Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
|  |

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информационных систем управления**

Котик Софья Валентиновна

ПРОЕКТИРОВНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ЗАПИСЬ НА УНИВЕРСИТЕТСКИЕ КУРСЫ

**КУРСОВАЯ РАБОТА**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  |  | Студентка гр. Б8219 | | | |  |
|  | | | | |  |  |  | | | | (подпись) |
|  | | | | |  |  | Руководитель ст. преподаватель | | | | |
|  | | | | |  |  |  | | | | Г. Л. Березкина |
|  | | | | |  |  |  | | | |  |
| Регистрационный № | | |  | |  |  | Оценка |  | | | |
|  | |  | |  |  |  |  | | | |  |
| подпись | |  | | И. О. Фамилия | |  |  | |  | |  |
| «\_\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | 2016 г. |  | подпись | | |  | И. О. Фамилия |
|  | |  | |  |  |  | «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г. | | | | |

г. Владивосток

2016

**Оглавление**

[Введение 3](#_Toc470269927)

[1 Анализ 4](#_Toc470269928)

[Диаграмма вариантов использования 4](#_Toc470269929)

[Диаграмма классов 5](#_Toc470269930)

[Словарь предметной области 5](#_Toc470269931)

[2 Проектирование 7](#_Toc470269932)

[Сценарии вариантов использования 7](#_Toc470269933)

[Диаграммы последовательностей 14](#_Toc470269934)

[3 Реализация 18](#_Toc470269935)

[4 Интерфейс 19](#_Toc470269936)

[Формы интерфейса 19](#_Toc470269937)

[Описание структур папок проекта 23](#_Toc470269938)

[5 Обобщённое описание работы программы 26](#_Toc470269939)

[Заключение 27](#_Toc470269940)

[Список литературы 28](#_Toc470269941)

[Приложение 1 30](#_Toc470269942)

[Приложение 2 35](#_Toc470269943)

# **Введение**

Необходимо разработать информационную систему для записи на университетские курсы.

Название проекта “Entry to University courses”. Данная система будет иметь простой и доступный консольный интерфейс.

Прежде чем переходить непосредственно к программированию необходимо построить несколько видов UML диаграмм, которые наглядно покажут необходимый функционал и взаимодействие между классами. В данном случае это будут:

* Диаграмма вариантов использования;
* Диаграмма классов;
* Сценарии;
* Диаграмма последовательностей.

# **1 Анализ**

В процессе анализа было решено рассмотреть диаграмму вариантов использования (Рисунок 1), словарь предметной области и диаграмму классов (Рисунок 2), данные элементы позволяют лучше понять структуру ИС и перейти к проектированию.

**Диаграмма вариантов использования**

Диаграмма вариантов использования (рисунок 1) показывает, как актёры взаимодействуют с системой, например, администратор напрямую работает с системой, а абитуриенты и преподаватели взаимодействуют только с администратором, сами действия записаны в эллипсах. Линии связи обозначены – соответственно линиями.



Рисунок 1 – Диаграмма вариантов использования

**Диаграмма классов**

Ниже представлена диаграмма классов (Рисунок 2), каждому прямоугольнику соответствует класс, классы интерфейсы обозначены ключевым словом <<Интерфейс>>. Стрелочки между классами показывают, как именно взаимодействуют классы между собой. Для создания этой диаграммы было удобно написать список всех существительных, которые могут фигурировать в ИС, а потом уже из них выбрать наиболее подходящие на роль классов.

