

**ВЗАИМОДЕЙСТВИЕ  
MICROSOFT VISUAL BASIC  
С ПРОГРАММНЫМИ  
ПРОДУКТАМИ  
MICROSOFT OFFICE**

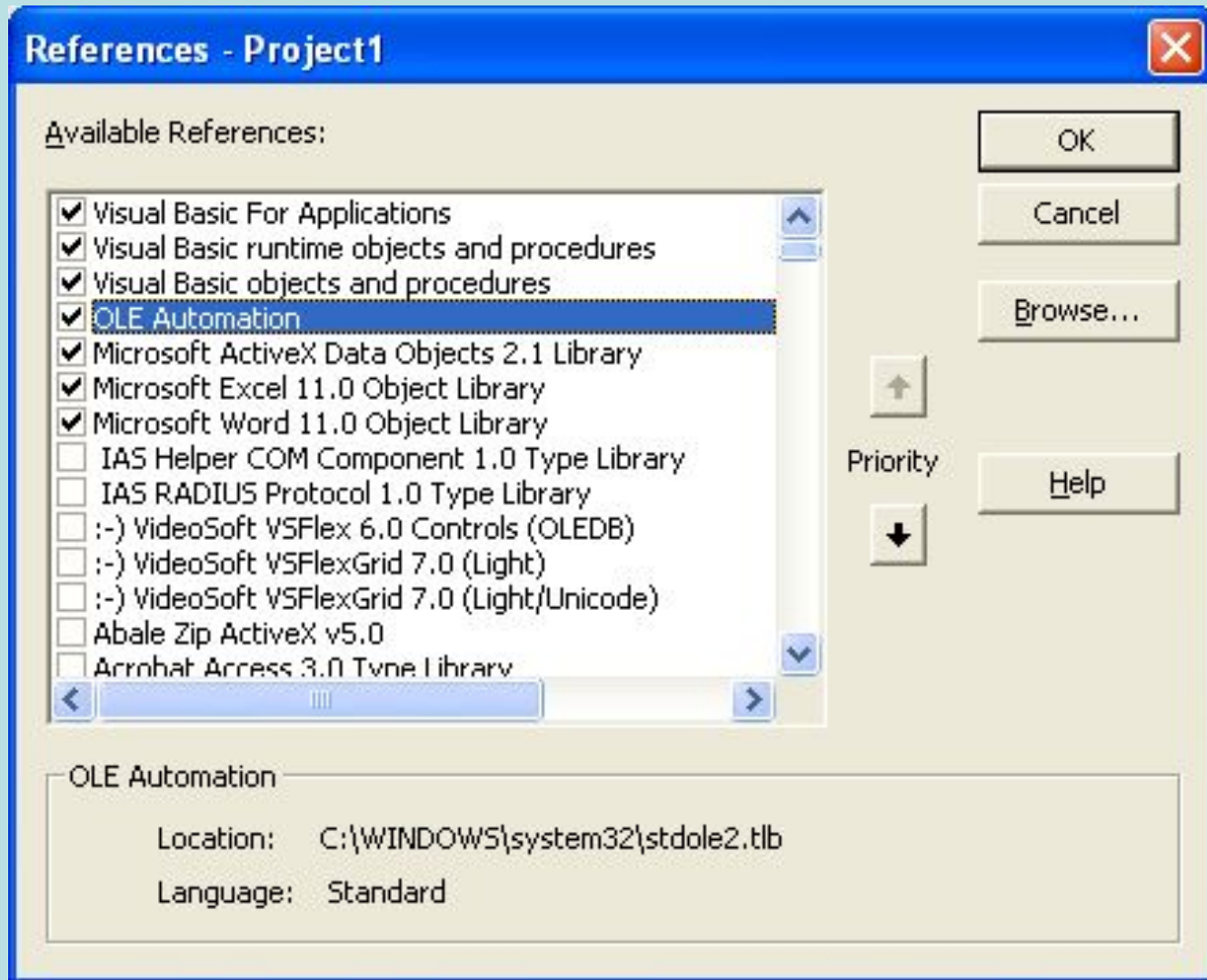
# ПОДКЛЮЧЕНИЕ БИБЛИОТЕК

The screenshot displays the Microsoft Visual Basic IDE interface. The main window is titled "Project1 - Microsoft Visual Basic [design]". The menu bar includes "File", "Edit", "View", "Project", "Format", "Debug", "Run", "Query", "Diagram", "Tools", "Add-Ins", "Window", and "Help". The "Project" menu is open, showing options such as "Add Form", "Add MDI Form", "Add Module", "Add Class Module", "Add User Control", "Add Property Page", "Add User Document", "Add WebClass", "Add Data Report", "Add DHTML Page", "Add Data Environment", "More ActiveX Designers...", "Add File...", "Remove Form1", "References...", "Components...", and "Project1 Properties...". The "References..." option is highlighted. In the background, a form window titled "Form1" is visible on a grid. On the right side, the "Project - Project1" window shows a tree view with "Project1 (Project1)" and "Forms" containing "Form1 (Form1)". Below it, the "Properties - Form1" window is open, showing a list of properties for "Form1 Form".

