Министерство образования и науки Республики Татарстан

Государственное автономное профессиональное образовательное учреждение

«Технический колледж им.В.Д. Поташова»

09.02.03 Программирование в компьютерных системах

МДК 03.02. Инструментальные средства разработки программного обеспечения

КУРСОВАЯ РАБОТА

На тему: Разработка автоматизированной информационной системы агентства рекламных услуг\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил студент \_\_IV\_\_ курса, \_\_\_\_\_Ханов Динар Раифович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

Шифр \_РК.5379.47.25.ПЗ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель преподаватель профессионального цикла, Миннуллина Р.Г.\_\_\_\_\_\_\_\_\_\_\_\_

должность, звание Ф.И.О.\_

К защите Защита принята с оценкой

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022\_\_г. \_\_\_\_\_«\_\_\_\_»\_\_\_\_\_\_ 2022\_\_г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись Подпись

Набережные Челны, 2021 год

СОДЕРЖАНИЕ

[Введение 3](#_Toc69330879)

[1 Описание предметной области 5](#_Toc69330880)

[1.1 Исследование автоматизируемой области 5](#_Toc69330881)

[1.2 Разработка алгоритма программы 8](#_Toc69330882)

[1.3. Обоснование выбора языка программирования 10](#_Toc69330883)

[2 Информационное обеспечение задачи 13](#_Toc69330884)

[2.1 Входные данные 13](#_Toc69330885)

[2.2 Выходные данные 13](#_Toc69330886)

[2.3 Описание программы 14](#_Toc69330887)

[2.4 Инструкция пользователю 14](#_Toc69330888)

[Заключение 18](#_Toc69330889)

[Список используемой литературы 20](#_Toc69330890)

[ПРИЛОЖЕНИЯ 21](#_Toc69330891)

# Введение

В процессе организации своей работы какие-либо документы, неизбежно сталкивается с необходимостью их обработки. Под обработкой, понимается сортировка данных, поиск данных по указанной совокупности параметров, добавление, изменение и удаление записей. Сложность обработки данных прямо пропорциональна объему хранимой информации.

Для разрешения проблемы, связанной с обработкой больших объемов данных, необходима разработка информационной системы, обеспечивающей обработку данных из выбранной предметной области. Грамотное проектирование структуры хранения данных и средств их обработки позволяют решить практически любую задачу, связанную с поиском и сортировкой исходного хранилища.

Автоматизация процесса работы издательского учета позволит сотрудникам решать свои ежедневные должностные обязанности на качественно новом уровне.

Организовав автоматизацию процесса учета, можно сократить временные затраты на выполнение сотрудниками рекламного агентства различных работ, что в свою очередь, позволит создать свободную временную нишу и уделить больше внимания другим не менее важным вопросам.

Целью курсовой работы является:

* систематизации и закрепления полученных теоретических знаний и практических умений по профессиональному модулю ПМ 03. Участие в интеграции программных модулей;
* формирования умений применять теоретические знания при решении поставленных вопросов;
* приобретение практических навыков по написанию программ на языке объектно-ориентированного программирования Object Pascal;

овладение профессиональными (ПК) и общими (ОК) компетенциями:

* ПК 2. Выполнять интеграцию модулей в программную систему.
* ПК 3. Выполнять отладку программного продукта с использованием специализированных программных средств.
* ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
* ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
* ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

Задачами курсовой работы являются:

* изучение особенностей конкретной предметной области, относящихся к теме курсовой работы;
* теоретический анализ научной и методической литературы по проблеме разработки приложения агентства рекламных услуг;
* анализ возможных подходов и методов решения с обоснованием выбранного метода;
* выбор или разработка информационной модели, необходимой для достижения цели;
* разработка информационных систем, включающая как проектирование баз данных, так и создание программного обеспечения;
* проектирование интерфейса программы и порядок ее взаимодействия с пользователем;
* разработка программной и эксплуатационной документации.

В итоге главной задачей курсового проекта является создание надежной, эффективной и незаменимой в работе системы, которая будет в помощь сотрудникам рекламного агентства по предоставлению услуг.

