

MS Access. Макросы

Лекции по дисциплине «Основы программирования и информационных технологий», прочитанные на кафедре ФМЭГ ФТ-факультета НТУ «ХПИ»

1. Расширение средств Microsoft Access

В основной набор средств Microsoft Access входят **макросы** (небольшие программы на языке макрокоманд системы **Access**) и **модули** (процедуры на языке **Visual Basic for Application, VBA**). С их помощью можно существенно расширить функциональные возможности создаваемого вами приложения и настроить его на нужды конкретных пользователей.

2. Рассматриваемые вопросы:

- Понятие макроса
- Использование макросов в приложениях
- Создание макросов и групп макросов
- Задание условий для управления процессом выполнения макроса
- Различные способы запуска макросов
- Советы по отладке макросов
- Обработка событий в формах и отчетах с помощью макросов
- Преобразование макросов в процедуры VBA

3. Использование макросов в базе данных Microsoft Access

С помощью макросов можно выполнить практически все действия над объектами **Access**.

Макрос в Access представляет собой **структуру**, состоящую из одной или нескольких макрокоманд, которые выполняются либо последовательно, либо в порядке, заданном определенными условиями. Набор макрокоманд в Access очень широк, с помощью макросов можно реализовать многое из того, что позволяют сделать процедуры на **VBA**.

3. Использование макросов в базе данных Microsoft Access

Каждая макрокоманда имеет определенное имя и, возможно, один или несколько аргументов, которые задаются пользователем. Например, при использовании макрокоманды **ОткрытьФорму (OpenForm)** в качестве аргументов необходимо задать, по крайней мере, имя открываемой формы и режим вывода ее на экран.

4. Макрокоманды

В нижеследующей **таблице** содержится список макрокоманд Microsoft Access, сгруппированных по категориям: **работа с данными в формах и отчетах; выполнение команд, макросов, процедур и запросов; работа с объектами; импорт/экспорт данных и объектов** и прочие. В категорию "**Прочие**" собраны все макрокоманды, которые позволяют влиять на интерфейс приложения.

4.1 Макрокоманды

Категория	Назначение	Макрокоманда
Работа с данными в формах и отчетах	Отбор данных	Применить Фильтр (ApplyFilter)
	Перемещение по данным	Следующая Запись (FindNext), Найти Запись (FindRecord), К Элементу Управления (GoToControl), На Страницу (GoToPage), На Запись (GoToRecord)

4.1 Макрокоманды

Категория	Назначение	Макрокоманда
Работа с данными в формах и отчетах	Отбор данных	Применить Фильтр (ApplyFilter)
	Обновление данных или экрана	Обновление (Requery), Показать Все Записи (ShowAllRecords)
Выполнение	Выполнение команды	Выполнить Команду (RunComraand)

4.1 Макрокоманды

Категория	Назначение	Макрокоманда
Выполнение	Выполнение макроса, процедуры или запроса	Запуск Макроса (RunMacro), Запуск Программы (RunCode), Открыть Запрос (OpenQuery), Запуск Запроса SQL (RunSQL)
	Выполнение другого приложения	Запуск Приложения (RunApp)

4.1 Макрокоманды

Категория	Назначение	Макрокоманда
Выполнение	Прерывание выполнения	Отменить Событие (CancelEvent) Остановить ВсеМакросы (StopAllMacros) ОстановитьМакрос (StopMacro)
	Выход из Microsoft Access	Выход (Quit)

4.1 Макрокоманды

Категория	Назначение	Макрокоманда
Импорт/экспорт	Передача объектов Microsoft Access в другие приложения	Вывести В Формате (OutputTo), Отправить Объект (SendObject)
12.04.2013	Преобразование формата данных доц. Шкалето В.И.	Преобразовать Базу Данных (Transfer Database), Перенос Базы Данных SQL (Transfer SQLDatabase), Преобразовать Электронную Таблицу (Transfer Spreadsheet), Преобразовать Таблицу

4.1 Макрокоманды

Категория	Назначение	Макрокоманда
Работа с объектами	Копирование, переименование и сохранение объекта	Копировать Объект (CopyObject), Копировать Файл Базы Данных (CopyDatabase File), Переименовать (Rename), Сохранить (Save)
Прочие	Вывод на экран или скрытие встроенной или специальной панели инструментов Подача звукового сигнала	Панель Инструментов (ShowToolbar), Сигнал (Beep)

4.2 Макрокоманды

Как видно из этой таблицы, действия, которые могут быть выполнены с помощью макросов, очень разнообразны. Даже если вы не будете использовать макросы в своих приложениях, имеет смысл изучить эти макрокоманды, поскольку их можно использовать и в процедурах VBA

4.2 Макрокоманды

Использование макросов оправдано тем, что их легко создавать. Для создания макроса требуется знать лишь основные приемы работы в Microsoft Access и Windows, такие как перетаскивание объектов из окна **База данных (Database)** в специальное окно — **Конструктор макросов (Macro Design)**, выбор действия из списка и ввод выражений в качестве аргументов макрокоманды. Применяя макросы можно получить достаточно функциональное приложение.

4.2 Макрокоманды

Основное назначение макросов — это создание удобного интерфейса приложения: чтобы формы и отчеты открывались при нажатии кнопок в форме или на панели инструментов или же привычным выбором команды меню; чтобы при открытии приложения пользователь видел на экране не окно **База данных (Database)**, наполненное множеством таблиц, запросов, форм и отчетов, а некую понятную форму, с помощью которой можно было бы сразу производить желаемые действия и т. д.

С помощью макросов можно создать удобные меню и панели инструментов для приложения.

4.3 Недостатки использования макросов

Однако использование макросов имеет и некоторые недостатки:

- Возможности макрокоманд ограничены по сравнению с возможностями языка **VBA**, поэтому в ряде случаев без программирования на **VBA** не обойтись. Язык **VBA** предоставляет более широкие возможности для работы с данными, позволяет использовать механизм программирования объектов для связи с другими приложениями, вызывать функции из библиотек динамической загрузки (**DLL**) **Windows** и создавать собственные специализированные функции.

4.3 Недостатки использования макросов

- Макросы можно использовать практически везде, где применяются процедуры **VBA**, однако процедуры **VBA**, как правило, выполняются быстрее.
- Макросы являются объектами, существующими отдельно от форм и отчетов, в которых они используются, поэтому, когда этих объектов становится очень много, их поддержка достаточно трудоемка. Процедуры обработки событий **VBA** являются неотъемлемой частью форм и отчетов, и в этом есть свои преимущества. Например, при переносе форм и отчетов из одной базы данных в другую с ними автоматически переносятся связанные процедуры.

5. Создание макросов

В Microsoft Access существует специальное средство для создания макросов — **Конструктор макросов (Macro Design)**. Рассмотрим процесс создания макроса с помощью **Конструктора**.