Form1 Form	
Alphabetic   Categorized	
(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	<input type="checkbox"/> &H8000000F
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	<input type="checkbox"/> &H00000000
FillStyle	1 - Transparent
Font	MS Sans Serif
FontTransparent	True
ForeColor	<input type="checkbox"/> &H80000012
HasDC	True
Height	5205
HelpContextID	0
Icon	(Icon)

**Caption**  
Returns/sets the text displayed in an object's title bar or below an object's

# ПОДКЛЮЧЕНИЕ БИБЛИОТЕК



# СОЗДАНИЕ ТАБЛИЦЫ В MS ACCESS

Microsoft Access

test : база данных (формат Access 2000)

Объекты

- Таблицы
- Запросы

Создание таблицы в режиме конструктора

Создание таблицы с помощью мастера

Создание таблицы путем ввода данных

Имя поля	Тип данных	Описание
ID_KLIENT	Счетчик	
FAM	Текстовый	
IMY	Текстовый	
OTCH	Текстовый	
GOROD	Текстовый	

Сохранение

Имя таблицы:

KLIENT

Общие Подстановка

Размер поля 50

Формат поля

Маска ввода

Подпись

Значение по умолчанию

Условие на значение

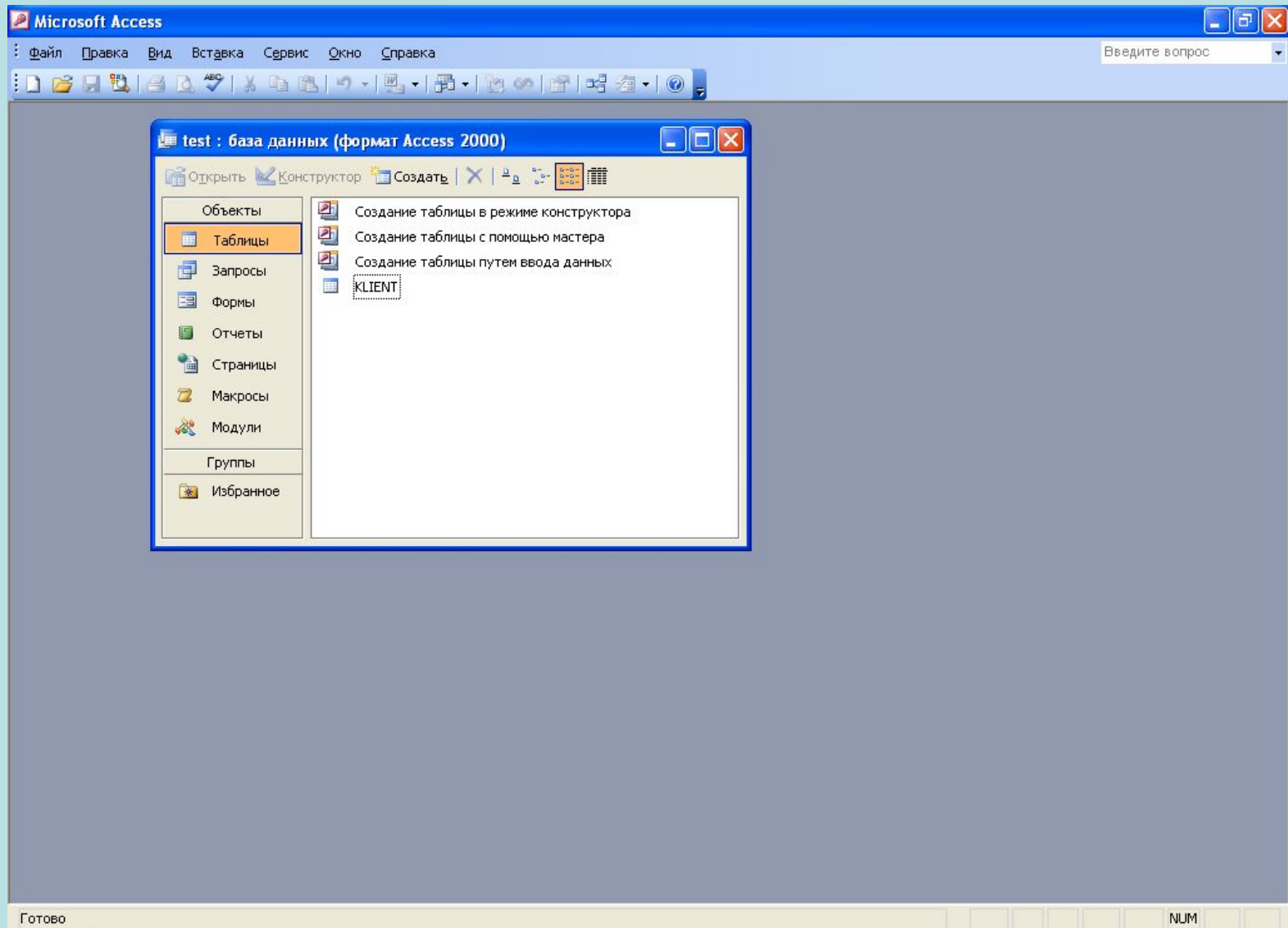
Сообщение об ошибке

Обязательное поле Нет

Имя поля может состоять из 64 знаков с учетом пробелов. Для справки по именам полей нажмите клавишу F1.

