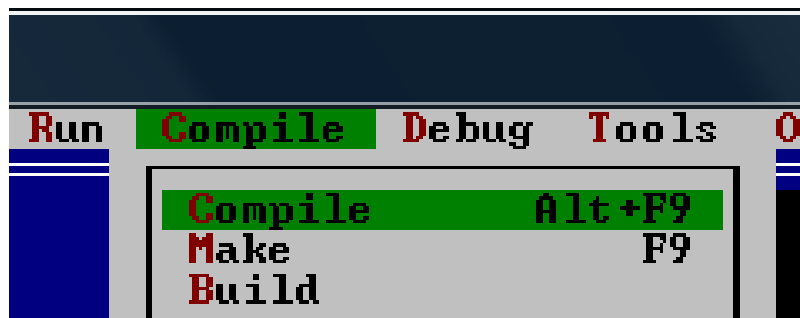
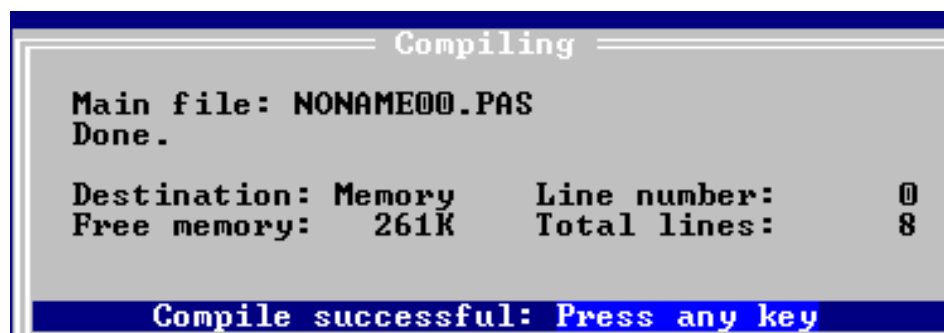


Рис.3.3 - Діалогове вікно для збереження файлу

4. Виконати компіляцію програми, використовуючи команду **Compile** (пункт меню **Compile**) або натиснути комбінацію клавіш **ALT+F9** (компіляція програми) чи **F9** (компіляція програми і створення завантажувального модуля).



Після компіляції програми на екрані з'являється вікно з результатами компіляції.



У разі виявлення помилок в тексті програми, компіляція припиняється і з'являється повідомлення про помилку. Необхідно виправити помилку і знову виконати компіляцію.

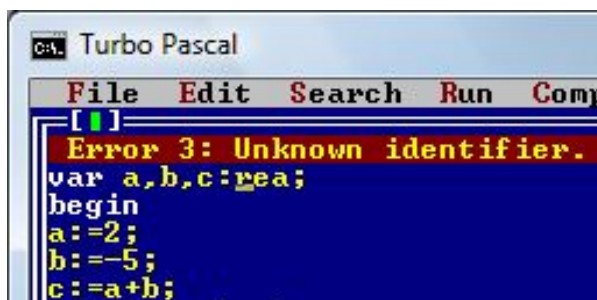


Рис.3.4. Повідомлення про помилки при виконанні компіляції програми

5. Запустити програму на виконання, використовуючи команду **Run** (пункт меню **Run**) або натиснути комбінацію клавіш **Ctrl+F9**. Команда **Trace info** використовується для відрядкового виконання програми.



Іноді на стадії виконання програми теж виникають помилки (неприпустимі операції – наприклад, ділення на нуль). При виникненні таких помилок виконання програми буде перервано. З'явиться код помилки і дана фізична інтерпретація її виникнення. Необхідно виправити помилку і знову виконати компіляцію.

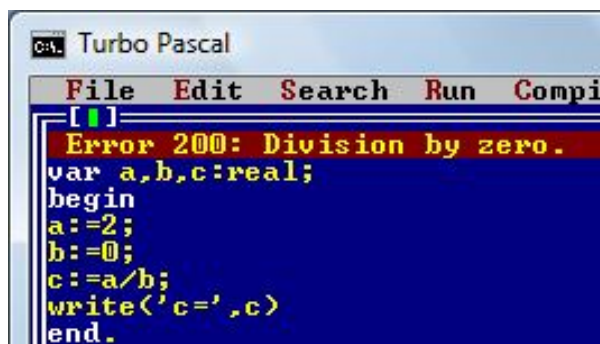


Рис.3.5. Повідомлення про помилки при виконанні програми

6. Щоб побачити результати розрахунків необхідно натиснути комбінацію клавіш **ALT+F5**.

3.2 Алфавіт мови Pascal

Алфавіт – це сукупність допустимих в мові символів або груп символів, що розглядаються як єдине ціле:

- букви від A до Z, від a до z;
- цифри від 0 до 9;
- спеціальні символи + - * / = < > . : ; ‘ ^ \$ # _ { } [] ()

З символів алфавіту формуються ключові слова і ідентифікатори.

Ключові слова – це зарезервовані слова, які використовуються в програмі тільки по своєму прямому призначенню.

Ідентифікатори (імена) використовуються для позначення констант, типів, змінних, процедур, функцій і т.д. Ідентифікатори складаються з букв, цифр, знаку підкреслення і повинні починатися з букви. Довжина ідентифікатора може бути довільною. Спеціальні символи не можуть входити до складу ідентифікаторів, а також зарезервовані слова не можна використовувати як ідентифікатори.

Приклади допустимих імен:

X	Y	Sum	P1_1	myprogram
---	---	-----	------	-----------

Приклади неприпустимих імен:

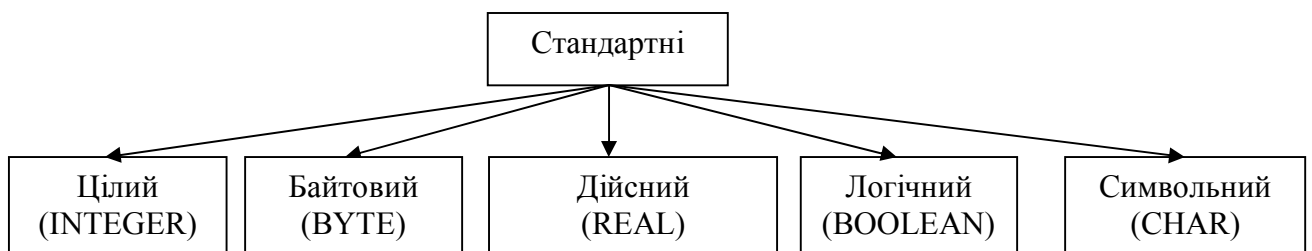
5th	починається з цифри
Type	зарезервоване слово
Sum.6	містить спеціальний символ
X 5	містить пропуск
Корінь	містить російські букви

У тексті програми можна використовувати коментарі. **Коментар** – це будь-яка послідовність символів, ув'язнена у фігурні дужки: { це коментар}.

Замість фігурних дужок можна використовувати пари символів (* і *). Коментар може займати будь-яке число рядків і під час виконання програми ігнорується. Основне призначення коментаря – пояснення по тексту програми.

3.3 Типи даних

Програми обробляють дані, які можуть приймати числові, логічні і буквені значення. Дані представляються у вигляді констант і змінних. **Константа** – це область пам'яті, яка має незмінне значення. **Змінною** називається ім'я, що визначає область пам'яті для зберігання величини, яка може змінюватися під час роботи програми. Всі типи даних можна розділити на прості і складні. Прості – це стандартні і змінні типи даних.



Цілий тип призначений для представлення цілих чисел в діапазоні від -32767 до 32767.

Байтовий тип призначений для представлення цілих чисел з інтервалу від 0 до 255.

Дійсний тип призначений для представлення дійсних чисел, тобто чисел, що мають дробову частину, нуль і відповідні від'ємні числа. Дійсні числа мають дві форми запису:

- з **фіксованою крапкою** $m.n$,

де m – ціла частина числа, n – дробова частина числа.

Наприклад, 27.3 5.0 -16.003

- з **плаваючою комою** mEr ,

де m – мантиса числа, E – ознака запису числа з десятковим порядком, r – порядок числа.

Наприклад,

математичний запис

запис на мові Pascal

$4 \cdot 10^{-5}$

4E-05

$0,62 \cdot 10^4$

0.62E+04

$-10,88 \cdot 10^{12}$

-10.88E+12

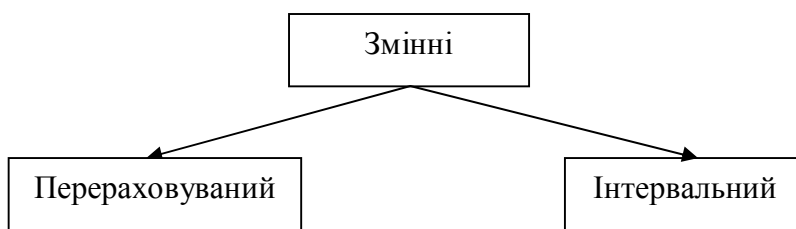
Логічний тип: дані цього типу можуть приймати одне з двох значень істина (TRUE) або лож (FALSE).

Наприклад,

$5 > 1.25$ результат - true $10 > 25.33$ результат - false

Символьний тип: дані символного типу призначені для зберігання одного символу. У тексті програми значення змінних і констант символного типу повинні бути розташовані в апострофах: 't' '*' '5'.

Змінні типи даних визначаються користувачем.



Перераховуваний тип задається перерахуванням тих значень, які він може отримувати. Наприклад, A: (red, white, blue)- це означає, що змінна A може приймати тільки одне з цих трьох значень.

Інтервальний тип задається двома константами, що визначають межі діапазону для змінної. Наприклад, M: 1..12 (змінна M може приймати значення від 1 до 12). Накладати обмеження на дійсний тип не можна.

Складні – це різні комбінації простих типів (масиви, множини, записи, файли). Введення складних типів даних робить мову програмування могутнішою і дозволяє складати програми широкого класу завдань.

У математиці, економіці, інформатиці часто використовуються впорядковані набори даних, наприклад, послідовності чисел, таблиці, списки прізвищ. Для обробки наборів даних одного типу вводиться поняття масиву.

Масив – це група однотипних змінних, які об'єднані одним загальним ім'ям. Доступ до заданого елемента масиву здійснюється за допомогою індексу.

Під **множиною** розуміють обмежений, нерегульований набір різних елементів однакового типу. Всій множині в цілому дається ім'я.

У багатьох інформаційних завданнях обробляються відомості, документи, списки. При цьому з'являється необхідність об'єднати дані різного типу в одну групу. Для роботи з групою даних введено поняття запису. **Запис** це сукупність обмеженого числа даних різного типу.

Під **файлом** розуміють будь-який набір даних. Так, початкові дані в програмі можна вважати файлом. Файлом можуть бути і результати, і сама програма.

Перед використанням змінної або константи в програмі їх необхідно спочатку оголосити, тобто задати їх тип і область використання.

Оголошення змінної здійснюється за допомогою службового слова `Var`.

Var ім'я змінної: тип змінної;

Наприклад,

```
Var a, b,c: integer;           // наприклад, 0  12  -6
    S: real;                   // наприклад, 1.2  1.8E-03
    K: char;                   // наприклад, 'ABC' '1234'
```

Оголошення константи здійснюється за допомогою службового слова `Const`.

Const ім'я константи = значення;

Наприклад,

```
Const pi=3.141492;           //константа дійсного типу з фіксованою
                               //крапкою
    K=100;                    // константа цілого типу
    Abc='p';                  // константа символного типу
```

3.4 Операції і стандартні функції мови Pascal. Вирази.

Для обчислення значень використовуються операції і стандартні функції (див. табл. 3.1). Операції визначають, які дії слід виконати над даними. Операції застосовуються до операндів. Наприклад, операція «+» виконує операцію складання двох чисел або виразів, що є операндами.

Всі операції можна розділити на декілька груп:

1) **арифметичні операції** використовуються для виконання основних арифметичних операцій над даними:

* множення

/ ділення

+ складання

- віднімання

div цілочисельне ділення двох чисел

mod визначення залишку при діленні двох чисел

Пояснення операцій: ділення, div і mod

Вираз	Результат
8/4	2,0
8 div 4	2
8 mod 4	0
6/4	1,5
6 div 4	1
6 mod 4	2

2) **логічні операції** призначені для виконання різних логічних операцій:

or логічне складання (або)

and логічне множення (і)

not логічне заперечення (ні)

До логічних зазвичай відносяться також дві операції зрушення над цілими числами (і та j - вирази цілого типу):

i shl j - зрушення змісту і на j розрядів вліво; молодші розряди, що звільнилися, заповнюються нулями;

i shr j - зрушення змісту і на j розрядів управо; старші розряди, що звільнилися, заповнюються нулями.

Таблиця 3.1 – Стандартні функції мови Pascal

Функція	Назва	Тип аргументу	Тип результату
abs(x)	абсолютне значення	ціле або дійсне	співпадає з типом аргументу
sin(x)	синус	ціле або дійсне	дійсний
cos(x)	косинус	ціле або дійсне	дійсний
exp(x)	експонента	ціле або дійсне	дійсний
ln(x)	натуральний логарифм	ціле або дійсне	дійсний
arctan(x)	арктангенс	ціле або дійсне	дійсний
sqr(x)	квадрат аргументу	ціле або дійсне	співпадає з типом аргументу
sqrt(x)	квадратний корінь	ціле або дійсне	дійсний
trunc(x)	виділення цілої частини дійсного числа	ціле або дійсне	цілий
round(x)	округлення числа	ціле або дійсне	цілий
pi	число 3,14159	аргумент відсутній	дійсний
odd(x)	непарність числа, якщо число непарне – true, парне - false	цілий	логічний
pred(x)	визначає попереднє значення x	цілий логічний символьний	цілий логічний символьний
succ(x)	визначає наступне значення x	цілий логічний символьний	цілий логічний символьний
ord(x)	визначає порядковий номер символу x в наборі символів	логічний символьний	цілий
chr(i)	визначає символ з набору символів по порядковому номеру i	цілий	символьний
frac(x)	дробова частина аргументу: $x - \text{int}(x)$	цілий	дійсний
int(x)	ціла частина аргументу	цілий	цілий

3) операції порівняння

призначені для виконання операцій порівняння логічних виразів:

= <> < > <= >=

Вирази визначають послідовність обчислення значення. Вирази можуть включати константи, змінні, стандартні функції, які розділяються дужками і знаками операцій.

Щоб уникнути помилок при написанні виразу рекомендується виконувати привласнення тільки між сумісними типами даних. Якщо вираз містить декілька операцій, то значення компонентів виразу розраховуються в певному порядку, який називається пріоритетом операцій (табл.3.2). Якщо вираз містить операції різних типів, то першими виконуються арифметичні операції, потім операції порівняння і в останню чергу логічні операції.

Таблиця 3.2 - Пріоритет операцій

Операції	Пріоритет
not	1 (вищий)
* / div mod and shl shr	2
+ - or	3
= <> < > <= >=	4 (нижчий)

Наприклад, запис виразу на мові програмування Pascal

1) $\ln|\arctg(x) - \sin(ax)| + \sqrt[3]{ax}$

має вигляд: $\ln(\text{abs}(\arctan(x) - \sin(a*x))) + \exp(1/3*\ln(a*x))$

2) $\sin^2 z + \sqrt{|x-y|}$ має вигляд: $\text{sqr}(\sin(z)) + \text{sqrt}(\text{abs}(x-y))$

3) $\frac{\sin a - b}{|b| + \text{tgb}}$ має вигляд: $(\sin(a) - b) / (\text{abs}(b) + \sin(b)/\cos(b))$

4) $\sqrt[3]{e^{x-1/\sin x}}$ має вигляд: $\exp((x-1/\sin(x))/3)$

3.5 Структура програми

Програма на мові Pascal складається із заголовка, розділу описів і розділу операторів:

```
Program ім'я;  
розділ описів  
begin  
розділ операторів  
end.
```

Заголовок програми починається з службового слова **Program**, після вказується ім'я програми. Закінчується заголовок символом «;» (крапка з комою), наприклад, Program p1_1;

Розділ описів призначений для оголошення всіх в програмі даних, які зустрічаються, і їх характеристик. Існує певний порядок в цьому розділі:

- розділ міток;
- розділ констант;
- розділ типів;
- розділ змінних ;
- розділ процедур і функцій.

Не всі перераховані розділи обов'язкові в кожній програмі. У простих програмах може бути потрібно, наприклад, тільки розділи оголошення констант і змінних. Після кожного розділу опису ставиться символ «;» (крапка з комою).

Розділ міток починається з службового слова **Label**, потім слідує список міток. Мітка – це ідентифікатор або ціле число без знаку.

Label ім'я мітки;

Наприклад, Label rb, 32;

Розділ констант починається з службового слова **Const**

Const ім'я константи = значення;

Наприклад, Const P = 2.26;

Розділ опису типів починається з службового слова **Type**

Type ім'я типу = значення типу;

Наприклад, Type answer = (No, Yes);

Розділ опису змінних починається з службового слова **Var**

Var ім'я змінної : тип;

Наприклад, Var x: real;

Розділ опису процедур і функцій мають структуру, подібну до основної програми.

Розділ операторів береться в операторні дужки вигляду: Begin (почати) і End (закінчити), при цьому після End ставиться крапка. Крапка – ознака кінця програми. У розділі операторів записується послідовність виконуваних операторів. Оператори відділяються один від одного символом «;».

```
begin
оператор 1;
оператор 2;
.....
оператор n
end.
```

3.6 Оператори

3.6.1 Оператор привласнення

Оператор привласнення – основний оператор будь-якої мови програмування. Він призначений для заміни поточного значення змінній новим значенням.

Форма **оператора привласнення** має вигляд:

Змінна := вираз;

При виконанні цього оператора значення **Виразу** обчислюється і привласнюється **Змінній**.

Наприклад,

$a := b+c;$

$w := \sin(\sqrt{t})/(s + \ln(v));$

$s := \text{'рядок'};$

3.6.2 Введення-виведення даних

Для завдання змінним їх числових значень можна використовувати оператор привласнення, наприклад: $A:=5;$ $B:=-0.12.$ Проте в цьому випадку програма стає не універсальною, оскільки виконується тільки при цих значеннях змінних. Для виконання програми при різних значеннях змінних використовується оператор введення READ.

Як тільки під час виконання програми зустрічається оператор READ, програма припиняє свою роботу і чекає, поки на клавіатурі будуть набрані потрібні дані і натиснута клавіша Enter.

Оператор введення має вигляд:

read (елемент1, елемент2, ...);

Після натиснення клавіші Enter, введені значення привласнюються змінним, імена яких вказані в операторі READ.

Числові значення повинні бути набрані в одному рядку і розділені пропусками.

Наприклад, змінним X, Y, Z необхідно привласнити значення в процесі виконання програми: $X:=5.25,$ $Y:=10,$ $Z:=-0.05.$ Оператор введення буде мати вигляд: read (x, y, z), а числові значення необхідно ввести наступним чином: 5.25 10 -0.05.

Допускається використання оператора введення

readln (елемент1, елемент2, ...);

який спочатку вводить значення змінних, а потім здійснює перехід на новий рядок.

Для виведення даних на екран дисплея використовується оператор виведення WRITE. Форма запису оператора має вигляд:

write (елемент1, елемент2, ...);

де елемент1, елемент2 і т.д. є в простому випадку або змінними, або рядком символів, які ув'язнені в апострофи.

Наприклад, оператор

write ('Значення В= ', B);

виводить на екран дисплея текст

Значення В=

а потім значення змінної В.

Для виведення цілих і дійсних чисел можна указувати формати в операторі WRITE. Формат указується через двокрапку після змінної. Для дійсних чисел формат складається з двох величин. Перша величина позначає загальне поле значення, яке виводиться, друге – поле дробової частини. Загальне поле включає знак числа, кількість цифр в цілій частині, крапку і кількість цифр в дробовій частині. Так, виведення значення Y відповідно формату WRITE(Y:5:2) означає, що на зображення всього значення Y відведено п'ять позицій, з них дві – на дробову частину.

Якщо формат відведений більше, ніж кількість позицій, займаних числом, то перед цілою частиною буде відведено відповідну кількість пропусків, а після дробової частини – відповідна кількість нулів.

Наприклад, оператор виведення для значення Y=1.76 має вигляд:

WRITE('Y=',Y:8:3);

Тоді інформація на екрані буде представлена наступним чином:

Y= ___ 1.760
 └───┘

8 позицій

Для виведення цілих чисел формат дробової частини не указується. Наприклад, необхідно вивести значення цілого числа N=125, то оператор

виведення буде мати вигляд: WRITE('N=',N:3). На зображення числа N відведено 3 позиції.

Якщо формат вказати більш ніж 3, наприклад WRITE('N=',N:5), то перед числом буде два пропуски: N= _ _ 125, а для від'ємного числа – один пропуск: N= _ - 125.

Допускається використання оператора виведення

writeln (елемент1, елемент2, ...);

який спочатку виводить значення змінних, а потім здійснює перехід на новий рядок.

Перед введенням даних рекомендується давати пояснюючий текст за допомогою оператора WRITE. Цим самим встановлюється діалог користувача і комп'ютера. Наприклад, для введення значень A, B, C краще вказати:

WRITE('Ввести значення A, B, C');

READ(A, B,C);

Таким чином, перед введенням числових значень на екрані з'явиться повідомлення: Ввести значення A, B, C після чого можна вводити значення.

Приклад.

```
Program rabota;
```

```
Var x,a,b:real;
```

```
Begin
```

```
Writeln('vvod a');
```

```
Readln(a);
```

```
x:=2*a; b:=x+a;
```

```
writeln('x=',x:5:2,' b=',b:5:2);
```

```
end.
```

Результат роботи програми:

```
vvod a
2.5
x= 5.00 b= 7.50
```

3.6.3 Умовний оператор If

Оператор If призначений для реалізації розгалуженого обчислювального процесу.

Умовний оператор **If ... Then ... Else** є простою формою перевірки умов і має наступний синтаксис:

If умова Then оператор 1 Else оператор 2;

Оператор 1 виконується в тому випадку, якщо **Умова** істинна, інакше виконується **Оператор 2**. Умова – це вираз логічного типу.

В умовному операторі допустиме використання складеного оператора. **Складений оператор** – це об'єднання декількох операторів в одну групу. Форма запису даного оператора:

If умова Then

Begin

оператор 1;

оператор 2;

....

оператор n

End

Else

Begin

оператор 1;

оператор 2;

....

оператор n

End;

У цій конструкції службові слова **Begin** і **End** називаються операторними дужками. Складений оператор використовується в тих випадках, коли за правилами мови програмування Pascal дозволяється використовувати один оператор, а програмістові для вирішення завдання необхідно виконати групу операторів.

Оскільки гілка **Else** є необов'язковою, умовний оператор може бути записаний в короткій формі:

If умова Then оператор;

або

If умова Then

Begin

оператор 1;

оператор 2;

....

оператор n

End;

Приклад.

Обчислити значення у використовуючи формулу:

$$y = \begin{cases} \frac{1}{2\sqrt{x}}, & x \geq 1 \\ \frac{1}{3\sqrt[3]{x}}, & 0 < x < 1 \\ \frac{1}{4x}, & \text{у інших випадках} \end{cases}$$

Текст програми рішення задачі має вигляд:

```
Program pr2;
```

```
Var x, y: real;
```

```
begin
```

```
writeln ('Enter x'); readln (x);
```

```
if x>=1 then y:=1/(2*sqrt(x))
```

```
else
```

```
If (x>0) and (x<1) then
```

```
y:=1/(3*exp(1/3*ln(x)))
```

```
else
```

```
y:=1/(4*x);
```

```
writeln ('y=', y:5:2);
```

```
end.
```

Результат роботи програми при різних значеннях змінної x:

<pre>Enter x 4 y= 0.25</pre>	<pre>Enter x 0.5 y= 0.42</pre>	<pre>Enter x -5 y=-0.05</pre>
------------------------------	--------------------------------	-------------------------------

3.6.4 Оператор безумовного переходу

У мові Pascal всі оператори виконуються послідовно один за іншим в тому порядку, як вони записані. Проте в практиці програмування завдань виникає необхідність порушення послідовності виконання операторів. Для цього призначений оператор переходу. Оператор переходу – це оператор, що передає управління в програмі на інший оператор, перед яким стоїть мітка, що вказана в операторі Goto.

Оператор безумовного переходу має вигляд:

Goto мітка;

Мітка – це довільний ідентифікатор (ім'я) або ціле число без знаку, яка повинна бути оголошена в розділі опису Label.

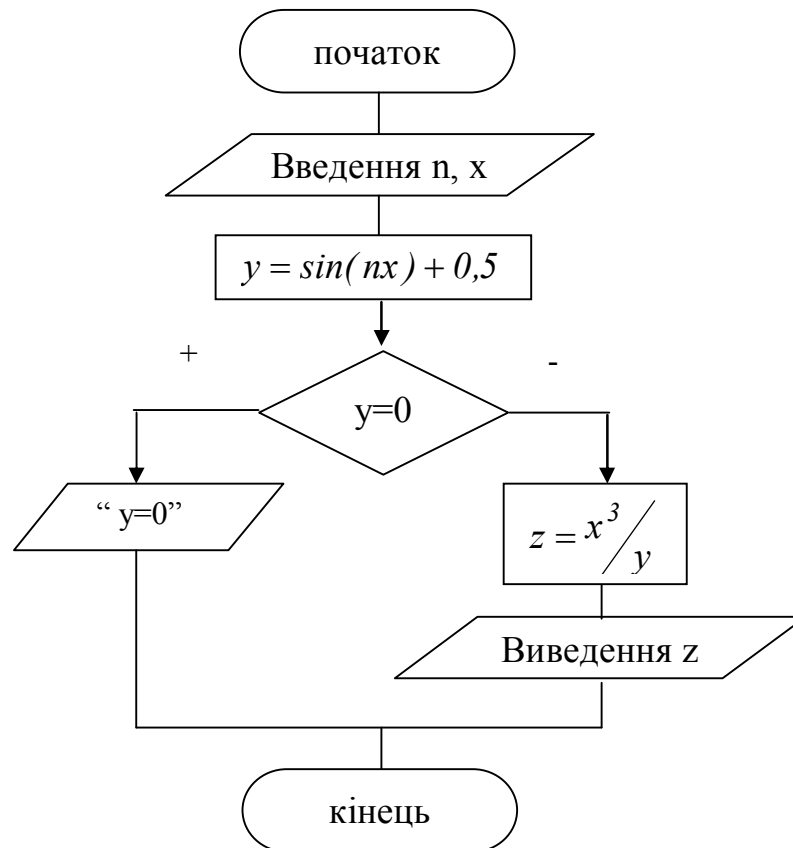
Оператор переходу слід використовувати у виняткових ситуаціях, оскільки він утрудняє розуміння програми, робить її заплутаною і складною у відладці.

Приклад використання умовного оператора і оператора безумовного переходу:

```
...
If x < 2 Then
begin  y := a + 2 * x ; Goto 1 end
Else y := sqr ( x );
b := sin(a)+ abs(x);
writeln ('b=', b:5:2);
1: writeln ('y=', y:5:2);
...
```

Приклад 1.

Обчислити значення $z = \frac{x^3}{y}$, де $y = \sin(nx) + 0,5$



Текст програми рішення задачі має вигляд:

1 варіант:

```
Program rabota;  
Label 1;  
var x, y, n, z: real;  
begin  
  writeln ('vvod x, n'); readln (x, n);  
  y:=sin(n*x)+0.5;  
  if y=0 then  
  begin writeln ('y=0'); goto 1 end  
  else z:=sqr(x)*x/y;  
  writeln('z=',z:5:2);  
  1: end.
```


2 варіант:

```
Program rabota;  
var x, y, n, z: real;  
begin  
  writeln ('vvod x, n'); readln (x, n);  
  y:=sin(n*x)+0.5;  
  if y=0 then writeln ('y=0')  
  else  
  begin  
    z:=sqr(x)*x/y; writeln('z=',z:5:2)  
  end  
end.
```

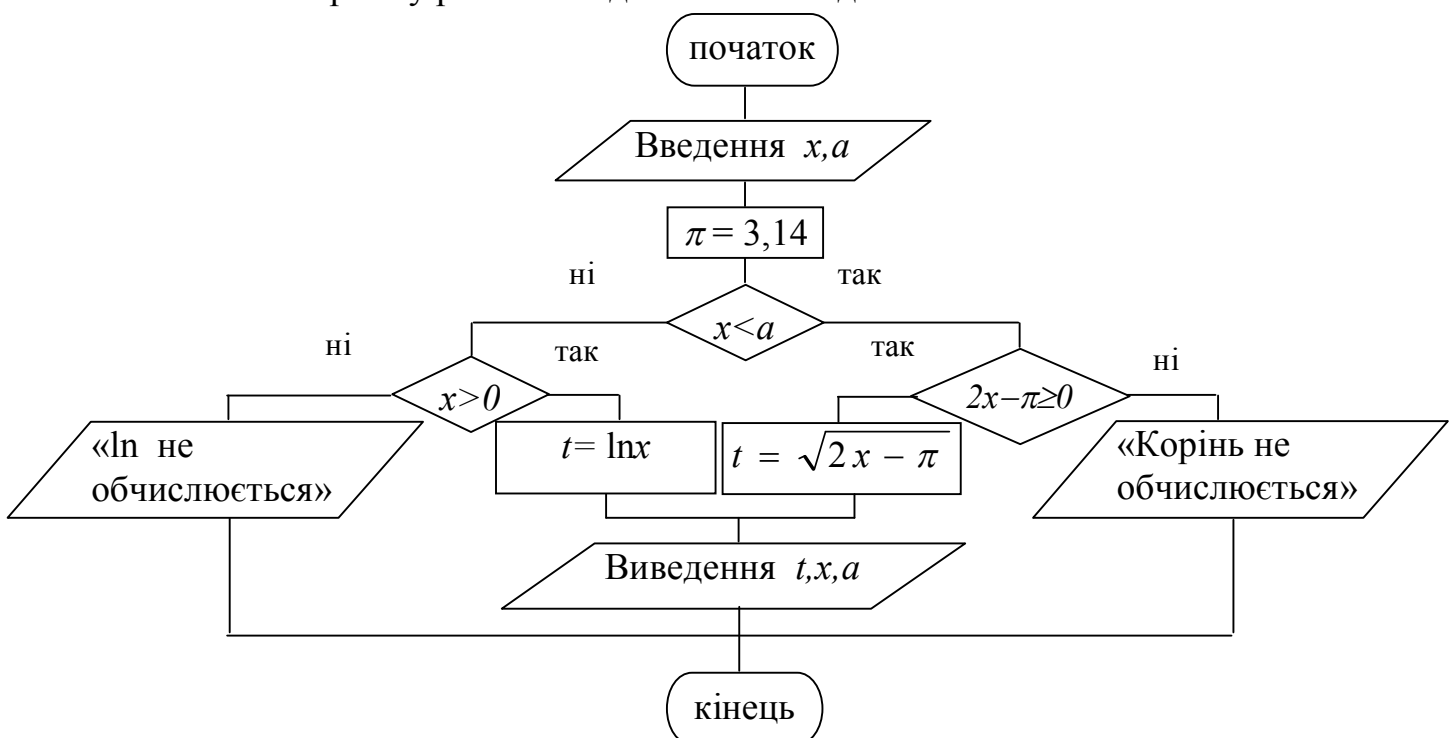
Приклад 2.

Обчислити значення t використовуючи формулу:

$$t = \begin{cases} \sqrt{2x - \pi}, & \text{якщо } x < a \\ \ln x, & \text{якщо } x \geq a \end{cases}$$

Початкові дані: x, a .

Схема алгоритму рішення задачі має вигляд:



Текст програми рішення задачі має вигляд:

```
Program rabota;
Label 1;
var x, a, t: real;
begin
writeln ('vvod x, a'); readln (x, a);
If x < a Then
    If  $2 * x - \pi \geq 0$  Then t: = Sqrt( $2 * x - \pi$ )
    Else begin writeln ('Корінь не обчислюється'); goto 1 end
Else
    If  $x > 0$  Then t: = Ln(x)
    Else begin writeln ('ln не обчислюється'); goto 1 end;
Writeln ('При x=', x:5:2, ' a=', a:5:2, ' t=', t:5:2);
1: end.
```

3.6.5 Оператор вибору

Оператор вибору використовується в тих випадках, коли залежно від значення якого-небудь виразу необхідно виконати один з декількох послідовних операторів. Оператор вибору відноситься до складних операторів і має наступну форму запису:

```
Case вираз of
список міток 1 : оператор 1;
список міток 2 : оператор 2;
.....
список міток N : оператор N;
else
оператор
end;
```

Вираз – це вираз порядкового типу, що визначає подальший хід виконання програми. Не може приймати значення дійсного числа або рядка.

Список міток є список констант, розділених комами; якщо використовується діапазон чисел – то указують першу і останню константи діапазону, розділені двома крапками.

Наприклад, список 1, 2, 3, 4 можна записати як список 1..4.

Оператор вибору працює таким чином. Обчислюється значення виразу. Потім значення виразу послідовно порівнюється з константами із списку міток, що стоять перед двокрапкою. Якщо значення виразу співпадає з константою з якого-небудь списку міток, то виконується відповідний цьому списку міток оператор. Якщо значення виразу не співпадає ні з однією константою зі всіх списків, то виконується оператор, розташований після ключового слова `else`.

Розділ `else` може бути відсутнім, тоді оператор буде мати вигляд:

```
Case вираз of  
список міток 1 : оператор 1;  
список міток 2 : оператор 2;  
.....  
список міток N : оператор N  
end;
```

Крім того, в операторові `Case` допустиме використання складеного оператора. Наприклад,

```
Case вираз of  
список міток 1 : begin оператор 1; оператор 2 end;  
список міток 2 : begin оператор 3; оператор 4; оператор 5 end;  
.....  
список міток N : оператор N  
end;
```

Приклад.

Скласти програму, що виводить на екран назву пори року, залежно від введеного номера місяця.

Оператора вибору Case можна записати у трьох варіантах:

1 варіант	2 варіант	3 варіант
<pre>case n of 1, 2, 12: s:='zima'; 3..5 : s:='vesna'; 6..8 : s:='leto'; 9..11 : s:= 'osen'; end;</pre>	<pre>case n of 3..5 : s:='vesna'; 6..8 : s:='leto'; 9..11 : s:= 'osen'; else s:='zima'; end;</pre>	<pre>case n of 1, 2, 12: s:='zima'; 3, 4, 5 : s:='vesna'; 6, 7, 8 : s:='leto'; 9, 10, 11 : s:= 'osen'; end;</pre>

Текст програми рішення задачі має вигляд:

```
Program pr;
Var n: integer; s: string;
Begin
writeln ('Enter n');
readln (n);
case n of
1, 2, 12: s:='zima';
3, 4, 5 : s:='vesna';
6, 7, 8 : s:='leto';
9, 10, 11 : s:= 'osen';
end;
writeln (s);
end.
```

Результат роботи програми, наприклад для n=12:



```
Enter n
12
zima
```

3.6.6 Оператори циклу

Циклічна структура програми дозволяє проводити багатократні обчислення групи операторів при зміні одного або декількох параметрів одночасно.

