

«Алгоритмы и способы их описания»

Алгоритмическое мышление

– искусство размышлять, умение планировать свои действия, способность предусматривать различные обстоятельства и поступать соответственно с ними.

Навыки алгоритмического мышления способствуют формированию:

- целеустремленность и сосредоточенность;
- объективность и точность;
- логичность и последовательность в планировании и выполнении своих действий;
- умение четко и лаконично выражать свои мысли;
- правильно ставить задачу и находить окончательные пути ее решения;
- быстро ориентироваться в стремительном потоке информации;

Слово «*алгоритм*» пришло с Востока, в результате перевода с арабского на европейские языки имени великого ученого IX века Аль-Хорезми, который изложил правила математических действий над числами в позиционной десятичной системе счисления.

Аль-Хорезми [имя] +
Аритмос [число]
→ алгоритм

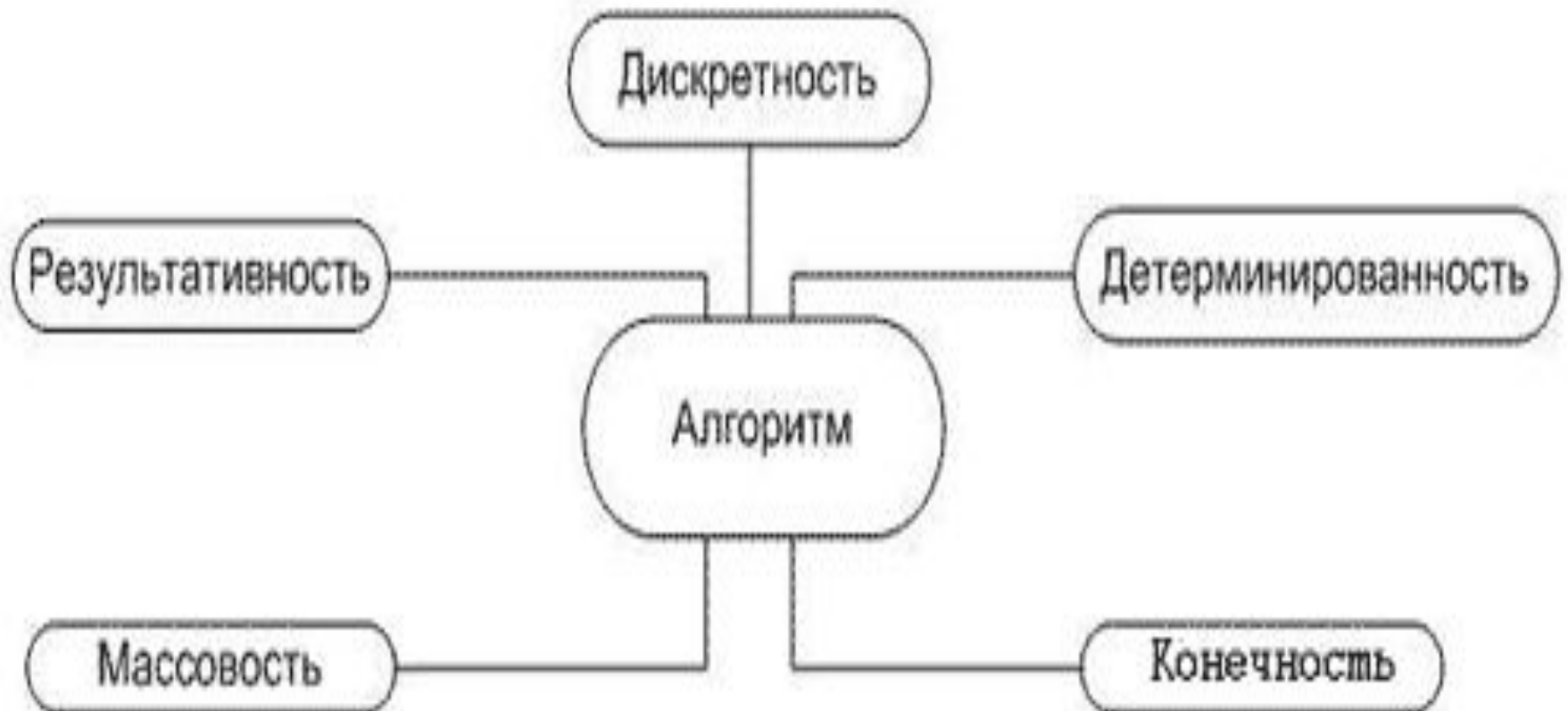
Алгоритм

● – система точных и понятных предписаний (команд, инструкций) о содержании и последовательности выполнения конечного числа действий, необходимых для решения любой задачи данного типа. Как всякий объект, алгоритм имеет название (имя). Также алгоритм имеет **начало и конец**.