Конструктор. F6 = переключение окон. F1 = справка. NUM

# СОЗДАНИЕ ТАБЛИЦЫ В MS ACCESS





# ОТЧЕТ В MS WORD

The image shows a screenshot of the Microsoft Word application window. The title bar reads "Документ1 - Microsoft Word". The menu bar includes "Файл", "Правка", "Вид", "Вставка", "Формат", "Сервис", "Таблица", "Окно", and "Справка". The toolbar contains various icons for file operations, editing, and formatting. The status bar at the bottom shows "Стр. 1", "Разд 1", "1/1", "На 1,2см", "Ст 1", "Кол 1", "ЗАП", "ИСПР", "ВДЛ", "ЗАМ", "русский (Ро)", and a spelling checker icon.

In the center of the document, the word "ОТЧЕТ" is centered. Below it is a table with five columns and one row. The columns are labeled "№", "ФАМИЛИЯ", "ИМЯ", "ОТЧЕСТВО", and "ГОРОД".

№	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ГОРОД







# ПРОГРАММА



```
Dim cnData As ADODB.Connection
Dim rsData As ADODB.Recordset
Dim objExcel As Excel.Application
```

```
Private Sub Command1_Click()
```

```
Dim number As Long
```

```
Set objExcel = New Excel.Application
```

```
Set cnData = New ADODB.Connection
```

```
Set rsData = New ADODB.Recordset
```

```
cnData.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;" +  
    "Data Source=" + App.Path + "\test.mdb"
```

```
cnData.Open
```

```
rsData.Open ("KLIENT"), cnData, adOpenStatic, adLockPessimistic
```

```
objExcel.Workbooks.Open (App.Path + "\test.xls")
```

```
number = 1
```

```
With objExcel.ActiveSheet
```

```
Do While .Cells(number + 1, 1).Value <> ""
```

```
rsData.AddNew
```

```
rsData!FAM = .Cells(number + 1, 2).Value
```

```
rsData!IMY = .Cells(number + 1, 3).Value
```

```
rsData!OTCH = .Cells(number + 1, 4).Value
```

```
rsData!GOROD = .Cells(number + 1, 5).Value
```

```
rsData.Update
```

```
number = number + 1
Loop
End With
rsData.Close
Set rsData = Nothing
cnData.Close
Set cnData = Nothing
objExcel.Quit
Set objExcel = Nothing
MsgBox "Загрузка данных завершена!", vbInformation + vbOKOnly,
      "ВНИМАНИЕ!"
End Sub
```

```
Private Sub Command2_Click()  
Dim wd As Word.Application  
Dim wddoc As Word.Document  
Dim sSQL As String  
Dim LTotalRecords As Long, i As Long  
Set cnData = New ADODB.Connection  
Set rsData = New ADODB.Recordset  
Set wd = New Word.Application  
wd.Visible = True  
Set wddoc = wd.Documents.Add(App.Path + "\test.dot")  
cnData.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;" + "Data  
Source=" + App.Path + "\test.mdb"  
cnData.Open  
sSQL = "SELECT FAM, IMY, OTCH, GOROD "  
sSQL = sSQL + "From KLIENT"  
rsData.Open (sSQL), cnData, adOpenStatic, adLockPessimistic  
If rsData.BOF Or rsData.EOF Then  
MsgBox "ТАБЛИЦА НЕ СОДЕРЖИТ ЗАПИСЕЙ", vbExclamation +  
vbOKOnly, "ВНИМАНИЕ!"  
rsData.Close  
Set rsData = Nothing  
cnData.Close  
Set cnData = Nothing
```

```
Exit Sub
End If
rsData.MoveLast
LTotalRecords = rsData.RecordCount
rsData.MoveFirst
z = 0
For i = 1 To LTotalRecords - 1
z = z + 1
wddoc.Tables(1).Rows.Add
wddoc.Tables(1).Cell(z + 1, 1).Range.Text = CStr(z)
wddoc.Tables(1).Cell(z + 1, 2).Range.Text = rsData!FAM
wddoc.Tables(1).Cell(z + 1, 3).Range.Text = rsData!IMY
wddoc.Tables(1).Cell(z + 1, 4).Range.Text = rsData!OTCH
wddoc.Tables(1).Cell(z + 1, 5).Range.Text = rsData!GOROD
rsData.MoveNext
Next i
z = z + 1
wddoc.Tables(1).Rows.Add
wddoc.Tables(1).Cell(z + 1, 1).Range.Text = CStr(z)
wddoc.Tables(1).Cell(z + 1, 2).Range.Text = rsData!FAM
wddoc.Tables(1).Cell(z + 1, 3).Range.Text = rsData!IMY
wddoc.Tables(1).Cell(z + 1, 4).Range.Text = rsData!OTCH
```

```
wddoc.Tables(1).Cell(z + 1, 5).Range.Text = rsData!GOROD
rsData.Close
Set rsData = Nothing
cnData.Close
Set cnData = Nothing
'.PrintOut
wddoc.SaveAs (App.Path + "\test2.doc")
wddoc.Close
wd.Quit
Set wd = Nothing
End Sub
```







