**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

Специальность 1-40 04 01 «Информатика и технологии программирования»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по дисциплине «Объектно-ориентированное программирование»

на тему: **«Расчет потребности в сырье и материалах на плановый объем продукции»**

Исполнитель: студент гр. ИП-22

Коваленко А.И.

Руководитель: преподаватель кафедры «Информатика»

В.Н. Шибеко

Дата проверки: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите: ­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписи членов комиссии

по защите курсовой работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель 2022

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc101806248)

[**1 АНАЛИЧЕСКИЙ ОБЗОР** 4](#_Toc101806249)

[**1.1 Аналитический обзор существующих методов и средств реализации** 4](#_Toc101806250)

[**1.2 Обзор средств автоматизации** 4](#_Toc101806251)

[**1.3 Требования к проектированному программному обеспечению** 4](#_Toc101806252)

[**1.4 Используемые средства** 4](#_Toc101806253)

[**2 АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ** 5](#_Toc101806254)

[**2.1 Анализ задания на проектирование** 5](#_Toc101806255)

[**2.2 Преценденты и их описание** 5](#_Toc101806256)

[**2.3 Описание предметной области** 5](#_Toc101806257)

[**2.4 Отношения между классами предметной области** 5](#_Toc101806258)

[**2.5 Описание вспомогательных классов** 5](#_Toc101806259)

[**3 СТРУКТУРА И ОСНОВНЫЕ АЛГОРИТМЫ СОЗДАННОГО ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ** 6](#_Toc101806260)

[**3.1 Структурная схема приложения и ее описание** 6](#_Toc101806261)

# **ВВЕДЕНИЕ**

Бжлдаспмриотльттрпьооьооооооооооооооооооооооооооооооооооооооооооооооооооооо

# **1 АНАЛИЧЕСКИЙ ОБЗОР**

# **1.1 Описание объекта**

Моделированием в информационных технология называется процесс целенаправленного отражения свойств некоторого объекта, называемого оригиналом, в другом объекте – модели, существенных для исследователя, посредством создания и использования этой модели, с целью получения данных и знаний об оригинале.

В данной работе в роли главного объекта выступает справочная служба аптек и лекарств города. Будет проведено не только моделирование самого объекта, но и реализация методов автоматизации, так как современные автоматизированные система улучшают качество обслуживания клиентской базы, снижают напряженность и трудоемкость труда персонала, повышают производительность работы всего персонала, а также уменьшают количество ошибок в его действиях.

Итогом проектирования модели будет приложение под управлением операционной системы Windows с подключенным источником данных, а именно базой данных, в которой будет хранится вся необходимая информация.

В результате накопленных данных будет производиться автоматическое формирование статистики, включающих построение графических диаграмм по деятельности предприятия.

# **1.2 Обзор существующих методов решения**

В качестве предметной области автоматизации рассматривается автоматизация справочной службы аптек и лекарств города. Данный проект удобен для обработки данных по оказанию услуг населению в получении необходимых лекарственных средств.

Автоматизация деятельности справочной службы аптек и лекарств города позволит составлять и обрабатывать заказы быстрее. Пользователь сможет выбрать удобный для него тип доставки заказа. Узнать цены на лекарства можно будет любому человеку, а также оформлять сами заказы, не выходя из дома.

Каждому клиенту предоставляются функции формирования заказа, составление корзины для формирования заказа, просмотр стоимости лекарств, а также просмотр своего личного кабинета для привязки банковского счета. Клиент сможет оплатить заказ в приложении, войдя в свой личный кабинет, или наличными придя в аптеку.

# **1.3 Используемые средства**

Для создания оконного Windows-приложения был выбран объектно-ориентированный язык программирования C#, который является языком разработки приложений для платформы Microsoft .NET Framework и основой для использования технологии Windows Forms. Данная технология была выбрана так как она имеет достаточно удобный инструмент для разработки стилизованных Windows приложений.

Для хранения данных используется база данных MS SQL Server 2022. Для удобного управления базой данных было использовано специальное средство администрирования SQL Server Management Studio.

