

# 3. Структура цикл

- Наиболее эффективно проявляются возможности компьютера при многократном выполнении одних и тех же действий с изменяющимися данными.
- Такой вычислительный процесс называется **циклическим**, а описывающая его структура – **циклом**.

# Будем использовать следующие термины и обозначения:

- параметр цикла –  $X$ ,
- начальное значение параметра цикла –  $X_0$ ,
- конечное значение параметра цикла –  $X_k$ ,
- шаг изменения параметра цикла –  $dX$ ,
- условие выполнения цикла –  $X \leq X_k$ ,
- тело цикла – группа повторяющихся операторов.

Различают циклы с параметром и итерационные:

- в циклах с параметром число его повторений (N) заранее известно и зависит от начального значения параметра цикла, его конечного значения и шага

$$N = (X_k - X_0) / dX + 1;$$

- в итерационных циклах повторения заканчиваются когда достигается заданная точность вычислений (нахождение предела функции, корней уравнений и т.п.).

## Циклы с параметром различают :

- циклы с предусловием, когда проверка на окончание цикла осуществляется до повторяющихся действий (в этом случае тело цикла может не выполниться ни разу)

И

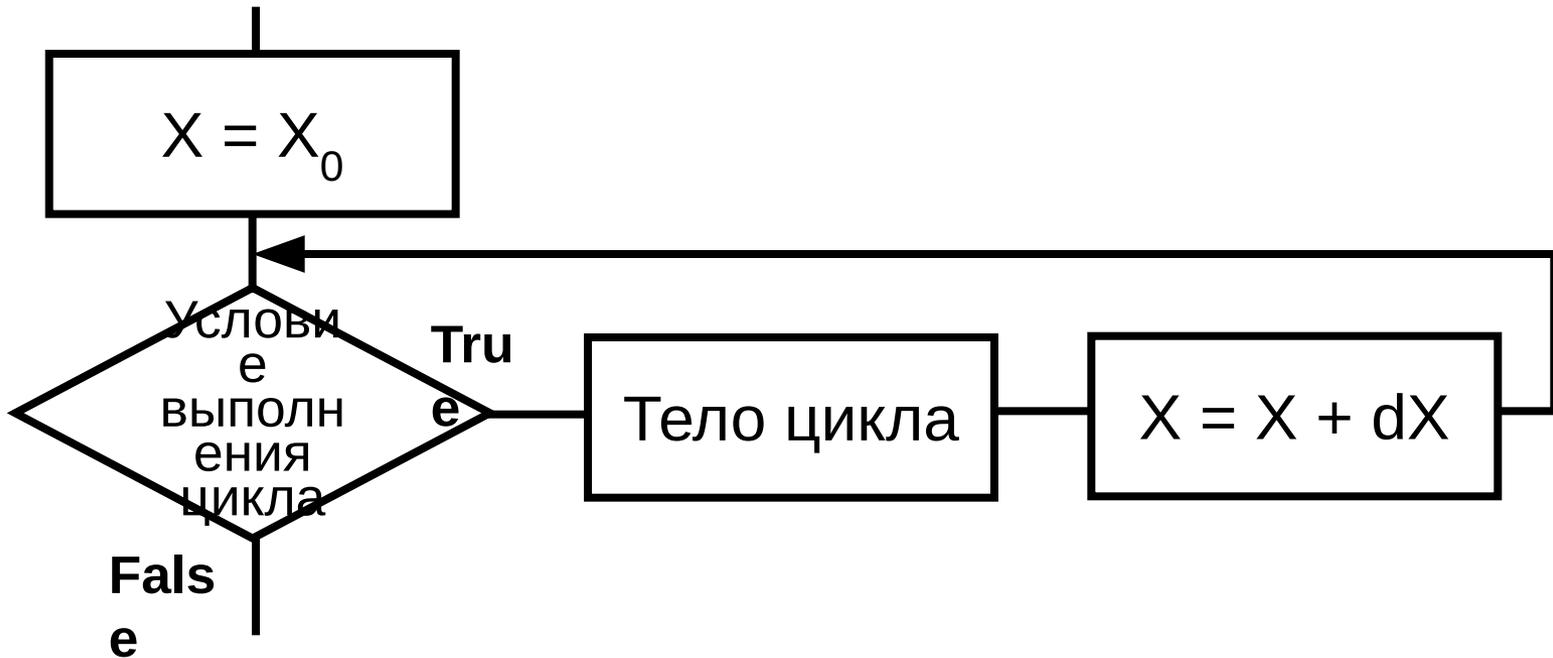
- циклы с постусловием, когда проверка на окончание цикла осуществляется после выполненных действий (в этом случае тело цикла выполнится хотя бы один раз).

# Цикл с условием

Организуется следующим образом:

1. Параметру цикла  $X$  присваивается начальное значение  $X_0$ .
2. Проверяется условие выполнения цикла.
3. Если это условие истинно, то выполняется тело цикла, если ложно, осуществляется переход к оператору следующему за циклом.
4. Значение параметра цикла изменяется на величину шага и далее снова выполняется пункт 2.

# Схема алгоритма:



Цикл с предусловием реализуется несколькими способами.

