|  |
| --- |
| **Министерство сельского хозяйства и продовольствия Республики Беларусь**  **УО «Столинский государственный аграрно-экономический колледж»**  **Отделение «Бухгалтерский учет и информационные технологии»**  ДОПУЩЕН К ЗАЩИТЕ  Заведующий учебно-производственной практикой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.С. Ярмошик  « 19 » декабря 2022г.  **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  **ПРОЕКТА**  « Обработка результатов экзамена »  (Тема проекта)  Специальность  2-40 01 01 «Программное обеспечение информационных технологий»  Специализация  2-40 01 01 35 «Программное обеспечение обработки экономической и деловой информации»  Учащийся  группа П-16 Пачко Н.Н.  (подпись) (Ф.И.О.)  Руководитель Пешко В.П.  (подпись) (Ф.И.О.)  Отметка  Столин 2022 |

|  |
| --- |
| Министерство сельского хозяйства и продовольствия Республики Беларусь  УО «Столинский государственный аграрно-экономический колледж  **ЗАДАНИЕ**  на выполнение проекта учащемуся  *Пачко Николай Николаевич*  (Фамилия, Имя, Отчество)   1. Тема работы: *Обработка результатов экзамена* 2. Исходные данные к проекту 3. Состав проекта    1. Содержание расчётно-пояснительной записки (перечень подлежащих разработке вопросов)          * 1. Перечень графического материала (с точным указанием обязательных чертежей)  1. Дата выдачи задания \_\_\_\_\_\_\_\_\_\_\_20\_\_г. 2. Срок сдачи учащимся законченной работы \_\_\_\_\_\_\_\_\_\_\_20\_\_г.   Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ф.И.О.  (подпись)  Задание принял к исполнению  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_20\_\_г. \_\_\_\_\_\_\_\_\_\_\_\_\_ *Ф.И.О.* |

|  |
| --- |
| **РЕФЕРАТ**  Пояснительная записка проекта: 32 с, 17 рис., 1 табл., 0 источник, 0 прил.  Объектом разработки является ПП для обработки результатов экзамена.  Цель проекта - автоматизация обработки информации и проверки сдачи экзаменов на высшем уровне.  В процессе проектирования выполнены следующие разработки создавались диаграммы разного вида, соответствующие теме проекта, проектировалась база данных, проводилось тестирование и отладка данного прототипа.  Областью возможного практического применения являются учебные заведения.  Область применения: созданный ПП применяется в учебных заведениях. |

**Перечень условных обозначений**

ПО – программное обеспечение

ОС – операционная система

ОЗУ – оперативное запоминающее устройство

БИС – большая интегральная схема

ИС – интегральная схема

СБИС – сверхбольшая интегральная схема

ПЛМ – программируемая логическая матрица

ГСП – генератор случайных последовательностей

ГПП – генератор псевдослучайных последовательностей

СЛН – средства локализации неисправностей

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc91447497)

[1. Анализ исходных данных и постановка цели и задач 7](#_Toc91447498)

[**1.Общие сведения** 8](#_Toc91447499)

[1.1 Формулировка задания 8](#_Toc91447500)

[1.2 Цели, достигаемые разработкой 8](#_Toc91447501)

[1.3 Категории пользователей 8](#_Toc91447502)

[1.4 Наименование организации-заказчика 8](#_Toc91447503)

[1.5 Основание для проведения работ 8](#_Toc91447504)

[**2. Описание предметной области** 8](#_Toc91447505)

[2.1 Описание процессов 8](#_Toc91447506)

[**3.Требования к разработке** 9](#_Toc91447507)

[3.1 Информационная модель 9](#_Toc91447508)

[**4.** **Порядок контроля и обеспечения качества** 12](#_Toc91447509)

[4.1 Экспертиза 12](#_Toc91447510)

[5. **Требования к документированию** 13](#_Toc91447511)

[*3. Функциональная модель ПС* 14](#_Toc91447512)

[*5. Концептуальная модель UML* 16](#_Toc91447513)

[*6.Проектирование интерфейса* 22](#_Toc91447514)

[*7.Кодирование модулей системы* 24](#_Toc91447515)

[*8.Реализация и тестирование* 31](#_Toc91447516)

# Введение

Информационно-справочная система разработана как набор программ, каждая из которых реализует заданное множество операций, где под операцией понимается некоторое действие над информацией, хранящейся в базе данных, инициированное пользователем системы. Все программы осуществляют диалоговое взаимодействие с пользователями.

Для автоматизации системы ПП была выбрана СУБДMSAccess, которая является лидером среди инструментов для создания приложений и систем, функционирующих на ОС Windows.

Цели создания информационно-поисковой системы – автоматизация обработки информации и проверки сдачи экзаменов на высшем уровне. Данная система позволила решать следующие задачи:

1. Навигация по таблицам.

2. Добавление записей.

3. Редактирование записей.

4. Удаление записей.

5. Подсчёт среднего балла.

Благодаря тому, что программа разаработана при помощи BorlandDelphiи MicrosoftAccess, она имеет понятный интерфейс, который прост и удобен для пользователя.