# 1 Описание предметной области

## 1.1 Исследование автоматизируемой области

Целью разрабатываемой информационной системы является автоматизация учета заказов услуг рекламного агентства.

Услуги рекламного агентства – дизайн фасадов, витрин, вывесок, табличек, интерьера, и любой наружной рекламы.

Производство всех видов наружной рекламы и входных групп и производится из высококачественных материалов на собственном современном оборудовании. В техпроцессе используется лазерная резка, и фрезерование нержавеющей стали и пластика, гибка металлопроката, широкоформатная печать, неон. Что позволяет добиться высоких эстетических и эксплуатационных качеств продукции.

Пользователями системы должна быть сотрудники рекламного агентства.

Система должна позволять:

* создавать новые записи о клиентах
* вносить информацию о заказах, путём заполнения накладной
* вносить/обновлять информацию о видах услуг

На предоставление услуг клиенту, заполняется накладная. Накладная может содержать несколько услуг, но услуги не повторяются, для заказа несколько услуг одного вида существует поле кол-во. Накладную заполняет работник рекламного агентства. Общая стоимость вычисляется в зависимости от выбранных услуг рекламного агентства. Накладная может быть выписана только на одного заказчика. Накладная может быть заполнено только одним сотрудником рекламного агентства. Цена услуги является изменяемым полем. В накладную проставляются цены, действующие на момент заказа. Изменение текущей цены услуги не повлияет на предыдущие накладные.

Входные документы:

1. Информация о заказчике:

* ФИО клиента
* паспортные данные;
* E-mail;
* контактный телефон.

1. Информация о предоставляемых услугах:

* ФИО
* информация о предоставленных услугах
* количество;
* цена продажи
* наименование

1. Информация о накладной

* номер накладной;
* дата заполнения;
* сумма.

Результатом работы информационной системы, должны быть отчеты

* популярные услуги;
* активность сотрудников;
* частота возвращения клиентов.

В реализуемой программе должны выполняться следующие требования:

* программа должна обладать понятным интерфейсом и относительной простотой работы;
* добавлять, редактировать, удалять и хранить данные по сотрудникам, заказам и предоставляемым услугам;
* легко отследить и управлять всеми основными данными рекламных услуг и заказов;
* формировать отчет по заказам, сотрудникам и рекламным услугам;
* выводом отчета на принтер.

Основные задачи, решаемые при разработке программного обеспечения:

* спроектировать структуру информационной системы и взаимодействие частей системы;
* определить виды данных, циркулирующих в информационной системе;
* разработать структуру базы данных;
* выбрать алгоритмы проектирования;
* обосновать методы реализации информационной системы.

После исследования предметной области и разработки постановки задачи, для хранения вводимых данных были созданы следующие таблицы:

* «Клиенты» - содержит информацию о клиентах;
* «Услуги» - содержит данные об услугах рекламного агентства;
* «Сотрудники» - содержит данные о сотрудниках;
* «Накладная» - содержит данные о накладных.

Таблица 1 Описание полей таблицы «Клиенты»

|  |  |  |
| --- | --- | --- |
| Наименование поля | Тип | Длина |
| Код клиента | Счетчик | 10 |
| ФИО | Текстовый | 50 |
| Паспортные данные | Текстовый | 60 |
| Телефон | Текстовый | 15 |
| Email | Текстовый | 20 |

Таблица 2 Описание полей таблицы «Услуги»

|  |  |  |
| --- | --- | --- |
| Наименование поля | Тип | Длина |
| Код услуги | Счетчик | 10 |
| Наименование | Текстовый | 25 |
| Цена | Денежный | 20 |

Таблица 3 Описание полей таблицы «Сотрудники»

|  |  |  |
| --- | --- | --- |
| Наименование поля | Тип | Длина |
| Код сотрудника | Счетчик | 10 |
| ФИО | Текстовый | 50 |

Таблица 4 Описание полей таблицы «Накладная»

|  |  |  |
| --- | --- | --- |
| Наименование поля | Тип | Длина |
| Номер накладной | Числовой | 30 |
| Дата оформления | Дата | дата |
| Код\_клиента | Числовой | 10 |
| Код\_сотрудника | Числовой | 20 |
| Сумма | Денежный | 20 |

**Физическая модель данных**

На рисунке 1 изображена физическая модель базы данных. Помимо структуры базы данных, на ней указаны конкретные типы данных. Данная схема нарисована средствами Microsoft Visio 2007 на базе диаграммы IDEF1X.



