

Информатика. Часть I

Информатика. Часть I

/Материалы к лекции 4/

Основные понятия объектно-ориентированного языка программирования

1. *Объект* – объединение данных с кодом, предназначенным для их обработки
2. *Семейство объектов* – объект, содержащий несколько других объектов, как правило, одного и того же типа
Worksheets(1) или Worksheets(“Лист1”)
3. *Класс* – проект, на основе которого будет впоследствии создан конкретный объект. Определяет имя объекта, его свойства и действия, выполняемые над объектом

Основные понятия объектно-ориентированного языка

программирования

4. *Свойство* – атрибут объекта, определяющий характеристики объекта
 $Object.Property = PropertyValue$
`Range("A1").Value="Пример"`
5. *Метод* – действие, выполняемое над объектом
 $Object.Method$
`Worksheets(1).Delete Worksheets("Лист2").Select`
`Worksheets("Лист2").Activate`
6. *Событие* – извещение, которое генерируется в результате действия пользователя или изменением состояния приложения или системы



Объектная модель MS Excel

Иерархия объектов, подчиненных одному объекту *Application*, который соответствует самому приложению MS Excel

Объект Семейство Описание

Application	Активное приложение MS Excel
-------------	------------------------------

Workbook	Workbooks	Открытая рабочая книга
----------	-----------	------------------------

Worksheet	Worksheets	Рабочий лист в рабочей книге
-----------	------------	------------------------------

Style	Styles	Встроенный или пользовательский стиль
-------	--------	---------------------------------------

WorksheetFunction	Объект, контейнер для функций рабочего листа
-------------------	--

Range	Диапазон ячеек
-------	----------------

Полная ссылка на объект

`Application.Workbooks("Книга1").Worksheets("Лист1").Range("A1")`

Применение WorksheetFunction

`Application.WorksheetFunction.Min()`

`Application.WorksheetFunction.Pi()`

Диалоговое окно InputBox

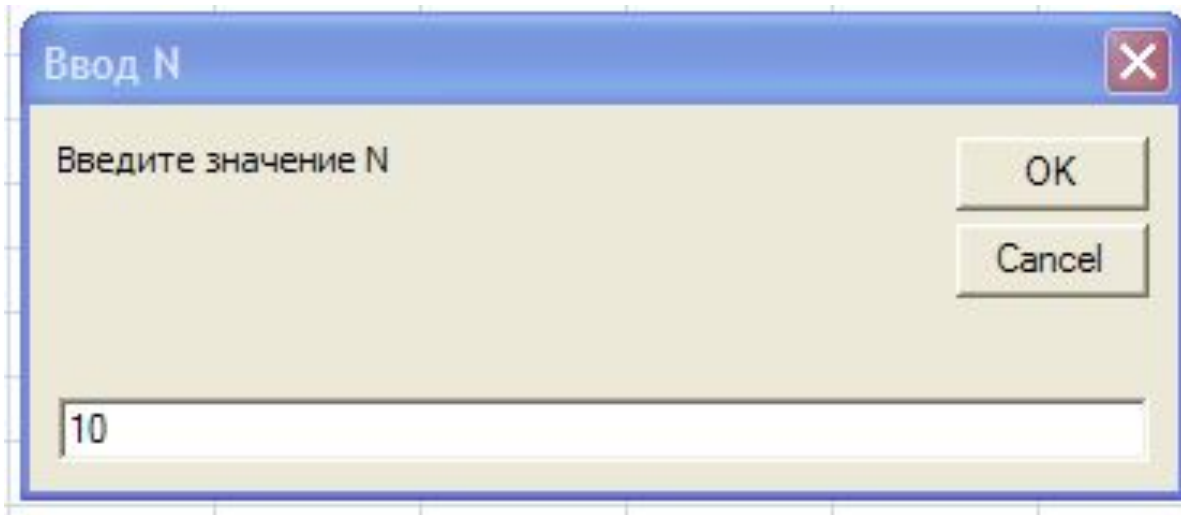
```
N = InputBox("Введите значение N", "Ввод N", 10)
```

Сообщение

Заголовок окна

Значение по умолчанию

Возвращает значение



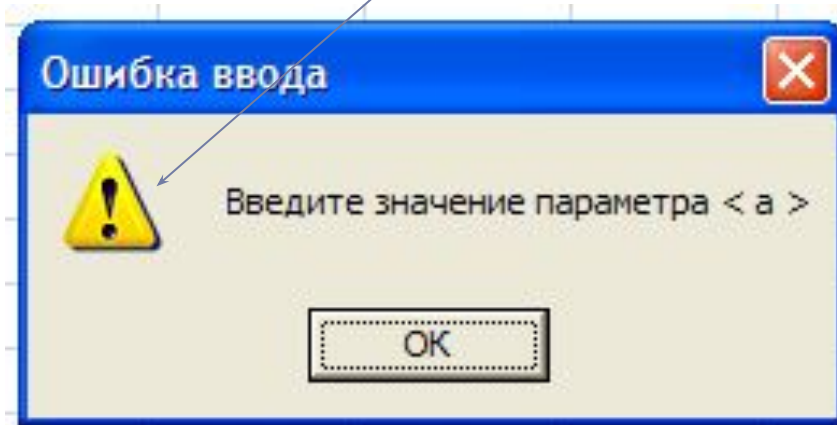
Диалоговое окно MsgBox

MsgBox "Введите значение параметра < a >", vbExclamation,
"Ошибка ввода"

Сообщение

Заголовок окна

Устанавливает режим ожидания



Оператор With

Позволяет избежать повторения имени одного и того же объекта при работе с его свойствами и методами

```
With Range("A1")  
    .Value = 3  
    .Font.Italic = True  
End With
```


Функция RGB

Позволяет получить любой цвет, смешивая красную, зеленую и синюю составляющую различной интенсивности

$\text{RGB}(\textit{Red}, \textit{Green}, \textit{Blue})$

Red, Green, Blue – целые числа от 0 до 255

$\text{RGB}(255, 175, 0)$ – оранжевый

Функция RGB

Цвет	Red	Green	Blue
Черный	0	0	0
Синий	0	0	255
Зеленый	0	255	0
Бирюзовый	0	255	255
Красный	255	0	0
Фиолетовый	255	0	255
Желтый	255	255	0
Белый	255	255	255

