Практическая работа 45
Разработка макета и создание экранной формы.
Создание кнопок и обработчиков событий на экранных формах

Цель работы: Получить практический опыт разработки макета и создания экранной формы.

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio, Access)

Краткие теоретические сведения

Access предоставляет широкие возможности для создания форм - графического диалогового интерфейса пользователя. Формы позволяют добавлять и удалять записи в таблицах, изменять значения полей, получать расчетные данные.

Простейшая форма на основе одной таблицы может быть создана при выборе одного из режимов АВТОФОРМЫ: В СТОЛБЕЦ, ЛЕНТОЧНАЯ, ТАБЛИЧНАЯ.

АВТОФОРМА В СТОЛБЕЦ является одноразовой. В ней поля располагаются в столбец, а в окне формы одна запись и кнопки перехода к другим записям (рекомендуется использовать автоформу в столбец, если исходная таблица содержит много полей).

Остальные автоформы многоразовые, т. е. поля размещаются в одной строке и несколько записей в окне.

Все автоформы не предоставляют возможности выбора стиля оформления и выбирают все поля таблицы. Однако после создания автоформы, войдя в режим конструктора, можно доработать ее в нужном направлении, т. е. отредактировать.

Формы могут быть получены и при помощи Мастера форм, который в режиме диалога с пользователем позволяет выбирать одну или несколько исходных таблиц, отображать только необходимые поля, предоставляет стили для их оформления.

Измените расположение полей в области данных.

Для изменения размера или положения элемента его необходимо предварительно выделить. Если указатель мыши принимает форму раскрытой ладони можно перетащить элемент (надпись и поле) в новое место. Перетащить поле и присоединенную к нему надпись можно независимо друг от друга, когда указатель мыши принимает форму сжатой ладони с вытянутым указательным пальцем). Для выделения нескольких элементов необходимо удерживать нажатой клавишу <Shift>. Чтобы

выровнять элементы по размеру или расположению можно воспользоваться пунктом меню Формат.

Добавьте рисунок.

Преобразуйте форму СТУДЕНТ в режиме конструктора.

Добавьте заголовок СТУДЕНТЫ в области заголовка. В области примечания разместите кнопки перехода между записями и кнопку закрытия формы.

Если запросы — это специальные средства для отбора и анализа данных, то формы — это средства для ввода данных. С помощью форм данные можно не только вводить, но и отображать. Запросы тоже отображают данные, но делают это в виде результирующей таблицы, не имеющей почти никаких средств оформления. При выводе данных с помощью форм можно применять специальные средства оформления

Пользователь имеет возможность создать форму самостоятельно или воспользоваться мастером. Мастер форм ускоряет процесс создания формы, так как автоматически выполняет всю основную работу. При использовании мастера Microsoft Access выдает приглашение на ввод данных, на основе которых создается форма. В созданных формах мастер можно использовать для быстрого создания элементов управления в форме. Для настройки формы следует переключиться в режим конструктора.

Кроме того, форму можно создать с помощью кнопки Новый объект, без помощи мастера, на основе нескольких таблиц, из записей, отфильтрованных в таблице, запросе или другой форме.

Рассмотрим создание формы с помощью мастера

1 В окне базы данных выберите вкладку Формы.

2 Нажмите кнопку Создать.

3 В диалоговом окне Новая форма выберите нужного мастера. Описание мастера появляется в левой части диалогового окна.

4 Выберите имя таблицы или запроса, содержащих данные, на основе которых будет создана форма. При использовании мастера форм источник данных для формы следует указывать в диалоговом окне мастера.

5 Нажмите кнопку ОК.

6 Если на шаге 3 были выбраны Мастер форм, Диаграмма или Сводная таблица, то при создании формы следуйте инструкциям, выводимым в диалоговых окнах соответствующего мастера.

При выборе элементов Автоформа: в столбец, Автоформа: ленточная или Автоформа: табличная форма создается автоматически. Для просмотра записей с небольшим количеством полей удобно использовать ленточную форму. В отличие от простой формы, которая показывает только одну запись, в ленточной форме отформатированные записи выводятся на экран одна за другой, как в таблице. Изменить созданную форму можно в режиме конструктора.

Структура формы. Форма имеет три основных раздела: область заголовка, область данных и область примечания. Линии, разделяющие

разделы, перетаскиваются по вертикали с помощью мыши, это позволяет изменять размеры разделов так, как требуется.

Разделы заголовка и примечания имеют чисто оформительское назначение — их содержимое напрямую не связано с таблицей или запросом, на котором основана форма. Раздел данных имеет содержательное значение — в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод. Разработчик формы может разместить здесь дополнительные элементы управления для автоматизации ввода данных (переключатели, флажки, списки и другие, типичные для приложений Windows).

Все сведения в форме или отчете содержатся в элементах управления. Элементы управления — это объекты формы или отчета, которые служат для вывода данных на экран, выполнения макрокоманд или оформления формы или отчета. Например, поле можно использовать для вывода данных на экран в форме или отчете, кнопку — для открытия другой формы или отчета, а линию или прямоугольник — для разделения и группировки элементов управления с тем, чтобы они лучше воспринимались пользователем.

В Microsoft Access существуют следующие **типы элементов управления,** которые содержатся на панели элементов в режиме конструктора формы или режиме конструктора запроса: поле, надпись, группа, переключатель, флажок, выключатель, поле со списком, список, кнопка, рисунок, присоединенная рамка объекта, свободная рамка объекта, набор вкладок, подчиненная форма/отчет, разрыв страницы, линия, прямоугольник и дополнительные элементы ActiveX. Элементы управления могут быть связанными, свободными или вычисляемыми. Связанный элемент управления присоединен к полю базовой таблицы или запроса. Такие элементы управления используются для отображения, ввода или обновления значений из полей базы данных. Для вычисляемого элемента управления в качестве источника данных используется выражение. В выражении могут быть использованы данные из поля базовой таблицы или запроса для формы или отчета, а также данные другого элемента управления формы или отчета. Для свободного элемента управления источника данных не существует. Свободные элементы управления используются для вывода на экран данных, линий, прямоугольников и рисунков.

Надписи. Надписи предназначены для отображения в форме или отчете описательных текстов, таких как заголовки, подписи или краткие инструкции. В надписях не выводятся значения полей или выражений; они всегда являются свободными и не меняются при переходе от записи к записи. Надпись может быть присоединена к другому элементу управления (такую надпись называют подписью).Например, поле создается с присоединенной надписью, которая содержит подпись этого поля. Эта надпись появляется как заголовок столбца в форме в режиме таблицы.

Надпись, созданная с помощью инструмента «Надпись», размещается отдельно и не присоединяется ни к какому элементу управления. Такие надписи используются для отображения разных сведений, например,

заголовков формы или отчета, а также для вывода поясняющего текста. Надписи, не присоединенные к элементам управления, не отображаются в режиме таблицы.

Поля. Поля используются в форме или отчете для отображения данных из таблицы, запроса или инструкции SQL. Поле такого типа называют присоединенным, потому что оно связано с данными в поле в источнике данных. Кроме того, существуют свободные поля. Например, можно создать свободное поле для отображения результатов вычислений или для приема данных, вводимых пользователем. Содержимое свободного поля нигде не сохраняется.

Группы. Группа используется в форме или отчете для вывода ограниченного набора параметров. Группа делает выбор параметра простым и наглядным. В каждый момент времени в группе может быть выбран только один параметр. Группа состоит из рамки группы и набора флажков, переключателей или выключателей. При присоединении группы к полю к нему присоединяется только рамка группы, а не находящиеся в ней флажки, выключатели или переключатели. Пользователь не должен определять свойство Данные (ControlSource) для каждого элемента управления в группе. Вместо этого следует задать в свойстве Значение параметра (OptionValue) каждого флажка, выключателя или переключателя число, являющееся допустимым для поля, к которому присоединена рамка группы. При выборе параметра в группе Microsoft Access вводит в поле значение, равное значению свойства Значение параметра (OptionValue) выбранного элемента. В свойстве Значение параметра (OptionValue) требуется задавать число, так как значением группы может быть только числовое, а не текстовое значение. Microsoft Access сохраняет это число в базовой таблице. Группа может быть также связана с выражением или быть свободной. Свободные группы применяются в специальных диалоговых окнах для принятия данных, вводимых пользователем, и для выполнения действий, основанных на этих данных.

Выключатели. В форме или отчете выключатель может быть использован как отдельный элемент управления, в котором отображаются значения логического поля из базовой таблицы, запроса или инструкции SQL.. Если кнопка выключателя нажата, поле в таблице имеет значение «Да»; если кнопка выключателя не нажата, поле имеет значение «Нет».