Рисунок 1 **–** Физическая модель базы данных

## 1.2 Разработка алгоритма программы

Целью разрабатываемого приложения является создание автоматизированной информационной системы рекламного агенства.

Автоматизированная информационная система должна позволять:

* хранение информации в базе данных;
* учет сведений по клиентам и предоставляемым услугам;
* формирование справочника изданий и справочника по услугам и клиентам;
* осуществления поиска, фильтрации и сортировка данных таблиц;
* формирование отчета по выполненным заказам за определенный период.

Разрабатываемая программа имеет следующую функциональную схему

Автоматизированная информационная система рекламного агентства

Каталог услуг

Ввод издания

Редактирование

Удаление

Сортировка

Просмотр

Сотрудники

Просмотр

Ввод сотрудника

Редактирование

Удаление

Сортировка

Клиенты

Просмотр

Ввод заказчика

Редактирование

Удаление

Сортировка

Отчеты

Рисунок 2 – Функциональная схема работы рекламного агентства

## 1.3. Обоснование выбора языка программирования

Для создания программы была выбрана система программирования Delphi версии 7. Delphi – это продукт Borland International для быстрого создания приложений.

Высокопроизводительный инструмент визуального построения приложений включает в себя настоящий компилятор кода и предоставляет средства визуального программирования, несколько похожие на те, что можно обнаружить в Microsoft Visual Basic (она не является RAD-системой) или в других инструментах визуального проектирования. В основе Delphi лежит язык Object Pascal, который является расширением объектно-ориентированного языка Pascal. В Delphi также входят локальный SQL-сервер, генераторы отчетов, библиотеки визуальных компонентов, и прочее, необходимое для того, чтобы чувствовать себя совершенно уверенным при профессиональной разработке информационных систем или просто программ для Windows-среды.

Преимущества Delphi по сравнению с аналогичными программными продуктами.

* быстрота разработки приложения (RAD);
* высокая производительность разработанного приложения;
* низкие требования разработанного приложения к ресурсам компьютера;
* наращиваемость за счет встраивания новых компонент и инструментов в среду Delphi;
* возможность разработки новых компонентов и инструментов собственными средствами Delphi (существующие компоненты и инструменты доступны в исходных кодах);
* удачная проработка иерархии объектов.

Система программирования Delphi рассчитана на программирование различных приложений и предоставляет большое количество компонентов для этого. К тому же работодателей интересует, прежде всего, скорость и качество создания программ, а эти характеристики может обеспечить только среда визуального проектирования, способная взять на себя значительные объемы рутинной работы по подготовке приложений, а также согласовать деятельность группы постановщиков, кодировщиков, тестеров и технических писателей. Возможности Delphi полностью отвечают подобным требованиям и подходят для создания систем любой сложности

В Delphi используются все самые современные методы программирования: модель составных объектов COM (Component Object Model), распределённая модель составных данных DCOM (Distributed Component Object Model), технология программных компонентов ActiveX и другие.

Благодаря сумме технологий dbExpess, BDE, DataSnap, ADO и СОМ очень легко разрабатывать приложения баз данных любой сложности. Связь с базой данных в Delphi устанавливается настройкой всего нескольких свойств и заданием пары-тройки дополнительных параметров.

В качестве хранилища данных используется база данных, созданная на основе СУБД Access.

На основе выявленных выше преимуществ, для разработки автоматизированной информационной системы целесообразно использовать язык программирования Delphi, и технологию ADO на базе той же среды, основанную для работы с базами данных MS Access. Для присоединения базы данных Access использована технология ADO. База данных присоединяется к проекту при помощи компонента ADOConnection.