Розрізняють цикли з відомим числом повторень, коли значення параметра циклу змінюється від деякого початкового до деякого кінцевого значення з постійним кроком; а також цикли з невідомим числом повторень, в яких умова повторення або закінчення циклу задається по деякому проміжному результату, наприклад, поки не буде досягнута необхідна точність обчислень.

У мові Pascal є оператори циклу REPEAT, WHILE та FOR.

Оператор циклу з постумовою REPEAT використовується для організації циклу з невідомим числом повторень.

Оператор циклу Repeat має вигляд:

```
Repeat  
оператор 1;  
оператор 2;  
.....  
оператор N  
Until логічний вираз;
```

Оператор циклу виконується наступним чином. Спочатку виконується група операторів циклу, потім обчислюється значення логічного виразу. Якщо значення виразу має дійсне значення, то цикл припиняється, інакше виконується знов група операторів циклу.

Оператор циклу з передумовою WHILE використовується для організації циклу з невідомим числом повторень.

Оператор циклу While має вигляд:

```
While логічний вираз do оператор;
```

Оператор циклу виконується наступним чином. Обчислюється значення логічного виразу. Якщо значення логічного виразу має дійсне

значення, то виконується оператор, що стоїть після слова ключового слова do. Потім управління знов передається на початок оператора While, знову обчислюється логічний вираз і процес повторюється. Цикл припиняється якщо значення логічного виразу дорівнює false.

Якщо в циклі необхідно виконати декілька операторів, то оператор циклу While буде мати наступний вигляд:

```
While логічний вираз do  
begin  
оператор 1;  
оператор 2;  
.....  
оператор n  
end;
```

Оператор циклу FOR використовується для організації циклу з відомим числом повторень.

Існує два види запису оператора:

- при збільшенні значення параметра циклу

```
For i:= n1 to n2 do оператор;
```

- при зменшенні значення параметра циклу

```
For i:= n2 downto n1 do оператор;
```

де

i - параметр циклу;

n1 – початкове значення параметра циклу;

n2 – кінцеве значення параметра циклу.

Оператор циклу виконується наступним чином. Спочатку обчислюється початкове і кінцеве значення параметра циклу. Параметру привласнюється початкове значення. Далі значення параметра порівнюється з кінцевим значенням. Поки параметр менше або рівний кінцевому значенню (перший варіант) або більше або рівний кінцевому значенню (другий варіант), виконується оператор. Інакше відбувається вихід з циклу. Після

виконання оператора циклу, параметру циклу привласнюється наступне значення.

Якщо в циклі необхідно виконати групу операторів, то необхідно використовувати складений оператор:

```
For i := n1 to n2 do  
begin  
оператор 1;  
.....  
оператор n  
end;
```

3.7 Приклади програмування обчислювальних процесів

Приклад 1.

Обчислити значення Y використовуючи формулу:

$$y = \begin{cases} x + 4, & \text{якщо } x \geq 5 \\ x^2, & \text{якщо } x < 5 \end{cases} \quad -2 \leq x \leq 8 \quad \Delta x = 1$$

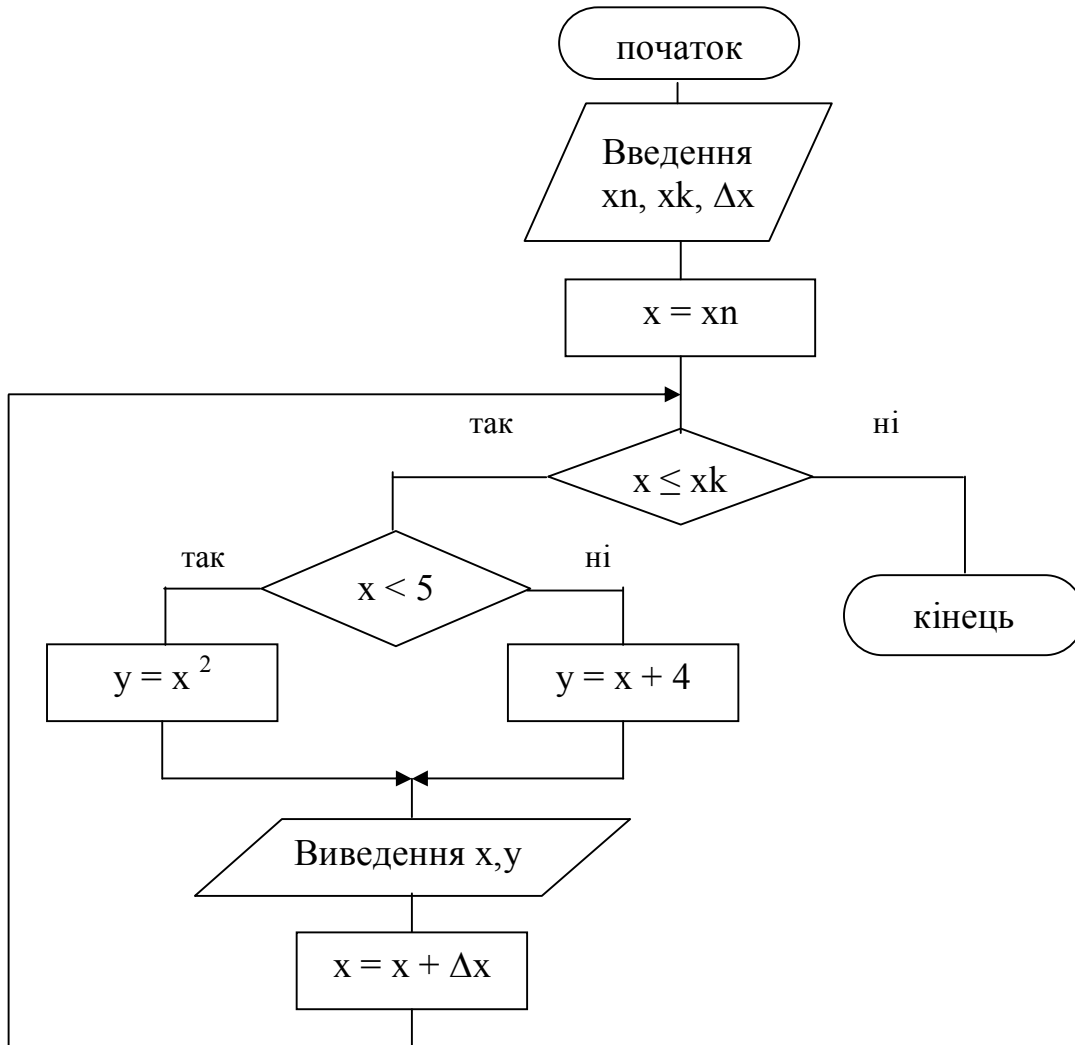
Початкові дані: $x_n, x_k, \Delta x$.

Для обчислення змінної у організувати цикл з передумовою.

Текст програми рішення задачі має вигляд:

```
Program pr1;  
Var  xn,xk,dx,x,y:real;  
begin  
writeln ('Vvod xn,xk,dx'); readln (xn,xk,dx);  
x := xn;  
while x <= xk do  
begin  
if x < 5 then y := sqr(x) else y := x + 4;  
writeln ('x =', x : 5 : 3, ' y =', y : 5 : 3);  
x := x + dx  
end; end.
```

Схема алгоритму рішення задачі має вигляд:



Приклад 2.

Обчислити значення X використовуючи формулу:

$$x = \frac{2,5a - c}{a^2 - 2ac}$$

$$-3 \leq a \leq 3; \quad \Delta a = 1,5$$

Початкові дані: $a_n, a_k, \Delta a, c$.

Для обчислення змінної X організувати цикл з постумовою.

Текст програми рішення задачі має вигляд:

Program pr2;

Label 1;

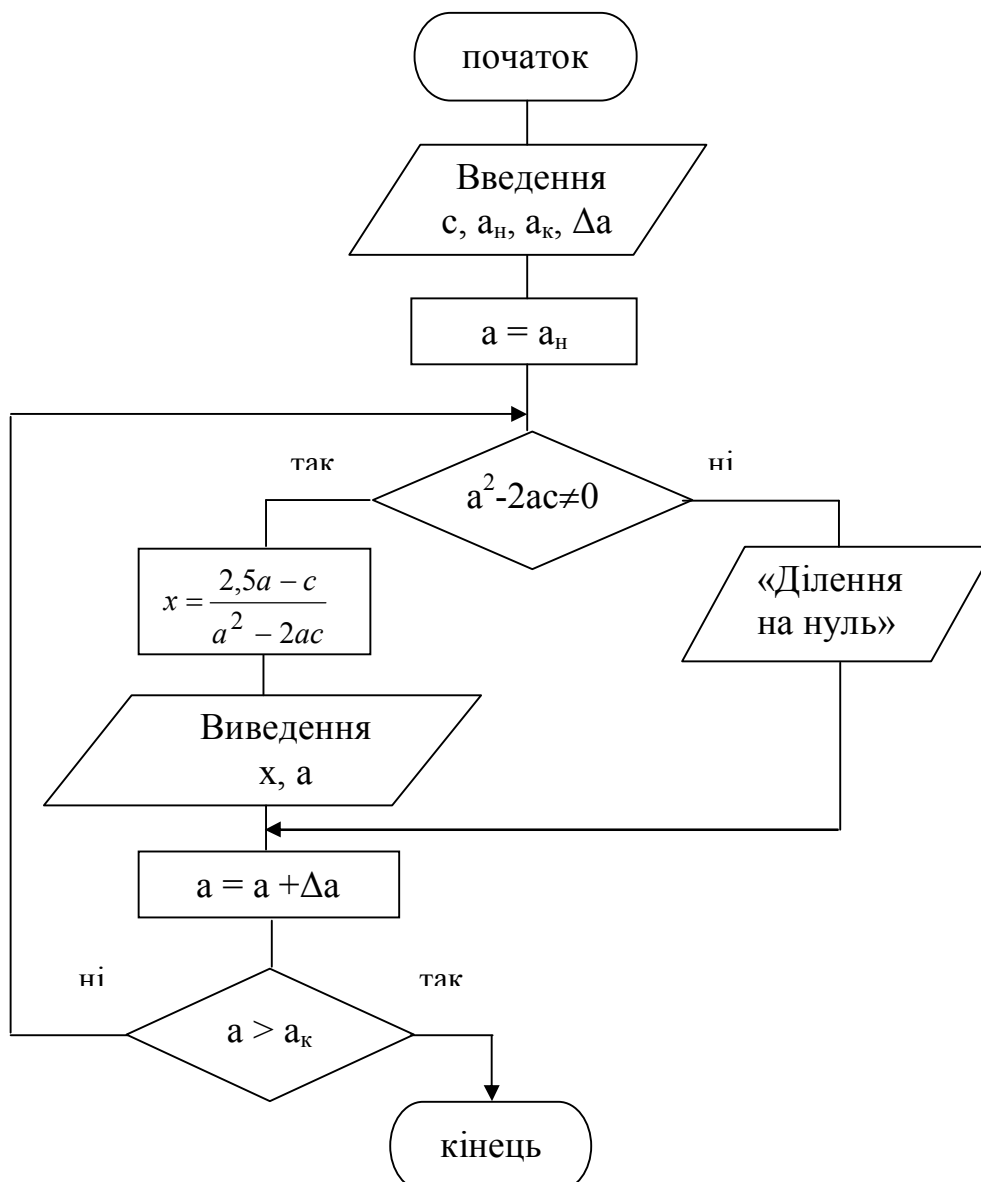
Var an,ak,da,a,c,x:real;


```

begin
writeln ('Vvod an,ak,da,c'); readln (an,ak,da,c);
a := an;
repeat
    if  $\text{sqr}(a) - 2 * a * c <> 0$  then  $x := (2.5 * a - c) / (\text{sqr}(a) - 2 * a * c)$ 
    else
        begin writeln ('a =', a : 5 : 3, ' ділення на нуль'); goto 1 end;
        writeln ('a =', a : 5 : 3, ' x =', x : 5 : 3);
    1: a := a + da
until a > ak;
end.

```

Схема алгоритму рішення задачі має вигляд:



Приклад 3.

Обчислити
$$\sum_{i=1}^n \frac{1}{i^2} = 1 + \frac{1}{4} + \frac{1}{9} + \dots + \frac{1}{n^2}$$

Використовується алгоритм накопичення суми: $S = S + 1/i^2$, де $i = 1, 2, 3, \dots, n$.

Якщо позначити $a_i = 1/i^2$, тоді $S = S + a_i$.

Для обчислення організувати цикл з відомим числом повторень.

Текст програми рішення задачі має вигляд:

```
Program pr3;
Var i,n: integer; s,ai: real;
begin
write ('Vvod n'); readln (n);
s := 0;
for i := 1 to n do
begin ai := 1/sqr(i); s := s + ai end;
writeln ('s =', s : 5 : 3);
end.
```

Приклад 4.

Обчислити
$$C_n^m = \frac{n!}{m!(n-m)!}$$

Факторіал обчислюється за формулою: $k! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot k = \prod_{i=1}^k i$

Позначення для рішення задачі: $n!$ – fact1; $m!$ – fact2; $(n-m)!$ – fact3; C_n^m – C.

Для обчислення організувати цикл з відомим числом повторень.

Текст програми рішення задачі має вигляд:

```
Program pr4;
Var i,n,m,c,fact1,fact2,fact3: integer;
begin
writeln ('Vvod n, m'); readln (n, m);
```

```

fact1 :=1; for i:= 2 to n do fact1 := fact1 * i;
fact2 := 1; for i:= 2 to m do fact2 := fact2 * i;
fact3 := 1; for i:= 2 to n - m do fact3 := fact3 * i;
c := round (fact1 / (fact2 * fact3));
writeln ('c =', c);
end.

```

Приклад 5.

Обчислити значення Y використовуючи формулу:

$$y = \frac{\sin(ax) + 2}{1 - x}$$

$$-5 \leq x \leq 5 \quad \Delta x = 1$$

Визначити $P = \prod_{y > 0} y$, $S = \sum y$ та кількість $y < 0.3$

Початкові дані: x_n , x_k , Δx , a .

Для обчислення змінної Y організувати цикл з відомим числом повторень.

Текст програми рішення задачі має вигляд:

```

Program pr5;
label 1;
var xn,xk,dx,x,a,s,p,y : real; i,m,k : integer;
begin
writeln ('Vvod xn,xk,dx,a'); readln (xn,xk,dx,a);
k := 0; s := 0; p := 1; x := xn;
m := trunc ((xk-xn) / dx) + 1;
for i := 1 to m do
begin
if 1 - x <> 0 then y := (sin (a * x) + 2) / (1 - x)
else
begin writeln ('x =', x : 5 : 3, ' delenie na 0'); goto 1 end;
if y < 0.3 then k := k + 1;

```

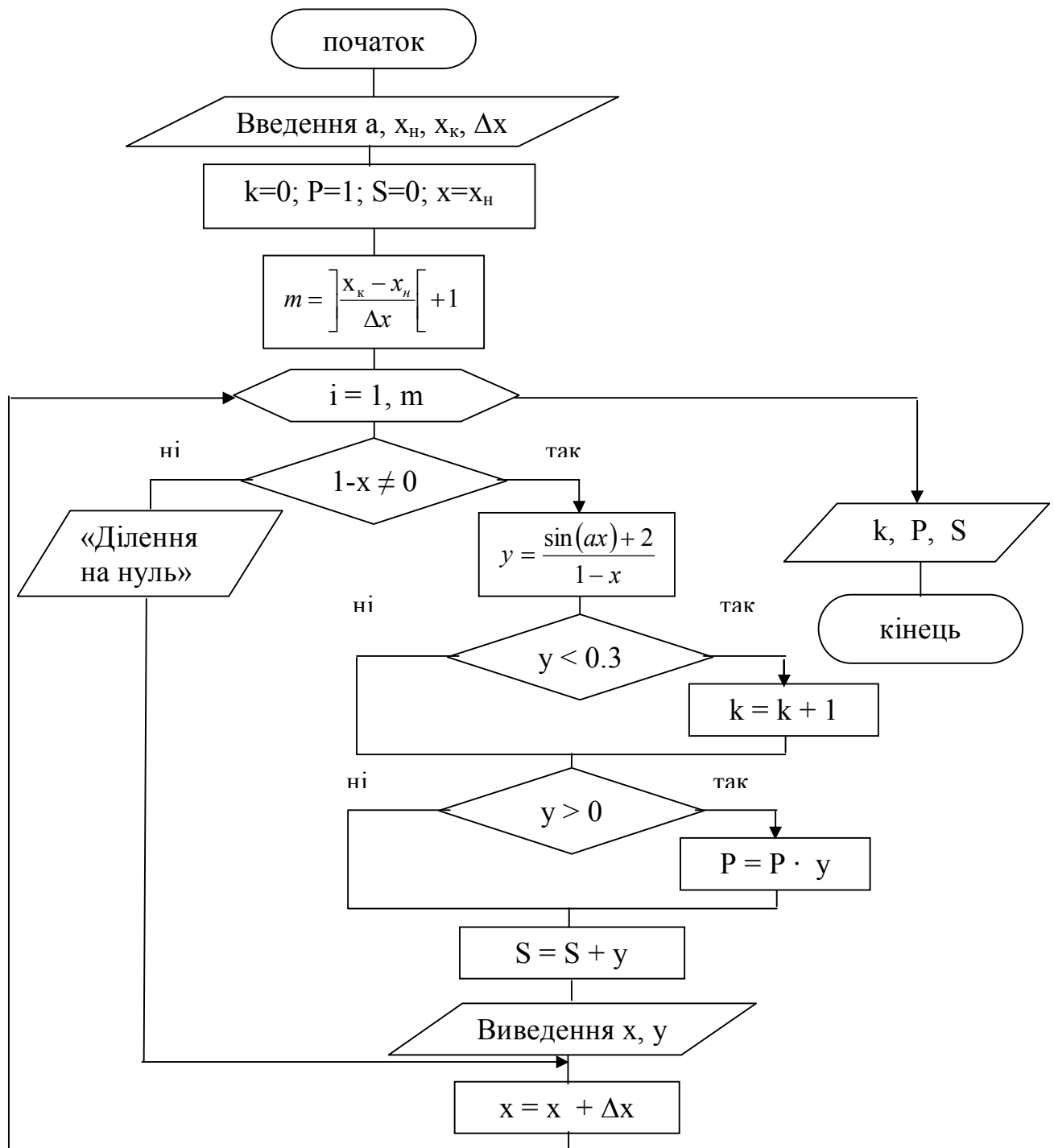
```

if y > 0 then p := p * y;
s := s + y;
writeln ('x =', x : 5 : 3, ' y =', y : 5 : 3);
1: x := x + dx;

end;
writeln ('k =', k : 3, ' s =', s : 5 : 3, ' p =', p : 5 : 3);
end.

```

Схема алгоритму рішення задачі має вигляд:



Приклад 6.

Обчислити значення X та Y використовуючи формули та вивести результати у формі таблиці:

$$y = \frac{x^2 - ac}{c^2 - 1} \quad x = \sqrt{|b^2 - a|}$$
$$2 \leq a \leq 7 \quad \Delta a = 1$$
$$-1,5 \leq c \leq 2 \quad \Delta c = 0,5$$
$$b = 3,2$$

Початкові дані: a_n , a_k , Δa , c_n , c_k , Δc , b .

Для організації зовнішнього і внутрішнього циклів використовуватиме цикли з відомим числом повторень.

Виведення значень Y оформити у вигляді таблиці:

$a \setminus c$	c_1	c_2	...	c_n
a_1				
...				
a_m				

Текст програми рішення задачі має вигляд:

```
program pr6;  
var b, an, ak, da, cn, ck, dc, c, a, x, y : real;  
m, n, i, j: integer;  
begin  
  writeln('Enter b,an,ak,da,cn,ck,dc');  
  readln(b,an,ak,da,cn,ck,dc);  
  n := trunc ((ck-cn) / dc) + 1;  
  m := trunc ((ak-an) / da) + 1;  
  c := cn; write ('a \ c');  
  for i := 1 to n do  
    begin write('  : 4, c : 5 : 3); c := c + dc  end;  
  writeln;  
  a := an;
```

```

for j := 1 to m do
  begin
  write (a : 5 : 3);
  c := cn;
  for i := 1 to n do
    begin
    x := sqrt (abs (sqr(b) - a));
    if sqr(c) - 1 = 0 then write ( ' *** ' )
    else
      begin
      y := (sqr(x) - a * c) / (sqr(c) - 1);
      write ( ' ' : 2, y : 5 : 3, ' ' : 2)
      end;
      c := c + dc
      end;
    a := a + da;
  writeln;
  end;
end.

```

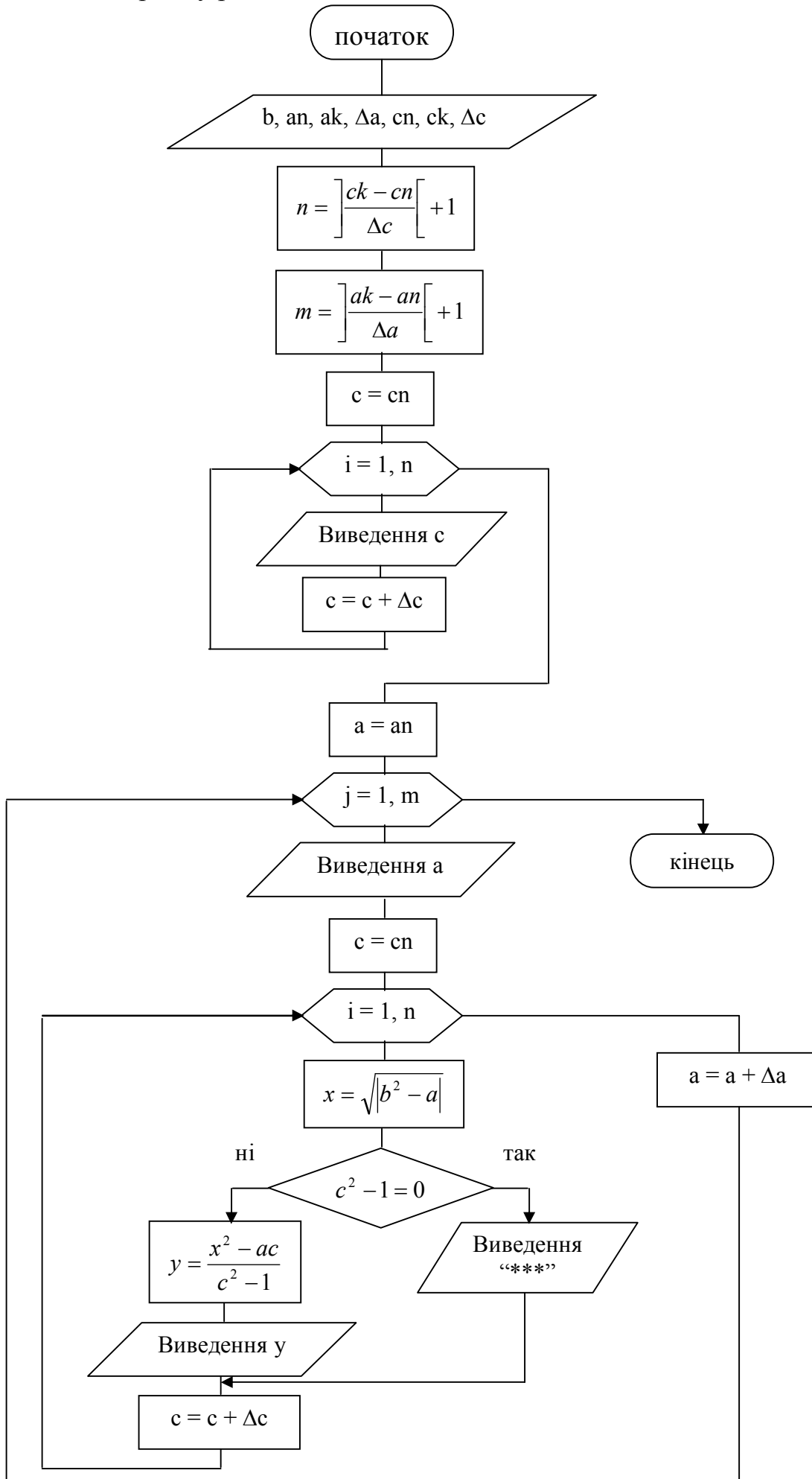
Результат роботи програми:

```

Enter b,an,ak,da,cn,ck,dc
3.2 2 7 1 -1.5 2 0.5
a\c  -1.500  -1.000  -0.500  0.000  0.500  1.000  1.500  2.000
2.00  8.99  ***  -12.32  -8.24  -9.65  ***  4.19  1.41
3.00  9.39  ***  -11.65  -7.24  -7.65  ***  2.19  0.41
4.00  9.79  ***  -10.99  -6.24  -5.65  ***  0.19  -0.59
5.00  10.19 ***  -10.32  -5.24  -3.65  ***  -1.81  -1.59
6.00  10.59 ***  -9.65  -4.24  -1.65  ***  -3.81  -2.59
7.00  10.99 ***  -8.99  -3.24  0.35  ***  -5.81  -3.59

```

Схема алгоритму рішення задачі має вигляд:



3.8 Обробка та формування масивів

Масив – це структурований тип даних, що складається з фіксованого числа елементів. Масив позначається одним ім'ям. Елементами масиву можуть бути дані будь-якого типу. Так, сукупність дійсних чисел

1.6 14.9 -5.2 0.45

можна вважати масивом і позначити одним ім'ям, наприклад А:

$A = (1.6 \quad 14.9 \quad -5.2 \quad 0.45)$

Всі елементи масиву пронумеровані. Номер елементу масиву називається **індексом**. Доступ до кожного елементу масиву здійснюється за допомогою індексів. Кожен елемент масиву позначається ім'ям масиву з індексом, наприклад $A[1]=1.6$; $A[2]=14.9$. Масив, що є простим списком даних одного і того ж типу, називають **одномірним масивом**.

Якщо виникає необхідність зберігання даних у вигляді таблиць, у форматі рядків і стовпців, необхідно використовувати багатовимірні масиви. Особливо широкого поширення набули **двомірні масиви**, які мають назву - матриця. Наприклад, зображення цілих чисел послідовне в декількох рядках є матрицею

$$B = \begin{pmatrix} 2.3 & -2.7 & 7.4 \\ -0.4 & 3.2 & -4.9 \\ 12.7 & -1.4 & 0.8 \end{pmatrix}$$

Для доступу до даних, що зберігаються в цьому масиві, необхідно вказати ім'я масиву і два індекси, перший відповідає номеру рядка, а другий – номеру стовпця в якому зберігається необхідний елемент.

Наприклад, $B[1,2]=-2.7$; $B[3,1]=12.7$.

Якщо в програмі використовується масив, то він повинен бути описаний або в розділі змінних VAR, або в розділі типів TYPE.

1 варіант

Var ім'я масиву: **array** [тип індексу] of тип елементів масиву;

Наприклад,

```
Var A: array [1..4] of real;    //A – це одномірний масив A, що  
                               складається з 4 дійсних чисел.
```

```
    B:array[1..3,1..3] of real; // B - це двомірний масив розміром 3  
                               на 3, що складається з дійсних чисел.
```

2 варіант

Type ім'я типу = array [тип індексу] of тип елементів масиву;

Var ім'я масиву: ім'я типу;

Наприклад,

```
Type mas1= array [1..4] of real;  
    mas2= array[1..3,1..3] of real;  
Var A: mas1;  
    B: mas2;
```

Для введення і виведення числових значень масиву використовуються цикли.

Введення і виведення елементів одномірного масиву A(n)

Введення елементів масиву

```
Write('Vvod n');  
Readln(n);  
for i := 1 to n do read(a[i]);
```

Виведення елементів масиву

```
for i := 1 to n do write(a[i]);
```

Введення і виведення елементів двомірного масиву B(n,m)

Введення елементів масиву

```
Write('Vvod n,m');  
Readln(n,m);  
for i := 1 to n do  
for j := 1 to m do read(b[i,j]);
```

Виведення елементів масиву

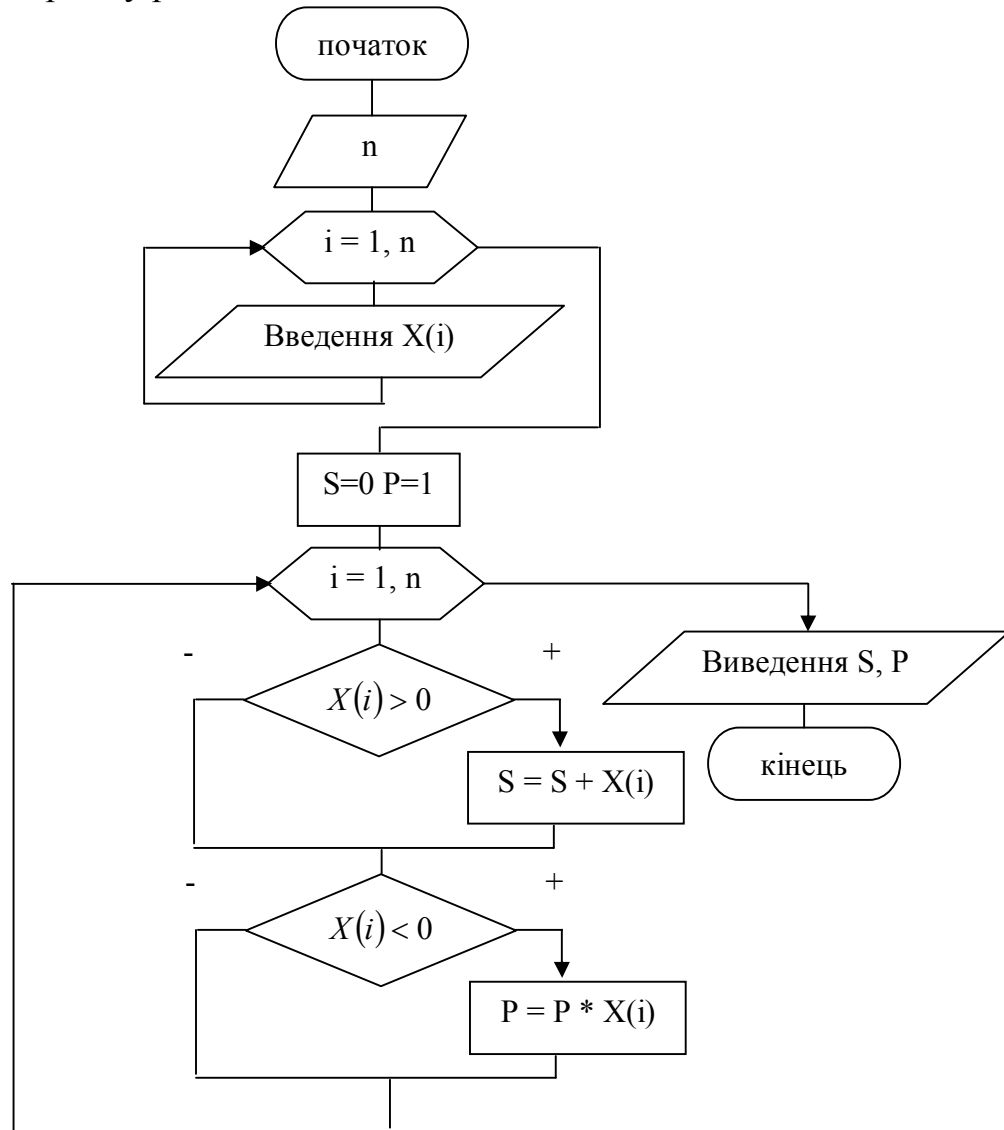
```
for i := 1 to n do  
for j := 1 to m do  
write(b[i,j]);
```

Приклад 1.

Визначити суму додатних і добуток від'ємних елементів масиву $X(N)$.

Початкові дані: масив $X(N)$.

Схема алгоритму рішення задачі має вигляд:



Текст програми рішення задачі має вигляд:

```
Program pr1;
```

```
var x: array [1 .. 10] of real; i,n: integer; s,p: real;
```

```
begin
```

```
writeln('Enter n'); readln (n);
```

```
writeln('vvod massiva');
```

```
for i := 1 to n do read (x [i]);
```

```
s := 0; p := 1;
```

```

for i := 1 to n do
    begin
        if x [i] > 0 then s := s + x [i];
        if x [i] < 0 then p := p * x [i]
    end;
writeln('S =', S : 5 : 3, ' P =', P : 5 : 3);
end.