Когда пользователь нажимает кнопку выключателя, присоединенного к логическому полю, Microsoft Access отображает значение в базовой таблице в формате, который определяется значением свойства поля Формат поля (Format) («Да»/«Нет», «Истина»/«Ложь» или «Вкл»/«Выкл»). Выключатели особенно удобны при использовании в группах. В такой группе легко видеть, какой из выключателей нажат. Вместо подписи на выключатель можно поместить рисунок.

Свободные выключатели используются также в специальных диалоговых окнах для приема данных, вводимых пользователем.

Переключатели. Флажки. Свойства аналогичны выключателям. Кроме того, флажки включаются в группу для отображения набора выбираемых значений.

Поля со списком. Во многих случаях удобнее выбрать значение из списка, чем вводить конкретное значение с клавиатуры по памяти. Поле со списком позволяет выбрать любой из этих способов ввода значения, не требуя при этом значительного места в форме. Поле со списком является комбинацией двух элементов: поля и раскрывающегося списка. Значение, выбранное или введенное в присоединенное поле со списком, вставляется в поле, к которому присоединено поле со списком. В поле со списком список состоит из строк с данными. Строки содержат один или несколько столбцов, с заголовками или без заголовков. Если поле со списком, содержащим нескольких столбцов, является присоединенным, то сохраняется значение одного из столбцов. Свободное поле со списком позволяет сохранять значение, используемое в другом элементе управления. Например, с помощью свободного поля со списком можно ограничить значения, отбираемые в другом поле со списком или в специальном диалоговом окне. Свободное поле применяется также для поиска записи с помощью значения, выбранного или введенного в поле со списком. Поля со списком имеют свойство Ограничиться списком (LimitToList), которое определяет, допускается ли ввод в поле любых значений или только совпадающих с одним из значений списка. Если в форме достаточно свободного места и требуется, чтобы список постоянно находился на экране, а также если требуется ограничить вводящиеся данные имеющимся списком, вместо поля со списком можно использовать список.

Списки. Во многих случаях удобнее выбрать значение из списка, чем вводить конкретное значение по памяти. Кроме того, выбор из списка позволяет быть уверенным, что введенное значение является допустимым. Список состоит из строк с данными. Строки содержат один или несколько столбцов, которые могут быть снабжены заголовками. Если список из нескольких столбцов является присоединенным, то сохраняется значения одного из столбцов. Свободный список позволяет хранить значение, используемое в другом элементе управления. Например, с помощью свободного списка можно ограничить значения, отбираемые в другом списке или в специальном диалоговом окне. Свободный список применяется также для поиска записи с помощью значения, выбранного в списке. В тех случаях, когда в форме недостаточно места для отображения списка, или если наряду с выбором из списка требуется вводить новые значения с клавиатуры, вместо списка следует использовать поле со списком.

Кнопки. Кнопки используются в формах для выполнения определенного действия или ряда действий. Например, можно создать в форме кнопку, открывающую другую форму. Чтобы кнопка выполняла какое-либо действие, следует создать макрос или процедуру обработки события и связать их со свойством кнопки Нажатие кнопки (OnClick). Мастер кнопок позволяет создавать кнопки более 30 разных типов. При

создании кнопки с помощью мастера для нее определяется процедура обработки события.

Текст надписи на кнопке задается в качестве значения свойства Подпись (Caption). Чтобы поместить на кнопку рисунок, следует указать его в свойстве кнопки Рисунок (Picture).

Подчиненные формы. Подчиненная форма - это форма, находящаяся внутри другой формы. Первичная форма называется главной формой, а форма внутри формы называется подчиненной формой. Комбинацию «форма/подчиненная форма» часто называют также иерархической формой или комбинацией «родительской» и «дочерней» форм. Подчиненная форма удобна для вывода данных из таблиц или запросов, связанных с отношением «один-ко-многим». Например, можно создать форму с подчиненной формой для вывода данных из таблицы «Типы» и из таблицы «Товары». Данные в таблице «Типы» находятся на стороне «один» отношения. Данные в таблице «Товары» находятся на стороне «многие» отношения s каждый тип может иметь несколько товаров. Главная форма и подчиненная форма в этом типе форм связаны таким образом, что в подчиненной форме выводятся только те записи, которые связаны с текущей записью в главной форме. При использовании формы с подчиненной формой для ввода новых записей текущая запись в главной форме сохраняется при входе в подчиненную форму. Это гарантирует, что записи из таблицы на стороне «многие» будут иметь связанную запись в таблице на стороне «один». Это также автоматически сохраняет каждую запись, добавляемую в подчиненную форму. Подчиненная форма может быть выведена в режиме таблицы, или она может быть выведена как простая или ленточная форма. Главная форма может быть выведена только как простая форма.

Главная форма может содержать любое число подчиненных форм, если каждая подчиненная форма помещается в главную форму. Имеется также возможность создавать подчиненные формы двух уровней вложенности. Это означает, что можно иметь подчиненную форму внутри главной формы, а другую подчиненную форму внутри этой подчиненной формы. Например, можно иметь главную форму, в которой выводятся данные о клиентах, подчиненную форму, выводящую данные о заказах и другую подчиненную форму, которая отображает то, что заказано.

Элементы управления, которыми может пользоваться разработчик, представлены на Панели элементов. Ее открывают щелчком на соответствующей кнопке панели инструментов Microsoft Access или командой Вид > Панель элементов. Выбор элемента управления выполняется одним щелчком на его значке в Панели элементов, после чего следующим щелчком в поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его присоединенная надпись. По умолчанию эта надпись стандартная, например для переключателей это Переключатель1, Переключатель2 и т. д. Редактированием свойства элемента управления (доступ к свойствам открывается через контекстное меню) можно дать элементу управления более содержательную подпись.

Основными элементами оформления формы являются текстовые надписи и рисунки. Для создания в форме текстовых надписей служат два элемента управления — Надпись и Поле. В качестве надписи можно задать произвольный текст. Элемент Поле отличается тем, что в нем отображается содержимое одного из полей таблицы, на которой основана форма, то есть при переходе от записи к записи текст может меняться.

Для создания графических элементов оформления служат элементы управления Рисунок, Свободная рамка объекта и Присоединенная рамка объекта. Рисунок выбирается из графического файла и вставляется в форму. Элемент Свободная рамка объекта отличается тем, что это не обязательно рисунок — это может быть любой другой объект OLE, например мультимедийный. Элемент Присоединенная рамка объекта тоже в какой-то степени может служить для оформления формы, но его содержимое берется не из назначенного файла, а непосредственно из таблицы базы данных (если она имеет поле объекта OLE). Естественно, что при переходе между записями содержимое этого элемента будет меняться.

Дизайн формы. В то время как таблицы базы данных скрыты от посторонних глаз, формы базы данных — это средства, с помощью которых с ней общаются пользователи. Поэтому к формам предъявляются повышенные требования по дизайну. В первую очередь, все элементы управления форм должны быть аккуратно выровнены. Это обеспечивается командой Формат > Выровнять. Если нужно равномерно распределить элементы управления по полю формы, используют средства меню Формат > Интервал по горизонтали или Формат - Интервал по вертикали.

Ручное изменение размеров и положения элементов управления тоже возможно, но редко приводит к качественным результатам. При работе вручную используют перетаскивание маркеров, которые видны вокруг элемента управления в тот момент, когда он выделен. Особый статус имеет маркер левого верхнего угла. Обычно элементы управления перетаскиваются вместе с присоединенными к ним надписями. Перетаскивание с помощью этого маркера позволяет оторвать присоединенную надпись от элемента.

Существенную помощь при разработке дизайна формы оказывает вспомогательная сетка. Ее отображение включают командой Вид - Сетка. Автоматическую привязку элементов к узлам сетки включают командой Формат > Привязать к сетке.

Управление последовательностью перехода. Пользователь, для которого, собственно, и разрабатывается форма, ожидает, что ввод данных в нее должен происходить по элементам управления слева направо и сверху вниз. Однако при проектировании сложных форм, когда в процессе дизайна элементы управления многократно перемещаются с места на место, очень легко перепутать их последовательность и создать неудобный порядок ввода данных.

Физически последовательность перехода — это порядок перехода к следующему полю по окончании работы с предыдущим. Она легко проверяется с помощью клавиши TAB. Если при последовательных нажатиях

этой клавиши фокус ввода «мечется» по всей форме, значит, последовательность перехода нерациональна, и ее надо править.

Для управления последовательностью перехода служит диалоговое окно Последовательность перехода. В нем представлен список элементов управления формы. Порядок элементов в списке соответствует текущему порядку перехода. Изменение порядка перехода выполняется перетаскиванием в два приема:

- щелчком на кнопке маркера слева от названия выделяется элемент управления (кнопка мыши отпускается);
- после повторного щелчка с перетаскиванием элемент перемещается на новое место.

Закончив разработку макета формы, ее следует закрыть и сохранить под заданным именем. После открытия формы в окне База данных, с ней можно работать: просматривать или редактировать данные из базовой таблицы. Проверку последовательности перехода выполняют клавишей TAB.

