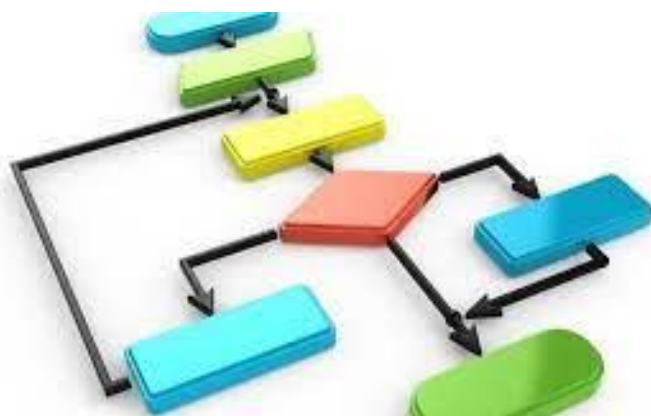


Валерій ФРАТАВЧАН, Тоня ФРАТАВЧАН, Василь ЛАЗОРИК



АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ

Навчальний посібник для закладів вищої освіти

Чернівці -2022

УДК 004.42(075.8)

Ф 854

Рекомендовано Вченою радою Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича (протокол №8 від 22 вересня 2022 року).

Рецензенти:

Волохов Віктор Миколайович – доцент кафедри теорії та технології програмування Київського національного університету імені Тараса Шевченка, кандидат фізико-математичних наук;

Ляшкевич Василь Яремович – доцент кафедри системного проектування Львівського національного університету імені Івана Франка, ІТ компанія «Technology at GlobalLogic, A Hitachi Group Company», Solution Architect.

Ф 854 *Фратавчан В.Г., Фратавчан Т.М., Лазорик В.В.* Алгоритмізація та програмування, навчальний посібник для закладів вищої освіти. – ЧНУ, 2022, – 286 с.

Посібник містить матеріали для засвоєння початкового курсу програмування мовами C/C++ для консольного та віконного програмного інтерфейсу. Призначенням посібника є ознайомлення з синтаксисом базових інструкцій мов програмування C/C++, отримання початкових навичок алгоритмізації та програмування, засвоєння набору елементарних алгоритмів.

Посібник побудований як лекційно-практичний курс. Послідовно розглядаються теми, які пов'язані з питаннями введення та виведення даних, проведення математичних та логічних обчислень і перетворень, моделювання множинних та матрично-векторних операцій, обробки текстових даних, використання динамічних даних та динамічних структур даних, обробки файлової інформації, застосування парадигми процедурного програмування. Лекційний теоретичний матеріал кожної теми доповнений великою кількістю прикладів та програмних стереотипних фрагментів. У плані практичної та самостійної підготовки до кожної теми додається комплект завдань до лабораторних робіт.

Посібник розрахований на широке коло користувачів, у тому числі без істотного попереднього досвіду алгоритмізації та програмування. Матеріал посібника буде корисним для студентів початкових курсів всіх спеціальностей галузі знань «12 Інформаційні технології».