```

Результат роботи програми:

```

enter n
10
vovod massiv
2 4 -3 -0.6 3.1 6.8 0.3 -5.9 -4.1 0.2
s=16.4000 p=43.542

```

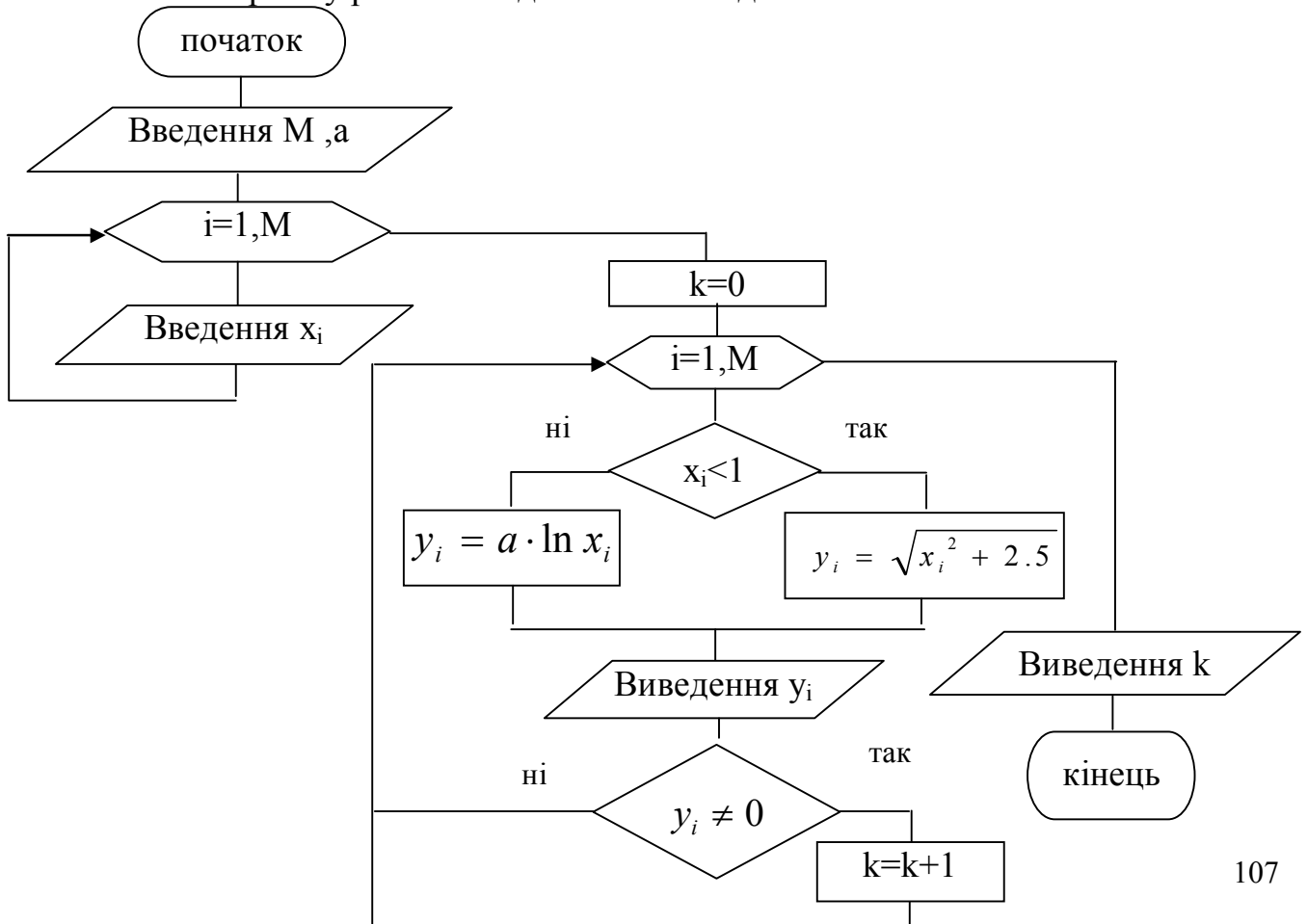
Приклад 2.

Визначити кількість ненульових елементів в масиві Y, елементи якого обчислюються за формулами:

$$y_i = \begin{cases} a \ln x_i, & \text{якщо } x_i \geq 1 \\ \sqrt{x_i^2 + 2,5}, & \text{якщо } x_i < 1 \end{cases} \quad \text{де } i = 1, M$$

Початкові дані: масив X(M), a.

Схема алгоритму рішення задачі має вигляд:



Текст програми рішення задачі має вигляд:

```
Program pr2;
type mas = array [1 .. 10] of real;
var x,y: mas;  i,m,k: integer;  a: real;
begin
writeln('Vvod m,a');  readln (m,a);
writeln(' vvod  elementiv massiva X');
for i := 1 to m do  read (x [i] );
k := 0;
for i := 1 to m do
    begin
        if x [i] < 1 then y[i] := sqrt (sqr (x[i]) + 2.5) else y[i] := a * ln (x[i]);
        writeln('y[', i, ']=', y[i] : 5 : 3);
        if y [i] <> 0 then k := k + 1;
        end;
writeln('k =', k: 3);
end.
```

Приклад 3.

Елементи масиву $X(N)$, які більше A записати в масив Y . Визначити мінімальне значення у масиві Y та його номер.

Початкові дані: масив $X(N)$, A .

Текст програми рішення задачі має вигляд:

```
Program pr3;
type mas = array [1 .. 10] of real;
var x,y: mas;  a, min: real;  i, k, n, nmin: integer;
begin
writeln('Enter a, n');  readln(a,n);
```

```

for i := 1 to n do
begin
write ('x [ , i , ' ]=' ); readln (x [i])
end;
k:=0;
for i := 1 to n do
if x [i] > a then
begin
k := k + 1;
y[k] := x[i];
writeln ('y[ , k, ' ]=', y[k] : 5 : 2);
end;
min := y [1]; nmin := 1;
for i := 2 to k do
if y [i] < min then
begin min := y [i]; nmin := i end;
writeln ('min =', min : 5 : 2, '    nmim = ', nmin : 2);
end.

```

} Введення елементів масиву
в стовбець

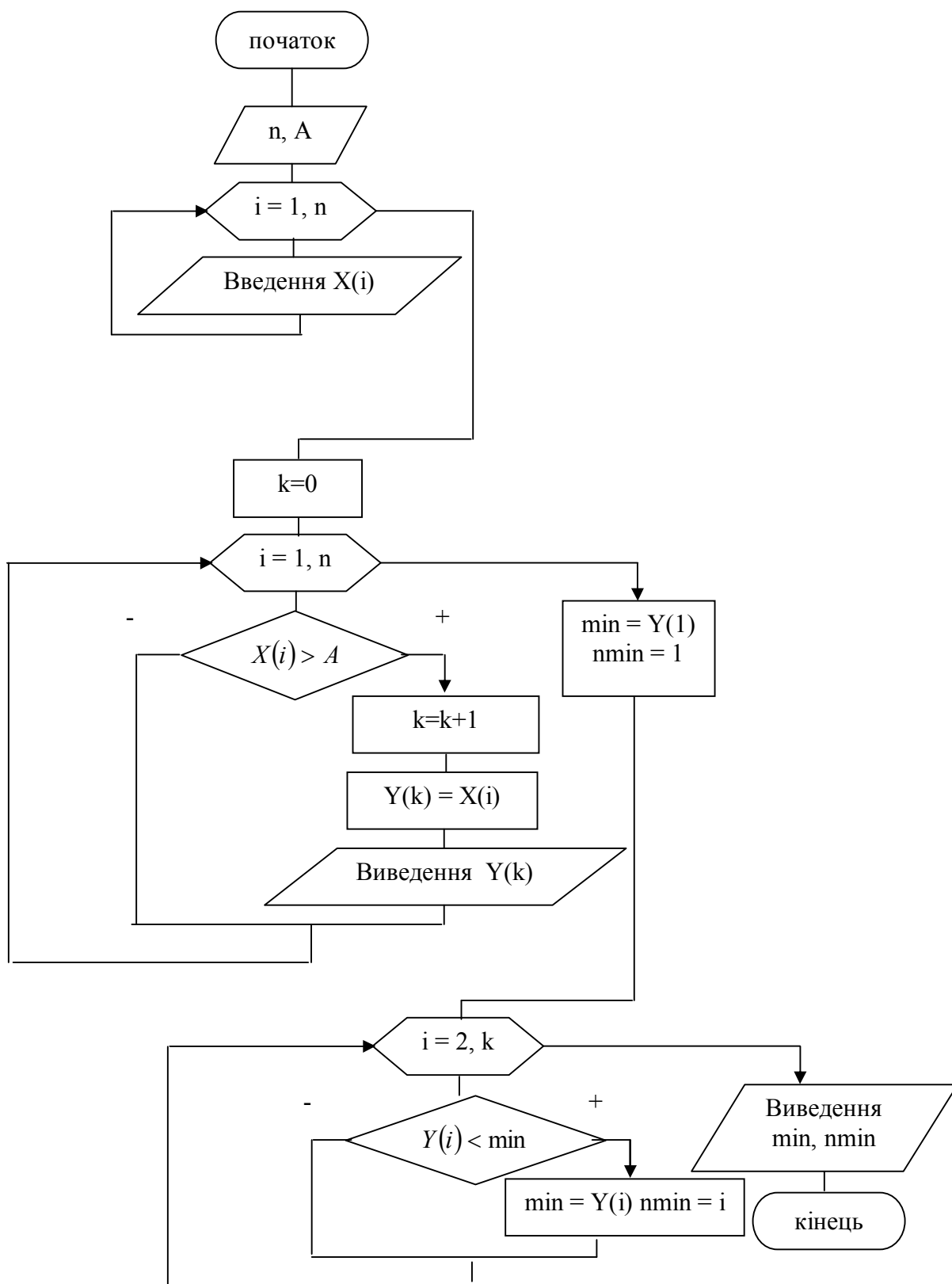
Результат роботи програми:

```

Enter a,n
2.5 10
X[1]=2.8
X[2]=-0.3
X[3]=1.2
X[4]=6.2
X[5]=7.9
X[6]=-4.6
X[7]=12.5
X[8]=-4.12
X[9]=2.6
X[10]=11.5
Y[1]= 2.80
Y[2]= 6.20
Y[3]= 7.90
Y[4]=12.50
Y[5]= 2.60
Y[6]=11.50
min= 2.60    nmin= 5

```

Схема алгоритму рішення задачі має вигляд:



Приклад 4.

Елементи масиву $X(N)$ упорядкувати по убутанню.

Початкові дані: масив $X(N)$.

Текст програми рішення задачі має вигляд:

```
Program pr4;
type mas=array[1..20] of real;
var x:mas; n, i, j: integer; a:real;
begin
writeln('Enter n'); readln(n);
for i:=1 to n do
    begin
        write('X[', i, ']='); readln(x[i])
        end;
for i:=1 to n-1 do
for j:=i to n do
    if x[j]>x[i] then
        begin
            a:=x[j]; x[j]:=x[i]; x[i]:=a
        end;
writeln;
for i:=1 to n do
writeln('X[', i, ']=', x[i]:5:2);
end.
```

Результат роботи програми:

```
Enter n
10
X[1]=3.9
X[2]=-2.1
X[3]=0.5
X[4]=-2.7
X[5]=12.8
X[6]=5.9
X[7]=2.6
X[8]=-0.8
X[9]=-2.4
X[10]=10.7
```

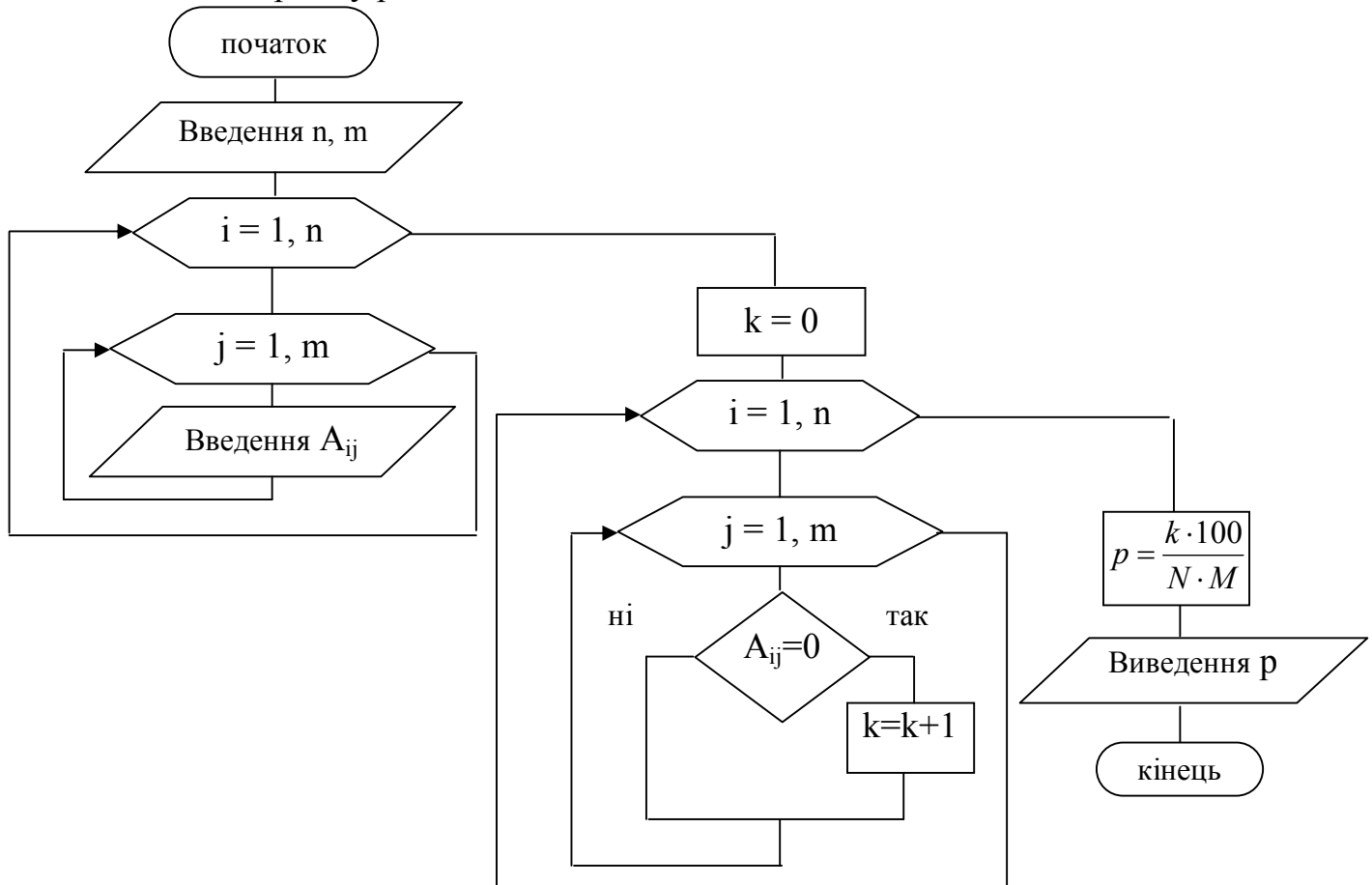
```
X[1]=12.80
X[2]=10.70
X[3]= 5.90
X[4]= 3.90
X[5]= 2.60
X[6]= 0.50
X[7]=-0.80
X[8]=-2.10
X[9]=-2.40
X[10]=-2.70
```

Приклад 5.

Для масиву $A(n,m)$ знайти відсоток нульових елементів.

Початкові дані: масив $A(n,m)$.

Схема алгоритму рішення задачі має вигляд:



Текст програми рішення задачі має вигляд:

```
program pr5;
type mas1=array[1..10,1..10] of integer;
var a:mas1; i,j,n,m,k: integer; p: real;
begin
writeln('vvod n,m'); readln(n,m);
writeln('vvod massiva A');
for i:=1 to n do
for j:=1 to m do
read (a[i,j]);
k:=0;
```



```

for i:=1 to n do
for j:=1 to m do
if a[i,j] =0 then k:=k+1;
p:=k*100/(n*m);
writeln('p=', p:5:2);
end.

```

Приклад 6.

У матриці $A(3,3)$ поміняти місцями елементи першого і останнього рядків.

Рішення. У елементів першого рядка відомим є індекс рядка - $a[1, j]$. Аналогічно, для останнього (третього) рядка - $a[3, j]$. Для того, щоб поміняти місцями елементи першого та третього рядків, які розташовані в одному стовпці, необхідно використовувати третю змінну як буфер обміну.

Текст програми рішення задачі має вигляд:

```

Program pr6;
Const n=3;
Var a: array [1..3, 1..3] of integer; c, i, j: integer;
begin
writeln('Enter massiv');
for i:=1 to n do
for j:=1 to n do
read (a[i,j]);
for j:=1 to n do
begin c:= a[1, j]; a[1, j]:= a[3, j]; a[3, j]:= c; end;
writeln ('Result massiv');
for i:=1 to n do
begin
for j:=1 to n do write(a[i,j]:3);
writeln
end; end.

```

```

Enter massiv
-2 3 12
11 0 -1
-6 4 -9
Result massiv
-6 4 -9
11 0 -1
-2 3 12

```

Виведення
елементів масиву
у вигляді матриці

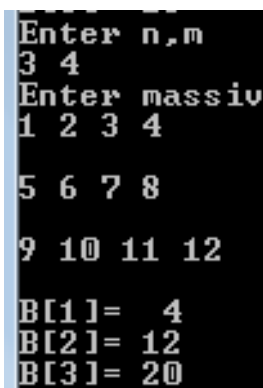
Приклад 7.

З масиву $A(n,m)$ сформувати масив $B(n)$, кожен елемент якого рівний сумі непарних елементів відповідного рядка матриці A .

Текст програми рішення задачі має вигляд:

```
program pr7;
type mas1=array[1..10,1..10] of integer;
      mas2=array[1..10] of integer;
var a:mas1; b:mas2; i,j,n,m,s: integer;
begin
writeln('Enter n,m');  readln(n,m);
writeln('Enter massiv');
for i:=1 to n do
for j:=1 to m do
read (a[i,j]);
for i:=1 to n do
begin
s:=0;
for j:=1 to m do
if a[i,j] mod 2 <>0 then s:=s+a[i,j];
b[i]:=s;  writeln('B[', i]='',b[i]:3);
end; end.
```

Результат роботи програми:



```
Enter n,m
3 4
Enter massiv
1 2 3 4
5 6 7 8
9 10 11 12
B[1]= 4
B[2]= 12
B[3]= 20
```

3.9 Загальні відомості про підпрограми

Частина програми, що реалізовує певний алгоритм і що допускає звернення до неї з різних частин основної програми, називається підпрограмою.

Підпрограми оформляються у вигляді замкнутих ділянок програми, що мають чітко позначений вхід і вихід.

У мові програмування Pascal виділяють два види підпрограм: процедуру (PROCEDURE) і функцію (FUNCTION).

Будь-яка програма може містити декілька процедур і функцій. Процедури і функції оголошуються в розділі опису услід за розділом змінних.

Приклад структури програми з двома підпрограмами.

```
Program ім'я;  
Label ..;  
Const...;  
Type...;  
Var...;  
Підпрограма P1;  
Підпрограма P2;  
Begin  
Оператори;  
Виклик підпрограми P1;  
Оператори;  
Виклик підпрограми P1;  
Оператори;  
Виклик підпрограми P2;  
Оператори;  
Виклик підпрограми P2;  
Оператори;  
End.
```

Виконання програми починається з операторів основної програми. При необхідності викликається підпрограма, і починають виконуватися її оператори. Потім управління передається в основну програму, яка продовжує виконуватися. Підпрограма оформляється аналогічно основній програмі, тобто складається із заголовка, розділу описів і розділу операторів.

3.9.1 Процедури

Структура процедури має вигляд:

```
Procedure ім'я (список формальних параметрів);  
Розділ описів процедури  
Begin  
Оператор 1;  
Оператор 2;  
...  
Оператор n  
End;
```

У заголовку вказується службове слово Procedure, за яким слідує ім'я процедури і формальні параметри в круглих дужках. Розділ описів процедури подібний до програми і складається з розділів міток, констант, типів, змінних і, у свою чергу, процедур і функцій. Розділ операторів розташовується в операторних дужках Begin... End.

Процедура викликається по її імені:

Ім'я (список фактичних параметрів);

За допомогою формальних і фактичних параметрів дані передаються з програми в процедуру і, навпаки, з процедури в програму. Кожен формальний параметр вказується разом з своїм типом. Відповідний йому фактичний параметр вказується без типу. Між формальними і фактичними параметрами повинна бути відповідність по кількості параметрів, за їх типом і порядком слідування.

Якщо декілька формальних параметрів мають однаковий тип, тоді їх можна об'єднати в список, тобто перерахувати параметри через кому, а потім вказати тип. Наприклад, заголовок процедури має вигляд:

```
Procedure Summa (a,b,c:integer);    де a,b,c – формальні параметри.
```

Викликати процедуру можна таким чином:

```
Summa (5, m, 7);    де 5, m, 7 – фактичні параметри.
```

Кожен формальний параметр приймає значення відповідного фактичного параметра. Таким чином, в процедурі матимемо наступні значення: a=5; b= m; c=7.

Серед параметрів процедури виділяють параметри-значення і параметри-змінні. Параметри-значення виконують роль вхідних параметрів процедури. Вони можуть приймати значення фактичних параметрів, але не можуть передавати свої значення фактичним параметрам. Параметри-змінні виконують роль як вхідних, так і вихідних параметрів процедури. Вони можуть приймати значення фактичних параметрів, змінювати їх в процедурі і повертати нові значення фактичним параметрам. Для виділення параметрів-змінних перед ними ставиться слово `Var` у формальних параметрах.

Приклад процедури.

```
Procedure calc (a,b,c: real; Var x,y: real);  
  Var z: real;  
  Begin  
    z:= a+b+c;  x:=sqr(z);  y:=sqrt(z)  
  End;
```

Цю процедуру можна викликати таким чином: `calc (25, 44, 30, x1, y1)`.

Результатом виконання процедури є `X,Y`, які передають свої значення відповідним фактичним параметрам `X1,Y1`.

Якщо в процедуру потрібно передати як параметр не просто одне значення, а масив, то фактичним параметром повинне бути ім'я масиву. При цьому формальний параметр указується після слова `Var` разом з типом масиву. Сам же опис масиву робиться в розділі `Type` основної програми.

Наприклад, в основній програмі є опис масиву `A`:

```
Const n=10;  
Type mas=array[1..n] of real;  
Var a:mas;
```

Тоді формальні параметри процедури `Prim` можуть мати вигляд:
`Procedure Prim (k:integer; Var x:mas)` при наступних фактичних параметрах:
`Prim (n, a)`.

Формальний параметр `k` приймає значення фактичного параметра `n`, а формальний параметр `x` – значення масиву `a`.

Приклад 1.

```
Program primer1;  
  Var a, b, c, d : real;      S1, S2, S3 : real;  
  Procedure lab (x, y, f : real; Var S : real);  
    Var k : real;  
    Begin k := x + y / 2;  S := 2 * k + f  End;  
Begin  
  Writeln ('a, b, c, d'); Readln (a, b, c, d);  
  Lab (a, b, c, S1);  Lab (b, d, a, S2);  
  S3 := S1 + S2;  Writeln('S3 =', S3:5:2);  
End.
```

Допускається використання процедур без параметрів. В цьому випадку формальні і фактичні параметри відсутні. В цьому випадку передача початкових даних в процедуру і вихідних даних з процедури в основну програму здійснюється за допомогою глобальних параметрів. А саме, початкові дані і вихідні дані процедури описуються як глобальні в розділі змінних основної програми. Оператор виклику процедури в цьому випадку складатиметься тільки з імені процедури без фактичних параметрів.

Приклад 2.

```
Program primer2;  
  Var a, b, c, d, x, y, f, S, S1, S2, S3 : real;  
  Procedure lab;  
    Var k : real;  
    Begin k := x + y / 2;  S := 2 * k + f  End;  
Begin  
  Writeln ('a, b, c, d');  Readln (a, b, c, d);  
  x := a; y := b; f := c;  Lab;  S1 := S;  
  x := b; y := d; f := a;  Lab;  S2 := S;  
  S3 := S1 + S2; Writeln ('S3 =', S3:5:2);  
End.
```

3.9.2 Функції

Структура функції має вигляд:

Function ім'я (список формальних параметрів) : тип результату;

Розділ описів функції

Begin

Оператор 1;

Оператор 2;

...

Оператор n;

Ім'я := результат

End;

У функції формальні параметри підкоряються тим же правилам, що і в процедурі. Результатом функції є одна проста змінна, значення якої привласнюється в тілі функції імені функції.

Викликається функція по її імені з вказівкою фактичних параметрів. Виклик функції можна робити безпосередньо усередині виразу.

Приклад.

```
Program primer;  
Var a, b, c, d, S : real;  
Function lab (x, y, f : real) : real;  
    Var k : real;  
    Begin  
        k := x + y / 2;  
        lab := 2 * k + f  
    End;  
Begin  
    Writeln ('a, b, c, d');  
    Readln (a, b, c, d);  
    S := Lab (a, b, c) + Lab (b, d, a);  
    Writeln('S =', S:5:2);  
End.
```

3.9.3 Приклад програмування завдання з використанням підпрограм

Задані масиви $B(i)$, $H(j)$, $U(k)$, $S(n)$, де $i=1 - 6$; $j=1 - 8$; $k= 1 - 9$; $n= 1 - 7$.

Скласти підпрограми :

- розрахунку значень елементів масивів:

$$B(i) = \sin(i + 0.5) + \cos(i - 2.9) \quad H(j) = \sin(j - 1.4) + \cos(j + 3.7)$$

$$U(k) = \sin(k + 2.2) + \cos(k + 7.8) \quad S(n) = \sin(n - 4.3) + \cos(n + 9.2)$$

- визначення різниці максимального і мінімального елементів масивів.

Рішення.

1) Елементи масивів B , H , U , S обчислюються по загальній формулі:

$$Y(f) = \sin(f + A) + \cos(f + C), \text{ де } f \text{ змінюється від } 1 \text{ до } M.$$

Для кожного масиву можна розрахувати елементи, якщо підставити відповідні коефіцієнти в цю формулу. Наприклад, для масиву B індекс f буде змінюється від 1 до 6, тобто $M=6$, а коефіцієнти $A=0.5$ і $C=-2.9$.

Обчислення елементів масиву $Y(M)$ можна оформити у вигляді окремої процедури з ім'ям Dan . Початковими даними для процедури будуть параметри A , C дійсного типу і розмірність масиву M цілого типу. Результатом виконання процедури Dan буде масив Y , тип якого описується в розділі типів основної програми. Індекс масиву f буде допоміжною змінною в процедурі при організації циклу, тому описується як локальна змінна в розділі описів процедури. Процедура Dan матиме вигляд:

```
Procedure Dan (A,C: real; M: integer; Var Y: mas);
```

```
Var f: integer;
```

```
Begin
```

```
  For f:=1 to M do Y[f]:=sin(f+A)+cos(f+C);
```

```
End;
```

Ця процедура використовуватиметься чотири рази для розрахунку елементів даних масивів.

2) Визначення різниці максимального і мінімального елементів масиву $Y(M)$ можна оформити у вигляді окремої процедури з ім'ям Ras . Початковими даними для процедури Ras буде розмірність масиву M цілого

типу і масив Y типу mas. Результатом виконання процедури Ras буде змінна R дійсного типу (різниця між максимальним і мінімальним елементом масиву). Індекс масиву f, змінна max (значення максимального елемента) і змінна min (значення мінімального елемента) будуть допоміжними в процедурі і описуються як локальні змінні в розділі описів процедури. Процедура Ras матиме вигляд:

```
Procedure Ras (M: integer; Y: mas; Var R: real);  
  Var f: integer; max, min: real;  
  Begin  
    max:=Y[1]; min:=Y[1];  
    For f:=2 to M do  
      If Y[f]>max then max:=Y[f] Else If Y[f]< min then min:=Y[f];  
    R:=max-min;  
  End;
```

3) Для виведення елементів масиву Y(M) можна оформити процедуру Vuvod. Початковими даними для процедури будуть розмірність масиву M цілого типу, масив Y типу mas, символна змінна Z для вказівки імені масиву при друці. Результатом виконання процедури Vuvod буде друк елементів масиву Y. Процедура Vuvod матиме вигляд:

```
Procedure Vuvod (M: integer; Y: mas; Z: char);  
  Var f: integer;  
  Begin  
    Writeln('Масив ', Z);  
    For f:=1 to M do Write(Y[f]:5:3, ' ');  
    Writeln;  
  End;
```

4) У основній програмі приймаються наступні позначення:
NB, NH, NU, NS – розмірності масивів B, H, U, S відповідно;
RB, RH, RU, RS – різниці між максимальним і мінімальним елементів масивів B, H, U, S відповідно. Програма матиме вигляд:

```

Program primer1;
Type mas=array[1..50] of real;
Var B, H, U, S: mas; NB, NH, NU, NS: integer; RB, RH, RU, RS: real;
(* Процедура Dan *)
Procedure Dan (A,C: real; M: integer; Var Y: mas);
Var f: integer;
Begin
    For f:=1 to M do Y[f]:=sin(f+A)+cos(f+C);
End;
(* Процедура Ras *)
Procedure Ras (M: integer; Y: mas; Var R: real);
Var f: integer; max, min: real;
Begin
    max:=Y[1]; min:=Y[1];
    For f:=2 to M do
        If Y[f]>max then max:=Y[f] Else If Y[f]< min then min:=Y[f];
    R:=max-min;
End;
(* Процедура Vuvod *)
Procedure Vuvod (M: integer; Y: mas; Z: char);
Var f: integer;
Begin
    Writeln('Масив ', Z);
    For f:=1 to M do Write(Y[f]:5:3, ' ');
    Writeln;
End;

(* Основна програма *)
Begin
Write ('Ввести розмірність масиву'); Readln(NB);
Dan (0.5,-2.9,NB, B); Ras (NB, B, RB); Vuvod (NB, B, 'B');
Write ('Ввести розмірність масиву H'); Readln(NH);
Dan (-1.4,3.7,NH, H); Ras (NH, H, RH); Vuvod (NH, H, 'H');
Write ('Ввести розмірність масиву U'); Readln(NU);
Dan (2.2,7.8,NU, U); Ras (NU, U, RU); Vuvod (NU, U, 'U');
Write ('Ввести розмірність масиву S'); Readln(NS);
Dan (-4.3,9.2,NS, S); Ras (NS, S, RS); Vuvod (NS, S, 'S');
Writeln ('RB=',RB:5:3 ' RH=',RH:5:3 ' RU=',RU:5:3 ' RS=',RS:5:3);
End.

```

У даному прикладі можна процедуру Ras оформити процедурою без параметрів. В цьому випадку початкові дані розмірність масиву М і масив Y, а також вихідна змінна R повинні оголошуватися в основній програмі як глобальні змінні. Програма в цьому випадку матиме вигляд:

```

Program primer2;
Type mas=array[1..50] of real;
Var B, H, U, S, Y: mas; NB, NH, NU, NS, M: integer; RB, RH, RU, RS, R: real;
(* Процедура Dan *)
Procedure Dan (A,C: real; M: integer; Var Y: mas);
Var f: integer;
Begin
    For f:=1 to M do Y[f]:=sin(f+A)+cos(f+C);
End;
(* Процедура Ras *)
Procedure Ras;
Var f: integer; max, min: real;
Begin
    max:=Y[1]; min:=Y[1];
    For f:=2 to M do
        If Y[f]>max then max:=Y[f] Else If Y[f]< min then min:=Y[f];
    R:=max-min;
End;
(* Процедура Vuvod *)
Procedure Vuvod (M: integer; Y: mas; Z: char);
Var f: integer;
Begin
    Writeln('Масив ', Z);
    For f:=1 to M do Write(Y[f]:5:3, ' ');
    Writeln;
End;
(* Основна програма *)
Begin
    Write ('Ввести розмірність масиву'); Readln(NB);
    Dan (0.5,-2.9,NB, B); Vuvod (NB, B, 'B'); M:=NB; Y:=B; Ras; RB:=R;
    Write ('Ввести розмірність масиву H'); Readln(NH);
    Dan (-1.4,3.7,NH, H); Vuvod (NH, H, 'H'); M:=NH; Y:=H; Ras; RH:=R;
    Write ('Ввести розмірність масиву U'); Readln(NU);
    Dan (2.2,7.8,NU, U); Vuvod (NU, U, 'U'); M:=NU; Y:=U; Ras; RU:=R;
    Write ('Ввести розмірність масиву S'); Readln(NS);

```

```

Dan (-4.3,9.2,NS, S); Vuvod (NS, S, 'S'); M:=NS; Y:=S; Ras; RS:=R;
Writeln ('RB=',RB:5:3 ' RH=',RH:5:3 ' RU=',RU:5:3 ' RS=',RS:5:3);
End.

```

Програма Primer2 є громіздкішою в порівнянні з програмою Primer1. Чим більше число початкових і вихідних даних процедури без параметрів, тим більше громіздкою буде основна частина програми.

У даному прикладі можна також використовувати функцію користувача. Процедури Dan і Vuvod представити у вигляді функції не можна, оскільки в них як дані використовується масив Y структурованого типу, а результатом функції може бути тільки проста змінна. Процедуру Ras можна оформити у вигляді функції, оскільки вихідною величиною в ній є проста змінна R дійсного типу. Цей результат можна привласнити імені функції. Програма в цьому випадку матиме вигляд:

```

Program primer3;
Type mas=array[1..50] of real;
Var B, H, U, S: mas; NB, NH, NU, NS: integer; RB, RH, RU, RS: real;
(* Процедура Dan *)
Procedure Dan (A,C: real; M: integer; Var Y: mas);
Var f: integer;
Begin
    For f:=1 to M do Y[f]:=sin(f+A)+cos(f+C);
End;
(* Функція Ras *)
Function Ras (M: integer; Y: mas): real;
Var f: integer; max, min: real;
Begin
    max:=Y[1]; min:=Y[1];
    For f:=2 to M do
        If Y[f]>max then max:=Y[f] Else If Y[f]< min then min:=Y[f];
    Ras:=max-min;
End;
(* Процедура Vuvod *)
Procedure Vuvod (M: integer; Y: mas; Z: char);
Var f: integer;
Begin
    Writeln('Масив ', Z);

```

```

    For f:=1 to M do Write(Y[f]:5:3, ' ');
    Writeln;
End;
(* Основна програма *)
Begin
Write ('Ввести розмірність масиву'); Readln(NB);
Dan (0.5,-2.9,NB, B); Vuvod (NB, B, 'B'); RB:=Ras (NB, B);
Write ('Ввести розмірність масиву H'); Readln(NH);
Dan (-1.4,3.7,NH, H); Vuvod (NH, H, 'H'); RH:=Ras (NH, H);
Write ('Ввести розмірність масиву U'); Readln(NU);
Dan (2.2,7.8,NU, U); Vuvod (NU, U, 'U'); RU:=Ras (NU, U);
Write ('Ввести розмірність масиву S'); Readln(NS);
Dan (-4.3,9.2,NS, S); Vuvod (NS, S, 'S'); RS:=Ras (NS, S);
Writeln ('RB=',RB:5:3 ' RH=',RH:5:3 ' RU=',RU:5:3 ' RS=',RS:5:3);
End.

```

4. ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО РІШЕННЯ

Скласти алгоритм і програму рішення задачі.

1. Обчислити значення у використовуючи формулу:

$$y = b^3 + \sqrt{|ax - b|} \quad b = \sin x + \cos(ax)$$

2. Обчислити значення z використовуючи формулу:

$$z = \begin{cases} a + x, & x^3 > y \\ \sqrt{a + x}, & x^3 = y \\ \ln |a + x|, & x^3 < y \end{cases} \quad y = e^{0.5x} + \sqrt{(a + x)^3}$$

3. Обчислити значення у використовуючи формулу:

$$y = \begin{cases} \sqrt{bx}, & x < 4 \\ a + bx, & 4 \leq x \leq 6 \\ (a - b)x, & x > 6 \end{cases}$$

4. Визначити, чи потрапляє крапка з координатами (x,y) в круг радіусу R (рівняння кола $R^2 = x^2 + y^2$). Вивести ознаку N=1, якщо крапка знаходиться усередині круга, і ознака N=0, якщо крапка знаходиться поза кругом.

