# Архитектура

Программа будет написана на языке С# с использованием среды разработки Microsoft Visual Studio Community 2019 на платформе .NET Framework 4.7.2, тип приложения: настольное с использованием Windows Forms.

Выбор языка С№ обуславливается тем, что на данный момент это один из наиболее развитых объектно-ориентированных языков, а задание подразумевает ООП подход. Он является наиболее совершенным и комфортным в использовании из существующих языков.

Мы выбрали С# .NЕТ Framework среди прочих платформ из-за его развитости и зрелости. Это полнофункциональная платформа с большим количеством управляющих элементов, которая поддерживает разработку современных приложений. Явным недостатком является отсутствие мультиплатформености, а именно возможность работы приложения лишь только на аппаратах под управлением ОС Windows.

Для взаимодействия с пользователем был выбран графический интерфейс из-за его понятности и интуитивной простоты использовании для клиента. Реализация происходит через Windows Forms, которые являются частью платформы .NET Framework.

# Структура проекта

При условиях и требованиях, поставленных в данном задании курсового проекта, максимально эффективно с точки зрения удобств программирования и использования, будет создание двух отдельных приложений: для клиента и администратора так как их функционал не пересекается.

Так же в программу будут включены наборы тестовых данных для проверки корректности ее работы и тестирования функций.

Таким образом модель данного проекта имеет следующий вид:

Клиент

Администратор

Наборы тестовых данных

Хранилище данных

Слой доступа к данным

Модель

# Объектная модель

## Описание

**Список товаров** включает в себя список всех доступных предметов и квартир.

**Администратор** добавляет **товары**, рассматривает заявки, просматривает **список товаров**, вносит изменения, отмечает проданные варианты, удаляет **товары** из **базы**.

**Клиент** просматривает доступные к покупке **товары**, добавляет понравившиеся в **корзину**, осуществляет покупку, оставляет заявку на продажу товара.

Есть два типа товаров: **квартиры** и **предметы**.

О **квартире** известно: **наименование**, **изображение**, **адрес**, **стоимость**, **дата приема на баланс ломбарда**.

О **предмете** известно: **наименование**, **изображение**, **стоимость**, **дата приема на баланс ломбарда**.

О **администраторе** известно: **имя**, **номер паспорта.**

О **покупателе** известно: **логин,** **имя**, **номер паспорта**, **пароль**.

**К простым типам не относятся:**

Список квартир, квартира, список предметов, предмет, пользователь (администратор, клиент), корзина, заявка.

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

Заявка, квартира, предмет, администратор, покупатель.

**Первые пользовательские типы:**

Предмет – название, фото, цена, дата приема.

Квартира – название, фото, цена, адрес, дата приема.

Администратор – имя, номер паспорта.

Покупатель – логин, имя, номер паспорта, пароль.

**Остальные типы:**

Список предметов – коллекция предметов.

Список квартир – коллекция квартир.

Список пользователей – коллекция пользователей.

Заявка – название, фото, адрес (если квартира).

Корзина – коллекция выбранных к покупке товаров.

Ломбард —несколько коллекций: пользователей, квартир, предметов, проданных товаров.

