

# *Алгоритмы и исполнители*



# Основные вопросы:

- Понятие алгоритма, свойства алгоритмов. Исполнители алгоритмов, система команд исполнителя.
- Способы записей алгоритмов. Формальное исполнение алгоритмов.
- Основные алгоритмические структуры (линейная, ветвления, выбор, цикл).
- Этапы решения задач с использованием компьютера: формализация, программирование и тестирование.

Само слово «алгоритм» происходит от **algorithmi** - латинской формы написания имени великого математика IX века **Абдулла (или Абу Джафар) Мухаммед Бен Мусса аль-Хорезми**, который сформулировал правила выполнения арифметических действий.

Первоначально под **алгоритмами** и понимали только правила выполнения четырех арифметических действий над многозначными числами.



# Алгоритм

- это описание некоторой последовательности действий, которую нужно совершить для достижения определенной цели



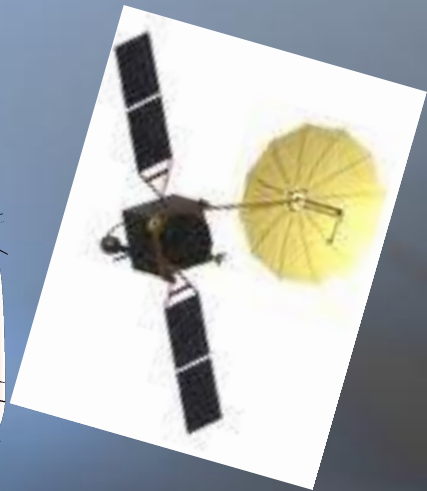




*Каждый  
алгоритм  
предназначен  
для  
определенного  
исполнителя.*

# Исполнитель алгоритма

- это человек, живое существо или автомат (в частности им может быть процессор ЭВМ), умеющий выполнять некоторый набор действий которые можно назвать командами.



# Исполнителя характерезуют

- **Среда** – это «место обитания» исполнителя.
- **Элементарные действия** – это действия, которые исполнитель способен выполнять.

# Исполнителя характерезуют

- Совокупность команд, которые могут быть выполнены исполнителем, называется *системой команд исполнителя.*



# Исполнителя характерезуют

- **Отказы** – возникают если по каким-либо причинам (неизвестная команда, недопустимое состояние среды и т.д.) исполнитель не может выполнить очередную команду алгоритма.

# Свойства алгоритма

1. **Дискретность** – разбиение алгоритма на последовательность отдельных законченных действий.
2. **Понятность** – однозначное понимание каждого шага алгоритма для исполнителя.
3. **Точность** – свойство, когда алгоритм не должен содержать команды, смысл которых может восприниматься неоднозначно.

# Свойства алгоритма

4. **Результативность (конечность)** – свойство, при котором любой алгоритм в процессе выполнения должен приводить к определённому результату. Отрицательный результат также является результатом.
5. **Массовость** – свойство, когда по данному алгоритму должна решаться не одна, а целый класс подобных задач.

# Свойства алгоритма

**6. Универсальность.** Алгоритм должен быть составлен так, чтобы им мог воспользоваться любой исполнитель для решения аналогичной задачи.

# Способы записи алгоритмов

1. **Словесно –  
формульный,**  
описание алгоритма с  
помощью слов и  
формул.





$$A x^2 + B x + C = 0$$

1. Начать.
2. Ввод  $A, B, C$ .
3.  $D = B^2 - 4 A C$ .
4. Если  $D < 0$ , то идти к п. 6.
5. Если  $D > 0$ , то идти к п. 8.
6. Действительных корней нет.
7. Идти к п. 10.
8.  $X_1 = (- B - \sqrt{D}) / 2 A$  ;  $X_2 = (- B + \sqrt{D}) / 2 A$  .
9. Вывести значения  $X_1$  и  $X_2$ .
10. Закончить.

# Способы записи алгоритмов

**2. Алгоязык (псевдокод)**- это средство для записи алгоритмов в аналитическом виде, промежуточном между записью алгоритма на естественном (человеческом) языке и записью на языке ЭВМ (языке программирования).

<u>Алг</u>	имя алгоритма
<u>Дано</u>	условия выполнимости алгоритма
<u>Надо</u>	цель выполнения алгоритма
<u>Нач</u>	
	тело алгоритма (последовательность команд)
<u>Кон</u>	

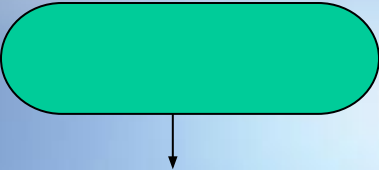
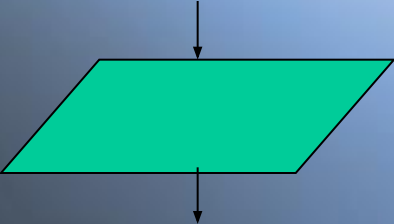
# Способы записи алгоритмов

## 3. Графический способ (Блок-схема)

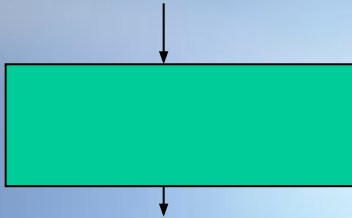
служит для графического изображения структуры алгоритма.

Любая блок-схема представляет набор геометрических фигур или блоков, в этих фигурах можно записывать текст, формулу и вообще любую информацию о том, что надо делать на данном этапе выполнения алгоритма. Последовательность действий указывается стрелкой. Основное преимущество блок-схемы - это наглядность.