5. Обчислити значення t використовуючи формулу:

$$t = \frac{b + a \cdot \sqrt{y}}{\sqrt{ay}}, \text{ якщо } 1 \leq y \leq 3 \text{ і } \Delta y = 0.1$$

6. Рахувати t до тих пір, поки підкорінний вираз стане від'ємним.

$$t = a \cdot \cos^2 x - \frac{1}{3} \sqrt{4x^2 - 3x - 2}$$
$$a = 1.7 \quad x \leq 5 \quad \Delta x = -0.4$$

7. Обчислити значення x використовуючи формулу:

$$x = \frac{c^2 + 1}{b} + \sqrt{cb}, \text{ якщо } -2 \leq b \leq 2 \text{ і } \Delta b = 1.$$

8. Обчислити значення x і y використовуючи формули:

$$x = \frac{|a - bc^2 + c^3|}{12 \cdot (bc + 1)} \quad y = \frac{a^2 - bx^3 + 16cx^2}{x + a}$$

Початкові дані:

$$b = 3.6 \quad 1 \leq a \leq 7 \quad \Delta a = 1 \quad -0.5 \leq c \leq 0.5 \quad \Delta c = 0.1$$

9. Обчислити значення x і y використовуючи формули:

$$x = 2ab^2 \quad y = \frac{x^5}{db} + \frac{x}{2a}$$

Початкові дані:

$$a = 2.9 \quad 1.2 \leq b \leq 4.8 \quad \Delta b = 0.2 \quad -1 \leq d \leq 4 \quad \Delta d = 1$$

10. Для масиву $X(N)$ виконати наступні дії:

- записати на місце від'ємних елементів масиву нулі;
- вивести номери елементів, що задовольняють умові $0 < x_i < 1$;
- вивести додатні елементи масиву.

11. У масиві $A(N)$ знайти елемент, значення якого рівне 5 і вказати його номер. Якщо таких елементів в масиві декілька, то вказати елемент з найменшим номером. Якщо ж елементу з таким значенням в масиві немає - вивести відповідне повідомлення.

12. Є масив $A(N)$. Знайти кількість елементів масиву A , що задовольняють умові $P \leq A_i \leq Q$.

13. Є масив $A(K)$ і число C . Визначити:

- чи зустрічається дане число серед елементів масиву;
- виключити з послідовності елементи рівні C , здвинувши інші на місця, що звільнилися;
- замінити всі елементи масиву, рівні числу C , на нуль.

14. Вивести додатні елементи головної діагоналі матриці $X(N, N)$.

15. Обчислити значення елементів масиву Y за формулою:

$$y_i = \sqrt{|x+a|} / 2, \text{ якщо:}$$

а) змінна x є елементом масиву $X(40)$, змінна a змінюється від 2 з кроком 0.3;

б) змінна x змінюється від 2.5 до 0 з кроком -0.5, змінна a змінюється від 0 з кроком h .

16. Записати елементи цілочисельного масиву $X(N)$, кратні 5, підряд в масив M . Якщо такі елементи відсутні, то вивести відповідне повідомлення.

17. Для масиву $X(N)$ виконати наступні дії:

а) записати елементи масиву X в масив Y в зворотному порядку;

б) записати додатні елементи масиву X підряд в масив Y ;

в) записати додатні елементи масиву X підряд в масив Y , а від'ємні елементи – підряд в масив Z ;

г) записати в масив M номери елементів масиву X , що задовольняють умові $0 < x_i < 1$;

д) записати в масив C підряд 10 перших додатних елементів масиву X ;

е) записати в масив Y елементи масиву X , що мають парні індекси;

ж) записати в масив Y елементи масиву X , що мають парні індекси, а в масив Z – елементи, що мають непарні індекси;

з) записати на початку масиву Y додатні, а потім від'ємні елементи масиву X .

18. Дана матриця $X(N, M)$. Записати в одновимірний масив B :

а) елементи x_{ii} головної діагоналі матриці X ;

б) елементи $x_{i,n-i+1}$ побічної діагоналі матриці X ;

в) елементи x_{ii} і елементи $x_{i,n-i+1}$ головних діагоналей матриці X .

19. Обчислити суму додатних і добуток від'ємних значень функції $z = \cos(nx+a) \cdot \sin(nx-a)$, де $n = 1, 2, \dots, 10$.

20. Для масиву $X(M)$, що має додатні і від'ємні елементи, обчислити:

а) середнє арифметичне додатних елементів масиву;

- б) суму елементів, що стоять на парних місцях;
- в) середнє геометричне елементів, що задовольняють умові $1 < x_i < 4$;
- г) кількість додатних, від'ємних і рівних нулю елементів.

21. Знайти різницю факторіалів $F = m! - k!$

22. Обчислити суму елементів головної і побічної діагоналей матриці $A(M, M)$.

23. Обчислити суму членів для наступних рядів:

а) $z = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$ з точністю до члена ряду, меншого 10^{-4} ;

б) $z = 1 + \frac{mx}{(m+1)!} + \frac{m(m-1)x^2}{(m+2)!} + \frac{m(m-1)(m-2)x^3}{(m+3)!} + \dots$

з точністю до члена ряду, меншого ε . Для обчислення поточного значення члена ряду використовувати рекурентну формулу:

$$y_n = y_{n-1} \frac{x(m-n+1)}{(m+n)}, \text{ де } n - \text{ номер члена ряду.}$$

Початкове значення у прийняти рівним $\frac{1}{m!}$

в) $z = \cos x + \frac{\cos 2x}{4} + \frac{\cos 3x}{9} + \dots + \frac{\cos nx}{n^3} + \dots$ з точністю до члена ряду,

меншого ε .

24. Визначити максимум функції $y = \sin x + 2 \sin x$. Аргумент x змінюється від 0 до π з кроком 0.1.

25. Для масиву $X(M)$ визначити:

- а) найбільший і найменший елементи масиву і їх індекси;
- б) найбільше і найменше значення модуля різниці між сусідніми елементами;
- в) найбільший і найменший елементи; поміняти їх місцями;
- г) найменший з додатних елементів;
- д) найбільший елемент, що задовольняє умові $1 < x_i < 2$;
- е) найбільший елемент; замінити його значенням суми попередніх йому елементів.

26. Визначити найбільший елемент головної діагоналі матриці $X(M, M)$ і вивести рядок, в якому він знаходиться.

27. Упорядкувати елементи масиву $A(N)$, розташувавши по убутанню в тому ж масиві.

28. Є впорядкований за збільшенням масив $A(N)$. Вилучити з масиву число Z так, щоб впорядкованість масиву не порушилася.

29. Дана матриця $X(N, M)$. Визначити:

а) суму елементів, розташованих вище за головну діагональ;

б) добуток додатних елементів, розташованих нижче за головну діагональ;

в) суму елементів кожного рядка матриці;

г) найбільший елемент матриці і номери рядка і стовпця, в яких він розташований;

д) найменший елемент матриці; записати нулі в рядок і стовець, в яких він знаходиться;

е) упорядкувати за збільшенням елементи того рядка, де знаходиться найбільший елемент матриці.

30. Перемножити матриці $A(N, M)$ і $B(M, L)$. Елементи результуючої матриці обчислити за допомогою формули: $C_{ik} = \sum_{j=1}^m a_{ij} b_{jk}$

31. Упорядкувати елементи кожного рядка матриці $A(H, B)$, розташувавши їх в тій же матриці по убутанню.

32. Упорядкувати елементи масиву $X(M)$ за збільшенням, якщо ознака $p=1$, що вводиться, і по убутанню, якщо $p=0$.

33. Скласти таблицю значень функцій, використовуючи допоміжні підзадачі

$g(x) = 3.7x^3 - 0.8\sqrt{|a \cdot \sin^2 x + 2.5|}$ $z(x) = -0.75x^3 + 1.2\sqrt{|a \cdot \sin^2 x - 0.6|}$, якщо $1 \leq x \leq 10$, $\Delta x = 0.5$

34. Записати вирази, використовуючи оператори мови Pascal:

$$f = \frac{a+b}{2x+b} (a+x) \cdot \sin x \quad a = \cos^2 x^3 + \sqrt{x+|z|} \quad c = \frac{x^2 - \cos^3 x}{1.4 + \ln(|x|+1)} \quad z = e^{\sin^3 x} + \ln(x + e^x)$$

5. ТЕКСТОВИЙ РЕДАКТОР MICROSOFT WORD

5.1 Екранний інтерфейс редактора

Текстовий редактор MICROSOFT WORD використовують практично всі, хто працює на комп'ютері, і професіонали і «чайники», які набирають текст одним пальцем по букві в хвилину.

Редактор WORD може виконувати всі процеси обробки тексту від звичайного введення до детальної обробки із завданням кольору, розміру, стилю окремих букв і всього документа, здійснювати перевірку орфографії, вставляти в текст формули і картинки, таблиці і слайди.

Для відкриття текстового редактора MICROSOFT WORD необхідно виконати команду **Пуск/Програми/ Microsoft Word**. На екрані з'явиться вікно редактора (див. рис.5.1), яке містить:

- 1 – рядок заголовка;
- 2 – кнопки управління вікном;
- 3 – рядок меню;
- 4 – панелі інструментів;
- 5 – горизонтальна і вертикальна лінійки;
- 6 – горизонтальна і вертикальна смуга прокрутки;
- 7 - рядок стану.

5.2 Початок роботи в текстовому редакторі

Починаючи роботу в текстовому редакторі, перш за все, слід підготувати вікно редактора до зручної роботи у ньому. Пропонується наступна послідовність дій, яка не є обов'язковою, але перевірена багато разів.

1. Встановити режим відображення документа, використовуючи пункт меню **Вид**.

Пропонується декілька варіантів:

Обычный – придатний в більшості випадків, але не відображає області полів документа; нагадує набір тексту на друкарській машинці;

Веб-документ – зручний для читання Web-документів;

Разметка страницы – найбільш оптимальний режим, дозволяє побачити всі невидимі в звичайному режимі елементи документа;

Структура документа – дозволяє переглядати, створювати і змінювати ієрархію заголовків і відповідних ним розділів тексту.

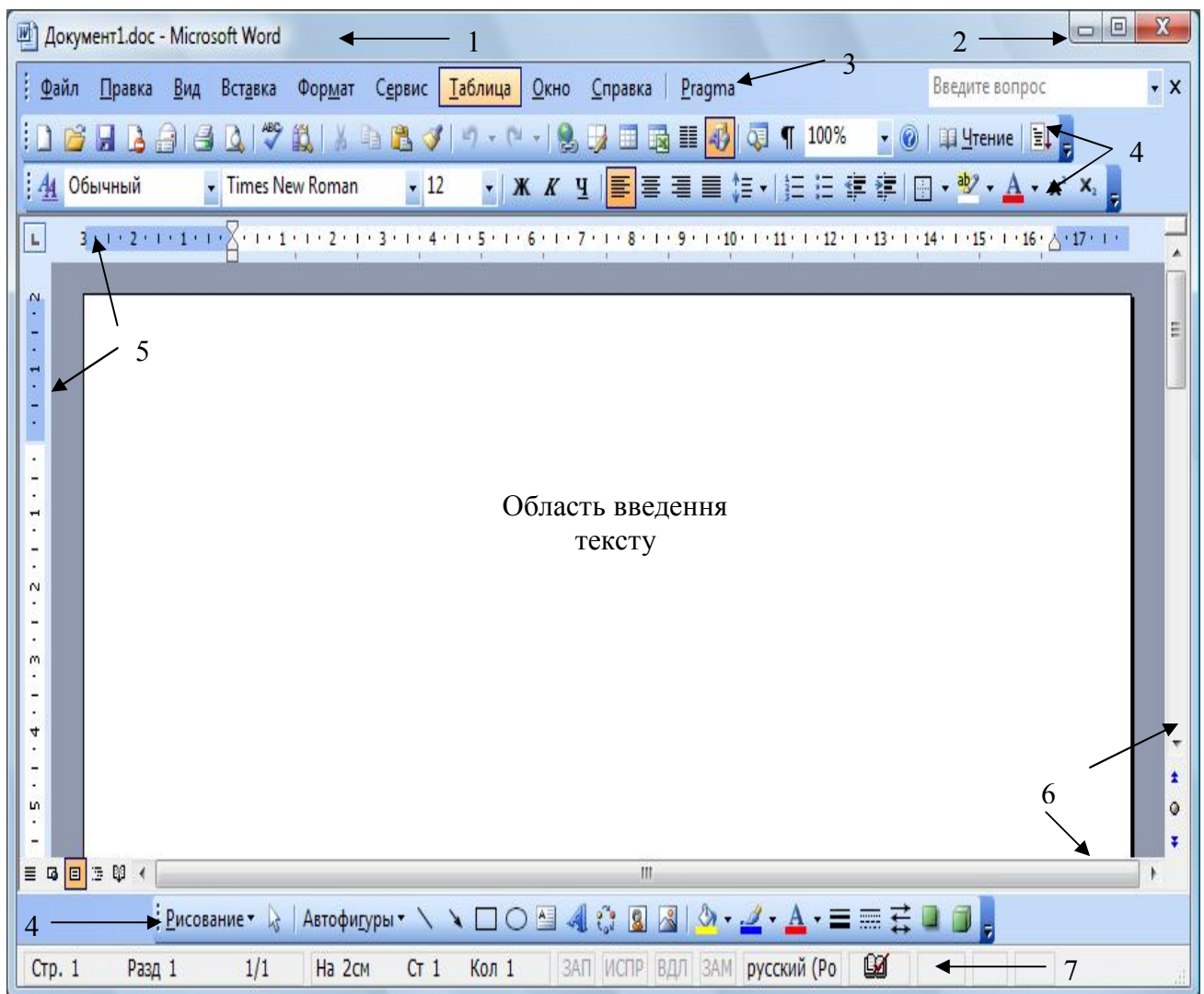
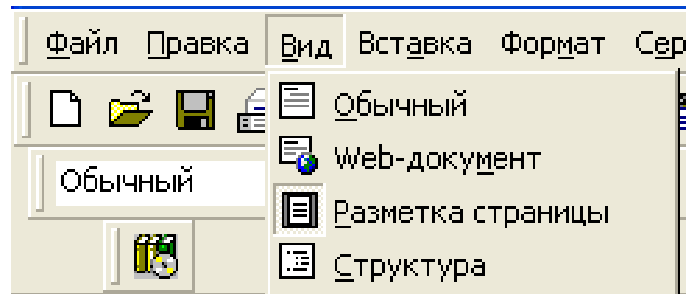


Рис.5.1 - Зовнішній вигляд вікна текстового редактора MICROSOFT WORD

2. Встановити масштаб відображення документа, використовуючи пункт меню **Вид/Масштаб** або кнопку на панелі інструментів.

Якщо в документі використаний дуже дрібний шрифт, то, можливо, потрібно буде збільшити масштаб його відображення. З іншого боку, іноді необхідно змінити масштаб так, щоб побачити одночасно велику частину документа.

У діалоговому вікні, що з'явиться, вибрати необхідне значення або ввести необхідний відсоток масштабування і натиснути кнопку ОК.

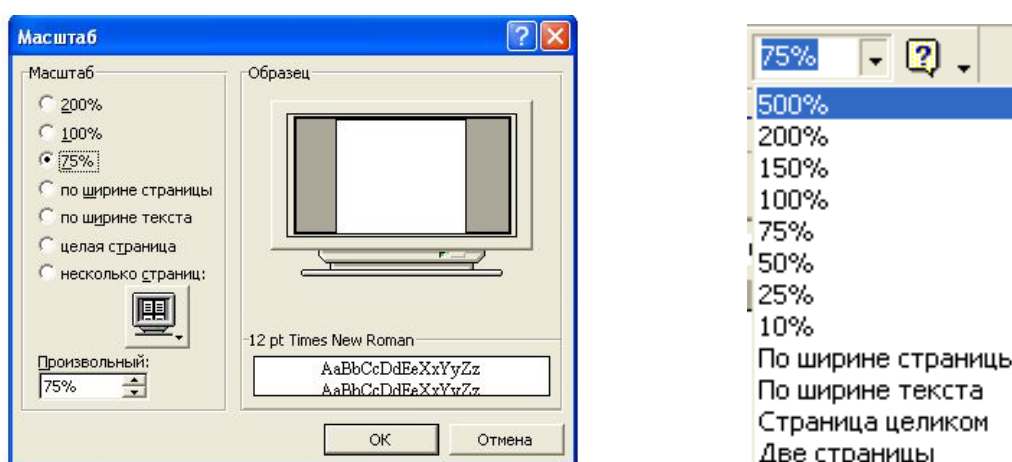
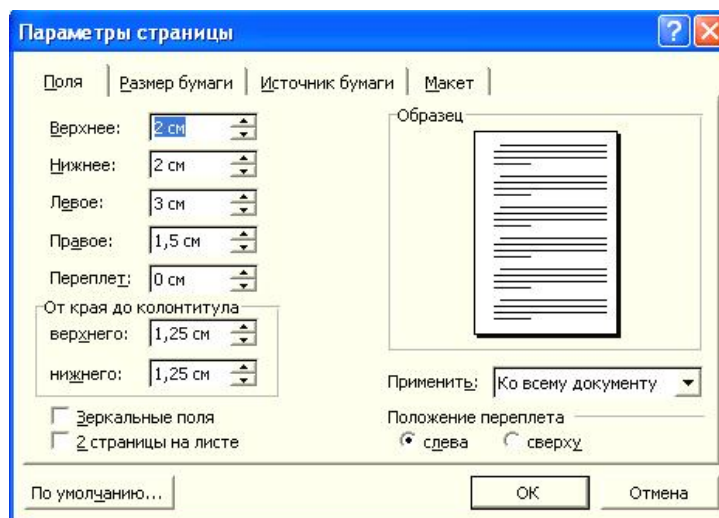


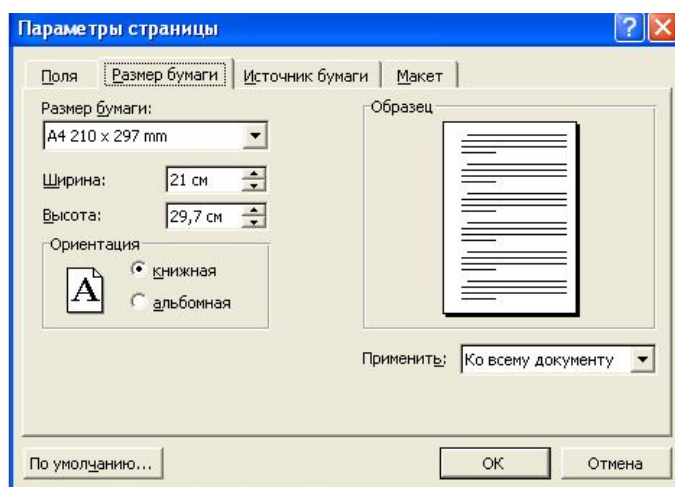
Рис.5.2 - Встановлення масштабу відображення документа

2. Встановити параметри сторінки та її орієнтацію, використовуючи пункт меню **Файл/Параметры страницы**.

У діалоговому вікні, що з'явиться, на вкладці **Поля** ввести значення розмірів полів у вікна **Верхнее, Нижнее, Левое и Правое**.



Для зміни орієнтації сторінки потрібно перейти на вкладку **Размер бумаги** та вибрати орієнтацію **Книжная** або **Альбомная**.



Після установки параметрів натиснути кнопку **ОК**.

5.3 Створення нового документа, введення тексту і його збереження

При запуску WORD надає вікно документа, в якому можна відразу ж вводити текст. Миготливий курсор показує, де з'явиться наступний введений символ.

Введення тексту в програмі WORD за допомогою клавіатури має ряд особливостей.

Клавіша **Enter** дозволяє закінчити введення рядка і перевести курсор в наступний рядок. Проте нею слід користуватися тільки для того, щоб закінчити введення чергового абзацу і перейти до наступного. В межах одного абзацу, як тільки курсор досягне правої межі документа, WORD сам переведе його в наступний рядок.

Не слід вставляти декілька пропусків підряд, наприклад, для того, щоб задати відступ першого рядка абзацу. Для цього краще скористатися клавішею табуляції (клавіша **Tab**).


При використанні клавіш **Пробел**, **Enter**, **Tab** програма вставляє в текст спеціальні символи. Щоб їх побачити необхідно клацнути на панелі інструментів на кнопку



Январь · · Февраль · · Март ¶
¶
Апрель · · Май · · Июнь ¶
¶
Текстовый редактор WORD ¶
¶

Повторне клацання на даній кнопці прибирає ці символи з екрану.


Документ, що вводиться, існує тільки в пам'яті комп'ютера, тому, якщо виникне необхідність повернутися до цього документа надалі, слід зберегти його на диску.

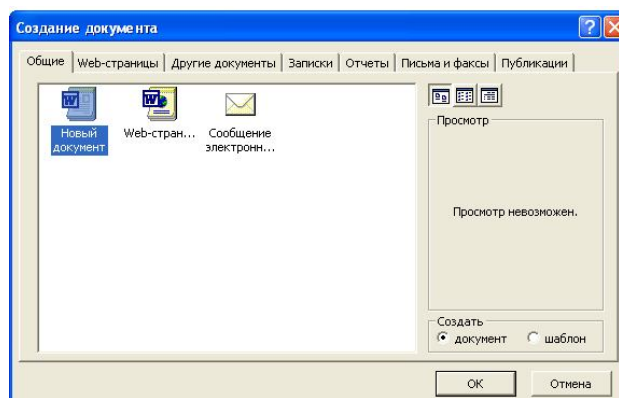
Для збереження документа можна скористатися кнопкою на панелі інструментів  або командою **Файл/Сохранить**.

У діалоговому вікні, що з'явиться **Сохранение документа** необхідно ввести ім'я файлу, вибрати із списку **Папка** папку, в яку необхідно зберегти документ. Потім клацнути на кнопці **Сохранить**.

Зберігаючи файл в процесі роботи, вікно **Сохранение документа** більше відображатися не буде.

Якщо ж необхідно мати в своєму розпорядженні і стару версію документа, і версію з внесеними змінами, зберегти документ необхідно під іншим ім'ям, використовуючи команду **Файл/Сохранить как**. Знов з'явиться діалогове вікно **Сохранение документа**, і можна ввести нове ім'я і вибрати іншу папку.

Щоб почати новий документ необхідно в панелі інструментів клацнути на кнопці  або вибрати команду **Файл/Создать**. У вікні, що з'явиться, вибрати пункт **Новий документ** и натиснути кнопку **ОК**.

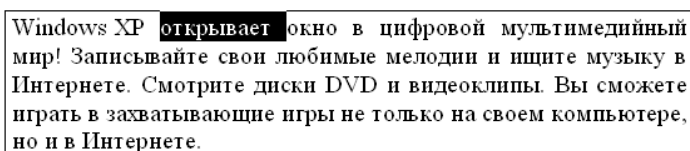


5.4 Основи редагування документа

Процес редагування може полягати як в простому виправленні друкарських помилок або перестановки слів, так і в повній переробці цілих абзаців або розділів. Незалежно від ступеня зміни тексту механізм редагування в WORD достатньо простий в освоєнні.

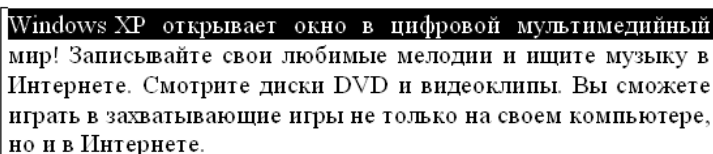
Перш ніж виконувати які-небудь операції редагування в документі, необхідно вказати програмі, якої частини тексту повинні торкнутися зміни. Для цього текст потрібно виділити.

1. **Виділення слова** – два рази лівою кнопкою миші усередині слова



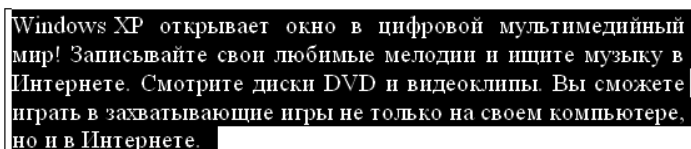
Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.

2. **Виділення рядка** – помістити курсор миші зліва від рядка на порожнє місце і клацнути один раз лівою кнопкою миші



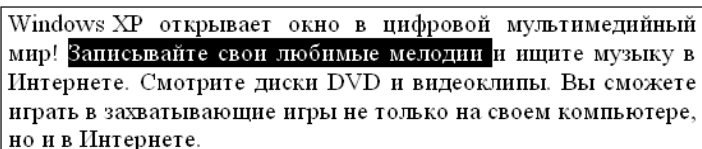
Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.

3. **Виділення абзацу** – три рази лівою кнопкою миші усередині абзацу



Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.

4. **Виділення довільної частини тексту** – один раз лівою кнопкою миші в тому місці, де необхідно почати виділення і, утримуючи її, протягнути до того місця, де необхідно закінчити виділення



Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.

5. **Посимвольне виділення** – помістити текстовий курсор в те місце, де необхідно почати виділення, натиснути на клавішу Shift і, утримуючи

її, натискати на клавіатурі клавішу із стрілкою вліво (управо) до кінця виділення

Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.

6. **Виділення прямокутного блоку** – утримуючи клавішу Alt, перетягувати мишу з початку блоку і до кінця

Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.

7. **Виділення всього тексту** – команда **Правка/Выделить все**

Для відміни виділення тексту необхідно клацнути в будь-якому місці області тексту.

Дії з текстом:

1. Видалення тексту

Для видалення тексту використовуються дві клавіші **Backspace** і **Delete**. Backspace видаляє символ зліва від курсора і зрушує курсор вліво. Delete видаляє символ праворуч від курсора і зрушує текст, що залишився справа, вліво. Щоб видалити цілий текстовий блок, необхідно виділити його і натиснути клавішу Backspace або Delete.

2. Копіювання і переміщення фрагмента тексту

Існує два методи:

- переміщення тексту за допомогою миші

Помістити покажчик миші на виділений фрагмент тексту і натиснути праву кнопку миші, утримуючи її, почати переміщення тексту. Встановити курсор перед тим словом, перед яким слід вставити текст і відпустити кнопку миші. У меню, що з'явиться, вибрати потрібну команду.


- через буфер обміну


Виділити текст і вибрати команду **Правка/Копировать** (для копіювання) або **Правка/Вырезать** (для переміщення). Текст буде

поміщений в буфер обміну. Встановити курсор перед тим словом, перед яким слід вставити текст і вибрати команду **Правка/Вставити**. Текст з'явиться на новому місці.

3. Відміна дій

WORD дає можливість відмінити більшість дій, включаючи введення, видалення, переміщення, копіювання і форматування тексту.

Для відміни найостаннішої дії використовується кнопка на панелі інструментів  або команда **Правка/Отменить**.

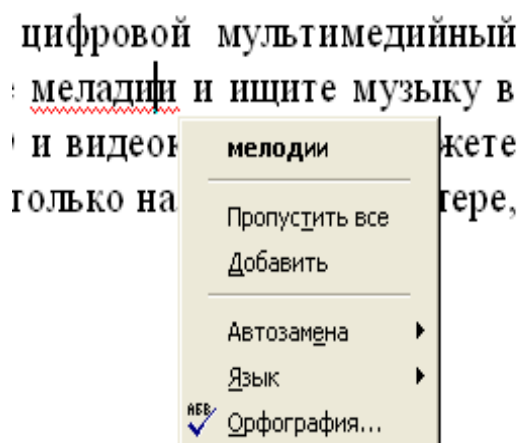
Дію команди **Отменить** теж може бути відмінено або кнопкою на панелі інструментів  або командою **Правка/Вернуть**.

WORD надає різні способи перевірки і виправлення орфографічних і граматичних помилок в документі. Слово, яке відсутнє в словнику WORD або не відповідає встановленим в WORD граматичним правилам, WORD підкреслить це слово кольоровою хвилястою лінією.

Для виправлення правопису слова існує два способи:

1 спосіб – виправлення «на льоту»

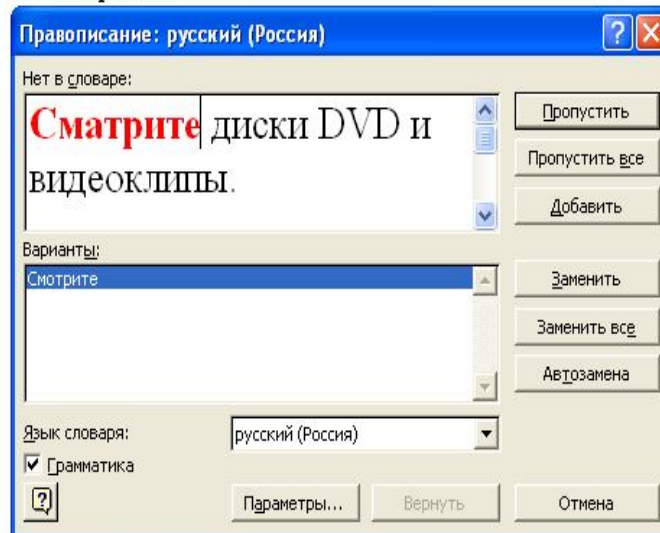
Клацнути на слові правою кнопкою миші і вибрати з контекстного меню правильний варіант написання. Якщо слово написано правильно, але відсутнє в словнику, його можна додати, використовуючи команду в контекстному меню **Добавить в словарь** або вибрати команду **Пропустить** для відміни підкреслення слова.



2 спосіб – команда *Сервис/Правописание*

Коли програма знайде слово, яке відсутнє в словнику WORD або не відповідає встановленим в WORD граматичним правилам, на екрані з'явиться вікно **Правописание**. У списку **Варианты** необхідно вибрати пропонований варіант заміни і клацнути на кнопці **Заменить**. Якщо слово написано правильно, але відсутнє в словнику, його можна додати, використовуючи кнопку **Добавить** або кнопку **Пропустить** для відміни підкреслення слова.

Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. **Смотрите** диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.



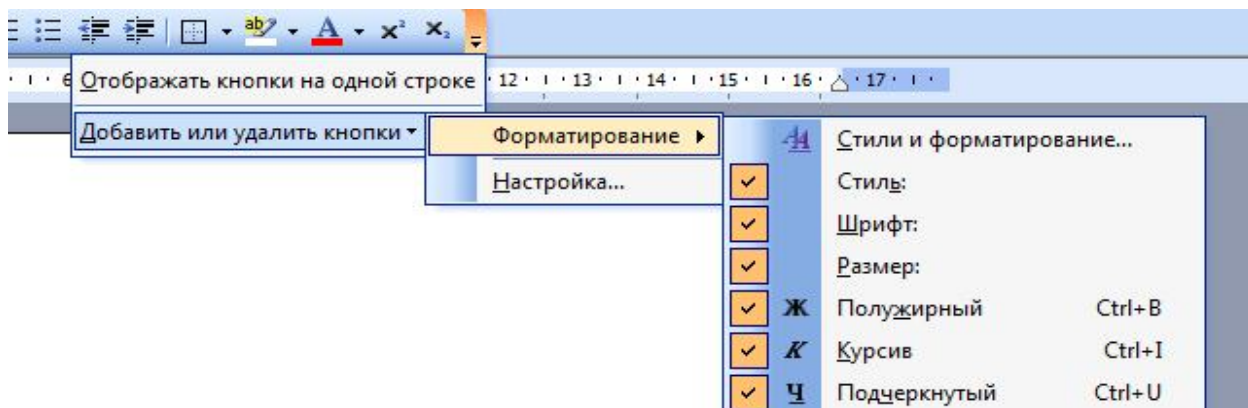
5.5 Основи форматування документа

Існує два методи виконання форматування символів і абзацу – для швидкого форматування використовуються кнопки на панелі інструментів і за допомогою діалогового вікна.

WORD має в своєму розпорядженні значно великий набір кнопок, ніж той, який представлений на панелях інструментів.

Для додавання кнопки на панель інструментів необхідно клацнути на стрілці в кінці панелі інструментів, до якої необхідно додати кнопки. У меню, що з'явиться, вибрати команду **Добавить** или **удалить** кнопки.

Клацнути на назві тієї кнопки, яку слід додати на панель інструментів. Поряд з назвою і зображенням кнопки встановиться прапорець, а сама кнопка з'явиться на панелі інструментів. Для видалення кнопки необхідно зняти прапорець поряд з кнопкою.



5.5.1 Форматування символів

Форматуванням символів визначається їх зовнішній вигляд. Параметрами форматування символів є: зображення, гарнітура шрифту, розмір шрифту, міжбуквений інтервал, кернинг.

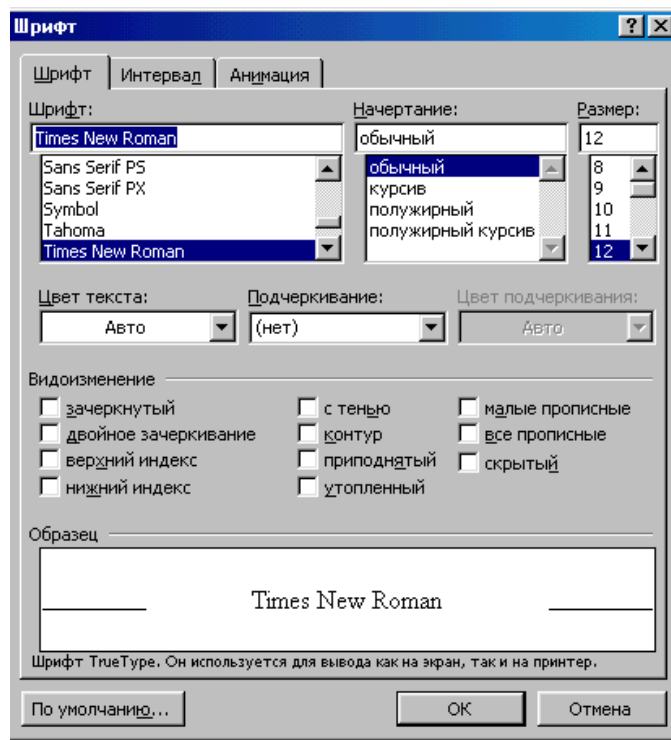
1) Використання панелі інструментів **Форматирование**.

Для зміни шрифту вибрати необхідний шрифт зі списку **Шрифт** Times New Roman. Розмір шрифту вибрати зі списку **Размер** 20, що розкривається або у текстовому полі ввести нове значення. Для зміни накреслення на панелі інструментів використовуються кнопки **Ж К Ч**, де **Ж** (напівжирний), **К** (курсив), **Ч** (підкреслений). Також можна змінити колір шрифту **A** і фон виділення **[pen icon]**.