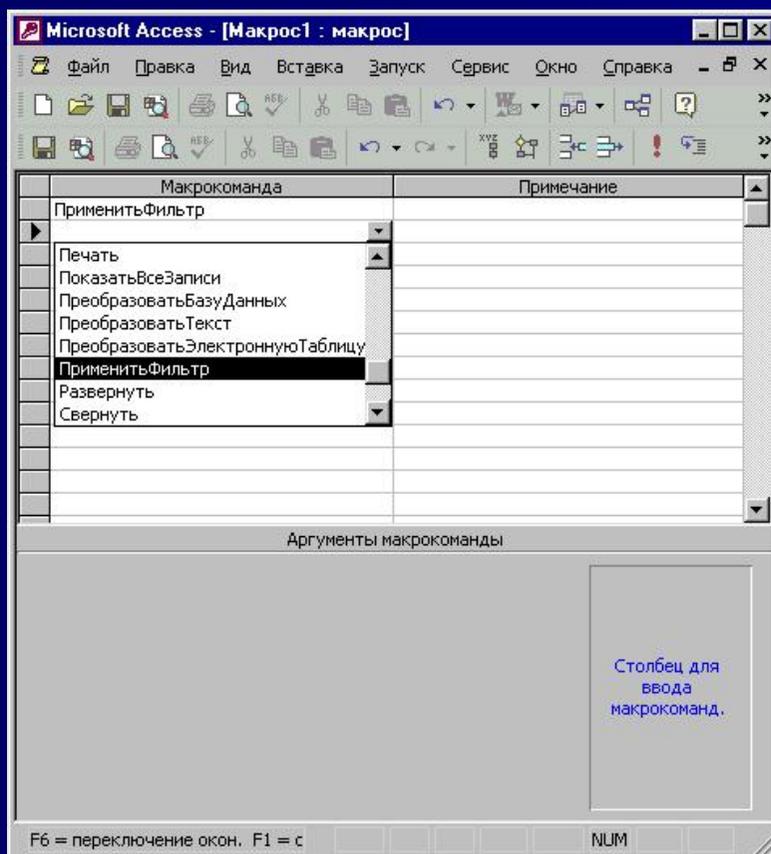
5.1 Конструктор макросов

Открыть окно **Конструктор макросов** (Macro Design) для создания нового макроса можно двумя способами:

1. Раскройте список макросов в окне **База данных (Database)**, щелкнув левой кнопкой мыши по ярлыку **Макросы (Macros)**, и нажмите кнопку **Создать (New)** в верхней части окна.

2. Щёлкните по стрелке вниз на кнопке **Новый объект (New Object)** на панели инструментов **База данных (Database)** и выберите элемент **Макрос (Macros)**.

5.1 Конструктор макросов



Окно **Конструктора** макросов построено аналогично окну **Конструктора таблиц**, т. е. разделено по горизонтали на две части: **панель описаний** и **панель аргументов**.

Для быстрого переключения между панелями описаний и аргументов, так же как при работе с таблицами и запросами в режиме Конструктора, можно использовать клавишу <F6>.

5.2 Окно Конструктора

Верхняя часть окна **Конструктора** — панель описаний — состоит из нескольких столбцов. По умолчанию на этой панели выводится два столбца: "**Макрокоманда**" (**Action**) и "**Примечание**" (**Comments**). Панель описаний позволяет определить последовательность макрокоманд, из которых состоит макрос.

- Строка в столбце "**Макрокоманда**" (**Action**) представляет собой поле со списком, в котором можно выбрать нужную макрокоманду.
- Строка в столбце "**Примечание**" (**Comments**) — это обычное текстовое поле, в которое можно ввести комментарий, описывающий выполняемое действие.

5.3 Панель аргументов

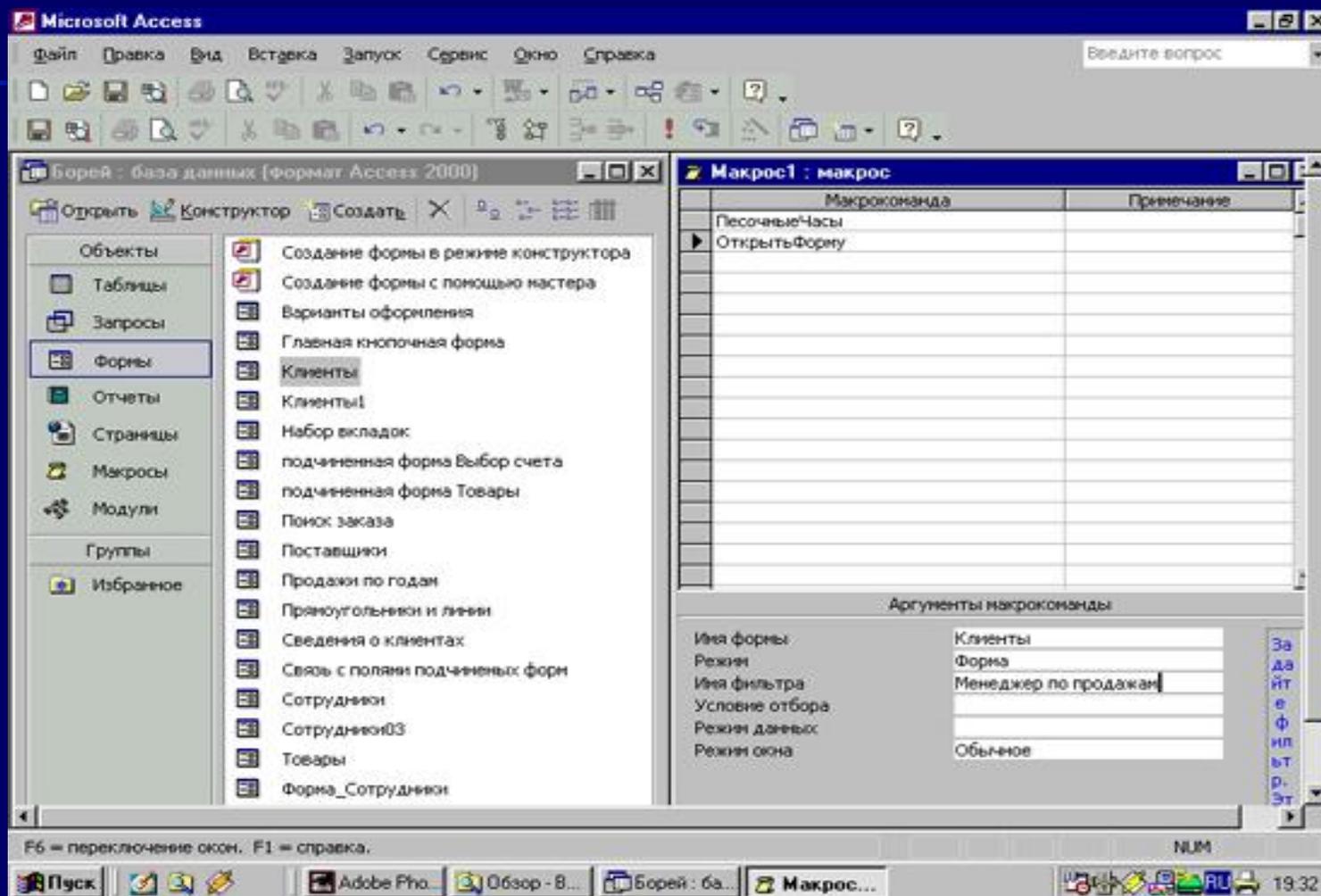
Когда поле "**Макрокоманда**" (**Action**) заполнено, в нижней части окна **Конструктора** макросов появляется **панель аргументов**, предназначенная для ввода значений аргументов соответствующей макрокоманды. **Список полей** на этой панели зависит от выбранной макрокоманды и может отсутствовать, если макрокоманда не имеет аргументов. Таким образом, при создании макросов не нужно запоминать список аргументов для каждой макрокоманды.

5.4 Простой способ ввода макрокоманды

Допустим, нужно создать макрос, который должен открывать форму "**Клиенты**" (Customers) в приложении "**Борей**" (Northwind). Для этого в окне **База данных** (Database) необходимо щелкнуть по ярлыку **Формы** (Forms), затем, открыть окно **Конструктор макросов** (Macro Design) и перетащить с помощью мыши форму "**Клиенты**" (Customers) в поле "**Макрокоманда**" (Action) окна **Конструктора макросов**.

*Для удобства перетаскивания объектов из окна **База данных** (Database) в форму выберите команду **Окно, Слева направо** (Window, Tile Vertically), чтобы расположить окна на экране рядом.*

Перетаскивание – способ создания макроса



5.4.1 Перетаскивание – способ создания макроса

Тогда в поле "**Макрокоманда**" (Action) автоматически появится макрокоманда **ОткрытьФорму** (OpenForm), а на панели аргументов автоматически заполнятся поля: в поле "**Имя формы**" (Form Name) появится имя формы "**Клиенты**" (Customers), в поле "**Режим**" (View) — значение по умолчанию **Форма** (Form), в поле "**Режим окна**" (Window Mode) — значение **Обычное** (Normal). Остальные поля останутся пустыми. Мы можем в этой же макрокоманде задать фильтр для отбора записей. Для этого введем в поле "**Имя фильтра**" (Filter Name) имя какого-нибудь фильтра, построенного ранее.