Для предоставления доступа к таблицам и запросам, графическим элементам управления используется компонент DataSource. Связав DataSource с необходимой таблицей (ADOTable) или запросом (ADOQuery), а затем связав необходимый графический элемент управления с ним, создается простой интерфейс для вывода данных на экран или принтер.

Ниже представлен набор необходимых компонентов для интеграции базы данных Microsoft Access в разрабатываемую программу с помощью технологии ADO.

В состав необходимых компонентов входят:

* DBGrid – компонент представляет собой таблицу, предназначенную для отображения БД в приложении.
* DataSource – предназначен для связи DBGrid и базы данных.
* ADOConnection – предназначен для управления соединением с базой данных.
* ADOTable – предназначен для выполнения запросов и получения нужных результатов из нашей БД.

# 2 Информационное обеспечение задачи

## 2.1 Входные данные

Для хранения вводимых данных были созданы следующие таблицы:

* «Клиенты» – содержит информацию о клиентах рекламного агентства;
* «Услуги» – содержит данные об оказываемых услугах;
* «Сотрудники» – содержит данные о сотрудниках агентства;
* «Накладная» – содержит данные о накладных.

## 2.2 Выходные данные

Результаты обработки входной информации можно просмотреть с помощью экранных форм и отчетов. Воспользовавшись удобной навигацией, пользователь может просмотреть информацию со справочников в формах, а также просмотреть и распечатать отчеты.

Перечень печатных форм и отчётов предоставляемых пользователю:

* форма Клиентов – просмотр и редактирование списка клиентов;
* форма Услуг – просмотр и редактирование услуг рекламного агентства;
* форма Сотрудники – просмотр и редактирование списка сотрудников;
* форма Накладная – просмотр и редактирование данных об проведенных операциях по накладным;
* отчет по услугам, общему количеству услуг и их цене;
* отчет по накладным с подробной информацией.

## 2.3 Описание программы

Программная часть дипломного проекта представлена исполнительным файлом Agentrec.exe и файлом с базой данных rec.mdb.

В процессе разработки исходного кода программы использовался объектно-ориентированный язык программирования Embarcadero Delphi 10.2.

Для функционирования программного продукта требуется следующее программное обеспечение:

* процессор не ниже Pentium 4;
* 1 Гб оперативной памяти;
* операционная система Windows XP;
* пакет прикладных программ Microsoft Office -2007 или 2010;
* среда программирования Embarcadero Delphi 10.2.

При работе необходимо иметь устройство вывода на печать.

При создании программного изделия были использованы такие языки как Object Pascal и SQL в Microsoft Access.

## 2.4 Инструкция пользователю

При разработке пользовательского интерфейса программы реализованы все необходимые окна.

Для открытия приложения необходимо запустить с папки установленную программу «Agentrec.exe».

После этого загрузится приложение и на экране появится главная форма программы с названием «Главная страница» (Рисунок 3).



Рисунок 3 – Главное окно программы

На рисунке 4 показано окно просмотра, редактирования и внесения новой информации по клиентам.

В левой части - список клиентов, в правой - поля для редактирования и добавления клиентов.



Рисунок 4 – Окно со списком клиентов

На рисунке 5 показано окно просмотра, редактирования и внесения новой информации по услугам.

В левой части - список услуг, в правой - поля для редактирования и добавления услуг.



Рисунок 5 – Окно со списком услуг

На рисунке 6 показано окно оформления новой накладной.

Для заполнения необходимо выбрать Работника, Клиента, и заполнить список услуг. Кнопка «В Корзину» добавляет услугу в список услуг. Кнопка «Очистить» очищает весь список услуг. Кнопка «Оформить» проводит накладную.



Рисунок 6 – Окно оформления накладной

**Работа с меню «Отчеты»**

Пункт меню «Отчеты», предназначен для формирования, просмотра и печати отчетов по хранящимся в системе данных. При выборе указанного меню, появится следующий подчиненный список меню: количество книг, операция.