© В.Г.Фратавчан, 2022, © Т.М.Фратавчан, 2022,

© В.В.Лазорик, 2022, © ЧНУ, 2022

ЗМІСТ

Передмова	9
Розділ 1. Програми та алгоритми. Базові поняття	11
Тема 1.1. Коротко про дані, програми та алгоритми	11
1. Програмне забезпечення обчислювального комплексу.....	11
2. Середовища програмування. Інтегровані середовища розробки.....	12
3. Дані. Інтерпретація, подання та збереження.....	13
4. Алгоритми. Способи подання та класифікація.....	16
Тема 1.2. Алгоритмічні мови та середовища програмування. Середовище Borland C++ Builder	19
1. Мови програмування.....	19
2. Середовища програмування.....	19
3. Структура середовища Borland C++ Builder.....	20
4. Створення нового програмного проєкту.....	21
5. Файлова конфігурація консольних та віконних програм.....	27
Розділ 2. Мови C/C++. Елементарні дії та інструкції	29
Тема 2.1. Структура C/C++ програми. Базові поняття мови програмування	29
1. Базові поняття мови програмування C/C++.....	29
2. Структура C/C++ програми.....	29
3. Визначення, опис, оголошення даних.....	32
4. Математичні операції.....	35
5. Математичні вирази. Оператор присвоєння.....	36
Тема 2.2. Операції введення даних та відображення результатів	39
1. Безпосереднє консольне введення та відображення окремих символів.....	39
2. Потокове консольне введення та відображення окремих символів.....	40
3. Форматне введення та відображення.....	41
4. Потокове введення та відображення.....	44
5. Організація обміну повідомленнями у віконному інтерфейсі.....	45

Завдання для лабораторної роботи.....	49
Тема 2.3. Розгалуження. Оператори умовного, альтернативного, варіантного виконання дій.....	51
1. Прості умовні розгалуження.....	51
2. Альтернативні розгалуження.....	52
3. Умовні вирази.....	53
4. Оператор варіанту.....	54
5. Вкладені розгалуження.....	55
Завдання для перевірки засвоєння теми.....	58
Завдання для лабораторної роботи.....	58
Тема 2.4. Циклічні алгоритми та програми.....	63
1. Поняття циклу.....	63
2. Типи циклічних алгоритмів.....	63
3. Організація циклів за допомогою міток.....	63
4. Стандартні цикли. Цикли із початковою умовою виконання.....	64
5. Цикли з кінцевою умовою повторення.....	66
6. Узагальнений цикл.....	68
7. Цикли з лічильником.....	70
Завдання для перевірки засвоєння теми.....	72
Завдання для лабораторних робіт.....	73
Розділ 3. Обробка регулярних даних.....	76
Тема 3.1. Одновимірні масиви.....	76
1. Поняття масиву. Способи опису масивів.....	76
2. Операції введення та відображення елементів масивів.....	77
3. Повторні цикли. Однотипні та змішані вкладені цикли.....	86
Завдання для перевірки засвоєння теми.....	88
Завдання для лабораторних робіт.....	88
Тема 3.2. Елементарні базові алгоритми з використанням одновимірних масивів.....	91
1. Прості алгоритми обробки одновимірних масивів.....	91

2. Застосування одновимірних масивів для моделювання операцій над множинами.....	94
Завдання для лабораторних робіт.....	101
Тема 3.3. Алгоритми пошуку та впорядкування.....	104
1. Алгоритми пошуку.....	104
2. Алгоритми впорядкування.....	110
Завдання для лабораторних робіт.....	117
Тема 3.4. Обробка двовимірних масивів. Операції введення та виведення.....	120
1. Поняття двовимірного масиву. Способи опису.....	120
2. Побудова циклів для обробки двовимірних масивів.....	121
Тема 3.5. Реалізація векторно-матричних операцій.....	131
1. Основні векторно-матричні операції.....	131
2. Операції над векторами.....	132
3. Обчислення матричних характеристик.....	136
4. Перетворення матриць. Елементарні операції.....	138
5. Векторно-матричне множення.....	142
6. Множення матриць.....	146
Завдання для перевірки засвоєння теми.....	152
Завдання для лабораторних робіт.....	152
Тема 3.6. Особливості обробки текстових даних.....	155
1. Операції введення текстових (символьних) даних.....	155
2. Операції відображення текстових даних.....	163
3. Введення та відображення текстових даних засобами потокових класів.....	165
4. Бібліотечні засоби обробки тестів.....	166
Завдання для лабораторних робіт.....	169
Тема 3.7. Регулярні дані структурного типу.....	171

