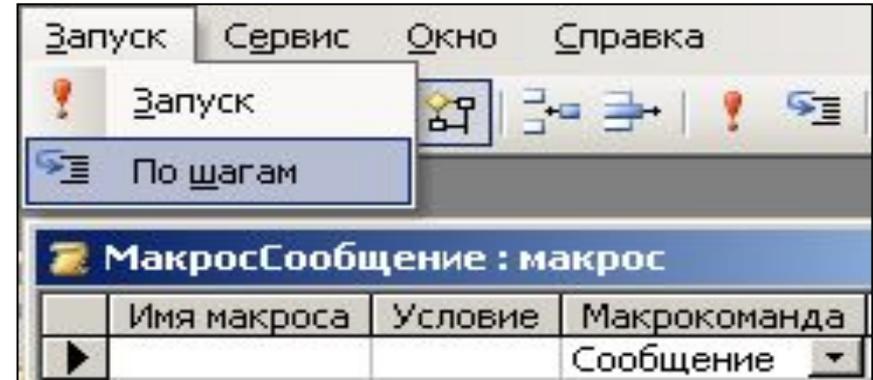
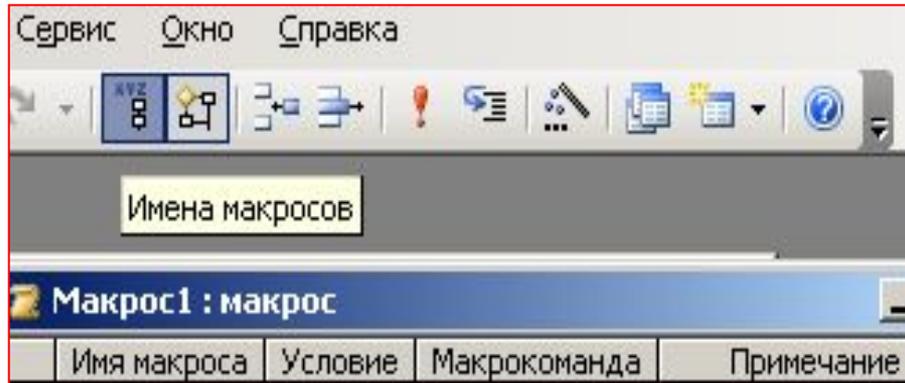


# **8\_ Объектно-ориентированное программирование БД**

# Макросы

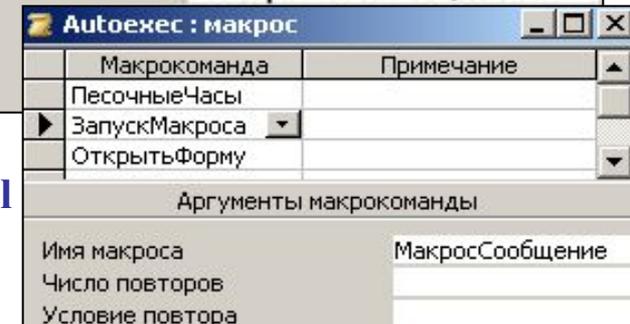
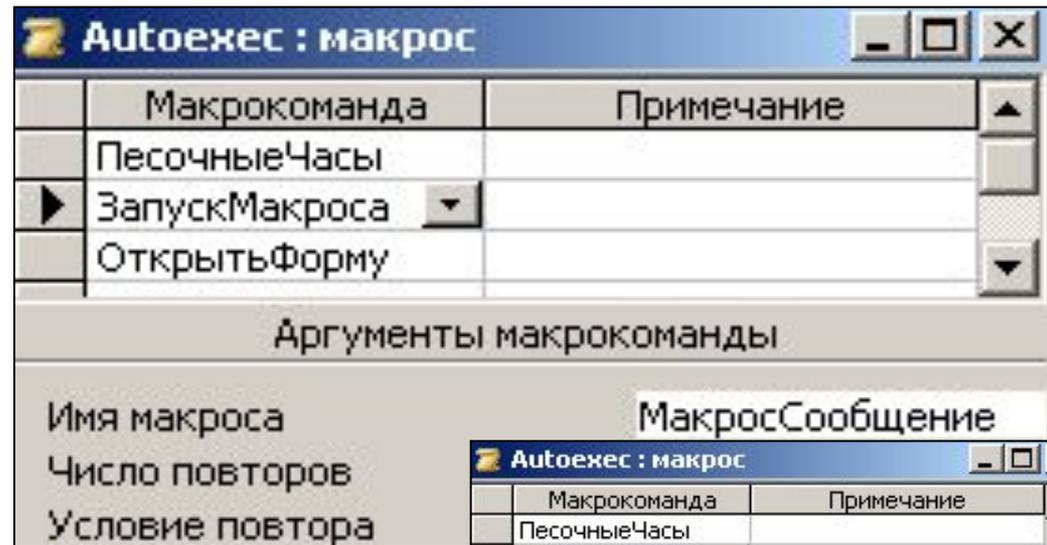
Макросы содержат понятные названия макрокоманд. Окно макроса изначально содержит поля для выбора макрокоманд и примечания, но может быть дополнено полями имени предопределенного макроса и условия для его выполнения. Покомандное выполнение макроса задается в режиме конструктора по команде *Запуск – По шагам*. Составим *МакросСообщение*, выводящий сообщение на экран.