5.4.2 Перетаскивание – способ создания макроса

Кроме того, можно сделать записи в этой форме недоступными для изменений, для чего в поле **Режим данных** (Data Mode) необходимо ввести значение **Только чтение** (Read Only). Аналогично можно создавать макрокоманды, перетаскивая из окна **База данных** (Database) таблицы, запросы, отчеты и другие макросы. При этом в окне **Конструктора макросов** будут появляться соответствующие макрокоманды: **ОткрытьТаблицу** (**OpenTable**), **ОткрытьЗапрос** (**OpenQuery**), **ОткрытьОтчет** (**OpenReport**), **ЗапускМакроса** (**RunMacro**).

5.4.3 Ввод аргументов

Для ввода аргументов макрокоманды чаще всего требуется выбирать значения из списков или вводить выражения. Для ввода выражений можно воспользоваться **Построителем выражений**, кнопка которого находится справа от поля аргумента. Как и в других случаях, перед выражением нужно ставить знак равенства (=). Исключения составляют аргумент **Выражение** (Expression) макрокоманды **ЗадатьЗначение** (Setvalue) и аргумент **Число повторений** (Repeat Count) макрокоманды - **ЗапускМакроса** (RunMacro). Если ввести знак равенства перед выражением, задающим значение этих аргументов, то оно будет вычисляться дважды, что может привести к нежелательным результатам.

6. Создание групп макросов

При разработке приложения с использованием макросов количество макросов может оказаться очень большим. Поэтому важно хорошо организовать доступ к нужным макросам. Для этого в одном объекте **Макрос (Macro)** можно объединить несколько макросов. Например, рекомендуется все макросы, связанные с событиями в форме или отчете, объединить в отдельный объект, соответствующий данной форме или отчету. Для того чтобы можно было это сделать, каждый макрос группы должен иметь свое имя, а имя объекта **Макрос (Macro)** будет являться именем группы макросов.

6.1 Создание групп макросов

Чтобы создать группу макросов:

1. Откройте Конструктор макросов одним из тех способов, которые были описаны в предыдущем разделе.

2. Нажмите кнопку **Имена макросов** (**Macro Name**) на панели инструментов. На панели описаний в окне Конструктора появится еще один столбец — **"Имя макроса"** (**Macro Name**)

6.2 Создание групп макросов

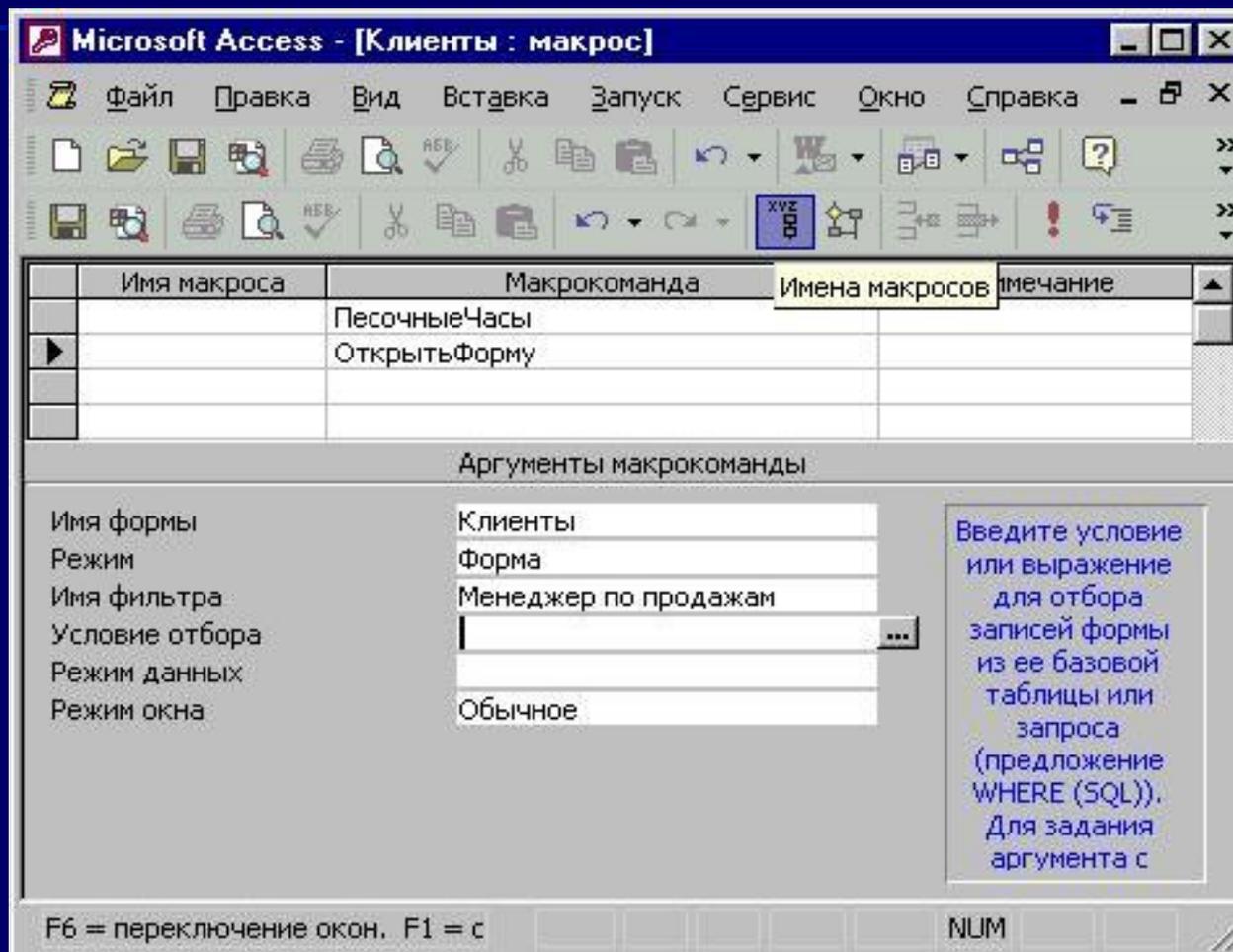
3. В этот столбец введите имя первого макроса. Остальные поля данной строки лучше оставить пустыми — это облегчит перемещение и копирование макрокоманд.

4. Начиная со следующей строки, введите все макрокоманды макроса и соответствующие аргументы для каждой макрокоманды.

5. Пропустите одну строку.

6. Повторите шаги 3—5 для каждого макроса.

6.3 Создание групп макросов



6.4 Создание групп макросов

При обработке макроса **пустые строки игнорируются**, поэтому их удобно использовать для **разделения макросов в группе**. Этот прием, а также ввод **комментариев** для сложных макросов, состоящих из большого числа макрокоманд, позволяет улучшить читаемость макросов.