# Оператор For . . . Next

(используется только для цикла с предусловием)

---

```
For < Параметр цикла > = < Начальное значение >  
  To < Конечное значение > [Step < Шаг >]  
< Тело цикла >  
Next [Параметр цикла]
```

```
For X = X0 To Xk Step dX  
< Тело цикла >  
Next X
```

- Параметр цикла может изменять свое значение в сторону уменьшения. Для этого его начальное значение задается больше конечного, а шагу присваивается отрицательное значение.
- Досрочный выход из цикла (прерывание) осуществляется оператором **Exit For**

# Пример 1.

## 1. Постановка задачи

Вычислить значение функции  $Y = \sin X$  при значениях аргумента меняющегося от 0 до 1 с шагом 0,1

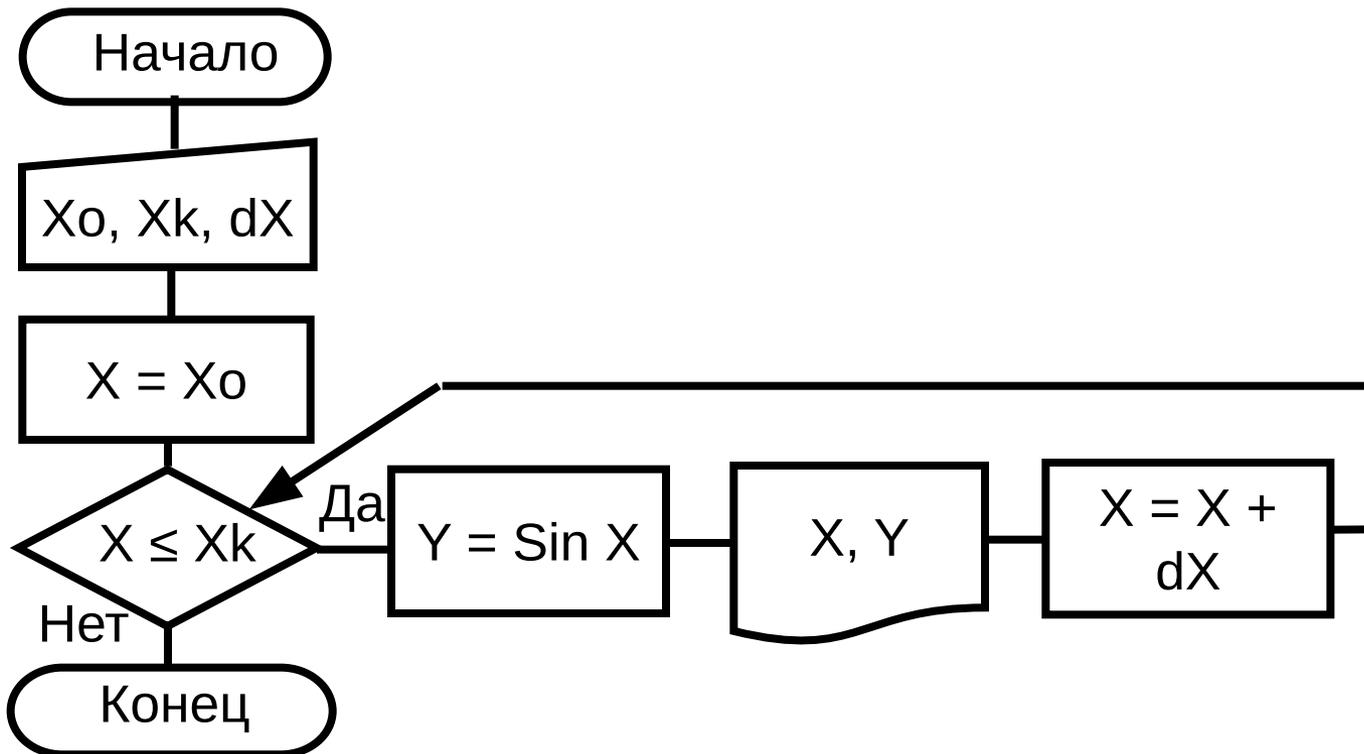
*Входные данные:*  $X_0, X_k, dX$