# Анализ исходных данных и постановка цели и задач

В данном случае постановщиком задачи и пользователем программы является преподаватель.

Основанием для разработки программы является задание, выданное учащемуся по разработке и сопровождению программного обеспечения.

Задача – написание качественного, отлаженного и работоспособного продукта для автоматизации проверки оценок.

Данная программа написана на языке программировании ObjectPascal в среде программирования Delphi7.

Данная система должна обеспечивать выполнение следующих функций. Предоставление пользователю возможность добавлять, изменять и удалять данную информацию в базе данных, делать подсчёт среднего бала.

Данные операции были автоматизированы для быстрого и более простого учета.

2. Техническое задание

**1.Общие сведения**

* 1. Формулировка задания

Разработать программный продукт (ПП), предназначенный для подсчёта среднего бала всех абитуриентов по предметной области. ПП должен обеспечивать ведение базы данных, учет поступления абитуриентов, учет отметок абитуриентов.

* 1. Цели, достигаемые разработкой

Автоматизация обработки информации, занимающегося продвижением данного ПП.

* 1. Категории пользователей

ПП должен быть рассчитан для работников учебных заведений.

* 1. Наименование организации-заказчика

Организация- ЗАО «Kaspersky Lab»

Город - Столин

Ф.И.О заказчика – Мозоль Павел Васильевич

Телефон заказчика- +375(12)345-67-89

Адрес электронной почты- mozolpavel@gmail.com

Ф.И.О генерального директора – Пачко Николай Николаевич

Телефон директора- +375(12)345-67-89

Адрес электронной почты- pachko2005@gmail.com

* 1. Основание для проведения работ

Основаниями для проведений работ данного ПП является договор между заказчиком (УО «СГАЭК») и разработчиком (ЗАО «Kaspersky Lab») от 20.10.2022 г.

**2. Описание предметной области**

2.1 Описание процессов

ПП выполняет следующие функции: добавление и изменение абитуриента и его отметки, подсчёт среднего бала, работа с базой.

С помощью разрабатываемого ПП должен осуществляться учет и обработка, как первичных документов: договоров с поставщиками, договоров с покупателями.

Доступ к информации имеет администратор после успешного завершения процедуры авторизации.

ПП должен осуществлять ведение БД.

По запросу пользователя может быть осуществлена настройка интерфейса каждого учебного предмета.

2.2 Состав данных и алгоритмы обработки информации

Входными данными для разрабатываемого ПП являются:

* пароль администратора (приложение Е);
* параметры настройки, перечисленные в разделе «Функциональные требования»;
* данные, вводимые пользователем, в таблицы БД;
* критерии поиска, фильтрации; – тип формируемого документа;
* параметры настройки интерфейса.

Выходными данными разрабатываемого ПП являются:

* информация, отображаемая на экран по запросу пользователя;
* документы, создаваемые разработанным ПП (структура документации приведена в приложениях А-Д).

Постоянными данными являются: – шаблоны документов, – БД.

Разрабатываемый ПП должен создавать и модифицировать перечисленные ниже БД.

БД проверки экзаменов, содержащая ожну таблицы:

* таблица поступления экзаменов: возраст учащегося, ФИО учащегося;
* таблица проверки экзаменов: ФИО учащегося, дата сдачи экзамена, оценка по окончанию проверки экзамена.

* 1. Недостатки существующих проектных решений

На данный момент не существует аналогов похожих на данный ПП.

* 1. Текущий уровень автоматизации

На данный момент Заказчик не располагает каким-либо ПП, обеспечивающим полноценную автоматизацию всей его деятельности. Заказчик располагает оборудованием:

* 5 компьютеров Р-III-500 (256RAM RIMM), Unix Free BSD, Star office; – 4 компьютера П-133 (16RAM 4VRAM), Windows 98, Microsoft office

97.

**3.Требования к разработке**

3.1 Информационная модель

* 1. Информационная модель

Информационная модель представлена на рисунке 3.1.