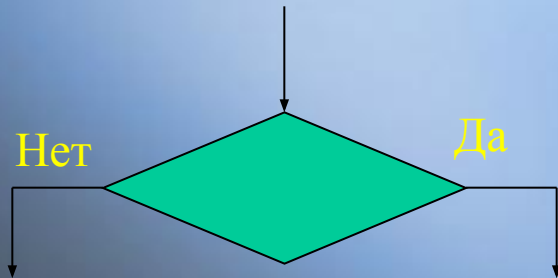
# Блок-схема

Обозначение	Название блоков
	начало
	конец
	Ввод/Вывод данных

# Блок-схема

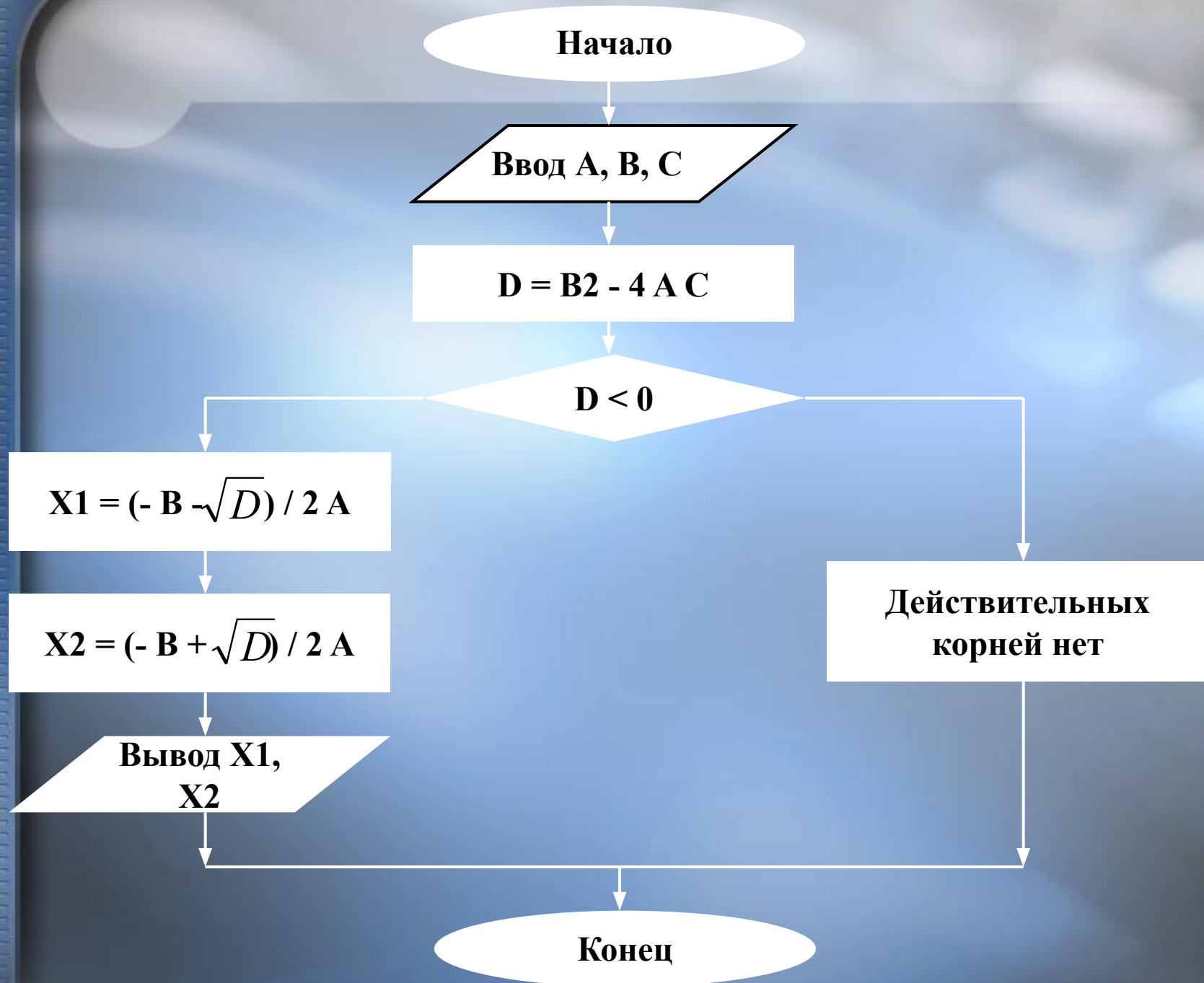


Выполнение операций  
(действий)



Условие





# Способы записи алгоритмов

4. Алгоритм, записанный на понятном компьютеру языке программирования, называется **программой**.



