**Архитектура**

Программа будет создаваться на языке C# с использованием 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.

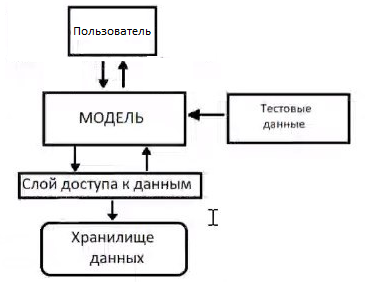
В процессе проектирования рассматривалось 2 платформы, с которыми язык C# имеет наилучшее взаимодействие, а именно .Net Core и .Net Framework. Наш выбор пал на последнюю по причине увеличенного количества управляющих элементов и более высокого уровня развития платформы. К сожалению, это означает, что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

Для реализации пользовательского интерфейса было решено использовать графический интерфейс, т. к. 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 пользовательского приложения.

**Структура проекта**

Будет неразумно делить проект, описанный в спецификации на два отдельных проекта, поскольку функции продавца и покупателя пересекаются. 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

На **бирже товаров** имеются **товары.**

**Пользователь** добавляет новые **товары** и оформляет **заказы**.

О **товаре** известно: **идентификатор, наименование, единица измерения, розничная цена, оптовая цена, минимальный заказ для опта, запас, продавец.**

О **пользователе** известно: **идентификатор, имя, пароль, номер телефона, электронная почта.**

О **заказе** известно: **Номер заказа, покупатель, продавец, товар, количество, дата и время заказа, адрес доставки, способ оплаты**.

Товар содержит только идентификатор продавца, а не тип данных Пользователь. Заказ содержит только идентификаторы товара, продавца и покупателя а не типы данных Товар и Пользователь соответственно.

**К простым типам не относятся:**

База товаров, товар, список пользователей, пользователь, база заказов, заказ.

**Из них не являются коллекциями и коллекций не содержат:**

Товар, пользователь, заказ.

**База товаров** – это коллекция **товаров**.

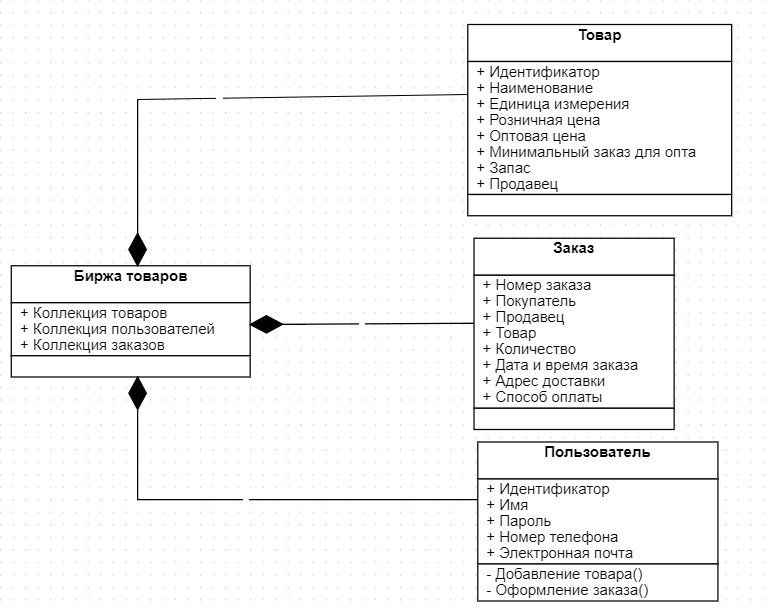
**База заказов** – это коллекция **заказов**.

**Список пользователей** – это коллекция **пользователей**.

Методы:

На данном этапе к главным методам можно отнести:

1. Добавление пользователем товара.
2. Оформление пользователем заказа.



**Хранилище и доступ к данным**

Все данные будут храниться в единственном бинарном файле «exchange.bin». Данные будут переводиться в поток байтов при помощи сериализации с использованием встроенной библиотеки BinaryFormatter. Защита данных в рамках работы не рассматривается.

Ссылка на Гитхаб: <https://github.com/Yurambolo/CommodityExchange>