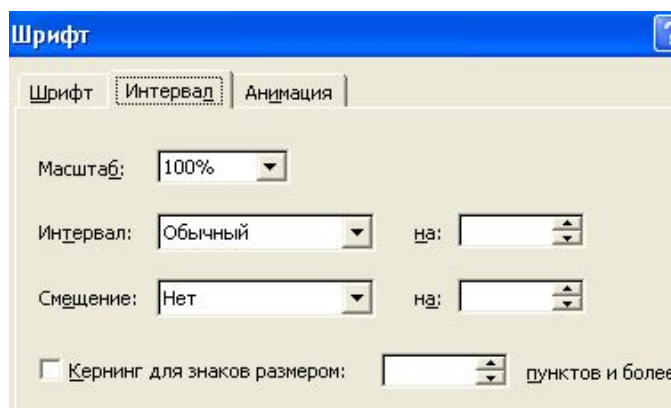
2) Використання діалогового вікна **Шрифт**.

Виберіть команду **Формат/Шрифт...** У діалоговому вікні, що з'явиться, на вкладці **Шрифт** необхідно вибрати шрифт, розмір, накреслення, вид підкреслення, колір тексту та натиснути кнопку **ОК**.

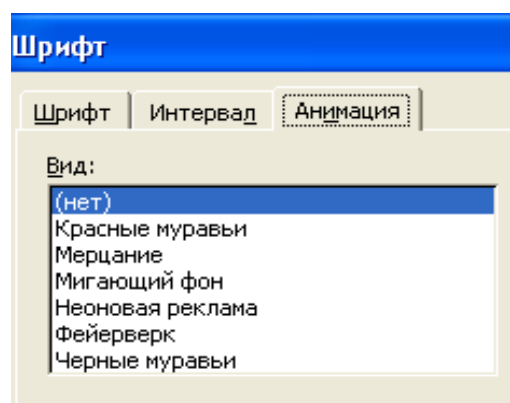
Діалогове вікно також має спеціальні ефекти (видозміна тексту) та дозволяє подивитися результати вибору у вікні перегляду.



Вкладка **Интервал** використовується для завдання інтервалу між символами. Звичайний – відповідає стандартній відстані між символами, розріджений – їх збільшує, ущільнений – зменшує.



Вкладка **Анимация** додає анімаційні ефекти, проте на друці вони не відображаються.



5.5.2 Форматування абзацу

Форматування абзаців має безліч можливостей: зміна міжрядкових інтервалів, відступів, вирівнювання і т.п.

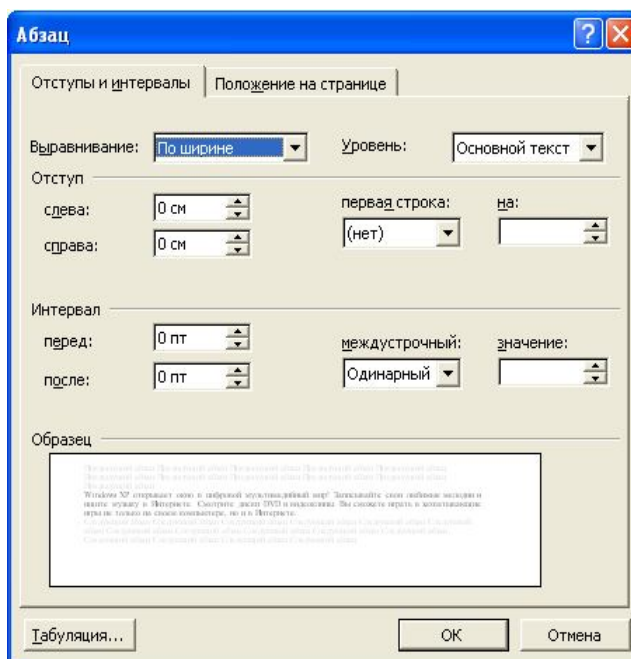
Для форматування абзацу треба виділити його та вибрати команду **Формат/Абзац** або використати кнопки панелі інструментів **Форматирование**.

У діалоговому вікні **Абзац**, що з'явиться, можна зробити:

- установити відступ для тексту абзацу ліворуч, праворуч;
- установити відступ для першого рядка абзацу;
- установити інтервал між абзацами (числові значення в полях

Перед та После);

- установити міжрядковий інтервал, використовуючи список **Междустрочный**. В списку вибрати необхідну опцію та при необхідності ввести числові значення у поле **Значение**;

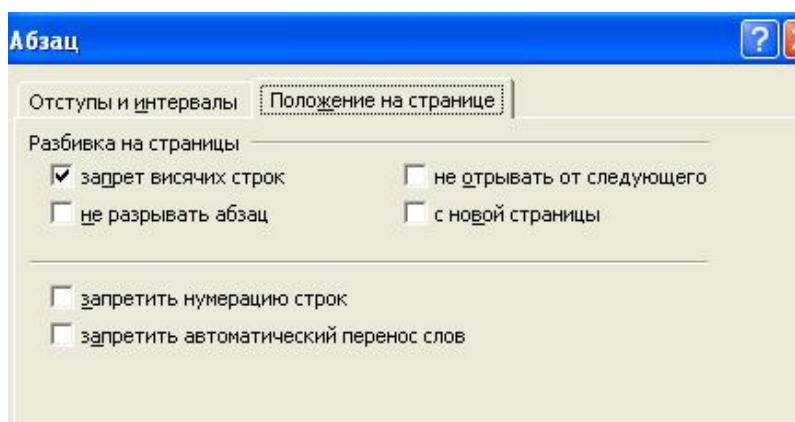


- установити спосіб вирівнювання тексту (по лівому краї, по правому, по центрі або по ширині).

Після установки параметрів форматування абзацу натиснути кнопку **ОК**.

Вкладка **Положение на странице** дозволяє встановити наступні параметри:

- **запрет висячих строк** забороняє відокремлювати від абзацу перший і останній рядок і переносити її на наступну сторінку;
- **не разрывать абзац** взагалі забороняє розривати абзац і переносити на іншу сторінку;
- **не отрывать от следующего** забороняє розривати сторінки між вибраним і наступним абзацом;
- **с новой страницы** вставляє розрив сторінки перед вибраним абзацом;
- **запретить нумерацию строк** забороняє нумерацію рядків починаючи з наступного абзацу;
- **запретить автоматический перенос слов** забороняє автоматичне перенесення слів у вибраному абзаці.



5.5.3 Створення рамок і заливок

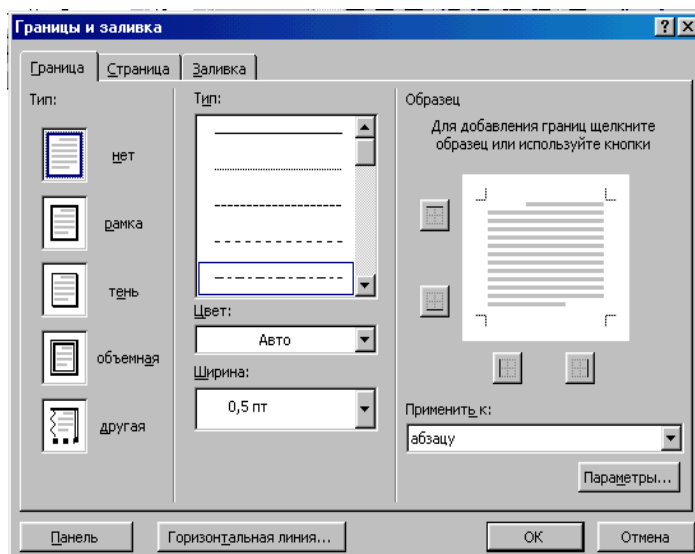
Для створення рамок і заливок зовсім не обов'язково що-небудь знати про графіку. WORD дозволяє створити рамку не тільки для абзаців або заголовків, але і для цілої сторінки.

Щоб обвести і/або залити абзац відтінком (або будь-який інший фрагмент), його необхідно спочатку виділити, а потім вибрати команду **Формат/Границы и Заливка ...** та установити необхідні параметри.

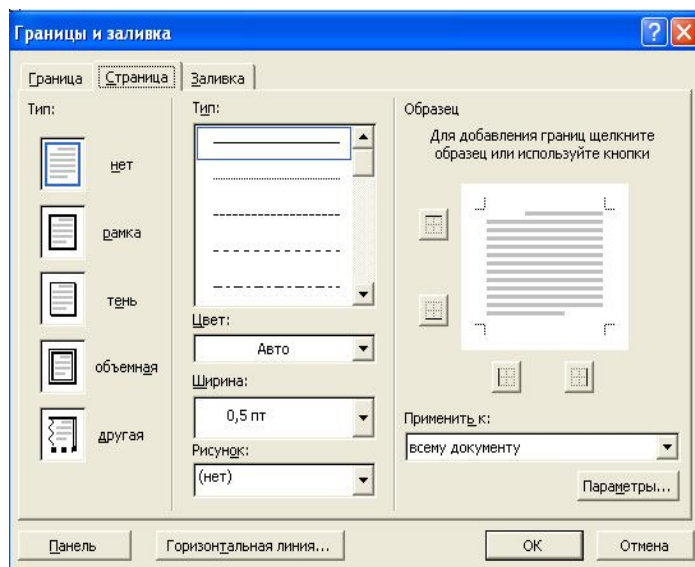
У розділі **Тип** вибрати варіант лінії рамки. За допомогою списку **Цвет** задати колір лінії, а за допомогою списку **Ширина** – її ширину. Щоб лінія з'явилася біля абзацу, клацнути на відповідній кнопці в розділі **Образец**. Для обведення абзацу зі всіх чотирьох боків клацнути на всіх чотирьох кнопках.

За допомогою вкладок **Граница** та **Заливка** неважко зробити різні варіанти обрамлення і заливання абзацу.

Щоб підкреслити абзац художньою лінією можна використовувати кнопку **Горизонтальная линия**.



Для створення рамки на сторінці використовується вкладка **Страница**. Багато опцій цієї вкладки такі ж і призначені для того ж, що і опції вкладки **Граница**. З тією лише різницею, що рамка створюється не навколо абзацу, а навколо сторінки.

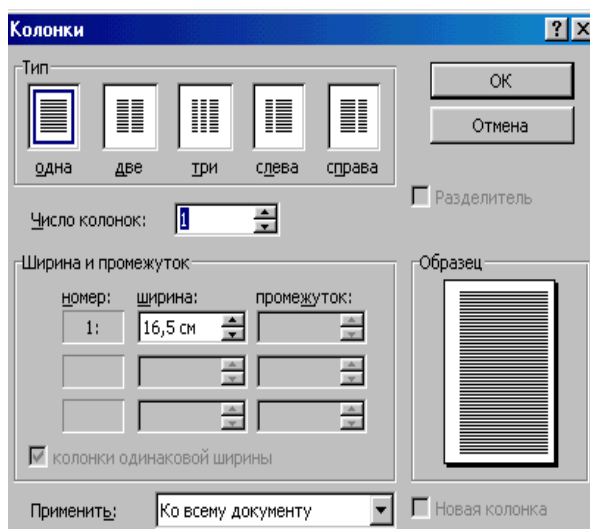


5.5.4 Створення колонок тексту

У WORD є можливість створювати газетні колонки, тобто такі, в яких текст розташовується зверху вниз в одній колонці, а потім перетікає у верхню частину наступної колонки.

Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и	видеокадры. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.
--	---

Для цього, нічого попередньо не виділяючи, вибрати команду **Формат/Колонки...** та вказати, на яку кількість стовпчиків варто розбити документ. Якщо треба розбити на стовпчики тільки частину документа, то для цього треба виділити той фрагмент документа, який треба розбити на стовпчики та вибрати команду **Формат/Колонки...**



За умовчанням створюються колонки рівної ширини. Щоб мати можливість задати ширину кожної колонки і величину проміжку між ними, необхідно скинути прапорець **Колонки однакової ширини** і ввести необхідні розміри для кожної колонки в розділі **Ширина і Промежуток**.



Для введення вертикальних ліній між колонками встановити прапорець **Разделитель**.

5.5.5 Створення нумерованих і маркірованих списків

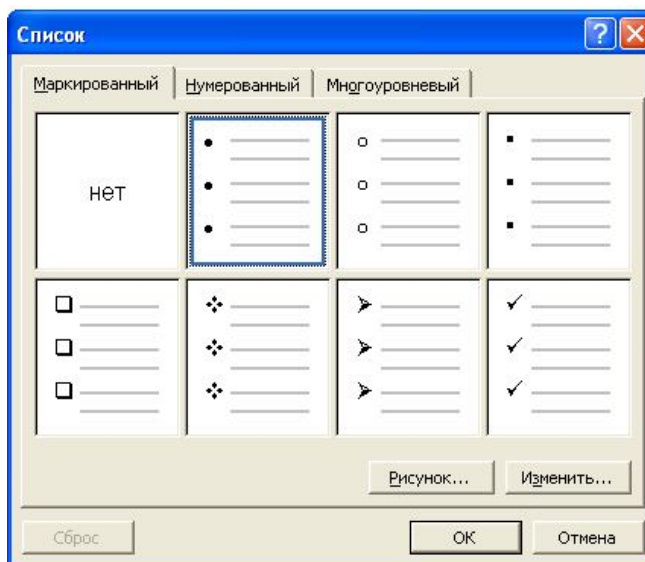
Можливості WORD по створенню списків з нумерованими і маркірованими позиціями дозволяють автоматично вводити номери або маркери при введенні списків. Абзац списку форматується з виступом першого рядка.

Для створення нумерованого списку ввести номер ("1." або "1"), а після нього пропуск. Ввести текст першої позиції і натиснути клавішу **Enter**. Після натискання клавіші **Enter**, у наступному абзаці буде відразу стояти другий номер. Якщо новий нумерований абзац не потрібен, натисніть клавішу **Backspace** і абзац стане "нормальний" (без номера).

Маркірований (ненумерований) список створюється аналогічним образом. Цей тип списку відрізняється тим, що абзаци починаються не з номера, а з маркера - деякого символу.

Для створення списку з вже введених абзаців необхідно виділити їх і клацнути на кнопці в панелі інструментів  для створення нумерованого списку або на кнопці  для створення маркірованого списку.

Команда **Формат/Список** також використовується для створення або зміни списку. Команда містить три вкладки (три типа списку), кожна вкладка містить вісім різних варіантів. Для зміни списку необхідно його виділити і вибрати команду **Формат/Список**, перейти на необхідну вкладку і вибрати варіант списку, натиснути кнопку ОК.



Список, який включає інші списки, називається багаторівневим.

При створенні такого списку: щоб перейти з рівня на підрівень необхідно натиснути клавішу **Tab**, а щоб перейти з підрівня на рівень натиснути комбінацію клавіш **Shift+Tab**.

1 Устройство компьютера

1.1 Монитор

1.2 Системный блок

1.2.1 Процессор

1.2.2 Системная плата

1.2.3 Оперативная память

1.2.4 Жесткий диск

1.3 Устройства ввода информации

1.3.1 Клавиатура

1.3.2 Мышь

Для видалення маркерів списку можна використовувати кнопки на панелі інструментів або клавішу **Backspace**.

5.6 Створення колонтитулів

Верхній колонтитул це текст, який друкується у верхній частині кожної сторінки, а нижній колонтитул – це текст, який друкується у нижній частині кожної сторінки.

Верхні і нижні колонтитули можна використовувати для виведення на сторінці таких відомостей, як номер сторінки, дата, найменування організації і т.п. Також WORD надає ряд елементів автотексту, спеціально призначених для верхніх і нижніх колонтитулів.

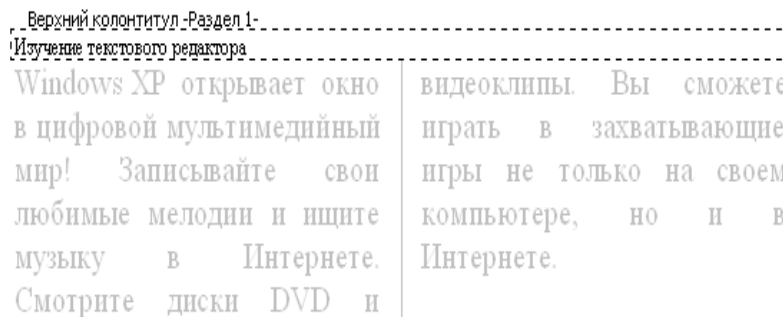
Для створення колонтитулів необхідно вибрати команду **Вид/Колонтитулы**. Для даної команди не має значення, де знаходиться курсор, оскільки колонтитули автоматично з'являються на кожній сторінці.

WORD активізує режим розмітки сторінки, встановлює курсор в області верхнього колонтитулу і відображає панель інструментів Колонтитули.






Кнопка для переходу у
верхній або нижній
колонтитул

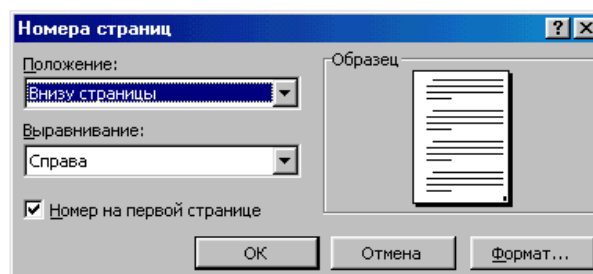
Ввести текст колонтитулу. Потім перейти в область нижнього колонтитулу і ввести його текст.



Для виходу з режиму створення колонтитулу необхідно клацнути на кнопці **Заккрыть**.

Кнопка  призначена для вставки номеру сторінки. Щоб вставити загальне число сторінок клацнути на кнопці . Кнопка  на панелі інструментів **Колонтитули** дає можливість створити вигляд номерів сторінок документа.

Номер сторінки можна поставити і використовуючись командою **Вставка/Номера страниц...**



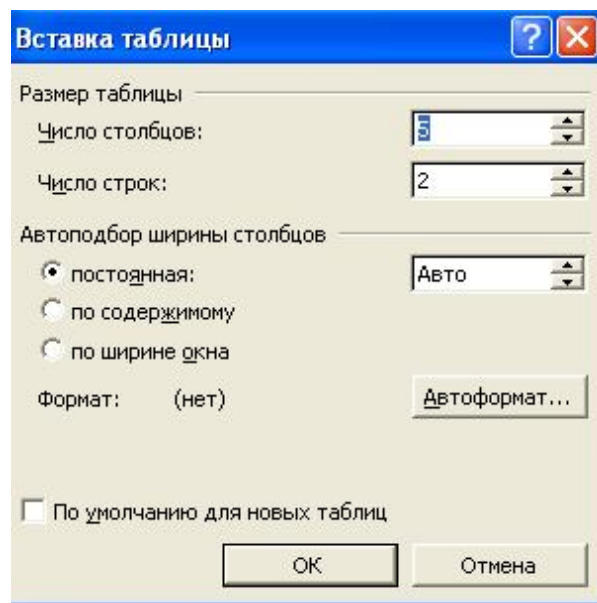
У діалоговому вікні можна вибрати де (угорі, внизу) та з яким вирівнюванням (уліво, вправо, по центрі) варто поставити номер сторінки. Якщо необхідно показувати номер першої сторінки, то позначити відповідний прапорець та натиснути кнопку **ОК**.

5.7 Робота з таблицями

Таблиці – надзвичайно гнучкий засіб. Вставивши таблицю в документ, можна змінити її структуру, вводячи або видаляючи рядки і стовпці, змінюючи ширину стовпів і т.п. Можна відформатувати таблиці різноманітними способами, вибравши відповідний вид обрамлення, заливку осередків, вирівнювання тексту усередині осередків і т.п. В таблиці можна вводити числа і виконувати прості обчислення над ними.

5.7.1 Створення таблиці

Для вставки у документ таблиці необхідно встановити курсор в місці, де повинна починатися таблиця і вибрати команду **Таблиця/ Додати/ Таблиця**. У діалоговому вікні **Вставка таблиці** ввести потрібну кількість рядків і стовпців таблиці.



Перемикач **Автоподбор ширины столбцов** дозволяє задати ширину стовпців таблиці:

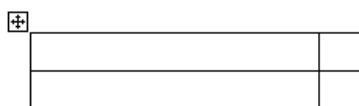
- 1) **постоянная** – стовпець має постійну ширину;
- 2) **по содержимому** – ширина стовпця міняється по міру введення тексту так, щоб весь текст помістився в один рядок;
- 3) **по ширине окна** – використовується тільки тоді, коли таблиця призначена для публікації в Internet.

Якщо заздалегідь невідомо, скільки рядків (стовпців) буде потрібно, необхідно вказати деяке первісне значення. Згодом рядки або стовпці можна додавати.

Для того, щоб додати рядок треба виділити рядок у таблиці та вибрати команду **Таблица/Добавить/Строки ниже** або **Таблица/Добавить/Строки выше**.

Для того, щоб додати стовець треба виділити стовець у таблиці та вибрати команду **Таблица/Добавить/Столбцы справа** або **Таблица/Добавить/Столбцы слева**.


Для переміщення таблиці необхідно встановити покажчик миші у будь-якому місці таблиці і у верхньому лівому кутку таблиці з'явиться маркер переміщення таблиці.



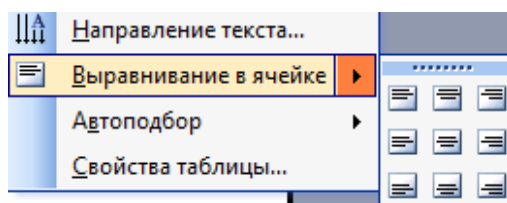
5.7.2 Форматування таблиці

Форматування таблиці складається із зміни виду тексту в елементах таблиці (за допомогою застосування форматування символів і абзаців), а також виду самих елементів таблиці (за допомогою обрамлення і заливки).

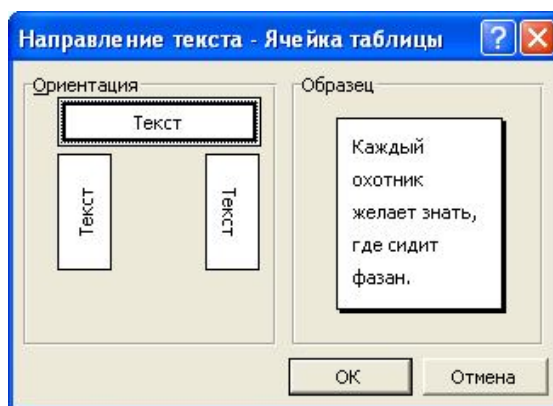
Перш ніж використовувати команди форматування осередок слід виділити. До тексту, поміщеного в елемент таблиці, можна застосувати всі види форматування, як і до звичайного тексту.

Для вирівнювання тексту по горизонталі в межах осередку можна скористатися кнопками на панелі інструментів .

Щоб задати вертикальне вирівнювання тексту необхідно виділити осередки і викликати контекстне меню. У меню вибрати команду **Выравнивание в ячейке**, а із списку необхідний варіант вирівнювання.



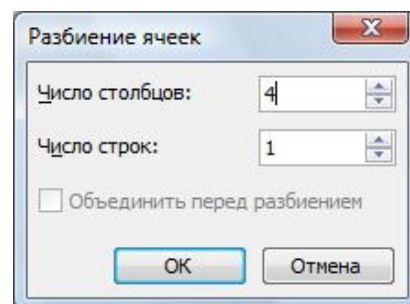
Для зміни напрямку тексту в елементах таблиці використовується команда **Формат/Направление текста**.



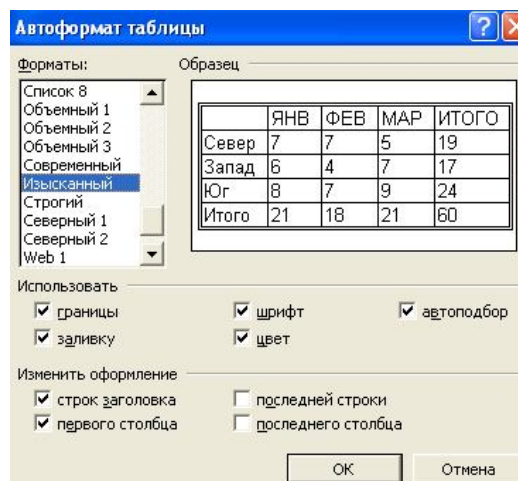
Маніпулюючи командами **Таблица/Объединить ячейки** і **Таблица/Разбить ячейки** можна створити таблиці різної структури.

Наприклад,

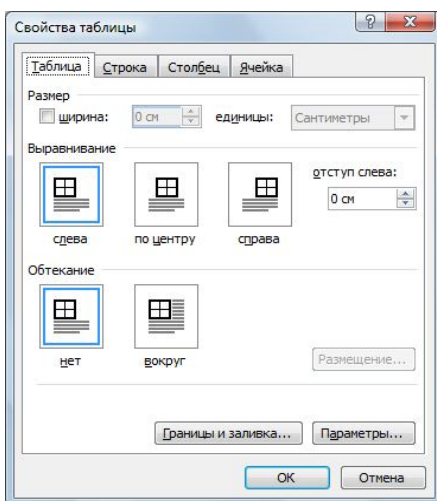
A diagram showing a table with 4 columns and 4 rows. An arrow points from a dialog box to the bottom-right cell of the table, which is being split into four equal-width columns.



Для створення рамок, виділення осередків кольором використовується команда **Формат/Границы и заливка** або можна скористатися командою **Таблица/Автоформат таблицы** для автоматичного форматування таблиці. У списку даної команди представлені різні стилі оформлення таблиці.



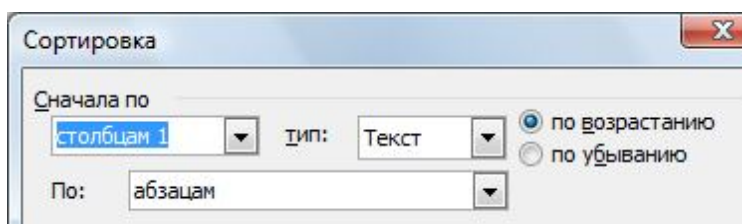
Щоб задати всі параметри форматування осередку, можна скористатися також командою **Таблиця/Свойства таблицы**.



5.7.3 Работа з даними в таблиці

WORD дозволяє не тільки вводити дані в таблицю, але і обробляти їх. Зокрема, дані можна упорядкувати за збільшенням або убуванням.

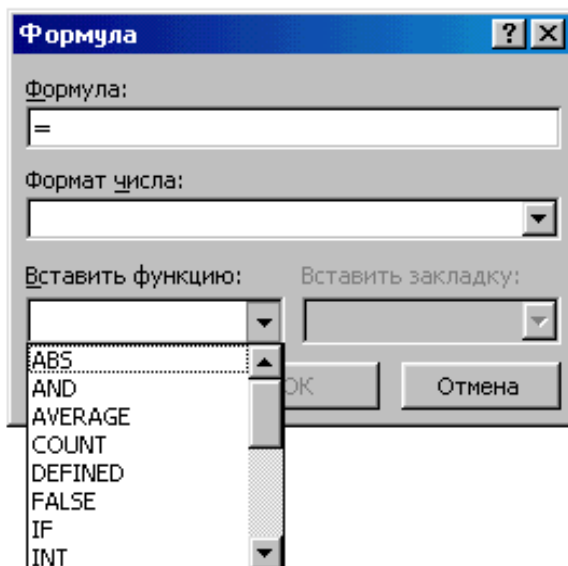
Для виконання сортування даних необхідно виділити данні у таблиці та вибрати команду **Таблиця/Сортировка...** У вікні треба вибрати стовпець, по якому необхідно сортувати таблицю, а також порядок (зростання або убування).



Елемент таблиці може містити не тільки дані, але і формули, в яких використані деякі математичні і логічні функції. Найчастіше такою формулою є підсумовування діапазону осередків вище або лівіше за ту, в яку введена формула. Але необхідно пам'ятати, що всі осередки таблиці мають свій індекс:

A1	B1	C1
A2	B2	...
A3

Отже, для того, щоб зробити розрахунок у стовпці або рядку, треба установити курсор у необхідному стовпці або рядку та вибрати команду **Таблиця/Формула...** З'явиться вікно, у якому варто вказати формулу для обчислення та діапазон осередків.



Формула обов'язково повинна починатися зі знака = (дорівнює). При створюванні формул можна використовувати функції зі списку **Вставити функцію**, а із списку **Формат числа** вибрати формат результату обчислень.

Наприклад, якщо у текстовому полі **Формула** записано =SUM(A1:D5), то результатом буде сума значень діапазону осередків A1:D5. Якщо формулу змінити: =SUM(A1;D5), то результатом буде сума значень осередків A1 і D5.

У WORD у відмінності від EXCEL при зміні даних значення функції не змінюється автоматично. Щоб відновити результат обчислень, необхідно виділити формулу і натиснути клавішу F9.

Найбільш часто використовувані функції в WORD:

- Average** середнє значення даних
- Max** найбільше значення в списку
- Min** найменше значення в списку
- Product** добуток значень
- Sum** сума значень

5.8 Графіка в документі WORD

5.8.1 Вставка картинки з колекції

Для того, щоб додати в документ картинку треба вибрати команду **Вставка/Рисунок/Картинки...** У вікні, що з'явиться, треба вибрати необхідну картинку та натиснути кнопку **Вставити**.

За допомогою миші можна переміщати картинку у документі, змінювати її розміри. Для зміни розмірів використовується кожний з маркерів, які виділяють картинку.

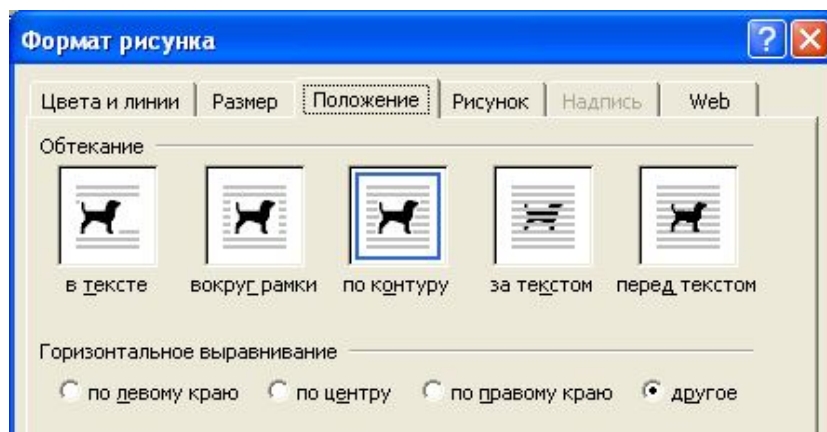
У редакторі WORD передбачена можливість "обтікання" картинки текстом.

Windows XP открывает окно в цифровой мультимедийный мир! Записывайте свои любимые мелодии и ищите музыку в Интернете. Смотрите диски DVD и видеоклипы. Вы сможете играть в захватывающие игры не только на своем компьютере, но и в Интернете.



Цей ефект можна застосовувати не до всіх картинок. Проте спробуйте клацнути правою кнопкою миші на картинці та у контекстному меню вибрати команду **Формат рисунка...**

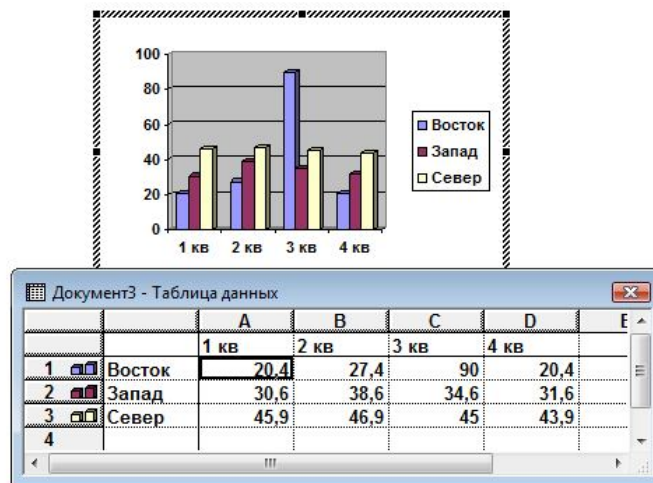
У вікні **Формат рисунка** треба переключитися на вкладку **Положение** та вибрати необхідний тип обтікання для малюнка. Зробивши вибір натиснути кнопку **ОК**.





5.8.2 Вставка діаграми

Діаграми є графічним представленням табличних даних і дозволяють зробити їх наочнішими. Демонструючи числа у вигляді картинок, добре розроблені діаграми допомагають представити інформацію більш дохідливо і наочно, ніж більші об'єми цифр.

Встановити текстовий курсор в те місце документа, де необхідно помістити діаграму і вибрати команду **Вставка/Рисунок/Діаграма**. У документі з'явиться зображення діаграми, таблиця даних, а у верхній частині вікна програми відобразиться панель інструментів Microsoft Graph.



Якщо запропонований тип діаграми не підходить, клацнути на кнопці в панелі інструментів  і вибрати інший тип діаграми. Ввести дані в осередки таблиці, які необхідно відобразити у вигляді діаграми. За умовчанням діаграма будується на основі рядів даних, розташованих в рядках. Якщо необхідно побудувати діаграму на основі рядів даних, розташованих в стовпцях, клацнути на кнопці в панелі інструментів .

Закінчивши побудову діаграми, клацнути в будь-якому місці документа за межами діаграми або таблиці даних. WORD повернеться в режим розмітки сторінки, таблиця зникне, а на екрані з'являться звичайні панелі інструментів.

Щоб відредагувати діаграму, яка вже вставлена в документ, необхідно двічі клацнути на ній мишею.

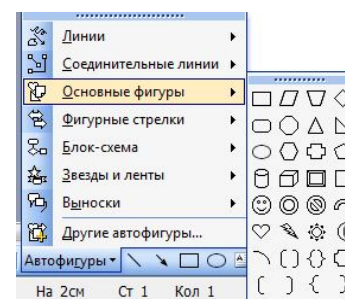
5.8.3 Використання автофігур і напису


Автофігури це об'єкти векторної графіки. Все, що можна намалювати в WORD, малюється з їх допомогою. Напис – це також об'єкт векторної графіки, в який вставлений текст.

Щоб намалювати автофігуру або напис необхідно підключити панель інструментів **Рисование** за допомогою команди **Вид/Панели инструментов/Рисование**. Ця панель включає різноманітні інструменти, необхідні для створення та редагування мальованих об'єктів і малюнків.



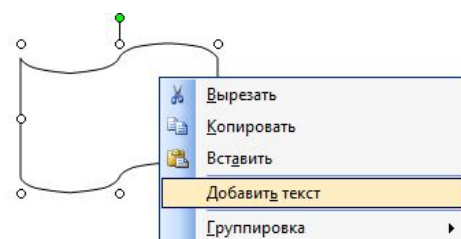
Для малювання автофігур використовується інструмент **Автофігури**.






Для малювання напису використовується кнопка  на панелі інструментів **Рисование**.