Кто может являться исполнителем алгоритмов?

В качестве исполнителя алгоритмов можно рассматривать человека, любые технические устройства, среди которых особое место занимает компьютер. Компьютер может выполнять только точно определенные операции, в отличие от человека, получившего команду.

Алгоритм обладает следующими свойствами:



- **Дискретность** - (от лат. discretus – разделенный, прерывистый) указывает, что любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке. Образованная структура алгоритма оказывается дискретной: только выполнив одну команду, исполнитель сможет приступить к выполнению следующей.

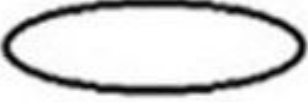
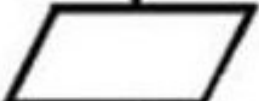
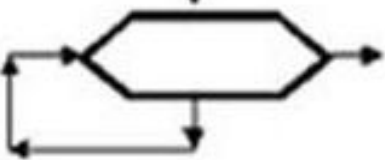
- **Детерминированность** (– определенность, точность) указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.
- **Конечность** определяет, что каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.

- **Результативность** требует, чтобы в алгоритме не было ошибок, т.е. при точном исполнении всех команд процесс решения задачи должен прекратиться за конечное число шагов и при этом должен быть получен определенный постановкой задачи результат (ответ).
- **Массовость.** Это свойство показывает, что один и тот же алгоритм можно использовать с разными исходными данными, т.е. применять при решении всего класса задач данного типа, отвечающих общей постановке задачи. Пример: алгоритмы «Решение квадратного уравнения», «Приготовить бутерброд».

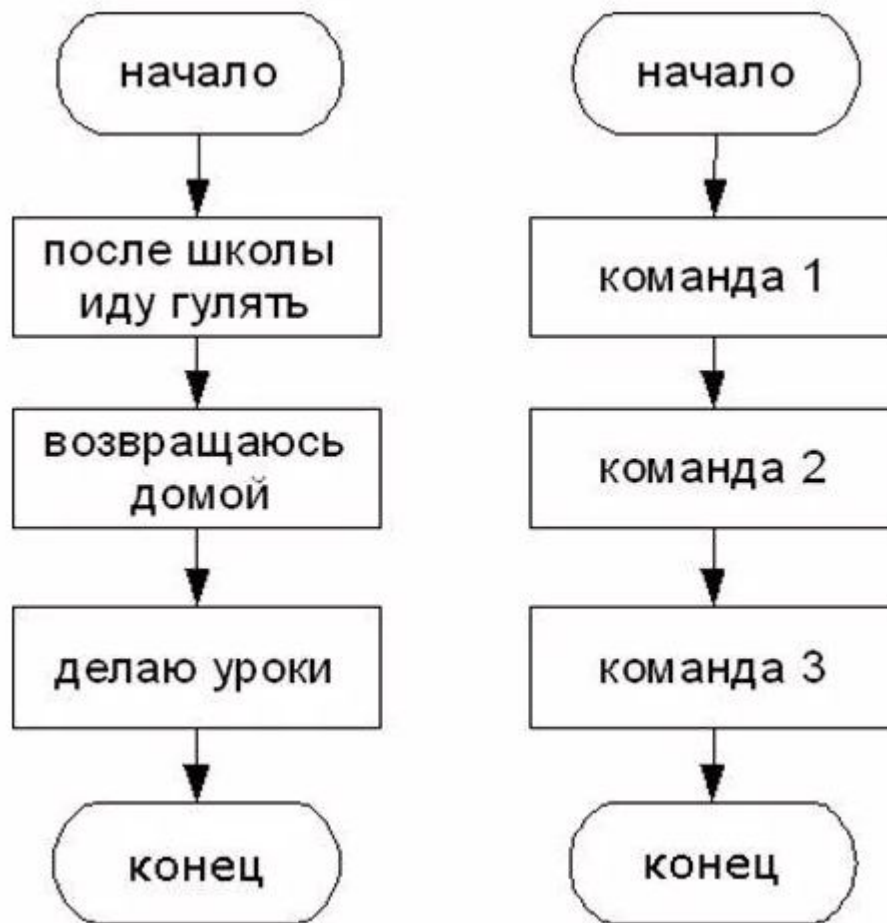
Типовые конструкции алгоритмов:

- Линейная
- Циклическая
- Разветвляющаяся
- Вспомогательная

Элементы блок-схем алгоритмов

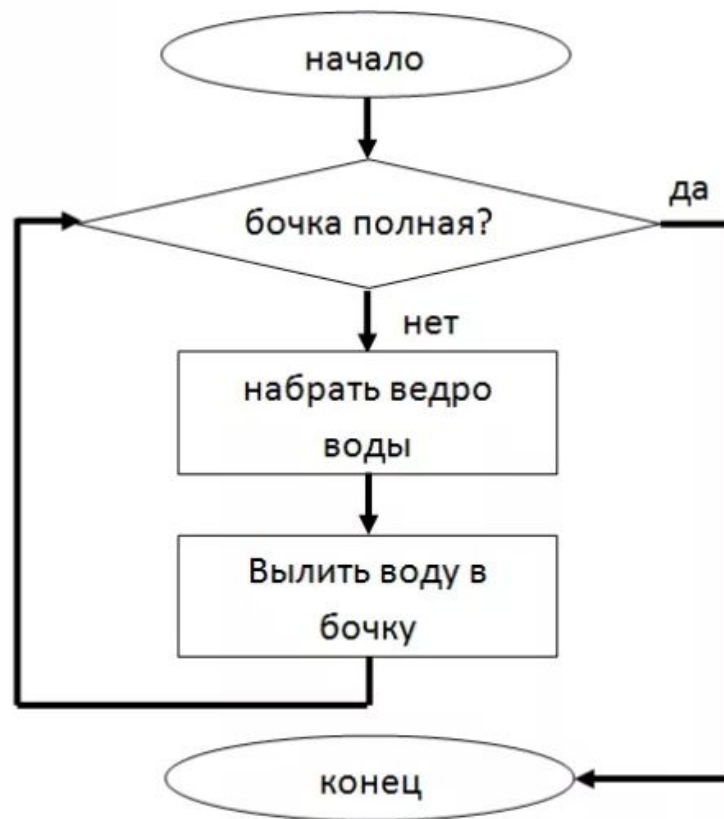
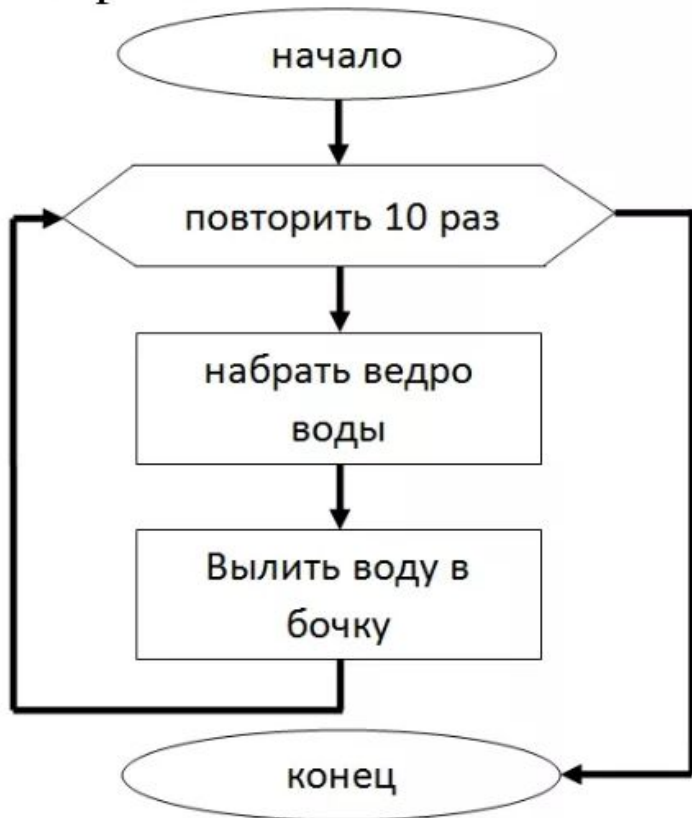
| № | фигура | Что обозначает | Кол-во входов и выходов |
|---|---|-------------------------------------|---------------------------------------|
| 1 |  | Начало Конец | 1 выход 1 вход |
| 2 |  | Действия | 1 вход и 1 выход |
| 3 |  | Ввод и вывод | 1 вход и 1 выход |
| 4 |  | Условие (если, циклы с условием) | 1 вход и 2 выхода |
| 5 |  | Цикл с параметром (для) | 1 вход, 1 выход и циклический круг |

- **Линейный** (последовательный) алгоритм – описание действий, которые выполняются однократно в заданном порядке.