*Выходные данные:*  $X, Y$

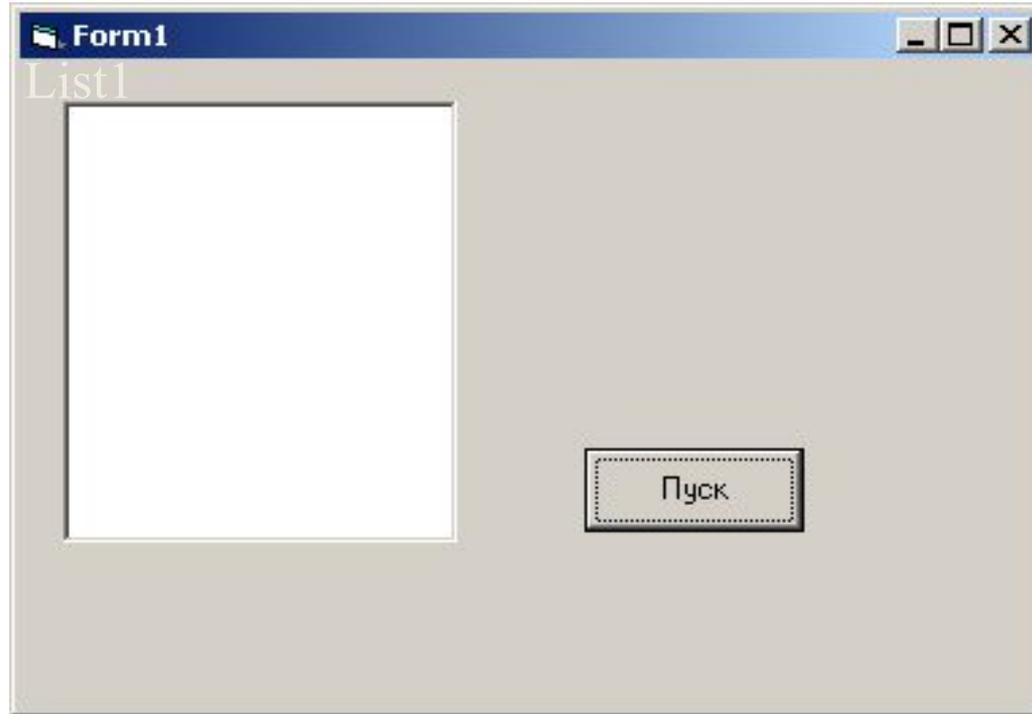
## 2. Математическая модель

$y = \sin x$  для всех  $0 \leq x \leq 1$  с шагом 0,1

## 3. Схема алгоритма



## 4. Разработка визуальной части проекта

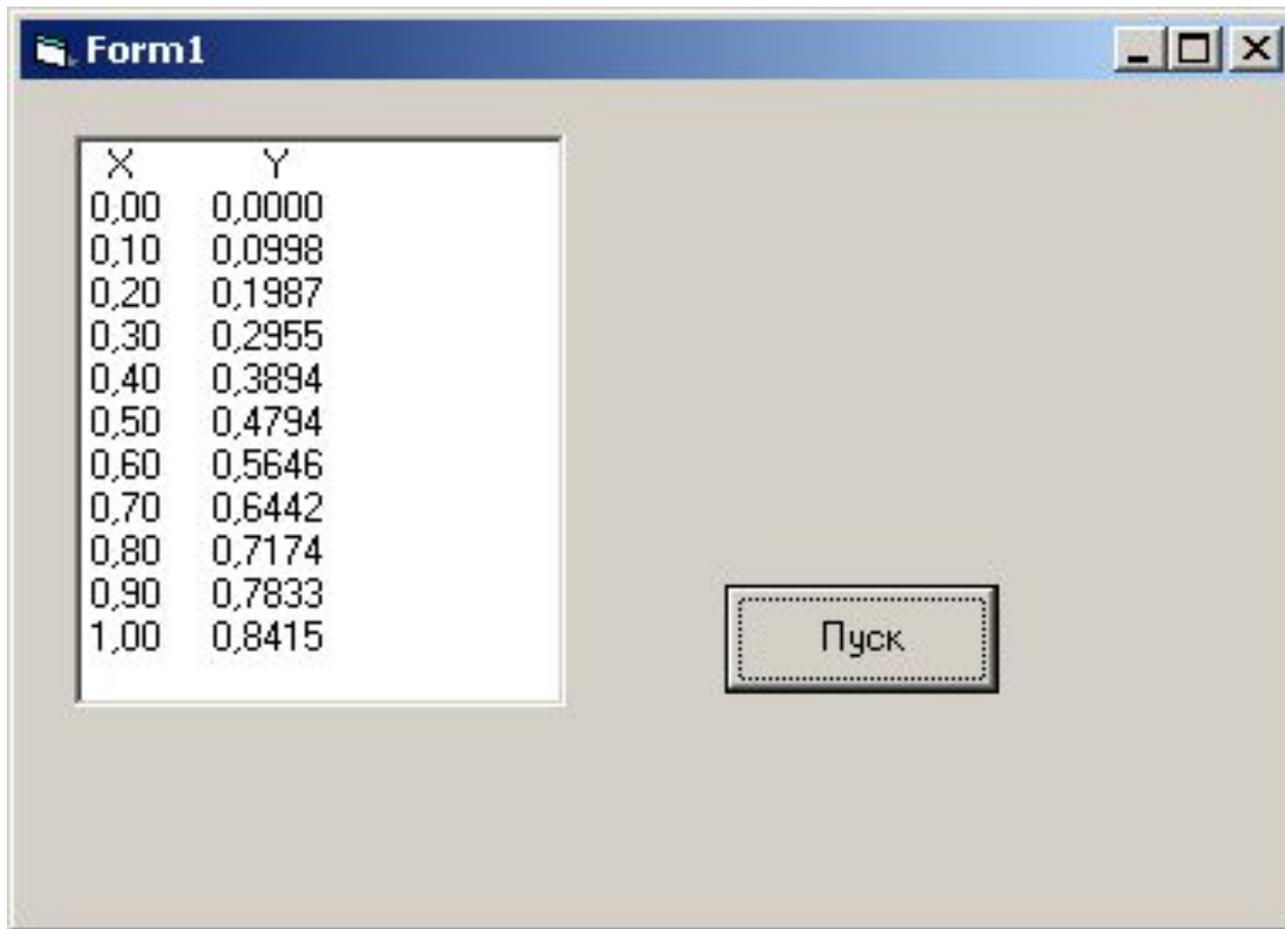


ListBox – поле списка  
(см. Основные элементы управления панели General)

## 5. Код приложения

```
Private Sub Command1_Click()  
Dim x0 As Single, xk As Single, dx As Single  
Dim y As Single, x As Single  
x0 = InputBox("Введите x0")  
xk = InputBox(" Введите xk")  
dx = InputBox(" Введите dx")  
List1.AddItem " X          Y "  
For x = x0 To xk + dx / 2 Step dx  
y = Sin(x)  
List1.AddItem Format(x, "0.00") & "    " & Format(y, "0.0000")  
Next  
End Sub
```

