Міністерство освіти і науки України
Інженерний навчально-науковий інститут
імені Ю.М. Потебні
Запорізького національного університету

О. М. Міхайлуца, А. В. Пожуєв

ІНФОРМАТИКА ТА КОМП'ЮТЕРНА ТЕХНІКА

Навчальний посібник
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Цивільна безпека»
освітньо-професійної програми «Охорона праці»

Затверджено
вченою радою ЗНУ
Протокол № 11 від 27.05.2025

Запоріжжя
2025

УДК 004(075.8)
М691

Міхайлуца О. М., Пожуєв А. В. Інформатика та комп’ютерна техніка : навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності «Цивільна безпека» освітньо-професійної програми «Охорона праці». Запоріжжя : Запорізький національний університет, 2025. 120 с.

Навчальний посібник містить основні теоретичні положення тем курсу «Інформатика та комп’ютерна техніка», питання для самоконтролю, навчальні та практичні завдання, тестові завдання для перевірки знань, перелік рекомендованої літератури.

Видання сприятиме набуттю системних знань і навичок із базових принципів роботи комп'ютерної техніки, ключових концепцій інформатики та практичного застосування сучасних програмних інструментів в інженерній діяльності. Отримані знання дозволять студентам використовувати сучасні програмні інструменти для вирішення конкретних завдань у сфері охорони праці, таких як ведення документації, аналіз даних про нещасні випадки, оцінка ризиків, моделювання небезпечних ситуацій.

Посібник орієнтований на студентів, які навчаються на рівні освіти бакалавр спеціальності «Цивільна безпека» за освітньо-професійною програмою «Охорона праці» Інженерного навчально-наукового інституту ім. Ю. М. Потебні Запорізького національного університету денної та заочної форм здобуття освіти. Може бути корисним студентам інших спеціальностей у розрізі наведених у посібнику розділів.

Рецензент

О.Ю. Небеснюк, к.т.н., доцент кафедри електроніки, інформаційних систем та програмного забезпечення

Відповідальний за випуск

Т.В. Критська, д-р техн. наук, професор, завідувач кафедри електроніки, інформаційних систем та програмного забезпечення Запорізького національного університету

ЗМІСТ

ВСТУП 6
РОЗДІЛ 1. АРИФМЕТИЧНІ ОСНОВИ КОМП'ЮТЕРІВ. СИСТЕМИ ЧИСЛЕННЯ 7
1.1 Історія розвитку обчислювальної техніки 7
1.2 Загальні відомості про системи числення 9
1.3 Переведення чисел з однієї позиційної системи числення в іншу 13
1.3.1 Переведення цілих чисел 13
1.3.2 Переведення правильних дробів 14
1.3.3 Переведення змішаних чисел 16
1.3.4 Переведення чисел між системами з основою степеня 2 16
1.3.5 Арифметичні операції над двійковими числами 17
Питання для самоконтролю 18
Індивідуальне завдання 18
Тести для самоконтролю 19
РОЗДІЛ 2. РОБОТА З ОФІСНИМ ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ MICROSOFT OFFICE 21
2.1 Основи роботи з текстовим редактором Microsoft Word 21
2.1.1 Основні функції роботи з додатком 21
2.1.2 Основи роботи з текстом у документі 24
2.1.3 Робота з таблицями, графікою та спільний доступ 29
Питання для самоконтролю 33
Індивідуальне завдання 34
2.2 Створення мультимедійних презентацій за допомогою Microsoft Power Point 40
2.2.1 Основні функції роботи з додатком 40
2.2.2 Оформлення презентацій 42
2.2.3 Робота з текстом доповіді презентації 44
2.2.4 Робота з графічними об'єктами 45
2.2.5 Організація текстової і графічної інформації слайда 45
Питання для самоконтролю 47
Індивідуальне завдання 48
Тести для самоконтролю 49
РОЗДІЛ 3. ОСНОВИ РОБОТИ З ЕЛЕКТРОНИМИ ТАБЛИЦЯМИ MICROSOFT EXCEL 52
3.1 Організація введення даних 52
3.2 Формати даних Microsoft Excel 54
3.3 Форматування вмісту комірок Microsoft Excel 55
3.4 Операції друку 56
3.5 Робота з великими робочими аркушами 57
3.6 Робота з графікою 58
3.7 Побудова формул 59
Питання для самоконтролю 61
Індивідуальні завдання 61

Тести для самоконтролю 71
РОЗДІЛ 4. ОСНОВИ АЛГОРИТМІЗАЦІЇ 73
4.1 Поняття алгоритму 73
4.2 Властивості алгоритму 73
4.3 Способи опису (представлення) алгоритму 74
4.4 Об'єкти дії в алгоритмах і програмах 78
4.5 Різновиди структур алгоритмів 80
Питання для самоконтролю 87
Індивідуальні завдання 88
Тести для самоконтролю 91
РОЗДІЛ 5. ОСНОВИ МОВИ ПРОГРАМУВАННЯ PYTHON 94
5.1 Введення в Python та комп'ютерне програмування 94
5.2 Основні конструкції мови 98
5.3 Базовий синтаксис функцій 104
5.4 Основи графічних інтерфейсів (GUI) і створення віконних додатків 106
5.4.1 Основні елементи графічного інтерфейсу користувача 106
5.4.2 Створення графічних інтерфейсів Python за допомогою Tkinter 108
Питання для самоконтролю 111
Індивідуальні завдання 111
Тести для самоконтролю 117
РЕКОМЕНДОВАНА ЛІТЕРАТУРА 119
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 119

ВСТУП

Сучасний етап розвитку суспільства характеризується стрімким зростанням можливостей комп'ютерної техніки та програмного забезпечення. Комп'ютерні технології стали невід'ємною складовою практично всіх галузей людської діяльності. У контексті офісного середовища їх повсюдне та інтенсивне використання є безальтернативною нормою. Комп'ютер де-факто є невід'ємним елементом робочого простору фахівців широкого спектра професій. Ключовими компетенціями сучасного інженера є впевнена робота з комп'ютером, глибоке розуміння сучасних технологій обробки електронних даних, а також вміння ефективно створювати технічну документацію, аналітичні звіти та наочні діаграми.

Метою викладання дисципліни «Інформатика та комп'ютерна техніка» є формування у студентів компетентностей, які сприяють оволодінню навичками роботи на сучасній обчислювальній техніці, засвоєнню основних пакетів прикладних програм, формуванню умінь і навичок ефективного використання сучасних комп'ютерно-інформаційних технологій у своїй діяльності, що має забезпечити формування у студентів основ інформаційної культури та інформатично-комунікативної компетентності, розвиток алгоритмічного мислення, оволодіння розробкою програм на алгоритмічній мові високого рівня.

Завданнями навчальної дисципліни є надання теоретичних знань з принципів роботи ЕОМ, основ алгоритмізації, основ програмування на мовах високого рівня та технології програмування, з операційних систем; набуття практичних навичок з текстової обробки інформації, з систем програмування, графічних засобів ЕОМ, основних понять сучасної технологій обробки інформації.

У результаті вивчення навчальної дисципліни студент повинен набути таких результатів навчання (знання, уміння тощо) та компетентностей:

- Здатність до абстрактного мислення, аналізу та синтезу.
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Програмні результати навчання:

- Використовувати у професійній діяльності сучасні інформаційні технології, системи управління базами даних та стандартні пакети прикладних програм.
- Розробляти та використовувати технічну документацію, зокрема з використанням сучасних інформаційних технологій.

Відповідно до структурно-логічної схеми освітньо-професійної програми, засвоєння навчального матеріалу курсу «Інформатика та комп'ютерна техніка» логічне пов'язане з використанням знань, вмінь та навичок, отриманих при вивченні комп'ютерних дисциплін на базі школи. Курс має міждисциплінарні зв'язки з ОК «Ведення документації та навчання з охорони праці», набуті при вивченні даного курсу знання необхідні у виробничій практиці та виконанні курсового проєкту з ОК «Пожежна безпека».

РОЗДІЛ 1. АРИФМЕТИЧНІ ОСНОВИ КОМП'ЮТЕРІВ. СИСТЕМИ ЧИСЛЕННЯ

Мета: розглянути фундаментальні основи систем числення, які використовуються в комп'ютерних системах; ознайомитися з принципами представлення чисел у комп'ютері; вивчити методи перетворення чисел між різними позиційними системами числення, такими як десяткова, двійкова, вісімкова та шістнадцяткова; опанувати виконання основних арифметичних операцій з двійковими числами.

План

1. Історія розвитку обчислювальної техніки.
2. Вступ до систем числення.
3. Позиційні системи числення.
4. Перетворення чисел між системами числення.
5. Арифметичні операції над двійковими числами.

Ключові терміни: *система числення, основа системи числення, двійкова система числення, десяткова система числення, вісімкова система числення, шістнадцяткова система числення, перетворення чисел, арифметичні операції, представлення чисел.*

1.1 Історія розвитку обчислювальної техніки

Історія обчислювальної техніки налічує понад шість тисячоліть. Перший механічний калькулятор, здатний виконувати додавання та віднімання, з'явився у 1642 році завдяки французькому математику Блезу Паскалю. Пізніше, у 1673 році, німецький вчений Готфрід Вільгельм Лейбніц удосконалив цю ідею, створивши машину, яка могла виконувати вже чотири арифметичні дії. У 1882 році російський математик Пафнутій Львович Чебишов також розробив електромеханічну обчислювальну машину, що виконувала чотири арифметичні операції.

Важливий крок у розвитку обчислювальної техніки зробив англійський науковець Чарльз Беббідж, який вперше запропонував використовувати програми для автоматизації обчислень. У 1833 році він розпочав роботу над створенням механічної цифрової обчислювальної машини, відомої як аналітична машина Беббіджа. Розроблений Беббіджем принцип програмного управління, що передбачав розділення інформації на дані для обробки та керуючі інструкції, став ключовою теоретичною основою для подальшої автоматизації обчислень.

У 1890 році в Америці Герман Голлеріт розробив перфораційні машини, які знайшли широке застосування на практиці. Варто зазначити, що навіть у 80-х роках XX століття задачі обліку матеріалів та нарахування заробітної плати вирішувалися за допомогою цих машин.

Першим по-справжньому електронним комп'ютером вважається ENIAC, створений у 1945 році. У 1946 році Джон фон Нейман запропонував концепцію

внутрішнього зберігання програми, яка вперше була реалізована в англійській ЕОМ EDSAC у 1949 році під керівництвом Моріса Вілкса.

Перша електронно-обчислювальна машина в Європі, «МЕМ», була створена в Україні, поблизу Києва, у Феофанії, під керівництвом академіка Сергія Олексійовича Лебедєва у 1951 році. Ця машина використовувалася переважно для інженерних та наукових розрахунків. ЕОМ першого покоління, побудовані на електронних лампах, були великими за розміром, повільними, ненадійними та споживали значну кількість електроенергії, проте саме вони стали основою для розвитку майбутніх комп'ютерів.

З 1958 року розпочалася ера комп'ютерів другого покоління, які використовували транзистори замість електронних ламп. Однією з найпоширеніших транзисторних ЕОМ стала IBM 1401, випущена у 1959 році. Друге покоління комп'ютерів відзначалося вищою швидкістю роботи, меншими розмірами, нижчим енергоспоживанням та більшою надійністю. У цей період активно розвивалися методи дослідження операцій, лінійного програмування, оптимізації та розв'язання транспортних задач, де швидкість обчислень була важливішою за високу точність. Комп'ютери другого покоління чудово підходили для цього, а також оснащувалися швидкими алфавітно-цифровими друкувальними пристроями, що дозволяло виводити великі обсяги бухгалтерської документації. Таким чином, бізнес відкрив для себе значний потенціал використання ЕОМ. Інвестиції почали надходити в розробку та проектування комп'ютерів, що дало початок їх стрімкому поширенню.

У 1968 році компанія IBM представила серію комп'ютерів третього покоління IBM/360, а першими комп'ютерами, що використовували монолітні інтегральні схеми, стали IBM/370. Комп'ютери третього покоління мали ще вищу швидкість обробки даних, більший обсяг оперативної пам'яті (від 64 до 1024 кілобайт), а вартість обробки інформації знизилася в 4-5 разів. Важливою особливістю стала сумісність програмного забезпечення в межах серії ЕОМ.

У 1971 році компанія Intel зробила революційний крок у розвитку мікроелектроніки, створивши перший мікропроцесор – 4004. За ним послідувала низка вдосконалених мікропроцесорних комплексів, таких як 8080, 80186, 80286, 80386, і аж до сучасних моделей 80680. Приблизно кожні два-три роки спостерігалося значне зростання рівня інтеграції мікросхем при одночасному поліпшенні всіх інших характеристик комп'ютерів.

З початку 80-х років розпочалася активна розробка надвеликих інтегральних схем (НВІС) та комп'ютерів п'ятого покоління в Японії, США та Західній Європі. Це призвело до фактичного об'єднання комп'ютерної та телекомунікаційної техніки в інформаційні обчислювальні мережі, подальший розвиток яких визначає загальний стратегічний потенціал суспільства.

На сьогоднішній день існує величезна кількість різноманітних комп'ютерів, і їхній асортимент постійно розширюється. Вивчення такої кількості машин було б складним без певної класифікації. Комп'ютери можна класифікувати за різними критеріями, серед яких важливими є швидкодія та сфера застосування. Крім того, усі комп'ютери поділяються на аналогові, аналого-цифрові та цифрові. Також можлива класифікація комп'ютерів за їхнім

апаратним забезпеченням, призначенням, надійністю, енергоспоживанням, архітектурою та за їхнім програмним забезпеченням.

1.2 Загальні відомості про системи числення

Числові дані в комп'ютерах характеризуються:

- системою числення (двійкова, десяткова та інш.);
- видом числа (числа дійсні, комплексні, масиви);
- формою представлення числа (місце коми) – з природною (змінною), фіксованою, плаваючою комами;
- розрядною сіткою і форматом числа;
- діапазоном і точністю подання чисел;
- способом кодування від'ємних чисел – прямим, оберненим та доповняльними кодами;
- алгоритмами виконання арифметичних операцій.

Система числення – сукупність прийомів і правил, що дають змогу встановлювати взаємно однозначну відповідність між будь-яким числом і його зображенням у вигляді сукупності скінченого числа символів.

Системою числення називається сукупність цифр і правил для записування чисел. Запис числа у деякій системі числення називається його ***кодом***. Залежно від способу зображення чисел за допомогою цифр системи числення поділяють на ***позиційні*** та ***непозиційні***.

У ***непозиційних системах*** будь-яке число визначають через деяку функцію від числових значень сукупності цифр, що зображують це число. Цифри в непозиційних системах числення відповідають деяким числам. Приклад непозиційної системи – римська система числення. В електронних цифрових пристроях застосовують позиційні системи числення. Систему числення називають ***позиційною*** через те, що позначення вхідної у число цифри залежить і змінюється від її положення в запису числа.

Для запису чисел у позиційній системі числення використовують певну кількість графічних знаків (цифр і букв), які відрізняються один від одного. Число таких знаків q називається основою позиційної системи числення.

У позиційних системах числення значення кожної цифри визначається її зображенням і позицією в числі. Окремі позиції в записі числа називають розрядами, а номер позиції – номером розряду. Число розрядів у записі числа називається його ***розрядністю*** і збігається з довжиною числа.

У непозиційних системах числення значення кожної цифри не залежить від її позиції. Найвідомішою непозиційною системою є римська, в якій використовуються сім знаків – I, V, X, L, C, D, M, що відповідають таким значенням:

Таблиця 1 – Значення римської системи числення

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Правила: кожен менший знак, що розташовується праворуч від більшого, додається до його значення, ліворуч — віднімається, однакові — додаються.

Наприклад: $III-3$, $LIN-59$, $DLV-555$, $CX-110$, $XC-90$, $XXX-30$.

Недоліком непозиційної системи є відсутність нуля та формальних правил запису чисел і відповідно арифметичних дій з ними (хоч за традицією римськими числами часто користуються при нумерації розділів в книгах, віків у історії та інш.).

Система числення повинна забезпечувати:

- можливість представлення будь-якого числа в заданому діапазоні;
- однозначність, стислість запису числа і простоту виконання арифметичних операцій;
- досягнення високої швидкодії машини в процесі обробки даних.

Число в позиційній системі числення значення числа, що має зображення

$$A_p = \pm a_{n-1}a_{n-2}...a_1a_0, a_{-1}a_{-2}...a_{-m} \qquad (1.1)$$

можна представити у вигляді наступної суми:

$$\left\|A_p\right\| = \pm\left(a_{n-1}\cdot p^{n-1} + a_{n-2}\cdot p^{n-2} + ... + a_0\cdot p^0 + a_{-1}\cdot p^{-1} + ... + a_{-m}\cdot p^{-m}\right), \qquad (1.2)$$

де n – загальна кількість розрядів у цілій, а m – у дробовій частині числа, a_k – цифра k-го розряду $0 \le a_k \le p-1$, p – основа системи числення, p^{n-i} $(i=1...n)$ – вага i-го розряду цілої, а p^{-j} $(j=1...m)$ – вага j-го розряду дробової частини числа.

Приклад 1.

1) Десяткова система, $p=10, a_i \in \{0,1,2...9\}$:

$$A_{10} = (182,254)_{10} = 182,254_d = 1\cdot 10^2 + 8\cdot 10^1 + 2\cdot 10^0 + 2\cdot 10^{-1} + 5\cdot 10^{-2} + 4\cdot 10^{-3} =$$
$$= 1\cdot 100 + 8\cdot 10 + 2\cdot 1 + 2\cdot 0{,}1 + 5\cdot 0{,}01 + 4\cdot 0{,}001 =$$
$$= 100 + 80 + 2 + \frac{200}{1000} + \frac{50}{1000} + \frac{4}{1000} = 182\frac{254}{1000}.$$

2) Двійкова система, $p=2, a_i \in \{0,1\}$:

$$A_2 = (111,01)_2 = 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 + 0\cdot 2^{-1} + 1\cdot 2^{-2} = 4 + 2 + 1 + 0 + \frac{1}{4} = 7{,}25_{10}.$$

3) Вісімкова система, $p=8, a_i \in \{0,1,...7\}$

$$A_8 = (45,21)_8 = 4\cdot 8^1 + 5\cdot 8^0 + 2\cdot 8^{-1} + 1\cdot 8^{-2} = 37{,}26510_{10}.$$

4) Шістнадцяткова система, $p=16, a_i \in \{0,1,...F\}$

$$A_{16} = (DE,1B)_{16} = D\cdot 16^1 + E\cdot 16^0 + 1\cdot 16^{-1} + B\cdot 16^{-2} =$$
$$= 13\cdot 16^1 + 14\cdot 16^0 + 1\cdot 16^{-1} + 11\cdot 16^{-2} = 222{,}10510_{10}.$$

Системи числення, застосовувані в комп'ютерах

Непозиційні системи числення не придатні для застосування в комп'ютерах в силу своєї громіздкості й складності виконання арифметичних операцій, а при виборі позиційної системи числення необхідно врахувати наступні фактори:

- наявність фізичних елементів, здатних зобразити символи системи числення;
- економічність системи числення, тобто кількість фізичних елементів, необхідних для подання багато розрядних чисел;
- трудомісткість виконання арифметичних операцій;
- швидкодія комп'ютера;
- наявність формального математичного апарата для аналізу й синтезу обчислювальних пристроїв;
- зручність роботи людини з машиною;
- найбільшу завадостійкість кодування цифр.

За сукупністю цих критеріїв найбільш придатною виявляється двійкова система числення, тому що вона має ряд переваг перед іншими системами:

- для її реалізації потрібні технічні засоби з двома стійкими станами;
- представлення інформації за допомогою лише двох станів надійне та завадостійке;
- можливе використання апарату булевої алгебри для виконання логічних перетворень інформації;
- двійкова арифметика набагато простіша, ніж десяткова. Недолік двійкової системи – швидке зростання кількості розрядів, необхідних для запису числа.

За допомогою найпростіших еквівалентних перетворень будь-яку інформацію можна представити у двійковому цифровому виді. Таким чином, двійкова система числення є основною системою числення, у якій виконуються всі арифметичні і логічні перетворення інформації в комп'ютерах. В цьому випадку цифровий сигнал може приймати тільки два значення: логічні «0» та «1».

Поняття «0» та «1» являються умовними, так як вони відображають два стани цифрових систем: «ввімкнене» та «вимкнено». При цьому, якщо «1» являється високим потенціалом (рівнем електричної напруги), а «0» низьким, то мають позитивну потенціальну логіку. Відповідно при представленні «1» низьким потенціалом, а «0» високим, отримують від'ємну потенціальну логіку. Потенціальна логіка широко використовується в інтегральній мікроелектроніці, яка реалізує функціональні вузли ЕОМ, оскільки при реалізації можна уникнути використовування трансформаторів, конденсаторів і інших приладів, які важко виготовити засобами мікроелектронних технологій.

Для двійкової системи числення $p = 2, a_i \in \{0,1\}$. Поряд із двійковою системою числення для більш короткого і зручного запису двійкових кодів машинних команд при складанні програм використовуються вісімкова і шістнадцяткова системи числення.

Для вісімкової системи числення $p = 8, a_i \in \{0,1,2,3,4,5,6,7\}$.

Для шістнадцяткової системи числення $p = 16,$

$a_i \in \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$, де $A-10, B-11, C-12, D-13, E-14, F-15$.

Оскільки одні й ті самі символи цифр можуть використовуватися для запису чисел у різних позиційних системах числення, а значення числа з одним

і тим самим записом залежить від того, в якій системі його розглядати, то для уточнення основи системи числення в записі числа використовуються спеціальні кінцеві символи (пост суфікси): «b» – для двійкової, «o» – для вісімкової, «d» – для десяткової, «h» – для шістнадцяткової.

Оскільки символ букви «o» («O») нагадує символ цифри 0, то при ручному записі на папері *символ нуля перекреслюють* справо наліво по діагоналі.

Оскільки шістнадцяткове число може починатися з символу букви (і взагалі, складатися лише з самих символів букв) і завершуватися також буквою «h» («H»). то при складанні програм для позначення того, що це все ж таки число, а не змінна програми, *зліва в таких числах завжди записують цифру* 0.

Приклад 2.

Перевести з 2-ї, 8-ї, 16-ї в 10-у систему числення.

Переклад здійснюється з використанням формули розкладання (1.2).

1) $(110011{,}101)_2 = 110011{,}101_b = 1\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 + 1\cdot 2^{-1} +$

$+ 0\cdot 2^{-2} + 1\cdot 2^{-3} = 32 + 16 + 0 + 0 + 2 + 1 + \frac{1}{2} + \frac{1}{8} = 51{,}625_{10}.$

2) $(135{,}14)_8 = 135{,}14_o = 1\cdot 8^2 + 3\cdot 8^1 + 5\cdot 8^0 + 1\cdot 8^{-1} + 4\cdot 8^{-2} =$

$= 64 + 24 + 5 + \frac{1}{8} + \frac{4}{64} = 93 + \frac{1}{8} + \frac{1}{16} = 93 + \frac{3}{16} = 93\frac{3}{16} = 93{,}1875_{10}.$

3) $(10C,E)_{16} = 10C,E_h = 1\cdot 16^2 + 0\cdot 16^1 + C\cdot 16^0 + E\cdot 16^{-1} =$

$= 1\cdot 16^2 + 0\cdot 16^1 + 12\cdot 16^0 + 14\cdot 16^{-1} = 256 + 0 + 12 + \frac{14}{16} = 256{,}875_{10}.$

Двійково-десяткова система є системою кодування (Binary Coded Decimal, BCD). В цій системі цифри числа (BCD-числа) зображуються 4-х розрядним двійковим кодом, що називають тетрадою, або кодом 8-4-2-1 (Таблиця 2). Двійково-десяткова система числення використовується в комп'ютерах у якості допоміжної при уведенні/виведенні даних, та у якості основної при розв'язанні задач із малою кількістю обчислень, але з великими об'ємами даних (статистичні, економічні задачі).

Для того, щоб записати десяткове число у двійково-десятковому коді, необхідно кожну десяткову цифру замінити відповідною тетрадою коду 8-4-2-1.

Для того, щоб перевести двійково-десяткове число в десяткове, необхідно кожну тетраду BCD-числа замінити відповідною десятковою цифрою.

Приклад 3.

$$(572{,}38)_{10} = (0101\ \ 0111\ \ 0010\ ,\ \ 0011\ \ 1000)_{2-10}$$

$$\downarrow\quad\ \downarrow\quad\ \downarrow\quad\ \downarrow\quad\ \downarrow$$

$$(\ \ 5\ \ \ 7\ \ \ 2\ ,\ \ 3\ \ \ 8\ )_{10}$$

Код 8-4-2-1 являє собою приклад спрощеного використання формули розкладання, в якій враховуються лише 4 члени, що відповідають молодшим ступеням числа $2: 2^3 = 8, 2^2 = 4, 2^1 = 2, 2^0 = 1.$

Таблиця 2 – Зображення чисел у різних системах числення

Десяткове	Код 8-4-2-1	Двійкове	Вісімкове	Шістнадцяткове
0	0000	0	0	0
1	0001	1	1	1
2	0010	10	2	2
3	0011	11	3	3
4	0100	100	4	4
5	0101	1001	5	5
6	0110	110	6	6
7	0111	111	7	7
8	1000	1000	10	8
9	1001	1001	11	9
10	1010	1010	12	A
11	1011	1011	23	B
12	1100	1100	14	C
23	1101	1101	15	D
14	1110	1110	16	E
15	1111	1111	17	F

Будь-яке ціле число може бути зображене у вигляді суми відповідних степенів двійки, зокрема, якщо це число не перевищує $15d$, то для цього достатньо лише чотирьох членів формули розкладання. Для подання десяткового числа від 0 до 15 у вигляді двійкової тетроди коду 8-4-2-1 потрібно подати його у вигляді суми будь-яких десяткових чисел 8,4,2,1.

Наприклад: 8+4+2+1

$3 = 2 + 1 \quad = 1 \cdot 2^1 + 1 \cdot 2^0 \quad = 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0011_b$

$6 = 4 + 2 \quad = 1 \cdot 2^2 + 1 \cdot 2^1 \quad = 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0110_b$

$10 = 8 + 2 \quad = 1 \cdot 2^3 + 1 \cdot 2^1 \quad = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 1010_b$

$13 = 8 + 4 + 1 \quad = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 \quad = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0101_b$

1.3 Переведення чисел з однієї позиційної системи числення в іншу

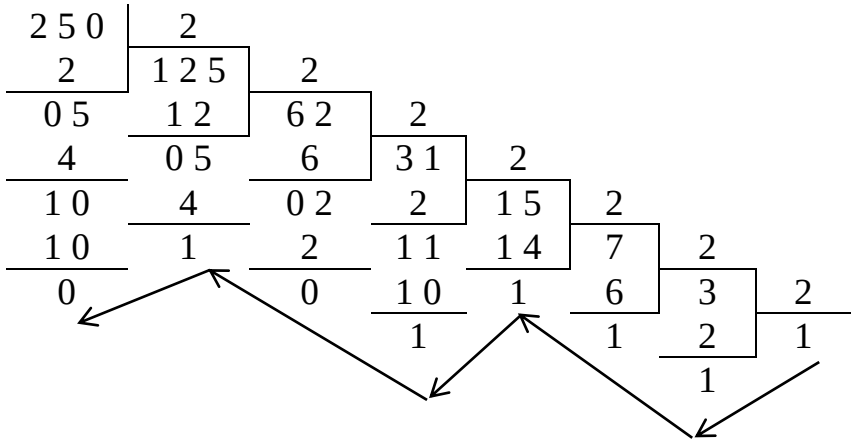
1.3.1 Переведення цілих чисел

Щоб перевести ціле число з однієї системи числення з основою p_1 в іншу систему числення з основою p_2, необхідно послідовно ділити це число й одержувані частки від ділення на основу p_2 нової системи, доти, поки чергова

отримана частка не стане менше основи нової системи числення p_2. Остання частка – старша цифра запису числа в новій системі числення, а наступні цифри – це залишки від ділення, виписані в порядку, зворотному їхньому одержанню.

Приклад 4.

Перевести $(250)_{10}$ в 2-у, 8-у, 16-у системи числення.



$(250)_{10} = 250_d = (11111010)_2 = 11111010_b.$

або так

250	125	62	31	15	7	3	1
0	1	0	1	1	1	1	1

8

2 5 0
2 4 | 3 1 | 8
1 0 | 2 4 | 3
8 | 7
2

$(250)_{10} = 250_d = (372)_8 = 372_o.$

2 5 0 | 1 6
1 6 | 1 5
9 0
8 0
1 0

$(250)_{10} = 250_d = (FA)_{16} = 0FA_h.$

Зауваження. *При виконанні переведення з однієї позиційної системи числення в іншу всі необхідні арифметичні дії виконуються за правилами тієї системи числення, у якій записане вихідне число.*

1.3.2 Переведення правильних дробів

Щоб перевести правильний дріб з системи числення із основою p_1 у систему з основою p_2, необхідно послідовно множити вихідний дріб і дробові частини отримуваних добутків на основу нової системи p_2 доти, поки чергова дробова частина не стане рівною нулю, або не буде досягнута необхідна точність перекладу. Точність перекладу визначається числом p_2^{-m}, де m –

кількість множень. Правильний дріб числа в новій системі із основою p_2 формується із цілих частин добутків, починаючи з першого.

Приклад 5.

Перевести $(0{,}3126)_{10}$ й $(0{,}625)_{10}$ у 2-ну і 16-ну ПСЧ.

0,3126	0,6252	0,2504	0,5008	00016
× 2	× 2	× 2	× 2	× 2
0,6252	1,2504	0,5008	1,0016	0,0032

Напрямок зчитування цілих частин.

Як видно, наступні множення будуть давати нульові значущі цифри (лише після 14-го з'явиться 1), тому обмежимося точністю 2^{-4}.

$(0{,}3126)_{10} = (0{,}0101)_2 = 0{,}0101_b$ з точністю 2^{-4} ($0{,}0101000000\,0001$ з точністю до 2^{-14}).

Перевірка: $(0{,}0101)_2 = 0\cdot 2^{-1} + 1\cdot 2^{-2} + 0\cdot 2^{-3} + 1\cdot 2^{-4} = \frac{1}{4} + \frac{1}{16} = \frac{5}{16} = (0{,}3125)_{10}$.

0,625	0,250	0,500
× 2	× 2	× 2
1,250	0,500	1,000

Напрямок зчитування цілих частин.

$(0{,}625)_{10} = (0{,}101)_2$. $(0{,}101)_2 = 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} = \frac{1}{2} + \frac{1}{8} = (0{,}625)_{10}$.

0,3126	0,0016	0,0256	0,4096
× 16	× 16	× 16	× 16
18756	00096	01536	24576
+ 3126	+ 00016	+ 00256	+ 4096
5,0016	0,0256	0,4096	6,5536

Напрямок зчитування цілих частин.

$(0{,}3126)_{10} = (0{,}5006)_{16} = 0{,}5006_h$ з точністю до 16^{-4}

Перевірка: $(0{,}5006)_{16} = 5\cdot 16^{-1} + 0\cdot 16^{-2} + 0\cdot 16^{-3} + 6\cdot 16^{-4} = \frac{5}{16} + \frac{6}{65536} = .$

$(0{,}5006)_{16} = 0{,}3125 + 0{,}9155\cdot 10^{-6} = 0{,}3125915541\,3138017853\,0556191$.

$$
\begin{array}{r}
0{,}625 \\
\times \quad 16 \\
\hline
3750 \\
+\,0625 \\
\hline
10{,}000
\end{array}
\qquad
\begin{array}{l}
(0{,}625)_{10} = (0, A)_{16} \\
(0, A)_{16} = 10 \cdot 16^{-1} = \dfrac{10}{16} = 0{,}625
\end{array}
$$

1.3.3 Переведення змішаних чисел

Щоб перевести змішане число з однієї системи числення із основою p_1 в іншу систему з основою p_2, необхідно окремо перевести його цілу частину за правилами переведення цілих чисел, окремо перевести його дробову частину за правилами переведення правильних дробів, а потім обидва отриманих результати об'єднати в одне змішане число у новій системі числення.

Приклад 6.

Перевести $(250{,}3126)_{10}$ у 2-у та 16-у системи числення.

$(250)_{10} = (11111010)_2 = (0, FA)_{16}$ (*Приклад 4*)

$(0{,}3126)_{10} = (0{,}0101)_2$ з точністю до 2^{-4} (*Приклад 5*)

$(0{,}3126)_{10} = (0{,}5006)_{16}$ з точністю до 16^{-4} (*Приклад 5*).

$(250{,}3126)_{10} = (11111010{,}0101)_2 = 11111010{,}0101_b$ з точністю до 2^{-4} .

$(250{,}3126)_{10} = (FA{,}5006)_{16} = FA{,}5006_h$ з точністю до 16^{-4}.

1.3.4 Переведення чисел між системами з основою степеня 2

Для переведення числа з 16-ної (8-ної) системи числення в 2-ну необхідно кожну цифру цього числа записати 4-х (3-х) розрядним двійковим числом – тетрадой (тріадой). Після чого об'єднати тетради (тріади) у єдине двійкове число, відкинувши за необхідності ліві та праві незначні нулі.

Приклад 7. Перевести $(372{,}24)_8$ й $(FA{,}5)_{16}$ в 2-ну систему числення.

$(3\ \ 7\ \ 2,\ \ 2\ \ 4)_8 = (011\ \ 111\ \ 010,\ \ 010\ \ 100)_{8-2} = (011111010{,}010100)_2$

$(F\ \ A,\ \ 5)_{16} = (1111\ \ 1010,\ \ 0101)_{16-2} = (11111010{,}0101)_2$.

Для перекладу числа з 2-ї системи числення в 16-у (8-у) необхідно розбити це число вправо й уліво від коми на групи по 4 (3) розряди – тетради (тріади). Крайню ліву й крайню праву тетради (тріади) при необхідності

доповнюють нулями, після чого кожну тетраду (тріаду) замінюють відповідною їй 16-ною (8-ною) цифрою.

Приклад 8. Перевести $(11111010,0101)_2$ у 8-у й у 16-у системи числення.

$(11111010,0101)_2 = (011\ \ 111\ \ 010,\ \ 010\ \ 100)_{8-2} = (3\ \ 7\ \ 2,\ \ 2\ \ 4)_8 = 372,24_o$

$(11111010,0101)_2 = (1111\ \ 1010,\ \ 0101)_{16-2} = (F\ \ A,\ \ 5)_{16} = 0FA,5_h$

$(101011,011)_2 = (0010\ \ 1011,\ \ 0110)_{16-2} = (2\ \ B,\ \ 6) = 2B,6_h$.

1.3.5 Арифметичні операції над двійковими числами

Арифметичні операції над багаторозрядними двійковими числами виконуються за тими ж загальними правилами, що й у десятковій системі числення. Правила виконання операцій над окремими розрядами наведені в наступній таблиці.

Таблиця 3 – Правила арифметичних дій над двійковими числами

Додавання	Віднімання	Множення
$0+0=0$	$0-0=0$	$0\cdot 0=0$
$0+1=1$	$10-1=1$	$0\cdot 1=0$
$1+0=1$	$1-0=1$	$1\cdot 0=0$
$1+1=10$	$1-1=0$	$1\cdot 1=1$

Приклад 9.

Виконати арифметичні дії з числами у двійковій системі числення:

1) $0101_2 + 0011_2 = 1000_2$

$$\begin{array}{r} 0101_2 = 5_{10} \\ +\ 0011_2 = 3_{10} \\ \hline 1000_2 = 8_{10} \end{array}$$

2) $1111_2 - 110_2 = 1001_2$

$$\begin{array}{r} 1111_2 = 15_{10} \\ -\ \ 110_2 = 6_{10} \\ \hline 1001_2 = 9_{10} \end{array}$$

3) $101_2 \cdot 11_2 = 1111_2 \qquad 5_{10} \cdot 3_{10} = 15_{10}$.

$$\begin{array}{r} 101 \\ \times\ \ 11 \\ \hline 101 \\ 101 \\ \hline 1111 \end{array}$$



Питання для самоконтролю

1. Предмет та зміст інформатики як навчальної дисципліни.
2. Основні віхи історії розвитку обчислювальної техніки.
3. Інформатика як комплексна наукова дисципліна, визначення її складових.
4. Що таке система числення?
5. Які типи систем числення Ви знаєте?
6. Що таке основа позиційної системи числення?
7. У чому полягає принцип побудови позиційної системи числення?
8. У чому полягає принцип побудови непозиційної системи числення?
9. Які основні фактори треба враховувати при виборі системи числення для застосування в комп'ютерах?
10. Яка система числення використовується для подання чисел у пам'яті комп'ютера? Чому?
11. Яким чином здійснюється переведення чисел, якщо основа нової системи числення дорівнює деякому степеню старої системи числення?
12. За яким правилом переводяться числа з десяткової системи числення?
13. За яким правилом переводяться числа в десяткову систему числення?

