**Проектная спецификация**

**Архитектура**

Курсовой проект будет создан на языке C# с использованием среды разработки Microsoft Visual Studio 2019 на платформе .Net Framework 4.7.2 с использованием интерфейса программирования приложений Windows Forms.

Объектно-ориентированный подход к написанию программы является решающим фактором при выборе языка. На данный момент C# является одним из самых востребованных, простых в использовании и полноценных объектно-ориентированных языков программирования.

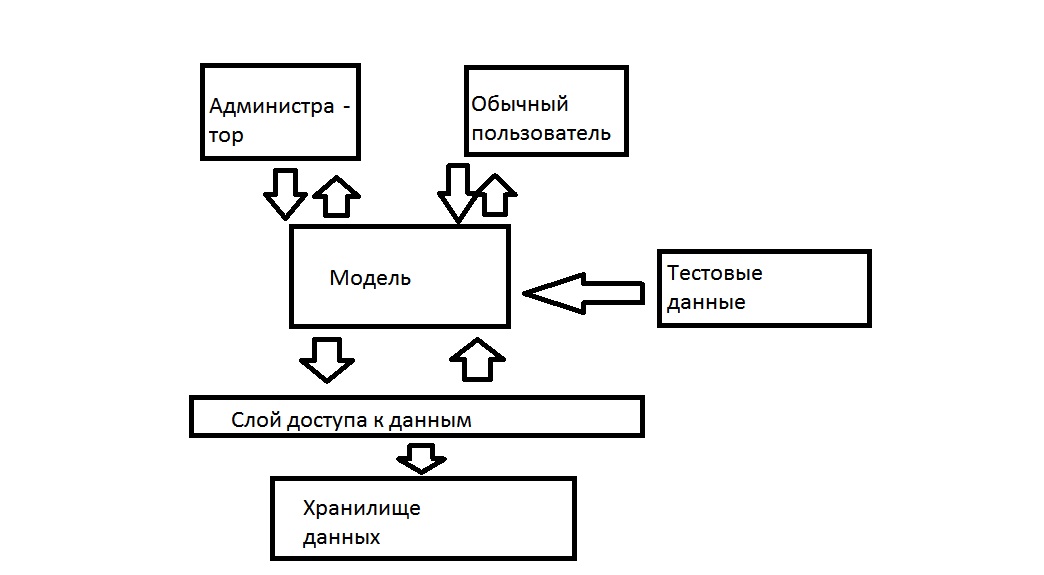
В процессе проектирования было рассмотрено две основные платформы, с которыми язык С# имеет наилучшее взаимодействие: .Net Core и .Net Framework. Платформа .Net Framework имеет более высокий уровень развития, поэтому и была выбрана нами. Хотя это означает, что приложение будет работать исключительно на устройствах с операционной системой Windows.

Пользовательский интерфейс было реализовано с помощью графического интерфейса, так как он простой для восприятия и интуитивно понятен. Интерфейс Windows Forms является частью платформы .Net Framework. С его помощью у нас будет возможность создать полноценный интерфейс для пользователей приложения.

**Структура проекта**

Логично будет разделить проект, описанный в спецификации, на два приложения: для обычного пользователя и администратора. Это обусловлено тем, что функционал администратора не включает в себя функции обычного пользователя (далее ОП). Также для тестирования программы необходимо создать тестовые данные.

Потому структура проекта будет выглядеть следующим образом.



Объектная модель

Описание

В **базе статей** содержатся **водные объекты** (реки, моря озёра и т.д.).

**Водные объекты** имеют **название**, **тип**, **режим**, **площадь** (площадь бассейна), **дополнительную информацию**.

В **базе тестов** содержатся **тесты**.

**Тесты** состоят из его **названия,** **вопросов** и **баллов за тест**.

**Вопросы** содержат **условие**, **варианты ответа**, **правильный вариант**.

Оба типа пользователей имеют доступ к **базе статей** и **базе тестов.**

В **базе пользователей** содержатся **пользователи**.

О пользователе известно **логин**(имя), **пароль**, **общий счет по тестам**.

К простым типам не относятся:

- база статей;

- водные объекты;

- база тестов;

- тесты;

- вопросы;

- варианты ответа;

- база пользователей;

- пользователь.

Из них не являются коллекцией и коллекций не содержат водный объект, пользователь.

База статей – коллекция водных объектов.

База тестов – коллекция тестов.

Тесты – название теста + коллекция вопросов + балл за тест.

Вопросы – условие вопроса + коллекция вариантов ответа + правильный ответ + номер вопроса.

База пользователей – коллекция пользователей.