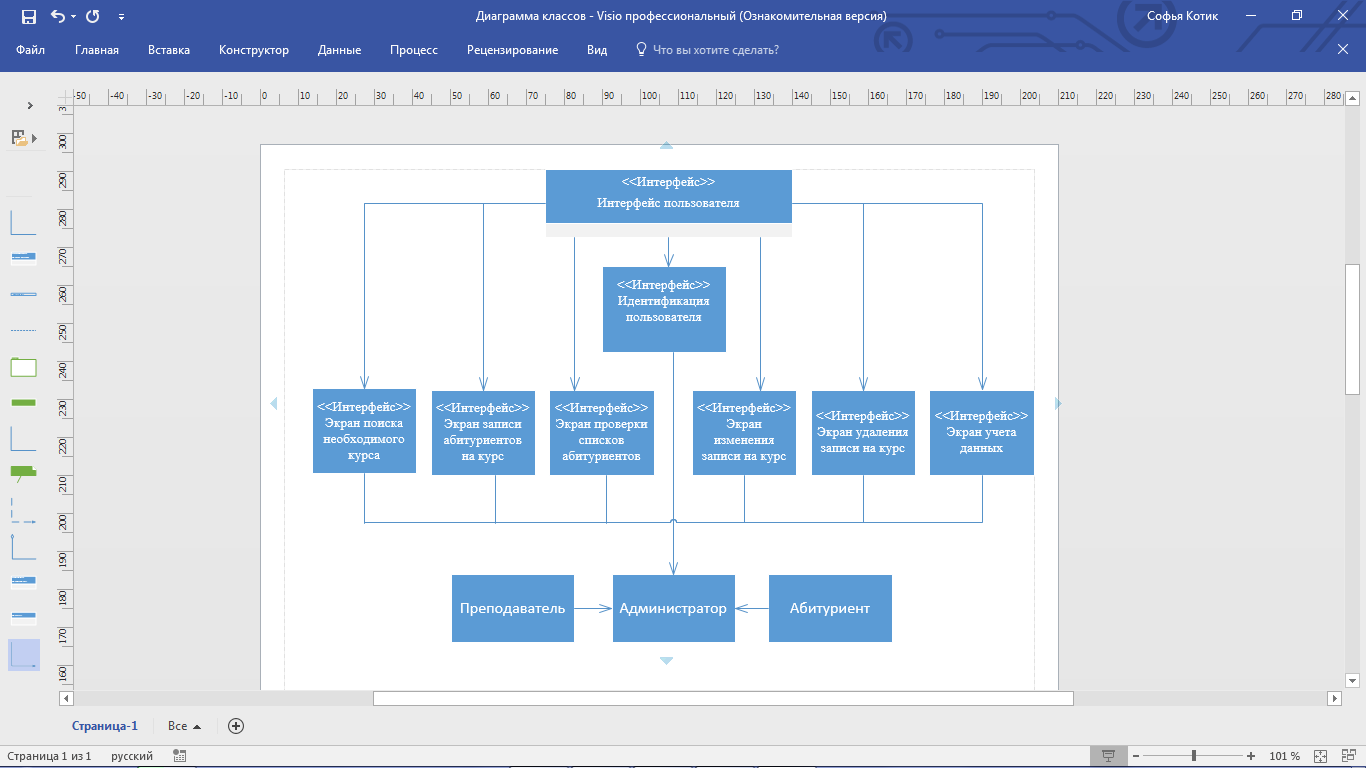


Рисунок 2 – Диаграмма классов

**Словарь предметной области**

1. Экран пользовательского интерфейса;
2. Администратор;
3. Абитуриент;
4. Преподаватель;
5. Идентификация пользователя;
6. Экран поиска необходимого курса;
7. Экран записи абитуриентов на курс;
8. Экран проверки списков абитуриентов;
9. Экран изменения записи на курс;
10. Экран удаления записи;
11. Фамилия абитуриента;
12. Имя абитуриента;
13. Аббревиатуры курсов;
14. Список курсов;
15. Список абитуриентов;
16. Учет данных;
17. Количество человек;
18. Дата проведения.

**2 Проектирование**

Начиная проектирование необходимо построить сценарии для каждого варианта использования. Стандартно – это две таблицы. В первой описывается сам сценарий, а во второй – ход событий в этом сценарии, если во время хода событий что-то может пойти не так, то создается третья таблица – таблица исключений. Сценарии подробно показывают, как выполняются варианты использования. Все сценарии ИС представлены ниже, за сценариями следуют диаграммы последовательностей. Диаграмма последовательностей связывают классы и действия, то есть показывается как именно выполняется тот или иной сценарий и связь между класса в процессе выполнения.

**Сценарии вариантов использования**

Сценарию «Войти в программу» соответствуют таблицы 1.1 – 1.2.

Т а б л и ц а 1.1 – Главный раздел сценария выполнения варианта использования «Войти в программу»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актер | Администратор |
| Краткое описание | Администратор запускает программу и попадает в главное меню |
| Цель | Получить доступ к остальным функциям системы |
| Тип | Базовый |
| Ссылки на другие варианты использования | Найти нужный курс, запись абитуриента на курс, проверка списка абитуриентов, составить расписание, изменить запись на курс, удалить запись на курс, учет данных (курс-дата проведения-количество человек) |

Т а б л и ц а 1.2 – Раздел типичный ход событий варианта использования «Войти в программу»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер запускает файл .exe | 2. Система выводит главное меню, из которого можно вызвать весь функционал информационной системы |

Сценарию «Поиск необходимого курса» соответствуют таблицы 2.1 – 2.2.

Т а б л и ц а 2.1 – Главный раздел сценария выполнения варианта использования «Поиск необходимого курс»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актеры | Администратор |
| Краткое описание | Программа осуществляет поиск заданного курса в предложенном списке данной информационной системы и вывод соответствующее сообщение |
| Цель | Поиск необходимого курса |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 2.2 – Раздел типичный ход событий варианта использования «Поиск необходимого курс»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Поиск необходимого курс»  3. Актеры вводят в подписанные поля наименование курса | 2. Система открывает окно ввода данных для поиска курса  4. Система соответствующее сообщение |

Сценарию «Запись абитуриента на курс» соответствуют таблицы 3.1 – 3.2.

