

Лабораторна робота № 1

1. АЛГОРИТМИ. ОСНОВНІ ПОНЯТТЯ ТА ВЛАСТИВІСТІ

Поняття алгоритму

В теперішній час поняття алгоритму – одне з фундаментальних понять науки «інформатика». Алгоритм — це послідовність дій над заданими об'єктами, чітко та однозначно визначаюча обчислювальний процес.

Ефективним методом побудови алгоритмів є метод покрокової деталізації, при якому завдання розбивається на кілька простих підзадач (модулів), і для кожного модуля створюється свій власний алгоритм.

Здебільшого модуль реалізує певний процес обробки інформації і застосовується як для окремого використання, так і для включення модуля в інші алгоритми. Застосування модульності при створенні алгоритмів дозволяє

розбити великі завдання на незалежні блоки (модулі), усуває повторення стандартних дій і значно прискорює процес відлагодження алгоритму в цілому.

Найчастіше головний модуль алгоритму містить декілька інших модулів, створених раніше. Використовуючи модулі як складові великої конструкції, можна створювати алгоритми будь-якого ступеня складності, і при цьому не втрачати контролю за функціонуванням алгоритму всієї задачі.

Такий метод називається структурним проектуванням алгоритму, він є універсальним і може використовуватися як для обчислювальних процесів (так зване системне програмування), так і для процесів реального життя.

Властивості алгоритму

1. **Дискретність** – процес розв'язку розбивається на кроки. Кожен крок – це одна дія або підпорядкований алгоритм. Таким чином полегшується процес знайдення помилок і редагування алгоритму.

2. **Визначеність (точність)** – кожен крок алгоритму має бути однозначно описаною дією і не містити двозначностей.

3. **Зрозумілість** – усі дії, включені до алгоритму, мають бути у межах компетенції виконавця алгоритму.

4. **Універсальність (масовість)** – алгоритм має виконуватися при будь-яких значеннях вхідних даних та початкових умов.

5. **Скінченність** – алгоритм має бути реалізований за кінцеве число кроків і повинен використовувати кінцевий набір вхідних значень.

6. **Результативність** – алгоритм має привести до отримання результату.

Способи подання алгоритмів

Алгоритми можуть бути подані

– словесно (засобами природної мови у вигляді плану дій)

– графічно (у вигляді блок-схем)

– у вигляді програм, написаних певною мовою програмування.

Найчастіше алгоритми обчислювальних процесів подаються у вигляді блок-схем, де кожний крок алгоритму представлено спеціальним блоком, який показує дію, яку треба виконати.

Графічному опису передую, як правило, побудова математичної моделі – математичного опису алгоритму. Такий опис полягає у формалізованому (із застосуванням математичних символів) поданні всіх розглянутих залежностей і методів відшукування значень вихідних даних на підставі вхідних.

Призначення блоків впливає з їхніх назв. Блоки поєднуються між собою лініями потоку. Природні напрями потоків зверху вниз і зліва направо. Якщо напрямок потоку інший то лінія повинна мати стрілку.

Рекомендується не перетинати лінії потоку, а використовувати поєднувач блоків. Для використання поєднувача блоки мають бути попередньо пронумеровані, а сам поєднувач має містити цифру – номер блоку, з яким відбувається поєднання або номер блоку, з якого відбувається поєднання. Інакше: при з'єднуванні блоків можливо використовувати спільний символ, який записується в поєднувач блоків.

Основні види блоків та їх призначення наведені у таблиці 1.1.

2. АЛГОРИТМИ ЛІНІЙНОЇ СТРУКТУРИ

Лінійна структура використовується в алгоритмах, де одна дія виконується слідом за іншою послідовно і при цьому жодна з дій не пропускається і не повторюється.

Розглянемо приклади алгоритмів лінійної структури.

Приклад 2.1. Обчислити висоти трикутника зі сторонами **a**, **b**, **c** за формулами

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

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

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

$$\text{де } p = \frac{a+b+c}{2}$$

Введемо позначення:

$$t = 2\sqrt{p(p-a)(p-b)(p-c)} \Rightarrow h_a = \frac{t}{a}, \quad h_b = \frac{t}{b}, \quad h_c = \frac{t}{c}$$

Блок-схема алгоритму наведена на рисунку 2.1.

