Алгоритмизация. Блок-схемы.

Для составления **программы**, предназначенной для решения на ЭВМ какой-либо задачи, требуется составление **алгоритма** ее решения.

Понятие алгоритма является в информатике одним из фундаментальных. Это неопределяемое понятие.

- Алгоритмами, например, являются правила сложения, умножения, решения алгебраических уравнений, умножения матриц и т.п.

- Под **алгоритмом** понимается определенная конечная последовательность действий, выполнение которых приводит к получению результатов.

Основные формы представления алгоритмов:

- словесная
- **блок-схема**
- таблица
- программа

Пример.

Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел.

Алгоритм в словесной форме может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;

3. определить большее из чисел;

4. заменить большее из чисел разностью большего и меньшего из чисел;

5. повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи.

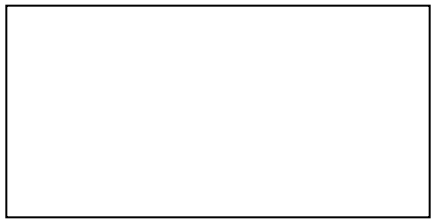
Задание на с/р . Определить с помощью этого алгоритма наибольший общий делитель чисел 125 и 75.

Виды блоков.

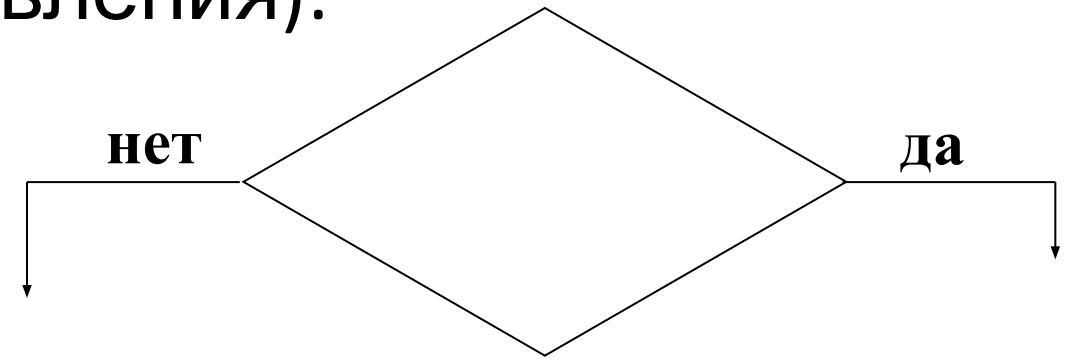
1. Блок начала /конца.

2. Блок ввода информации (список вводимых величин, имена переменных, названия массивов, функции):

3. Блок присваивания (Имя переменной = выражение):



4. Блок условного перехода (или блок ветвления):



Внутри блока записывается операция сравнения:

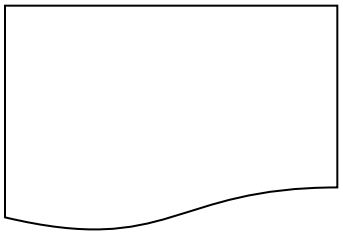
1. Математические операции

(=, >, <, <=, >=, < >).

2. Логические операции

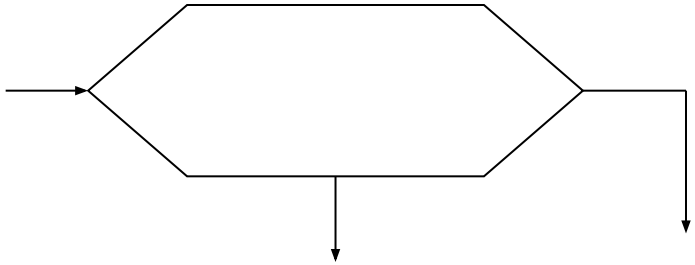
(NOT – «не», AND – «и», OR – «или»).

