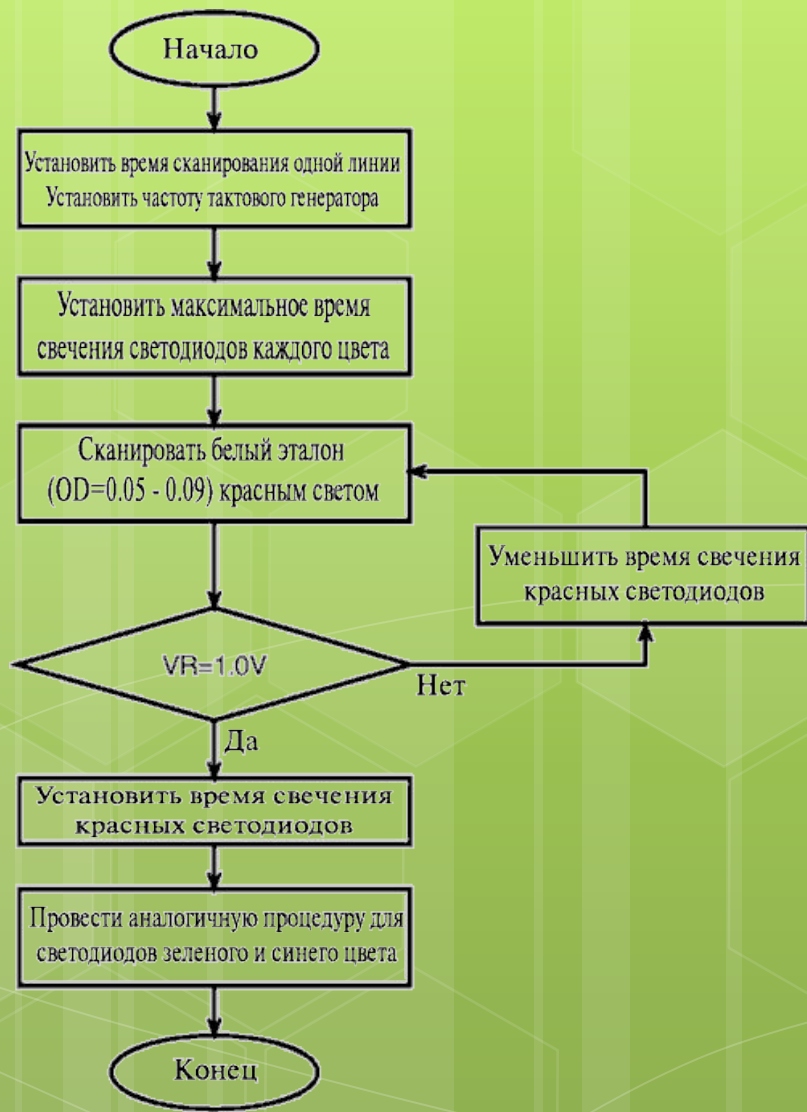
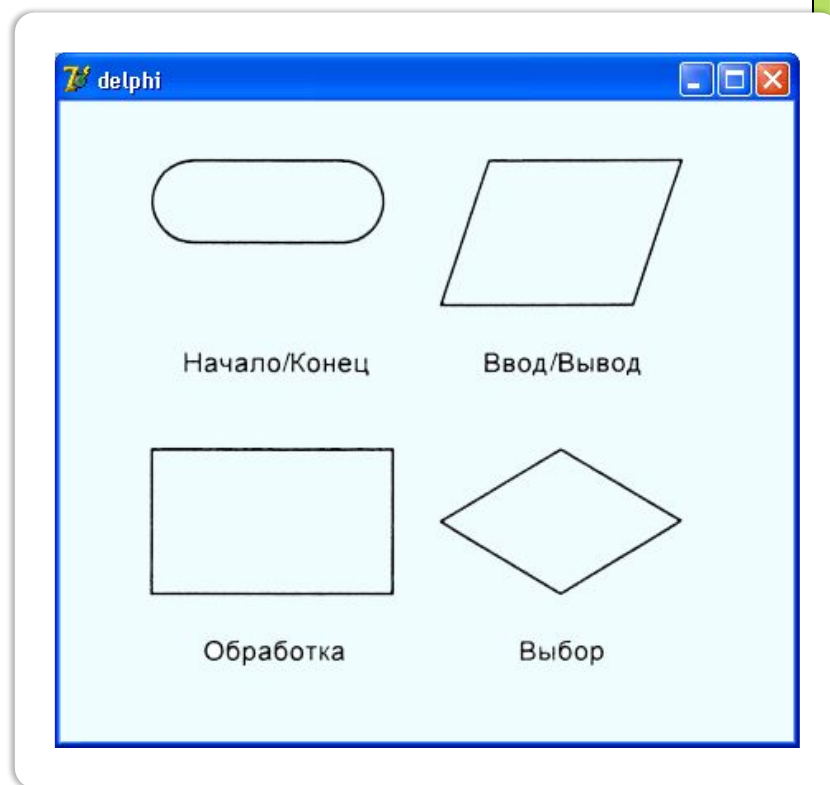