Для малювання об'єктів треба клацнути на кнопці обраного об'єкта (для активізації інструмента). Поставити покажчик у те місце, де повинен з'явитися об'єкт. Натиснути ліву кнопку миші, утримуючи її, перемістити покажчик, щоб створити об'єкт. Одержавши об'єкт необхідного розміру, відпустити кнопку миші. При створенні напису після створення рамки усередині неї з'явиться текстовий курсор, тобто необхідно ввести текст напису.

За умовчанням написи мають прямокутну форму, але зовсім не складно перетворити будь-яку автофігуру на напис. Для введення тексту в мальоване зображення клацнути правою кнопкою миші і вибрати команду **Добавить текст**. З'явився курсор введення в полі набору тексту даної фігури, набрати текст.





Для об'єкта, який виділено (клацнути на об'єкті) можна зробити:

- переміщати у документі шляхом хапання "об'єкта" усередині контуру та перетаскування у необхідне місце;
- масштабувати об'єкт шляхом перетаскування не самого об'єкта, а його маркерів;
- здійснювати поворот за допомогою переміщення маркерів у круговому напрямку;
- залити об'єкт за допомогою кнопки **Цвет заливки**  та **Способы заливки**, які надають можливість використовувати текстурні, візерункові та градієнтні заливання;
- надати ефект тіні за допомогою інструмента **Тень** . Набір тіней можна розширити, якщо вибрати на панелі тіней команду **Настройка тени** (змістити тінь, змінити колір тіні);
- створити ефект об'єму за допомогою інструмента **Объем** .

З об'єктами можна виконувати такі дії:


- групувати;
- розгрупувати;
- перегрупувати;
- змінити порядок розташування.

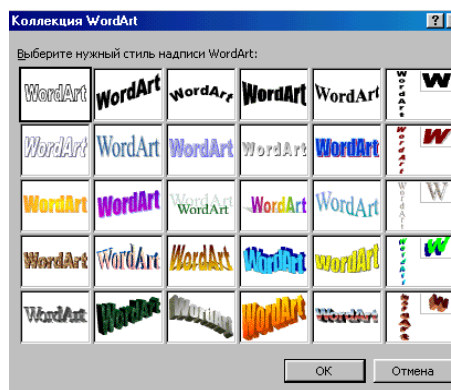
Для виконання цих дій використовується інструмент **Действия**.

Для роботи з групою об'єктів їх необхідно спочатку виділити тобто послідовно виділити кожен об'єкт, утримуючи при цьому клавішу **Shift**, або скористатися інструментом **Выбор объектов**  на панелі інструментів **Рисование**. Натисніть кнопку **Выбор объектов**  і, при натиснутій кнопці миші, перемістите покажчик так, щоб виділювані об'єкти укласти в рамку. Завершивши цю операцію, знову клацніть кнопку **Выбор объектов**, щоб відключити режим виділення.

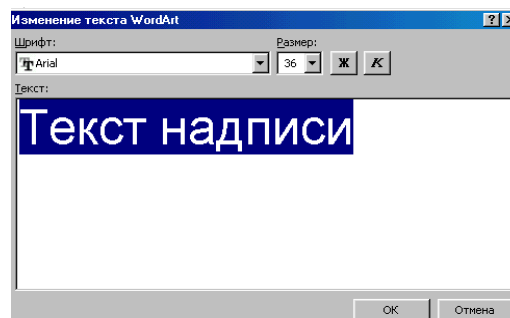
5.8.4 Використання об'єкту WordArt

Якщо написи це текст, нанесений на графіку, то об'єкти WordArt – це текст, який сам є графікою. Його можна використовувати для створення заголовків, рекламних текстів. WORD має в своєму розпорядженні цілу колекцію заготовок WordArt.

Для створення об'єкту треба намітити в документі місце, куди необхідно вставити об'єкт та вибрати команду **Вставка/Рисунок/Об'єкт WordArt...** або натиснути на кнопку  на панелі інструментів **Рисование**. З'явиться наступне діалогове вікно:



У даному вікні треба вибрати стиль напису та натиснути кнопку **ОК**. Після того, як обран стиль напису, необхідно вказати текст у наступному діалоговому вікні:



У даному вікні можна вказати тип і розмір шрифту, який буде використовуватися для напису.

Після завдання всіх параметрів треба натиснути кнопку **ОК**. Напис з'явиться в документі.

Об'єкти WordArt можна розтягувати, стискати, згинати і вписувати в різні геометричні фігури. Як до всякого векторного об'єкту до них можна

застосувати заливку, змінити колір і товщину ліній обведення, додати їм об'єм або тінь. Для цього використовується спеціальна панель, яка з'явиться якщо клацнути мишею на об'єкті.

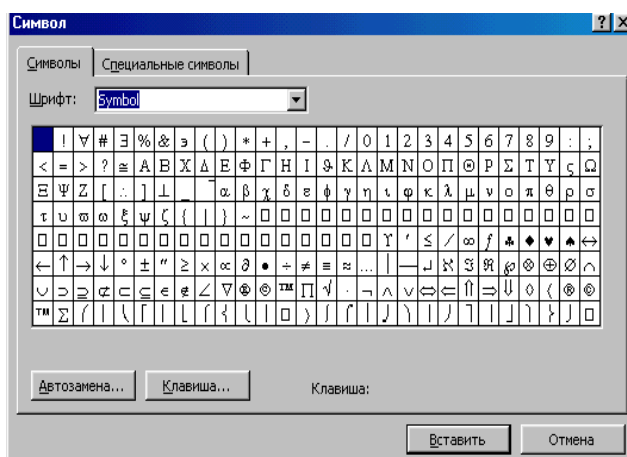


5.8.5 Набір математичних формул

Для набору щодо простих формул використовується наступний метод: символи, відсутні на клавіатурі вводяться з використанням команди **Вставка/Символ** і кнопок x^2 x_2 на панелі інструментів для позначення верхніх і нижніх індексів.

Для додавання у формулу нестандартного символу необхідно поставити текстовий курсор у необхідну позицію та вибрати команду **Вставка/Символ...** З'явиться таблиця символів поточного шрифту. Її можна поміняти (список **Шрифти**). Усі нестандартні символи знаходяться в шрифтах Symbol і Wingdings.

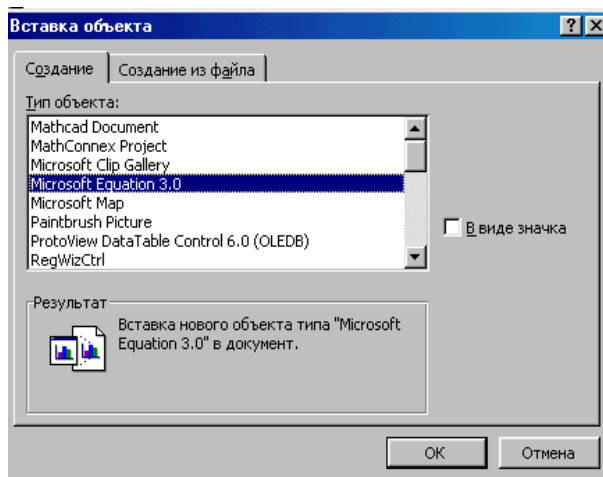
У таблиці виберіть той символ, що потрібний і двічі клацніть по ньому. Якщо вставляти більше нічого не треба, то закрити вікно.



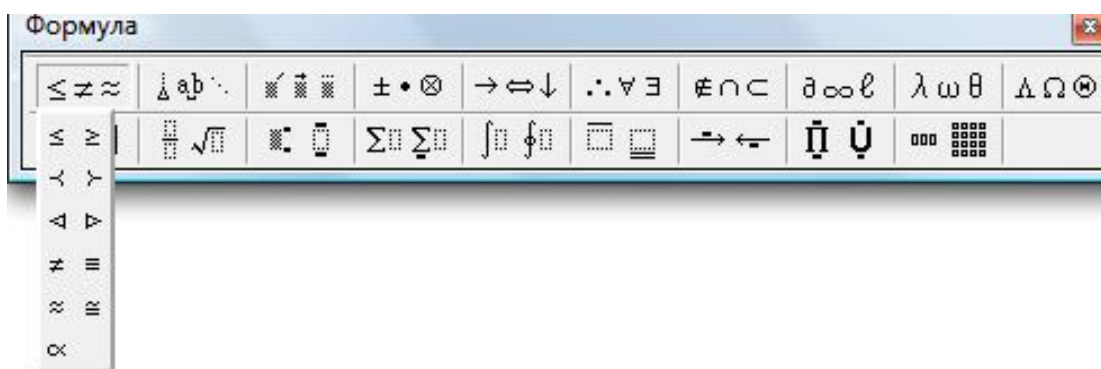
Для створення складних формул використовується спеціальний редактор математичних формул **Microsoft Equation 3.0**.

Для набору формули необхідно установити текстовий курсор у ту позицію, де варто помістити формулу, і вибрати команду **Вставка/Об'єкт...** У діалоговому вікні варто вибрати тип об'єкта - **Microsoft Equation 3.0** і

натиснути кнопку **ОК**. Потім можна приступити до редагування формули.

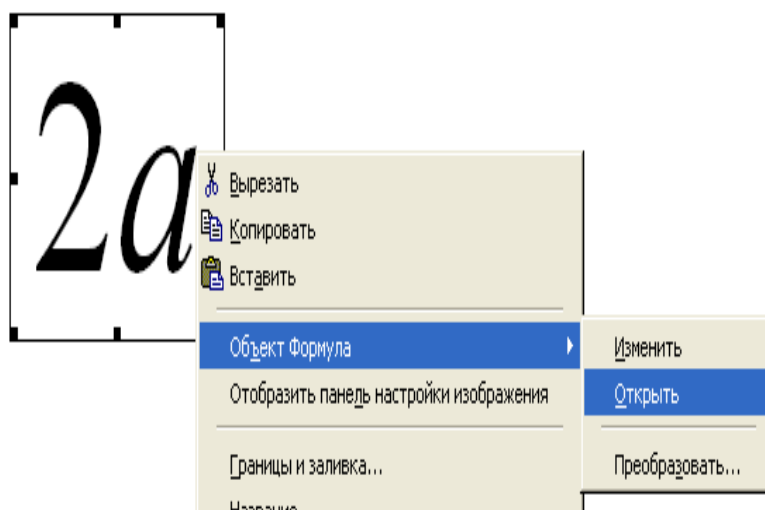


Якщо треба додати у формулу нові об'єкти (знаків сум, інтегралів і інші) необхідно використовувати спеціальну панель, що з'явиться на екрані. Кожна з кнопок цієї панелі містить цілий набір інструментів.



Для виходу з режиму введення і редагування формули необхідно клацнути за межами вікна редактора формул.

Створена формула схожа на графічний об'єкт. Її можна переміщати у документі та форматувати.



6. ЕЛЕКТРОННА ТАБЛИЦЯ MICROSOFT EXCEL

6.1 Інтерфейс електронної таблиці EXCEL

Електронна таблиця MICROSOFT EXCEL одна з найпопулярніших програм електронних таблиць. Нею користуються ділові люди, вчені, працівники бухгалтерії. З її допомогою ведуть різні списки, каталоги, таблиці; складають фінансові і статистичні звіти; обробляють результати наукового експерименту і т.п.

У програмі EXCEL можна використовувати безліч стандартних функцій для обчислення результату (фінансові, математичні, логічні, статистичні). Програма EXCEL має величезні можливості форматування даних, а також дозволяє побудувати за табличними даними графіки і діаграми. Оформлення таблиць може бути найрізноманітнішим.

Для відкриття електронної таблиці MICROSOFT EXCEL необхідно виконати команду **Пуск/Програми/ Microsoft Excel**. На екрані з'явиться вікно програми (див. рис.6.1), яке містить:

- 1 – рядок заголовка;
- 2 – кнопки управління вікном;
- 3 – рядок меню;
- 4 – панелі інструментів;
- 5 – адреса активного осередку;
- 6 – рядок формул;
- 7 – зміст активного осередку;
- 8 – активний осередок;
- 9 – заголовки стовпців;
- 10 – заголовки рядків;
- 11 – ярлики робочих листів;
- 12 – рядок стану;
- 13 – горизонтальна і вертикальна смуга прокрутки.

При запуску EXCEL відкривається чиста робоча книга, якою привласнюється ім'я **Книга1**. Подібно до всіх програм, написаних для

Windows, в EXCEL є рядок меню, за допомогою якого можна виконати різноманітні команди. На панелях інструментів розміщений цілий ряд кнопок, що дозволяють швидко виконати багато команд EXCEL.

На рис.6.1 показано, що виділений ярлик листа з ім'ям **Лист1**, тобто активний перший лист книги. Кожному стовпцю привласнена буква, а кожному рядку – номер. Таким чином, кожен осередок має свою адресу – комбінацію з букви, відповідної стовпцю, і номера рядка. Наприклад, осередок, що знаходиться на перетині стовпця Е і рядка 7, називається осередком Е7. **Активний осередок**, тобто осередок, до якого застосовуються всі виконувані в даний момент дії, виділяється товстою рамкою. Зміст активного осередку показується в рядку формул.

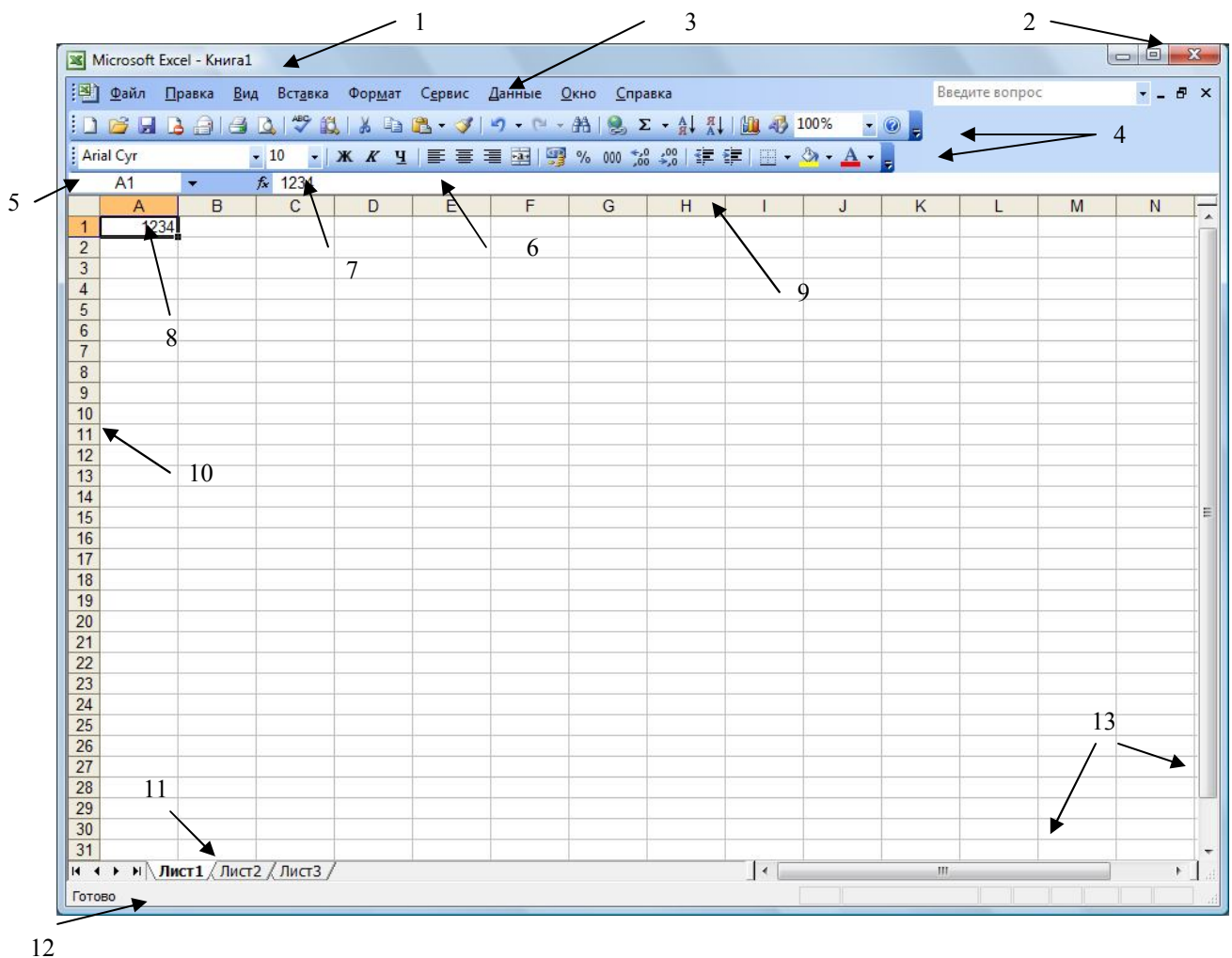




Рис.6.1 - Зовнішній вигляд вікна електронної таблиці MICROSOFT EXCEL

Рядок стану, що знаходиться у нижній частині екрану, містить корисну інформацію про те, що в даний момент робить EXCEL. На рис.6.1. показано, що в рядку стану написано **Готово**, це означає, що EXCEL виконав попередню операцію і готовий прийняти нові дані.

6.2 Основні принципи роботи в EXCEL

Будь-який осередок в EXCEL можна зробити активним, помістивши покажчик миші на осередок і натиснувши ліву кнопку миші.

Активізувавши осередок, можна вводити в нього дані: текст, числа, дати, час або формули. Під час набору дані з'являються в активному осередку і в рядку формул. У момент початку введення в рядку формул з'являються дві кнопки: **Ввод**  і **Отмена** . У рядку стану написано **Ввод**. Це свідчить про те, що йде процес введення даних. Після введення даних натиснути клавішу **Enter**.

Програма EXCEL дозволяє вводити дані, ширина яких перевищує ширину осередку. Довгі рядки тексту можуть переходити на сусідні осередки, якщо вони не зайняті. Інакше текст обривається на межі осередку. Якщо числа виявляються ширшими за осередок, то вони виводяться або в показовій формі (наприклад, 1.23E+12) або замінюються знаками фунта (#). Але навіть якщо дані показуються лише частково, EXCEL все одне зберігає їх і використовує в обчисленнях повністю. Крім того, і текст, і числа будуть виведені цілком, якщо збільшити ширину стовпця до потрібного розміру.

Для зміни ширини стовпця за допомогою миші необхідно виконати наступні дії:

- помістити покажчик миші на праву межу заголовка стовпця, ширину якого необхідно змінити. Якщо правильно встановлений покажчик миші, він перетвориться на горизонтальну двонаправлену стрілку;

- для збільшення або зменшення ширини стовпця необхідно перемістити стрілку управо або вліво відповідно. Пунктирна лінія показуватиме ширину стовпця;

- відпустити кнопку миші, якщо ширина стовпця відповідає вимогам.


Для зміни висоти рядка за допомогою миші необхідно виконати наступні дії:

- помістити покажчик миші на нижню межу заголовка того рядка, висоту якого необхідно змінити. Покажчик миші прийме вид вертикальної двонаправленої стрілки;
- перемістити вертикальну двонаправлену стрілку вниз або вгору для збільшення або зменшення висоти рядка відповідно;
- відпустити кнопку миші, якщо висота рядка відповідає вимогам.

Існує багато причин, з яких виникає необхідність вносити зміни до осередку: зміна запису, коректування даних, уточнення введеної формули і т.п. Для редагування необхідно активізувати осередок, зміст якого необхідно змінити, і клацнути в рядку формул. При зміні змісту осередку можна використовувати клавіші-стрілки або мишу для переміщення курсору, а для видалення символів клавіші **Delete** або **Backspace**. Після закінчення редагування натиснути клавішу **Enter**.

6.3 Створення робочої книги і її збереження


Коли вперше створюється в EXCEL робоча книга, вона тимчасово зберігається в оперативній пам'яті комп'ютера. Якщо вимкнути комп'ютер або в мережі випадково пропаде напруга, результати роботи зникнуть миттєво і безповоротно. Точніше кажучи, це відбудеться в тому випадку, якщо вчасно не зберегти дані.

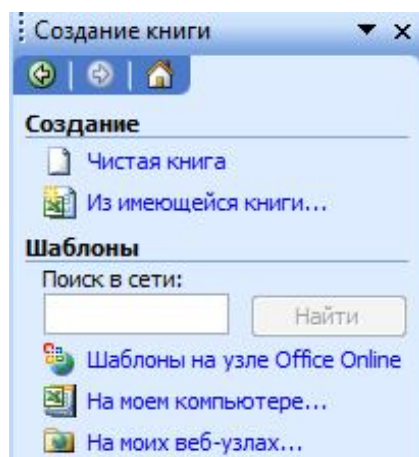
Для збереження робочої книги можна скористатися кнопкою на панелі інструментів  або командою **Файл/Сохранить**.


У діалоговому вікні, що з'явиться **Сохранение документа** необхідно ввести ім'я файлу, вибрати із списку **Папка** папку, в яку необхідно зберегти документ. Потім клацнути на кнопці **Сохранить**.

Зберігаючи файл в процесі роботи, вікно **Сохранение документа** більше відображатися не буде.

Якщо ж необхідно мати в своєму розпорядженні і стару версію робочої книги, і версію з внесеними змінами, зберегти робочу книгу необхідно під іншим ім'ям, використовуючи команду **Файл/Сохранить как**. Знов з'явиться діалогове вікно **Сохранение документа**, і можна ввести нове ім'я і вибрати іншу папку.

Щоб почати нову робочу книгу необхідно в панелі інструментів клацнути на кнопці  або вибрати команду **Файл/Создать**. У вікні, що з'явиться, вибрати пункт **Чистая книга**.



Після закінчення роботи з EXCEL доцільно його закрити. Для цього в меню **Файл** вибрати команду **Выход** або клацнути на кнопці закриття EXCEL (кнопка  в правому кутку рядка заголовка). Якщо робоча книга збережена, вікно EXCEL закриється, інакше EXCEL запитає, чи хочете зберегти книгу. Необхідно клацнути на кнопці **Да**, якщо необхідно зберегти дані, і на кнопці **Нет**, якщо зміни потрібно відмінити. Щоб повернутися до роботи над книгою, не покидаючи програми EXCEL, клацнути на кнопці **Отмена**.

6.4 Дії з робочими листами

Робоча книга подібна до папки-швидкозшивача, в якій зібрана безліч робочих листів. Якийсь час можна користуватися яким-небудь одним з них, проте іноді зручно в одній книзі розмістити відразу декілька взаємозв'язаних робочих листів.

Щоб проглянути новий лист, клацнути на вкладці листа в нижній частині вікна робочої книги. Це називається активізацією робочого листа.

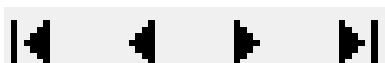
Спочатку програма EXCEL привласнює робочим листам такі імена **Лист1**, **Лист2** і т.п. Краще всього перейменувати листи так, щоб їх назви відображали зміст; тоді можна буде швидко знайти потрібні дані.

Для перейменування листа необхідно двічі клацнути на його вкладці лівою кнопкою миші, ввести ім'я робочого листа і натиснути клавішу **Enter** або на вкладці листа клацнути правою кнопкою миші і в контекстному меню вибрати команду **Переименовать**.

Нові робочі книги включають спершу три робочих листа. Якщо необхідна більша кількість робочих листів, їх можна додати. Для вставки нового листа необхідно вибрати лист, розташований праворуч від того листа, де повинен з'явитися новий, оскільки при вставці робочого листа програма EXCEL поміщає його перед поточним листом. На вкладці листа клацнути правою кнопкою миші і в контекстному меню вибрати команду **Добавить**. Якщо ж новий лист з'явився не в тому місці, де необхідно, його можна перемістити. Для переміщення листа необхідно клацнути на його вкладці лівою кнопкою миші і, не відпускаючи її, перемістити його на нове місце.

Для видалення робочого листа, клацнути правою кнопкою миші на вкладці листа і в контекстному меню вибрати команду **Удалить**. Програма EXCEL запитає підтвердження на видалення. Необхідно у відповідь клацнути на кнопці **ОК**.

Якщо робоча книга містить багато робочих листів, і не видно вкладки потрібного листа, на який необхідно перейти, можна скористатися кнопками прокрутки вкладок, розташованих зліва від вкладок. Прості стрілки дозволяють переміщатися на один лист за раз в напрямі, вказаному стрілкою. Стрілки з рисками дозволяють переміститися до останнього або до першого робочого листа.



6.5 Основи форматування робочого листа

EXCEL володіє цілим арсеналом функцій форматування, за допомогою яких можна поліпшити зовнішній вигляд робочих листів як при відображенні їх на екрані, так і при друці. Перш ніж виконувати які-небудь операції форматування, необхідно вказати програмі, яких осередків повинні торкнутися зміни. Для цього осередки необхідно виділити.

6.5.1 Виділення групи осередків

Група осередків називається **інтервалом**, вони забарвлюються в синій колір. Виняток становить активний осередок, який залишається білим. Можна виділити будь-який чотирикутний блок осередків, а також цілі стовпці, рядки або всю таблицю даних.

Для виділення інтервалу необхідно:

- натиснути ліву кнопку миші на осередку, з якого необхідно почати виділення;
- не відпускаючи кнопки, перемістити курсор миші по інтервалу осередків;
- коли досягнутий останній осередок вибраного інтервалу, відпустити кнопку миші.

Для того, щоб виділити декілька інтервалів відразу необхідно:

- відзначити один інтервал;
- натиснути клавішу **Ctrl** і, не відпускаючи її, відзначити інший інтервал.




Для виділення всього рядка або стовпця клацнути мишею на заголовку рядка або стовпця відповідно. Показчик миші повинен мати вигляд **➔** або **⬇** відповідно. Для виділення декількох сусідніх рядків або стовпців необхідно протягнути по їх заголовках курсор миші при натиснутій лівій кнопці миші.

Щоб відмінити виділення осередків, клацнути в будь-якому місці робочого листа.

6.5.2 Зміна зовнішнього вигляду інформації осередків

Перед зміною зовнішнього вигляду необхідно виділити осередок або інтервал осередків, до яких необхідно застосувати однакові режими форматування і використовувати один із способів.

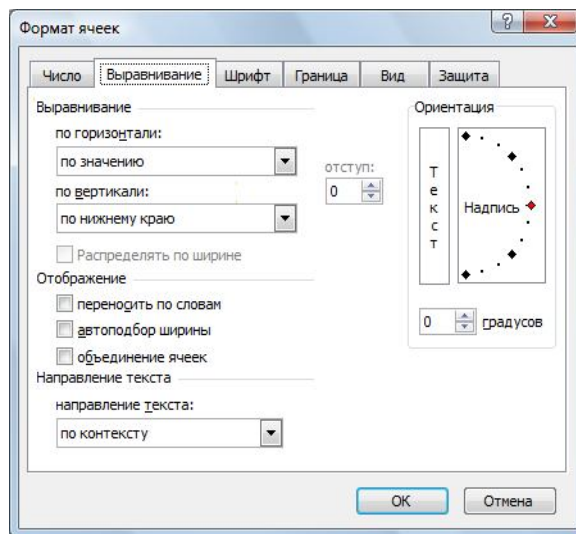
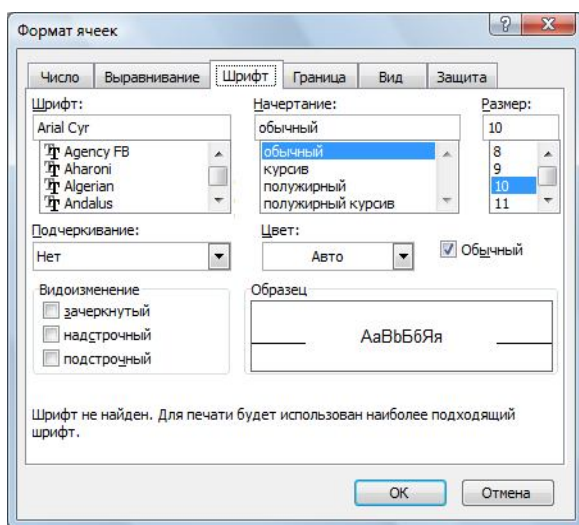
1) Кнопки на панелі інструментів **Форматування**.

Для зміни шрифту вибрати необхідний шрифт з розкриваючого списку **Шрифт** *Arial Cyr* . Розмір шрифту вибирається з розкриваючого списку **Розмір** *10*  або вводиться нове значення. Для зміни зображення на панелі інструментів використовуються кнопки **Ж** (напівжирний), **К** (курсив), **Ц** (підкреслений) . Можна використовувати комбіноване зображення по своєму вибору.

2) Команда **Формат/Ячейки**.

У діалоговому вікні, що з'явиться, **Формат ячеек** вибрати закладку **Шрифт**. На вкладці вибрати необхідний тип шрифту, його розмір, вид зображення. При необхідності вибрати тип підкреслення, колір шрифту. Можна встановити різні ефекти.

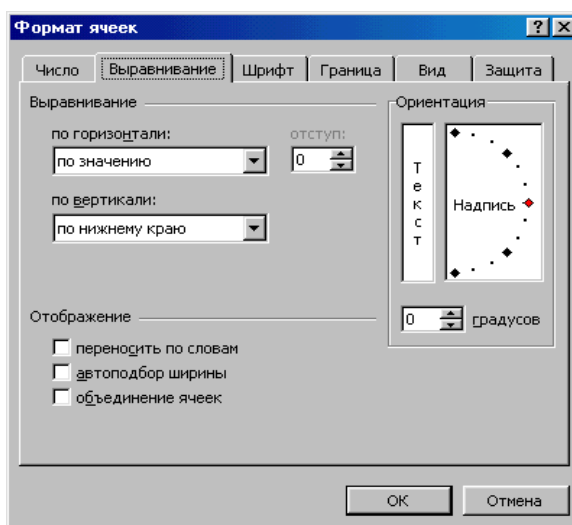
На вкладці **Выравнивание** можна змінити розташування тексту або ввівши кут повороту тексту в полі введення градусів, або клацнувши на відмітці з потрібним значенням кута у вікні перегляду справа. Слово **Текст** обернеться на заданий кут.







Після закінчення всіх установок клацнути на кнопці **ОК**.

За умовчанням EXCEL вирівнює текстові дані в осередку по лівому краю, а числа – по правому, включаючи і результати розрахунків по формулах. Для вирівнювання даних необхідно виконати наступні дії:

- виділити осередок або інтервал осередків, в яких необхідно змінити спосіб вирівнювання;
- вибрати команду **Формат/Ячейки**. У діалоговому вікні, що з'явиться, **Формат ячеек** вибрати вкладку **Выравнивание**;
- вибрати спосіб вирівнювання і клацнути на кнопці **ОК**.



Для вирівнювання даних в осередку можна також використовувати кнопки на панелі інструментів: по лівому краю , по центру , по правому краю .

Щоб розташувати заголовок по центру відносно декількох стовпців, необхідно виділити його, а також стовпці, щодо яких необхідно відцентрувати даний заголовок. Потім клацнути на кнопці панелі інструментів **Объединить и поместить в центре**  або скористатися командою **Формат/Ячейки**, вкладка **Выравнивание**, прапорець на опції **Объединение ячеек**.

В результаті установки прапорця **Переносить по словам** на вкладці **Выравнивание** широкий текстовий рядок розпадається на декілька рядків в межах одного осередку, але не «заповзатиме» управо на сусідні осередки. Внаслідок цього може збільшитися висота рядка.

6.5.3 Форматування чисел

Програма EXCEL володіє широким асортиментом форматів чисел, вони дозволяють забезпечити числові дані знаком долара, комами, що розділяють кожні три розряди, позначенням відсотків і т.п.

Числові формати розповсюджуються на весь осередок, а не тільки на окремі частини її змісту.

Для зміни формату чисел необхідно виконати наступні дії:

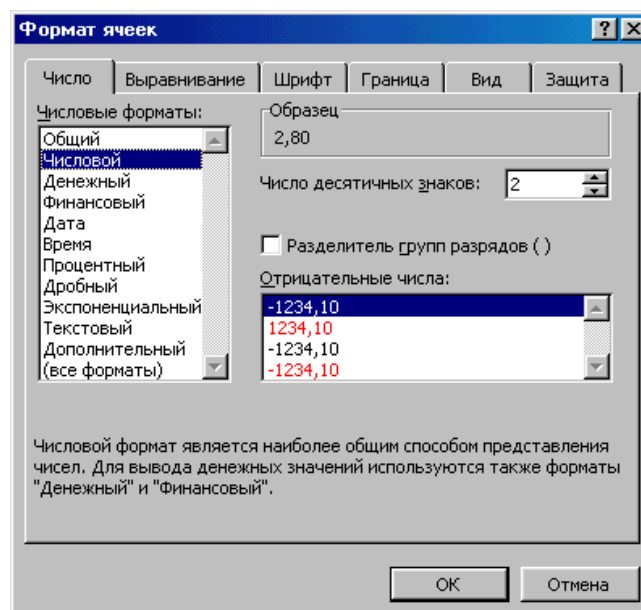
- виділити осередок, що містить число, яке необхідно відформатувати;

- вибрати команду **Формат/Ячейки**.

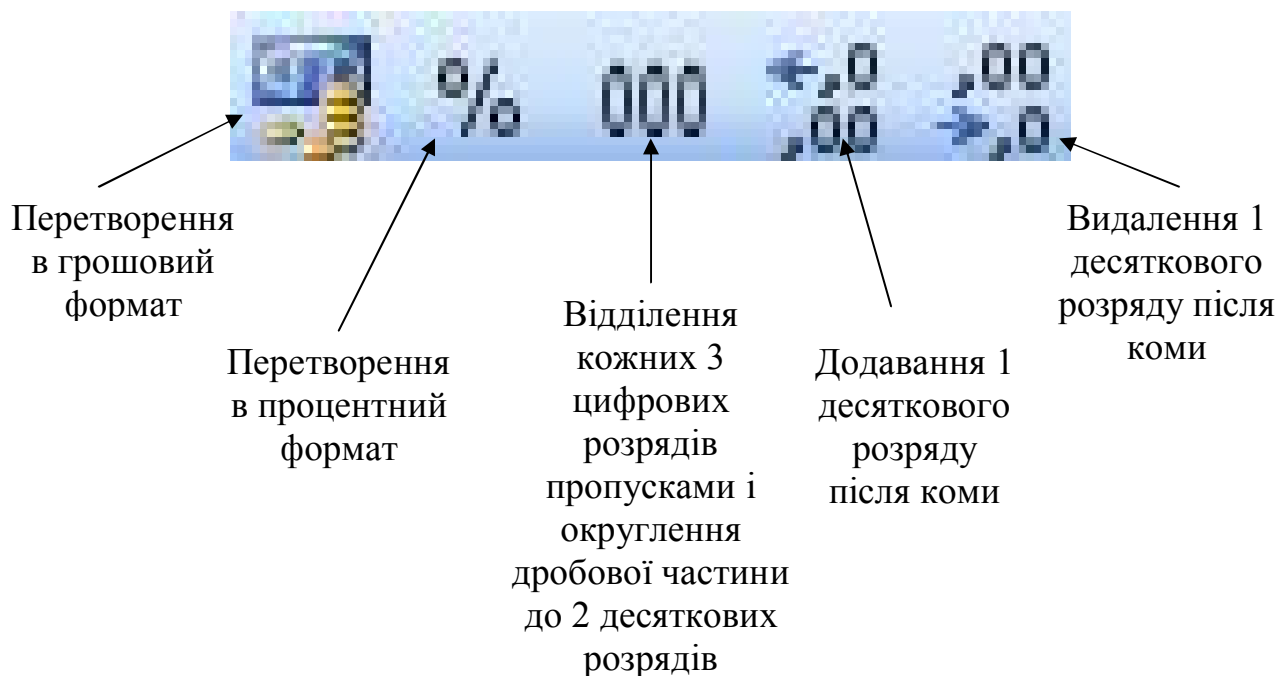
- у діалоговому вікні, що з'явиться, **Формат ячеек** клацнути на вкладці **Число**;

- із списку **Числовые форматы** вибрати потрібний тип числових даних. Залежно від вибраного числового формату, можна задавати кількість десяткових знаків після коми в представленні чисел, символ грошової одиниці і спосіб уявлення на екрані від'ємних чисел (за умовчанням перед від'ємними числами ставиться знак мінус). Приклад числа з вибраним форматом відобразиться в полі **Образец**;

- клацнути на кнопці **ОК**.