Access, как и любая другая развитая программная система, обладает средствами разработки программных приложений, ориентированных на конечных пользователей. Эти средства базируются на инструментах двух типов: макросах и модулях. Само понятие макроса подразумевает наличие набора некоторых стандартных команд системы, или макрокоманд (допустим, таких, как открытие формы, выполнение запроса, вывод отчета), из которых и конструируется сам макрос.

Макрос может быть как собственно макросом, состоящим из последовательности макрокоманд, так и группой макросов. Группой макросов называют их набор, сохраняемый под общим именем. В некоторых случаях для решения, должна ли в запущенном макросе выполняться определенная макрокоманда, может применяться условное выражение.

Особый интерес вызывает механизм вызова макросов в Access. Для этого существует две принципиальных возможности:

- вызов макроса по команде пользователя (либо непосредственно из раздела Макросы главного окна базы данных, либо с помощью меню или панели инструментов, с которыми он также может быть ассоциирован);
- вызов макроса по некоторому системному событию (открытие или закрытие формы, изменение управляющего элемента и т. п.).

Весьма полезной является возможность организовать автоматическое выполнение ряда действий при открытии базы данных. Для этого они должны быть описаны в специальном макросе с именем Autoexec. Возможности применения макросов при работе в среде СУБД Access можно наглядно продемонстрировать на следующем примере.

Пример. Создадим таблицу в СУБД Access с нужными вам таблицами. Для заполнения таблиц создадим формы. Для объединения форм единым интерфейсом создадим пустую форму, и расположим на ней кнопки вызова созданных форм. Можно воспользоваться встроенной обработкой событий на кнопках (рисунок 1).

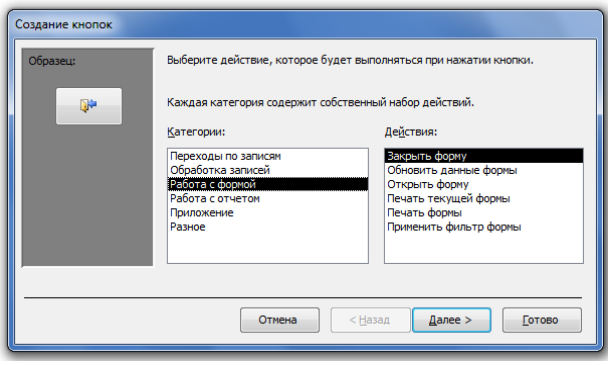


Рисунок 1 – Выбор обработчика событий

Кроме этого, существует возможность самостоятельной обработки событий заданием макросов (рисунок 2)

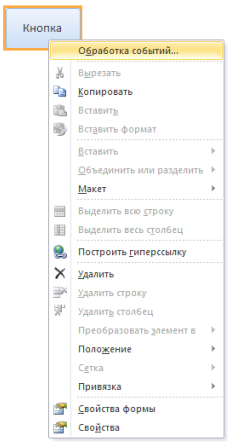
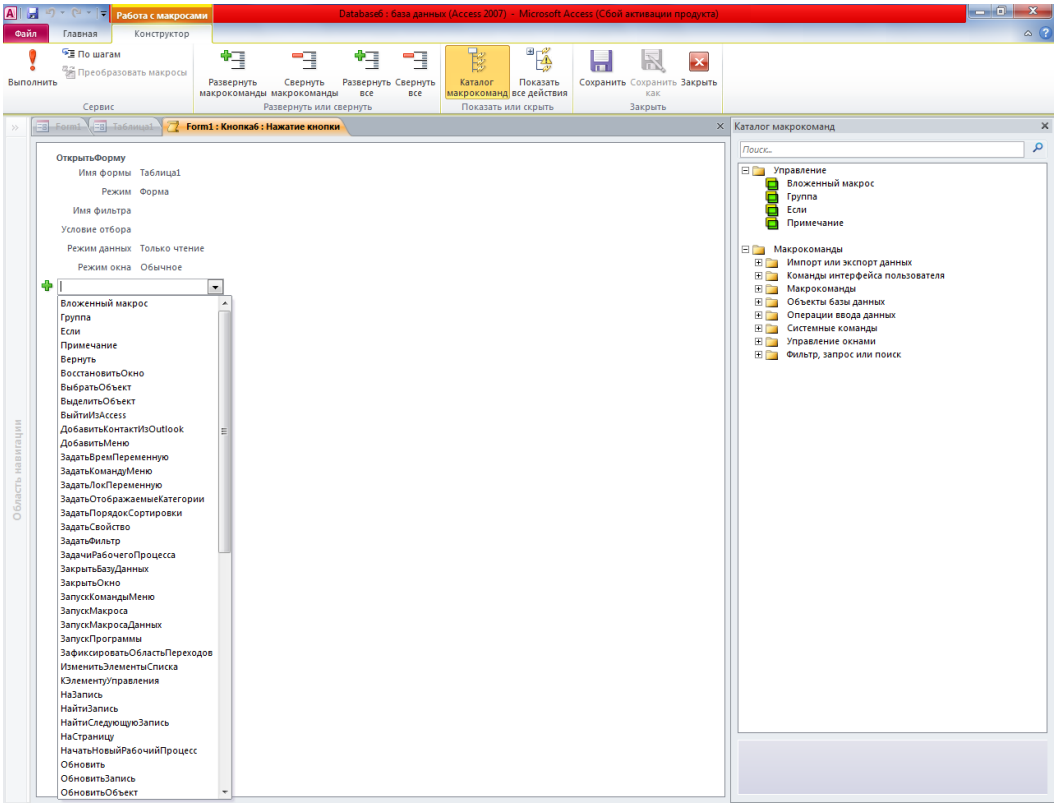


Рисунок 2 – Выбор ручной обработки событий

Для каждого события может быть предусмотрена процедура обработки данного события. И на этом механизме строится большая часть пользовательских приложений, поскольку основой диалогового режима являются формы и отчеты. В таком приложении пользователь или некоторое системное событие запускает процедуру обработки событий, которая написана на языке VBA. Порядок выполнения программ зависит от порядка возникновения событий. В простых приложениях в программе на VBA ограничиваются созданием процедур обработки событий и процедур функций, которые используются в выражении. Программа на VBA записывается как совокупность процедур типа Sub и Function. Процедура состоит из последовательности инструкций и методов, с помощью которых решаются задачи, требуемые действия или производятся расчеты. Процедуры, входящие в состав приложения, хранятся в модулях в БД, однако запросить выполнение модуля в целом невозможно. Выполняться могут только процедуры, содержащиеся в модуле. Модули содержат набор

процедур и описание переменных, констант и объектов, которые используются в процедурах.



Программы на Visual Basic используют вместо макросов в следующих случаях.

1. Упрощение управления базой данных. Поскольку макросы являются объектами, существующими отдельно от использующих их форм и отчетов, поддержание базы данных, в которой реакция на события в формах и отчетах определяется многими макросами, становится достаточно затруднительным. В отличие от этого процедуры обработки события Visual Basic являются встроенными в описания соответствующих форм и отчетов. При переносе формы или отчета из одной базы данных в другую встроенные процедуры обработки события автоматически переносятся вместе с формой или отчетом.

2. Создание пользовательских функций. В MS Access определен ряд встроенных функций, например функция IPmt, которая рассчитывает проценты по платежам. Встроенные функции можно использовать для выполнения вычислений без необходимости разрабатывать сложные выражения. Язык Visual Basic позволяет пользователям создавать также собственные функции как для решения задач, выходящих за рамки возможных для встроенных функций, так и для замены сложных выражений, содержащих встроенные функции.

3. Скрытие сообщений об ошибках. Стандартные сообщения об ошибках Microsoft Access, выводимые на экран при возникновении нештатных ситуаций во время работы с базой данных, могут оказаться

малопонятными для пользователя, особенно если этот пользователь плохо знаком с MS Access. С помощью Visual Basic можно перехватывать ошибку при ее возникновении и либо выводить собственное сообщение об ошибке, либо предпринимать определенные действия.

4. Создание или обработка объектов. В большинстве случаев удобнее создавать или изменять объекты в режиме конструктора. Однако в некоторых ситуациях приходится работать с описанием объекта в программе. Средства Visual Basic позволяют выполнять обработку всех объектов в базе данных, а также самой базы данных.

5. Выполнение действий на уровне системы. Выполнение в макросе макрокоманды Запуск Приложения (RunApp) позволяет запускать из собственного приложения другое приложение, работающее в среде Microsoft Windows или MS-DOS, однако этим возможности использования макроса вне Microsoft Access практически исчерпываются. Средства Visual Basic позволяют проверять существование файлов, использовать программирование объектов или динамический обмен данными (DDE) для связи с другими приложениями для Windows, например Microsoft Excel, а также вызывать функции из библиотек динамической компоновки (DLL) Windows.