Індивідуальне завдання

Перетворити вказане число у відповідну систему числення

Варіант	Завдання		
1	$11001100_2 \to_{10}$ $11010101_2 \to_{10}$ $101001000_2 \to_{10}$	$10011010_2 \to_8$ $E1_{16} \to_{10}$ $147_8 \to_{10}$	$46_{10} \to_2$ $E1_{16} \to_2$ $DA_{16} \to_8$
2	$10111101_2 \to_{10}$ $11000001_2 \to_{10}$ $101101010_2 \to_{10}$	$10000010_2 \to_8$ $A1_{16} \to_{10}$ $127_8 \to_{10}$	$52_{10} \to_2$ $A1_{16} \to_2$ $FB_{16} \to_8$
3	$10000010_2 \to_{10}$ $10010001_2 \to_{10}$ $100001111_2 \to_{10}$	$10011010_2 \to_8$ $A3_{16} \to_{10}$ $141_8 \to_{10}$	$46_{10} \to_2$ $A3_{16} \to_2$ $FA_{16} \to_8$
4	$11001111_2 \to_{10}$ $10000101_2 \to_{10}$ $101111000_2 \to_{10}$	$11100010_2 \to_8$ $A5_{16} \to_{10}$ $122_8 \to_{10}$	$59_{10} \to_2$ $A5_{16} \to_2$ $CC_{16} \to_8$
5	$10000000_2 \to_{10}$ $10011111_2 \to_{10}$ $101001000_2 \to_{10}$	$10000111_2 \to_8$ $A7_{16} \to_{10}$ $133_8 \to_{10}$	$51_{10} \to_2$ $A7_{16} \to_2$ $BA_{16} \to_8$
6	$11111110_2 \to_{10}$ $10000111_2 \to_{10}$	$10101010_2 \to_8$ $A9_{16} \to_{10}$	$50_{10} \to_2$ $A9_{16} \to_2$

	$101001110_2 \rightarrow_{10}$	$130_8 \rightarrow_{10}$	$FB_{16} \rightarrow_8$
7	$11011110_2 \rightarrow_{10}$ $10010001_2 \rightarrow_{10}$ $101111000_2 \rightarrow_{10}$	$11000011_2 \rightarrow_8$ $C1_{16} \rightarrow_{10}$ $140_8 \rightarrow_{10}$	$49_{10} \rightarrow_2$ $C1_{16} \rightarrow_2$ $DA_{16} \rightarrow_8$
8	$11111111_2 \rightarrow_{10}$ $10011101_2 \rightarrow_{10}$ $101001000_2 \rightarrow_{10}$	$10011011_2 \rightarrow_8$ $C3_{16} \rightarrow_{10}$ $141_8 \rightarrow_{10}$	$54_{10} \rightarrow_2$ $C3_{16} \rightarrow_2$ $AF_{16} \rightarrow_8$
9	$11110100_2 \rightarrow_{10}$ $10000001_2 \rightarrow_{10}$ $101111000_2 \rightarrow_{10}$	$11111110_2 \rightarrow_8$ $C5_{16} \rightarrow_{10}$ $138_8 \rightarrow_{10}$	$56_{10} \rightarrow_2$ $C5_{16} \rightarrow_2$ $AB_{16} \rightarrow_8$
10	$10011101_2 \rightarrow_{10}$ $11100111_2 \rightarrow_{10}$ $101000011_2 \rightarrow_{10}$	$10111010_2 \rightarrow_8$ $C7_{16} \rightarrow_{10}$ $142_8 \rightarrow_{10}$	$57_{10} \rightarrow_2$ $C7_{16} \rightarrow_2$ $AD_{16} \rightarrow_8$

Тести для самоконтролю

1. Система числення – це:
   a) сукупність правил віднімання, додавання, множення та ділення;
   b) сукупність мов різних народів;
   c) сукупність цифр;
   d) сукупність способів і засобів запису чисел для проведення підрахунків.
2. Система числення, яка має найбільше застосування в інформатиці та програмуванні:
   a) двійкова;
   b) п’ятирічна;
   c) вісімкова;
   d) десяткова.
3. Щоб перевести число з десяткової системи числення в будь-яку, потрібно:
   a) це число помножити на основу системи числення, в яку переводимо число;
   b) це число поділити на основу системи числення, в яку переводимо число;
   c) від цього числа відняти основу системи числення, в яку переводимо число;
   d) до цього числа додати основу системи числення, в яку переводимо число.
4. Перевести число у десяткову систему числення: $23_4=?_{10}$:
   a) 5_{10};
   b) 6_{10};
   c) 9_{10};
   d) 11_{10}.
5. Перевести шістнадцяткове число $4C5_{16}$ у десяткове:

a) 1221;
b) 2141;
c) 3212;
d) 10112.

6. Щоб перевести число в десяткову систему числення, потрібно:
   a) це число помножити на основу системи числення, з якої переводимо число;
   b) це число поділити на основу системи числення, з якої переводимо число;
   c) від цього числа відняти основу системи числення, з якої переводимо число;
   d) до цього числа додати основу системи числення, з якої переводимо число.
7. Перевести шістнадцяткове число $6B_{16}$ у десяткове:
   a) 107;
   b) 108;
   c) 109;
   d) 106.
8. Перевести шістнадцяткове число FB_{16} у двійкове:
   a) $1111\ 1011_2$;
   b) $1000\ 1011_2$;
   c) $1010\ 1011_2$;
   d) $1100\ 1011_2$.
9. Перетворіть в римську систему десяткові числа:
   15, 234, 1999, 2666, 3888.
10. Перетворіть в десяткову систему римські числа:
    MMMCXI, MCMXCIX, MCXCVII, MMDCCCLXXXIX.

РОЗДІЛ 2. РОБОТА З ОФІСНИМ ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ MICROSOFT OFFICE

2.1 Основи роботи з текстовим редактором Microsoft Word

Мета: опанувати базові навички роботи з текстовим редактором Microsoft Word для ефективного створення та редагування документів.

План

1. Основні функції роботи з додатком.
2. Основи роботи з текстом у документі.
3. Робота з таблицями, графікою та спільний доступ.

Ключові терміни: *Microsoft Word, текстовий редактор, форматування тексту, таблиці, графіка, шаблони, граматика, спільна робота, інтерфейс користувача.*

Microsoft Word – потужний текстовий редактор, призначений для створення листів, факсів, звітів, графіків, записок, різноманітних ділових паперів. Крім того, Word надає широкі можливості для роботи з графічними об'єктами й таблицями, здійснює підтримку перевірки граматики, редагування і перегляду готових документів перед друком.

2.1.1 Основні функції роботи з додатком

Вікно редактора Microsoft Word містить графічні компоненти, що дозволяють спростити і полегшити роботу з додатком. Основними його частинами є панель заголовка, що містить ім'я відкритого в даний момент документа і кнопки мінімізації додатка; панель меню, що містить усі команди роботи з текстовим редактором; панелі інструментів Стандартна і Форматування, що полегшують доступ до основних команд роботи з документом і його змістом; смуги прокручування і лінійки, область задач, що відображає тематику задачі, над якою працює користувач у даний момент; панель стану, що містить основну інформацію про документ і основне вікно документа, що відображає поточне місце в документі, що відкритий або редагується користувачем. Основні функції додатка описані в таблиці 4 і зображені на рисунку 1.

Меню містить список команд. Поруч з деякими з цих команд розташовані значки, що дозволяють швидко зв'язати команду зі значком. Більшість меню знаходиться в рядку меню, що являє собою панель інструментів, розташовану у верхній частині екрана. Панелі інструментів можуть містити кнопки, меню або сполучення кнопок і меню. Microsoft Office автоматично корегує меню і панелі інструментів відповідно до частоти використання команд. При першому запуску додатка Office відображаються всі команди. Потім, в процесі роботи, меню і панелі інструментів корегуються таким чином, щоб відображати тільки команди, що найбільш часто використовуються. Щоб знайти команду, що використовується рідко або взагалі ніколи раніше не використовувалася,

необхідно скористатися стрілками в нижній частині меню, щоб відобразити повний список команд. Меню, організоване таким чином, називається персоналізованим меню. Крім основних команд, що найчастіше використовуються користувачем, існують додаткові можливості, що відображені в підменю (каскадному меню), яке знаходиться праворуч від основної команди, яка поєднує їх в одну категорію. Контекстне меню дозволяє одержати доступ до команд, доступних у даному місці документа. Основними елементами вікна Word є: стрічка команд, рядок стану, лінійки, панель швидкого доступу, робоча область документа (див. рис. 1).

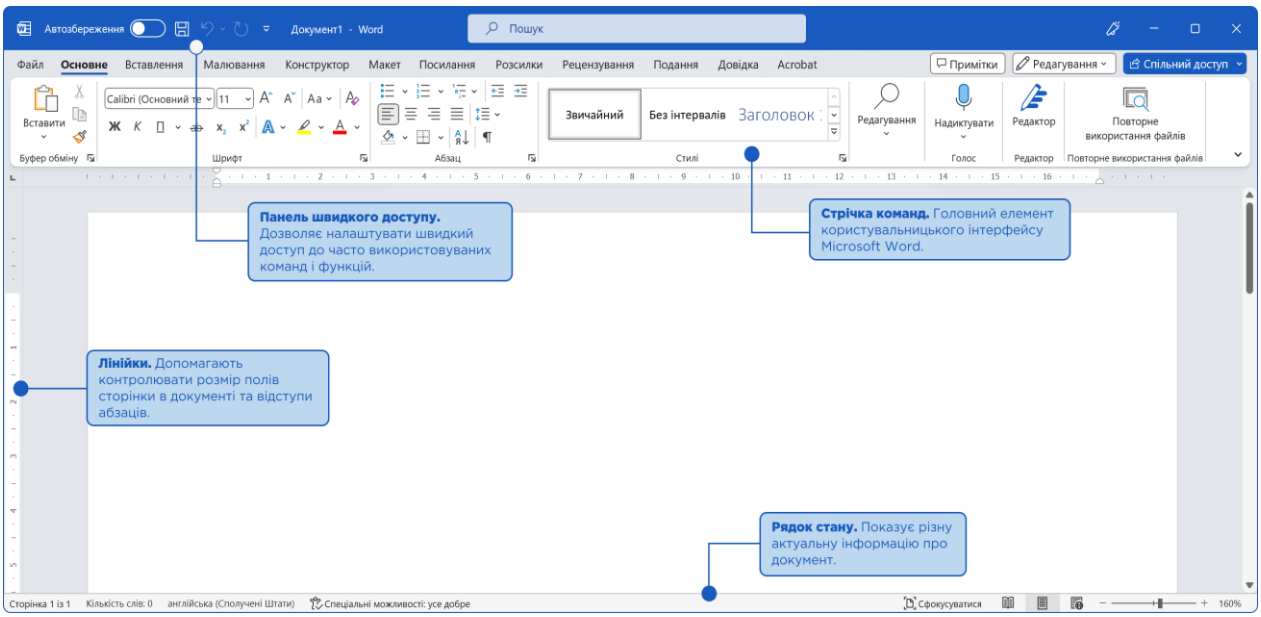


Рисунок 1 – Зовнішній вигляд вікна Microsoft Word і його основних елементів

Рядок основного меню містить наступні групи команд:

Файл: Містить команди для роботи з файлами, такі як «Створити», «Відкрити», «Зберегти», «Друк» тощо.

Основне: Містить основні інструменти для форматування тексту, такі як шрифт, розмір, колір, вирівнювання, стилі тощо.

Вставлення: Дозволяє вставляти різні об'єкти в документ, такі як таблиці, зображення, фігури, діаграми, посилання тощо.

Конструктор: Містить інструменти для створення та редагування дизайну документа, такі як теми, кольори, шрифти, ефекти тощо.

Макет: Дозволяє налаштовувати параметри сторінки, такі як поля, орієнтація, розмір, відступи тощо.

Посилання: Дозволяє додавати посилання на інші документи, веб-сторінки, електронні адреси тощо.

Розсилки: Містить інструменти для створення та надсилання електронних листів, наклейок, конвертів тощо.

Рецензування: Дозволяє перевіряти орфографію, граматику, перекладати текст, додавати коментарі тощо.

Подання: Дозволяє налаштовувати відображення документа, такі як масштаб, режим читання, розмітка сторінки тощо.

Таблиця 4 – Опис основних елементів додатка Microsoft Word

Елемент додатка	Опис
Панель заголовку	Область вікна, що містить ім'я даного додатка (Microsoft Word) і ім'я поточного документа. Розташована у верхній частині вікна.
Панель меню	Область, що містить список імен меню, доступних у Word. Меню — це набір різних команд, за допомогою яких проводиться робота з документом. Розташована панель меню під панеллю заголовка.
Панель інструментів Стандартна	Панель інструментів дозволяє здійснити швидкий доступ до функцій-команд меню, що часто використовуються. Панель інструментів Стандартна містить основні команди роботи з документом і текстом і за замовчуванням розташована під панеллю меню.
Панель інструментів Форматування	Дозволяє здійснювати швидкий доступ до функцій форматування, що найчастіше використовуються.
Крапка введення	Мерехтлива вертикальна лінія у вікні документа відображає місце введення наступного символу.
Область виділення	Область між лівою частиною вікна і лівою частиною документа, що дозволяє за допомогою покажчика миші робити виділення частини тексту документа цілими рядками. При приміщенні покажчика в область виділення він змінюється на стрілку.
Лінійка	Шкала розмічена в сантиметрах або інших одиницях виміру, що дозволяє змінювати розмітку абзаців, параметрів сторінки і встановлювати розміри стовпчиків тексту. Лінійка розташовується у верхній і лівій частині основного вікна документа.
Смуги прокручування	Смуги призначені для перегляду виду документа. Вертикальна і горизонтальна смуги прокручування розташовані в правій і нижній частині основного вікна відповідно.
Кнопки навігації	Розташовуються на вертикальній смузі прокручування і призначені для перегляду великих документів.
Панель стану	Містить інформацію про поточне положення покажчика в тексті й основні параметри стану документа. Розташована в нижній частині вікна Microsoft Word.
Кнопки мінімізації, максимізації/ відновлення і закриття	Призначені для згортання додатка Microsoft Word на панель інструментів Windows, розгортання додатка і його закриття.
Підказка	Елемент допомоги, що відображає ім'я кнопки або елемента екрана, що спливає при зупинці покажчика миші на ньому.
Панель задач	У Microsoft Word команди організовані по групах задач, що відображаються на панелі задач. Зміст панелі задач варіюється в залежності від потреб користувача. Первісна панель задач служить для створення порожнього документа/документа із шаблона або вибору документа, який необхідно відкрити для редагування. Розташовується в правій частині основного вікна документа.
Введіть питання	Текстове вікно в правому кінці панелі меню дозволяє одержати швидкий доступ до довідки Microsoft Word.

2.1.2 Основи роботи з текстом у документі

При роботі з великим документом навігація по ньому здійснюється за допомогою миші, стрілок і комбінацій клавіш, що призводять до переміщення крапки введення. При введенні тексту видалення символів здійснюється за допомогою клавіш Backspace і Delete, що видаляють символ ліворуч або праворуч від курсору відповідно. Для роботи з великими обсягами тексту при копіюванні, видаленні і вставці використовується виділення мишею або стрілками клавіатури, виділення по лівій частині документа. Скасування змін здійснюється за допомогою клавіш Скасувати/Повернути (Undo/Redo), що знаходяться на панелі інструментів Стандартна.

Форматування тексту дозволяє звернути увагу користувача на яку-небудь важливу інформацію, що знаходиться в документі. Панель інструментів Форматування (група команд Формат) призначена для роботи з такими форматами введення тексту як стиль, тип і розмір шрифту, виділення тексту курсивом, напівжирним або підкресленням, розмітка тексту. Ці ж команди дозволяють скасувати відповідне форматування.

Стиль тексту це набір форматованих характеристик тексту, який дозволяє швидко змінити різні параметри формату тексту в документі. Діалогове вікно форматування тексту надає можливості роботи з параметрами формату з одночасним відображенням результату застосування параметрів до виділеної ділянки тексту. Доступ до діалогового вікна здійснюється за допомогою пункту меню формат-шрифт або за допомогою контекстного меню для виділеної ділянки тексту (див. рис. 2).

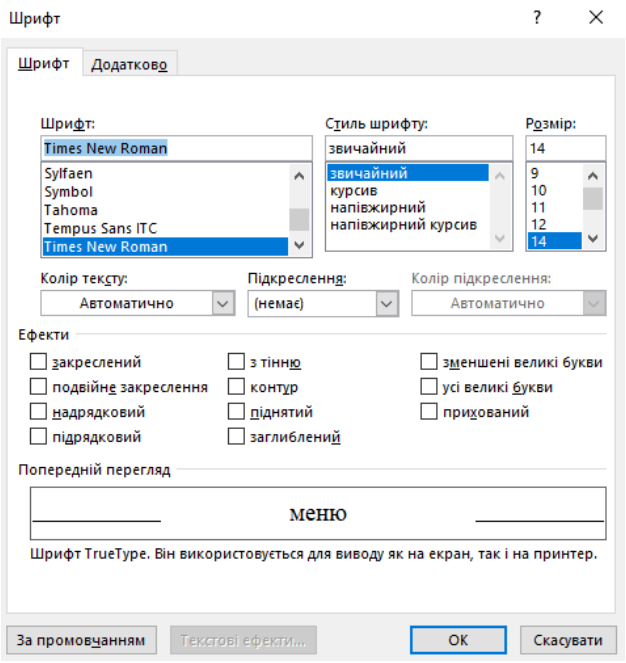


Рисунок 2 – Діалогове вікно формату шрифту

Вирівнювання тексту в документі визначає вид розміщення символів тексту в абзаці. За замовчуванням у документах, що створюються, використовується вирівнювання по лівому краю. При цьому текст вирівняний

по лівій стороні сторінки. Застосування типу вирівнювання по ширині призводить до розташування тесту рівномірно між правою і лівою сторонами сторінки, текст, вирівняний по правому краю, розташований праворуч сторінки, вирівнювання по центру призводить до рівномірної появи порожніх місць у сторін сторінки при не цілком заповнених рядках абзацу. Спеціальна команда формат за зразком призначена для копіювання шрифту, розміру, кольору тексту, а також для копіювання параметрів параграфів (табуляції, вирівнювання, параметрів абзацу).

Крім форматування параметрів тексту, Microsoft Word надає такі можливості форматування абзаців, як вирівнювання, що визначає вид вирівнювання тексту абзацу, і відступ, що визначає розмір нового рядка тексту в абзаці. Крім того, існує можливість зміни відстані міжстрокових інтервалів тексту усередині абзацу і відстаней перед і після абзацу в тексті документа (див. рис. 3).

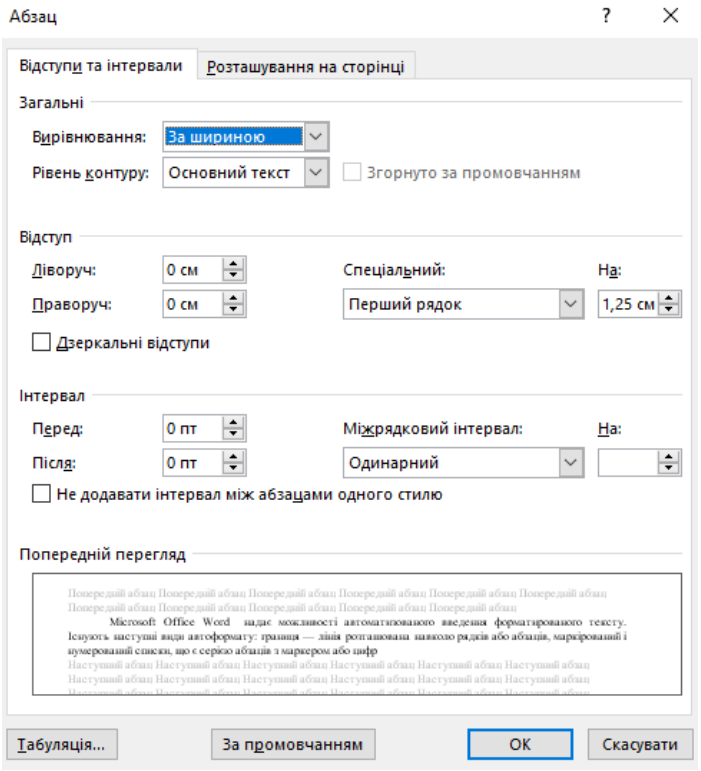


Рисунок 3 – Діалогове вікно завдання параметрів абзацу

Microsoft Office Word надає можливості автоматизованого введення форматованого тексту. Існують наступні види автоформату: границя – лінія розташована навколо рядків або абзаців, маркірований і нумерований списки, що є серією абзаців з маркером або цифрою перед ними. Модифікації зовнішнього вигляду списків задаються в спеціальному діалоговому вікні, при цьому Microsoft Word дозволяє задавати різні види знаків для ієрархічних і структурованих списків (див. рис. 4).

Автотекст – опція, що дозволяє зробити заміну визначених абревіатур при введенні повними назвами. Можливості автокорекції (автозаміну) у Microsoft Word дозволяють вводити спеціальні символи без їхнього додаткового копіювання, шляхом уведення визначених груп символів.

Список абревіатур, груп символів і їхніх замін може бути розширений користувачем.

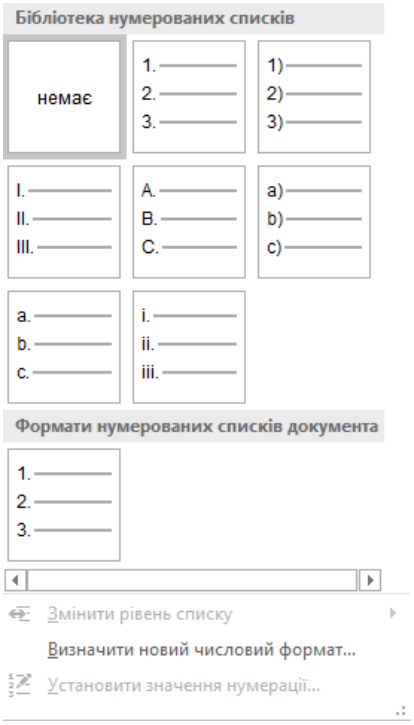


Рисунок 4 – Зовнішній вигляд діалогового вікна для роботи зі списками

Вид документа

Microsoft Word підтримує різні види представлення документа.

Звичайний. Документ відображається в спрощеному виді для полегшення введення і редагування тексту. При представленні документа в цьому виді розриви сторінок показуються пунктирними лініями і деякі параметри, такі як колонтитули, не проглядаються.

Веб-документ. Документ відображається у виді веб-страниці.

Розмітка сторінки. Документ відображається в тому виді, у якому він буде надрукований. Відображаються розриви сторінок, колонтитули, границі сторінок і графічні об'єкти.

Режим читання. Документ відображається на всю сторінку у виді, найбільш зручному для читання.

Структура. Відображення документа у виді існуючих рівнів заголовків і тексту. Призначений для перегляду і зміни ієрархічної структури тексту документа.

Схема документа. Цей вид може бути використаний поряд з будь-яким іншим видом документа. Схема документа являє собою список заголовків документа у виді окремої панелі, розташованої в лівій частині основного вікна Microsoft Word.

Ескізи сторінок. Цей вид може бути використаний поряд з будь-яким іншим видом документа. При цьому на панелі, розташованої в лівій частині основного вікна Microsoft Word відображаються ескізи сторінок документа, що можуть бути використані для швидкої навігації по документу.

Попередній перегляд документа дозволяє до друку документа побачити вид сторінок, що будуть надруковані. Параметри сторінок документа дозволяють для всього документа одразу установити розміри полів угорі внизу і по сторонах сторінки й орієнтацію тесту в документі. Зміна параметрів сторінки проводиться за допомогою лінійки або в спеціальному діалоговому вікні (див. рис. 5). Колонтитули (верхній і нижній заголовки) використовуються для додавання такої корисної інформації, як ім'я документа, номер сторінки, кількість сторінок у документі, розташування файлу документа тощо. При цьому доданий текст розташовується в полях сторінки і відображається під час друку і при попередньому перегляді документа.

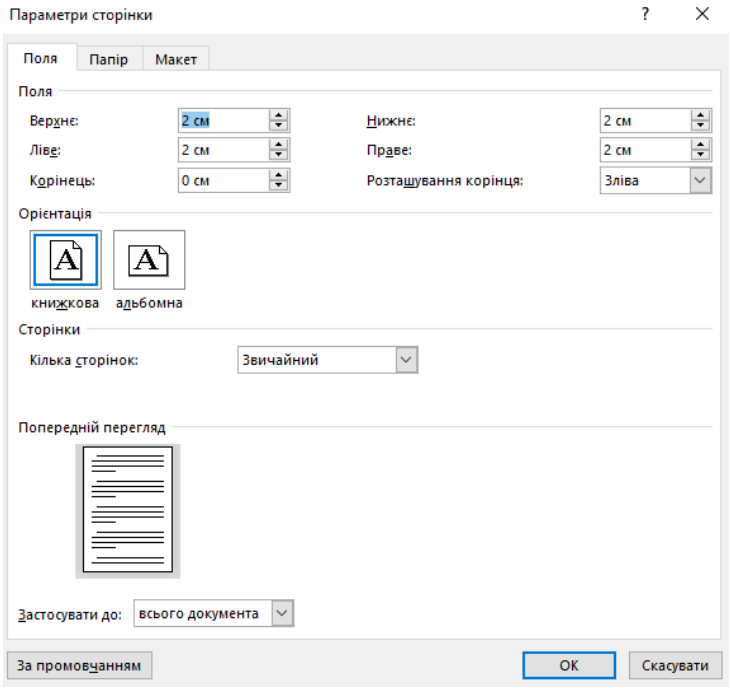


Рисунок – 5 Діалогове вікно встановлення параметрів сторінки

Табуляція – це ділянки сторінки, що можуть використовуватися для вирівнювання тексту документа. Додавання елементів табуляції дозволяє розмітити рядки тексту певним чином для створення стовпчиків тексту з можливістю заповнення інтервалів між ними спеціальними символами. Діалогове вікно установки параметрів табуляції та результат застосування табульованої розмітки до тексту (див. рис. 6).

Шаблони

Для спрощення створення документів, що містять стандартну інформацію і мають стандартний вид, використовуються шаблони. Шаблон це документ який містить маркірований текст, стилі і форматування і який може використовуватися як основа для створення інших документів з аналогічними параметрами форматування. Діалогове вікно роботи із шаблонами Microsoft Word представлене на рисунку 7.

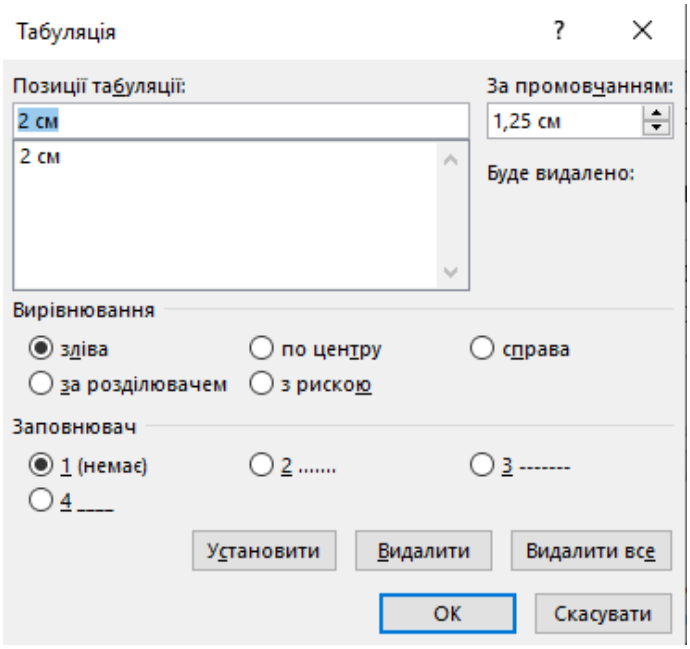


Рисунок 6 – Табуляція в Microsoft Word

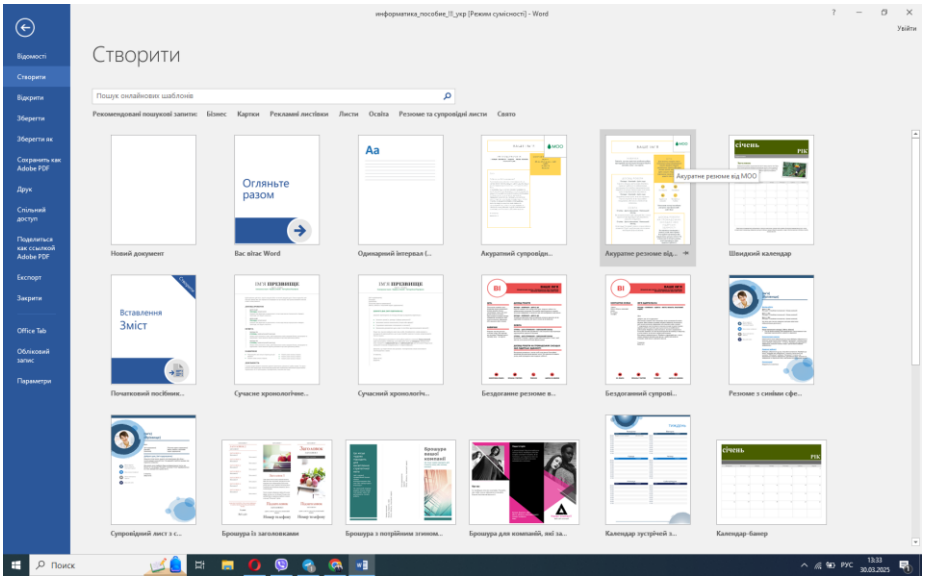


Рисунок 7 – Шаблони в Microsoft Word

Існуючі в Microsoft Word шаблони розділені на тематичні групи, при цьому крім використання готових існуючих документів-шаблонів можна скористатися майстром і створити документ або шаблон, що містить ту або іншу необхідну користувачеві інформацію. Документ, що містить визначену інформацію, може бути збережений як шаблон і використаний у подальшій роботі.

Граматика

При введенні тексту в документ Microsoft Word автоматично здійснює перевірку граматики і правопису в документі. При цьому помилки в словах виділяються червоним підкресленням, помилки побудови пропозицій –зеленим. Усі граматичні зауваження Microsoft Word не переглядаються при роздрукуванні тексту або в режимі попереднього перегляду документа.

Часто при редагуванні документів існує необхідність знайти визначене слово або словосполучення в документі і замінити його. Microsoft Word дозволяє за допомогою діалогового вікна Пошуку і Заміни (див. рис. 8) здійснювати пошук, заміну і перехід як слів і словосполучень, так і визначених параметрів форматування тексту і спеціальних символів.

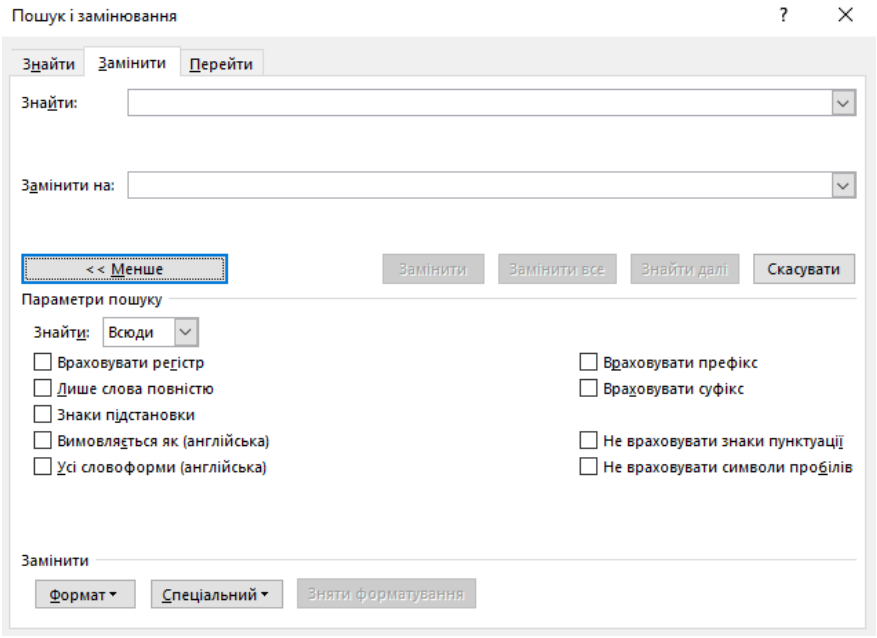


Рисунок 8 – Діалогове вікно пошуку і заміни

2.1.3 Робота з таблицями, графікою та спільний доступ

При введенні тексту у великих обсягах може виникнути необхідність розбивки тексту на колонки – блоки тексту на сторінці, кожний з яких містить продовження тексту попереднього блоку.

Перед модифікацією документа і створенням колонок необхідно врахувати ширину колонок відповідно до ширини сторінки, їхню кількість, ширину кожної з колонок і відстань між ними. Спеціальне діалогове вікно, зовнішній вигляд якого відображений на рисунку 9, дозволяє організувати текст у колонки відповідно до бажань користувача.

Ще одною широкою можливістю організації тексту є створення таблиць. Таблиця є шляхом угруповання, організації і форматування даних.

У Microsoft Word **таблиця** – це структура, що складається з прямокутних розподілів, що називаються комірками, які організовані в рядки і стовпчики. **Рядок**(ряд) – це горизонтальна група осередків, **стовпець**(стовпчик) – вертикальна. **Комірець** визначається перетинанням ряду і стовпця.

Microsoft Word надає 4 способи створення таблиць:

1. Промальовування рядів, стовпчиків і комірок з використанням покажчика миші.
2. Використання кнопки Таблиця на панелі інструментів Стандартна і завдання кількості рядів і колонок майбутньої таблиці.
3. Використання діалогового вікна Вставка таблиці з меню Таблиця.

4. Конвертування існуючого тексту в таблицю, використовуючи роздільники.

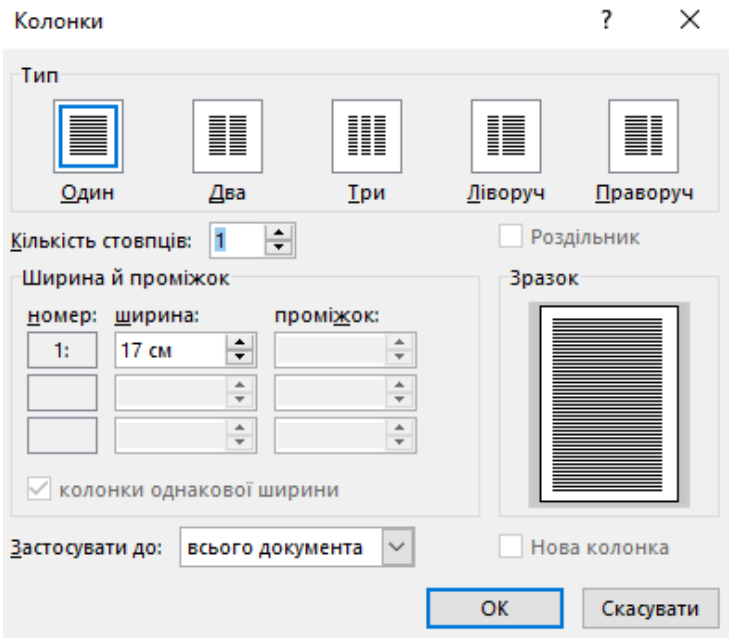


Рисунок 9 – Робота з колонками тексту

Після створення таблиці користувач може змінити її розміри шляхом додавання рядків і стовпців, а також відрегулювати їхню ширину і висоту.

Комірки таблиці можуть бути в наступному розділені на кілька комірців або об'єднані в один за допомогою контекстного меню або відповідних команд у меню Таблиця.

Microsoft Word дозволяє змінювати напрямок і вирівнювання тексту в комірцях таблиці, а також задавати спеціальний стиль таблиці за допомогою діалогового вікна Автоформат таблиці, що визначає зовнішній вигляд таблиці і її крайніх рядків і стовпців.

Виділення абзацу для залучення уваги користувача може здійснюватися шляхом додавання границь абзацу, заливання його визначеним кольором. Дані можливості представлені в спеціальній панелі інструментів Таблиці і Границі.

Робота з графікою

Рисунком у Microsoft Word є будь-який графічний об'єкт, яким може бути картинка з файлу або галереї Microsoft Word, автофігура, WordArt – об'єкт, і комбінація будь-яких графічних об'єктів у групі. Автофигурою називається графічна фігура визначеного виду – стрілка, прямокутник, напис. **WordArt** – декоративний текст, вставлений у документ Microsoft Word.

Вставка рисунка з існуючого файлу здійснюється за допомогою відповідної команди меню. Крім того, може бути використаний буфер обміну. Галерея картинок Microsoft Word містить різні рисунки, об'єднані ключовими словами. Пошук картинок у галереї по ключових словах дозволяє спростити розбивку рисунків на тематичні групи.

Після вставки рисунка в Microsoft Word можна змінити його розміри за допомогою покажчиків миші або в діалоговому вікні, яке може бути викликане з контекстного меню рисунка.

Обтікання рисунка текстом визначає спосіб розташування впровадженого рисунка в рядках тексту, що існує в документі.

Декоративний текст також є графічним об'єктом і додається для залучення уваги користувача до частини документа. У Microsoft Word розроблені різні види накреслення тексту, що включають у себе не тільки вид напису, але і спосіб розташування заливання кольором. Параметри WordArt задаються спеціальним діалоговим вікном, представленим на рисунку 10.

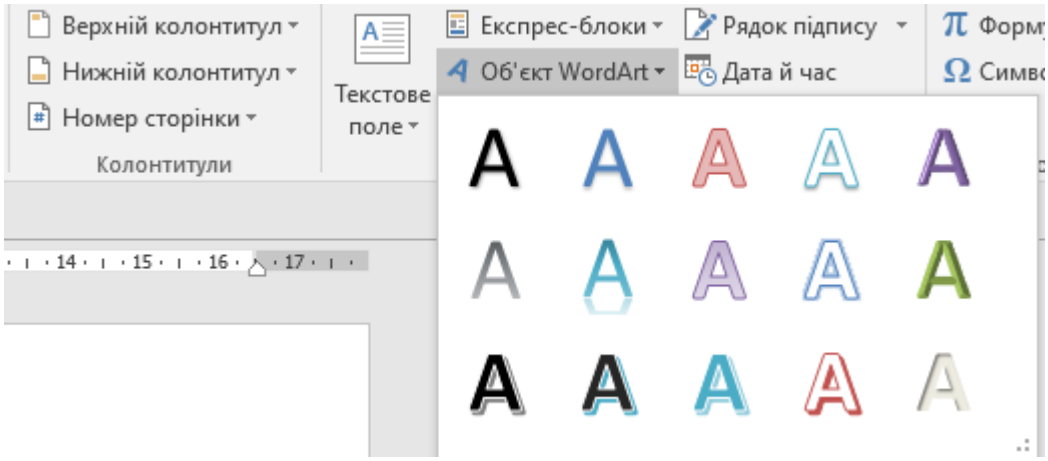


Рисунок 10 – Робота з декоративним текстом

Додавання автофігур у документ дозволяє створювати схеми і діаграми.

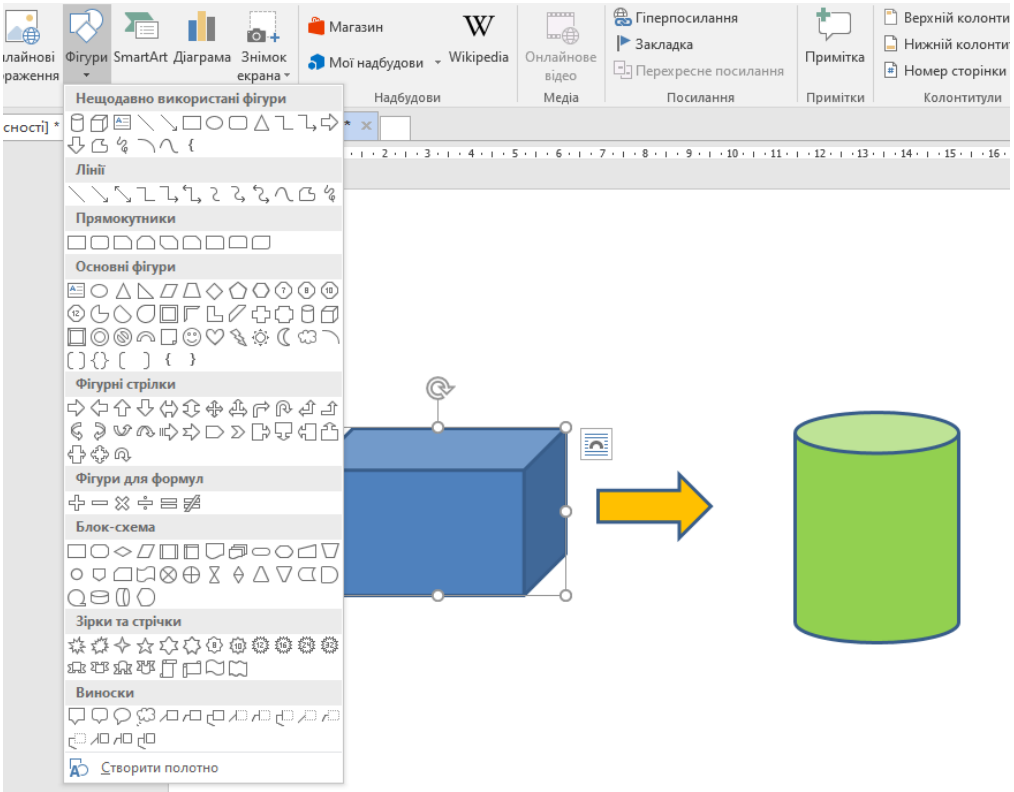


Рисунок 11 – Робота з графічними об'єктами

Всі існуючі автофігури об'єднані у тематичні групи (лінії, сполучні лінії, основні фігури, стрілки, елементи блок-схем, зірки і стрічки, винесення). Microsoft Word дозволяє користувачеві здійснювати над автофигурами різні дії – вирівнювання, зрушення і розподіл, поворот, зміна порядку автофигур, їхнє групування та разгрупування.