Наиболее качественной средой разработки для языка C# на операционной системе Windows является Visual Studio – официальный продукт от компании Microsoft – разработчика платформы .NET. Разработка этого проекта велась в среде Visual Studio 2019 Community.

# **1.4 Постановка задачи**

На основании произведённого анализа существующих методов, можно сформировать список требований, для разрабатываемой системы:

1. Приложение не должно иметь лишнего функционала.

2. Доступ к приложению должен быть удобным и простым.

3. Приложение должно иметь достаточно удобный и интуитивно понятный интерфейс для различных пользователей.

4. Приложение должно быть масштабируемым.

Программный продукт должен предоставлять:

1. Различные виды справочников, которыми пользователи с определенной ролью могут заниматься , а именно обработкой которая включает в себя такие функции как: добавление, удаление, изменение данных, а пользователи с другой ролью ими пользоваться: осуществлять поиск, сортировку, фильтрацию данных.

2. Способ совершения покупки выбранного лекарства в аптеке на основе заказа.

3. Возможность выбора типа доставки заказа.

4. Возможность привязки счета для оплат в личном кабинете.

5. Получение статистики в графическом виде. Также необходимо разделение доступа к функционалу приложения по ролям: Гость, Клиент, Администратор, Работник, Владелец.

Каждой роли будет доступен определенный функционал приложения. Дополнительно следует учесть возможность пользователю иметь сразу несколько ролей одновременно.

Возможности гостя:

1. Авторизироваться в приложении.

2. Зарегистрироваться в приложении.

3. Просмотреть прейскурант интересующей аптеки.

Возможности клиента:

1. Формировать корзину из требуемых лекарств.

2. На основе корзины формировать заказ по выбранным лекарствам.

3. Просматривать счета и заказы в личном кабинете.

4. Просматривать лекарства в выбранной аптеке.

5. Осуществлять сортировку лекарств по необходимым критериям.

6. Осуществлять фильтрацию аптек и лекарств.

Возможности работника:

1. Редактировать справочник лекарств.

2. Редактировать данные прейскурант.

3. Обрабатывать заказ клиента.

Возможности администратора:

1. Удалять и добавлять сотрудников аптеки.

Возможности владельца:

1. Просмотр прейскуранта выбранной аптеки.

2. Просмотр статистики.

3. Просмотр наличия лекарства в заданной аптеке.

# **2 АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

# **2.1 Прецеденты и актеры**

Для реализации работы справочной службы аптек и лекарств города были добавлены следующие роли:

* владелец;
* клиент;
* работник;
* администратор;
* гость.

***2.1.1*** Роль «Владелец»

Для реализации модели владелицу доступны следующие возможности:

* просмотр прейскуранта заданной аптеки;
* просмотр статистики по аптекам;
* просмотр наличия определенного лекарства в аптеках.

На рисунке 2.1 представлена диаграмма прецедентов роли «Владелец».

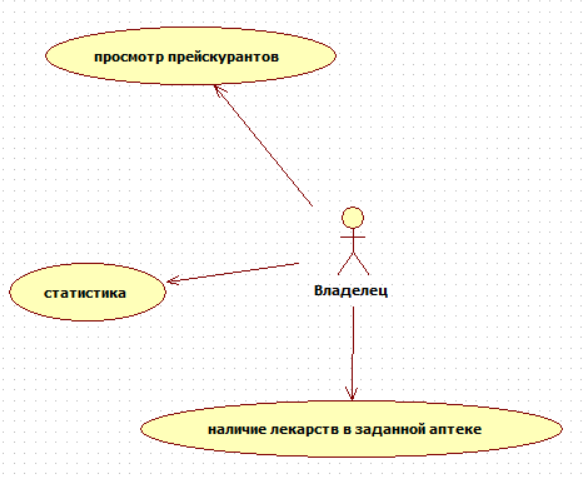
****

Рисунок 2.1 – Диаграмма прецедентов роли «Владелец».

***2.1.2*** Роль «Администратор»

Для реализации модели администратор доступны следующие возможности:

* работа с сотрудниками (редактирование, добавление, удаление);
* сохранение данных в базу данных.