5. Блок вывода информации на печать:



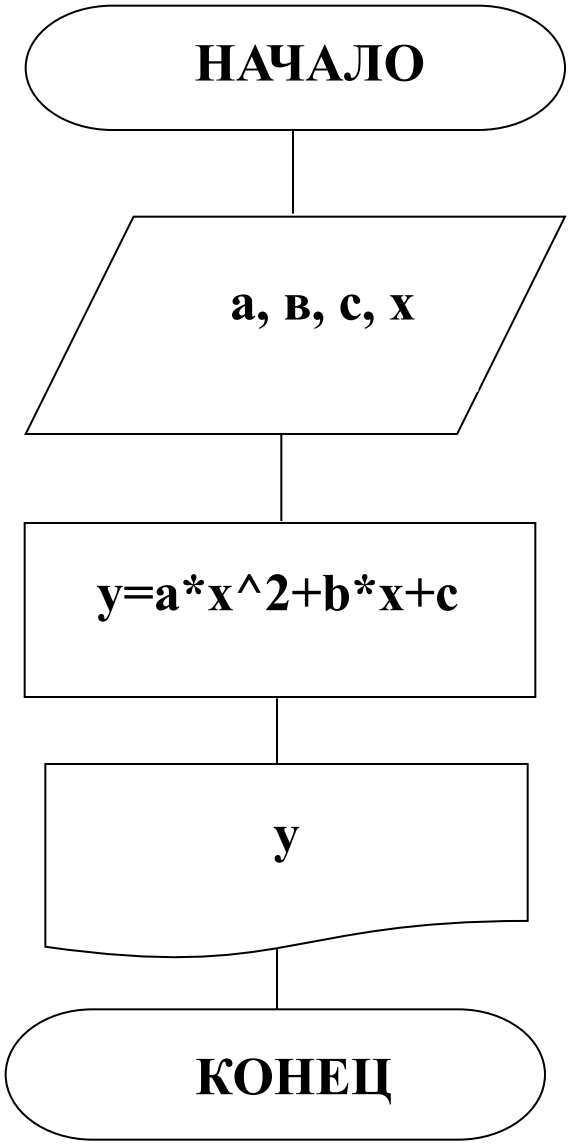
Внутри блока - список выводимых величин.

6. Блок модификации команд (начало цикла):



Используется для изображения циклов повторений.

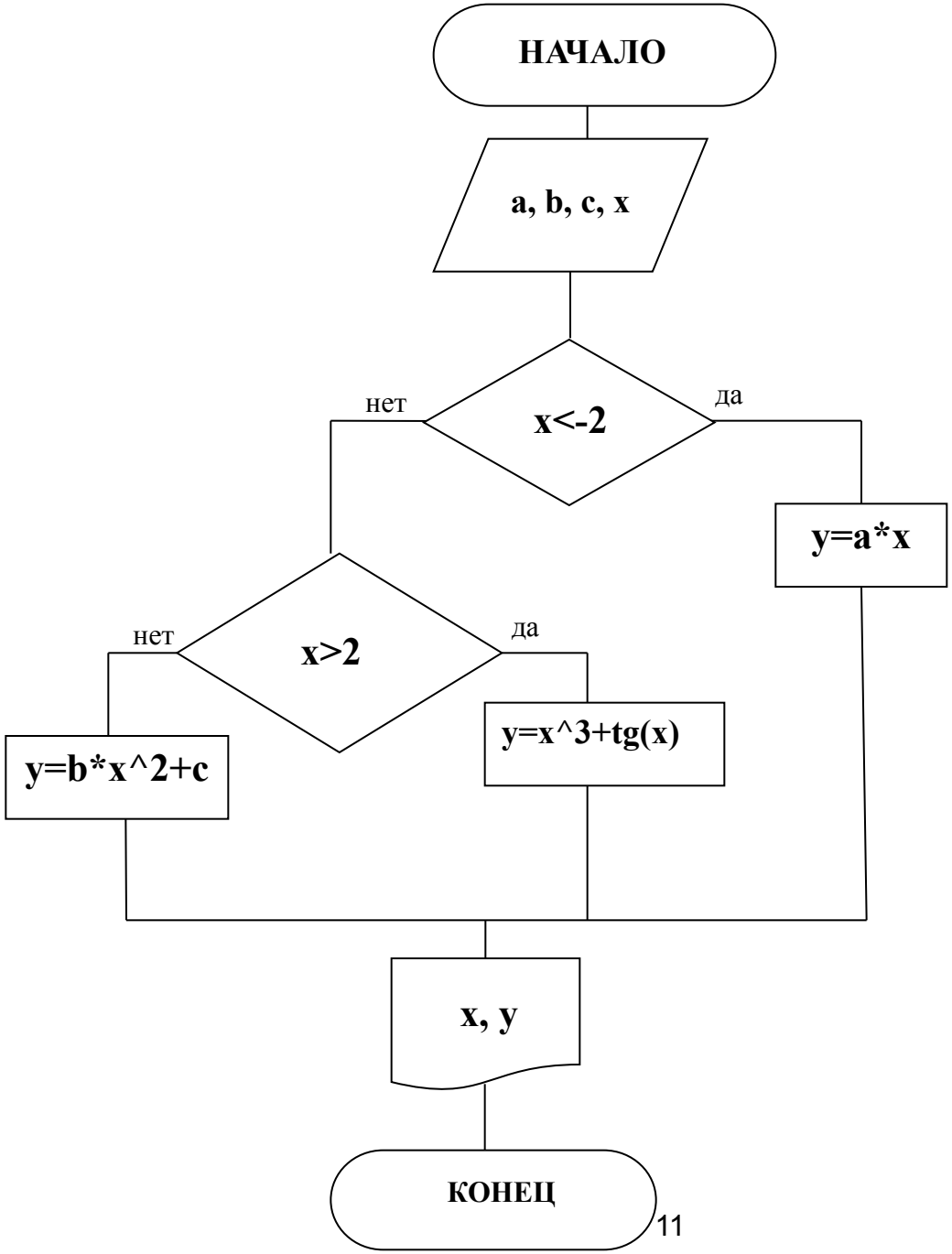
- Пример. Вычислить значение функции $y=ax^2+bx+c$.