6. Обработка записей по одной. Инструкции Visual Basic позволяют перебирать наборы записей по одной и выполнять определенные действия над отдельной записью. В отличие от этого макросы позволяют работать только с целым набором записей.

7. Передача аргументов в процедуры Visual Basic. Аргументы для макрокоманд можно задавать в нижней части окна макроса при его создании, но при выполнении макроса изменять их невозможно. Однако при помощи Visual Basic можно передавать аргументы в выполняемую программу или использовать в качестве значений аргументов переменные; макросы не позволяют делать это. Передача аргументов повышает гибкость выполнения процедур Visual Basic.

Главная кнопочная форма создается с целью навигации по базе данных. Эта форма может использоваться в качестве главного меню БД. Элементами главной кнопочной формы являются объекты форм и отчётов.

Запросы и таблицы не являются элементами главной кнопочной формы. Поэтому для создания кнопок Запросы или Таблицы на кнопочной форме можно использовать макросы. Сначала в окне базы данных создают макросы «Открыть Запрос» или «Открыть Таблицу» с уникальными именами, а затем в кнопочной форме создают кнопки для вызова этих макросов.

Для одной базы данных можно создать несколько кнопочных форм. Кнопки следует группировать на страницах кнопочной формы таким образом, чтобы пользователю было понятно, в каких кнопочных формах можно выполнять определенные команды (запросы, отчеты, ввода и редактирования данных). Необходимо отметить, что на подчиненных

кнопочных формах должны быть помещены кнопки возврата в главную кнопочную форму.

Технология создания кнопочных форм следующая:

– создать страницу главной кнопочной формы (ГКФ);

– создать необходимое количество страниц подчиненных кнопочных форм (например, формы для ввода данных, для отчетов, для запросов и т.д.);

– создать элементы главной кнопочной формы;

– создать элементы для кнопочных форм отчетов и форм ввода или изменения данных;

– создать макросы для запросов или для таблиц с уникальными именами;

– создать элементы для кнопочных форм запросов или таблиц.

Структура кнопочных форм может быть представлена в следующем виде.

Задания

1 Изучить теоретические сведения.

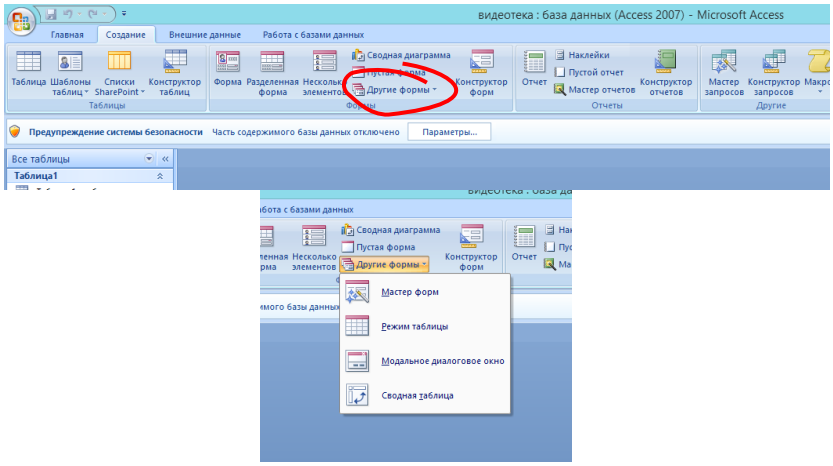
2 В соответствии с вариантом задания создать экранные формы с кнопками, списками, подчиненными формами и другими элементами управления.

3 В соответствии с вариантом задания создать обработчики событий на элементах управления.

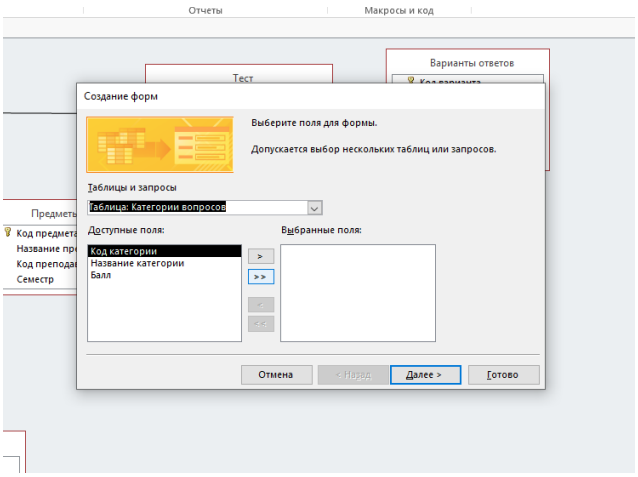
4 В соответствии с вариантом задания создать кнопочные формы на элементах управления.

Порядок выполнения работы

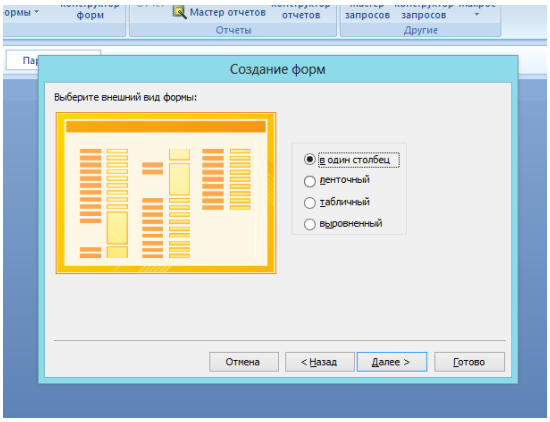
1 Создание форм с помощью мастера, который находится в разделе меню «Создание», «Другие формы», «Мастер форм».



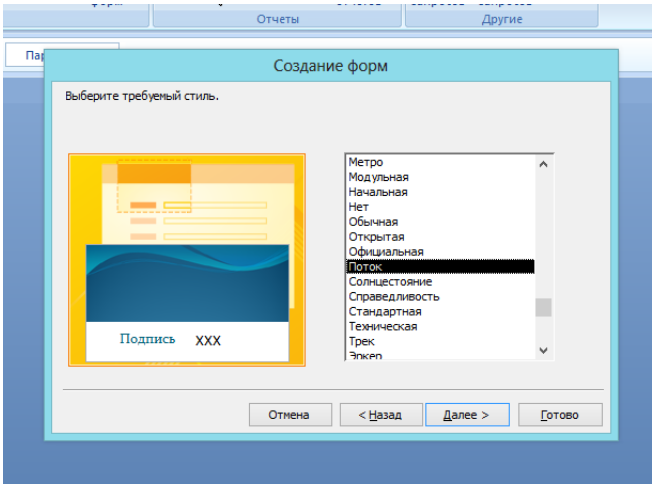
2 На первом этапе работы Мастера форм выбирают таблицы и поля, которые войдут в будущую форму.



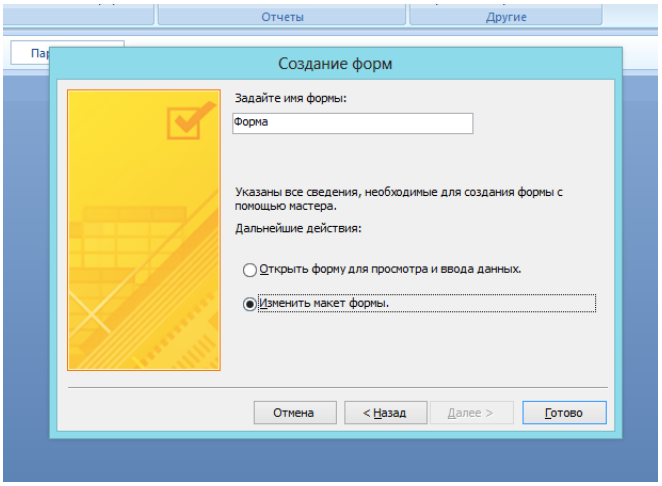
3 На втором этапе выбирается внешний вид формы.



4 На третьем этапе выбирается стиль оформления формы. В некоторых версиях Ms Access этот этап пропущен, т.к. оформление можно задать с помощью конструктора.



5 На последнем этапе выполняется сохранение формы под заданным именем. Здесь же можно включить переключатель - Изменить макет формы, который открывает только что созданную форму в режиме Конструктора. Этим удобно воспользоваться в учебных целях, чтобы рассмотреть структуру формы на готовом примере.