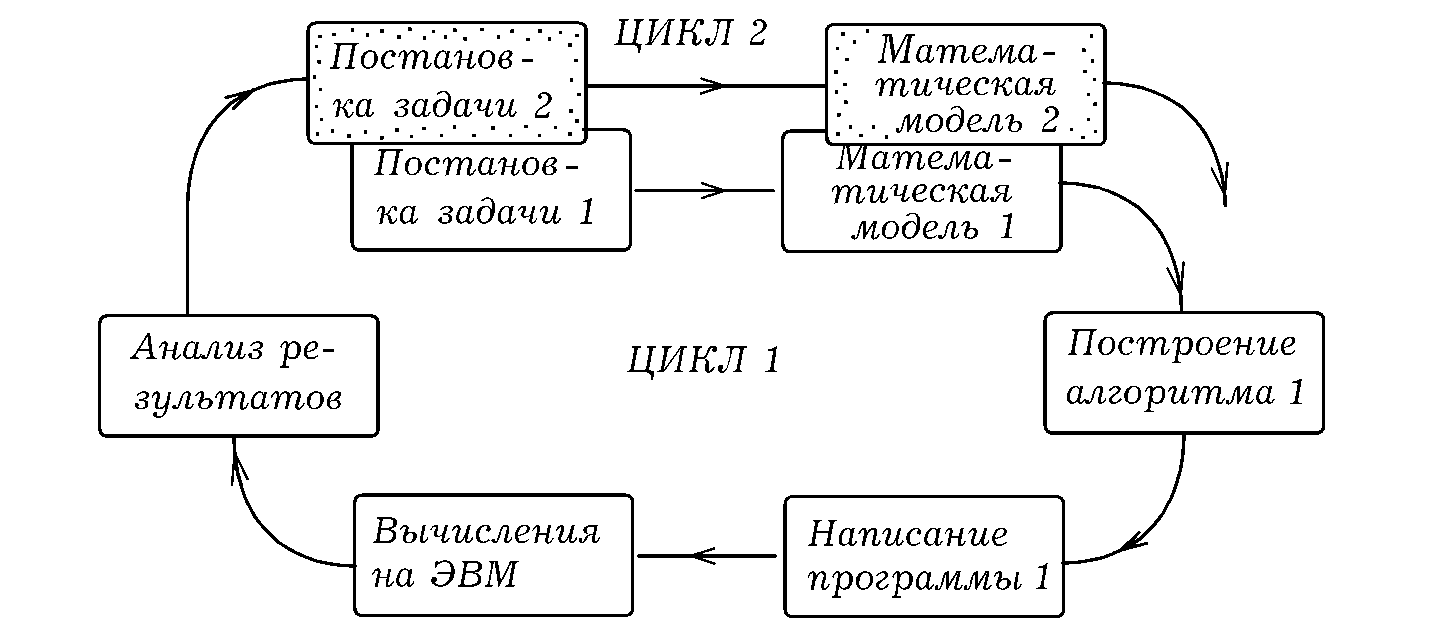


Рисунок 3.1 – Информационная модель проекта

* 1. Структура меню

Сайт в продаже:

Вывести список

Внесение изменений:

Сайт отключён:

Вывести список

Внесение изменений:

Документы

Просмотр созданных

Создать новый

Акт на списание

Накладная на возврат

Договор

Гарантийный талон

Внесение изменений

Поисковая система

Поиск подходящего экзамена

Расчет скидки

Сохранить

Help

3.3 Функциональные требования

Разрабатываемое ПО должно реализовать следующие функции: 1. 1. Ведение базы данных по продаже обуви.

Функция предназначена для:

* ведение учета поступления учащихся;
* ведение учета проверки экзаменов;

2. Контроль сроков выполнения работ.

3. Ведение БД учащихся.

4. Авторизация. Предназначена для контроля допуска к информации.

5. Поиск, фильтрация. Формирует списки и осуществляет поиск по заданным критериям.

6. Расчет скидки для постоянных клиентов.

7. Формирование документации.

3.4 Требования к информационному обеспечению

Разрабатываемый ПП должен:

* сохранять данные, вводимые пользователем;
* формировать и сохранять отчетные документы;
* содержать необходимые классификаторы и таблицы;
* обеспечивать обмен данными с другими ПП по локальной сети;
* иметь изменяемые настройки поиска, классификации данных, сопровождаемые кратким описанием;
* иметь изменяемые настройки обрабатываемых объектов (склады, филиалы).

3.5 Требования к пользовательскому интерфейсу

Разрабатываемый ПП должен быть обеспечен упрощенной формой интерфейса:

* глубина вложенности окон должна быть не более трех; – содержать краткие описания элементов меню; – должен быть понятным и простым в использовании.

Панель интерфейса должна содержать кнопку, предназначенную для сохранения необходимой информации

Интерфейс должен содержать в себе все необходимые функции для работы Заказчика. Каждая функция оснащается всплывающей подсказкой и кратким описанием, что будет способствовать быстрому освоению разрабатываемого ПП.

* При необходимости интерфейс изменяется только Разработчиком. глубина вложенности окон должна быть не более трех;
* содержать краткие описания элементов меню;
* должен быть понятным и простым в использовании.

Панель интерфейса должна содержать кнопку, предназначенную для сохранения необходимой информации.

Интерфейс должен содержать в себе все необходимые функции для работы Заказчика. Каждая функция оснащается всплывающей подсказкой и кратким описанием, что будет способствовать быстрому освоению разрабатываемого ПП.

При необходимости интерфейс изменяется только Разработчиком.

* 1. Требования к алгоритмам.

Разрабатываемые алгоритмы должны легко модифицироваться. В течение сроков, установленных в договоре между Заказчиком и Разработчиком в ПП могут быть внесены изменения, не требующие глобальной переработки алгоритмов.

* 1. Прочие требования

Разрабатываемый ПП должен функционировать в ОС Win 9x, NT, 2K.

Другие ОС не поддерживают формат создаваемых файлов в процессе работы с ПП.

Хранение данных пользователя должно производиться в файлах с встроенной защитой, не подлежащих удалению без пароля Администратора (приложение Е).

ПП должен использовать нижние регистры памяти, что обеспечивает защиту от сбоев на 95 %.

Каждая созданная таблица должна записываться на HDD в виде файла, открываемого любым текстовым редактором без права редактирования, что обеспечивает удобство переноса данных и возможность максимально быстрого просмотра данных.

