

# Лекция

---

**Тема: «Алгоритмы. Принципы обработки алгоритмов»**

**Дисциплина : «Программирование на Python»**

**Образовательная программа:  
«6B06104 – «Вычислительная техника  
и программное обеспечение»»**

**Ст.преп. каф. ИВС Томилов А.Н.**

# Цель лекции

Целью лекции является приобретение теоретических знаний в области алгоритмизации и программирования

# План

1. Понятие алгоритма и его свойства
2. Способы описания алгоритмов
3. Основные алгоритмические конструкции
4. Базовые алгоритмы

# 1. Понятие алгоритма и его свойства

**Алгоритм** (от *algoritmi*)- предписание, однозначно задающее процесс преобразования исходной информации в виде последовательности элементарных дискретных шагов, приводящих за конечное число их применений к результату.



Мухаммед ибн Муса  
аль-Хорезми (783-850)

## Разновидности алгоритмов:

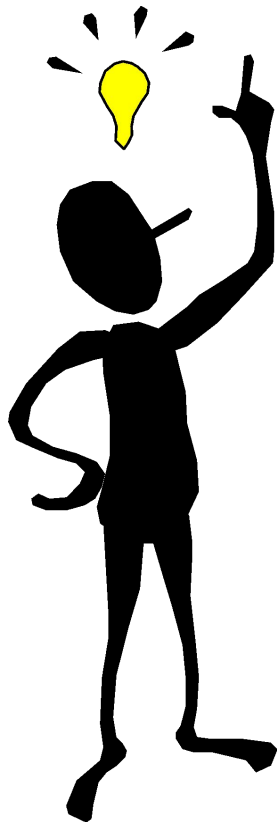
**вычислительные** – работают с простыми видами данных (числа, векторы, матрицы), но процесс вычисления может быть длинным и сложным;

**информационные** – реализуют небольшие процедуры обработки (например, поиск элементов, удовлетворяющих определенному признаку), но для больших объемов информации;

**управляющие** – непрерывно анализируют информацию, поступающую от тех или иных источников, и выдают результирующие сигналы, управляющие работой тех или иных устройств.

# Свойства алгоритма

- *Дискретность* – последовательное выполнение простых или ранее определённых (подпрограммы) шагов. Преобразование исходных данных в результат осуществляется дискретно во времени.
- *Понятность* – каждая команда алгоритма должна быть понятна тому, кто исполняет алгоритм; в противном случае, эта команда и, следовательно, весь алгоритм в целом не могут быть выполнены.
- *Определенность* - каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвольного толкования.
- *Результативность* означает возможность получения результата после выполнения конечного количества операций.
- *Корректность* - решение должно быть правильным для любых допустимых исходных данных.
- *Массовость* заключается в возможности применения алгоритма к целому классу однотипных задач, различающихся конкретными значениями исходных данных (разработка в общем виде).



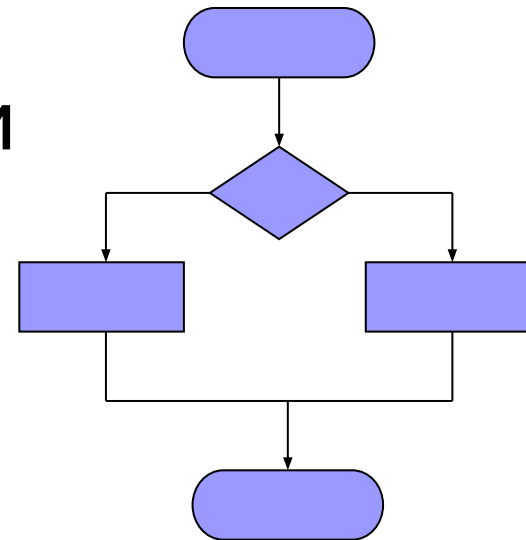
Составление  
алгоритма является  
обязательным  
этапом  
автоматизации  
любого процесса.

## 2. Способы описания алгоритмов

- словесный (на естественном языке);
- формульно-словесный;
- табличный (обычно носит вспомогательный характер);
- графический (использует элементы блок-схем).



**Блок-схема** - графическое изображение структуры алгоритма, в котором каждый этап процесса переработки данных представляется в виде геометрических фигур (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых при этом операций.

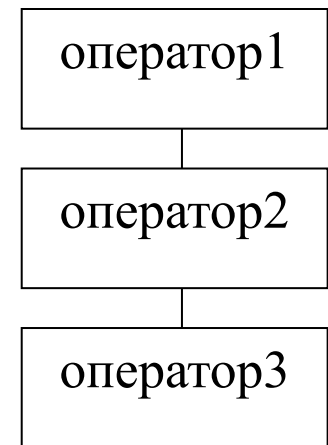


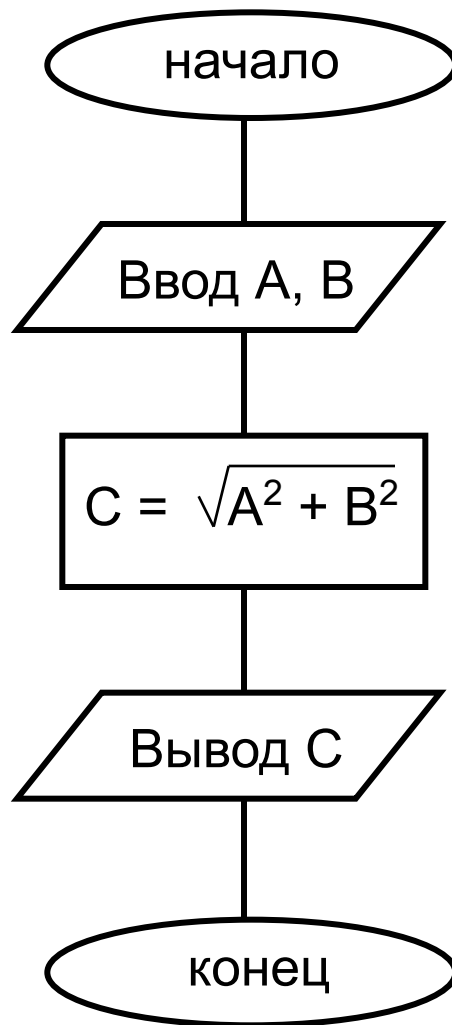
	- начало или конец алгоритма
	- ввод/вывод данных или результата на экран монитора
	- процесс – арифм.выражение или операция присваивания
	- проверка условия
	- подпрограмма
	- вывод на принтер
	- циклический процесс.