Т а б л и ц а 3.1 – Главный раздел сценария выполнения варианта использования «Запись абитуриента на курс»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актеры | Администратор |
| Краткое описание | Администратор производит соответствующую запись абитуриента на нужный ему курс |
| Цель | Запись абитуриента на курс |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 3.2 – Раздел типичный ход событий варианта использования «Запись абитуриента на курс»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Записаться на курс»  3. Актер заносит соответствующие данные | 2. Система запрашивает данные абитуриента, для которого производится запись на курс  4. Система сохраняет данные в файле |

Сценарию «Проверка списка абитуриентов» соответствуют таблицы 4.1 – 4.3.

Т а б л и ц а 4.1 – Главный раздел сценария выполнения варианта использования «Проверка списка абитуриентов»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актеры | Администратор |
| Краткое описание | Администратор получает доступ к списку абитуриентов нужного курса |
| Цель | Проверка списка абитуриентов |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 4.2 – Раздел типичный ход событий варианта использования «Проверка списка абитуриентов»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Список абитуриентов по курсам»  3. Актер заносит наименование курса | 2. Система открывает окна ввода данных для поиска по курсам  4. Система вывод нужную информацию |

Т а б л и ц а 4.3 – Раздел исключений сценария варианта события «Проверка списка абитуриентов»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| \*\*\*\*\*  3. Актер вводит неверное наименование курса  \*\*\*\*\* | \*\*\*\*\*  4. Система выводит сообщение об ошибке  \*\*\*\*\* |

Сценарию «Составить расписание» соответствуют таблицы 5.1 – 5.2.

Т а б л и ц а 5.1 – Главный раздел сценария выполнения варианта использования «Составить расписание»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актер | Администратор |
| Краткое описание | Администратор составляет расписание для каждого курса в функционале данной информационной системы |
| Цель | Составить расписание для курсов |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 5.2 – Раздел типичный ход событий варианта использования «Составить расписание»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Составить расписание»  3. Актер в соответствующие поля добавляет запрашиваемые данные | 2. Система открывает окно ввода данных  4. Система сохраняет данные в файле |

Сценарию «Изменить запись на курс» соответствуют таблицы 6.1 – 6.2.

Т а б л и ц а 6.1 – Главный раздел сценария выполнения варианта использования «Изменить запись на курс»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актеры | Администратор |
| Краткое описание | Администратор производит изменение записи абитуриента на курс |
| Цель | Изменить запись на курс |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 6.2 – Раздел типичный ход событий варианта использования «Изменить запись на курс»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Изменить запись на курс»  3. Актер вносит запрашиваемые данные  5. Актер вносит запрашиваемые данные | 2. Система запрашивает данные, которые нужно изменить  4. Система запрашивает новые данные  6. Система сохраняет данные в файле |

Сценарию «Удалить запись на курс» соответствуют таблицы 7.1 – 7.3.

Т а б л и ц а 7.1 – Главный раздел сценария выполнения варианта использования «Удалить запись на курс»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актеры | Администратор |
| Краткое описание | Администратор производит удаление записи на курс |
| Цель | Удалить запись на курс |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 7.2 – Раздел типичный ход событий варианта использования «Удалить запись на курс»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Удалить запись»  3. Актер вносит данные | 2. Система запрашивает введение данных, которые нужно удалить  4. Система сохраняет результат |

Т а б л и ц а 7.3 - Раздел исключений сценария варианта события «Удалить запись на курс»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| \*\*\*\*\*  3. Актер вводит неверные данные  \*\*\*\*\* | \*\*\*\*\*  4. Система выводит сообщение об ошибке  \*\*\*\*\* |

Сценарию «Учет данных (курс-дата проведения-количество человек)» соответствуют таблицы 8.1 – 8.3.

Т а б л и ц а 8.1 – Главный раздел сценария выполнения варианта использования «Учет данных (курс-дата проведения-количество человек)»

|  |  |
| --- | --- |
| Вариант использования | Вход в систему |
| Актеры | Администратор |
| Краткое описание | Администратор получает информацию об учете данных |
| Цель | Учет данных |
| Тип | Базовый |
| Ссылки на другие варианты использования | Отсутствуют |

Т а б л и ц а 8.2 - Раздел типичный ход событий варианта использования «Учет данных (курс-дата проведения-количество человек)»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Актер в интерфейсе пользователя нажимает на кнопку «Учет данных (курс-количество человек-дата проведения)»  3. Актер вводит советующее наименование курса | 2. Система запрашивает аббревиатуру нужного курса  4. Выводит учет данных в формате курс-количество человек-дата проведения |

Т а б л и ц а 8.3 - Раздел исключений сценария варианта события «Учет данных (курс-дата проведения-количество человек)»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| \*\*\*\*\*  3. Актер вводит неверную аббревиатуру курса  \*\*\*\*\* | \*\*\*\*\*  4. Система выводит сообщение об ошибке  \*\*\*\*\* |

**Диаграммы последовательностей**

Диаграммы последовательностей показывают более подробное описание вариантов использования (Рисунок 1) на уровне классов (диаграмма классов – рисунок 2).