Основні види блоків, їх призначення

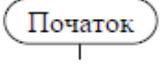
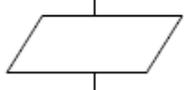
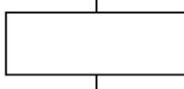
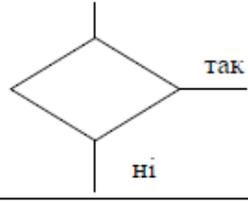
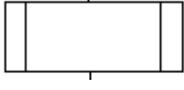
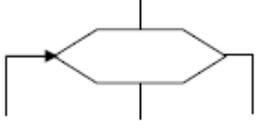
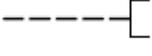
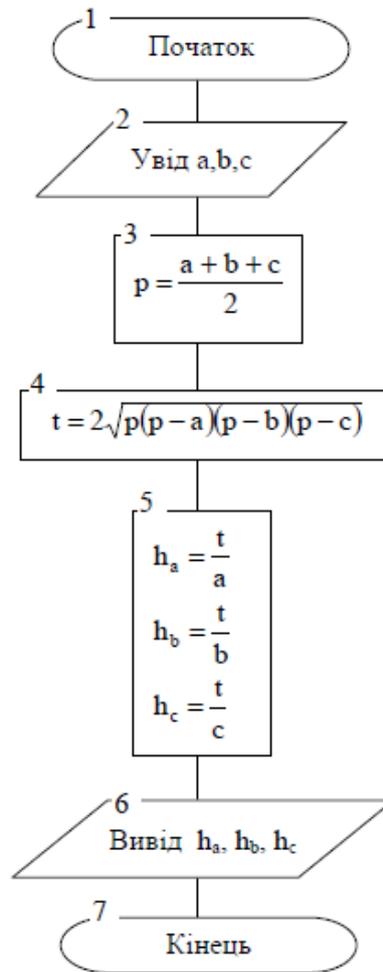
Графічне зображення	Назва, виконувана дія
	Початок алгоритму
	Закінчення алгоритму
	Блок вводу-виводу
	Виконання обчислень або присвоєння значень
	Перевірка умови. Якщо умова справедлива (набуває значення ІСТИНА), виконується перехід по лінії Так , а якщо не справедлива (набуває значення БРЕХНЯ), то виконується перехід по лінії Ні
	Виклик раніше створених алгоритмів (модулів)
	Блок організації циклу
	Коментар. Короткі пояснення до показаного блоку
	П'єднувач блоків

Рис. 2.1. Блок-схема алгоритму прикладу 2.1.



Завдання 1. Для заданих значень **a** і **c** знайти значення виразу:

$$Y = \sqrt{(|x^2 + c|)} - \sin(ax - c) ,$$

де $x = \cos(a^2 c^2)$

1. Скласти словесний спосіб надання алгоритму;
2. Скласти блок-схему алгоритму знаходження значення виразу.

АЛГОРИТМИ РОЗГАЛУЖЕНОЇ СТРУКТУРИ

Розгалужена структура передбачає вибір виконання дії залежно від виконання певної умови, при цьому деякі дії можуть не виконуватися взагалі (пропускатися).

Проста умова містить два вирази (значення), поєднані знаком операції відношення:

> більше за...

< менше за...

≥ більше або дорівнює...

≤ менше або дорівнює ...

≠ не дорівнює...

Результатом перевірки умови є логічний вираз ІСТИНА, якщо умова виконується, або БРЕХНЯ, якщо умова не виконується.

Приклад 3.1. Знайти значення дійсних коренів квадратного рівняння

$$ax^2 + bx + c = 0, \quad a \neq 0$$

Словесний спосіб подання алгоритму:

1. Уведення значень коефіцієнтів a , b , c .
2. Обчислення значення дискримінанта за формулою $D = b^2 - 4ac$.
3. Перевірка отриманого значення дискримінанта: якщо дискримінант ≥ 0 , то перехід на п.4, в противному разі перехід на п.6.
4. Обчислення дійсних коренів рівняння за формулами:

$$x_1 = \frac{-b + \sqrt{D}}{2a}$$

$$x_2 = \frac{-b - \sqrt{D}}{2a}$$

5. Виведення отриманих результатів x_1 та x_2 . Кінець алгоритму.
6. Виведення повідомлення «Дійсних коренів немає». Кінець алгоритму.