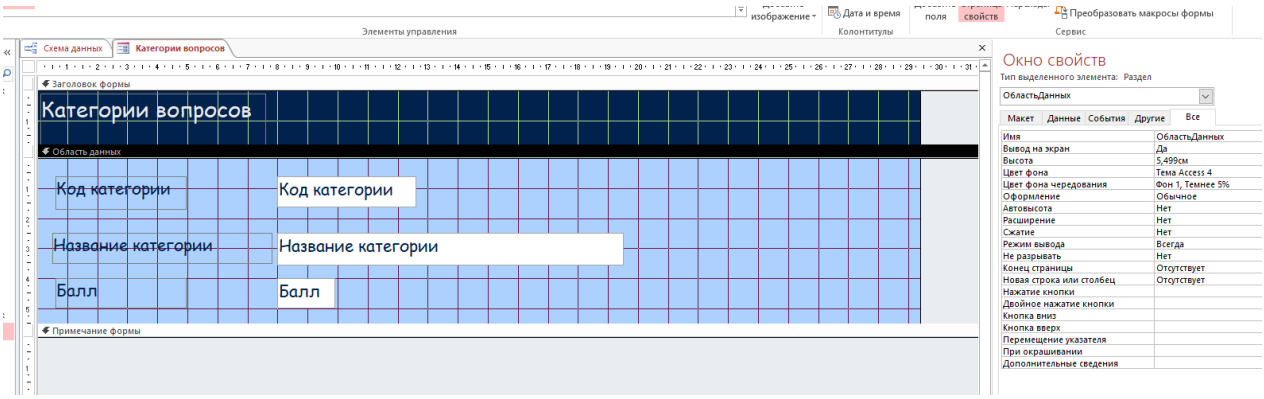


Форма в режиме конструктора имеет три основных раздела:

- область заголовка.
- область данных.
- область примечания.

Линии, разделяющие разделы, перетаскиваются по вертикали с помощью мыши — это позволяет изменять размеры разделов так, как требуется.

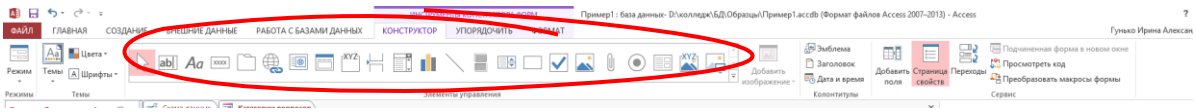
Разделы заголовка и примечания имеют чисто оформительское назначение — их содержимое напрямую не связано с таблицей или запросом, на котором основана форма. Раздел данных имеет содержательное значение — в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод.



Разработчик формы может разместить здесь дополнительные элементы управления для автоматизации ввода данных (переключатели, флажки, списки и другие, типичные для приложений Windows).

Элементы управления формы.

6 Элементы управления, которыми может пользоваться разработчик, представлены на Панели элементов. Ее открывают щелчком на соответствующей кнопке панели инструментов Microsoft Access или командой Вид --> Панель элементов.



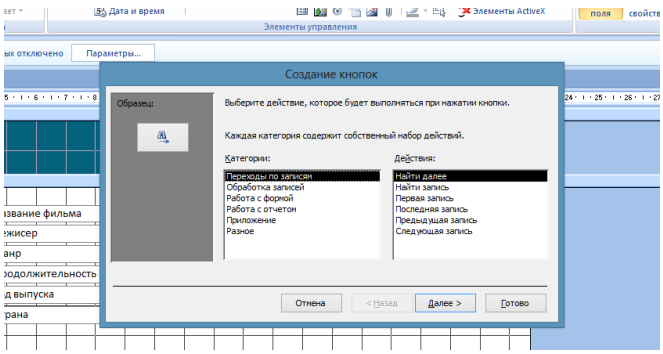
Выбор элемента управления выполняется одним щелчком на его значке в Панели элементов, после чего следующим щелчком в поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его присоединенная надпись. По умолчанию эта надпись стандартная, например, для переключателей это Переключатель 1, Переключатель 2 и т. д. Редактированием свойства элемента управления (доступ к свойствам открывается через контекстное меню) можно дать элементу управления более содержательную подпись.

Основными элементами оформления формы являются текстовые надписи и рисунки. Для создания в форме текстовых надписей служат два элемента управления — Надпись и Поле. В качестве надписи можно задать произвольный текст. Элемент Поле отличается тем, что в нем отображается содержимое одного из полей таблицы, на которой основана форма, то есть при переходе от записи к записи текст может меняться.

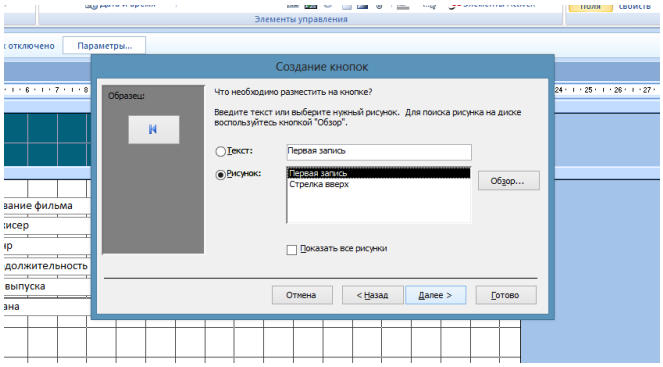
Для создания графических элементов оформления служат элементы управления Рисунок, Свободная рамка объекта и Присоединенная рамка объекта. Рисунок выбирается из графического файла и вставляется в форму. Элемент Свободная рамка объекта отличается тем, что это не обязательно рисунок — это может быть любой другой объект OLE, например мультимедийный. Элемент Присоединенная рамка объекта тоже в какой-то степени может служить для оформления формы, но его содержимое берется не из назначенного файла, а непосредственно из таблицы базы данных (если

она имеет поле объекта OLE). Естественно, что при переходе между записями содержимое этого элемента будет меняться.

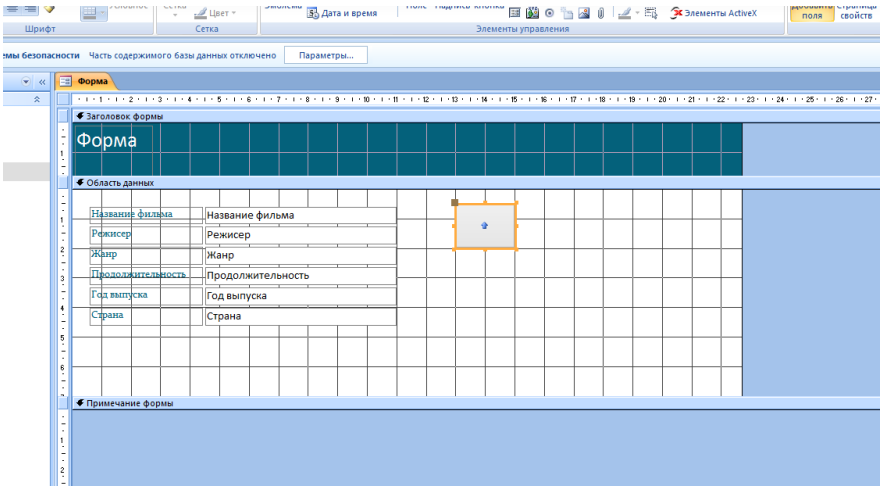
7 Добавим в форму базы данных управляющие кнопки, для этого выберем на панели элементов раздел кнопка, если включен режим «Использовать мастер», то при размещении кнопки на форме, появится окно мастера «Создание кнопок». В нем выберем категорию «Переходы по записям», а в действиях «Первая запись», затем нажмем «Далее»



Следующее окно – внешний вид кнопки:



Выберем подходящий вариант и нажмем «Готово», в окне формы появится кнопка, которую можно перемещать, изменять ее размер мышью.

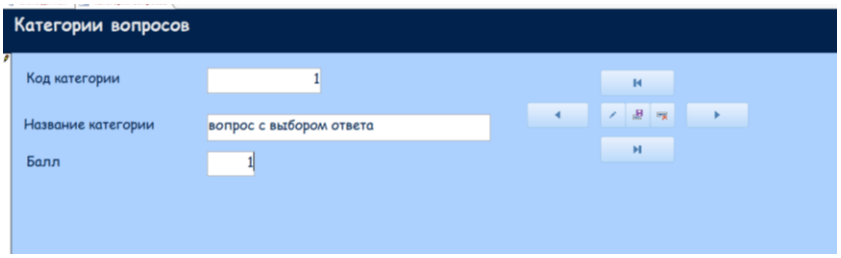


8 Таким образом создать в форме следующие кнопки:

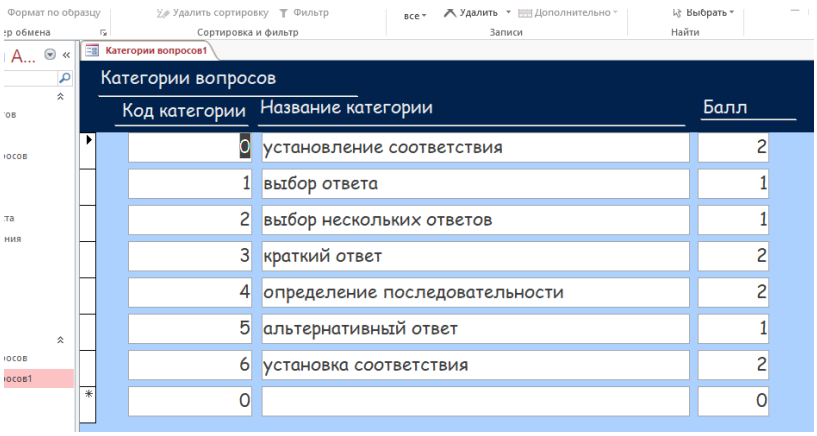
- Первая запись
- Последняя запись
- Следующая запись
- Предыдущая запись
- Добавить запись
- Сохранить запись
- Удалить запись

9 Сохраним форму и откроем ее в режиме использования двойным нажатием левой кнопки мыши.

