

# АЛГОРИТМИЗАЦИЯ

## 1.1 Функциональная схема ЭВМ.

ЭВМ состоит из

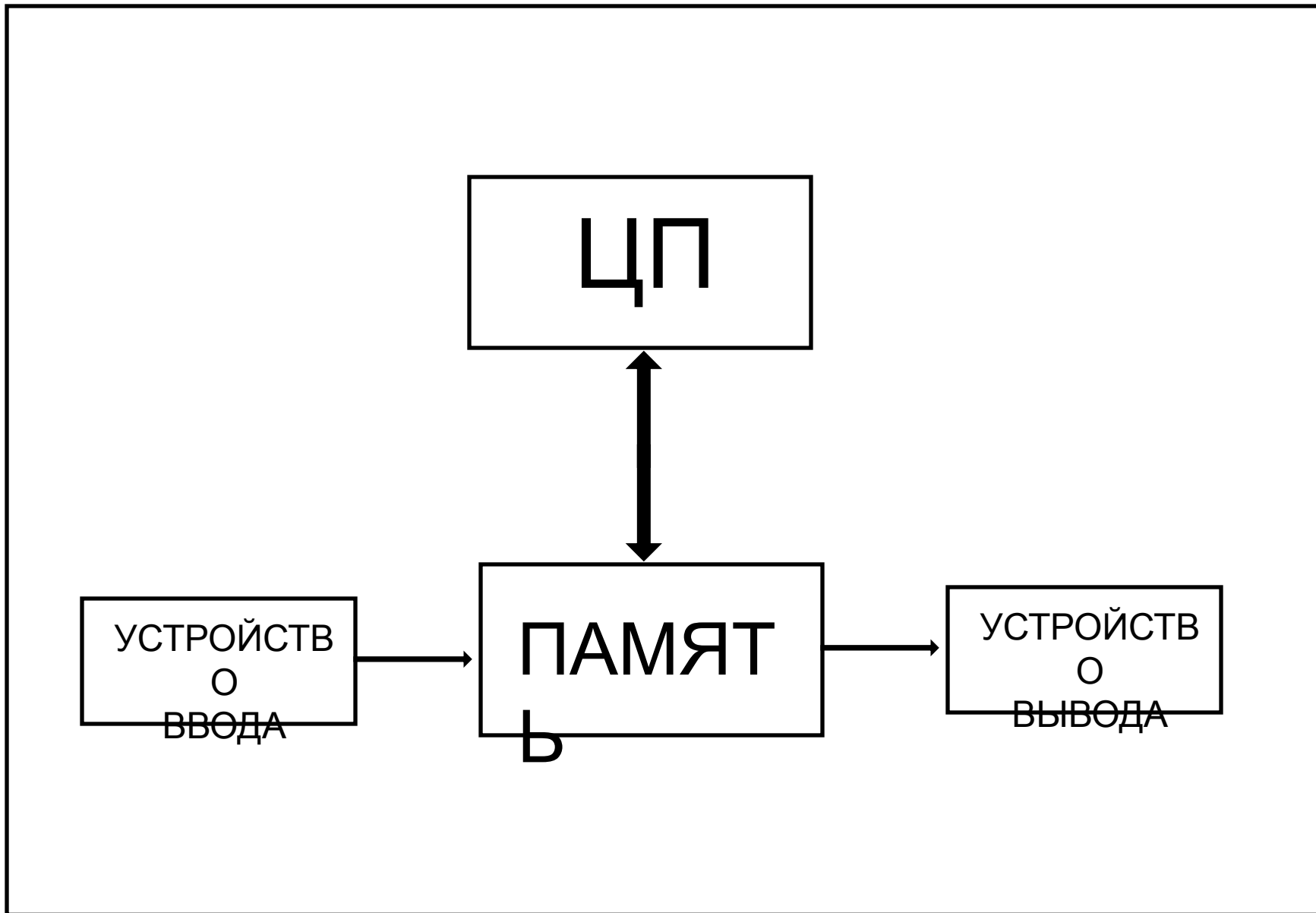
центрального процессора (ЦП) -

устройства, осуществляющего  
обработку информации,

памяти, в которой хранятся данные и  
программы,

устройства ввода, позволяющего вводить  
информацию в память

и устройства вывода.



- ЭВМ работает под управлением программы.
- Программа это набор инструкций.
- Инструкция из памяти поступает в процессор и выполняется.
- Исходные данные для выполнения инструкции загружаются из памяти, результаты - записываются в память.