1. Способи оголошення типів структурованих даних. Оголошення структурованих даних.....	171
2. Операції над структурами.....	178
3. Введення та відображення структур.....	180
4. Вкладені структури.....	182
Завдання для лабораторних робіт.....	183
Розділ 4. Адреси, вказівники, динамічні дані.....	184
Тема 4.1. Робота з адресами та вказівниками.....	184
1. Поняття вказівника. Оголошення даних вказівникового типу.....	184
2. Операції з вказівниками.....	186
3. Використання вказівників на структуру	189
Тема 4.2. Динамічний розподіл пам'яті.....	190
1. Керування пам'яттю . Загальні положення.....	190
2. Виділення пам'яті для скалярних даних.....	192
3. Виділення пам'яті та використання вказівників на одновимірні масиви.....	193
4. Виділення пам'яті та використання вказівників на двовимірні масиви.....	195
5. Нестереотипна модель роботи з динамічними двовимірними масивами.....	198
6. Вказівники на динамічні структури та масиви структур.....	201
Завдання для лабораторних робіт.....	203
Розділ 5. Парадигма процедурного програмування в мовах C/C++.....	205
Тема 5.1. Функції C/C++.....	205
1. Поняття підпрограми. Функції.....	205
2. Опис функції у прототипній версії мови C.....	207
3. Сучасна форма опису функції.....	209
4. Функції з прототипами.....	210
5. Область дії змінної.....	213

6. Правила передачі параметрів.....	217
Завдання для лабораторних робіт.....	227
Тема 5.2. Рекурсивні функції	228
1. Поняття рекурсії. Класифікація рекурсій.....	228
2. Класичні рекурсивні задачі.....	230
3. Рекурсивні алгоритми впорядкування.....	233
Завдання для лабораторних робіт.....	241
Розділ 6. Динамічні структури даних.....	243
Тема 6.1. Динамічні структури даних. Списки.....	244
1. Списки. Основні положення.....	244
2. Черга. Алгоритм створення.....	246
3. Стек. Алгоритм створення.....	247
4. Перегляд та обробка елементів списку.....	248
5. Списки. Динамічні «маніпуляції» із елементами.....	248
6. Створення систематизованого списку.....	250
7. Особлива операція – знищення повного списку.....	252
Завдання для лабораторних робіт.....	255
Тема 6.2. Динамічні структури даних. Деревя.....	256
1. Ієрархічні структури даних. Деревя.....	256
2. Бінарні деревя.....	257
3. Операції з бінарними деревями.....	258
Завдання для лабораторних робіт.....	266
Розділ 7. Обробка файлової інформації.....	267
Тема 7.1. Основи обробки файлової інформації.....	267
1. Файлова інформація. Основні положення.....	267
2. Класифікація файлів.....	268
3. Стандартні засоби доступу до файлів та виконання операцій	

введення та збереження інформації.....	268
Тема 7.2. Робота з текстовими файлами послідовного доступу.....	270
1. Робота з текстовими та бінарними файлами у консольному режимі послідовного доступу.....	270
2. Робота з текстовими та бінарними файлами у віконному режимі послідовного доступу.....	272
Тема 7.3. Бінарні файли. Режим прямого доступу.....	276
1. Особливості бінарних файлів. Операції блочного введення та збереження інформації.....	276
2. Методи та програмні засоби роботи у режимі прямого доступу....	277
3. Приклад реалізації операцій блочного обміну інформації та роботи у режимі прямого доступу.....	280
Завдання для лабораторних робіт.....	282
Література.....	287

ПЕРЕДМОВА

Читачам пропонується навчальний посібник, у якому синхронно проводиться знайомство з компонентами та засобами мови програмування, а також демонструються та аналізуються елементарні стереотипні алгоритми, в яких задіяні ці засоби.

