**Задание**

**Использование макросов в базе данных Microsoft Access**

**Исследовать возможности Access по созданию макросов**

 1. Введение

Использование макросов в базе данных Microsoft Access

 С помощью макросов можно выполнить практически все действия над объектами Access. Существенной особенностью Access по сравнению с такими компонентами Microsoft Office, как Word и Excel, является возможность с помощью макросов и модулей VBA Определять реакцию приложения на различные события в формах и отчетах, такие как нажатие кнопок, изменения данных, открытие и закрытие форм и отчетов, получение формой или элементом управления формы фокуса и т. д.

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

 В Microsoft Access 2000 содержится список макрокоманд, 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 ” ” собранны все макрокоманды, которые позволяют влиять на интерфейс приложения.

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.

Использование макросов оправдано тем, что их легко создавать, и для этого не нужно изучать синтаксис языка программирования. Как будет показано в следующем разделе, для создания макроса требуется знать лишь основные приемы работы в Microsoft Access и Windows, такие как перетаскивание объектов из окна **База данных** в специальное окно – **Конструктор макросов**, выбор действия из списка и ввод выражений в ка­честве аргументов макрокоманды. Поэтому, если очень не хочется изучать синтаксис языка VBA или это представляется слишком трудным, смело применяйте макросы, и вы получите приложение с достаточно большими функциональными возможностями.

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**, 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.

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, хотя сначала нужно быть уверенным, что эти дополнительные возможности действительно нужны. 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.
* Макросы можно использовать практически везде, где используются 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 становится достаточно затруднитель­ной. Процедуры обработки событий VBA, используемые в формах и от­четах, являются неотъемлемой частью этих 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.

Тем не менее, во многих случаях использование макро­сов вполне оправдано, т. к. это существенно упрощает и ускоряет разработ­ку приложения. Их можно использовать при создании небольших однополь­зовательских приложений, хотя при создании многопользовательских или клиент-серверных приложений, где скорость работы приложения критична, лучше применять VBA.

#  2. Основная часть.

# 2.1. Создание макросов

В Microsoft Access существует специальное средство для создания макросов – **Конструктор** **макросов**. Рассмотрим процесс создания макроса с помощью Конструктора.

# Конструктор макросов

Открыть окно Конструктор макросов для создания нового макроса можно двумя способами:

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

**Рис. 1.** Кнопка для создания макроса

1. Щелкните по стрелке вниз на кнопке **Новый объект** на па­нели инструментов **База данных** и выберите элемент Макрос.

**Рис. 2. Новый объект** на па­нели инструментов **База данных**

Окно Конструктора построено аналогично Конструктору таблиц, т. е. разде­лено по горизонтали на две части — панель описаний и панель аргументов.

**Рис. 3.** Конструктор макросов и панель аргументов

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

Строка в столбце **Макрокоманда** представляет собой поле со спи­ском, в котором можно выбрать нужную макрокоманду.

Строка в столбце **Примечание** — это обычное текстовое поле, в которое можно ввести комментарий, описывающий выполняемое действие.

Когда поле **Макрокоманда** заполнено, в нижней части окна Конст­руктора макроса появляется панель аргументов, которая служит 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.

На панели аргументов имеются два поля, которые позволя­ют задать имя фильтра и условие отбора записей. Для того чтобы задать условие отбора записей, можно воспользоваться построителем выражений.

 Такой макрос можно использовать в форме или отчете для того, чтобы ограничить число отображаемых записей.

Существует еще один очень простой способ ввода макрокоманды. Допус­тим, нужно создать макрос, который должен открывать форму “Вычисление” в приложении “**db1**”. Для этого в окне **База данных** нужно щелкнуть по **Форме**, затем, используя панель инструментов (способ 2), открыть Конструктор макросов и перетащить с помощью мыши Форму “Вычисление” в поле **Макрокоманда** окна Конструктора макроса.

Тогда в поле **Макрокоманда** автоматически появится макрокоманда ОткрытьФорму, а на панель аргументов автоматически заполняются поля: в поле **Имя формы** появляется имя формы “Вычисление”, в поле **Режим** – значение по умолчанию **Форма**,в поле **Режим окна** – значение **Обычное.** Остальные поля останется пустыми.

