# Лабораторная работа №8. Программирование в Access.

**Макросы.**

Макрос — это средство для автоматизации задач и добавления функциональных возможностей в формы, отчеты и элементы управления. Например, при добавлении командной кнопки в форму событие кнопки OnClick (Нажатие кнопки) связывается с макросом, который содержит команды, выполняемые при каждом нажатии кнопки.

В приложении Access макросы можно рассматривать как упрощенный язык программирования, на котором программа записывается в виде списка макрокоманд. Основной компонент макроса; замкнутая инструкция, самостоятельно или в комбинации с другими макрокомандами определяющая выполняемые в макросе действия. В других макроязыках макрокоманды иногда называют просто командами.) для выполнения. При создании макроса каждая макрокоманда выбирается из раскрывающегося списка, после чего к ней добавляется необходимая информация. Макросы позволяют добавлять функциональные возможности в формы, отчеты и элементы управления без необходимости написания кода в модуле Макросы обеспечивают выполнение части команд, доступных в VBA, и для большинства пользователей создание макроса оказывается проще, нежели написание кода VBA.

Предположим, требуется запустить отчет непосредственно из формы ввода данных. Можно добавить в форму кнопку и затем создать макрос, который будет запускать отчет. Макрос может быть изолированным (отдельный объект в базе данных), который затем связывается с событием OnClick (Нажатие кнопки) для кнопки, или же он может быть внедрен непосредственно в событие кнопки OnClick  (Нажатие кнопки). В любом случае, при нажатии кнопки выполняется макрос, который запускает отчет.

***ПРИМЕР 1.***

Рассмотрим пример создания простейшего макроса для открытия таблицы и перехода на последнюю запись. Выберите на ленте «Создание» в группе «Макросы и код» команду «Макрос». На экране появится новое окно для создания нового макроса (Конструктор макросов). Любой макрос состоит либо из последовательности одного или нескольких шагов, либо из действий (макрокоманд). Для создания макроса вы формируете список макрокоманд программы. У программы Access есть предварительно подготовленный список макрокоманд, которые можно использовать для приготовления макросов. Когда добавляется команда, ее просто выбирают из этого списка. Выберите макрокоманду «ОткрытьТаблицу». Далее устанавливаются параметры для вашего макроса.

Большинству макрокоманд для выполнения их работы нужна от вас некоторая инфор­мация. В макрокоманде «ОткрытьТаблицу» мало смысла до тех пор, пока вы не сообщите программе Access, какую именно таблицу хотите открыть. Эти дополни­тельные сведения и называют аргументами.

У макрокоманды «ОткрытьТаблицу» три аргумента.

* Имя таблицы ссылается на таблицу, которую вы хотите открыть. Его можно выбрать из раскрывающегося списка таблиц. В данном примере можно использовать любую непустую таблицу.
* Режим позволяет выбрать применяемый режим представления. Можно выбрать обычный режим «Таблица» для ввода информации, «Конструктор» для изменения струк­туры таблицы, «Просмотр» для подготовки данных к печати или «Сводную таблицу» и «Сводную диаграмму» для работы с итогами сводной таблицы. В данном примере вы­берите режим «Таблица».
* Режим данных определяет, какой тип изменений разрешен. Можно использовать стандартный вариант «Изменение» для разрешения любых изменений, «Только чтение» для запрета каких бы то ни было изменений или «Добавление» для разрешения только добавления записей. В данном примере выберите «Только чтение».

Вы можете вставить в макрос практически неограниченное число макрокоманд. Каждая команда занимает отдельную строку, и программа Access выполняет их по порядку, перемещаясь сверху вниз.

Для завершения данного примера добавьте макрокоманду «НаЗапись». Это действие обеспечивает получение нужной записи в таблице, которую вы только что от­крыли. При использовании аргументов следует указывать корректный объект. Затем можно использовать аргументы «Запись» и «Смещение» для точного задания той позиции, в которую вы хотите перейти. С помощью аргумента «Запись» можно выбрать переход к предыдущей строке, к следующей строке, к заполнителю новой строки в конце таблицы к заданной строке, к первой строке или, как в данном примере, к последней строке. Если вы­брать «Конкретная», можно использовать аргумент «Смещение» для указания точной пози­ции, например, если задать «Смещение 5», обеспечивается переход к 5-й записи.

Сохраните созданный макрос (рис. 1).

Рис. 1. Создание макроса.

***ПРИМЕР 2.***

Рассмотрим следующий пример. В форме с категориями товаров создать кнопку, при нажатии на кнопку, возле наименования категории, открывается форма со списком товаров именно этой категории. То есть необходимо реализовать макрос, открывающий список товаров по категориям.

Открываем форму «Категории» (2-е задание 4 лабораторной работы) в режиме конструктора. На панели элементов управления выбираем элемент «Кнопка» и размещаем элемент в области данных формы. Отменяем «Мастер создания кнопок». Далее необходимо вызвать окно свойств для созданной кнопки. В окне свойств, на вкладке «События» в строке «Нажатие кнопки» вызываем построитель с помощью кнопки . В появившемся окне необходимо выбрать «Макросы» и нажать кнопку «ОК». Появится конструктор макросов.