10 Просмотрим таблицу, создадим и сохраним запись таблицы с помощью формы.

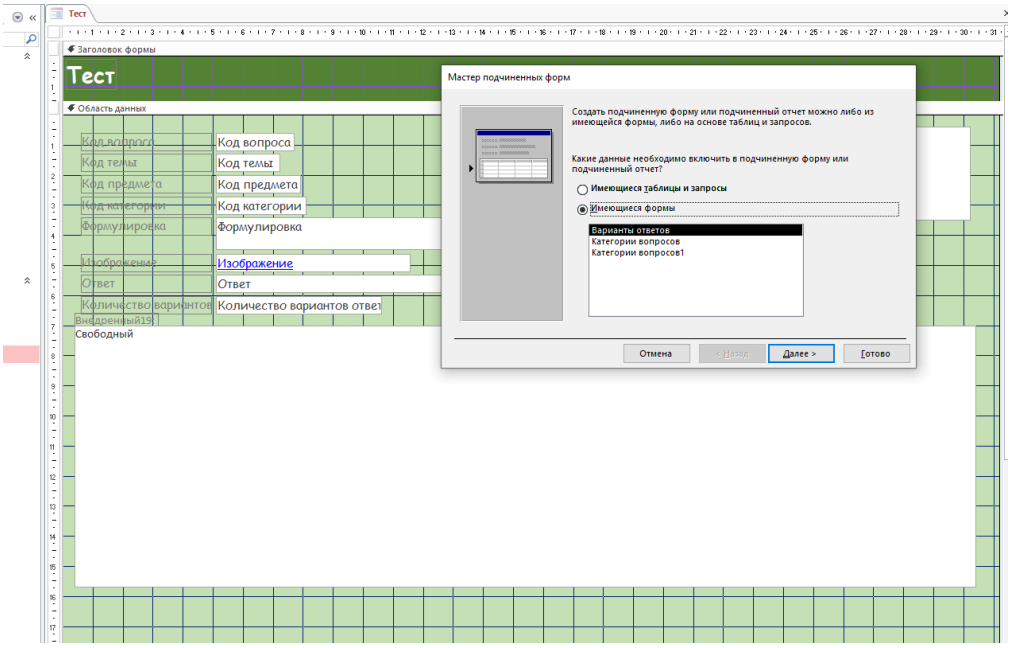


Или ленточный вид, который можно использовать в подчиненной форме:



11 Для включения подчиненной формы в главную надо сначала создать эту форму, причем выбрать вид отображения данных в этой форме: **ленточный** (tabular) или **табличный** (datasheet).

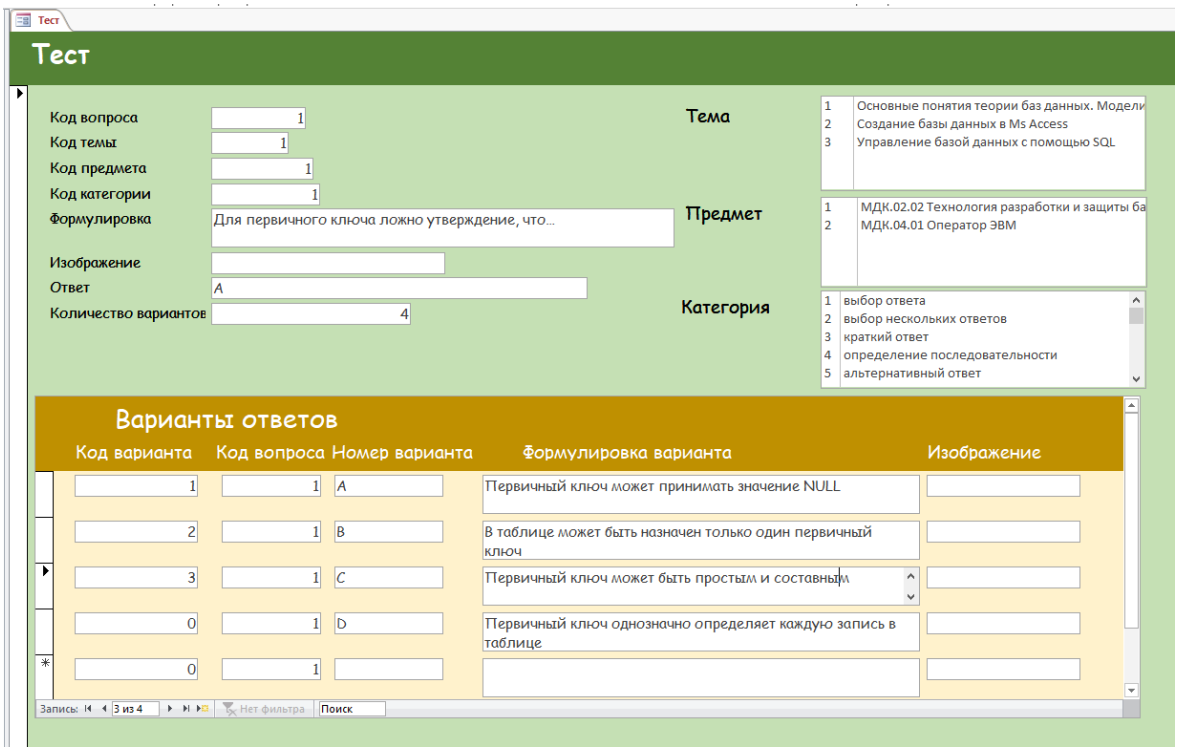
12 На вкладке Конструктор в группе Элементы управления выберите элемент Подчиненная форма/отчет. В форме щелкните место, куда нужно поместить подчиненную форму. В открывшемся окне выбрать Имеющиеся формы, а в списке созданных форм выбрать нужную.



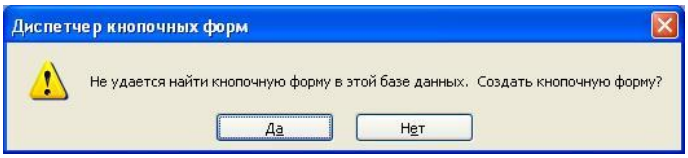
В процессе доработки формы, используя технику редактирования формы, можно перемещать поля в главной форме, менять их свойства, в том числе шрифт и размеры, подпись поля, формировать текст в заголовке формы. Аналогичные действия по доработке выполняются для подчиненной формы.

Добавим на форму список полей из другой таблицы, для этого на панели инструментов необходимо выбрать инструмент Список, в окне мастера выбрать нужную таблицу и поля в ней, следуя инструкции мастера.

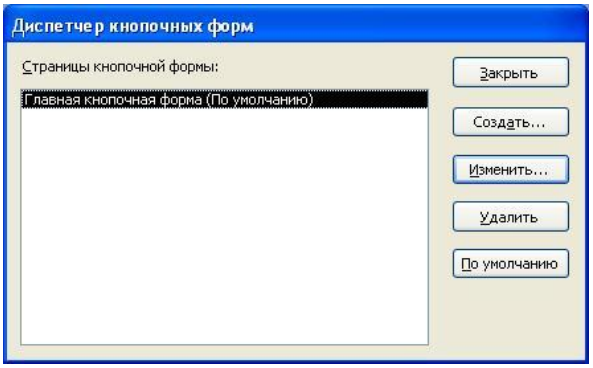
По окончанию редактирования формы, сохраним изменения и запустим ее на выполнение.



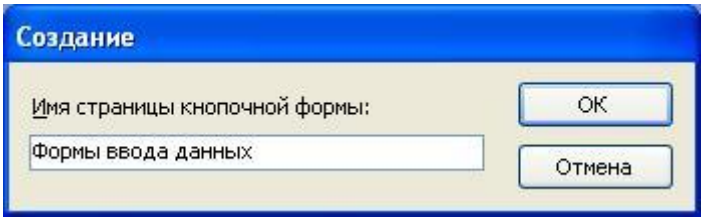
Для создания главной кнопочной формы и ее элементов необходимо открыть базу данных, (например, «Успеваемость_ студентов») и выполнить команду Диспетчер кнопочных форм. Если эта команда отсутствует на панели быстрого доступа, щелкнуть правой кнопкой по этой панели и выбрать Настроить панель …, в окне команд выбрать все команды, найти диспетчер и добавить его на панель, предварительно создав группу. Если кнопочная форма ранее не создавалась, то откроется окно диалога «Диспетчер кнопочных форм».