# Заключение

Информационная система - это система для поддержки принятия решений и предоставления услуг, используя компьютерную информационную технологию. Основная цель информационной системы - организация хранения поиск и передача информации.

Программное обеспечение – все или часть программ, процедур, правил и соответствующей документации системы обработки информации.

Сегодня появилась реальная возможность с помощью моделирования на современных многофункциональных средствах обработки и отображения информации, таких как Embarcadero Delphi, конкретизировать тип и характеристики используемых информационных моделей, выявить основные особенности будущей деятельности операторов, сформулировать требования к параметрам аппаратно-программных средств интерфейса взаимодействия.

Для этого было собрана исчерпывающая информация об организации учета предоставляемых услуг рекламным агентством. После чего выявлена область для автоматизации, выбрана информация, которая должна храниться в базах данных для её лучшей обработки и анализа. После были разработаны необходимые хранилища информации, программные модули и интерфейс пользователя.

Благодаря автоматизации учета и регистрации рекламных услуг, которые позволят оперативное сохранение данных и их реализации способствуют снижению влияния личных качеств персонала на выполнение процессов обработки документов, повышению надежности хранения, а также минимизации временных затрат на поиск и анализ необходимых изданий.

В ходе выполнения курсовой работы были закреплены знания по профессиональному модулю ПМ 03 «Участие в интеграции программных модулей», приобретены практические навыки по написанию программ на языке программирования Delphi 7, освоил профессиональные (ПК) и общие (ОК) компетенции:

* ПК 2. Выполнять интеграцию модулей в программную систему.
* ПК 3. Выполнять отладку программного продукта с использованием специализированных программных средств.
* ПК 4. Осуществлять разработку тестовых наборов и тестовых сценариев.
* ПК 5. Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования
* ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
* ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
* ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

Я считаю, что поставленные передо мной задачи выполнены в полном объеме.

# Список используемой литературы

1. Культин, Н. Б. Основы программирования в Embarcadero Delphi, - Интернет-издание, 2017. – 232с;
2. Фленов, М. Е. Библия Delphi. — СПб.: БХВ-Петербург, 2017.— 686 с.: ил.;
3. Фленов, М. Е. Программирование в Delphi глазами хакера. — СПб.: БХВ-Петербург, 2017. — 368 с.: ил.
4. Осипов, Д. Л. Базы данных и Delphi. Теория и практика. — СПб.: БХВ-Петербург, 2018.— 752 с.: ил.;
5. Горошков, Б.И. Автоматическое управление: Учебник для студентов учреждений СПО – М.:ИРПО: Издательский центр «Академия», 2017. – 304.
6. Сайт про программирование http://thedelphi.ru/
7. Википедия, язык программирования Delphi <http://ru.wikipedia.org/wiki/Delphi>
8. Самоучитель Delphi 7 <http://am-ru.delphi.com/>
9. Миннуллина, Р.Г. Лекционный материал по МДК 03.02 «Информационные системы разработки программного обеспечения».

# ПРИЛОЖЕНИЯ

Листинг программы

**unit Unit1;**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, DB, ADODB, StdCtrls, ExtCtrls, jpeg, Buttons, Menus;

type

TGlavForm = class(TForm)

Button1: TButton; Button2: TButton; Button3: TButton;

Image1: TImage; Button4: TButton; Button5: TButton;

Button6: TButton; MainMenu1: TMainMenu;

N1: TMenuItem; N2: TMenuItem; W1: TMenuItem;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure W1Click(Sender: TObject);

procedure N3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

GlavForm: TGlavForm;

implementation

uses Unit2, Unit4, Unit5, Unit3, Unit6, Unit7, Unit9, otchetUnit10, Unit11,

Unit12;