Пример.

$$y = \begin{cases} ax & \text{при } x<-2 \\ bx^2+c & \text{при } -2 \le x \le 2 \\ x^3+\operatorname{tg}(x) & \text{при } x>2 \end{cases}$$

-


НАЧАЛО
a, b, c, x
x<-2
нет
да
y=a*x
x>2
нет
да
y=b*x^2+c
y=x^3+tg(x)
x, y
КОНЕЦ

Программирование на VBA

Visual Basic for Application (VBA–Visual Basic для приложений) – объектно-ориентированный язык программирования.

- Запуск VBA осуществляется из приложения Excel командой **Разработчик - Макросы**.

Появляется окно, в котором задается имя Макроса, далее нажимается кнопка **Создать**. После чего появляется окно среды разработки VBA.

- Запуск программы осуществляется клавишей **F5**.

Данные и их описание

Byte Положительное число без знака от 0 до 255

Integer Короткое целое от -32768 до 32767

Long Длинное целое от -2147483648 до 2147483648

Single Числовое значение с плавающей точкой (вещественные числа).

String Для хранения строковых переменных

Арифметические операции

^ Возведение в степень

***** Умножение

/ Деление

**** Целочисленное деление

Mod Определение остатка от деления

+ Сложение

- Вычитание

& Объединение (конкатенация) строк

- **Логические (булевые) операции:**

AND -логическое «И»- конъюнкция (a AND b)

OR -логическое «ИЛИ» (дизъюнкция) (a OR b)

NOT -отрицание (NOT (a))

Функции

MsgBox – функция вывода

InputBox - функция ввода

Математические функции в VBA:

Abs (число) – модуль числа

Cos(x)- cos x, Sin(x) – sin x

Tan(x) – tg x, Log(x) – ln x

Atn(x) – arctg x Sqr(x) - $\sqrt{x}$

Exp(x) – e^x,

Rnd(x) – генератор случайных чисел

Int(x) – целая часть числа

A mod B – целый остаток от деления числа A на число B.

Формулы для перевода некоторых функций:

$$\arccos x = arctg\frac{\sqrt{1-x^2}}{x}$$

$$\arcsin x = arctg\frac{x}{\sqrt{1-x^2}}$$

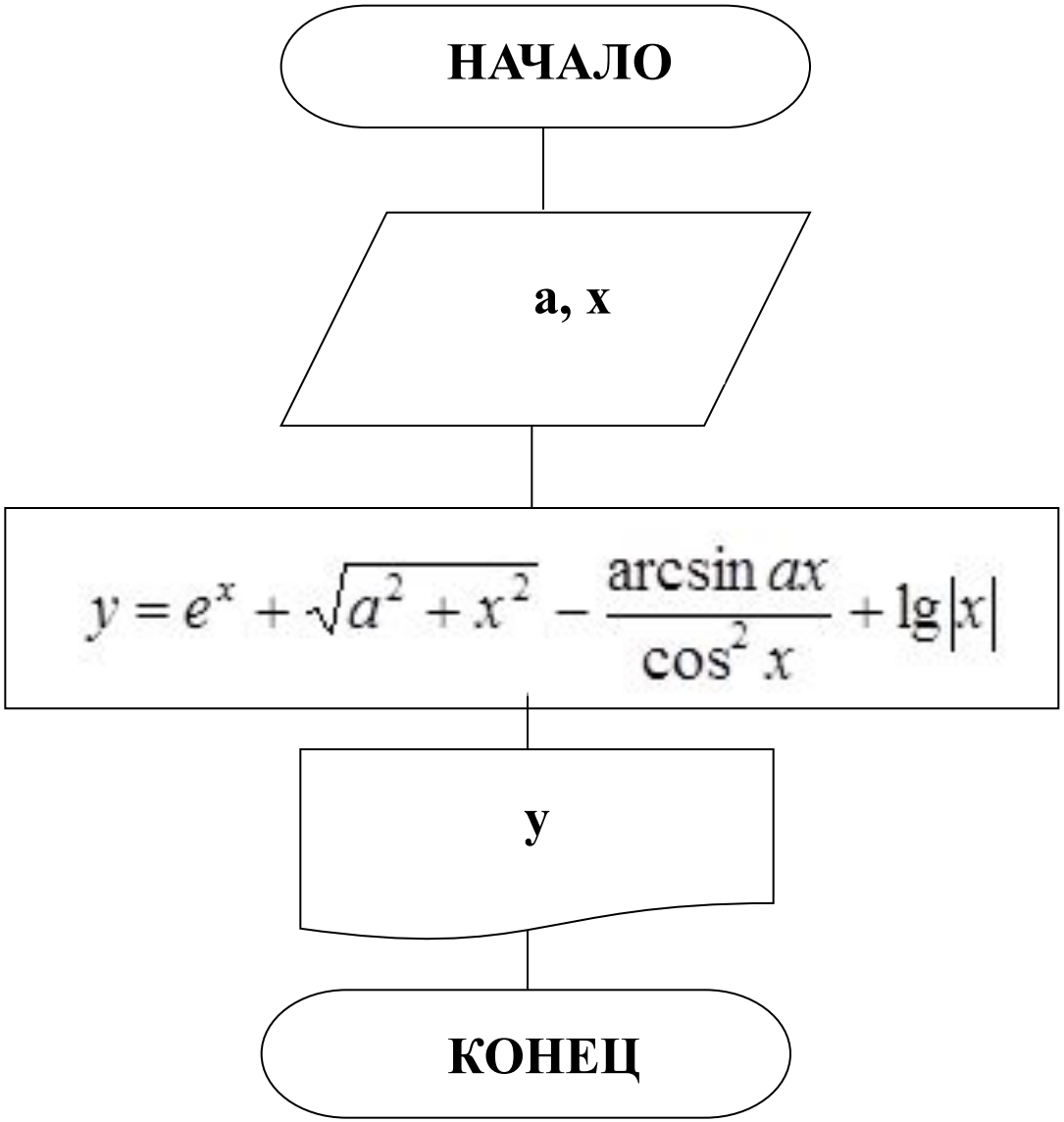
$$\log_a b = \frac{\log_c b}{\log_c a}$$

Пример: Вычислить значение выражения

$$y = e^x + \sqrt{a^2 + x^2} - \frac{\arcsin ax}{\cos^2 x} + \lg|x|$$

при x=3, a=2

- Блок-схема



```
Sub Primer1()
Dim y As Single, x,a As Integer
a=2
x=3
y=Exp(x)+Sqr(a^2+x^2)-
  Atn(a*x/Sqr(1-(a*x)^2))/cos(x)^2+Log(Abs
  (x))/log(10)
MsgBox “y=” & y
End Sub
```

Оператор условия

- Он используется для того, чтобы выполнить какой-либо оператор, если некоторое условие будет истинным.
- Синтаксическая конструкция линейного оператора имеет две формы – ***безальтернативную*** *и* ***альтернативную****.*

Безальтернативная форма условного оператора:

If условие **Then** выражение

Альтернативная форма оператора позволяет выбирать одну из двух различных ветвей операторов в зависимости от определенного условия:

If условие **Then** оператор1 **Else** оператор2

Пример. Даны три произвольных числа x, y, z. Найти f = min(x, y, z)

```
Sub primer()
Dim x, z, y, f, min As Single
x = Val(InputBox(«введите число x», x))
y = Val(InputBox(«введите число y», y))
z = Val(InputBox(«введите число z», z))
If x < y Then min = x Else min = y
If z < min Then min = z
MsgBox «f=» & min
End Sub
```

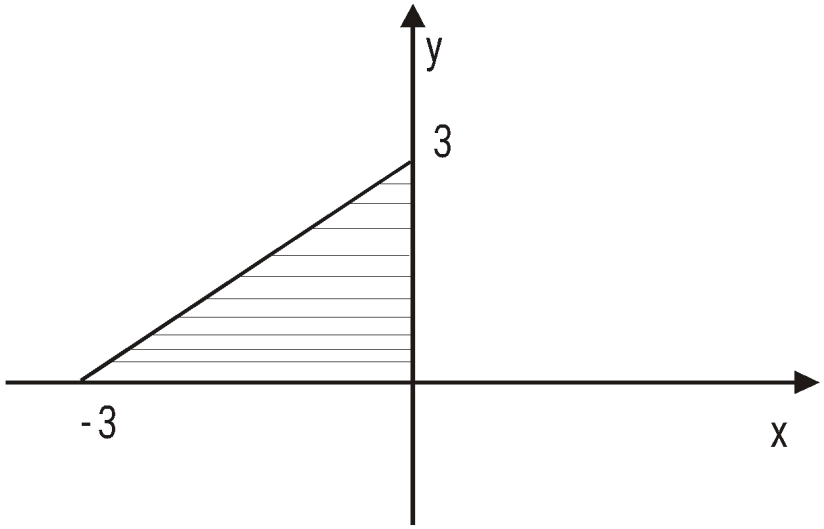
Пример. Вычислить

при k=1 a=2,3 b=4,5

$$y = \begin{cases} x^2 + bx, & \text{при } x < -10 \\ ax^3, & \text{при } -10 \le x \le 10 \\ k \sin x, & \text{при } x > 10 \end{cases}$$

```
Sub Prim3()
Dim x, y, a, b, k As Single
k=1 : a=2.3 : b=4.5
If x<-10 Then y= x^2 +b*x
If -10<=x  and x<= 10  Then y= a*x^3
If x>10 Then y= sin(x)*k
MsgBox “x=” & x & “y=” & y
End Sub
```

Пример : Попадает ли точка с координатами (x, y) в заштрихованную часть плоскости?




```
Sub zd4()
Dim x, y As Single
x = Val(InputBox("x"))
y = Val(InputBox("y"))
If y < (x + 3) And y > 0 And x < 0 Then
MsgBox («попадает") Else MsgBox ("не
   попадает")
End Sub
```

Циклы со счетчиком For …..Next.

Так они называются потому, что начало и конец этих циклов определяются операторами

For и Next.

Синтаксис:

For счетчик = Начальное значение To Конечное значение [Step Шаг]

Операторы

[Exit For]

Next [счетчик цикла]

Пример1: $x^2 + bx$ при $x<-10$

$$y = \begin{cases} ax^3 & \text{при } -10 \le x \le 10 \\ k*\sin(x) & \text{при } x > 10 \end{cases}$$

где $-20 \le x \le 20$ $\Delta x = 2$ (шаг)

$k=1$ $a=2,3$ $b=4,5$

```
Sub Prim1()
Dim x, y, a, b, k As Single
k=1 : a=2.3 : b=4.5
For x=-20 To 20 Step 2
If x<-10 Then y= x^2 +b*x
If -10<=x  and x<= 10  Then y= a*x^3
If x>10 Then y= sin(x)*k
MsgBox “x=” & x & “y=” & y
Next x
End Sub
```

Пример 2. Найти max функции Y (из примера 1)

```
Sub prim2()
Dim x, y, a, b, k, max, p As Single
max = -10^10
For x = -20 To 20 Step 2
```

Операторы для вычисления ***y*** (см. пример 1)

Продолжение (пример 2)

```
If y > max Then
max = y
p = x
End If
Next x
MsgBox "max y=" & max & "при x=" & p
End Sub
```