МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО

СПЕЦИАЛЬНОГО

ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени АБУ РАЙХАНА БЕРУНИ

Факультет: Электроники и Автоматики

КУРСОВАЯ РАБОТА

По предмету: Информатика и информационные технологии.

Выполнил: Мир-Исаев Ф.А.

Группа: 118-14 И.Т. (у)

Приняла : Каримова Д.К.

Ташкент 2015

СОДЕРЖАНИЕ

1. Введение……. 3
2. Постановка задачи ……. 7
3. Теоретическая часть ……. 8
4. Блок схема б.д. ……. 14
5. Последовательность создании базы данных…… 15
6. Описание программы……. 25
7. Результат ……. 26
8. Заключение……. 27

Введение

Современное материальное производство и другие сферы деятельности все больше нуждаются в информационном обслуживании, переработке огромного количества информации. Универсальным техническим средством обработки любой информации является компьютер, который играет роль усилителя интеллектуальных возможностей человека и общества в целом, а коммуникационные средства, использующие компьютеры, служат для связи и передачи информации. Появление и развитие компьютеров - это необходимая составляющая процесса информатизации общества.

Информатизация общества является одной из закономерностей современного социального прогресса. Этот термин все настойчивее вытесняет широко используемый до недавнего времени термин «компьютеризация общества». При внешней похожести этих понятий они имеют существенное различие.

При *компьютеризации общества* основное внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение результатов переработки информации и ее накопление.

При *информатизации общества* основное внимание уделяется комплексу мер, направленных на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех видах человеческой деятельности.

Таким образом, «информатизация общества» является более широким понятием, чем «компьютеризация общества», и направлена на скорейшее овладение информацией для удовлетворения своих потребностей. В понятии «информатизация общества» акцент надо делать не столько на технических средствах, сколько на сущности и цели социально-технического прогресса. Компьютеры являются базовой технической составляющей процесса информатизации общества.

Информатизация на базе внедрения компьютерных и телекоммуникационных технологий является реакцией общества на потребность в существенном увеличении производительности труда в информационном секторе общественного производства, где сосредоточено более половины трудоспособного населения. Так, например, в информационной сфере США занято более 60% трудоспособного населения, в СНГ — около 40%.

С современной точки зрения использование телефона в первые годы его существования выглядит довольно смешно. Руководитель диктовал сообщение своему секретарю, который затем отправлял его из телефонной комнаты. Телефонный звонок принимали в аналогичной комнате другой компании, текст фиксировали на бумаге и доставляли адресату. Потребовалось много времени, прежде чем телефон стал таким распространенным и привычным способом сообщения, чтобы его стали, использовать, так, как мы это делаем сегодня: сами звоним в нужное место, а с появлением сотовых телефонов – и конкретному человеку.

В наши дни компьютеры, в основном, применяются как средства создания и анализа информации, которую затем переносят на привычные носители (например, бумагу). Но теперь, благодаря широкому распространению компьютеров и созданию Интернета, впервые можно при помощи своего компьютера общаться с другими людьми через их компьютеры. Необходимость использования распечатанных данных для передачи коллегам устраняется подобно тому, как бумага исчезла из телефонных переговоров. Сегодняшний день, благодаря использованию Web, можно сравнить с тем временем, когда люди перестали записывать текст телефонных сообщений: компьютеры (и их связь между собой посредством Интернета) уже настолько широко распространены и привычны, что мы начинаем использовать их принципиально новыми способами. WWW – это начало пути, на котором компьютеры по – настоящему станут средствами связи.

Интернет предоставляет беспрецедентный способ получения информации. Каждый, имеющий доступ к WWW, может получить всю имеющуюся на нем информацию, а также мощные средства ее поиска. Возможности для образования, бизнеса и роста взаимопонимания между людьми становятся просто ошеломляющими. Более того, технология Web позволяет распространять информацию повсюду. Простота этого способа не имеет аналогов в истории. Для того чтобы сделать свои взгляды, товары или услуги известными другим, больше нет необходимости покупать пространство в газете или журнале, платить за время на телевидении и радио. Web делает правила игры одинаковыми для правительства и отдельных лиц, для малых и больших фирм, для производителей и потребителей, для благотворительных и политических организаций. WorldWideWeb (WWW) на Интернете – это самый демократичный носитель информации: с его помощью любой может сказать и услышать сказанное без промежуточной интерпретации, искажения и цензуры, руководствуясь определенными рамками приличия. Интернет обеспечивает уникальную свободу самовыражения личности и информации.

Подобно использованию внутренних телефонов компаний для связи сотрудников между собой и внешним миром, Web применяется как для связи внутри организации, так и между организациями и их потребителями, клиентами и партнерами. Та же самая технология Web, которая дает возможность небольшим фирмам заявить о себе на Интернете, крупной компанией может использоваться для передачи данных о текущем состоянии проекта по внутренней интрасети, что позволит ее сотрудникам всегда быть более осведомленными и, значит, более оперативным по сравнению с небольшими, проворными конкурентами. Применение интрасети внутри организации для того, чтобы сделать информацию более доступной для своих членов, также является шагом вперед по сравнению с прошлым. Теперь, вместо того, чтобы хранить документы в запутанном компьютерном архиве, появилась возможность (под контролем средств защиты) легко производить поиск и описание документов, делать ссылки на них и составлять указатели. Благодаря технологии Web бизнес, равно как и управления, становится более эффективным.

Информационные технологии

Информационная технология - это совокупность методов, производственных процессов и программно-технических средств, объединенных в технологическую цепочку, обеспечивающую сбор, обработку, хранение, передачу и отображение информации.

Цель функционирования этой цепочки, т.е. информационной технологии, - это снижения трудоемкости процессов использования информационного ресурса и повышение их надежности и оперативности.

Эффективность информационной технологии определяется, в конечном счете, квалификацией субъектов процессов информатизации. При этом технологии должны быть максимально доступны потребителям.

Можно классифицировать информационные технологии с различных точек зрения. Например:

Информационные технологии можно различать по типу обрабатываемой информации. Разделение достаточно условное, т.к. большинство информационных технологий позволяет поддерживать и другие виды информации. Например, в текстовых процессорах возможна и несложная расчетная деятельность, а табличные процессоры обрабатывают не только цифровую информацию, но и могут генерировать графики. Однако каждая из видов технологии в основном ориентирована на работу с информацией определенного вида. Модификация элементов, составляющих информационные технологии, дает возможность образования новых технологий в различных компьютерных средах.

Информационные технологии можно разделить на обеспечивающие (ОИТ) и функциональные (ФИТ).

Обеспечивающие технологии - это технологии обработки информации, которые могут использоваться как инструментарий в различных предметных областях. При этом они могут обеспечивать решение задать разного плана и разной степени сложности. ОИТ могут быть разделены по классам задач, в зависимости от класса ОИТ используют разные виды компонентов и программных средств. При объединении ОИТ по предметному признаку возникает проблема системной интеграции, т.е. приведение различных технологий к единому стандартному интерфейсу.

Функциональные информационные технологии (ФИТ) - это модификация обеспечивающих технологий для задач определенной предметной области, т.е. реализуется предметная технология. Предметные технологии и информационная технология влияют друг на друга. Например, появление пластиковых карточек как носителей финансовой информации принципиально изменила предметную технологию. При этом пришлось создавать совершенно новую информационную технологию. Но, в свою очередь, возможности, представленные новой ИТ, повлияли на предметную технологию пластиковых носителей (в области их защиты, например).

Информационные технологии классифицируются по типам пользовательского интерфейса. Можно выделить системный и прикладной интерфейс.

Прикладной интерфейс связан с реализацией функциональных информационных технологий. Системный интерфейс - это набор приемов взаимодействия с компьютером, который реализуется операционной системой или ее надстройкой.

Большинство обеспечивающих и функциональных технологий используются пользователем без посредников - программистов. Пользователь может самостоятельно изменять последовательность применения тех или иных технологий. С точки зрения участия или не участия пользователя в информационном процессе, технологии можно разделить на пакетные и диалоговые.

Задачи, решаемые в пакетном режиме, характеризуются следующими свойствами:

- алгоритм решения задачи формализован, процесс не требует вмешательства человека.
- имеется большой объем входных и выходных данных; значительная их часть хранится на магнитных носителях.
- большое время решения задач, обусловленное объемами данных.
- регламентность, т.е. задачи решаются с заданной периодичностью.

Диалоговый режим это не альтернатива пакетному режиму, а его развитие. Диалоговый режим позволяет пользователю вмешаться в процесс решения задачи, он отпускает пользователя, отменяет жестко закрепленную последовательность обработки данных. Применение режимов зависит в первую очередь от предметной технологии.

Можно классифицировать информационные технологии по степени их взаимодействия между собой. Например, дискретное и сетевое взаимодействие; взаимодействие с использованием различных вариантов обработки и хранения данных; распределенная информационная база и распределенная обработка данных. Эту классификацию информационных технологий можно изобразить с помощью схемы.

Постановка задачи

15- Вариант.

Создать базу данных: Сотрудников кафедры.

Разработать структуру базы данных, каждая запись которой содержит не менее 10 полей. Подготовить исходные данные для разработанной БД (не менее 15 записей). Создать разработанную БД на ПК и заполнить ее исходными данными. Просмотреть созданную БД и, в случае необходимости, внести необходимые корректировки. Перекопировать исходную БД в БД2. Предусмотреть выборку данных из исходной БД по какому- то признаку и выдачу их на

Теоретическая часть

Работа с базами данных в Borland C++ Builder.

Используя Borland C++ Builder, можно создать приложения, работающие как с однопользовательскими базами данных (БД), так и с серверными СУБД, такими как Oracle, Sybase, Informix, Interbase, MS SQL Server, DB2, а также с ODBC-источниками. Возможности C++ Builder, связанные с созданием приложений, использующих базы данных, весьма обширны для того, чтобы описать их в одной статье. Поэтому сегодня мы рассмотрим лишь простейшие возможности работы с таблицами баз данных.

Набор данных в C++ Builder - это объект, состоящий из набора записей, каждая из которых, в свою очередь, состоит из полей, и указателя текущей записи. Набор данных может иметь полное соответствие с реально существующей таблицей или быть результатом запроса, он может быть частью таблицы или объединять между собой несколько таблиц.

Набор данных в C++ Builder является потомком абстрактного класса TDataSet (абстрактный класс - это класс, от которого можно порождать другие классы, но нельзя создать экземпляр объекта данного класса). Например, классы TQuery, TTable и TStoredProc, содержащиеся на странице палитры компонентов Data Access, - наследники TDBDataSet, который, в свою очередь, является наследником TDataSet. TDataSet содержит абстракции, необходимые для непосредственного управления таблицами или запросами, обеспечивая средства для того, чтобы открыть таблицу или выполнить запрос и перемещаться по строкам.

Компонент TDataSource

Компонент DataSource действует как посредник между компонентами TDataSet (TTable, TQuery, TStoredProc) и компонентами Data Controls - элементами управления, обеспечивающими представление данных на форме. Компоненты TDataSet управляют связями с библиотекой Borland Database Engine (BDE), а компонент DataSource управляет связями с данными в компонентах Data Controls.

