



Поняття алгоритму. Блок-схемний метод задання алгоритмів

к.т.н., доцент кафедри прикладної математики
Рижа Ірина Андріївна

Програмні продукти

- ▶ додаток на мобільному телефоні;
- ▶ комп'ютерна гра;
- ▶ розрахунок за надані послуги чи придбаний у крамниці товар.

Програмування

– це мистецтво, яке можна досягнути вивченням засад і положень, використовуючи вже існуючі прийоми та напрацювання чи створюючи нові надбання.

- ▶ Це **НЕ** означає просто навчитись писати команди на тій чи іншій мові програмування.
- ▶ Це вироблений спосіб мислення, скерований на творення якісного й оптимального способу одержання результату.

Про що ця лекція???

- ▶ Розглянемо поняття алгоритму та його властивості.
- ▶ Викладемо особливості зображення алгоритму у вигляді блок-схеми.



Задання алгоритму



Алгоритм

- ▶ конструктивно задане правило (закон), за яким вхідній інформації (умовам задачі) ставиться у відповідність нова вихідна інформація (розв'язок задачі);
- ▶ деякий скінченний набір операцій, виконання яких одна за однією через скінченне число кроків приводить до поставленої мети (розв'язку задачі);
- ▶ базове поняття і не означається через простіші поняття.

Властивості алгоритму

1. Скінченність

- ▶ Алгоритм є скінченим об'єктом, що є необхідною умовою його механічної реалізованості.

2. Масовість

- ▶ Алгоритм повинен бути застосовним до **широкого класу задач**, які відрізняються тільки вхідними даними.

3. Визначеність (детермінованість)

- ▶ Описання множини операцій, якою визначається алгоритм, **НЕ повинні допускати двояких тлумачень**.
- ▶ При виконанні операцій **НЕ** повинно виникати питань, що саме і як треба робити. Строго визначеним повинен бути і порядок виконання операцій.

Властивості алгоритму

4. Дискретність

- ▶ Процес, який визначається алгоритмом, повинен мати дискретний (перервний) характер, тобто являти собою **послідовність окремих завершених кроків**.
- ▶ Кожна операція алгоритму повинна виконуватися за скінченний час, а виконання наступної операції повинно починатися після завершення попередньої.

5. Результативність

- ▶ Виконання послідовності операцій, якою визначається алгоритм, через **скінчення** число кроків приводить до цілком певного результату.
- ▶ Виконання алгоритму **НЕ може закінчуватися невизначеною ситуацією** або ж зовсім **НЕ закінчуватися**.
- ▶ Кожен алгоритм передбачає наявність деяких вхідних даних і його виконання за скінченний час приводить до цілком певних результатів.

Властивості алгоритму

6. Формальність

- ▶ Будь-який виконавець, здатний сприймати і виконувати вказівки алгоритму, виконає поставлене завдання.

7. Захищеність

- ▶ Алгоритм повинен бути захищеним від **несанкціонованого використання** (використання без дозволу авторів) та **некваліфікованого користувача** (некоректне задання початкових даних).

8. Дружелюбність

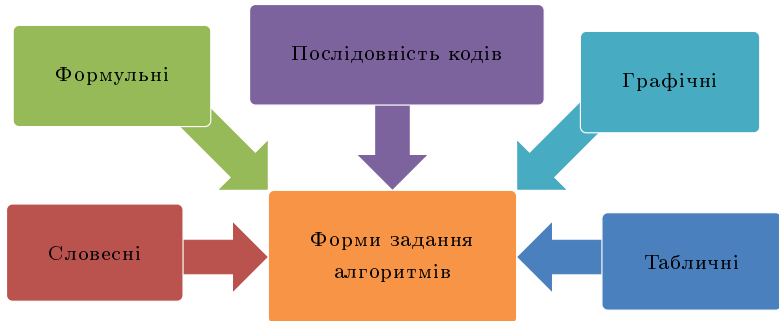
- ▶ Алгоритм завжди готовий вказати виконавцю на його помилки.

Система операцій (вказівок) виконавця

– набір операцій, виконанню яких навчено виконавця і які можуть бути включені у множину операцій, якою визначається алгоритм.

- ▶ Якщо вказівка про виконання завдання входить до системи допустимих для виконавця вказівок, то множина операцій, якою визначається алгоритм, містить *єдину* операцію.
- ▶ **Покрокова деталізація “зверху вниз”**
Якщо серед вказаних операцій є такі, що НЕ входять до системи операцій виконавця, то такі операції розкладаються на сукупність *простіших* операцій. Таке розкладання операцій на простіші продовжується доти, поки утвориться сукупність операцій, кожна з яких входить до системи операцій виконавця.

Форми задання алгоритмів



Зображення алгоритму у вигляді блок-схеми

Блок-схема

– графічне зображення алгоритму, при якому окремі кроки (етапи) алгоритму зображаються з допомогою геометричних фігур (символів, блоків), кожна з яких несе відповідне логічно-змістове навантаження.