Кроме того, можно сделать записи в этой форме недоступными для изменений, для этого в поле **Режим данных** нужно ввести значение Только чтение.

**Рис. 4.** Автоматическое создание макрокоманд путем перетаскивания объектов

Аналогично можно создавать макрокоманда, перетаскивая из окна **База данных** таблицы, запросы, отчеты и другие макросы. При этом будут создаваться соответствующие макрокоманды: ОткрытьТаблицу, ОткрытьЗапрос, ОткрытьОтчет, ЗапускМакроса.

Для ввода аргументов макрокоманды чаще всего требуется выбрать значения из списков или вводить выражения. Для ввода выражений можно воспользоваться построителем выражений, кнопка которого находится справа от поля аргумента. Еще одна кнопка построителя выражений находится на панели инструментов. Как и в других случаях, перед выражением нужно ставить знак равенства (=). Исключения составляют аргумент **Выражение**  макрокоманды задать Значение и аргумент **Число повторений** макрокоманды ЗапускМакроса. Если ввести знак равенства перед выражением, задающим значение этих аргументов, то оно будет вычисляться дважды, что может привести к нежелательным результатам.

Если в качестве аргумента макрокоманды необходимо задать имя объекта базы данных, то его можно либо ввести с помощью клавиатуры, либо выбрать из раскрывающегося списка, либо указать название объекта, перетащив его из окна базы данных.

Итак, созданная нами макрокоманда будет открывать форму “Вычисление” в режиме Только чтение и показывать в ней только менеджеров по продажам. Как уже было сказано ранее, макрос может содержать последовательность из нескольких макрокоманд. Поэтому в этом же макросе в следующей строке мы выберем макрокоманду Сообщение, которая после открытия формы будет отображать диалоговое окно с сооб­щением о том, какие клиенты представлены в форме. Для этого введем в поле **Сообщение** на панели аргументов текст сообщения, например

Установлен фильтр по Фамилиям, остальные поля можем оставить без изменения. Аргумент **Сигнал** определяет, будет ли вывод сообще­ния сопровождаться звуковым сигналом. Аргумент **Тип** определяет вид диалогового окна сообщения. Аргумент **Заголовок** позволяет за­дать текст, который будет выведен в заголовке окна сообщения.

Если база данных большая, отбор записей в форме может занять некоторое время, поэтому рекомендуется вывести на экран курсор в форме песочных часов, который будет показывать, что идет обработка данных. Чтобы сде­лать это, воспользуемся макрокомандой ПесочныеЧасы. Макро­команды в макросе выполняются в той последовательности, в которой они записаны в столбце **Макрокоманда**, т. е. сверху вниз. Макрокоманду ПесочныеЧасы нужно поставить первой, т. к. курсор должен из­мениться перед выводом формы. Таким образом, нам нужно вставить новую строку на панели описаний. Для этого установим курсор в первую строку панели и нажмем кнопку **Добавить** **строки** на панели инстру­ментов или просто клавишу <Insert>. Появляется пустая строка. В этой строке в поле **Макрокоманда** выберем макрокоманду ПесочныеЧасы. В поле **Включить** панели аргументов появится значение **Да**, установленное по умолчанию, что нам и нужно. Чтобы вернуть первоначальный вид указателю мыши, в конце созданного макроса нужно ввести такую же макрокоманду ПесочныеЧасы, но со зна­чением **Нет** аргумента **Включить**.

Строго говоря, эту макрокоманду лучше поставить перед макрокомандой вывода сообщения на экран. Это легко исправить, поскольку макрокоманды легко меняются местами с помощью мыши. Чтобы поменять две последние макрокоманды местами, выделите последнюю строку, щелкнув левой кноп­кой мыши по области выделения строк (слева в окне Конструктора) и пере­тащите строку вверх. Горизонтальная линия показывает место нового поло­жения строки. Отпустите кнопку мыши, когда эта линия окажется над предпоследней строкой.

Теперь остается только сохранить созданный макрос. Для этого можно вы­брать в меню **Файл** команду **Сохранить** **как** или нажать кла­вишу <F12>. Появится окно сохранения макроса. Введите в поле ввода имя макроса, соблюдая правила именования объектов Access, например, менеджеры по продажам.

