**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное ГОСУДАРСТВЕННОЕ бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики и вычислительной техники

Кафедра ИиСП

**Курсовая работа**

**по дисциплине «Базы данных»**

Выполнил:

студент группы: ПС-31

Чибирев А.Д.

Проверил:

Лучинин З.С.

г. Йошкар-Ола

2016

**Постановка задачи**

Создать приложение, включающее:

1. главное меню;
2. формы для ввода данных в таблицы;
3. формы для согласованного просмотра каждой из родительских таблиц совместно с дочерней таблицей (для текущей записи родительской таблицы должны показываться только связанные записи дочерней таблицы).

# Использовать:

# Поля с рисунками (фотографиями).

# Вычисляемые поля. Например, произведение цены на количество.

# Форматирование полей ввода. Например, представление телефона (42-35-18).

# Проверка правильности ввода некоторых значений. Например, нахождение числовых значений в некотором диапазоне.

# Программное редактирование и ввод записей с проверкой полноты ввода информации и связи значений разных полей.

# Сортировка записей по различным полям.

# Поиск и фильтрация записей по частичному заданию значений нескольких полей.

# Вычисление показателей по всем записям таблицы и по отфильтрованной части таблицы. Примеры показателей: сумма, среднее значение, максимум некоторого числового поля и т. п.

# Бизнес-правила, то есть программирование реакции приложения на вставку, редактирование и удаление записей.

1. Отчеты в виде экранных форм с возможностью распечатки:
   * по записям родительских таблиц;
   * по связанным записям каждой пары родительской и дочерней таблиц;
   * сгруппированных по значениям заданных полей записей.

Среда разработки:

Embarcadero RAD Studio 10.1 Berlin, язык Delphi

База Данных:

в формате Access

**Структура базы данных**

Таблицы:

**Пациенты**: Patients(idPatient, name, nPolicy, nPassport, nTelephone, address, nClinic),

где IDPatient – номер пациента, name – ФИО, nPolicy – полис, nPassport – паспорт, nTelephone – телефон, address – адрес, nClinic – клиника.

**Прием у врача**: appointments(idCommunication, idPatient, idDoctor),

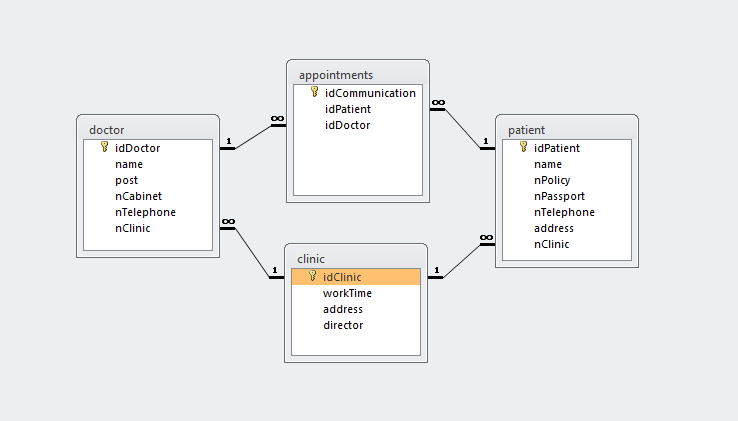
где idCommunication – номер связи, idPatient – номер пациента, idDoctor – номер доктора.

**Врач:** Doctor(idDoctor, name, post, nCabinet, nTelephone, nClinic),

где idDoctor - номер доктора, name – имя доктора, post –должност, nCabinet –номер кабинета, nTelephone – номер телефона, nClinic – номер клиники.

**Поликлиника:** Clinic (idClinic, workTime, address, director),

где idClinic - номер клиники, worktime – время работы, address – адрес, director – директор.

**Схема данных**

**Форма просмотра данных**

Для просмотра данных используется компонент ADOQuery, позволяющий на основе SQL-запроса SELECT получать данные из таблицы. Для изменения хранящейся в БД информации формируются запросы с операторами INSERT, UPDATE и DELETE.

Просмотр информации в разных разрезах обеспечивают как параметрические запросы SELECT, так и динамические запросы, когда программа модифицирует текст запроса перед его выполнением.

Создадим пары компонентов ADOQuery-DataSource, назовем компоненты запросов ADOQPatient, ADOQMedication, ADOQTherapy1, ADOQTherapy2.