# 3. Основные алгоритмические конструкции


**Линейным** принято

называть вычислительный процесс, в котором этапы вычислений выполняются в линейной последовательности и каждый этап выполняется только один раз.





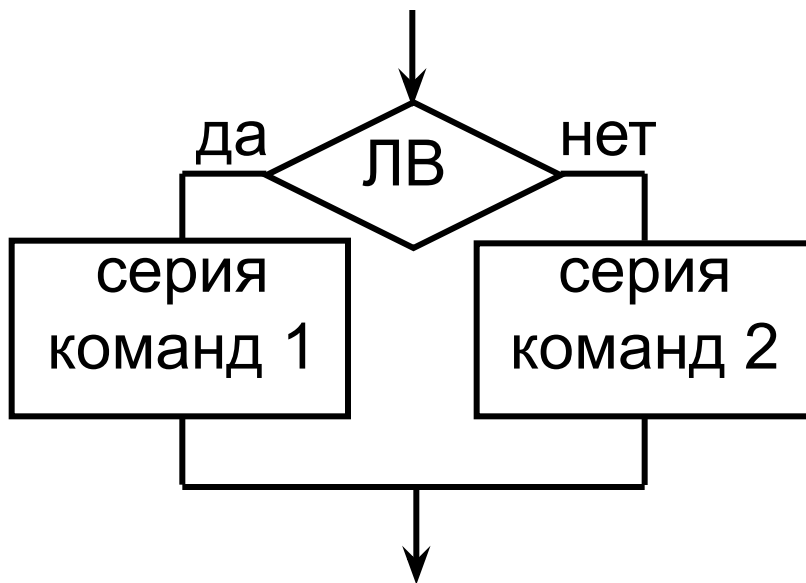
Блок-схема вычисления гипотенузы по  
теореме Пифагора



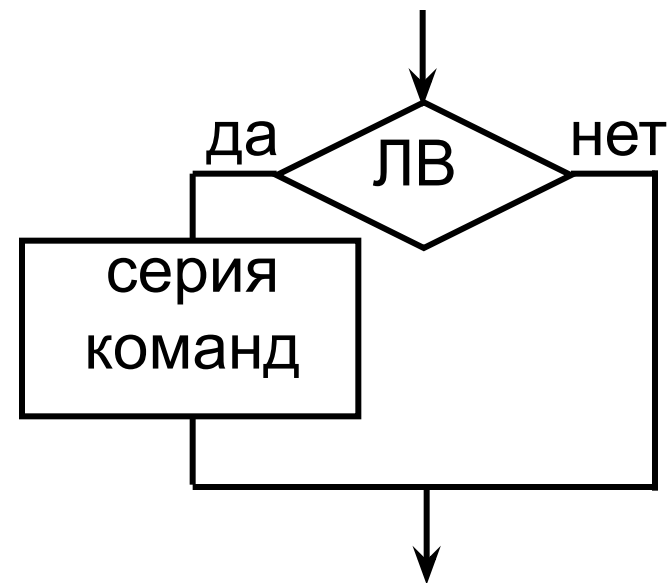
**Разветвляющийся** вычислительный процесс реализуется по одному из нескольких заранее предусмотренных направлений (ветвей) в зависимости от выполнения некоторого условия (логического выражения - ЛВ).

Ветвящийся процесс, включающий в себя две ветви, называется *простым*, более двух ветвей - *сложным*.