# Виды алгоритмов

```
graph TD; A[Виды алгоритмов] --> B[Линейный]; A --> C[Разветвляющийся]; A --> D[Циклический]
```

Линейный

Разветвляющийся

Циклический

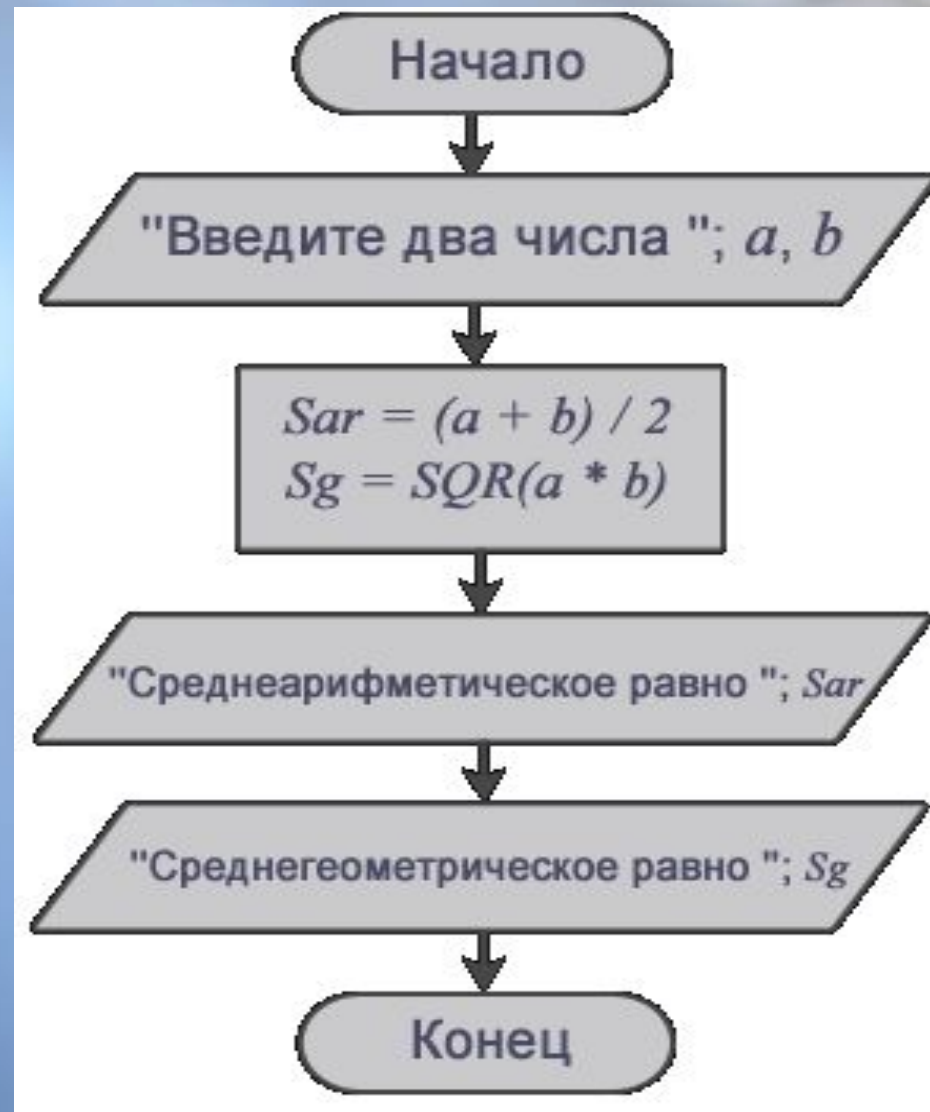
# Основные алгоритмические структуры

1. Линейным называется алгоритм, в котором все этапы решения задачи выполняются строго последовательно.



# Задача № 1

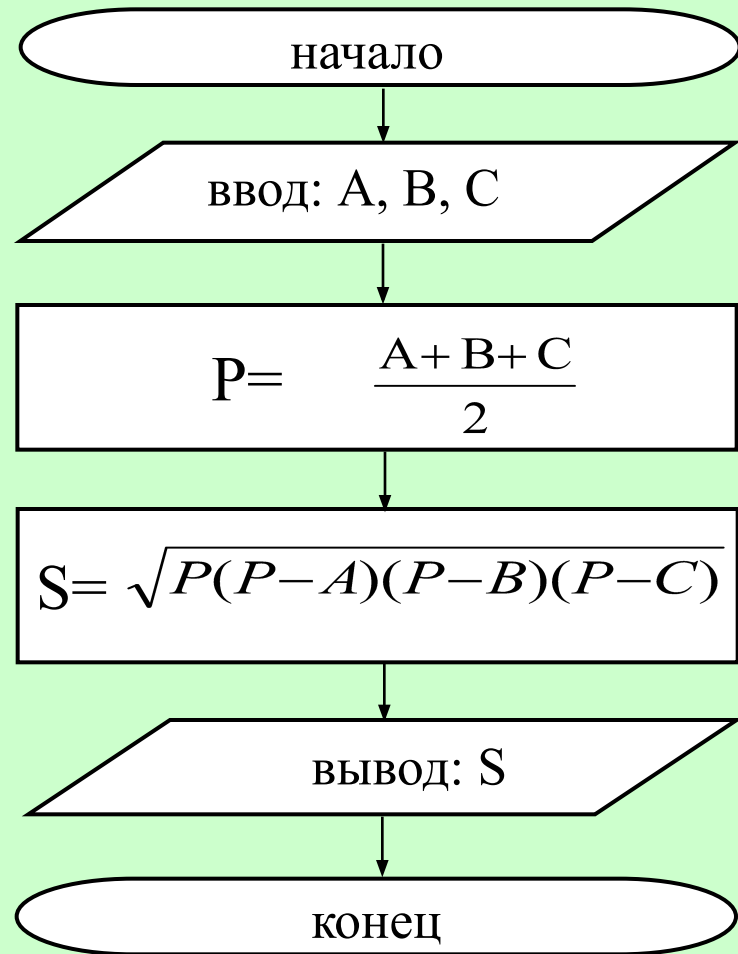
**Составить алгоритм вычисления среднеарифметического и среднегеометрического двух чисел и записать его в виде блок-схемы.**





# Задача № 2

Даны длины сторон  
треугольника  $A, B, C$ .  
Найти площадь  
треугольника  $S$ .  
Составьте блок-  
схему алгоритма  
решения  
поставленной  
задачи.