Заливання графічних об'єктів кольором здійснюється різними методами – градиєнтним, текстурою, візерунком або з використанням рисунка.

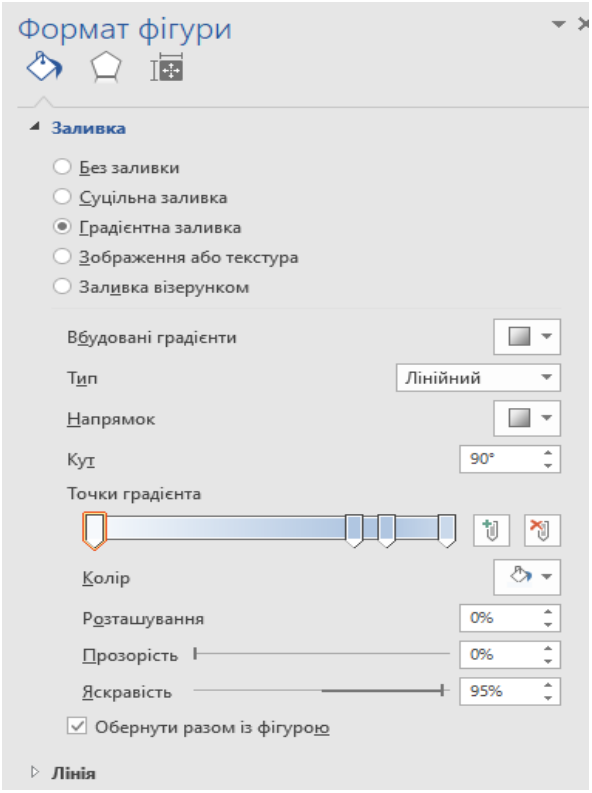


Рисунок 12 – Діалогове вікно керування фоном

Створення діаграм здійснюється за допомогою спеціального об'єкта. Діаграма це візуальне відображення зв'язаної інформації. Види діаграм, представлені в Microsoft Word наступні (див. рис.13): організаційна, циклічна, радіальна, пірамідальна, цільова діаграми і діаграма Венна.

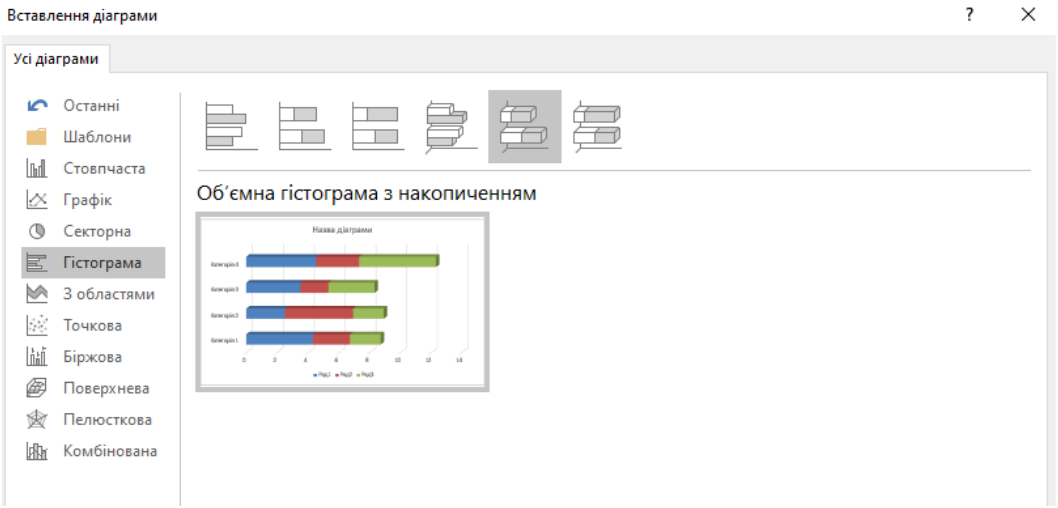


Рисунок 13 – Діалогове вікно вибору виду діаграми в Microsoft Office

Елементи діаграм, що знаходяться на одному ієрархічному рівні є співробітниками, діаграми на різних рівнях є керівником і підлеглим один щодо одного, відгалужений елемент діаграми має назву асистента. Microsoft Word підтримує також спеціальний графічний об'єкт – графік Microsoft Excel, що дозволяє представляти табличні дані в різних графічних інтерпретаціях.

Спільна робота

Існують ситуації, коли над документом працює кілька людей одночасно. Для перегляду і редагування змін інших осіб існує можливість спільної роботи над документом у Microsoft Word з відстеженням змін, зробленим кожним з авторів документа. Порівняння й об'єднання виправлень у документі за допомогою відповідної команди меню дозволяє відобразити всі зміни документа щодо вихідного варіанта і прийняти або скасувати кожне з них, залишити коментарі.

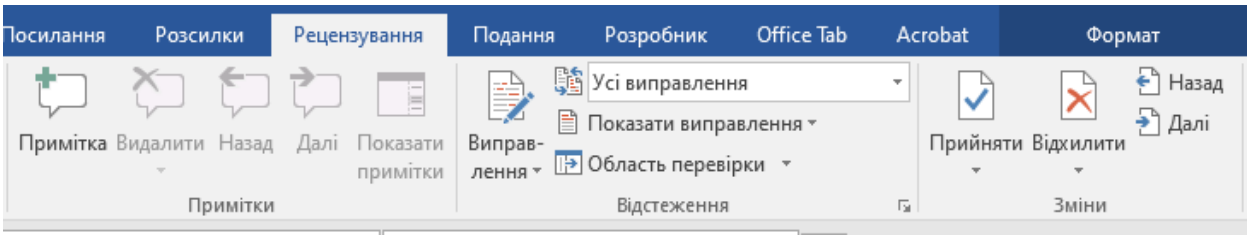


Рисунок 14 – Інструменти рецензування та внесення змін у документі

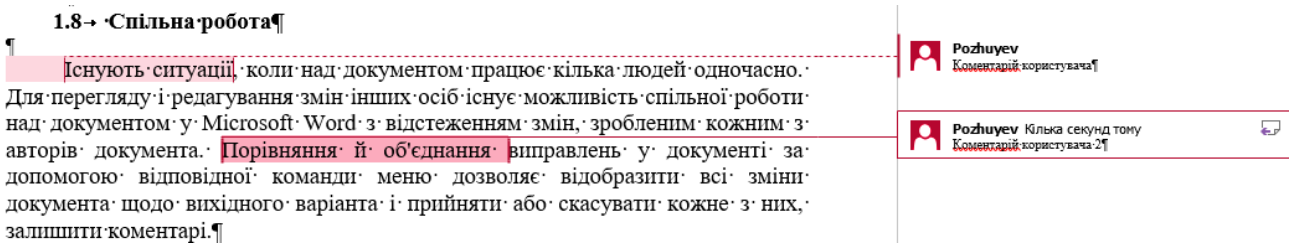


Рисунок 15 – Робота з коментарями і виправленнями



Питання для самоконтролю

1. Що таке Microsoft Word?
2. Які основні функції Microsoft Word?
3. Які формати файлів підтримує Microsoft Word?
4. Що таке інтерфейс користувача Microsoft Word?
5. Як створити новий документ у Microsoft Word?
6. Як зберегти документ у Microsoft Word?
7. Як відкрити існуючий документ у Microsoft Word?
8. Як змінити шрифт тексту в Microsoft Word?
9. Як змінити розмір тексту в Microsoft Word?
10. Як зробити текст жирним, курсивом або підкресленим у Microsoft Word?
11. Як змінити колір тексту в Microsoft Word?
12. Як вирівняти текст у Microsoft Word?

13. Як створити маркований або нумерований список у Microsoft Word?
14. Як змінити міжрядковий інтервал у Microsoft Word?
15. Як додати межі та заливку до тексту в Microsoft Word?
16. Що таке абзац у звичайному тексті й комп'ютерному тексті?
17. Як зробити перевірку правопису?
18. Що таке Колонтитули й для чого призначені?
19. Що таке Табуляція? Що можна зробити, застосовуючи табуляцію?
20. Який основній метод розриву рядка в MS Word?
21. Яким чином створити таблицю в Word?
22. Що є елементом таблиці в Word? Що може бути поміщене в комірку?
23. Яким чином додати стовпці (рядки)? Яким чином видалити стовпці (рядки)?
24. Яким чином можна змінити ширину стовпців і висоту рядків?
25. Як розбити комірку на декілька комірок? Об'єднати декілька комірок в одну?
26. Як ввести формулу в елемент таблиці?
27. Яким чином відсортувати таблицю?
28. Як вставити зображення в документ Microsoft Word?
29. Як вставити таблицю в документ Microsoft Word?
30. Як вставити діаграму в документ Microsoft Word?
31. Як створити колонтитули в документі Microsoft Word?
32. Як додати нумерацію сторінок у документі Microsoft Word?
33. Як створити зміст у документі Microsoft Word?
34. Як роздрукувати документ Microsoft Word?
35. Як перевірити орфографію та граматику в документі Microsoft Word?
36. Як використовувати функцію пошуку та заміни в документі Microsoft Word?
37. Як створити шаблон документа в Microsoft Word?

Індивідуальне завдання
Основи роботи з текстом і таблицями в текстовому редакторі Microsoft Word

Існує декілька способів створення таблиці в текстовому документі:

1) вставити таблицю простої структури;
2) накреслити таблицю довільної структури;
3) вставити таблицю з колекції шаблонів;
4) перетворити фрагмент тексту в таблицю.

Всі команди створення таблиць знаходяться в списку кнопки ***Таблиця*** групи ***Таблиці*** вкладки ***Вставка***.

І спосіб. Вставити в документ таблицю простої структури можна так:

1. Вибрати в документі місце, де потрібно вставити таблицю.
2. Виконати ***Вставка → Таблиці → Таблиця*** .
3. Виділити на схемі таблиці необхідну кількість рядків і стовпців і клацнути ліву кнопку миші.

Цим способом можна створити таблицю, в якій не більше 10 стовпців і 8 рядків. Якщо ж потрібно створити велику таблицю, то її можна вставити в текстовий документ, виконавши ***Вставка → Таблиці → Таблиця → Вставити таблицю***. Потім у відповідних полях діалогового вікна ***Вставка таблиці*** потрібно вказати кількість стовпців і рядків, спосіб визначення ширини стовпців нової таблиці і вибрати кнопку ***ОК***. Для ширини стовпців можна встановити такі значення:

- *постійна* – ширина всіх стовпців таблиці однакова і указується користувачем в полі з лічильником;
- *за змістом* – спочатку ширина стовпців автоматично встановлюється мінімальною, а при введенні даних в клітки таблиці ширина стовпця автоматично збільшується;
- *по ширині вікна* – ширина стовпців визначається автоматично діленням ширини робочої області документа на кількість стовпців таблиці.

ІІ спосіб . Таблицю будь-якої структури можна намалювати. Особливо це доцільно, коли таблиця має складну структуру. Для цього потрібно:

1. Виконати ***Вставка → Таблиці → Таблиця → Намалювати таблицю***.
2. Покажчиком, який матиме вид олівця, намалює контур всієї таблиці.
3. Намалювати лінії, які розділяють рядки і стовпці.

Якщо під час малювання таблиці складної структури були створені зайві або помилкові лінії, їх можна «стерти», скориставшись інструментом **Гумка**. Для цього слід виконати ***Конструктор → Намалювати межі → Гумка*** , навести покажчик на зайву лінію і вибрати її. Після видалення ліній кнопку ***Гумка*** слід вибрати повторно.

ІІІ спосіб дає можливість вставити в документ шаблон Експрес-таблиць з колекції що відформатували і заповнених деякими зразками даних. Для застосування вказаного способу слід виконати такий алгоритм:

1. Вибрати в документі місце, де потрібно вставити таблицю.
2. Виконати ***Вставка → Таблиці → Таблиця → Експрес-таблиці***.
3. Вибрати в списку **Вбудований потрібний шаблон**.

Користувач може створити власну експрес-таблицю, зберегти її в колекції і використовувати при необхідності. Для цього слід відформатувати вже створену таблицю за власним бажанням, виділити її і виконати ***Вставка → Таблиці → Таблиця → Експрес-таблиці → Зберегти виділений фрагмент в колекцію експрес-таблиць***.

IV спосіб полягає у перетворенні попередньо виділеного текстового фрагмента в таблицю. Щоб це зробити, необхідно виконати наступні кроки: ***Вставка → Таблиці → Перетворити в таблицю***. У відкритому діалоговому вікні «Перетворення тексту на таблицю» потрібно визначити необхідну кількість стовпців, а також обрати або вказати символи, які використовуються для розділення вмісту майбутніх клітинок таблиці (це можуть бути пробіли, знаки табуляції, символи кінця абзацу, крапки з комою та інші).

Введення даних в таблицю і переміщення по таблиці

Після того, як таблиця створена, її потрібно заповнити даними. Текст вводиться в поточну клітку таблиці по відомих вам правилах введення тексту

або, у разі використання експрес-таблиці, дані в таблиці замінюються на потрібні.

При введенні даних ширина стовпця і висота рядка автоматично змінюються, якщо введений текст не вміщається в клітку, – такий режим встановлений за умовчанням. Якщо цей режим потрібно відмінити, досить вибрати довільну клітку таблиці і виконати ***Макет → Таблиця → Властивості → Таблиця → Параметри*** і зняти мітку прапорця ***Автопідбір розмірів по вмісту***.

Щоб перемістити курсор в певну клітку, її потрібно вибрати покажчиком або скористатися клавішами управління курсором (табл. 5).

Таблиця 5 – Переміщення курсору в таблиці

Клавіші	**Результат дії**	**Клавіші**	**Результат дії**
Tab	Наступна клітка таблиці	**Shift+Tab**	Попередня клітка таблиці
→	Наступний символ тексту в клітці або наступна клітка	←	Попередній символ тексту в клітці або попередня клітка
↓	Наступний рядок тексту в клітці або в таблиці	↑	Попередній рядок тексту в клітці або в таблиці

Виділення об'єктів таблиці

Текст в таблиці, саму таблицю і її об'єкти можна редагувати і форматувати. Для виконання цих операцій об'єкти таблиці, над якими виконуються дії, необхідно зробити поточними або виділити. При виборі будь-якого об'єкту таблиці на Стрічці з'являється дві тимчасові вкладки Конструктор і Макет в тимчасовому розділі Робота з таблицями.

Виділення об'єктів можна виконати одним з двох способів:

1) Використовуючи елементи управління Стрічки: зробити потрібний об'єкт таблиці поточним, виконати ***Робота з таблицями → Макет → Таблиця → Виділити*** і вибрати в списку потрібну команду: Виділити комірку, Виділити стовпець, Виділити рядок або Виділити таблицю.

2) Використовуючи мишу:

Для виділення однієї клітки таблиці – вибрати внутрішню область клітки біля її лівої межі, коли покажчика набуває вигляд .

Для виділення одного рядка – вибрати зовнішню область рядка таблиці біля її лівої межі.

Для виділення одного стовпця – вибрати зовнішню область стовпця таблиці біля його верхньої межі .

Для виділення всієї таблиці – вибрати маркер над лівим верхнім кутом таблиці .

Для виділення декількох суміжних об'єктів таблиці – виділити область, в яку потрапляють потрібні об'єкти таблиці.

Для виділення декількох несуміжних об'єктів таблиці – виділити один об'єкт, потім, утримуючи натиснутою клавішу Ctrl, виділити решту об'єктів.

Зняти виділення об'єкту – вибрати місце поза межами виділення.

Редагування таблиці і її об'єктів

Редагування тексту в таблиці здійснюється звичайними для Word способами. Також для швидкого очищення змісту всієї таблиці або окремих її об'єктів досить їх виділити і натиснути клавішу Delete: дані будуть видалені, а сама таблиця залишиться.

Редагування структури таблиці передбачає додавання або видалення окремих її об'єктів, об'єднання або розділення елементів таблиці і ін. Всі елементи управління, які використовуються для виконання цих операцій, містяться на вкладці Макет. Більшість з цих команд розміщена в контекстному меню об'єктів таблиці.

У групі Рядка і стовпці розміщені кнопки для вставки і видалення відповідних об'єктів. Всі ці операції застосовуються до виділених або поточних об'єктів таблиці. Наприклад, для вставки додаткового рядка потрібно виділити рядок, поряд з яким вставлятиметься нова, і вибрати одну з команд Вставити зверху або Вставити знизу залежно від бажаного результату. Додатковий рядок в кінці таблиці можна також вставити, натиснувши клавішу Tab в останній клітці таблиці.

Аналогічно в таблицю можна вставити стовпці, скориставшись командами Вставити зліва або Вставити справа.

Для вставки в таблицю окремої клітки потрібно відкрити діалогове вікно вказаної групи і вибрати спосіб вставки - із зрушенням управо або вниз. У разі вибору способу із зрушенням управо в поточному місці з'явиться нова клітка, а всі інші клітки рядка перемістяться правіше. При виборі способу із зрушенням вниз в таблиці на поточному місці з'явиться новий рядок.

Додати до таблиці рядок або стовпець також можна, вибравши відповідний перемикач у вікні Додавання комірок.

Видалити будь-який виділений об'єкт (рядок, стовпець, клітку таблиці або всю таблицю) можна за допомогою команд списку Видалити або клавішею BackSpace.

Якщо декілька кліток таблиці, які розташовані поряд, потрібно *об'єднати в одну*, то їх слід виділити і вибрати кнопку Об'єднати комірки в групі Об'єднати. В цьому випадку вміст кліток «склеюється» і розміщується в об'єднаній комірці.

Для розділення однієї або декількох суміжних кліток потрібно виконати такий алгоритм:

1. Виділити потрібні клітки таблиці.
2. Вибрати на вкладці Макет в групі Об'єднати кнопку Розбити комірку .
3. Вказати в діалоговому вікні Розбиття комірки, на скільки рядків і стовпців слід розділити комірку.
4. Вказати спосіб розміщення тексту після розділення, використавши мітку відповідного прапорця.
5. Вибрати ОК.

Розділити або об'єднати клітки можна також, використавши інструменти Намалювати таблицю або Гумку групи Намалювати межі вкладки Конструктор.

Кнопку Розбити таблицю групи Об'єднати використовують, коли таблицю в документі потрібно розділити на дві частини для введення між ними тексту або перенесення частини таблиці на іншу сторінку. Для цього курсор потрібно розмістити в тому рядку таблиці, перед якою планується розрив таблиці.

Якщо таблиця велика і розміщується більш ніж на одній сторінці текстового документа, то заголовки стовпців на інших сторінках можна вставити автоматично. Для цього слід виділити рядок заголовка на початку таблиці і виконати ***Макет → Дані → Повторити рядки заголовків*** .

У тому випадку, коли потрібно видалити таблицю з документа, а її вміст представити у вигляді звичайного тексту, можна виконати автоматичне перетворення таблиці в текст. Для цього потрібно виділити таблицю, виконати ***Макет → Дані → Перетворити в текст*** і в діалоговому вікні Перетворення в текст вказати роздільник – символ, який слід вставляти в місцях попереднього ділення тексту на окремі клітки.



Завдання для самостійного виконання

Завдання 1.

Створіть таблицю за зразком. Відформатуйте таблицю по вказаних значеннях властивостей. Збережіть документ у власній папці у файлі з ім'ям завдання 2.1.docx.

Держава	Площа (тис. кв. км)	Населення (млн. чол.)	Столиця
Німеччина	357	81,3	Берлін
Франція	552	65,6	Париж
Велика Британія	244	62,9	Лондон
Бельгія	31	10,4	Брюссель
Швейцарія	41	7,9	Берн
Польща	323	38,4	Варшава
Болгарія	111	7,1	Софія
Греція	231	10,7	Афіни
Італія	301	61,3	Рим
Іспанія	505	47,1	Мадрид
Україна	603	44,9	Київ

Об'єкт таблиці	Шрифт	Розмір символів	Колір символів	Вид начертання	Вирівню-вання
Перша строчка	Comic Sans MS	14	Темно-зелений	Напів-жирний	По центру
Четверта строчка	Arial	12	Темно-синій	Курсив	По лівому краю
Перший стовпець	Times New Roman	14	Чорний	Звичайний	По центру

Завдання 2.

Створіть таблицю за зразком.

№ п/п	ПІБ студента	Предмети					Середній бал
		Інформа-тика	*Біологія*	*Хімія*	*Матема тика*	*Іноземна мова*	
1	Орел Д.В.	10	7	9	8	5	
2	Назаренко К.І	7	3	4	6	9	
3	Сидоров А.А.	5	9	10	8	4	
4	Ворон С.А.	8	6	5	4	7	
5	Арбузов В.І.	10	11	11	11	9	
6	Чайка Н.О.	4	4	5	3	2	
7	Маркова Т.П.	6	3	6	7	5	
8	Ігоров М.П.	10	6	9	8	7	
9	Местний С.С.	8	4	8	5	3	
10	Антонов В.І.	3	7	7	6	9	
11	Озерна Т.Г.	6	9	5	4	7	
12	Бубка Л.М.	10	5	7	10	8	
13	Заяць К.Д.	2	3	4	4	2	

Виконання обчислень в таблиці

Текстовий процесор має спеціальні засоби для виконання обчислень над числовими даними, розміщеними в клітках таблиці. Для цього потрібно у відповідних клітках таблиці записати формули для обчислення. Формула задається виразом, який починається із знаку =. У цьому виразі можуть бути використані числа, знаки арифметичних дій, а також:

– посилання на діапазони кліток:

- LEFT (англ. left – лівий, зліва) – клітки, розташовані зліва від клітки з формулою;
- RIGHT (англ. right – правий, справа) – клітки, розташовані праворуч від клітки з формулою;
- ABOVE (англ. above – над, вище) – клітки, розташовані вище клітки з формулою;
- BELOW (англ. below – нижче, внизу) – клітки, розташовані нижче клітки з формулою;

– вбудовані функції Word, наприклад:

Функція	Призначення	Приклад запису
AVERAGE (англ. average - в середньому, середнє число)	Знаходження середнього арифметичного значення чисел у вказаному діапазоні кліток	=AVERAGE (LEFT)
MAX (англ. maximum - максимальний, найбільший)	Знаходження найбільшого значення у вказаному діапазоні кліток	=MAX (RIGHT)
MIN (англ. minimum - мінімальний, найменший)	Знаходження найменшого значення у вказаному діапазоні кліток	=MIN (ABOVE)
SUM (англ. sum - сума, підсумок)	Знаходження суми чисел у вказаному діапазоні кліток	=SUM (BELOW)

Щоб здійснити обчислення, необхідно виконати такий алгоритм:

1. Встановити курсор в клітку, в якій повинен розміститися результат обчислень.
2. Виконати команду ***Макет → Дані → Формула*** , яка відкриває діалогове вікно ***Формула***
3. Записати у відповідному полі діалогового вікна формулу, ввівши її з клавіатури або вставивши необхідну функцію із списку ***Вставити функцію***.
4. Вибрати при необхідності формат отримуваного результату.
5. Вибрати кнопку ОК.

Зміна даних в клітках таблиці не змінює автоматично результати обчислень. Для оновлення результатів необхідно виділити таблицю або клітку з формулою і натиснути клавішу ***F9*** або в контекстному меню клітки вибрати команду ***Відновити поле*** .

2.2 Створення мультимедійних презентацій за допомогою Microsoft Power Point

Мета: сформувати базові навички роботи з програмним забезпеченням Microsoft PowerPoint з метою створення інформативних та візуально привабливих мультимедійних презентацій шляхом освоєння основних інструментів структурування контенту, інтеграції різноманітних об'єктів (текст, зображення, відео, аудіо, діаграми) та застосування ефектів анімації та переходів для оптимізації сприйняття інформації.

План

1. Основні функції роботи з додатком
2. Оформлення презентацій
3. Робота з текстом доповіді презентації
4. Робота з графічними об'єктами.
5. Організація текстової і графічної інформації слайда

Ключові терміни: *Microsoft PowerPoint, презентація, слайд, текст, форматування, зображення, фігура, SmartArt, анімація, показ слайдів, дизайн презентації, мультимедіа, об'єкт, шаблон, макет.*

Microsoft PowerPoint – офісна програма, призначена для створення презентацій, що супроводжують доповіді, заміток доповідача, роздавального матеріалу тощо. Усі ці можливості створюються й утримуються в одному файлі презентації і можуть бути змінені або додані в будь-який момент роботи над презентацією і доповіддю.

2.2.1 Основні функції роботи з додатком

Документом у Microsoft PowerPoint є файл презентації. Основне вікно додатка містить графічні компоненти, що значно полегшують роботу користувача над проектом доповіді.

Таблиця 6 – Опис основних елементів додатка Microsoft PowerPoint

Елемент додатка	Опис
Рядок заголовка	Область вікна, що містить ім'я даного додатка (Microsoft PowerPoint) і ім'я поточного документа. Розташована у верху вікна.
Панель меню	Область, що містить список імен меню, доступних у PowerPoint. Меню це набір різних команд, за допомогою яких здійснюється робота над презентацією. Розташовано панель меню під панеллю заголовка.
Панель інструментів Стандартна	Панель інструментів дозволяє здійснити швидкий доступ до найбільш популярних функцій-команд меню. Панель інструментів Стандартна містить основні команди роботи з презентацією і її даними і за замовчуванням розташована під панеллю меню.
Панель інструментів Форматування	Дозволяє здійснювати швидкий доступ до функцій форматування.
Підказка	Елемент допомоги, який відображує ім'я кнопки або елемента екрана, що спливає при зупинці покажчика миші на ньому.
Панель задач	У Microsoft PowerPoint команди організовані по групах задач, що відображаються на панелі задач. Зміст панелі задач варіюється в залежності від потреб користувача. Первісна панель задач служить для створення порожньої презентації із шаблону або вибору презентації, яку необхідно відкрити для редагування. Розташовується в правій частині основного вікна документа.
Введіть питання	Текстове вікно в правому кінці панелі меню дозволяє одержати швидкий доступ до довідки Microsoft PowerPoint
Панель Структура/Слайди	Панель із закладками, що відображає всі аркуші презентації або у виді іконок, або у виді форматованого тексту, що міститься на кожнім листі.
Панель слайда	Основне вікно додатка Microsoft PowerPoint, що відображає зовнішній вигляд слайда, над яким у даний момент працює користувач.

Види перегляду презентації

У Microsoft PowerPoint існує чотири види, що дозволяють полегшити роботу користувача по створенню, організації і показу презентацій: Звичайний, Сортувальник слайдів, Сторінки заміток і Показ слайдів.

У **Звичайному** режимі користувач може вводити текст і працювати з мініатюрами слайдів у панелі структури/слайдів, працювати з окремим слайдом в основному вікні і створювати замітки до презентації на панелі заміток. Цей режим найбільш зручний для введення тексту і різних графічних і мультимедийных об'єктів.

У режимі **Сортувальника слайдів** проглядаються у виді мініатюр відразу всі слайди презентації. Цей вид дуже зручний при реорганізації порядку слайдів у презентації.

Режим **Сторінки заміток** відображає коментарі до слайдів у весь екран. Тільки в цьому режимі користувач може додавати до заміток графічні об'єкти.

Показ слайдів – режим, що використовується безпосередньо при доповіді перед аудиторією. Слайди відображаються на весь екран і змінюють один одного в порядку, визначеному користувачем.

2.2.2 Оформлення презентацій

При створенні нової презентації за замовчуванням створюється презентація з одного порожнього білого листа. У процесі роботи над доповіддю користувач додає слайди, змінює їх зовнішній вигляд і встановлює порядок зміни слайдів у презентації. При роботі з оформленням презентації рекомендується використовувати шаблони оформлення. Шаблон оформлення – це презентація з професійно розробленими форматами і кольоровими схемами, у яку необхідно тільки додати потрібний текст. Шаблон оформлення містить у собі фонову графіку, місце розташування і формат тексту, що вводиться, визначає вид маркірованих і нумерованих списків.

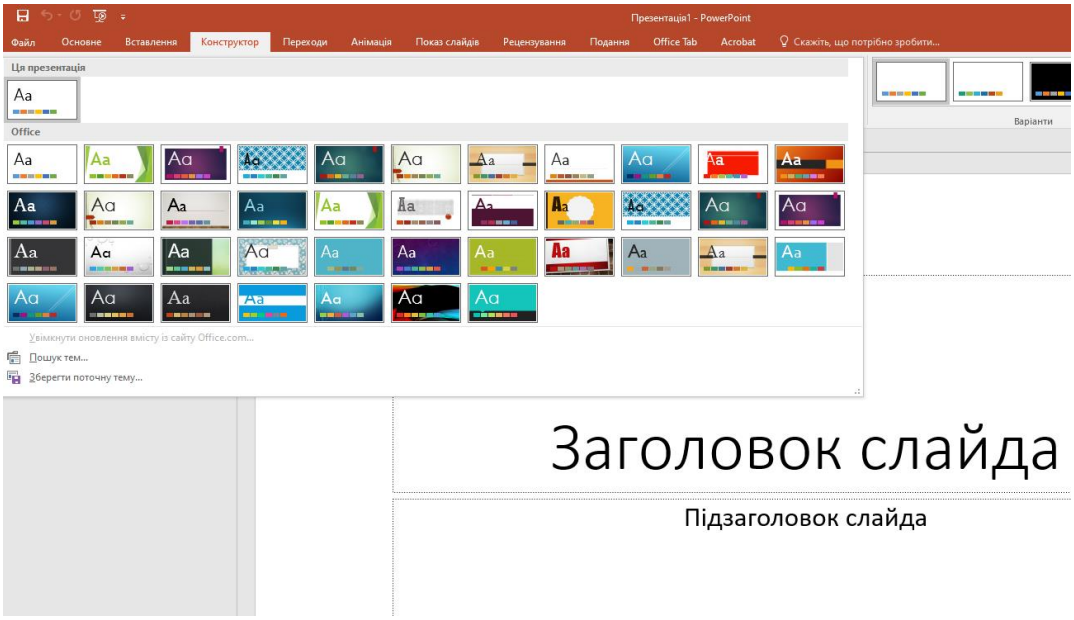


Рисунок 16 – Робота із шаблонами оформлення презентацій

У поставку Microsoft PowerPoint входять два спеціальних слайди, називаних майстрами – Майстер Слайда і Майстер Заголовка. Це зразки, що відображають шаблони оформлення першого і наступного слайдів презентації. З їхньою допомогою користувач може легко створити і змінити частини зовнішнього вигляду презентації. У презентації може бути декілька пар майстрів, у залежності від кількості шаблонів оформлення, що використовуються у презентації.

Майстер Слайда контролює властивості кожного слайда з аналогічним оформленням у презентації. Усі характеристики (фон, колір тексту, тип і розмір

шрифту, вид розташування рівнів тексту) є загальними для всіх подібних слайдів у презентації і при їх зміні на Майстру Слайда відбувається одночасна їх заміна на підлеглих слайдах. Це дозволяє легко оформлювати слайди презентації в одному загальному стилі.

Майстер Заголовка відповідає спеціальному слайдові презентації з розміткою заголовка. Такі слайди звичайно випереджають розділи доповіді і також можуть бути оформлені в єдиному стилі за допомогою зразка.

Після видозміни шаблона оформлення слайда користувач може зберегти вид майстрів презентації для наступного використання як шаблон.

При визначенні кольорового оформлення дизайну слайда використовуються кольорові схеми – набір з восьми кольорів, розроблених для фону, тексту, ліній, тіней, заливань і акцентів слайдів презентації.

Кожен шаблон оформлення підтримує кілька кольорових схем, які користувач може застосувати до слайда для додання презентації привабливого виду. Панель кольорових схем для поточного шаблона оформлення слайда містить усі кольорові схеми даного шаблона і дозволяє створювати нові кольорові схеми на основі наявних.

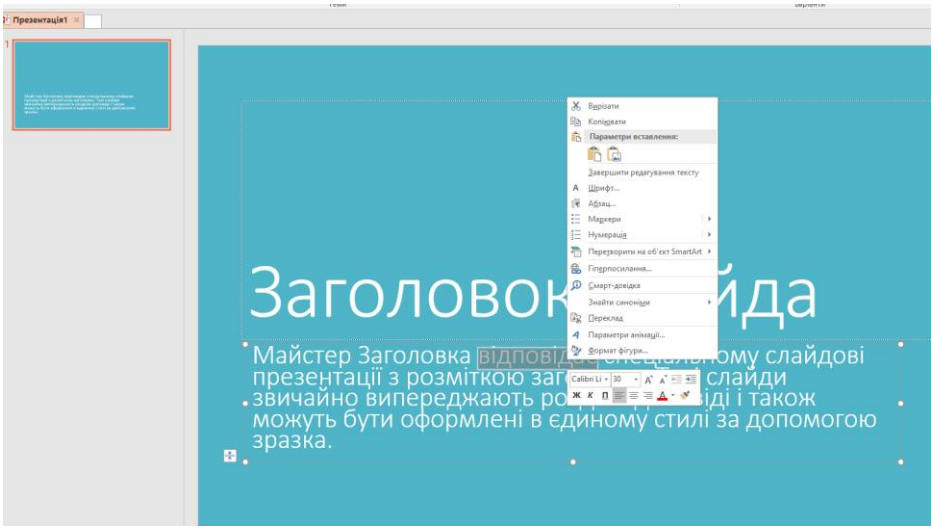


Рисунок 17 – Вид основного вікна Microsoft PowerPoint при роботі зі зразками слайдів

При роботі над зовнішнім виглядом презентації користувач може змінити фон слайду, застосувавши не тільки різні кольори, але і способи заливання – градієнтну, текстуру, візерунок або вставити який-небудь рисунок.

При роботі зі змістом слайда користувач може змінювати вид його наповнення за допомогою спеціальної панелі розмітки. Розмітка слайда визначає тип і кількість об'єктів наповнення з використанням заголовка, текстових блоків, графічних об'єктів. Існує велика кількість шаблонів розмітки слайда, розбитих на тематичні групи, що можуть бути застосовані до поточного слайда, над яким працює користувач, у будь-який момент – як до створення наповнення, так і для видозміни взаємо розташування існуючих об'єктів слайда.

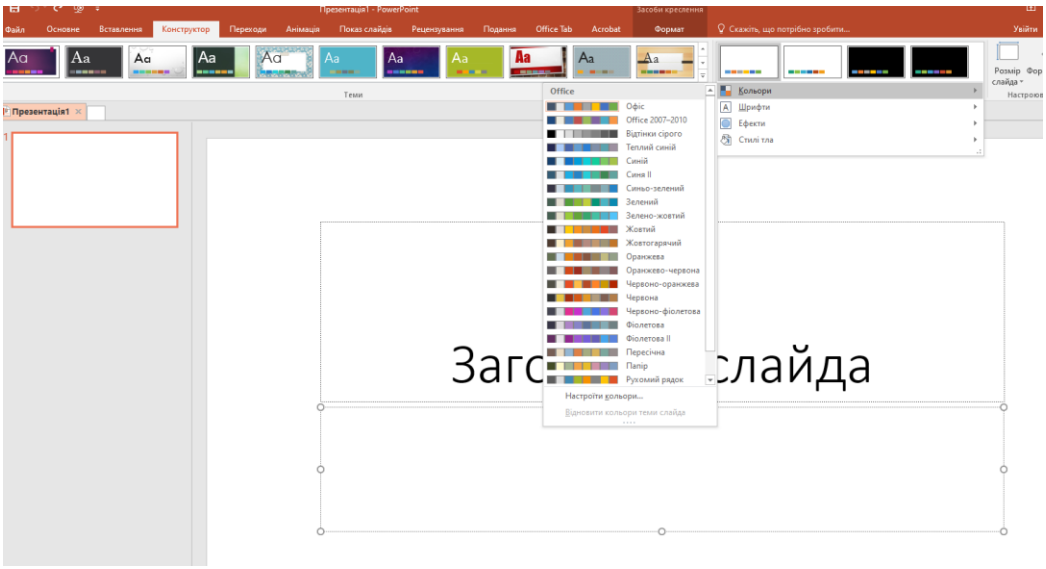


Рисунок 18 – Створення нової кольорової схеми на основі існуючої для обраного шаблона оформлення

2.2.3 Робота з текстом доповіді презентації

Додавання в презентацію нових слайдів відбувається двома способами – додавання порожніх слайдів і додавання слайдів із вже існуючої презентації. Найчастіше це слайди з титульним листом і атрибутами підприємства, слайди співавторів.

Якщо в користувача вже є готова доповідь, на основі якого необхідно створити презентацію, Microsoft PowerPoint дозволяє створити слайди презентації зі структури Microsoft Word. Аналогічно структуру слайдів можна зберегти у форматі документа з використанням форматування тексту.

Друк презентації

Найчастіше при доповіді необхідно видати слухачам роздавальний матеріал. Для цього найпростіше роздрукувати готову презентацію.

Аркуші презентації можуть містити додаткову інформацію про доповідь, номера сторінок, ім'я автора тощо. Додавання цієї інформації організовується за допомогою верхнього і нижнього колонтитулів у спеціальному діалоговому вікні (див. рис. 19).

При друці презентації задаються наступні параметри: діапазон друку, інформація для друку (слайди, замітки, видачі або структура), число копій, параметри принтера.

При друці слайдів роздруковується кожен слайд на окремому листі. Видачі дозволяють надрукувати одну, дві, три, чотири, шість і дев'ять слайдів на одній сторінці, розташованих по вертикалі або горизонталі.

Вибір друку заміток приводить до роздруківки заміток до кожного слайда на окремому листі. У режимі печатки структури видається текст слайдів відповідно до заданого форматування.