Алгоритм

Алгоритм -

послідовність, система, набір систематизованих правил виконання обчислювального процесу, що обов'язково приводить до розв'язання певного класу задач після скінченного числа операцій. Щ При написанні комп'ютерних програм алгоритм описує логічну послідовність операцій. Для візуального зображення алгоритмів часто використовують блок-схеми.



Історія

Слово алгоритм походить від імені перського вченого, астронома та математика Аль-Хорезмі. Приблизно 825 до н. е. він написав трактат, в якому описав придуману в Індії позиційну десяткову систему числення.



Сучасне поняття алгоритму

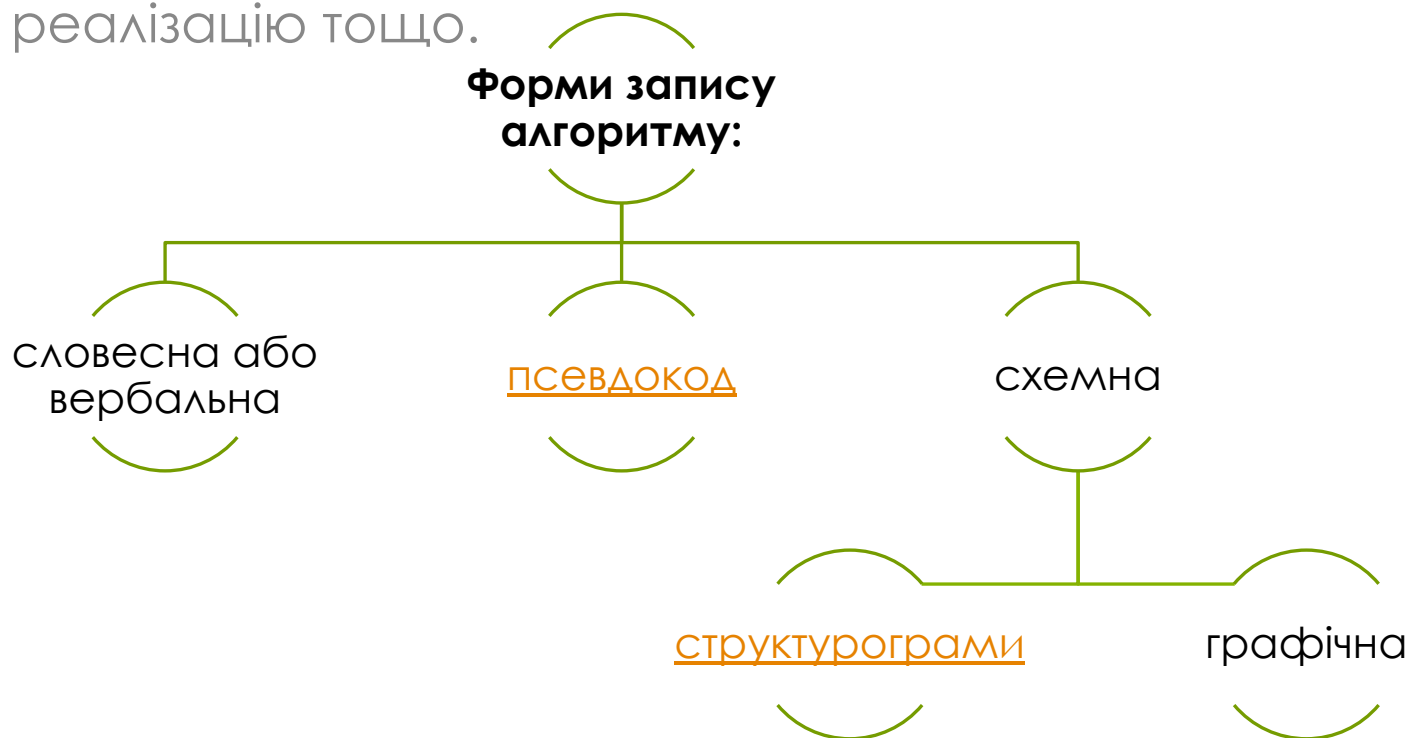
- Сучасне поняття слова «алгоритм» більш широке ніж було раніше при його виникненні. Воно для багатьох співзвучне зі словами метод, спосіб, процедура, програма.