Автоматизация проекта с помощью элементов управления

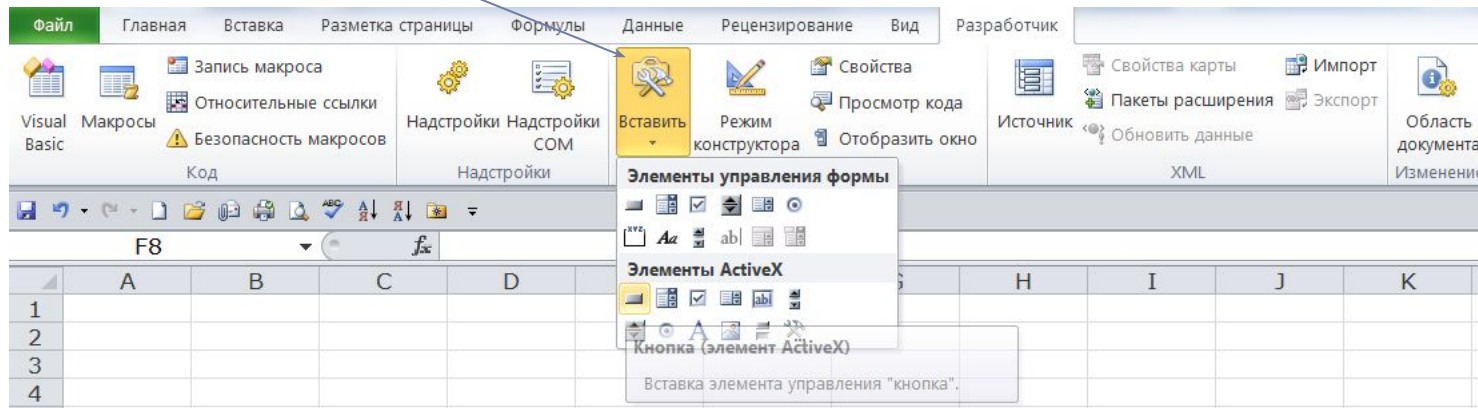
- Добавить в книгу листы: «Содержание», «Табулирование», «Формы», «Массивы»
- На листе «Содержание» создать кнопки перехода на каждый из листов книги, включая уже существующие листы
- На каждом листе сделать кнопки перехода на лист «Содержание»
- На листе «Содержание» создать кнопки «Об авторе», «О книге», «Выход»

Пример создания кнопки перехода

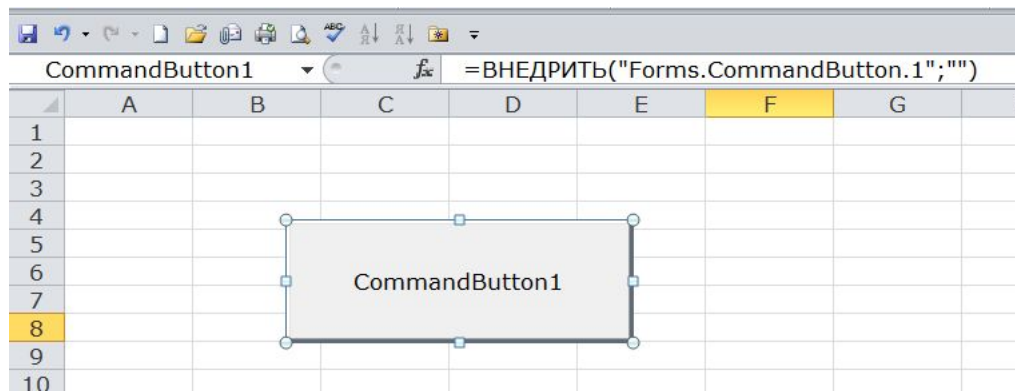
Создадим на листе «Содержание» кнопку перехода на лист «Отчет»

1. Открыть вкладку *Разработчик*.

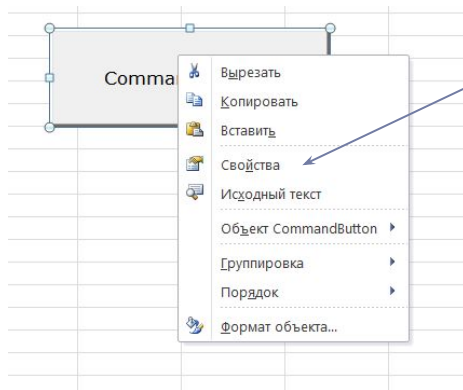
2. Выбрать в *Вставить; Элементы ActiveX* вставку элемента управления *Кнопка*



3. И как в графическом редакторе нарисовать прямоугольник

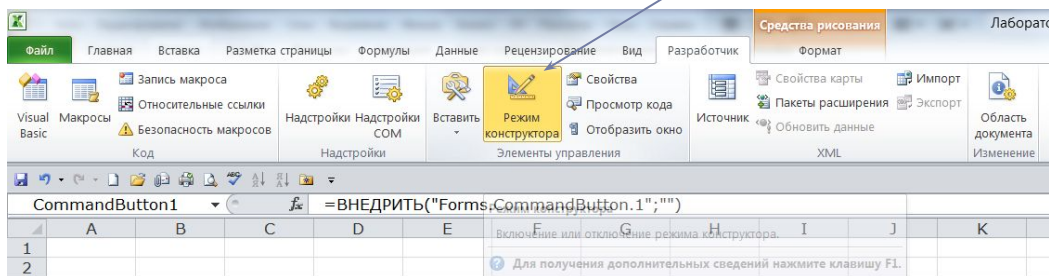


4. Изменить свойства кнопки, используя контекстное меню или кнопку *Свойства* на вкладке *Разработчик*.