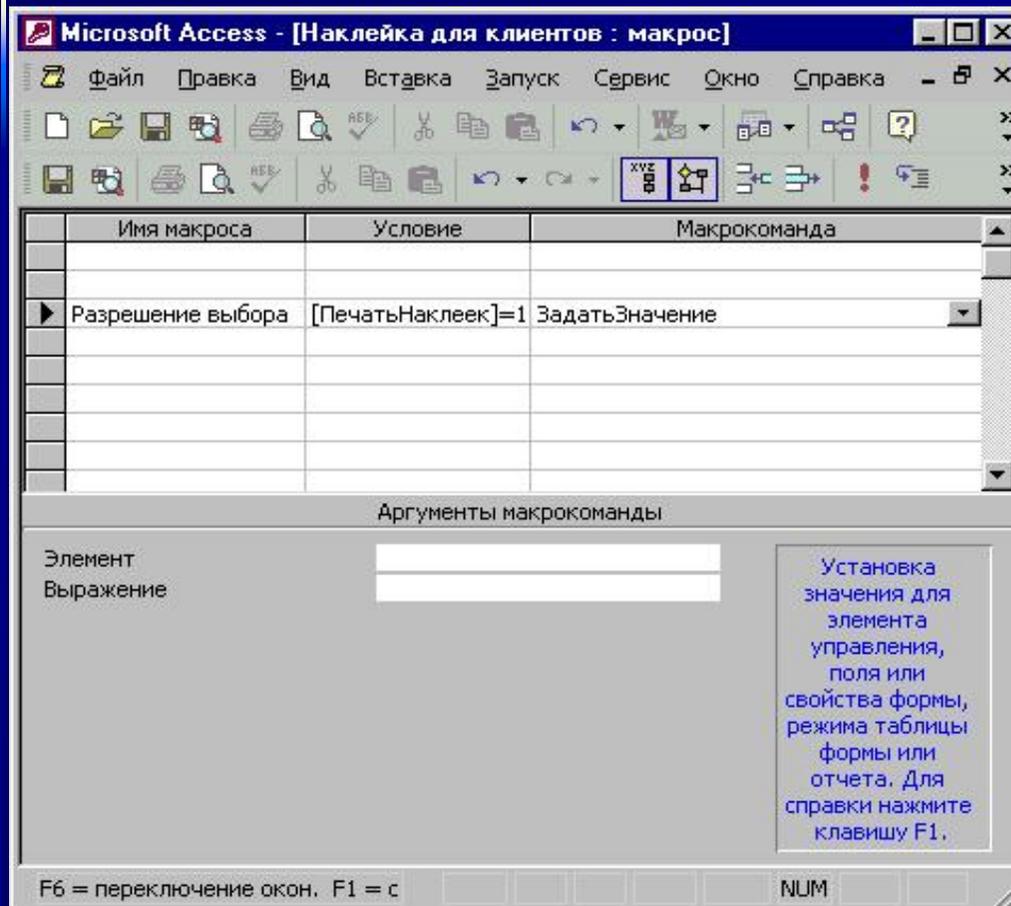
Для **запуска одного из макросов группы** используется полное имя для ссылки на макрос. Полное имя макроса формируется таким образом: **имяГруппы.имяМакроса**.

7. Применение условий в макросах

В примерах макросов, которые приводились ранее, использовался последовательный порядок выполнения макрокоманд.

Однако, как и в любой программе, порядок выполнения макрокоманд в макросе может быть изменен. Для этого вводятся условия выполнения или пропуска макрокоманд. Эти условия задаются в виде выражений в специальном столбце, который появляется в окне **Конструктора** макросов, если нажата кнопка **Условия** (**Conditions**) на панели инструментов.

7.1 Применение условий в макросах



*Как и в случае с группами макросов, можно настроить параметры системы таким образом, чтобы этот столбец появлялся в **Конструкторе** макросов всегда.*

7.2 Применение условий в макросах

На рисунке приведено окно **Конструктора** для группы макросов "**Наклейки для клиентов**" (**Customer Label Dialog**), используемой для печати отчета в виде наклеек на конверты. Причем можно напечатать наклейки сразу для всех клиентов или для клиентов одной страны, выбранной из списка. В окне виден столбец "**Условие**" (**Condition**). Если условие, заданное в этом столбце, истинно, выполняется макрокоманда, находящаяся в этой строке. Если условие ложно, соответствующая макрокоманда пропускается и выполняется следующая.

7.3 Применение условий в макросах

Если требуется при истинности условия выполнить сразу несколько макрокоманд, то для всех макрокоманд, кроме первой, в столбце "**Условие**" (**Condition**) ставится многоточие (...). Условие, которое должно проверяться, пишется в строке первой макрокоманды. Тогда, если оно истинно, выполняется весь набор макрокоманд от этого условия и до следующей макрокоманды с заданным условием, до следующего макроса или до конца макроса. Если условие ложно, пропускаются все команды, помеченные многоточием, включая макрокоманду с заданным условием. Далее выполняется макрокоманда, следующая за пропущенными, в которой содержится новое условие или поле "**Условие**" (**Condition**) не заполнено.

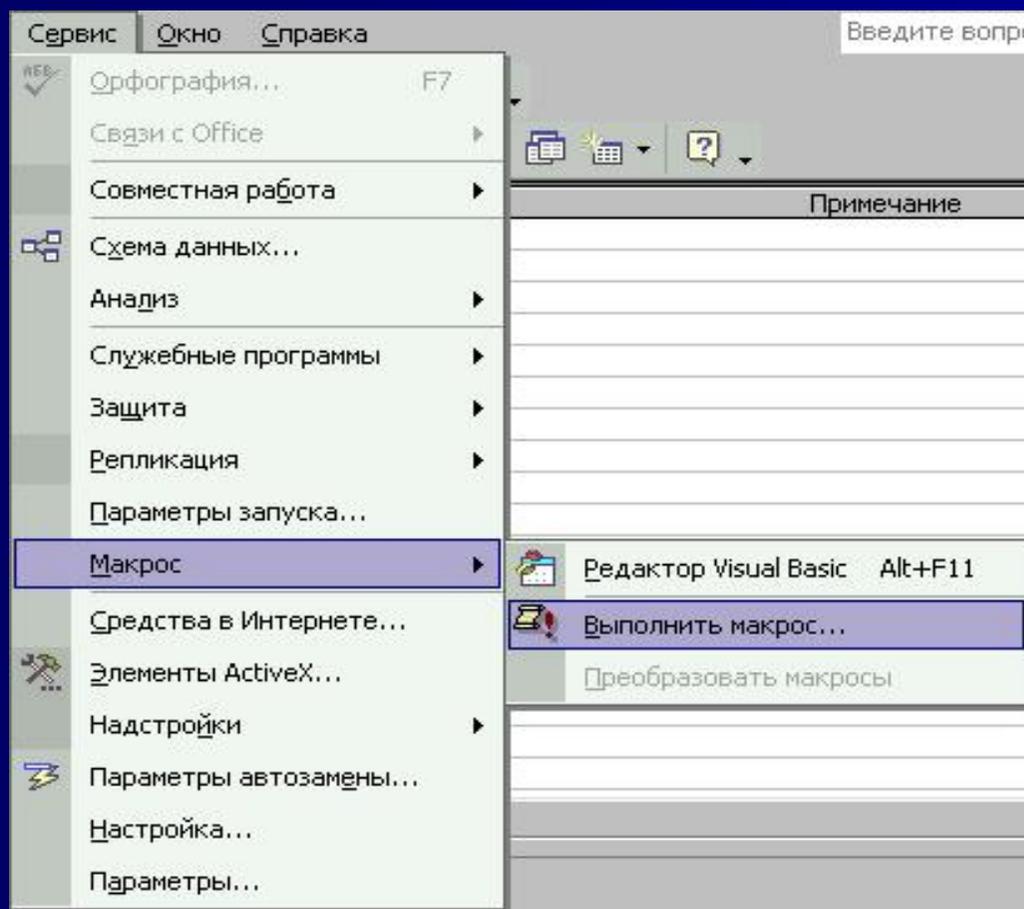
7.4 Применение условий в макросах

Таким образом, в отличие от большинства языков программирования, в макросах нет альтернативного ветвления. Для того чтобы создать макрос с двумя альтернативными ветвями, нужно сначала ввести условие и определить макрокоманды, исполняемые при выполнении этого условия. Сразу после них необходимо указать обратное условие и определить макрокоманды, исполняемые в альтернативной ветви. Затем можно ввести макрокоманды, которые исполняются безусловно.

8. Выполнение и отладка макросов

Способов запуска макросов очень много. Среди них есть прямые способы, когда макрос запускается непосредственно пользователем. Эти способы используются обычно при тестировании макросов. Когда макрос отлажен, создают специальную команду меню или кнопку на панели инструментов или связывают макрос с каким-либо событием в форме или отчете.