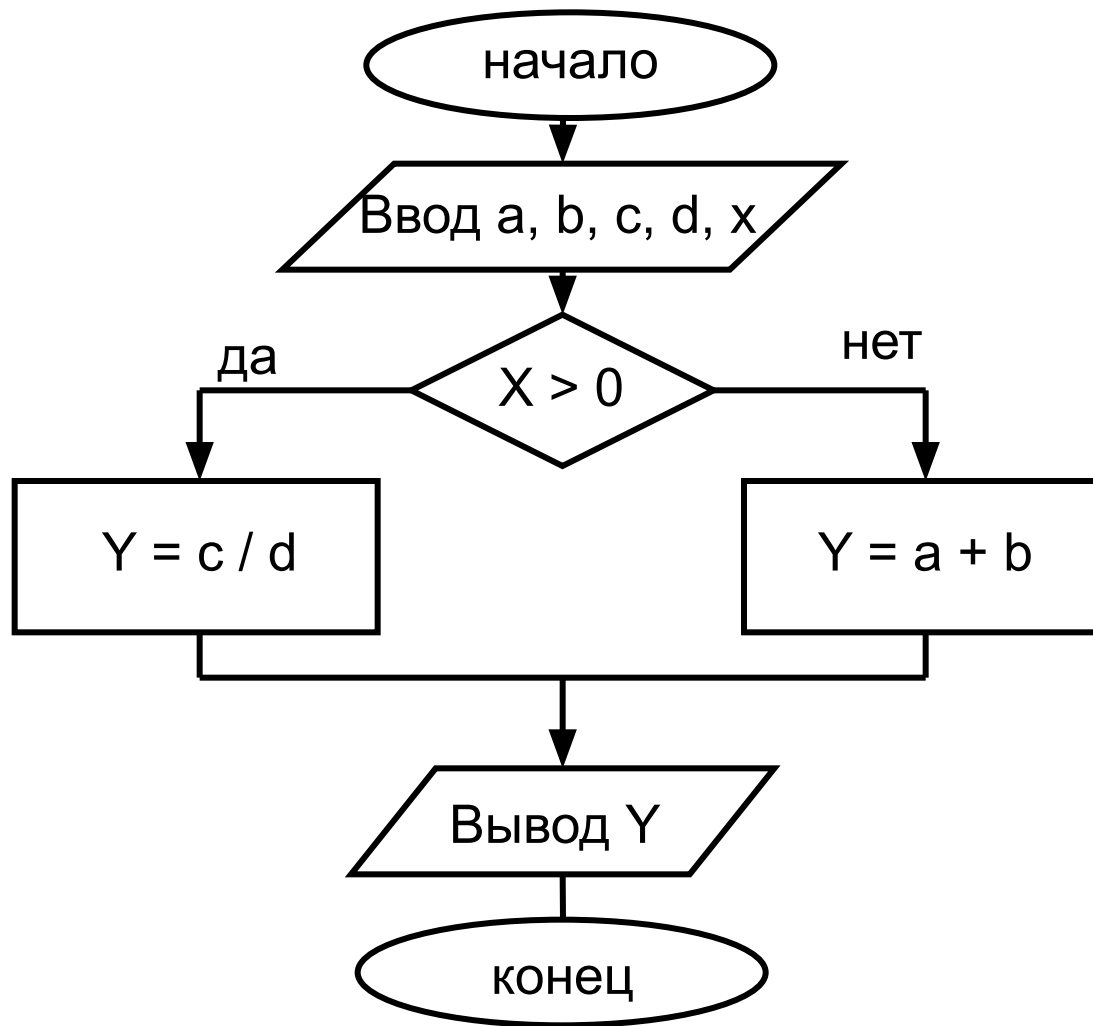
полное ветвление  
*если-то-иначе*




неполный вариант  
ветвления  
*если-то*




Алгоритм вычисления функции:  $y = \begin{cases} a + b, & \text{если } X \leq 0 \\ c/d, & \text{если } X > 0 \end{cases}$





**Циклический** вычислительный процесс (цикл) включает участки, на которых вычисления выполняются многократно по одним и тем же математическим формулам, но при разных значениях исходных данных.

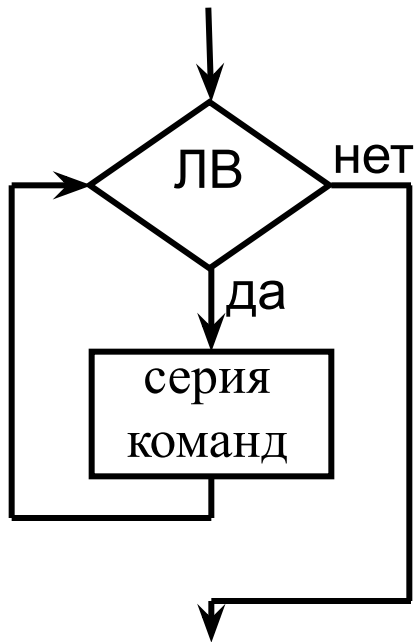




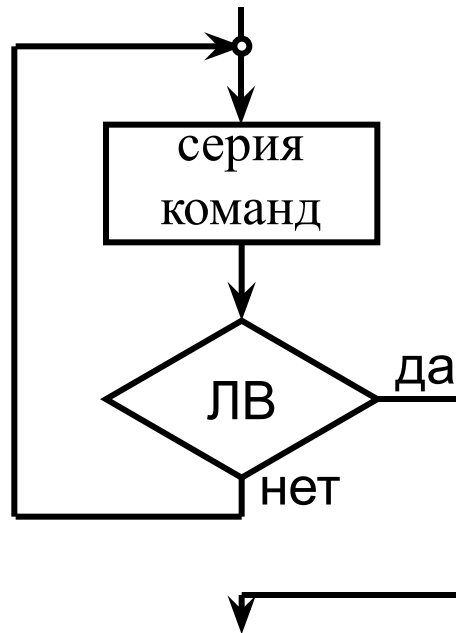
Цикл называется **детерминированным (цикл с параметром)**, если число повторений тела цикла заранее известно или определено.

Цикл называется **итерационным (с пред- и постусловием)**, если число повторений тела цикла заранее неизвестно, а зависит от значений переменных, участвующих в вычислениях.