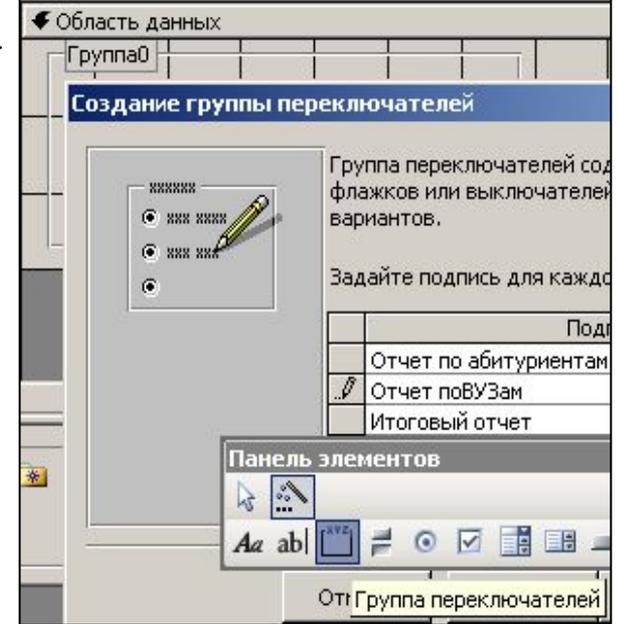
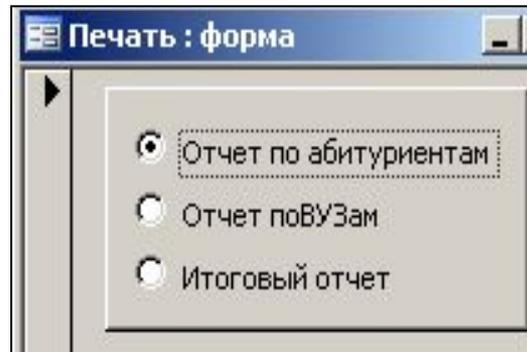
Сложный макрос с созданным ранее макросом в качестве одной из макрокоманд. (макрос с именем *AutoExec* автоматически запускается в момент открытия БД).

Код макроса можно преобразовать в код VB по команде *Сервис – Макрос – Преобразовать макросы*:

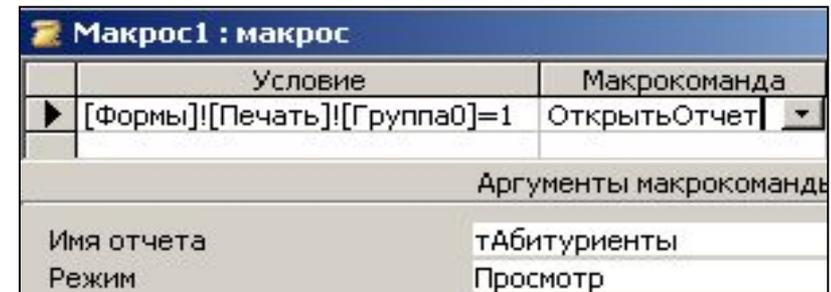
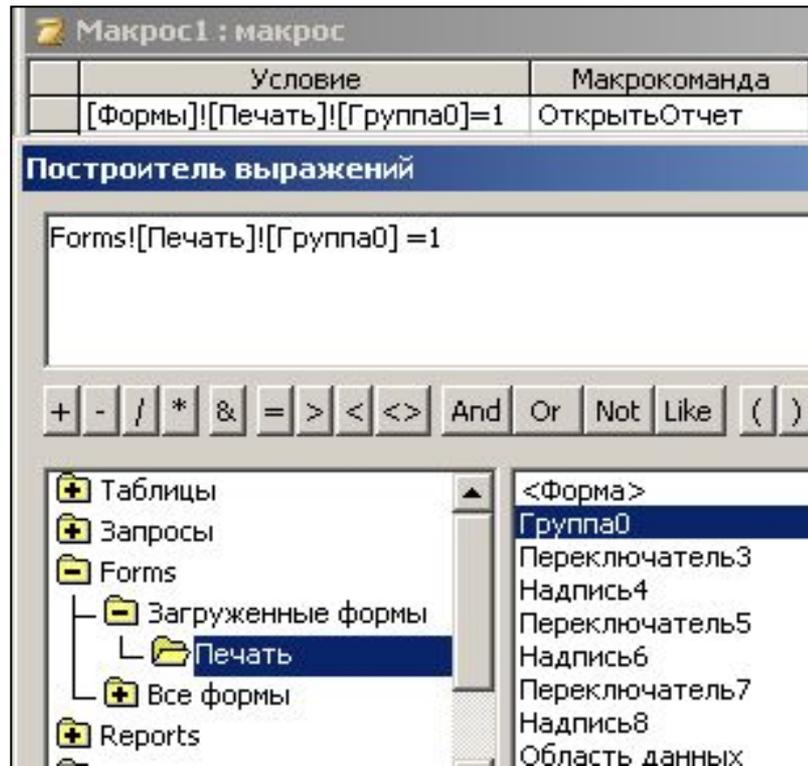
```
Function Autoexec()  
DoCmd.Hourglass True  
DoCmd.RunMacro "МакросСообщение", , ""  
DoCmd.OpenForm "тАбитуриенты", acNormal, "", "", , acNormal  
End Function
```



3. Пример использования условий при выполнении команд макроса. Разработаем форму для формирования отдельных отчетов. Форма содержит группу переключателей

