

# **ОСНОВНЫЕ ТИПЫ АЛГОРИТМИЧЕСКИХ СТРУКТУР**

Презентация создана  
учителем математики и информатики  
Ковалевой Анной Леонидовной  
ЦО №1679 г.Москва  
2012-2013

# ЛИНЕЙНЫЙ АЛГОРИТМ -

- это алгоритм, в котором команды выполняются последовательно одна за другой

БЛОК-СХЕМА  
ЛИНЕЙНОГО АЛГОРИТМА

ЛИНЕЙНЫЙ АЛГОРИТМ  
НА ~~PASCAL~~ BASIC



```
Program <ИМЯ ПРОГРАММЫ>;
```

```
Var <СПИСОК ПЕРЕМЕННЫХ> : <ТИП  
ПЕРЕМЕННЫХ>;
```

```
Begin
```

```
<КОМАНДА ВВОДА> (НАПРИМЕР, ReadLn)
```

```
<КОМАНДА ПРИСВАИВАНИЯ 1>
```

```
<КОМАНДА ПРИСВАИВАНИЯ 2>
```

```
.....
```

```
<КОМАНДА ПРИСВАИВАНИЯ n>
```

```
<КОМАНДА ВЫВОДА> (НАПРИМЕР, WriteLn)
```

```
End .
```

```
End Sub
```

# Пример программы линейного алгоритма

Задание: Составить программу вычисления площади произвольного треугольника по трем известным сторонам  $a$ ,  $b$ ,  $c$  (по формуле Геррона)

```
Private Sub Command1_Click()
```

```
a = Val(Текст1.Text)
```

```
b = Val(Текст2.Text)
```

```
c = Val(Текст3.Text)
```

```
p = (a + b + c) / 2
```

```
Метка5.Caption = Int(Sqr(p * (p - a) * (p - b) * (p - c)))
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Текст1.Text = "": Текст2.Text = ""
```

```
Текст3.Text = "": Метка5.Caption = ""
```