## Методы

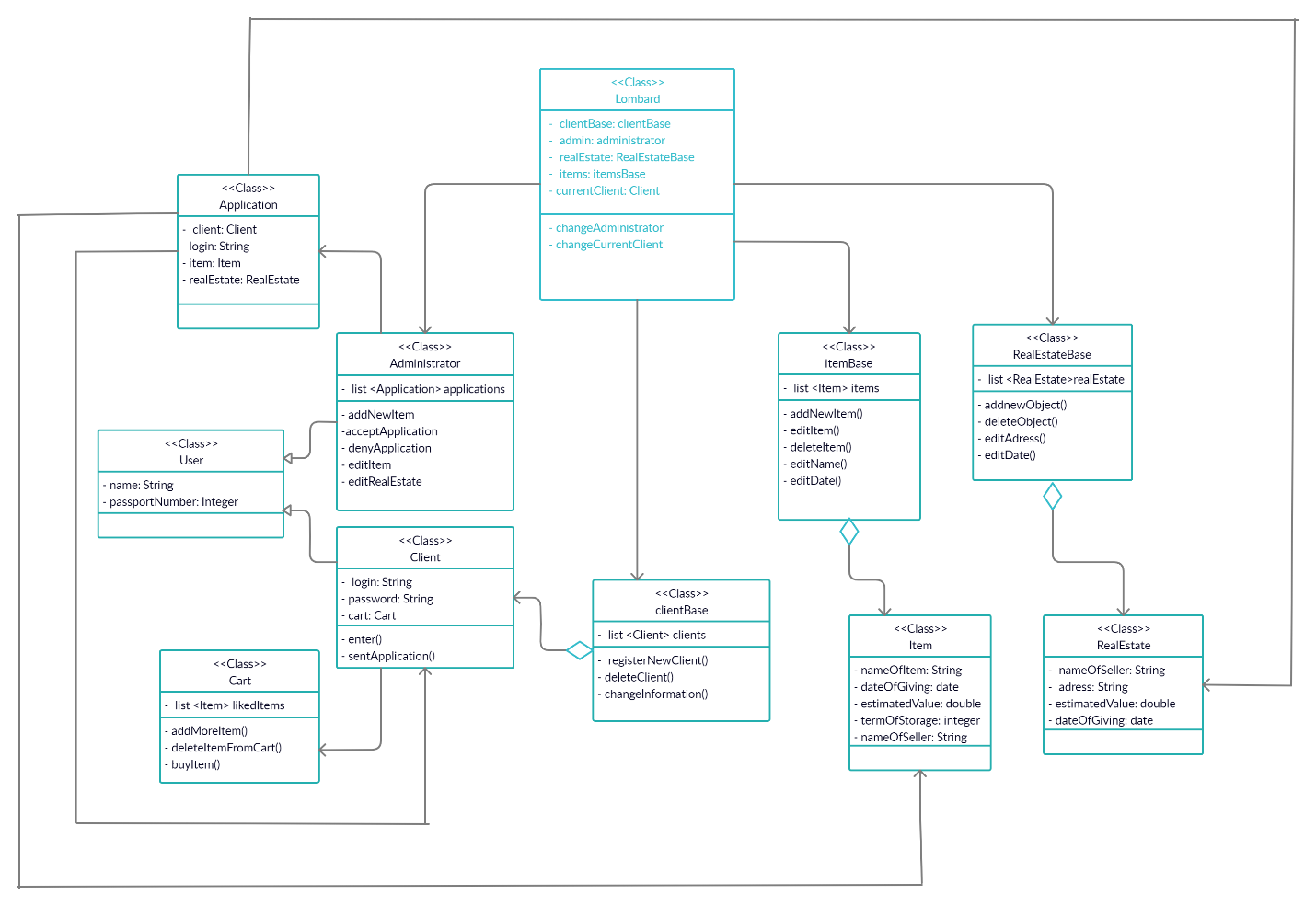
Основные методы:

* Регистрация новых клиентов программы (пользователь, список пользователей).
* Отправка заявки на продажу товара (пользователь, заявка).
* Одобрение заявки (администратор, заявка).
* Просмотр списка товаров доступных к покупке (клиент, список доступных товаров).
* Покупка выбранных товаров (покупатель, корзина).
* Изменение товаров (администратор, список квартир, квартира, список предметов, предмет).
* Удаление товаров (администратор, список квартир, квартира, список предметов, предмет).

Список методов будет расширяться по мере работы над проектом и написании программного кода.

Методы более детально оговорены в спецификации к проекту.

## Диаграмма классов



# Хранение и доступ к данным

Все данные, отмеченные словами “база” или “список” будут представлены в виде отдельных бинарных файлов в каталоге данного проекта. Данные будут переводиться из текстовых в поток байтов при помощи сериализации с использованием встроенной библиотеки ВinaryFormatter. Файл под названием «lombard.bin» расположен по адресу “Gavryushenko\_project\bin\lombard.bin”. Защита данных в рамках данного проекта не рассматривается.

### Программирование проекта

Контроль версий проекта будет осуществляться с помощью сайта GitHub.

<https://github.com/ArtemGavr/Lombard>