

“Алгоритми.

Властивості алгоритмів.

Форми подання алгоритму”



План:

- Алгоритм
- Властивості алгоритмів
- Форми подання алгоритму
- Базові структури алгоритмів
- Типи алгоритмів



Процес купівлі хліба можна представити так

- 1) взяти гроші;
- 2) піти в магазин;
- 3) вибрати потрібні хлібобулочні вироби;
- 4) оплатити вартість покупки;
- 5) привнести хліб додому.



Поняття алгоритму

Слово «алгоритм» походить від «algorithmi» – латинської форми написання імені великого математика **Аль-Хорезмі**, який сформулював правила виконання арифметичних дій. Тому спочатку під алгоритмом розуміли тільки правила виконання чотирьох арифметичних дій над багатоцифровими числами в десятковій системі числення. Зараз він є одним із фундаментальних понять інформатики.

Алгоритмізація
процес розробки алгоритму (плану дій) для розв'язування задачі

- **Алгоритм – це послідовність дій, спрямованих на розв'язання поставленої задачі.**

Алгоритм – це точні розпорядження (вказівки, команди) виконавцеві відносно здійснення послідовності дій, які спрямовані на розв'язання певної задачі.



Виконавець алгоритму

- Під виконавцем алгоритму ми розуміємо будь-яку істоту (живу чи неживу), яка спроможна виконати алгоритм.
- Відмінність між виконавцями алгоритмів — людьми і комп'ютерами: якщо люди виконують багато дій, навіть не усвідомлюючи, що при цьому вони виконують якісь алгоритми, то комп'ютери не можуть функціонувати без програм, вказівок яких вони точно додержують.



Основні характеристики виконавця алгоритму

- ◎ **Середовище виконавця** – умови, у яких може діяти виконавець
- ◎ **Елементарні дії** – найпростіші дії, які може виконати виконавець
- ◎ **Система команд виконавця** – сукупність допустимих команд виконавця.

Допустимі команди – команди, які зрозумілі виконавцю і можуть бути ними виконані.
Недопустимі команди – команди, які не можуть бути виконані виконавцем.



Залежно від цілей та шляхів її вирішення алгоритми поділяються на:

- ◎ *Механічні* — задають певні дії, позначаючи їх у єдиній послідовності, забезпечуючи тим самим однозначний результат.
- ◎ *Імовірнісні* — дають програму вирішення задачі кількома шляхами, що приводить до ймовірного досягнення результату.
- ◎ *Евристичні* — досягнення кінцевого результату програми дій однозначно не визначено, використовуються універсальні логічні способи прийняття рішень, засновані на аналогіях, асоціаціях і минулому досвіді розв'язання схожих задач.



Властивості алгоритмів

Дискретність.

- Будь-який алгоритм може бути розбитий на визначеність (чи однозначність)
- Кожна команда алгоритму повинна

Результативність.

- В кожній команді алгоритму строго визначено

Масовість.

- Вхідні дані алгоритму розглядаються як множина

Скінченність.

- Для кожного заданого входу алгоритму повинно завершитися за

Зрозумілість.

- Алгоритм повинен бути зрозумілим для людини, яка його використовує, і не повинен містити жодних двозначностей, які можуть зовсім не закінчуватися.



Властивості алгоритмів

ДИСКРЕТНІСТЬ

Алгоритм розв'язання задачі повинен складатися з послідовності окремих кроків – відокремлених одна від одної команд (указівок), кожна з яких виконується за кінцевий час.

Тільки закінчивши виконання однієї команди, виконавець переходить до виконання іншої.

ВИЗНАЧЕНІСТЬ

Визначеність (однозначність).

Кожна команда алгоритму однозначно визначає дії виконавця і не припускає подвійного тлумачення.

Суворо визначеним є й порядок виконання команд.



Властивості алгоритмів

РЕЗУЛЬТАТИВНІСТЬ

Виконання алгоритму не може закінчуватися невизначеною ситуацією або зовсім не закінчуватися.

Будь-який алгоритм передбачає, що його виконання при допустимих початкових даних за кінцеве число кроків приведе до очікуваного результату.

МАСОВІСТЬ

Алгоритм має передбачати можливість зміни початкових (вхідних) даних у деяких допустимих межах і можливість використання його для розв'язання задач одного класу (універсальність алгоритму).



Властивості алгоритмів

ЗРОЗУМІЛІСТЬ

Щоб виконавець міг досягти поставленої перед ним мети, використовуючи даний алгоритм, він повинен уміти виконувати кожну його вказівку, тобто розуміти кожну з команд, що входять до алгоритму.

