

---

# СУБД ACCESS

---

Автор: Тутыгин В.С.

---

# Назначение макросов

Макросы используются для автоматизации исполнения последовательности рутинных операций с БД, например:

- а) автоматическое открытие и/или печать нескольких форм или отчетов;
  - б) автоматическое экспортирование данных из БД в другой формат (например, в dBase).
-

---

# Виды макрокоманд в макросах

Макрос представляет собой одну или последовательность нескольких макрокоманд из числа следующих:

Восстановить

ВывестиВформате

Звуковой сигнал

Песочные часы

ОткрытьЗапрос

ОткрытьМодуль

и т. д.

---

# Макрос с условием в форме

Можно устанавливать запуск макроса при выполнении какого-либо условия в форме или отчете, например:

Условие	Макрокоманда	
	ОткрытьФорму	ЗапросЧитателя
[Forms]![ЗапросЧитателя]![КолЭкз]>1	ОткрытьФорму	Ответ3
[Forms]![ЗапросЧитателя]![КолЭкз]=1	ОткрытьФорму	Ответ2
[Forms]![ЗапросЧитателя]![КолЭкз] Is Null	ОткрытьФорму	Ответ1

---

# Создание макроса с условием

- [Макрос]-[Создать];
- Создать макрос;
- [Вид] – [Условие].

В результате появится дополнительный столбец УСЛОВИЕ, в который можно внести условие запуска для каждой макрокоманды. Например:

```
[Forms]![Запрос читателя]![КолЭкз]>1
```

---

# Запуск макроса

Запуск макроса может осуществляться:

- а) автоматически при открытии БД;
- б) при активизации специально созданной для запуска макроса кнопке управления. Кнопка может быть создана в форме или отчете;
- в) автоматически при возникновении определенного события в форме, отчете или в элементе управления типа поле или поле со списком.

# Использование модулей на VBA

VBA имеет значительно большие функциональные возможности по сравнению с макросами, а именно:

- VBA позволяет использовать в выражениях и в качестве аргументов процедур переменные, значения которых могут динамически изменяться, в макросах же применяются только статические значения;
- VBA обеспечивает очень гибкие возможности управления всеми объектами Access, особенно объектами, связанными с данными. Например, VBA дает возможность работать с отдельными записями набора. Макросы позволяют работать только с набором записей.

---

# Использование модулей на VBA

- VBA обеспечивает возможность создавать средства перехвата и обработки ошибок. Макросы этого не позволяют.

Поэтому макросы целесообразно использовать для создания прототипов приложений пользователя, а «доведение до ума» приложений следует производить с использованием VBA.

---



## Пример задачи, связанной с вводом данных в БД из внешней аппаратуры

БД предназначена для учета отгрузки предприятием металлического проката. Взвешивание отгружаемой продукции производится на автоматических весах. Результат взвешивания считывается в компьютер через СОМ-порт и записывается в файл. Программа чтения данных представляет собой EXE-модуль. Запуск EXE-модуля и перенос данных из файла в БД нужно осуществить из БД.

# Интерфейс пользователя

товары1

<b>№ п/п</b>	<input type="text" value="1"/>
<b>Дата/Время</b>	<input type="text" value="09.10.2001 12:48:28"/>
<b>№ вагона</b>	<input type="text" value="355"/>
<b>Описание груза</b>	<input type="text" value="Лента стальная"/>
<b>№ накладной</b>	<input type="text" value="789-2000"/>
<b>масса,т</b>	<input type="text" value="1,94"/>

Record:       of 17

# Алгоритм решения

- В форме ТОВАРЫ пользователь заполняет поля: №вагона, Описание груза, № накладной;
- По событию ПОСЛЕ ОБНОВЛЕНИЯ запускается макрос, который запускает модуль;
- Модуль запускает EXE-модуль, затем после паузы, соответствующей времени взвешивания, считывает результаты взвешивания из файла и заносит их в БД.

