ПЗПИ-19-4

Гриненко Анастасия

# Архитектура

Программа будет создаваться на языке С# с использованием среды разработки Visual Studio 2019 на платформе .NЕТ Framework 4.7.2 с использованием интерфейса программирования приложений Windows Forms.

Выбор языка обусловлен объектно-ориентированным подходом к написанию программы. На данный момент язык С# является одним из самых популярных, удобных и полноценных среди объектно-ориентированных языков программирования.

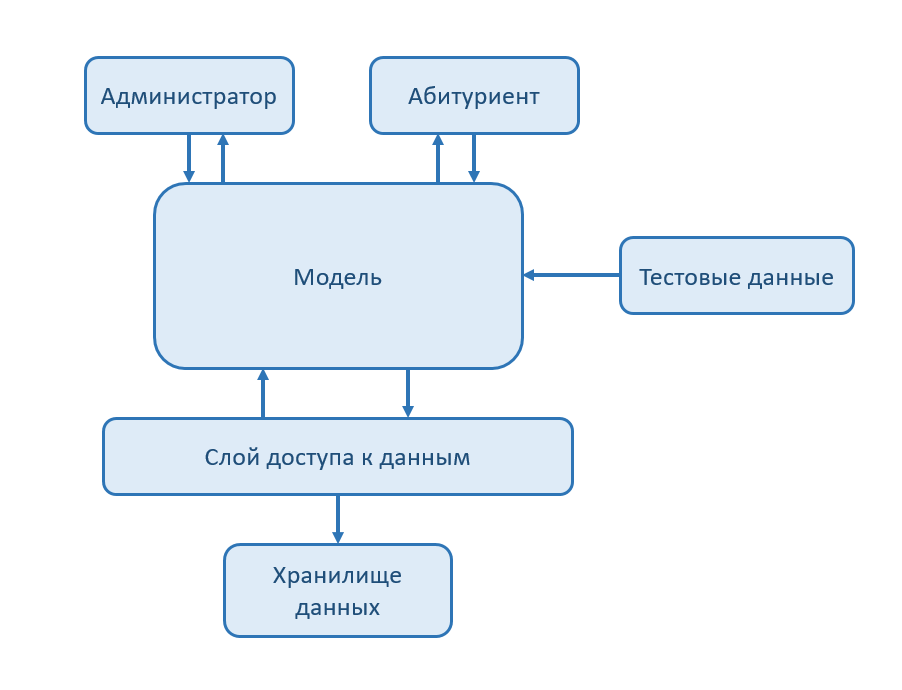
В процессе проектирования рассматривались 2 платформы, с которыми язык С# имеет наилучшее взаимодействие, а именно .NЕТ Core и .NЕТ Framework. Наш выбор пал на последнюю по причине увеличенного количества управляющих элементов и более высокого уровня развития платформы. К сожалению, это означает что приложение будет работать исключительно на машинах с операционной системой Windows.

Для реализации пользовательского интерфейса было решено использовать графический интерфейс, так как он простой для восприятия и интуитивно понятен. Для реализации интерфейса мы будем использовать интерфейс Windows Forms, который является частью платформы .NЕТ Framework. С его помощью у нас будет возможность создать полноценный интерфейс для пользователей приложения.

# Структура проекта

Будет разумно поделить проект, описанный в спецификации, на два отдельных приложения: для администратора и для абитуриентов, поскольку функции этих типов пользователей между собой не пересекаются. Кроме того, для тестирования программы на этапе программирования уместно создать несколько тестовых данных для проверки правильности работы программы.

На этом основании, можем составить следующую структуру проекта.



# Объектная модель

## Описание

В базе абитуриента имеются университеты.

Абитуриент ищет университеты по разным параметрам, просматривает информацию про университеты, добавляет и удаляет университеты из Списка избранных.

Администратор редактирует информацию об университетах.

Об университете известно: идентификатор, название, изображение, адрес, специальности.

Об абитуриенте известно: пароль, имя, город.

Об администраторе известно: пароль, имя.

О специальности известно: название, цена обучения, проходной балл на бюджет прошлого года, кол-во мест на обучение на бюджетной основе, кол-во всех мест, кол-во мест на дневную форму, кол-в мест на заочную форму обучения.

Абитуриент – это пароль + имя + город.

База абитуриентов – это коллекция всех абитуриентов.

Администратор – это пароль + имя.

Специальность – это название + цена обучения + проходной балл на бюджет прошлого года + кол-во мест на обучение на бюджетной основе + кол-во всех мест + кол-во мест на дневную форму + кол-в мест на заочную форму обучения.

Университет – это коллекция специальностей + идентификатор + название + изображение + адрес.

База университетов – это коллекция всех университетов.

Список избранных – это коллекция университетов.

Справочник абитуриента – это коллекция абитуриентов + коллекция университетов + администратор.

**Замечание**. Множественное число будет означать коллекцию.

## Методы

Программа находится на этапе разработки, поэтому методы будут добавляться по мере необходимости. На данном этапе к основным методам можно отнести:

* Регистрация нового абитуриента

Проверка при входе абитуриента

* Регистрация администратора

Проверка при входе администратора

* Добавление новой специальности

Удаление специальности

Редактирование информации о специальности

* Добавление нового университета

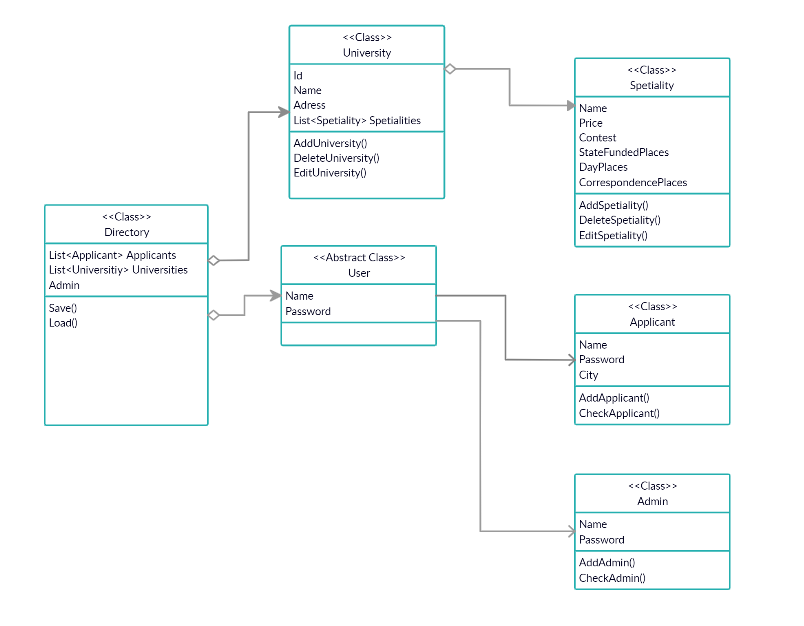
Удаление университета

Редактирование информации об университете

* Сохранение данных

Загрузка данных

## Диаграмма классов



## Хранилище и доступ к данным

Все данные, отмеченные в работе словом «база», а также «список пользователей» будут представлены в виде отдельных бинарных файлов в каталоге проекта. Данные будут переводиться из текстовых в поток байтов при помощи сериализации с использованием встроенной библиотеки ВinaryFormatter. Защита данных в рамках роботы не рассматривается.

GitHub: <https://github.com/NyotaUhura/Handbook>