В типичных приложениях БД компонент DataSource, как правило, связан с одним компоненом TDataSet (TTable или TQuery) и с одним или более компонентами Data Controls (такими, как DBGrid, DBEdit и др.). Связь этого компонента с компонентами TDataSet и DataControls осуществляется с использованием следующих свойств и событий:

- Свойство DataSet компонента DataSource идентифицирует имя компонента TDataSet. Можно присвоить значение свойству DataSet на этапе выполнения или с помощью инспектора объектов на этапе проектирования.
- Свойство Enabled компонента DataSource активизирует или останавливает взаимосвязь между компонентами TDataSource и Data Controls. Если значение свойства Enabled равно true, то компоненты Data Controls, связанные с TDataSource, воспринимают изменения набора данных. Использование свойства Enabled позволяет временно разъединять визуальные компоненты Data Controls и TDataSource, например, для того, чтобы в случае поиска в таблице с большим количеством записей не отображать на экране пролистывание всей таблицы.
- Свойство AutoEdit компонента DataSource контролирует, как инициируется редактирование в компонентах Data Controls. Если значение свойства AutoEdit равно true, то режим редактирования начинается непосредственно при получении фокуса компонентом Data Controls, связанным с данным компонентом TDataSet. В противном

случае режим редактирования начинается, когда вызывается метод Edit компонента TDataSet, например, после нажатия пользователем кнопки Edit на компоненте DBNavigator. · Событие OnDataChange компонента DataSource наступает, когда происходит изменение значения поля, записи, таблицы, запроса.

- Событие OnUpdateData компонента DataSource наступает, когда пользователь пытается изменить текущую запись в TDataSet. Обработчик этого события следует создавать, когда требуется соблюсти условия ссылочной целостности или ограничения, накладываемые на значения полей изменяемой базы данных.

Компонент TTable

Наиболее простым способом обращения к таблицам баз данных является использование компонента TTable, предоставляющего доступ к одной таблице. Для этой цели наиболее часто используются следующие свойства:

- Active - указывает, открыта (true) или нет (false) данная таблица.
- DatabaseName - имя каталога, содержащего искомую таблицу, либо псевдоним (alias) удаленной БД (псевдонимы устанавливаются с помощью утилиты конфигурации BDE, описание которой присутствует во многих источниках, посвященных продуктам Borland, либо с помощью SQL Explorer, вызываемого с помощью пункта меню Database/Explore). Это свойство может быть изменено только в случае, если таблица закрыта (ее свойство Active равно false), например:

```
Table1->Active = false;
Table1->DatabaseName = "BCDEMOS"
Table1->Active = true;
```

- TableName - имя таблицы.
- Exclusive - если это свойство принимает значение true, то никакой другой пользователь не может открыть таблицу, если она открыта данным приложением. Если это свойство равно false (значение по умолчанию), то другие пользователи могут открывать эту таблицу.
- IndexName - идентифицирует вторичный индекс для таблицы. Это свойство нельзя изменить, пока таблица открыта.
- MasterFields - определяет имя поля для создания связи с другой таблицей.
- MasterSource - имя компонента TDataSource, с помощью которого TTable будет получать данные из связанной таблицы.
- ReadOnly - если это свойство равно true, таблица открыта в режиме "только для чтения". Нельзя изменить свойство ReadOnly, пока таблица открыта.
- Eof, Bof - эти свойства принимают значение true, когда указатель текущей записи расположен на последней или соответственно первой записи таблицы.
- Fields - массив объектов TField. Используя это свойство, можно обращаться к полям по номеру, что удобно, когда заранее неизвестна структура таблицы:

```
Edit1->Text=Table1->Fields[2]->AsString;
```

Наиболее часто при работе с компонентом TTable используются следующие методы:

- Open и Close устанавливают значения свойства Active равными True и False соответственно.
- Refresh позволяет заново считать набор данных из БД.
- First, Last, Next, Prior перемещают указатель текущей записи на первую, последнюю, следующую и предыдущую записи соответственно, например:

```
Table1->First();
while (!Table1->Eof)
```

```
{
//что-то делаем...
Table1->Next();
};
```

- MoveBy перемещает указатель на указанное число строк (оно может быть и отрицательным) в пределах таблицы
- Insert, Edit, Delete, Append - переводят таблицу в режимы вставки записи, редактирования, удаления, добавления записи соответственно.
- Post - осуществляет физическое сохранение измененных данных. Например:

```
Table2->Insert();
Table2->Fields[0]->AsInteger = 100;
Table2->Fields[1]->AsString =Edit1->Text;
Table2->Post();
```

- Cancel - отменяет внесенные изменения, не сохраненные физически.
- FieldByName - предоставляет возможность обращения к данным в полях по имени поля:

```
S=Table1->FieldByName("area")->AsString;
```

- SetKey переключает таблицу в режим поиска.
- GotoKey начинает поиск строки, значение Fields[n] которой равно выбранному, где n - номер колонки таблицы, начиная с 0:

```
Table1->SetKey();
Table1->Fields[0]->AsString=Edit1->Text;
Table1->GotoKey();
```

- SetRangeStart, SetRangeEnd, ApplyRange позволяют выбрать нужные строки на основе диапазона значений какого-либо поля.

```
Table1->SetRangeStart();
Table1->Fields[0]->AsString = Edit1->Text;
Table1->SetRangeEnd();
Table1->Fields[0]->AsString = Edit2->Text;
Table1->ApplyRange();
```

- FreeBookmark, GetBookmark, GotoBookmark- позволяют создать помеченную строку в таблице и затем вернуться к ней позже. Методы Bookmark используют класс TBookmark. Метод GetBookmark устанавливает закладку на текущей строке таблицы. GotoBookmark осуществляет перемещение в таблице к строке, ранее отмеченной закладкой. Метод FreeBookmark используется для уничтожения объекта типа TBookmark:

```
TBookmark Marker =Table1->GetBookmark();
Table1->GotoBookmark(Marker);
Table1->FreeBookmark(Marker);
```

События компонента TTable позволяют строить и контролировать поведение приложе ий БД. Например, событие BeforePost наступает перед вставкой или изменением записи, событие AfterPost - после сохранения вставленной или измененной записи, событие AfterDelete - после удаления записи и т.д.

Чтобы внести компонент TTable в форму, нужно выполнить следующее:

1. Используя страницу Data Access палитры компонентов, разместить компонент TTable на форме или в модуле данных.

2. Свойству DatabaseName присвоить имя каталога, где находится БД, либо псевдо има БД.

3. Свойству TableName присвоить имя таблицы или выбрать таблицу из выпадающего списка.

4. Внести в форму компонент DataSource и установить значение свойства DataSet равным имени компонента TTable.

5. Внести компоненты Data Controls и связать их с компонентом DataSource для того, чтобы отобразить на экране данные из таблицы БД.

Компонент TField

Объекты класса TField являются свойством объекта TDataSet (напомним, что некоторые свойства объектов сами являются объектами с их собственными наборами свойств, и TField - один из них).

Свойство Fields объекта типа TDataSet позволяет обращаться к отдельным полям набора данных. Свойство Fields является массивом или набором объектов TField, динамически создающимся во время выполнения приложения. Элементы массива соответствуют колонкам таблицы.

Объект TField не делает никаких предположений относительно типов данных, с которыми он связан. Он имеет несколько свойств, позволяющих установить или вернуть обратно значения поля, например, AsString, AsBoolean, AsFloat, AsInteger. Наиболее часто используются свойства Text (строка текста, выводимого в связанный с данным полем интерфейсный элемент) и FieldName (имя поля базы данных).

Fields Editor позволяет создать так называемый статический список полей таблицы, добавляемых к описанию класса формы. Когда впервые используются такие компоненты TDataSet, как компонент TTable или TQuery, список полей для них динамически генерируется в процессе выполнения приложения на основе имеющихся столбцов таблиц или результатов SQL-запроса. Fields Editor позволяет определить и затем модифицировать статический список компонентов Field на этапе проектирования приложения. При внесении колонок с использованием Fields Editor для каждого из полей, добавленных к TDataSet, возникают объекты TField, после чего можно увидеть эти поля в инспекторе объектов и использовать в приложениях их свойства, события и методы.

Использовать Fields Editor нужно следующим образом:

1. Разместить компонент TTable или TQuery на форме.
2. Установить свойство DatabaseName для TTable или TQuery.
3. Установить свойство TableName компонента TTable или свойство SQL компонента TQuery.
4. Выбрать компонент TDataSet на форме и нажать правую клавишу мыши, после чего появится контекстное меню.
5. Из контекстного меню выбрать Fields Editor. Появится пустое окно с заголовком, совпадающим с именем компонента TTable.
6. Снова нажать правую клавишу мыши над пустым окном и из контекстного меню выб ать опцию Add Fields.Имена всех колонок таблицы или запроса появятся в диалоговой панели Add Fields.
7. Выбрать поля, которые нужно внести в список объектов, и нажать OK.

8. Если требуется создать вычисляемое поле на основе имеющихся полей, нажать правую клавишу мыши и из контекстного меню выбрать New Field для создания нового поля на основе существующего или для создания вычисляемого поля (в дальнейшем следует создать код обработчика события OnCalcFields компонента TTable, где и производятся необходимые вычисления).

9. Если необходимо удалить статическое поле из списка полей в наборе данных, нужно нажать правую клавишу мыши и из контекстного меню выбрать Delete.

После того, как в Fields Editor добавлены поля, они появятся в инспекторе объектов, а ссылки на них - в h-файле формы.

Если теперь применить операцию drag-and-drop к выделенным в Fields Editor полям, перенеся их на форму, то можно получить готовую форму с необходимым набором интерфейсных элементов (в нашем случае - DBEdit, позволяющий отображать и редактировать строковые, числовые, денежные и другие поля, чьи значения представимы в виде строки символов, и DBImage, позволяющий отображать графические поля и использовать Clipboard для их редактирования). Если к такой форме добавить компонент TDBNavigator (этот компонент реализует основные методы TTable и TQuery, связанные с редактированием данных) и связать его с имеющимся компонентом TDataSource, а затем скомпилировать проект, получим приложение для просмотра и редактирования данных в таблице.

При работе Fields Editor создаются объекты, соответствующие видимым в инспекторе объектов полям. Эти объекты являются потомками объектного типа TField. Таблица 1 описывает существующие классы таких объектов:

Таблица 1. Потомки TField

Потомок	Описание
TStringField	Текстовые данные фиксированной длины до 8192 символов.
TAutoIncField	Целые числа от -2,147,483,648 до 2,147,483,647. Предназначен для нумерации ст ок в наборе данных. Потомок TIntegerField.
TIntegerField	Целые числа от -2,147,483,648 до 2,147,483,647.
TSmallIntField	Целые числа от -32768 до 32767.
TWordField	Целые числа от 0 до 65535.
TFloatField	Действительные числа с абсолютной величиной от 1.2x10e-324 до 1.7x10e308 с точностью до 15-16 цифры.
TCurrencyField	Действительные числа с абсолютной величиной от 1.2x10e-324 до 1.7x10e308 с точностью до 15-16 цифры.
TBoolea	Значения true или false.

nField	
TDateTimeField	Значения даты и времени.
TDateField	Значения даты.
TTimeField	Значения времени.
TBlobField	Произвольное поле данных без ограничений размера.
TBytesField	Произвольное поле данных без ограничений размера.
TVarBytesField	Произвольное поле данных до 65535 символов с фактической длиной, представленной в первых двух байтах.
TMemoField	Текст произвольной длины.
TGraphicField	Графическое поле произвольной длины, например, битовый массив.