---

# Пример запуска exe-модуля из модуля VBA

```
Dim a, b, p As Variant
```

```
Shell "d:\gast20.exe" 'Запуск exe-модуля, выполняющего
```

```
'чтение данных с автоматических весов и запись в файл .dat
```

```
a = Timer 'Запуск функции чтения кода времени с таймера
```

```
b = a
```

```
While (b - a < 2) 'Цикл ожидания на 2 сек
```

```
b = Timer '(За 2 сек производится взвешивание)
```

```
Wend
```

---

# Пример ввода данных в БД из файла

**Open "D:\b1" For Input As 1** 'Открытие файла b1.dat для чтения

**Input #1, p** 'Чтение результата взвешивания p из файла

**Close #1** 'Заккрытие файла

**Beep** 'Звуковой сигнал

**Forms![товары1]![масса,г] = p \* 0.01** 'Ввод данных в БД

**Forms![товары1]![Дата/время] = Now()** 'Ввод текущего времени в БД

Запуск модуля с помощью макроса

# Пример задачи, связанной с необходимостью автоматической корректировки данных о запасах на складе по результатам продаж

**Расход**

НомерРасхОрдера	<input type="text" value="1"/>
КодТовара	<input type="text" value="Карандаш"/>
Заказчик	<input type="text" value="ЗАО " аврора""=""/>
ДатаЗаказа	<input type="text" value="14.11.2004"/>
КолЗаказаТовара	<input type="text" value="50"/>
СтоимЗаказа	<input type="text"/>

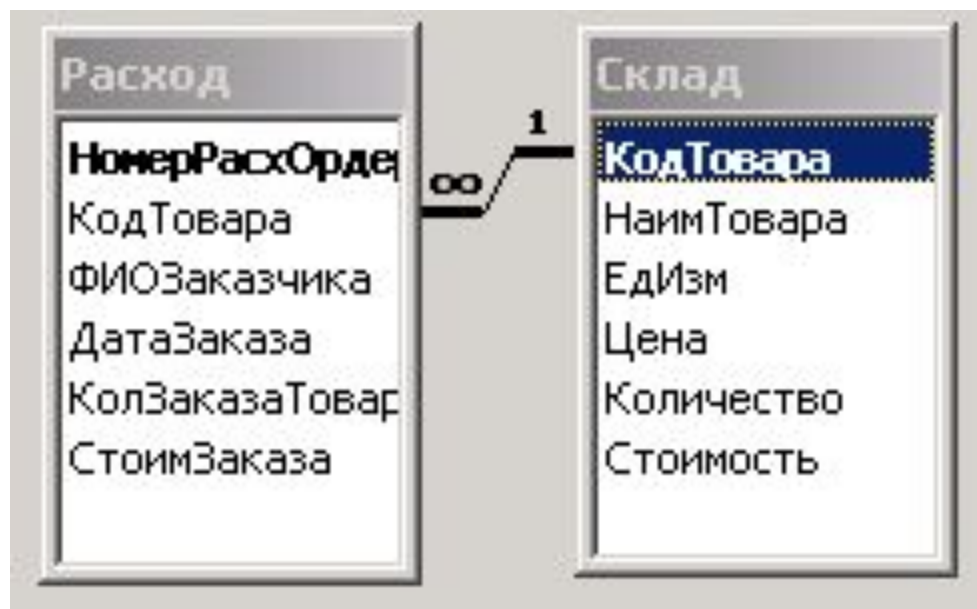
Record:       of 4

**Склад**

КодТовара	<input type="text" value="1"/>
НаимТовара	<input type="text" value="Карандаш"/>
ЕдИзм	<input type="text" value="шт"/>
Цена	<input type="text" value="2,00р."/>
Количество	<input type="text" value="200"/>
Стоимость	<input type="text" value="400,00р."/>

Record:       of 12

# Пример использования модулей на VBA



Автоматический перерасчет остатка товаров на складе после каждой продажи можно произвести только с помощью модуля на VBA

---

# Модуль учета продаж

```
Private Sub КолЗаказаТовара_AfterUpdate()  
Dim stDocName As String  
Dim stLinkCriteria As String  
stDocName = "Склад"  
stLinkCriteria = "[КодТовара]=" & Me![КодТовара]  
Продано = Forms("Расход")("КолЗаказаТовара").Value  
<Операторы, выполняющие перерасчет остатка на  
складе>  
End Sub
```