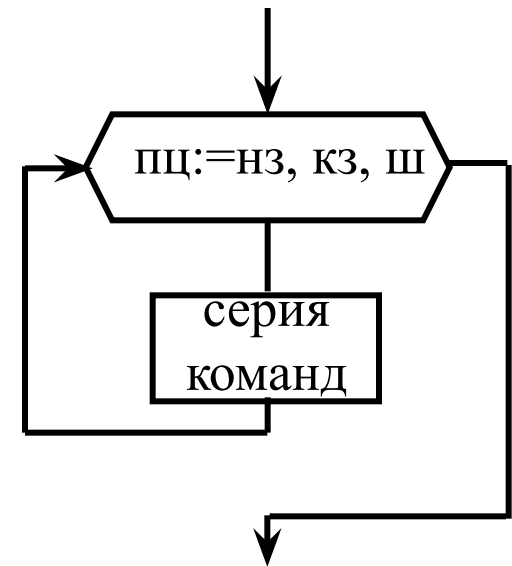
ЦИКЛ С  
предусловием



ЦИКЛ С  
постусловием



ЦИКЛ С  
параметром

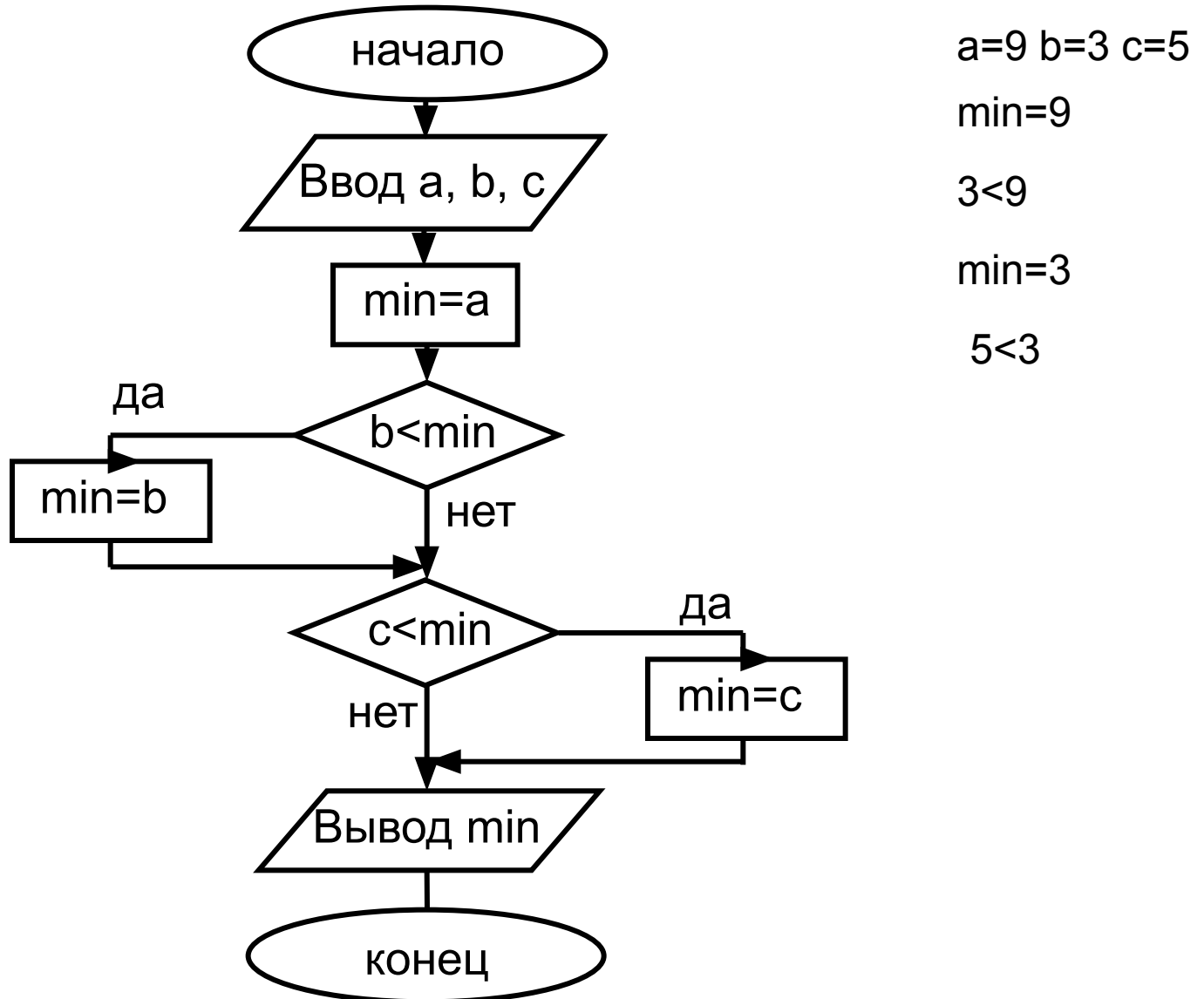


## 4. Базовые алгоритмы

**Алгоритм поиска наибольшего (наименьшего) значения:**

за  $\max$  ( $\min$ ) принимаем значение любого из данных и поочередно их сравниваем. Если окажется, что очередное значение входного данного больше (меньше)  $\max$  ( $\min$ ), то  $\max$  ( $\min$ ) присваиваем это значение. Алгоритм использует неполное ветвление.

