Использование макросов в среде Access

9.1. Введение

Макросы являются удобным средством выполнения команд в СУБД Access. Макрос представляет собой одну или более макрокоманд, объединенных под одним именем. Каждой макрокоманде ставится в соответствие какое-либо действие. Например, открыть запрос, открыть форму, перейти к элементу управления, запустить приложение, вывести сообщение. Всего в СУБД Accesssзадействовано более пятидесяти макрокоманд.

У макрокоманды могут быть аргументы, число которых зависит от ее типа, но есть и макрокоманды без аргументов. Посредством аргументов конкретизируется действие, которое выполняет макрокоманда. Например, в макрокоманде «Открыть таблицу» в качестве аргумента выступает имя таблицы. Некоторые аргументы имеют список возможных значений. Например, у макрокоманды «Выполнить команду» всего один аргумент, который имеет более пятисот возможных значений. Для каждой макрокоманды может быть сформировано условие ее выполнения. В качестве аргументов условий можно задействовать значение полей форм и отчетов.

Макрокоманда связывается с каким-либо событием и выполняется при его наступлении. В качестве событий могут быть использованы все события, возможные в формах и отчетах. Например, можно запустить макрос при наступлении события «Нажатие кнопки», можно также запустить макрос при наступлении события «Закрыть форму». Таким образом, макросы обеспечивают программирование действий БД в ответ на возникающие события, которые могут быть инициированы пользователем или самим приложением.

9.2. Создание макроса

Для того чтобы сформировать макрос, необходимо в базовом окне СУБД Access выбрать вкладку «Макросы» и щелкнуть по инструменту «Создать». В результате этих действий сформируется окно, показанное на рисунке 9.1:

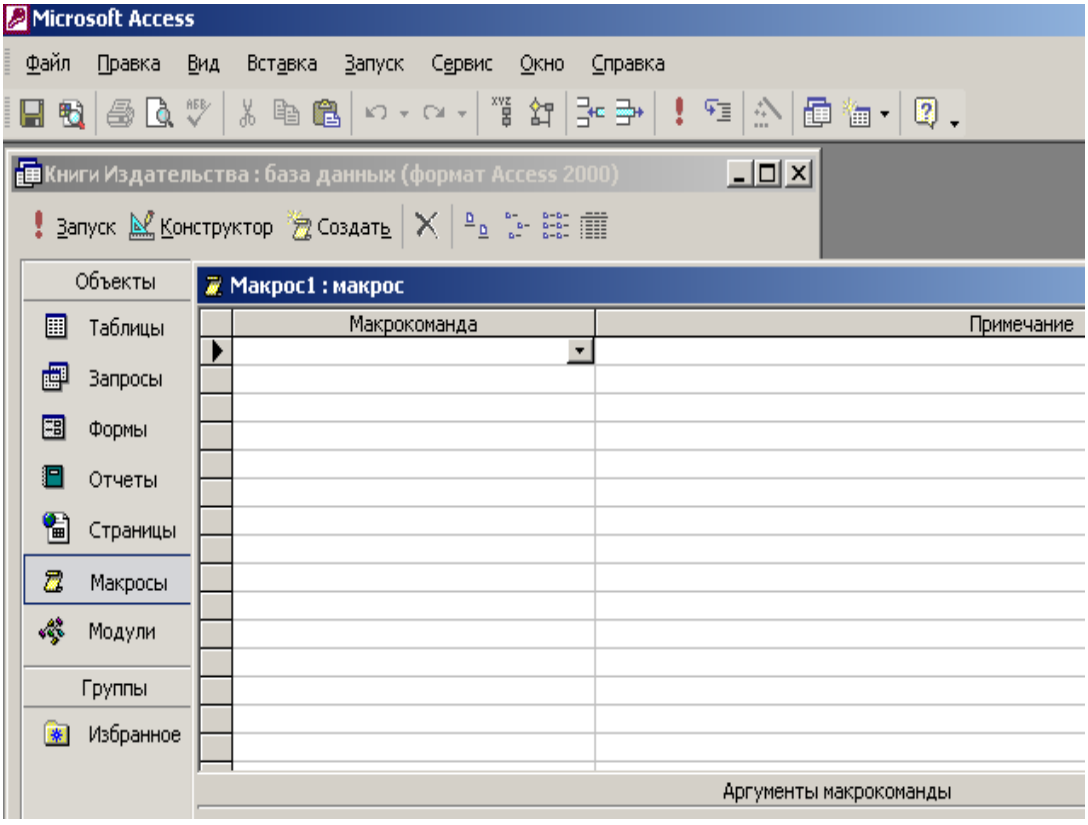


Рис. 9.1.Начальный этап создания макроса

В этом окне в области «Макрокоманда» вводятся или выбираются из предлагаемого списка макрокоманды. В области «Примечание» вводятся необязательные комментарии.

С помощью меню «Вид/Имя макроса» можно отобразить область для ввода имен групп. С помощью меню «Вид/Условия» можно отобразить область для ввода условий, в зависимости от которых будут выполняться макрокоманды. После выбора указанных позиций меню окно макроса примет вид, показанный на рисунке 9.2:

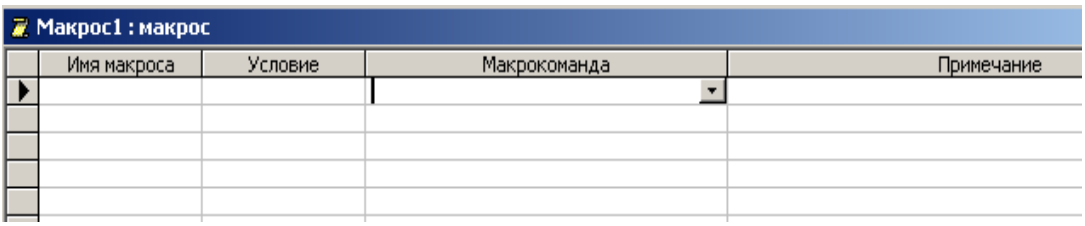


Рис. 9.2.Окно создания макроса с четырьмя областями

Области «Имя макроса» и «Условие» используются не очень часто, поэтому они по умолчанию не выводятся. Чтобы их скрыть, применяются соответственно команды меню «Вид/Имя макроса» и «Вид/Условия».

Чтобы сформировать макрокоманду, необходимо в области «Макрокоманда» в поле со списком выбрать нужную макрокоманду и в случае необходимости ввести ее аргументы.

Поле с раскрытым списком представлено на рисунке 9.3:

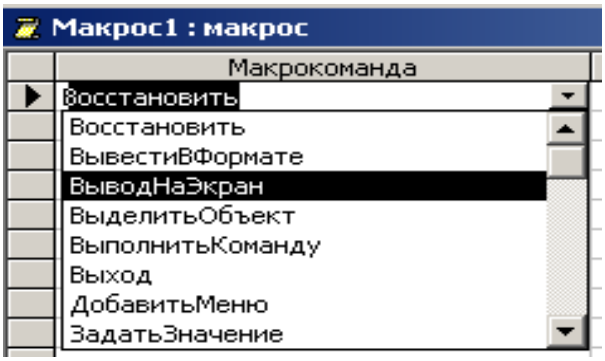


Рис. 9.3.Список макрокоманд

Здесь представлен список из 57-ми макрокоманд с аргументами и без них. Например, макрокоманда «Восстановить» не имеет аргументов. Она служит для восстановления прежних размеров того окна, из которого макрос запускается, и поэтому не нуждается в уточнениях.