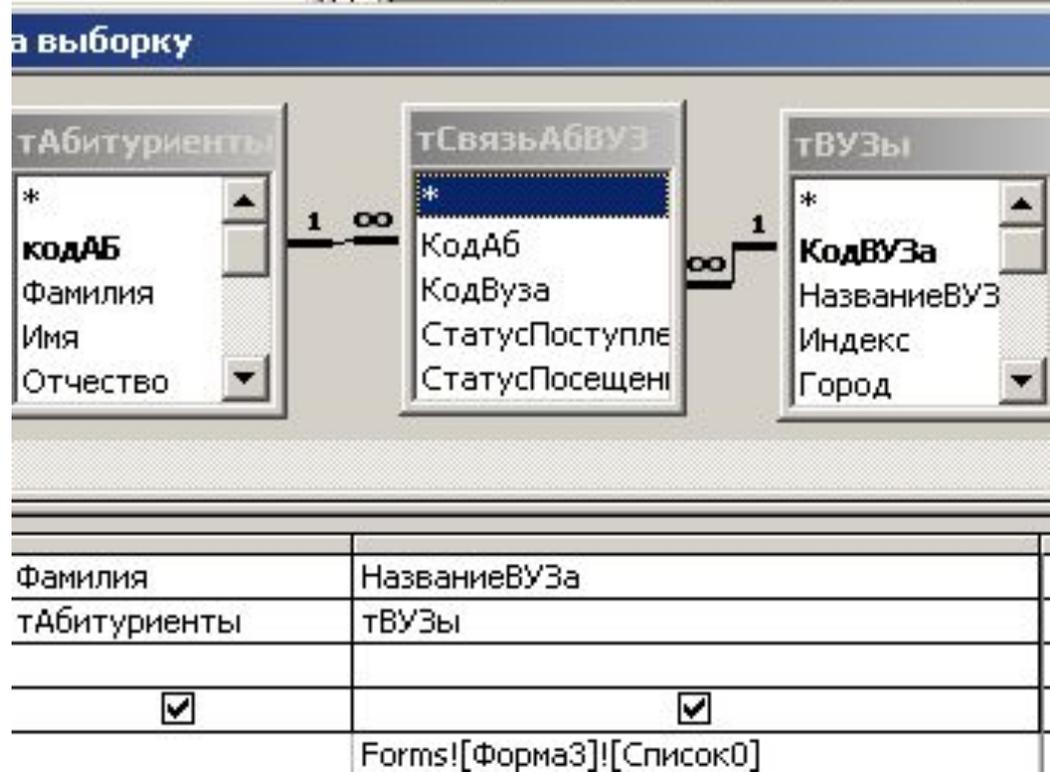
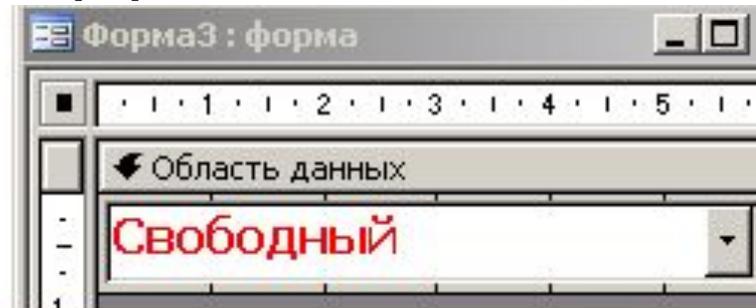
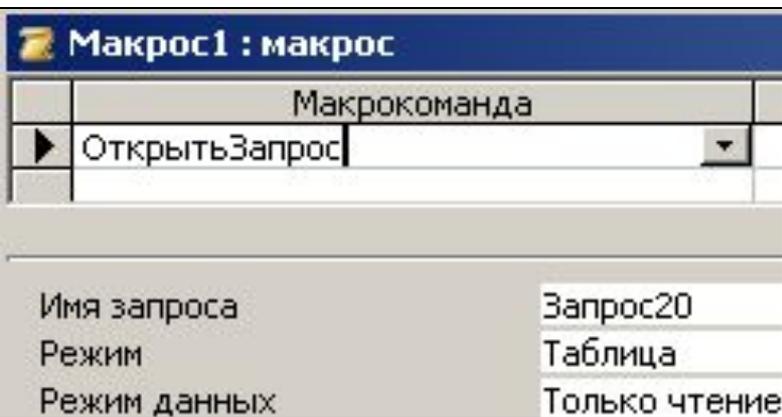


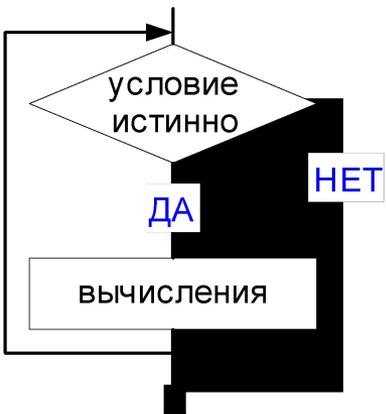
4. Запустим форму Печать и построим условия для выбора отчета с помощью нового макроса



# Свободная форма

Запрос с параметром предполагает точный ввод значений параметра. Выбор этих значений из списка возможных – эффективнее. Создадим пустую форму и поместим на нее поле со списком, указав в качестве источника поле НазваниеВУЗа из таблицы тВУЗы. Не закрывая форму, создадим запрос, указав на выбор названия ВУЗа из значения поля формы. Создадим макрос, открывающий запрос на событии ПОСЛЕ ОБНОВЛЕНИЯ поля формы.





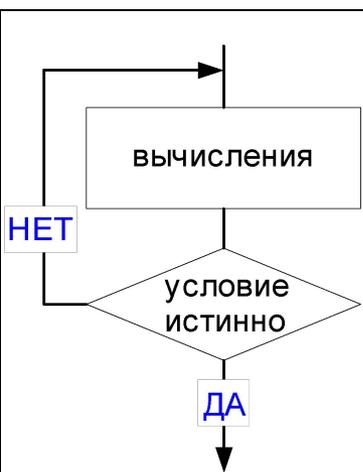
**DO WHILE** *условие истинно*

....

**LOOP**

**ЦИКЛ С ПРЕДУСЛОВИЕМ**

*Выполнять, пока условие истинно*



**DO**

....