**Пример.** Заданы три числа  $a$ ,  $b$ ,  $c$ . Найти значение наименьшего из них.



# Алгоритм Евклида –

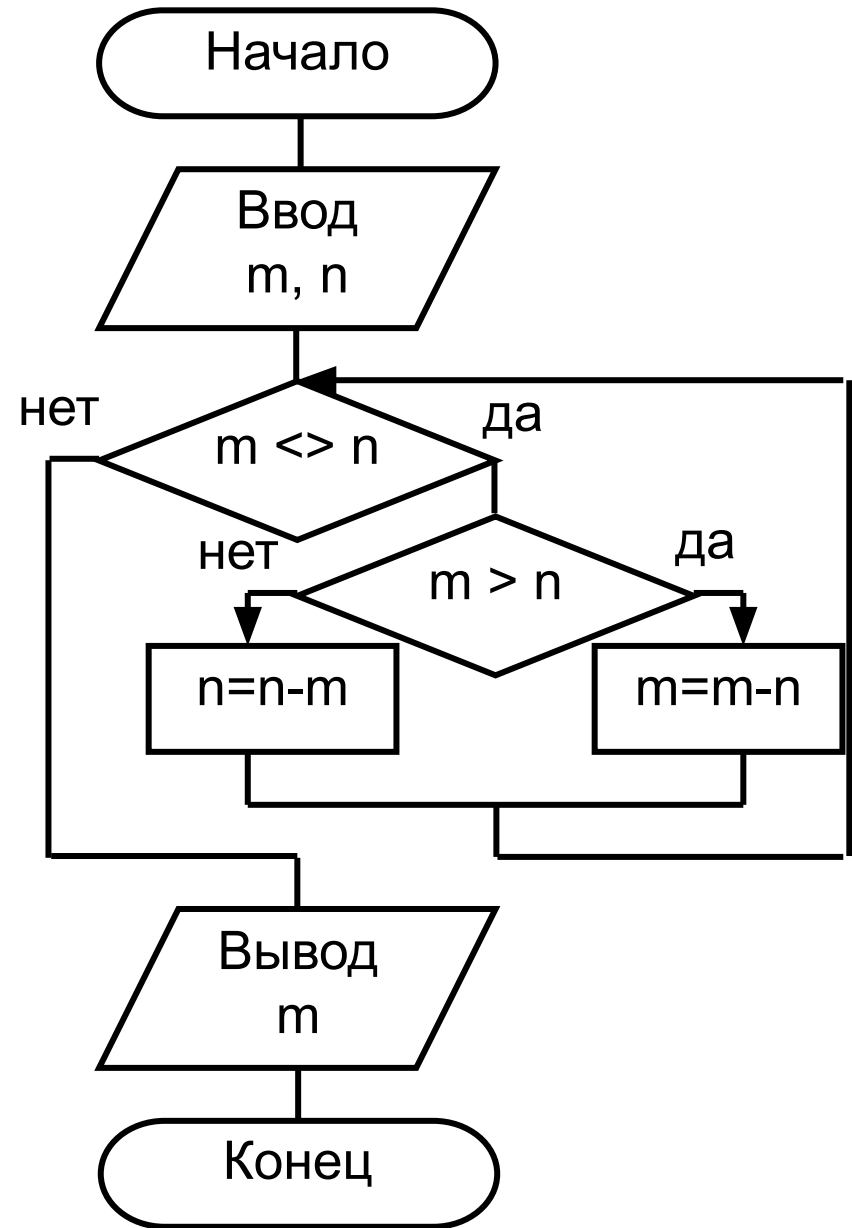
алгоритм нахождения НОД (наибольшего общего делителя) двух натуральных чисел  $m$  и  $n$  ( $m \neq n$ ).  
Используется цикл с предусловием, в который вложена операция ветвления

$m=18$   $n=12$

$m=6$

$n=6$

НОД=6



**Пример.** Вычислить факториал  $F$  натурального числа  $N$  ( $N! = 1 \cdot 2 \cdot 3 \dots \cdot N$ ).  
Используется цикл со счетчиком  $i$ .