**2.2. 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 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 будет являться именем группы макросов.

Чтобы создать группу макросов:

* Откройте Конструктора макросов.

**Рис 5.** Открытие Конструктора макросов

* Нажмите кнопку Имена макросов на панели инструментов. На панели описаний в окне Конструктора макросов появится еще один столбец Имя макроса.

**Рис 6.** Дополнительный столбец “**Имя Макроса**”

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

**Рис 7.** Ввод в столбец первого имени макроса

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

**Рис 8.** Ввод всех макрокоманд и аргументы для каждой макрокоманды в макрос

Для указания конца макроса не требуется отдельной макрокоманды. Access продолжает выполнение мак­роса, входящего в группу, пока не обнаружит строку, которая содержит имя в столбце Имя макроса, либо пока не дойдет до последней строки окна 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: имяГрупы.имяМакроса.

**2.3. Запуск макроса из окна Конструктора макроса**

Этот способ применения для тестирования только что созданного или исправленного макроса. Если макрос в окне Конструктора макроса один, то для его запуска нужно просто нажать на кнопку **Запуск** на панели инструментов или выбрать команду **Запуск, По шагам** (последний способ позволяет отладить макрос). Однако если макросов не­сколько, то с помощью этой кнопки или команды можно запустить только первый макрос в группе. Чтобы запустить любой другой макрос из группы, придется воспользоваться командой меню **Сервис, Марко, Выполнить мак­рос**.

Если вы часто используете этот способ запуска макроса, имеет смысл застроить панель инструментов, добавив на нее стандартную кнопку **Запуск макроса**. Она эквивалентна выполнению команды меню.

# 2.4. Запуск макроса из окна базы данных

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

**2.5. 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 с нашей точки зрения способ запуска макроса — это создание специальной кнопки на панели инструментов. Правда, панелей инструментов очень много, причем в разных режимах работы с Access на экране появляются разные 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.

Чтобы добавить кнопку на стандартную панель инструментов Access:

* Щелкните правой кнопкой мыши по панели инструментов и выберите в контекстном меню команду **Настройка**.
* В появившемся диалоговом окне **Настройка** раскройте вкладку **Панели инструментов**.
* Установите флажок для одной из стандартных панелей инструментов, например, **Запрос в режиме таблицы**, чтобы она отобра­зилась на экране.
* Щелкните по ярлыку **Макросы** на панели объектов в окне **База данных** и найдите нужный макрос.
* Перетащите макрос с помощью мыши на панель инструментов

**2.6. Запрос в режиме таблицы.**

На панели инструментов появится кнопка с характерным значком макроса. Если вы подержите несколько секунд курсор на этой кнопке, то увидите подсказку **Запуск макроса <имя макроса>**. Вы можете таким образом переместить на панель инструментов созданный нами макрос «Open». Теперь всякий раз, когда эта панель инструментов будет отображаться на экране, вам будет дос­тупна кнопка запуска этого макроса.

**Рис 9.** Кнопка запуска макроса на панели инструментов

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

* Щелкните по ярлыку **Макросы** на панели объектов в окне **База данных** и выделите нужный макрос.
* Выберите команду **Сервис, Макрос, Создать панель инструментов из мак­роса**. На экране появляется пустая панель инструментов.

**Рис 10.** Пустая панель инструментов

* Перетащите с помощью мыши нужный макрос на эту панель.

На панели появляется кнопка с уже знакомым значком макроса.

**Рис 11.** Значок макроса

 Кнопка получает такое же название, как имя макроса, для которого она создана.

Тот же самый результат можно получить другим способом:

* Откройте окно **Настройка**, щелкнув правой кнопкой мыши по панели инструментов.
* В появившемся диалоговом окне раскройте вкладку **Панели инструментов** и нажмите кнопку Создать.

**Рис 12.** Диалоговое окно “**Настройка**” с вкладкой “**Панель Инструментов**”, кнопка “**Создать**”

* В появившемся диалоговом окне введите имя новой панели инструмен­тов мои макросы и нажмите кнопку ОК. Появится пустая панель инстру­ментов.