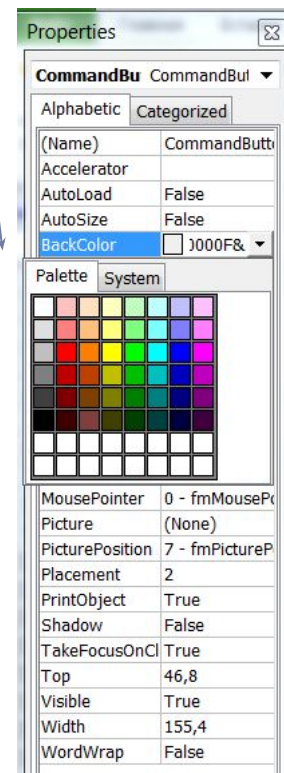
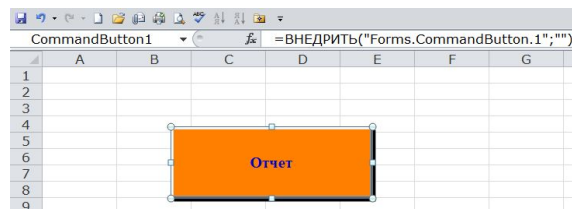


Меняем свойство Caption (заголовок) [не Name !] – на «Отчет» и, например, BackColor (цвет фона), Font (шрифт), ForeColor (цвет текста).

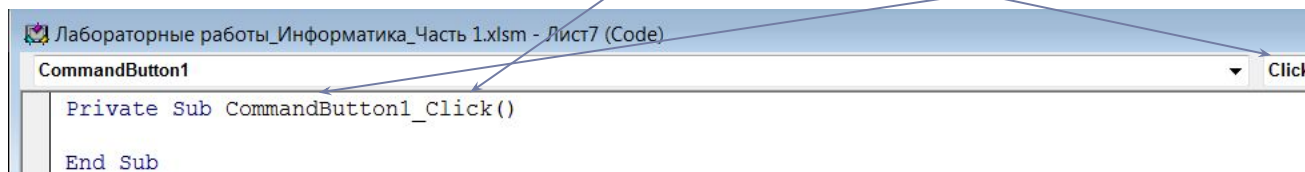
Обратите внимание: при работе с объектом (изменении свойств, перемещении, создании кода и т.п.) включен *Режим конструктора*. Когда все необходимые поправки внесены, *Режим конструктора* можно отключить.



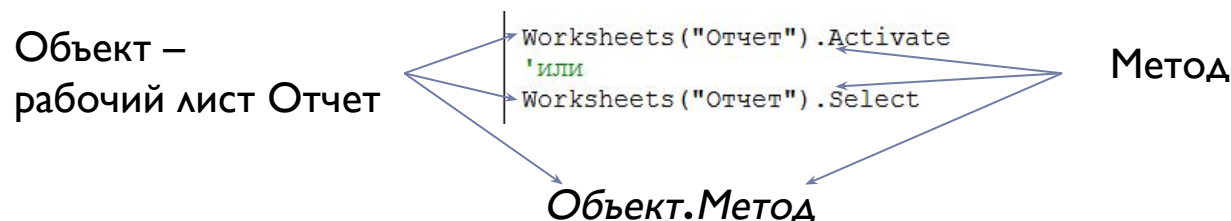
5. В результате, например, получилось:



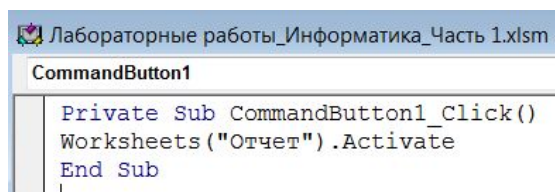
4. Теперь создадим программу, которая будет выполняться при нажатии на кнопку Отчет. После двойного щелчка мыши на кнопке Отчет (в режиме конструктор) откроется окно редактора Visual Basic и автоматически будет создана процедура, выполняющаяся при возникновении в системе события Click (щелчок мыши) с объектом CommandButton1 (свойство Name кнопки, которое мы не меняли)



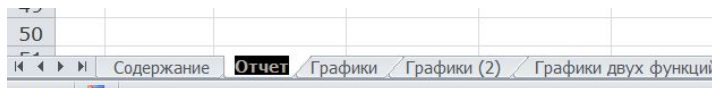
Кнопка должна выполнить переход на лист Отчет. Напишем:



Получилось:



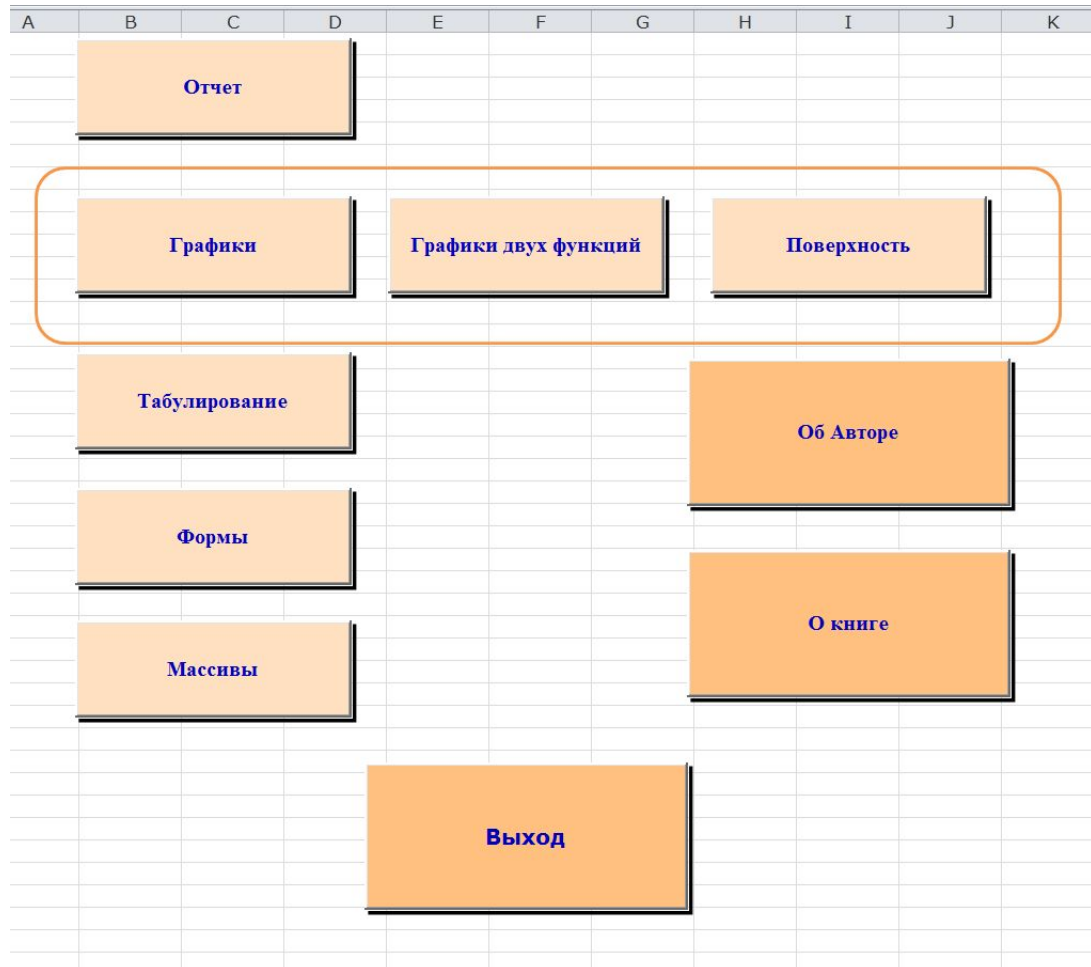
Обратите **внимание**: название листа в программе должно точно соответствовать названию на ярлычке. Для этого его можно скопировать с ярлычка, используя Ctrl+C (копировать), Ctrl+V (вставить)