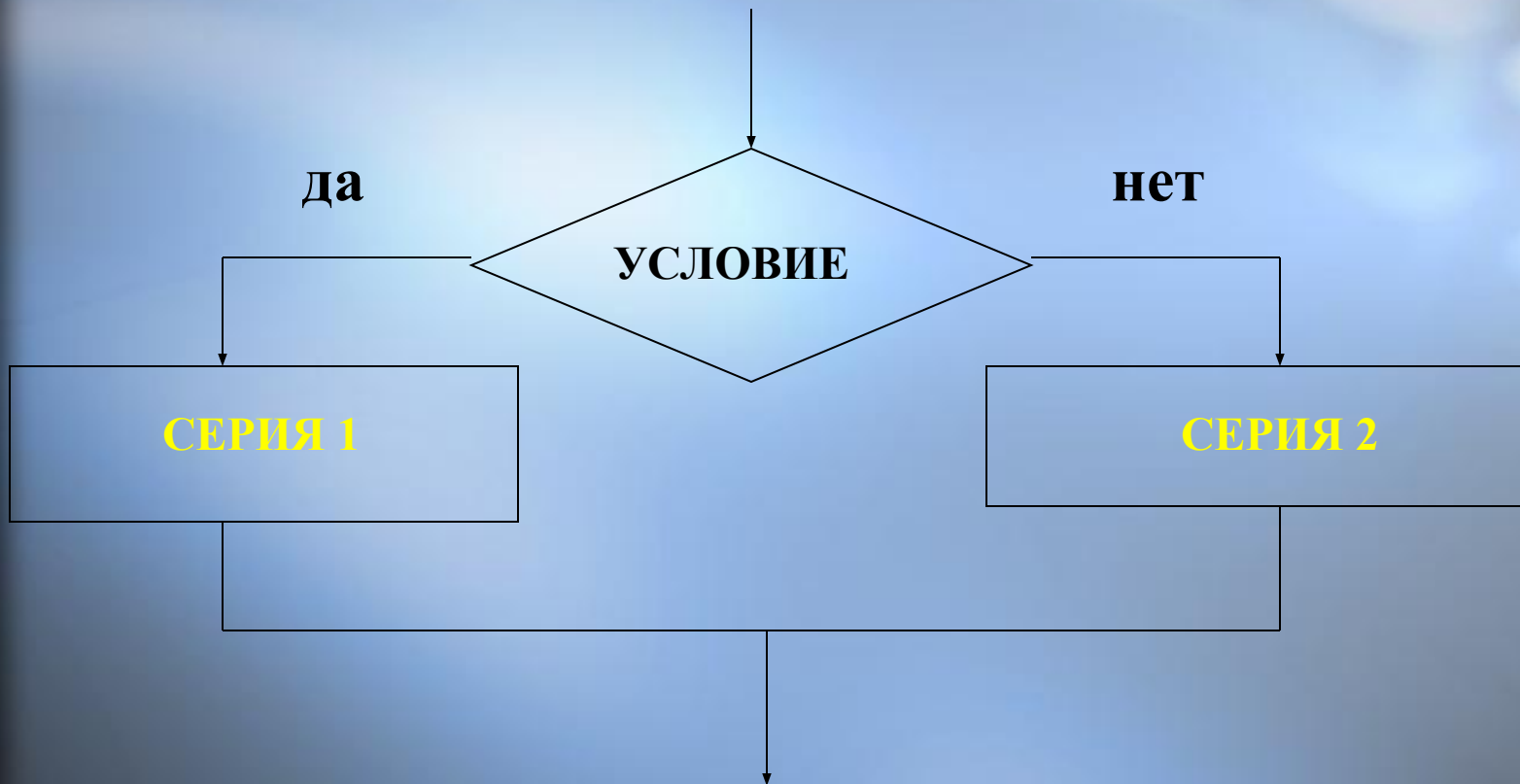


# Основные алгоритмические структуры

2. **Ветвление** - это алгоритмическая конструкция, в которой в зависимости от **условия** выполняется та или иная последовательность действий.



# Основные алгоритмические структуры



**Полная форма ветвления** *если*  
*условие истинно, выполняется Серия 1, иначе -*  
*выполняется Серия 2.*

# Основные алгоритмические структуры

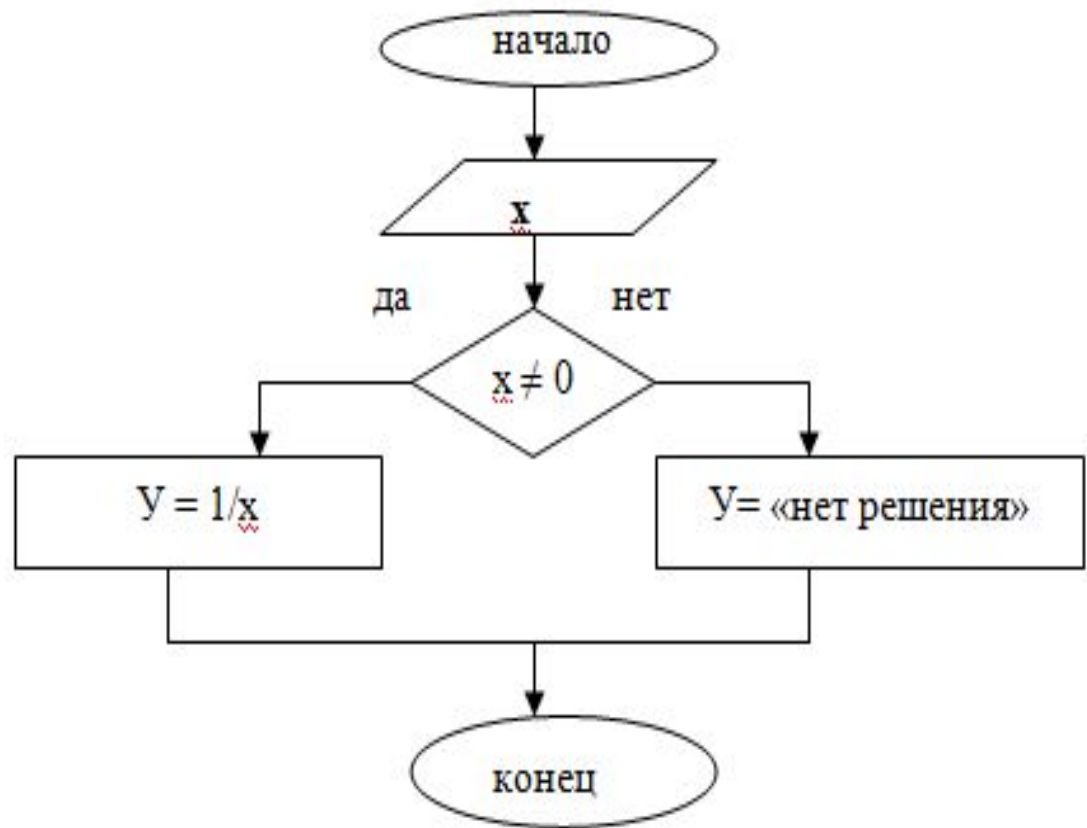


**Неполная форма ветвления** *если* условие истинно, выполняется Серия 1, иначе - ничего не делается, управление переходит к следующим командам.

# Задача № 3

Составьте схемы  
алгоритмов  
вычисления  
значения  
функции:

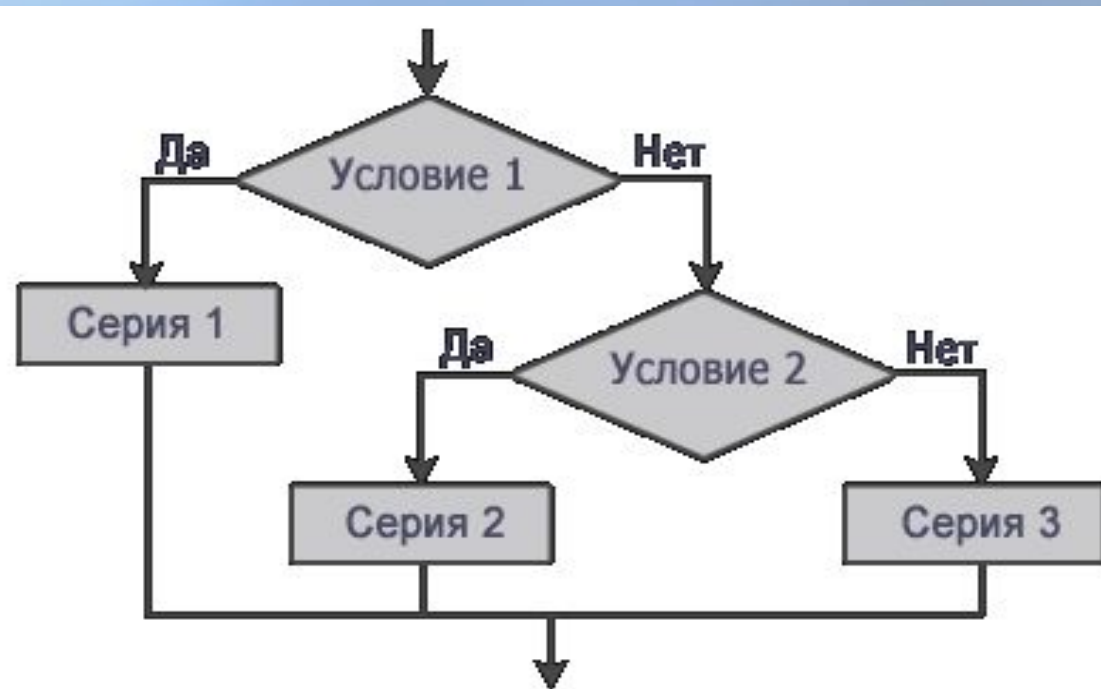
$$y = \frac{1}{x}$$



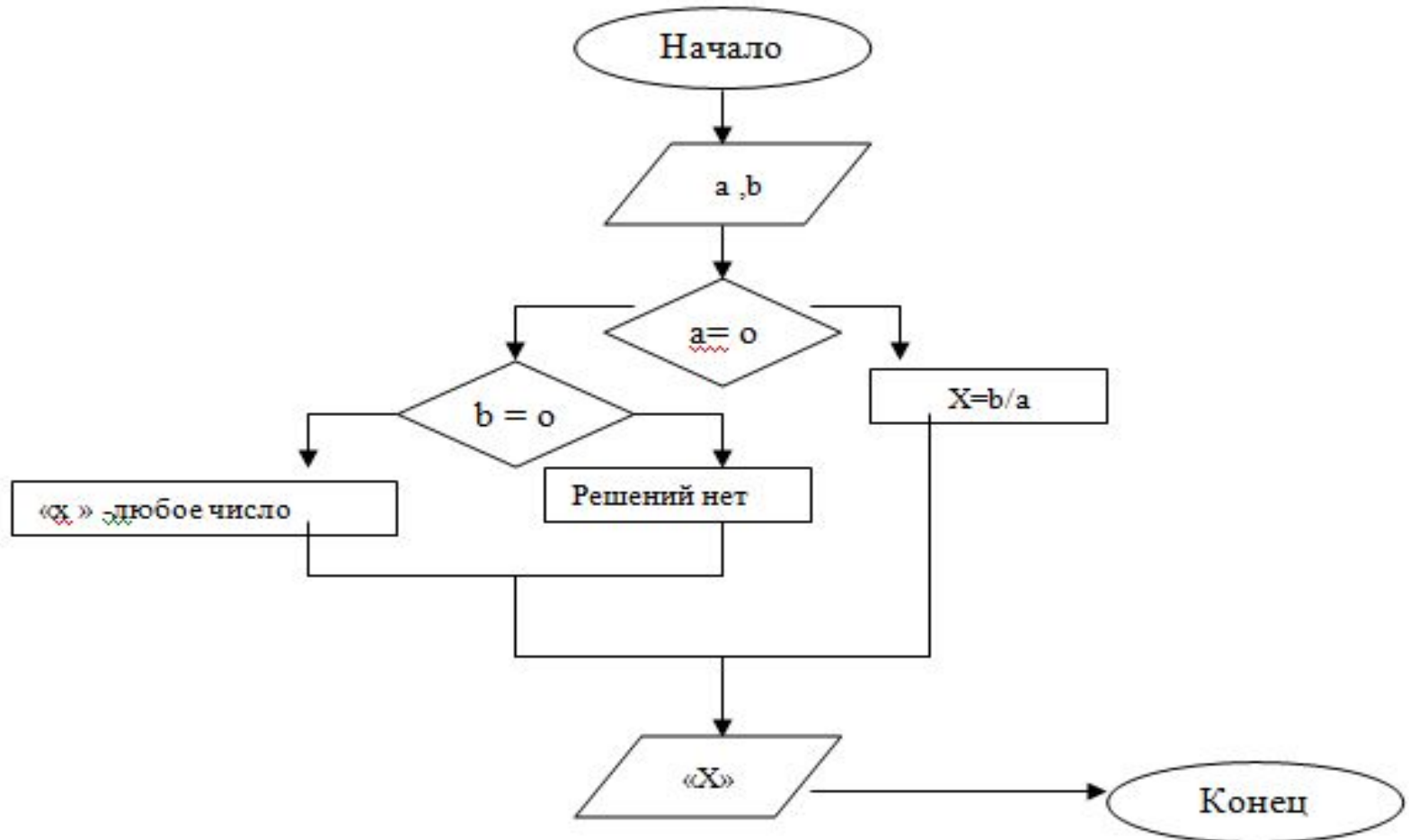


# Основные алгоритмические структуры

Алгоритмическая структура **ВЫБОР** применяется для реализации ветвления со многими вариантами серий команд.



**Задача №4. Алгоритм решения уравнения  $ax=b$  в общем случае может быть описан схемой, изображенной на рис.**



# Основные алгоритмические структуры

3. **Цикл** - это алгоритмическая конструкция, в которой в зависимости от условия повторяется определённая последовательность действий.



**Любой цикл  
характеризуется тремя  
стадиями:**

- **началом цикла;**
- **телом цикла;**
- **концом цикла.**

• Начало цикла – это условие вхождения в цикл.

Цикл может никогда не наступить, если не выполняется условие вхождения в цикл.



- Телом цикла называют последовательность действий, которые выполняются многократно в цикле.