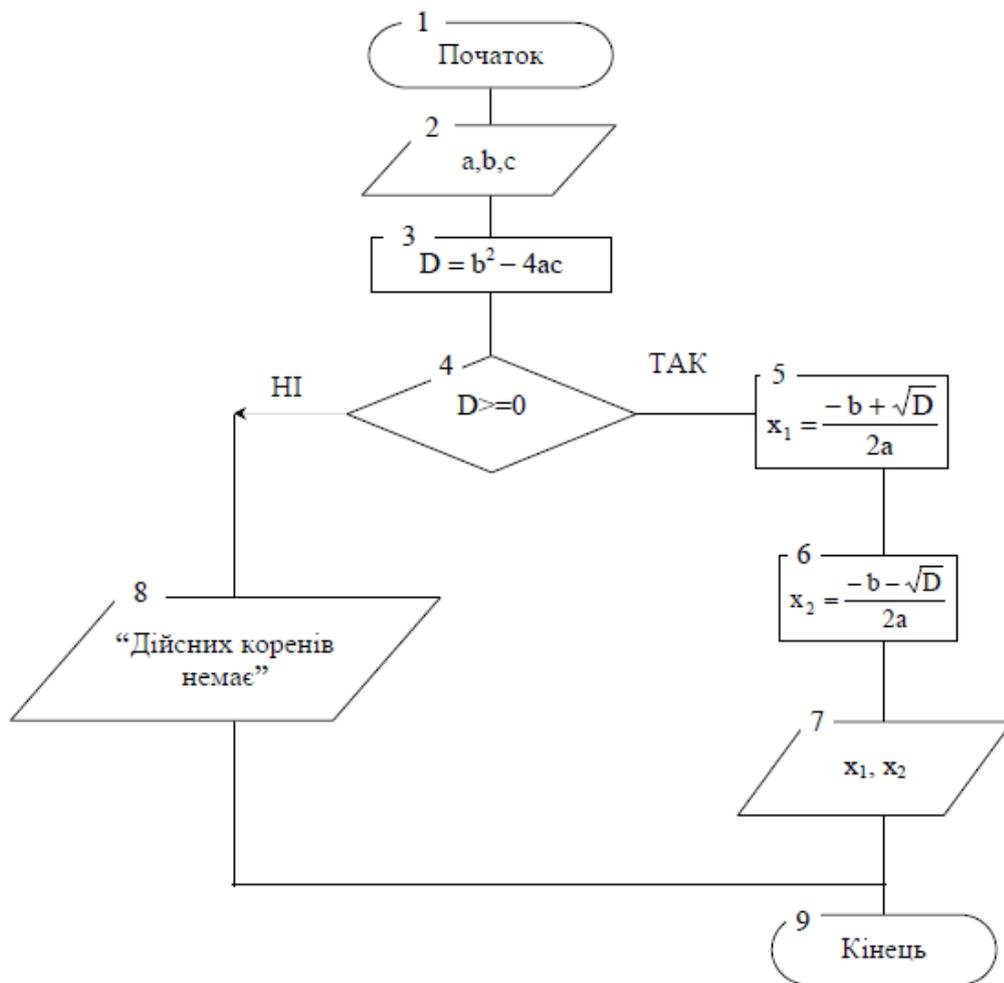


Рис. 3.1. Блок-схема алгоритму прикладу 3.1.

У блок-схемі алгоритму блок №4 використовується для перевірки умови і реалізує розгалуження: якщо умова набуває результату **істина**, то алгоритм продовжується по блоках 5, 6, 7, 9, а блок 8 зовсім не виконується. Навпаки, коли умова набуває значення **брехня**, алгоритм продовжується по блоках 8, 9, при цьому блоки 5, 6, 7 не виконуються.

Завдання 2. Для заданих значень x , a , b обчислити значення виразу:

$$Y = \begin{cases} \frac{abx^2}{b+x}, & x \leq 2 \\ a + \sin(bx), & 2 < x \leq 8 \\ \sqrt{|ab+x|}, & x > 8 \end{cases}$$

1. Скласти словесний спосіб надання алгоритму;
2. Скласти блок-схему алгоритму знаходження значення виразу.

