

Алгоритмизация и программирование

Этапы решения задачи на ПК

I. Составление модели задачи

- a) Выделить существенные предположения
- b) Выделить *исходные данные*
- c) Определить *результат*
- d) Установить связь между *исходными данными* и *результатом* (формулы, неравенства и т.д.)

II. Составление алгоритма

III. Составление программы

IV. Ввод исходных данных и анализ результатов

V. Исправление ошибок



СЛОВО

АЛГОРИТМ

произошло от латинского
написания имени
величайшего ученого из
города Хорезма,
Абдуллы (или абу
Джафара) Мухаммеда
бен Муса аль-Хорезми
(Alhorithmi), жившего в
783 – 850 гг.

Алгоритм — это строго определенная последовательность действий при решении задачи.

Алгоритм содержит несколько шагов.

Шаг алгоритма — это каждое отдельное действие алгоритма.

Алгоритмизация:

1) этап решения задачи, состоящий в нахождении по формулировке задачи алгоритма ее решения.

2) раздел информатики, изучающий методы, приемы построения алгоритмов и их свойства (иногда также называемый алгоритмикой).

Исполнитель – это объект, умеющий выполнять определенный набор действий. Исполнителем может быть человек, робот, животное, компьютер.

Система команд исполнителя (СКИ) – это все команды, которые исполнитель умеет выполнять.

Среда исполнителя – обстановка, в которой функционирует исполнитель.

Свойства

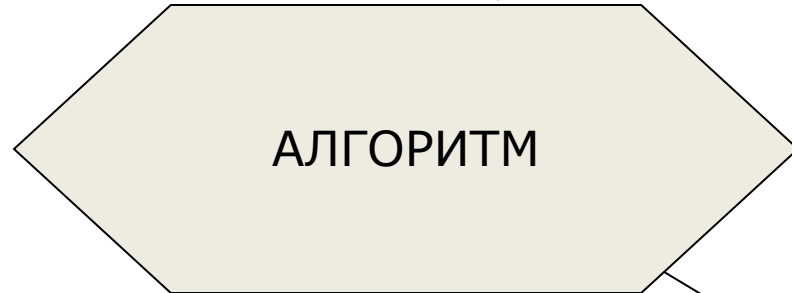
АЛГОРИТМА

Дискретность (прерывность, отдельность) – разбиение алгоритма на шаги

Детерминированность (определенность, точность) – каждое действие должно строго и недвусмысленно определено

Конечность – каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения

Понятность – указания, которые понятны исполнителю



АЛГОРИТМ

Результативность – получение результата за конечное количество шагов

Массовость – использование алгоритма для решения однотипных задач

Классификация алгоритмов по форме представления:

- Словесные
- Табличные
- Графические (блок-схемы)
- Программные

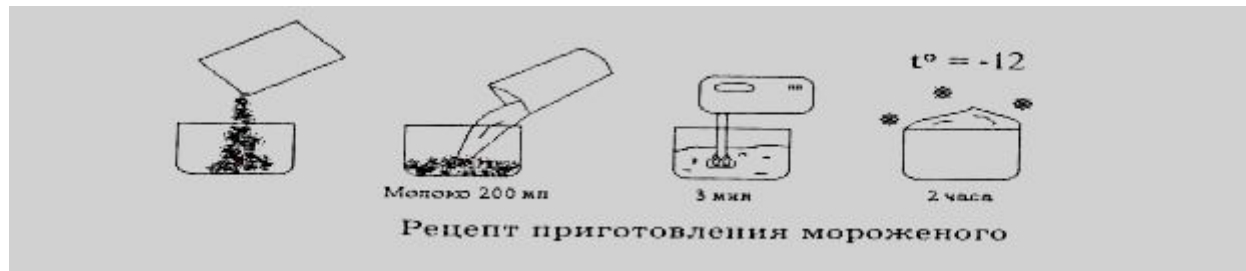







Таблица основных условных обозначений в блок-схемах

| Условное обозначение | Назначение блока |
|---|---|
|  | Начало или конец алгоритма |
|  | Ввод или вывод данных. Внутри блока перечисляются данные через запятую. |
|  | Процесс. Внутри блока записываются математические формулы и операции для обработки данных. |
|  | Проверка условия. Внутри блока записываются логические условия. Имеет два выхода Да(+) и Нет(-) . |
|  | Направление. |