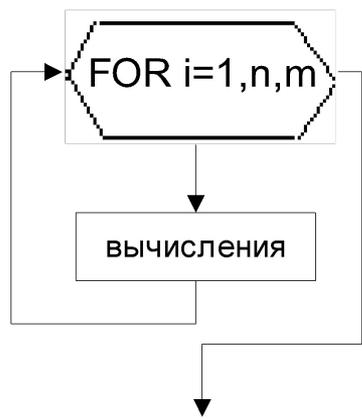
**LOOP WHILE** *условие истинно*

**ИЛИ**

**LOOP UNTIL** *условие не станет ложно*

**ЦИКЛ С ПОСТУСЛОВИЕМ**

*Выполнять, пока условие истинно*



**FOR**  $i = 1$  **TO**  $n$  **STEP**  $m$

....

**NEXT**

**ЦИКЛ С ОПРЕДЕЛЕННЫМ ЗНАЧЕНИЕМ ПАРАМЕТРА**

*Выполнять, пока параметр цикла  $i$  не превысит  $n$*

## Операторы условной обработки

<b>IF ...THEN...</b>	If rst.RecordCount <> 0 Then rst.MoveLast
<b>IF ... THEN ... ELSE ... END IF</b>	IF Forms![Счет]![P0] <> "" THEN Forms![Счет]![P0] = Forms![Счет]![P0] + vbCrLf + me.Имя ELSE Forms![Счет]![P0] = me.Наименование END IF

### Вложенный условный оператор

```
IF (a <> 0 Or a <> Null) THEN  
    IF b>0 THEN  
        b=sqrt(b)  
    ELSE  
        b=sqrt(abs(b))  
    END IF  
END IF
```

### Конструкция WITH...END WITH

```
WITH my_recordset  
    .Edit  
    ![Выбор] = False  
    .Update  
    .MoveNext  
END WITH
```

Конструкция WITH полезна при многократном обращении к форме по ее полному имени **Forms!Абитуриенты!Год\_рождения.Visible** или

**WITH Forms!Абитуриенты**

**!Год\_рождения.Visible=FALSE**

**!Адрес.Visible=TRUE**

**END WITH**

# Оператор выбора SELECT и безусловного перехода GOTO

<pre><b>Select Case</b> <b>переменная</b>   <b>Case</b> <i>значение1</i> или <i>условие1</i>:     <u><b>вычисления1</b></u>   <b>Case</b> <i>значение2</i> или <i>условие2</i>:     <u><b>вычисления2</b></u>   <b>Case Else</b>     <u><b>вычисления</b></u> <b>End Select</b></pre>	<p><i>Вместо вложенного условного оператора</i></p> <pre><b>IF</b> <b>переменная</b> = <i>значение1</i> <b>THEN</b> <u><b>вычисл1</b></u> <b>ELSE</b>   <b>IF</b> <b>переменная</b> = <i>значение2</i> <b>THEN</b> <u><b>вычисл2</b></u>   <b>ELSE</b>     <b>IF...THEN...</b>     <b>ELSE...</b>   <b>END IF</b> <b>END IF</b> <b>END IF</b></pre>
---	---

Оператор безусловного перехода **GOTO метка** используется для изменения последовательности выполнения операторов, например, принудительного завершения цикла

```
1:
у = InputBox("Ваш возраст ?", "Окно ввода")
```

```
Select Case у
  Case Is <= 0: MsgBox ("Ошибка")
    GoTo 1
  Case Is < 55: i = MsgBox("Ваш возраст " & у & ". Вам рано на пенсию")
  Case Is >= 55: i = MsgBox("Ваш возраст " & у & ". Вы на пенсии")
End Select
```

# Вызов процедуры

```
Sub main()
```

```
  a = InputBox("Введите коэфф. a")
```

```
  b = InputBox("Введите коэфф. b")
```

```
  c = InputBox("Введите коэфф. c")
```

```
  Do While a <> 0
```

```
    Call kv_kor (a, b, c) 'вызов процедуры
```

```
  Loop
```

```
End Sub
```

```
Sub kv_kor (a, b, c) ' процедура
```

```
  d = b * b - 4 * a * c формируется
```

```
  If d < 0 Then в составе модуля
```

```
    MsgBox ("Нет решения")
```

```
    GoTo 1
```

```
  Else
```

```
    x1 = (-b + Sqr(d)) / (2 * a)
```

```
    x2 = (-b - Sqr(d)) / (2 * a)
```

```
    MsgBox ("x1=" & x1 & ", x2=" & X2)
```

```
  End If
```