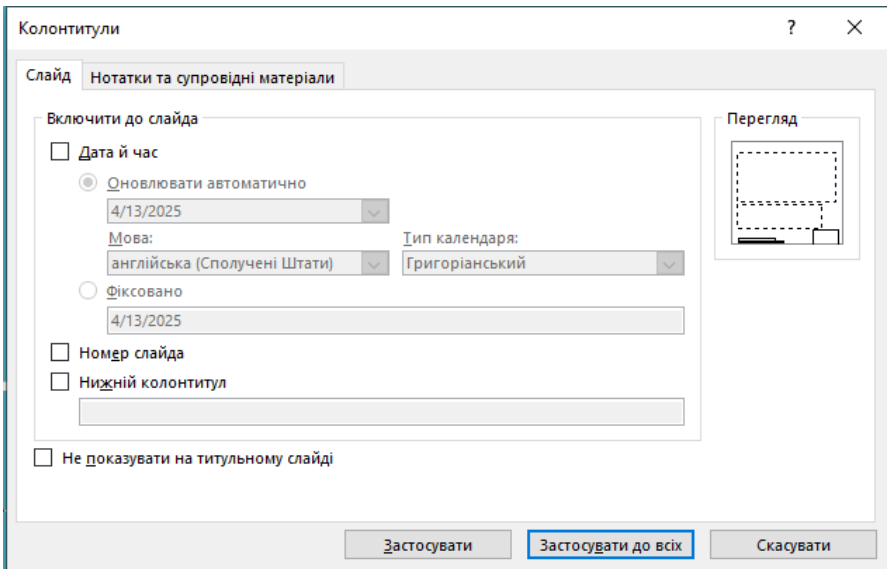


Рисунок 19 – Діалогове вікно установки колонтитулів слайдів

2.2.4 Робота з графічними об'єктами

Крім звичайного тексту, ефективні презентації містять різні фігури і графіки, що уточнюють думку доповідача і залучають увагу слухача. Користувач може створювати різні графічні об'єкти і модифікувати такі їх атрибути, як розмір, стиль ліній, колір, тіні, обсяг тощо.

Зображення об'єктів у Microsoft PowerPoint здійснюється за допомогою спеціальної панелі, на яку винесені і згруповані усі функції роботи з графічними об'єктами. Вкладка малювання основних об'єктів дозволяє легко створювати основні геометричні фігури, лінії, елементи блок-схем, стрічки, стрілки, винесення. При необхідності важливий текст може бути створений за допомогою об'єкта WordArt, зовнішній вигляд якого задається одним з 30 шаблонів з різними параметрами заливання. Над створеними фігурами можна робити різні дії – поворот, відображення, зсув, вирівнювання і розподіл, змінювати їх порядок і групувати між собою. Для будь-якої геометричної фігури в Microsoft PowerPoint задається вид її обтікання текстом, що визначає її місце розташування на слайді, заповненому інформацією, розмір, колір, параметри ліній, заливання, можуть бути додані тінь і об’єм.

2.2.5 Організація текстової і графічної інформації слайда

Користувач може для зручності слухачів організовувати на основі даних доповіді таблиці і графіки. Таблиці призначені для розташування даних у рядки і стовпці. Організовані подібним чином дані можуть бути відформатовані за допомогою спеціальних шаблонів для таблиць з виділенням крайніх рядів даних, для комірок таблиці може бути задані заливання і вид границь. На основі табличних числових даних можуть бути побудовані графіки і діаграми за допомогою Microsoft Excel.

Взаємозв'язок між різними об'єктами може бути відображений за допомогою діаграм різних видів: організаційної, циклічної, радіальної, пірамідальної, цільовий або діаграми Венна.

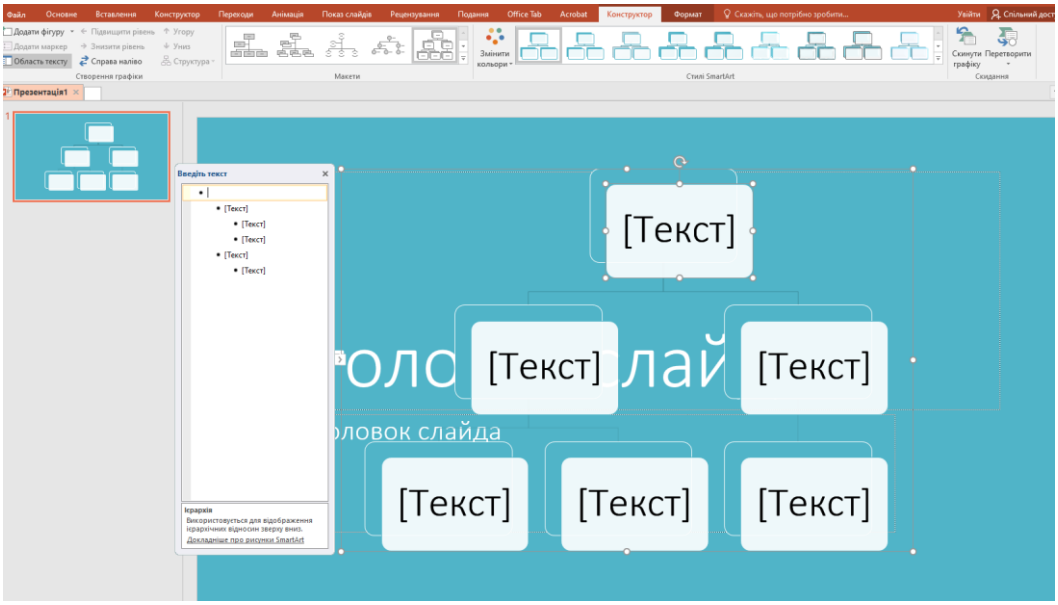


Рисунок 20 – Слайд з організаційною діаграмою

Показ презентацій. При показі презентації використовується спеціальний вид, при цьому слайди змінюють один одного в порядку, визначеному користувачем і займають весь екран монітора або проектора. При показі слайдів навігація може здійснюватися як від слайда до слайда, з можливістю повернення на попередній, так і шляхом переходу на будь-який слайд презентації. У процесі показу слайдів користувач може робити замітки прямо на аркушах презентації з використанням фломастера або маркера виділення різних колборів з можливістю наступного збереження заміток.

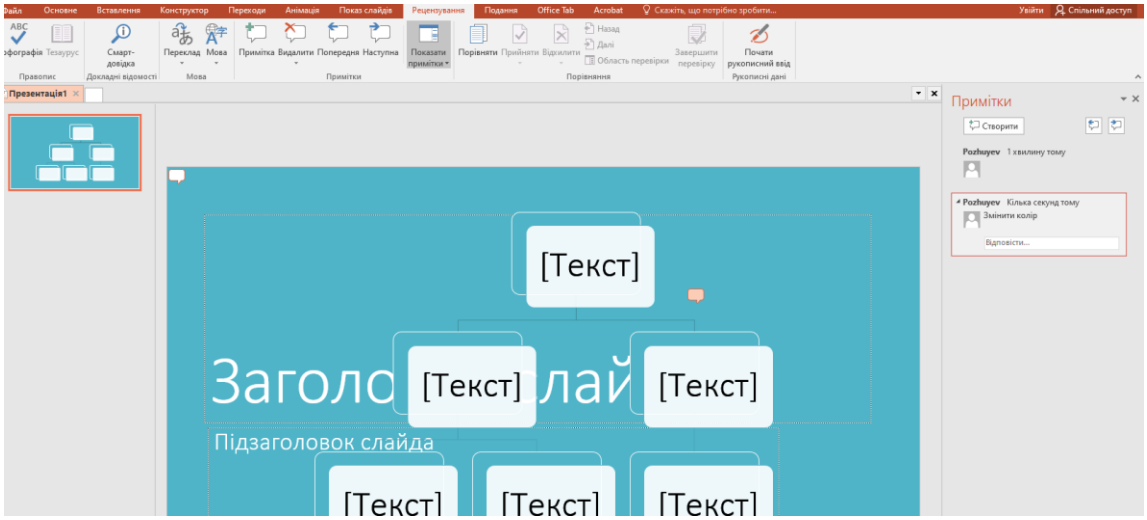


Рисунок 21 – Замітки на слайді, створені в процесі показу презентації

Зміна слайдів з використанням різних переходів дозволяє удержати інтерес слухачів і виділити ключові моменти доповіді. За допомогою спеціальної панелі користувач може задати вид зміни слайдів, швидкість графічного ефекту і час показу кожного слайду в секундах.

Всі об'єкти слайдів Microsoft PowerPoint можуть з'являтися в процесі доповіді користувача з застосуванням різних анімаційних ефектів. Анімаційна

схема задає порядок і тип появи кожного об'єкта слайда. Всі існуючі типи анімаційних схем розбиті на тематичні групи та знаходяться на спеціальній панелі оформлення слайда з застосуванням ефектів анімації. Крім того, для будь-якого об'єкту або частини тексту можуть бути задані власні анімаційні схеми появи на екрані у виді входу, виділення, виходу або намальованого користувачем шляху переміщення (див. рис. 22).

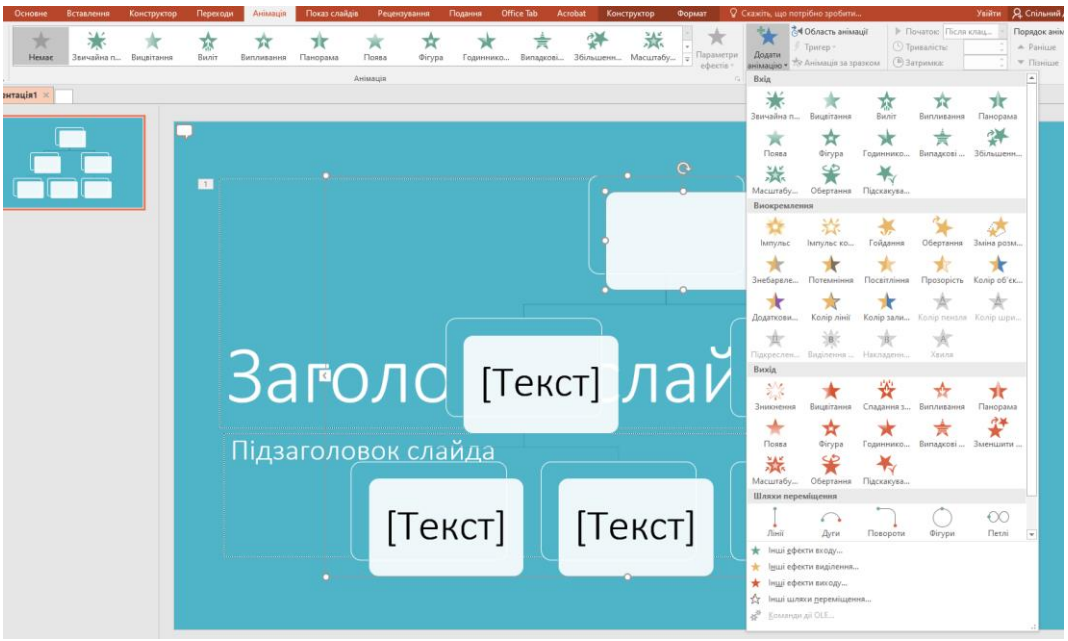


Рисунок 22 – Завдання параметрів анімації елементів слайду Microsoft PowerPoint

Для різних аудиторій слухачів доповіді можуть бути створені різні покази слайдів презентації шляхом створення довільного показу, у якому визначається, які саме слайди будуть показані й у якому порядку. Крім того, видозмінювати презентацію для різних слухачів можна шляхом приховування деяких слайдів, що при цьому не будуть вилучені з презентації, а просто виключені з показу.

Для додавання мультимедійності презентації користувач може додати до слайдів звук або відео з існуючих файлів або з галереї, що входить у поставку Microsoft PowerPoint. Програвання звуку і відео здійснюється або при показі визначеного слайда, або по щиглику користувача.



Питання для самоконтролю

1. Які основні елементи інтерфейсу Microsoft PowerPoint ви знаєте?
2. Як створити нову презентацію в PowerPoint? Які є варіанти створення?
3. Як відкрити існуючу презентацію в PowerPoint?
4. Які основні режими перегляду презентації ви знаєте? Для чого використовується кожен з них?
5. Як зберегти презентацію? Які формати збереження вам відомі і для чого вони використовуються?
6. Як налаштувати параметри автозбереження?

7. Що таке тема оформлення в PowerPoint? Як її змінити або застосувати до презентації?
8. Як змінити макет слайда? Які стандартні макети вам відомі?
9. Як змінити фон слайда? Які є варіанти заливки фону?
10. Що таке зразок слайдів? Для чого він використовується і як його редагувати?
11. Як додати колонтитули до слайдів? Як їх налаштувати?
12. Як додати текстове поле на слайд?
13. Як форматувати текст у PowerPoint (шрифт, розмір, колір, вирівнювання, списки тощо)?
14. Як вставити символ або спеціальний знак у текст?
15. Для чого використовується область нотаток до слайда?
16. Які типи графічних об'єктів можна вставляти на слайди (зображення, фігури, SmartArt, діаграми)?
17. Як вставити зображення з файлу або з онлайн-джерел?
18. Як змінити розмір, перемістити та обернути графічний об'єкт?
19. Які основні інструменти форматування графічних об'єктів ви знаєте (заливка, контур, ефекти)?
20. Що таке SmartArt? Які типи SmartArt вам відомі і для чого вони використовуються?
21. Як вставити та редагувати діаграму в PowerPoint?
22. Як змінити порядок розташування об'єктів на слайді (на передній план, на задній план)?
23. Що таке вирівнювання та розподіл об'єктів на слайді? Як їх використовувати?
24. Як згрупувати кілька об'єктів в один? Для чого це потрібно?

Індивідуальне завдання
Вивчення базових можливостей середовища MS PowerPoint зі створення презентацій

Завдання 1. Розробити презентацію (8-12 слайдів), призначену для інформування населення про порядок дій при надзвичайних ситуаціях техногенного характеру.

Оберіть 2-3 найбільш актуальні для вашого регіону види НС та детально опишіть алгоритм дій населення в кожному випадку. Обов'язково включіть інформацію про попереджувальні сигнали та склад «тривожної валізки». Використовуйте MS PowerPoint для створення презентації, застосовуючи текст, зображення, списки та просте форматування для ефективної передачі інформації.

Завдання 2. Підготувати інформаційну презентацію на тему «Правила поведінки в умовах воєнного стану» (8-12 слайдів).

- Презентація повинна базуватися на наданому плані та містити актуальну інформацію.

- Під час підготовки презентації використовувати офіційні джерела інформації (ДСНС, органи державної влади, перевірені ЗМІ).
- Презентація повинна бути чіткою, структурованою, візуально привабливою та легко зрозумілою для аудиторії.
- Інформаційна презентація про правила поведінки в умовах воєнного стану. Надати чітку, структуровану та актуальну інформацію населенню про основні правила поведінки під час воєнного стану для збереження життя та здоров'я.

Тести для самоконтролю

1. Що таке форматування тексту?
   a) внесення змін у зовнішній вигляд тексту: встановлення шрифту, кольору, накреслення символів, вирівнювання абзаців, відступів абзаців тощо;
   b) збереження текстового документа у файлі, відкриття текстового файлу в текстовому процесорі та інше;
   c) розміщення в документі графічних зображень, таблиць, діаграм тощо;
   d) введення тексту з використанням клавіатури.
2. Організаційні діаграми використовують для:
   a) схематичного подання пов'язаних між собою даних;
   b) налаштування параметрів форматування даних;
   c) зміни розташування зображення відносно тексту.
3. Який об'єкт у середовищі текстового процесора має такі параметри форматування, відступ, вирівнювання, інтервал?
   a) сторінка;
   b) абзац;
   c) символ.
4. Файли шаблонів створені засобами Microsoft Word можуть мати розширення:
   a) ppt, pot;
   b) doc, docx;
   c) dot, dotx.
5. Рисунок, фотографія, схема, організаційна діаграма в текстовому документі це:
   a) текстові об'єкти;
   b) графічні об'єкти.
6. Редагування тексту це:
   a) внесення змін у текст;
   b) внесення змін у зовнішній вигляд тексту;
   c) збереження тексту на диску у вигляді текстового файла.
7. Які основні функції систем опрацювання текстів?
   a) створення презентацій;
   b) форматування та збереження тексту;
   c) уведення та редагування тексту;
   d) створення власних зображень.
8. Списки**,** яких типів можна створювати в текстовому документі Word?
   a) маркований;

b) нумерований;
c) багаторівневий;
d) безрівневий.

9. Курсор – це:
a) клавіша на клавіатурі;
b) пристрій введення текстової інформації;
c) мітка на екрані монітора, яка вказує позицію, в якій буде відображений символ, введений з клавіатури;
d) найменший елемент відображення на екрані.

10. Які дії дає можливість виконувати текстовий процесор Microsoft Word?
a) будувати графіки функцій за наведеною формулою ;
b) редагувати та форматувати текст;
c) вводити текст з клавіатури;
d) додавати до текстового документа графічні зображення.

11. Виберіть операції, які можна виконувати над текстовим документом у текстовому процесорі:
a) збереження документа у файлі;
b) редагування тексту;
c) уведення тексту з клавіатури;
d) вставка рисунків, формул, таблиць.

12. Програма призначена для створення й опрацювання текстових документів:
a) текстовий процесор;
b) текстовий документ;
c) текстовий редактор.

13. Яка вкладка в Microsoft PowerPoint містить основні інструменти для вставлення нових слайдів?
a) файл;
b) головна;
c) вставлення;
d) показ слайдів.

14. Який режим перегляду в PowerPoint найкраще підходить для редагування окремих слайдів?
a) звичайний;
b) сортувальник слайдів;
c) показ слайдів;
d) режим нотаток.

15. Що таке «тема» в Power Point?
a) набір анімаційних ефектів для слайдів;
b) готовий набір кольорів, шрифтів та ефектів оформлення для презентації;
c) спосіб переходу між слайдами;
d) область для введення нотаток доповідача.

16. Який тип об'єкта НЕ можна безпосередньо вставити на слайд PowerPoint через вкладку «Вставлення»?

a) зображення;
b) фігуру;
c) текстовий документ Word;
d) діаграму.

17. Який інструмент використовується для створення графічних організаційних схем та діаграм у PowerPoint?
a) WordArt; б) SmartArt; в) діаграма; г) фігури.

18. Як змінити фон окремого слайда, не впливаючи на інші слайди з тією ж темою?
a) змінити фон зразка слайдів;
b) вибрати слайд і змінити стиль фону на вкладці «дизайн»;
c) вибрати слайд, клацнути правою кнопкою миші, обрати «формат фону»;
d) вставити велике зображення як фон.

19. Яка вкладка містить інструменти для додавання анімації до об'єктів на слайді?
a) переходи;
b) вставлення;
c) анімації;
d) показ слайдів.

20. Який формат файлу Power Point використовується для демонстрації презентації без можливості редагування за замовчуванням?
a) .pptx; б) .potx; в) .ppsx; г) .pptm.

РОЗДІЛ 3. ОСНОВИ РОБОТИ З ЕЛЕКТРОНИМИ ТАБЛИЦЯМИ MICROSOFT EXCEL

Мета: Опанувати основні навички роботи з електронними таблицями Microsoft Excel, розглянути основні принципи аналізу даних засобами використання логічних функцій MS Excel.

План

1. Організація введення даних. Формати даних Microsoft Excel.
2. Форматування вмісту комірок Microsoft Excel. Операції друку.
3. Робота з графікою.
4. Побудова формул.

Ключові терміни: *Microsoft Excel, електронні таблиці, введення даних, формати даних, форматування комірок, друк, великі аркуші, навігація, графіка, зображення, фігури, формули, функції, посилання на комірки.*

Microsoft Office Excel – офісна програма, розроблена для організації, форматування та обчислення числових даних. Microsoft Excel відображає дані в строчно-стовбчиковому форматі, з розділовими лініями між рядами, аналогічно бухгалтерським книгам. Excel добре пристосований для роботи з чисельними даними в ході фінансових розрахунків, наукових досліджень, статистичних підрахунків і інших ситуацій у процесі яких виникає необхідність організації даних у табличному форматі.

3.1 Організація введення даних

Документом у Microsoft Excel є робоча книга. Кожна робоча книга може містити набір інформації відносно визначеного об'єкта, яку можна розділити на окремі робочі аркуші. Робочий лист це таблиця з рядків і стовпців, у які користувач уводить дані для наступної роботи.

Зовнішній вигляд додатка Microsoft Excel і опис його основних елементів див. рис. 23 і табл. 7 відповідно.

Для позначення рядів таблиці робочої книги використовуються цифри, для позначення стовпчиків – букви англійського алфавіту. Визначення обраної користувачем комірки здійснюється за її адресою (ім'ям комірки), що визначається номером рядка і буквою стовпця, на перетині яких комірка знаходиться. Аналогічно зміні даних в одній комірці, Microsoft Excel дозволяє працювати з діапазоном комірок. Діапазон звичайно ідентифікується описом його першого та останнього комірців, з колонками між ними.

Розміри комірок можна змінювати в залежності від даних, що утримуються в них, за допомогою покажчика миші або шляхом уведення конкретного розміру в діалоговому вікні Microsoft Excel.

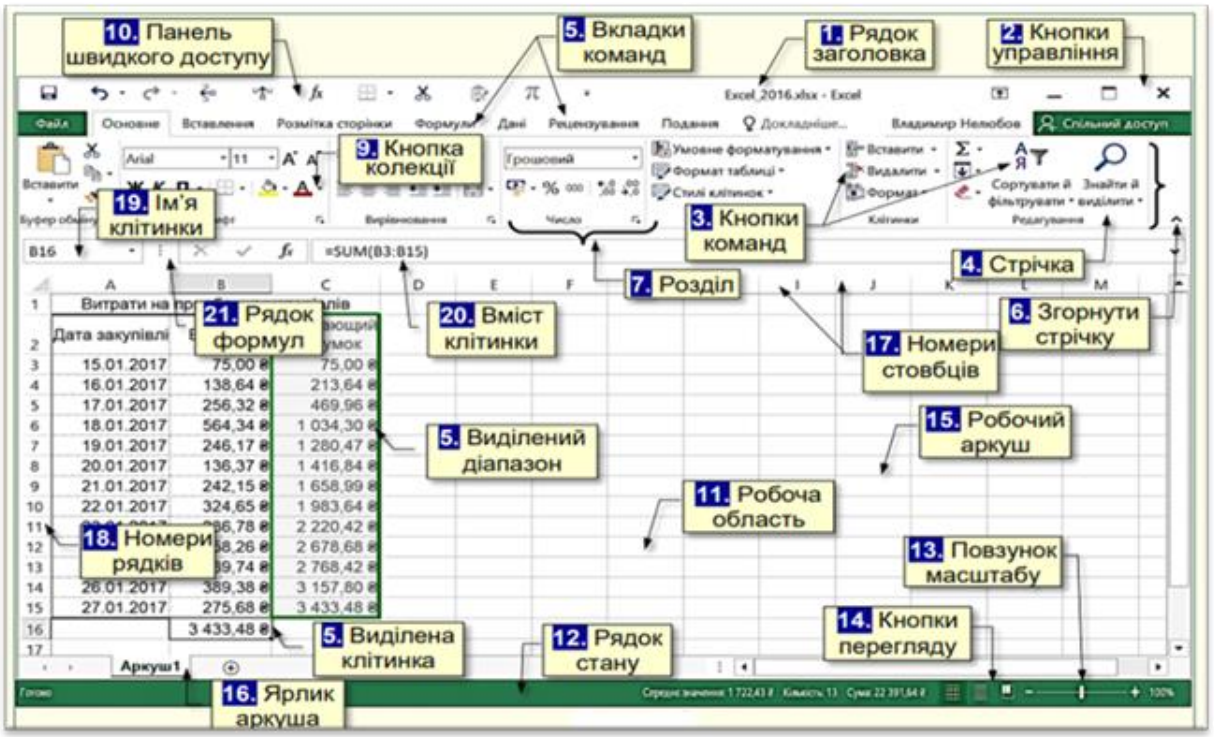


Рисунок 23 – Інтерфейс вікна Microsoft Excel 2016

Таблиця 7 – Опис основних елементів додатка Microsoft Excel

Елемент додатка	**Опис**
Рядок заголовка	Область вікна, що містить ім'я даного додатка (Microsoft Excel) і ім'я поточного документа. Розташована у верхній частині вікна.
Панель меню	Область, що містить список імен меню, що доступні у Excel. Меню це набір різних команд, за допомогою яких здійснюється робота з документом. Розташовано панель меню під панеллю заголовка.
Панель інструментів	Панель інструментів дозволяє здійснити швидкий доступ до популярних функцій-команд меню. Панель інструментів Стандартна містить основні команди роботи з робочим листом і його даними і за замовчуванням розташована під панеллю меню.
Панель інструментів Форматування	Дозволяє здійснювати швидкий доступ до функцій форматування, що найчастіше використовуються.
Ім'я комірки	Відображає адресу активної комірки.
Рядок формули	Відображає вміст активного комірця.
Смуги прокручування	Смуги призначені для перегляду виду робочого листа. Вертикальна і горизонтальна смуги прокручування розташовані в правій і нижній частині основного вікна відповідно.
Кнопки навігації	Розташовуються на горизонтальній полосі прокручування і призначені для перегляду великих документів.
Кнопки мінімізації, максимізації/ відновлення і закриття	Призначені для згортання додатка Microsoft Excel на панель інструментів Windows, розгортання додатка і його закриття.
Підказка	Елемент допомоги, що відображає ім'я кнопки або елементу екрана, що спливає при зупинці покажчика миші на ньому.

Панель задач	У Microsoft Excel команди організовані по групах задач, що відображені на панелі задач. Зміст панелі задач варіюється в залежності від потреб користувача. Первісна панель задач служить для створення порожньої робочої книги/книги з шаблону або вибору книги, яку необхідно відкрити для редагування. Розташовується в правій частині основного вікна документа.
Введіть питання	Текстове вікно в правому кінці панелі меню дозволяє отримати швидкий доступ до довідки Microsoft Excel
Вибрати всі	Вибір всіх комірців робочого листа
Активна комірка	Виділена користувачем для введення і редагування даних комірка, відзначена чорною рамкою

3.2 Формати даних Microsoft Excel

Microsoft Excel передбачає введення даних 3 типів: текст, число і формула.

Таблиця 8 – Формати даних Microsoft Excel

Категорія	**Опис**	**Введення**	**Відображення**
Основний	Відображає дані в тому виді, у якому вони вводяться	1234	1234
Числовий	Відображає число з двома знаками після коми за замовчуванням.	1234	1234.00
Грошовий	Відображає число зі знаком валюти, що прийнятий у даному регіоні з двома знаками після коми і роздільником між розрядами.	1234	$1,234.00
Фінансовий	Відображає грошовий тип з вирівнюванням по групах розрядів	1234 12	$1,234.00 $12.00
Дата	Відображає день, місяць і рік у різних форматах	18.03.2005	May 18, 2005, 18-May, 18/3/2005
Час	Відображає години, хвилини і секунди в різних форматах	12.34	12:34 AM 12:34 12:34:00
Процентний	Відображає число помноженим на 100 і зі знаком відсотка	12	1200.00%
Дробовий	Число відображається як відношення до будь-якої основи	12.34	12 1/3
Експонентний	Відображає число у виді експонентного запису	1234	1.23E+03
Текстовий	Дані відображаються в тому виді, у якому були введені, навіть якщо це число	1234	1234
Спеціальний	Призначений для відображення спеціальних номерів, ідентифікаційних кодів, номерів соціального страхування	12345 123-555-1234	12345 123-555-1234

3.3 Форматування вмісту комірок Microsoft Excel

Після введення даних в комірку її зовнішній вигляд можна відформатувати у спеціальному діалоговому вікні (див. рис. 24), задавши вирівнювання й орієнтацію тексту, його відображення і напрямок. Крім того, Microsoft Excel дозволяє задати формат шрифту, границі і кольори комірців.

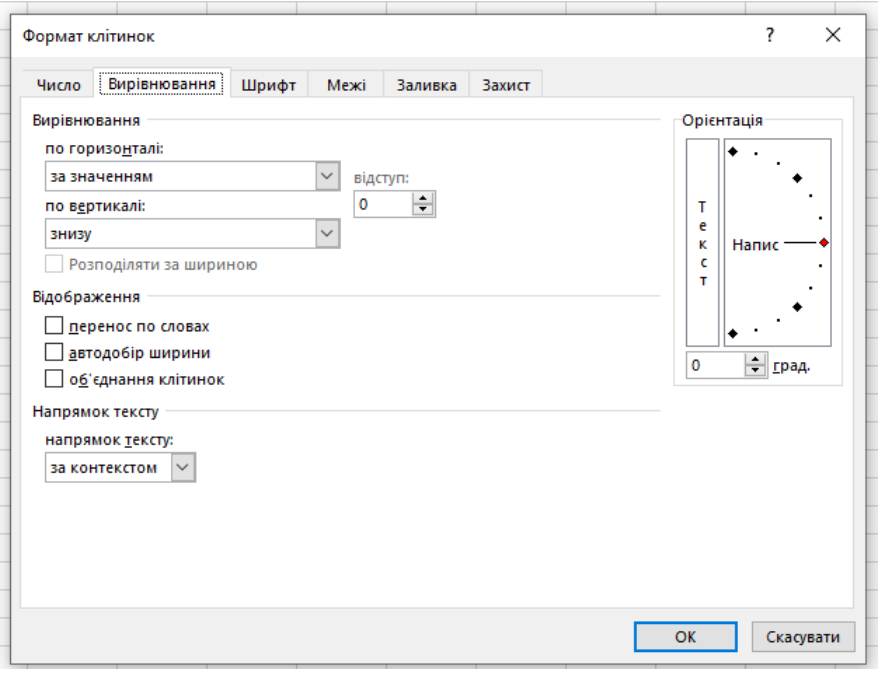


Рисунок 24 – Завдання формату осередків Microsoft Excel

Для виділення важливої інформації в робочих аркушах найчастіше використовуються границі комірок. Microsoft Excel підтримує не тільки різні стилі ліній границь, але і дозволяє додати границю в кожної з чотирьох сторін комірця або діапазону. Заливання кольором також має розширені можливості – крім вибору кольору, користувач може визначити тип замальовки або візерунок заливання комірки.

До групи комірок може бути застосована опція Автоформат, що дозволяє виділити таблицю по заздалегідь обраному шаблону з виділенням крайніх рядків і стовпців.

При частому повторенні форматованих характеристик комірців рекомендується створити стиль, що дозволяє легко застосовувати числовий формат, вирівнювання, шрифт, границю, заливання і захист комірки по заздалегідь визначеному зразку.

Пошук вмісту комірок здійснюється за допомогою спеціальних функцій Знайти і Замінити, що дозволяють не тільки перейти до даних визначеного змісту, але і з заданим форматом (див. рис. 25).

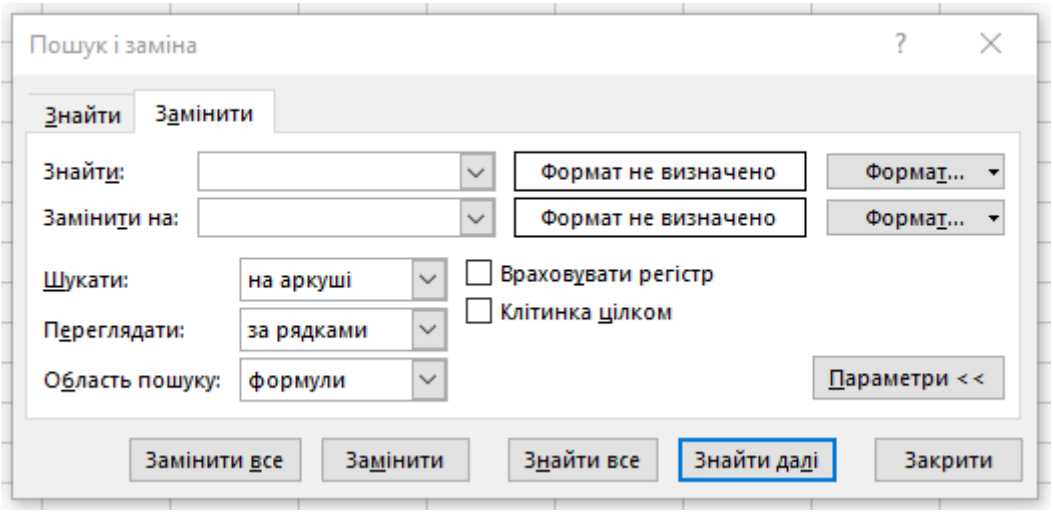


Рисунок 25 – Пошук і заміна інформації в комірках Microsoft Excel

При копіюванні даних між комірками різних форматів Microsoft Excel дозволяє користувачеві вибирати між форматом джерела і приймача, використовуючи одну з наступних функцій, відображених у таблиці 9.

Таблиця 9 – Функції вибору формату копіювання

Функція	**Опис**
Зберегти формати оригіналу	Вміст комірки-джерела поміщується в комірку-приймач із застосуванням формату оригінальних комірок.
Використовувати формати кінцевих комірок	Вміст комірки-джерела поміщується в комірку-приймач із застосуванням формату кінцевих комірок.
Значення і формати чисел	В комірку-приймач поміщується значення з комірки-джерела без заміни форматування.
Зберегти ширину стовпців оригіналу	При копіюванні даних ширина стовпців приймача стає рівній ширині стовпців джерела.
Тільки формати	Дані комірки-приймача форматуються аналогічно даним комірки-джерела.
Посилання на комірці	В комірку-приймач копіюються значення джерела, з підтримкою посилання на оригінальні комірки.

3.4 Операції друку

При створенні документа найчастіше необхідно виводити на кожній його сторінці яку-небудь загальну інформацію – ім'я автора, назву документа, дату створення, номер сторінки тощо.

Колонтитули – верхній і нижній заголовки – рядок заздалегідь визначеного тексту, виведеного при друці на кожній сторінці робочого листа.

Параметри сторінок виведених на друк з даної робочої книги задаються спеціальним діалоговим вікном, що визначає ширину полів, розміри колонтитулів, розташування інформації на сторінці, масштаб даних, що друкуються.

При необхідності користувач може задати область друку – діапазон робочого листа, що буде подаватися на принтер.

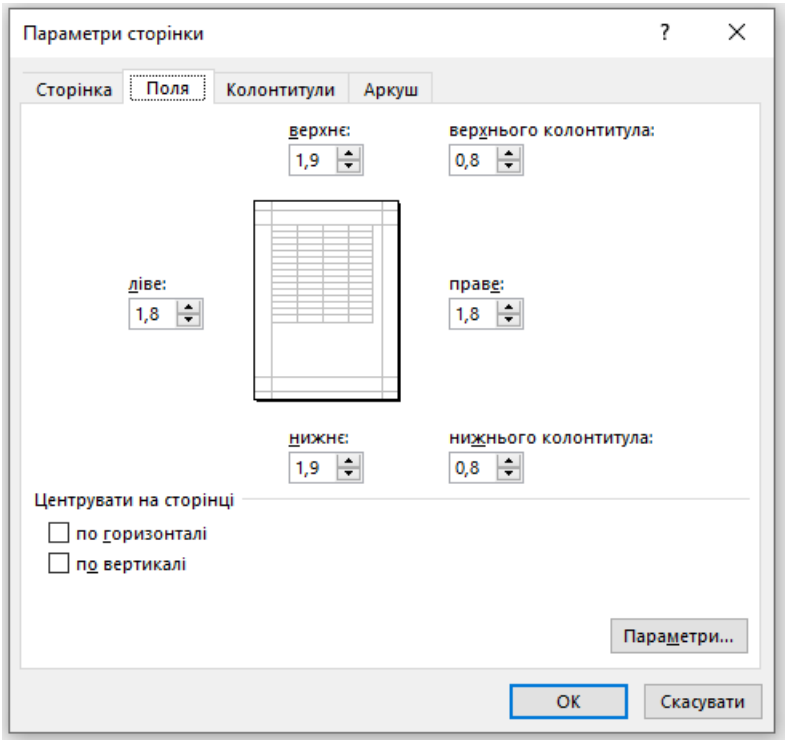


Рисунок 26 – Діалогове вікно встановлення параметрів сторінки

3.5 Робота з великими робочими аркушами

При роботі з даними великих діапазонів, що займають розмір більш одного екранного листа, Microsoft Excel надає можливість утримання і приховання рядів і колонок.

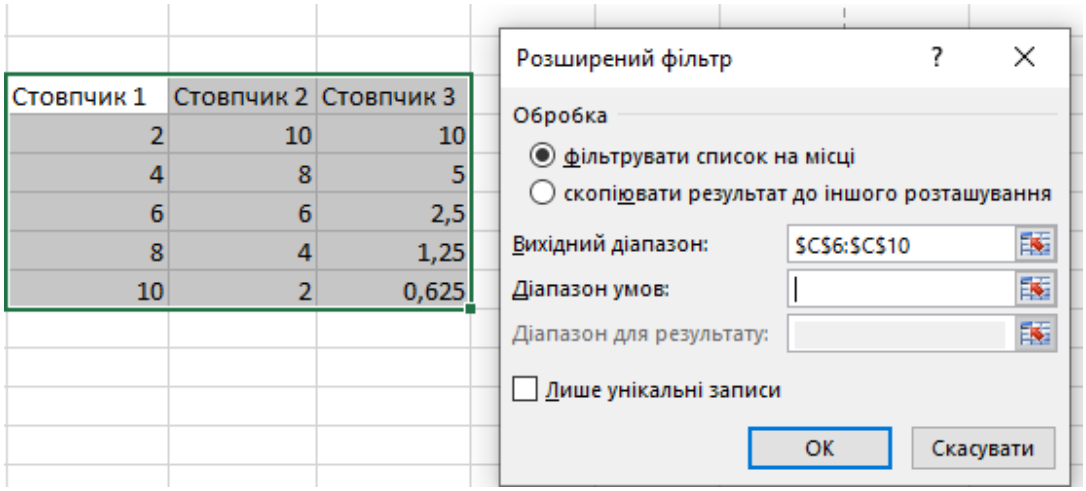


Рисунок 27 – Установка параметрів сортування даних в комірках Microsoft Excel

Приховання стовпчиків тимчасово ховає для перегляду визначені користувачем горизонтальні або вертикальні діапазони комірок, при цьому дані з них не зникають, а доступні для використання в будь-який момент. Утримання області даних дозволяє заморозити одну з частин екрана (найчастіше заголовок рядів даних), забороняє приховання з екранної області

утриманих комірців. Порядок, у якому користувач уводить дані, не завжди зручний для логічної інтерпретації й аналізу. Сортування даних комірок значно прискорює процес визначення залежностей рядів даних. Microsoft Excel дозволяє сортувати до 3 рядів комірок з різними параметрами. Відображення даних, що задовольняють фільтрові — правилу або наборові правил, дозволяє тимчасово сховати від користувача частину діапазону даних. Функція Автофільтр дозволяє задавати параметри фільтрації даних у Microsoft Excel.