**Рис 13.** Ввод **имени** новой **Панели Инструментов**

* Выберите вкладку **Команды**.
* Выберите в списке **Категории** значение **Все макросы**, а в списке Команды найдите макрос «Open».
	+ Перетащите этот макрос с помощью мыши на пустую панель инструмен­тов.

**Рис 14.** Помещение макроса на пустую **Панель Инструментов**

Теперь посмотрим, как можно изменить внешний вид этой кнопки.

В диалоговом окне **Настройка** нажмите кнопку **Изменить выде­ленный объект** или щелкните правой кнопкой мыши по только что созданной кнопке. Отобразится контекстное меню.

**Рис 15.** Контекстное меню только что созданной кнопки

Видим, что данный режим отображения кнопки помечен, как **Основной стиль**. Вообще говоря, на кнопке можно помес­тить не только значок, но и текст или текст и значок вместе. Если вы хотите поместить на кнопке текст (название кнопки), введите этот текст в поле **Имя**. По умолчанию в этом поле отображается имя макроса.

Вы можете поместить на кнопку другой значок, выбрав понравившуюся картинку на специальной панели. Для отображения набора стандартных значков выберите в контекстном меню команду **Выбрать значок для кнопки**. Если же ни одна из картинок не подходит, можно выбрать наиболее близкий по оформлению значок, а потом исправить его в специальном редакторе. Чтобы открыть окно **Редактор кнопок** выберите в контекстном меню команду **Изменить значок на кнопке**.

**Рис 16.** Редактор кнопок

Наконец, можно просто скопировать понравившийся вам значок с другой кнопки.

Чтобы скопировать значок с одной кнопки на другую:

* + Откройте диалоговое окно **Настройка** и выделите ту кнопку на панели инструментов, с которой нужно скопировать значок.
	+ Нажмите кнопку **Изменить выделенный объект** в диа­логовом окне **Настройка**.
	+ Выберите в контекстном меню команду **Копировать значок на кнопке**.
	+ Выделите кнопку, значок которой нужно изменить.

**Рис 17.** Выбранный рисунок на кнопке для изменения

* Нажмите кнопку **Изменить выделенный объект** в диа­логовом окне **Настройка**.
* Выберите в контекстном меню команду **Вставить значок для кнопки**.

**Рис 18.** Контекстное меню

С помощью диалогового окна свойств кнопки можно изменить текст всплы­вающей подсказки и задать клавиши быстрого вызова.

Чтобы изменить текст всплывающей подсказки для кнопки на панели ин­струментов:

* В режиме настройки выделите кнопку на панели инструментов. В диа­логовом окне **Настройка** нажмите кнопку **Изменить выделен­ный объект**.
* Выберите в контекстном меню команду **Свойства**. Появится диалоговое окно свойств кнопки.

**Рис 19.** Свойство кнопки “**Open**”

* Введите желаемый текст в поле **Всплывающая подсказка**.

На созданной вами панели инструментов можно разместить любые другие полезные кнопки, не только кнопки для запуска макросов. Когда создание панели инструментов закончено, вы можете переместить ее на свободное место в верхней части экрана (или поместить у любой другой границы эк­рана).

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). Создание такой панели ин­струментов можно осуществить с помощью одной команды **Сервис, Макрос, Создать панель инструментов из макроса.** Попробуйте сделать это для одной из групп макросов в базе данных «db1», например, «Open». С помощью указан­ной команды создается панель инструментов с кнопками для вызова каж­дого макроса из группы, причем на кнопках отображаются имена соответст­вующих макросов. Новая панель инструментов получает такое же имя, как имя группы макросов. При желании можно заменить текст на кнопках значками, как это было описано выше.

**2.7. Связывание макросов с событиями.**

 В базе данных хранится информация о состоянии её объектов. Любое изменение состояния формы или отчета называется событием. Каждый из этих объектов имеет свой набор событий.

 С событиями удобно связывать макросы. Попробуем связать созданный нами макрос *Сообщение* с открытием базы данных:

* Маркируйте макрос *Размещение на экране* в окне базы данных.
* Выберите в меню **Павка** команду **Переименовать** и присвойте макросу *Сообщение* новое имя *AutoExec* (макрос с таким именем автоматически выполняется при загрузке базы данных). Вы можете произвольным образом использовать строчные и прописные литеры.