Начнем с простой формы просмотра данных по медикаментам. Раскроем свойство SQL компонента и введем следующий текст SQL-запроса для компонента ADOQTherapy1:

**Select Title, Treatment**

**From Medications, Therapy**

**Where №Medication = IDMedication and №Patient = :IDPatient**

текст SQL-запроса для компонента ADOQTherapy2:

**Select Surname, Treatment**

**From Patients, Therapy Where**

**№Patient = IDPatient and №Medication = :IDMedication**

текст SQL-запроса для компонента ADOQPatient:

**Select IDPatient, Surname, NamePatient, Patronymic, DOB, Cellphone, Photo, Sector**

**From Patients**

**Where Surname Like :Par1 and NamePatient Like :Par2**

**Order By IDPatient**

В форме просмотра нужно сетку Grid1 и навигатор связать с результатами запроса ADOQMedication, указав в свойстве DataSource значение DataModule3.DataSource2, а сетку Grid2 связать с запросом ADOQTherapy1, задав значение DataModule3.DataSource4. Здесь и далее DataModule3 определяет имя модуля данных. В обработчике события OnCreate модуля данных откроем все запросы операторами типа ADOQStud.Open.

Параметры запроса Par1 и Par2 потребуется для выбора пациентов по частичным значениям полей имени и фамилии, задаваемых пользователем. Кроме того, потребуется сортировка по фамилиям и по имени с фамилиями, поэтому добавим в запрос условие сортировки.

Укажем для обоих параметров в свойстве Value значение '%', которое соответствует любой последовательности символов. В свойстве DataType задается ftString.

Займемся режимами сортировки. Вид сортировки определяет режим Order By, находящийся в третьей, считая с нуля, строке запроса. По клавише Отмена будем аналогичным образом восстанавливать исходный порядок сортировки.

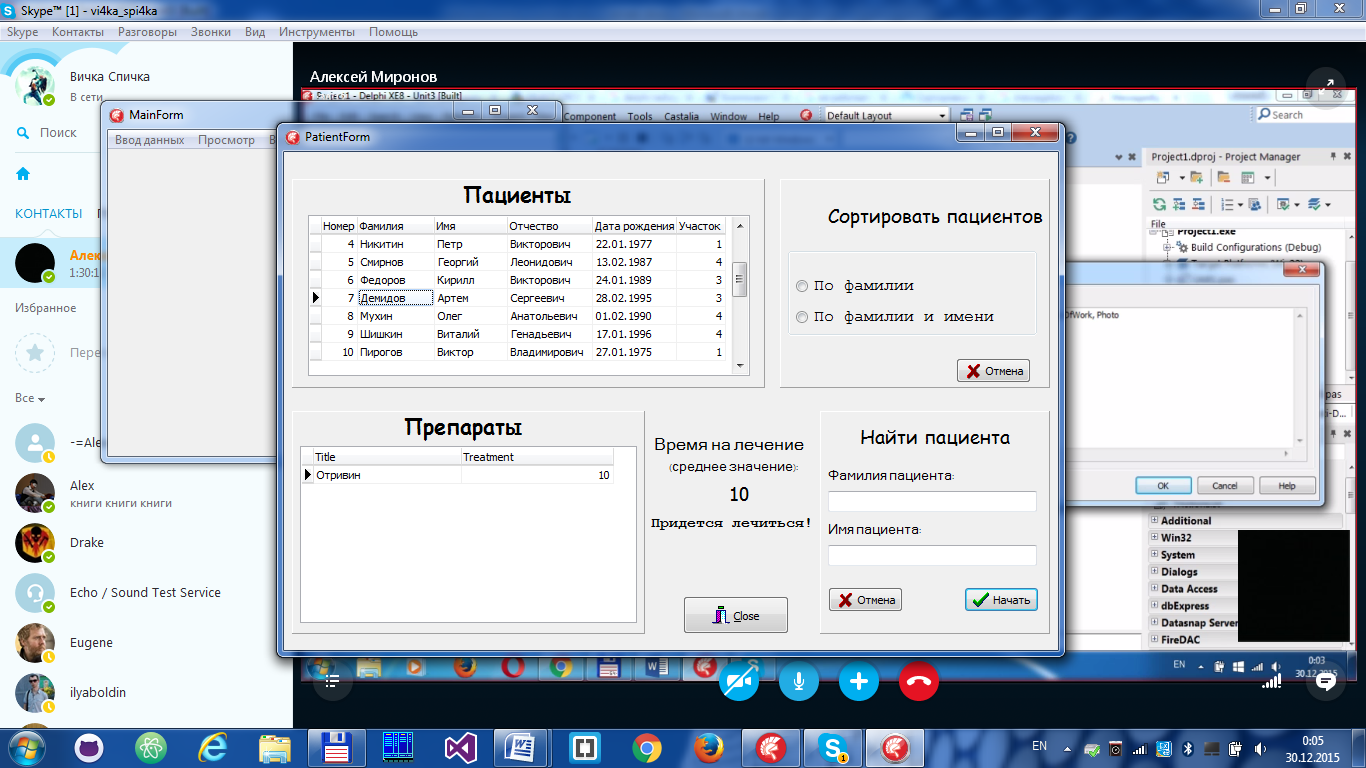
Для поиска пациентов по частичным значениям имени и фамилии, задаваемым в компонентах редактирования Edit1 и Edit2, будем использовать параметры запроса Par1 и Par2. При задании новых значений параметров требуется запрос закрыть, а потом открыть его повторно. В форме просмотра для текущего студента в сетке Grid1 будем показывать его среднее время лечения с соответствующим комментарием. Для получения такой оценки будем использовать SQL-запрос, компонент ADOQAvgTherapy