Сценарию «Поиск необходимого курса» соответствует Рисунок 3.



Рисунок 3 – Диаграмма последовательностей «Поиск необходимого курса»

Сценарию «Запись абитуриента на курс» соответствует Рисунок 4.



Рисунок 4 – Диаграмма последовательностей «Запись абитуриента на курс»

Сценарию «Проверка списка абитуриентов» соответствует Рисунок 5.



Рисунок 5 – Диаграмма последовательностей «Проверка списка абитуриентов»

Сценарию «Изменение записи на курс» соответствует Рисунок 6.



Рисунок 6 – Диаграмма последовательностей «Изменение записи на курс»

Сценарию «Удаление записи на курс» соответствует Рисунок 7.



Рисунок 7 – Диаграмма последовательностей «Удаление записи на курс»

Сценарию «Учет данных (курс-дата проведения-количество человек)» соответствует Рисунок 8.



Рисунок 8 – Диаграмма последовательностей «Учет данных (курс-дата проведения-количество человек)»

**3 Реализация**

Для реализации продукта использовалась среда разработки Visual Studio Сommunity 2016. Ниже представлено описание классов, реализованных в информационной системе.

Admin – класс администратора. Данный класс хранит общие переменные и реализует множество функций для работы с абитуриентом такие, как: «Печать списка курсов», «Поиск необходимого курса», «Заполнение списка студентов», «Запись абитуриента на соответствующий курс», «Вывод списка абитуриентов определенного курса», «Учет данных (курс-дата проведения-количество человек)», «Изменить запись на определенном курсе», «Удалить запись на курс».

//Класс администратор

class Admin{

private:

string last\_name, //Фамилия

name; //Имя

public:

//Печать списка курсов

void print\_list\_of\_course() {}

//Поиск необходимого курса

void find\_course(string name\_of\_course) {}

//Заполнение списка студентов

void list\_of\_students(string name\_of\_course, int a) {}

//Запись абитуриента на соответствующий курс

void record\_new\_student(string name\_of\_course) {}

//Вывод списка абитуриентов определенного курса

void print\_list\_of\_students(string name\_of\_course){}

//Учет данных (курс-дата проведения-количество человек)

void data\_records(string name\_of\_course) {}

//Изменить запись на определенном курсе

void change\_registration(string name\_of\_course) {}

//Удалить запись на курс

void delete\_registration(string name\_of\_course) {}

};

**4 Интерфейс**

Все описанные ниже диалоговые окна являются шаблонами. Они описывают должный вид законченной информационной системы. На данном этапе разработки данный графический интерфейс реализован не был, поэтому описание классов не соответствует должным элементам приведённых ниже форм. Для создания данных форм использовалась среда разработки Microsoft Visual Studio Community 2016.

**Формы интерфейса**

Окно приветствия запрашивает у пользователя (администратора) логин и пароль идентификации для дальнейшей работы, при вводе не правильных данных программа завершает работу (Рисунок 9).

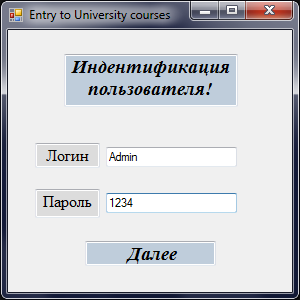


Рисунок 9 – Экран идентификации администратора

Окно главного меню предлагает пользователю действия, соответствующие сценариям соответствующие таблицам 1.1 – 8.3 (Рисунок 10). Подробно взаимодействие с каждым окном описано в руководстве пользователя (приложение 2).