3. АЛГОРИТМИ ЦИКЛІЧНОЇ СТРУКТУРИ

Циклом називають частину алгоритму, яка повторюється. При кожному черговому виконанні циклу перевіряється умова на продовження роботи і, якщо умова набуває результату ІСТИНА, цикл виконується, а якщо умова набуває результату БРЕХНЯ – цикл не виконується.

Перевірка умови може бути організована на початку циклу, і такий цикл називається циклом з передумовою, або у кінці циклу – тоді такий цикл називається циклом з післяумовою.

Різниця між такими циклами полягає в тому, що цикл з післяумовою виконується хоча б один раз, а цикл з передумовою може не виконуватися жодного разу.

Цикл по лічильнику характерний тим, що заздалегідь відома кількість повторень циклу, і цикл буде виконуватися, доки значення лічильника циклу не перевищить зазначену кількість повторень.

Якщо відомі початкове та кінцеве значення параметра циклу, а також закон (формула), за яким це значення змінюється, то цикл буде виконуватися, доки параметр циклу лежатиме у межах від початкового до кінцевого значення.

Приклад 4.1. Побудувати таблицю значень функції

$$Y = a + \sin(bx)$$

для заданих коефіцієнтів **a** і **b** та аргументу **x**, що змінюється від -4 до 6 з кроком 2 .

Словесний спосіб подання алгоритму:

1. Вводяться коефіцієнти **a** і **b**.
2. Задається початкове значення аргументу $x = -4$.
3. Обчислюється значення функції **Y** для поточного аргументу.
4. Виводиться отримане значення функції **Y**.
5. Значення аргументу **x** збільшується на 2 .
6. Перевіряється умова продовження циклу: якщо нове значення аргументу не перевищує заданого кінцевого значення **6**, то цикл (пункти – б) виконується ще раз, у противному випадку — кінець алгоритму.

На рисунку 4.1. подана блок-схема алгоритму.

Цей цикл є циклом з післяумовою, тому що перевірка умови проводиться у кінці циклу.

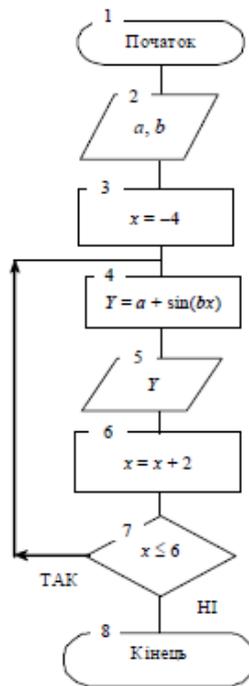


Рис. 4.1. Блок-схема прикладу 4.1.

Завдання 3. Обчислити значення функції

$$Y = \frac{b^2}{x^2 + b},$$

де x змінюється від $x_{\text{початкове}} = 0$ до $x_{\text{кінцеве}} = 3$
з кроком $\Delta x = 0,1$; $b = 3,8$

Число повторень циклу n обчислюється за формулою

$$n = \left[\frac{x_k - x_n}{\Delta x} \right] + 1$$

Навести блок-схему алгоритму двома способами.

4. ІТЕРАЦІЙНИЙ ЦИКЛ

Ітераційний цикл характерний тим, що виконується, доки проміжне значення не досягне зазначеної величини, тобто кількість повторень циклу заздалегідь невідома.

Приклад 5.1. Обчислити значення членів нескінченного ряду

$$x; \frac{x^2}{2!}; \frac{x^3}{3!}; \dots; \frac{x^n}{n!}, \dots \quad \text{де } x \text{ — задане число}$$

Обчислення проводити, доки не виконається умова $\frac{x^n}{n!} \leq \varepsilon^{(1)}$,
де ε – задане значення (точність обчислення).

У даному випадку ми не знаємо, при якому значенні n виконається умова (1).

Число повторень циклу залежить від проміжного результату, де:

$$y_1 = x \text{ – перший член ряду;}$$

$$y_2 = y_1 \cdot \frac{x}{2} \text{ – другий член ряду;}$$

$$y_3 = y_1 \cdot \frac{x}{3} \text{ – третій член ряду;}$$

.....

$$y_n = y_{n-1} \cdot \frac{x}{n} \text{ – рекурентна формула для } n\text{-го члена ряду}$$

Для того, щоб використати цю формулу для 1-го члена ряду $y_1 = y_0 \cdot \frac{x}{1}$
необхідно, щоб $y_0 = 1$.