**Select Avg(Treatment) as AvgO from Therapy**

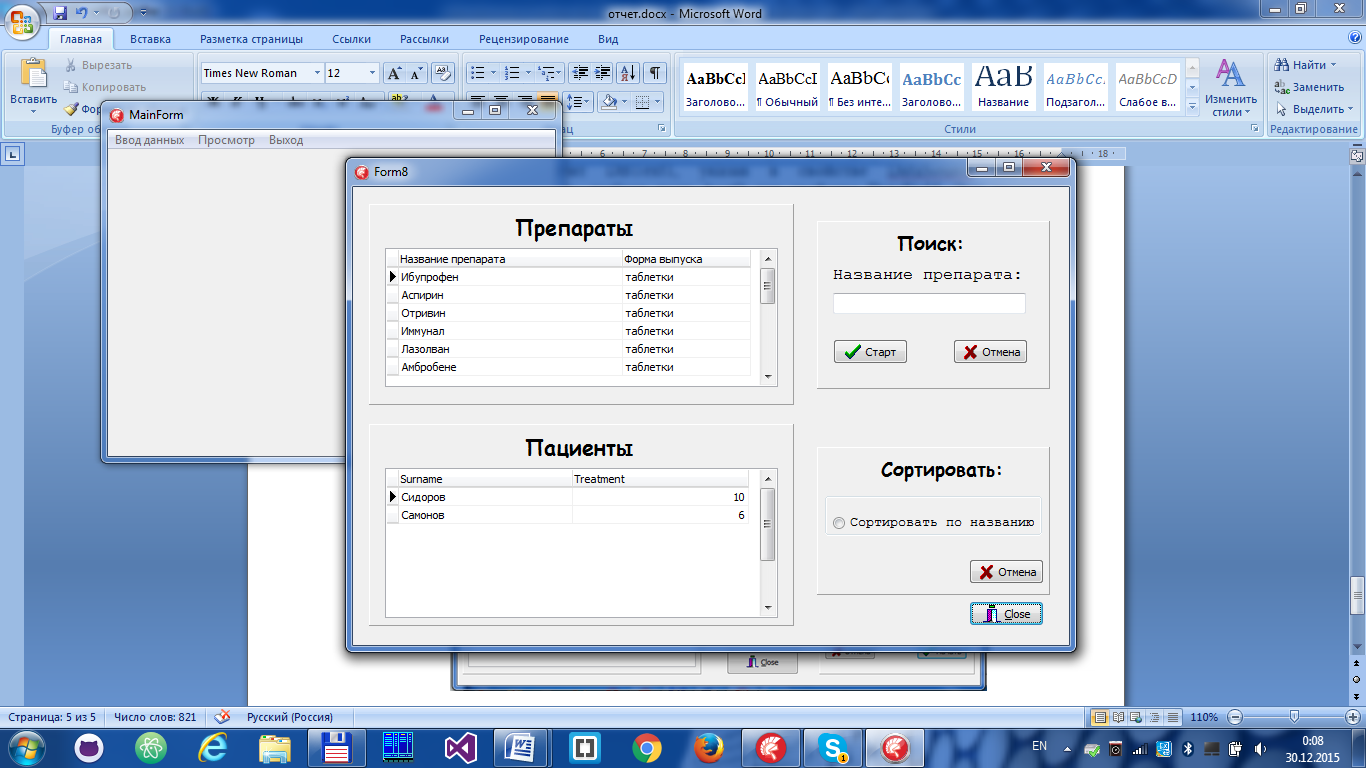
**Where №Patient=:IDPatient**

Параметр IDPatient будет определяться автоматически, если в свойстве DataSource задать значение DataSource1, что соответствует запросу ADOQPatient.

В форме просмотра по пациентам для демонстрации среднего времени лечения предназначим компонент DBText1, указав в свойстве DataSource значение DataModule3.DataSource5 и выбрав поле AvgO для свойства DataField. Эти значения связывают компонент DBText1 с запросом ADOQAvgTherapy.

Результат запроса представляет собой обычную таблицу или набор данных в терминах Delphi. Для иллюстрации этого приведем обработчик события AfterScroll запроса ADOQPatient, дающий с помощью метки Label7 комментарий по среднем времени лечения пациента.