Посібник призначений для студентів молодших курсів всіх спеціальностей галузі знань «Інформаційні технології», але основний акцент ставиться на відповідність стандарту вищої освіти спеціальності «Комп'ютерні науки» (в редакції 2019 року). Посібник розрахований на студентів початкових курсів і не потребує конкретної попередньої підготовки. Але засвоєння матеріалу буде легшим та ефективнішим при наявності хороших знань та навичок, отриманих при вивченні курсів інформатики та обчислювальної техніки за програмою загальноосвітньої школи. Також засвоєнню матеріалу сприятиме попереднє знайомство (за шкільним курсом програмування) з мовами програмування Object Pascal або Python.

Основною мовою програмування при викладанні матеріалу обрано мову C/C++. Дані мови вважаються мовами загального призначення, вони отримали розвиток у вигляді декількох поколінь та різновидів професійних мов програмування, таких як C#, Java, тощо. При цьому прототипна версія мови залишається основою для багатьох крос-компіляторних платформ, вимагає від розробників програм високої дисципліни та культури програмування, розуміння обчислювальних процесів на алгоритмічному, системному та апаратному рівні. Це робить дану мову привабливою для початкового періоду підготовки професіоналів ІТ - галузі.

Додатковим призначенням даного підручника є формування у студентів алгоритмічного мислення. Паралельно з вивченням мови програмування, при ознайомленні з програмними мовними засобами та при засвоєнні техніки програмування, у матеріалі підручника демонструються стереотипні та базові алгоритмічні конструкції для виконання математичних обчислень, опису, обробки і перетворення даних та інформаційних структур. В процесі вивчення даного матеріалу студенти знайомляться з різними техніками та парадигмами програмування, велика увага приділяється питанням ефективності та надійності алгоритмів та програм. Результатом засвоєння даного курсу також може бути підвищення рівня культури та дисципліни оформлення програмних текстів, розуміння та опанування

методиками поетапного створення програмних продуктів, побудови математичних та алгоритмічних моделей системних та прикладних задач, реалізації, тестування програм, оформлення звітної документації.

Підручник складається з семи розділів. У першому розділі містяться базові відомості про алгоритми та програми, відбувається ознайомлення з термінами та основними компонентами, з формою збереження даних, зі складом середовищ програмування, складом програмного проєкту та структурою програм. Наступні розділи мають змішану структуру і містять відомості про мовні інструкції та відповідні алгоритмічні елементарні конструкції. Теоретичний матеріал чергується з прикладами практичного застосування цих інструкцій при розв'язуванні математичних або прикладних задач. Тематика раціонально структурована, матеріал поступово ускладнюється, охоплюючи широкий спектр знань та навичок в області алгоритмізації та техніки програмування. Вивченню кожної теми сприяють супроводжуючий набір контрольних питань, тестових задач для самоперевірки, комплектів лабораторних завдань.

На завершення передумови автори вважають за потрібне відмітити декілька хрестоматійних першоджерел, якими автори користувалися впродовж багаторічної педагогічної діяльності та які вплинули на зміст та стиль викладання даних матеріалів. Це підручники «Б.Керніган, Д.Річчі. Мова програмування С», «Т.В.Ковалюк. Основи програмування» та «Т.П.Караванова. 777 алгоритмів».

Автори виражають вдячність своїм вчителям, які формували їх знання та навички в області алгоритмізації та програмування і яких вони вважають «співавторами» даного підручника: Ф.О.Сопронюку, М.Ф.Кириченку, Л.І.Ясинській, Л.К.Шеляг, М.А.Руснаку, Б.Є.Ковалю, А.Д.Семенюку, Є.М.Тимофіївій. Також автори вдячні своїм колегам – співробітникам та студентам кафедри математичних проблем управління і кібернетики та кафедри математичного моделювання Чернівецького національного університету імені Юрія Федьковича за цінні ідеї, допоміжні матеріали, поради та оцінку роботи.

Розділ 1. Програми та алгоритми. Базові поняття

Тема 1.1. Коротко про дані, програми та алгоритми

1. Програмне забезпечення обчислювального комплексу.

