**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 функціюванням 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 набор 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 передує, як правило, побудова математичної моделi –

математичного опису алгоритму. Такий опис полягає у формалiзованому (із

застосуванням математичних символів) поданні всіх розглядуваних

залежностей і методів відшукання значень вихiдних даних на підставі вхiдних.

Призначення блоків випливає 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









Рис. 2.1. Блок-схема алгоритму прикладу 2.1.

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



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







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 ***х*, *a*, *b*** обчислити значення виразу:



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



для заданих коефіцієнтів **а** і **b** та аргументу **x**, що змінюється від –4 до 6 з

кроком 2.

**Словесний спосіб подання алгоритму:**

1. Вводяться коефіцієнти **а** 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 – 6)

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.



Завдання 3. Обчислити значення функції



Число повторень циклу **n** обчислюється за формулою 

Навести блок-схему алгоритму двома способами.

1. **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







Завдання 4. Обчислити вираз



1. Скласти словесний спосіб надання алгоритму;
2. Скласти блок-схему алгоритму знаходження значення виразу.
3. **ОДНОМІРНІ МАСИВИ**

**Табулювання функції**

**Приклад 6.1.** Побудувати таблицю значень функції



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

Завдання 5. Обчислити суму елементів массиву 



Скласти блок-схему алгоритму.