Форма просмотра по пациентам



Форма просмотра по препаратам

**Формы ввода-редактирования данных таблицы Patient**

Организуем ввод и редактирование информации таблицы Patient. На форме ввода разместим компоненты DBEdit1 – DBEdit6 для полей Surname, NamePatient, Patronymic, DOB, Cellphone, Sector соответственно и DBImage1 для поля Photo, CheckBox1 с подписью "Редактировать запись", DBNavigator1 для перемещения по записям и кнопки с рисунками BitBtn1 - BitBtn3 с надписями "Сохранение", "Отмена", "Выход". Все компоненты свяжем с источниками из модуля данных, заполнив свойства DataSource.

Для поля Tel в свойстве EditMask зададим маску ввода 8(###)###-##-##, а для поля DOB маску ##.##.#### (число, месяц, год).

Необходимо проверять правильность заполнения даты рождения. Для этого в обработчике события OnSetText поля Birth (объект StudBirth в дереве объектов) напишем функции проверки.

Здесь Form8.DBEdit4.Text определяет поле ввода даты. Событие OnSetText наступает после изменения значения поля, но до его запоминания в таблице.

Перейдем к форме ввода данных. При открытии формы будем допускать только просмотр данных. В этом режиме нужно запретить корректировку значений полей, сделать недоступными кнопки сохранения и отмены, активизировать навигатор.

При активизации формы (событие OnActivate) обратимся к приведенной процедуре и сделаем текущей первую запись таблицы

**DataModul3.ADOQPatient.First;**

**DBEdit1.SetFocus; // курсор на поле фамилии**

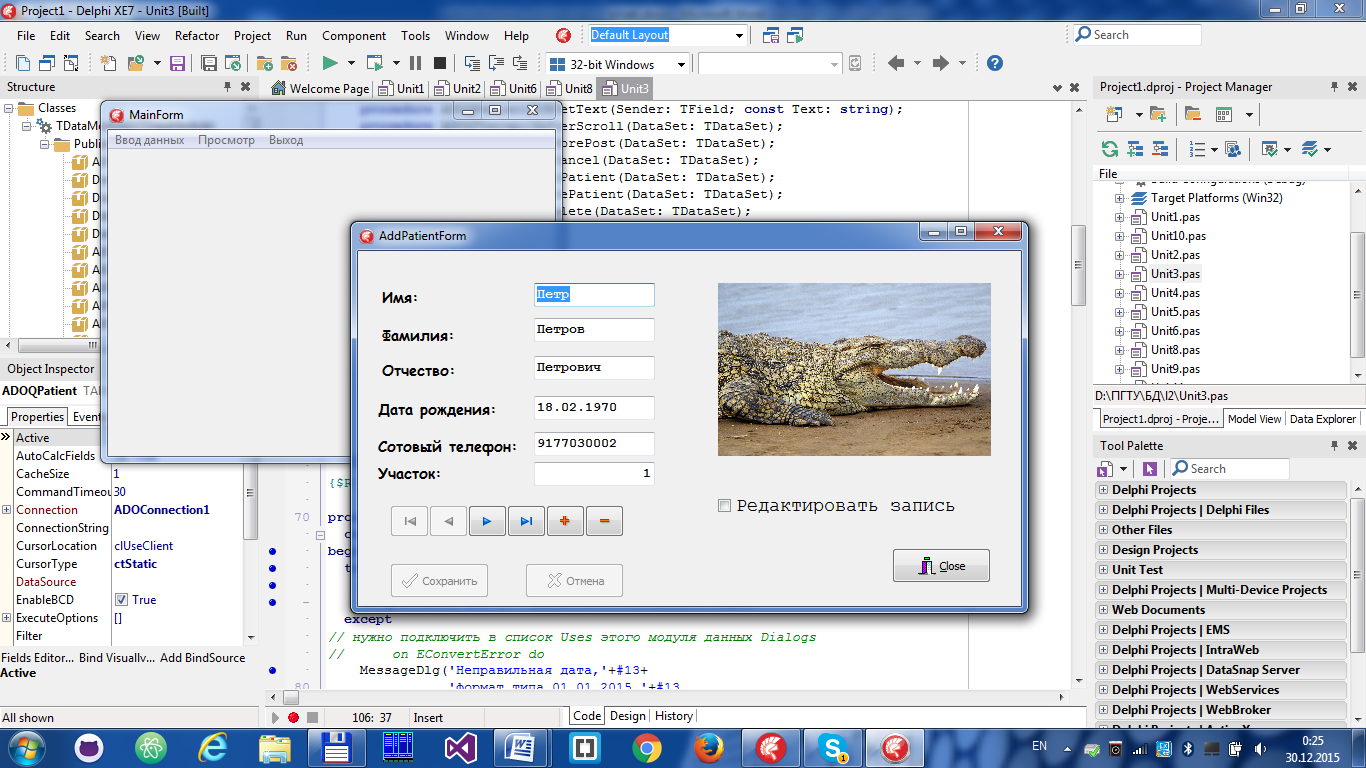
В обработчике события OnClick переключателя CheckBox1 будем вызывать процедуру, соответствующую режиму работы с данными

По клавише Отмена нужно отказаться от сделанных изменений данных и вернуться в режим редактирования.