Функціонування обчислювальної системи (комп'ютера), незалежно від складності, призначення та функціональних можливостей, базується на двох складових – апаратного та програмного забезпечення. Програмне забезпечення ділиться на дві категорії – системні та прикладні програми. Системні програми – це ресурси, які відповідають за ефективне та коректне функціонування самої обчислювальної системи. Прикладні програми призначені задовольнити потреби користувачів.

До системних програм відносять:

А) **Операційні системи** – програмні комплекси, до функцій яких входить:

- початкове завантаження обчислювальної системи;
- діагностика роботи підсистем комп'ютера та приєднаних периферійних апаратних засобів;
- завантаження програм у оперативну пам'ять та диспетчеризація пам'яті між програмами;
- організація інтерфейсу «комп'ютер-користувач»;
- керування файловою структурою комп'ютера.

Б) **Драйвери** – програми, які забезпечують перетворення даних та їх супроводження під час передачі між підсистемами комп'ютера та під час взаємодії із зовнішніми апаратними пристроями;

В) **Утиліти** – «корисні програми», за допомогою яких можна покращити інтерфейс користувача, керувати процесом налагодження роботи комп'ютера, підвищити ефективність його функціонування.

Г) **Системи та середовища програмування** – програмні комплекси, які призначені для створення нових системних та прикладних програмних ресурсів.

До категорії **прикладних програм** можна віднести:

- текстові редактори;
- графічні редактори;
- системи керування базами даних;
- електронні таблиці;

- пакети прикладних програм наукового, інженерного, професійного призначення;
- інтернет – браузері;
- мультимедійні програми;
- диспетчерські та облікові інформаційні системи;
- ігрові програми, тощо.

2. Середовища програмування. Інтегровані середовища розробки

Техніка та методика створення програмних продуктів розвивалася та вдосконалювалася синхронно з розвитком апаратних та програмних засобів. На перших етапах розвитку комп'ютерної техніки розробники програм створювали програмні тексти за допомогою текстових редакторів і керували процесом створення програм у «ручному режимі». Зростання складності програм, долучення до програм додаткових «напівфабрикатних» програмних ресурсів спровокували появу спеціалізованих систем та середовищ програмування. Сучасне середовище програмування – це складний програмний комплекс, який базується на одній або декількох мовах програмування і може складатися з наступних компонентів:

- інтерфейс середовища – сервісна оболонка, яка дозволяє здійснити загальне керування процесом створення нової програми;
- текстовий редактор – для створення та редагування нових програмних текстів;
- мовні бібліотечні ресурси – для розв'язування стереотипних задач в різних предметних областях;
- мовний транслятор – для аналізу програмних текстів та перетворення нової програми з текстового або структурного формату у системну або прикладну програму.

Слід сказати декілька слів про мови програмування та мовні транслятори. Мови програмування сьогодні ділять на дві категорії:

- **мови програмування низького рівня** – програма складається з послідовності простих інструкцій, що відповідають елементарним діям комп'ютерного процесора (до таких мов відносять машинні коди та асемблери);
- **мови програмування високого рівня** – програмні тексти мають складну структуровану форму і складаються із змістовних інструкцій, що нагадують речення загальних людських мов.

До окремої категорії інколи відносять спеціалізовані мови оформлення завдань у пакетах прикладних програм.

Мовні транслятори також діляться на декілька категорій в залежності від призначення, базової мови програмування та вигляду початкового та кінцевого стану програми. Можна виділити наступні види мовних трансляторів:

- **інтерпретатори команд операційних систем** – «розуміють» та виконують інструкції «ручних» команд та системних файлів завантаження;
- **асемблери** – перевіряють коректність програмних текстів та переводить програми з мови асемблера або машинних кодів у виконавчу програму;
- **компілятори** - перевіряють коректність програмних текстів та переводить програми з мови високого рівня у виконавчу програму (отримана програма функціонує незалежно від середовища програмування);
- **інтерпретатори** – перевіряють інструкції мови високого рівня та синхронно виконують кожну інструкцію. В деяких інтерпретаторах вхідний програмний текст переводиться у деяку проміжну форму. Сама вихідна програма при цьому потребує наявності середовища розробки або наявності іншої операційної платформи;
- **препроцесори** – додатковий допоміжний засіб компілятора або інтерпретатора, який дозволяє розширити можливості самої базової мови програмування створенням нових мовних ресурсів або підключенням і адаптацією бібліотечних ресурсів;
- **крос-компілятори** – спеціалізовані компілятори, які встановлені на повнофункціональних комп'ютерах і які призначені для створення програм, що будуть функціонувати на інших обчислювальних платформах з обмеженими можливостями або для створення програм для мікроконтролерів.

3. Дані. Інтерпретація, подання та збереження

Для подальшого однозначного трактування матеріалу слід зупинитись на декількох «аксіоматичних» термінах, відомих нам ще зі шкільних курсів інформатики та комп'ютерної техніки.

Дані – це будь-які відомості про об’єкти, явища та процеси у реальному або віртуальному світі.

Інформація – дані, які були використані для вирішення деяких задач (різниця між цими термінами досить умовна і часто ці терміни трактуються ідентично).

Концепт «інформація» породжує термін «**інформатика**» – наука про засоби та методи накопичення, збереження, передачі, аналізу, перетворення, обробки та подання інформації.

На формальному рівні дані (інформацію) в інформаційній галузі подаються у числовому (цифровому) вигляді, у тому числі текстову, аудіо, відео, графічну інформацію тощо. Тому варто ознайомитись з декількома методами подання оцифрованої інформації. На апаратному рівні елементарним носієм інформації є електричний елемент, який може знаходитися у двох станах («зарядженому» або «розрядженому»), і який таким чином «моделює» два цифрових значення – 1 та 0. Тобто є потреба у деякій системі перекодування будь-яких чисел цифрами 0 та 1. Оперувати різними наборами цифр дозволяє **система числення**.

Система числення – це множина цифр та набір правил для запису чисел та виконання математичних операцій.

В інформаційній галузі, безпосередньо, використовуються чотири позиційні системи числення – десяткова, двійкова, вісімкова, шістнадцяткова.

Позиційна система числення визначається кількістю цифр (основа системи числення) та множиною цифр. Ціле додатне число в позиційній системі числення можна трактувати як скорочений запис многочлена. Наприклад, якщо система має основу N , тоді число C із записом $C=(c_k c_{k-1} \dots c_1 c_0)_N$, де c_0, c_1, \dots, c_k – цифри в системі з основою N , буде відповідати поліному (у цій самій системі числення) $C=(c_k N^k + c_{k-1} N^{k-1} + \dots + c_1 N^1 + c_0 N^0)_N$.

Для інших систем можна навести такі приклади:

$$(2603)_{10}=(2*10^3+6*10^2+0*10^1+3)_{10},$$

$$(1001)_2=(1*2^3+0*2^2+0*2^1+1)_{10},$$

$$(6014)_8=(6*8^3+0*8^2+1*8^1+4)_{10}.$$

Зауважимо, що 16-кова система має основу 16 та множину **цифр (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)**.

Чому в ІТ галузі використовуються саме ці системи числення стане зрозуміло, коли проаналізуємо таблицю відповідності між цифрами цих систем:

10-кова	2-кова	8-кова	2-кова	16-кова	2-кова
0	0	0	000	0	0000
1	1	1	001	1	0001
2	10	2	010	2	0010
3	11	3	011	3	0011
4	100	4	100	4	0100
5	101	5	101	5	0101
6	110	6	110	6	0110
7	111	7	111	7	0111
8	1000			8	1000
9	1001			9	1001
10	1010			A	1010
11	1011			B	1011
12	1100			C	1100
13	1101			D	1101
14	1110			E	1110
15	1111			F	1111

Кожну вісімкову цифру можна кодувати трьома цифрами у двійковій системі, а кожна 16-кова цифра – чотирма цифрами у двійковій системі. Це ще дозволяє використати двійкову систему як «посередника» для переходу між 8-ковою та 16-ковою системами. Нажаль, такої однозначності немає між двійковою та десятковою системами, тому для переходу від десяткової до інших систем потрібно застосувати інші спеціальні правила. А саме – при переведенні цілого додатного числа A з десяткової системи числення до системи числення з основою N потрібно послідовно та рекурентно ділити націло число A на N (поки число A більше 0), фіксувати після кожного ділення остачу, після чого записати послідовність остач у зворотному порядку.