Макрокоманда «ВывестиВформате» имеет несколько аргументов, часть из которых необходимо обязательно задать. На рисунке 9.4 представлен вид окна макросов с выбранной макрокомандой «ВывестиВформате» и соответствующим перечнем аргументов, который располагается в нижней части окна:

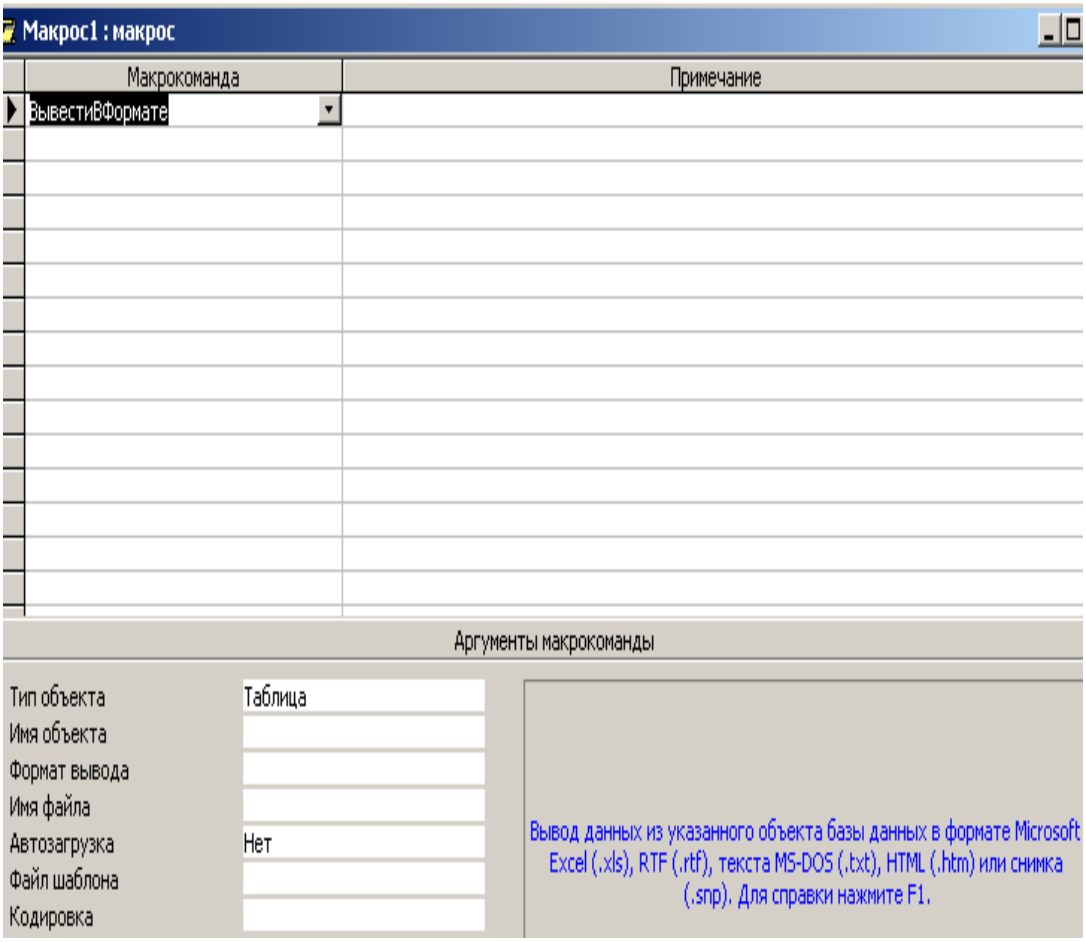


Рис. 9.4.Окно макросов с макрокомандой «ВывестиВформате»

Как видно из этого рисунка, макрокоманда содержит 7 аргументов. В правой нижней части окна выводится краткая справка по выбранной макрокоманде или выбранному полю. Для получения более детальной справки, как и в любом приложении Microsoft Office, необходимо выбрать меню «Справка/Что это такое?» и щелкнуть по нужному объекту.

Рассматриваемая макрокоманда предназначена для вывода данных из объекта базы данных в другой формат вывода. В качестве объекта вывода могут быть задействованы практически все объекты СУБД Access (таблица, форма, отчет и другие). В качестве формата вывода могут использоваться форматы HTML (*.htm; *.html), текстовые файлы (*.txt), Microsoft Excel (*.xls), Rich Text Format (*.rtf) и другие. На рисунке 9.5 представлен заполненный список аргументов:

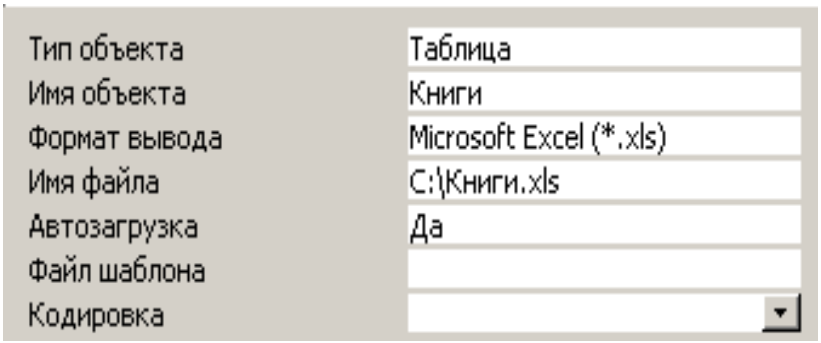


Рис. 9.5. Список аргументов макрокоманды «ВывестиВформате»

Набор заполненных аргументов показывает, что с помощью этой макрокоманды данные выводятся из таблицы «Книги» в формате табличного процессора Microsoft Excel, причем имя создаваемого

файла – «Книги.xls». Значение «Да» аргумента «Автозагрузка» указывает на то, что после запуска данного макроса будет загружен файл «Книги.xls».

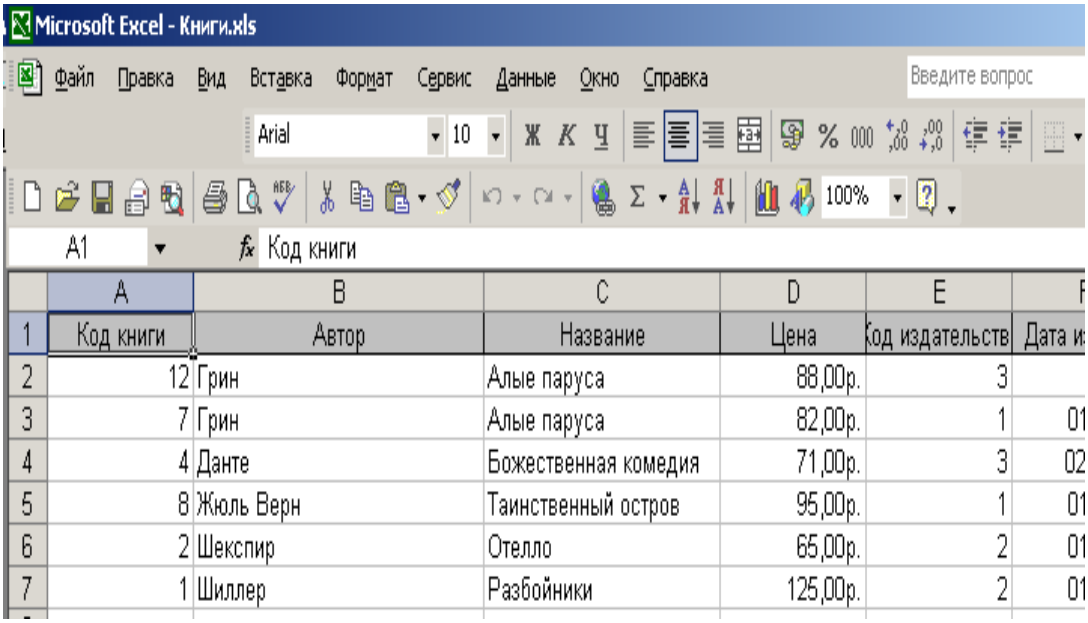
Для сохранения созданного макроса можно воспользоваться, как обычно, меню «Файл/Сохранить». Можно также закрыть окно и подтвердить сохранение макроса, при этом будет запрошено его имя (рисунок 9.6):

Рис. 9.6.Ввод имени макроса

В рассматриваемом случае макрос сохранен с именем «Запуск приложения». Значок данного макроса разместится на вкладке «Макросы» наряду с другими макросами (рисунок 9.7):

Рис. 9.7.Пиктограмма созданного макроса «Запуск приложения»

Для выполнения макроса достаточно дважды щелкнуть по его значку или имени. Результат выполнения макроса «Запуск приложения» представлен на рисунке 9.8:



	A	B	C	D	E	F
1	Код книги	Автор	Название	Цена	Код издательств	Дата и
2	12	Грин	Алые паруса	88,00р.	3	
3	7	Грин	Алые паруса	82,00р.	1	01
4	4	Данте	Божественная комедия	71,00р.	3	02
5	8	Жюль Верн	Таинственный остров	95,00р.	1	01
6	2	Шекспир	Отелло	65,00р.	2	01
7	1	Шиллер	Разбойники	125,00р.	2	01

Рис. 9.8.Результат выполнения макроса «Запуск приложения»

Как видно из этого рисунка, загрузился табличный процессор Microsoft Excel, а в нем в качестве таблицы представлена таблица базы данных. Файл «Книги.xls» будет храниться на диске вплоть до следующего вызова макроса и обновления.

Этот макрос, как и любой другой макрос, можно связать с каким-либо событием любого объекта БД. Тогда он будет запускаться при наступлении этого события.

Рассмотренный макрос может оказаться очень полезным. В частности, полученную таблицу можно передать или переслать специалисту Excel для обработки в данной системе и использования ее возможностей.