С точки зрения хранения данных, память состоит из ячеек, обладающих следующими свойствами:

1. в ячейке может храниться одно данное;
2. при записи в ячейку нового значения, бывшее там ранее значение стирается;
3. при считывании значения из ячейки, информация в ней сохраняется, т.е. берётся копия значения, хранящегося в ячейке.

- С понятием ячейки памяти тесно связано понятие переменной.
- Переменная может принимать ряд значений, в каждый момент времени одно значение.
- Эти значения хранятся в ячейках.
- Переменная имеет имя. Это же имя относится к ячейке памяти, в которой хранится значение переменной.

- Чтобы ввести значение в ячейку памяти (присвоить значение переменной) существует два способа: при помощи устройства ввода, т.е. оператором ввода, и из процессора, т.е. оператором присваивания.
- **READ(x)** - переменная  $x$  получает значение, считываемое устройством ввода;
- $x := 10$  - переменная  $x$  получает значение 10;
- $x := z + y$  - переменная  $x$  получает значение суммы значений переменных  $z$  и  $y$ .

## 1.2. Этапы решения задач на ЭВМ.

**Каждая задача, решаемая на ЭВМ, проходит ряд подготовительных этапов:**

- Постановка задачи.
- Разработка метода решения задачи.
- Составление алгоритма решения задачи.
- Написание программы.
- Отладка программы.
- Решение задачи.

## **Рассмотрим более подробно эти этапы.**

- Постановка задачи может быть чисто математической или словесной.
- Разработка метода предполагает или выбор известного математического метода, или построение комбинации таких методов, или словесное описание процесса, приводящего к искомому решению.
- Процесс построения алгоритма - это процесс формализации разработанного метода.



# Понятие алгоритма

Алгоритм - это точное и полное описание метода решения задачи, составленное из инструкций, взятых из заданного набора инструкций.

Алгоритм обладает рядом свойств.

# Свойства алгоритма.

## 1. Определенность

(детерминированность). Это свойство заключается в том, что инструкции просты, понятны и однозначны. Кроме того, выполнив одну инструкцию, известно, какая инструкция должна выполняться следующей.

Благодаря этому свойству процесс выполнения алгоритма носит механический характер, и его может выполнять автомат (ЭВМ).

## 2. Результативность.

Свойство результативности означает, что алгоритм остановиться через конечное число шагов и даст ответ.

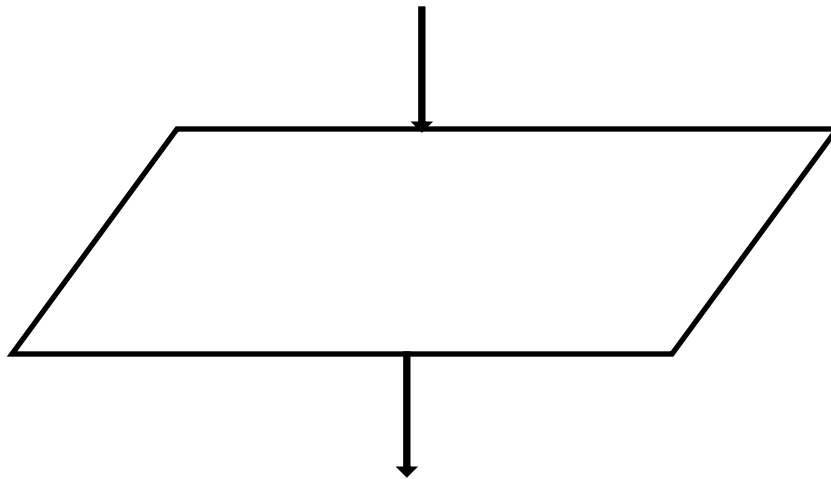
(Может быть отрицательный. Например: *“Система уравнений не имеет решения”*).

3. Массовость. Это свойство подчеркивает то, что алгоритм решает класс задач, а не одну конкретную задачу.