## 6. Отладка программы



1. Назовите основные алгоритмические структуры
2. Перечислите типы данных, которые поддерживает VB
3. Какой оператор языка программирования VB позволяет запрограммировать разветвляющиеся вычислительные процессы
4. Какое из арифметических действий имеет больший приоритет (т.е. выполняется раньше) – умножение или возведение в степень
5. Как будет выглядеть на VB следующее

$$\gamma = \sqrt[3]{1 - ctgx^2}$$

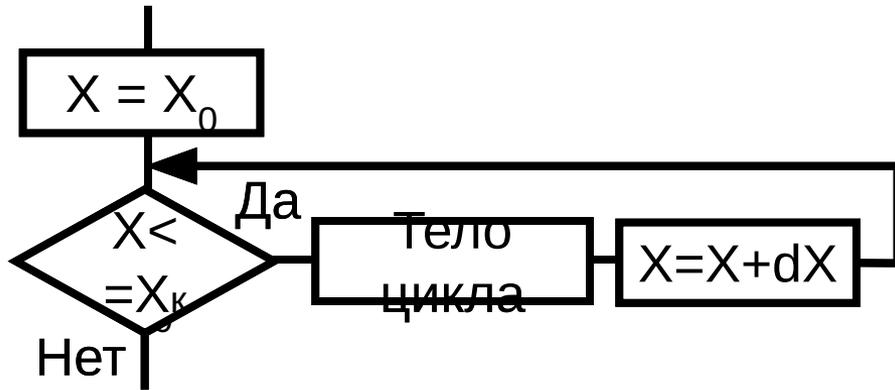
# Цикл с постусловием

Организуется следующим образом:

1. Задается начальное значение параметра цикла  
 $X=X_0$
2. Выполняется тело цикла.
3. Значение параметра цикла изменяется на величину шага.
4. Проверяется условие продолжения цикла.
5. Если условие истинно, то вновь выполняется тело цикла (переход к пункту 2), если же условие ложно, то выполняется следующий после цикла оператор.

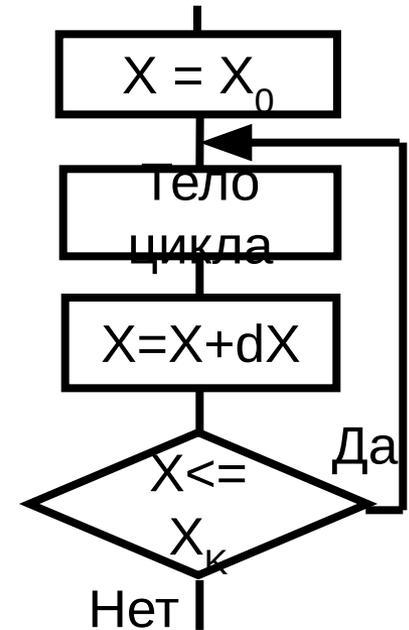
# Оператор Do While ... Loop

С предусловием:



$X = X_0$   
Do While  $X \leq X_k$   
< Тело цикла >  
 $X = X + dX$   
Loop

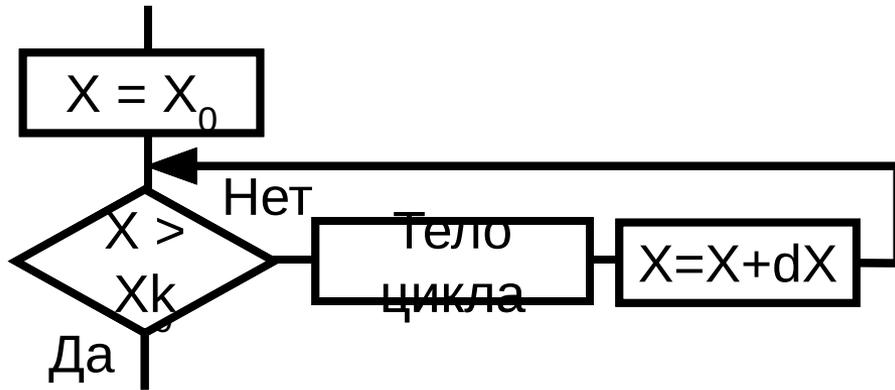
С постусловием:



$X = X_0$   
Do  
< Тело цикла >  
 $X = X + dX$   
Loop While  $X \leq X_k$

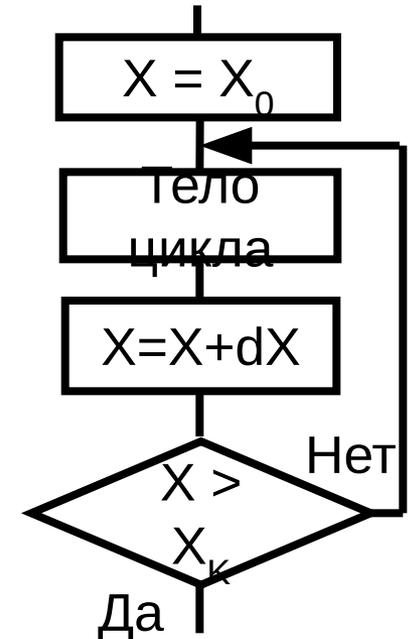
# Оператор Do Until . . . Loop

С предусловием:



$X = X_0$   
Do Until  $X > X_k$   
< Тело цикла >  
 $X = X + dX$   
Loop

С постусловием:

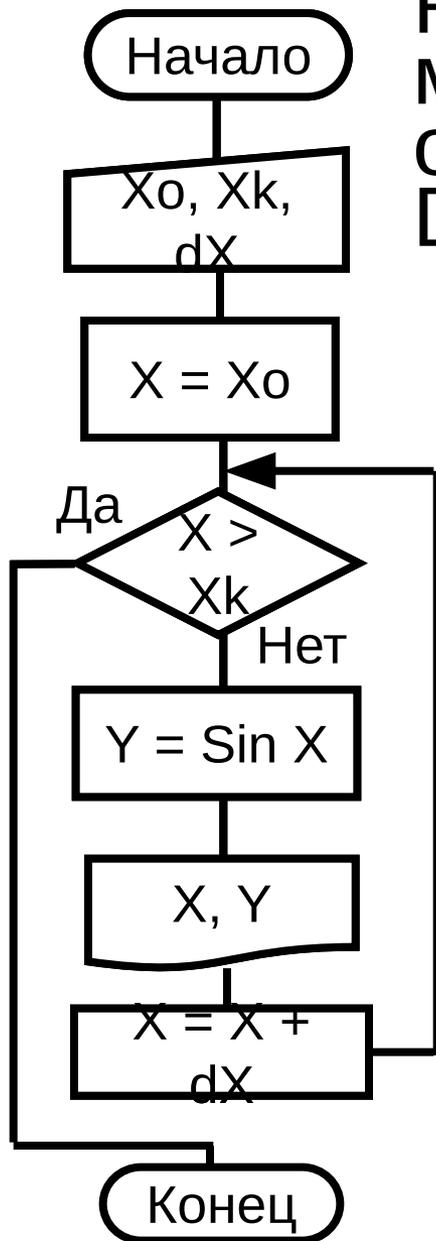


$X = X_0$   
Do  
< Тело цикла >  
 $X = X + dX$   
Loop Until  $X > X_k$

## NB

1. При использовании операторов **Do While** и **Do Until** перед их записью следует присвоить параметру цикла начальное значение (чего не надо делать перед оператором For).
2. После тела цикла должен стоять оператор изменяющий параметр цикла на величину шага.
3. Цикл **Do While** выполняется пока логическое выражение остается истинным.
4. Цикл **Do Until** выполняется пока логическое выражение не станет истинным.
5. Для досрочного выхода из этих циклов используют команду **Exit Do**

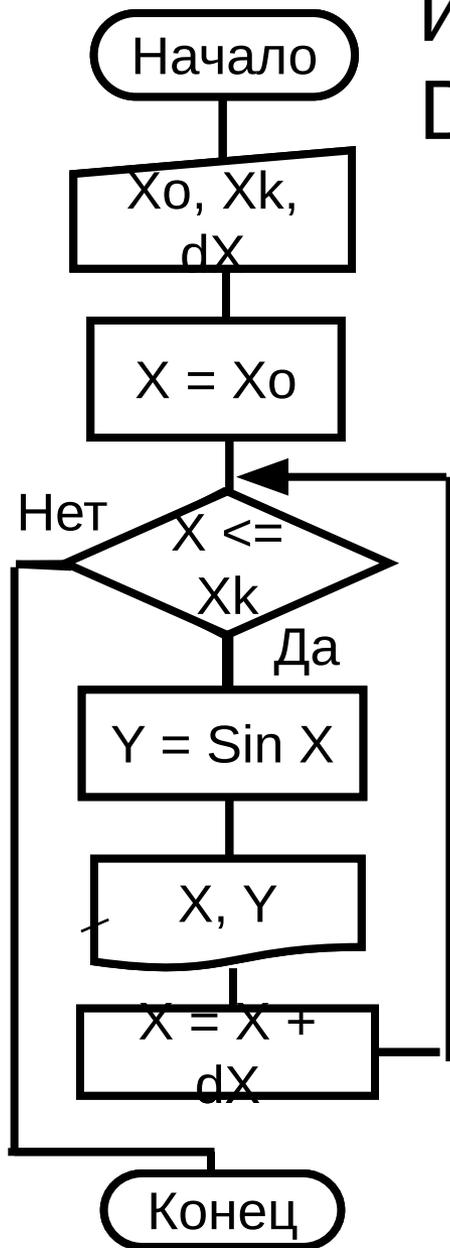
Рассмотренный выше пример  
можно реализовать с помощью  
оператора  
Do Until ... Loop (с предусловием)



...  
 $X = X_0$   
Do until  $X > X_k$   
 $Y = \text{Sin } (X)$   
List1.AddItem ...  
 $X = X + dX$   
Loop

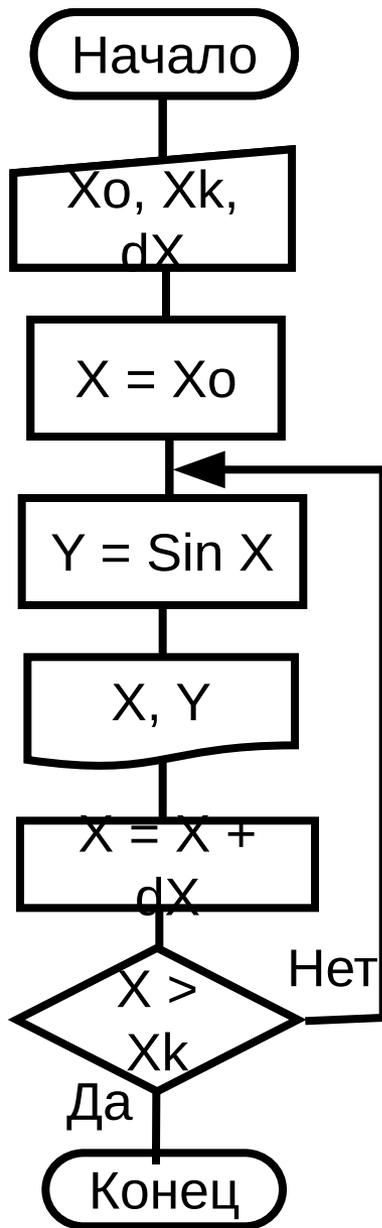
...