9.3. Примеры макрокоманд

Число макрокоманд довольно велико, поэтому рассмотреть их все не представляется возможным. Некоторые из них используются исключительно редко, назначение многих – очевидно. В связи с этим вначале рассматриваются часто используемые относительно простые макрокоманды, а затем – наиболее распространенные более сложные макрокоманды.

Макрокоманда «ВыводНаЭкран» позволяет отменить вывод информационных сообщений, которые сопутствуют выполнению некоторых макросов.

Макрокоманда «ВыделитьОбъект» позволяет сделать активным объект, который указан в аргументах макрокоманды. После этого имеется возможность выполнять макрокоманды для этого объекта.

Макрокоманда «Выход» позволяет завершить работу с СУБД Access.

Макрокоманда «Закрыть» позволяет закрыть активный объект, причем имя и тип этого объекта задаются в аргументах макрокоманды. Это может быть таблица, форма, отчет, запрос и другие объекты.

Макрокоманда «ЗапускПрограммы» позволяет запустить нужную программу, причем ее имя задается в аргументах макрокоманды.

Макрокоманда «КэлементуУправления» позволяет перейти к элементу управления в активной форме. Обычно это поле. Имя элемента управления вводится в качестве аргумента макрокоманды.

Макрокоманда «ПесочныеЧасы» позволяет вывести или отменить вывод изображения указателя мыши в виде песочных часов. Обычно часы выводятся до начала выполнения продолжительных действий БД и убираются по их завершению.

Макрокоманда «ПоказатьВсеЗаписи» позволяет отменить фильтр таблицы запроса или формы, если он использовался.

Макрокоманда «Развернуть» позволяет развернуть активное окно. Нередко это бывает полезно, так как иногда активное окно располагается на фоне окон, открытых ранее.

Макрокоманда «Сигнал» позволяет выдать звуковой сигнал. Обычно он используется, чтобы обратить внимание пользователя на нестандартную ситуацию.

Макрокоманда «УстановитьСообщения» позволяет запретить или разрешить вывод режимных сообщений, например, сообщений о том, что предполагается удаление из таблицы 23-х записей.

9.3.1. Макрокоманда «ВыполнитьКоманду»

Эта макрокоманда позволяет выполнить одну из пятисот команд. Такие команды, в основном, являются аналогами команд системы Access, которые задействованы в меню, подменю, на панелях инструментов. Макрокоманда имеет всего один аргумент, который выбирается из предлагаемого списка. В качестве примера на рисунке 9.9 представлена примерно десятая часть команд, которые можно выполнить посредством макрокоманды «ВыполнитьКоманду»:

Рис. 9.9.Примеры команд, доступных макрокоманде «ВыполнитьКоманду»

Эта макрокоманда является очень полезной и позволяет пользователю в рамках своей БД использовать практически все возможности СУБД Access. Например, если ее аргументом является «СхемаДанных», то после выполнения макрокоманды отобразится схема данных (рисунок 9.10):

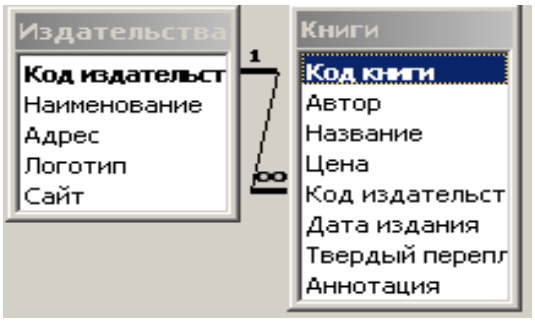


Рис. 9.10.Результат макрокоманды «ВыполнитьКоманду»

с аргументом «СхемаДанных»

Таким образом, эта команда может быть доступна пользователю БД. В принципе, реальная БД должна быть освобождена от меню среды Access. Это делается из соображений невозможности доступа пользователя к модификации ее объектов, а также для избежания перегруженности окон приложения.

9.3.2. Макрокоманда «ЗадатьЗначение»

Эта макрокоманда позволяет присвоить какому-либо элементу определенное значение. Элемент указывается с помощью аргумента «Элемент», а требуемое значение задается посредством аргумента «Выражение». Например, в определенной ситуации может возникнуть необходимость занесения в поле «Код издательства» формы «Книги» значения «Код издательства» активной записи формы «Издательства», что необходимо для связывания записей. Тогда элементу [Forms]![Книги]![Код издательства] необходимо присвоить значение [Forms]![Издательства]![Код издательства]. В этом примере использована нотация, принятая в СУБД Access. Ключевое слово [Forms] определяет, что элемент принадлежит к классу «Формы». Далее следуют имена формы и поля, разделенные восклицательным знаком. Все элементы такой конструкции необходимо брать в квадратные скобки.

В окне макрокоманды ее аргументы выглядят следующим образом (рисунок 9.11):

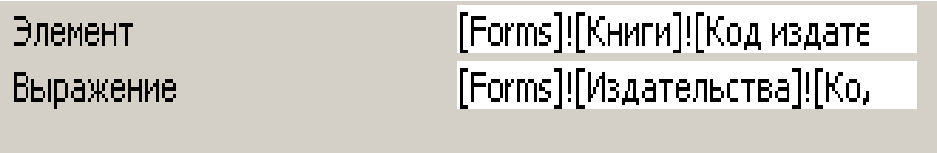


Рис. 9.11.Аргументы макрокоманды «ЗадатьЗначение»

В качестве выражения может быть задействовано любое математическое выражение.

После сохранения макроса его можно преобразовать в процедуру на языке Visual Basic. Для этого надо выделить преобразуемый макрос в базовом окне системы Access, а затем выбрать в меню команду «Сервис/Макрос/Преобразовать макросы», после чего сформируется окно (рисунок 9.12), в котором задается вид программы:

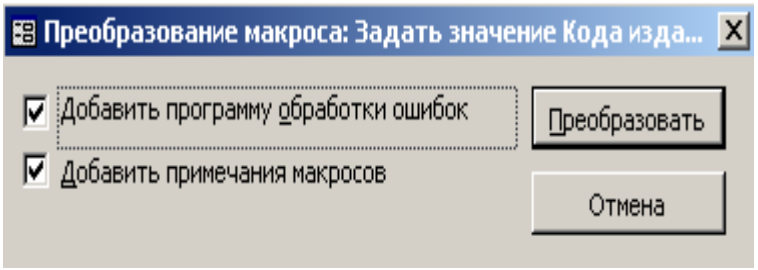


Рис. 9.12.Окно преобразования макроса

После выбора флажков и нажатия кнопки «Преобразовать» сформируется функция, которая разместится в группе объектов «Модули» (рисунок 9.13):

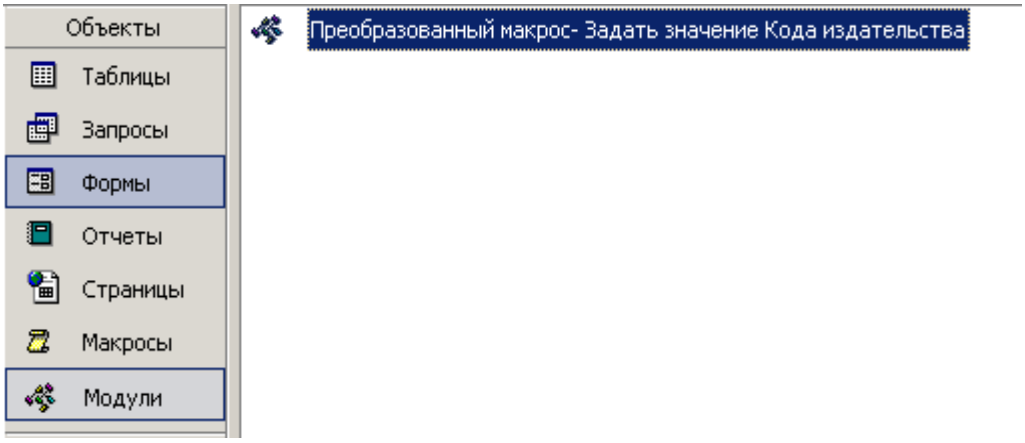


Рис. 9.13.Размещение сформированной процедуры

Этот модуль в режиме конструктора показан на рисунке 9.14:

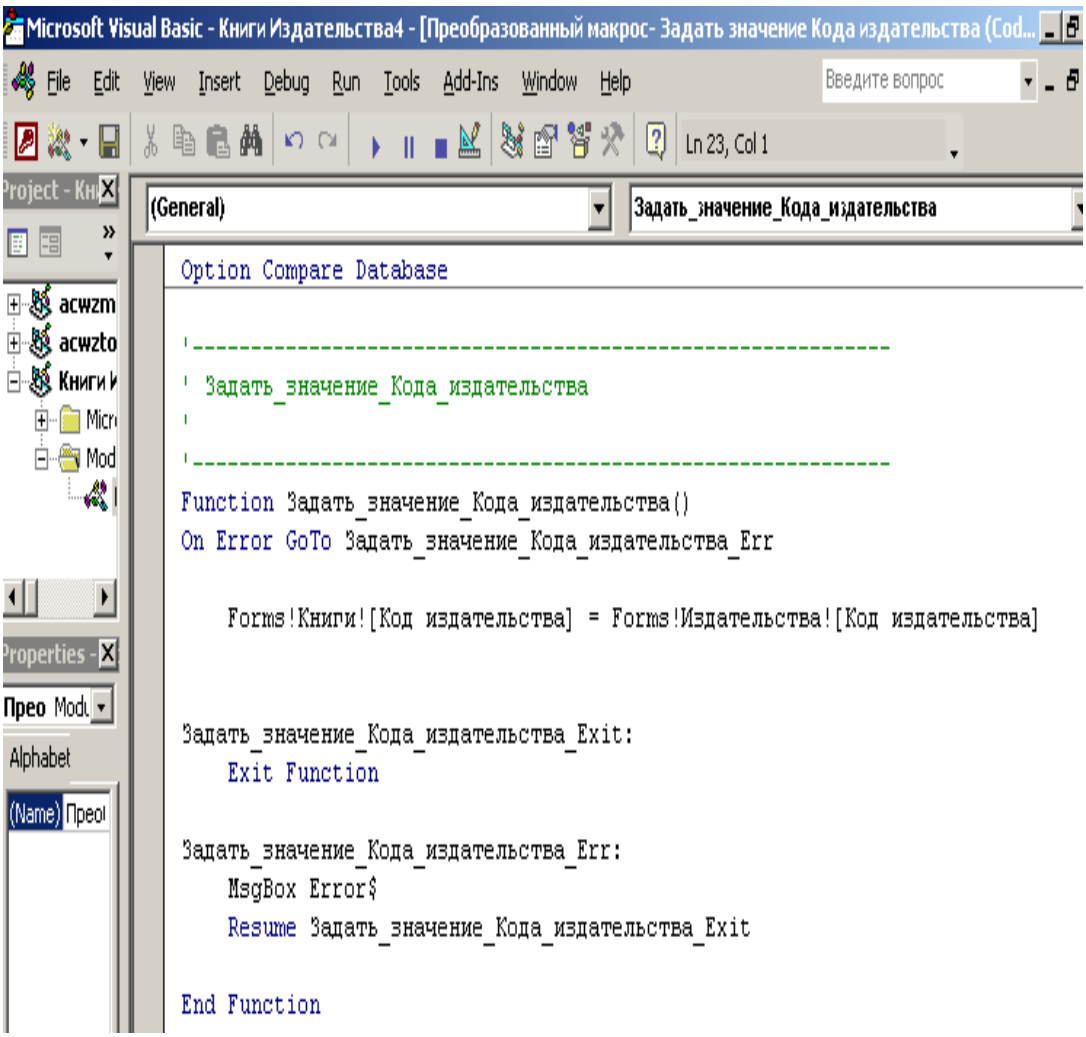


Рис. 9.14.Сформированная функция на языке Visual Basic

Это окно языка Visual Basic. В нем доступны все средства системы программирования Visual Basic. В частности разработчик БД может редактировать, изменять программу, а также выявлять допущенные ошибки программирования.

Функция, представленная в окне на рисунке 9.14, содержит средства обработки ошибок и комментарии, причем единственная исполняемая команда этой функции следующая:

Forms!Книги![Код издательства] = Forms!Издательства![Код издательства].

Средства преобразования макросов в процедуры следует использовать в том случае, когда в разрабатываемой программе задействованы команды, которые реализуются с помощью каких-либо макросов, а также другие команды, не реализованные посредством макросов. Кроме того, рассмотренные средства можно использовать для изучения основ языка программирования Visual Basic.

9.3.3. Макрокоманда «ЗапускМакроса»

Эта макрокоманда используется для циклического запуска макроса. Ее аргументы представлены на рисунке 9.15:

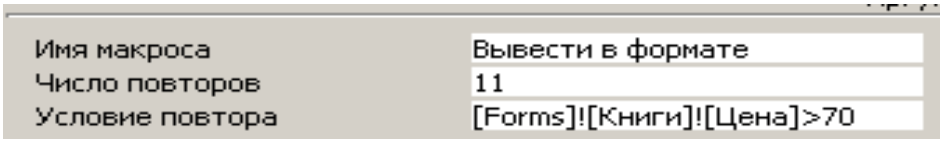


Рис. 9.15.Аргументы макрокоманды «ЗапускМакроса»

С помощью аргумента «Имя макроса» задается, естественно, имя макроса, который необходимо выполнять в цикле.

Аргумент «Число повторов» задает максимальное количество повторов выполнения макроса.

С помощью аргумента «Условие повтора» задается условие завершения циклического вызова макроса. Цикл будет прекращен, если условие примет значение “Ложь”.

Если преобразовать этот макрос в функцию, то она примет следующий вид:

```
Function Запуск_макроса()

On Error GoTo Запуск_макроса_Err

DoCmd.RunMacro "Вывести в формате", 11, "[Forms]![Книги]![Цена]>70"

Запуск_макроса_Exit:

Exit Function

Запуск_макроса_Err:

MsgBox Error$

Resume Запуск_макроса_Exit

End Function
```

В качестве объекта языка Visual Basic используется объект DoCmd, который позволяет выполнять макрокоманды СУБД Access. В данном случае выполняется макрокоманда ЗапускМакроса (соответствующий метод RunMacro). Сама команда выглядит так:

```
DoCmd.RunMacro "Вывести в формате", 11, "[Forms]![Книги]![Цена]>70".
```

В ней, как и во всех объектах DoCmd, перечисляются аргументы соответствующей макрокоманды. В связи с этим, для нашего случая преобразовывать макрокоманды в функции имеет смысл только тогда, когда эти функции впоследствии будут расширены.

9.3.4. Макрокоманда «ЗапускПриложения»

Эта макрокоманда позволяет запустить приложение ОС Windows. Она имеет единственный аргумент, в котором необходимо ввести командную строку для запуска приложения. Например, если надо запустить из БД приложение Acrobat 5.0, то в качестве аргумента макрокоманды нужно ввести командную строку вида

D:\Program Files\Adobe\Acrobat 5.0\Reader\AcroRd32.exe.

На рисунке 9.16 показано соответствующее окно макрокоманды:

Рис. 9.16.Окно макрокоманды «ЗапускПриложения»

Соответствующая команда выглядит следующим образом:

Call Shell("D:\Program Files\Adobe\Acrobat 5.0\Reader", 1).

9.3.5. Макрокоманда «КопироватьОбъект»

Эта макрокоманда позволяет копировать любой объект из одной БД в другую или в ту же БД, но с другим именем. Она может быть полезна для создания архивных копий таблиц или для использования в новой БД объектов существующей БД.

Заполненный список аргументов макрокоманды приведен на рисунке 9.17:

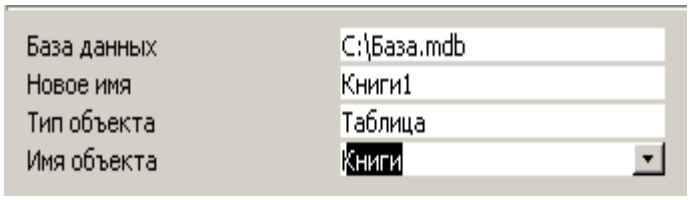


Рис. 9.17.Аргументы макрокоманды «КопироватьОбъект»

Аргумент «База данных» позволяет задать полное имя БД, в которую будет копироваться объект, она должна существовать. Если копирование осуществляется в текущую БД, то этот аргумент не заполняется.

Аргумент «Новое имя» задает новое имя объекта. Если имя объекта не меняется, этот аргумент также не заполняется.

Аргумент «Тип объекта» позволяет задать тип копируемого объекта.

Аргумент «Имя объекта» позволяет задать имя копируемого объекта.

Создавать и использовать эту макрокоманду имеет смысл, когда часто выполняются однотипные копирования и к тому же копирование осуществляется пользователем. Гораздо проще открыть две БД и перенести объект из одной БД в другую. В рамках одной БД можно скопировать объект, например, с помощью контекстного меню и вставить его с другим именем.

Соответствующая команда выглядит следующим образом:

DoCmd.CopyObject "C\База.mdb", "Книги1", acTable, "Книги".