3.6 Робота з графікою

Microsoft Excel дозволяє працювати з діапазонами даних, що найчастіше мають потребу в графічній інтерпретації. За допомогою діаграм Excel користувач може підсумовувати, підсвітлювати, визначати лінії залежностей і трендів значень, не очевидних при перегляді числових рядів даних.

Майстер діаграм дозволяє користувачеві задати параметри графіків, дотримуючись наступних кроків: тип графіка, джерело даних, настроювання, розміщення діаграми. Microsoft Excel підтримує 14 видів діаграм, при цьому в кожному з них доступні кілька видів відображення даних.

Таблиця 10 – Види діаграм Microsoft Excel

Відображення графіка	**Назва графіка**	**Функція діаграми**
	Гістограма	Відображення окремих значень за часом або іншими категоріями у виді вертикальних ліній
	Лінійчата	Відображення окремих значень за часом або іншими категоріями у виді горизонтальних ліній
	Графік	Відображення зміни значень за часом або іншими категоріями у виді крапок,що з'єднані лінією
	Кругова	Відображення значень як частин цілого у виді секцій кола
	Точкова	Порівняння значень двох наборів даних за часом або іншими категоріями. Відображення у виді точок даних, що можуть бути з'єднані лініями
	З ділянками	Відображення залежності значень за часом або іншими категоріями у виді залитих ділянок
	Кільцева	Відображення значень як частин цілого у виді секцій кільця
	Пелюсткова	Відображення залежності даних щодо центрального значення у виді точок, радіально розбіжних від центра. Кожна категорія має власну вісь, значення однієї серії з'єднані лініями.
	Поверхня	Відображення залежності значень із двох наборів даних у виді тривимірної поверхні.
	Бульбашкова	Порівняння трьох наборів даних
	Біржова	Відображення залежності значень за часом, що використовується для відображення зміни біржових цін з маркерами вищої, нижчої ціни і цінами на момент відкриття і закриття біржі. Дані

		відображаються у виді точок, ліній або стовпчиків.
	Циліндрична	Порівняння окремих значень набору даних за часом або іншими категоріями у виді вертикальних або горизонтальних циліндрів
	Конічна	Порівняння окремих значень набору даних за часом або іншими категоріями у виді вертикальних або горизонтальних конусів
	Пірамідальна	Порівняння окремих значень набору даних за часом або іншими категоріями у виді вертикальних або горизонтальних пірамід

При розміщенні діаграми існує можливість створення приєднаних діаграм на поточному робочому листі або винесення діаграми на окремий лист.

У процесі роботи з графіком користувач може змінювати вид діаграми і джерело даних, модифікувати осі значень, проводити лінії тренда відповідно до набору існуючих залежностей, змінювати вид частин діаграми.

3.7 Побудова формул

У Microsoft Excel користувач може робити обчислення з даними комірців робочого листа. Крім основних операцій існує можливість проведення складних обчислень з використанням функцій шляхом побудови формул.

Формула – вираження, що обчислюється та виконується в Excel. При уведенні формули в комірець вона знаходиться в ньому до моменту видачі результату. Після цього переглянути формулу комірця можна в рядку формул, а комірець буде відображати вже обчислене значення. Формула складається з двох елементів: операндів, що ідентифікують значення, яке використовується в обчисленні і математичних операторів. Операндом може виступати число, посилання на комірку або діапазон, інша формула. У формулах можуть бути використані оператори зведення в ступінь (^), множення (*), розподілу (/), суми(+), вирахування(-). При цьому читання формул, що містять два і більш оператори здійснюється зліва направо з урахуванням пріоритету операцій відповідно до правил математики.

При копіюванні формул, що містять посилання на комірки або діапазони, варто враховувати тип адресації, що використовується у Microsoft Excel. Усього існує три типи адресації в комірці: відносна, абсолютна і змішана. Відносна адресація прийнята за замовчуванням і при зсуві комірця приводить до зсуву його адреси, тобто при копіюванні формул дані будуть автоматично зміщені щодо вихідного місця розташування формули й комірки з використаними в ній даними. Абсолютна адресація позначається знаком $ і при зсуві комірки її адресація не змінюється, тобто дані при копіюванні формул будуть братися саме з зазначеного комірця. Змішана адресація дозволяє задати зсув адреси комірця тільки в горизонтальному або вертикальному порядку.

Найчастіше можливостей, наданих формулою, недостатньо для складних обчислень, і виникає необхідність використовувати функції. Функції розроблені для здійснення всіх типів обчислень і можуть працювати з групою комірок.

Функція складається з двох елементів – імені функції і списку аргументів. Список аргументів, обмежений круглими дужками, містить дані (операнди), що використовуються функцією в ході обчислень. Уведення функцій здійснюється безпосередньо в комірку або з використанням діалогового вікна побудови функцій, що дозволяє вибрати функцію з існуючого списку і задати необхідні аргументи. Усі функції Microsoft Excel для зручності користувача розділені на 9 груп (див. табл. 11).

Таблиця 11 – Функції Microsoft Excel

Група функцій	**Призначення**
Фінансові	Функції здійснення біржових, бухгалтерських і фінансових обчислень
Дата і час	Функції обробки даних, представлених у часовому форматі
Математичні	Інженерні, арифметичні і тригонометричні функції
Статистичні	Обчислення основних функцій, що використовуються у математичній статистиці і теорії імовірностей
Посилання і масиви	Функції роботи з діапазонами даних і покажчиками на комірці
Робота з базою даних	Функції робочого листа, що використовуються для аналізу даних зі списків або баз даних
Текстові	Функції для роботи з текстом, пошук і підстановка в текстових рядках
Логічні	Основні операції реляційної алгебри
Перевірка властивостей і значень	Функції визначення істинності виражень

Часто при роботі з робочими книгами необхідно використовувати дані, що знаходяться на різних робочих аркушах. У цьому випадку застосовується тривимірна формула, що містить посилання з указівкою робочих аркушів з необхідними значеннями.

Використання Microsoft Excel в Інтернет

За допомогою Microsoft Excel користувач може зробити дані доступними різним користувачам в Інтернет. Крім того, існує можливість відправлення робочих книг по електронній пошті, збереження файлів Excel у виді Web-сторінок і розширення можливостей робочих аркушів за допомогою гіперпосилань на інші аркуші, книги або реферативні матеріали Інтернет.

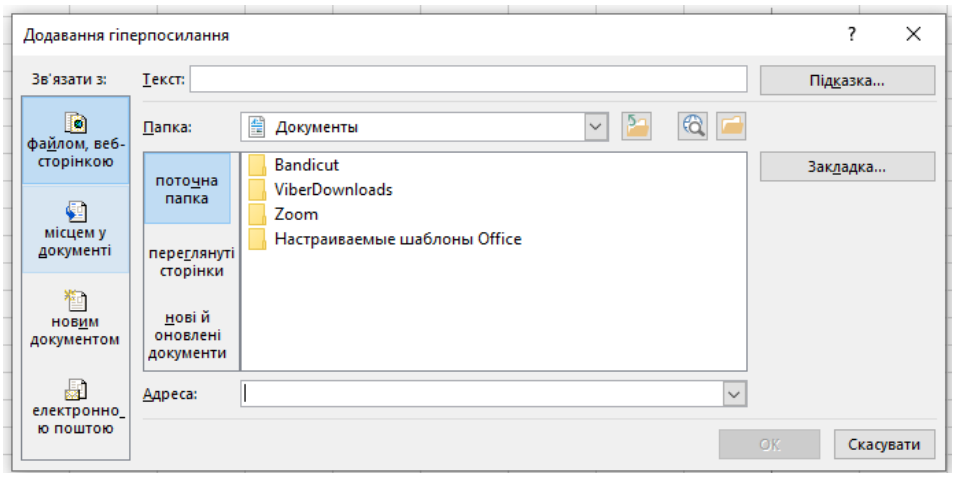


Рисунок 28 – Діалогове вікно роботи з посиланнями в Microsoft Excel



Питання для самоконтролю

1. Для чого призначена програма Excel?
2. Як називається файл в програмі Excel?
3. Що таке Адреса осередку?
4. У якому форматі можуть бути внесені дані в таблицю? У якій вкладці може бути виставлений потрібний формат даних?
5. Як змінити параметри шрифту? Що таке вирівнювання в комірці і як воно виставляється?
5. Який порядок проведення розрахунків в програмі Excel?
6. Для чого призначений рядок формул?
7. Що таке вирівнювання в осередку і як воно виставляється?
8. Який порядок проведення розрахунків в програмі Excel?
9. Для чого призначений рядок формул?
10. Що слід зробити для побудови діаграми?
11. Які набори інструментів призначені для зміни діаграм?
12. Що таке сортування?
23. Порядок виконання сортування?
14. Для чого використовуються фільтри?
15. Порядок виконання авто фільтрації?
16. Як створити зведену таблицю?

Індивідуальні завдання

I. Використання основних інструментів обчислень MS Excel

Введення чисел

Числа, які вводять в комірку, інтерпретуються як константи. У Microsoft Excel число може складатися тільки з наступних символів: 0 1 2 3 4 5 6 7 8 9 + - ( ) / $ % . Необхідні перед числом знаки плюс (+) ігноруються, а кома інтерпретується як роздільник десяткових розрядів. Всі інші з'єднання клавіш, які складаються з цифр і нецифрових символів, розглядаються як текст. Символи, які розглядаються як числа, залежать від встановлених параметрів в компоненті *Мова* і *стандарти панелі управління*.

Перед раціональним дробом варто вводити 0 (нуль), щоб уникнути її інтерпретації як формату дати; наприклад, вводите 01/2. Перед негативним числом необхідно вводити знак мінус (-) або узяти його в круглі дужки ( ).

Введені числа вирівнюються в комірці по правому краю. Щоб змінити вирівнювання, виділите комірці, виберіть ***Головна / Вирівнювання*** *і* встановіть необхідні параметри. Формат числа, призначений комірці, визначає спосіб перегляду числа на листі. Якщо число вводиться в комірку, яка має загальний числовий формат, то може бути використаний інший формат. Наприклад, якщо вводиться 14,73р., то буде використаний грошовий формат. Щоб змінити числовий формат, виділіть комірки, які містять числа, виберіть ***Головна / Число*** виберіть необхідний формат.

Для інтерпретації чисел, наприклад, інвентаризаційних номерів, як тексту, необхідно призначити текстовий формат незаповненим коміркам. Тільки після цього варто вводити числа. Якщо числа вже введені, то призначте коміркам текстовий формат і, виділяючи кожну комірку, натискайте клавішу **F2**, а потім – клавішу **ENTER**, щоб ввести наново дані.

Введення тексту

У Microsoft Excel текстом є будь-яка послідовність, яка складається з цифр, пропусків і нецифрових символів, інтерпретуються як текст. Введений текст вирівнюється в комірці по лівому краю. Щоб змінити вирівнювання, виберіть ***Головна / Вирівнювання*** і встановите необхідні параметри. Щоб переглянути текст, який займає в комірці декілька рядків, виберіть ***Перенесення тексту*** на стрічці ***Головна / Вирівнювання***. Щоб почати в комірці новий рядок, натисніть клавіші **ALT+ENTER**.

Установка меж комірки

1. Виділите комірку, де необхідно додати рамку.
2. Щоб встановити рамку на стрічці ***Головна*** вибрати ***Комірки / Формат*** виберіть команду ***Комірки***, а потім перейдіть до вкладки ***Границя***. Виберіть бажаний тип лінії. Для зміни типу лінії вже існуючої рамки виділите комірки, на яких рамка відображена. На вкладці ***Границя*** (діалогове вікно ***Комірці***) в полі ***Тип лінії*** виберіть необхідний тип, а потім вкажіть границю, яку необхідно змінити.

Відносні посилання

До цих пір ми використовували відносну адресацію комірок. При авто заповненні в кожному наступному записі у формулі змінювалися імена комірок. Такі імена комірок або, точніше сказати, посилання на комірки називаються відносними. У цьому полягає основне правило при роботі з відносними адресами. При зміні позиції комірки, що містить формулу, змінюється і посилання. При копіюванні формули уздовж рядків і уздовж стовпців посилання автоматично коректується. За умовчанням в нових формулах використовуються відносні посилання.

Абсолютні посилання

Якщо виникла необхідність вказати у формулі комірку, який не можна міняти при автозаповненні, використовується знак **$**. Їм фіксуються як стовпці, так і рядки. Наприклад: **A10.** Абсолютне посилання комірки у формулі завжди посилається на комірку, розташовану у визначеному місці. При зміні позиції комірки, що містить формулу, абсолютне посилання не змінюється. При копіюванні формули уздовж рядків і уздовж стовпців абсолютне посилання не коректується.

Змішані посилання

Змішане посилання містить або абсолютний стовпець і відносний рядок, або абсолютний рядок і відносний стовпець. Абсолютне посилання стовпців набуває вигляду $A1, $B1 і т.д. Абсолютне посилання рядка набуває вигляду A$1, B$1 і т.д. При зміні позиції комірки, що містить формулу, відносне посилання змінюється, а абсолютне посилання не змінюється. При копіюванні

формули уздовж рядків і уздовж стовпців відносне посилання автоматично коректується, а абсолютне посилання не коректується.



Завдання для самостійного виконання

Завдання 1.

Підготувати електронний журнал групи, що включає відомості по одному предмету. Передбачити 5 оцінок в семестрі, середні бали за 2 семестри і за рік для 10 студентів, середній бал групи за семестри і за рік (див. рис), при цьому: розрахунок середніх балів за семестр і за рік здійснювати строго по формулах! за допомогою функції =СРЗНАЧ() =ОКРУГЛ(=СРЗНАЧ();0) введення оцінок здійснювати різними способами: шляхом безпосереднього введення чисел в комірку.

Завдання 2.

Створіть в Excel таблицю обліку витрат на обладнання та матеріали з охорони праці. Заповніть комірки формулами для обчислення загальної вартості кожної позиції. Продемонструйте використання різних типів посилань при автозаповненні. Для статей витрат, сума яких перевищує 500 грн, застосуйте автоматичне коригування бюджету (наприклад, зменшення на 1%), використовуючи функцію «ЕСЛИ».

Завдання 3.

На окремому листі електронного журналу групи оформити таблицю, що містить дані про студентів групи «Відомості про студентів». Таблиця повинна містити: номер, прізвище, ім'я, дату народження, місто народження, домашній телефон (встановите формат комірок – текстовий), успішність. При створенні таблиці повинні бути виконані наступні вимоги:

1. Ввести в таблицю дані, при цьому імена і місто народження з використанням функції автозавершення значення комірок, для чого перед заповненням таблиці виконати: **Файл → Параметри → Додатково**. При введенні тексту в комірку електронна таблиця Excel спочатку перевірятиме вміст інших комірок даного стовпця. Якщо буде виявлений текст, перші символи якого співпадають з введеними символами, то введення даних закінчується автоматично. Можна проігнорувати пропозицію Excel, якщо продовжити введення. Для підтвердження треба натиснути **ENTER**.

2. Для отримання формули в стовпці «успішність» необхідно використовувати логічні функції **ЯКЩО, І**. При додаванні функції видається інформація по її використанню, ознайомтеся з правилами використання цих функцій. Формула буде аналогічна наступною:

=ЕСЛИ(І(матем!B3=5;био!B3=5;информ!B3=5); «відмінник»;
ЕСЛИ(І(матем!B3>=4;био!B3>=4;иформ!B3>=4); «хорошист»;
ЕСЛИ(І(матем!B3>=3;био!B3>=3;информ!B3>=3); «трієчник»; «двоєчник»))),
де матем, біо, інформ – імена листів.

Для заповнення формули краще всього використовувати вибір значень, що підставляються, за допомогою кнопки миші. Тобто записали формулу, а замість імен комірок, на які йде посилання, вставляєте комірку шляхом натиснення на ній кнопкою миші.

Завдання 4. Сортування списку

Після введення даних потрібно упорядкувати їх. Процес впорядковування записів в базі даних називається сортуванням. При сортуванні змінюється порядок проходження записів в базі даних або таблиці. Сортування за збільшенням припускає наступний порядок: числа, текст, логічні значення, значення помилок, порожні комірки. Сортування по убуванню відбувається в зворотному порядку. Виключенням є порожні комірки, які завжди розташовуються в кінці списку. Текстові дані упорядковуються в алфавітному порядку. Можна задати три рівні сортування одночасно за одну операцію, можна виконати сортування спочатку по першому рівню, потім в отриманому списку – по другому, а потім – по третьому рівню. Другий і третій рівень дозволяють визначити порядок вторинного сортування для записів, в яких є співпадаючі значення. При використанні функцій списку, виділення області списку відбувається автоматично. Проте користувач може заздалегідь виділити діапазон комірок із записами, що підлягають сортуванню.

Виконати різні види сортувань списку, відкривши діалогове вікно, сортування діапазону: Дані/ Сортування і фільтр. Зробити сортування трьом рівням: успішність, дата народження, прізвище. Познайомитися з видами установок різних параметрів в діалоговому вікні «параметри сортування».

II. Графічний аналіз статистичних даних засобами MS Excel

Побудова графіків і діаграм

Для побудови діаграми потрібно виконати наступні дії:

1. Виділити діапазон даних, які будуть відображені на діаграмі.

2. На вкладці Вставка і в групі Діаграми вибрати тип створюваної діаграми (Гістограма, Круг, Лінійчата і ін.). При вставці діаграма переходить в режим редагування і стає активна контекстна вкладка Робота з діаграмами.

Робота з діаграмами

Вкладка має три набори інструментів для зміни діаграм:

Конструктор – містить параметри, що визначають тип діаграми, джерело даних і їх впорядковування, макети діаграм, а також команду Перемістити діаграму.

Макет – дозволяє указувати властивості діаграм, додавати або редагувати елементи діаграми (підписи даних і осей, додавання легенди і елементів малювання), а також вибирати параметри, пов'язані з тривимірними графіками.

Формат – містить опції вибору різних елементів діаграми, привласнення стилів її графічним елементам, включаючи тривимірні краї, тіні, фаски і ін.



Завдання для самостійного виконання

Завдання 1.

1. **Аналіз витрат на засоби індивідуального захисту.**

На основі наданої таблиці з даними про вартість різних видів засобів індивідуального захисту (ЗІЗ) для працівників підприємства:

- Побудуйте **кругову діаграму**, яка відображає частку витрат на кожен вид ЗІЗ у загальному бюджеті.
- Побудуйте **гістограму**, яка показує розподіл кількості придбаних одиниць кожного виду ЗІЗ.
- Проаналізуйте отримані діаграми та зробіть висновки щодо структури витрат на ЗІЗ та кількості придбаних одиниць різних видів.

2. **Розподіл нещасних випадків за причинами**

На основі статистики нещасних випадків на виробництві за минулий рік, де вказано основні причини кожного випадку та їх кількість:

- Створіть **кругову діаграму**, що відображає відсоткову частку кожної причини у загальній кількості нещасних випадків.
- Побудуйте **гістограму**, яка показує кількість нещасних випадків для кожної категорії причин.
- Інтерпретуйте отримані візуалізації для визначення найбільш поширених причин травматизму на підприємстві.

3. **Аналіз порушень правил охорони праці**

Маючи дані про виявлені порушення правил охорони праці на різних ділянках підприємства та кількість зафіксованих порушень кожного типу:

- Зобразіть розподіл типів порушень у вигляді **кругової діаграми**, показавши їхню питому вагу.
- Створіть **гістограму**, яка ілюструє кількість виявлених порушень для кожного типу.
- Надайте аналіз діаграм з метою визначення найбільш проблемних аспектів дотримання правил охорони праці.

При створенні діаграми відкривається доступ до інструментів для роботи з діаграмою: відображаються вкладки Конструктор, Макет і Формат. Команди цих вкладок можна використовувати для зміни представлення даних на діаграмах. Наприклад, вкладка Конструктор використовується для відображення рядів даних по рядках або по стовпцях, внесення змін до початкових даних, змін розміщення діаграми, зміни типу діаграми, збереження діаграми як шаблон або вибір заздалегідь певних параметрів макету і форматування.

Вкладка Макет використовується для зміни таких елементів діаграми, як заголовки діаграм і підпису даних, використання інструментів малювання, а також додавання до діаграми текстових полів і рисунків.

Вкладка Формат дозволяє додавати заливку кольором, змінювати тип ліній або використовувати спеціальні ефекти:

– перевірити правильність вказівки діапазону даних, у разі потреби виправити його, перевірити розташування рядів даних і у разі потреби змінити його;

– задати Параметри діаграми: назва, видалите легенду, підписи даних, вибираючи різні варіанти і проглядаючи результат. Остаточно вибрати опцію Категорія і частка;

– використовуючи команду Перемістити діаграму (група Розташування, вкладка Конструктор) вибрати варіант розміщення діаграми.

На цьому побудова діаграми закінчується. Побудована діаграма є упровадженим об'єктом. Його можна виділити, копіювати, перемістити, видалити, змінити розміри так само, як це виконувалося для рисунків.

Завдання 2.

Побудувати таблицю значень функції $y = \left(sin^2 4x\right)/\left(x+1\right)$ для аргументу *x*, що змінюється від 0 до 1,5 з кроком 0,1. Побудувати графік функції. (функція однієї змінної для крокового аргументу).

Хід роботи

Рішення розбивається на два основні етапи: побудова таблиці значень функції і побудова графіка функції.

Побудова таблиці. Наберемо заголовки стовпців для *x* і *y* в комірках A1, B1. Наберемо перше значення *x*, рівне 0, в комірці A2. Виконаємо команду Прогресія (вкладка Головна, група Редагування, кнопка Заповнити), задаємо в діалозі Розташування в стовпці, Арифметична прогресія, Крок 0,1, Граничне значення 2. Заповняться комірки A4:A22.

В комірку B2 введемо формулу: =SIN(4*A2)^2/(A2+1) і скопіюємо її в комірки B3:B22 . Виконаємо форматування даних (чисел) і обрамлення таблиці. Фрагмент робочого листа з таблицею показаний на рисунку.

Побудова графіка функції

Для побудови графіка виділимо діапазон даних (комірки A1:B22) і побудуємо точкову діаграму, вид якої представлений на рисунку 29.

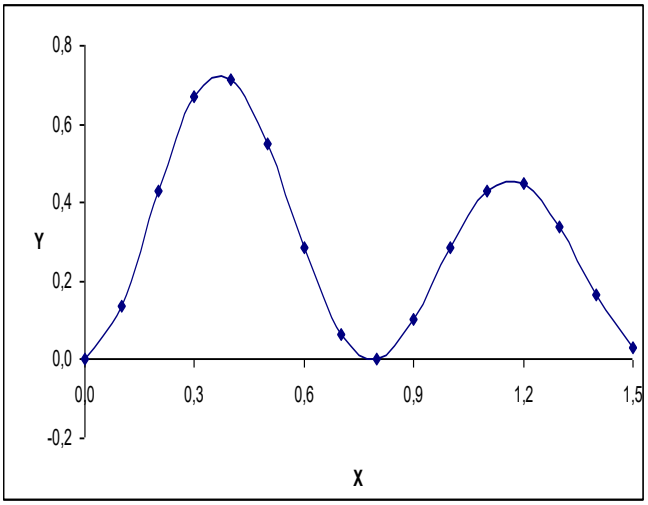


Рисунок 29 – Графік функції $y = \dfrac{sin^2 4x}{x+1}$

III. Практичне застосування методів обробки баз даних

Фільтрація даних

При роботі з великими списками, буває зручнішим відображати не весь список, а тільки його частину, звану підмножиною списку. Для того, щоб вибрати записи що задовольняють певним умовам і тимчасово приховати записи ***використовується фільтр***. Фільтр використовується вводячи умови фільтрації. Щоб встановити фільтр, потрібно натиснути на будь-який осередок списку з даними і виконати команду **Дані/(Сортування і Фільтр) Фільтр**. Після цього вид списку зміниться. У кожного стовпця з'явиться додаткова кнопка, натиснення якої приводить до відкриття переліку умов фільтрування відповідного поля. Ці умови є списком назв полий у відповідному стовпці, а також ряд загальних умов: Все, перші 10, і т.д. Застосувавши одну з умов ми змінимо вид списку. При використанні фільтру, всі записи не входять в підмножину, тобто що не задовольняють умовам фільтру ховаються. У рядку стану програми відображається загальне число записів і число записів що задовольняє умові.

При фільтрації нумерація рядків не змінюється, а номери відфільтрованих записів виділяються іншим кольором. Кнопка у назви стовпця також змінює свій колір, показуючи що вона використовується як умова для фільтру. Щоб відмінити застосування фільтру потрібно вибрати пункт Все у випадному списку. Програма дозволяє створювати досить складні фільтри, використовуючи пункт умова. Діалог, що з'являється при цьому, містить декілька випадних списків. Лівий верхній список містить логічні операції, такі як: рівно, більше, менше і т.д., так і умова для пошуку даних: що містять або не містять даний текст. Полів для операції два, і вони об'єднуються за правилом **І** або **АБО**. Праві списки містять значення умов із списку фільтрації. Після завершення форматування умови натисніть кнопку ОК. Щоб завершити роботу з автофільтром необхідно ще раз виконати команду Дані/(Сортування і Фільтр) Фільтр.



Завдання для самостійного виконання

Завдання № 1.

1. Створити таблицю, що містить зведення про надходження товарів, скопіювати її на п'ять листів Книги Microsoft Excel

2. На другому листі за допомогою автофільтру (Вкладка **Дані / Сортування і фільтр**) вибрати товари, відпущені до 8 січня. Назвати лист «Різдво».

3. На третьому листі побудувати діаграму зміни попиту на мармелад, заздалегідь скопіювати на цей лист початкові дані. На діаграмі вставити мітки значення, для цього:

– перейти в режим редагування діаграми, виділивши її;

– в області Робота з діаграмами на вкладці Конструктор вибрати макет діаграми, наприклад, для кругової діаграми можна вибрати Макет 4 , на якому як мітка використовуються значення елементів ряду.

Склад:

№	Дата	№ накладної	Найменування	Одержу-вач	Кількість	Ціна	Вартість
1	04.01.24	1	Печиво	Весна	23	2500	57500
2	04.01.24	1	Вафлі	Весна	43	3000	129000
3	04.01.24	2	Карамель	Літо	65	12500	812500
4	04.01.24	2	Мармелад	Літо	67	23000	871000
5	04.01.24	3	Шоколад	Сезам	45	3800	171000
6	05.01.24	4	Печиво	Сезам	67	2500	187500
7	05.01.24	4	Вафлі	Сезам	34	3500	119000
8	05.01.24	5	Мармелад	Весна	76	14000	1064000
9	05.01.24	5	Карамель	Весна	45	16000	720000
10	05.01.24	8	Печиво	Літо	8	3000	18000
11	06.01.24	6	Шоколад	Літо	2	4200	8400
12	06.01.24	6	Вафлі	Літо	33	4500	148500
13	06.01.24	7	Печиво	Весна	98	2500	245000
14	08.01.24	8	Мармелад	Сезам	57	12500	712500
15	08.01.24	8	Карамель	Сезам	89	12000	1068000
16	08.01.24	9	Шоколад	Весна	45	4200	189000
17	08.01.24	9	Карамель	Весна	34	23500	459000
18	09.01.24	10	Газ.вода	Літо	45	4000	180000
19	09.01.24	10	Печиво	Літо	67	3000	201000
20	09.01.24	10	Мармелад	Літо	43	23000	559000
21	09.01.24	11	Газ.вода	Весна	58	3500	196000
22	09.01.24	11	Печиво	Весна	78	3000	234000

4. На четвертому листі отримати щоденні підсумки для кожної фірми за загальною вартістю, отриманих нею товарів. Для цього:

– скопіювати дані з першого листа;

– відсортувати дані по датах (за збільшенням), а потім по фірмах (за збільшенням). На вкладці **Головна** область **Редагування**

– вибрати вкладку **Дані / Структура /**. Встановити При кожній зміні в Одержувач вибрати операцію Сума, додати підсумки по полю Вартість

5. На п'ятому листі отримати відповідь про вартість і кількість кожного товару для кожної фірми. Для цього:

– скопіювати дані з першого листа на п'ятий;

– вибрати Вставка / Зведена таблиця;

– вказати діапазон всієї таблиці;

– далі в макеті здійснити розмітку таблиці (на місце стовпців помістити поле Одержувач, на місце рядків помістити поле Найменування, в області даних помістити поля Кількість і Вартість);

– відключити отримання загальних підсумків по рядках (область на стрічці Робота із зведеними таблицями вкладка Конструктор).

Завдання № 2.

Проаналізувати ефективність роботи двох автосалонів за наданими даними.

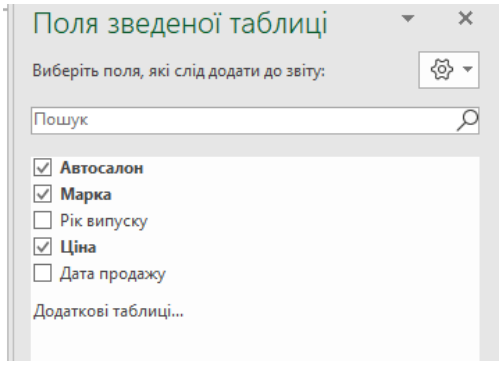
Хід роботи

1. Створіть нову електронну таблицю, що містить дані про продажі автомобілів двома автосалонами. Назвіть аркуш Продажі.

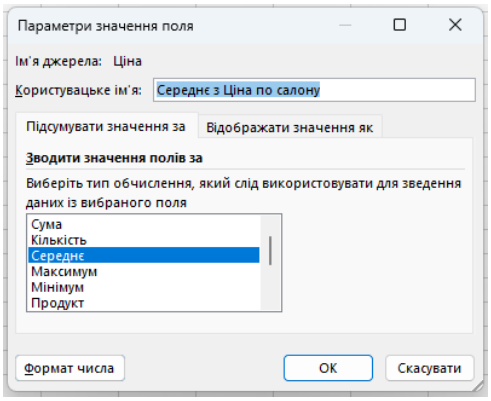
Автосалон	Марка	Рік випуску	Ціна	Дата продажу
RIOCAR	Opel	2023	400 000	11.02.2022
Ілта Львів	Ford	2022	525 000	12.09.2023
RIOCAR	Volvo	2023	1000 000	14.08.2023
Ілта Львів	Opel	2022	300 000	15.09.2022
RIOCAR	Volvo	2022	950 000	16.08.2022
Ілта Львів	Fiat	2023	450 000	06.06.2023
RIOCAR	Renault	2023	350 000	08.03.2023
Ілта Львів	Ford	2023	650 000	05.05.2022
RIOCAR	Mazda	2022	550 000	09.08.2022
Ілта Львів	Jeep	2023	650 000	20.09.2023
RIOCAR	Mazda	2024	625 000	18.09.2022

2. Виділіть будь-яку клітинку в таблиці та виберіть команду Вставлення -» Зведена таблиця. У вікні Зведена таблиця натисніть кнопку **ОК**.

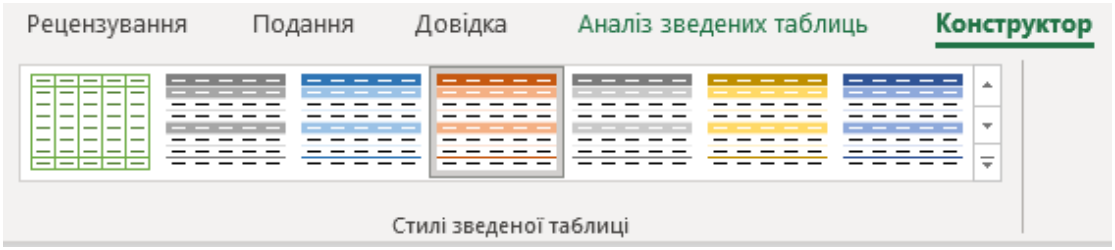
Примітка: До книги додастся новий аркуш, на якому буде розміщено макет зведеної таблиці, список полів та панель інструментів зведеної таблиці.



3. Перетягніть до вікна Фільтр звіту поле Автосалон.
4. Перетягніть до вікна Позначки рядків поле Марка.
5. До поля Значення помістіть поле Ціна.
6. Змініть функцію підсумовування по полю Ціна на середнє значення у вікні Параметри значення поля, яке відкривається подвійним клацанням заголовка поля.

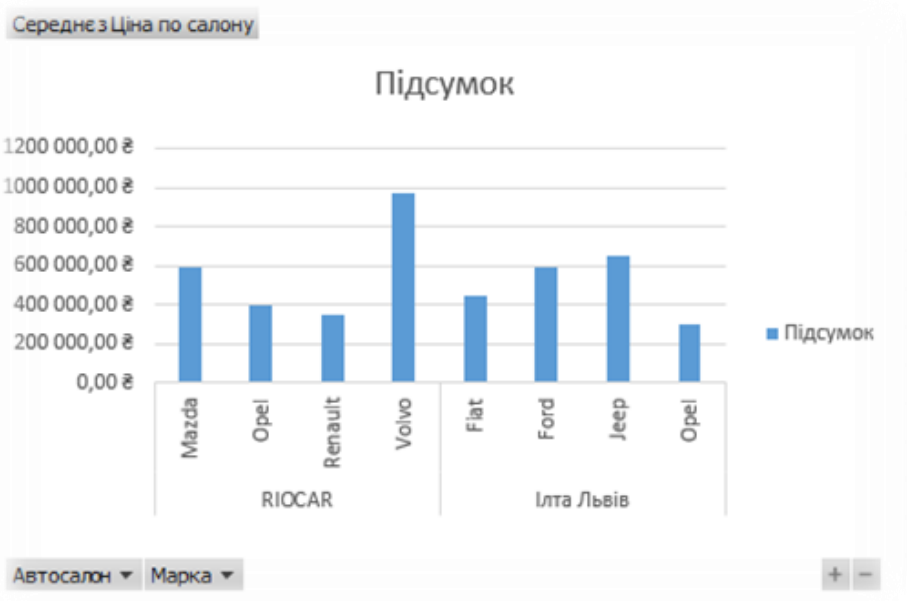


7. Змініть ім'я поля Середня ціна на Середня ціна по автосалону, використовуючи текстове поле Користувацьке ім'я в тому ж діалоговому вікні.
8. Змініть форматування числових значень зведеної таблиці, застосувавши грошовий формат (Робота зі зведеними таблицями -» Параметри -» Активне поле -» Параметри поля).
9. Застосуйте стилі до зведеної таблиці, виконавши панель Конструктор - > Стилі зведеної таблиці.

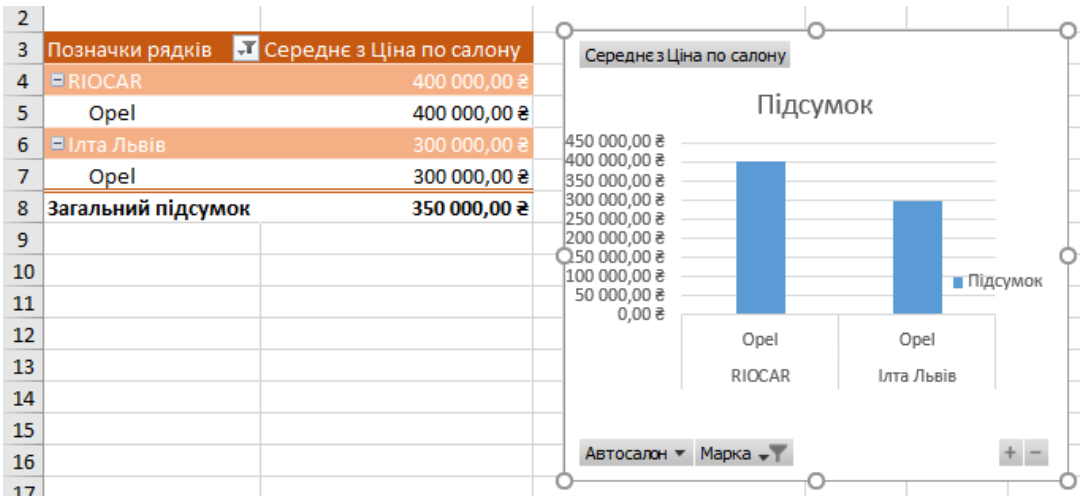


	Позначки рядків	Середнє з Ціна по салону
4	RIOCAR	645 833,33 ₴
5	Mazda	587 500,00 ₴
6	Opel	400 000,00 ₴
7	Renault	350 000,00 ₴
8	Volvo	975 000,00 ₴
9	Ілта Львів	515 000,00 ₴
10	Fiat	450 000,00 ₴
11	Ford	587 500,00 ₴
12	Jeep	650 000,00 ₴
13	Opel	300 000,00 ₴
14	**Загальний підсумок**	**586 363,64 ₴**

10. Побудуйте діаграму, що відображає дані зведеної таблиці, використовуючи команду Вставлення -» Зведена діаграма.
11. Сформуйте зведену таблицю, що дозволяє оцінити кількість і суму продажів машин різних марок. На рисунку відображено звіт за маркою Opel.



Зробіть висновок: які можливості для аналізу великих обсягів даних надають зведені таблиці.



Тести для самоконтролю

1. Електронні таблиці Excel призначені для:
   a) роботи лише з числовою інформацією;
   b) складання бухгалтерських, інженерних, статистичних документів
   c) лише побудови діаграм.
2. Як називається файл обробки даних у табличному процесорі?
   a) текст; b) сторінка; c) лист; d) книга.
3. В комірку таблиці EXCEL можна вносити:
   a) лише текст;
   b) лише числа;
   c) текст, числа і формули;
   d) малюнки.
4. Вкажіть правильно записаний адрес комірки в EXCEL:
   a) 1E23; b) a:12 ; c) d3.
5. Введення формули в комірку починають зі знаку:
   a) =; b) $; c) (; d) [.
6. Для запису математичних функцій в формули для обчислень в EXCEL потрібно:
   a) пам’ятати назви математичних функцій;
   b) вибрати потрібну функцію у вікні Майстер функцій;
   c) знайти потрібні піктограми в панелях інструментів.
7. Який з виразів записано правильно?
   a) =СУММ(A2:A10);
   b) СУММ(A2;A10);
   c) =СУММ(A2.A10);
   d) СУММА(A2:A10).
8. Для копіювання формули з однієї комірки в сусідню в EXCEL потрібно:
   a) навести мишею курсор на нижній правий кут комірки (курсор перетвориться на знак "+ ") і протягнути;
   b) навести мишею курсор на верхню межу комірки і протягнути одночасно натискаючи клавішу Shift;

c) навести мишею курсор на верхній лівий кут комірки і протягнути, одночасно натискаючи клавішу Shift.