# МОДЕЛИРОВАНИЕ РЕАКТОРА ИДЕАЛЬНОГО ВЫТЕСНЕНИЯ В РЕАЛЬНОМ РЕЖИМЕ ВРЕМЕНИ (ПРОГРАММНЫЙ КОД)

**Const k1 = 0.8**

**Const k2 = 0.7**

**Dim ca As Single, cb As Single, cd As Single, ce As Single, tau As Single**

**Dim dt As Single, k As Integer, m As Integer**

**Dim ca1 As Single, cb1 As Single, cd1 As Single, ce1 As Single**

**Private Sub Command1\_Click()**

**Timer1.Interval = 250**

**Timer1.Enabled = True**

**End Sub**

**Private Sub Command2\_Click()**

**Timer1.Enabled = False**

**End Sub**

**Private Sub Command3\_Click()**

**Timer1.Enabled = True**

**End Sub**

# МОДЕЛИРОВАНИЕ РЕАКТОРА ИДЕАЛЬНОГО ВЫТЕСНЕНИЯ В РЕАЛЬНОМ РЕЖИМЕ ВРЕМЕНИ (ПРОГРАММНЫЙ КОД)

```
Private Sub Command4_Click()  
Dim i As Integer, j As Integer  
Timer1.Enabled = False  
Picture1.Cls  
For i = 1 To 11  
MSFlexGrid1.Row = i  
For j = 0 To 4  
MSFlexGrid1.Col = j  
MSFlexGrid1.Text = ""  
Next j  
Next i  
tau = 0: k = 0: m = 0  
dt = 0.1: ca = 0.8  
cb = 0.5: cd = 0: ce = 0  
ca1 = ca: cb1 = ca: cd1 = cd: ce1 = ce  
Label6.Caption = CStr(ca)  
Label7.Caption = CStr(cb)  
Label8.Caption = CStr(cd)  
Label9.Caption = CStr(ce)  
Label11.Caption = CStr(dt)  
Picture1.PSet (m, Picture1.Height - Int(ca * Picture1.Height)), RGB(255, 0, 0)  
Picture1.PSet (m, Picture1.Height - Int(cb * Picture1.Height)), RGB(0, 255, 0)  
Picture1.PSet (m, Picture1.Height - Int(cd * Picture1.Height)), RGB(0, 0, 255)  
Picture1.PSet (m, Picture1.Height - Int(ce * Picture1.Height)), RGB(255, 255, 0)  
End Sub
```

```
Private Sub Command5_Click()  
End  
End Sub
```

```
Private Sub Form_Load()  
MSFlexGrid1.Row = 0  
MSFlexGrid1.Col = 0  
MSFlexGrid1.Text = "Tau"  
MSFlexGrid1.Col = 1  
MSFlexGrid1.Text = "CA"  
MSFlexGrid1.Col = 2  
MSFlexGrid1.Text = "CB"  
MSFlexGrid1.Col = 3  
MSFlexGrid1.Text = "CD"  
MSFlexGrid1.Col = 4  
MSFlexGrid1.Text = "CE"  
tau = 0: k = 0: m = 0: dt = 0.1: ca = 0.8  
cb = 0.5: cd = 0: ce = 0  
ca1 = ca: cb1 = cb: cd1 = cd: ce1 = ce  
Label6.Caption = CStr(ca)  
Label7.Caption = CStr(cb)  
Label8.Caption = CStr(cd)  
Label9.Caption = CStr(ce)  
Label11.Caption = CStr(dt)  
Picture1.PSet (m, Picture1.Height - Int(ca * Picture1.Height)), RGB(255, 0, 0)  
Picture1.PSet (m, Picture1.Height - Int(cb * Picture1.Height)), RGB(0, 255, 0)  
Picture1.PSet (m, Picture1.Height - Int(cd * Picture1.Height)), RGB(0, 0, 255)  
Picture1.PSet (m, Picture1.Height - Int(ce * Picture1.Height)), RGB(255, 255, 0)  
End Sub
```

```
Private Sub HScroll5_Change()  
'dt  
Label11.Caption = CStr(HScroll5.Value / 100)  
End Sub
```

```
Private Sub HScroll5_Scroll()  
HScroll5_Change  
End Sub
```

```
Private Sub HScroll4_Change()  
'ce  
Label9.Caption = CStr(HScroll4.Value / 10000)  
End Sub
```

```
Private Sub HScroll4_Scroll()  
HScroll4_Change  
End Sub
```

```
Private Sub HScroll3_Change()  
'cd  
Label8.Caption = CStr(HScroll3.Value / 10000)  
End Sub
```



```
Private Sub HScroll3_Scroll()  
HScroll3_Change  
End Sub  
Private Sub HScroll2_Change()  
'cb  
Label7.Caption = CStr(HScroll2.Value / 10000)  