По клавише Выход переходим к режиму редактирования данных, выбранному во всем приложении по умолчанию, закроем форму и вернемся в меню.

По клавише Сохранение происходит, проверка возраста и сохраняется новый результат.

Ввод новой фотографии можно выполнить, обрабатывая событие щелчка мышью в области рисунка Image1:

Форма редактирования таблицы Patient

# Форма ввода-редактирования данных таблицы Therapy

Создадим форму для просмотра и редактирования способа лечения. В модуле данных добавим пару компонент ADOQTherapy3 – DataSource7. Свойством SQL зададим базовый запрос:

**Select IDPatient, Surname, IDMedication, Title, №Patient, №Medication, Treatment**

**From Patients, Medications, Therapy**

**Where (IDPatient=№Patient) and (IDMedication=№Medication)**

**and (Surname Like :Par3) and (Title Like :Par4)**

**Order By Surname, Title**

Параметры Par3 и Par4 предназначены для поиска оценки по имени студента и названию предмета.

В свойстве FieldKind для полей Surname и Title выберем тип fklookup. Это позволит выбирать значения полей из имеющихся значений в таблицах Patients и Medications. Поля типа fklookup требуют определенных свойств для связи с базовыми таблицами. Так для поля Surname указываются свойства:

**KeyFields – Patient;**

**LookupKeyFields – IDPatient;**

**LookupResulnField – Surname.**

Поля редактирования Edit1 и Edit2 потребуются для поиска оценки по имени пациента и названию препарата с помощью параметров запроса Par3 и Par4. Фильтрация записей происходит сразу же после изменения значений полей Edit1 и Edit2.

С помощью навигатора обеспечим только продвигаться по записям НД ADOQTherapy3, а остальные функции реализуем кнопками. Режим редактирования будем включать и отключать объектом CheckBox.

Для удаления оценки разместим в модуле данных компонент ADOQDelExam3 с параметрическим запросом SQL:

**Delete from Therapy**

**Where (№Patient= :StId) and (№Medication=:PrId)**

Здесь параметры Stid и Prid должны задавать идентификаторы пациента и терапии удаляемого лечения.

Редактирование и вставка могут завершаться как клавишей сохранения, так и переходом на другую запись. Добавим в модуль данных запрос на вставку записи со следующим параметрическим запросом SQL:

**insert into Therapy(№Patient,№Medication,Treatment)**

**Values(:EPat,:EMed,:ETher)**

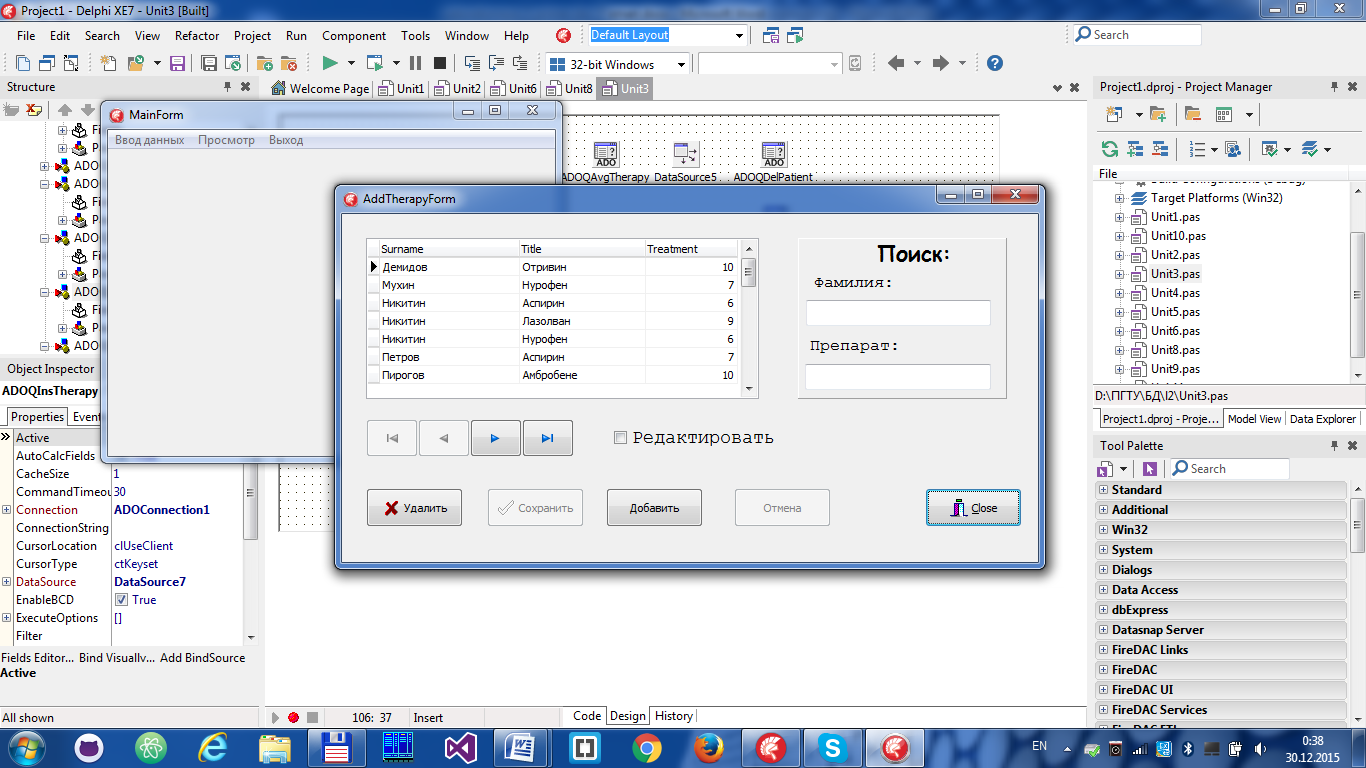
Параметры EPat, EMed, ETher задают содержимое вставляемой записи и должны быть инициализированы перед выполнением запроса. Фактическая вставка происходит при нажатии клавиши сохранения либо при переходе на другую запись.

Как и для формы просмотра, необходимо связать модуль данных с формой редактирования лечения, указав имя соответствующего форме модуля в операторе Uses.

Наконец, вставка записи может быть прервана пользователем. В этом случае нужно восстановить свойства доступности объектов для режима просмотра информации. Отмена редактирования выполняется оператором

DataModule3.ADOQExam3.Cancel .

Изменение параметров запросов требует закрытия и повторного открытия соответствующих компонентов.

Форма просмотра и редактирования терапии.

# Отчеты

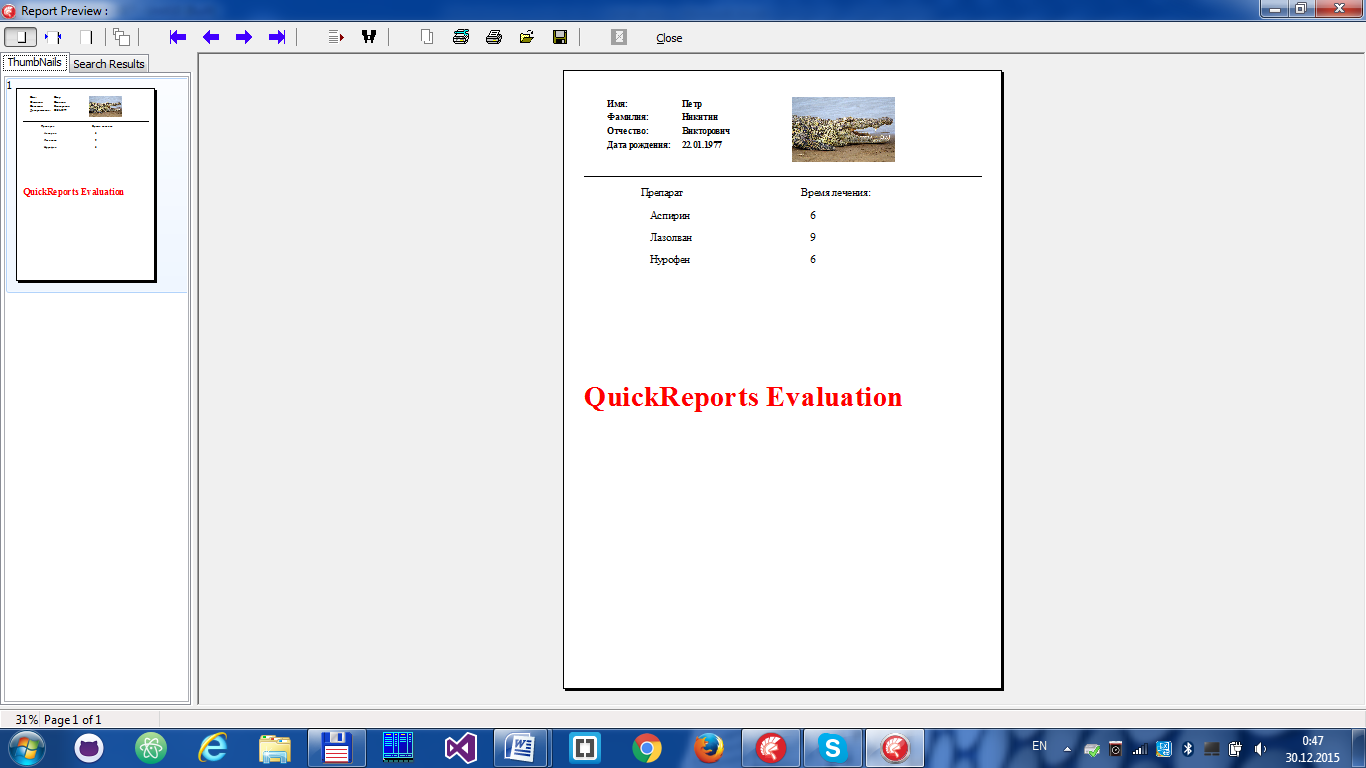
Отчеты создаются с помощью компонента QReport.