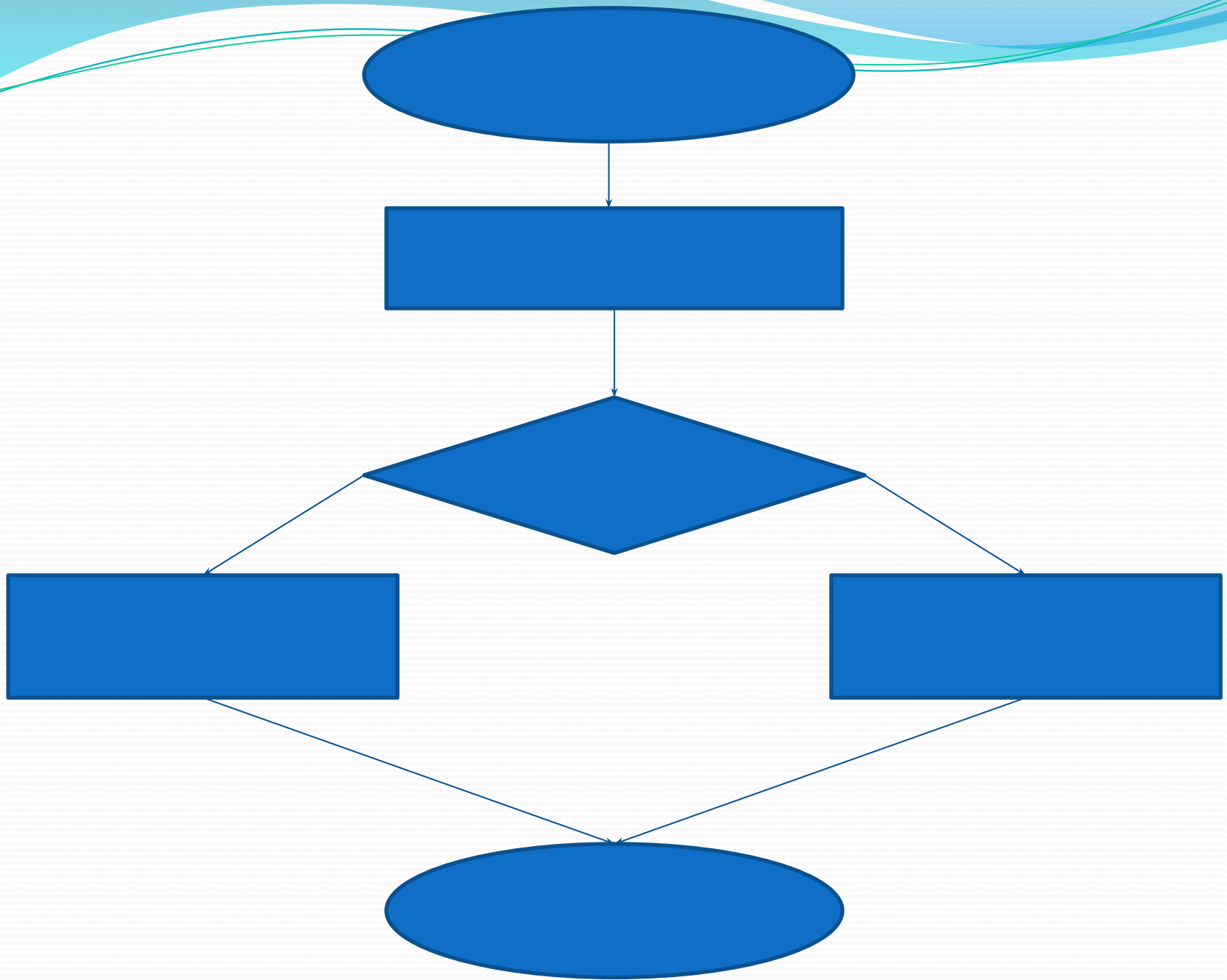


Циклический – описание действий или группы действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Совокупность повторяющихся действий – тело цикла.

Блок-схемы этих алгоритмов выглядят следующим образом:



● **Разветвляющийся** – алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий. Условие – выражение, находящееся между словом «если» и словом «то» и принимающее значение «истина» (ветвь «да») или «ложь» (ветвь «нет»). Возможна полная и неполная форма ветвления.



Вспомогательный – алгоритм, который можно использовать в других алгоритмах, указав только его имя. Вспомогательному алгоритму должно быть присвоено имя.