End Sub
```

```
Private Sub HScroll2_Scroll()  
HScroll2_Change  
End Sub  
Private Sub HScroll1_Change()  
'ca  
Label6.Caption = CStr(HScroll1.Value / 10000)  
End Sub
```

```
Private Sub HScroll1_Scroll()  
HScroll1_Change  
End Sub
```

```
Private Sub Timer1_Timer()  
Dim i As Integer, j As Integer  
Picture1.AutoRedraw = True  
k = k + 1: m = m + 1  
If m * 100 > Picture1.Width Then  
m = 1  
Picture1.Cls  
End If  
dt = CSng(Label11.Caption)  
ca = CSng(Label6.Caption)  
cb = CSng(Label7.Caption)  
cd = CSng(Label8.Caption)  
ce = CSng(Label9.Caption)  
tau = tau + dt  
MSFlexGrid1.Row = k  
MSFlexGrid1.Col = 0  
MSFlexGrid1.Text = CStr(Int(tau * 1000) / 1000)  
ca = ca - dt * (k1 * ca * cb)  
Picture1.Line ((m - 1) * 100, Picture1.Height - Int(ca1 * Picture1.Height))-(m * _  
100, Picture1.Height - Int(ca * Picture1.Height)), RGB(255, 0, 0)  
MSFlexGrid1.Col = 1  
MSFlexGrid1.Text = CStr(Int(ca * 10000) / 10000)
```

```
Label6.Caption = CStr(ca)
cb = cb - dt * (k1 * ca * cb + k2 * cd * cb)
Picture1.Line ((m - 1) * 100, Picture1.Height - Int(cb1 * Picture1.Height))-(m * _
    100, Picture1.Height - Int(cb * Picture1.Height)), RGB(0, 255, 0)
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = CStr(Int(cb * 10000) / 10000)
Label7.Caption = CStr(cb)
cd = cd + dt * (k1 * ca * cb - k2 * cd * cb)
Picture1.Line ((m - 1) * 100, Picture1.Height - Int(cd1 * Picture1.Height))-(m * _
    100, Picture1.Height - Int(cd * Picture1.Height)), RGB(0, 0, 255)
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = CStr(Int(10000 * cd) / 10000)
Label8.Caption = CStr(cd)
ce = ce + dt * (k2 * cd * cb)
Picture1.Line ((m - 1) * 100, Picture1.Height - Int(ce1 * Picture1.Height))-(m * _
    100, Picture1.Height - Int(ce * Picture1.Height)), RGB(255, 255, 0)
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = CStr(Int(10000 * ce) / 10000)
Label9.Caption = CStr(ce)
```

```
HScroll1.Value = Int(10000 * ca)
HScroll2.Value = Int(10000 * cb)
HScroll3.Value = Int(10000 * cd)
HScroll4.Value = Int(10000 * ce)
HScroll5.Value = Int(100 * dt)
ca1 = ca: cb1 = cb: cd1 = cd: ce1 = ce
If ca = 0 Or cb = 0 Then
Timer1.Enabled = False
End If
If k > 10 Then
k = 0
For i = 1 To 11
MSFlexGrid1.Row = i
For j = 0 To 4
MSFlexGrid1.Col = j
MSFlexGrid1.Text = ""
Next j
Next i
End If
End Sub
```

## **РАБОТА С ФОРМАМИ. СОБЫТИЯ ФОРМ**

**Формы в Visual Basic являются каркасом пользовательского интерфейса приложения. Под приложением будем понимать программу, работающую в среде Windows.**

**Под интерфейсом будем понимать внешнюю оболочку приложения вместе с программами управления доступом и другими скрытыми для пользователя механизмами управления, дающую возможность работать с документами, данными и другой информацией, хранящейся в компьютере или за его пределами.**

# ДОБАВЛЕНИЕ ФОРМЫ В ПРОЕКТ

## ПЕРВЫЙ СПОСОБ:

1) Выполнить последовательность команд главного меню:

**Project ► Add Form.**

2) В появившемся меню перейти на вкладку **New**, где дважды щелкнуть левой кнопкой мышки по значку **Form**.