5. Проверим работоспособность кнопки Отчет, щелкнув на ней при отключенном *Режиме конструктора*.

Создадим другие кнопки, в том числе перехода на лист «Содержание». (Кнопки можно копировать).

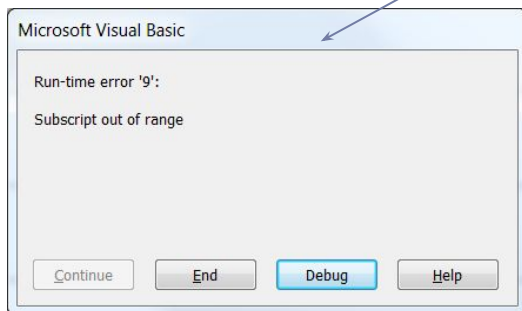
Пример листа «Содержание» :



Для кнопок «Об Авторе», «О книге», «Выход» изменить свойство Name

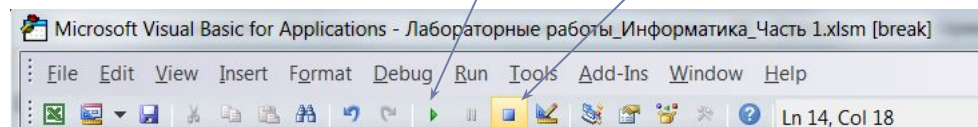
```
Private Sub CmdAvtor_Click()  
MsgBox "Лабораторная работа" + vbCrLf + "студента Иванова И.И." + vbCrLf + "группа  ", vbInformation, "Об авторе"  
End Sub  
  
Private Sub CmdBook_Click()  
MsgBox "В рабочей книге " + Str(Worksheets.Count) + " листов", vbInformation, "О книге"  
End Sub  
  
Private Sub CmdExit_Click()  
Application.Quit  
End Sub
```

Замечание. Если возникает ошибка при отладке, например, появляется сообщение, выбрать Debug.



Строка, в которой обнаружена ошибка, будет выделена. Исправить ошибку.
Для продолжения нажать Continue или Reset.

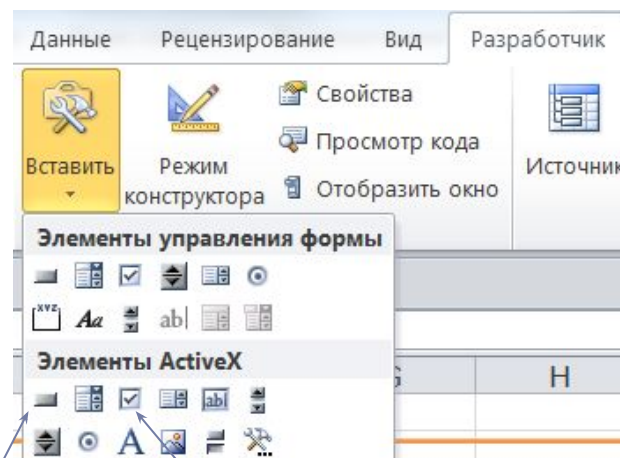
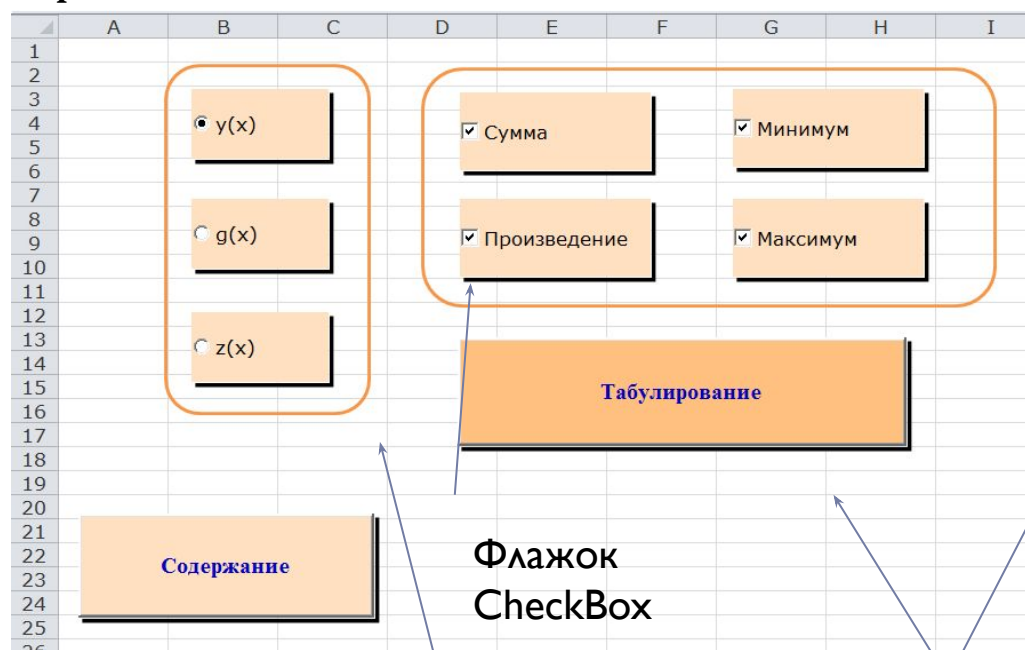
```
Private Sub CommandButton1_Click()  
Worksheets("Отчет ").Activate  
End Sub
```



Табулирование функции

Написать программу табулирования функций $y(x)$, $g(x)$, $z(x)$ из примера I (графики).
Функция, значения которой необходимо вычислить, выбирается с помощью переключателя.