Компонент TDBGrid

Компонент TDBGrid обеспечивает табличный способ отображения на экране строк данных из компонентов TTable или TQuery. Приложение может использовать TDBGrid для отображения, вставки, уничтожения, редактирования данных БД. Обычно DBGrid используется в сочетании с DBNavigator, хотя можно использовать и другие интерфейсные элементы, включив в их обработчики событий методы First, Last, Next, Ptior, Insert, Delete, Edit, Append, Post, Cancel компонента TTable.

Внешний вид таблицы (например, надписи в заголовках столбцов) может быть изменен с помощью редактора свойств Columns Editor. Для вызова Columns Editor нужно либо выбрать соответствующую опцию в контекстном меню компонента DBGrid или щелкнуть мышью в колонке значений напротив свойства Columns в инспекторе объектов.

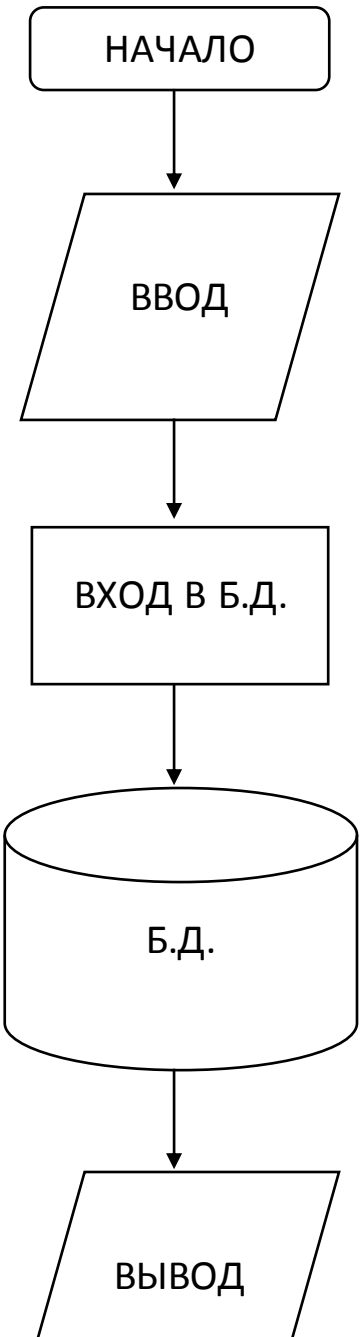
Вторым способом получения контроля над характеристиками DBGrid или другими компонентами является создание описанным выше способом статического набора компонентов TField. Имея компонент типа TField, созданный для каждого из полей в наборе данных, можно установить ширину, формат, маску, расположение, метку для отображения в DBGrid и другие характеристики.

Поля Float, Integer и Date обладают свойством DisplayMask. Это свойство можно использовать, чтобы форматировать данные в компоненте DBGrid или другом компоненте Data Controls. Например, экранный формат mm-dd-yy может использоваться для размещения полей типа дата.

Некоторые компоненты TField (например, TStringField) обладают свойством EditMask, которое можно установить, вводя данные в DBGrid и другие компоненты Data Controls. Для

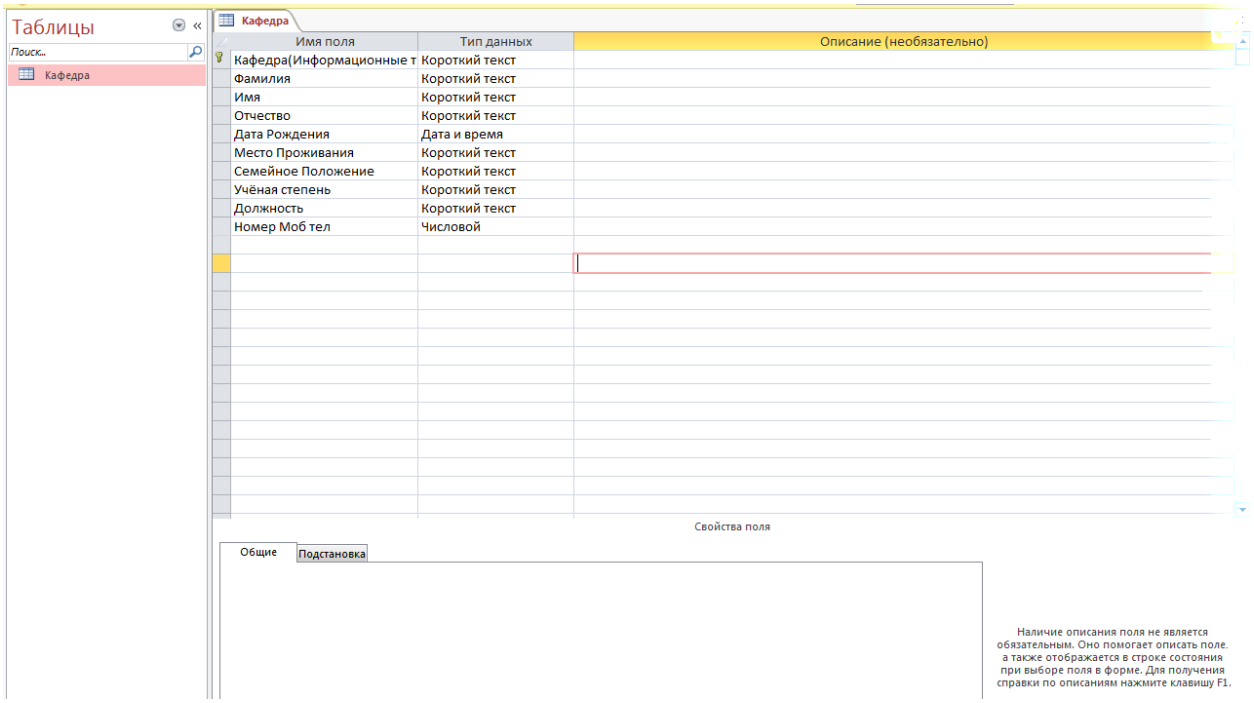
установки свойства EditMask нужно установить компонент Field в Object Inspector и выбрать свойство EditMask, после чего появится диалоговая панель Input Mask Editor,. Чтобы проверить маску редактирования, нужно ввести значение в поле Test Input.

БЛОК СХЕМА Б.Д.

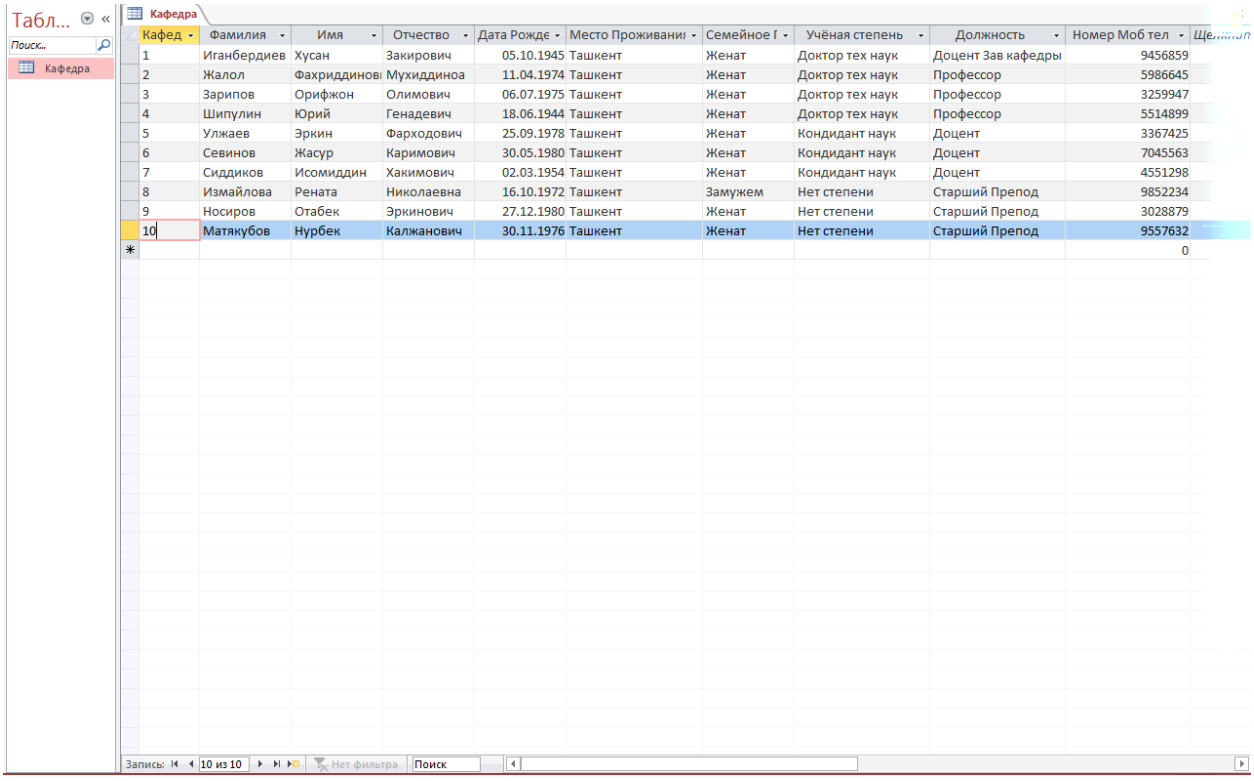


Последовательность создания базы данных

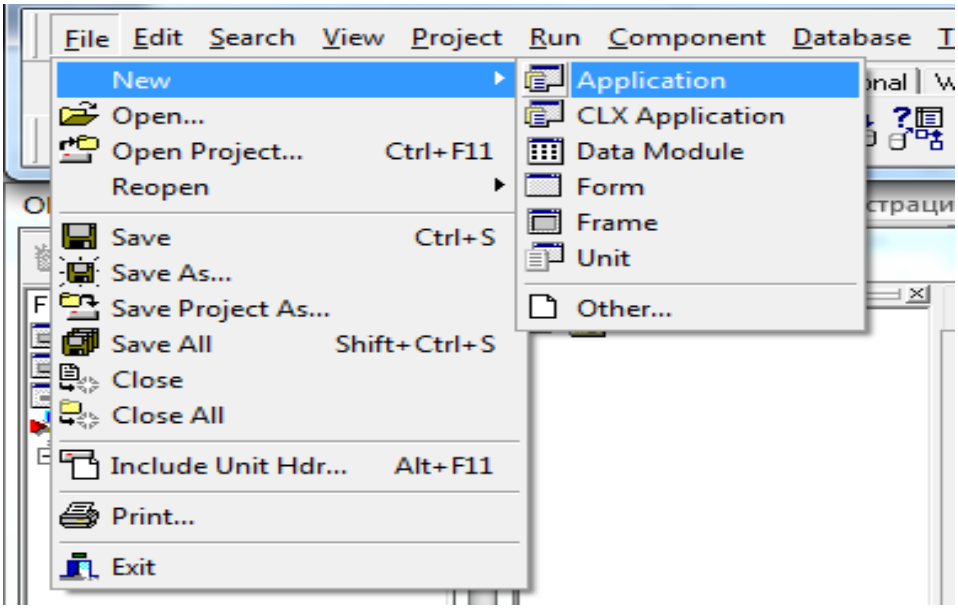
Начнем нашу работу с создании Базы Данных с помощи Программы Microsoft Access 2013