9.3.6. Макрокоманда «ОткрытьЗапрос»

Эта макрокоманда позволяет открыть запрос, например, в режиме просмотра. Ее аргументы представлены на рисунке 9.18:

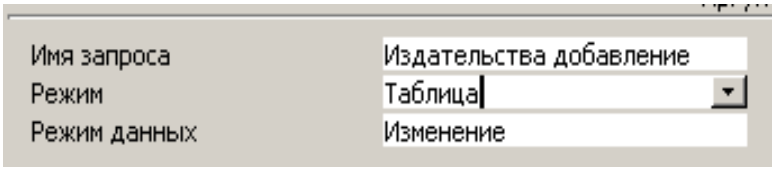


Рис. 9.18.Аргументы макрокоманды «ОткрытьЗапрос»

С помощью аргумента «Имя запроса» задается имя запроса, который предполагается открыть посредством макрокоманды.

Аргумент «Режим» позволяет выбрать режим, в котором будет открыт запрос. Список возможных режимов приведен на рисунке 9.19:

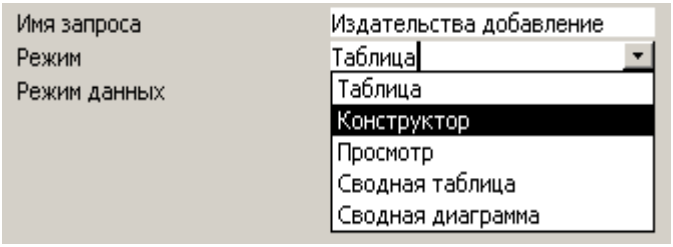


Рис. 9.19.Список режимов открытия запроса

Аргумент «Режим данных» позволяет указать, в каком режиме данных открывается запрос. Список возможных режимов данных приведен на рисунке 9.20:

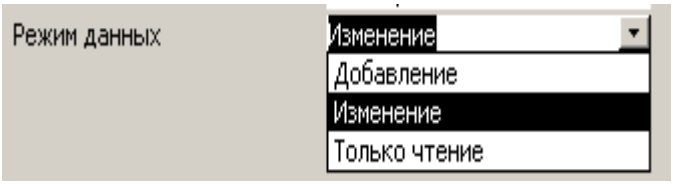


Рис. 9.20.Список режимов данных открытия запроса

Необходимо отметить, что существует несколько макрокоманд, подобных рассмотренной макрокоманде. Например, возможно использование макрокоманд «Открыть таблицу», «Открыть форму», «Открыть отчет».

Соответствующая команда выглядит следующим образом:

```
DoCmd.OpenQuery "Издательства добавление", acViewNormal, acEdit.
```

9.3.7. Макрокоманда «ПреобразоватьБазуДанных»

Эта макрокоманда позволяет реализовать импорт или экспорт объектов БД, а также реализовать связь с таблицами внешней БД. Со связанными таблицами можно работать, как и с любыми другими таблицами текущей БД.

Список аргументов макрокоманды представлен на рисунке 9.21:

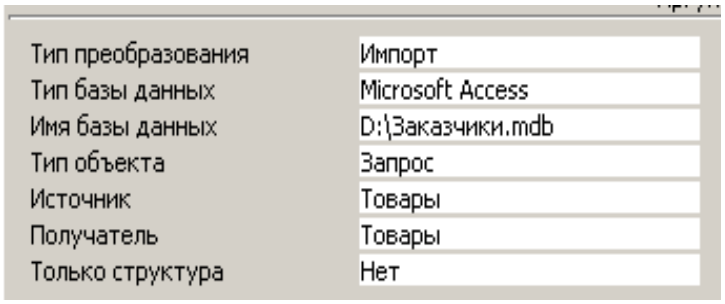


Рис. 9.21.Аргументы макрокоманды «ПреобразоватьБазуДанных»

Аргумент «Тип преобразования» может принимать следующие значения (рисунок 9.22):

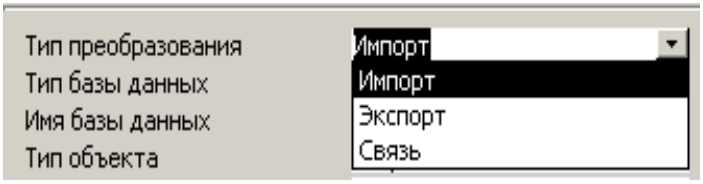


Рис. 9.22.Значения аргумента «Тип преобразования»

Аргумент «Тип базы данных» может принимать следующие значения (рисунок 9.23):

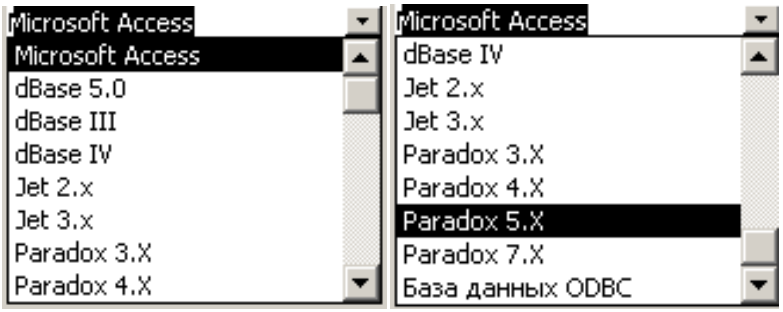


Рис. 9.23.Значения аргумента «Тип базы данных»

С помощью аргумента «Имя базы данных» задается имя БД, из которой импортируется объект в активную БД. С помощью этого же аргумента можно задать имя БД, в которую экспортируется объект активной БД. Он же позволяет задать имя БД, с которой связывается таблица текущей БД. Смысл этого аргумента определяется аргументом «Тип преобразования».

Аргумент «Тип объекта» может принимать следующие значения (рисунок 9.24):

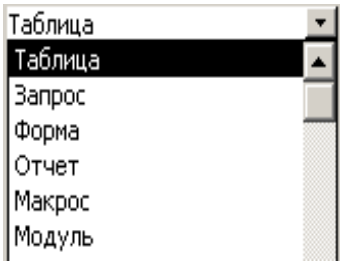


Рис. 9.24.Значения аргумента «Тип объекта»

Аргумент «Источник» позволяет указать имя объекта внешней БД.

Аргумент «Получатель» указывает имя объекта активной БД.

Аргумент «Только структура» определяет, что импортируется или экспортируется только структура объекта или структура вместе с данными.

Соответствующая команда выглядит следующим образом:

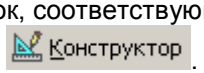
```
DoCmd.TransferDatabase acImport, "Microsoft Access", _ "D:\Заказчики.mdb", acQuery, "Товары",
"Товары", False.
```

9.4. Связывание макросов с событиями

Макрос можно запустить из базового окна СУБД Access, щелкнув по нему два раза. Кроме того, для выполнения макроса можно использовать меню «Сервис/Макрос/Выполнить макрос». Эти способы запуска макросов удобно использовать при разработке БД и отладке макросов. Для пользователей БД такие средства неприемлемы.

Для запуска макросов из интегрированной среды пользователя применяются связи макросов с событиями. Например, макрос может быть выполнен при наступлении события «Нажатие кнопки». Далее рассматривается как раз этот пример.

Пусть в форме «Книги ленточная» нам надо создать кнопку, нажатие которой позволит загрузить в формате Excel таблицу «Книги».

Для реализации этого необходимо открыть форму «Книги ленточная» в режиме конструктора, для чего надо в базовом окне среды Access выделить значок, соответствующий форме «Книги ленточная», и щелкнуть по инструменту «Конструктор» 

. В результате этих действий загрузится окно, показанное на рисунке 9.25:

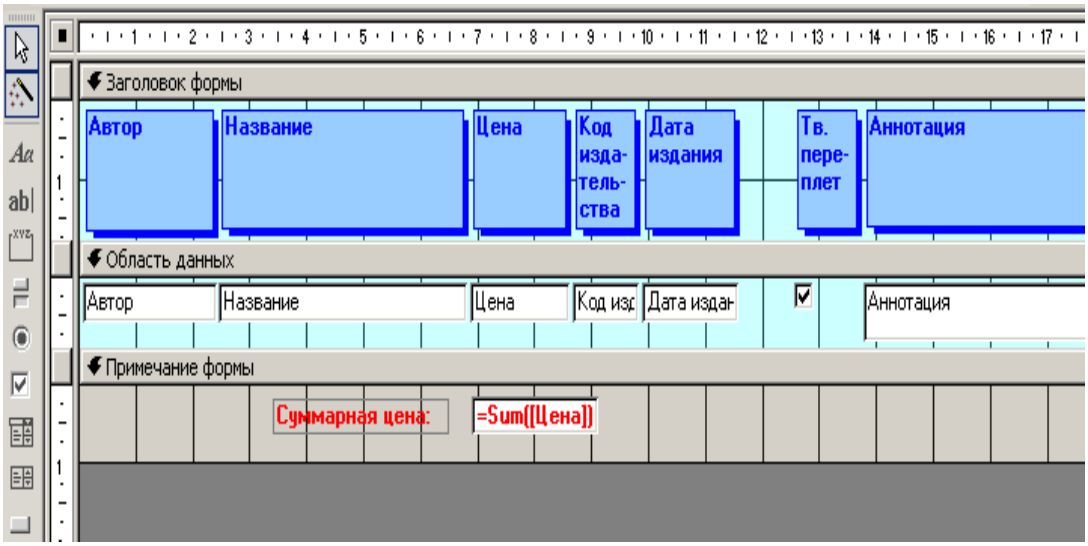


Рис. 9.25.Создание кнопки в форме «Книги ленточная»