ПП должен обеспечивать сетевой обмен информацией с другими БД. При создании БД происходит резервное сохранение информации, вводимой пользователем в файлы с расширением «pole». Сетевой обмен происходит за счет передачи/ получения файлов с таким же расширением от других систем, для этого необходимо всю нужную информацию другой БД сохранить в файл с расширением «txt» (текстовый), затем переименовать текстовый файл с расширением «pole». После этого полученный файл поместить в директорию:

c:\Program Files\AutoServAutomatisation\main\bd\info\pole.

1. **Порядок контроля и обеспечения качества** 
   1. Экспертиза

Не требуется разработка макета на стадии «Технический проект». Требуется проведение экспертизы при участии представителя Заказчика.

* 1. Тестирование

Тестирование должно проводиться Разработчиком в соответствии с установленными правилами. Присутствие представителя Заказчика необязательно.

Тестирование проводится Разработчиком по предложенной схеме Заказчика:

1. Создается БД небольшого объема по всем таблицам и пунктам меню;
2. В течение срока, указанного в договоре, Разработчик проводит тестирование ПП и передает Заказчику готовый программный продукт, который будет сопровождаться Разработчиком в течении сроков, предусмотренных договором.

4.3 Опытная эксплуатация

Опытная эксплуатация проводится Заказчиком, при контроле Разработчика, в соответствии с договором.

1. **Требования к документированию** 
   1. Требования к справочной система

Справочная система должна содержать описание всех команд, использующихся в ПП. Справочная система должна быть встроенной в интерфейс и располагаться непосредственно в окнах.

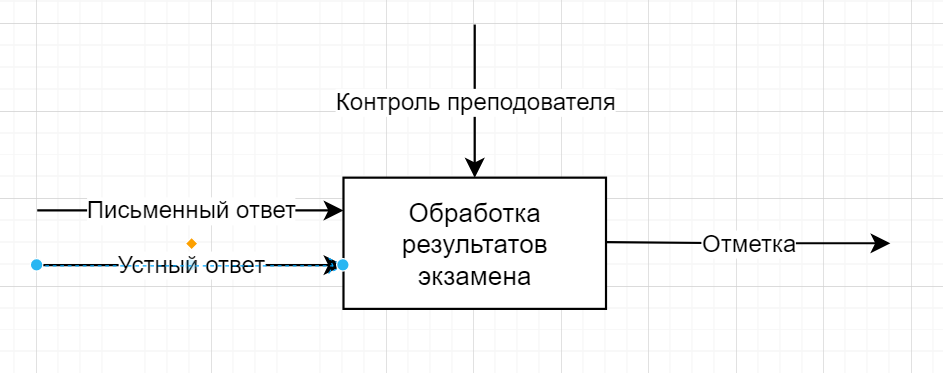
* 1. Требования к документации пользователя

Разрабатываемый ПП должен сопровождаться полным пакетом конструкторской и эксплуатационной документации, перечисленной в договоре.

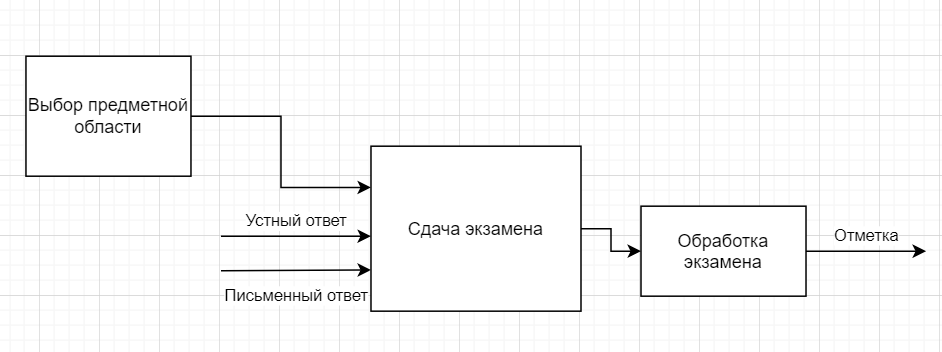
Разработанный ПП сопровождается полным пакетом документов, предусмотренным ГОСТ 7685 – 2000, а именно: лицензионное право использования разработанного ПП; документация по эксплуатации; талон на последующее обслуживание ПП; документы, подтверждающие монопольное использование разработанного ПП.