Певні формати зручно застосовувати, клацаючи на відповідній кнопці панелі інструментів. На панелі інструментів є також кнопки, за допомогою яких можна вводити або прибирати додаткові десяткові розряди після десяткової крапки.



Команда **Формат/Ячейки** також використовується для зміни поточного формату часу і дати. Для форматування дати необхідно вибрати в списку **Числовые форматы** пункт **Дата**, а для форматування часу – пункт **Время**.

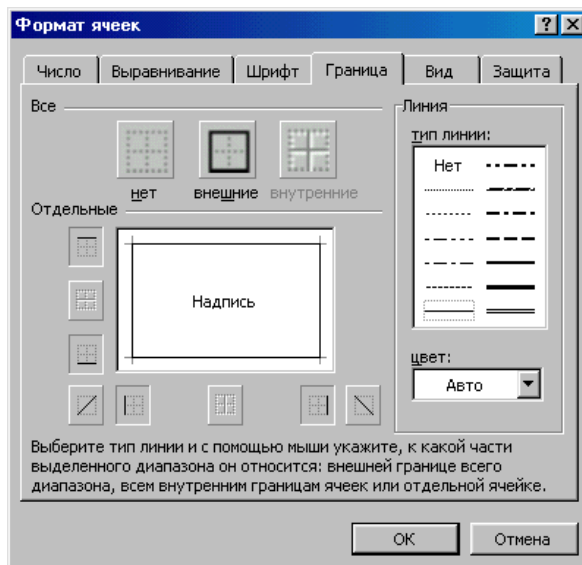
6.5.4 Обрамлення і заливка осередків

Обрамлення і заливка є основними засобами виділення важливої інформації і додання таблицям елегантного вигляду як на екрані монітора, так і при виведенні даних на друк.

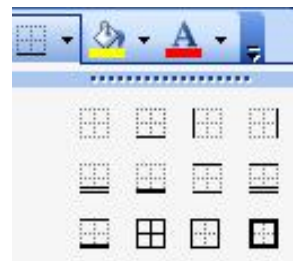
Для виконання обрамлення необхідно виконати наступні дії:

- виділити осередок або інтервал осередків, які необхідно відформатувати;
- вибрати команду **Формат/Ячейки**. У діалоговому вікні, що з'явиться, **Формат ячеек** вибрати вкладку **Граница**;

- вибрати вид межі: контур, зліва, справа, зверху або знизу;
- у зоні **Линия** вибрати тип лінії; колір лінії можна вибрати з розкриваючого списку **Цвет**;
- клацнути на кнопці **ОК**.

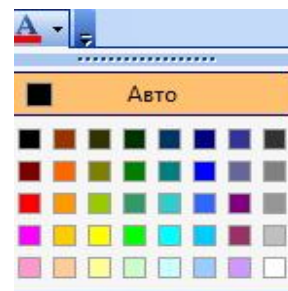


Для виконання обрамлення осередків можна використовувати кнопку на панелі інструментів **Границы**. Із списку даної кнопки, що розкривається, можна вибрати необхідний тип лінії.



Щоб створити заливку у виділеному осередку або групі осередків, клацнути на вкладці **Вид** діалогового вікна **Формат ячеек**. Потім вибрати потрібний колір або заливку. За бажанням можна також вибрати фоновий узор осередку.

Кнопка на панелі інструментів **Цвет заливки** також дозволяє швидко змінити забарвлення осередків. Для цього досить клацнути на стрілці, розташованій поряд з кнопкою і вибрати відповідний колір з колірної палітри, що з'явилася.



6.6 Зміна структури робочої таблиці

6.6.1 Додавання і видалення рядків і стовпців

Оскільки робота не стоїть на місці, зазнають зміни і робочі таблиці: введення нових даних, перенесення на інше місце і видалення наявних даних і т.п. Можна додавати нові дані за допомогою вставки в таблицю стовпців і рядків, а також видаляти цілі рядки і стовпці із застарілими даними і звільняти місце для нових даних.

Для вставки рядка необхідно виконати наступні дії:

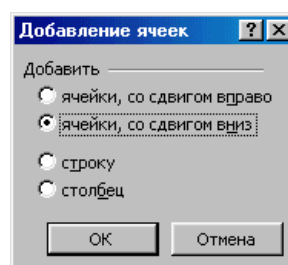
- виділити осередок в рядку, над яким необхідно вставити новий рядок;
- вибрати команду **Вставка/Строки**. Програма EXCEL вставить новий рядок, а існуючі рядки зрушаться вниз.

Для вставки стовпця необхідно виконати наступні дії:

- виділити осередок в стовпці, зліва від якого необхідно вставити новий стовець;
- вибрати команду **Вставка/Столбцы**. Програма EXCEL вставить новий стовець, а існуючі стовпці зрушаться управо.

Для вставки осередку або інтервалу осередків необхідно виконати наступні дії:

- виділити осередок або інтервал і вибрати команду **Вставка/Ячейки**. З'явиться діалогове вікно **Добавление ячеек**;
- щоб вставлені осередки з'явилися зліва від виділених, необхідно вибрати опцію «**ячейки, со сдвигом вправо**». Щоб нові осередки з'явилися над виділеними, необхідно вибрати опцію «**ячейки, со сдвигом вниз**»;
- клацнути на кнопці **ОК**. Виділені осередки зрушаться у вказаному напрямі.



Існує два способи видалення даних з робочих таблиць. Можна видалити осередки, виключивши їх з робочої таблиці і заклавши вакантні місця, що утворилися, шляхом переміщення осередків знизу або справа. Можна також очистити осередки даних, прибравши з них зміст і залишивши їх порожніми, без видалення осередків, що звільнилися, з робочої таблиці. Часто звільняється цілий рядок або стовпець.

Для видалення рядка необхідно виконати наступні дії:

- клацнути на номері рядка, який необхідно видалити. Для видалення декількох рядків необхідно виділити кожен рядок окремо;
- вибрати команду **Правка/Удалить**. Виділені рядки будуть видалені, а існуючі рядки зрушаться вгору.

Для видалення стовпця необхідно виконати наступні дії:

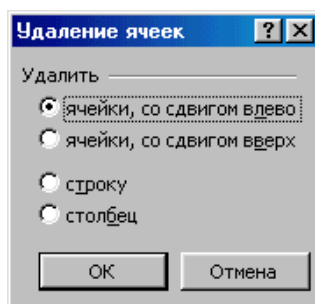
- клацнути на заголовку стовпця, який необхідно видалити. Для видалення декількох стовпців необхідно виділити їх;
- вибрати команду **Правка/Удалить**. Виділений стовпець зникне, а існуючі стовпці зрушаться вліво.

Для видалення осередку або інтервалу осередків необхідно виконати наступні дії:

- виділити осередки або інтервал, які необхідно видалити;
- вибрати команду **Правка/Удалить**. З'явиться діалогове вікно

Удаление ячеек;

- щоб існуючі дані після видалення перемістилися вліво, необхідно вибрати опцію «**ячейки, со сдвигом влево**». Для переміщення даних після видалення вгору, необхідно вибрати опцію «**ячейки, со сдвигом вверх**»;
- клацнути на кнопці **ОК**.



6.6.2 Копіювання і переміщення даних

Існує декілька способів для виконання цих операцій.

Метод «перетягнути і відпустити» дозволяє переміщати або копіювати дані шляхом переміщення їх при натиснутій кнопці миші і потім фіксації в тому місці, де кнопка миші буде відпущена. Цей метод особливо ефективний для переміщення і копіювання даних, коли вона знаходяться в полі видимості в межах екрану.

Для здійснення цього способу необхідно виконати наступні дії:

- виділити осередок або інтервал осередків, які необхідно перемістити або скопіювати на інше місце;
- встановити покажчик миші на межу осередку або інтервалу осередків, при цьому покажчик миші прийме форму стрілки;
- утримуючи натиснутою ліву кнопку миші, перетягнути виділені осередки в потрібне місце. Прямокутний контур показує, де розташовуватимуться переміщені або копійовані осередки, якщо в даний момент відпустити кнопку миші.

Щоб перемістити осередки, досить відпустити кнопку миші, коли контур опиниться в потрібному місці.

При копіюванні осередків, відпускаючи кнопку миші, утримувати клавішу **Ctrl** в натиснутому положенні. Не можна відпускати клавішу **Ctrl** раніше, ніж кнопку миші, тому що у такому разі дані не скопіюються, а перемістяться.

Даний метод зручний, але у нього є деякі недоліки. Зокрема, цей метод не дуже зручно використовувати при переміщенні або копіюванні даних на великі відстані, наприклад з осередку A1 в осередок Z115 або за межі видимості на екрані. В цьому випадку можна скористатися буфером обміну. Дані з буфера обміну можуть бути вставлені в інше місце робочого листа, нової робочої книги або навіть в інші програми.




Для копіювання або переміщення даних необхідно виконати наступні дії:

- виділити осередок або інтервал осередків, які необхідно перемістити або скопіювати на інше місце;

- вибрати команду **Правка/Копировать** (для копіювання) або команду **Правка/Вырезать** (для переміщення). Навколо виділеної області з'явиться рамка, що біжить, яка указує, що дані поміщені в буфер обміну;

- активізувати осередок, в який необхідно розмістити дані;

- вибрати команду **Правка/Вставить**.

Для виконання цих операцій також існують кнопки на панелі інструментів: **Вырезать** , **Копировать** , **Вставить** .

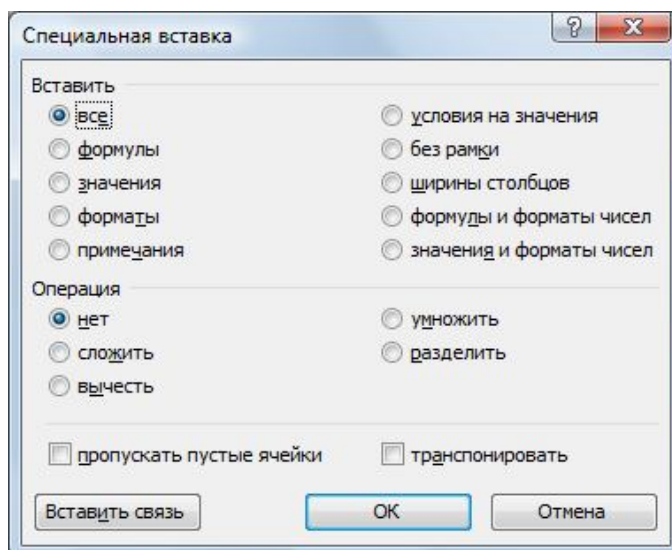
Для того, щоб дані у всіх копіях змінювалися при зміні їх в початковому інтервалі осередків, необхідно виконати копіювання з використанням команди **Специальная вставка**.

Для копіювання даних необхідно виконати наступні дії:

- виділити осередок або інтервал осередків, які необхідно скопіювати на інше місце;

- вибрати команду **Правка/Копировать**. Навколо виділеної області з'явиться рамка, що біжить, яка указує, що дані поміщені в буфер обміну;

- вибрати команду **Правка/Специальная вставка**. У діалоговому вікні, що з'явиться, **Специальная вставка** необхідно клацнути на кнопці **Вставить связь** і клацнути на кнопці **ОК**.



6.7 Створення формул

6.7.1 Створення простих формул

У програмі EXCEL формула може бути простою, як $=2+2$ (у EXCEL всі формули починаються із знаку рівності). Але корисно застосовувати формули з посиланнями на осередки замість фіксованих чисел.

Посилання на осередки – це просто адреси осередків, повідомляючи EXCEL, що слід проводити обчислення над поточним змістом вказаних осередків.

Якщо згодом дані, на які посилаються, будуть змінені, то результат формули зміниться автоматично.

Для введення формули необхідно виділити осередок, в який необхідно ввести формулу і ввести знак рівності, а потім написати формулу.

Формула, що вводиться, відобразиться в активному осередку і в рядку формул. Після введення формули натиснути клавішу **Enter**.

У активному осередку з'явиться результат, а в рядку формул буде показана формула.

Для виконання обчислень використовуються наступні математичні дії: + для складання, - для віднімання, * для множення, / для ділення, ^ для піднесення до ступеня.

Більшість помилок у формулах відбуваються із-за порушення необхідного порядку дій, послідовності виконання математичних операцій. Пріоритет виконання операцій:

1 рівень - піднесення до ступеня

2 рівень – множення, ділення

3 рівень – складання, віднімання

Якщо формула включає математичні операції однакового рівня, обчислення проводяться зліва направо.

У більшості випадках програма EXCEL відображає повідомлення про помилки у формулах, що дозволяє її виправити (див. табл.6.1).

Таблиця 6.1 - Повідомлення про помилки у формулах.

Помилка	Сенс
#ДЕЛ/0	у формулі присутнє ділення на нуль
#Н/Д	немає доступного значення
#ИМЯ?	програма EXCEL не може розпізнати ім'я, використовуване у формулі
#ЧИСЛО!	некоректні дані
#ССЫЛКА!	у формулі не правильне посилення на осередок
#ЗНАЧ!	недоступний тип аргументу

Створення формули розглянемо на прикладі.

Припустимо, що необхідно в осередок В3 ввести формулу, яка дозволить обчислити суму чисел, значення яких знаходяться в осередках А1 і А2. Для цього необхідно виконати наступні дії:

- виділити осередок В3 і ввести знак рівності (=) ;
- клацнути на осередку А1, щоб ввести у формулу адрес першого осередку;
- ввести знак складання (+);
- клацнути на осередку А2, щоб ввести у формулу адрес другого осередку;
- для отримання результату натиснути клавішу **Enter**.


СУММ		=A1+A2	
	A	B	C
1	12,5		
2	5,72		
3		=A1+A2	

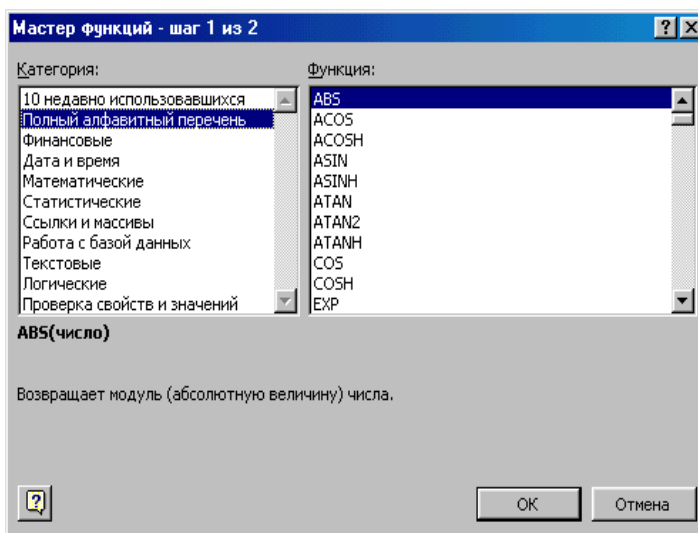
6.7.2 Створення формул з використанням Майстра функцій

Формула є деяким набором інструкцій для комп'ютера, за допомогою яких проводяться обчислення і виводяться отримані результати. Функція є вбудованою формулою, яку EXCEL використовує для швидкої організації спеціальних і деколи складних розрахунків.

Вбудована функція складається із знаку рівності (=), імені і аргументу – це діапазон осередків, зміст яких використовується для обчислень.

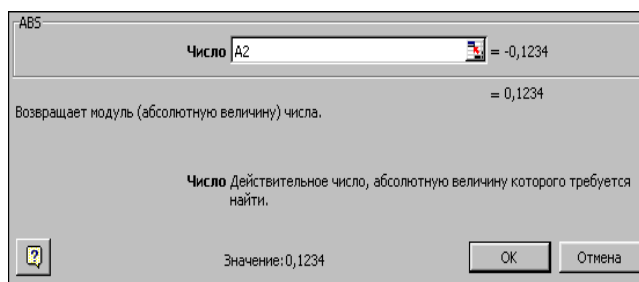
Для введення функції в активний осередок необхідно набрати знак рівності (=), ім'я функції і відкриваючу дужку. Потім задати діапазон і закрити дужку. Після натиснення клавіші **Enter** програма відображає результат.

Якщо немає упевненості в тому, як працює та або інша функція, то можна використовувати Майстер функцій. Для цього необхідно вибрати команду **Вставка/Функция** або скористатися кнопкою на панелі інструментів . У діалоговому вікні, що з'явиться, **Мастер функций** в списку **Категория** вибрати тип вбудованої функції, а потім в списку **Функция** – ім'я функції і клацнути по кнопці **ОК**.



У вікні вибраної функції, що з'явиться, пропонується ввести аргумент функції (адрес осередку або інтервал осередків) і клацнути по кнопці **ОК**. Для введення значень аргументів в текстове поле **Число** використовується

миша, за допомогою якої необхідно виділити осередки або інтервал осередків.



6.7.3 Приклади створення формул

Приклад 1.

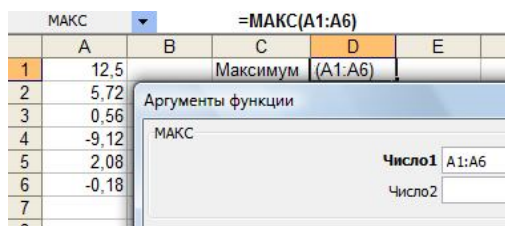
На робочому листі в стовпці А розміщені початкові дані. Необхідно визначити найбільше, середнє арифметичне значення з набору заданих величин і суму всіх значень.

	A	B	C
1	12,5		Максимум
2	5,72		Среднее
3	0,56		Сумма
4	-9,12		
5	2,08		
6	-0,18		
7			

Для отримання найбільшого числа із заданого діапазону необхідно в осередок D1 ввести формулу =МАКС(діапазон). У даному прикладі діапазон осередків A1:A6.

Порядок виконання дій:

- виділити осередок D1 і викликати Майстер функцій;
- у списку Категорія вибрати тип Статистичні, а в списку Функція – МАКС і клацнути на кнопці ОК;
- у вікні функції МАКС вказати її аргумент, тобто виділити за допомогою миші діапазон осередків A1:A6 і клацнути на кнопці ОК.

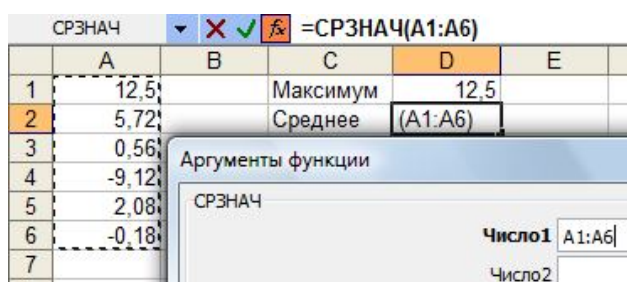


Щоб обчислити середнє арифметичне значення осередків в деякому діапазоні, необхідно ввести функцію =СРЗНАЧ(діапазон). Функція СРЗНАЧ

складає числа, розташовані в межах діапазону, і ділить результат на кількість величин, що складаються.

Порядок виконання дій:

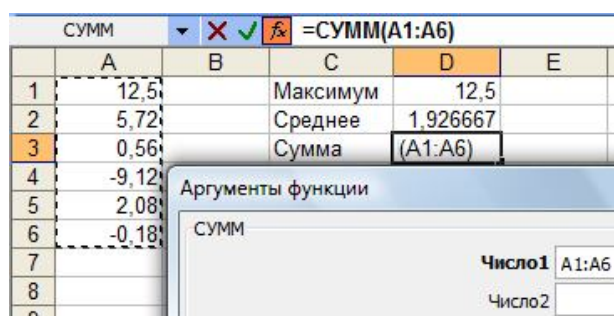
- виділити осередок D2 і викликати Майстер функцій;
- у списку Категорія вибрати тип Статистичні, а в списку Функція – СРЗНАЧ і клацнути на кнопці ОК;
- у вікні функції СРЗНАЧ вказати її аргумент, тобто виділити за допомогою миші діапазон осередків A1:A6 і клацнути на кнопці ОК.



Для отримання суми всіх величин необхідно ввести функцію =СУММ(діапазон).

Порядок виконання дій:

- виділити осередок D3 і викликати Майстер функцій;
- у списку Категорія вибрати тип Математичні, а в списку Функція – СУММ і клацнути на кнопці ОК;
- у вікні функції СУММ вказати її аргумент, тобто виділити за допомогою миші діапазон осередків A1:A6 і клацнути на кнопці ОК.

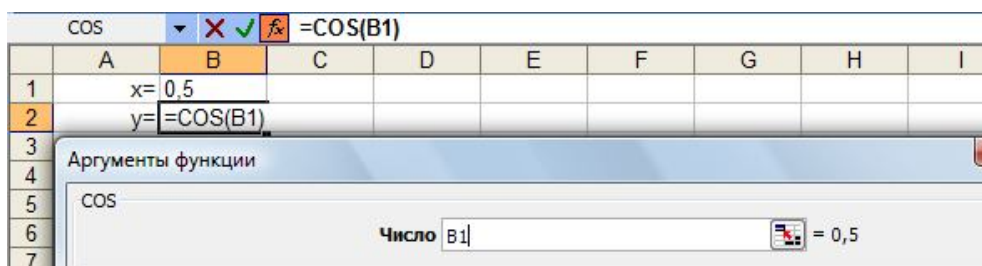


Приклад 2.

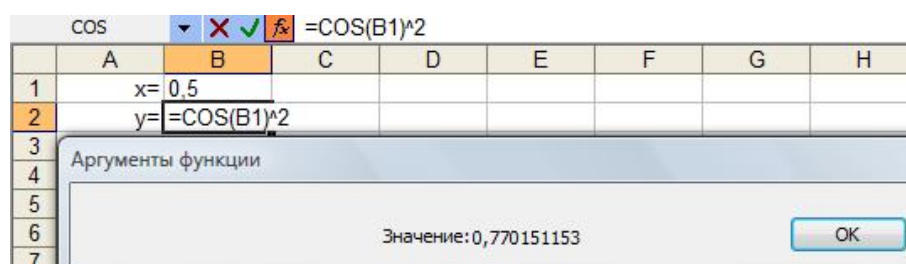
На робочому листі в осередку B1 розміщені початкові дані – значення змінної x. Необхідно визначити значення змінної використовуючи формулу: $y = \cos^2(x)$.

Порядок виконання дій:

- виділити осередок B2 і викликати Майстер функцій;
- у списку Категорія вибрати тип Математичні, а в списку Функція – COS і клацнути на кнопці ОК;
- у вікні функції COS вказати її аргумент, тобто виділити за допомогою миші осередок B1;



- клацнути в рядку формул і ввести знак ^ для піднесення до ступеня та показник ступеню;
- для отримання результату клацнути на кнопці ОК.



Приклад 3.

На робочому листі в стовпці A розміщені початкові дані. Необхідно визначити суму всіх значень, які більше 2.

	A	B	C	D
1	12,5	Сумма чисел, которые больше 2		
2	5,72			
3	0,56			
4	-9,12			
5	2,08			
6	-0,18			

Для отримання результату необхідно використовувати функцію СУММЕСЛИ. Ця функція дозволяє скласти числа, що задовольняють певним критеріям.

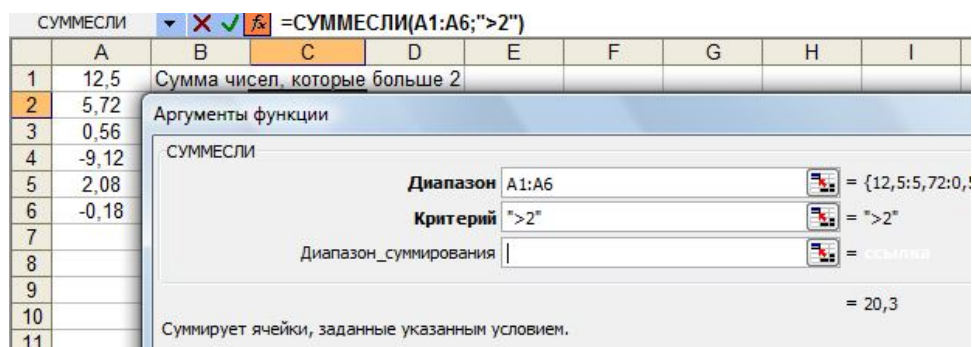
Порядок виконання дій:

- виділити осередок C2 і викликати Майстер функцій;

- у списку Категорія вибрати тип Математичні, а в списку Функція – СУММЕСЛИ і клацнути на кнопці ОК;

- у вікні функції СУММЕСЛИ вказати які осередки піддати аналізу на предмет виконання заданого критерію (поле **Диапазон**), які критерії застосувати до змісту осередків (поле **Условие**), що проглядаються, а також зміст якої групи осередків буде підсумований при виконанні критерію (поле **Диапазон суммирования**).

Якщо в полі **Диапазон суммирования** діапазон осередків не заданий, то програма EXCEL вважає, що потрібно скласти зміст осередків, вказаних в полі **Диапазон**.



У EXCEL є також функція СЧЕТЕСЛИ. Ця функція обчислює кількість осередків, в яких містяться тільки дані, що задовольняють вказаним в ній критеріям.

Приклад 4.

На робочому листі у осередок A1 введено число. Якщо число в осередку A1 більше 1000, то в осередок A2 записати 12% від даного числа, інакше - 20%.

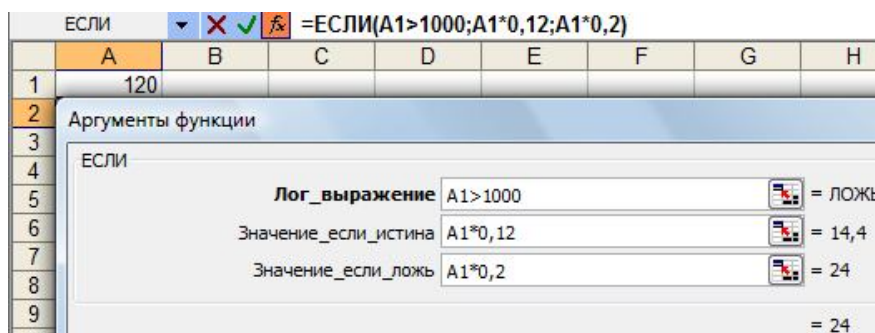
Для отримання результату необхідно використовувати функцію ЕСЛИ. Ця функція перевіряє задану умову і, якщо воно виконується, то проводить одну дію, якщо ні – інше.

Порядок виконання дій:

- виділити осередок A2 і викликати Майстер функцій;
- у списку Категорія вибрати тип Логічні, а в списку Функція – ЕСЛИ і клацнути на кнопці ОК;

- у вікні функції ЕСЛИ в рядку **Логическое выражение** вказати умову.
У рядку **Значение если истина** ввести величину, яка буде повернена, якщо опиниться, що задана умова виконується. У рядку **Значение если ложь** ввести величину, яка буде повернена, якщо опиниться, що задана умова не виконується.

- клацнути на кнопці **ОК**.



Приклад 5.

На робочому листі у осередок B1 введено значення величини x.
Необхідно обчислити значення змінної g використовуючи формулу:

$$g = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}; & x \leq 0 \\ 2x + \frac{\sin^2(x)}{2+x}; & x > 0 \end{cases}$$

Для отримання результату необхідно використовувати функцію ЕСЛИ.

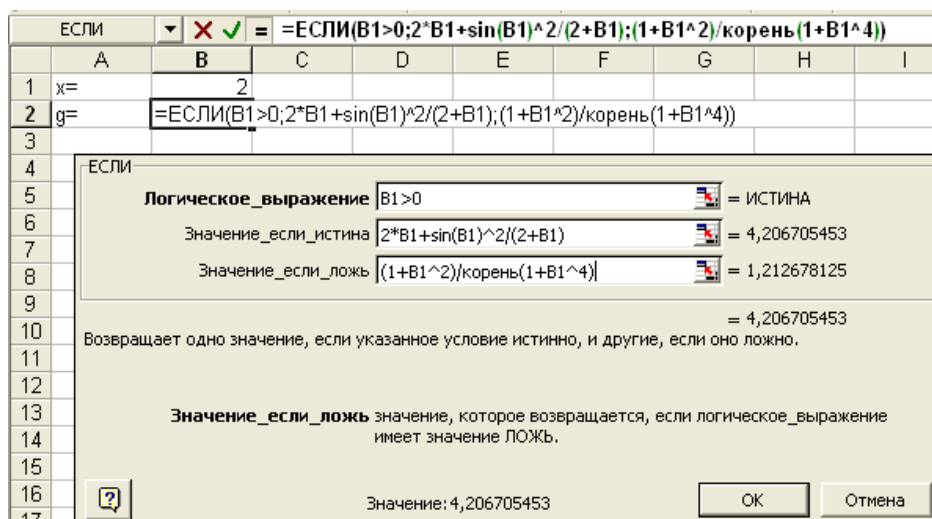
Порядок виконання дій:

- виділити осередок B2 і викликати Майстер функцій;
- у списку Категорія вибрати тип Логічні, а в списку Функція – ЕСЛИ і клацнути на кнопці ОК;

- у вікні функції ЕСЛИ в рядку **Логическое выражение** вказати умову (наприклад, $x > 0$). У рядку **Значение если истина** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова виконується. У рядку **Значение если ложь** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова не виконується.

- клацнути на кнопці **ОК**.

Функція ЕСЛИ буде мати вигляд: $ЕСЛИ\left(x > 0; 2x + \frac{\sin^2(x)}{2+x}; \frac{1+x^2}{\sqrt{1+x^4}}\right)$



Приклад 6.

На робочому листі у осередок B1 введено значення величини x . Необхідно обчислити значення змінної z використовуючи формулу:

$$z = \begin{cases} \frac{1+x}{\sqrt[3]{1+x^2}}; & x \leq 0 \\ -x + 2e^{-2x}; & 0 < x < 1 \\ |1-x|^{1/3}; & x \geq 1 \end{cases}$$

Для отримання результату необхідно використовувати функцію ЕСЛИ. Функція ЕСЛИ буде мати вигляд:

$$\text{ЕСЛИ}\left(x \leq 0; \frac{1+x}{\sqrt[3]{1+x^2}}; \text{ЕСЛИ}\left(x \geq 1; |1-x|^{1/3}; -x + 2e^{-2x}\right)\right)$$

Порядок виконання дій:

- виділити осередок B2 і викликати Майстер функцій;
- у списку Категорія вибрати тип Логічні, а в списку Функція – ЕСЛИ і клацнути на кнопці ОК;
- у вікні функції ЕСЛИ в рядку **Логическое выражение** вказати умову (наприклад, $x \leq 0$). У рядку **Значение если истина** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова виконується. У рядку **Значение если ложь** необхідно викликати знов функцію ЕСЛИ в якості другого аргументу для перевірки другої умови;

ЕСЛИ		X	✓	✓	=	=ЕСЛИ(B1<=0;(1+B1)/(1+B1^2)^(1/3);)			
	A	B	C	D	E	F	G	H	
1	x=	-2							
2	z=	2)^(1/3);)							
3									
4	ЕСЛИ								
5	Логическое выражение		B1<=0				= ИСТИНА		
6	Значение_если_истина		(1+B1)/(1+B1^2)^(1/3)				= -0,584803548		
7	Значение_если_ложь						= любое		
8									
9									
10									= -0,584803548
11	Возвращает одно значение, если указанное условие истинно, и другое, если оно ложно.								

- у вікні другої функції ЕСЛИ, що з'явиться, в рядку **Логическое выражение** вказати умову (наприклад, $x \geq 1$). У рядку **Значение если истина** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова виконується. У рядку **Значение если ложь** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова не виконується;

- клацнути на кнопці **ОК**.

ЕСЛИ		X	✓	✓	=	=ЕСЛИ(B1<=0;(1+B1)/(1+B1^2)^(1/3);ЕСЛИ(B1>=1;abs(1-B1)^(1/3);)			
	A	B	C	D	E	F	G	H	
1	x=	-2							
2	z=	(-2*B1);)							
3									
4	ЕСЛИ								
5	Логическое выражение		B1>=1				= ЛОЖЬ		
6	Значение_если_истина		abs(1-B1)^(1/3)				= 1,44224957		
7	Значение_если_ложь		-B1+2*exp(-2*B1)				= 111,1963001		
8									
9									

Для отримання результату також можна використовувати не тільки функцію ЕСЛИ, а і інші логічні функції, наприклад формула для обчислення змінної z може мати такий вигляд:

$$\text{ЕСЛИ}\left(x \leq 0; \frac{1+x}{\sqrt[3]{1+x^2}}; \text{ЕСЛИ}\left(\text{И}(x > 0; x < 1); \left(-x + 2e^{-2x}; |2-x|^{1/3}\right)\right)\right)$$

Приклад 7.