# 1.3. Язык блок-схем для представления алгоритмов

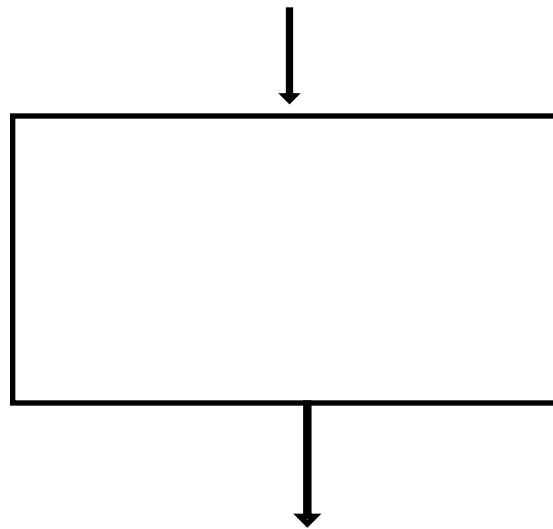
- Язык блок-схем, это наиболее простой и наглядный способ, пригодный для представления алгоритмов любой сложности.
- Блок-схема - это графическое представление алгоритма, где все инструкции пишутся в некоторых геометрических фигурах, а стрелками указывается последовательность выполнения операций.

Операторы ввода и вывода помещаются  
в параллелограмм.



- Если это оператор ввода, то после его выполнения, все перечисленные в нем переменные получают значения, введенные в момент выполнения алгоритма с устройства ввода.
- Если это оператор вывода, то в момент выполнения алгоритма, на устройство вывода будут выведены взятые из памяти значения переменных, перечисленных в операторе.

- Вычислительные операторы записываются в прямоугольных блоках.

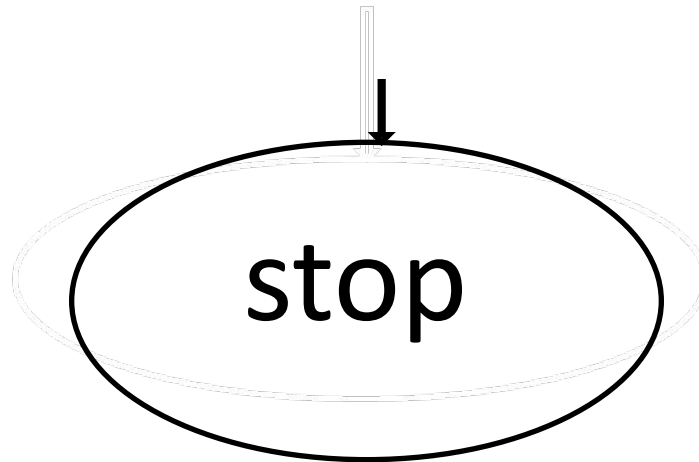


- Блок в виде ромба имеет две выходящие стрелки, но это не нарушает однозначности алгоритма, т.к. в таком блоке записывается условие и, в зависимости от результата его проверки, выполнение алгоритма продолжается по стрелке с надписью “да”, если условие выполнено, или - с надписью “нет”, если условие не выполнено.





В овале записывается оператор **STOP**,  
останавливающий процесс вычисления.



- Блок-схема алгоритма может состоять из любого числа блоков.
- Для наглядности они должны помещаться на одной странице.
- Блок схемы рисуют сверху вниз и, если есть стрелки идущие вверх, то на них не должно быть блоков.

# Пример

Задано целое, положительное значение переменной  $n$  и вещественное значение переменной  $x$ .

Подсчитать значение  $S$ , получаемое по формуле:

$$S = \sum_{i=1}^n \frac{x + i}{i^2}$$

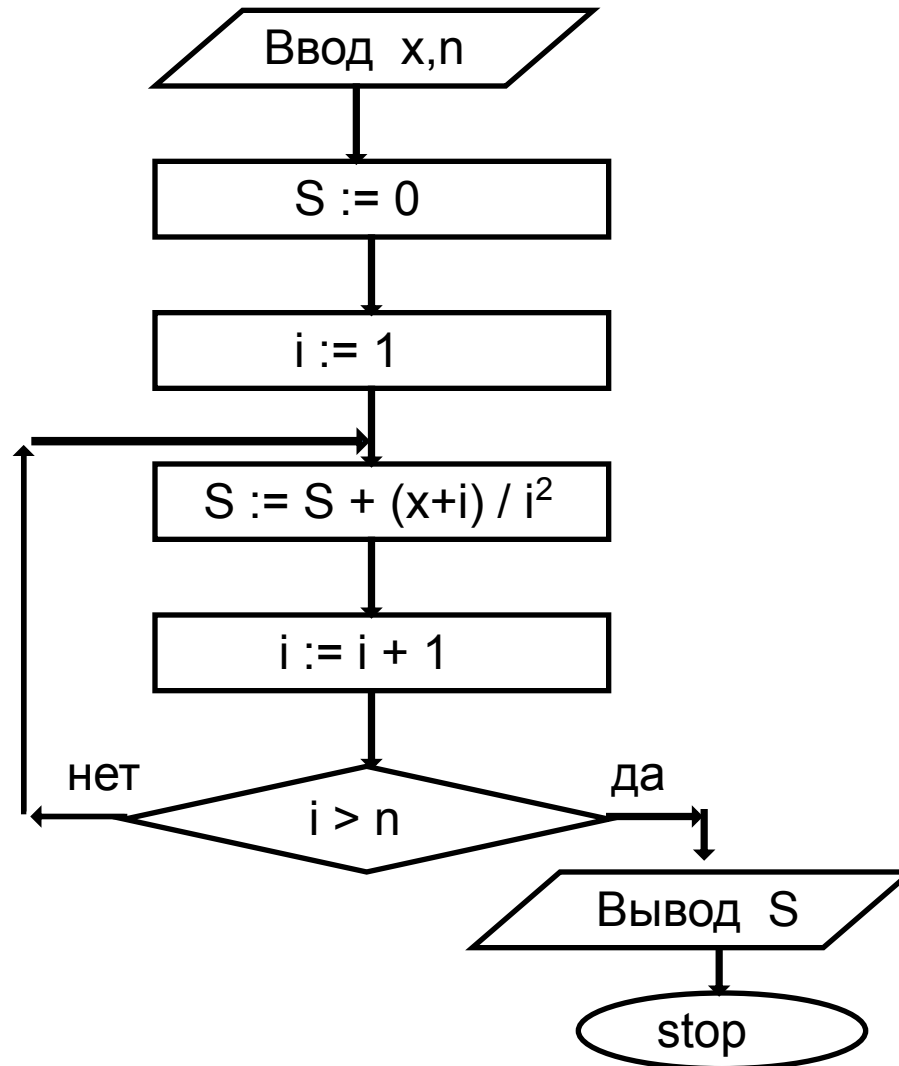
Необходимо составить алгоритм решения этой задачи.

Решим задачу методом накопления.

Это значит сначала положим начальное значение переменной  $S$  равным нулю.

Потом к значению  $S$  прибавим первое слагаемое, затем второе, третье и так далее, до тех пор пока не просуммируем все  $n$  слагаемых.

# Алгоритм решения задачи.



# Пример

Вычислить значение  $K$ -го члена ряда Фибоначчи.

Числовой ряд, в котором каждое последующее число равно сумме двух предыдущих, носит название чисел Фибоначчи.

Первое и второе число ряда равны единице.

# Метод решения

Исходное данное  $K$  – номер искомого члена ряда. По смыслу задачи значение  $K$  должно быть больше двух.

Обозначим через  $X$  искомое значение, а через  $X_1$  и  $X_2$  – значения двух предыдущих членов ряда.

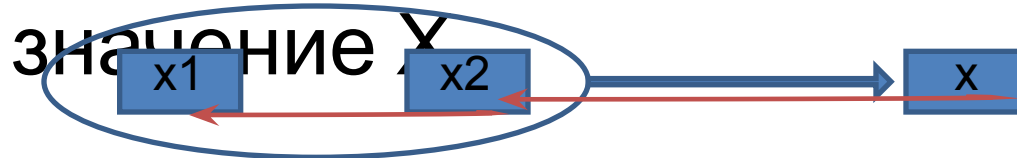
Тогда  $X = X_1 + X_2$  .

Обозначим через  $i$  номер подсчитанного значения  $X$ .

Решение начинается со ввода значения  $K$ .  
После этого надо проверить, является ли полученное значение допустимым, это называется проверкой аномалий.

Если аномалии нет, можно решать задачу.

Решение начинается с присвоения переменным  $X_1$  и  $X_2$  начального значения, после этого можно рассчитать





При этом  $i$  получает следующее значение.

Если  $i = K$ , то результат получен, и его надо напечатать (значение  $X$ ), в противном случае надо подготовить следующий шаг подсчёта  $X$ , для этого надо переписать значение  $X_2$  в  $X_1$  и значение  $X$  в  $X_2$ .

Таким образом, в  $X_1$  и  $X_2$  будут находиться два последних рассчитанных члена ряда, и можно получить следующий член  $X$ .

# Блок-схема алгоритма