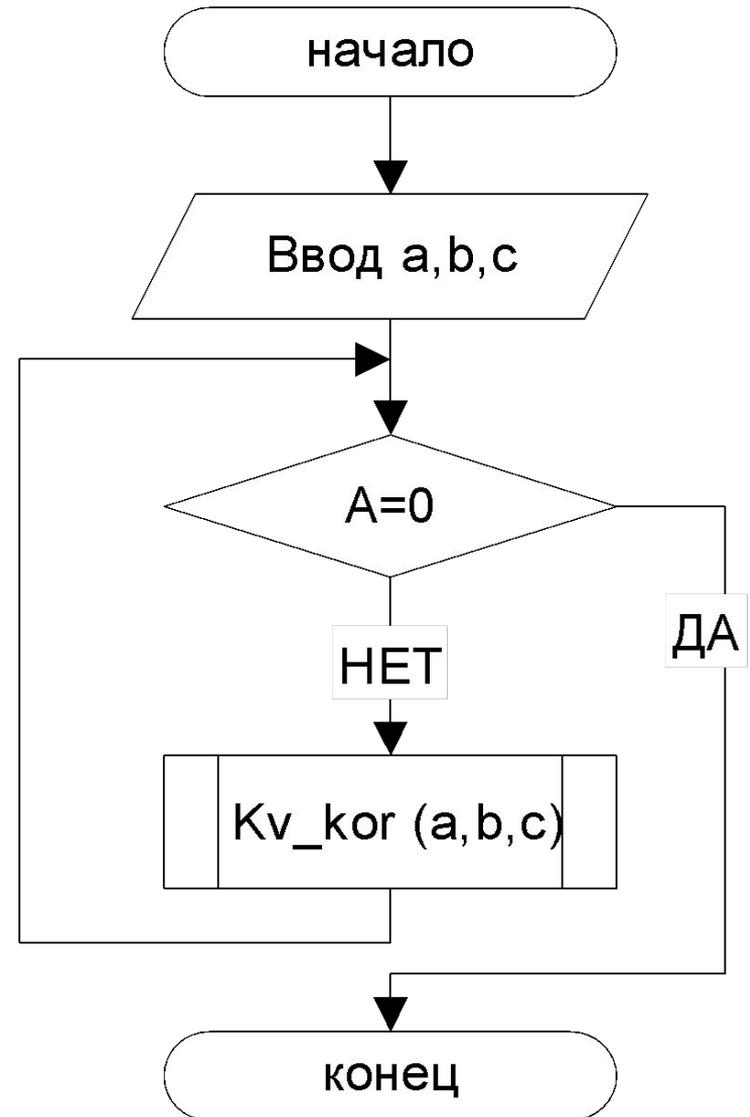
```
  1:
```

```
  a = InputBox("Введите коэфф. a")
```

```
  b = InputBox("Введите коэфф. b")
```

```
  c = InputBox("Введите коэфф. c")
```

```
End Sub
```



# Модули - процедуры и функции

**Модуль** - совокупность объявлений и процедур, объединенных в единое целое. Различают **стандартный модуль**, **модуль формы** и **модуль отчета**.

Стандартный модуль создается как объект БД и может выполнять любые вычисления. Модуль форм или отчета разрабатывается для обработки событий форм или отчетов. Каждый модуль состоит из *области описания* и *процедур*.

**Процедура** представляет собой последовательность *операторов*. Входящие в модуль процедуры объединены общей *областью описания*.

Процедуры делятся на две категории: подпрограммы и функции. Каждая процедура может быть вызвана из любого модуля, а функция - из таких объектов, как форма, запрос или отчет. Для передачи значений из вызывающих операторов в вызываемые процедуры служат *аргументы*.

Наряду с *общедоступными процедурами* **Public** (по умолчанию), существуют *локальные процедуры* **Private**, доступные только в том модуле, в котором они описаны.

Обращение к функции выполняется по ее имени на вкладке СОБЫТИЕ вместо выражения [**Процедура обработки события**], например, **=имя\_функции()**

# Переменные

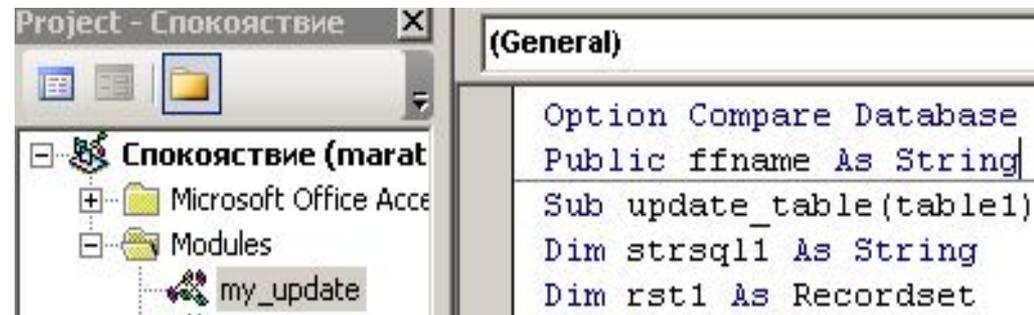
Переменные, объявленные без использования слова AS, принимают тип VARIANT (**16 байт для десятичных чисел и 22 байта для строк !**), например, переменная “а” в объявлении:

```
DIM a, b AS STRING
```

Переменные, объявленные в процедуре или функции, по умолчанию являются локальными. Они объявляются оператором **DIM**. Оператор **PRIVATE** явно объявляет локальную переменную для процедуры или модуля.

Переменную, значение которой можно использовать во всех процедурах модуля, объявляют со словом **PUBLIC** в разделе **DECLARATIONS** модуля.

**Переменную, значение которой можно использовать во всех модулях и процедурах приложения, объявляют со словом GLOBAL в разделе DECLARATIONS модуля.**



Директивы кода:

- **Option Explicit** - требование обязательного объявления переменных.
- **Option Compare Database** – метод сравнение текстовых строк в процессе их сортировки.

# СОБЫТИЯ

Объекты на формах и отчетах имеют **СВОЙСТВА** (например, скрытое поле) и **МЕТОДЫ** (например, поле обновить). Обращаться к ним можно в окне свойств объекта и в коде VBA, открываемом при выборе события (Click, DblClick), связанного с объектом (*поле события может содержать имя макроса*).

Если поле свойства события содержит строку [**Процедура обработки события**], то VBA выполнит процедуру обработки события, которая имеет вид

```
Sub Объект_Событие()... End Sub
```

Открытие **формы** включает события **Open** (открытие), **Load** (загрузка), **Activate** (включение).