- В теле цикла может быть всего одно действие, может быть несколько действий, может не быть ни одного действия.

- **Конец цикла – это условие выхода из цикла или прекращение цикла. Если условие выхода из цикла нарушено – цикл может никогда не кончиться, произойдет зацикливание (бесконечный цикл).**

# Виды циклических алгоритмов:

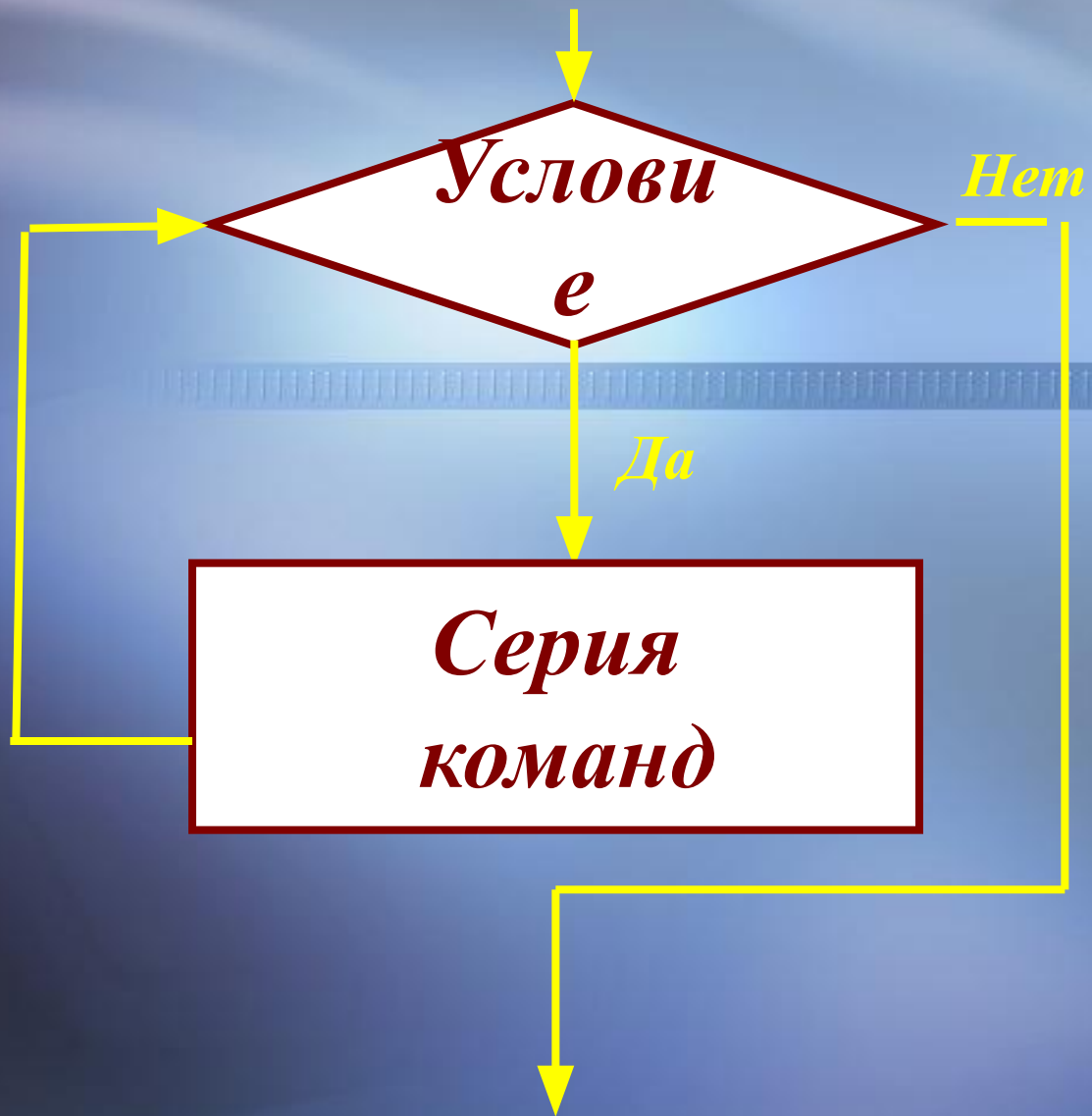
**Циклическая структура**

Цикл с предусловием  
- ЦИКЛ «ПОКА»