На рисунку 5.1. подана блок-схема алгоритму.

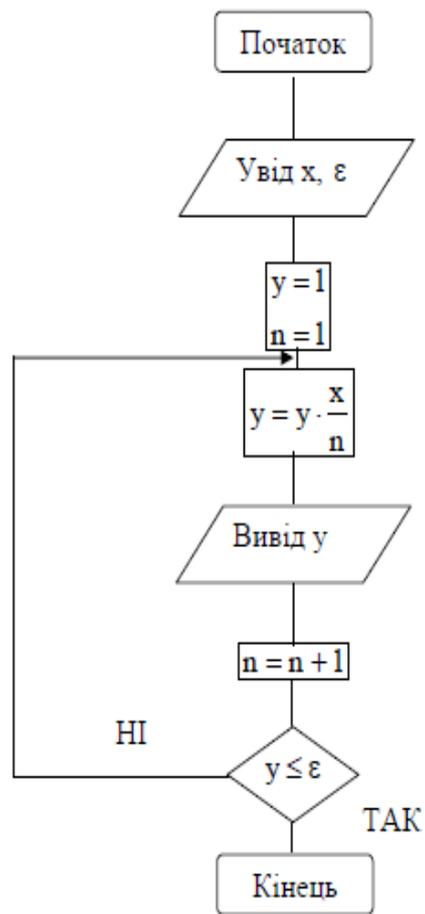


Рис. 5.1. Блок-схема прикладу 5.1.

Завдання 4. Обчислити вираз

$$Y = \sqrt[n]{100}, \quad \text{де } n = 2, 3, \dots$$

Визначити таке n , при якому $Y \leq \epsilon$,
якщо ϵ – задане значення (точність обчислення).

1. Скласти словесний спосіб надання алгоритму;
2. Скласти блок-схему алгоритму знаходження значення виразу.

5. ОДНОМІРНІ МАСИВИ

Таблювання функції

Приклад 6.1. Побудувати таблицю значень функції

$$y = \frac{b^2}{x^2 + b}$$

де X – масив чисел, $X = (x_1, x_2, \dots, x_n)$, $b = 4,8$.

На рисунку 6.1. зображено блок-схему алгоритма.

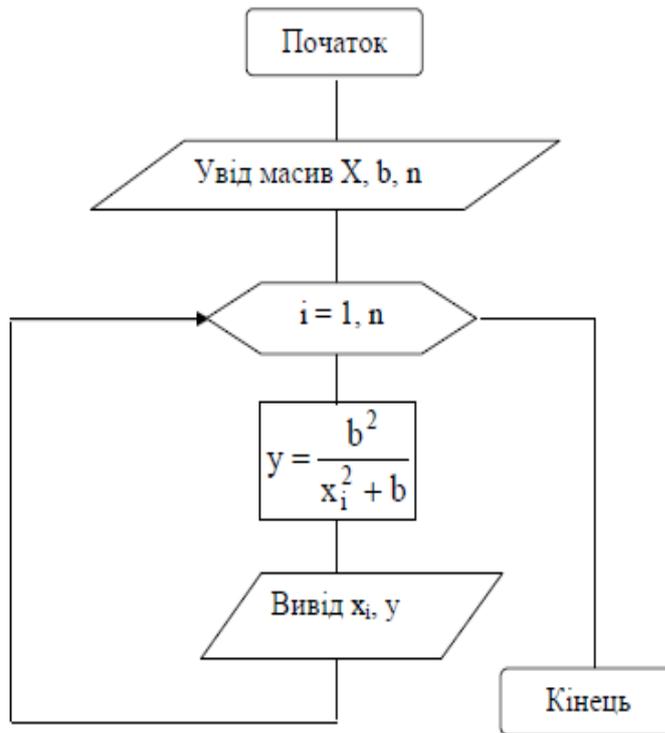


Рис. 6.1. Блок-схема прикладу 6.1

Нехай $X = (3; -0,8; 16,4; 100; -17; 2,4)$.

У даному випадку цикл буде працювати 6 разів, i – параметр циклу.

Завдання 5. Обчислити суму елементів масиву $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$.

$$S = \sum_{i=1}^n \alpha_i = \alpha_1 + \alpha_2 + \dots + \alpha_n$$

Скласти блок-схему алгоритму.