Можна сказати, що **алгоритм - це точна інструкція**, а інструкції зустрічаються практично у всіх сферах нашого життя.

Алгоритм є фундаментальним поняттям інформатики.

Представлення алгоритмів

У процесі розробки алгоритму можуть використовуватись різні способи його опису, які відрізняються за простотою, наочністю, компактністю, мірою формалізації, орієнтації на машинну реалізацію тощо.



Як виникла теорія алгоритмів?

- У 30-х роках ХХ століття виникла теорія алгоритмів.
- До цього часу поняття алгоритму зводилось до набору елементарних кроків: арифметичних дій, перевірки рівностей, нерівностей та інших відношень такого типу.

Але на початку ХХ століття об'єкти, з якими оперували алгоритми, почали ускладнюватися, з'явилась необхідність виконувати операції над векторами, таблицями, множинами тощо.

Постали питання щодо трактовки поняття елементарності кроків, тлумачення однозначності алгоритма, виникла думка, що не для всяких математичних задач можна знайти процедуру розв'язку за кінцевий проміжок часу.

Що досліджує теорія алгоритмів?

- **Теорія алгоритмів** досліджує питання побудови конкретних алгоритмічних моделей, кожна з яких містить конкретний набір елементарних кроків, способів визначення наступного кроку.

Завданням теорії алгоритмів є також дослідження питання про існування чи не існування ефективних алгоритмів розв'язання окремих задач. Найбільшу цінність представляють моделі, які одночасно були б і універсальними, і простими.

Бурхливий розвиток обчислювальної техніки, використання її в дослідженнях багатьох наук привів до створення великої кількості різноманітних алгоритмів в різних прикладних галузях.

Властивості алгоритмів

Скінченність

алгоритм має завжди завершуватись після виконання скінченної кількості кроків. Процедуру, яка має решту характеристик алгоритму, без, можливо, скінченності, називають методом обчислень.

Дискретність

процес, що визначається алгоритмом, можна розчленувати (розділити) на окремі елементарні етапи (кроки), кожен з яких називається кроком алгоритмічного процесу чи алгоритму.

Визначеність

кожен крок алгоритму має бути точно визначений. Дії, які необхідно здійснити, повинні бути чітко та недвозначно визначені для кожного можливого випадку.