Методы

1. регистрация пользователя (пользователь);
2. добавление статьи (водный объект);
3. добавление теста (тест);
4. удалить тест (название теста);
5. удалить статью (название водного объекта);
6. добавить баллы за тест (пользователь, баллы за тест);
7. завершить тест ();
8. добавить вопрос (вопрос, тест);
9. удалить вопрос (номер, тест);
10. сменить название (название);
11. редактировать вопрос (условие, коллекция вариантов ответа, правильный ответ) – по умолчанию все аргументы равны советующим полям вопроса;
12. редактировать статью (название, тип, режим, площадь, дополнительную информацию) – по умолчанию все аргументы равны советующим полям водного объекта;
13. вход ().

Прочие методы будут добавляться по необходимости*.*

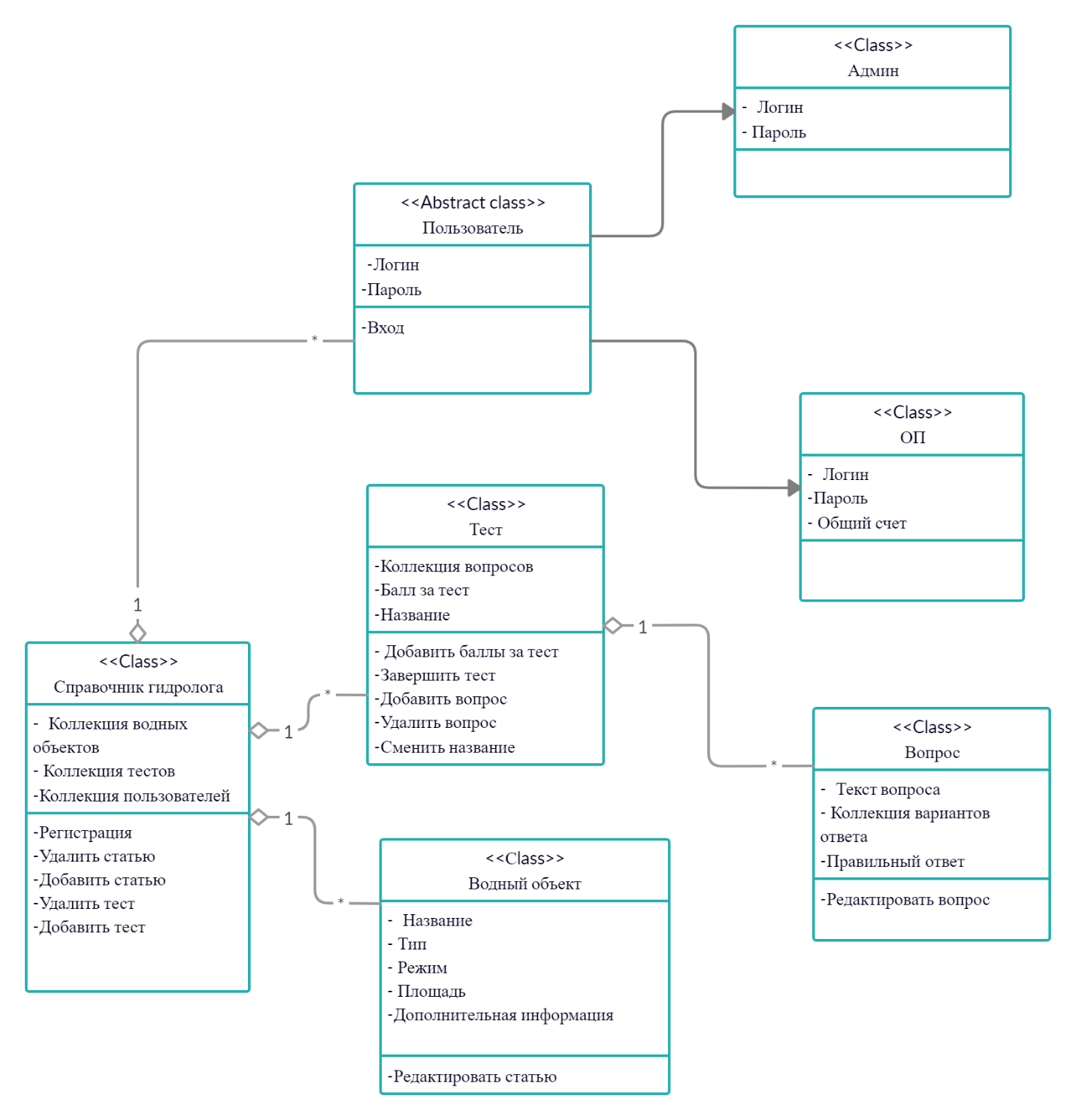


Рисунок 1. Классовая модель проекта

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 Binary Formatter. Защита данных в рамках роботы не рассматривается.

Ссылка на будущий проект - <https://github.com/avojarot/Course_Project_Kizitskyi>