На робочому листі у осередок B1 введено значення величини x. Необхідно обчислити значення змінної у використовуючи формулу:

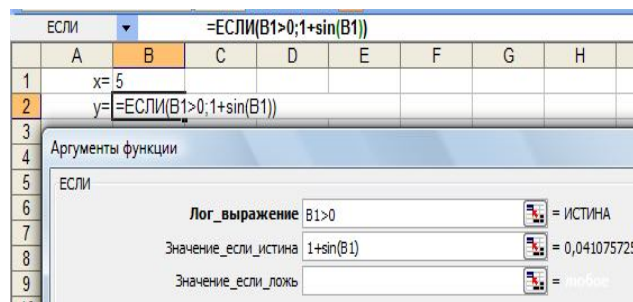
$$y = \begin{cases} 1 + \sin x; & x > 0 \\ x + \frac{1}{2+x}; & x \leq 0 \end{cases}$$

Для отримання результату необхідно використовувати функцію ЕСЛИ. Функція ЕСЛИ буде мати вигляд:

$$\text{ЕСЛИ}\left(x > 0; 1 + \sin x; \text{ЕСЛИ}\left(2 + x \neq 0; x + \frac{1}{2 + x}; \text{"деление на ноль"}\right)\right)$$

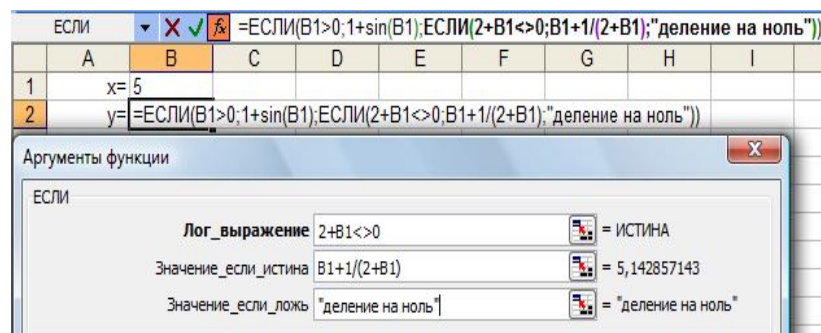
Порядок виконання дій:

- виділити осередок B2 і викликати Майстер функцій;
- у списку Категорія вибрати тип Логічні, а в списку Функція – ЕСЛИ і клацнути на кнопці ОК;
- у вікні функції ЕСЛИ в рядку **Логическое выражение** вказати умову (наприклад, $x > 0$). У рядку **Значение если истина** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова виконується. У рядку **Значение если ложь** необхідно викликати знов функцію ЕСЛИ в якості другого аргументу для перевірки другої умови;



- у вікні другої функції ЕСЛИ, що з'явиться, в рядку **Логическое выражение** вказати умову (наприклад, $2 + x \neq 0$). У рядку **Значение если истина** ввести формулу, результат якої, буде повернено, якщо опиниться, що задана умова виконується. У рядку **Значение если ложь** ввести текстове повідомлення «ділення на нуль», яке буде виводитись на екран, якщо опиниться, що задана умова не виконується;

- клацнути на кнопці **ОК**.



Отримання результату значення у при різних значеннях змінної x:

	A	B	C
1	x= 5		
2	y=	0,041076	

	A	B	C
1	x= -5		
2	y=	-5,33333	

	A	B	C
1	x= -2		
2	y=	деление на ноль	

6.7.4 Копіювання формул

Багато таблиць містять дуже схожі формули. Хоча можна вводити всі формули роздільно, набагато зручніше користуватися маркером заповнення програми EXCEL, що дозволяє створити набір ідентичних формул, які розрізняються лише посиланнями.

Для копіювання формули необхідно виконати наступні дії:

- виділити осередок, в якому записана формула;
- встановити покажчик миші на маркері заповнення (маленький квадратик в нижньому правому кутку активного осередку). Покажчик миші прийме вид маленького знаку «плюс»;
- натиснути на ліву кнопку миші і, не відпускаючи її, протягнути маркер заповнення на необхідну кількість осередків, а потім відпустити кнопку миші.

При копіюванні формули відбудеться автоматичне настроювання посилань на осередки з урахуванням нових положень формул. Такі посилання на осередки називаються **відносними**. Але буває, що посилання на осередки не можна змінювати. Посилання, що залишаються незмінними при копіюванні, називаються **абсолютними**.

Розглянемо приклад.

У осередку C2 введена формула, яка розраховує правильне значення для Іванова, множаючи кількість відпрацьованих часів на тарифну ставку в осередку C8.

	A	B	C
1	Фамилия	Количество отработанных часов	Начислено, грн.
2	Иванов	120	3900
3	Петров	109	
4	Сидоров	96	
5	Макаров	123	
6	Семенов	80	
7			
8	Часовая тарифная ставка		32,5

Щоб скопіювати цю формулу в решту осередків стовпця C, необхідно перетягнути маркер заповнення. Якщо відпустити кнопку миші, то буде наступна картина. У чому помилка?

Якщо виділити осередок C3, то можна побачити формулу: =B3*C9. Ссылка на осередок B3 указує на рядок, в якому записана кількість відпрацьованих часів Петровим. Тут все гаразд. Але

	A	B	C
1	Фамилия	Количество отработанных часов	Начислено, грн.
2	Иванов	120	3900
3	Петров	109	0
4	Сидоров	96	0
5	Макаров	123	0
6	Семенов	80	0
7			
8	Часовая тарифная ставка		32,5

посилання на тарифну ставку змінилося і тепер указує не на осередок С8, а на осередок С9, в якому нічого не немає! Щоб все було правильно, посилання на осередок С8 слід було зберегти – тобто зробити її абсолютною.

Для цього необхідно виділити осередок з формулою С2 і натиснути клавішу **F2**. Використовуючи клавіші переміщення (клавіші із стрілками) або мишу, встановити курсор введення зліва від осередку С8 і натиснути клавішу **F4**. Програма EXCEL вставить знак долара (\$) і осередок С8 прийме вигляд: **\$C\$8**. Натиснути клавішу **Enter**.

	А	В	С
1	Фамилия	Количество отработанных часов	Начислено, грн.
2	Иванов	120	=B2*\$C\$8
3	Петров	109	0
4	Сидоров	96	0
5	Макаров	123	0
6	Семенов	80	0
7			
8	Часовая тарифная ставка		32,5

Для отримання копії формули в решті всіх осередків, протягнути маркер заповнення вниз по стовпцю С.

	А	В	С
1	Фамилия	Количество отработанных часов	Начислено, грн.
2	Иванов	120	3900
3	Петров	109	3542,5
4	Сидоров	96	3120
5	Макаров	123	3997,5
6	Семенов	80	2600
7			
8	Часовая тарифная ставка		32,5

Іноді у формулах використовується і **змішане посилання на осередок**, наприклад **\$C6** (стовпець залишається тим самим, а рядок міняється) або **D\$2** (стовпець міняється, а рядок залишається тим самим).

6.8 Автоматичне введення даних

У програмі EXCEL є декілька витончених трюків, прискорюючих введення даних. Наприклад, якщо створюються мітки для місяців в році, не потрібно вводити кожного місяця окремо. Досить ввести лише перший

місяць і скористатися засобом заповнення EXCEL, яке автоматично виконає введення назв решти місяців.

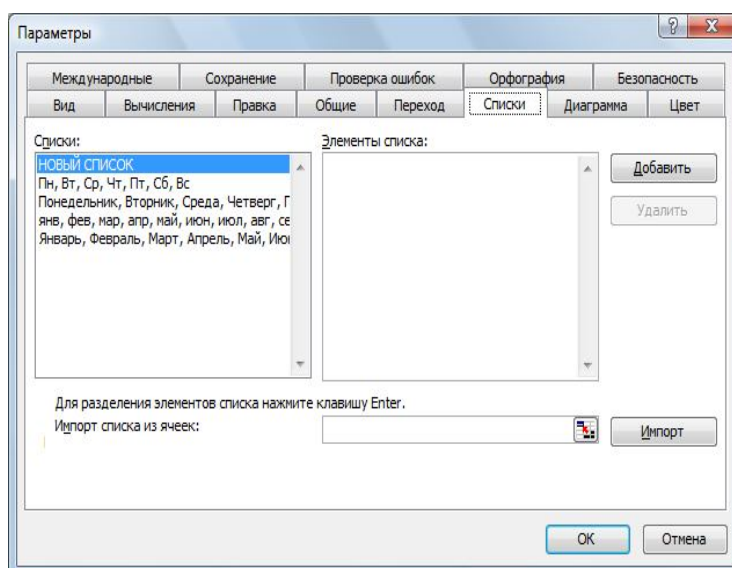
Для заповнення інтервалу осередків текстовою послідовністю необхідно:

- у першій осередок ввести першу мітку даних (наприклад, день тижня або місяць);
- встановити покажчик миші на маркер заповнення;
- натиснути на ліву кнопку миші і, не відпускаючи її, протягнути маркер заповнення на необхідний інтервал осередків, а потім відпустити кнопку миші.

	A1	Пн	
	A	B	C
1	Пн		
2			
3			Ср

Програма EXCEL автоматично заповнить вибраний інтервал осередків відповідними текстовими даними.

Якщо немає потрібної послідовності в списку програми EXCEL, її можна додати, використовуючи команду **Сервіс/Параметри/Списки**.



Для додавання нового списку даних необхідно:

- на вкладці **Списки** натиснути кнопку **Добавить**;
- у полі **Элементы списка** ввести елементи нового списку в тому порядку, в якому вони повинні будуть з'являтися на робочому листі. Для розділення елементів списку необхідно натискати клавішу **Enter**;

- натиснути кнопку **ОК**.

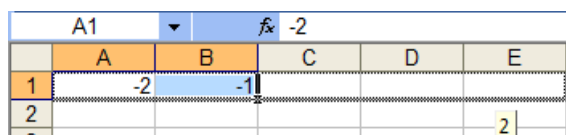
Так само формується послідовність чисел.

Для заповнення інтервалу осередків послідовністю чисел кроком одиниця або іншим необхідно виконати наступні дії:

- ввести перші два значення в сусідніх осередках. Програма EXCEL використовуватиме ці два значення для визначення кроку, з яким зростатимуть або убуватимуть числа;

- виділити ці осередки і встановити покажчик миші на маркері заповнення;

- натиснути ліву кнопку миші і, не відпускаючи її, протягнути маркер заповнення на необхідний інтервал осередків, а потім відпустити кнопку миші.



	A1	A	B	C	D	E
1	-2	-2	-1			
2						2

Програма EXCEL заповнить відмічену область відповідними значеннями.

6.9 Створення й редагування діаграм


Демонструючи числа у вигляді картинок, добре розроблені діаграми допомагають представити інформацію більш дохідливо і наочно, ніж більші об'єми цифр. Підкреслюючи тенденції і порівняльні чинники, діаграми можуть розповісти те ж саме, що і колонки цифр, але більш чітко і виразно.

Більш того, діаграми автоматично оновлюються при зміні даних, на яких вона заснована.

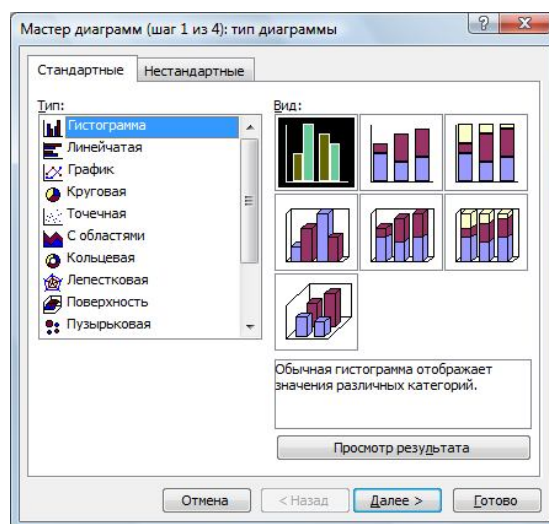
Для створення діаграми використовується **Майстер діаграм EXCEL**. Ця програма виконує крок за кроком процес створення діаграми.

Для створення діаграми за допомогою Майстра діаграм необхідно виконати наступні дії:

- виділити початкові дані, на основі яких буде побудована діаграма. Обов'язково включити заголовки рядків і стовпців;

- клацнути по кнопці в панелі інструментів **Мастер диаграмм**  або вибрати команду **Вставка/Диаграмма**.

На першому кроці майстер діаграм пропонує вибрати тип і вид діаграми, і натиснути кнопку **Далее**.



На другому кроці перевіряється, чи правильно введений в поле **Диапазон** інтервал осередків, на основі яких буде побудована діаграма.

На третьому кроці пропонується ввести параметри діаграми: назву діаграми і назви осей координат, включити або відмінити лінії сітки і вказати, чи слід включати в діаграму легенду і де вона повинна знаходитися.

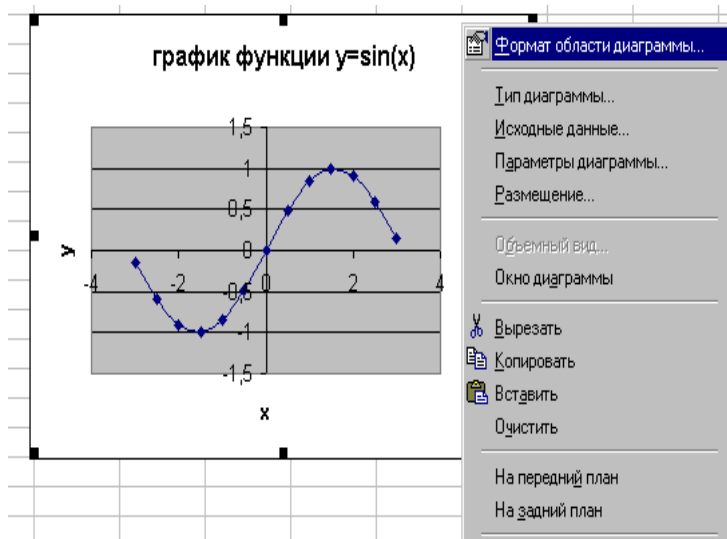
На четвертому кроці пропонується вказати, де повинна бути розташована діаграма: на вже існуючий робочий лист або на окремий лист, і натиснути кнопку **Готово**.

Після вставки діаграми в робочий лист можна змінити її розміри або перемістити в інше місце, заздалегідь виділивши її. Щоб виділити діаграму, необхідно помістити на ній покажчик миші і клацнути лівою кнопкою миші. Навколо діаграми з'явиться тонка прямокутна рамка з маркерами у вигляді чорних квадратів по кутах і на середині кожній із сторін.

Для зміни розмірів діаграми необхідно встановити покажчик миші на один з маркерів (він прийме вид двонаправленої стрілки) і при натиснутої лівої кнопки миші, перемістити пунктирну лінію - межу діаграми.

Для переміщення діаграми – встановити на діаграмі покажчик миші, натиснути ліву кнопку миші і, не відпускаючи її, перемістити діаграму на нове місце.

У будь-який момент можна змінити тип або формат діаграми. Для цього виділити діаграму або окремий елемент діаграми (крива, легенда, заголовки, координатні осі і т.п.) і натиснути праву кнопку миші. У контекстному меню, що з'явиться, вибрати відповідну команду.



Приклад 1.

Побудувати графік функції $y=\sin(x)$.

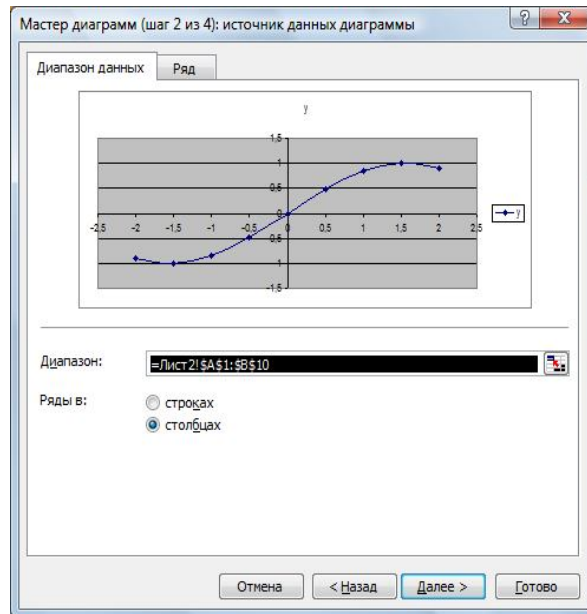
Порядок дій:

1. На робочому листі створити числову послідовність значень x .
2. Для кожного значення x обчислити за заданою формулою значення y .

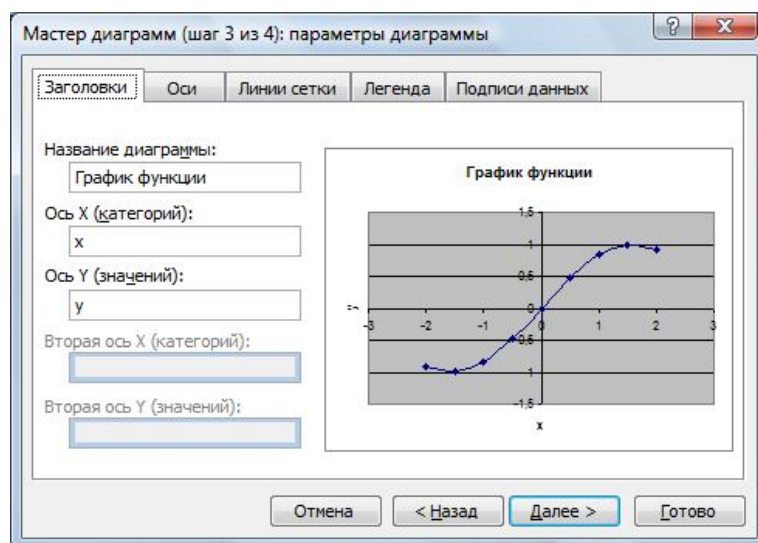
	B2		fx =SIN(A2)
	A	B	C
1	x	y	
2	-2	-0,9093	
3	-1,5	-0,99749	
4	-1	-0,84147	
5	-0,5	-0,47943	
6	0	0	
7	0,5	0,479426	
8	1	0,841471	
9	1,5	0,997495	
10	2	0,909297	

3. Виділити таблицю значень і вибрати команду **Вставка/Діаграма**. З'явиться діалогове вікно **Мастера діаграмм**.

4. На 1 кроці вибрати тип діаграми – «точечная», вигляд – «точечная диаграмма со значениями, соединенными сглаживающими линиями», кнопка **Далее**. На 2 кроці перевірити в полі **Диапазон** інтервал осередків, на основі якого буде побудована діаграма - A1:B10, кнопка **Далее**.



На 3 кроці задати параметри діаграми: на вкладці **Заголовки** вказати назву діаграми і назви координатних осей, на вкладці **Легенда** зняти прапорець **Добавить легенду**, кнопка **Далее**.



На 4 кроці вказати розміщення діаграми – в наявному листі, кнопка **Готово**.

На робочому листі з'явиться графік функції $y=\sin(x)$.

Приклад 2.

Побудувати поверхню $Z = X^2 - Y^2$ для $x \in [-1, 1]$ і $y \in [-1, 1]$ з кроком $\Delta=0.2$ для обох змінних.

Порядок дій:

1. На робочому листі в діапазоні осередків A2:A12 створити числову послідовність значень X: -1, -0.8, ..., 1.

2. У діапазоні осередків B1:L1 створити числову послідовність значень Y: -1, -0.8, ..., 1.

3. У осередок B2 ввести формулу: $= \$A2^2 - B\1^2 .

Знак \$ що стоїть перед буквою в імені осередку, дає абсолютне посилання на стовпець з даним ім'ям, а знак \$, що стоїть перед цифрою – абсолютне посилання на рядок з цим ім'ям. Знаки \$ дозволяють правильно обчислити значення Z при відповідних значеннях X і Y.

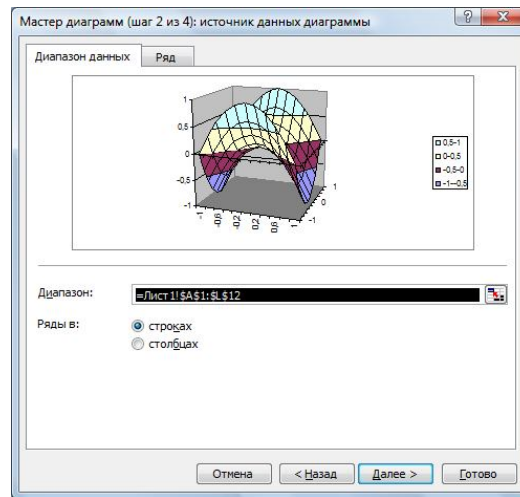
4. Скопіювати формулу на діапазон осередків B2:L12.

	A	B	C	D	E	F	G	H	I	J	K	L
1		-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1
2	-1	0	0.36	0.64	0.84	0.96	1	0.96	0.84	0.64	0.36	0
3	-0.8	-0.36	0	0.28	0.48	0.6	0.64	0.6	0.48	0.28	0	-0.36
4	-0.6	-0.64	-0.28	0	0.2	0.32	0.36	0.32	0.2	0	-0.28	-0.64
5	-0.4	-0.84	-0.48	-0.2	0	0.12	0.16	0.12	0	-0.2	-0.48	-0.84
6	-0.2	-0.96	-0.6	-0.32	-0.12	0	0.04	0	-0.12	-0.32	-0.6	-0.96
7	0	-1	-0.64	-0.36	-0.16	-0.04	0	-0.04	-0.16	-0.36	-0.64	-1
8	0.2	-0.96	-0.6	-0.32	-0.12	0	0.04	0	-0.12	-0.32	-0.6	-0.96
9	0.4	-0.84	-0.48	-0.2	0	0.12	0.16	0.12	0	-0.2	-0.48	-0.84
10	0.6	-0.64	-0.28	0	0.2	0.32	0.36	0.32	0.2	0	-0.28	-0.64
11	0.8	-0.36	0	0.28	0.48	0.6	0.64	0.6	0.48	0.28	0	-0.36
12	1	0	0.36	0.64	0.84	0.96	1	0.96	0.84	0.64	0.36	0

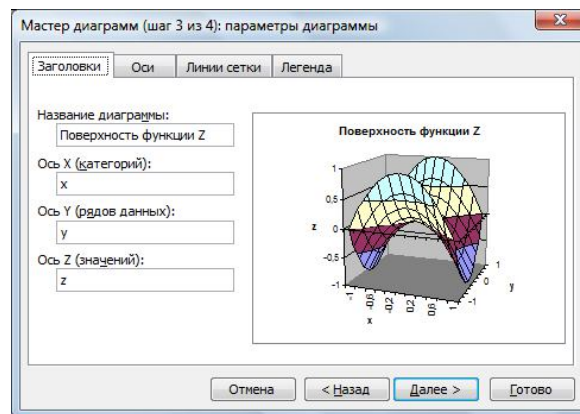
5. Виділити таблицю значень функції Z (діапазон осередків A1:L1) і вибрати команду **Вставка/Діаграма**. З'явиться діалогове вікно **Мастер діаграмм**.

6. На 1 кроці вибрати тип діаграми – «**Поверхність**», вигляд – «**Поверхність. Отображает изменение значений по двум измерениям, в виде поверхности**», кнопка **Далее**.

На 2 кроці перевірити в полі **Діапазон** інтервал осередків, на основі якого буде побудована поверхня, це – інтервал осередків A1:L1, кнопка **Далее**.



На 3 кроці задати параметри поверхні: на вкладці **Заголовки** вказати назву діаграми і назви координатних осей, на вкладці **Легенда** зняти прапорець **Добавить легенду**, кнопка **Далее**.



На 4 кроці вказати розміщення діаграми – в наявному листі, кнопка **Готово**.

Приклад 3.

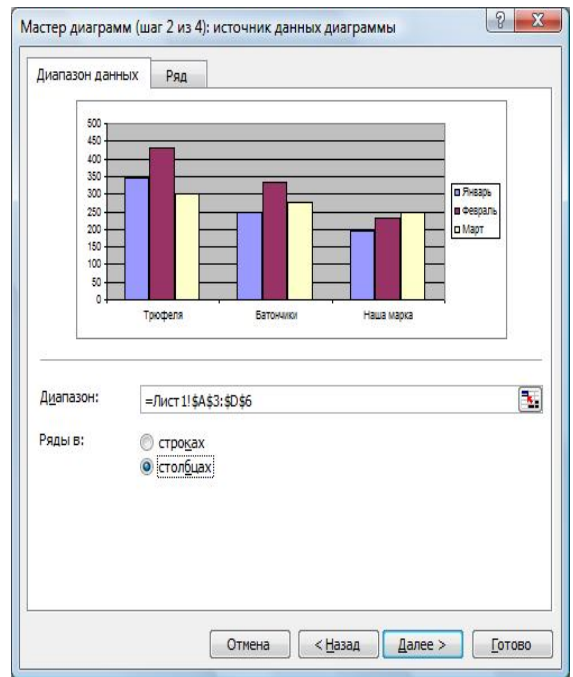
Побудувати гістограму на основі таблиці початкових даних і кругову діаграму на основі підсумкових результатів.

	A	B	C	D
1	Реализация конфет			
2		Январь	Февраль	Март
3	Трюфеля	341	426	305
4	Батончики	250	340	270
5	Наша марка	200	230	250
6	Итого за месяц	791	996	825
7				

Порядок дій для побудови гістограми:

1. Виділити таблицю початкових даних: діапазон осередків A3:D6 і вибрати команду **Вставка/Диаграмма**. З'явиться діалогове вікно **Мастер диаграмм**.

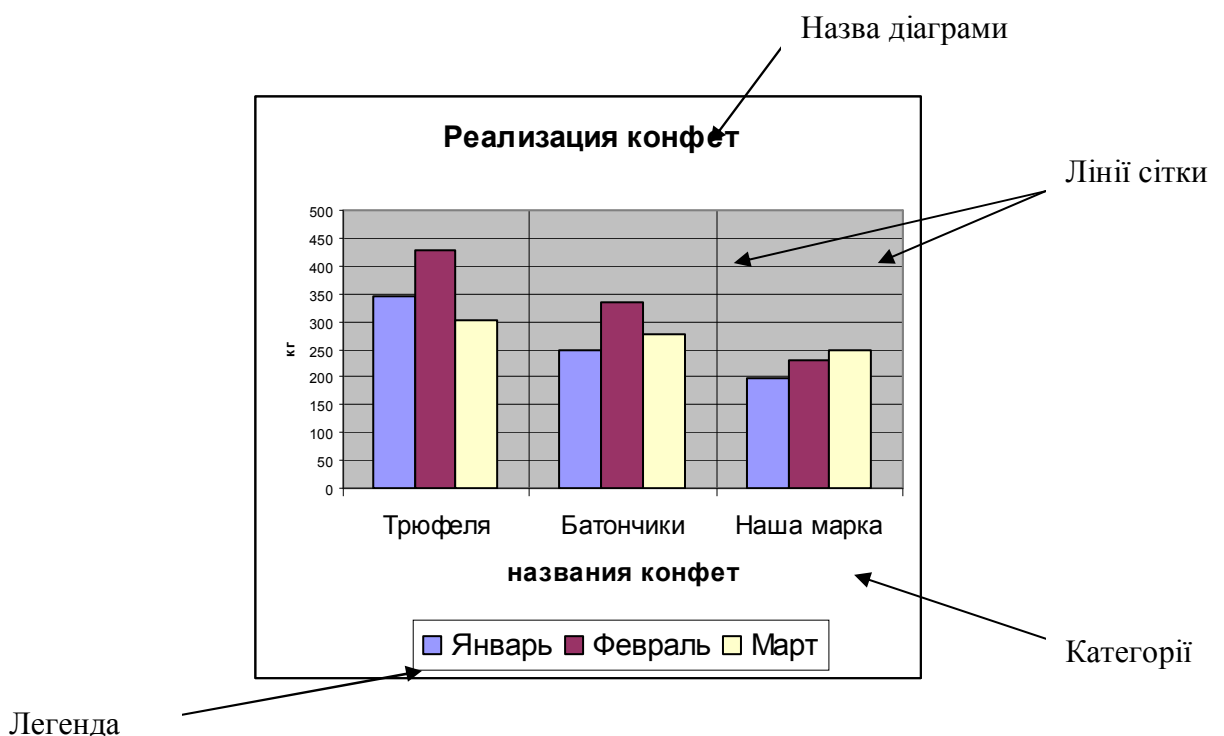
2. На 1 кроці вибрати тип діаграми – «Гистограмма», вигляд – «Обычная гистограмма отображает значения различных категорий», кнопка **Далее**. На 2 кроці перевірити в полі **Диапазон** інтервал осередків, на основі якого буде побудована діаграма, це інтервал осередків - A3:D6. Побудувати діаграму можна по рядках або по стовпцях таблиці, вибравши потрібну опцію **Ряды в строках** або



Ряды в столбцах. Якщо зображена у вікні діаграма виглядає як потрібно, клацнути на кнопці **Далее**.

На 3 кроці задати параметри гистограми: на вкладці **Заголовки** вказати назву діаграми і назви координатних осей, на вкладці **Легенда** вказати розміщення легенди на діаграмі, на вкладці **Линии сетки** включити основні лінії сітки; кнопка **Далее**.

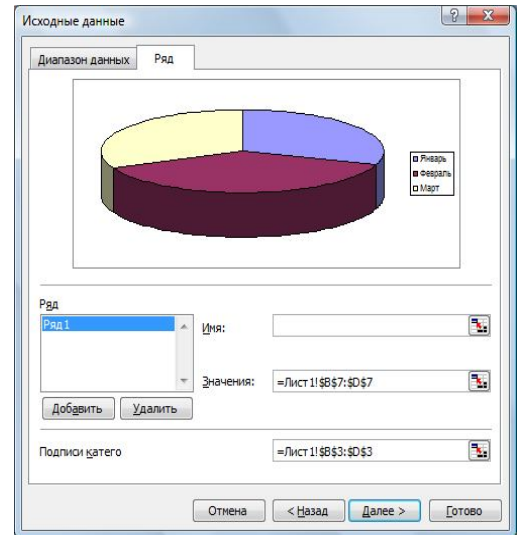
На 4 кроці вказати розміщення діаграми – в наявному листі, кнопка **Готово**.



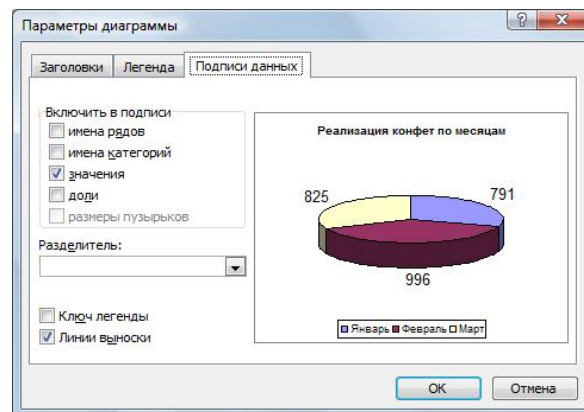
Порядок дій для побудови кругової діаграми:

1. Виділити в таблиці рядок підсумкових результатів: діапазон осередків B7:D7 і вибрати команду **Вставка/Діаграма**. З'явиться діалогове вікно **Мастер діаграм**.

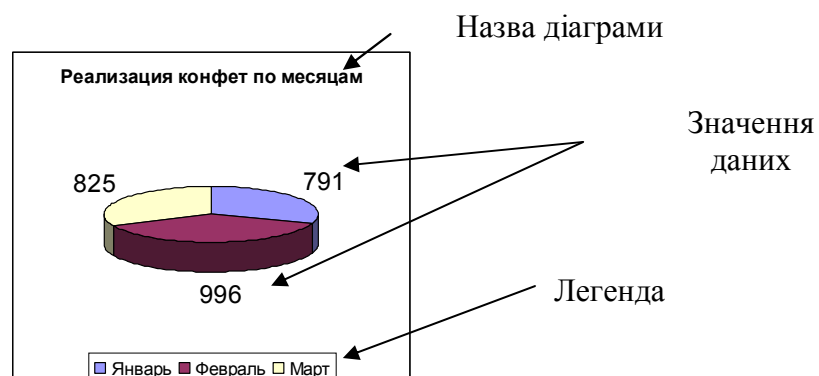
2. На 1 кроці вибрати тип діаграми – «Круговая», вигляд – «Объемный вариант круговой диаграммы», кнопка **Далее**. На 2 кроці перевірити в полі **Диапазон** інтервал осередків, на основі якого буде побудована діаграма, - B7:D7. На вкладці **Ряд** в полі **Подписи категорий** вказати інтервал осередків B3:D3, для відображення змісту цих осередків в легенді. Якщо зображена у вікні діаграма виглядає як потрібно, клацнути на кнопці **Далее**.



На 3 кроці задати параметри діаграми: на вкладці **Заголовки** вказати назву діаграми, на вкладці **Легенда** вказати розміщення легенди на діаграмі, на вкладці **Подписи данных** включити відображення значень; кнопка **Далее**.



На 4 кроці вказати розміщення діаграми – в наявному листі; кнопка **Готово**.



СПИСОК ЛІТЕРАТУРИ

1. К. Стинсон. Эффективная работа в Windows 95 / Перев. сангл. - СПб: Питер, 1998. - 784 с.
2. Информатика. Базовый курс. 2-е издание / Под ред. С.В.Симоновича. – СПб.: Питер, 2004. – 640 с.
3. Клименко А. Эффективный самоучитель работы на ПК. Основной курс.- Киев: ДиаСофт, 2002.- 496 с.
4. Кишик А.Н. Word 2002. Эффективный самоучитель. Быстро... просто... наглядно...- СПб.: ООО ДиаСофт, 2001.- 256 с.
5. Кишик А.Н. Excel 2002. Эффективный самоучитель. Быстро... просто... наглядно...- СПб.: ООО ДиаСофтЮП, 2001.- 240 с.
6. Р.Борланд. Эффективная работа с Microsoft Word 7.0. – СПб: Питер, 1997. – 1104с.
7. Гончаров А. Excel 97 в примерах. – СПб: Питер, 1997. – 336с.
8. В.Я.Гельман. Решение математических задач средствами Excel/ Практикум. – СПб.: Питер, 2003. – 237с.
9. Вычислительная техника и программирование. Учеб. для техн. вузов / А.В.Петров, В.Е.Алексеев, А.С.Ваулин и др. – М.: Высш.шк., 1990. – 479с.
10. Методические указания и задания к выполнению лабораторных работ по теме “Алгоритмизация” / Сост.: Л.А. Лазебная, И.В. Дынник, С.В. Масло – Донецк, ДонНТУ, 2005. – 67с.
11. Методичні вказівки і завдання до курсової роботи з дисципліни «Обчислювальна математика і програмування» (для студентів спеціальності «Хімічна технологія і інженерія» / Уклад.: Л.О. Лазебна, С.В. Масло – Донецьк: ДонНТУ, 2010. – 36 с.
12. Методичні вказівки до виконання лабораторних робіт у текстовому редакторі WORD / Укл.: Л.О. Лазебна, С.В. Масло - Донецьк: ДонНТУ, 2008. - 35 с.

Курс лекцій з дисципліни
«Обчислювальна математика та
програмування»

(для студентів напрямку підготовки «Хімічна технологія»)

Укладачі: Лазебна Людмила Олександрівна