

ПОНЯТИЕ АЛГОРИТМА



ПОНЯТИЕ АЛГОРИТМА

- Понятие алгоритма является одним из основных понятий вычислительной математики и информатики.

- **Алгоритм — строго определенная последовательность действий для некоторого исполнителя, приводящая к поставленной цели или заданному результату за конечное число шагов.**

- Любой алгоритм составляется в расчете на конкретного исполнителя с учетом его возможностей.

Исполнитель — субъект, способный исполнять некоторый набор команд.

Совокупность команд, которые исполнитель может понять и выполнить, называется **системой команд исполнителя**.

Для выполнения алгоритма исполнителю недостаточно только самого алгоритма. Выполнить алгоритм — значит применить его к решению конкретной задачи, т. е. выполнить запланированные действия по отношению к определенным входным данным. Поэтому исполнителю необходимо иметь **исходные (входные) данные** — те, что задаются до начала алгоритма. В результате выполнения алгоритма исполнитель должен получить искомый результат — **выходные данные**, которые исполнитель выдает как результат выполненной работы. В процессе работы исполнитель может создавать и использовать данные, не являющиеся выходными, — промежуточные данные.

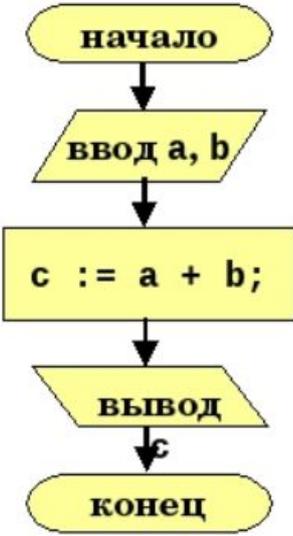
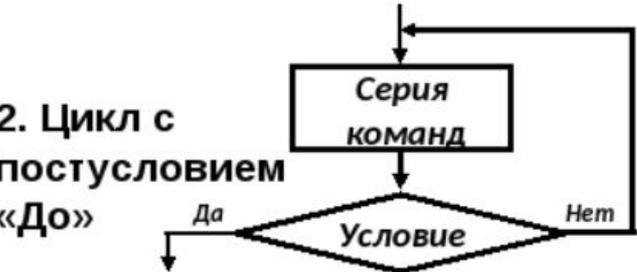
СВОЙСТВА АЛГОРИТМОВ

- **Дискретность.** Процесс решения задачи должен быть разбит на последовательность отдельных шагов — простых действий, которые выполняются одно за другим в определенном порядке. Каждый шаг называется командой (инструкцией). Только после завершения одной команды можно перейти к выполнению следующей.
 -
- **Конечность.** Исполнение алгоритма должно завершиться за конечное число шагов; при этом должен быть получен результат.
 -
- **Понятность.** Каждая команда алгоритма должна быть понятна исполнителю. Алгоритм должен содержать только те команды, которые входят в систему команд его исполнителя.
 -
- **Определенность (детерминированность).** Каждая команда алгоритма должна быть точно и однозначно определена. Также однозначно должно быть определено, какая команда будет выполняться на следующем шаге.
- **Результативность.** Результат выполнения команды не должен зависеть ни от какой дополнительной информации. У исполнителя не должно быть возможности принять самостоятельное решение (т. е. он исполняет алгоритм формально, не вникая в его смысл). Благодаря этому любой исполнитель, имеющий необходимую систему команд, получит один и тот же результат на основании одних и тех же исходных данных, выполняя одну и ту же цепочку команд.
 -
- **Массовость.** Алгоритм предназначен для решения не одной конкретной задачи, а целого класса задач, который определяется диапазоном возможных входных данных.

СПОСОБЫ ПРЕДСТАВЛЕНИЯ АЛГОРИТМОВ:

- **словесная запись** (на естественном языке). Алгоритм записывается в виде последовательности пронумерованных команд, каждая из которых представляет собой произвольное изложение действия;
- **блок-схема (графическое изображение)**. Алгоритм представляется с помощью специальных значков (геометрических фигур) — блоков;
- **формальные алгоритмические языки**. Для записи алгоритма используется специальная система обозначений (искусственный язык, называемый алгоритмическим);
- **псевдокод**. Запись алгоритма на основе синтеза алгоритмического и обычного языков. Базовые структуры алгоритма записываются строго с помощью элементов некоторого базового алгоритмического языка.