В конструкторе макросов выберите макрокоманду «ОткрытьФорму». Установите следующие аргументы макрокоманды:

* «Имя формы» – «Товары по категории» (1-е задание 5 лабораторной работы).
* «Режим» - «Форма».
* «Условие отбора» - [Наименование категории] =Forms![Категории]![Наименование категории] . (То есть необходимо указать по каким полям происходит связь между формами «Категории» и «Товары по категории»). Если в форме «Товары по категории» отсутствует поле «Наименование категории», то такое поле обязательно следует добавить.

Макрос готов (рис. 2). Сохраняем и закрываем макрос. Далее проверяем его работоспособность через форму «Категории» нажав на кнопку.



Рис. 2. Макрос для открытия формы.

**Макросы с условиями.**

Все рассмотренные макросы, до настоящего момента, были линейные. Они выполняли свои макрокоманды по порядку от начала к концу.

Для создания макроса с условием необходимо применить команду «Если». Для этого необходимо в «Каталоге макрокоманд», который обычно отображается в правой части окна, двойным кликом мыши выбрать команду «Если». В появившейся строке «Если» задаются условия. Условие похоже на выражение, но оно в результате всегда выдает одно из двух значений: «Истина» или «Ложь». Программа Access проверяет условие и решает с его помощью выполнить или нет соответ­ствующую макрокоманду.

***ПРИМЕР 3.***

Рассмотрим пример, Допустим, вы хотите создать простое условие, которое проверяет добавление или обновление опре­деленной записи. Например, в форме «Поставщики» (3-е задание 4 лабораторной работы) есть поле «Адрес», необходимо при событии «Потеря фокуса» для этого поля, чтобы появлялось сообщение «Вы не ввели адрес!», если поле осталось пустым.

Для этого в свойстве поля «Адрес», на вкладке «События» необходимо вызвать построитель  для свойства «Потеря фокуса». Выбрать «Макросы».

В конструкторе макросов добавить строку «Если» и прописать условие IsNull([Формы]![Поставщики]![Адрес]). Далее выбрать макрокоманду «ОкноСообщения» и указать соответствующие аргументы. Далее после окончания блока «Если» для установки фокуса можно выбрать макрокоманду «КЭлементуУправления» и в качестве аргумента указать «Адрес» (рис.3).

Рис. 3. Макрос с условием.

Макрос готов. Необходимо сохранить и закрыть его.

**Модули.**

В Access пользователь может управлять выполнением приложения, например, изменить информацию в таблице или щелкнуть на командной кнопке. Используя в приложении макросы и процедуры, можно реализовать реакцию пользовате­ля на эти действия.

Для создания процедур обработки событий используется язык VBA (Visual Basic for Application). Язык VBA является современным языком структурного программирования. С помощью VBA можно настроить формы и отчеты, запус­тить макросы, а также отобразить объект Access в других приложениях или извлечь данные. Несмотря на то, что программный код на Visual Basic хранится в вашей БД, вы должны применять специальное средство для его - просмотра и редактирования. Оно называется ре­дактором Visual Basic. Редактор Visual Basic работает во взаимодействии с программой Access, но открывается в отдельном окне. Для перехода в редактор Visual Basic выберите на ленте «Работа с базами данных» в группе «Макрос» команду «Visual Basic».

Для создания процедур VBA необходимы базовые знания по алгоритмизации и знания языка Visual Basic.

**Контрольные вопросы:**

1. Макрос. Назначения макроса.
2. Основные макрокоманды.
3. Конструктор макросов.
4. Условия выполнения макросов.
5. Группы макросов.
6. Запуск макроса.
7. Связывание макроса с событием.
8. Модуль. Назначение модулей.
9. Модули класса.
10. Стандартные модули.
11. Преобразование макросов в модули.
12. Элементы программы.
13. Переменные, константы, массивы.
14. Процедуры и функции.
15. Управление выполнением программы.
16. Семейства, объекты, методы, свойства.
17. Иерархическая структура объектов.
18. Взаимосвязь макросов и программ VBA.
19. События. Виды событий.

**Задания к лабораторной работе:**

База данных «Магазин».

1. В форме с категориями товаров создать кнопку, при нажатии на кнопку, возле наименования категории, открывается форма со списком товаров именно этой категории. (Пример 2)
2. В форме с поставщиками товаров создать кнопку, при нажатии на кнопку, возле наименования поставщика, открывается форма со списком товаров именно этого поставщика. (по примеру 2)
3. В форме с продавцами товаров создать кнопку, при нажатии на кнопку, возле ФИО продавца, открывается форма со списком товаров, проданных этим продавцом. (по примеру 2).
4. Создать макрос AutoExec (При открытии базы данных автоматически открывается Кнопочная форма).
5. Создать макрос AutoKeys. Создать три горячие комбинации клавиш для запуска примеров 1, 2, 3.