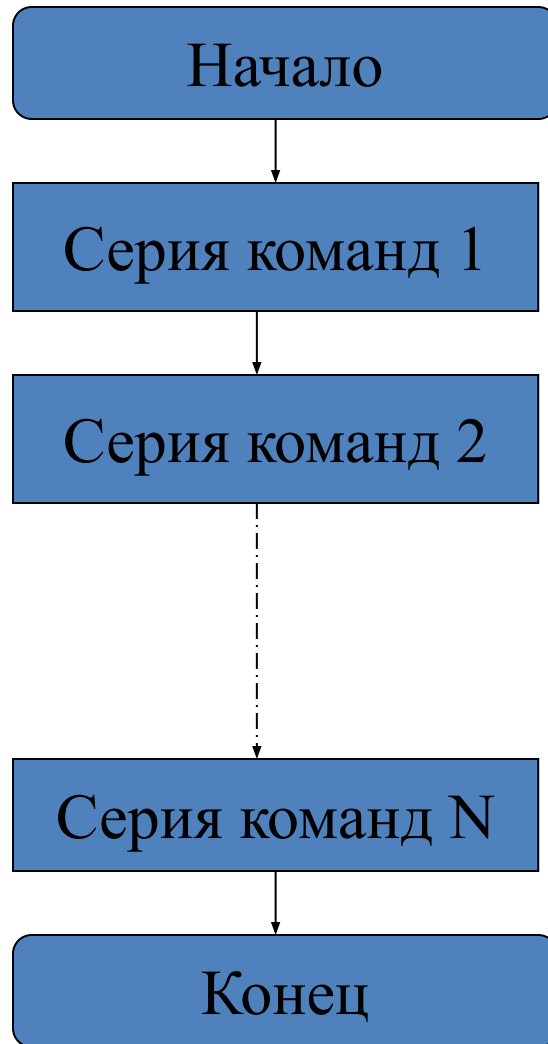
Классификация алгоритмов по структуре:

- Линейный (следование)
- Разветвленный (ветвление, выбор, альтернатива)
- Циклический (повтор)
- Вспомогательный
- Комбинированный

Линейный алгоритм

Линейный алгоритм – это алгоритм, шаги которого выполняются последовательно друг за другом.

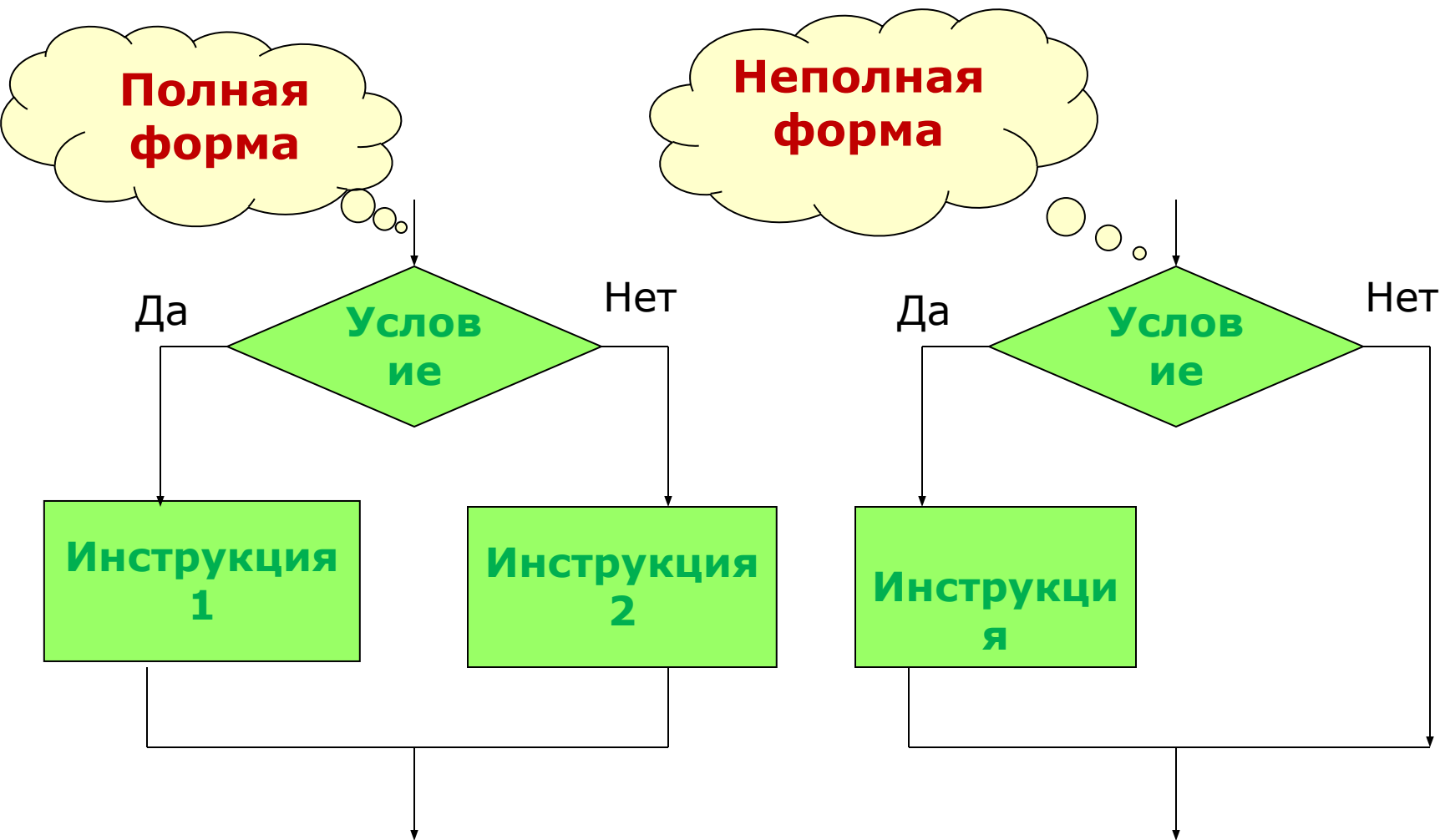
Базовая структура линейного алгоритма:



Разветвляющийся алгоритм –

это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

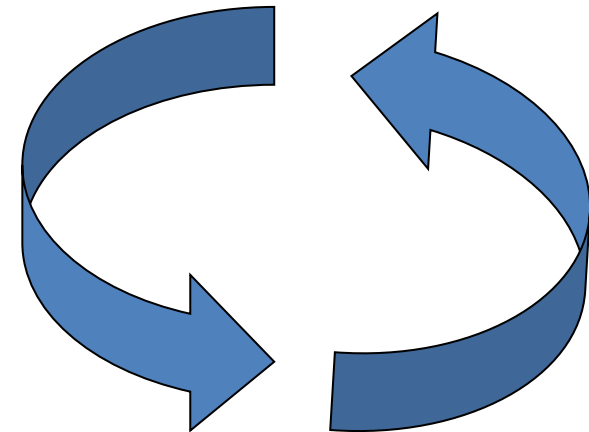
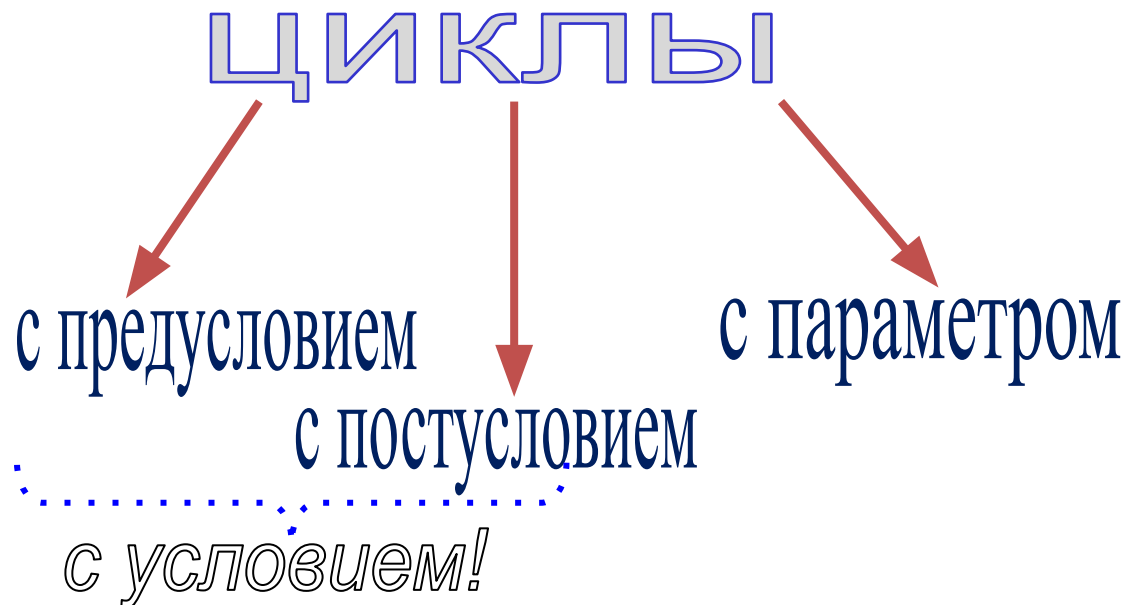
Базовая структура ветвления