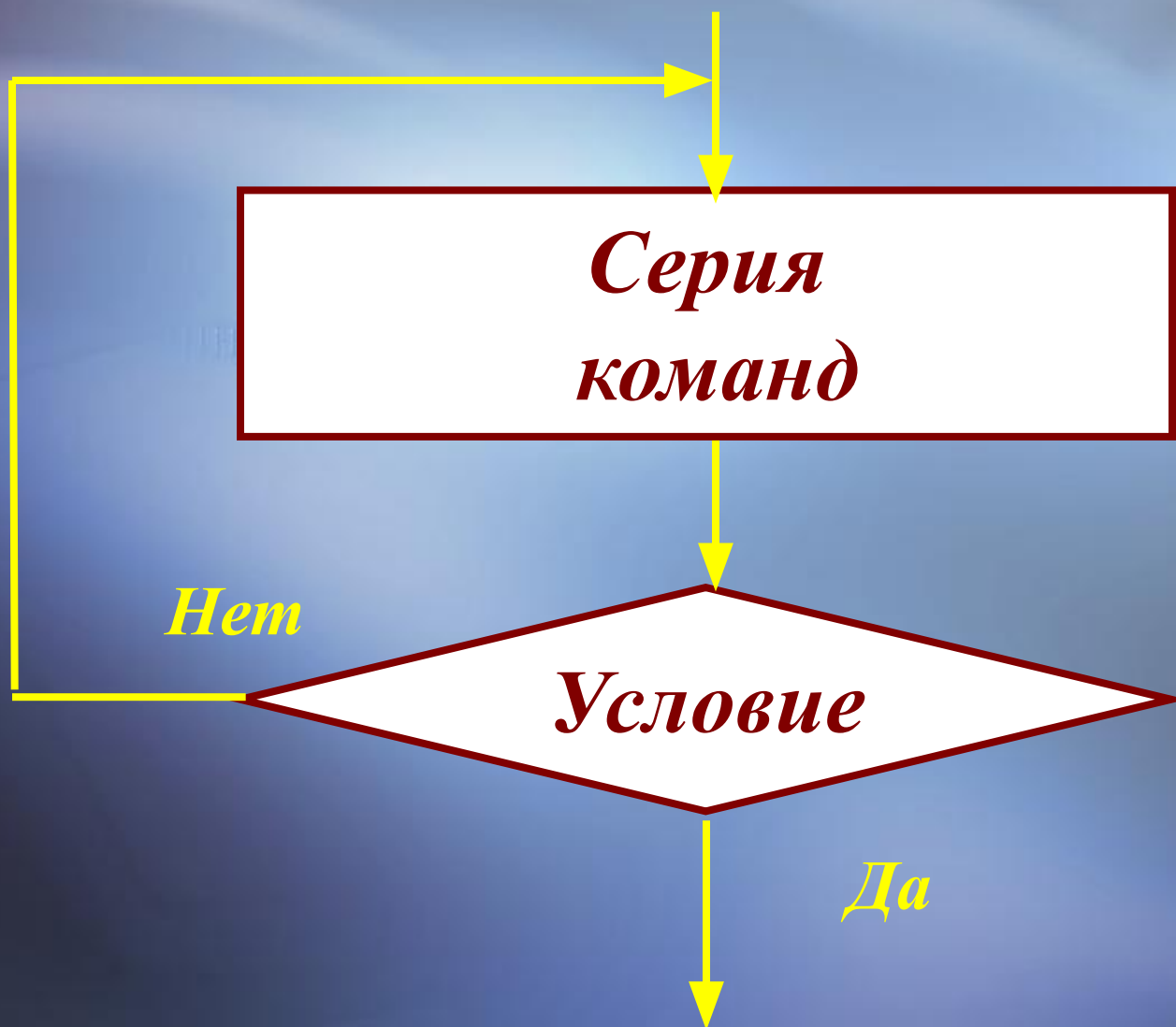
Цикл с постусловием -  
ЦИКЛ «ДО»

Цикл с параметром -  
**Арифметические циклы**

# *А) Цикл с предусловием*



## *б) Цикл с постусловием*



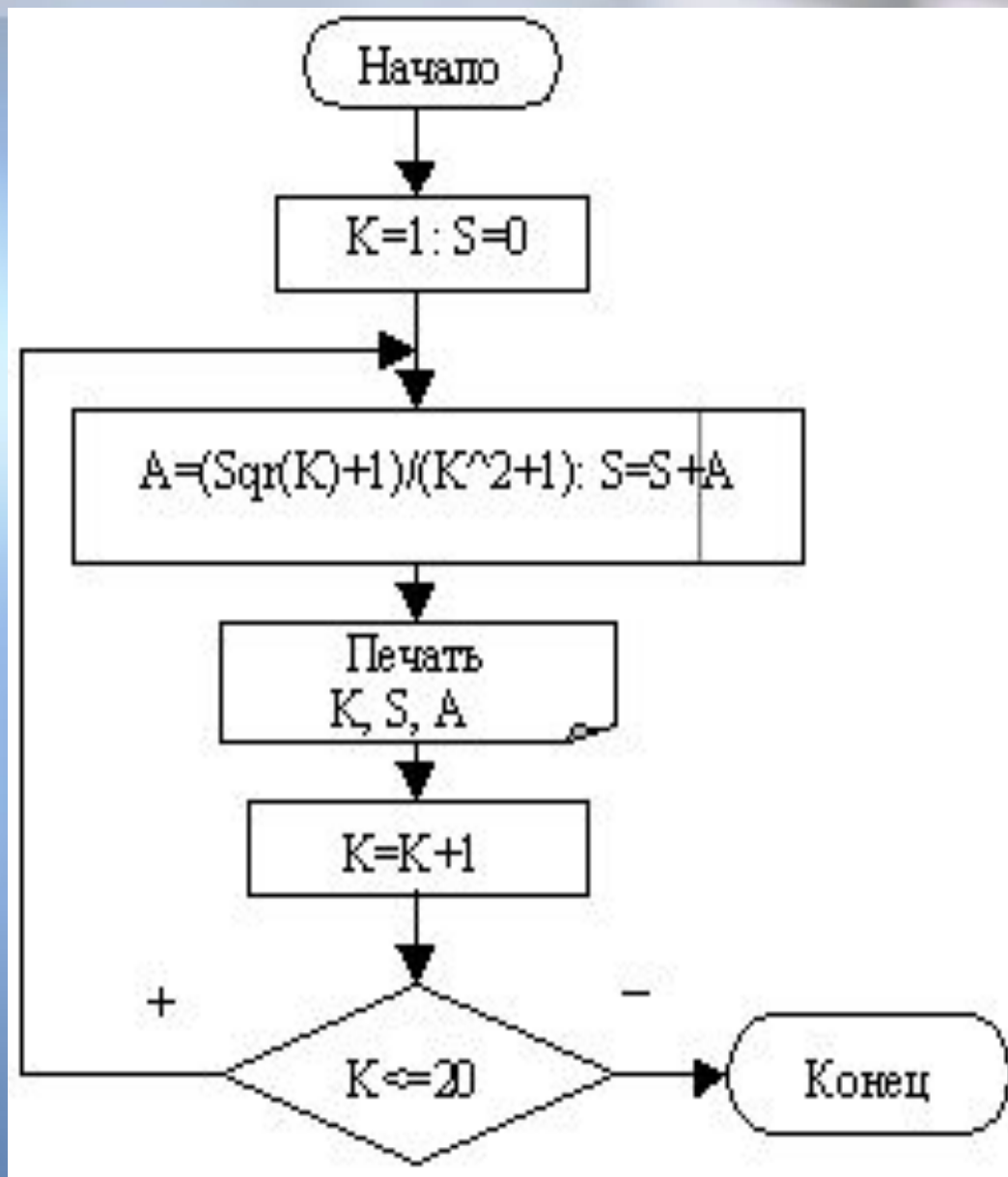


# Задача № 5

- Вычислите и их 20 первых членов последовательности **сумму**, если общий член последовательности

$$a_k = \frac{\sqrt{k} + 2}{k^2 + 1}$$

где  $(k=1, 2, 3 \dots 20)$ .