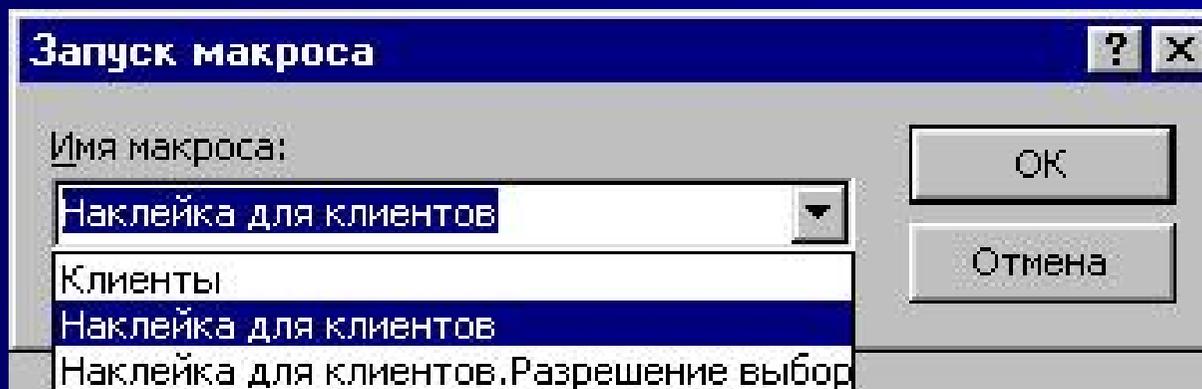
8.1 Выполнение и отладка макросов



Универсальным способом запуска макроса является использование команды меню **Сервис, Макрос, Выполнить макрос** (Tools, Macro, Run Macro).

8.2 Выполнение и отладка макросов

В появившемся при выборе этой команды диалоговом окне нужно указать (ввести или выбрать из списка) полное имя макроса, т. е. **имяГруппы.имяМакроса**. Этот способ позволяет выполнить макрос из любого активного окна, будь то **окно макросов, окно базы данных** или, например, **Конструктор форм**.



8.3. Запуск макроса из окна базы данных

Чтобы запустить макрос из окна **База данных (Database)**, щелкните по ярлыку **Макросы (Macros)** на панели объектов, найдите в списке нужный макрос и дважды щелкните по нему кнопкой мыши или нажмите кнопку **Запуск (Run)** на панели инструментов окна **База данных**. Последний способ подходит, только если в выбранном объекте содержится один макрос. В противном случае нужно воспользоваться универсальным способом выполнения макроса, описанным выше.

8.4 Запуск макроса с помощью кнопки на панели инструментов

Прямые способы запуска макросов являются простыми, но не самыми быстрыми. Существуют более удобные и быстрые способы. Наиболее удобный способ запуска макроса — это создание специальной кнопки на панели инструментов. Правда, стандартных панелей инструментов в Access много, и в разных режимах работы на экране появляются соответствующие панели. Поэтому наиболее разумным является создание специальной панели инструментов, на которую вы можете поместить все нужные кнопки. Тем не менее операция добавления специальной кнопки на стандартную панель инструментов чрезвычайно проста.

8.5.1 Запуск макроса с помощью кнопки на панели инструментов

Чтобы добавить кнопку на стандартную панель инструментов Access:

1. Щелкните правой кнопкой мыши по панели инструментов и выберите в контекстном меню команду **Настройка (Customize)**.

2. В появившемся диалоговом окне **Настройка (Customize)** раскройте вкладку **Панели инструментов (Toolbars)**.

3. Установите флажок для одной из стандартных панелей инструментов, например **Запрос в режиме таблицы (Query Datasheet)**, чтобы она отобразилась на экране.

8.5.2 Запуск макроса с помощью кнопки на панели инструментов

4. Закройте диалоговое окно **Настройка (Customize)**, нажав кнопку **Заккрыть (Close)**.
5. Щелкните по ярлыку **Макросы (Macros)** на панели объектов в окне **База данных (Database)** и найдите нужный макрос.
6. Перетащите макрос с помощью мыши на панель инструментов **Запрос в режиме таблицы (Query Datasheet)**.

8.5.3 Создание специальной панели инструментов

Теперь рассмотрим создание специальной панели инструментов и размещение на ней кнопки вызова макроса.

Щелкните по ярлыку **Макросы** (Macros) на панели объектов в окне **База данных** (Database) и выделите нужный макрос.

Выберите команду **Сервис, Макрос, Создать панель инструментов из макроса** (Tools, Macro, Create Toolbar from Macro). На экране появляется пустая панель инструментов.

Перетащите с помощью мыши нужный макрос на эту панель.

8.5.4 Создание специальной панели инструментов

На панели появляется кнопка с уже знакомым значком макроса. Кнопка получает такое же название, как имя макроса, для которого она создана.

Если нужно создать панель инструментов для вызова нескольких макросов, мы рекомендуем объединить эти макросы в одну группу (создание группы макросов было описано в предыдущем разделе). Тогда при выполнении команды меню **Сервис, Макрос, Создать панель инструментов из макроса** (Tools, Macro, Create Toolbar from Macro) будет создана панель инструментов, на которой каждому макросу группы будет соответствовать своя кнопка.

8.5.5 Создание специальной панели инструментов

При этом на кнопках отображаются имена соответствующих макросов. Новая панель инструментов получает такое же имя, как имя группы макросов. Попробуйте выполнить эту команду для одной из групп макросов в базе данных "Борей" (Northwind), например "Клиенты" (Customers). При желании можно заменить текст на кнопках значками.

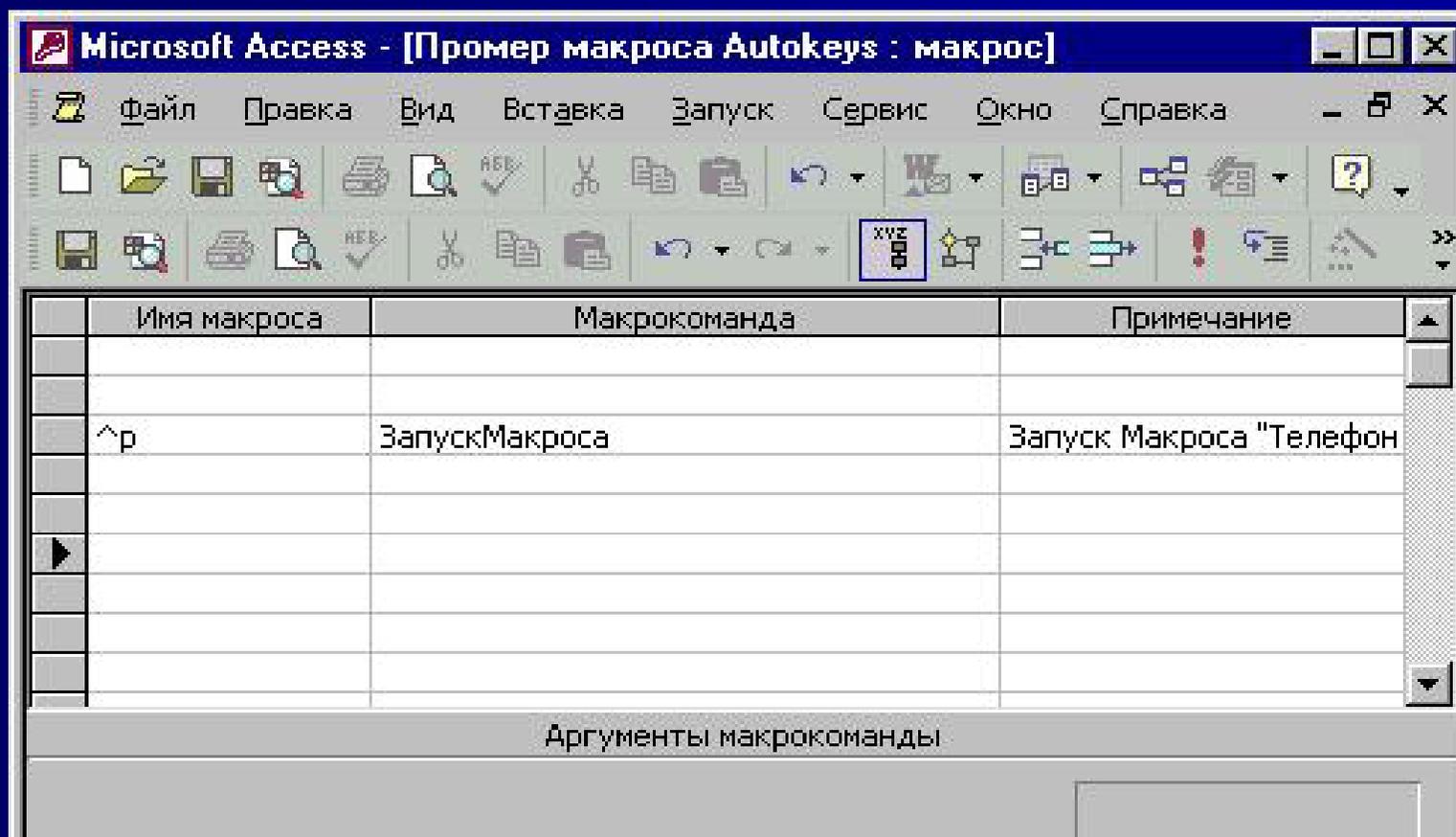
8.6 Запуск макроса с помощью команды меню