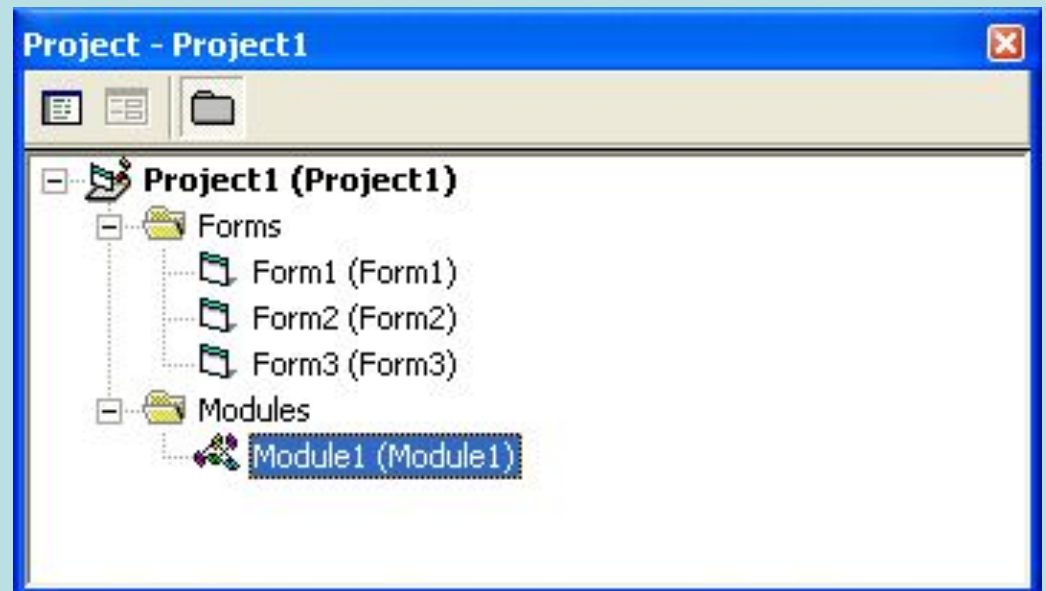
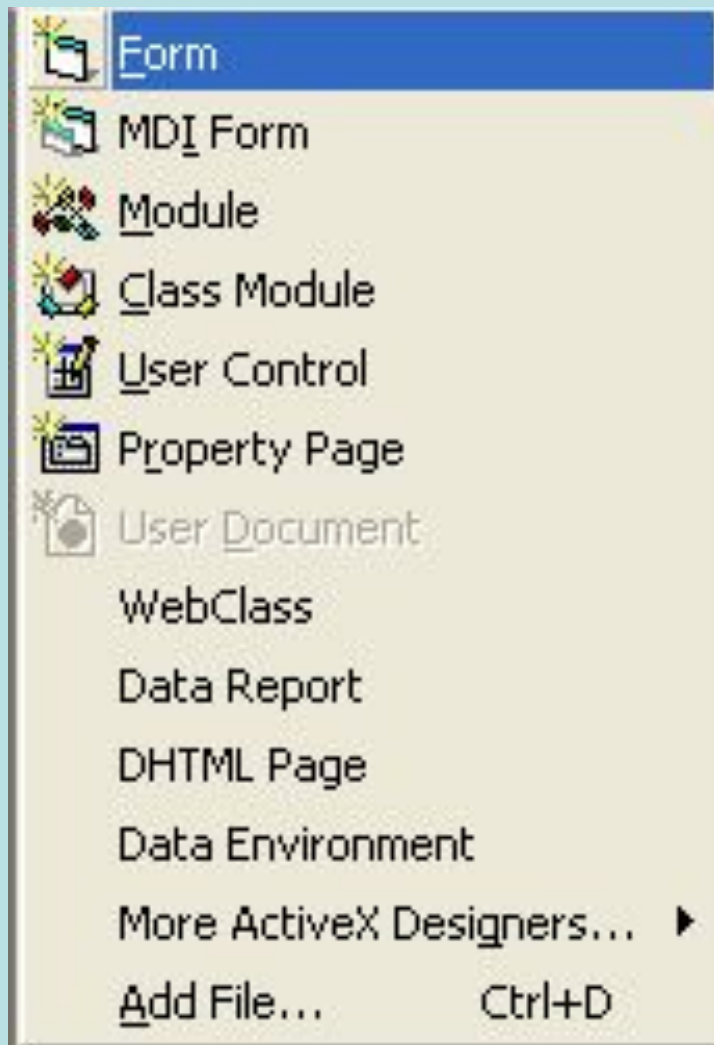


## ВТОРОЙ СПОСОБ:

- Нажать на стрелку направленную вниз, расположенную справа от кнопки.