В этом окне необходимо отключить мастера, для чего надо щелкнуть по соответствующему элементу. Мастера отключаются для того, чтобы при формировании кнопки за ней не предлагалось закрепить какие-либо функции. Затем надо щелкнуть по элементу «Кнопка» и расположить ее в нужном месте формы. В результате форма примет вид, представленный на рисунке 9.26:

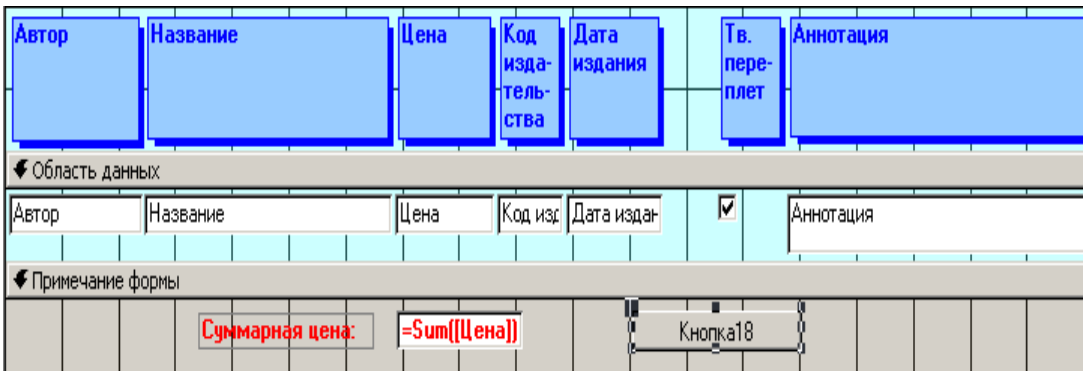


Рис. 9.26.Форма с внедренной кнопкой

Кнопку можно увеличить с помощью маркеров выделения, а затем обратиться к ее свойствам. Для того чтобы обратиться к свойствам любого объекта, можно дважды по нему щелкнуть или щелкнуть правой клавишей мыши и из контекстного меню выбрать «Свойства». Свойства сформированной кнопки приведены на рисунке 9.27:

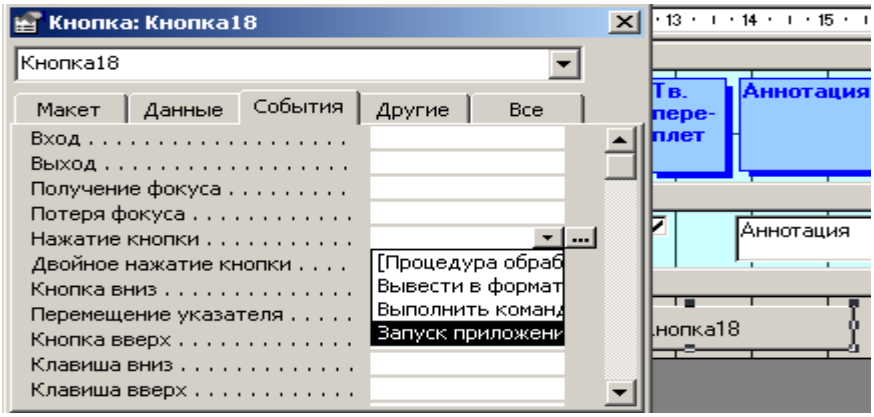


Рис. 9.27.Свойства сформированной кнопки

Свойству «Нажатие кнопки» необходимо поставить в соответствие макрос, который загружает в формате Excel таблицу «Книги». Этот макрос был предварительно сформирован под именем «Запуск приложения», поэтому необходимо щелкнуть по маркеру списка в свойстве «Нажатие кнопки» и выбрать позицию «Запуск приложения».

Если бы макрос не был предварительно создан, можно было бы в свойстве «Нажатие кнопки» щелкнуть по построителю выражения [...]. После этого загружается окно (рисунок 9.28), в котором можно сформировать выражение, макрос или программу, не выходя из конструктора формы.

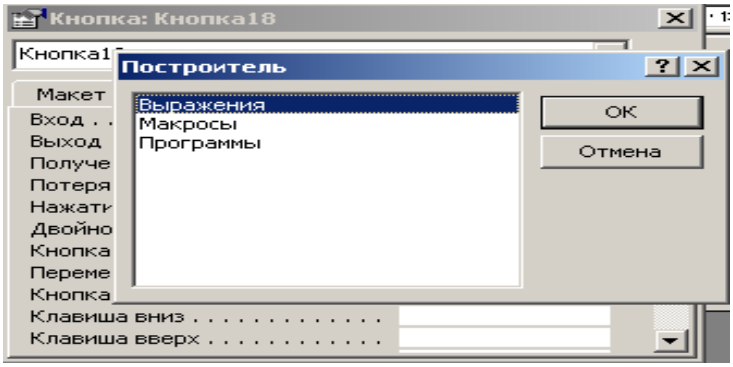


Рис. 9.28.Окно построителя выражения

Для задания информативной надписи на кнопке используется свойство «Подпись» на вкладке «Макет». Незначительное изменение кнопки позволит привести ее к виду, показанному на рисунке 9.29:

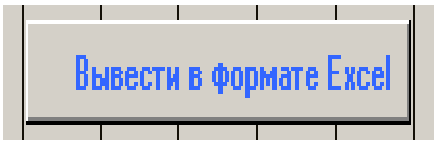


Рис. 9.29.Окончательный вид сформированной кнопки

Теперь, после сохранения отредактированной формы и открытия ее в режиме просмотра, она примет следующий вид (рисунок 9.30):

