

Алгоритм. Властивості
алгоритмів. Форми запису
алгоритмів. Лінійні
алгоритми.

Походження терміна «алгоритм» пов'язане з математикою. У IX столітті в Багдаді жив вчений ал (аль) -Хорезмі (повне ім'я - Мухаммед бен Муса ал-Хорезмі), математик, астроном, географ. В одній зі своїх праць він описав десяткову систему числення і вперше сформулював правила виконання арифметичних дій над цілими числами і звичайними дробами. Арабський оригінал цієї книги був загублений, але залишився латинський переклад XII в., за яким Західна Європа ознайомилася з десятковою системою числення і правилами виконання арифметичних дій.



Правила в книгах ал-Хорезмі в латинському перекладі починалися словами «Алгорізмi сказав». В інших латинських перекладах автор іменувався як Алгорітмус. Згодом було забуто, що Алгорізмi (Алгорітмус) - це автор правил, і ці правила стали називати алгоритмами.

Алгоритмом називається точний припис, що визначає послідовність дій виконавця, спрямованих на вирішення поставленого завдання. У ролі виконавців алгоритмів можуть виступати люди, роботи, комп'ютери.

Властивості алгоритму:

- **результативність** алгоритму означає, що за кінцеве число кроків повинен бути отриманий результат;
- **дискретність** алгоритму означає, що алгоритм повинен бути розбитий на послідовність виконуваних кроків;
- **зрозумілість** алгоритму означає, що алгоритм повинен містити тільки ті команди, які входять в набір команд, який може виконати конкретний виконавець;
- **точність** алгоритму означає, що кожна команда повинна розумітися однозначно;
- **масовість** алгоритму означає, що одного разу складений алгоритм повинен підходити для вирішення подібних завдань з різними вихідними даними.
- **детермінованість (визначеність)**. Алгоритм має властивість детермінованості - для одних і тих же наборів вихідних даних він буде видавати один і той же результат, тобто результат однозначно визначається вихідними даними.

Правила побудови алгоритмів.

Перше правило - при побудові алгоритму необхідно задати безліч об'єктів, з якими він буде працювати. Формалізоване (закодоване) представлення цих об'єктів носить назву дані. Алгоритм приступає до роботи з деяким набором даних, які називаються вхідними, і в результаті своєї роботи видає дані, які називаються вихідними.

Друге правило - для роботи алгоритму потрібна пам'ять. У пам'яті розміщуються вхідні дані, з якими алгоритм починає працювати, проміжні дані і вихідні дані, які є результатом роботи алгоритму. Пам'ять є дискретною, тобто складається з окремих осередків. Пойменованій осередок пам'яті носить назву змінної. У теорії алгоритмів розміри пам'яті не обмежуються.

Третє правило - дискретність. Алгоритм будується з окремих кроків (дій, операцій, команд).

Четверте правило - детермінованість. Після кожного кроку необхідно вказувати, який крок виконується наступним, або давати команду зупинки.

П'яте правило - збіжність (результативність). Алгоритм повинен завершувати роботу після кінцевого числа кроків. При цьому необхідно вказати, що вважати результатом роботи алгоритму.

Існують наступні форми подання алгоритму:

- словесна (вербальна) на неформальній мові;
- на мовах програмування;
- графічна.

Словесна форма подання алгоритму є найпоширенішою формою подання алгоритмів адресована людині. Форму словестного запису мають багато так звані «побутові алгоритми», які часто використовуються в повсякденній практиці (наприклад, інструкції).

Приклад 1.

Нехай потрібно записати послідовність елементарних дій для обчислень за формулою:

$$y = \frac{3x}{\sqrt{8x} + 1}$$

При цьому припущенні шуканий словесний алгоритм може мати вигляд:

1. Прочитати задане значення x .
2. Помножити x на 8.
3. З результату другої дії знайти квадратний корінь.
4. До результату третьої дії додати 1.
5. Помножити x на 3.
6. Результат п'ятої дії розділити на результат четвертого дії.
7. Записати значення результату y .

Наведений вище запис можна зробити більш компактним, скориставшись операцією присвоювання: =

Сенс операції присвоювання полягають в наступному:

Нехай є припис виду

$$y: = A$$

y - змінна,

A - деякий вираз.

Припис означає наступне: виконати всі дії, передбачені формулою A і отриманий результат (число) вважати значенням (тобто привласнити) змінної y .