Для отмены события используется аргумент Cancel. Если ему присвоить **True**, то событие отменяется. Можно отменить события: **BeforeDelConfirm** (до подтверждения Del), **BeforeInsert** (до вставки), **BeforeUpdate** (до обновления), **DblClick** (двойное нажатие кнопки), **Delete** (удаление), **Error** (ошибка), **Exit** (выход), **Format** (форматирование), **Open** (открытие), **Print** (печать), **Unload** (выгрузка).

Объекты **Screen** (свойство ActiveControl) : **ActiveForm** (активная форма), **ActiveReport** (активный отчет) и **PreviousControl** (предыдущий элемент управления).

Например, **Screen.PreviousControl.SetFocus** (установка фокуса на предыдущий объект).

Событие	Назначение
<b>GotFocus</b>	<b>Получение фокуса</b>
<b>KeyDown</b>	<b>Клавиша вниз</b>
<b>KeyPress</b>	<b>Нажатие клавиши</b>
<b>KeyUp</b>	<b>Клавиша вверх</b>
<b>LostFocus</b>	<b>Потеря фокуса</b>
<b>MouseDown</b>	<b>Кнопка вниз</b>
<b>MouseMove</b>	<b>Перемещение указателя</b>
<b>MouseUp</b>	<b>Кнопка вверх</b>

# Методы RunCommand и DoCmd

1. Метод **RunCommand** используется для выполнения команд меню/панелей инструментов с **вызовом диалоговых окон**.

**[DoCmd].RunCommand *command*** или **[Application].RunCommand *command***  
или **RunCommand *command***

где *command* - это соответствующая команда меню или панели инструментов.  
Префикс [Application] или [DoCmd] не обязателен.

2. Метод **DoCmd** выполняет эти же действия с заданием аргументов команды **без необходимости использования диалоговых окон команд**.

**RunCommand acCmdPrint** ' вывод *диалогового окна "Печать"*.

**RunCommand acCmdSelectRecord** ' выбор текущей записи (команда Выделить запись из меню Правка)

**DoCmd.PrintOut acSelection** ' печать текущей записи

**Вызов макросов с помощью команды DoCmd:**

**DoCmd DoMenuItem** A\_FormBar, A\_EditMenu, A\_UndoField, , A\_Menu\_Ver20 - *восстановление записи.*

**DoCmd GoToRecord** , , A\_NEWREC - *переход к новой записи.*

**DoCmd Close** - *закрытие текущего объекта.*

**DoCmd Quit** - *закрытие приложения.*

**DoCmd DoMenuItem** A\_FormBar, A\_EditMenu, 10, , A\_MenuVer20 - *вывод на экран диалога Поиск.*

# Модули

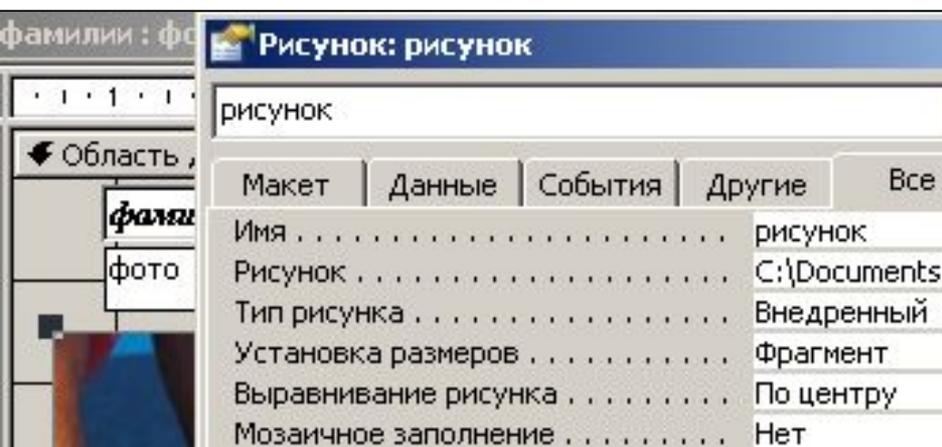
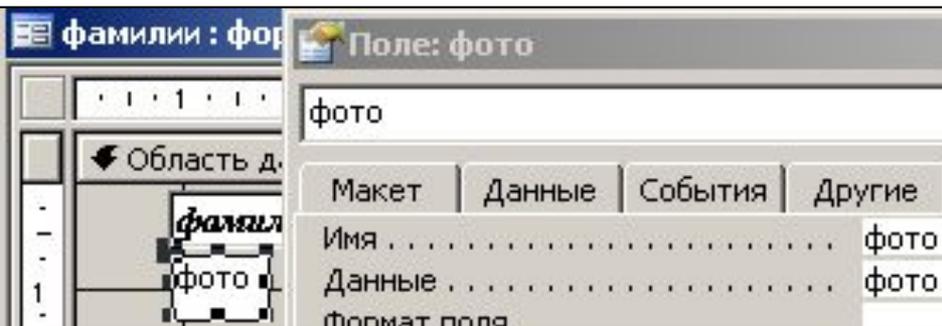
1. Столбчатая форма для таблицы *т.Абитуриенты*.

В поле *Фото* записано **имя файла с фото**.

Хранение не файла, а его имени в файловой системе эффективнее, т.к. уменьшает размер БД.