$N=4$

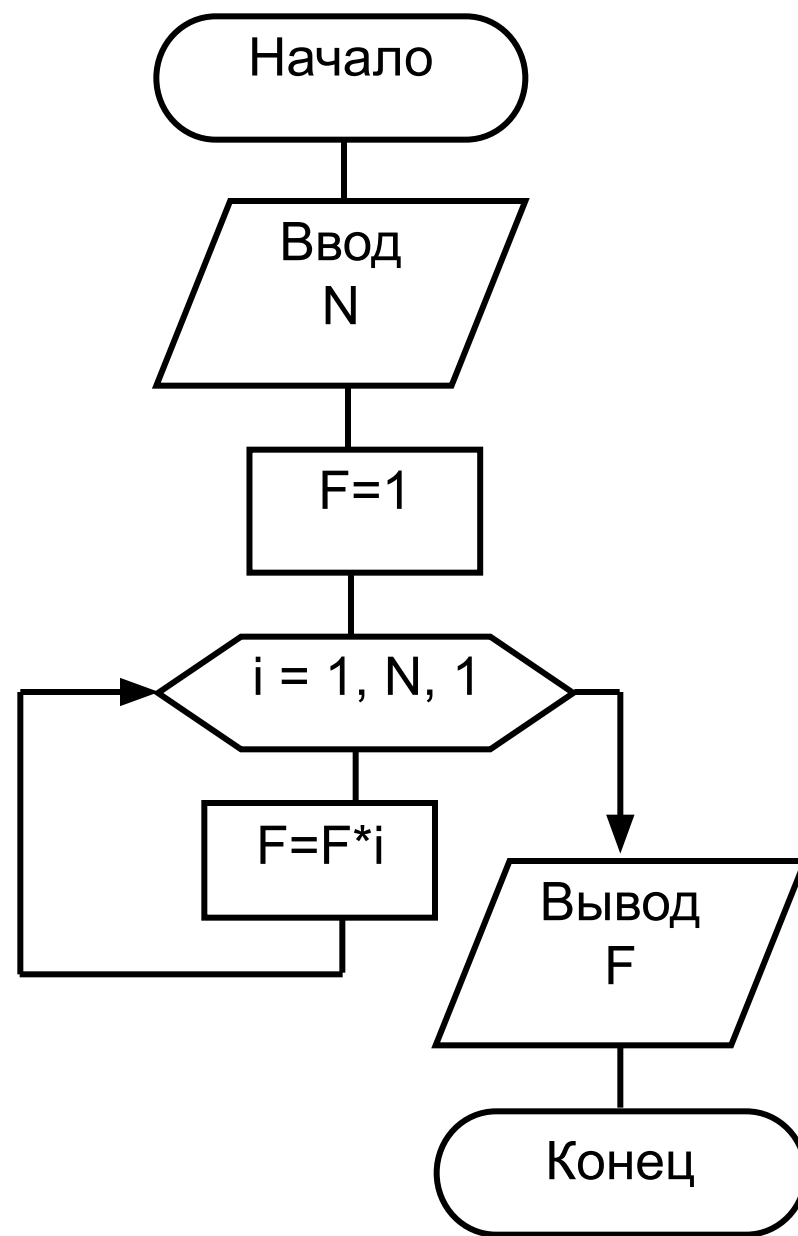
$F=1$

$i=1 \quad F=1 \cdot 1=1$

$i=2 \quad F=1 \cdot 2=2$

$i=3 \quad F=2 \cdot 3=6$

$i=4 \quad F=6 \cdot 4=24$



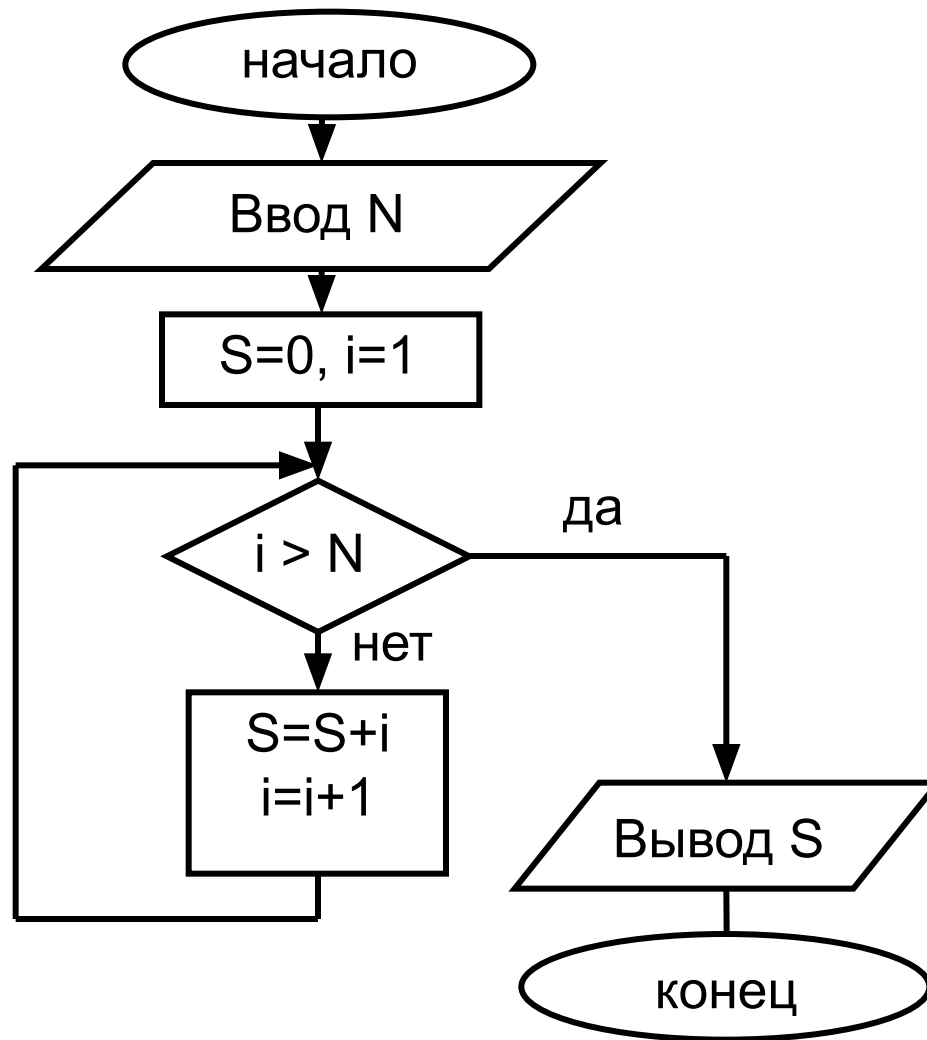


## Правило произведения:

- начальное значение произведения  $P=1$ ;
- в теле некоторой циклической конструкции выполнить команду:  
 $P = P * \langle \text{множитель} \rangle$

**Пример.** Составим алгоритм вычисления суммы  $N$  первых натуральных чисел. Используется цикл с предусловием.

$N=5$   
 $S=0$        $i=1$   
 $S=0+1=1$      $i=2$   
 $S=1+2=3$      $i=3$   
 $S=3+3=6$      $i=4$   
 $S=6+4=10$     $i=5$   
 $S=10+5=15$   $i=6$   
  