# 3. Функциональная модель ПС

**1.Контекстная диаграмма**

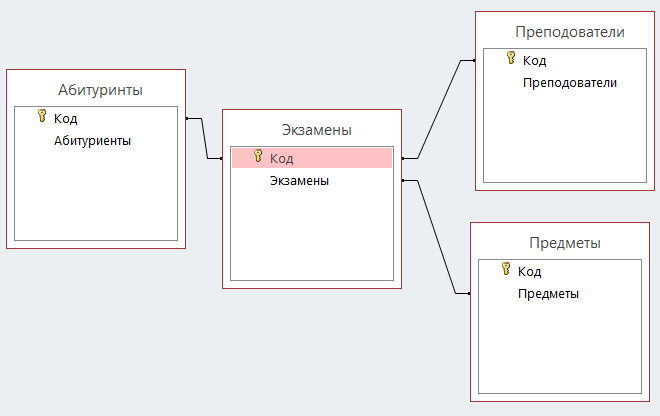


**2.Декомпозиция функциональной диаграммы**



4. Информационная БД проекта

**1.Концептуальная модель БД.**



**2. База данных на основе концептуальной модели.**

Таблица 1 - (Абитуриенты)

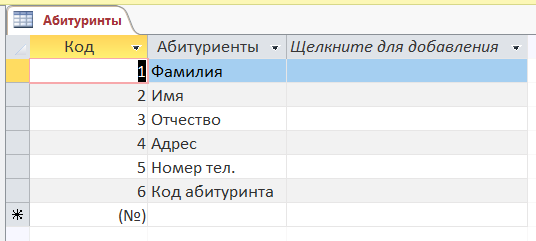
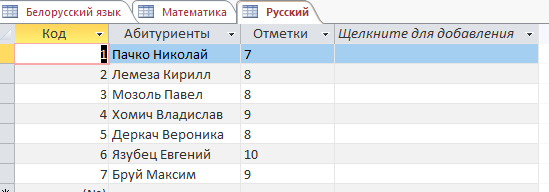


Таблица 2 - (Предметы)



**3. Создание модели БД в виде таблиц.**

Таблица 1 — Абитуриенты

|  |  |  |
| --- | --- | --- |
| Поле | Тип поля | Ключевое поле |
| Код | Счётчик | **+** |
| Абитуриенты | Короткий текст |  |

Таблица 2 — Предметы

|  |  |  |
| --- | --- | --- |
| Поле | Тип поля | Ключевое поле |
| Код | Счётчик |  |
| Предметы | Короткий текст | **+** |

# 5. Концептуальная модель UML

***1.Диаграмма вариантов использования***

**Диаграмма вариантов использования(рис-1)**

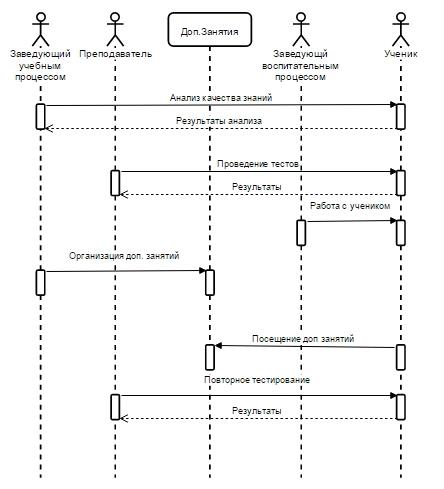


**Рисунок-1**

На рисунке 1 изображена диаграмма вариантов использования, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

***2.Диаграмма последовательности***

**Диаграмма вариантов последовательности(рис-2)**



**Рисунок-2**

На рисунке 2 изображена диаграмма вариантов последовательности, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

***.Диаграмма деятельности***

**Диаграммы деятельности(рис-1)**

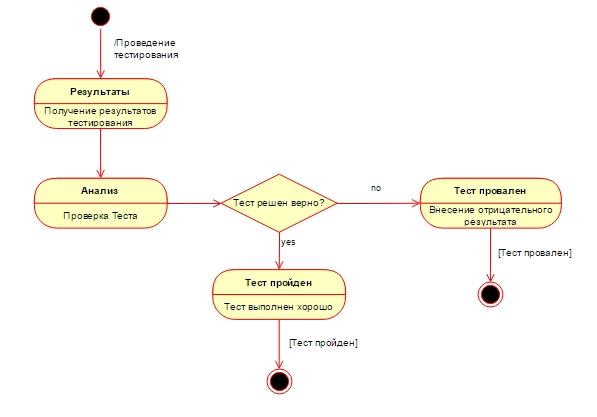


Рисунок-1

На рисунке-1 изображена диаграмма деятельности, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

***4.Диаграмма состояния***

**Диаграмма состояний(рис-2)**

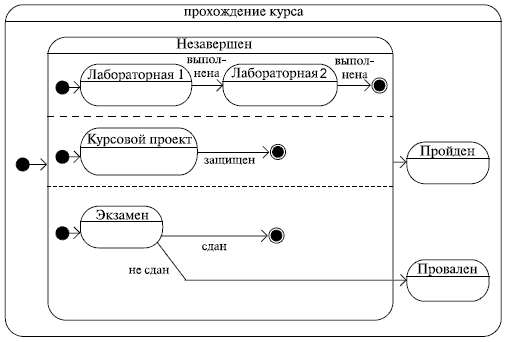


Рисунок-2

На рисунке-2 изображена диаграмма состояний, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

***5.Диаграмма классов***

**Диаграмма классов(рис-1)**

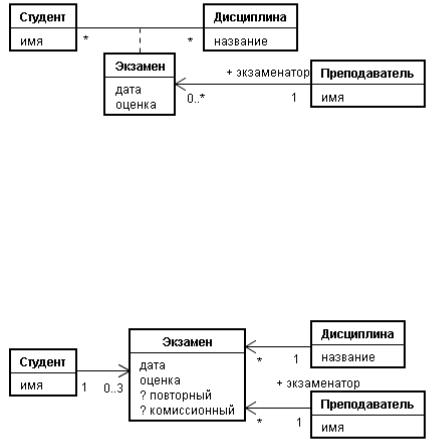


Рисунок-1

На рисунке-1 изображена диаграмма классов, выполненная в Rational Rose. В диаграмме отображена работа программного продукта. Она определяет типы классов системы и различного рода статические связи, которые существуют между ними.