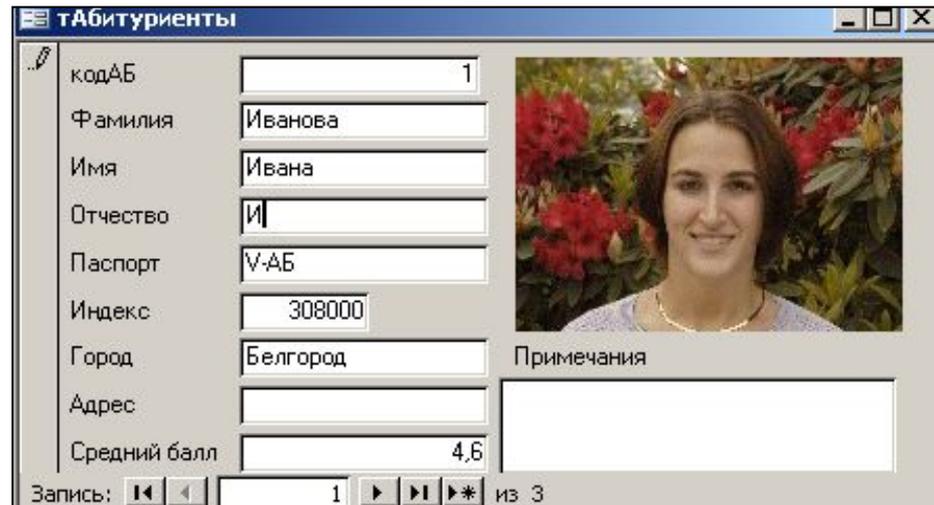
фамилии : таблица			
	код_фамилии	фамилия	фото
▶	2	Петров Петр Петрович	d:\ФОТО\empid1_1.jpg
	8	Сидоров Сидор Сидорович	d:\ФОТО\empid2.bmp
	9	Ахметов Ахмет Ахметович	d:\ФОТО\empid3.bmp
	16	Ибрагимов Ибрагим	d:\ФОТО\empid4.bmp

2. На форму поместим поле **ФОТО** со свойством **ВЫВОД НА ЭКРАН=НЕТ** и объект **РИСУНОК** с именем **РИСУНОК**.



3. В свойствах формы событие **ТЕКУЩАЯ ЗАПИСЬ** содержит код:

```
Private Sub Form_Current()  
    On Error Resume Next  
    If Me![фото] <> "" Then  
        Me.Рисунок.Picture = Me.фото  
    Else  
        Me.Рисунок.Picture = ""  
    End If  
End Sub
```



# Окно отладки программного кода

The screenshot shows the Microsoft Visual Basic IDE in a debugging state. The main window displays the code for a form's event procedure. A red dot on the left margin indicates a breakpoint at the line `If Me.Группа56 = 3 Then`. The code is as follows:

```
Private Sub Form_Current ()  
    If Me.Группа56 = 3 Then  
        Me.ПолеСоСписком29.Visible = False  
        Me.Надпись19.Visible = False  
    Else  
        Me.ПолеСоСписком29 = Me.код_субсидии  
        Me.ПолеСоСписком29.Requery  
    End If  
End Sub
```

The Immediate window shows the value of `?Me.Группа56` as `3`. The Watches window shows the value of `Me.Группа56` as `3`. The Properties window shows the value of `Me` as `Form_Субсидии`.

**Код события Текущая запись** (Code event Current record) - points to the code window.

**Точка останова** (Breakpoint) - points to the red dot in the left margin.

**Шаг отладки кода (F8)** (Step through code (F8)) - points to the yellow arrow in the left margin.

**Подручный процессор** (Immediate window) - points to the Immediate window.

**Просмотр значений переменных** (Watch variables) - points to the Watches window.

**Текущий объект** (Current object) - points to the Properties window.

# “Полосатый” отчет

Чередование темных и светлых полос задается на событии **ФОРМАТИРОВАНИЕ** выделенного мышкой заголовка **ОБЛАСТИ ДАННЫХ** отчета (**Свойство BACKCOLOR** означает цвет фона) :

```
Dim BackForth As Variant
```

```
Private Sub ОбластьДанных_Format(Cancel As Integer, FormatCount As Integer)
```

```
Const ColorWhite = 16777215
```

```
Const ColorGray = 12632256
```

```
Select Case BackForth
```

```
Case True
```

```
Me.ОбластьДанных.BackColor = ColorWhite
```

```
Case False
```

```
Me.ОбластьДанных.BackColor = ColorGray
```

```
End Select
```

```
BackForth = Not BackForth
```

```
End Sub
```

Лицей				
Группа: 201				
фамилия	имя	отчество	школа	класс
1 Акметов	Ахмет	Акметович	114	10А
2 Власов	Валей	Власович	114	10Б
3 Данилов	Данил	Данилович	111	9А

## Бегущая строка

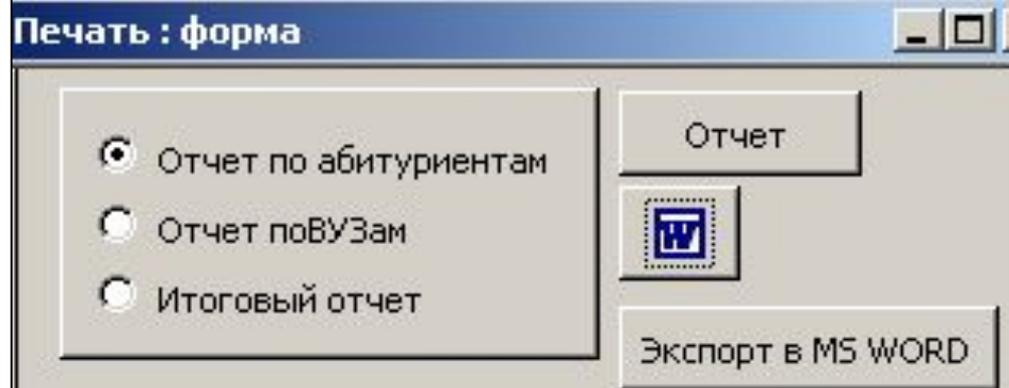
Для демонстрации действий с таймером создадим на форме надписи с именами НАДПИСЬ1, НАДПИСЬ2, НАДПИСЬ3, НАДПИСЬ4. В свойствах первой надписи на вкладке МАКЕТ параметр ВЫВОД НА ЭКРАН установим в значение ДА, у остальных надписей - в значение НЕТ. На событие формы ИНТЕРВАЛ ТАЙМЕРА указываем 200 (миллисекунд), а в поле ТАЙМЕР наберем код:

```
Private Sub Form_Timer()  
Select Case Me.TimerInterval  
Case 200  
    Me.Надпись2.Visible = True  
    Me.Надпись1.Visible = False  
    Me.TimerInterval = Me.TimerInterval + 200  
Case 400  
    Me.Надпись3.Visible = True  
    Me.Надпись2.Visible = False  
    Me.TimerInterval = Me.TimerInterval + 200  
Case 600  
    Me.Надпись4.Visible = True  
    Me.Надпись3.Visible = False  
    Me.TimerInterval = Me.TimerInterval + 200  
Case 800  
    Me.Надпись1.Visible = True  
    Me.Надпись4.Visible = False  
    Me.TimerInterval = 200  
End Select  
End Sub
```