Еще одним распространенным способом запуска макроса является запуск с помощью специально созданной для этого команды меню. Способы создания команд меню для запуска макросов очень похожи на те, что применяются для создания кнопок на панели инструментов. Например, можно воспользоваться для этого командой меню **Сервис, Макрос, Создать меню из макроса** (Tools, Macro, Create Menu from Macro).

8.7 Запуск макроса с помощью комбинации клавиш

Для запуска макроса можно назначить комбинацию клавиш. Для этого необходимо создать специальную группу макросов — "**AutoKeys**". Эта группа макросов должна для каждой назначенной вами комбинации клавиш содержать макрокоманду запуска соответствующего макроса. Пример группы макросов "**AutoKeys**" приведен в базе данных "**Борей**". Откройте эту группу макросов в режиме **Конструктора**. Она содержит один макрос.

8.7 Запуск макроса с помощью комбинации клавиш



8.7.1 Запуск макроса с помощью комбинации клавиш

«**Имя макроса**» — это запись комбинации клавиш **<Ctrl>+<P>**. Сам макрос состоит из одной макрокоманды **ЗапускМакроса** (RunMacro), которая запускает макрос "Телефоны клиентов. Печать" (Customer Phone List.Print). Макрос "**AutoKeys**" просматривается каждый раз, когда пользователь вводит специальные комбинации клавиш, например **<Ctrl>+<P>**. Если введенная комбинация клавиш найдена в "**AutoKeys**", то запускается соответствующий макрос.

8.7.2 Запуск макроса с помощью комбинации клавиш

Этот способ запуска макроса, безусловно, является самым быстрым, однако он имеет существенный **недостаток**: свободных комбинаций клавиш очень немного. И хотя комбинации, определенные в "**AutoKeys**", имеют более высокий приоритет, чем стандартные (например, **<Ctrl>+<C> — копировать**), заменять стандартное назначение комбинаций клавиш не рекомендуется. Разрешенные комбинации клавиш приведены в нижеследующей таблице. Комбинации с клавишей **<Alt>** в данном случае не применяются, т. к. используются для запуска команд меню и нажатия кнопок.

8.7.2 Запуск макроса с помощью комбинации клавиш

Обозначение	Комбинации клавиш
^A или ^4	<Ctrl> + буква или <Ctrl> + цифра
{F1}	Любая функциональная клавиша
^{F1}	<Ctrl> + любая функциональная клавиша

8.7.2 Запуск макроса с помощью комбинации клавиш

Обозначение	Комбинации клавиш
+ {F1}	<Shift> + любая функциональная клавиша
{Insert}	<Ins>
^{Insert}	<Ctrl> + <Ins>
+{Insert}	<Shift> + <Ins>

8.7.2 Запуск макроса с помощью комбинации клавиш

Обозначение	Комбинации клавиш
{Delete} или {Del}	
^{Delete} или ^{De1}	<Ctrl> +
+{Delete} или +{Del}	<Shift> +

8.8. Запуск макроса при открытии базы данных

При открытии базы данных Microsoft Access могут выполняться некоторые действия. Чаще всего это открытие специальной формы, так называемой **Главной кнопочной формы**, вывод **специальных меню** или **панелей инструментов**, скрывание **стандартных меню** и т. д. Для определения этих действий используется диалоговое окно **Параметры запуска (Startup)**. Однако иногда требуется при запуске приложения выполнить более сложный набор действий, чем тот, что позволяет задать это диалоговое окно. Например, может потребоваться заранее открыть несколько форм (не показывая их на экране), чтобы потом их отображение не занимало много времени, проверить некоторые условия или запросить ввод некоторых данных.

8.8.1 Запуск макроса при открытии базы данных

Все это можно сделать с помощью специального макроса, который называется "**AutoExec**". При открытии базы данных Access проверяет наличие этого макроса и, если он существует, выполняет его. При создании макроса "**AutoExec**" следует помнить, что **Access** сначала выполнит действия, определенные в окне **Параметры запуска (Startup)**, а затем — макрос "**AutoExec**", поэтому в них не должно быть противоречивых действий.

8.9 Запуск макроса из другого макроса

Иногда требуется вызвать макрос из другого макроса. Это можно сделать с помощью **макрокоманды ЗапускМакроса (RunMacro)**. Мы уже встречались с этой макрокомандой при описании макроса "**AutoKeys**". Здесь нужно отметить, что эта макрокоманда имеет три аргумента: кроме имени макроса, задается число повторов выполнения макрокоманды и условие повтора. Таким образом, эта макрокоманда позволяет организовывать циклы. Аргумент **Число повторов (Repeat Count)** задает количество вызовов макроса. Аргумент **Условие повтора (Repeat Expression)** является выражением, которое может принимать значение **Истина** или **Ложь**.

8.9.1 Запуск макроса из другого макроса

Перед выполнением макрокоманды **ЗапускМакроса** (`RunMacro`) проверяется значение этого выражения. Если оно **Истина**, то макрос выполняется, если **Ложь**, то макрос не выполняется и управление передается следующей макрокоманде. Если эти два аргумента не заданы, макрос выполняется только один раз. Если заданы оба эти аргумента, цикл вызова прекращается, когда макрос выполнился заданное число раз либо когда заданное условие окажется невыполненным и получит значение **Ложь**.

8.10 Назначение макроса событию

Наиболее часто макросы используются в приложении Access для обработки событий. **Событие** — это любое действие, распознаваемое объектом, и можно определить реакцию объекта на событие. События происходят в результате действий пользователя, выполнения инструкций **VBA** или генерируются системой. Примером событий является вывод на экран формы, отчета, ввод данных в текстовое поле, нажатие кнопки мыши или клавиши. Каждому из этих событий можно назначить макрос или процедуру **VBA**, которые будут автоматически выполняться в ответ на произошедшее событие.

8.10.1 Назначение макроса событию

Практически все программирование в Access сводится к написанию макросов или процедур, обрабатывающих события, т. е. программируется реакция объектов на события. Существует большое количество различного рода событий, на которые реагируют объекты, причем часто возникает не одно, а целая последовательность событий. Поэтому необходимо обладать определенным умением, чтобы решить, какому событию следует назначить созданный вами макрос или процедуру **VBA**. Рассмотрим **способ назначения макроса событию** и приведем примеры обработки событий с помощью макросов.