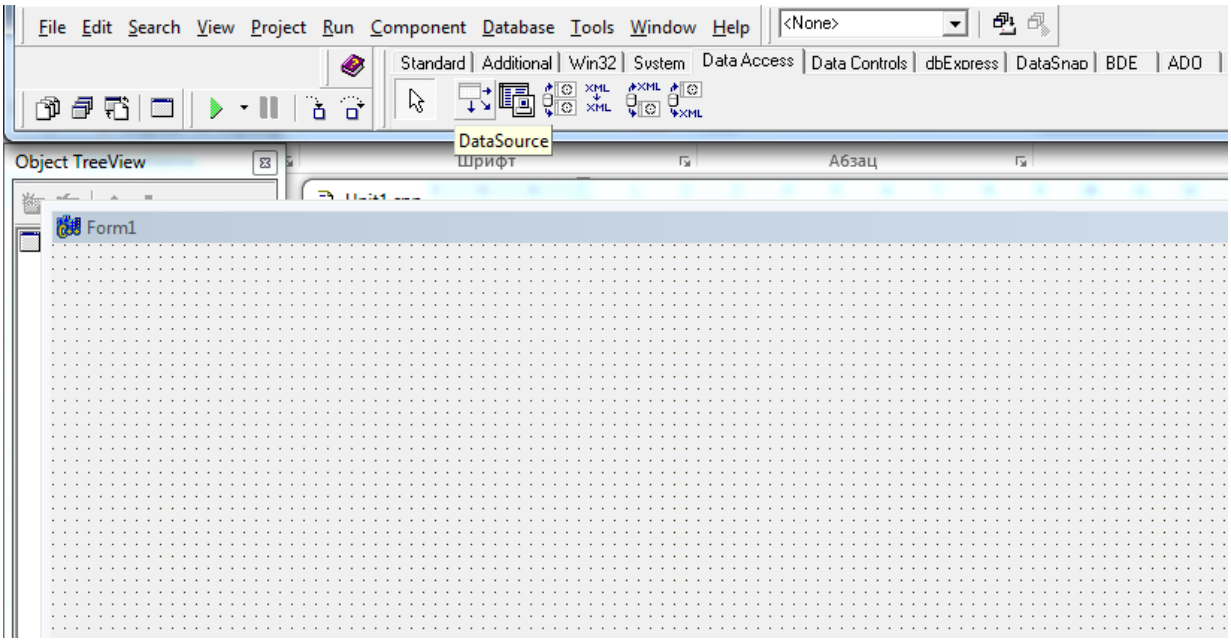
После создания (Конструктор таблиц) заполняем таблицу «Кафедра».



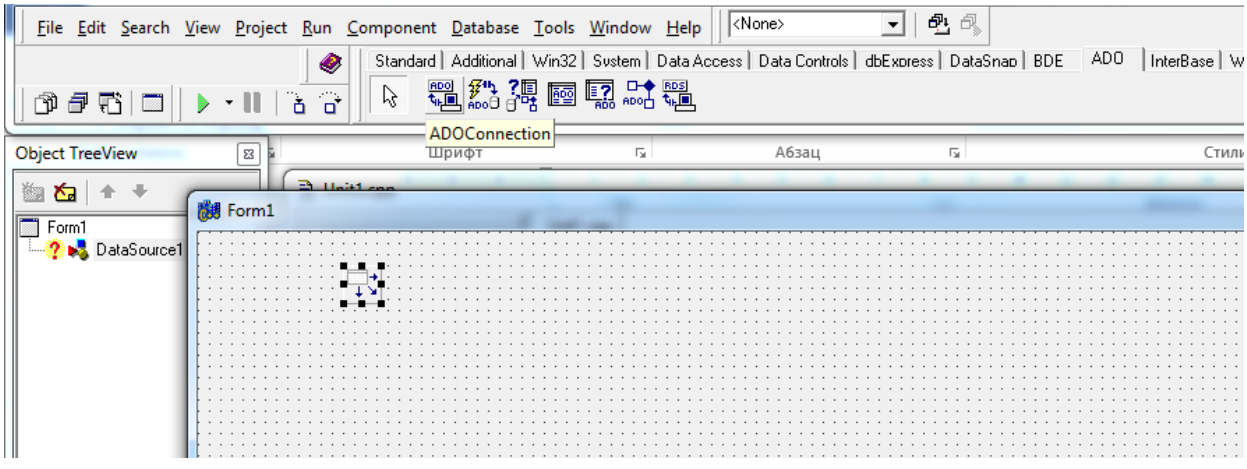
После Сохраняем нашу базу и переходим к созданию базы через Программу C++ Builder 6
Создаем новую прогамму для Базы:



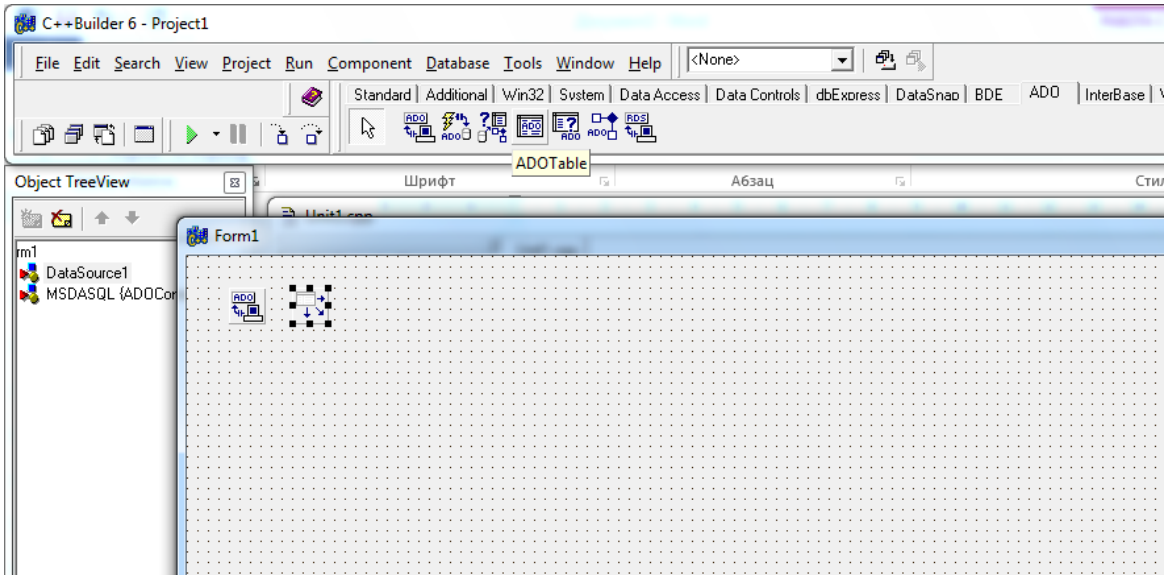
Выбираем DataSource из Data Access:



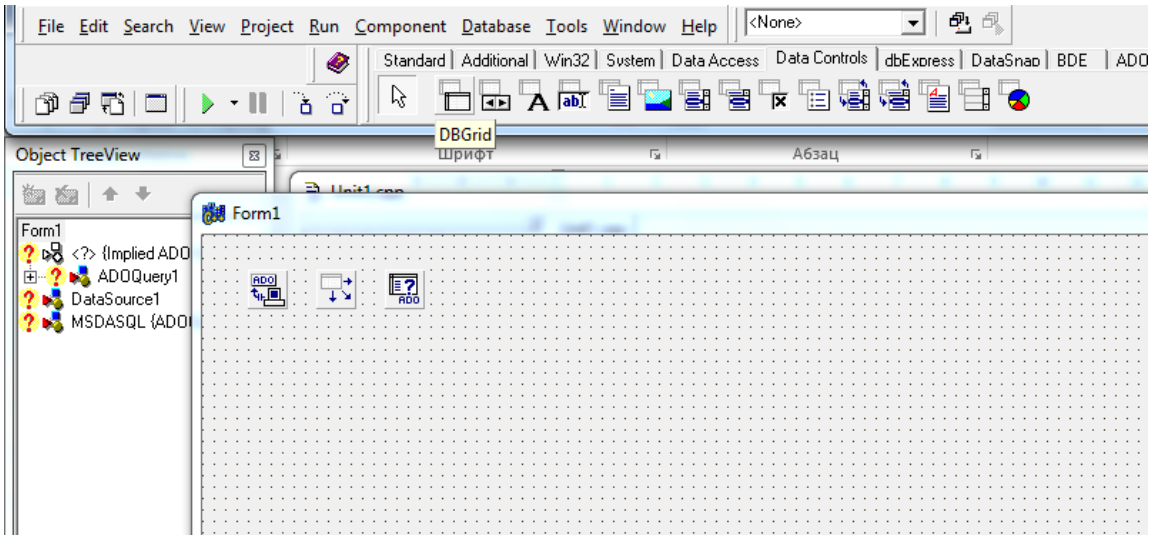
Далее ADOConnection из ADO:



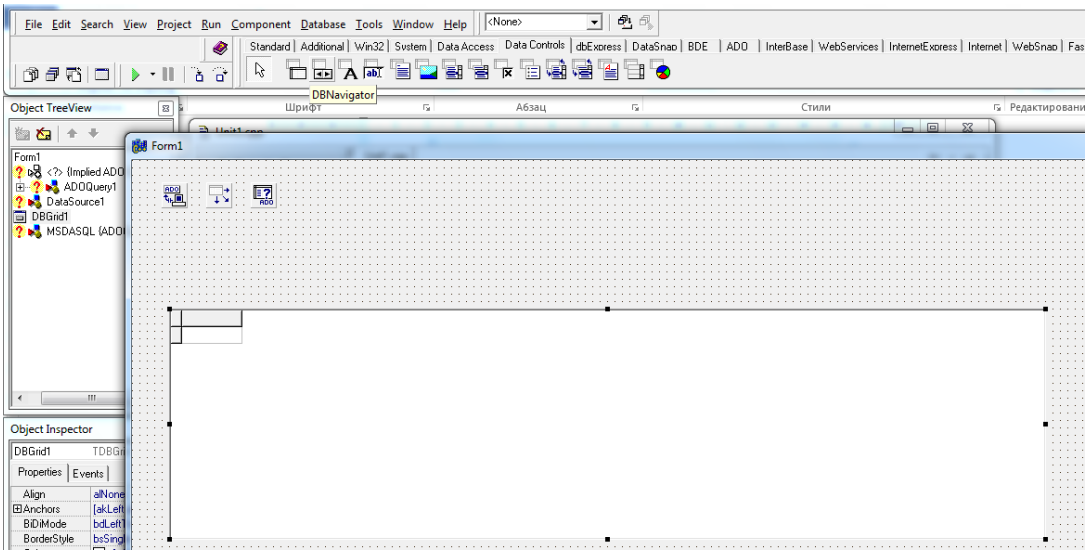
Из той же ADO выбираем ADOTable:



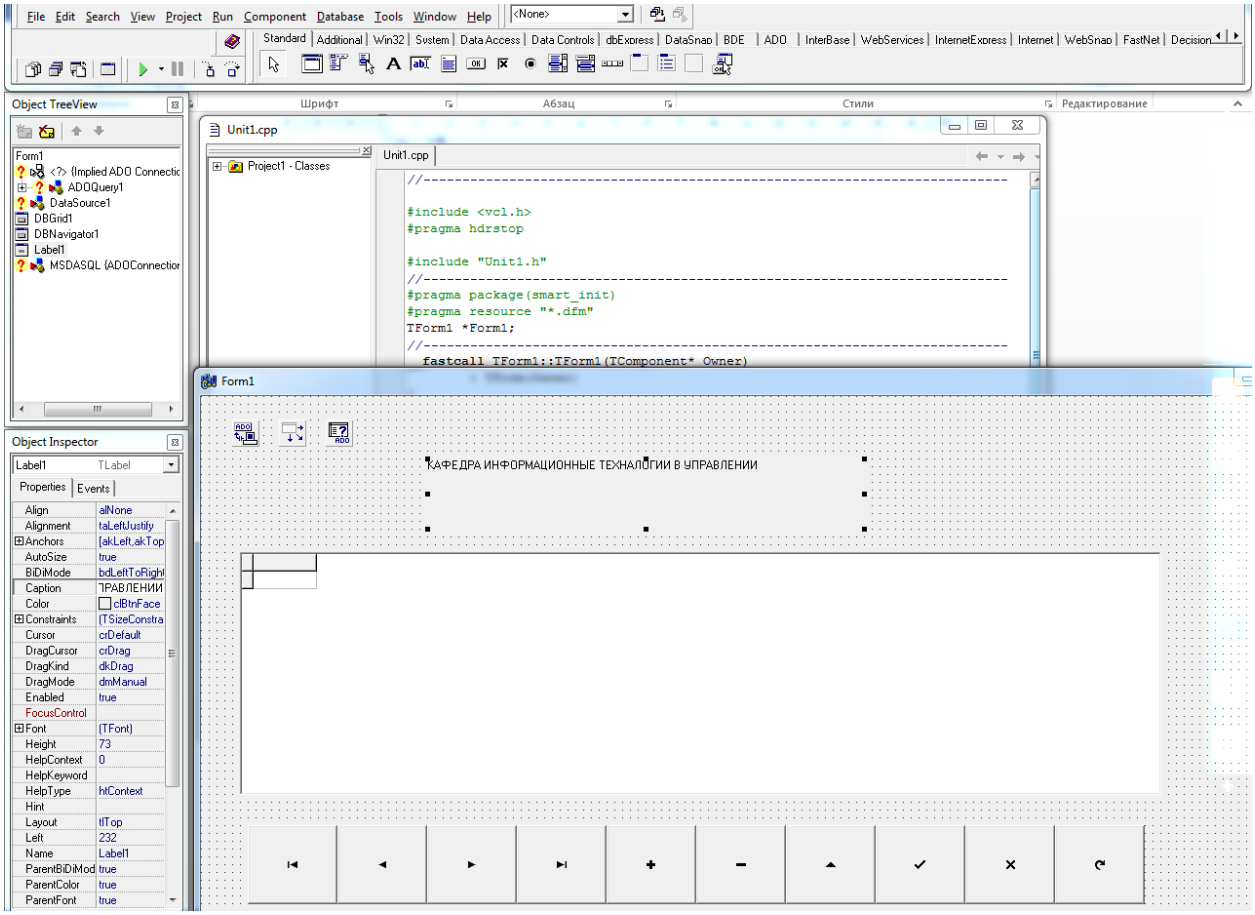
Далее из Data Controls выбираем DBGrid:



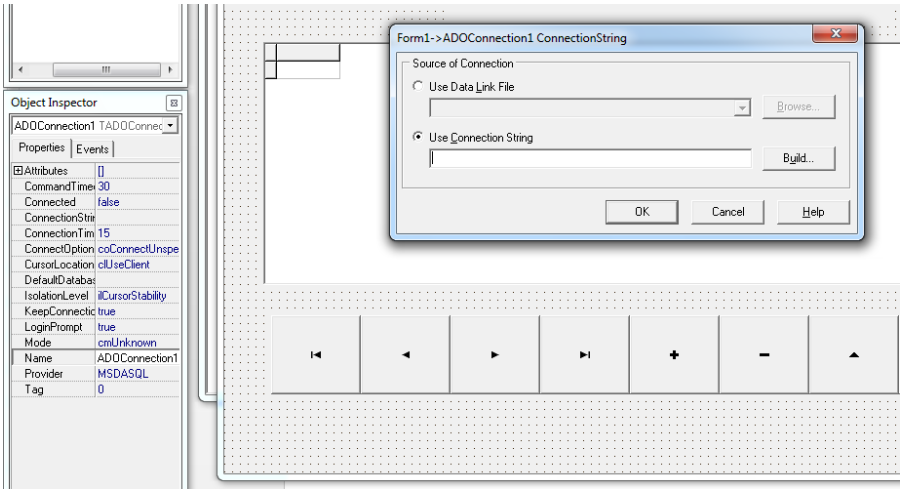
Далее из Data Controls выбираем DBNovigator:

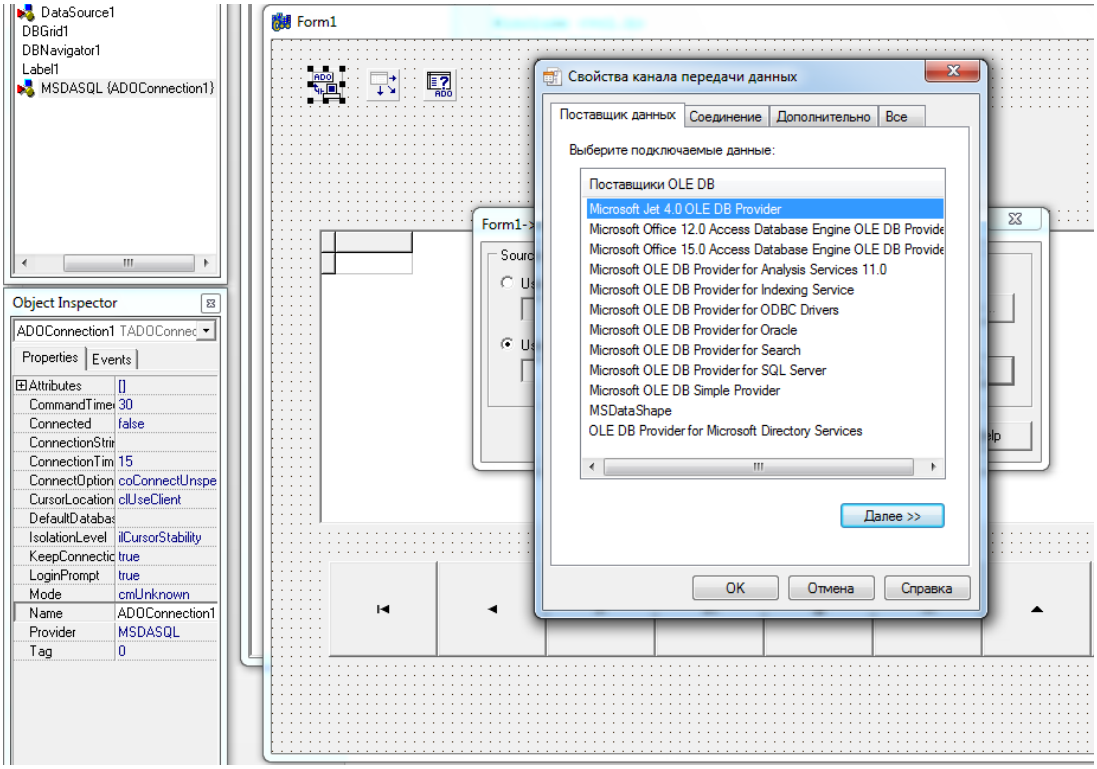


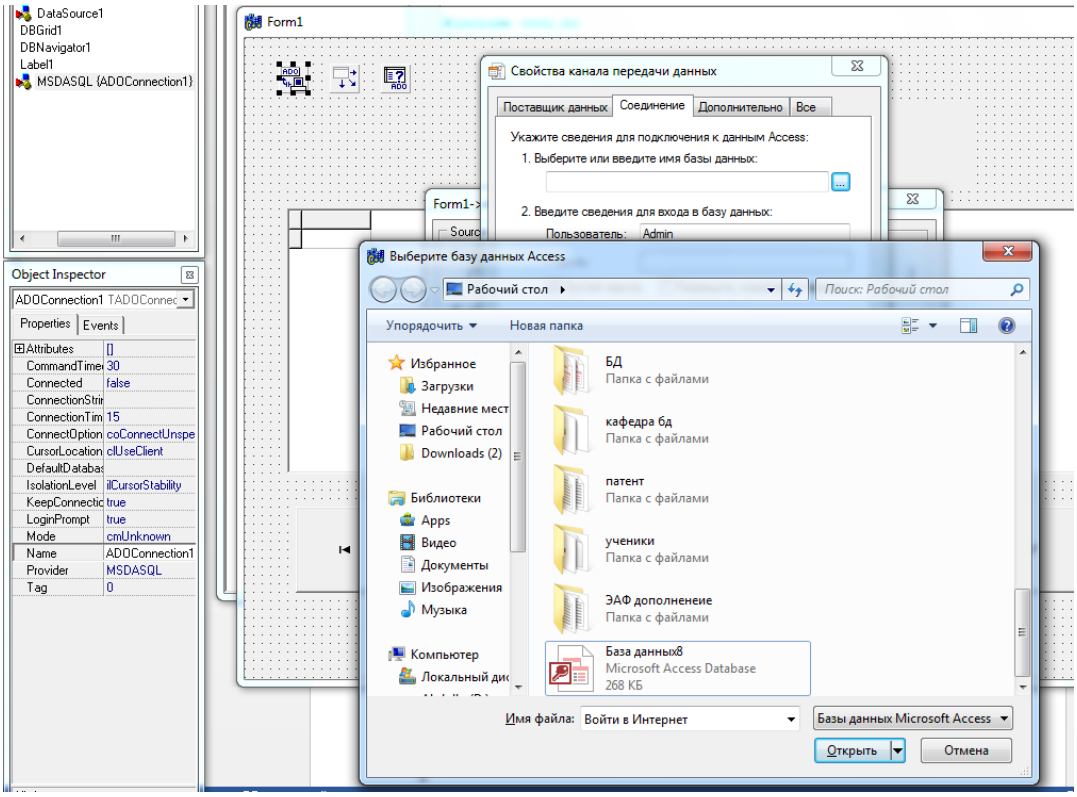
После создаем Label и изменяем из Label1 на «Кафедра Информационные техналогии в управлении»



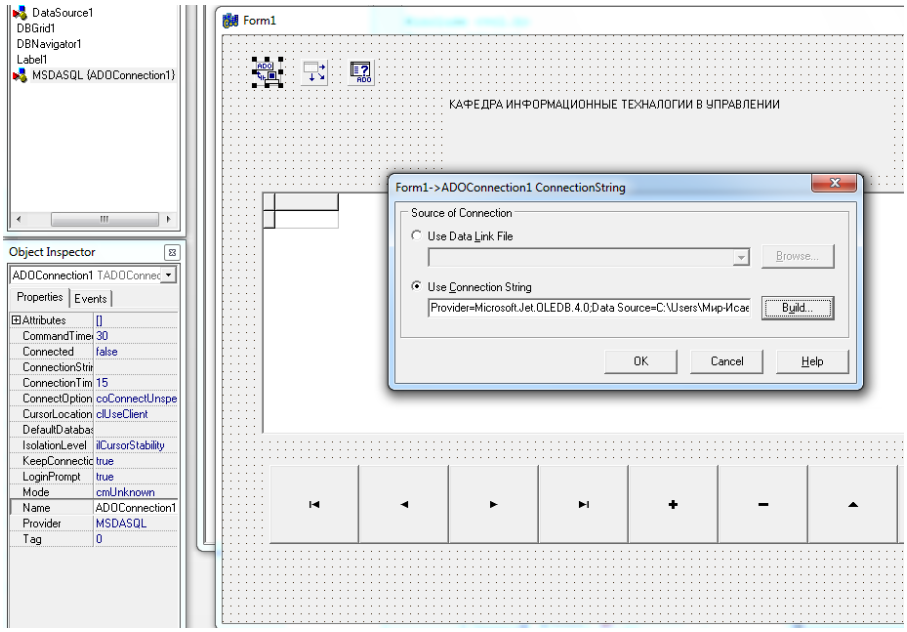
Далее нажимаем два раза на ADOConnection и указываем путь:



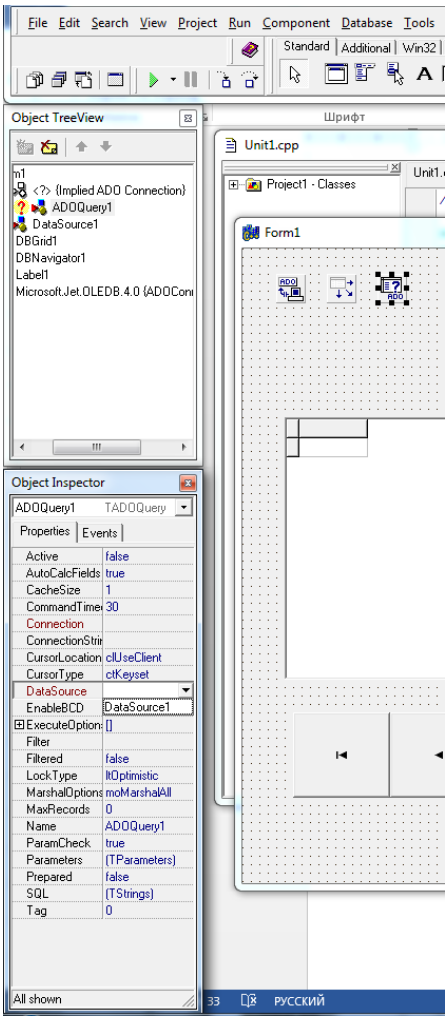

DataSource1
DBGrid1
DBNavigator1
Label1
MSDASQL (ADOConnection1)
Object Inspector
ADOConnection1 TADOConnec
Properties
Events
Attributes
CommandTimer 30
Connected false
ConnectionStri
ConnectionTim 15
ConnectOption coConnectUnspe
CursorLocation clUseClient
DefaultDatabas
IsolationLevel ilCursorStability
KeepConnectic true
LoginPrompt true
Mode cmUnknown
Name ADOConnection1
Provider MSDASQL
Tag 0
Form1
Свойства канала передачи данных
Поставщик данных
Соединение
Дополнительно
Все
Выберите подключаемые данные:
Поставщики OLE DB
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Provide
Microsoft Office 15.0 Access Database Engine OLE DB Provide
Microsoft OLE DB Provider for Analysis Services 11.0
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
Далее >>
OK
Отмена
Справка

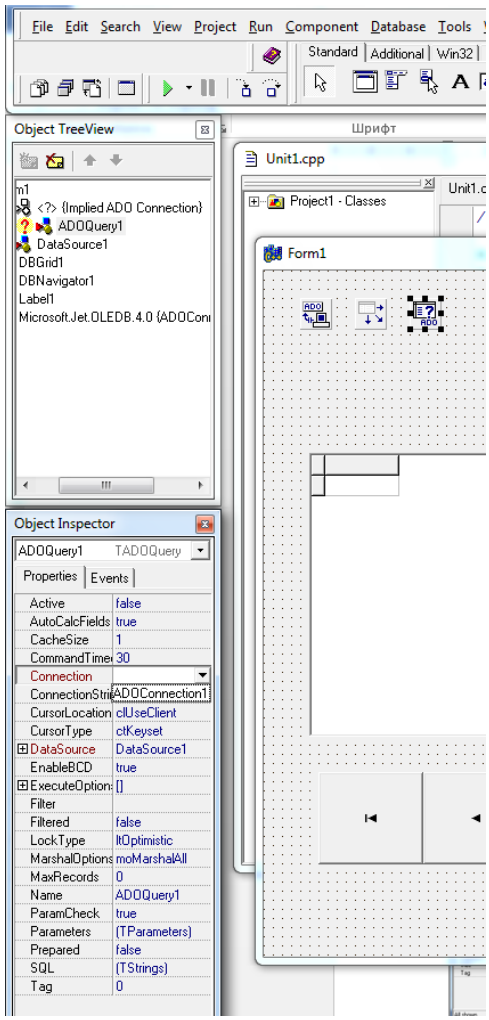

DataSource1
DBGrid1
DBNavigator1
Label1
MSDASQL (ADOConnection1)
Object Inspector
Form1
Свойства канала передачи данных
Поставщик данных
Соединение
Дополнительно
Все
Укажите сведения для подключения к данным Access:
1. Выберите или введите имя базы данных:
2. Введите сведения для входа в базу данных:
Пользователь:
Admin
Выберите базу данных Access
Рабочий стол
Поиск: Рабочий стол
Упорядочить
Новая папка
Избранное
Загрузки
Недавние мест
Рабочий стол
Downloads (2)
Библиотеки
Apps
Видео
Документы
Изображения
Музыка
Компьютер
БД
Папка с файлами
кафедра бд
Папка с файлами
патент
Папка с файлами
ученики
Папка с файлами
ЭАФ дополнение
Папка с файлами
База данных8
Microsoft Access Database
268 КБ
Имя файла: Войти в Интернет
Базы данных Microsoft Access
Открыть
Отмена

Нажымаем ОК и изменяем значение LogoPrompt на true:

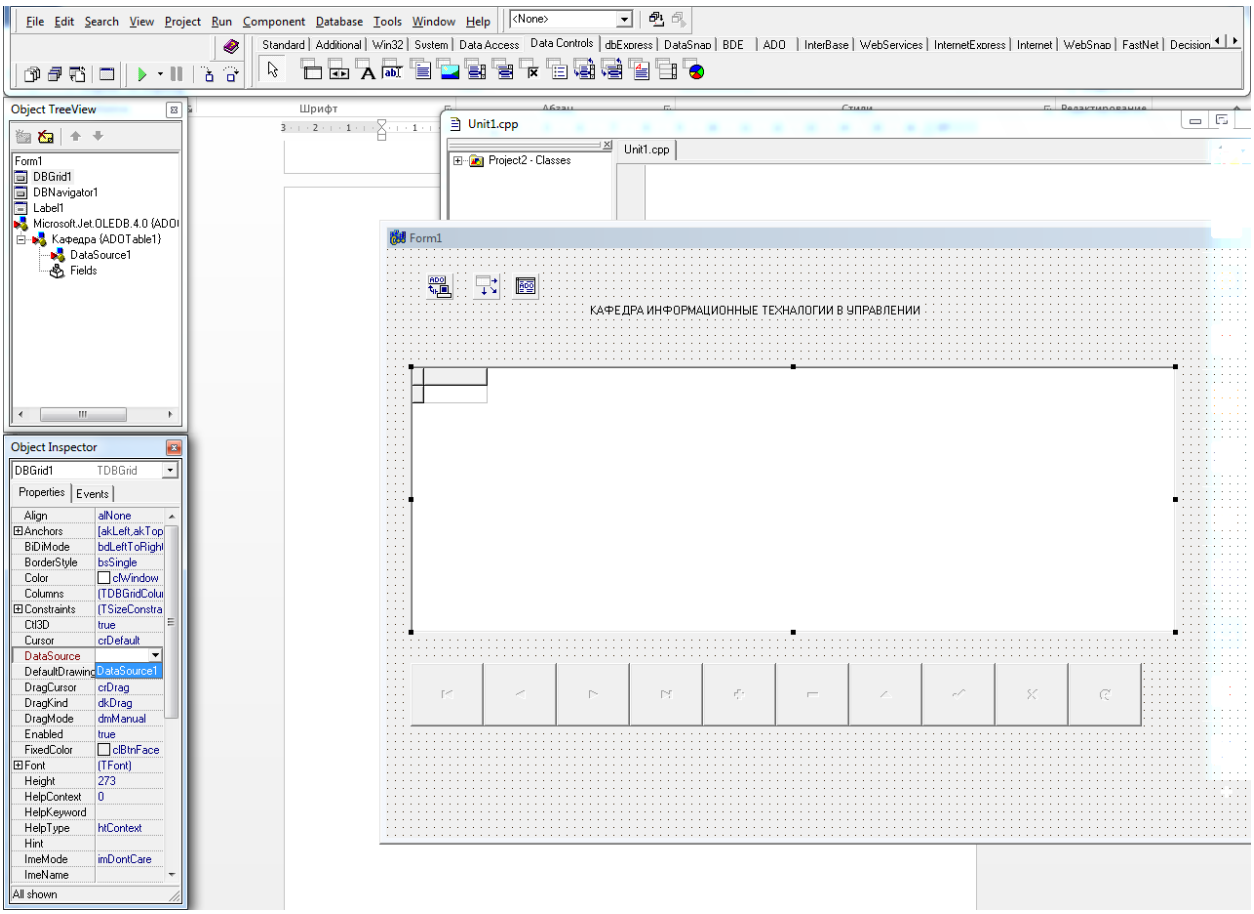


Выбираем ADOTable, добавляем в DataSource - DataSource1 и в Connection - ADOConnection:

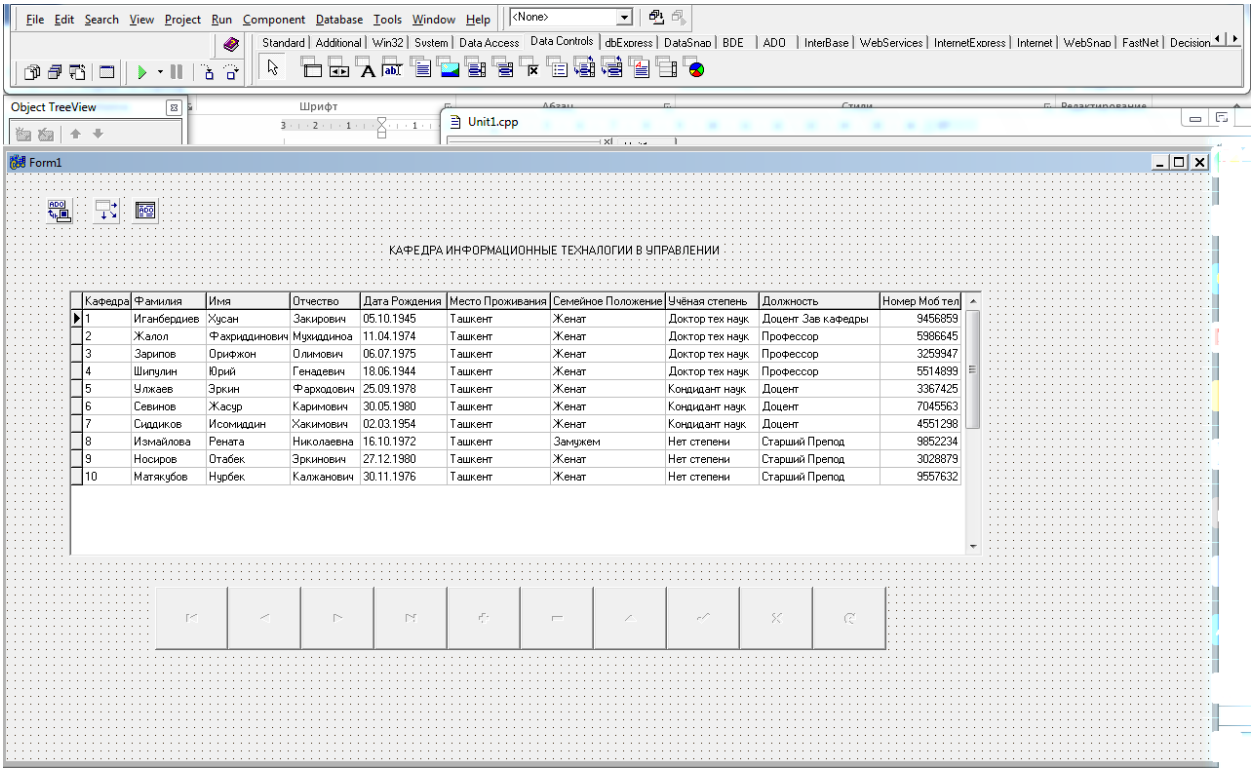




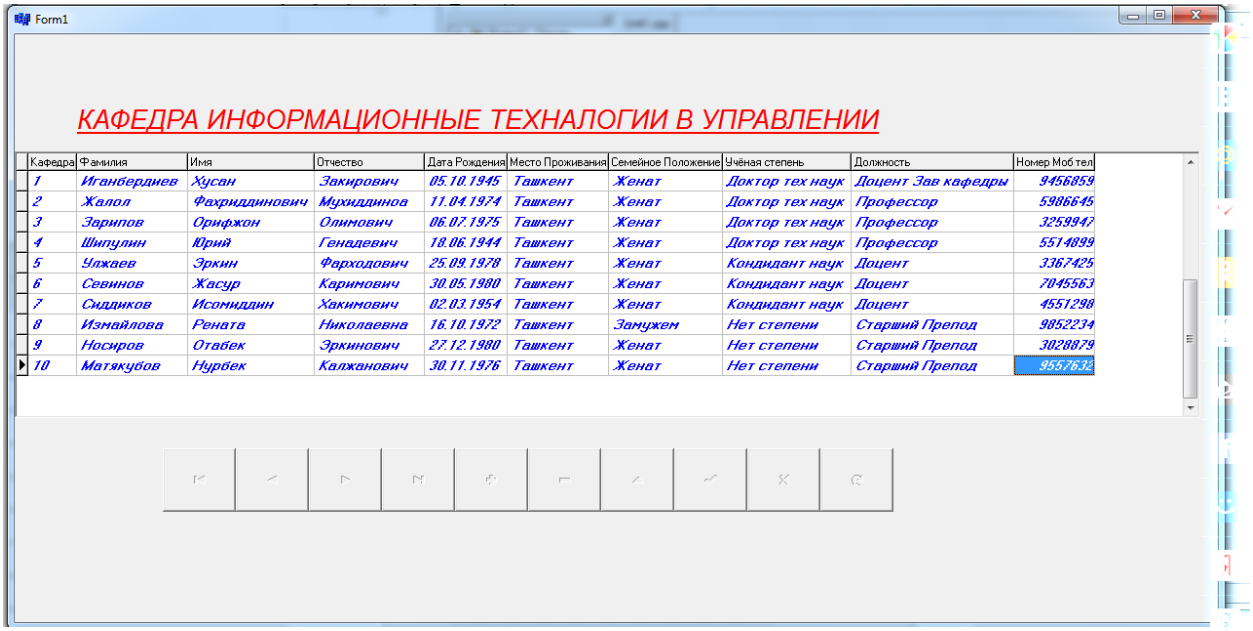
После выбераем DBGrid и добавляем в окно DataSource – DataSource1:



Далее Появляется наша База Данных:

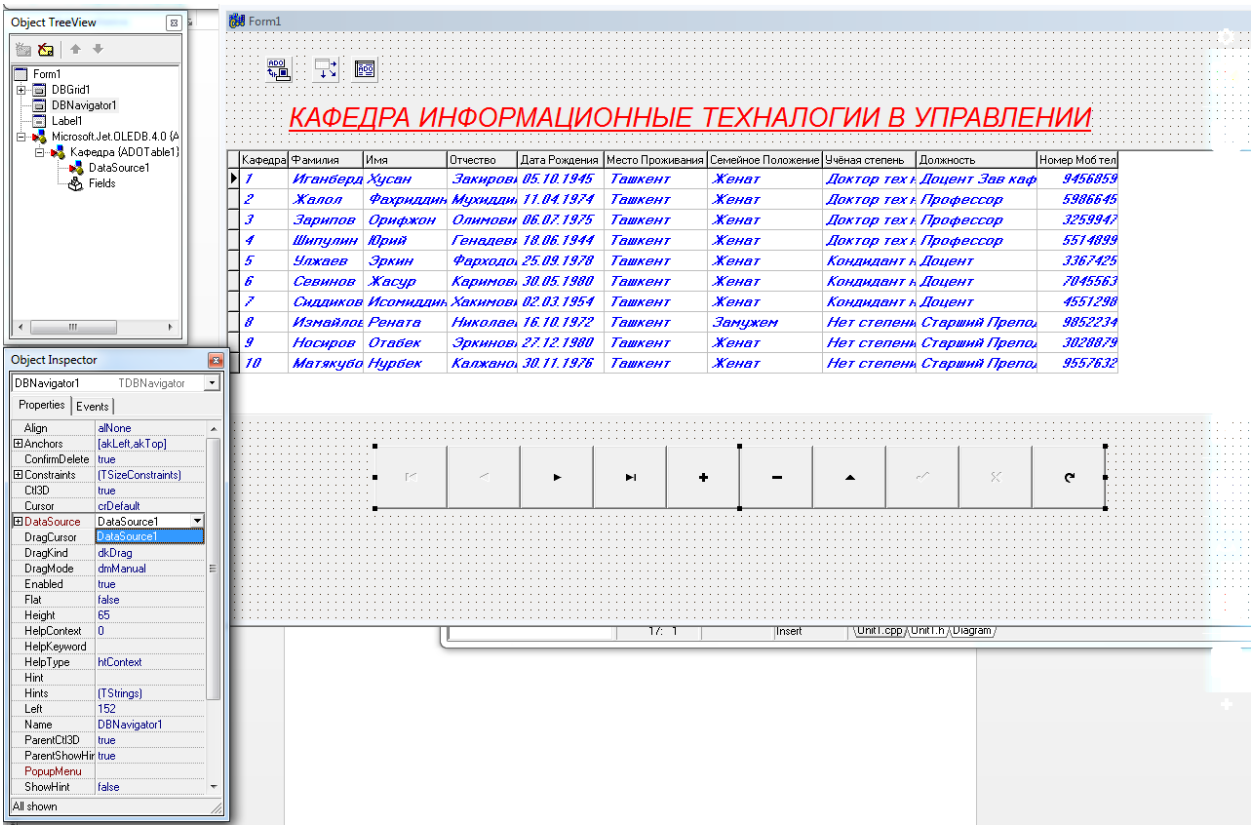


Кафедра	Фамилия	Имя	Отчество	Дата Рождения	Место Проживания	Семейное Положение	Учёная степень	Должность	Номер Моб тел
1	Иганбердиев	Хусан	Закирович	05.10.1945	Ташкент	Женат	Доктор тех наук	Доцент Зав кафедры	9456859
2	Жалол	Фахриддинович	Мухиддинов	11.04.1974	Ташкент	Женат	Доктор тех наук	Профессор	5986645
3	Зарипов	Орифжон	Олимович	06.07.1975	Ташкент	Женат	Доктор тех наук	Профессор	3259947
4	Шипулин	Юрий	Генадевич	18.06.1944	Ташкент	Женат	Доктор тех наук	Профессор	5514899
5	Улжаев	Эркин	Фарходович	25.09.1978	Ташкент	Женат	Кондидант наук	Доцент	3367425
6	Севинов	Жасур	Каримович	30.05.1980	Ташкент	Женат	Кондидант наук	Доцент	7045563
7	Сиддиков	Исомиддин	Хакимович	02.03.1954	Ташкент	Женат	Кондидант наук	Доцент	4551298
8	Измайлова	Рената	Николаевна	16.10.1972	Ташкент	Замужем	Нет степени	Старший Препод	9852234
9	Носиров	Отабек	Эркинович	27.12.1980	Ташкент	Женат	Нет степени	Старший Препод	3028879
10	Матякубов	Нурбек	Калжанович	30.11.1976	Ташкент	Женат	Нет степени	Старший Препод	9557632



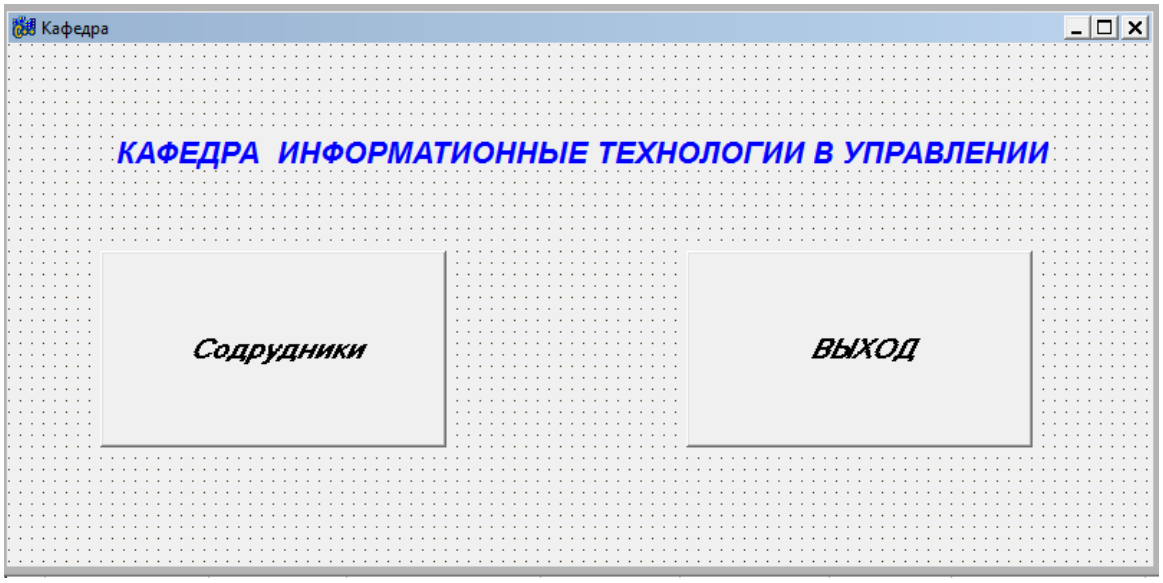
Кафедра	Фамилия	Имя	Отчество	Дата Рождения	Место Проживания	Семейное Положение	Учёная степень	Должность	Номер Моб тел
1	Иганбердиев	Хусан	Закирович	05.10.1945	Ташкент	Женат	Доктор тех наук	Доцент Зав кафедры	9456859
2	Жалол	Фахриддинович	Мухиддинов	11.04.1974	Ташкент	Женат	Доктор тех наук	Профессор	5986645
3	Зарипов	Орифжон	Олимович	06.07.1975	Ташкент	Женат	Доктор тех наук	Профессор	3259947
4	Шипулин	Юрий	Генадевич	18.06.1944	Ташкент	Женат	Доктор тех наук	Профессор	5514899
5	Улжаев	Эркин	Фарходович	25.09.1978	Ташкент	Женат	Кондидант наук	Доцент	3367425
6	Севинов	Жасур	Каримович	30.05.1980	Ташкент	Женат	Кондидант наук	Доцент	7045563
7	Сиддиков	Исомиддин	Хакимович	02.03.1954	Ташкент	Женат	Кондидант наук	Доцент	4551298
8	Измайлова	Рената	Николаевна	16.10.1972	Ташкент	Замужем	Нет степени	Старший Препод	9852234
9	Носиров	Отабек	Эркинович	27.12.1980	Ташкент	Женат	Нет степени	Старший Препод	3028879
10	Матякубов	Нурбек	Калжанович	30.11.1976	Ташкент	Женат	Нет степени	Старший Препод	9557632