Процедура выполняется **по истечении** времени, указанного в поле ИНТЕРВАЛ ТАЙМЕРА. Оператор Select Case ... End Select проверяет достижение таймером значений и поочередно задает для надписей режимы видимости (свойство Visible). По истечении 800 миллисекунд таймеру вновь задается 200 миллисекунд для бесконечного цикла таймера.

# Экспорт данных Access в документ Word

Создадим форму для контроля вывода данных на печать. На диске С создадим шаблон заполняемого из БД документа “С:\letter.doc”



ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ВУЗ	ТЕЛЕФОН	РЕЙТИНГ

```
Private Sub Кнопка10_Click()  
Dim wdd As Word.Document  
Dim wda As Word.Application  
Dim db As Database  
Set db = CurrentDb()  
Dim str_sql As String  
'проверяем вариант отчета в Word  
Select Case Me.Группа0.Value  
Case 1:  
'открываем документ с готовым шаблоном  
Set wdd = GetObject("C:\letter.doc")  
'создаем указатель на приложение  
Set wda = wdd.Parent  
'делаем приложение видимым  
wda.Visible = True  
'размер шрифта, установленный по умолчанию в MS  
WORD
```

```
a = wda.Selection.Font.Size  
  
' можно создать строку статического запроса  
' str_sql = "SELECT тАбитуриенты.кодАб," _  
' & "тАбитуриенты.Фамилия, тАбитуриенты.Имя, "  
' & " тАбитуриенты.Отчество, тВУЗы.НазваниеВУЗа, "  
' & "тВУЗы.Телефон, тВУЗы.Рейтинг FROM тВУЗы" _  
' & " INNER JOIN (тАбитуриенты INNER JOIN "  
' & "тСвязьАбВУЗ ON тАбитуриенты.кодАб = "  
' & "тСвязьАбВУЗ.КодАб) ON "  
' & " тВУЗы.КодВУЗа = тСвязьАбВУЗ.КодВУЗа"  
' Set rst = db.OpenRecordset(str_sql, dbOpenDynaset,  
dbReadOnly)  
  
' можно использовать существующий в БД запрос  
Set rst = db.OpenRecordset("АбВУЗ", dbOpenDynaset,  
dbReadOnly)
```

ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ВУЗ	ТЕЛЕФОН	РЕНТИНГ
Иванова	Ивана	И			
			УГНИ		0
			ВЭГУ		0
<i>ВУЗов - 2</i>					
Петров	Петр	П			
			УГАТУ		0
<i>ВУЗов - 1</i>					

'1-я табл. в открытом документе WDD

Set tbl = wdd.Tables(1)

With tbl

j = 2

rst.MoveFirst

Id = 0

kolz = 0

If rst.RecordCount <> 0 Then

Do Until rst.EOF

If rst.кодАБ <> Id Then

If Id <> 0 Then

.Cell(j, 1).Select

'ссылка на приложение

wda.Selection.InsertRowsBelow 1

wda.Selection.Font.Bold = True

wda.Selection.Font.Italic = True

wda.Selection.Font.Size = 14

j = j + 1

.Cell(j, 1) = "ВУЗов - " & kolz

.Cell(j, 1).Select

wda.Selection.InsertRowsBelow 1

wda.Selection.Font.Bold = False

wda.Selection.Font.Italic = False

wda.Selection.Font.Size = 12

```

j = j + 1
kolz = 0
End If
Id = rst.кодАБ
For i = 1 To 3
    .Cell(j, i) = Nz(rst.Fields(i), "")
Next i
End If
.Cell(j, 1).Select
wda.Selection.InsertRowsBelow 1
j = j + 1
For i = 4 To 6
    .Cell(j, i) = Nz(rst.Fields(i), "")
Next i
j = j + 1
kolz = kolz + 1
rst.MoveNext

```

Loop

.Cell(j, 1).Select

wda.Selection.InsertRowsBelow 1

wda.Selection.Font.Bold = True

wda.Selection.Font.Italic = True

wda.Selection.Font.Size = 14

.Cell(j, 1) = "ВУЗов - " & kolz

wda.Selection.Font.Bold = False

wda.Selection.Font.Italic = False

wda.Selection.Font.Size = a

wda.Selection.MoveDown

End If

End With

Case 2:

End Select

Set wdd = Nothing

Set wda = Nothing

End Sub

**Экспорт в MS WORD с использованием встроенной команды**

```

DoCmd.OutputTo acOutputQuery, "А6ВУЗ", acFormatRTF, "document.doc", True
'
acOutputTable

```

If Me.ActiveControl.Controls.Count <> 0 Then

clt = Me.ActiveControl.ControlType

End If