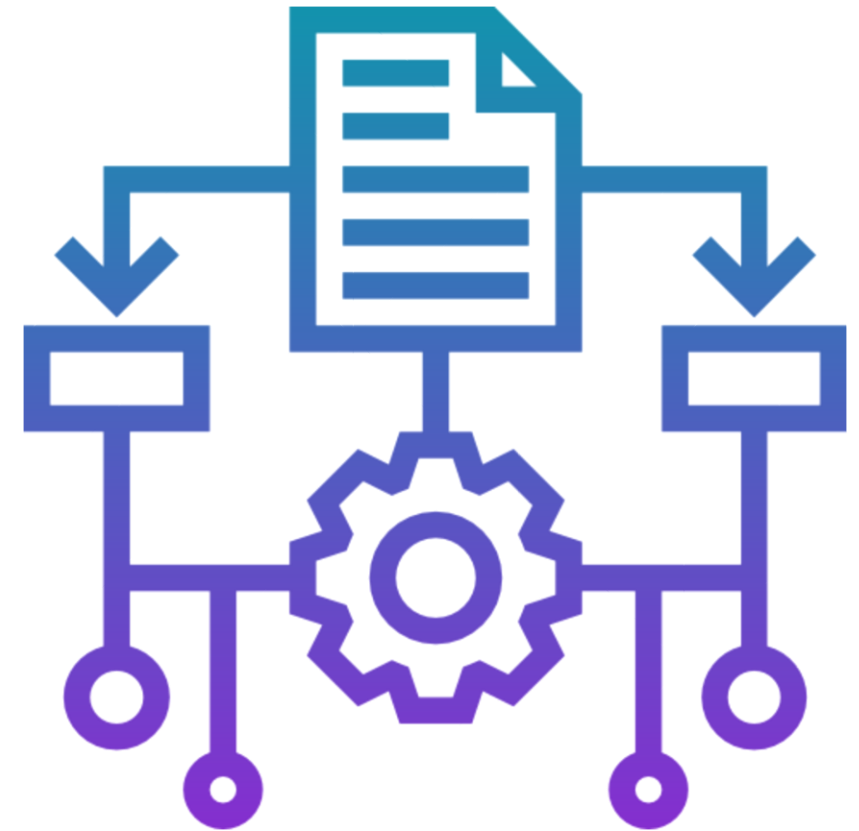
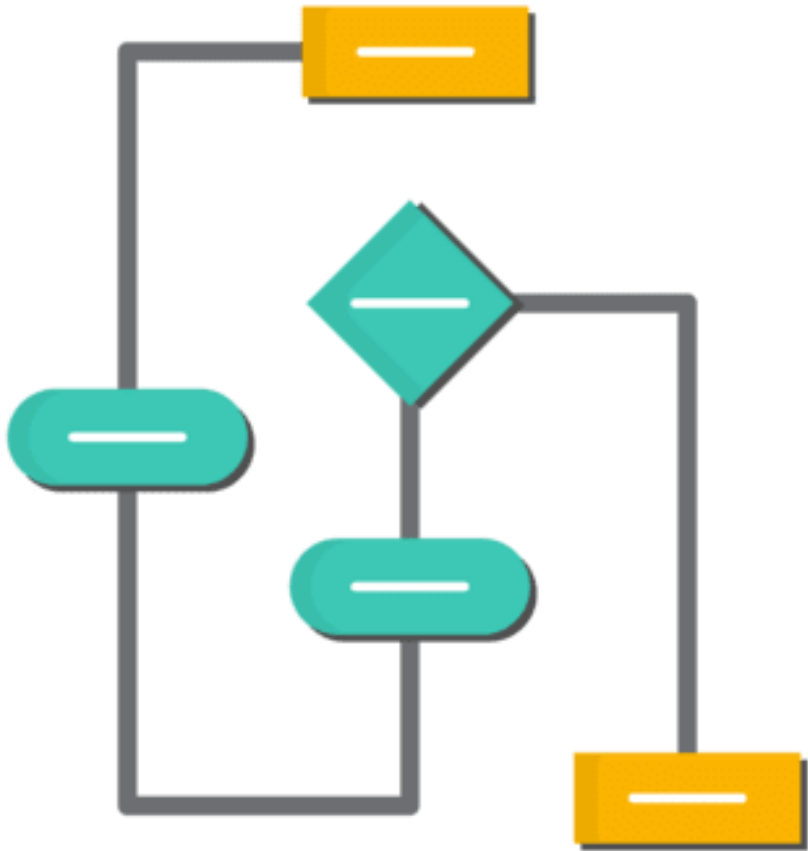
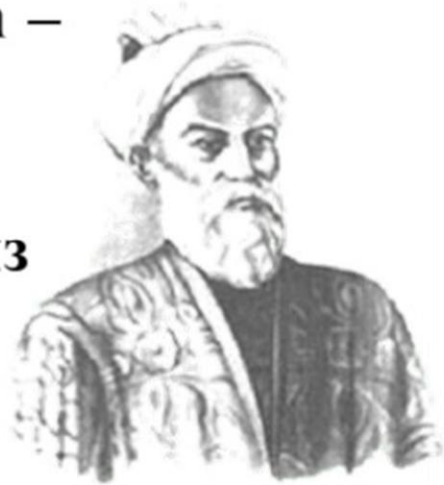


# Алгоритмы и способы их описания



Слово алгоритм произошло от **algorithm** – латинского написания имени аль – Хорезми, величайшего ученого из города Хорезма, Мухамеда бен Мусу, жившего в 783 – 850 гг.