Приклад. Перевести число 157 з десяткової у двійкову систему числення.
Розв’язання.

$$157 : 2 = 78 (1)$$

$$78 : 2 = 39 \text{ (0)}$$

$$39 : 2 = 19 \text{ (1)}$$

$$19 : 2 = 9 \text{ (1)}$$

$$9 : 2 = 4 \text{ (1)}$$

$$4 : 2 = 2 \text{ (0)}$$

$$2 : 2 = 1 \text{ (0)}$$

$$1 : 2 = 0 \text{ (1)}$$

Результат: $157_{10} = 10011101_2$.

Після знайомства з 16-ковою позиційною системою можемо навести приклад і для поліноміального представлення 16-кових чисел:

$$(6A1E)_8 = (6 \cdot 16^3 + 10 \cdot 16^2 + 1 \cdot 16^1 + 14)_{10}.$$

Зауваження: Детально системи числення на ІТ спеціальностях вивчаються за програмою спеціальних дисциплін «Архітектура комп'ютерів», «Комп'ютерна арифметика», тощо.

4. Алгоритми. Способи подання та класифікація

Алгоритм – це скінчена множина дій або правил, виконання яких забезпечує розв'язування задачі визначеного змісту за скінчену кількість кроків.

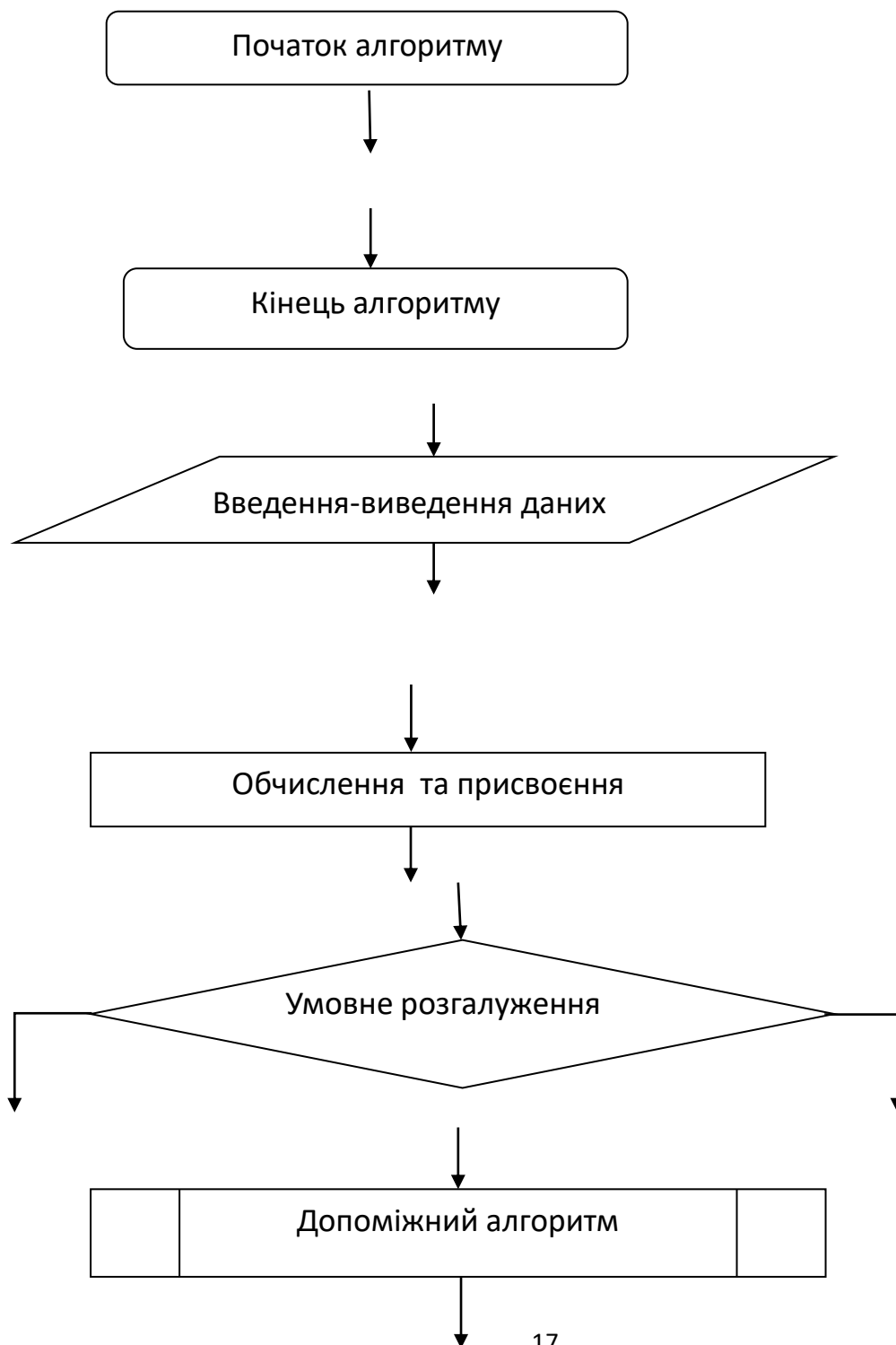
За цим означенням можна виділити декілька основних властивостей алгоритму: визначеність, результативність, дискретність, ефективність, масовість...