9. У правому нижньому куті активної комірки є маленький чорний хрестик. Для чого він призначений?

a) для автоматичного заповнення комірок засобами Автозаповнення;
b) для виділення за допомогою миші блоку порожніх комірок;
c) для копіювання вмісту активної комірки у комірки, розташовані поруч з нею;
d) для виділення за допомогою миші блоку клітинок, які містять дані.

10. При зміні даних, внесених в комірки:

a) всі результати обчислень зберігаються в незмінному вигляді;
b) всі результати перераховуються;
c) старі результати потрапляють в буфер обміну.

11. Щоб побудувати діаграму в EXCEL, потрібно:

a) виділити дані, на основі яких будується діаграма і відкрити вікно Майстер діаграм ;
b) виділити дані, на основі яких будується діаграма і відкрити вікно Майстер функцій;
c) просто відкрити вікно Майстер діаграм.

12. Тип діаграми вибирають, враховуючи:

a) особливості даних;
b) естетичні вподобання;
c) модель вашого ПК;
d) особливості друку друкувального пристрою чи екрана для їх перегляду.

13. Що необхідно зробити для поновлення діаграми при зміні значень в таблиці, для якої побудовано діаграму Excel?

a) побудувати діаграму за новими даними;
b) виділити діаграму та натиснути клавішу F9;
c) діаграма змінюється автоматично.

14. Як виділити несуміжний діапазон комірок електронної таблиці?

a) клацнути на першій комірці, натиснути клавішу Ctrl і, утримуючи її, клацнути на інші комірки;
b) клацнути на першій комірці, натиснути клавішу Shift і, утримуючи її, клацнути на інші комірки;
c) клацнути на першій комірці, натиснути клавішу Alt і, утримуючи її, клацнути на інші комірки;
d) виконати команду Правка → Знайти та виділити.

15. Якщо в Excel після введення числа у комірку на екрані з'являється #######, це означає, що:

a) замість числа набрано текст;
b) при відображенні на екрані число не вміщується у комірці;
c) перед числом немає знаку «=».

РОЗДІЛ 4. ОСНОВИ АЛГОРИТМІЗАЦІЇ

Мета: Ознайомити з фундаментальними поняттями алгоритмізації, її основними властивостями, способами представлення, об'єктами дії та різновидами структур алгоритмів.

План

1. Поняття алгоритму.
2. Властивості алгоритму.
3. Способи опису (представлення) алгоритму.
4. Об'єкти дії в алгоритмах і програмах.
5. Різновиди структур алгоритмів.

Ключові терміни: *Алгоритм, властивості алгоритму, способи опису, об'єкти дії, структури алгоритмів, блок-схема, псевдокод, константа, змінна, оператор присвоєння, умовний оператор, оператор вибору, оператори циклу.*

4.1 Поняття алгоритму

Під ***алгоритмом*** розуміється «точне приписання, що визначає обчислювальний процес, який веде від початкових даних, що змінюються, до шуканого результату». Алгоритм включає систему правил, що визначають зміст і кінцеву послідовність дій (кроків і операцій), виконуваних над деякими об'єктами з метою переробки вихідних і проміжних даних у шуканий результат. Це приписання конкретному виконавцеві про те, які дії, над якими об'єктами й у якому порядку слід виконувати для отримання розв'язку поставленого завдання.

Поняття алгоритму, що ставиться до фундаментальних концепцій інформатики, виникло задовго до появи ПЕОМ. Алгоритми з IX століття й дотепер використовуються в математиці для позначення правил виконання чотирьох арифметичних дій: додавання, вирахування, множення й ділення. Протягом багатьох століть люди інтуїтивно користувалися алгоритмами знаходження кореня квадратного рівняння, розрахунків площі квадрата або кола, визначення об'єму куба, циліндра, конуса. У цей час сфера застосування цього поняття суттєво розширена. Про алгоритми говорять під час обговорення процесів керування виробництвом, проблем прийняття рішень або виконання деяких технологічних операцій.

4.2 Властивості алгоритму

При розробці алгоритмів слід ураховувати ряд вимог, сукупність яких формує його властивості. З основних властивостей алгоритму виділимо: *визначеність, дискретність, скінченність, результативність, раціональність, масовість*.

Вказівки, що становлять алгоритм, повинні бути чіткими й однозначними, не допускати довільного або двоякого тлумачення. Цю

властивість називають *визначеністю.* Вона є гарантією того, що алгоритм може бути виконаний об'єктами, що не мають інтелектуальних здатностей, зокрема ПЕОМ. При складанні алгоритму не можна розраховувати на професійного виконавця, який може проаналізувати і якось виправити хід розв'язку завдання, якщо буде потреба.

Визначений алгоритмом обчислювальний процес може бути розділений на окремі етапи (кроки), що представляють собою елементарні операції. У результаті чого виникає впорядкований запис сукупності приписань (директив, команд), що утворюють безперервну структуру алгоритму. У цьому впорядкованому записі виконання дій чергового приписання припустиме лише після виконання попереднього. Можливість поетапної деталізації алгоритму шляхом розкладання будь-якої складної структури на ряд простих, строго обкреслених дій, визначає властивість алгоритму, називану *дискретністю.* Обчислювальний процес після виконання заданої алгоритмом скінченної послідовності дій повинен закінчуватися видачею результатів або *повідомленням* про неможливість розв'язання завдання. Ці взаємозалежні властивості алгоритму називаються *скінченністю й результативністю.*

Алгоритм обчислювального процесу повинен привести до результату не просто за скінченне число кроків, що передбачене властивістю результативності, але й за найменший час при мінімальнім використанні ресурсів комп'ютера (оперативної пам'яті і ємності магнітних дисків). Ці вимоги визначаються властивістю *раціональності* алгоритму, що припускають багатоваріантність шляхів розв'язку будь-якого завдання, з яких слід вибирати найефективніший.

Нарешті, алгоритми повинні мати властивість *масовості,* щоб їх можна було використовувати для розв'язку безлічі однотипних завдань із різними вихідними даними.

4.3 Способи опису (представлення) алгоритму

Розроблені алгоритми можуть бути представлені на фізичному носії інформації різними способами, найбільш відомими з яких є: *словесний* (засобами мови людського спілкування з ретельно відібраним набором слів і фраз), *структурно-стилізований* (мовою псевдокодів), *графічний* (схемами із графічних блок-символів) або *програмний* (текстами програм).

Словесний спосіб запису алгоритмів являє собою опис послідовності дій по обробці даних природною мовою людського спілкування. Очевидні переваги: доступність і простота використання. Недоліки: алгоритми погано формалізуються, громіздкі й не наочні.

Структурно-стилізований спосіб запису заснований на застосуванні природної мови у формалізованому представленні приписань, конструкції якого близькі до конструкцій структурних мов програмування. У цій мові введений обмежений набір типових синтаксичних конструкцій (псевдокодів), представлених у зрозумілому для розроблювача й користувача виді, що й дозволяють запис алгоритму здійснити у формалізованій і більш виразній

формі. Наприклад, у мові АМРН початок алгоритму кодується як **нач,** кінець **кін,** ознакою заголовка є ключове слово **алг** і т.д.

Найпоширенішим способом представлення алгоритму є *графічний.* У графічному описі алгоритми зображуються у вигляді блок-схеми, доповненої елементами словесного або математичного запису. Схема алгоритму включає геометричні фігури (блокові символи), з'єднані між собою стрілками (лініями), що вказують порядок виконання операцій. Блокові символи стандартизовані й різняться по типу виконуваних дій. У схемі початок і завершення алгоритму, а також вхід і вихід з допоміжних алгоритмів, відзначаються відповідно блоковими символами «початок» і «кінець» (рис. 30а і 30б). Ці блоки, на відміну від більшості інших, мають один вхід або один вихід, що помічають як би початок і кінець шляху обробки інформації. Кожна схема обов'язково повинна починатися й закінчуватися цими символами.

Зображені на рисунку 30в й 30г блокові символи у вигляді паралелограма використовують для позначення операцій введення-виведення даних. Блок, що відображає обчислювальний процес, застосовують для позначення однієї або групи операцій, які змінюють значення, форму представлення або розміщення даних (рис. 30д). Виконувані операції в цьому блоці записують у будь-якій формі з використанням математичних формул, виразів і пояснень природною мовою. Логічний блоковий символ «розв'язок» (рис. 30ж, 30з, 30і) використовують для позначення вибору напрямку виконання алгоритму залежно від деякої умови (умов). У блоці вказують умову, питання або розв'язок, що визначають подальший напрямок виконання алгоритму. Умови можуть бути простими (рис. 30ж) і складовими (рис. 30і). У них повинні бути враховані абсолютно всі можливі варіанти проходження процесу при розв'язку завдання.

Із блоку перевірки умови може виходити два або три інформаційні потоки, що відрізняє його від інших блокових символів, що мають не більш одного виходу. Вихідні із блоку лінії повинні забезпечуватися поясненнями про напрямки виконання алгоритму при виконанні або невиконанні наведеної умови (наприклад, «так», «ні», <«0», «=0», «>0», «+», «–», або інші). Блоковий символ модифікації (рис. 30л) символізує початок циклічних обчислень (заголовок циклу), для керування якими він використовується. Усередині блоку вказується змінна циклу й параметри, що характеризують закон її зміни.

Крім вхідної лінії блок модифікації має одну вихідну (на рисунку 30л позначену «Вихід»), а також лінії, що відзначають передачу обчислювального процесу на обробку для циклічних обчислень «Тіло циклу» і повернення в початок для зміни змінної циклу «Зміна парам».

Для звертання до обчислення по підпрограмі (стандартної або розробленої користувачем) у схемі використовують блок-символ «визначений процес», зображений на рисунку 30к. Він як би заміняє алгоритм підпрограми (допоміжний алгоритм) і вказує, що інформаційний потік передається підпрограмі. По завершенню обчислювального процесу в підпрограмі результати розрахунків вертаються в основний алгоритм, у якому процес обчислень відновляється із блоку, наступного за блоком звертання до

підпрограми. Блок «визначений процес» використовують при організації допоміжних алгоритмів, оформлених автономно у вигляді окремого модуля, або при звертанні до бібліотечних підпрограм.

Для зменшення кількості перетинань і довжини ліній, що символізують шляхи проходження інформаційних потоків, допускається їх розривати, вставляючи в місця розриву з'єднувачі. Якщо лінія розривається між блоками, розміщеними на одній сторінці, то в якості з'єднувача використовують відповідні символи (рис. 30м і 30н). У блоковий символ вписують номери блоків, у які обчислювальний процес передається або з яких він вийшов. Так, з'єднувач на рисунку 30м указує, що обчислювальний процес передається на вхід блоку 15, а з'єднувач на рисунку 30н – що обчислювальний процес вийшов з виходу блоку 10. Якщо ж лінії з'єднують блоки, розташовані на різних сторінках, то використовується символ міжсторінкового з'єднувача, у який вписують не тільки номери блоків, але й номери сторінок. У зображеному на рисунку 30п міжсторінковому з'єднувачі зазначено, що обчислювальний процес передається на вхід блоку 21, який буде стояти на сторінці 10, а наведений на рисунку 30о міжсторінковий з'єднувач зазначає, що інформація передана з виходу блоку 14, розташованого на сторінці 6.

Для пояснень особливостей функціонування окремих блоків або груп блоків, прийнятих допущень і призначень окремих елементів, позначень змінних у схеми алгоритмів можуть включатися коментарі (рис. 30р).

Схема є самим наочним і простим способом представлення алгоритму. У ній чітко простежуються порядок виконання дій, потоки інформації й шляхи їх проходження, які помічаються лініями зі стрілками (стрілки допускається опускати, якщо потоки спрямовані зверху вниз і ліворуч праворуч). Лінії стосовно блоків можуть бути вхідними й вихідними. Кількість вхідних ліній для всіх блоків не обмежена – їх може бути одна, дві, три й більше. Вихідна ж лінія для більшості блоків може бути тільки одна (виключення – блоки перевірки умови).

У схемі блоки, за винятком з'єднувачів, можуть нумеруватися для простоти подальшого опису їх роботи, організації коментарів і використання з'єднувачів. Номери проставляють у верхній частині графічного символу в розриві його накреслення, як це зроблено на рис. 30ж (2-й блок).

Усередині блоків і поруч із ними робляться записи й позначення, що уточнюють виконувані функції. Ці записи можуть проводитися в будь-якій зручній для розробника формі. Вони не мають яких-небудь істотних обмежень (на мову, позначення, символи й інше), однак повинні бути зрозумілі всім, хто буде користуватися алгоритмом. Порядок обробки записів є детермінованим і не залежить від потоків інформації, а саме: зліва направо по кожному рядку, починаючи з верхнього.

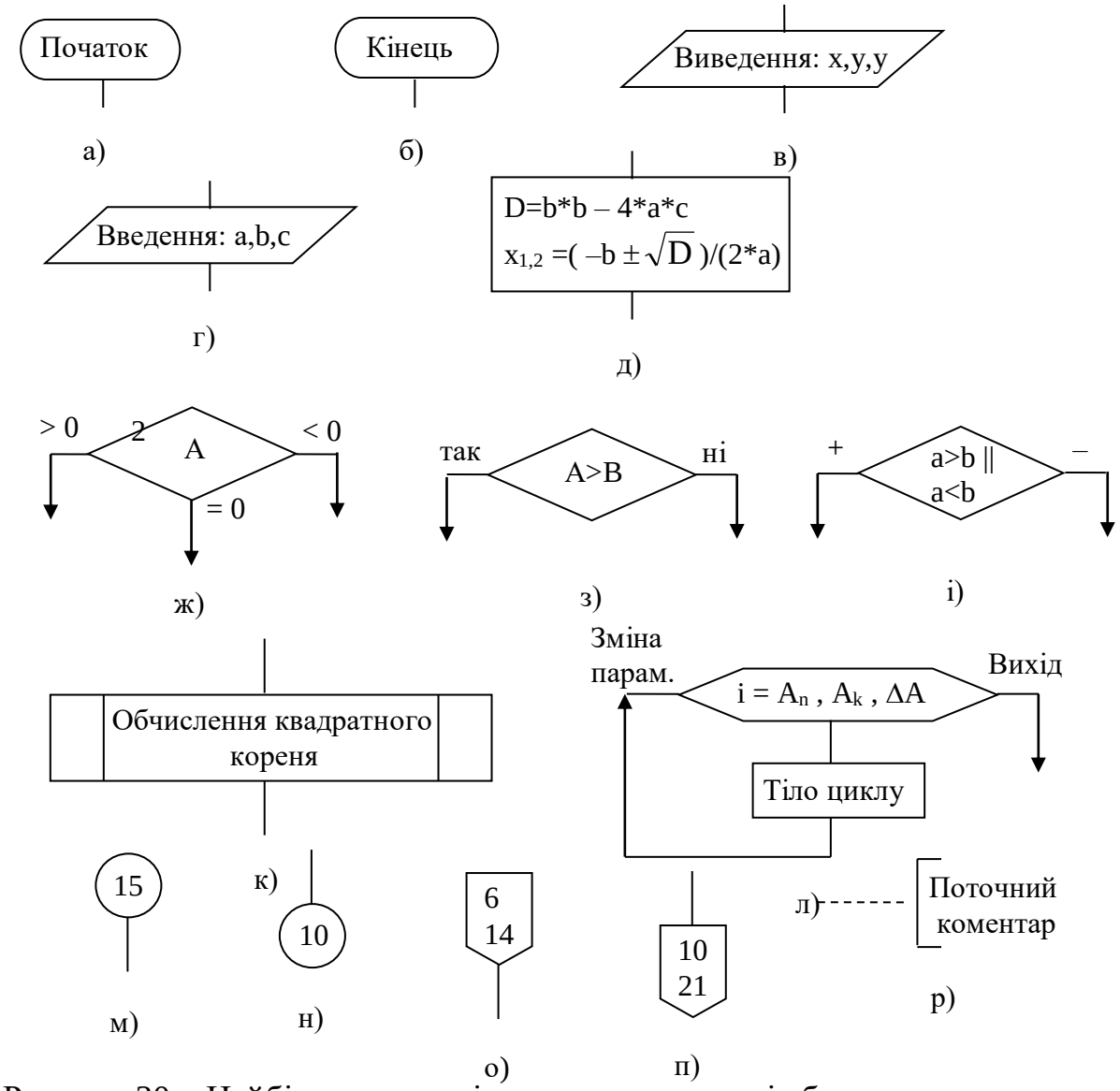


Рисунок 30 – Найбільш вживані в схемах алгоритмів блок-символи

Алгоритми доцільно розробляти поетапно (по кроках). Складні завдання слід розбивати на досить прості, легко сприймані частини. Логіка алгоритму повинна опиратися на мінімальне число досить простих керуючих базових структур. При розробці схем алгоритмів необхідно дотримувати деяких вимог:

– У схемі алгоритму всі лінії від блоку «початок» до блоку «кінець» не повинні мати розривів, не позначених з'єднувачами.
– Усі лінії, що вказують послідовність виконання дій, повинні бути замкнені. У схемі повинні чітко простежуватися потоки інформації. Блоки слід розміщати таким чином, щоб уникати перетинання ліній. При передачі керування в схемі « знизу-нагору» або «праворуч – ліворуч» лінії обов'язково позначають стрілками.
– Не допускається передача керування в нікуди. «Джерело» передачі керування й «одержувач» повинні бути чітко позначені.

4.4 Об'єкти дії в алгоритмах і програмах

Оброблювана інформація представляється в ПЕОМ у вигляді даних: *числових, логічних, символьних* (строкових) і *графічних*. Кожний із зазначених типів визначає множину значень, які може приймати змінна або повертати операція (функція), і в пам'яті ПЕОМ займає строго певний обсяг.

Числовий тип визначає множину значень із цифр, у своїй сукупності утворюючих число, яке бере участь в арифметичних операціях обробки. Числові дані представляються в цілому або дійсному виді.

Числові дані *цілого* типу включають деяку підмножину цілих числових значень, наприклад 10, –25, 237. Цей тип даних в основному використовують для позначення номерів, індексів, організації лічильників подій, введення цілочислових ознак при обробці інформації. В арифметичних операціях над цілочисловими величинами результат є ціла величина.

Числові дані *дійсного* типу представляються з фіксованою або плаваючою точкою (для обробки на ПЭВМ ціла й дробова частина дійсного числа розділяється точкою). Загальноприйнятою (природною) формою представлення числа є його відображення з фіксованою точкою, наприклад, 27.6 і 0.14. У так званій експонентній (нормалізованій) формі запису воно представляється у вигляді мантиси, помноженої на 10 у відповідному степені, наприклад число 1450.0 відображається як $14.5*10^2$, $1.45*10^3$ або $0.145*10^4$.

Дані дійсного типу зберігаються й обробляються в наближеному виді, наприклад, число 1.0 у ПЭВМ представляється, як 0.99999999 або 1.00000001 (число 1.0 відображає дійсний тип на відміну від цілого 1). На відміну від дій над цілими числами, операції над дійсними даними приводять до наближеного результату, точність якого багато в чому визначається можливостями ПЕОМ і помилками округлення.

Логічний тип даних представляється всього двома значеннями, які позначаються як *істина* або *хибність* (true або false). Цей тип даних може брати участь в операціях логічного множення (кон’юнкція), логічного додавання (диз'юнкція) і логічного заперечення.

Символьний тип поєднує множину символів, призначених для організації обробки й виводу (на екран або принтер) текстової інформації й коментарів. Це можуть бути літери, що входять у різні алфавіти (кирилиця, латинь і інші), символи арифметичних операцій (+, –, / і *), операцій порівняння (<, >, >=, <=, == або !=), цифрові (0, 1, ..., 9) і багато інших (#, [,], (,), і т.п.). Символьні дані зберігаються в ПЕОМ і на зовнішніх носіях у вигляді двійкових кодів і при обробці можуть брати участь у деяких характерних для них операціях, наприклад порівняння, зчленування (об'єднання), пошуку.

Графічний тип даних містить інформацію про місце розташування, кольорах й відтінках точок на екрані комп'ютера, яка у своїй сукупності дозволяє відображати графічні об'єкти.

Дані в оперативній пам'яті комп'ютера зберігаються під якимось унікальним іменем, яке в явному або неявному виді визначає їхній тип. *Ім'я* (*ідентифікатор*) може складатися з літер і цифр і починається з літери. Ім'я

вказує операційній системі тип даних і місце розташування їх конкретних значень в області оперативної пам'яті. Під певним іменем оброблювані дані можуть бути представлені у формі констант або змінних.

Константи – це такі дані (числа або символи), які задані в явному виді й не змінюються в процесі виконання алгоритму або програми. Константи зберігаються й обробляються під фіксованим іменем, мають незмінні тип і значення. Так, якщо в програмі необхідно багаторазово обчислювати значення площі кругу або довжину кола, то в алгоритм доцільно ввести константу 3.141592 дійсного типу й привласнити їй деяке ім'я, наприклад pi (PI).

У формі *змінних* представляють пойменовані дані, що характеризуються постійним типом і значення, що змінюється у процесі виконання операцій. Тип змінної визначається іменем або явно вказується операторами опису типів. У кожний момент часу змінна може мати тільки одне дискретне значення, яке фіксується у відведеній для неї області пам'яті. При одержанні нового значення змінної, наприклад після виконання якої-небудь операції, воно записується замість старого в ту ж область пам'яті.

При виконанні алгоритму або програми будь-якій змінній може бути привласнене значення іншої змінної, константи, результатів виконання операції або обчислення функції. *Присвоювання* – одна з основних дій, виконуваних ПЭВМ. В алгоритмах операція присвоювання записується знаком «=». Наприклад, вказівка A = B визначає присвоювання змінній з іменем A значення змінної B. Виконуючи приписання, значення змінної B переправиться в область змінної A, зберігаючись одночасно й в області змінної B.

Часто в алгоритмах і програмах використовують операцію присвоювання типу $j = j + 1$, що приписує читання значення змінної j з відведеної для неї області пам'яті, додавання до неї одиниці й запис її нового значення в ту ж область пам'яті замість старого. У результаті виконання операції значення змінної j збільшилося рівно на 1. Змінні, що мають у якості поточного значення тільки одну величину, називають *простими* (*скалярними*). Прості змінні можуть поєднуватися в деяку структуру, на яку допускається посилатися як на єдиний об'єкт, незважаючи на те що він складається з окремих елементів. Прикладом такої *структурованої* змінної є масив.

Масив – це підмножина впорядкованих щодо індексу (номера) однотипних даних, об'єднаних за якою-небудь ознакою. Масив характеризується типом елементів, розміром і розмірністю. Будь-який елемент масиву займає відведену для нього область пам'яті, є однаково доступним і може вибиратися довільно.

Усі елементи масиву беруть участь в обробці під одним загальним іменем. Їхнє положення в масиві впорядковано по одному, двом або більш вимірам. Число вимірів визначає *розмірність* масиву. Якщо елементи масиву впорядковані по одному виміру, то масив називають одномірним, двом – двовимірним і так далі. За іменем масиву слідує список індексів, що представляє послідовність цілих чисел, змінних цілого типу або арифметичних виразів. Кожний індекс задає номер позиції елемента по відповідному виміру й може змінюватися в заданому діапазоні. Індекси однозначно визначають місце

елемента в масиві, тобто його місце розташування в просторі, описаному масивом.

Кожний масив характеризується *розміром,* який слід указати при заявці місця під його розміщення в області оперативної пам'яті. Під розміром розуміється сумарна кількість усіх елементів масиву. Якщо масив багатомірний, то розмір визначають як добуток кількості вимірів на кількість елементів в одному вимірі. Змінні, що описують елементи масиву, називають *індексованими* (змінними з індексом).

4.5 Різновиди структур алгоритмів

За структурою алгоритми розділяють на *лінійні, що розгалужуються* й *циклічні*.

Визначення. *Лінійними* називають алгоритми, у яких операції виконуються послідовно одна за іншою, у природному і єдиному порядку проходження. У таких алгоритмах усі блоки мають послідовне з'єднання логічним зв'язком передачі інформаційних потоків. У них можуть використовуватися всі блоки, за винятком блоків перевірки умови й модифікації. Лінійні алгоритми, як правило, є складовою частиною будь-якого алгоритмічного процесу.

Наприклад, для того щоб відправити SMS-повідомлення, необхідно:

1) початок;
2) включити телефон;
3) вибрати команду «відправити SMS»;
4) написати текст повідомлення;
5) ввести номер телефону абонента;
6) натиснути кнопку відправити;
7) кінець.

Приклад 1. Знайти значення функції

$$Y = \operatorname{arctg}\left[\sqrt{\frac{ax^2+b}{c}} + \ln\left(\frac{ax^2+b}{c}\right)\right]$$

при значенні аргументу x=1,5 і заданих *a, b, c.*

Очевидно, що функцію Y доцільно обчислювати в такій послідовності: попередньо ввівши вихідні дані *a, b, c* і привласнивши значення змінній *x,* спочатку знайдемо значення вираз $(ax^2+b)/c$, яке позначимо змінною *z*, і далі визначимо вираз $\operatorname{arctg}(\sqrt{z} + \ln z)$. Використовуючи загальноприйняті символи блоків (рис.30), зобразимо схему розроблювального алгоритму (рис. 31).

При складанні схем алгоритмів часто виникає необхідність проведення аналізу вихідних даних або проміжних результатів обчислень і визначення подальшого порядку виконання обчислювального процесу залежно від результатів цього аналізу.

Визначення. Алгоритми, у яких залежно від виконання деякої логічної умови відбувається розгалуження обчислень по одному з декількох можливих напрямків, називають, алгоритмами, що *розгалужуються*. Подібні алгоритми

передбачають вибір одного з альтернативних шляхів продовження обчислень. Кожний можливий напрямок обчислень називається гілкою. Логічну умову називають *простою,* якщо процес, що розгалужується, має дві гілки, *і складною,* – якщо процес розгалужується на три й більше гілок.

Будь-яка складна логічна умова може бути представлена у вигляді простих.

Розглянемо приклад алгоритму, що розгалужується, із простою логічною умовою.

Приклад 2. Дано два числа a й b. Знайти $x = \begin{cases} a/b, & a > b \\ b/a, & a \le b \end{cases}$.

Очевидно, що для визначення областей, по яким необхідно робити процес обчислення значення *x,* досить перевірити виконання однієї з умов, наприклад a>b.

Якщо умова a>b не виконується, то очевидно й без додаткової перевірки, що буде виконана умова a < b. Отже, варіант схеми алгоритму буде виглядати в такий спосіб (рис. 32).

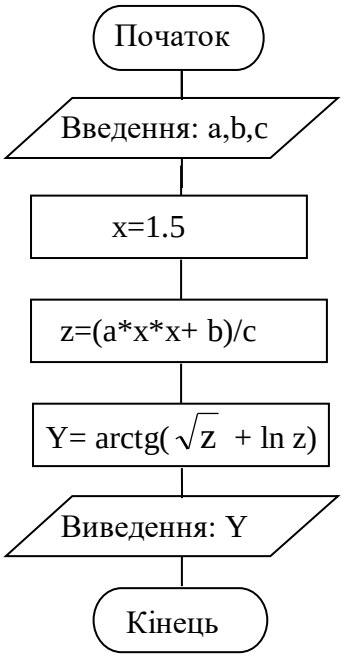


Рисунок 31 – Схема лінійного алгоритму

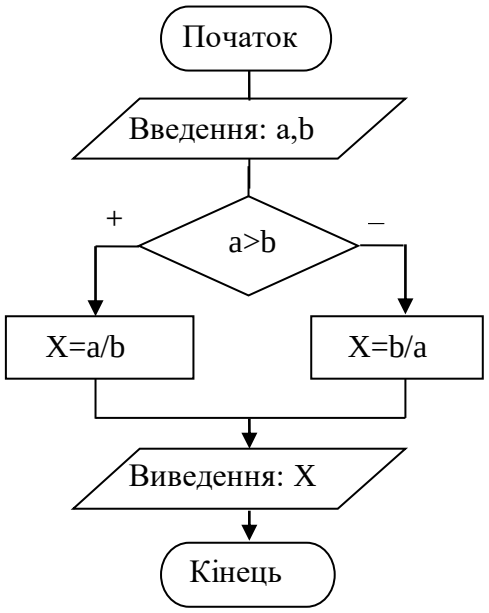
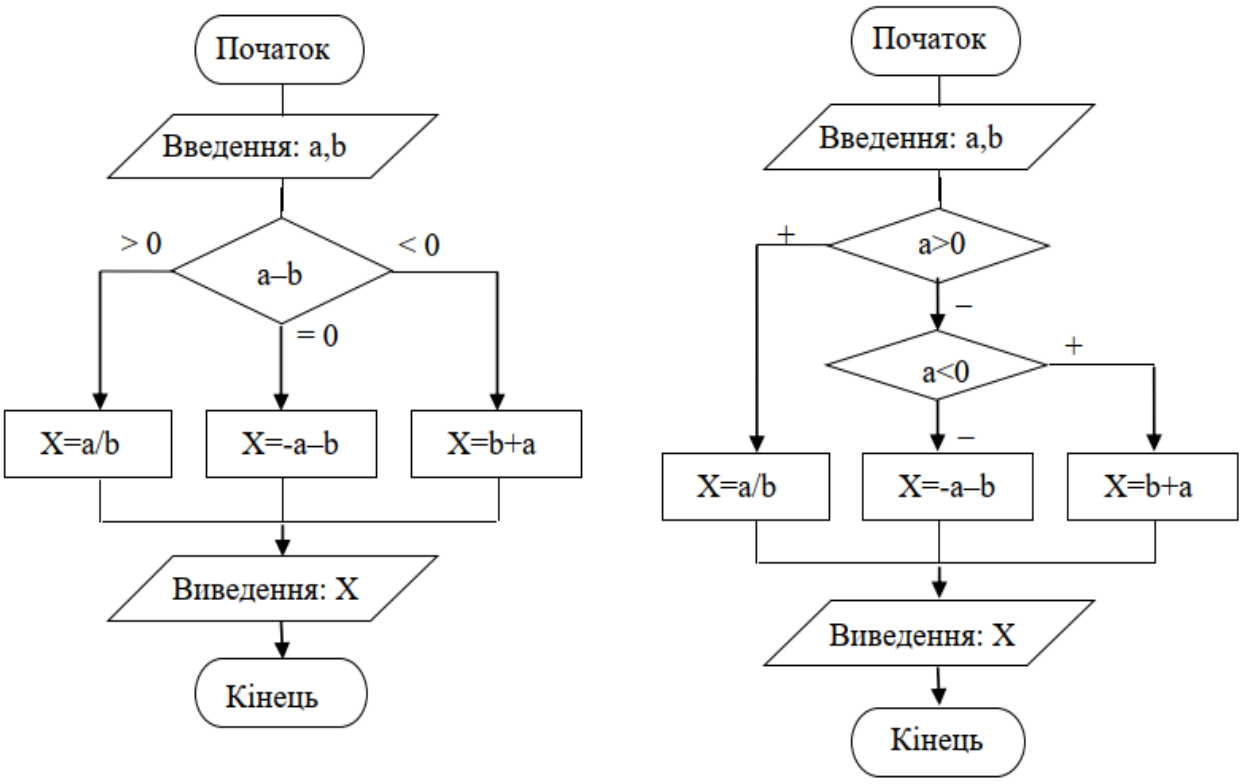


Рисунок 32 – Схема простого алгоритму, що розгалужується

Приклад 3. Дано два числа a й b. Знайти $x = \begin{cases} a/b, & a > 0 \\ -a-b, & a = 0 \\ a+b, & a < 0 \end{cases}$.

З умови очевидно, що обчислювальний процес повинен передбачати вибір одного з альтернативних шляхів обчислень залежно від значення змінної *a* (див. рис. 33).



а) б)

Рисунок 33 – Схеми алгоритмів, що розгалужується: а) зі складною логічною умовою; б) при розкладанні складної логічної умови на прості.

Приклад 4. Обчислити значення *Y* при заданих значеннях a, *x*.

$$Y = \begin{cases} \sin(a), & x < -2 \\ a^2 + x^2, & -2 \le x \le 0 \\ \cos(x), & 0 < x \le 5 \\ |x - a|, & x > 5 \end{cases}$$

При цьому алгоритм може бути представлено в одному із двох варіантів: з розгалуженням по складній логічній умові (рис. 33*а)* і з розгалуженням при розкладанні складної логічної умови на прості (рис. 33б).

Незважаючи на те, що в прикладі пропонуються чотири логічні умови, дві з яких складні, при складанні алгоритму досить організувати тільки три прості порівняння. Інші порівняння необов'язкові, оскільки відповідні умови й так є єдино можливими при невиконанні запропонованих порівнянь (рис. 34). Наприклад, перевірка логічної умови *x*<=0 можлива тільки при невиконанні умови *x*< –2, що у свою чергу можливо тільки при *x*>= –2.

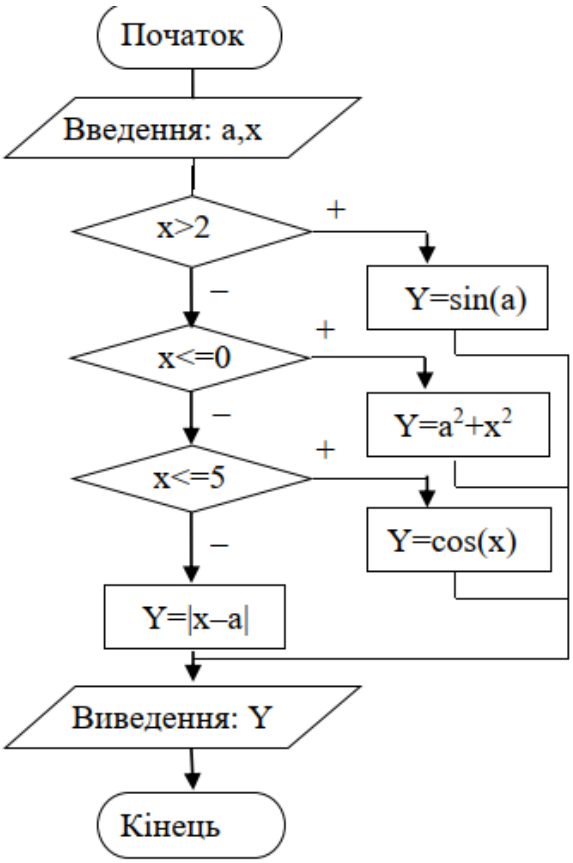


Рисунок 34 – Схема алгоритму, що розгалужується

Приклад 5. Дано три нерівні між собою числа *a, b, c.* Визначити найбільше з них.

В алгоритмі (рис. 35) спочатку зрівняємо між собою два перші числа *a* й *b*. Потім більше з них зрівняємо із третім числом, що залишилося, *c*. Більше число в цьому порівнянні і є найбільшим із трьох.

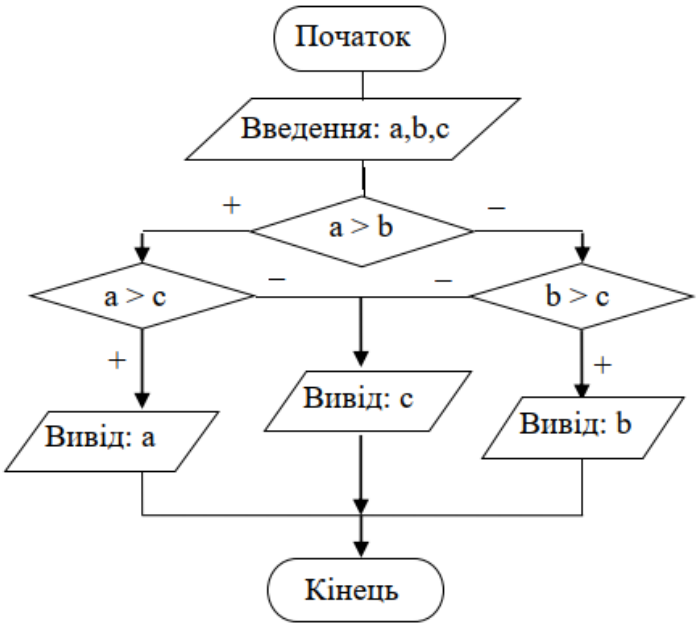


Рисунок 35 – Схема алгоритму пошуку найбільшого із трьох чисел

При збільшенні числа змінних до 4 і більше схема алгоритму зажадає великої кількості порівнянь, що ускладнить аналіз умов їх виконання й може привести до помилок при розробці алгоритму. До того ж збільшиться кількість ліній, що з'єднують ці блоки, що ускладнить читаність алгоритму.

Тому при розв'язку подібних завдань раціональніше змінити підхід до знаходження максимального значення.

Приклад 6. Задано чотири нерівні між собою величини *a, b, c, d.* Визначити найбільшу з них.

В алгоритмі (рис. 36) змінній *max* оператором присвоювання задається первісне значення, рівне значенню змінної *a*. Якщо при порівнянні значення змінної *max* зі значенням інших змінних *b, c, d* знайдеться більше max-го число, то його значення стане новим значенням змінної max. В блоці виводу здійснює вивід найбільшого із чисел *a, b, c, d,* привласненого в якості значення змінній max.

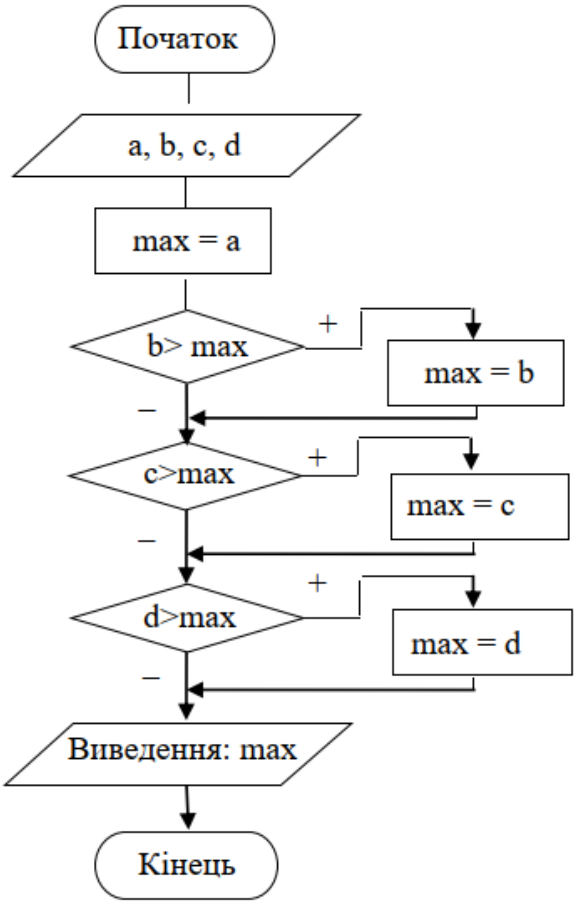


Рисунок 36 – Схема алгоритму знаходження найбільшого з 4-х заданих чисел

Алгоритм **циклічної** структури передбачає багаторазове повторення дій в одній і тій же послідовності по тим самим математичним залежностям, але при різних значеннях деякої спеціально змінюваної величини. Циклічні алгоритми дозволяють суттєво скоротити обсяг програми за рахунок багаторазового виконання групи повторюваних обчислень, так званого тіла

циклу. Спеціально змінюваний за заданим законом параметр, що входить у тіло циклу, називається змінною циклу. Змінна циклу використовується для підготовки чергового повторення циклу й відстеження умов його закінчення. У якості змінної циклу використовують будь-які змінні, індекси масивів, аргументи функцій, що обчислюються, і тому подібні величини. Під час виконання тіла циклу параметри змінної циклу змінюються в інтервалі від початкового до кінцевого значення із заданим кроком. Отже, при організації циклічних обчислень необхідно передбачити завдання початкового значення змінної циклу, закон її зміни перед кожним новим повторенням і її кінцеве значення, при досягненні якого відбудеться завершення циклу.