Циклический алгоритм

Цикл – это алгоритмическая конструкция, обеспечивающая многократное повторение оператора(ов) – команд исполнителю.

- Виды циклов

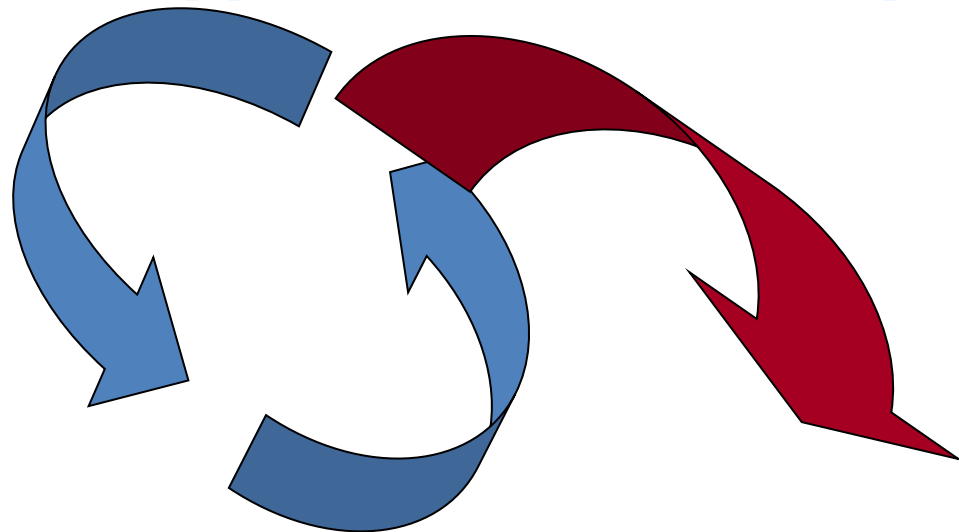


ЦИКЛЫ с условием

Условие – это некоторое утверждение, которое **обязательно** принимает одно из значений:

- а) истина
- б) ложь.

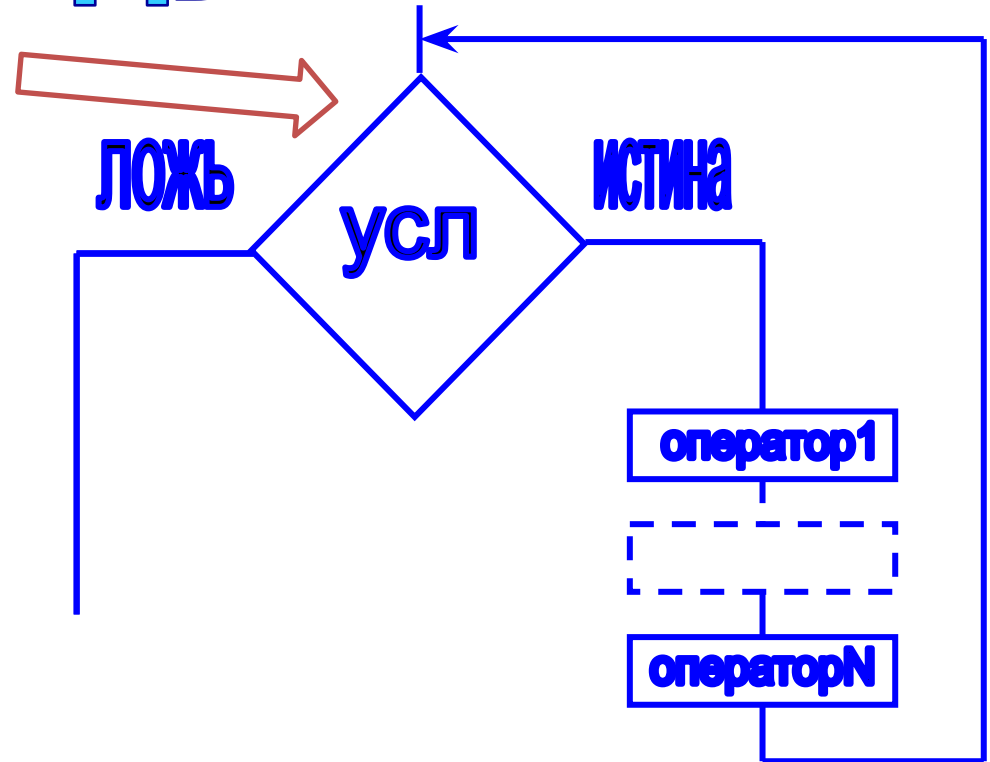
УСЛОВИЕМ
обозначают (определяют)
путь выхода -завершения цикла.