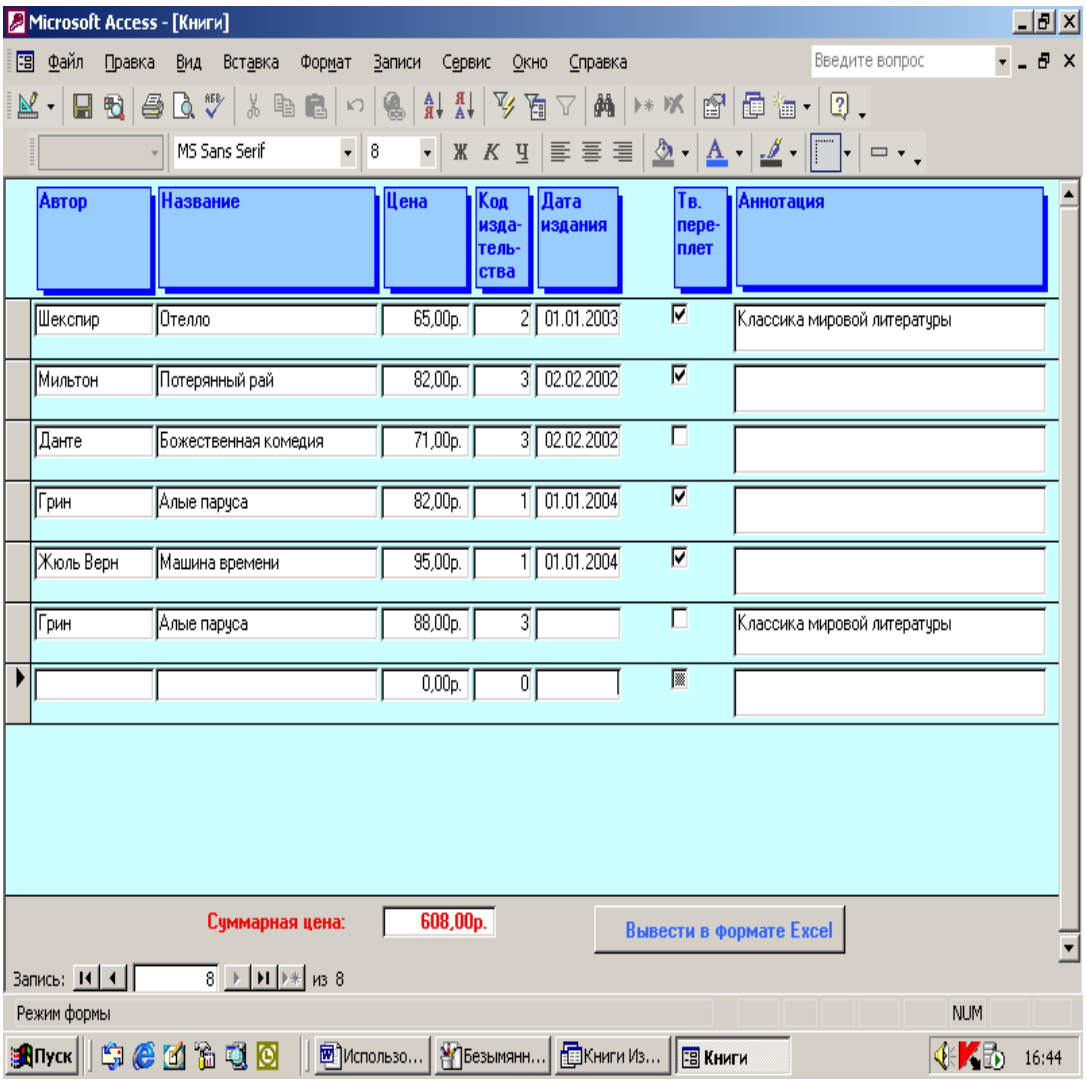
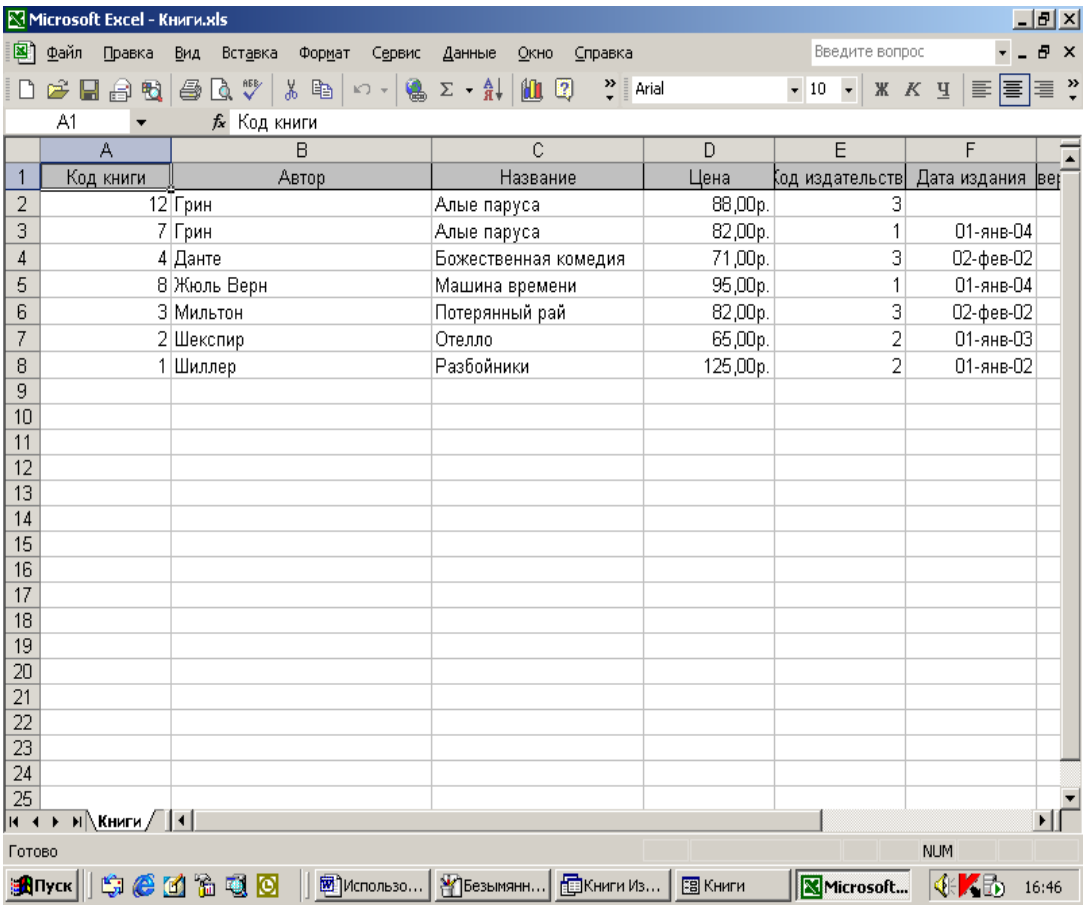


Рис. 9.30.Форма «Книги ленточная» с кнопкой

Если щелкнуть по сформированной кнопке, то загрузится окно, показанное на рисунке 9.31:



	A	B	C	D	E	F
1	Код книги	Автор	Название	Цена	Код издательств	Дата издания
2	12	Грин	Алые паруса	88,00р.	3	
3	7	Грин	Алые паруса	82,00р.	1	01-янв-04
4	4	Данте	Божественная комедия	71,00р.	3	02-фев-02
5	8	Жюль Верн	Машина времени	95,00р.	1	01-янв-04
6	3	Мильтон	Потерянный рай	82,00р.	3	02-фев-02
7	2	Шекспир	Отелло	65,00р.	2	01-янв-03
8	1	Шиллер	Разбойники	125,00р.	2	01-янв-02

Рис. 9.31.Результат нажатия на кнопку «Вывести в формateExcel»

Таким образом, сформирована и опробована связь макроса с событием.

9.5. Создание макросов с условиями

В качестве условий выполнения макросов чаще всего используются значения полей формы. Например, при вводе цены в форме «Книги» необходимо осуществлять частичную проверку ее значения. Для этого формируется макрос, показанный на рисунке 9.32:

Условие	Макрокоманда
[Forms]![Книги]![Цена]<5	Сообщение
[Forms]![Книги]![Цена]>10000	Сообщение

Рис. 9.32.Макрос проверки значения поля «Цена»

В первой макрокоманде этого макроса проверяется, меньше ли значение поля «Цена» формы «Книги», чем 5. Если это так, то выдается сообщение. Во второй макрокоманде макроса проверяется, больше ли значение этого же поля, чем 10000. Если это так, то снова выдается сообщение.

Аргументы первой макрокоманды представлены на рисунке 9.33:

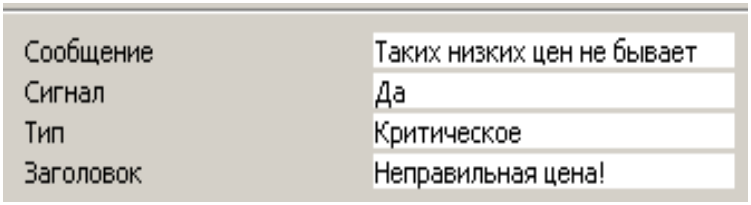


Рис. 9.33.Аргументы первой макрокоманды макроса

С помощью аргумента «Сообщение» задается текст, который выводится на экран, если условие выполняется. С помощью аргумента «Сигнал» можно предусмотреть выдачу звукового сигнала.

Типы выдаваемых сообщений представлены на рисунке 9.34:

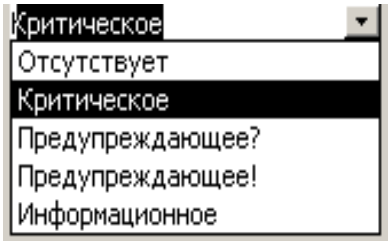


Рис. 9.34.Типы выдаваемых сообщений

С помощью аргумента «Заголовок» можно ввести текст заголовка выводимого сообщения.