**Алгоритм** (лат. *algorithmi* — от имени среднеазиатского математика Аль-Хорезми<sup>[1]</sup>) — совокупность точно заданных правил решения некоторого класса задач или набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.

Понятие алгоритма относится к первоначальным, основным, **базисным** понятиям математики.

Вычислительные процессы алгоритмического характера известны человечеству с глубокой древности:

- ✓ арифметические действия над целыми числами;
- ✓ нахождение наибольшего общего делителя двух чисел и т. д.

Однако в явном виде понятие алгоритма сформировалось лишь в начале XX века.





Алгоритм — это четкая **последовательность (порядок)** действий, выполнение которой дает какой-то заранее известный **результат**.

Простыми словами, это **набор инструкций** для конкретной задачи.

Известнее всего этот термин в информатике и компьютерных науках, где под ним понимают инструкции для решения задачи эффективным способом.

**Алгоритм** — это точный набор инструкций, описывающих порядок действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное число шагов.

## «Алгоритм» - последовательность, порядок... А что еще?

- ‡ Ранее в русском языке писали «алгори**ф**м», сейчас такое написание используется редко, но тем не менее имеет место исключение (нормальный алгори**ф**м Маркова).
- ‡ Часто в качестве исполнителя выступает компьютер, но понятие алгоритма необязательно относится к компьютерным программам

### ***Например:***

- ✓ Чётко описанный рецепт приготовления блюда также является алгоритмом, в таком случае исполнителем является человек.
- ✓ Алгоритм работы оператора (продавца) с клиентами.
- ✓ Алгоритм действий преподавателя и студентов при эвакуации
- ✓ А может быть и некоторый механизм, например, ткацкий или токарный станок с числовым управлением.

# Классификация алгоритмов

## *Вычислительные*

преобразуют некоторые начальные данные в выходные, реализуя вычисление некоторой функции

## *Управляющие*

сводится к выдаче необходимых управляющих воздействий либо в заданные моменты времени, либо в качестве реакции на внешние события

**В отличие от вычислительного алгоритма, управляющий может оставаться корректным при бесконечном выполнении.**

# Способы описания алгоритма:

- Словесный
- Формульный
- Табличный
- Графический – с помощью блок-схем
- С помощью алгоритмических языков программирования
- С помощью псевдокода

## Справка

**Алгоритмический язык** — формальный язык, используемый для записи, реализации или изучения алгоритмов.

**Псевдокод** — компактный, зачастую неформальный язык описания алгоритмов, использующий ключевые слова императивных языков программирования, но опускающий несущественные для понимания алгоритма подробности и специфический синтаксис. Предназначен для представления алгоритма человеку, а не для компьютерной трансляции и последующего исполнения программы.



## Как записывается алгоритм?

Алгоритм изображается с помощью **блок-схемы** — последовательности геометрических фигур, в которых записываются команды

Элементы блок-схемы между собой соединяются линиями и стрелками, которые показывают ход выполнения алгоритма.

### Основные элементы блок-схемы:



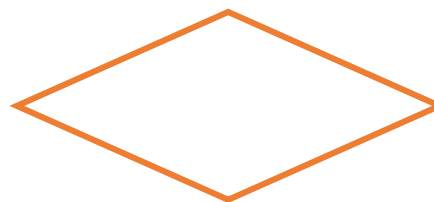
– *Начало, конец*



– *Ввод, вывод данных*



– *Операция, действие*



– *Ветвление по условию*

## Виды алгоритмов:

- **Линейные алгоритмы** – это алгоритм, в котором все действия выполняются последовательно одно за другим.
- **Ветвящиеся алгоритмы** содержат блок «решение» и обеспечивают в зависимости от результата проверки условия выбор одного из альтернативных путей работы алгоритма.
- **Циклические алгоритмы** – это алгоритм, в котором некоторая часть операций (тело цикла — последовательность команд) выполняется многократно.
- **Рекурсивные алгоритмы** – это алгоритм, в описании которого прямо или косвенно содержится обращение к самому себе.





## Неполное ветвление



## Полное ветвление



## Многовариантное ветвление в алгоритме



# Основные свойства алгоритмов:

- **Понятность для исполнителя** — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.
- **Дискретность (прерывность, раздельность)** — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).
- **Определенность** — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.
- **Результативность (или конечность)** состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.
- **Массовость** означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

# Работаем в паре

**Задание:** договоритесь (определитесь), кто какой алгоритм выбирает. Прочитайте алгоритм, объясните его соседу