8.10.2 Назначение макроса событию

Вначале откроем форму "**Клиенты**" (Customers) в базе данных "**Борей**". В этой форме отображается информация о клиенте. Допустим, мы хотели бы видеть не только эту информацию, но и данные о купленных клиентом продуктах. Было бы хорошо создать кнопку **Заказы клиента**, при нажатии которой появлялась бы форма "**Заказы**" (Orders) с заказами только того клиента, который в данный момент выбран в форме "**Клиенты**" (Customers). Для того чтобы получить желаемое, создадим макрос, который будет выполняться, когда произойдет событие **Нажатие кнопки** (On Click) в форме "**Клиенты**" (Customers).

8.10.3 Назначение макроса событию

Чтобы создать макрос, определяющий описанную реакцию приложения на событие **Нажатие кнопки (On Click)**:

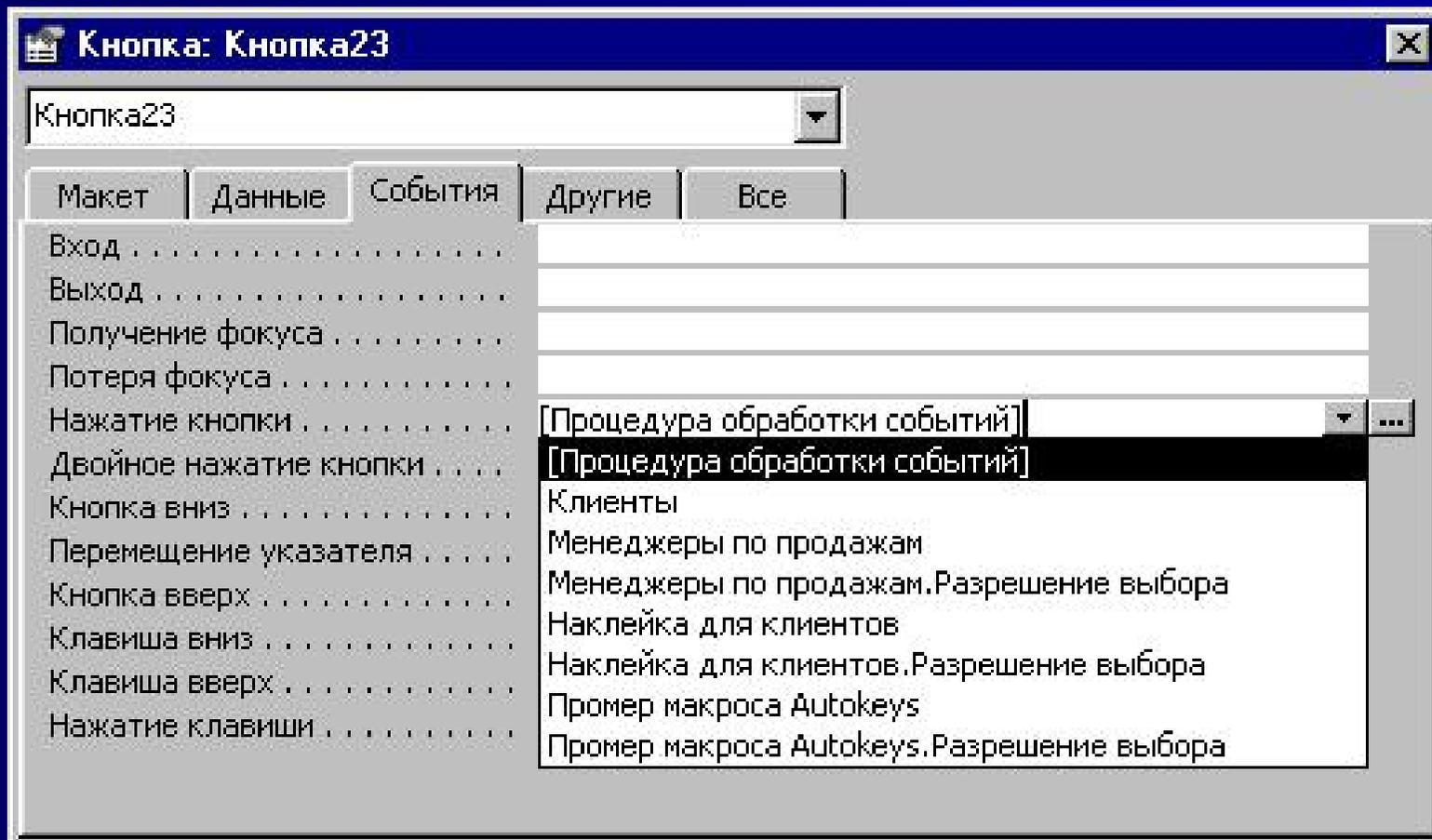
1. Откройте форму "**Клиенты**" (**Customers**) в режиме **Конструктора**.
2. Создайте кнопку в области заголовка формы. При этом **кнопка мастера на панели элементов должна быть отжата**, т. к. в противном случае вам будет предложено создать процедуру обработки событий, а не макрос.
3. Откройте окно свойств только что созданной кнопки, если оно еще не открыто, и раскройте вкладку **События (Event)**.

8.10.4 Назначение макроса событию

4. Помимо обычного **нажатия** (Click), которое будем использовать, в набор событий кнопки входят: **получение и потеря фокуса, двойной щелчок кнопкой мыши, простое перемещение указателя мыши над кнопкой** и пр.

Теперь найдите в списке событие **Нажатие кнопки** (On Click) и установите курсор в соответствующую ячейку. Это поле со списком, и в данный момент оно пустое. Если открыть этот список, то первым его элементом будет **[Процедура обработки событий]** ([Event procedure]), а далее идет перечень всех макросов, существующих в приложении.

8.10.5 Назначение макроса событию

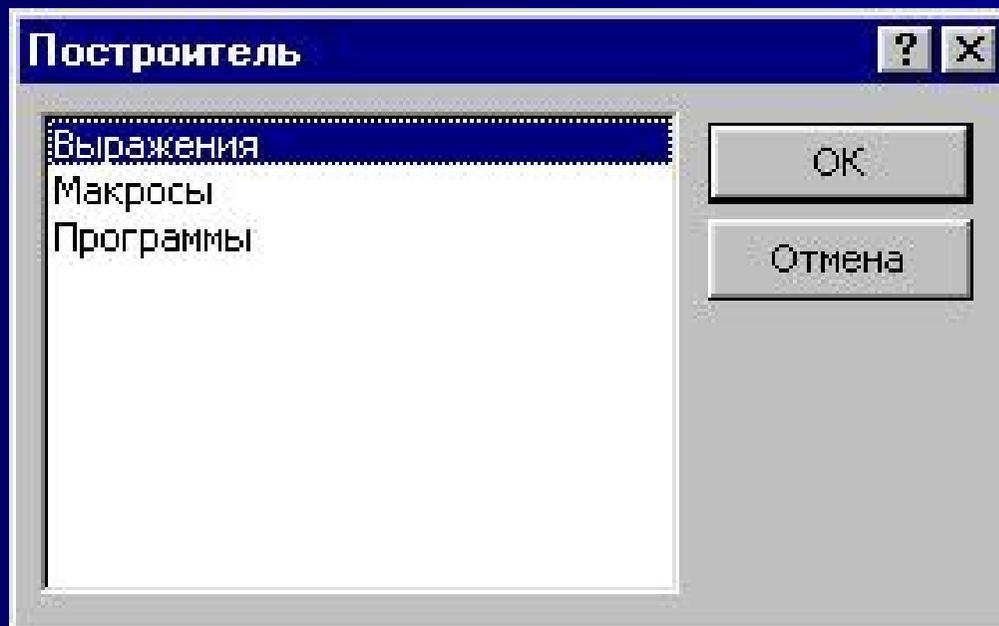


8.10.6 Назначение макроса событию

5. Для создания макроса нажмите кнопку **Построителя**, находящуюся справа от поля. Появится окно **Построитель (Choose Builder)**, в котором предлагается выбрать один из трех Построителей: **Выражения (Expression Builder)**, **Макросы (Macro Builder)** и **Программы (Code Builder)**.