СКІНЧЕННІСТЬ

Виконання алгоритму повинно завершитися за скінченну кількість кроків. Виконання алгоритму не може закінчуватися невизначеною ситуацією або ж зовсім не закінчуватися.



Подання алгоритмів

Алгоритми
подаються за
допомогою
природних або
штучних мов,
схем, рисунків,
знаків тощо

- **Словесний** – подання алгоритмів, як правило, у формі звичайної мови (української, англійської, французької тощо) або спеціально розроблених мов (псевдокодів), при цьому вони можуть бути подані як усно, так і письмово.
- **Словесно-формульний** – подання алгоритмів у навчальній та науковій діяльності за допомогою мови математичних формул, хімічних процесів тощо зі словесними поясненнями природною мовою.
- **Графічний** – подання алгоритмів у вигляді графічних схем (блок-схем або структурних схем) та строків ядро мови програмування, алгоритмів розв'язання задач у вигляді алгоритму до написання програм.
- **Програмні** – подання алгоритмів мовою програмування для їх подальшого опрацювання на комп'ютері.
- **Навчальні алгоритмічні мови (псевдокоди)** – подання алгоритмів за допомогою псевдокодів



Приклад

Словесний запис

- *алгоритму* задаємо конкретні числові значення кутів A, B, C .
- Якщо сума кутів дорівнює 180° , то трикутник існує, в іншому випадку не існує.

Алгоритм кипіння води в чайнику

- Налити воду в чайник
- Запалити сірник
- Відкрити газ
- Запалити газ
- Поставити чайник на плиту
- Зачекати доки чайник закипить