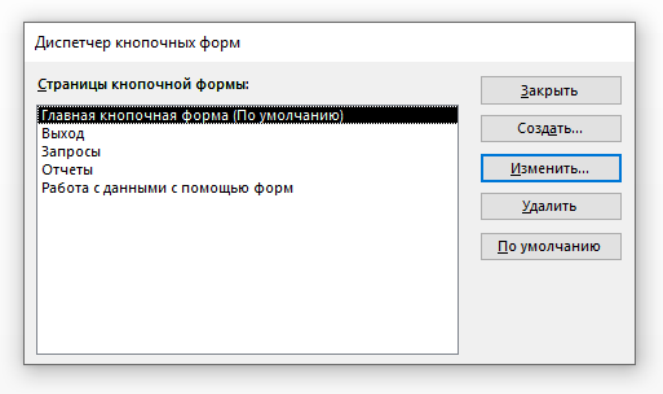
В окне диалога надо нажать кнопку «Да», тем самым подтвердить создание кнопочной формы. В результате будет создана страница Главной кнопочной формы.



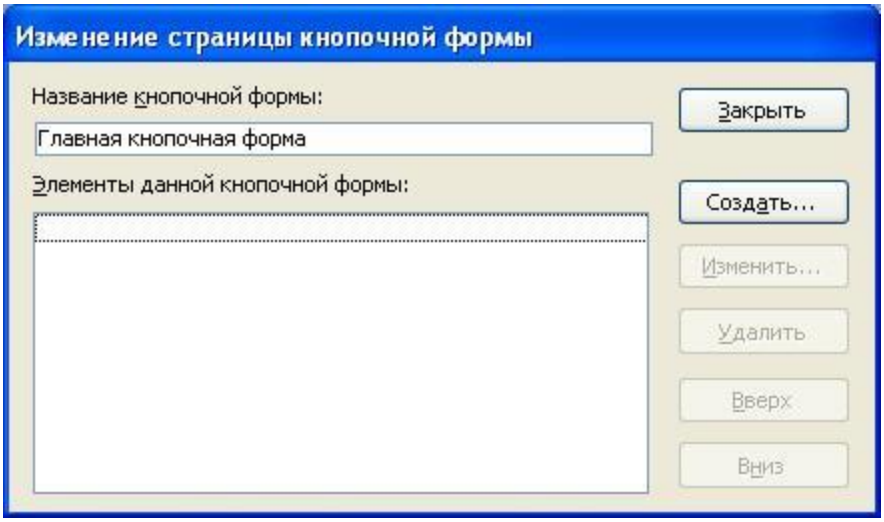
Далее можно создать еще три страницы кнопочной формы: Формы ввода данных, Отчеты и Запросы. Для этого следует щелкнуть на кнопке «Создать» и в появившемся окне ввести имя новой страницы «Формы ввода данных» и щелкнуть на кнопке «ОК».



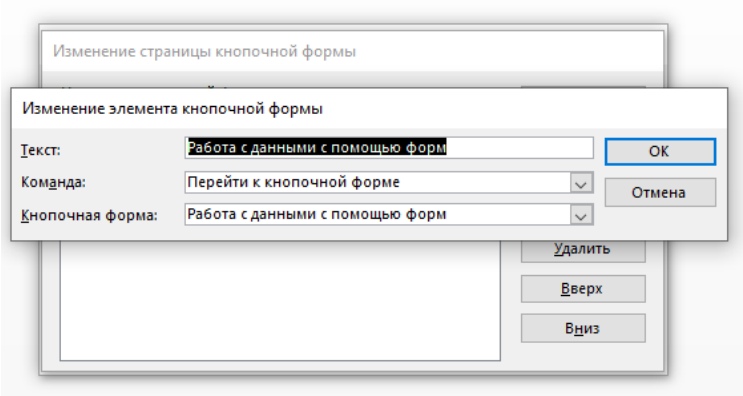
Будет создана страница кнопочной формы «Формы ввода данных». Аналогичным образом надо создать еще две страницы, в итоге получим четыре страницы кнопочных форм, которые отображаются в окне «Диспетчер кнопочных форм».



После этого создаем элементы ГКФ, для этого в «Окне диспетчер кнопочных форм» выделяем страницу «Главная кнопочная форма» и щелкаем «Изменить», откроется новое окно «Изменение страниц кнопочной формы».



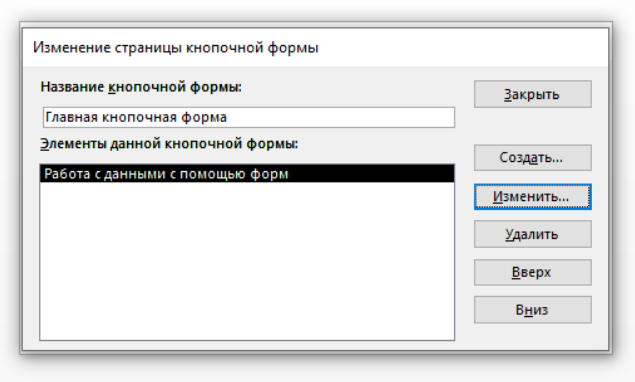
В этом окне щелкаем на кнопке «Создать», откроется новое окно «Изменение элемента кнопочной формы».



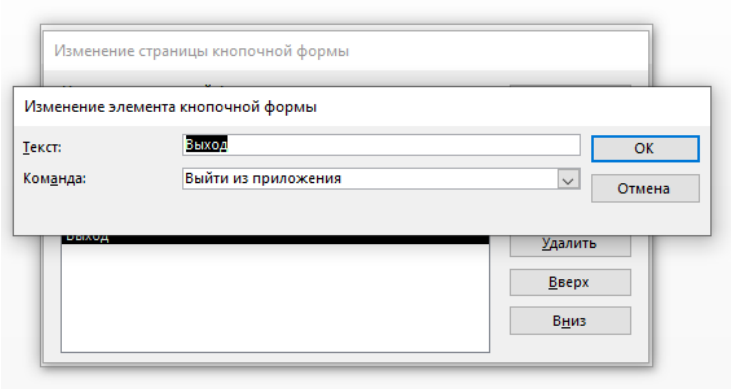
В окне выполняем следующее:

- вводим текст: Формы для ввода данных;
- выбираем из раскрывающегося списка команду: Перейти к кнопочной форме;

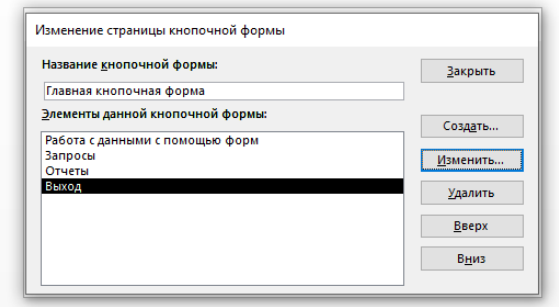
- выбираем из списка кнопочную форму: Ввод данных в формы, щелкаем на кнопке «ОК».
  В окне «Изменение страницы кнопочной формы» отобразится элемент кнопочной формы «Формы для ввода данных».



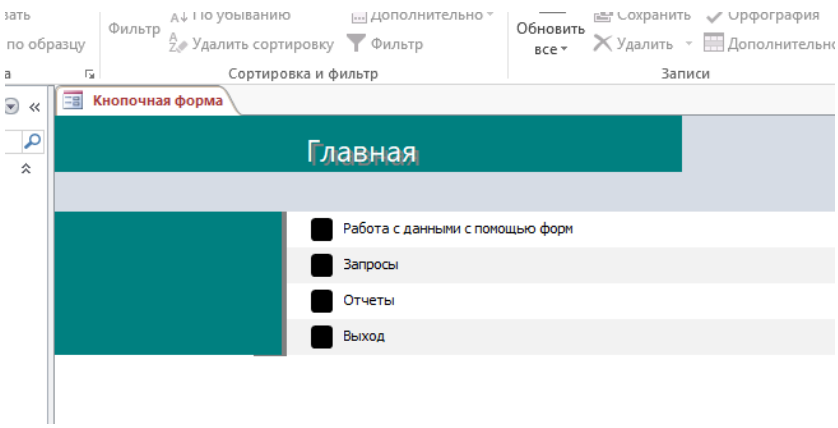
Аналогичным методом надо создать элементы: «Запросы» и «Отчеты», а затем элемент (кнопку) "Выход из БД".