или с помощью оператора  
Do While...Loop (с предусловием)



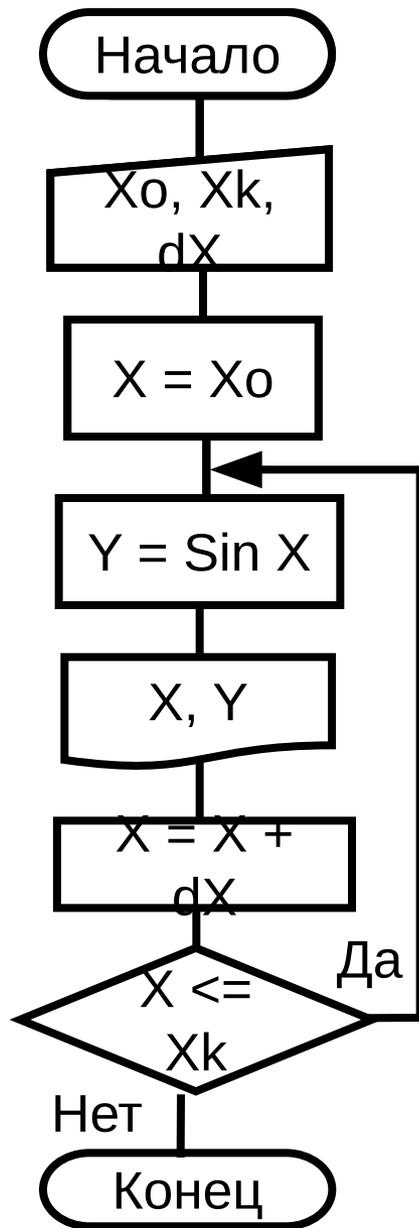
...  
**X = X0**  
**Do While X <= Xk+dX/2**  
**Y = Sin (X)**  
**List1.AddItem ...**  
**X = X + dX**  
**Loop**  
...

# А теперь с помощью оператора Do...Loop Until (с постусловием)



```
...
X = X0
Do
Y = Sin (X)
List1.AddItem ...
X = X + dX
Loop until X > Xk
```

...



или с помощью оператора  
Do...Loop While (с постусловием)

```

...
X = X0
Do
Y = Sin (X)
List1.AddItem ...
X = X + dX
Loop while X <= Xk+dX/2

```

...

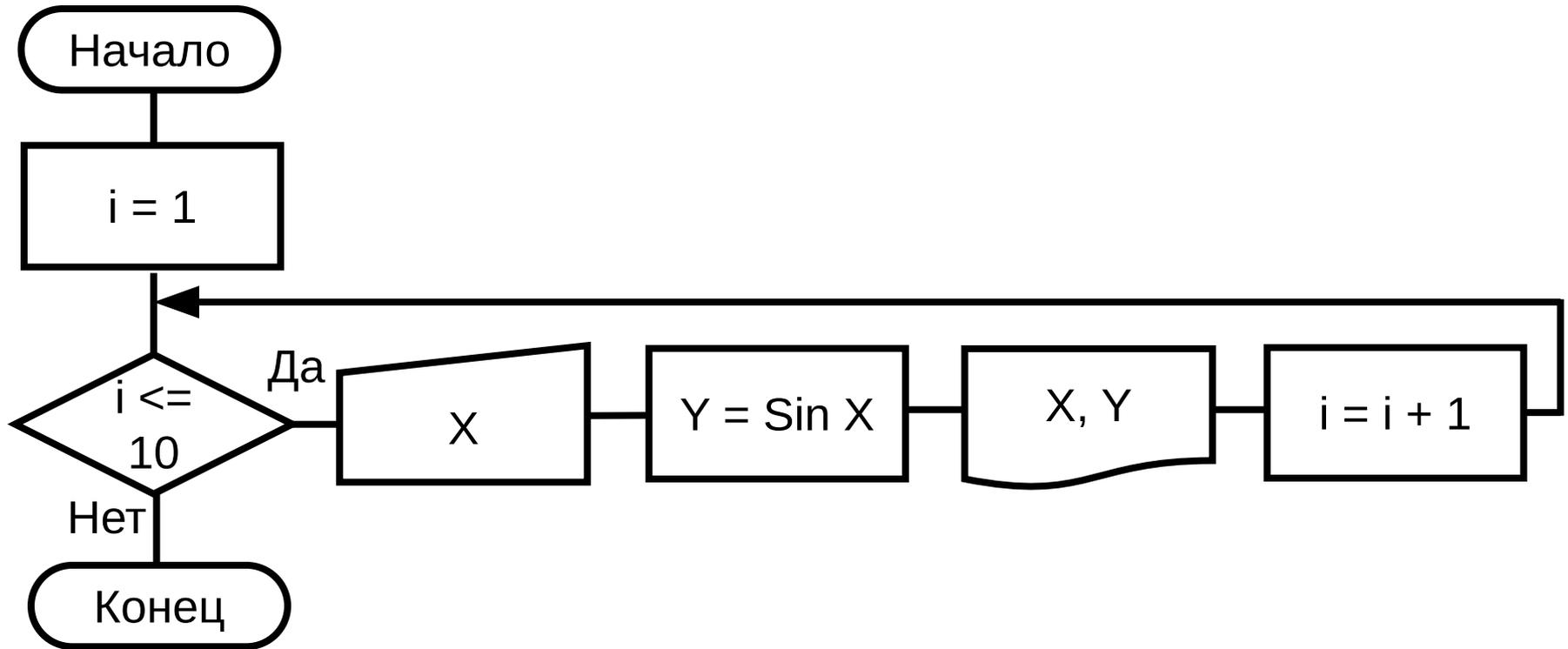
## Пример 2.