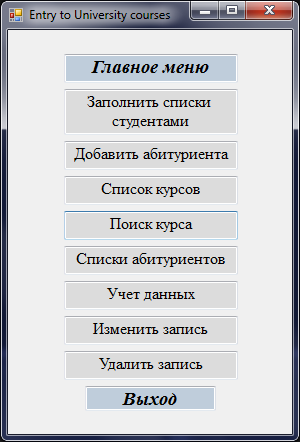


Рисунок 10 – Экран главного меню

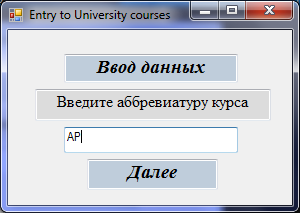


Рисунок 11 – Экран ввода данных. Запрос наименования курса

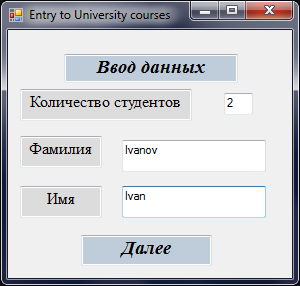


Рисунок 12 – Экран ввода данных. Заполнение списка студентами

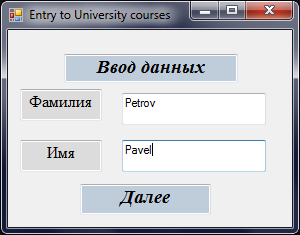


Рисунок 13 – Ввод данных. Добавление абитуриента

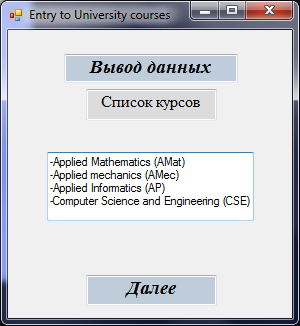


Рисунок 14 – Вывод данных. Список курсов

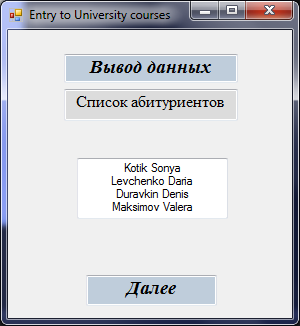


Рисунок 15 – Вывод данных. Список абитуриентов, указанного курса

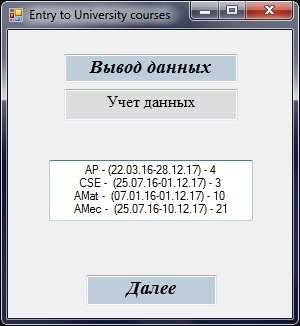


Рисунок 16 – Вывод данных. Учет данных (курс-дата проведения-количество студентов)

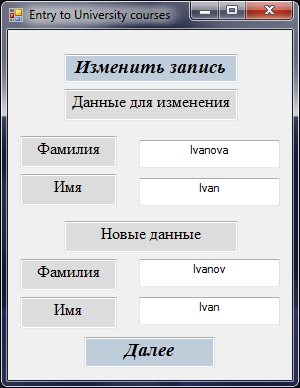


Рисунок 17 - Изменить запись

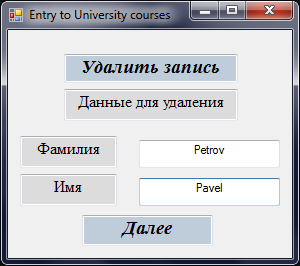


Рисунок 18 – Удаление записи

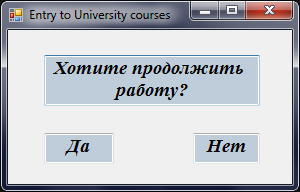


Рисунок 19 – Диалог с пользователем

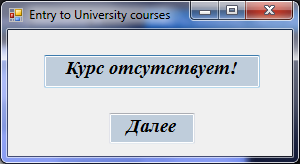


Рисунок 20 – Вывод об ошибке

**Описание структур папок проекта**

Так как на данном этапе разработки программа не имеет оконного интерфейса, то структура папок проекта соответствует консольному приложению (Рисунок 21).

В процессе реализации написания информационной системы активно использовался отладчик. Все действия отладчик производил в соответствующей папке «Debug» (Рисунок 21, отмечен цифрой 2). По окончанию работы производилось тестирование итоговой работоспособности проекта, данные по которому хранятся в папке «Курсовая работа» (Рисунок 21, отмечен цифрой 1).

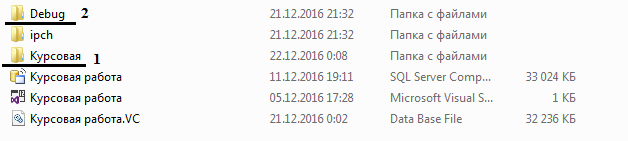


Рисунок 21 – Состав папки с проектом

В папке «Debug» проводилось тестирование продукта, её составляющими являются три программы. (Рисунок 22).