На листе «Табулирование» добавить элементы управления. Изменить свойства Name, Caption.



Флажок
CheckBox

Переключатель
OptionButton

Кнопка
CommandButton

Переключатель
OptionButton

Name	Caption
CmdTab	Табулирование
OptY	y(x)
OptG	g(x)
OptZ	z(x)
CheckSum	Сумма
CheckPrv	Произведение
CheckMin	Минимум
CheckMax	Максимум

Программа табулирования функции

Описание переменных

```
Private Sub CmdTab_Click()
```

```
Dim x As Double, f As Double
```

```
Dim xn As Double, xk As Double, dx As Double
```

```
Dim i As Integer
```

```
Dim c As Range
```

Программа табулирования функции

Ввод исходных данных

```
'Ввод исходных данных
```

```
xn = InputBox("Введите xn=", "Ввод начального значения x", -2)
```

```
xk = InputBox("Введите xk=", "Ввод конечного значения x", 2)
```

```
dx = InputBox("Введите dx=", "Ввод шага dx", 0.2)
```

Программа табулирования функции

Вывод заголовка таблицы

'Удаление содержимого и форматирование столбцов "J" и "K"

```
Range("J:K").Clear
```

'Заголовок (10 - номер столбца J; 11 - номер столбца K)

'в ячейку K2 вводим y, независимо от выбранного переключателя

```
Cells(2, 10) = "x": Cells(2, 11) = "y"
```

'Для того, чтобы заголовок изменялся в зависимости от переключателя,

'использовать условный оператор. Сделать самостоятельно, например:

```
'If OptY Then Cells(2, 11) = "y"
```

Программа табулирования функции Форматирование заголовка таблицы

```
'Форматирование заголовка

With Range("J2:K2")

'Выравнивание по центру
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter

'Цвет фона
    .Interior.Color = RGB(255, 175, 0)

'Шрифт
    .Font.Name = "Times New Roman"
    .Font.Size = 12

'Рамки
    .Borders(xlEdgeLeft).Color = RGB(0, 0, 0)
    .Borders(xlEdgeRight).Color = RGB(0, 0, 0)
    .Borders(xlEdgeBottom).Color = RGB(0, 0, 0)
    .Borders(xlEdgeTop).Color = RGB(0, 0, 0)
    .Borders(xlInsideVertical).Color = RGB(0, 0, 0)

End With
```

Программа табулирования функции

Оператор цикла

```
x = xn: i = 3
```

```
Do
```

```
  If OptY Then f = y(x)
```

```
  If OptG Then f = g(x)
```

```
  If OptZ Then f = z(x)
```

```
  Cells(i, 10) = x: Cells(i, 11) = f
```

```
  x = x + dx: i = i + 1
```

```
Loop Until x > xk
```

Программа табулирования функции Форматирование столбцов значений x и y

'Форматирование результата

```
Range(Cells(3, 10), Cells(i, 10)).NumberFormat = "#0.0"
```

```
Range(Cells(3, 11), Cells(i + 4, 11)).NumberFormat = "#0.000"
```

```
Range(Cells(3, 10), Cells(i, 10)).NumberFormat = "#0.0"
```

```
Range(Cells(3, 10), Cells(i, 10)).HorizontalAlignment = xlCenter
```

Программа табулирования функции

Проверка флажков

```
If CheckSum Then Cells(i + 1, 10) = "Сумма":      Cells(i + 1, 11) = "=sum(k3:k" + CStr(i - 1) + ")"  
If CheckPrv Then Cells(i + 2, 10) = "Произведение": Cells(i + 2, 11) = "=product(k3:k" + CStr(i - 1) + ")"  
If CheckMin Then Cells(i + 3, 10) = "Минимум":    Cells(i + 3, 11) = "=min(k3:k" + CStr(i - 1) + ")"  
If CheckMax Then Cells(i + 4, 10) = "Максимум":    Cells(i + 4, 11) = "=max(k3:k" + CStr(i - 1) + ")"
```


Программа табулирования функции Оператор For Each...Next

```
'отрицательные значения функции покажем синим цветом,  
For Each c In Range(Cells(3, 11), Cells(i - 1, 11))  
    If c.Value < 0 Then c.Font.Color = RGB(0, 0, 255)  
Next c  
  
End Sub
```

CmdTab

```

Private Sub CmdTab_Click()
Dim x As Double, f As Double
Dim xn As Double, xk As Double, dx As Double
Dim i As Integer

'Ввод исходных данных
xn = InputBox("Введите xn=", "Ввод начального значения x", -2)
xk = InputBox("Введите xk=", "Ввод конечного значения x", 2)
dx = InputBox("Введите dx=", "Ввод шага dx", 0.2)

'Удаляем содержимое и форматирование столбцов "J" и "K"
Range("J:K").Clear

'Заголовок (10 - номер столбца J; 11 - номер столбца K)
'в ячейку K2 вводим y, независимо от выбранного переключателя
Cells(2, 10) = "x": Cells(2, 11) = "y"
'Для того, чтобы заголовок изменялся в зависимости от переключателя,
'использовать условный оператор. Сделать самостоятельно, например:
'If OptY Then Cells(2, 11) = "y"

```

Значения по умолчанию задать по
примеру I (графики)