На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.

***6.Диаграмма объектов***

**Диаграмма объектов(рис-2)**

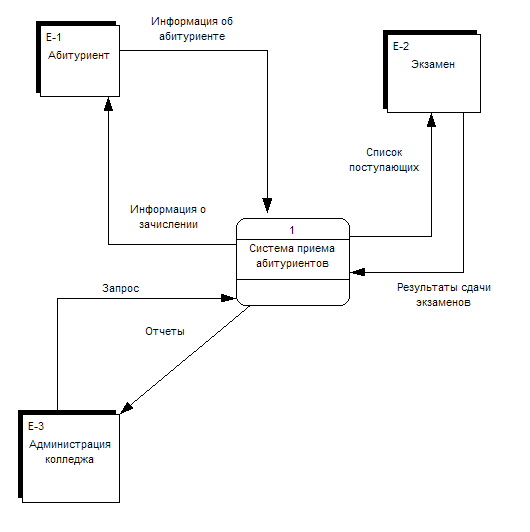


Рисунок-2

На рисунке-2 изображена диаграмма объектов, выполненная в Rational Rose. В диаграмме отображена работа программного продукта. Диаграммы объектов являются производными от диаграмм классов, поэтому диаграммы объектов зависят от диаграмм классов.

Диаграммы объектов представляют собой экземпляр диаграммы классов. Основные понятия одинаковы для диаграмм классов и диаграмм объектов. Диаграммы объектов также представляют статическое представление системы, но это статическое представление представляет собой снимок системы в определенный момент.

Диаграммы объектов используются для визуализации набора объектов и их отношений в качестве экземпляра.

***7.Диаграмма компонентов***

**Диаграмма компонентов(рис-1)**

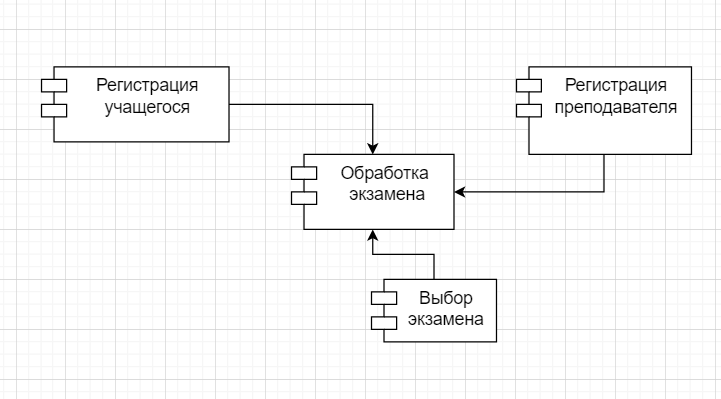


Рисунок-1

На рисунке 1 изображена диаграмма компонентов, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу.

Интерфейсы на компонентных схемах показывают, как компоненты соединены друг с другом и взаимодействуют друг с другом. Соединитель сборки позволяет соединить требуемый интерфейс компонента с предусмотренным интерфейсом другого компонента.

***8.Диаграмма развёртывания***

**Диаграмма развёртывания(рис-2)**

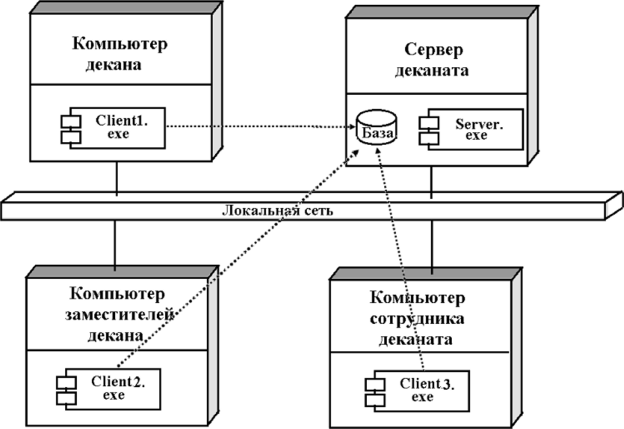


Рисунок-2

На рисунке 2 изображена диаграмма развёртывания, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

Диаграмма развертывания – это тип UML-диаграммы, которая показывает архитектуру исполнения системы, включая такие узлы, как аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их.

Диаграммы развертывания помогают моделировать аппаратную топологию системы по сравнению с другими типами UML-диаграмм, которые в основном описывают логические компоненты системы.

На данной диаграмме есть коммуникационная ассоциация, так же компоненты и узлы, и зависимость.

Изображено развёртывания компании, такие как сервер и поставщик, описание товара и физического лица.