Цикли, у тілі яких немає розгалужень і інших вбудованих у них циклів, називають *простими.* А якщо ні, то їх відносять до *складних.* Циклічні алгоритми розділяють на *детерміновані* й ітераційні.

Цикли, у яких число повторень заздалегідь відомо з вихідних даних або визначене в ході розв'язку завдання, називають *детермінованими.* Для організації детермінованих циклів найбільше доцільно використовувати блок модифікації, усередині якого вказується змінна циклу, її початкове й кінцеве значення, а також крок її зміни (якщо крок зміни рівний 1, то його допускається не вказувати). Організувати подібний цикл можливо й при використанні блоку перевірки умови замість блоку модифікації, однак при цьому трохи ускладнюється алгоритм і губиться його раціональність.

Приклад 7. Дане натуральне число N. Знайти суму перших N членів натурального ряду.

Варіанти схеми алгоритму циклічної структури розв'язку поставленого завдання наведені на рисунку 37. При цьому в схемі на рис. 37*а* цикл організований з використанням блоку модифікації, а на рис. 37б – блоку перевірки умови. В обох алгоритмах операція знаходження суми, при попередньому обнулінні значення змінної S, повторюється N раз.

У тілі циклу використана операція присвоювання $S = S + i$, по якій і здійснюється обчислення суми шляхом додавання до попереднього значення змінної S усіх нових значень змінної i.

Цикл є детермінованим і кількість його повторень заздалегідь визначена (N раз). У якості змінної циклу i прийняте поточне значення членів натурального ряду. Використання алгоритму із блоком модифікації переважніше, тому що він має кращу наочність.

Цикли, у яких число повторень невідомо з вихідних даних і не визначене по ходу розв'язку завдання, називають *ітераційними.* В ітераційних циклах для організації виходу з тіла циклу передбачається перевірка деякої заздалегідь заданої умови, для чого використовують блок перевірки умови. В ітераційних циклах неможливо використовувати блоки модифікації, тому що при організації таких циклів заздалегідь невідома кількість змін змінної циклу і її кінцеве значення. Залежно від місцезнаходження блоку перевірки умови ітераційні цикли можуть бути організовані як цикли із передумовою (блок перевірки умови розміщено перед тілом циклу) або з постумовою (блок

перевірки умови розміщено після тіла циклу). Якщо в циклі із передумовою команди, що входять в тіло циклу, можуть не виконуватися жодного разу (якщо початкове значення параметра циклу задовольняє умові виходу із циклу), то в циклі з постумовою команди, що входять в тіло циклу, виконуються як мінімум один раз (навіть якщо початкове значення параметра циклу задовольняє умові виходу з нього).

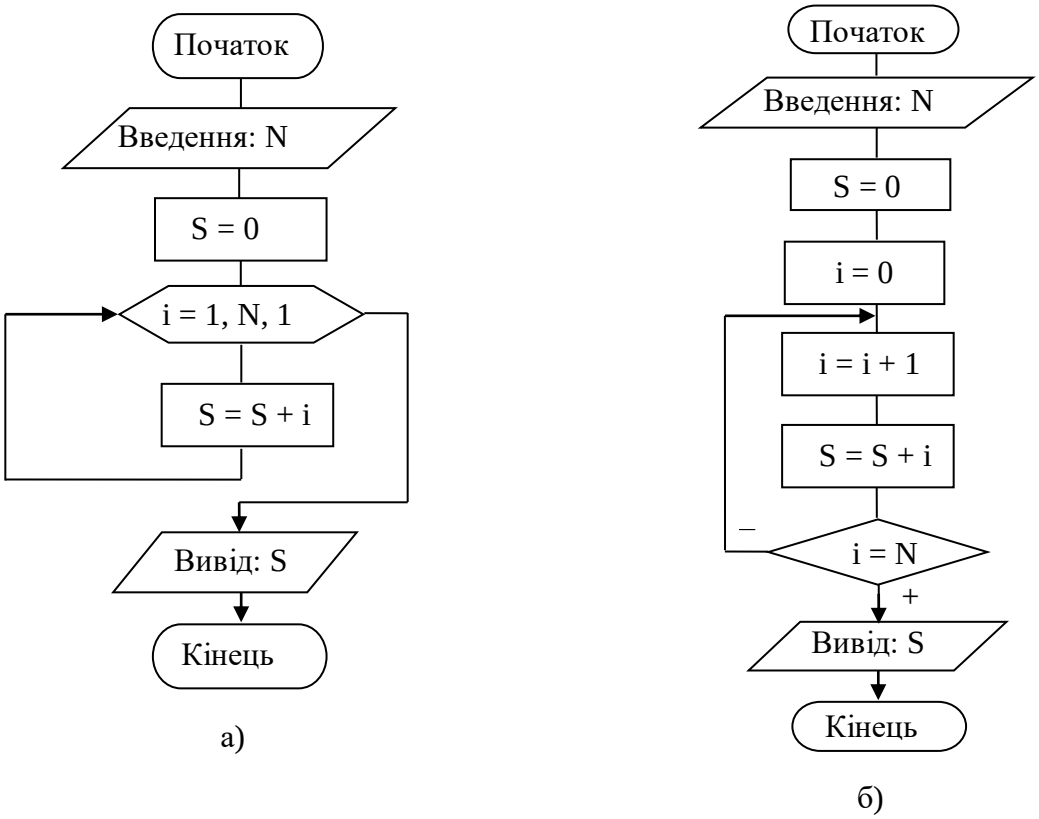


Рисунок 37 – Схема алгоритму циклічної структури для знаходження суми *N* перших членів натурального ряду: а) з використанням блоку модифікації; б) з використанням блоку перевірки умови

Приклад 8. Дано ряд натуральних чисел 1,2,3,..., ∞ і число *N*. Потрібно знайти суму перших членів ряду. Обчислення суми припинити, як тільки її значення буде рівно або перевищить задане *N*.

Варіант схеми алгоритму розв'язку завдання, що включає ітераційний цикл із постумовою, наведений на рисунку 38б. Цикл організований у вигляді ітераційного тому, що число його повторень заздалегідь невідомо. В алгоритмі вихід із циклу або його продовження визначається виконанням (або невиконанням) умови *S*>=*N*. Якщо умова не виконується, то обчислення суми триває шляхом додавання до попереднього значення суми (змінної *S*) значення чергового члена ряду, що відслідковується змінною циклу *i*. Однак наведений алгоритм дає невірний розв'язок при *N*=0, тому що в цьому випадку не буде виконано перше по порядку порівняння *S*=*N*=0.

Правильний алгоритм розв'язку цього завдання з використанням ітераційного циклу із передумовою наведено на рисунку 38а. У цьому алгоритмі тіло циклу розташоване після перевірки умови виходу з нього. У ньому на початку циклу

здійснюється перевірка ($S>=N$), і якщо N задане рівним нулю, то тіло циклу не буде виконуватися жодного разу.

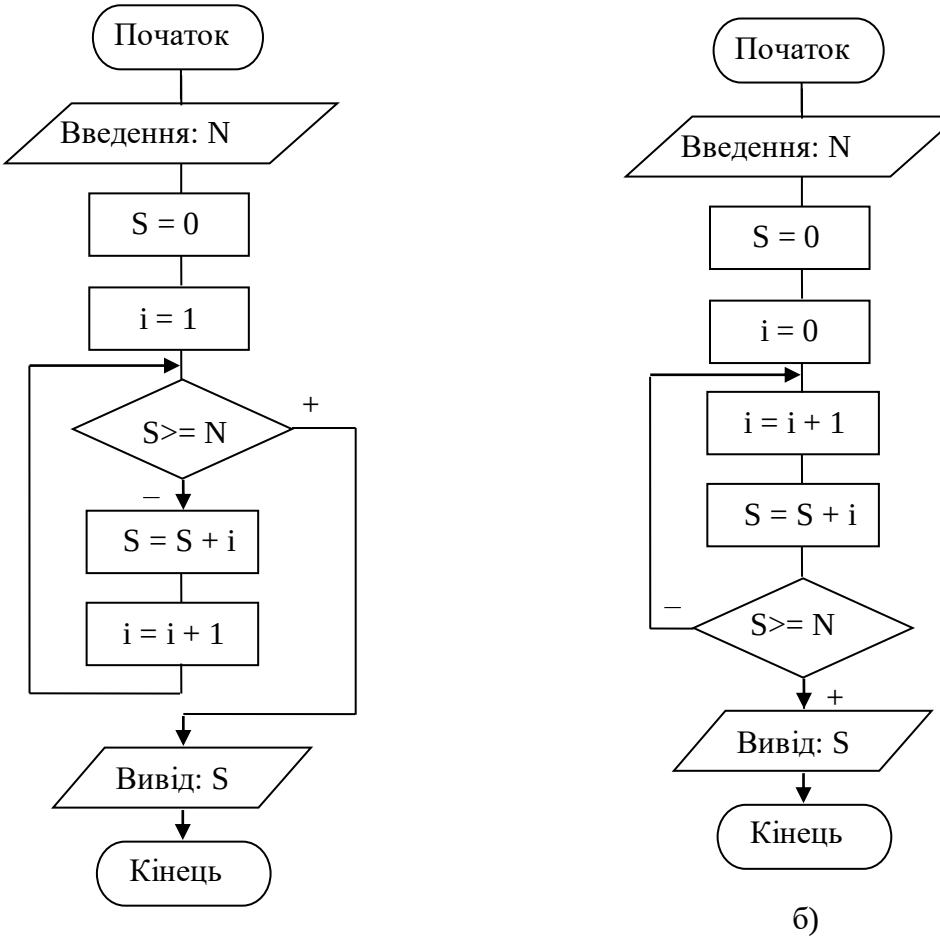


Рисунок 42 – Схема ітераційного алгоритму знаходження суми перших членів натурального ряду: а) з передумовою; б) з постумовою.



Питання для самоконтролю

1. Для чого служить алгоритм?
2. Що таке лінійний алгоритм?
3. Як записати будь-який алгоритм?
4. Де зустрічаються лінійні алгоритми?
5. Що таке визначеність?
6. Що таке дискретність?
7. Що таке масовість?
8. Що таке результативність?
9. Назвіть три типи обчислювальних процесів, які виконуються на ЕОМ?
10. Що таке блок-схема?
11. Для чого служить блок-схема?
12. Який елемент блок-схеми має один вхід і два виходи?
13. Який елемент блок-схеми не має входу?

14. Який елемент блок-схеми не має виходу?
15. Чому елемент блок-схеми «умова» має два виходи?
16. За допомогою якого елемента описується виконання однієї або декількох операцій?
17. У яких випадках використовується елемент блок-схеми «умова»?
18. Для чого використовується елемент блок-схеми «підпрограма»?

Індивідуальні завдання

1. Алгоритми лінійної структури. Скласти блок-схеми алгоритмів обчислення значення функції *Y* за варіантами:

1. $y = \dfrac{x^4 - bx^3 - a}{(x+a)(x-b)}$, де $b = 2^x;\ x = \lg 0{,}05 + 2;\ a = b + 2$

2. $y = \dfrac{x + 3a - k_1}{k_1 x + k_2}$, де $x = 7a + k_1 \cdot k_2;\ k_1 = 0{,}8;\ k_2 = 4{,}8;\ a = 0{,}25$

3. $y = \dfrac{\sin^3 ax + b}{\cos^2 x}$, де $x = -3{,}8;\ a = 0{,}5c + x^2;\ c = \ln 0{,}08;\ b = x^2 + c$

4. $y = \sin\dfrac{a}{x} + \lg 0{,}08e^x$, де $x = -2{,}5a^2;\ a = 0{,}8b + c;\ c = 0{,}5;\ b = c^3$

5. $y = \dfrac{\left(a\sqrt[3]{b} - cd\right)^2}{a + b + c}$, де $a = 7{,}14;\ b = a^2 - 1;\ c = a^3 - b^3;\ d = \sqrt{(b-c)^3}$

6. $y = \dfrac{\alpha_1\beta_1 - \alpha_2\beta_2}{\alpha_1 + \beta_1^2}$, де $\alpha_1 = \sin 0{,}18;\ \beta_1 = e^4;\ \alpha_2 = \ln 0{,}5 + \beta_1;\ \beta_2 = 1{,}7 \cdot \beta_1$

7. $y = \dfrac{(a+b)^2 + m}{1 + x^a + b^a}$, де $x = 0{,}81;\ a = \lg 0{,}83;\ b = e^a;\ m = x - 1$

2. Алгоритми розгалуженої структури. Скласти блок-схеми алгоритмів обчислення значення функції *Y* за варіантами:

1. $$y=\begin{cases}x^2-\sin\gamma, & x\le 0\\ \sqrt{x}+\cos\gamma, & x>0\end{cases};$$
$\gamma=0{,}35a;\ a=x+3;\ x=1{,}8.$

2. $$y=\begin{cases}4x^3+\gamma, & x\le 2\\ (x+3\gamma)\cdot x, & x>2\end{cases}$$
$x=1{,}4;\ a=3{,}5x;\ \gamma=\cos(a+x).$

3. $$y=\begin{cases}x^3+a, & x<0\\ \sin\dfrac{x}{a}, & x=0\\ \sqrt{x}+\dfrac{a}{2}, & x>0\end{cases}$$
$x=1{,}3-\ln a;\ a=0{,}27.$

4. $$y=\begin{cases}e^{-x+2}+a, & x\ge 1\\ \dfrac{\sin(x+3{,}2)}{a+3}, & x<1\end{cases}$$
$a=14{,}8;\ x=e^a.$

5. $$y=\begin{cases}\operatorname{tg}\dfrac{1}{x^2+8{,}2}, & x\ge 0\\ \dfrac{0{,}32x}{x^2+3}, & x<0\end{cases}$$
$x=0{,}5k+b;\ b=\sin 1{,}7;\ k=-0{,}8.$

6. $$y=\begin{cases}\dfrac{9bx}{x-2bx^2}, & x<2\\ \cos(b+x), & x\ge 2\end{cases}$$
$x=\ln 0{,}7;\ b=0{,}3.$

7. $$y=\begin{cases}\cos x-\sin^3 x, & x\ge 1{,}5\\ xe^{-x}, & x<1{,}5\end{cases}$$
$x=0{,}9\cdot\sin a;\ a=e^{0{,}17}.$

8. $$y=\begin{cases}x(A-C), & A=C\\ x^3-A, & A>C\\ x^3+A, & A<C\end{cases}$$
$x=8{,}6;\ a=2\cos x;\ C=A+1.$

9. $$y=\begin{cases}0{,}5\cos x, & x<1\\ 0{,}25x^a, & x=1\\ 0{,}9\sqrt{x}, & x>1\end{cases}$$
$x=1{,}7-e^{0{,}35}.$

10. $$y=\begin{cases}6z^2-5, & x\le 1\\ 5z^3+1, & x>1\end{cases}$$
$z=0{,}5;\ x=\operatorname{arccos} z.$

3. Створення алгоритмів лінійної, розгалуженої та циклічної структури для виконання математичних обчислень, логічних перевірок та обробки простих інженерних даних.

3.1 Розрахунок площі прямокутника: Напишіть алгоритм, який отримує на вхід довжину та ширину прямокутника, а потім обчислює та виводить його площу. Представите цей алгоритм у вигляді блок-схеми.

3.2 Перетворення температури з Цельсія у Фаренгейт: Напишіть алгоритм, який отримує на вхід значення температури в градусах Цельсія,

переводить його у градуси Фаренгейта за формулою: $F = (C * 9/5) + 32$, і виводить результат.

3.3 Обчислення середнього арифметичного двох чисел: Напишіть алгоритм, який отримує на вхід два числа, обчислює їхнє середнє арифметичне та виводить результат.

3.4 Знаходження гіпотенузи прямокутного трикутника: Напишіть алгоритм, який отримує на вхід довжини двох катетів прямокутного трикутника (a і b), обчислює довжину гіпотенузи (c) за теоремою Піфагора ($c^2 = a^2 + b^2$) та виводить результат.

3.5 Перевірка парності числа: Напишіть алгоритм, який отримує на вхід ціле число та визначає, чи є воно парним або непарним. Алгоритм повинен вивести відповідне повідомлення.

3.6 Визначення знаку числа: Напишіть алгоритм, який отримує на вхід число та визначає, чи є воно додатним, від'ємним або нулем. Алгоритм повинен вивести відповідне повідомлення.

3.7 Перевірка можливості існування трикутника: Напишіть алгоритм, який отримує на вхід довжини трьох сторін. Алгоритм повинен перевірити, чи може існувати трикутник з такими сторонами (сума будь-яких двох сторін повинна бути більшою за третю) та вивести відповідне повідомлення.

3.8 Виведення чисел у заданому діапазоні: Напишіть алгоритм, який отримує на вхід два цілих числа (початок і кінець діапазону) та виводить всі цілі числа в цьому діапазоні включно.

3.9 Обчислення суми перших N натуральних чисел: Напишіть алгоритм, який отримує на вхід натуральне число N та обчислює суму всіх натуральних чисел від 1 до N.

3.10 Обчислення факторіала числа: Напишіть алгоритм, який отримує на вхід невід'ємне ціле число N та обчислює його факторіал ($N!$).

3.11 Виведення таблиці множення на задане число: Напишіть алгоритм, який отримує на вхід ціле число (наприклад, від 1 до 10) та виводить таблицю множення цього числа на числа від 1 до 10.

3.12 Розрахунок опору складної електричної мережі: Опишіть алгоритм для розрахунку загального опору електричної мережі, що складається з послідовно та паралельно з'єднаних резисторів. На вхід алгоритм отримує значення опорів кожного резистора та інформацію про їхнє з'єднання. Розробіть блок-схему або псевдокод для цього алгоритму.

3.13 Сортування даних вимірювань: У процесі експерименту отримано набір числових даних (наприклад, значення температури, тиску, швидкості). Напишіть алгоритм для сортування цих даних у порядку зростання або спадання. Порівняйте ефективність (кількість порівнянь) двох різних алгоритмів сортування (наприклад, бульбашкового сортування та сортування вставками) для заданого набору даних.

3.14 Пошук екстремуму функції: Задано дискретну функцію, представлену набором точок (x, y). Розробіть алгоритм для знаходження

точки з максимальним та мінімальним значенням y. Як можна модифікувати цей алгоритм для знаходження локальних екстремумів?

3.15 Алгоритм перетворення одиниць вимірювання: Напишіть алгоритм, який на вхід отримує значення фізичної величини, її поточну одиницю вимірювання та бажану одиницю вимірювання. Алгоритм повинен виконати перетворення та вивести результат. Розгляньте приклади перетворень між одиницями довжини (метри, сантиметри, дюйми) або маси (кілограми, грами, фунти).

3.16 Лінійна інтерполяція: Задано дві точки на площині (x_1, y_1) та (x_2, y_2). Напишіть алгоритм, який за заданим значенням x (що знаходиться між x_1 та x_2) обчислює відповідне значення y за допомогою лінійної інтерполяції.

3.17 Обчислення площі під кривою методом прямокутників: Задано функцію $f(x)$ на певному інтервалі $[a, b]$. Розробіть алгоритм для наближеного обчислення площі під кривою за допомогою методу прямокутників (розбиття інтервалу на n рівних частин та сумування площ прямокутників).

3.18 Розв'язання системи лінійних рівнянь (метод Гауса): Опишіть основні кроки алгоритму Гауса для розв'язання системи лінійних алгебраїчних рівнянь. (Можна обмежитися системою 2x2 або 3x3 для спрощення).

3.19 Алгоритм цифрової фільтрації (простий ковзний середній): Задано послідовність виміряних значень сигналу. Напишіть алгоритм для застосування простого ковзного середнього фільтра з заданим вікном (кількістю попередніх значень для усереднення).

3.20 Алгоритм керування температурою в термостаті: Розробіть алгоритм для керування нагрівальним елементом на основі заданої цільової температури та поточного значення температури, отриманого від датчика. Врахуйте гістерезис (діапазон температур, в якому нагрівання не вмикається і не вимикається).

Тести для самоконтролю

1. Що таке алгоритм?
   a) набір комп'ютерних програм для виконання конкретного завдання;
   b) точний та скінченний набір інструкцій, що визначає послідовність дій для розв'язання певної задачі;
   c) будь-яка послідовність дій, що призводить до бажаного результату;
   d) математична формула для обчислення значення.
2. Яка з наведених властивостей НЕ є обов'язковою для алгоритму?
   a) скінченність;
   b) визначеність;
   c) результативність;
   d) простота реалізації на конкретній мові програмування.
3. Які основні типи алгоритмічних структур ви знаєте?
   a) лінійні, циклічні, розгалужені;

b) послідовні, умовні, повторювальні;
c) прості, складні, комбіновані;
d) математичні, логічні, символьні.

4. Що таке блок-схема алгоритму?
   a) текстовий опис алгоритму на псевдокоді;
   b) графічне представлення алгоритму у вигляді послідовності блоків, з'єднаних стрілками;
   c) список команд для комп'ютера;
   d) словесний опис кроків розв'язання задачі.
5. Що таке псевдокод?
   a) спеціальна мова програмування для опису алгоритмів;
   b) наближений до мови програмування опис алгоритму, який легко читається людиною;
   c) машинний код алгоритму;
   d) графічне представлення алгоритму.
6. Яка структура алгоритму передбачає послідовне виконання інструкцій одна за одною?
   a) розгалужена;
   b) циклічна;
   c) лінійна;
   d) рекурсивна.
7. Яка структура алгоритму дозволяє виконувати різні набори інструкцій залежно від виконання певної умови?
   a) розгалужена;
   b) циклічна;
   c) лінійна;
   d) ітераційна.
8. Які основні типи циклів існують в алгоритмах?
   a) прості та складні;
   b) з лічильником та з умовою;
   c) внутрішні та зовнішні;
   d) початкові та кінцеві.
9. Що таке тіло циклу?
   a) умова виконання циклу;
   b) змінна, яка використовується для контролю кількості ітерацій;
   c) набір інструкцій, які повторюються під час виконання циклу;
   d) перша інструкція в циклі.
10. Яка структура алгоритму використовується для повторення певної послідовності дій до тих пір, поки не буде виконана задана умова?
    a) лінійна; b) розгалужена; c) циклічна; d) умовна.
11. Який алгоритм найкраще підходить для пошуку певного елемента в невпорядкованому списку?
    a) двійковий пошук;
    b) сортування бульбашкою;
    c) лінійний пошук;

d) сортування злиттям.

12. Який алгоритм використовується для впорядкування елементів списку за певним критерієм?
    a) пошуковий алгоритм;
    b) алгоритм сортування;
    c) алгоритм обходу графа;
    d) алгоритм стискання даних.
13. Наведіть приклад задачі, для розв'язання якої доцільно використовувати розгалужений алгоритм:
    a) обчислення суми елементів масиву;
    b) визначення, чи є введене число додатним, від'ємним або нулем;
    c) виведення послідовності чисел від 1 до 10;
    d) обчислення площі прямокутника за заданими сторонами.
14. Наведіть приклад задачі, для розв'язання якої доцільно використовувати циклічний алгоритм:
    a) порівняння двох чисел;
    b) обчислення факторіала заданого числа;
    c) визначення найбільшого з трьох чисел;
    d) перевірка, чи є число простим.
15. Який етап розробки алгоритму включає перевірку його правильності на різних наборах вхідних даних?
    a) опис алгоритму;
    b) реалізація алгоритму;
    c) тестування алгоритму;
    d) аналіз складності алгоритму.

РОЗДІЛ 5. ОСНОВИ МОВИ ПРОГРАМУВАННЯ PYTHON

Python – це експеримент з визначення ступеня свободи програмістів. Занадто багато свободи, і ніхто не може читати чужий код; занадто мало, і виразність знаходиться під загрозою.
(Гвідо ван Россум)

Мета: розглянути базові принципи комп'ютерного програмування та їх застосування в мові Python; вивчити основні елементи мови програмування Python, включаючи типи даних, оператори, структури управління та базові операції вводу-виводу; закріпити навички структурного підходу до програмування шляхом написання простих програм на Python.

План

1. Введення в Python.
2. Основні конструкції мови.
3. Базовий синтаксис функцій.
4. Основи графічних інтерфейсів (GUI) і створення віконних додатків.

Ключові терміни: *мова програмування, Python, алгоритм, програма, синтаксис, структура програми: організація коду програми, типи даних, змінні, оператори, введення-виведення, структурний підхід, інструкція, вираз.*

5.1 Введення в Python та комп'ютерне програмування

Історія мови

Основний автор мови програмування Python – Гвідо ван Россум. Перші версії мови з'явилися в 1991 р. Основних версій на даний момент три:

- Python 1.0 – з'явився в 1994 р.;
- Python 2.0 – 2000 р.;
- Python 3.0 – 2008 р.

Особливості мови

Особливості мови програмування Python наступні:

- вільно поширюється;
- з відкритим вихідним кодом – можна розробляти свої модулі, а також вносити пропозиції щодо покращення;
- переносний (кросплатформовий) – є можливість інтерпретувати Python на багатьох платформах та архітектурах, у тому числі використовувати його для написання програм управління пристроями (прошивок);
- має непогану стандартну бібліотеку.

З точки зору класифікації мова відноситься до наступних категорій:

- високорівнева;
- інтерпретована – скрипти на Python компілюються в деякі високорівневі інструкції (байт-код), які послідовно виконуються інтерпретатором;

- мультипарадигменна – побудована на принципах об'єктно-орієнтованого програмування та реалізує його основні механізми у вигляді успадкування, інкапсуляції та поліморфізму;
- жорстка (сильна, strong) динамічна типізація – з одного боку, змінна пов'язується з типом під час присвоювання значення, а не в момент оголошення змінної, з іншого боку – помилки, пов'язані з неправильним використанням типів, запобігають під час виконання програми (в run-time викидається виняток), а також допускається деяке неявне перетворення типів (наприклад, приведення чисел або перетворення до булевого типу).

Ефективність

З точки зору швидкості розробки Python характеризується зрозумілістю коду:

- код легко писати – синтаксис максимально спрощений;
- код легко читати – ідеологія коду, написаного на мові Python, така, що є можливість читати його як текст на природній мові;
- код легко налагоджувати – є можливість подивитися значення практично будь-якої змінної, а також стан стека викликів і присутні вбудовані засоби налагодження у вигляді дебагера;
- для розробки є офіційний посібник зі стилю коду (style guide; PEP-81, PEP-2572).

З точки зору продуктивності він повільніший за багато мов. Наприклад, програма, що виконує прості операції з базовими типами (числами, рядками), інтерпретована «чистим» Python без додаткових оптимізацій, повільніша за аналогічну програму, написану мовою C++, в 5-100 разів.

Основні напрямки розробки на Python

Python – це універсальна мова програмування, яка може бути використана в багатьох галузях, а саме:

- Прототипування: Якщо є необхідність порівняти певну кількість підходів до розробки або деякі архітектури, досить просто написати прототип на Python.
- Інтеграція, реалізація допоміжних програм: Прості програми для обробки текстів, перетворення форматів, дій з великою кількістю файлів легко реалізувати на Python через велику стандартну бібліотеку та наявність великої кількості сторонніх бібліотек.
- Дослідження: На Python можна відносно швидко кодувати складні математичні обчислення, одночасно зберігаючи та інтерпретуючи проміжні результати, працювати з різними форматами даних.
- Web-сервіси та застосунки: Синтаксис мови Python дозволяє максимально спростити та прискорити розробку web-застосунків, оскільки написання всього застосунку зводиться до чіткої та диференційованої реалізації окремих його компонентів, причому код залишається легко читабельним. Мова програмування Python є зручною обгорткою над багатьма низькорівневими примітивами: дозволяє їх однаково використовувати та не думати про роботу з пам'яттю.

- Аналіз даних та машинне навчання: Завдяки великій кількості бібліотек, таких як NumPy, Pandas, Scikit-learn, TensorFlow, Python дуже широко використовується для аналізу даних та машинного навчання.
- Автоматизація завдань (скрипти): Python часто використовується для автоматизації рутинних завдань, таких як обробка файлів, системне адміністрування та веб-парсинг.
- Розробка ігор: З допомогою бібліотеки Pygame, можливо розробляти 2D ігри.

Ідеологія Python

У Python є звід правил та ідей, якими рекомендується керуватися, так званий Python Zen. Якщо ви наберете в інтерпретаторі наступний рядок, вам відкриється повний список.

```
>>> import this
```

Нижче наведено деякі з них, мабуть, одні з основних:

- *Красиве краще за потворне (beautiful is better than ugly)* – намагайтеся код писати виразно, дотримуючись чистоти, відступів, коректно називаючи змінні;
- *Просте краще за складне (simple is better than complex)* – намагайтеся писати максимально простий код без довгих функцій та без високої вкладеності;
- *Складне краще за заплутане (complex is better than complicated)* – якщо не вдається написати простий код, напишіть якісний складний, але так, щоб у ньому можна було розібратися;
- *Читабельність дуже важлива (readability counts)* – узгоджується з правилом про красиве та потворне; важливо, щоб код можна було досить швидко та легко сприймати;
- *Виняткові випадки не настільки виняткові, щоб порушувати правила (special cases aren't special enough to break the rules)* – оскільки мова Python динамічно типізована, є можливість писати досить загальні функції та умовні конструкції, що може значно спростити код;
- *Якщо реалізацію складно пояснити, то це погана ідея (if the implementation is hard to explain, it's a bad idea);*
- *Якщо реалізацію легко пояснити, це може бути гарна ідея (if the implementation is easy to explain, it may be a good idea).*

Версії Python

Існують дві офіційні та несумісні версії Python: Python 2.x та Python 3.x. Версія 3.x активно розвивається (на серпень 2019 р. це версії 3.7.4 та 3.8.0b3). Версія 2.x все ще підтримується (остання на вересень 2018 р. – версія 2.7.16), але закінчення її підтримки планується на початок 2020 р. У версію Python 2.x було перенесено деякі покращення, спочатку реалізовані для Python 3.x, однак ці покращення стосуються скоріше бібліотек. Варто зазначити, що деякі основні версії Python, наприклад Python 3.6 та Python 3.7, несумісні між собою.

Інтерпретатори

Оскільки мова програмування Python скриптова, для її виконання написані різні інтерпретатори. Вони інтерпретують скрипти, написані на

Python, у вигляді конструкцій на інших високорівневих мовах (наприклад, на C). Нижче представлені найпоширеніші інтерпретатори Python:

- *CPython* – офіційний інтерпретатор Python, написаний на мові програмування C;
- *Jython, IronPython* – інтерпретатори на Java та .Net; особливість їх у тому, що вони дозволяють розширити стандартну бібліотеку Python бібліотекою мови Java (Jython) або бібліотекою .Net (IronPython);
- *PyPy, Numba* – інтерпретатори, які дозволяють передкомпілювати деякі конструкції або функції (Just-in-Time compilers);
- *Cython, Nuitka* – інтерпретатори на базі CPython, але вони допускають вказівку типів, а також проводять деякі інші оптимізації для прискорення виконання програм;
- *Stackless Python* – реалізація інтерпретатора, що відрізняється специфічною реалізацією багатопотоковості;
- *Julia* – інша мова, в деякому роді надбудова над Python.

Середовища розробки

Зазвичай для розробки пропонується використовувати один із наступних інструментів:

- деякий текстовий редактор з підсвічуванням синтаксису та автодоповненням: SublimeText, Notepad++, Visual Studio Code;
- блокнот Jupyter [Notebook] / IPython [Notebook] – специфічний редактор тексту, що містить набір комірок, кожна з яких може містити або код, а також результат його виконання (наприклад, графік або результат обчислення формули), а також текст, що описує та пояснює код або процес експерименту;
- PyCharm (Community Edition) – одне з найпопулярніших та найзручніших середовищ розробки в 2018 р.;
- Visual Studio with plugin (native in VS2017) – класичне середовище розробки від компанії Microsoft.

Програми Python виконуються інтерпретатором. На комп'ютерах з системами Unix та Linux інтерпретатор можна викликати, набравши команду *python.* У системах Windows та Macintosh інтерпретатор можна запустити як програму (або з меню Start, або подвійним клацанням на піктограмі інтерпретатора). Після запуску інтерпретатора з'являється підказка, в якій можна почати налагодження операторів програми в простому циклі читання/виконання. Вихідні файли Python мають розширення *.py. У системі Windows програми Python можна запускати подвійним клацанням на файлі з розширенням .py. При цьому відбувається запуск інтерпретатора та виконання програми у вікні термінала. У такому випадку вікно термінала негайно зникає після того, як програма завершує своє виконання (найчастіше перш ніж вдається прочитати її вивід). Щоб уникнути цього, можна скористатися середовищем інтегрованої розробки (Idle або Pythonwin).

5.2 Основні конструкції мови

Інтерпретатор можна запустити у звичайному режимі, тобто виконати за його допомогою вихідний код (запустити новий процес із програмою):

$ python main.py

Інтерпретатор можна запустити в інтерактивному режимі – інтерпретатор без коду:

$ python

Слід зазначити, що в інтерпретаторі доступні наступні глобальні функції:

- help(...) – довідка за вказаним об'єктом (вихід: q)
- exit() або quit() – вихід

Робота з числами в Python

У Python доступні наступні типи для роботи з числами:

- int – тип, що представляє ціле число довільної довжини;
- float – тип, що представляє число з плаваючою комою;
- complex – тип, що представляє комплексне число.

У стандартній бібліотеці також є можливість скористатися типами Fraction (раціональний дріб) та Decimal (число з плаваючою комою довільної точності).

Над типами доступні звичайні арифметичні операції: додавання (+), віднімання (-), множення (*), зведення в ступінь (**), ділення (звичайне "/" та цілочисельне "//"), взяття залишку від ділення (%) та функція, що виконує ділення з залишком (divmod).

Розглянемо особливості ділення на наступному прикладі:

```
>>> -2 * 7 // 3 + 4 ** (4 % 2)
```

-4

Тут при цілочисельному діленні частка є найближчим знизу до результату ділення цілим числом.

У наступному прикладі показано особливість оператора зведення в ступінь – він правоасоціативний:

```
>>> 3 ** 2 ** 3 == 3 ** (2 ** 3) == 6561
```

True

Оператор цілочисельного ділення працює таким чином:

```
>>> print(7 / 3, 7 / 3.0, 7 // 3.0)
2.33333333333 2.33333333333 2.0
```

Логічні вирази

У Python доступні порівняння об'єктів на рівність і нерівність (== та !=), визначення передування об'єктів (<, >, <=, >=), оператор логічного «і» (`and`), логічного «або» (`or`) та заперечення (`not`).

```
>>> 0 != 0
False
>>> 3 * 3 == 9
True
```

```
>>> False or 0 > -1 and True
True
>>> not 0 < 1 <= 5
False
```

Змінні

Оскільки мова Python динамічно типізована, оголошення змінної поєднане з її визначенням, а вказівка типу змінної не потрібна. Як і прийнято в інших мовах, до арифметичних типів застосовується неявне перетворення: у прикладі нижче ціле число при додаванні з числом з плаваючою комою приведено до типу останнього.

```
>>> x = 1
>>> y = 2.5
>>> x + y
3.5
>>> True and y
2.5
```

Обчислення складних логічних виразів відбувається ліниво: гілки обчислюються згідно з пріоритетами, і як тільки можна визначити, що весь вираз буде істинним або хибним, обчислення припиняється. У Python також допустимі скорочені оператори, які модифікують результат змінної:

```
>>> y /= 2 * x # y = 1.25
1.25
```

Коментарем у Python є код, написаний після символу «#».

Умовний оператор if

Умовний оператор if є фундаментальною конструкцією в Python, яка дозволяє виконувати різні блоки коду залежно від того, чи виконується певна умова.

Синтаксис:

`if` умова:

Блок коду, який виконується, якщо умова істинна `elif` умова2: # Необов'язкова гілка

Блок коду, який виконується, якщо умова2 істинна

`else`: # Необов'язкова гілка

Блок коду, який виконується, якщо всі попередні умови хибні

Пояснення:

1. `if` умова:
   - умова – це будь-який вираз, який повертає булеве значення (`True` або `False`).
   - Якщо умова істинна, виконується блок коду, що йде після двокрапки.
2. `elif` умова2: (`else if`):
   - `elif` дозволяє перевірити додаткові умови, якщо перша умова `if` хибна.

- Можна використовувати кілька блоків `elif` для перевірки різних умов.

3. `else::`
   - `else` виконується, якщо жодна з попередніх умов `if` або `elif` не є істинною.
   - `else` є необов'язковим, але рекомендується для обробки випадків, коли жодна з умов не виконується.

Приклад 1.

```
a = -5
if a < 0:
    b = -a
elif a > 0:
    b = a
else:
    b = 0
print(f"Абсолютне значення a: {b}")  # Виведе: Абсолютне значення a: 5
```

Python також підтримує тернарний оператор, який дозволяє записати умовний вираз в одному рядку:

```
b = a if a >= 0 else -a
```

pass – це оператор-заглушка, який нічого не робить. Він використовується, коли синтаксично потрібен блок коду, але ви не хочете нічого виконувати.

```
if a > 10:
  pass  # Нічого не робити, якщо a > 10
else:
    print("a <= 10")
```

У Python немає вбудованої конструкції switch, як у мові C. Замість цього використовується ланцюжок `if-elif-else` для досягнення того ж результату. Також, починаючи з Python 3.10, використовується оператор `match`.

Приклад 2. Написати програму, яка перевіряє, чи є задане число парним.

```
number = 10
if number % 2 == 0:
    print("Число парне")
else:
    print("Число непарне")
```

Приклад 3. Написати програму, яка визначає знак числа:

```
number = -3
if number > 0:
    print("Число додатне")
elif number < 0:
    print("Число від'ємне")
else:
    print("Число дорівнює нулю")
```

Приклад 4. Розглянемо оператор умовного розгалуження на прикладі реалізації функції, яка обчислює абсолютне значення величини

```
>>> if a < 0: # first way
>>>     b = -a
>>> elif a > 0:
>>>     b = a
>>> else:
>>>     b = 0
>>>
>>> b = a if a >= 0 else -a # second way
>>>
>>> b = abs(a) # third way
```

Цикл while

Цикл while використовується для багаторазового виконання блоку коду, поки задана умова залишається істинною. Це дозволяє створювати програми, які продовжують працювати, доки не буде досягнуто певного результату або виконана певна умова.