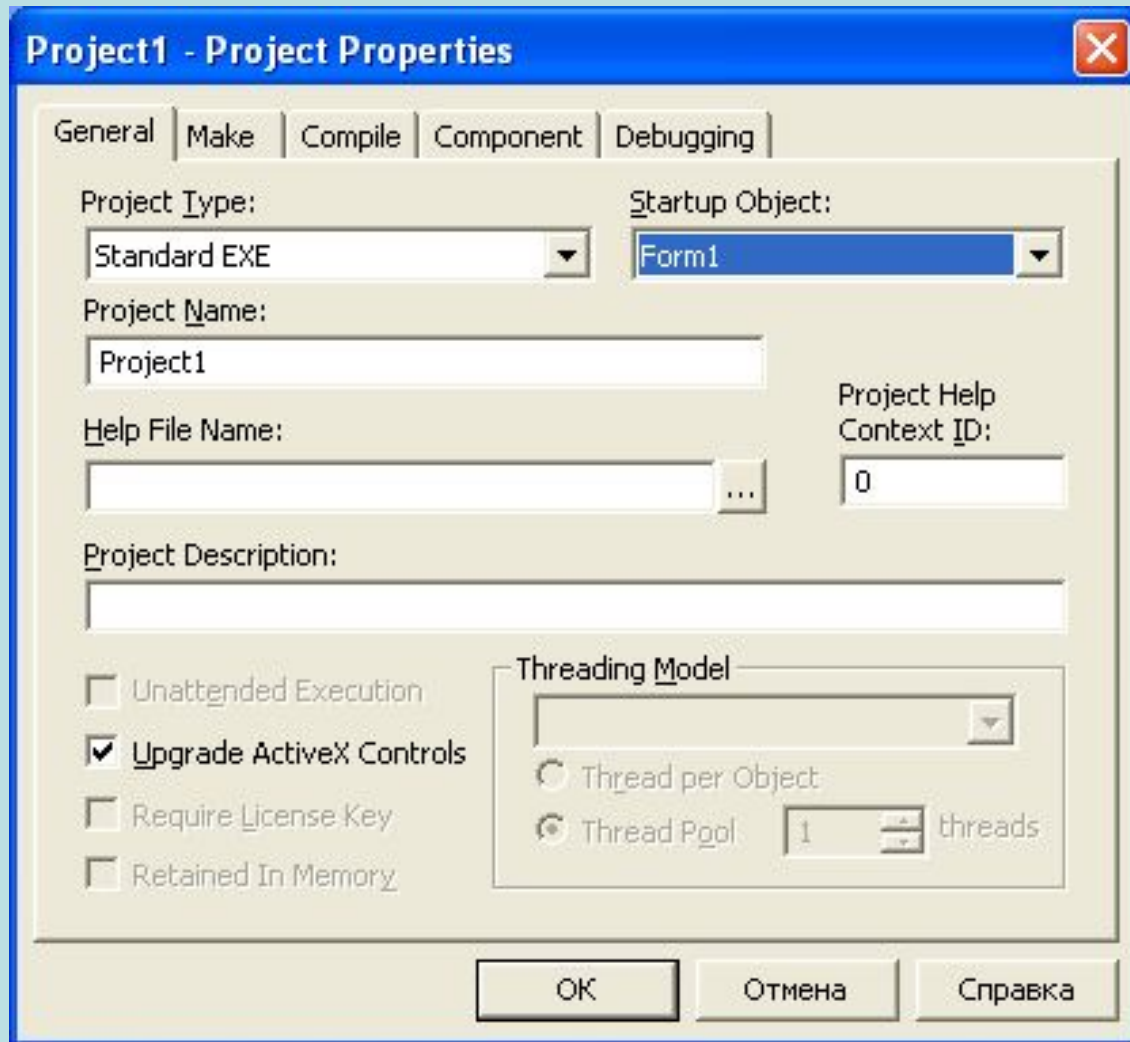
- В появившемся выпадающем списке выбрать пункт **Form**, затем нажать на клавиатуре клавишу **Enter**.



# СМЕНА СТАРТОВОЙ ФОРМЫ

- Выполнить последовательность команд меню: **Project ► Project Properties.**
- В появившемся окне перейти на вкладку **General** .
- В выпадающем списке Startup Object необходимо выбрать имя формы, которую предполагается сделать стартовой.
- Нажать на кнопку **Ok.**

# СМЕНА СТАРТОВОЙ ФОРМЫ



# СОБЫТИЯ, СВЯЗАННЫЕ С ФОРМАМИ

Системные события, связанные с формой выполняются в следующем порядке:

- Initialize
- Load
- Activate
- Deactivate
- QueryUnload
- Unload
- Terminate.