6. Выберите **Макросы (Macro Builder)** и нажмите кнопку **ОК**. Откроется «**окно макросов**» и диалоговое окно, в которое нужно ввести имя создаваемого макроса.

8.10.7 Назначение макроса событию



В макрос нужно добавить единственную макрокоманду **ОткрытьФорму** (**OpenForm**). Значения аргументов этой макрокоманды приведены ниже в таблице.

8.10.8 Назначение макроса событию

*Значения аргументов макрокоманды
ОткрытьФорму (OpenForm) :*

Аргумент	Значение
Имя формы (Form Name)	Заказы (Orders)
Режим (View)	Форма (Form)
Условие отбора (Condition)	[КодКлиента] = [Forms] ! [Клиенты] ! [КодКлиента]
Режим данных (Data Mode)	Только чтение (Read Only)
Режим окна (Window Mode)	Обычное (Normal)

8.10.9 Назначение макроса событию

Аргумент **Режим** (*View*) определяет режим, в котором форма должна быть открыта. Он может принимать значения: **Форма** (*Form*), **Конструктор** (*Design*), **Просмотр** (*Print Preview*), **Таблица** (*Datasheet*), **Сводная таблица** (*PivotTable*) и **Сводная диаграмма** (*Pivot Chart*). Аргумент **Условие отбора** (*Condition*) определяет условие для отбора записей, отображаемых в форме. **Условие** представляет собой выражение. В данном случае это выражение содержит ссылку на элемент управления в форме, что позволяет выбрать все записи из таблицы, в которых поле имеет значение равное значению, заданному в элементе управления формы.

8.10.10 Назначение макроса событию

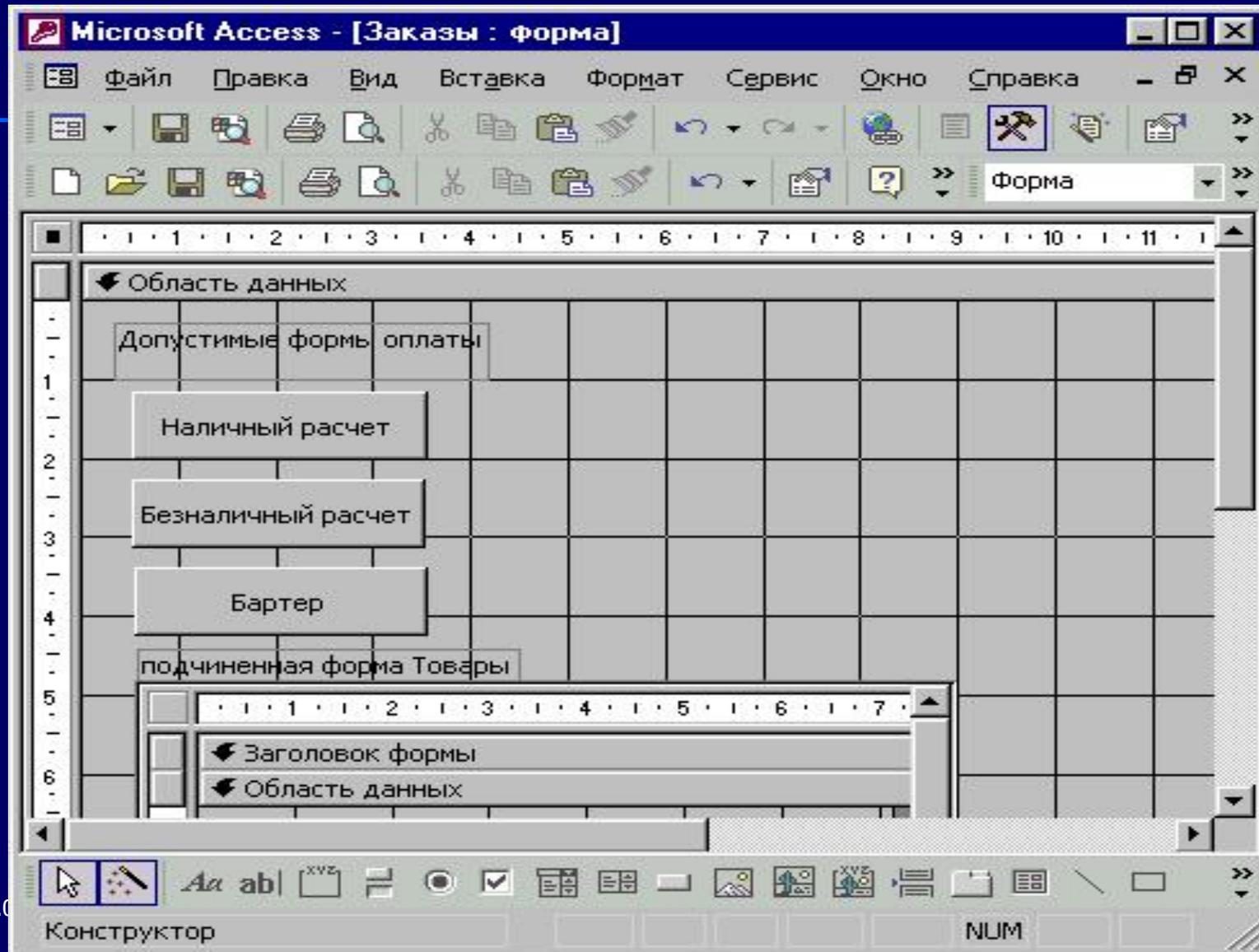
Аргумент **Режим данных** (Data Mode) определяет способ работы с данными и может принимать одно из значений: **Добавление** (Add), **Изменение** (Edit) или **Только чтение** (Read Only). И наконец, аргумент **Режим окна** (Window Mode) определяет тип окна: **Обычное** (Normal), **Невидимое** (Hidden), **Значок** (Icon) и **Окно диалога** (Dialog).

8.10.11 Назначение макроса событию

8. Закройте окно макроса, сохранив изменения. В окне свойств кнопки в поле **Нажатие кнопки** (On Click) появится имя макроса.

9. Раскройте вкладку **Макет** (Format) и введите в поле **Подпись** (Caption) название кнопки. То же самое имя рекомендуется ввести в поле **Имя** (Name) на вкладке **Другие** (Other).

8.10.12 Назначение макроса событию



8.11. Вызов макроса из процедуры VBA

Процедуры **VBA** и **макросы** в Access являются объектами близкими и часто взаимозаменяемыми, т. е. вместо макроса можно использовать процедуру **VBA** и наоборот. Существуют способы запуска макроса из процедуры **VBA**, запуска процедуры VBA из макроса, кроме того, Access позволяет преобразовать макрос в процедуру **VBA**.

Для запуска процедуры **VBA** из макроса существует специальная макрокоманда **ЗапускПрограммы(RunCode)**. Эта макрокоманда имеет один аргумент — имя вызываемой процедуры, хотя вызываться может только функция, а не подпрограмма применяется специальный метод **ЗапускМакроса (RunMacro)** объекта **DoCmd**, например:
DoCmd.RunMacro "Макрос!".

8.11.1 Вызов макроса из процедуры VBA

Объект **DoCmd** используется в процедуре **VBA** для выполнения макрокоманд **Access**. При этом английское имя нужной макрокоманды должно быть указано в качестве **Метода** объекта **DoCmd**, например строка процедуры

DoCmd.OpenForm "Клиенты"
позволяет открыть форму "Клиенты". Здесь "Клиенты"— аргумент макрокоманды. Аргументы перечисляются в предложении **DoCmd** через запятую. Таким способом может быть выполнено большинство макрокоманд.

Источники дополнительных сведений

- На сайте Realcoding.net:

Статьи:: Базы данных :: Учебник по Access 2002 :: Глава 11. Работа с макросами.

- Проектирование баз данных. СУБД Microsoft Access: Учебное пособие для вузов / Н. Н. Гринченко, Е. В. Гусев, Н. П. Макаров., А. Н. Пылькин, Н. И. Цуканова. - М.: Горячая линия -Телеком, 2004, 240с.