# 6.Проектирование интерфейса

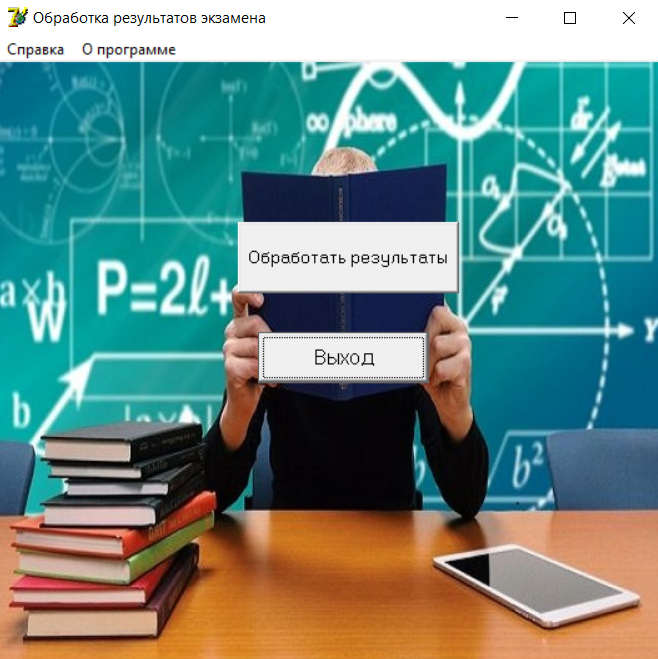
Разработал модель пользовательского интерфейса для своего варианта задания.

**Форма авторизация (рис.1)**



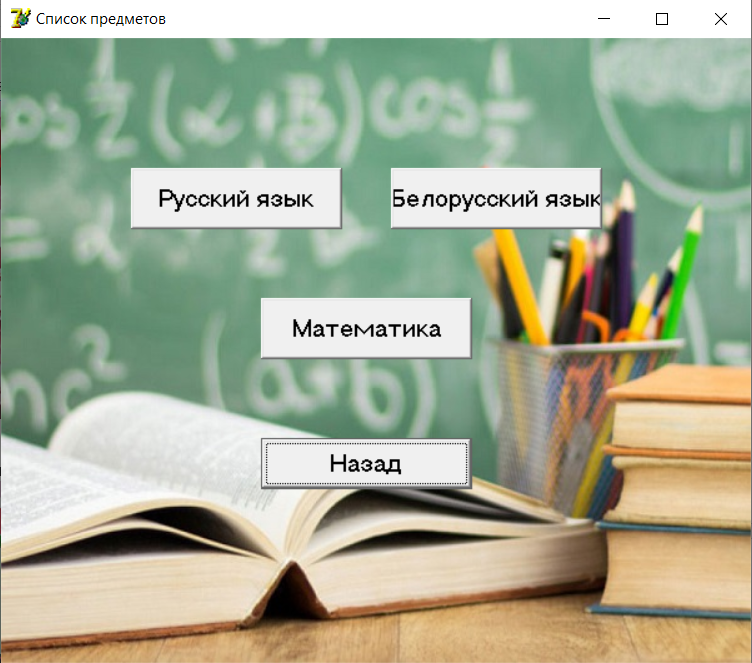
**Рисунок 1. Авторизация**

**Главная форма(рис.2)**

******

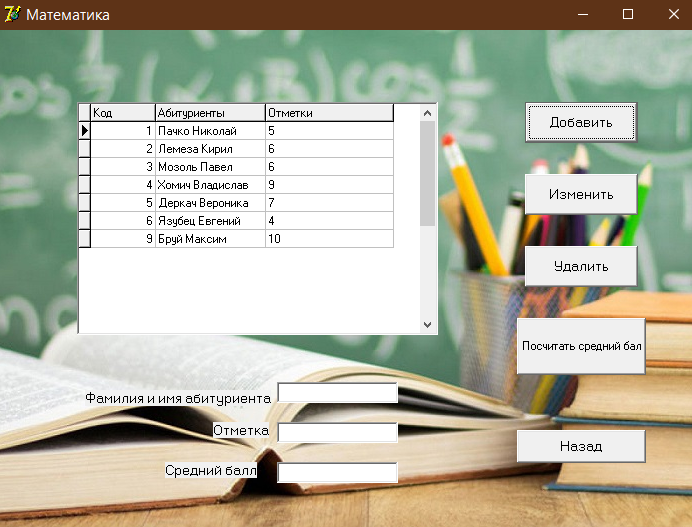
**Рисунок 2. Главное окно программы**

**Форма список предметов(рис.3)**

****

**Рисунок 3. Список предметов**

**Форма Математика(рис.4)**



**Рисунок 4. *Математика***

# 7.Кодирование модулей системы

**Листинг программных модулей**

**Листинг 1 – Модуль Unit7.pas**

unit Unit7;

interface

uses

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

Dialogs, StdCtrls;

type

TForm7 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Label2: TLabel;

Edit2: TEdit;

Label3: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form7: TForm7;

implementation

uses Unit1, Unit6;

{$R \*.dfm}

procedure TForm7.Button1Click(Sender: TObject);

var

a,b:string;

begin

if (edit1.Text = 'user') and (edit2.Text = '1234') then

begin

form1.show;

Form6.Hide;

end

else

label3.Caption:='Авторизация не удалась';

end;

end.