Заголовок меняется в зависимости от
выбранного переключателя. Сделать
самостоятельно.

```

'Форматирование заголовка
With Range("J2:K2")
'Выравнивание по центру
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
'Цвет фона
.Interior.Color = RGB(255, 175, 0)
'Шрифт
.Font.Name = "Times New Roman"
.Font.Size = 12
'Рамки
.Borders(xlEdgeLeft).Color = RGB(0, 0, 0)
.Borders(xlEdgeRight).Color = RGB(0, 0, 0)
.Borders(xlEdgeBottom).Color = RGB(0, 0, 0)
.Borders(xlEdgeTop).Color = RGB(0, 0, 0)
.Borders(xlInsideVertical).Color = RGB(0, 0, 0)
End With

```

```

x = xn: i = 3
Do
    If OptY Then f = Y(x)
    If OptG Then f = g(x)
    If OptZ Then f = z(x)
    Cells(i, 10) = x: Cells(i, 11) = f
    x = x + dx: i = i + 1
Loop Until x > xk

'Форматирование результата
Range(Cells(3, 10), Cells(i, 10)).NumberFormat = "#0.0"
Range(Cells(3, 11), Cells(i + 4, 11)).NumberFormat = "#0.000"
Range(Cells(3, 10), Cells(i, 10)).NumberFormat = "#0.0"
Range(Cells(3, 10), Cells(i, 10)).HorizontalAlignment = xlCenter

If CheckSum Then Cells(i + 1, 10) = "Сумма": Cells(i + 1, 11) = "=sum(k3:k" + CStr(i - 1) + ")"
If CheckPrv Then Cells(i + 2, 10) = "Произведение": Cells(i + 2, 11) = "=product(k3:k" + CStr(i - 1) + ")"
If CheckMin Then Cells(i + 3, 10) = "Минимум": Cells(i + 3, 11) = "=min(k3:k" + CStr(i - 1) + ")"
If CheckMax Then Cells(i + 4, 10) = "Максимум": Cells(i + 4, 11) = "=max(k3:k" + CStr(i - 1) + ")"

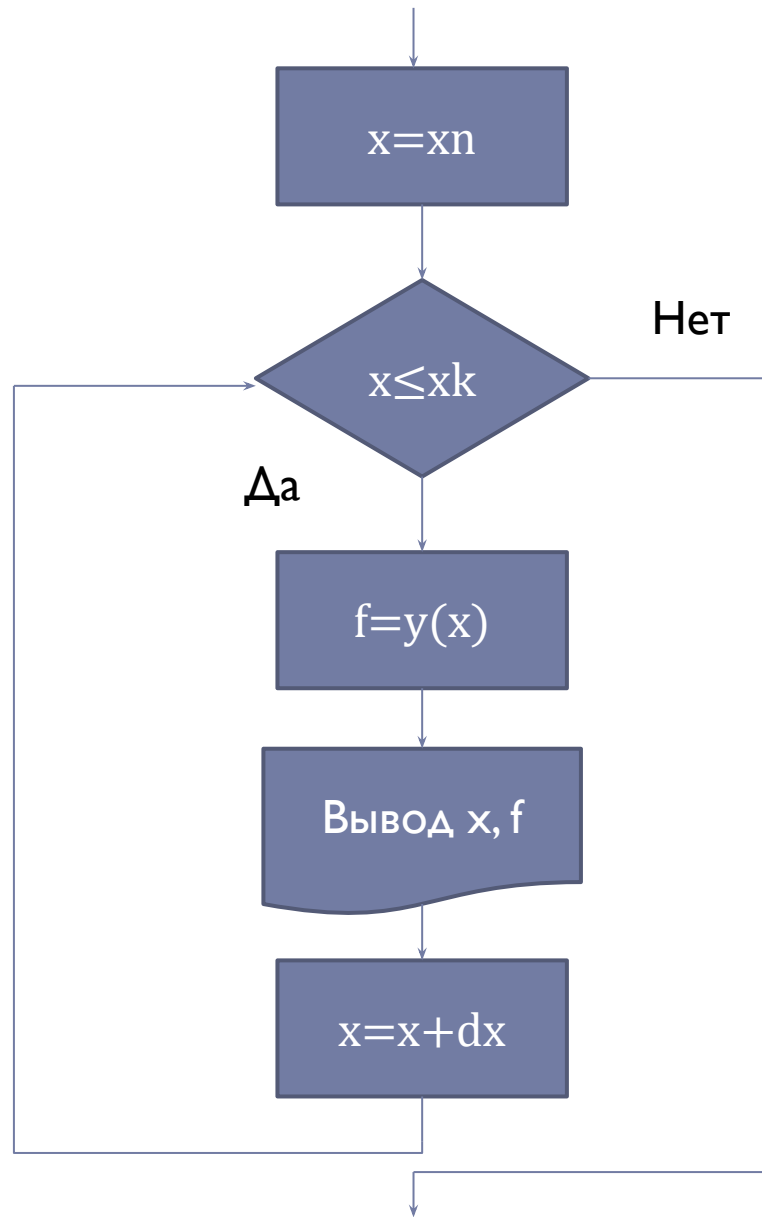
'отрицательные значения функции покажем синим цветом,
For Each c In Range(Cells(3, 11), Cells(i - 1, 11))
    If c.Value < 0 Then c.Font.Color = RGB(0, 0, 255)
Next c
End Sub