Вычислить значения функции  $y = \sin x$ , где значения аргумента заданы последовательностью из десяти случайных чисел не отличающихся постоянным шагом (0 0.1 0.16 0.21 0.34 и т.д.).

*Выходные данные: X, Y*

*Входные данные: X*

# Схема алгоритма



## Код процедуры

```
Private Sub Command1_Click()  
Dim I As Integer, X As Single, Y As Single  
For I = 1 To 10  
X = InputBox ( "Введите значение X")  
Y = Sin ( X )  
Print "X=" & Format(X,"0.00") & " Y=" & Format(Y,"0.00")  
Next I  
End Sub
```

# Структура «Цикл в цикле»

- это структура с одним или несколькими вложенными циклами.

Рассмотрим пример:

$$Z = \text{Sin } X + \text{Cos } Y ,$$

где

$X_0 \leq X \leq X_k$  с шагом  $dX$

$Y_0 \leq Y \leq Y_k$  с шагом  $dY$

Введем следующие понятия и обозначения:

$X$  – параметр внешнего цикла,

$Y$  – параметр внутреннего цикла,

$X_0$  – нач. значение параметра внешнего цикла,

$Y_0$  – нач. знач. параметра внутреннего цикла,

$X_k$  – кон. значение параметра внешнего цикла,

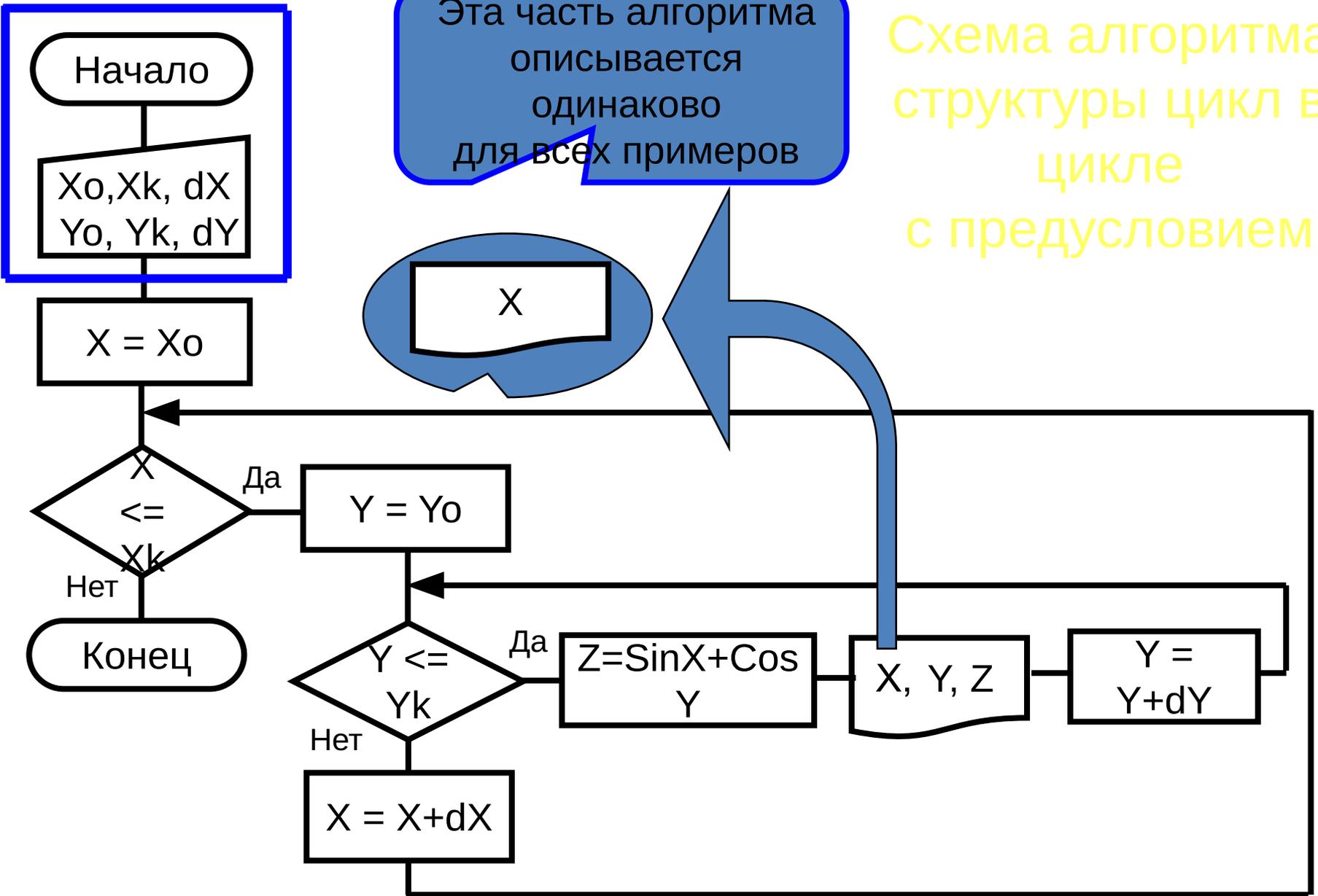
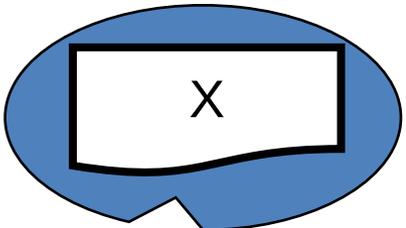
$Y_k$  – кон. знач. параметра внутреннего цикла,