Теперь макрос связан с открытием базы данных. Таким образом автоматизируются операции, которые должны всегда выполняться в момент загрузки базы. При каждом открытии базы данных Accesss 2000 поверяет присутствует ли макрос с именем *AutoExec*, и, если находит, выполняет его. В противном случае открытие базы не сопровождается дополнительными действиями. Внутри одной базы данных может использоваться только один макрос *AutoExec*.

* Чтобы проверить работоспособность макроса, закройте, а затем откройте базу данных.

Если имя *AutoExec* присвоено макрогруппе, то при открытии базы данных будет выполнен только первый макрос этой группы! Чтобы при загрузке базы последовательно приводились в исполнение несколько макросов, введите в макрос *AutoExec* макрокоманду **ЗапускМакроса,** что позволит выполнить макрос с произвольным именем (имя задается аргументом) и вернуться к следующей за **ЗапускМакроса** макрокоманде в макросе *AutoExec.*

**Макрос** *AutoExec* не выполняется, если перед открытием **базы данных** нажать и удерживать клавишу [Shift].

**2.8. Связывание макроса с кнопкой**

Целесообразно создавать элементы управления (например, кнопки) для вызова макросов. Это значительно упрощает и ускоряет доступ к ним. Удобным местом для размещения таких элементов является заголовок или примечание формы.

В качестве упражнения попробуем связать с кнопкой макрос, который маркирует активную запись формы и копирует ее в буфер обмена:

* Создайте макрос *Копировать,* состоящий из следующих макрокоманд:

**Макрокоманда Аргумент**

ВыполнитьКоманду Select Record

ВыполнитьКоманду Copy

* Сохраните макрос и закройте его окно.
* Откройте форму *Переключатель* и активизируйте режим конструктора.
* Вызовите команду Слева направо меню Окно.
* Маркируйте макрос *Копирование записи* в окне базы данных и перетащите его в область заголовка проекта формы.

**Рис 20.** Перетаскивание макроса из окна базы данных в область заголовка проекта формы

Как только вы отпустите кнопку мыши, позиция кнопки для макроса зафиксируется. При необходимости ее размер можно изменить с помощью специальных манипуляторов. Перемещение кнопки осуществляется с помо­щью манипулятора, расположенного в левом верхнем углу маркировочной рамки.

Имя макроса автоматически появляется внутри кнопки. Если оно слиш­ком длинное, размер кнопки следует увеличить. Чтобы на кнопке была видна вся надпись, можно уменьшить размер шрифта или создать надпись, отли­чающуюся от имени макрокоманды. Редактирование надписи выполняется после щелчка на кнопке. Для выхода из режима редактирования достаточно щелкнуть вне кнопки.

Для изменения параметров управляющего элемента **Кнопка** надлежит выполнить на нем двойной щелчок мышью, вследствие чего откроется окно свойств, если до этого оно отсутствовало на экране.

**Рис 21.** Окно свойств кнопки

У каждого элемента управления свой список доступных событий. Однако действия при связывании всегда одинаковы. Например, чтобы связать неко­торый макрос с перемещением мыши по форме, следует в режиме конструк­тора выполнить двойной щелчок в свободной области вне примечания формы и задать для события **Перемещение указателя** имя связываемого макроса.

**2.9. Вызов приложений и управление ими**

С помощью макроса можно запустить из Access 2000 приложения Microsoft Office 2000, а также программы, работающие под управлением Windows 98, Windows NT или DOS (например, чтобы выполнить некоторые вычисления с помощью Excel 2000 или сделать небольшие заметки посредством редакторов Notepad и Word 2000). Вызов выполняется одинаково во всех случаях и заключается главным образом в указании имени запускаемой программы. Макрос для вызова прикладной программы можно связать с кнопкой формы, ускорив таким образом доступ к нужному приложению:

* Создайте новый макрос. В первой строке введите макрокоманду **ЗапускПриложения.**

**Рис 22.** Создание макрокоманды “**ЗапускПриложения**”

* В качестве значения аргумента *Командная строка* введите имя запускае­мой программы, например *WinWord.*

**Рис 23.** Командная строка с аргументом запускаемой программы WinWord

* Сохраните макрос под именем *Заметки.*

**Рис 24.** Ввод имени макроса

* Откройте форму *Переключатель* в режиме конструктора и перетащите макрос из окна базы данных в область заголовка формы.