**Листинг 2 – Модуль Unit1.pas**

Модуль главного окна программы

unit Unit1;

interface

uses

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

Dialogs, StdCtrls, jpeg, ExtCtrls, Menus, ShellApi;

type

TForm1 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2, Unit6;

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

Form1.Close

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

Form1.Visible:=false;

Form2.Visible:=true;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

Form1.Visible:=false;

Form6.Visible:=true;

end;

procedure TForm1.N1Click(Sender: TObject);

begin

ShellExecute(0,Pchar('Open'),Pchar('spravka.chm'),nil,nil,SW\_SHOW);

end;

end.

**Листинг 3 – Модуль Unit2.pas**

Модуль формы «обработка результатов экзамена»

unit Unit2;

interface

uses

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

Dialogs, jpeg, ExtCtrls, StdCtrls;

type

TForm2 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1, Unit3, Unit5, Unit4;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject);

begin

Form2.Visible:=false;

Form1.Showmodal;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

Form3.Showmodal;

end;

procedure TForm2.Button4Click(Sender: TObject);

begin

Form5.Showmodal;

end;

procedure TForm2.Button3Click(Sender: TObject);

begin

Form4.Showmodal;

end;

end.

**Листинг 4 – Модуль Unit3.pas**

Модуль формы «Математика»

unit Unit3;

interface

uses

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

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

type

TForm3 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

ADOConnection1: TADOConnection;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

Edit3: TEdit;

ADOQuery1: TADOQuery;

Label3: TLabel;

procedure Button5Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses Unit2;

{$R \*.dfm}

procedure TForm3.Button5Click(Sender: TObject);

begin

Form3.Close;

end;

procedure TForm3.Button4Click(Sender: TObject);

var

k,sr:real;

i:integer;

begin

k:=0;

for i:=0 to ADOQuery1.RecordCount-1 do

begin

k:=k+ADOQuery1.FieldValues['Отметки'];

ADOQuery1.Next;

end;

sr:=(k/ADOQuery1.RecordCount);

Edit3.Text:=FloatToStr(sr);

end;

procedure TForm3.Button1Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') or (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Insert;

ADOQuery1['Абитуриенты']:=Edit1.Text;

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm3.Button2Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') and (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Edit;

if Edit1.Text<>'' then

ADOQuery1['Абитуриенты']:=Edit1.Text;

if Edit2.Text<>'' then

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

ADOQuery1.delete;

end;

end.

# 8.Реализация и тестирование

Тестирование своего ПС.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Выполненные действия | Шаги | Ожидаемый результат |
| 1 | Выбор обработки результатов | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты» | 1. Программа запуститься  2. Открывается новое окно |
| 2 | Просмотр справки | 1. Запустить программу  2. Нажать на кнопку «Справка» | 1. Программа запуститься  2. Произойдёт открытие справки в новом окне |
| 3 | Просмотр «О программе» | 1. Запустить программу  2. Нажать на кнопку «О программе» | 1. Программа запуститься  2. Произойдёт открытие окно «О программе» в новом окне |
| 4 | Просмотр списка предметов | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты»  3. Выбрать нужный предмет | 1. Программа запуститься  2. Откроется окно «Список предметов»  3. Выбрать нужный предмет |
| 5 | Обработка результатов для русского языка | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты»  3. Выбрать предмет «Русский язык» | 1. Программа запуститься  2. Откроется окно «Список предметов»  3. Окно «Русский язык» открыто |
| 6 | Просмотр журнала учащихся | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты»  3. Выбрать предмет «Русский язык» | 1. Программа запуститься  2. Откроется окно «Список предметов»  3. Окно «Русский язык» открыто |
| 7 | Добавление абитуриента | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты»  3. Выбрать предмет «Русский язык»  4. Заполнить поля  5. Нажать на кнопку «Добавить» | 1. Программа запуститься  2. Откроется окно «Список предметов»  3. Окно «Русский язык» открыто  4. Поля заполнены  5. Абитуриент добавлен |
| 8 | Удаление абитуриента | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты»  3. Выбрать предмет «Русский язык»  4. Установить курсор на нужного абитуриента  5. Нажать на кнопку «Удалить» | 1. Программа запуститься  2. Откроется окно «Список предметов»  3. Окно «Русский язык» открыто  4. Абитуриент удален |
| 9 | Изменение записи | 1. Запустить программу  2. Нажать на кнопку «Обработать результаты»  3. Выбрать предмет «Русский язык»  4. Установить курсор на нужного абитуриента  5. Заполнить поля  6. Нажать на кнопку «Добавить» | 1. Программа запуститься  2. Откроется окно «Список предметов»  3. Окно «Русский язык» открыто  4. Поля заполнены  5. Абитуриент добавлен |
| 10 | Выход из программы | 1. Запустить программу  2. Нажать на кнопку «Выход» | 1. Программа запуститься  2. Произойдёт выход из программы |