 $S=15$



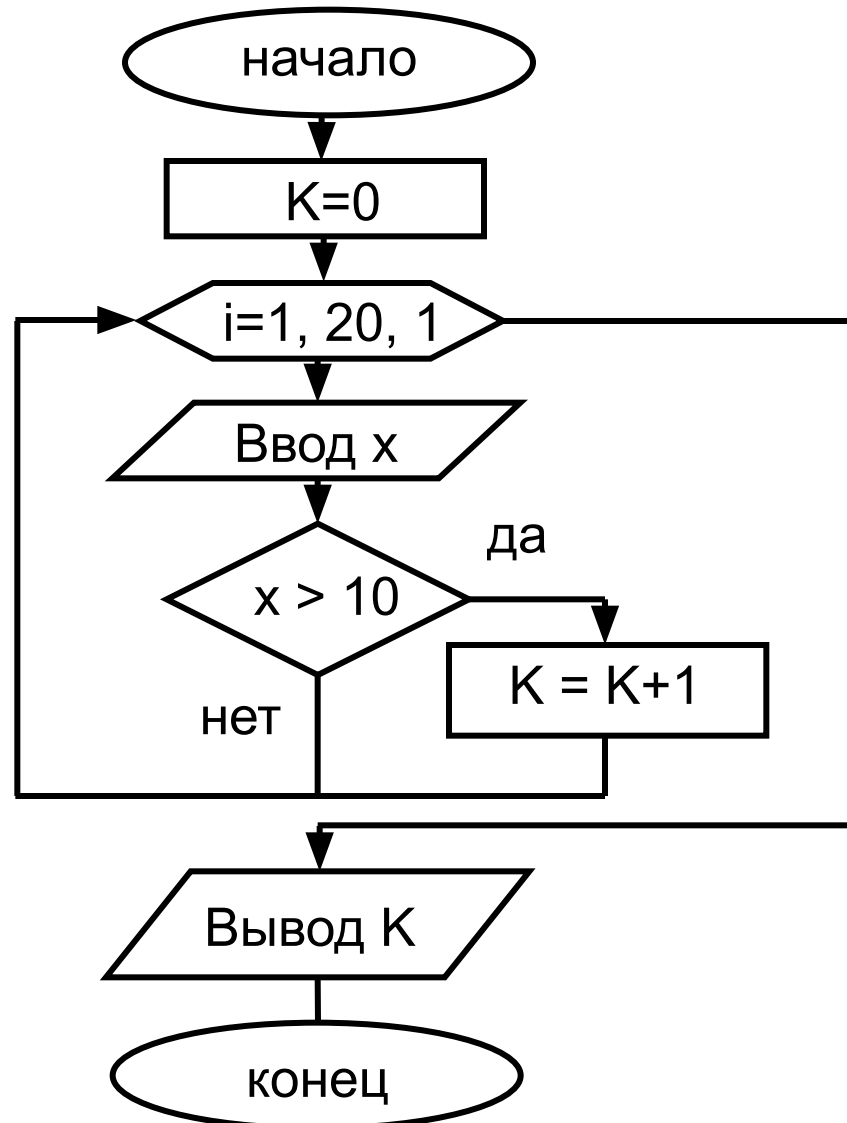




## Правило суммирования:

- начальное значение суммы  $S=0$ ;
- в теле некоторой циклической конструкции выполнить команду:  
 $S = S + \langle \text{слагаемое} \rangle$

Пример. Задано 20 чисел. Сколько среди них чисел, больших 10?





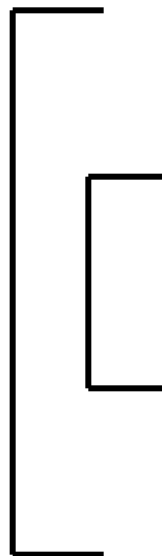
## Правило счетчика:

- начальное значение счетчика  $K=0$ ;
- в теле некоторой циклической конструкции выполнить команду:  
 $K = K + 1$



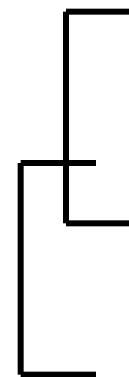
а

последовательные



б


вложенные




в

запрещенные

Рис. Расположение циклов



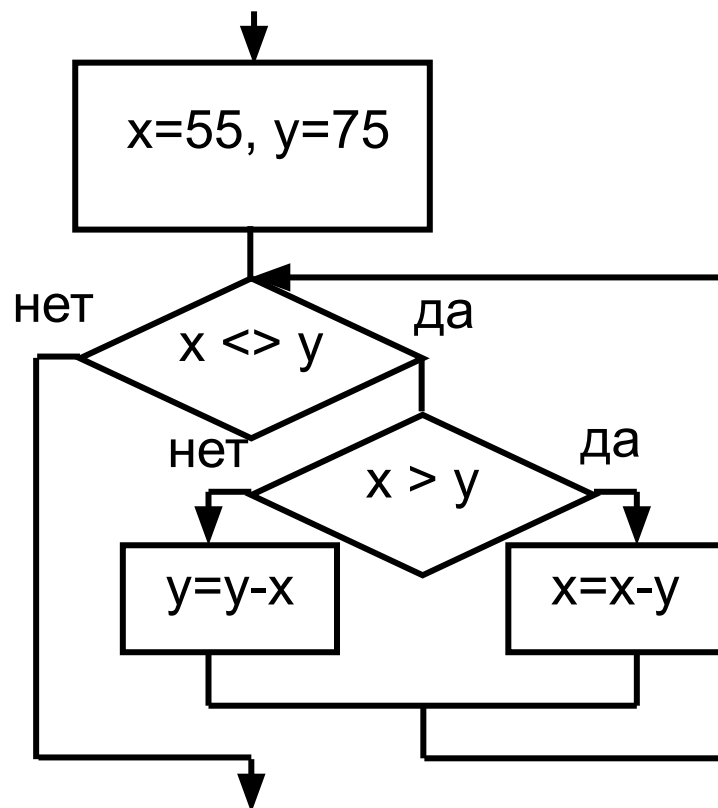
Алгоритм любой задачи может быть представлен как комбинация представленных выше элементарных алгоритмических структур, поэтому данные конструкции: линейную, ветвящуюся и циклическую, называют **базовыми**.