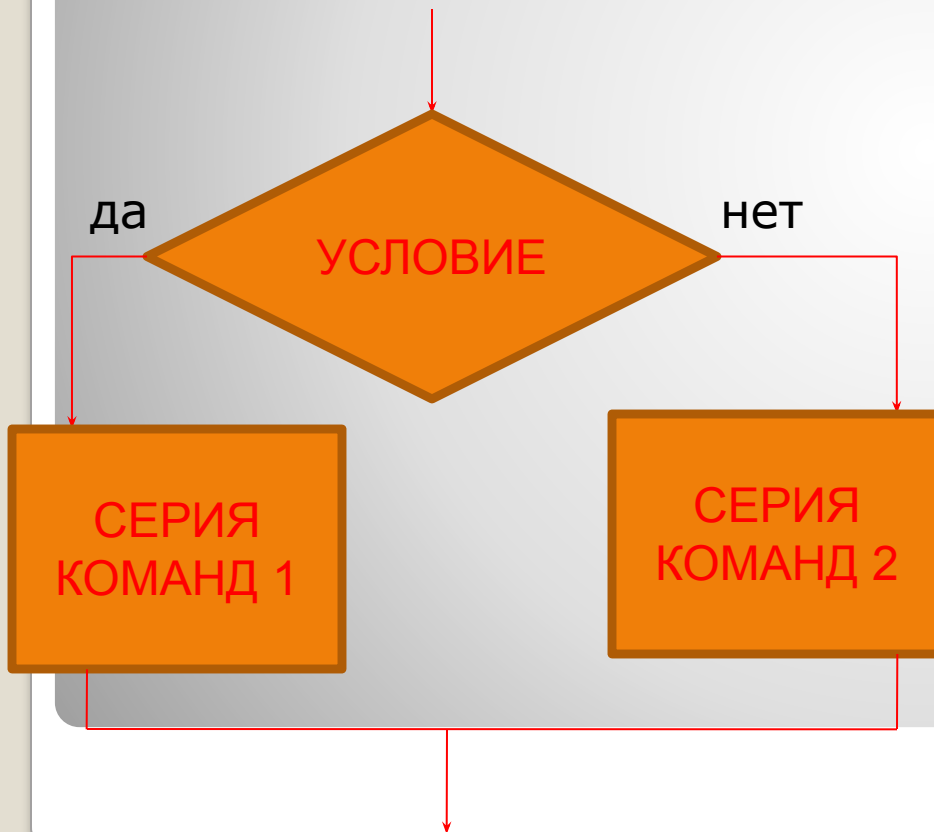
```
End Sub
```

The screenshot shows a Windows application window with a grid background. It contains three text input fields labeled "Сторона а", "Сторона б", and "Сторона с" arranged vertically. To the right of the "Сторона а" field is a button labeled "Площадь треугольника". Below the "Сторона б" field is a label "Ответ:" followed by a large empty rectangular area for the result. At the bottom right is a button labeled "Очистить".

# ВЕТВЯЩИЙСЯ АЛГОРИТМ -

- это алгоритм, в котором выполняется одна или другая серия команд в зависимости от истинности или ложности условия

ФРАГМЕНТ БЛОК-СХЕМЫ  
ВЕТВЯЩЕГОСЯ АЛГОРИТМА



ФРАГМЕНТ  
ВЕТВЯЩЕГОСЯ  
АЛГОРИТМА НА PASCAL

```
If <УСЛОВИЕ> then  
begin  
  <СЕРИЯ КОМАНД 1>  
end  
Else  
begin  
  <СЕРИЯ КОМАНД 2>  
end;
```

# Пример программы ветвящегося алгоритма

Задание: Составить программу вычисления корней квадратного уравнения в зависимости от дискриминанта

```
Private Sub Command1_Click()
```

```
a = Val(Текст1.Text)
```

```
b = Val(Текст2.Text)
```

```
c = Val(Текст3.Text)
```

```
d = b ^ 2 - 4 * a * c
```

```
If d > 0 Then
```

```
Метка6.Caption = (-b + Sqr(d)) / (2 * a)
```

```
Метка7.Caption = (-b - Sqr(d)) / (2 * a)
```

```
Else
```

```
If d = 0 Then
```

```
Метка6.Caption = -b / (2 * a)
```

```
Else
```

```
Метка6.Caption = "КОРНЕЙ НЕТ"
```

```
End If
```

```
End If
```

```
End Sub
```

The screenshot shows a Windows application window with a dotted border. It contains three text input fields for coefficients: "Коэффициент а", "Коэффициент б", and "Коэффициент с". Below these is a button labeled "Корни квадратного уравнения". At the bottom, there are two labels for the roots: "X1=" and "X2=", each followed by a dotted rectangular area for the result.

# АЛГОРИТМИЧЕСКАЯ СТРУКТУРА «ВЫБОР» -

- это алгоритм, в котором выполняется одна из нескольких последовательностей команд при истинности соответствующего условия (удобно использовать вместо вложенного ветвления IF)

ФРАГМЕНТ БЛОК-СХЕМЫ  
АЛГОРИТМА ВЫБОР



ФРАГМЕНТ АЛГОРИТМА  
«ВЫБОР» НА BASIC

**Select Case**

<ВЫРАЖЕНИЕ>

**Case** <УСЛОВИЕ 1>

<СЕРИЯ КОМАНД 1>

**Case** <УСЛОВИЕ 2>

<СЕРИЯ КОМАНД 2>

.....

**Case Else**

<СЕРИЯ КОМАНД N>

**End.select**

# Пример программы алгоритма «выбор»

Задание: Составить программу-тест вычисления отметки по количеству ошибок

```
Private Sub Command1_Click()
```

```
fi = Текст1.Text
```

```
k = Текст2.Text
```

```
sh = Текст3.Text
```

```
n = Текст4.Text
```

**Select Case n**

**Case 0**

```
Текст5.Text = fi + " " + k + " " + sh + " " + "ОТЛИЧНО"
```

**Case 1**

```
Текст5.Text = fi + " " + k + " " + sh + " " + "ХОРОШО"
```

**Case 2**

```
Текст5.Text = fi + " " + k + " " + sh + " " +  
"УДОВЛЕТВОРИТЕЛЬНО"
```

**Case Else**

```
Текст5.Text = fi + " " + k + " " + sh + " " +  
"НЕУДОВЛЕТВОРИТЕЛЬНО"
```

**End Select**

ФИ ученика	Текст1
Класс	Текст2
Школа	Текст3
Количество ошибок в итоговом тесте по информатике	Текст4
Итоговая отметка	Текст5

Запустить проект  
"ОТМЕТКА"

# АЛГОРИТМИЧЕСКАЯ СТРУКТУРА «ЦИКЛ» -

- это алгоритм, в котором серия команд (она называется телом цикла) выполняется многократно.
- Тело цикла определяет ЧТО повторять, а заголовок цикла (вид цикла) определяет СКОЛЬКО РАЗ повторять.