В форме просмотра пациенты-препараты через контекстное меню сетки с информацией о пациентах, которое вызывается правой кнопкой мышки, будем выводить три вида отчетов об лечении: по конкретному пациенту, по всем пациентам и в совокупности и по участкам. Начнем с первого отчета.

Поместим в форму центральный компонент отчета QuickRep. Добавим на форму все необходимые поля.

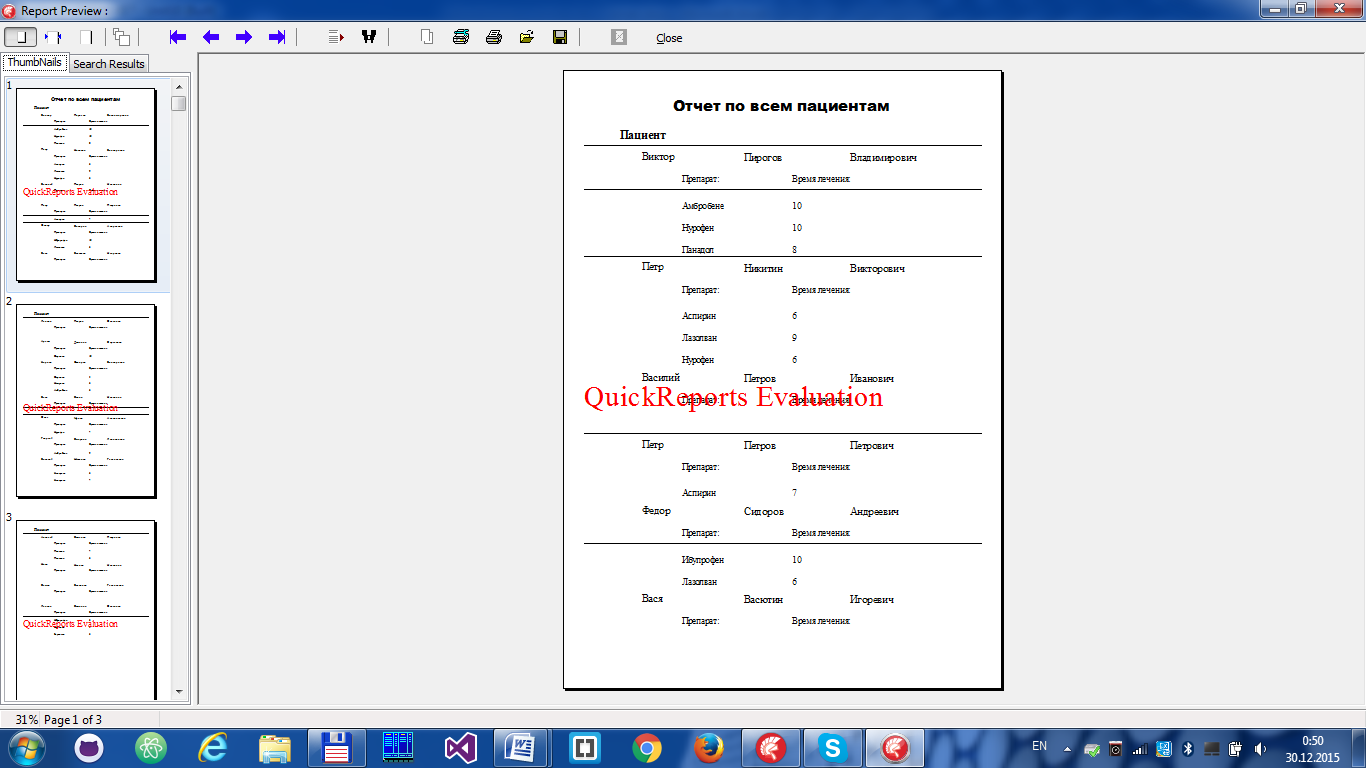
Свяжем модуль, соответствующий форме просмотра пациенты-препараты с разработанным отчетом. Выберем компонент контекстного меню PopupMenu из страницы Standart. В свойстве PopupMenu сетки с информацией о пациентах укажем имя нового компонента. Зададим три опции "Отчет о пациенте", "Отчет о всех пациентах " и "Отчет по участкам", соответствующие видам отчетов. Приведем обработчик первой опции меню для вызова отчета по одному студенту.