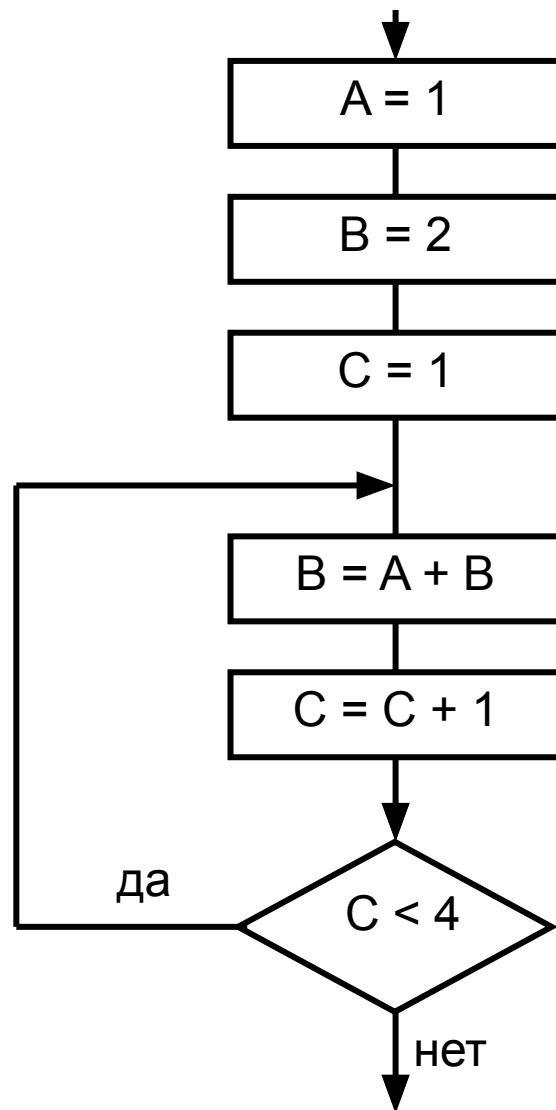
***Рекурсивным*** называется алгоритм, организованный таким образом, что в процессе выполнения команд на каком-либо шаге он прямо или косвенно *обращается сам к себе*.

# Задания для СРС

- 1) Определите значение целочисленной переменной  $x$  после выполнения следующего фрагмента алгоритма:

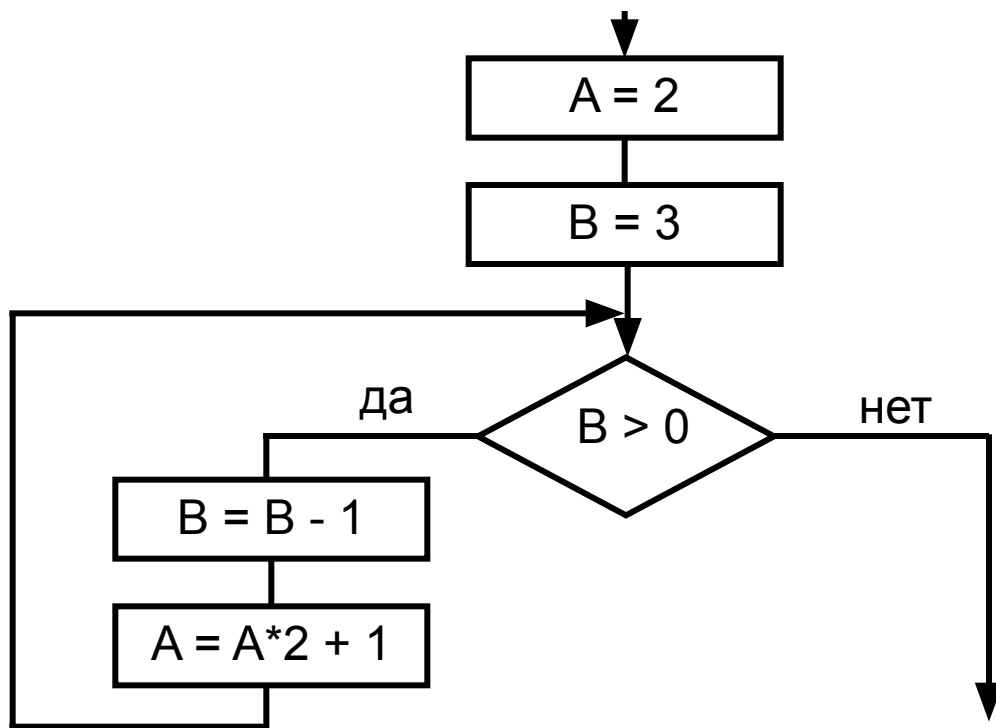


2) Определите значение переменной В :





3) Определите значение переменной A :



# Контрольные вопросы

1. Дайте определение понятия алгоритма и перечислите его свойства
2. Перечислите способы описания алгоритмов
3. Перечислите основные алгоритмические конструкции
4. Какие вы знаете базовые алгоритмы:

# Список рекомендуемых источников

1. Марк Лутц. Программирование на Python. Тома 1 и 2, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с
2. Майк МакГрат. Программирование для начинающих. Производственно-практическое издание. – М.: Эксмо, 2015. – 192.
3. Электронный учебник по дисциплине «Структуры данных», КарГТУ, 2015г
4. Информатика и программирование: Алгоритмизация и программирование, ред. Б. Г. Трусов. - М.: Изд. центр "Академия", 2012
5. Парфилова Н.И. Программирование: Основы алгоритмизации и программирования: учебник для студентов вузов / Н. И. Парфилова, А. Н. Пылькин, Б. Г. Трусов. - М. : Изд. центр "Академия", 2012.
6. Структуры и методы обработки данных: учебное пособие для студентов, магистрантов / Н. И. Томилова [и др.]; М-во образования и науки РК, Карагандинский государственный технический университет. Кафедра "Информационно-вычислительные системы". - Караганда : КарГТУ, 2015. - 155

# Список дополнительных источников

1. [www.intuit.ru](http://www.intuit.ru) Язык программирования Python.
2. Саммерфилд М. Программирование на Python 3. Подробное руководство. Пер. с англ. Киселев А. – М.: Символ-Плюс, 2009. – 608 с.
3. Доусон М. Програмируем на Python. - СПб.: Питер, 2014. - 416 с.
4. Видеолекции на Youtube (открытая библиотека видеолекций): <https://www.youtube.com/watch?v=xhoX3-NdM9k>
5. Язык программирования Python. Сузи Р.А. Учебное пособие. - М.: Интернет Университет информационных технологий, 2007. – 327 с.
6. А. А. Ключарев, В. А. Матьяш, С. В. Щекин. Структуры и алгоритмы обработки данных. СПбГУАП. СПб., 2004. 172 с.



**Благодарю за внимание!**