*в) Цикл с параметром  
арифметический цикл*

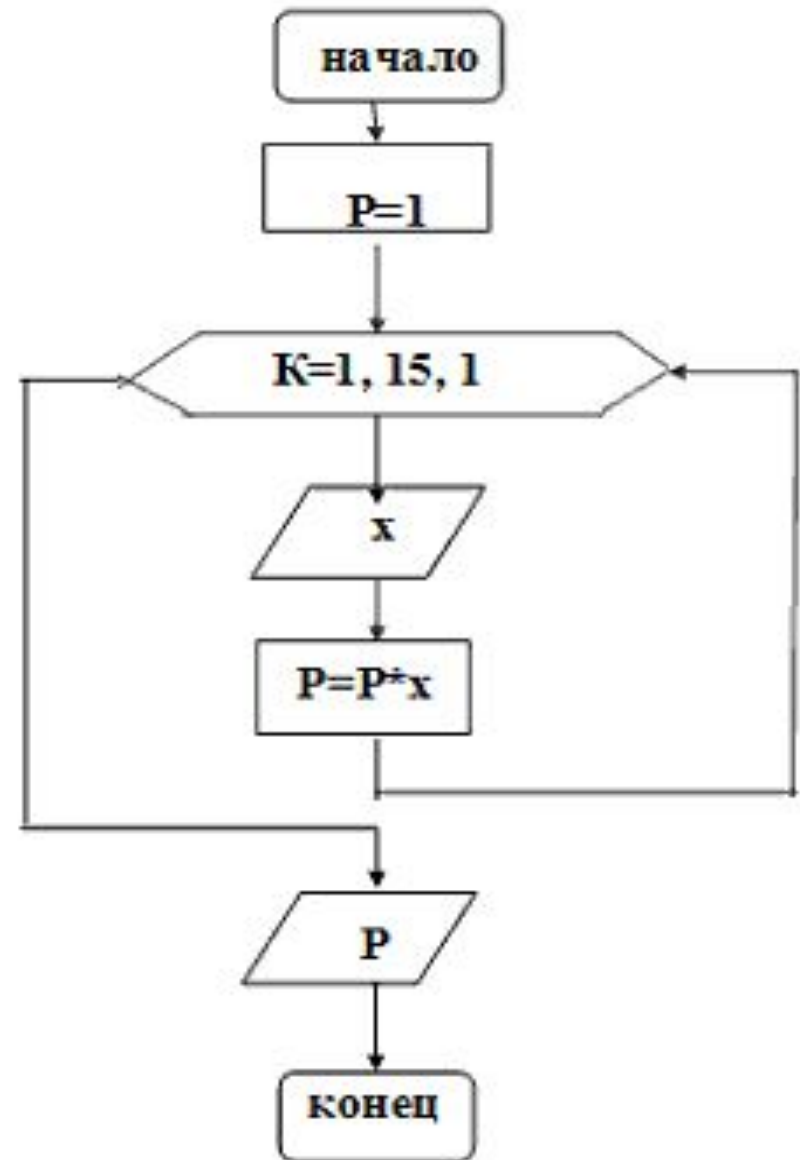


# Задача № 6

**Нахождение произведения 15 вещественных чисел.**

**Дано:**  $K$  - счётчик чисел (1;15),  $x$  - произвольные вещественные числа.

**Найти:**  $P$  - произведение 15 чисел.



**Этапы решения задач с  
использованием  
компьютера:  
формализация,  
программирование и  
тестирование.**

# Решение задачи на компьютере

**- это процесс  
автоматического  
преобразования  
информации в  
соответствии с  
поставленной целью.**

*Под процессом решения задачи на ЭВМ  
надо понимать совместную*



# ПОСТАНОВКА ЗАДАЧИ

- Это этап работы человека, хорошо представляющего предметную область задачи.

Он должен:

- *определить цель решения задачи;*
- *определить необходимый объем информации;*
- *дать точную формулировку задачи;*
- *предложить идею решения задачи;*
- *описать исходные данные и указать способы их хранения;*
- *определить форму выдачи результатов.*

# ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ.

Чтобы решить задачу, связанную с исследованием реального объекта, необходимо описать этот объект в математических терминах, связанных определенными соотношениями (формулами), то есть построить его **математическую модель.**

Этот этап выполняет человек, способный разработать математическое описание поставленной задачи. При этом он должен выполнить:

- анализ похожих решённых задач;
- анализ технических и программных возможностей;
- анализ условий существования решения.

# РАЗРАБОТКА АЛГОРИТМА.

Алгоритм решения задачи разрабатывается на основе построенной математической модели и представляет конечную последовательность предписаний (правил), которая определяет процесс преобразования исходных данных в результаты решения задачи.

Алгоритм разрабатывается одним из существующих способов, чаще всего в виде блок-схемы. Этот этап выполняет человек, умеющий программировать. Уровень его квалификации определяет эффективность разработанного алгоритма.

# СОСТАВЛЕНИЕ ПРОГРАММЫ.

Этот этап также выполняет человек, умеющий программировать, так как программа - это один из способов представления алгоритма с использованием языка программирования, наиболее удобного для решения данной задачи.

*Составление программы - это кодирование алгоритма для реализации его в компьютере.*

# ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ

- *Отладка программы - это проверка ее на наличие технических, грамматических и, возможно, алгоритмических ошибок.* Кроме того, отладка предполагает совершенствование (оптимизацию) программы. Ошибки в ходе отладки исправляют путем контрольного тестирования.
- *Тесты (контрольные примеры) составляются так, чтобы проверить все возможные варианты работы алгоритма. Проверка осуществляется путем сравнения заранее известных результатов тестов с результатами, полученными компьютером.* Этот этап - совместная работа программиста и машины.



# ПОЛУЧЕНИЕ И АНАЛИЗ РЕЗУЛЬТАТОВ.

После устранения всех ошибок, выявленных отладкой и тестированием, получают результаты решения поставленной задачи.

Получение результатов может быть многократным в зависимости от смены исходных данных, поскольку решение должно быть универсальным для задач подобного класса. На этом этапе машина работает совместно с человеком, выдающим задание.

# Домашнее задание:

□ Лекционный материал