## ВИДЫ ЦИКЛОВ

Безусловный  
цикл

**Цикл  
со счетчиком  
(цикл n раз)**

Условный  
цикл

**Цикл  
с условием**

**Цикл  
с предусловием**

**Цикл  
с постусловием**





## Цикл со счетчиком (цикл n раз)

Используется, когда заранее известно, сколько раз необходимо выполнить цикл.

ФРАГМЕНТ АЛГОРИТМА ЦИКЛА  
n РАЗ НА АЯ  
(алгоритмическом языке)

**НЦ** <ПАРАМЕТР ЦИКЛА> **ОТ** <НАЧАЛЬНОЕ ЗНАЧЕНИЕ> **ДО** <КОНЕЧНОЕ  
ЗНАЧЕНИЕ> **ШАГ**  $h$

**ПОВТОРЯТЬ**

<СЕРИЯ КОМАНД – ТЕЛО ЦИКЛА >

**КЦ** ФРАГМЕНТ АЛГОРИТМА  
ЦИКЛА n РАЗ НА АЯ

**For** <СЧЕТЧИК> := <НАЧАЛЬНОЕ ЗНАЧЕНИЕ> **to** (или **downto**) <КОНЕЧНОЕ  
ЗНАЧЕНИЕ> **do**

**begin**

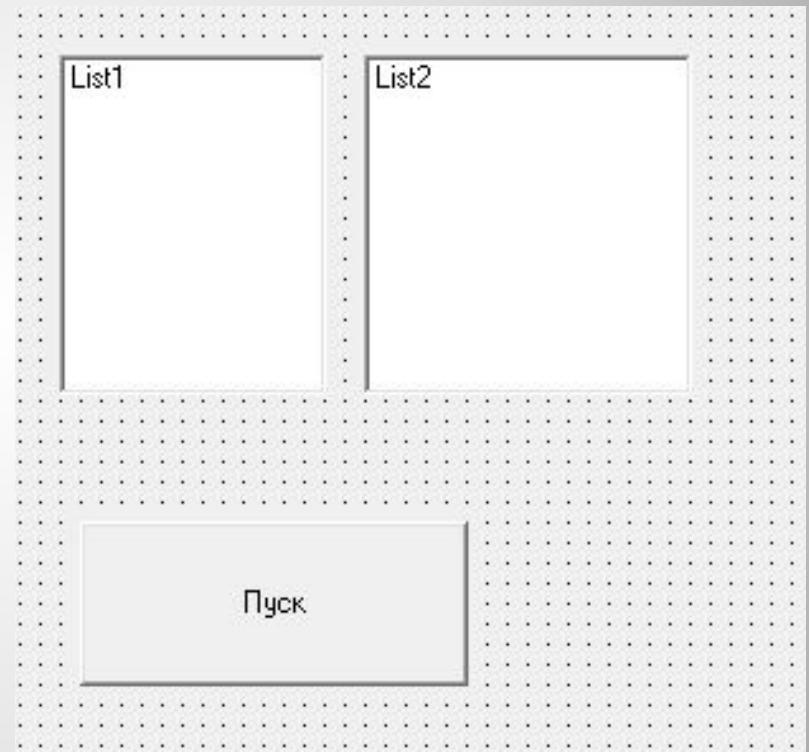
<СЕРИЯ КОМАНД – ТЕЛО ЦИКЛА >

**End;**

# Пример программы алгоритма «цикл n раз»

Задание: Составить программу вывода символов и их кодов (код ASCII)

```
Dim n As Integer  
Private Sub Command1_Click()  
For n = 127 To 32 Step -1  
List1.AddItem (n)  
List2.AddItem (Chr(n))  
Next n  
End Sub
```



## Цикл с предусловием

Используется, когда заранее неизвестно, сколько раз необходимо выполнить цикл.