Синтаксис:

```
while умова:
    # Блок коду, який виконується, поки умова істинна
else: # Необов'язковий блок
    # Блок коду, який виконується, якщо цикл завершився без break
```

Пояснення:

1. `while` умова::
   - умова – це будь-який вираз, який повертає булеве значення (True або False).
   - Поки умова істинна, виконується блок коду, що йде після двокрапки.
2. `else`: (необов'язково):
   - Блок коду в `else` виконується лише тоді, коли цикл `while` завершився природним чином, тобто умова стала хибною, а не через `break`.

Керуючі інструкції

- break: негайно перериває виконання циклу, навіть якщо умова залишається істинною.
- continue: пропускає поточну ітерацію циклу та переходить до наступної.

Приклад 5. Зчитування чисел до введення 0

```
number = int(input("Введіть число (0 для завершення): "))
while number != 0:
    print("Ви ввели:", number)
    number = int(input("Введіть число (0 для завершення): "))
```

Приклад 6. *Гра «Вгадай число»*

```
import random
secret_number = random.randint(1, 100)
guess = None
attempts = 0
```

```
while guess != secret_number:
    guess = int(input("Вгадайте число від 1 до 100: "))
    attempts += 1
    if guess < secret_number:
        print("Занадто мало!")
    elif guess > secret_number:
        print("Занадто багато!")
print(f"Ви вгадали число {secret_number} за {attempts} спроб!")
```

Приклад 7. *Нескінченний цикл*

```
while True:
    user_input = input("Введіть команду (q для виходу): 
")
    if user_input == "q":
        break  # Вихід з нескінченного циклу
    print("Ви ввели:", user_input)
```

Приклад 8. *Використання else.*

```
count = 0
while count < 3:
    user_input = input("Введіть 'stop' для виходу: ")
    if user_input == "stop":
        break
    count += 1
else:
    print("Ви не ввели 'stop' за 3 спроби.")
```

Цикл `while` є потужним інструментом для створення програм, які реагують на умови та повторюють дії до досягнення потрібного результату.

Цикл for

Цикл for в Python використовується для ітерації по елементах послідовності (такої як список, кортеж, рядок) або по інших об'єктах, які підтримують ітерацію. Він виконує блок коду для кожного елемента в цій послідовності.

Синтаксис:

```
for змінна in послідовність:
    # Блок коду, який буде виконано для кожного елемента
    # Змінна буде послідовно приймати значення кожного 
елемента
```

Пояснення:

- `for`: Ключове слово, що позначає початок циклу.
- `змінна`: Змінна, яка буде послідовно приймати значення кожного елемента з послідовності під час кожної ітерації циклу. Ім'я змінної ви обираєте самі.
- `in`: Ключове слово, що вказує на те, що ми ітеруємося по елементах послідовності.
- `послідовність`: Об'єкт, що підтримує ітерацію (наприклад, список, кортеж, рядок, результат функції range()).
- `:`: Обов'язковий символ, що позначає початок блоку коду.

- `# Блок коду`: Одна або кілька інструкцій Python, які будуть виконані для кожного елемента в послідовності. Важливо пам'ятати про відступи (зазвичай 4 пробіли або 1 табуляція), які визначають тіло циклу.

Приклад 9. Ітерація по списку

```
programming_languages = ["Python", "JavaScript", "Java", "C++"]
# ітерація над кожним елементом всередині списку
for language in programming_languages:
  print(language)
# вивід
# Python
# JavaScript
# Java
# C++
```

Приклад 10. Ітерація по кортежу

```
my_info = ("John", "Doe", 26, "Software Engineer")
for data in my_info:
  print(data)
# вивід
# John
# Doe
# 26
# Software Engineer
```

Приклад 11. Ітерація по словнику

```
my_info = {
  'name':'John Doe',
  'job title':'software engineer',
  'country':'USA'
}

for info in my_info:
  print(info)
# name
# job title
# country
```

Використання функції range() для ітерації по числах

Функція `range(start, stop, step)` створює послідовність чисел.

- `range(stop)`: Від 0 до `stop` (не включно) з кроком 1.
- `range(start, stop)`: Від `start` до `stop` (не включно) з кроком 1.
- `range(start, stop, step)`: Від `start` до `stop` (не включно) з кроком `step`.

Приклад 12.

```
for число in range(5):  # Від 0 до 4
    print(число)
```

```
for число in range(2, 7):  # Від 2 до 6
    print(число)

for число in range(0, 10, 2):  # Від 0 до 9 з кроком 2
    print(число)
```

5.3 Базовий синтаксис функцій

Функції в Python – це блоки коду, які виконують певні завдання. Вони дозволяють організувати код, зробити його більш читабельним та багаторазово використовувати.

Основні характеристики функцій у Python:

- Функції визначаються за допомогою ключового слова `def`.
- Функція може приймати аргументи (параметри), які використовуються всередині функції.
- Функція може повертати значення за допомогою ключового слова `return`.
- Якщо функція не має оператора return або return без аргументів, вона повертає `None`.

Типи функцій:

1. Вбудовані функції:
   - Python має багато вбудованих функцій, таких як `print()`, `len()`, `max()`, `min()`, `type()` тощо.
   - Ці функції доступні для використання без імпорту додаткових модулів.
2. Функції користувача:
   - Розробники можуть створювати власні функції для виконання конкретних завдань.
   - Це дозволяє розбивати складні програми на менші, керовані блоки коду.

Синтаксис визначення функції:

```
def ім'я_функції(параметри):
    # Тіло функції
    # ...
    return значення  # Необов'язково
```

Приклад 13.

```
def додати_числа(a, b):
    сума = a + b
    return сума

результат = додати_числа(5, 3)
print(результат)  # Виведе: 8
```

Переваги використання функцій:

- Організація коду: Функції дозволяють розбивати код на логічні блоки, що полегшує його розуміння та підтримку.
- Багаторазове використання коду: Функції можна викликати кілька разів у програмі, що дозволяє уникнути дублювання коду.
- Читабельність коду: Використання функцій робить код більш читабельним та зрозумілим.
- Модульність: Функції дозволяють створювати модульний код, який можна легко модифікувати та розширювати.

Аргументи функцій:

- Функції можуть приймати позиційні аргументи, ключові аргументи та аргументи зі значеннями за замовчуванням.
- Позиційні аргументи передаються у функцію в тому порядку, в якому вони визначені.
- Ключові аргументи передаються у функцію з вказівкою імені параметра.
- Аргументи зі значеннями за замовчуванням мають значення, які використовуються, якщо аргумент не передано при виклику функції.

Повернення значень:

- Функція може повертати одне або кілька значень.
- Якщо функція повертає кілька значень, вони повертаються у вигляді кортежу.

Приклад 14. Функція для обчислення факторіала числа. Функція використовує рекурсію для обчислення факторіала заданого числа.

```
def factorial(n):
    """Обчислює факторіал числа n."""
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

result = factorial(5)
print(result)  # Виведе: 120
```

Приклад 15. Функція для перевірки, чи є число простим. Ця функція перевіряє, чи ділиться задане число на будь-яке число від 2 до квадратного кореня з *n*.

```
def is_prime(n):
  """Перевіряє, чи є число n простим."""
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

```
print(is_prime(17))  # Виведе: True
print(is_prime(18))  # Виведе: False
```

Приклад 16. Функція для обчислення середнього значення списку чисел.

```
def average(numbers):
    """Обчислює середнє значення списку чисел."""
    if not numbers:
        return 0
    return sum(numbers) / len(numbers)

numbers = [1, 2, 3, 4, 5]
print(average(numbers))  # Виведе: 3.0
```

Приклад 17. Функція для форматування імені.

```
def format_name(first_name, last_name):
    """Форматує ім'я та прізвище."""
    return                    f"{first_name.capitalize()}
{last_name.capitalize()}"

print(format_name("john", "doe"))  # Виведе: John Doe
```

5.4 Основи графічних інтерфейсів (GUI) і створення віконних додатків

У сучасному світі програмне забезпечення стало невід'ємною частиною нашого життя. Від складних інженерних систем до простих утиліт, більшість програм взаємодіють з користувачем через графічний інтерфейс користувача (GUI). GUI забезпечує візуальне представлення програми, дозволяючи користувачам інтуїтивно взаємодіяти з її функціоналом за допомогою графічних елементів, таких як вікна, кнопки, меню, значки та інші.

На відміну від ранніх текстових інтерфейсів командного рядка (CLI), де користувачі повинні були вводити текстові команди для виконання дій, GUI значно спрощує взаємодію, роблячи програми більш доступними та зручними для широкого кола користувачів, незалежно від їхнього рівня технічної підготовки.

5.4.1 Основні елементи графічного інтерфейсу користувача

Для розуміння принципів створення віконних додатків необхідно ознайомитися з основними елементами GUI:

- *Вікна (Windows)*: Основні контейнери, в яких розміщуються інші елементи інтерфейсу та відображається інформація. Вікна можуть бути головними (що представляють основну програму) або діалоговими (що використовуються для відображення повідомлень або збору додаткової інформації від користувача).
- *Меню (Menus)*: Систематизовані списки команд, доступні користувачеві через панель меню у верхній частині вікна або контекстні меню, що з'являються при натисканні правою кнопкою миші.

- *Кнопки (Buttons)*: Елементи керування, які ініціюють певну дію при натисканні користувачем.
- *Текстові поля (Text Fields)*: Елементи для введення та відображення текстової інформації.
- *Мітки (Labels)*: Статичний текст, який використовується для позначення інших елементів інтерфейсу або відображення інформаційних повідомлень.
- *Списки (Lists) та випадаючі списки (Dropdown Lists):* Елементи для відображення переліку опцій, з яких користувач може обрати одну або кілька.
- *Прапорці (Checkboxes) та радіокнопки (Radio Buttons):* Елементи для вибору однієї або кількох опцій з набору.
- *Повзунки (Sliders):* Елементи для вибору значення в певному діапазоні.
- *Значки (Icons):* Графічні зображення, що символізують певні дії, програми або типи файлів.

Принципи створення віконних додатків

Створення віконних додатків передбачає використання спеціальних інструментів та бібліотек, які надають програмісту набір готових компонентів та механізмів для побудови GUI. Процес створення зазвичай включає наступні етапи:

1. *Вибір інструментарію (Toolkit) або фреймворку:* Існує безліч бібліотек та фреймворків для створення GUI, таких як Qt, Tkinter (для Python), Swing та JavaFX (для Java), Windows Forms та WPF (.NET) та інші. Вибір залежить від мови програмування, вимог до проекту та вподобань розробника.
2. *Створення головного вікна:* Першим кроком є створення основного вікна програми, яке слугуватиме контейнером для інших елементів інтерфейсу.
3. *Розміщення елементів керування (віджетів):* На головне вікно та інші контейнери додаються необхідні елементи GUI (кнопки, текстові поля, меню тощо).
4. *Визначення компонування (Layout Management):* Для забезпечення правильного та адаптивного розміщення елементів у вікні використовуються менеджери компонування. Вони автоматично змінюють розташування та розміри віджетів при зміні розміру вікна або інших подій.
5. *Обробка подій (Event Handling):* Ключовим аспектом GUI-програмування є обробка подій, таких як натискання кнопок, введення тексту, вибір елементів меню тощо. Програміст визначає, які дії повинні виконуватися програмою у відповідь на ці події.
6. *Реалізація функціональності програми:* Логіка програми пов'язується з елементами GUI через механізми обробки подій.

Значення GUI для інженерних додатків

У інженерній сфері GUI відіграють важливу роль, забезпечуючи зручний та інтуїтивно зрозумілий інтерфейс для складного програмного забезпечення, що використовується для:

- *CAD/CAM/CAE систем:* Візуалізація тривимірних моделей, керування процесами проектування та моделювання.
- *Систем збору та обробки даних:* Відображення графіків, діаграм, таблиць та іншої інформації, отриманої з датчиків та вимірювальних пристроїв.
- *Систем керування та автоматизації:* Візуалізація стану технологічних процесів, надання інструментів для моніторингу та контролю.
- *Інструментів аналізу та симуляції:* Відображення результатів розрахунків, анімацій та візуалізацій.

Ефективно розроблений GUI може значно підвищити продуктивність інженера, зменшити кількість помилок та спростити роботу зі складними даними та інструментами.

5.4.2 Створення графічних інтерфейсів Python за допомогою Tkinter

Tkinter є стандартною бібліотекою Python для створення GUI на основі бібліотеки Tk. Tkinter надає набір інструментів (віджети) для створення різноманітних елементів GUI, таких як вікна, кнопки, текстові поля, мітки тощо. Розглянемо основні кроки та приклади:

1. Імпорт модуля Tkinter

Першим кроком є імпорт модуля `tkinter`:

```
import tkinter as tk
```

Зазвичай його імпортують під псевдонімом `tk` для зручності.

2. Створення головного вікна

Кожен додаток `Tkinter` починається зі створення головного вікна (`Tk object`):

```
root = tk.Tk()
root.title("Мій перший GUI додаток") # Заголовок вікна
```

Тут `tk.Tk()` створює основне вікно програми, а `root.title()` встановлює заголовок цього вікна.

3. Додавання віджетів

Для створення інтерактивного інтерфейсу до головного вікна додаються різні віджети. Ось приклади найпоширеніших:

Мітка (Label) Для відображення тексту

Python

```
Label = tk.Label(root, text="Привіт, Tkinter!")
label.pack() # Розміщення віджета у вікні
```

`tk.Label()` створює віджет мітки, де `root` вказує, що мітка розміщується у головному вікні, а `text` задає текст мітки. Метод `pack()` є одним із менеджерів компонування, який розміщує віджет у вікні з базовими параметрами.

Кнопка (Button) Для виконання дії при натисканні

```
def button_click():
    print("Кнопку натиснуто!")
```

```
button = tk.Button(root, text="Натисни мене",
command=button_click)
button.pack()
```

`tk.Button()` створює кнопку. `text` задає текст на кнопці, а `command` вказує на функцію (`button_click` у цьому випадку), яка буде викликана при натисканні кнопки.

Текстове поле (Entry) Для введення однорядкового тексту

```
entry = tk.Entry(root)
entry.pack()

def get_text():
    user_input = entry.get()
    print(f"Введено: {user_input}")

get_button = tk.Button(root, text="Отримати текст",
command=get_text)
get_button.pack()
```

`tk.Entry()` створює текстове поле. Метод `get()` використовується для отримання тексту, введеного користувачем.

4. Менеджери компонування

Для організації розміщення віджетів у вікні Tkinter пропонує кілька менеджерів компонування:

- `pack()`: Найпростіший менеджер, розміщує віджети один за одним або поруч, залежно від параметрів (`side, fill, expand` тощо).
- `grid()`: Розміщує віджети у вигляді таблиці (сітки) з рядків та стовпців, використовуючи параметри `row` та `column`.
- `place()`: Дозволяє точно задавати положення та розміри віджетів за допомогою координат `x, y, width` та `height`.

Вибір менеджера залежить від складності інтерфейсу та необхідності гнучкого розміщення елементів.

5. Запуск головного циклу

Для відображення вікна та обробки подій (натискань кнопок, введення тексту тощо) необхідно запустити головний цикл `Tkinter`:

```
root.mainloop()
```

Цей метод блокує виконання програми та чекає на події GUI.

Приклад 18. Приклад простого віконного додатка з міткою та кнопкою:

```
import tkinter as tk

def button_click():
    label.config(text="Кнопку натиснуто!") # Зміна тексту
мітки

root = tk.Tk()
root.title("Простий додаток")

label = tk.Label(root, text="Початковий текст")
```

```
label.pack(pady=20) # Додавання вертикального відступу

button = tk.Button(root, text="Натисни мене",
command=button_click)
button.pack()

root.mainloop()
```

У наведеному прикладі при натисканні кнопки текст мітки змінюється. `label.config()` використовується для зміни властивостей вже існуючого віджета. `pady` додає вертикальний відступ навколо віджета при використанні `pack()`.

Приклад 19. Створення складніших інтерфейсів

Для створення складніших інтерфейсів можна комбінувати різні віджети, використовувати різні менеджери компонування, а також організовувати віджети у фрейми (`tk.Frame`), які слугують контейнерами для групування інших віджетів. Наприклад, використовуючи `grid()` для більш структурованого розміщення.

```
import tkinter as tk

root = tk.Tk()
root.title("Додаток з сіткою")

label_name = tk.Label(root, text="Ім'я:")
label_name.grid(row=0, column=0, sticky="w") # Розміщення
в рядку 0, стовпці 0, вирівнювання по західному краю

entry_name = tk.Entry(root)
entry_name.grid(row=0, column=1)

label_age = tk.Label(root, text="Вік:")
label_age.grid(row=1, column=0, sticky="w")

entry_age = tk.Entry(root)
entry_age.grid(row=1, column=1)

button_submit = tk.Button(root, text="Підтвердити")
button_submit.grid(row=2, column=0, columnspan=2) #
Займає 2 стовпці

root.mainloop()
```

Тут `sticky="w"` вказує на вирівнювання віджета по західному краю комірки сітки, а `columnspan=2` змушує кнопку займати дві колонки.

Tkinter є потужним інструментом для створення GUI-додатків на Python, особливо для невеликих та середніх проектів, а також для навчальних цілей. Для більш складних та візуально насичених інтерфейсів можуть використовуватися інші бібліотеки, такі як PyQt або Kivy.



Питання для самоконтролю

1. Що таке Python і які його основні особливості?
2. Які переваги та недоліки Python як мови програмування?
3. Де найчастіше застосовується мова Python?
4. Як встановити Python на різних операційних системах?
5. Що таке інтерпретатор Python та як він працює?
6. Які існують основні версії Python та їх відмінності?
7. Що таке «Zen of Python» і як його застосовувати?
8. Які основні типи даних існують в Python?
9. Як оголошуються та використовуються змінні в Python?
10. Які існують оператори в Python (арифметичні, логічні, порівняння)?
11. Як використовуються умовні оператори (if, elif, else)?
12. Як працюють цикли for та while в Python?
13. Що таке інструкції break та continue і як вони використовуються?
14. Як обробляти винятки за допомогою try, except та finally?
15. Що таке списки, кортежі, словники та множини в Python і як з ними працювати?
16. Як визначити функцію в Python?
17. Як передавати аргументи у функцію?
18. Як повернути значення з функції?
19. Що таке аргументи за замовчуванням та як їх використовувати?
20. Що таке локальні та глобальні змінні?
21. Що таке рекурсія та як її використовувати у функціях?

Індивідуальні завдання

Частина 1. Програмування лінійного обчислювального процесу

1. Напишіть програму, яка отримує три цілих числа, введені з клавіатури (кожне число вводиться на окремому рядку), і друкує на екрані їх суму, добуток, результат піднесення першого числа до степеня різниці другого і третього чисел.

2. Напишіть програму, яка приймає ціле число *n* і обчислює значення виразу *n + nn + nnn.*

3. Напишіть програму, яка отримує такі дані: ім’я, вік, хобі, введені з клавіатури (вводяться на окремих рядках), і друкує на екрані одним повідомленням повну інформацію на основі введених даних.

4. Напишіть програму, яка зчитує довжину основи та висоту прямокутного трикутника (цілі числа), обчислює площу і друкує її значення на екрані у відформатованому вигляді (два символи після десяткової крапки). Кожен параметр вводиться на окремому рядку.

5. Напишіть програму, яка зчитує ціле число, і друкує попереднє та наступне числа відносно введеного.

6. Вводиться ціле додатне число. Надрукуйте останню цифру числа.

7. Напишіть програму, яка приймає ім’я та прізвище від користувача у різних рядках та роздруковує їх у зворотному порядку з пропуском між ними у вітальному повідомленні.

8. Вводиться ціле невід’ємне число n ($n \leq 100$). Виведіть 2^n.

9. Напишіть програму, яка вітає користувача, виводячи слово *Hello*, введене ім’я і розділові знаки за зразком (дивитися приклади вхідних і вихідних даних). Програма повинна зчитувати в рядкову змінну значення і виводити відповідне вітання. Зверніть увагу, що після коми повинен обов’язково стояти пропуск, а перед знаком оклику пропуску немає. Операцією конкатенації (об’єднанням) рядків (+) користуватися не можна.

10. Дано двоцифрове число. Знайдіть число десятків у ньому.

11. Напишіть програму для друку чисел, у яких розділювачами груп розрядів (групи по три цифри) є коми.

12. Напишіть програму, яка отримує значення радіуса кола (дробове число) від користувача та обчислює площу круга і довжину колу. Виведення результату відбувається з трьома символами після десяткової крапки. Значення числа Пі округлити до сотих.

13. Напишіть програму для запису і розв’язування виразу $(a + b) * (a + b)$, де a і b — натуральні цілі числа.

14. Напишіть програму для перетворення висоти (вказується окремо у футах (1 фут = 30,48 см) і дюймах (1 дюйм = 2,54 см) у сантиметри.

15. Напишіть програму для перетворення відстані (вказана у футах) у дюйми (1 фут = 12 дюймів), ярди (1 фут = 0,333333333 ярда) і милі (1 фут = 0,000189393939 милі).

16. Напишіть програму, щоб отримати ASCII значення введеного з клавіатури символа.

17. Напишіть програму, щоб вивести символ за введеним ASCII значенням.

18. Користувач вводить значення чисел a і b з клавіатури. Напишіть програму для друку дії додавання і результату обчислення як у вихідних даних.

19. Напишіть програму, яка отримує три цифри і друкує їх суму. Кожна цифра вводиться на окремому рядку.

20. Напишіть програму, щоб конвертувати усі введені користувачем одиниці часу (дні, години, хвилини, секунди) в загальну кількість секунд.

21. Необхідно певну суму грошей залишити на банківському рахунку, щоб отримувати відсотки по закінченню певного часу. Наприклад, через 10 років ви хотіли б отримати 10 000 доларів США на власному рахунку. Скільки потрібно внести грошей на депозит сьогодні, щоб це сталося? Для розв’язування цієї задачі можна використовувати формулу $p = f/(1+r)*n$, де p – це поточна сума, яку потрібно внести сьогодні, f – майбутнє значення, яке ви хочете отримати (у нашому випадку становить 10 000 доларів США), r – річна відсоткова ставка, n – кількість років перебування грошей на рахунку.

22. Вивести на екран три цілих числа в один рядок через пропуск у порядку, зворотному введенню чисел.

23. Дано два цілих додатних числа a і b, які не перевищують 1000. Обчисліть і виведіть гіпотенузу трикутника із заданими катетами.

24. Вводяться три додатних числа – довжини сторін трикутника. Обчисліть площу трикутника за формулою Герона.

25. Напишіть програму, яка виконує усі арифметичні дії і піднесення до степеня з двома змінними a і b в такому порядку: додавання, віднімання, множення, ділення, піднесення до степеня.

Частина 2. Програмування розгалуженого обчислювального процесу

Розв'язати задачі, використовуючи алгоритми з розгалуженням – алгоритми, у яких, залежно від виконання деякої умови, виконується одна або інша послідовність команд.

1. У чемпіонаті з футболу команді за виграш дається 3 очка, за програш – 0, за нічию – 1. Відомо кількість очок, отриманих командою за гру. Вивести результат гри у вигляді відповідних слів: «виграш», «програш» або «нічия».

2. Напишіть програму, в якій користувач вводить значення температури, і, якщо це значення менше або дорівнює 0 градусів Цельсія, необхідно вивести повідомлення A cold, isn't it?. Якщо ж температура становить більше 0 і менше 10 градусів Цельсія повідомлення буде Cool, у інших випадках Nice weather we're having.

3. Дано радіус кола і сторона квадрата (дійсні числа). У якої фігури площа більше?

4. Серед учнів школи проводилося тестування з трьох предметів, по кожному з яких учні отримали певну кількість балів (цілі числа). Напишіть програму, яку можуть використати учні для обчислення їхнього середнього балу трьох тестів і виведення середнього значення. Окрім того, необхідно передбачити виведення повідомлення Congratulations! That is a great average!, якщо середній бал більший ніж 95.

5. Напишіть програму, на вхід якої подається два цілих числа - вік Сашка і вік Тетянки. Програма має вивести повідомлення про те, хто є старшим серед них.

6. Напишіть програму, яка запитує два цілих числа. Якщо добуток чисел перевищує їх суму, надрукувати добуток чисел, у протилежному випадку - вивести їх суму. Якщо добуток дорівнює сумі, вивести різницю чисел.

7. Напишіть програму, щоб перевірити, чи є введене число додатним, від'ємним або це нуль.

8. Напишіть програму, яка зчитує два цілих числа a і b (від 1 до 1000) та виводить найбільше значення з них.

9. Дано три цілих числа. Визначте, скільки серед них збігаються. Програма повинна вивести одне з чисел: 3 (якщо усі однакові), 2 (якщо два однакові) або 0 (якщо усі числа різні).

10. В університеті використовується наступна шкала для інтерпретації результатів тестування студентів: 90 балів і вище (A), 80-89 (B), 70-79 (C), 60-69

(D), нижче 60 (F). Напишіть програму, яка дозволить студенту ввести тестовий бал, а потім відобразити оцінку для цього балу.

11. Автомобіль подолав відстань *s* км через населений пункт за *t* хв. Визначити, чи не порушив водій правил дорожнього руху, якщо швидкість автомобіля при цьому не повинна перевищувати 60 км/год.

12. Користувач вводить значення температури *t* у градусах Цельсія. Напишіть програму, яка виводить назву стану, у якому знаходиться вода при цій температурі за нормальних умов.

13. Дано п’ять цілих чисел *a, b, c, d, e*. Визначити, скільки з них додатних.

14. Напишіть програму, яка запитує користувача номер в діапазоні від 1 до 7. Програма повинна відображати відповідний день тижня, де 1 – це понеділок, а 7 – неділя. Програма має враховувати варіант, коли користувач вводить номер, що знаходиться за межами діапазону від 1 до 7.

15. Напишіть програму, яка просить користувача ввести вік людини (ціле число). Програма повинна вивести повідомлення про те, чи є особа немовлям, дитиною, підлітком або дорослим за такими правилами: якщо людині 1 рік або менше, він або вона є немовлям, якщо особа старше 1 року, але молодше 23 років, вона є дитиною, якщо особа не молодше 23 років, але молодше 20 років, вона є підлітком і якщо особа віком старше 20 років, вона є дорослою.

16. Напишіть програму, на вхід якої подається назва місяця, а програма виводить кількість днів у ньому. Врахуйте те, що місяць «Лютий» може мати 28 або 29 днів.

17. Напишіть програму виведення текстового варіанта шкільних оцінок: 1, 2, 3 (початковий рівень – initial level), 4, 5, 6 (середній рівень – average level), 7, 8, 9 (достатній рівень – sufficient level), 10, 11, 12 (високий рівень – high level).

18. Розробіть програму для виведення інформації про операційну систему, яка встановлена на комп’ютері користувача, при введенні відповідного номера: 1 – Window, 2 – Linux, 3 – MacOS. Передбачте у роботі програми ситуації, коли користувач вводить інший номер, відмінний від наведених, або програма на вхід отримує порожній рядок.

19. Напишіть програму, яка виводить назви введених чисел. Користувач вводить ціле число. Якщо це число або 1 або 2 або 3, то виводиться повідомлення – назва числа, відповідно, One, Two, Three. В усіх інших випадках виводиться слово Unknown.

20. Перевести число, введене користувачем, у байти або кілобайти в залежності від його вибору. Користувач вводить значення у мегабайтах і, відповідно, вводить напрямок переведення: B (у байти) або KB (у кілобайти).



Завдання для самостійного виконання

Створення простого GUI додатку на Tkinter

Загальні вимоги до всіх завдань:

- *Графічний інтерфейс:* Додаток повинен мати простий та інтуїтивно зрозумілий графічний інтерфейс на основі Tkinter.
- *Введення даних:* Користувач повинен мати можливість вводити необхідні дані через текстові поля або інші елементи керування.
- *Відображення результатів:* Результати розрахунків або перевірок повинні чітко відображатися на екрані.
- *Обробка помилок (базово):* Додаток повинен передбачати прості механізми обробки некоректного введення даних (наприклад, спроба ввести нечислове значення там, де очікується число).
- *Збереження результатів (за вимогою в завданні):* Забезпечити можливість збереження введених даних та результатів у простий файл (текстовий або CSV).
- *Коментарі до коду:* Код програми повинен бути добре прокоментований.

1. Розрахунок індексу теплового стресу (WBGT)

Розробити функцію мовою програмування Python, яка обчислює індекс теплового стресу WBGT на основі вхідних даних про температуру вологого термометра, температуру сухого термометра та температуру кулі. Індекс WBGT (Wet Bulb Globe Temperature) використовується для оцінки теплового навантаження на працівників. Він враховує температуру вологого термометра, температуру сухого термометра та температуру кулі.

Напишіть функцію `calculate_wbgt(tw, tg, ta)` яка приймає три аргументи:

- `tw`: температура вологого термометра (°C).
- `tg`: температура кулі (°C).
- `ta`: температура сухого термометра (°C).

Функція повинна повертати значення індексу WBGT (°C) за наступною формулою для зовнішніх робіт на сонці: $WBGT=0.7\cdot tw+0.2\cdot tg+0.1\cdot ta$.

Протестуйте функцію з різними значеннями температур. Реалізуйте можливість вказувати одиниці вимірювання температур (наприклад, за допомогою ще одного аргументу) та передбачте їх конвертацію, якщо необхідно.

2. Визначення рівня шуму за еквівалентним рівнем звуку (Leq)

Розробити функцію мовою програмування Python, яка обчислює еквівалентний рівень звуку (Leq) на основі набору вимірювань рівня звуку та відповідних часових інтервалів їхньої дії.

Еквівалентний рівень звуку (Leq) є інтегральною характеристикою непостійного шуму та представляє собою рівень постійного шуму, який за певний період часу має таку ж саму енергію, що й досліджуваний непостійний шум. Для його розрахунку необхідно враховувати величину кожного рівня звуку та тривалість його дії.

$$L_{eq} = 10 \cdot lg\left(\frac{1}{T} \cdot \sum_{i=1}^{n} t_i \cdot 10^{0.1 \cdot L_i}\right),$$

де L_{eq} – еквівалентний рівень звуку (в дБА), T – загальний час вимірювання (сума всіх t_i), n – кількість вимірювань рівня звуку, L_i – i-тий виміряний рівень звуку (в дБА), t_i – час дії i-того рівня звуку (в секундах).

Напишіть функцію `calculate_leq(levels, durations)` яка приймає два аргументи:

- `levels`: список значень рівня звуку в дБА.
- `durations`: список значень часу дії кожного рівня звуку в секундах.

Протестуйте функцію з різними наборами рівнів звуку та тривалостей. Всередині функції реалізуйте логіку розрахунку Leq згідно з наведеною формулою. Переконайтеся, що ви правильно обчислюєте загальний час вимірювання (T) та сумуєте внески кожного рівня звуку з урахуванням його тривалості. Функція повинна повертати обчислене значення Leq у дБА (число з плаваючою комою). Напишіть невеликий приклад використання цієї функції, де ви створюєте списки значень рівнів звуку та їх тривалостей, передаєте їх до функції `calculate_leq` та виводите отриманий результат на екран.

Додайте до функції перевірку вхідних даних, щоб переконатися, що обидва списки мають однакову довжину і містять лише числові значення. Реалізуйте обробку можливих помилок, наприклад, якщо вхідні списки порожні. Розширте програму таким чином, щоб вона могла зчитувати дані про рівні шуму та їх тривалість з файлу (наприклад, у форматі CSV).

3. Перевірка допустимого рівня хімічної речовини в повітрі робочої зони

Для кожної хімічної речовини існують гранично допустимі концентрації (ГДК) в повітрі робочої зони.

Напишіть функцію `check_chemical_exposure (concentration, gdk)` яка приймає два аргументи:

- `concentration`: фактична концентрація хімічної речовини в повітрі (наприклад, мг/м³).
- `gdk`: гранично допустима концентрація цієї речовини (наприклад, мг/м³).

Функція повинна повертати повідомлення: «Концентрація в межах норми» якщо `concentration <= gdk`, і «Перевищення ГДК!» якщо `concentration > gdk`.

Протестуйте функцію з різними значеннями концентрацій та ГДК.

4. Класифікація умов праці за рівнем небезпеки

Створіть спрощену функцію `classify_work_conditions(noise_level, dust_level, temperature)` яка приймає три аргументи:

- `noise_level`: рівень шуму в дБА.
- `dust_level`: концентрація пилу в мг/м³.
- `temperature`: температура повітря в °C.

Функція повинна повертати рядок з класифікацією умов праці на основі наступних критеріїв (ці критерії є спрощеними та наведені для прикладу):

- Якщо рівень шуму > 80 дБА або концентрація пилу > 5 мг/м³ або температура > 30 °C: «Небезпечні умови праці».

- Інакше: «Допустимі умови праці».

Продемонструйте роботу функції з різними вхідними даними.

5. Розрахунок необхідної кількості повітря для вентиляції

Для забезпечення безпечних умов праці часто необхідно розраховувати об'єм повітря, який потрібно подавати для вентиляції приміщення.

Напишіть функцію `calculate_ventilation_rate(num_workers, volume_per_worker)` яка приймає:

- `num_workers`: кількість працівників у приміщенні.
- `volume_per_worker`: нормативний об'єм повітря на одного працівника (наприклад, м³/год).

Функція повинна повертати загальний необхідний об'єм повітря для вентиляції (м³/год).

Протестуйте функцію для різної кількості працівників та норм вентиляції.

Тести для самоконтролю

1. У якому році було створено мову програмування Python?
   a) 2000; b) 1990; c) 1975; d) 2010.
2. Хто створив мову програмування Python?
   a) Марк Цукерберг; b) Біл Гейтс; c) Гвідо ван Россум; d) Стів Возняк.
3. Яким буде вивід наступного коду?
   ```
   x = 5
   y = "привіт"
   print(x + y)
   ```
   a) 5привіт; b) привіт 5; c) Помилка; d) 5 привіт.
4. Який з наступних типів даних є незмінним (immutable) в Python?
   a) list (список); b) dictionary (словник); c) tuple (кортеж); d) set (множина).
5. Яким буде значення змінної `z` після виконання наступного коду?
   ```
   a = 10
   b = 3
   z = a // b
   ```
   a) 3.333...; b) 3; c) 3.0; d) 1.
6. Як перевірити тип даних змінної в Python?
   a) `typeof()`; b) `gettype()`; c) `type()`; d) `datatype()`.
7. Яким буде вивід наступного коду?
   ```
   name = "Світ"
   print(f"Привіт, {name}!")
   ```
   a) Привіт, {name}! b) Привіт, Світ!
   c) "Привіт, " + name + "!" d) помилка.
8. Яким буде вивід наступного коду?
   ```
   вік = 17
   if вік >= 18:
       print("Дорослий")
   else:
       print("Неповнолітній")
   ```
   a) дорослий; b) неповнолітній; c) немає виводу; d) помилка.

9. Скільки разів виконається наступний цикл?
```
for i in range(15):
    print(i)
```
a) 14; b) 15; c) 16; d) 0.

10. Яким буде вивід наступного коду?
```
x = 0
while x < 3:
    x += 1
print(x)
```
a) 0; b) 1; c) 2; d) 3.

11. Яке ключове слово використовується для дострокового виходу з циклу в Python?
a) `stop`; b) `exit`; c) `break`; d) `continue`.

12. Яким буде вивід наступного коду?
```
числа = [1, 2, 3, 4, 5]
for num in числа:
    if num == 3:
        continue
    print(num)
```
a) 1 2 3 4 5; b) 1 2 4 5; c) 3; d) немає виводу.

13. Як визначити функцію в Python?
a) `function моя_функція():`;
b) `def моя_функція():`;
c) `func моя_функція():`;
d) `define моя_функція():`.

14. Як імпортувати конкретну функцію sqrt з модуля math?
a) `import sqrt from math`;
b) `from math import sqrt`;
c) `import math.sqrt`;
d) `use math.sqrt`.

15. Яка мета оператора `return` у функції Python?
a) для виведення інформації на консоль;
b) для визначення імені функції;
c) для повернення значення з функції до того місця, де вона була викликана;
d) для позначення кінця визначення функції.

16. Що таке модуль в Python?
a) один рядок коду python;
b) збірка пов'язаних функцій та змінних, зазвичай збережених у файлі з розширенням .py;
c) ключове слово, що використовується для визначення циклу;
d) вбудований тип даних.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Баженов В. А. Інформатика. Комп'ютерна техніка. Комп'ютерні технології : підручник. 4-те вид. Київ : Каравела, 2023. 496 с.
2. Маляров М. В., Христич В. В., Журавський М. М. Основи інформаційних технологій : курс лекцій. Харків : НУЦЗУ, 2019. 184 с.
3. Нелюбов В. О., Куруца О. С. Основи інформатики. Microsoft Excel 2016 : навч. посіб. Ужгород : ДВНЗ «УжНУ», 2018. 58 с.
4. Нелюбов В. О., Куруца О. С. Основи інформатики. Microsoft PowerPoint 2016 : навч. посіб. Ужгород : ДВНЗ «УжНУ», 2018. 122 с.
5. Нелюбов В. О., Куруца О. С. Основи інформатики. Microsoft Word 2016 : навч. посіб. Ужгород : ДВНЗ «УжНУ», 2018. 96 с.
6. Костюченко А. О. Основи програмування мовою Python : навч. посіб. Чернігів : ФОП Баликіна С. М., 2020. 180 с.
7. Шаповалова М. І. Програмування на Python : метод. вказівки до виконання лаб. завдань. Харків : НТУ «ХПІ», 2024. 72 с.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Костюченко А. О. Основи програмування мовою Python : навч. посіб. Чернігів : ФОП Баликіна С. М., 2020. 180 с.
2. Шаповалова М. І. Програмування на Python : метод. вказівки до виконання лаб. завдань. Харків : НТУ «ХПІ», 2024. 72 с.
3. Архітектура, принципи функціонування та керування ресурсами IBM PC : навч. посіб. / А. М. Гуржій, С. Ф. Коряк, В. В. Самсонов, О. Я. Скляров. Харьков : Компанія СМІТ, 2003. 511 с.
4. Інформатика та комп’ютерна техніка : навч.-метод. посіб. / Д. О. Рзаєв, О. Д. Шарапов, В. М. Ігнатенко, Л. М. Дибкова. Київ : КНЕУ, 2012. 486 с.

Навчальне видання
(*українською мовою*)

Міхайлуца Олена Миколаївна
Пожуєв Андрій Володимирович

ІНФОРМАТИКА ТА КОМП'ЮТЕРНА ТЕХНІКА

Навчальний посібник
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Цивільна безпека»
освітньо-професійної програми «Охорона праці»

Рецензент *О. Ю. Небеснюк*
Відповідальний за випуск *Т. В. Критська*
Коректор *О. М. Міхайлуца*