---



# Операторы перерасчета остатка на складе

‘Фрагмент модуля открывает форму Склад на записи,  
‘связанной по ключевому полю КодТовара с записью,  
‘открытой в данный момент в форме Расход, затем  
‘ производит перерасчет

```
DoCmd.OpenForm ("Склад"), , , stLinkCriteria  
Остаток = Forms("Склад")("Количество").Value  
Forms("Склад")("Количество").Value = Остаток - Продано
```

Пример использования модулей на VBA

---

# СУБД ACCESS

---

Защита баз данных

# Способы защиты информации

- защита базы данных (mdb-файла) с помощью пароля и шифрования;
- защита приложения Microsoft Access путем сокрытия объектов в окне базы данных и настройки параметров запуска;
- защита паролем программы на языке VBA;
- защита программ VBA путем создания файла, в котором отсутствует программный код;
- защита базы данных и ее объектов средствами защиты на уровне пользователей.

# Защита базы данных (mdb-файла) с помощью пароля

- [Файл] – [Открыть...];
- найти в каталогах файл базы данных и пометить его;
- [Открыть] – [Монопольно].

После открытия окна базы данных сохранить ее с паролем:

- [Сервис] – [Защита] – [Задать пароль базы данных...],
- задать пароль доступа в окнах всплывающей панели «Задание пароля базы данных».

# Соккрытие объектов в окне базы данных

- пометить объект, затем вызвать контекстное меню, выбрать пункт «Свойства», затем в окне «Атрибуты» всплывающей панели «Свойства» установить флажок «скрытый»;
- [Сервис] – [Параметры...] ;
- открыть вкладку «Вид» и снять флажок «скрытые объекты».

---

# Открытие скрытых объектов

- [Tools] – [Options...];
- открыть вкладку «Вид» и установить флажок «скрытые объекты».

В результате значки скрытых объектов появятся, но будут неконтрастными. Для восстановления скрытого объекта нужно пометить его значок, затем вызвать контекстное меню, выбрать пункт «Свойства», затем в окне «Атрибуты» всплывающей панели «Свойства» сбросить флажок «скрытый».

---

# Скрытие окна базы данных

- [Сервис] – [Параметры запуска...];
- во всплывающей панели «Параметры запуска» сбросить флажок «Окно базы данных».

---

# Открытие скрытого окна базы данных

Для того, чтобы увидеть скрытое окно базы данных, нужно найти файл базы данных и открывать его при нажатой клавише <Shift>. После этого:

- а) [Сервис] – [Параметры запуска...];
  - б) во всплывающей панели «Параметры запуска» установить флажок «Окно базы данных».
-



---

# Защита путем создания файла MDE базы данных

Файл MDB, в котором хранится база данных, может быть преобразован в файл MDE. В файле MDE весь программный код базы данных хранится в скомпилированном виде. Это исключает возможность внесения изменений в модули, формы и отчеты базы данных в режиме Конструктор и, таким образом, частично защищает базу данных от несанкционированных изменений.

---

# Создание файла MDE

- [Файл] – [Открыть...];
- найти в каталогах файл базы данных и пометить его;
- [Открыть] – [Монопольно],  
затем выполнить следующую последовательность действий:
- [Сервис] – [Служебные программы] – [Создать MDE-файл...].

*Примечание. Файл MDE нельзя вновь преобразовать в MDB! Поэтому, чтобы сохранить возможность внесения изменений в базе данных, перед преобразованием файла MDB в MDE нужно сохранить резервную копию файла MDB!*

# Защита базы данных (mdb-файла) с помощью шифрования

Чтобы зашифровать базу данных, ее нужно закрыть, а затем выполнить следующую последовательность действий:  
[Сервис] – [Защита] – [Закодировать или раскодировать базу данных...]  
и указать имена шифруемого и результирующего файлов. Можно одновременно защитить базу данных паролем.

---

# Расшифровка файла базы данных

[Сервис] – [Защита] – [Закодировать или раскодировать базу данных...] и указать имена исходного и результирующего файлов.

---