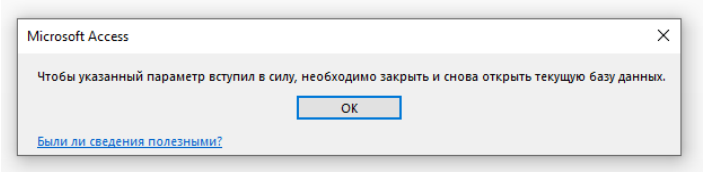
В результате в окне «Изменение страницы кнопочной формы» будут отображаться все элементы главной кнопочной формы.



Кнопочная форма появится в списке в области окна базы данных на вкладке Формы на панели Объекты, а на вкладе Таблицы в списках появится таблица Switchboard Items. Дважды щелкнув на надписи "Кнопочная форма", откроется Главная кнопочная форма.

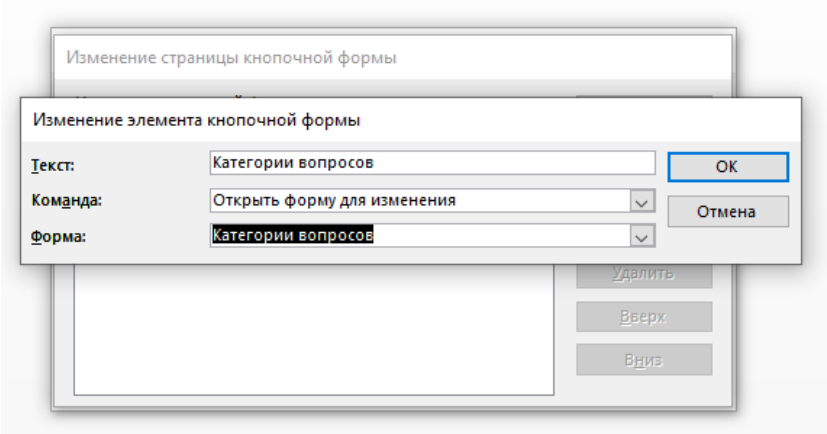


Для того чтобы эта форма отображалась при запуске базы данных, необходимо выполнить команду Файл/Параметры/Текущая база данных, и в открывшемся окне в разделе Параметры просмотра выбрать "Кнопочная форма" из раскрывающегося списка, можно также ввести заголовок и значок приложения. Нажать Ок, появится сообщение Ms Access.

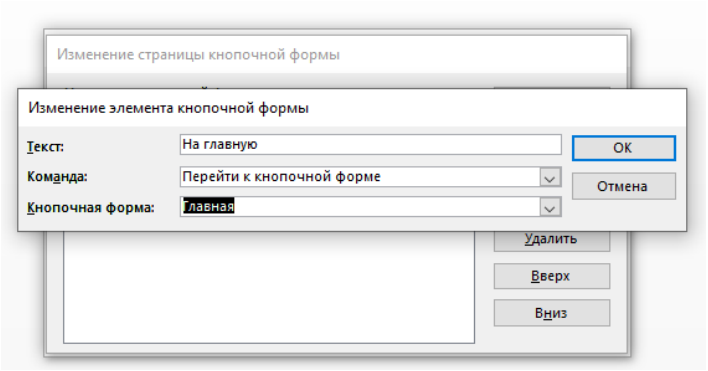


Но на этом создание кнопочных форм еще не законченно, так как на подчиненных кнопочных формах (Формы ввода данных, Отчеты, Запросы) нет элементов.

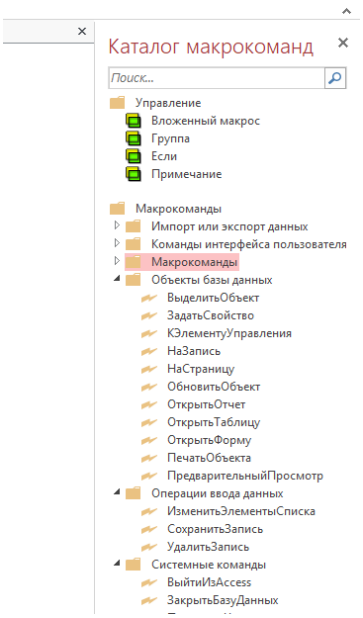
В «Окне диспетчер кнопочных форм» выделяем страницу «Работа с данными с помощью форм» и щелкаем «Изменить», откроется новое окно «Изменение страниц кнопочной формы». Аналогично заполняем страницу окна Работа с данными с помощью форм. Для этого выбираем Создать, заполняем появившееся окно, нажимаем Ок.



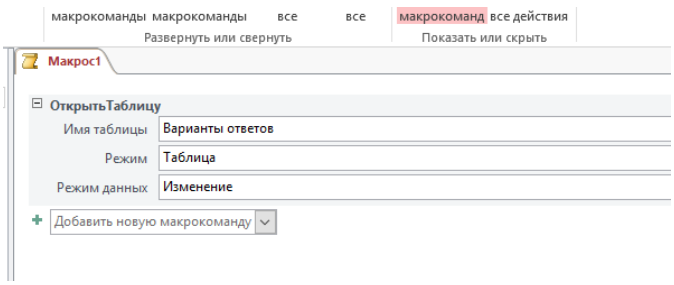
Так же выписываем все формы в базе данных, затем добавляем кнопку возврата на главную форму.



Аналогично можно сформировать форму Отчеты. В разделе Команда выбирать Открыть отчет. Для организации форм Таблицы и Запросы необходимо создать макросы. Для этого в разделе меню Создание выбираем Макрос. В правой части экрана появился список макрокоманд.

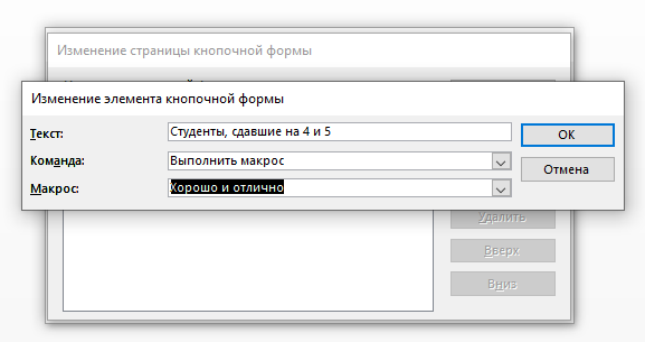


В каталоге выбираем «Открыть таблицу» в разделе «Объекты базы данных», перетаскиваем команду на лист макроса.

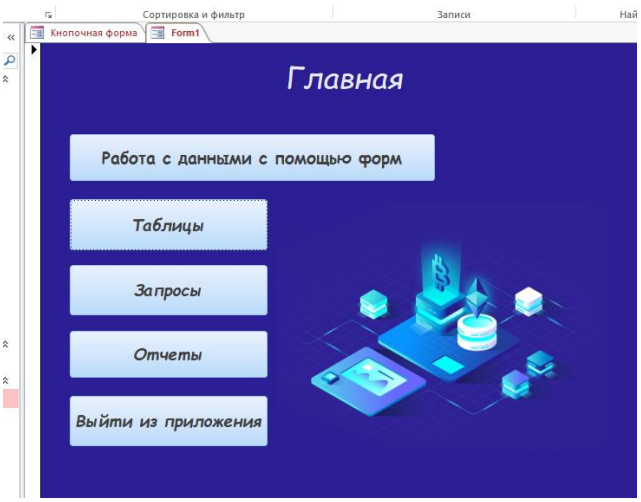


Выбираем имя таблицы и режим данных, сохраняем макрос с именем по имени таблицы и запускаем его. В результате открывается таблица для редактирования. Аналогично создать макросы для каждой таблицы. Макрос для открытия запроса создается аналогично таблице. Если необходимо внести изменения в имеющийся макрос, откроем его в режиме конструктора и продолжим его разработку.

Продолжим создание кнопочной формы, для чего выберем в меню диспетчер кнопочных форм и оформим формы запросов и таблиц



После создания кнопочной формы можно немного изменить ее внешний вид с помощью конструктора, правда больших возможностей для изменения не предоставляется, можно поменять цвета, начертание и размер шрифта, добавить рисунок в область заголовка. Можно создать собственную кнопочную форму для работы, начав ее создание с пустой формы, разместить кнопки и другие элементы управления, например.



Содержание отчета

1 Название работы
2 Цель работы
3 Перечень технических средств обучения
4 Порядок выполнения работы
5 Ответы на контрольные вопросы
6 Вывод

Варианты заданий

Варианты заданий представлены в практической работе № 39

Контрольные вопросы:

1. Назначение форм
2. С помощью каких инструментов формы осуществляется работа с данными таблиц?
3. Какие управляющие элементы используются в формах?
4. Как в форму добавить нужные элементы управления?
5. Что такое макрос и для каких целей используется?
6. Что такое модуль и чем он отличается от макроса?
7. В каких случаях используется Visual Basic в Ms Access?

Используемая литература

– Г.Н.Федорова Основы проектирования баз данных. М.: Академия, 2020

– Г.Н.Федорова Разработка, администрирование и защита баз данных. М.: Академия, 2018