Созданный макрос необходимо связать с каким-либо событием формы «Книги». Для этого она открывается в режиме конструктора (рисунок 9.35):

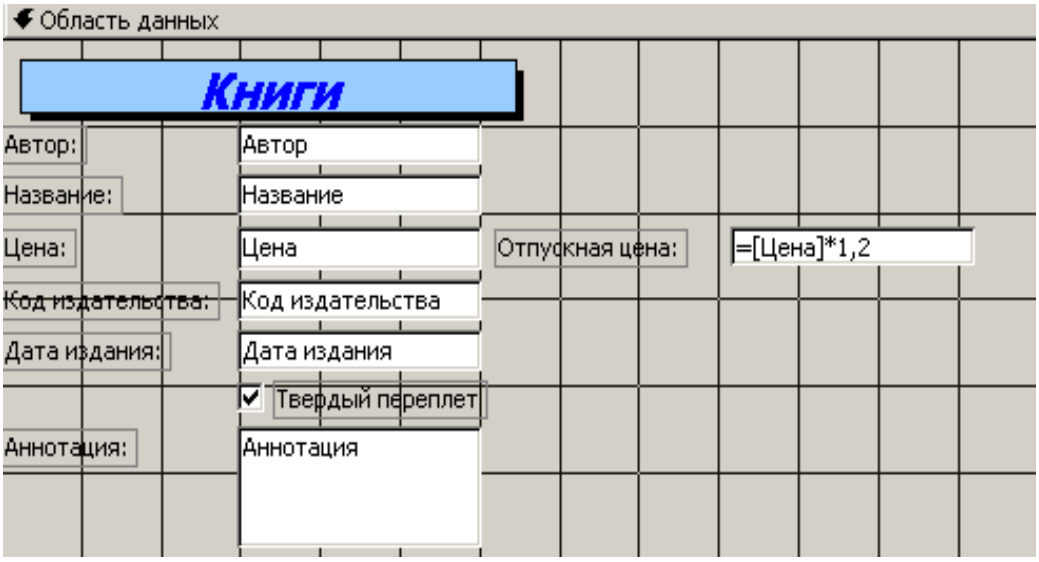


Рис. 9.35.Форма «Книги» в режиме конструктора

Логичнее всего созданный макрос связать с событием «Выход» поля «Цена» (рисунок 9.36):

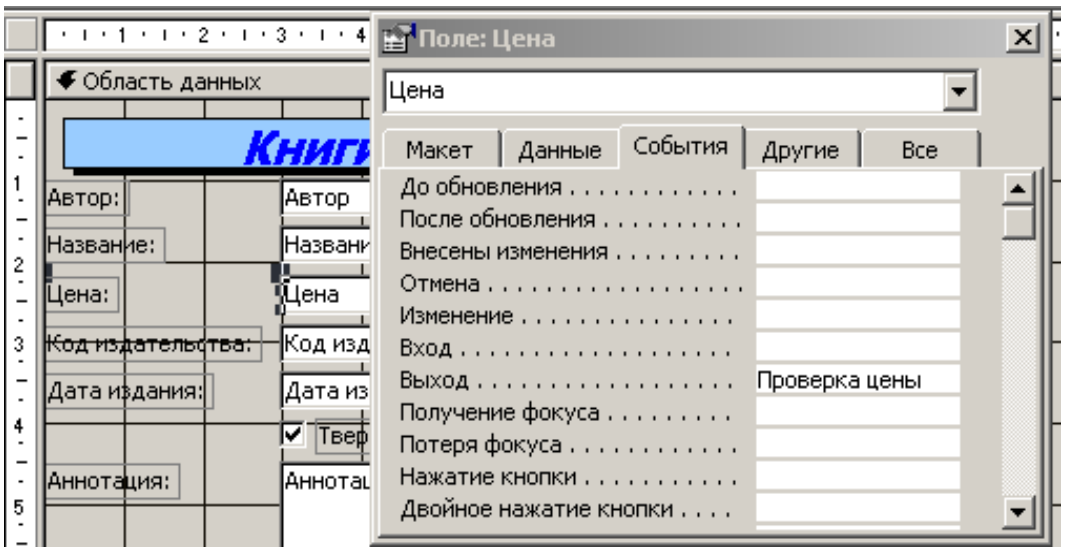


Рис. 9.36.Связывание макроса с событием «Выход» поля «Цена»

После редактирования формы и ее сохранения можно проверить правильность работы макроса. Для этого необходимо открыть форму «Книги» в режиме просмотра, затем ввести в поле «Цена» цену меньшую, чем 5 или большую, чем 10000 рублей.

Результат проверки очень низкой цены приведен на рисунке 9.37, а результат проверки очень большой цены – на рисунке 9.38.

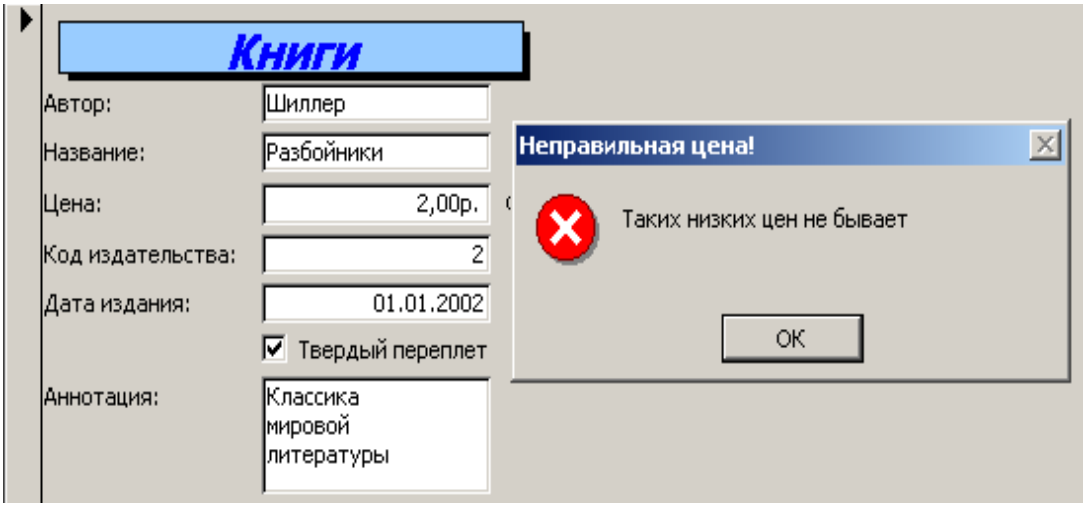


Рис. 9.37.Результат проверки очень низкой цены

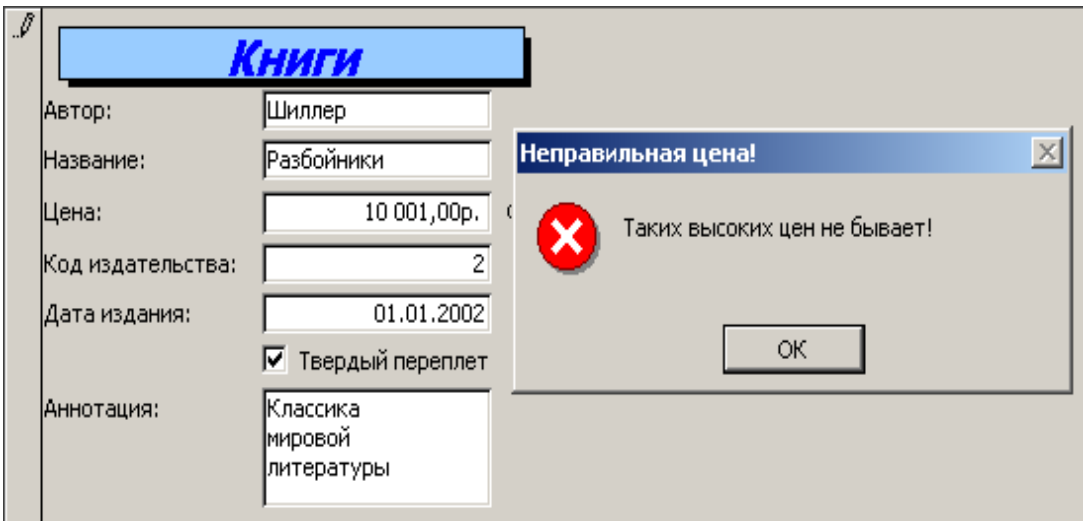


Рис. 9.38.Результат проверки очень высокой цены

Таким образом, макрос сформирован верно, и он откликается на событие выхода из поля «Цена».

Если преобразовать сформированный макрос в процедуру, то она будет выглядеть следующим образом:

```
'-------------------------------------------------------------

' Проверка_цены

'

'-------------------------------------------------------------

Function Проверка_цены()

On Error GoTo Проверка_цены_Err

If (Forms!Книги!Цена < 5) Then

Beep

MsgBox "Таких низких цен не бывает", vbCritical, "Неправильная цена!"

End If

If (Forms!Книги!Цена > 10000) Then

Beep

MsgBox "Таких высоких цен не бывает!", vbCritical, "Неправильная цена!"

End If

Проверка_цены_Exit:

Exit Function

Проверка_цены_Err:
```

MsgBox Error$

ResumeПроверка_цены_Exit

EndFunction

Здесь с помощью конструкций If (Forms!Книги!Цена < 5) и If (Forms!Книги!Цена > 10000) проверяется правильность ввода цены. Если цена введена неверно, то выдается звуковой сигнал (Beep) и выводится соответствующее сообщение посредством команды типа MsgBox "Таких низких цен не бывает", vbCritical, "Неправильная цена!"