ВИДЫ АЛГОРИТМОВ

Линейный	Разветвляющийся	Циклический
<p data-bbox="67 435 483 649">Шаги выполняются последовательно друг за другом</p>	<p data-bbox="627 421 1052 685">Порядок выполнения шагов изменяется в зависимости от условия</p>	<p data-bbox="1284 435 1777 649">Определенная последовательность шагов повторяется несколько раз</p>
 <pre data-bbox="112 806 405 1342">graph TD; Start([начало]) --> Input[/ввод a, b/]; Input --> Process[c := a + b;]; Process --> Output[/ВЫВОД/]; Output --> End([конец]);</pre>	<p data-bbox="521 756 743 835">1. Полное ветвление</p>  <pre data-bbox="540 792 1130 1056">graph TD; Entry(()) --> Cond{Условие}; Cond -- Да --> S1[Серия 1]; Cond -- Нет --> S2[Серия 2]; S1 --> Exit(()); S2 --> Exit;</pre> <p data-bbox="521 1099 772 1185">2. Неполное ветвление</p>  <pre data-bbox="540 1128 1052 1385">graph TD; Entry(()) --> Cond{Условие}; Cond -- Да --> S1[Серия 1]; S1 --> Cond; Cond -- Нет --> Exit(());</pre>	 <pre data-bbox="1188 771 1796 1042">graph TD; Entry(()) --> Cond{Условие}; Cond -- Да --> S[Серия команд]; S --> Cond; Cond -- Нет --> Exit(());</pre> <p data-bbox="1477 913 1787 1049">1. Цикл с предусловием «Пока»</p>  <pre data-bbox="1188 1106 1825 1378">graph TD; Entry(()) --> S[Серия команд]; S --> Cond{Условие}; Cond -- Да --> S; Cond -- Нет --> Exit(());</pre> <p data-bbox="1178 1206 1477 1342">2. Цикл с постусловием «До»</p>

Виды алгоритмов

линейный

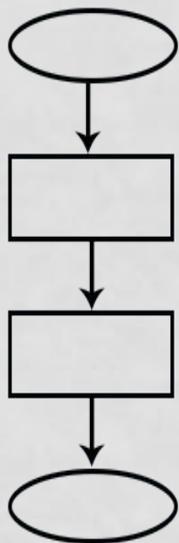
циклический

разветвляющийся

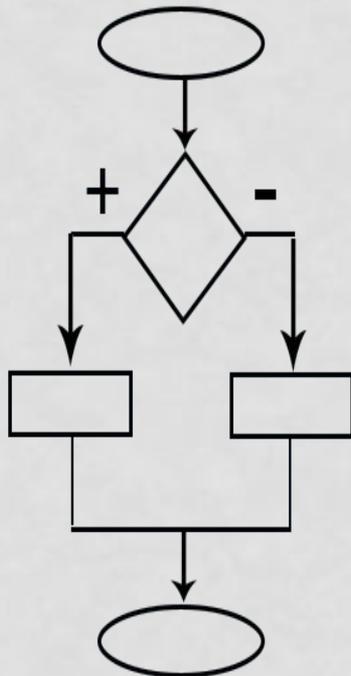
цикл с предусловием

цикл с постусловием

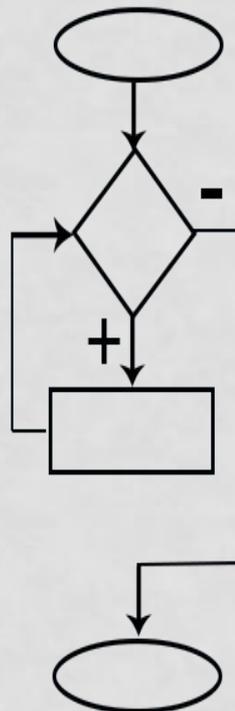
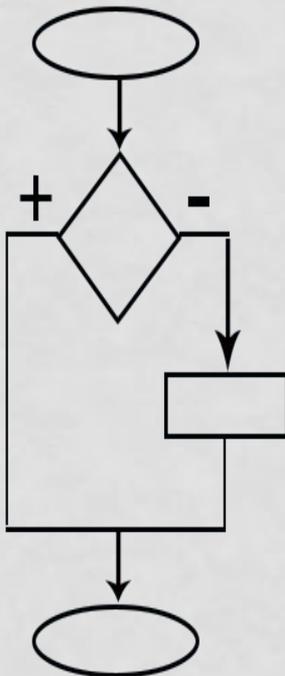
цикл с параметром



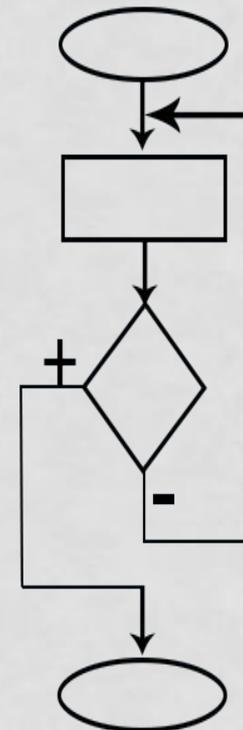
полное ветвление



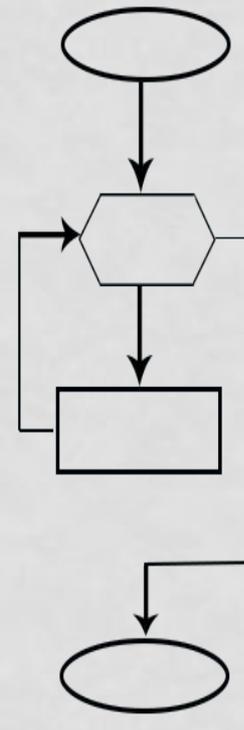
не полное ветвление



цикл типа ПОКА



цикл типа ДО



цикл типа ПОКА