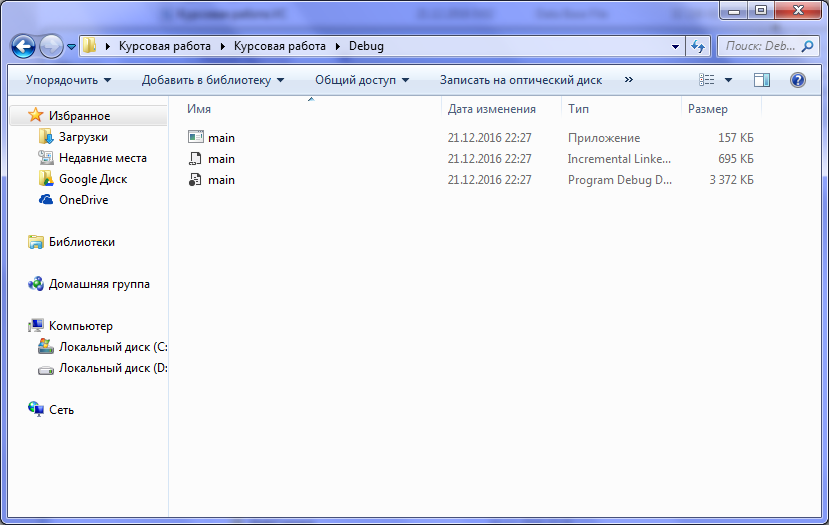


Рисунок 22 – Состав папки Debug

В папке «Курсовая работа» происходила основная разработка проекта, параллельно с отладкой и сборкой конечной версии, поэтому все сборочные файлы (под цифрой 1), программные файлы (под цифрой 4), заголовочные файлы (под цифрой 2) и файлы \*.cpp, содержащие коды программы (под цифрой 3) и используемые базы данных (под цифрой 5) (Рисунок 23).

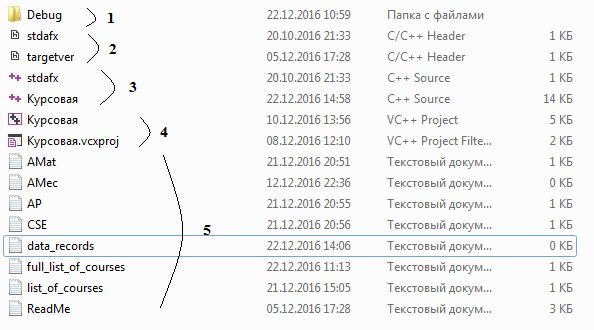


Рисунок 23 – Состав паки «Курсовая»

**5 Обобщённое описание работы программы**

При запуске программы открывается окно идентификации администратора. Пользователь вводит логин и пароль, после чего попадает в меню выбора действий. После чего пользователь выбирает соответствующие действия и следует подсказкам программы или же Руководству пользователя (приложение 2), приложенного к программе. Сама программа заносит изменения, которые производились пользователем при работе в соответствующие файлы. Прежде, чем начать работу с данной информационной системой желательно ознакомиться с руководством.

**Заключение**

В результате были выполнены все задачи, поставленные преподавателем при рассмотрении предоставленной темы. При изучении дополнительной информации по объектно-ориентированному языку программирования C++ была привлечена дополнительная литература, а также некоторые интернет ресурсы. Вся дополнительно использовавшаяся информация описана ниже (с ссылками на используемые интернет ресурсы).

**Список литературы**

1. Литвиненко, Н.А. Технология программирования на С. Современный учебник программирования. [Электронный ресурс] — Электрон. дан. — БХВ-Петербург, 2010. — 281 с. — Режим доступа: <http://www.proklondike.com/books/cpp/technology_of_programming_on_cplus.html>
2. Стефан Р. Дэвис - С++ Для чайников. [Электронный ресурс] — Электрон. дан. — Вильямс, 2003. — 336 с. — Режим доступа: <http://www.proklondike.com/books/cpp/cplus_dlja_chainikov.html>
3. Энтони Уильямс — Параллельное программирование на С++ в действии. [Электронный ресурс] — Электрон. дан. — Москва, 2012. — 672 с. — Режим доступа: <http://www.proklondike.com/books/cpp/Paralelnie_programirovanie.html>
4. Андрианова А.А, Исмагилов Л.Н., Мухтарова Т.М. — Объектно ориентированное программирование на C++. [Электронный ресурс] — Электрон. дан. — Казань: Приволжский Федеральный Университет, 2010. — 230 с. — Режим доступа: <http://www.proklondike.com/books/cpp/Adrianova_o_o_p_C2plus_2010.html>
5. Программирование в C++ Builder: метод. Пособие. [Электронный ресурс] — Электрон. дан. — БГПУ имени М. Акмуллы, 2007. — 80 с. — Режим доступа: <http://e.lanbook.com/book/43288>
6. Р. Лафоре - Объектно-ориентированное программирование в С++. [Электронный ресурс] — Электрон. дан. — Питер, 2013. — 192 с. — Режим доступа: <http://cppstudio.com/post/9497/>
7. Страуструп Б. – Язык программирования С++. [Электронный ресурс] — Электрон. дан. — Бином, 2004. — 369 с. — Режим доступа: <http://cppstudio.com/post/8439/>
8. Шилдт Г. — C++. Базовый курс. [Электронный ресурс] — Электрон. дан. — Бонус, 2014. — 624 с. — Режим доступа: <http://cppstudio.com/post/9507/>
9. Шлее М. — Профессиональное программирование на C++. [Электронный ресурс] — Электрон. дан. — Бонус, 2010. — 851 с. — Режим доступа: <http://cppstudio.com/post/8634/>
10. Х.М. Дейтел, П.Дж. Дейтел — Как программировать на С++. [Электронный ресурс] — Электрон. дан. — Бонус, 2006. — 1454 с. — Режим доступа: <http://cppstudio.com/post/8446/>