Метод Preview позволяет выдать отчет на экран и управлять его просмотром. Для распечатки отчета служит метод Print.



Отчет по выбранному пациенту

Второй отчет

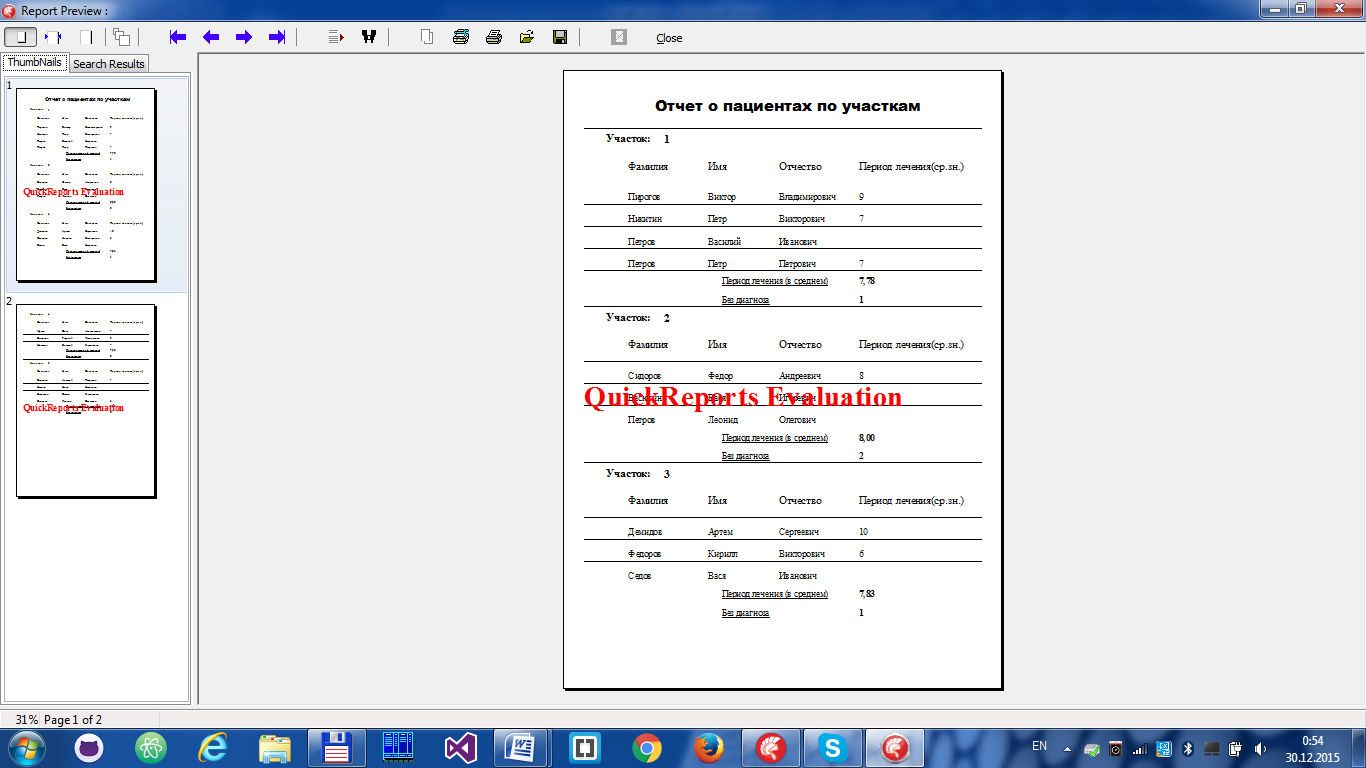


Отчет по всем пациентам

Третий отчет должен содержать сгруппированную информацию по участкам. НД ADOQPatient, поэтому в свойстве DataSet установим значение DataModule3. ADOQPatient. В обработчике события OnCreate для формы отчета зададим сортировку по участку.

**DataModule3.ADOQPatient.SQL[3]:='Order By IDPatient';**

**DataModule3.ADOQPatient.Open;**



Отчет о пациентах по участкам