Лінії потоків

– лінії, які з'єднують символи і задають зв'язки між етапами алгоритму.

Символи у блок-схемах

Символ ПРОЦЕСУ (*арифметичний блок*)

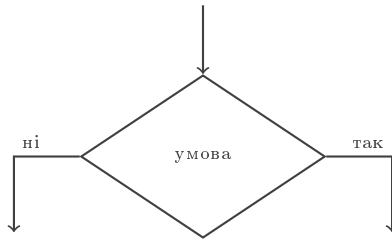
- ▶ безумовне виконання дії: обчислення певного виразу (арифметичного, логічного тощо) чи надання значення відповідному даному;
- ▶ прямокутник з відношенням сторін $a:b = 2:3$, де $a \in \{10, 15, 20, 50, 75, 100 \text{ мм}\}$;
- ▶ має тільки один вихід та, як правило, один вхід.

$$\begin{aligned} Y &= \sin(x), \\ I &= I + 1 \end{aligned}$$

Символи у блок-схемах

Символ РОЗГАЛУЖЕННЯ (блок умовного переходу)

- ▶ прийняття відповідного рішення;
- ▶ ромб з діагоналями a та b ;
- ▶ має один вхід і два виходи, кожен з яких у залежності від виконання умови, позначається “так” або “ні” (+ або -) і задає напрям продовження обчислень.



Символи у блок-схемах

Символ ПОЧАТКУ-КІНЦЯ обчислень

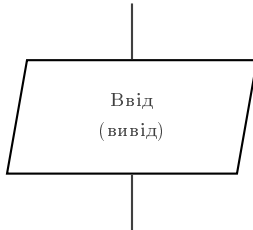
- ▶ початок і кінець алгоритму;
- ▶ прямокутник з заокругленими кінцями і висотою $\frac{a}{2}$;
- ▶ блок-початок має єдиний вихід (з'єднання з наступним блоком), а блок-кінець – єдиний вхід.



Символи у блок-схемах

Символ ВВОДУ-ВИВОДУ

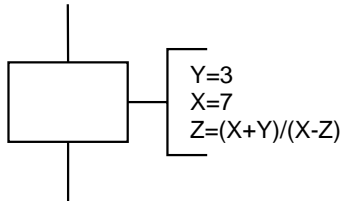
- ▶ початкова ініціалізація змінних (ввід) із зовнішнього носія;
- ▶ вивід значень змінних та виразів на зовнішній носій;
- ▶ паралелограм з висотою a та шириною $0,8b$.



Символи у блок-схемах

Винесення інформації за межі символу

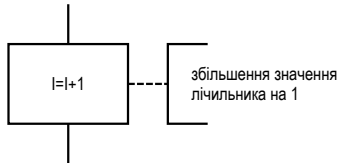
- ▶ використовується у випадках , коли не всю інформацію про перетворення, які виконує символ, можна у ньому розмістити;
- ▶ порядок виконання перетворень відповідає порядку їхнього запису у символі.



Символи у блок-схемах

Документування перетворень

- ▶ використовується для полегшення аналізу алгоритму та підвищення його читабельності;
- ▶ коментарі записуються у довільній формі.



Приклад 1.

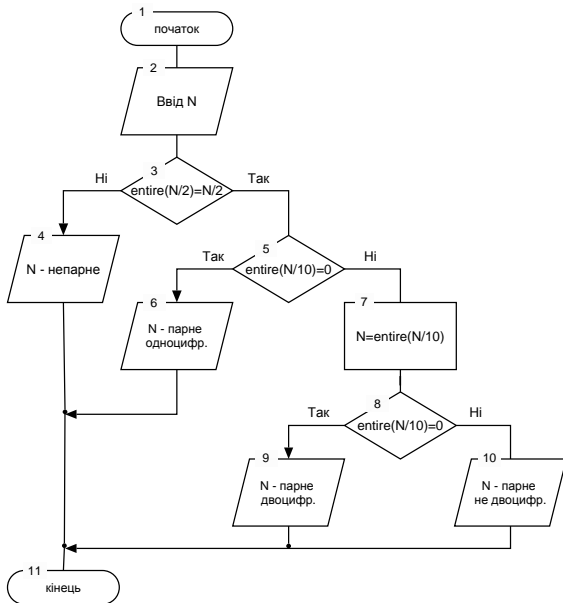
Розглянемо довільне ціле число N . Перевірити, чи задане число є парним і двоцифровим.

1. N – парне?
2. N – двоцифрове?

Функція $entire(x)$ виділяє цілу частину аргументу x , тобто

$$entire(5,7) = 5, \quad entire(-4,2) = -4.$$

1. $2 * entire(N/2) = N$
2. $entire(N/10)$



Дякую за увагу!

Далі буде...