{$R \*.dfm}

procedure TGlavForm.Button1Click(Sender: TObject);

begin

Form2.Show;

end;

procedure TGlavForm.Button2Click(Sender: TObject);

begin

Form3.Show;

end;

procedure TGlavForm.Button3Click(Sender: TObject);

begin

Form5.Show;

end;

procedure TGlavForm.Button4Click(Sender: TObject);

begin

close;

end;

procedure TGlavForm.Button5Click(Sender: TObject);

begin

form6.show;

end;

procedure TGlavForm.Button6Click(Sender: TObject);

begin

form7.show;

end;

procedure TGlavForm.N2Click(Sender: TObject);

begin

form9.QuickRep1.PreviewModal;

end;

procedure TGlavForm.W1Click(Sender: TObject);

begin

Zapros\_otcheta.show;

end;

procedure TGlavForm.N3Click(Sender: TObject);

begin

Form12.show;

end;

end.

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, jpeg, ExtCtrls, Menus,

DBCtrls;

type

TForm2 = class(TForm)

DBGrid1: TDBGrid; MainMenu1: TMainMenu;

N1\_File: TMenuItem; N2\_Zakr: TMenuItem;

N3\_Redak: TMenuItem; N4\_Dobav: TMenuItem;

N5\_Redak: TMenuItem; N6\_sohran: TMenuItem;

N7\_Ydal: TMenuItem; N8\_Sort: TMenuItem;

N9\_avtor: TMenuItem; N10\_nazvan: TMenuItem;

N11\_kolich: TMenuItem; N12\_cena: TMenuItem;

Btn1\_zakr: TButton; Button1: TButton;

Btn2\_Redak: TButton; Button3: TButton;

DBNavigator2: TDBNavigator; Panel4: TPanel;

procedure N9\_avtorClick(Sender: TObject);

procedure N10\_nazvanClick(Sender: TObject);

procedure N11\_kolichClick(Sender: TObject);

procedure N12\_cenaClick(Sender: TObject);

procedure N2\_ZakrClick(Sender: TObject);

procedure N4\_DobavClick(Sender: TObject);

procedure N6\_sohranClick(Sender: TObject);

procedure N7\_YdalClick(Sender: TObject);

procedure N5\_RedakClick(Sender: TObject);

procedure Btn1\_zakrClick(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Btn2\_RedakClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses DataModuleUnit, dobavlenie\_knig\_Unit;

{$R \*.dfm}

procedure TForm2.N9\_avtorClick(Sender: TObject);

begin

DataModule1.ADOTable1.IndexFieldNames:= 'Автор';

end;

procedure TForm2.N10\_nazvanClick(Sender: TObject);

begin

DataModule1.ADOTable1.IndexFieldNames:= 'Название';

end;

procedure TForm2.N11\_kolichClick(Sender: TObject);

begin

DataModule1.ADOTable1.IndexFieldNames:= 'Kolvo';

end;

procedure TForm2.N12\_cenaClick(Sender: TObject);

begin

DataModule1.ADOTable1.IndexFieldNames:= 'Cena';

end;

procedure TForm2.N2\_ZakrClick(Sender: TObject);

begin

close;

end;

procedure TForm2.N4\_DobavClick(Sender: TObject);

begin

DataModule1.AdoTable1.Insert;

dobavlenie\_knig.ShowModal

end;

procedure TForm2.N6\_sohranClick(Sender: TObject);

begin

if DataModule1.ADOTable1.Modified then

DataModule1.ADOTable1.Post;

end;

procedure TForm2.N7\_YdalClick(Sender: TObject);

begin

if Application.MessageBox('Вы уверен что хотите удалить строку?', 'Внимание', MB\_ICONQUESTION + MB\_YESNOCANCEL) = IDYES then DataModule1.ADOTable1.Delete;

end;

procedure TForm2.N5\_RedakClick(Sender: TObject);

begin

DBGrid1.setFocus;

dobavlenie\_knig.ShowModal

end;

procedure TForm2.Btn1\_zakrClick(Sender: TObject);

begin

close;

end;

procedure TForm2.Button3Click(Sender: TObject);

begin

if Application.MessageBox('Вы уверен что хотите удалить строку?', 'Внимание', MB\_ICONQUESTION + MB\_YESNOCANCEL) = IDYES then DataModule1.ADOTable1.Delete;

end;

procedure TForm2.Button1Click(Sender: TObject);

begin

DataModule1.AdoTable1.Insert;

dobavlenie\_knig.ShowModal

end;

procedure TForm2.Btn2\_RedakClick(Sender: TObject);

begin

DBGrid1.setFocus;

dobavlenie\_knig.ShowModal

end;

end.