У лівій частині команди присвоювання завжди повинна стояти змінна. Вираз y правій частині може бути змінною або числом.

Для того щоб зробити наш запис компактнішим скористаємося допоміжними змінними, які широко використовуються в алгоритмізації.

$$y = \frac{3x}{\sqrt{8x + 1}}$$

1. читання x	2
2. a:= 8x	16
3. v:= \sqrt{a}	4
4. c:= v+1	5
5. d:= 3x	6
6. y:=d/c	1,2
7. запис y	
8. кінець	

Операція присвоювання допускає випадки, коли одна змінна може бути зліва і праворуч від знака присвоювання. Наприклад,

$$k := k + 1$$

Це означає, що потрібно до значення змінної, яке вона мала до початку виконання операції присвоювання, додати число 1 і вважати отримане значення новим значенням змінної k .

1. читання x

2. $a := 8x$

3. $a := \sqrt{a}$

4. $a := a + 1$

5. $y := 3x$

6. $y := y/a$

7. запис y

8. кінець

Алгоритм, записаний на мові програмування, називається програмою.

Графічна форма подання алгоритмів є більш наочною. Алгоритм зображується у вигляді послідовності пов'язаних між собою блоків, кожен з яких відповідає виконанню одного або декількох операторів. Таке графічне представлення називається блок-схемою алгоритму.

Умовні графічні позначення символів, які використовуються для складання блок-схеми алгоритму, стандартизовані.

Блок-схемою називається наочне зображення алгоритму, в якій окремі дії (етапи алгоритму) зображуються за допомогою різних геометричних фігур (блоків), а зв'язки між етапами (послідовність виконання етапів) вказуються за допомогою стрілок, що з'єднують ці фігури.







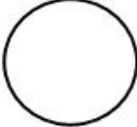
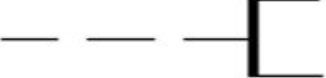
При виконанні блок-схем всередині кожного блоку вказується пояснювальна інформація, яка характеризує дії, що виконуються цим блоком.

Потоки даних в схемах показуються лініями. Напрямок потоку зліва направо і зверху вниз вважається стандартним. У випадках, коли необхідно внести більшу ясність в схему або потік має напрямок відмінний від стандартного, на лініях використовуються стрілки, що вказують цей напрямок.

У схемах слід уникати перетину ліній. Пересічні лінії не мають логічного зв'язку між собою, тому зміни напрямку в точках перетину не допускаються. Якщо дві або більше вхідних ліній об'єднуються в одну вихідну лінію, то місце об'єднання ліній зміщується.

Кількість вхідних ліній не обмежена, лінія що виходить з блоку повинна бути одна, за винятком логічного блоку.

Основні блоки, які використовуються при складанні алгоритмів.

Позначення	Призначення
	Початок-кінець алгоритму
	Будь-яка обчислювальна дія
	Перевірка умови (логічний блок)
	Блок циклу
	Введення даних
	Виведення на друкуючий пристрій
	Символ з'єднувача, що відображає перехід потоку управління між листами
	Коментар

Подання алгоритму у вигляді блок-схеми є проміжним, так як алгоритм в такому вигляді не може бути безпосередньо виконаний комп'ютером, але допомагає користувачеві при створенні (написанні) програми для ПК.

Використання блок-схем дає можливість:

- наочно відобразити базові конструкції алгоритму;
- зосередити увагу на структурі алгоритму, а не на синтаксисі мови;
- аналізувати логічну структуру алгоритму;
- перетворювати алгоритм методом укрупнення (зведення до єдиного блоку) або деталізації - розбиття на ряд блоків;
- використовувати принцип блочності при колективному вирішенні складного завдання;
- здійснити швидку перевірку розробленого алгоритму (на рівні ідеї);
- розібрати більшу кількість навчальних завдань.

Виділяють три основні структури алгоритмів:

1. Лінійна.
2. Розгалужена (альтернатива «якщо-то-інакше» або «якщо-то»).
3. Циклічна (повторення).

ЛІНІЙНІ СТРУКТУРИ

Лінійна структура - є основною. Вона означає, що дії виконуються одна за одною.

Прямокутник, показаний на малюнку, може представляти як одну єдину команду, так і безліч операторів, необхідних для виконання складної обробки даних, де F1 і F2 - деякі команди для відповідного виконавця. Команди записуються за допомогою операції привласнення.