**Приложение 1**



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** |

**Кафедра информационных систем управления**

|  |
| --- |
|  |
| Обозначение листа утверждения |

|  |
| --- |
| Информационная система записи на университетские курсы |
| наименование программы |

|  |
| --- |
| Техническое задание |
| наименование документа |

|  |
| --- |
| 5 страниц |
| объем документа |

|  |  |
| --- | --- |
| Исполнитель |  |
| Котик С.В. | |  | | --- | |  |   подпись |
| Группа Б8219 | |  | | --- | |  |   дата подписания |

|  |  |  |
| --- | --- | --- |
| Руководитель курсового проекта | |  | | --- | |  |   подпись |
| Березкина Г.Л. | |  | | --- | |  |   дата подписания |

2016

**Введение**

Данная программа применима в учебной области. Она была разработана для записи на университетские курсы, предусматривает сбор и хранение данных об абитуриентах, курсах, которые они посещают, информацию о группах и о расписании занятий. Программа позволяет осуществлять поиск абитуриента и посещаемых им курсах, а также поиск преподавателей по названию курса.

Программа позволяет выдать нужную информацию, например, по вводу фамилии и имени абитуриента, дате подаче заявления или наименованию курса.

**Основания для разработки**

В ряд документов, на основании которых ведется разработка входит: «Учебный план специальности «Прикладная информатика», «График учебного процесса», а также материалы дисциплины «Объектно-ориентированное программирование», которая читается Берёзкиной Галиной Леонидовной.

Тема разработки, согласованная с преподавателем, звучит как «Информационная система записи на университетские курсы».

**Назначение разработки**

Данная программа предназначена для упорядочивания данных абитуриентов, записавшихся на университетские курсы, а именно дат подачи заявлений, паспортные данные (ФИО, дата рождения), наименование курса, срок обучения.

**Требования к программе или программному изданию**

* Требования к функциональным характеристикам
* Статистика должна вестись в автоматическом режиме.
* Должны быть предусмотрена и доступны все материалы, необходимые для осуществления деятельности предприятия, включая хранилища для используемых данных, документы и требующие отчеты.
* Должна быть предусмотрена возможность ограничения прав некоторых пользователей.
* Требования к надежности
* Должны быть предусмотрены механизмы проверки корректности вводимых пользователем данных.
* Необходимо настроить периодически резервное копирование имеющихся данных.
* Условия эксплуатации
* Присваивание каждому пользователю собственного логина и пароля для входа в программу, неизвестных другим пользователям.
* Настраивание ролей для каждой группы пользователей, ограничивающих им доступ к отдельным частям информационной базы и некотором функциям программы.
* Требования к составу и параметрам технических средств
* Рабочие станции сотрудников должны обладать адекватной аппаратной конфигурацией, соизмеримой с размерами обрабатываемой информации для нормальной работы и быстрого функционирования разработанной системы.
* Требования к информационной и программной совместимости
* На рабочих станциях должна быть установлена операционная система семейства Windows (XP, 7, 8).
* Программное обеспечение, устанавливаемое на кассы, должно поддерживать формат выходных данных, корректно интерпретируемых разработанной ИС.
* Требования к маркировке и упаковке

Программный продукт передается клиенту на компакт-диске с соответствующей маркировкой, либо на съемном носителе с теми же подписями.

**Требования к программной документации**

Предварительный состав программной документации:

- Техническое задание.

- Текст программы с комментариями.

- Руководство пользователя.

- Пояснительная записка.

**Технико-экономические показатели**

Ориентировочно, данная программа должна повысить качество работы университета. Расходы на годовое обслуживание минимальны.

**Стадии и этапы разработки**

Этапы и содержание работ:

* Определение задачи

Определение цели приложения, выполняемые задачи. Поиск путей решения, анализ альтернатив, имеющих решений.

* Планирование работ

Планирование работ по созданию системы/ПО. Определение и анализ риска.

* Макет

Предоставление доработанного макета программы, с учетом всех поправок «заказчика», с утвержденным функционалом и интерфейсом.

* Установление границ приложения

Построение модели приложения. Анализ с целью установить, что является, а что не является частью разрабатываемого приложения.

* Специфицирование требований ПО приложения

Построение моделей ПО с целью детализированного описания перечня сервисов, которые должны выполнять приложения с указанием как приложение реагирует на те или иные входные данные, как ведет себя в определённых ситуация и т.д. Выявление функциональных требований к ПО приложения, системных требований, проверка их реализуемости и тестируемости. Выявление нефункциональных требований. Определение ограничений, накладываемых на действия и функции, выполняемые приложением, временные ограничения, стандарты и т.д. Обеспечение качества ПО.