```

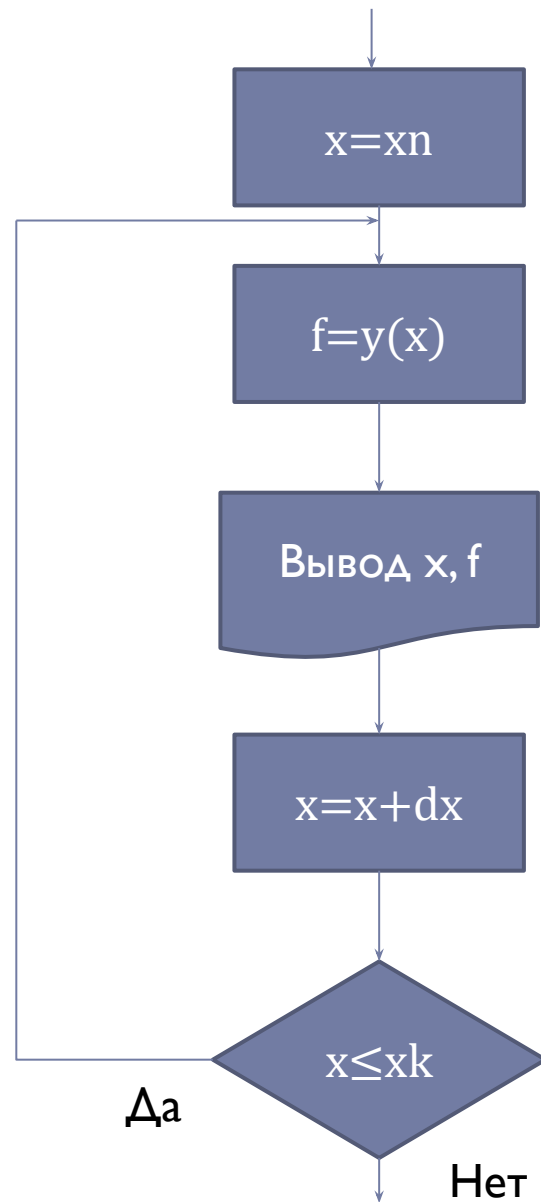
Результат:

J	K
x	y
-2,0	-20,157
-1,8	-17,674
-1,6	-14,853
-1,4	-11,989
-1,2	-9,283
-1,0	-6,862
-0,8	-4,790
-0,6	-3,087
-0,4	-1,743
-0,2	-0,728
0,0	0,000
0,2	0,488
0,4	0,783
0,6	0,930
0,8	0,967
1,0	0,929
1,2	0,842
1,4	0,729
1,6	0,605
1,8	0,483
2,0	0,369
Сумма	-84,039
Произведение	0,000
Минимум	-20,157
Максимум	0,967

Цикл с предусловием



Цикл с постусловием

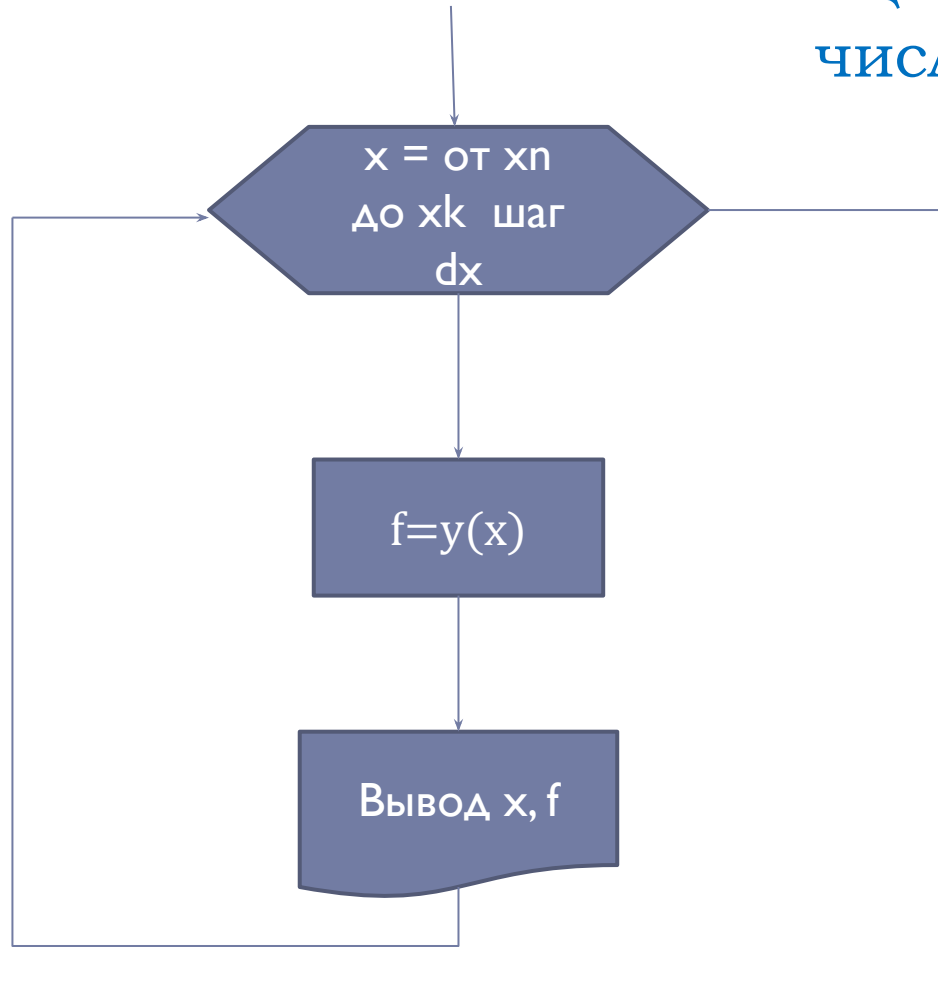


For $x = x_n$ To x_k Step dx

.....

Next

Цикл с заданным
числом повторений



Оператор цикла Do ... Loop While

```
x = xn: i = 36  
  
Do  
  
    If OptY Then f = y(x)  
    If OptG Then f = g(x)  
    If OptZ Then f = z(x)  
  
    Cells(i, 4) = x  
    Cells(i, 5) = f  
  
    x = x + dx: i = i + 1  
  
Loop While x <= xk
```

Оператор цикла Do While ... Loop

```
x = xn: i = 36
```

```
Do While x <= xk
```

```
    If OptY Then f = y(x)
```

```
    If OptG Then f = g(x)
```

```
    If OptZ Then f = z(x)
```

```
    Cells(i, 7) = x
```

```
    Cells(i, 8) = f
```

```
    x = x + dx: i = i + 1
```

```
Loop
```


Оператор цикла Do Until ... Loop

```
x = xn: i = 36

Do Until x > xk

    If OptY Then f = y(x)
    If OptG Then f = g(x)
    If OptZ Then f = z(x)

    Cells(i, 10) = x
    Cells(i, 11) = f

    x = x + dx: i = i + 1

Loop
```