Включаем наш Navigator также как и DBGrid:

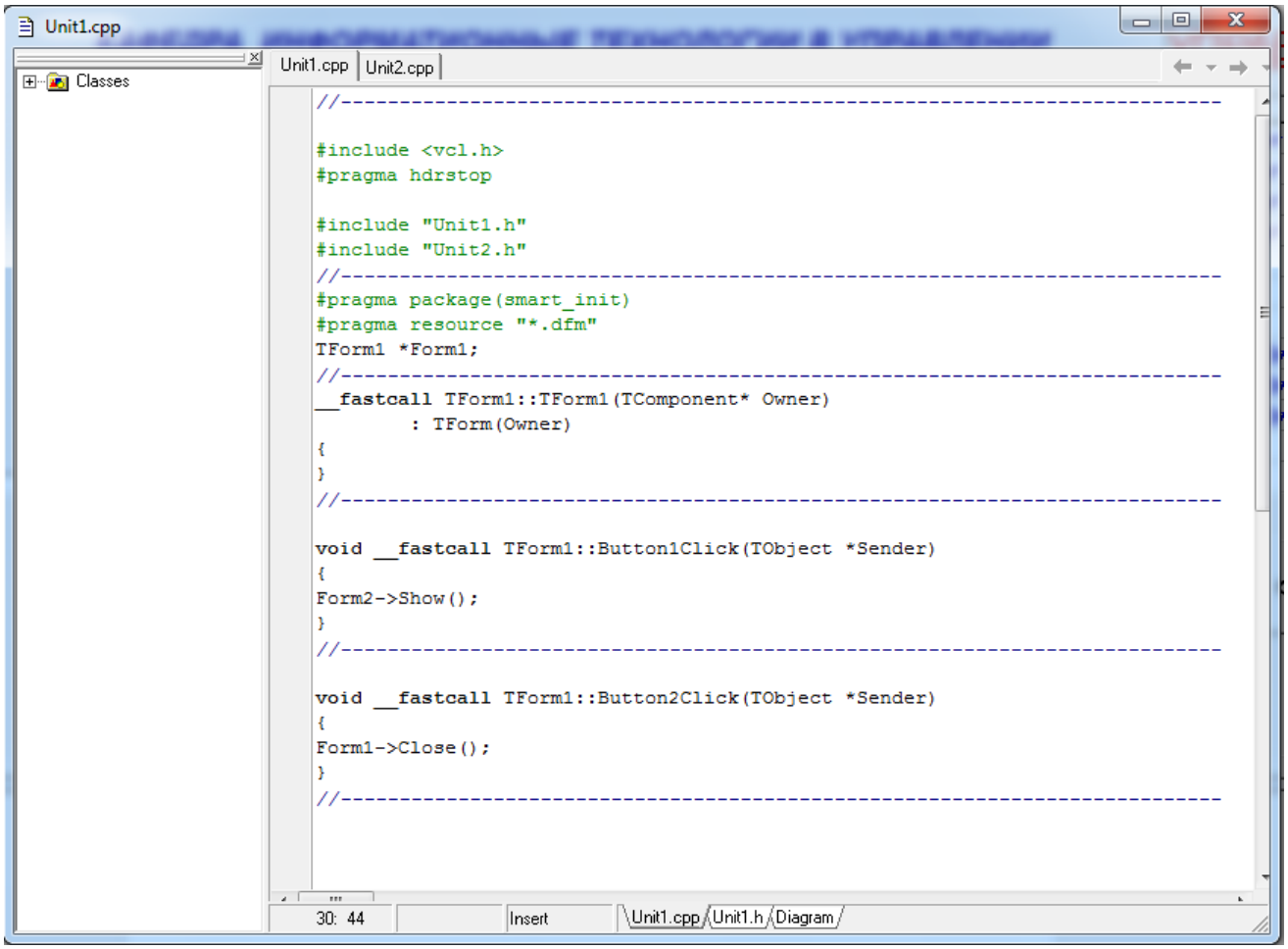


С помощю «Novigator» ра мы можем добавлять таблицу, удалять и мого другое:

Создаем ещё одну форму. назовём ее «Кафедра»:

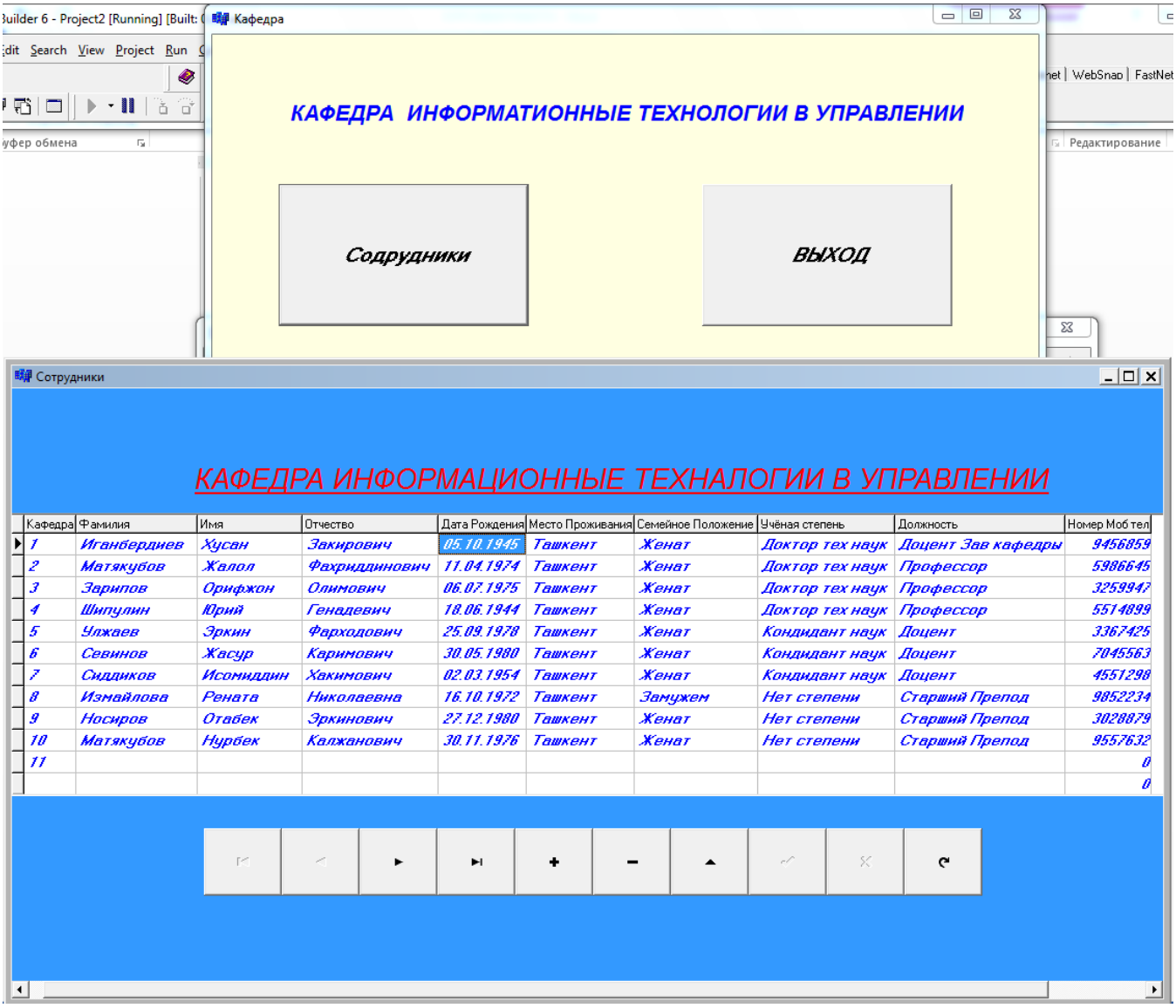


Вводим код программы:



Результат:

Мы создали базу данных через программу C++ Builder 6.



Кафедра	Фамилия	Имя	Отчество	Дата Рождения	Место Проживания	Семейное Положение	Учёная степень	Должность	Номер Моб тел
1	Игамбердиев	Хусан	Закирович	05.10.1945	Ташкент	Женат	Доктор тех наук	Доцент Зав кафедры	9456859
2	Матякубов	Жалол	Фахриддинович	11.04.1974	Ташкент	Женат	Доктор тех наук	Профессор	5986645
3	Зарипов	Орифжон	Олимович	06.07.1975	Ташкент	Женат	Доктор тех наук	Профессор	3259947
4	Шипулин	Юрий	Генадевич	18.06.1944	Ташкент	Женат	Доктор тех наук	Профессор	5514899
5	Уляаев	Эркин	Фарходович	25.09.1978	Ташкент	Женат	Кондидант наук	Доцент	3367425
6	Севинов	Жасур	Каримович	30.05.1980	Ташкент	Женат	Кондидант наук	Доцент	7045563
7	Сиддиков	Исомиддин	Хакимович	02.03.1954	Ташкент	Женат	Кондидант наук	Доцент	4551298
8	Измайлова	Рената	Николаевна	16.10.1972	Ташкент	Замужем	Нет степени	Старший Препод	9852234
9	Носиров	Отабек	Эркинович	27.12.1980	Ташкент	Женат	Нет степени	Старший Препод	3028879
10	Матякубов	Нурбек	Калжанович	30.11.1976	Ташкент	Женат	Нет степени	Старший Препод	9557632
11									0
									0

Заключение:

Написанная курсовая работы позволило мне близко познакомиться с программированием, формальными правилами записи алгоритмов и их последующего выполнения задачи компьютером. При помощи таких программ облегчается робота и ускоряется работа способность. Такие базы данных можно создать в MS Access, но в среде в C++ Builder 6 все создают свои базы в любом стиле.

В ходе работы я создал базу данных кафедры состоящих из 10 сотрудников. В ходе создании я ощутил легкость в создании базы данных в программе C++ Builder 6 . Я при помощи своей задачи научился создавать базы данных. Моя база помогает обновлять данные, сортировать и создавать базы. С помощью этой программы любой пользователь может создать свою базу, не соглашаясь, например с дизайном или функционалом, многих других программ создающих базы данные.

Литература:

1.WWW-серверы: Borland, Miller Friman, Turbo Power, ProtoView, Popkin Software, InterSolv, AOL и др.

2. http://technologies.su/

3.«Объектно – ориентированное программирование », Павловская Т. А., Щупак Ю. А., «Питер», 2005 г.

4. https://www.youtube.com/watch?v=0DFuOZ_fiNk . Создания баз данных в C++ Builder 6.

5. http://www.cyberforum.ru/