Приклад

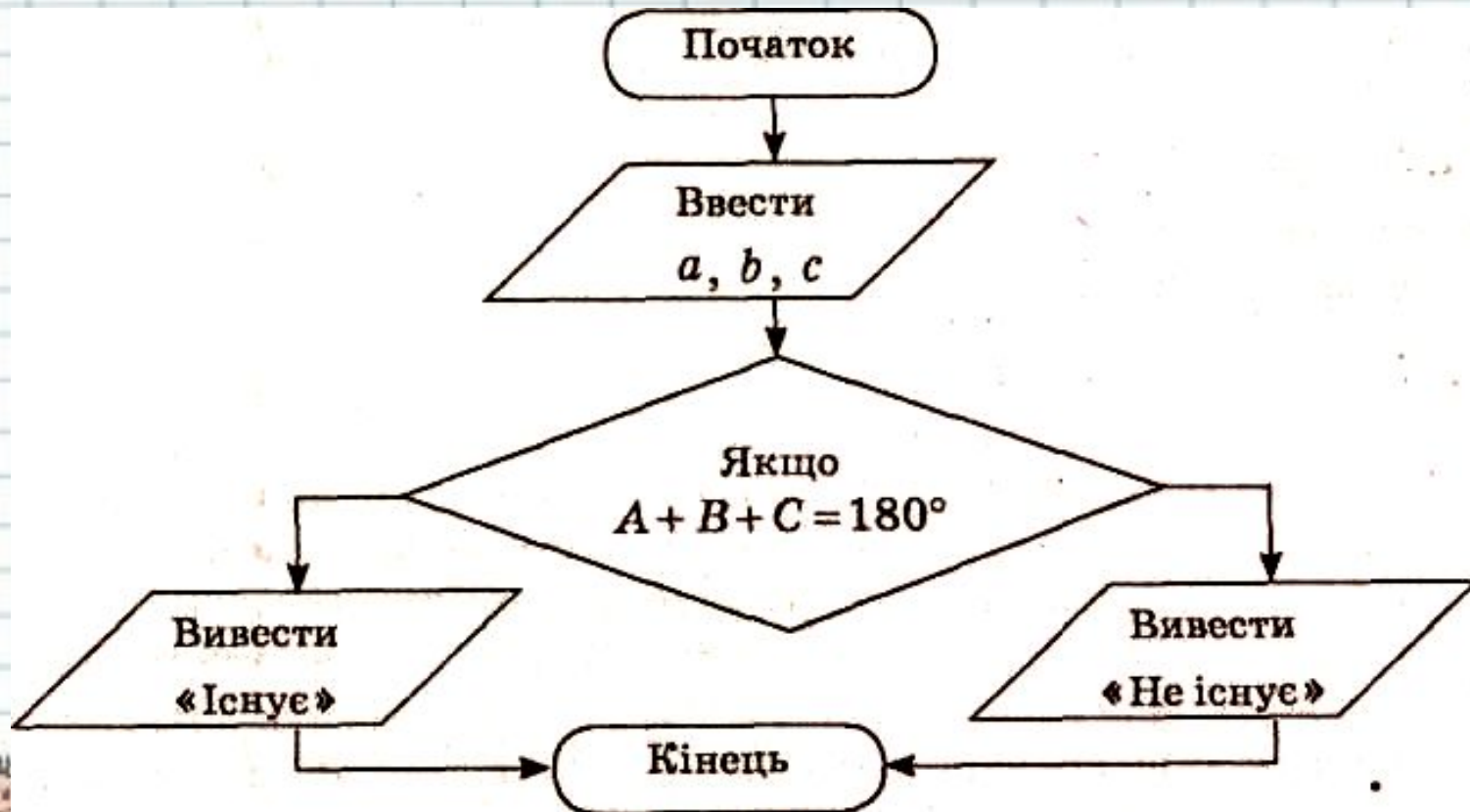
Словесно-формульний запис алгоритму

- ⊙ Задаємо конкретні числові значення кутів A, B, C .
- ⊙ Якщо $A + B + C = 180^\circ$, то трикутник існує, в іншому випадку не існує.



Приклад

Графічний запис алгоритму (блок-схема)



Приклад

Реалізація алгоритму у вигляді програми

Код програми на мові Паскаль

```
Program Z1;  
var a, b, c: Real;  
begin write(' Введіть значення кутів трикутника  
  ');  
readln(a, b, c);  
if a+b+c=180 then writeln('Існує') else writeln('  
  Не існує');
```

end.



Типи алгоритмів:

- ◎ лінійні алгоритми;
- ◎ алгоритми з розгалуженнями;
- ◎ алгоритми з повтореннями.



Лінійні алгоритми

Алгоритм, у якому команди виконуються в порядку їх запису, тобто послідовно один за одним, називається лінійним.

Наприклад, лінійним є наступний алгоритм посадки дерева:

- 1) викопати в землі ямку;
- 2) вилучити в ямку саджанець;
- 3) засипати ямку із саджанцем землею;
- 4) полити саджанець водою.



Алгоритми з розгалуженнями

Форма організації дій, при якій залежно від виконання деякої умови відбувається одна або інша послідовність кроків, називається **розгалуженням**.

Логікові ухвалення рішення можна описати так: **ЯКЩО ТО ІНАКШЕ**

Приклади:

- ЯКЩО прагнеш бути здоровий, ТО загартовуйся, ІНАКШЕ валяйся весь день на дивані;
- ЯКЩО низько ластівки літають, ТО буде дощ, ІНАКШЕ дощу не буде;
- ЯКЩО уроки виучені, ТО йди гуляти, ІНАКШЕ вчи уроки.

У деяких випадках можуть бути відсутні: **ЯКЩО ТО**

Приклад:

- ЯКЩО назвався грибом, ТО лізь в кошик



Алгоритми з повтореннями

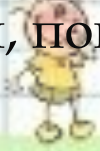
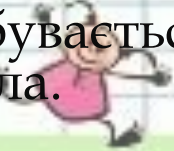
Форма організації дій, при якій виконання однієї й тієї ж послідовності команд повторюється, поки виконується деяке заздалегідь установлене умова, називається **циклом (повторенням)**.

Алгоритм, що містить цикли, називається **циклічним алгоритмом** або **алгоритмом з повтореннями**.

Приклад

- **Натуральне число називають простим, якщо воно має тільки два дільники: одиницю й саме це число** 2, 3, 5, 7 – прості числа; 4, 6, 8 – ні.
- 1) **виписати всі натуральні числа від 1 до n;**
 - 2) **викреслити 1;**
 - 3) **підкреслити найменше з невідмічених чисел;**
 - 4) **викреслити всі числа, кратні підкресленому на попередньому кроці;**
 - 5) **якщо в списку є невідмічені числа, то перейти до кроку 3, а якщо ні, то всі підкреслені числа – прості.**

Це **циклічний алгоритм**. При його виконанні повторення кроків 3-5 відбувається, поки у вихідному списку залишаються невідмічені числа.



-
- ⦿ Що таке алгоритм?
 - ⦿ Які властивості алгоритмів ви знаєте?
 - ⦿ Які форми подання алгоритмів ви знаєте?
 - ⦿ Які ви знаєте типи алгоритмів?