Оператор цикла For ... Next

```
Dim n As Integer, k As Integer

x = xn: i = 36

n = (xk - xn) / dx

For k = 1 To n + 1

    If OptY Then f = y(x)
    If OptG Then f = g(x)
    If OptZ Then f = z(x)

    Cells(i, 16) = x
    Cells(i, 17) = f

    x = x + dx: i = i + 1

Next
```

Оператор цикла For ... Next (2 вариант)

```
i = 36  
  
For x = xn To xk Step dx  
  
    If OptY Then f = y(x)  
    If OptG Then f = g(x)  
    If OptZ Then f = z(x)  
  
    Cells(i, 19) = x  
    Cells(i, 20) = f  
  
    i = i + 1  
  
Next
```

Оператор цикла While ... Wend

```
x = xn: i = 36

While x <= xk

    If OptY Then f = y(x)
    If OptG Then f = g(x)
    If OptZ Then f = z(x)

    Cells(i, 13) = x
    Cells(i, 14) = f

    x = x + dx: i = i + 1

Wend
```

Модифицировать программу: написать несколько программ табулирования, используя различные операторы цикла

30	Do...Loop Until		Do...Loop While		Do While...Loop		Do Until...Loop		While...Wend		For...Next	
31												
32												
33												
34												
35	x	y	x	y	x	y	x	y	x	y	x	y
36	-2	-20,1565	-2	-20,1565	-2	-20,1565	-2	-20,1565	-2	-20,1565	-2	-20,1565
37	-1,8	-17,6743	-1,8	-17,6743	-1,8	-17,6743	-1,8	-17,6743	-1,8	-17,6743	-1,8	-17,6743
38	-1,6	-14,8528	-1,6	-14,8528	-1,6	-14,8528	-1,6	-14,8528	-1,6	-14,8528	-1,6	-14,8528
39	-1,4	-11,9886	-1,4	-11,9886	-1,4	-11,9886	-1,4	-11,9886	-1,4	-11,9886	-1,4	-11,9886
40	-1,2	-9,28344	-1,2	-9,28344	-1,2	-9,28344	-1,2	-9,28344	-1,2	-9,28344	-1,2	-9,28344
41	-1	-6,86207	-1	-6,86207	-1	-6,86207	-1	-6,86207	-1	-6,86207	-1	-6,86207
42	-0,8	-4,78952	-0,8	-4,78952	-0,8	-4,78952	-0,8	-4,78952	-0,8	-4,78952	-0,8	-4,78952
43	-0,6	-3,08654	-0,6	-3,08654	-0,6	-3,08654	-0,6	-3,08654	-0,6	-3,08654	-0,6	-3,08654
44	-0,4	-1,74283	-0,4	-1,74283	-0,4	-1,74283	-0,4	-1,74283	-0,4	-1,74283	-0,4	-1,74283
45	-0,2	-0,72797	-0,2	-0,72797	-0,2	-0,72797	-0,2	-0,72797	-0,2	-0,72797	-0,2	-0,72797
46	-2,8E-16	-8,3E-16	-2,8E-16	-8,3E-16	-2,8E-16	-8,3E-16	-2,8E-16	-8,3E-16	-2,8E-16	-8,3E-16	-2,8E-16	-8,3E-16
47	0,2	0,48797	0,2	0,48797	0,2	0,48797	0,2	0,48797	0,2	0,48797	0,2	0,48797
48	0,4	0,783105	0,4	0,783105	0,4	0,783105	0,4	0,7831	0,4	0,783105	0,4	0,783105
49	0,6	0,929647	0,6	0,929647	0,6	0,929647	0,6	0,92965	0,6	0,929647	0,6	0,929647
50	0,8	0,966987	0,8	0,966987	0,8	0,966987	0,8	0,96699	0,8	0,966987	0,8	0,966987
51	1	0,92868	1	0,92868	1	0,92868	1	0,92868	1	0,92868	1	0,92868
52	1,2	0,842174	1,2	0,842174	1,2	0,842174	1,2	0,84217	1,2	0,842174	1,2	0,842174
53	1,4	0,729027	1,4	0,729027	1,4	0,729027	1,4	0,72903	1,4	0,729027	1,4	0,729027
54	1,6	0,605431	1,6	0,605431	1,6	0,605431	1,6	0,60543	1,6	0,605431	1,6	0,605431
55	1,8	0,482928	1,8	0,482928	1,8	0,482928	1,8	0,48293	1,8	0,482928	1,8	0,482928
56	2	0,36918	2	0,36918	2	0,36918	2	0,36918	2	0,36918	2	0,36918

```

Do
  If OptY Then f = Y(x)
  If OptG Then f = g(x)
  If OptZ Then f = z(x)
  Cells(i, 1) = x
  Cells(i, 2) = f
  x = x + dx: i = i + 1
Loop Until x > xk

Do
  If OptY Then f = Y(x)
  If OptG Then f = g(x)
  If OptZ Then f = z(x)
  Cells(i, 4) = x
  Cells(i, 5) = f
  x = x + dx: i = i + 1
Loop While x <= xk

Do While x <= xk
  If OptY Then f = Y(x)
  If OptG Then f = g(x)
  If OptZ Then f = z(x)
  Cells(i, 7) = x
  Cells(i, 8) = f
  x = x + dx: i = i + 1
Loop

Do Until x > xk
  If OptY Then f = Y(x)
  If OptG Then f = g(x)
  If OptZ Then f = z(x)
  Cells(i, 10) = x
  Cells(i, 11) = f
  x = x + dx: i = i + 1
Loop

While x <= xk
  If OptY Then f = Y(x)
  If OptG Then f = g(x)
  If OptZ Then f = z(x)
  Cells(i, 13) = x
  Cells(i, 14) = f
  x = x + dx: i = i + 1
Wend

Dim n As Integer, k As Integer
n = (xk - xn) / dx
For k = 1 To n + 1
  If OptY Then f = Y(x)
  If OptG Then f = g(x)
  If OptZ Then f = z(x)
  Cells(i, 16) = x
  Cells(i, 17) = f
  x = x + dx: i = i + 1
Next
  
```