**Рис 25.** Режим конструктора формы “**Переключатель”**

* Активизируйте режим формы и щелкните мышью на кнопке Заметки. После запуска этого макроса (щелчком на кнопке **Заметки** в форме *Переключатель)* на экране появится окно текстового редактора Word 2000 пакета Microsoft Office 2000. Теперь пользователь может ввести свои заметки, распечатать или сохранить их.

**Рис 26.** Окно текстового редактора **Word**

При вызове программы всегда приводится полная спецификация ее файла. При этом вводить расширение не следует. Каталог, в котором хранится файл, необходимо указывать только в том случае, если он не задан командой PATH в файле AUTOEXEC.BAT. Например:

C:\Program Files\NC\nc

Как правило, при вызове прикладных программ, входящих в комплект поставки Windows 98, папка Windows указывается в команде PATH. Поэтому имя редактора в данном макросе приведено без указания каталога.

Следите за тем, чтобы после окончания сеанса работы редактор был закрыт корректно.

**2.10. Запуск прикладных программ с параметрами**

Запуская прикладные программы с помощью макрокоманды **ЗапускПриложения,** можно указывать ключи, которые обычно вводятся в командной строке после имени программы при запуске с уровня DOS. Эти ключи служат для задания режима работы прикладной программы. Если, к примеру, из макроса запускается редактор Word 2000, то параметр /t *имя документа* приме­няется для загрузки документа с указанным именем, который используется в качестве шаблона. Ключ */n* указывается при запуске Word 2000 без открытия нового документа. Способ указания ключей в аргументе *Командная строка* такой же, как в командной строке DOS:

**C:\Program Files\Microsoft Office\0ffice\winword.exe /n**

Использование ключей допускается при запуске большинства приклад­ных программ.

Вернемся к макросу, разработанному для вызова редактора Word 2000, и несколько модифицируем его. При каждом вызове редактора целесообразно использовать для заметок один и тот же документ (по умолчанию каждый вызов Word 2000 сопровождается открытием нового документа).

* Перейдите в окно Word 2000 и вставьте в первую строку документа поля **Date** и **Time**.

**Рис 27.** Ввод даты и времени в текстовый редактор **Word**

* Сохраните документ Word 2000 под именем *Заметки* в папке *Мои доку­менты,* не указывая расширение doc, и закройте окно Word 2000.
* Маркируйте макрос *Заметки* в окне базы данных и активизируйте режим конструктора.

**Рис 28.** Режим Конструктора для макроса “**Заметки”**

* Присвойте аргументу *Командная строка* значение *WinWord Заметки* и сохраните макрос.

**Рис 29.** Ввод аргумента в Командную строку WinWord Заметки

При каждом новом вызове редактора в документ заметок вводятся

системная дата и системное время — таким образом, содержимое автомати­чески протоколируется.

 **Заключение**

В своей дипломной работе, я, исследовал возможности СУБД Access 2000 по созданию макросов. 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 процедура на VBA. Каждая макрокоманда имеет определенное имя и, возможно, один или несколько аргументов, которые задаются пользователем.

 **Содержание**

**Задание**

1. **Введение**
2. **Основная часть**
	1. **Создание макросов**
	2. **Создание групп макросов**
	3. **Создание макроса из окна Конструктора макроса**
	4. **Запуск макроса из окна базы данных**
	5. **Запуск макроса с помощью кнопки на панели инструментов**
	6. **Запрос в режиме таблицы**
	7. **Связывание макросов с событиями**
	8. **Связывание макроса с кнопкой**
	9. **Вызов приложений и управление ими**
	10. **Запуск прикладных программ с параметрами**
3. **Заключение**
4. **Приложение**
5. **Список литературы**

 Список литературы

1. И. Харитонова, В. Михеева, Microsoft Access 2000 в подлиннике.
2. В. Пасько, Для пользователей ПК русифицированная версия Access 97.