ФРАГМЕНТ АЛГОРИТМА ЦИКЛА с предусловием  
(условие впереди) НА АЯ (алгоритмическом языке)

**НЦ ПОКА** <УСЛОВИЕ>

**ПОВТОРЯТЬ**

<СЕРИЯ КОМАНД – ТЕЛО  
ЦИКЛА >

Данный вид цикла может не  
выполниться ни разу. Ответьте,  
почему?

КЦ  
ФРАГМЕНТ АЛГОРИТМА ЦИКЛА С ПРЕДУСЛОВИЕМ (НА ЯЗЫКЕ BASIC)  
ЦИКЛ С ИСТИННЫМ ПРЕДУСЛОВИЕМ ЦИКЛ С ЛОЖНЫМ ПРЕДУСЛОВИЕМ

**Do while** <УСЛОВИЕ>

<СЕРИЯ КОМАНД – ТЕЛО ЦИКЛА >

Тело цикла выполняется пока условие  
истинно

**Loop**

**Do Until** <УСЛОВИЕ>

<СЕРИЯ КОМАНД – ТЕЛО ЦИКЛА >

Тело цикла выполняется пока условие

Ложно

**Loop**

# Пример программ алгоритма «цикл с предусловием»

Задание: Составить программу вычисления количества отрицательных чисел среди шести чисел (каждое число вводить отдельной командой ввод в цикле)

```
Dim a As Integer

Private Sub Form_Load()

Show

i = 1: k = 0

Do While i < 7

a = InputBox("введите число")

Print a

If a < 0 Then

k = k + 1

End If

i = i + 1

Loop

Print "Количество отрицательных чисел=";
k

End Sub
```

```
Dim a As Integer

Private Sub Form_Load()

Show

i = 1: k = 0

Do Until i >= 7

a = InputBox("введите число")

Print a

If a < 0 Then

k = k + 1

End If

i = i + 1

Loop

Print "Количество отрицательных чисел=";
k

End Sub
```

## Цикл с постусловием

Используется, когда заранее неизвестно, сколько раз необходимо выполнить цикл.

ФРАГМЕНТ АЛГОРИТМА ЦИКЛА с постусловием  
(условие после) НА АЯ (алгоритмическом языке)

**НЦ  
ПОВТОРЯТЬ**

<СЕРИЯ КОМАНД – ТЕЛО  
ЦИКЛА >

Данный вид цикла выполнится хотя бы один раз. Ответьте, почему?

**КЦ ПОКА** <УСЛОВИЕ>

ФРАГМЕНТ АЛГОРИТМА ЦИКЛА С ПОСТУСЛОВИЕМ НА ЯЗЫКЕ BASIC  
ЦИКЛ С ИСТИННЫМ ПОСТУСЛОВИЕМ ЦИКЛ С ЛОЖНЫМ ПОСТУСЛОВИЕМ

**Repeat**

<СЕРИЯ КОМАНД – ТЕЛО ЦИКЛА >

Тело цикла выполняется пока условие

Ложно

**Until** <УСЛОВИЕ>;

**Do**

<СЕРИЯ КОМАНД – ТЕЛО ЦИКЛА >

Тело цикла выполняется пока условие

Ложно

**Loop Until** <УСЛОВИЕ>

# Пример программ алгоритма

## «цикл с постусловием»

Задание: Составить программу вычисления количества отрицательных чисел среди шести чисел (каждое число вводить отдельной командой ввод в цикле)

```
Dim a As Integer
```

```
Private Sub Form_Load()
```

```
Show
```

```
i = 1: k = 0
```

```
Do
```

```
a = InputBox("введите число")
```

```
Print a
```

```
If a < 0 Then
```

```
k = k + 1
```

```
End If
```

```
i = i + 1
```

```
Loop While i < 7
```

```
Print "Количество отрицательных чисел=";
```

```
k
```

```
End Sub
```

```
Dim a As Integer
```

```
Private Sub Form_Load()
```

```
Show
```

```
i = 1: k = 0
```

```
Do
```

```
a = InputBox("введите число")
```

```
Print a
```

```
If a < 0 Then
```

```
k = k + 1
```

```
End If
```

```
i = i + 1
```

```
Loop Until i >= 7
```

```
Print "Количество отрицательных чисел=";
```

```
k
```

```
End Sub
```