# Событие Initialize

Обычно, данное событие используется для подготовки приложения к работе. В обработчике данного события (в процедуре обработки данного события) присваиваются переменным начальные значения, расставляются на форме элементы управления, задаются их свойства.

```
Private Sub Form_Initialize()
```

```
.....
```

```
End Sub
```

Событие Initialize возникает в момент создания экземпляра формы (до ее загрузки в память компьютера и отображения на экране). Это событие генерируется только один раз в течение всего сеанса работы приложения. Для того, чтобы еще раз вызвать событие Initialize, потребуется завершить программу и перезапустить приложение. Таким образом, если какой-то код необходимо выполнить в приложении более чем один раз, то он должен быть помещен в обработчик другого события.

**Событие Initialize генерируется в методах:**

***Form1.Show u Load Form1***

**Также, событие Initialize возникает, когда возвращается одно из свойств формы, или вызывается метод, определенный в модуле (процедуре или функции) формы. Например, в форме Form2 определена процедура общего вида Proc1. При вызове этой процедуры из модуля другой формы Form2.Proc1, создается экземпляр формы Form2, что приводит к генерации события Initialize. Переменные уровня формы при инициализации существуют, пока выполняется приложение, даже если форма выгружена из памяти. При повторном запуске формы событие Initialize не генерируется.**

# Событие Load

Событие Load используется для выполнения каких-либо действий перед выводом формы на экран. Оно позволяет присвоить исходные значения свойствам формы и ее элементов управления. Данное событие происходит при каждой загрузке формы в память. При первой загрузке событие Load следует за событием Initialize.

Событие Load генерируется в результате применения оператора Load или метода Show, также может генерироваться после ссылки на свойства, методы или элементы управления незагруженной формы.

Пример.

```
Private Sub Form_Load()  
List1.Clear  
List1.AddItem ("Элемент1")  
List1.AddItem ("Элемент2")  
List1.AddItem ("Элемент3")  
End Sub
```



# События Activate / Deactivate

Если в приложении предполагается переключение между формами, то для определения поведения этих форм могут быть использованы события **Activate** и **Deactivate**.

Событие **Activate** возникает, когда фокус ввода переходит на данную форму от другой формы того же приложения. При этом форма должна быть видима. Например, форма, загруженная оператором **Load**, остается невидимой, пока к ней не применен метод **Show** или не установлено свойство формы **Visible** в положение **True**. Событие **Activate** генерируется до события **GotFocus**.

**Событие Deactivate происходит, когда ввода переходит с данной формы на другую форму того же приложения. Оно генерируется после события LostFocus.**

**События Activate и Deactivate генерируются только в том случае, если фокус ввода переходит между формами одного и того же приложения. Если пользователь переходит в другое приложение, а потом вернется в программу на Visual Basic, то ни одно из этих событий не происходит.**

# Событие QueryUnload

Данное событие может быть использовано, если требуется узнать, как именно пользователь закрывает форму. Оно генерируется перед событием Unload. Аргумент процедуры обработки данного события UnloadMode сообщает, как закрывается форма и при необходимости позволяет аннулировать это действие.

## **Событие QueryUnload возникает:**

- **когда пользователь выбирает из системного меню формы команду Close**
- **когда в программе выполняется оператор Unload**
- **когда текущий сеанс работы в операционной системе Windows завершается**
- **когда Диспетчер задач операционной системы Windows закрывает данное приложение**
- **когда закрывается дочерняя MDI-форма, так как закрывается основная MDI-форма.**

**Чтобы предотвратить выгрузку формы,  
необходимо аргументу Cancel присвоить  
значение True.**

**Пример.**

```
Private Sub Form_QueryUnload(Cancel as  
Integer, UnloadMode As Integer)
```

```
If UnloadMode <> vbFormCode then
```

```
MsgBox “Используйте для закрытия формы  
кнопку Выход”
```

```
Cancel = True
```

```
End If
```

```
End Sub
```

# Событие Unload

**Данное событие генерируется перед событием Terminate.**

**Обработчик события Unload может быть использован для проверки того, надо ли выгрузить форму, или для определения операций, выполняемых при выгрузке формы. Сюда можно включить программный код проверки введенных значений на уровне формы, код для сохранения данных в файле.**

**При обработке события Unload можно добавить оператор End – он гарантирует выгрузку всех форм до завершения программы.**

**Присвоение аргументу Cancel любого ненулевого значения предотвращает удаление формы, но не запрещает другие события, вроде выхода из среды Windows. Чтобы не допустить выхода из Windows нужно использовать событие QueryUnload.**

# Событие Terminate

Это событие генерируется, когда из памяти удаляются все ссылки на экземпляр формы. Чтобы убрать из памяти переменные этой формы и освободить занимаемые системные ресурсы, объектной переменной формы нужно присвоить значение `Nothing`.

**Set Form2=Nothing**

Для всех объектов, кроме классов, событие `Terminate` генерируется после события `Unload`.

Но, оно не возникает, если экземпляр формы или класса удаляется из памяти вследствие некорректного завершения программы. Если Ваша программа рухнет перед удалением из памяти всех существующих экземпляров класса или формы, то событие `Terminate` не произойдет.