ЦИКЛ

с предусловием

- Условие проверяется перед входом в цикл и называется «Условием выполнения цикла».
- Если условие принимает значение **ЛОЖЬ**, то цикл не



ЦИКЛ

С ПОСТУСЛОВИЕМ

- ✓ Предполагает сначала выполнение команд, а затем проверку условия.
- ✓ Условие называется «Условием продолжения цикла».
- ✓ Если условие продолжения цикла принимает значение ЛОЖЬ, то происходит возврат к выполнению операторов цикла



ЦИКЛ

с параметром (счетчиком)

- ✓ Предполагает выполнение команд определенное количество раз.
- ✓ Цикл завершается при достижении параметрической переменной предельного значения.



Программирование

Программирование — процесс создания компьютерных программ.

В узком смысле (так называемое **кодирование**) под программированием понимается написание инструкций (программ) на конкретном языке программирования.

В более широком смысле под **программированием** понимают весь спектр деятельности, связанный с созданием и поддержанием в рабочем состоянии программного обеспечения ЭВМ. Иначе это называется «программная инженерия» («инженерия ПО»).

Программирование

Программа – это логически упорядоченная последовательность команд необходимая для управления компьютером.

Программа, с которой работает процессор, представляет собой последовательность чисел, называемую **МАШИНЫМ КОДОМ**.

Написать программу в машинном коде достаточно сложно и поэтому для представления алгоритма в виде, понятном компьютеру, служат **ЯЗЫКИ программирования**.

Программирование

Языки программирования – формальная знаковая система, предназначенная для записи компьютерных программ.

Определяет набор *лексических, синтаксических и семантических правил*, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под ее управлением.

Общее количество языков программирования – *более двух с половиной тысяч*.

Языки программирования

- *Языки программирования* - это искусственные языки.
- Они отличаются от естественных ограниченным, достаточно малым числом слов, значение которых понятно компьютеру (транслятору), и очень строгими правилами записи команд (операторов).
- Совокупность требований для записи команд образуют *синтаксис* языка, а смысл каждой команды – *семантику* языка.
- Процесс поиска ошибок в программе называют *тестированием*, процесс устранения ошибок – *отладкой* программы.

УРОВНИ ЯЗЫКОВ

ПРОГРАММИРОВАНИЯ

- Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется **языком программирования низкого уровня**.
- “Низкий уровень” – это значит, что операторы близки к машинному коду и ориентированы на конкретный тип процессора.
- Языком самого низкого уровня является язык Ассемблера, который представляет каждую машинную команду в виде символьных условных обозначений.
- С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора.

УРОВНИ ЯЗЫКОВ

ПРОГРАММИРОВАНИЯ

Высокоуровневый язык программирования

(язык программирования высокого уровня) – язык программирования, разработанный для быстроты и удобства использования программистом.

Программы, написанные на языках высокого уровня, проще для понимания программистом, но менее эффективны, чем их аналоги, создаваемые при помощи низкоуровневых языков.

Примеры: *C, C++, Java, Python, PHP, Perl, Delphi, Lisp и др.*

Транслятор (англ. *translator* — переводчик) — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Компилятор (англ. *compiler* — составитель, собиратель) читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. *interpreter* — истолкователь, устный переводчик) переводит и выполняет программу строка за строкой.

Откомпилированные программы работают быстрее, но интерпретируемые проще исправлять и изменять.

Спасибо за внимание!!!

