# Архитектура

Программа будет создаваться 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.

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

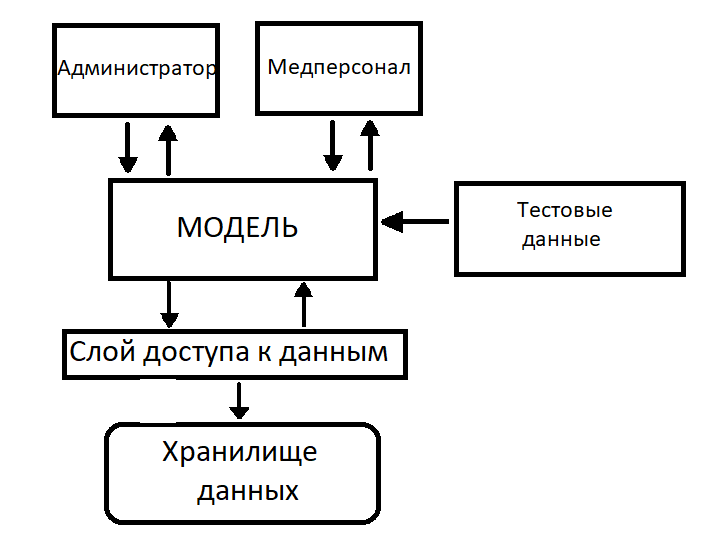
В процессе проектирования рассматривались 2 платформы, с которыми язык С# имеет наилучшее взаимодействие, а именно .NET Core и .NET Framework. Наш выбор пал на последнюю по причине увеличенного количества управляющих элементов и более высокого уровня развития платформы. К сожалению, это означает что 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 .NET 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 медработников (врачи и младший медперсонал), 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 следующею структуру проекта.



# Объектная модель

## Описание

В **базе медикаментов** имеются **товары** (препараты либо расходники)

**Администратор** дополняет **список пользователей**, добавляет товары, регистрирует **поставки**

**Медперсонал** имеет доступ **к базе пациентов**, базе медикаментов, текущим назначениям пациента.

Врач, имеет доступ к функциям медперсонала, доступ к **базе болезней**.

О товаре известно: **идентификатор, название, изображение, описание**

О пользователе известно: **должность, логин** (имя и фамилия), **пароль**

О пациенте известно: **ФИО, Дата рождения, Адрес, пол, история болезни, текущие лечение, скрытая информация**

О болезни известно: **симптомы, описание, способы лечения, информация о распространении**.

**Текущее лечение пациента и способы лечения болезней содержат только название препаратов, а не тип данных Товар**

**К простым типам не относятся:**

База медикаментов, товар, пользователь (администратор, медперсонал, врач), список пользователей, база пациентов, пациент, база болезней, болезнь.

**Из этих отделяем те, что не являются коллекциями и коллекций не содержат:**

Товар, пользователь, пациент.

**Первые пользовательские типы:**

Товар – это идентификатор + название + изображение + описание

Пользователь – это логин + пароль + доступ

Пациент – это ФИО, + Дата рождения + Адрес + пол + история болезни + текущие лечение + скрытая информация

**Остальные типы:**

База медикаментов – это коллекция товаров

список пользователей – это коллекция пользователей

база пациентов – это коллекция пациентов

Болезнь – это коллекция симптомов + описание + способы лечения+ информация о распространении

база болезней – это коллекция болезней

Поставка — это дата + коллекция порций товара

База поставок — это коллекция поставок

Порция — это товар + его количество.

Медучреждение – это несколько коллекций: пользователей, медикаментов, пациентов, болезней, поставок

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

## Методы:

На данном этапе к главным методам мы можем отнести:

1. Процедура приема пациента (пользователь, база пациентов, база медикаментов)
2. Регистрация новых пользователей программы (пользователь, список пользователей)
3. Учёт поставок медикаментов (пользователь, база медикаментов, поставка, база поставок)
4. Просмотр и изменение информации о пациенте (пользователь, база пациентов)

В дальнейшем мы будем добавлять методы по мере необходимости.

Более подробное описание методов можно посмотреть в спецификации к проекту.

## Диаграмма классов

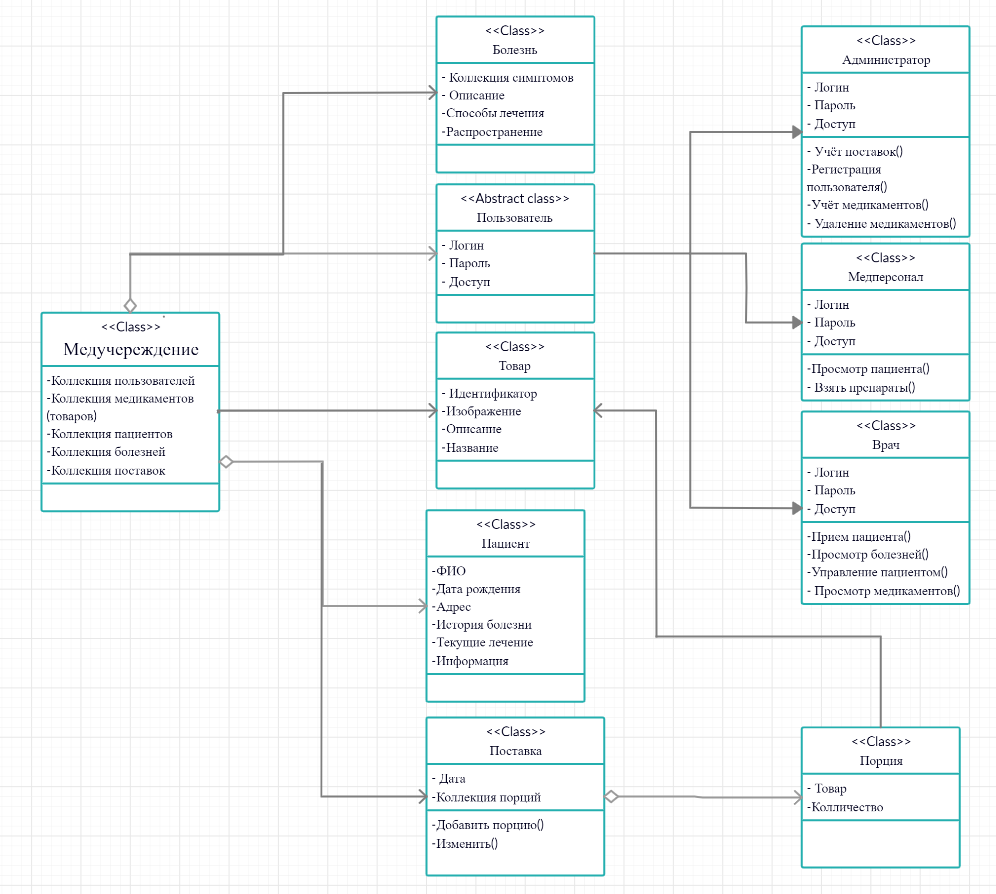


Диаграмма 1. Диаграмма классов объектной модели

Ссылка на диаграмму: <https://app.creately.com/diagram/FzQ8SAG6S0U/edit>

# 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 сериализаци с использованием втроенной библиотеки BinaryFormatter. Защита данных в рамках роботы не рассматривается.

# План выполнения проекта

1. Составление спецификации (до 12.04)
2. Проектирование (до 20.04)
3. Программирование модели (до 26.04)
4. Программирование пользовательского интерфейса (до 17.05)
5. Написание пояснительной записки (до 31.05)

Тестирование провидится параллельно с программированием пользовательского интерфейса