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, DBCtrls, StdCtrls, Mask, jpeg, ExtCtrls;

type

TEditRecord = class(TForm)

Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

DBLookupComboBox2: TDBLookupComboBox;

DBEdit1: TDBEdit; Button1: TButton; DBEdit2: TDBEdit; Image1: TImage;

Button2: TButton; Label5: TLabel;

DBLookupComboBox3: TDBLookupComboBox;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

EditRecord: TEditRecord;

implementation

uses DataModuleUnit;

{$R \*.dfm}

procedure TEditRecord.Button1Click(Sender: TObject);

begin

if DataModule1.VzbTable.Modified then

DataModule1.VzbTable.Post;

Close;

end;

procedure TEditRecord.Button2Click(Sender: TObject);

begin

close;

end;

end.

unit Unit8;

interface

uses Windows, SysUtils, Messages, Classes, Graphics, Controls,

StdCtrls, ExtCtrls, Forms, QuickRpt, QRCtrls;

type

TQuickReport20 = class(TQuickRep)

ColumnHeaderBand1: TQRBand;

DetailBand1: TQRBand; PageFooterBand1: TQRBand;

PageHeaderBand1: TQRBand; QRLabel2: TQRLabel;

QRLabel12: TQRLabel; QRLabel5: TQRLabel;

QRLabel6: TQRLabel; QRLabel8: TQRLabel;

QRLabel9: TQRLabel; QRLabel10: TQRLabel;

QRLabel7: TQRLabel; QRShape1: TQRShape;

QRShape2: TQRShape; QRLabel1: TQRLabel;

QRDBText1: TQRDBText; QRLabel13: TQRLabel;

QRLabel14: TQRLabel; QRLabel11: TQRLabel;

QRLabel15: TQRLabel; QRSysData3: TQRSysData;

QRDBText4: TQRDBText; QRDBText5: TQRDBText;

QRDBText6: TQRDBText; QRDBText7: TQRDBText;

QRSysData2: TQRSysData; QRDBText3: TQRDBText;

QRLabel3: TQRLabel; QRLabel4: TQRLabel;

QRDBText2: TQRDBText; QRSysData1: TQRSysData;

private

public

end;

var

QuickReport20: TQuickReport20;

implementation

uses unit2;

{$R \*.DFM}

end.

unit Unit17;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ComCtrls, DBCtrls, StdCtrls;

type

TForm17 = class(TForm)

Label1: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

Label2: TLabel; Label3: TLabel; Label4: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

DateTimePicker10: TDateTimePicker;

DateTimePicker20: TDateTimePicker;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form17: TForm17;

Implementation

uses unit2, Unit6, Unit18;

var date1, date2:string;

{$R \*.dfm}

procedure TForm17.Button1Click(Sender: TObject);

begin

DataModule2.AdoQuery1.Active:=false;

DataModule2.AdoQuery1.SQL.Clear;

DataModule2.ADOQuery1.sql.text:='select \* from Application where (EquipName2 LIKE '''+ DBLookupComboBox2.Text+''') and (PerformName2 LIKE '''+ DBLookupComboBox1.Text+''') and (EquipDate between :Date1 AND :date2)';

DataModule2.ADOQuery1.Parameters.ParamByName('date1').Value:=datetostr(DateTimePicker10.Date);

DataModule2.ADOQuery1.Parameters.ParamByName('date2').Value:=datetostr(DateTimePicker20.Date);

DataModule2.ADOQuery1.Active:= true;

QuickReport18.QRLabel9.Caption := datetostr(DateTimePicker10.Date);

QuickReport18.QRLabel10.Caption := datetostr(DateTimePicker20.Date);

QuickReport18.Preview;

end;

end.