* Проектирование интерфейса пользователя

Анализ деятельности пользователя. Создание проекта прототипа интерфейса.

* Проектирование

Выявление классов. Шаблоны проектирования. Проектирование базы данных. Тестирование. Документирование. Организация сопровождения и развития ПО.

* Прототипирование

Создание прототипа ПО, демонстрирующего правильность принятых в ходе проектирования решений

Работа принимается и считается завершенной в том случае, если работает и отлажен утвержденный на этапе проектирования функционал, с заранее оговоренным интерфейсом.

**Приложение 2**

**Руководство пользователя**

Данное руководство содержит информацию и иллюстрации для помощи осуществления взаимодействия с программой «Entry to University courses», информационной системой для записи абитуриента на университетские курсы.

1. **Вход в программу**

Для входа в программу вам необходимо ввести логин (Admin) и пароль доступа (1234) (Рисунок 9). При неправильном вводе пароля, программа завершит свою работу. Внимательно вводите пароль!

1. **Вывод данных**
   1. Вывод списка курсов

Чтобы вывести список всех курсов вам необходимо:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Список курсов» (Рисунок 10);
3. Откроется дополнительно окно, введите аббревиатуру курса, который нужно найти (Рисунок 11);
4. Если курс с такой аббревиатурой не найден, программа выведет соответствующее сообщение «Курс отсутствует!» (Рисунок 20);
5. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
   1. Вывод списка абитуриентов определенного курса

Чтобы вывести список всех абитуриентов, находящихся в указанном списке:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Списки абитуриентов» (Рисунок 10);
3. Откроется дополнительно окно, введите аббревиатуру курса, который нужно найти (Рисунок 11);
4. Если курс с такой аббревиатурой не найден, программа выведет соответствующее сообщение «Курс отсутствует!» (Рисунок 20);
5. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
   1. Вывод учета данных (курс-дата проведения-количество человек)

Чтобы вывести учет данных:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Учет данных» (Рисунок 10);
3. Откроется дополнительно окно, введите аббревиатуру курса, который нужно найти (Рисунок 11);
4. Если курс с такой аббревиатурой не найден, программа выведет соответствующее сообщение «Курс отсутствует!» (Рисунок 20);
5. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
6. **Поиск указанного курса**

Чтобы найти нужный курс:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Поиск курса» (Рисунок 10);
3. Откроется дополнительно окно, введите аббревиатуру курса, который нужно найти (Рисунок 11);
4. Если курс с такой аббревиатурой не найден, программа выведет соответствующее сообщение «Курс отсутствует!» (Рисунок 20);
5. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
6. **Заполнение данных об абитуриенте**
   1. Заполнение списка студентов

Чтобы заполнить список студентов:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Заполнить списки студентами» (Рисунок 10);
3. Откроется дополнительно окно, введите аббревиатуру курса (Рисунок 11);
4. Если курса с такой аббревиатурой нет, программа выведет сообщение об ошибке (Рисунок 20);
5. Введите количество человек и запрашиваемые данные об абитуриента (Рисунок 12);
6. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
   1. Занесение данных об абитуриенте

Чтобы внести данные об абитуриенте:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Добавить абитуриента» (Рисунок 10);
3. Введите запрашиваемые данные об абитуриенте (Рисунок 14);
4. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
5. **Изменение записи на курс**

Чтобы изменить данные записи абитуриента на курс необходимо:

1. Войти в программу (смотрите раздел 1 данного руководства);
2. В основном меню выбрать пункт «Изменить запись» (Рисунок 10);
3. Откроется дополнительно окно, введите аббревиатуру курса (Рисунок 11);
4. Если курса с такой аббревиатурой нет, программа выведет сообщение об ошибке (Рисунок 19);
5. Затем задайте в поля ввода фамилию и имя абитуриента для редактирования (Рисунок 17);
6. Если абитуриента с такими данными в списке нет, программа выведет сообщение об ошибке;
7. Программа запросит ввести новые данные об абитуриенте (Рисунок 17);
8. Для перехода в основное окно в дополнительном окне выбрать пункт «Да». Для выхода из программы – «Нет» (Рисунок 19).
9. **Удаление записи на курс**
10. Войти в программу (смотрите раздел 1 данного руководства);
11. В основном меню выбрать пункт «Удалить запись» (Рисунок 10);
12. Откроется дополнительно окно, введите аббревиатуру курса (Рисунок 11);
13. Если курса с такой аббревиатурой нет, программа выведет сообщение об ошибке (Рисунок 20);
14. Затем занести запрашиваемые данные об абитуриента (Рисунок 18);
15. Для перехода в основное окно в дополнительном окне выбрать пункт «Назад». Для выхода из программы – «Нет» (Рисунок 19).
16. **Выход из программы**

Чтобы завершить работу программы выберите в меню пункт «Выход» (Рисунок 10), либо нажмите на крестик в верхнем правом углу.