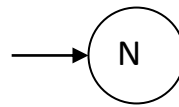
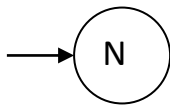
Історична довідка: Термін «алгоритм» пов'язана з ім'ям перського математика, філософа та астронома Аль Хорезмі (780 — 850 роки), якому приписують вперше зареєстрований опис розв'язку математичної задачі у вигляді пронумерованих впорядкованих абстрактних дій.

Застосовують декілька методів подання алгоритмів:

- вербальний, тобто у вигляді послідовності текстових правил (до цієї категорії можна відносити підкатегорію «формально-математичний» або «аналітичний», тобто у вигляді послідовності формул);
- графічний (або схематичний);
- алгоритмічною або програмною мовою.

У даній роботі часто будуть застосовані візуально-графічні методи демонстрації, тому детальніше зупинимося на графічному (схематичному) методі. Графічний образ алгоритму називається **блок-схемою**. Блок-схема складається з блоків та зв'язків (з'єднань). Блоки визначають алгоритмічні дії, з'єднання – хронологічну послідовність їх виконання. У блок-схемах використовуються наступні елементарні блоки:





Точка розриву та продовження.

З'єднання позначаються стрілками, які визначають напрямок переходу від поточної дії до наступної. Напрямки руху «зліва-направо» та «зверху-вниз» вважаються природними і стрілочки на зв'язках можна не відмічати.

В залежності від змісту та складності задач та їх структури, алгоритми (або фрагменти в складних алгоритмах) можна розбити на такі категорії:

- 1) лінійні – містять тільки операції введення-виведення та операцій присвоєння;
- 2) розгалужені – містять хоча б одне розгалуження;
- 3) циклічні – система з'єднань побудована таким чином, що деяка частина дій може виконуватися багаторазово.

Алгоритми, в яких присутні алгоритмічні фрагменти різного типу, називаються комбінованими.