Вихідні дані

алгоритм має одне або декілька вихідних даних, тобто, величин, що мають досить визначений зв'язок із вхідними даними.

Ефективність

Алгоритм вважають ефективним, якщо всі його оператори досить прості для того, аби їх можна було точно виконати за скінченний проміжок часу з допомогою олівця та аркушу паперу.

Масовість

властивість алгоритму, яка полягає в тому, що алгоритм повинен забезпечувати розв'язання будь-якої задачі з класу однотипних задач за будь-якими вхідними даними, що належать до області застосування алгоритму.

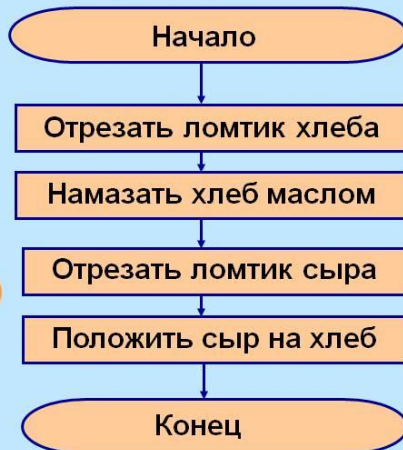
Приклад

В якості прикладу можна навести алгоритм Евкліда. Алгоритм Евкліда — ефективний метод обчислення найбільшого спільного дільника (НСД). Названий на честь грецького математика Евкліда, один з найдавніших **алгоритмів**, що досі використовують. Описаний в Началах Евкліда (приблизно 300 до н. е.), а саме, в книгах VII та X. У сьомій книзі алгоритм описано для цілих чисел, а в десятій — для довжин відрізків. Існує декілька варіантів алгоритму, нижче записано в псевдокоді рекурсивний варіант:

```
function GCD (a, b)
if b = 0
Return a
or
Return GCD (b, a mod
b)
```

Линейный алгоритм

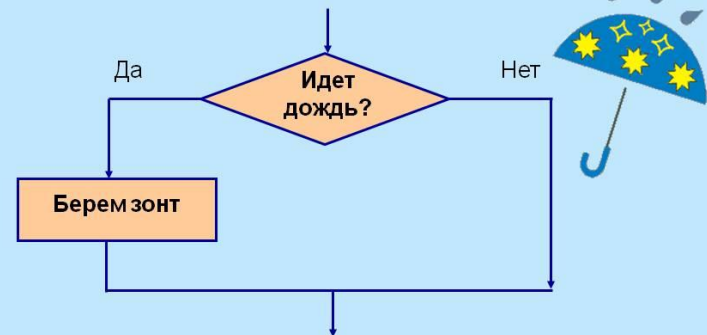
Приготовление бутерброда



4 из 23

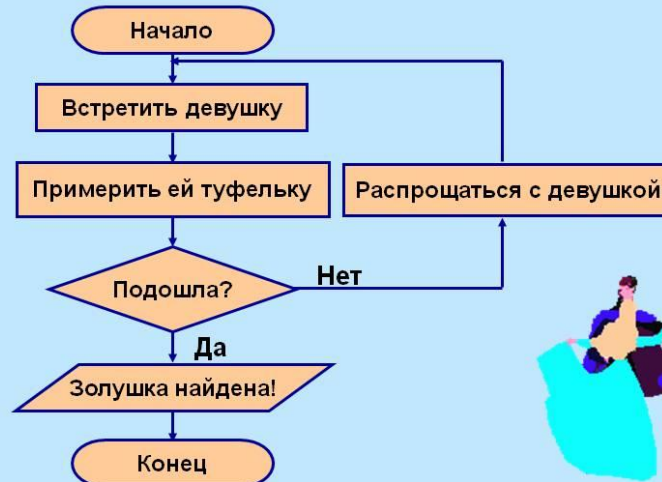
Разветвляющийся алгоритм

Неполное ветвление



8 из 23

Алгоритм поиска Золушки



15 из 23