$dX$  – шаг изменения параметра внешнего  
цикла,

$dY$  – шаг изменения параметра внутреннего  
цикла.

Схема алгоритма  
структуры цикл в  
цикле  
с предусловием

Эта часть алгоритма  
описывается  
одинаково  
для всех примеров



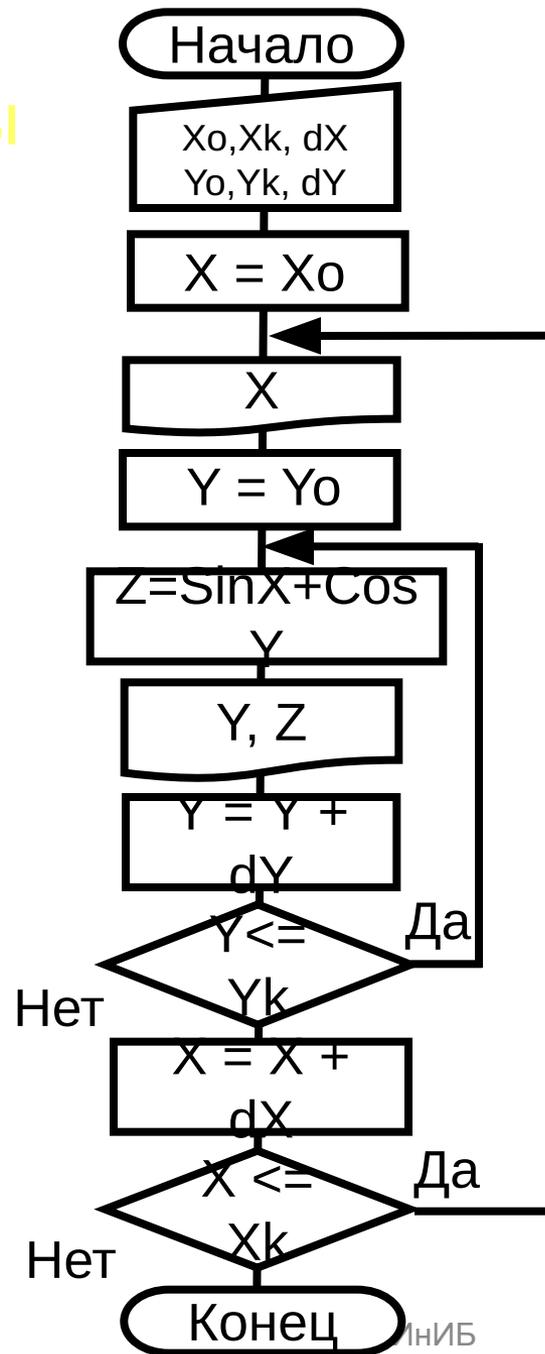
```
Private Sub Command1_Click()  
Dim X0 As Single, Xk As Single, dX As Single  
Dim Y0 As Single, Yk As Single, dY As Single  
Dim Z As Single  
X0 = InputBox (“Введите начальное значение X”)  
Xk = InputBox (“Введите конечное значение X”)  
dX = InputBox (“Введите шаг изменения X”)  
Y0 = InputBox (“Введите начальное значение Y”)  
Yk = InputBox (“Введите конечное значение Y”)  
dY = InputBox (“Введите шаг изменения Y”)  
...  
...
```

# Варианты продолжения кода процедуры

```
...  
X = X0  
Do While X <= Xk  
Print X  
Y = Y0  
Do While Y <= Yk  
Z = Sin (X) + Cos (Y)  
Print Y, Z  
Y = Y + dY  
Loop  
X = X + dX  
Loop  
End Sub
```

```
...  
For X=X0 To Xk Step dX  
Print X  
For Y=Y0 To Yk Step dY  
Z = Sin (X) + Cos (Y)  
Print Y, Z  
Next Y  
Next X  
End Sub
```

Схема структуры  
цикл в цикле  
с постусловием



На следующем слайде  
представлен фрагмент  
кода процедуры  
этой части алгоритма

```

...
X = Xo
Do
Print "X=" & X
Y = Yo
Do
Z = Sin (X) + Cos (Y)
Print "Y = " & Y & " Z = " & Z
Y = Y + dY
Loop While Y <= Yk
X = X + dX
Loop While X <= Xk
End Sub

```

Фрагмент кода процедуры  
без описаний переменных  
и ввода значений входных  
переменных