На рисунке 2.2 представлена диаграмма прецедентов роли «Администратора».

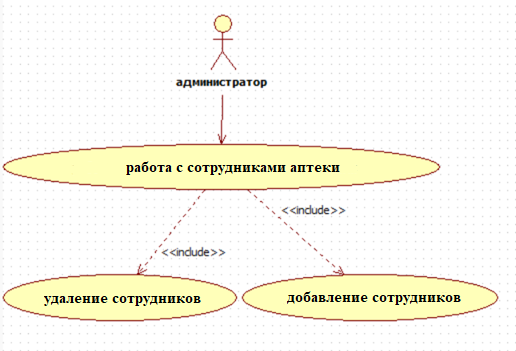


Рисунок 2.2 – Диаграмма прецедентов роли «Администратора».

***2.1.3*** Роль «Клиент»

Для реализации модели клиент доступны следующие возможности:

* изменение корзины (редактирование, добавление, удаление);
* просмотр корзины;
* оформление заказа;
* просмотр лекарств в аптеках;
* просмотр счета карты в личном кабинете;
* просмотр заказов в личном кабинете.

На рисунке 2.3 изображена диаграмма прецедентов роли «Клиент».

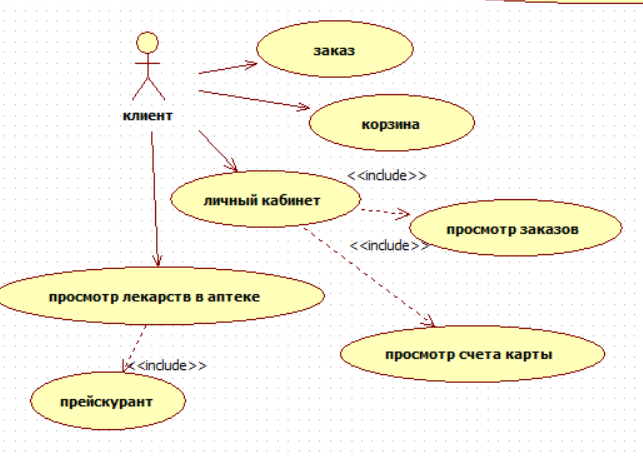
****

Рисунок 2.3 – Диаграмма прецедентов роли «Клиент».

На рисунке 2.4 изображена диаграмма прецедентов взаимодействия клиента с корзиной:

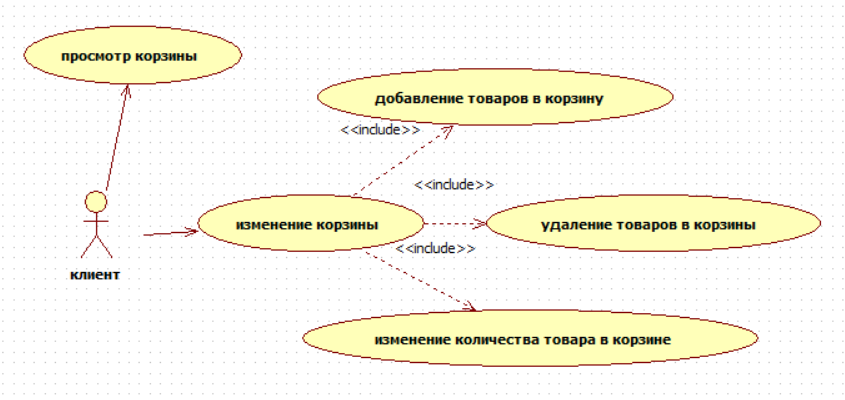
****

Рисунок 2.4 – Диаграмма прецедентов взаимодействия клиента с корзиной.

***2.1.4*** Роль «Работник»

Для реализации модели клиент доступны следующие возможности:

* изменение прейскуранта аптеки (редактирование лекарства, добавление лекарства, удаление лекарства);
* просмотр прейскуранта;
* ведения справочника списка лекарств;
* обработка заказа клиента.

На рисунке 2.5 изображена диаграмма прецедентов роли «Работник».

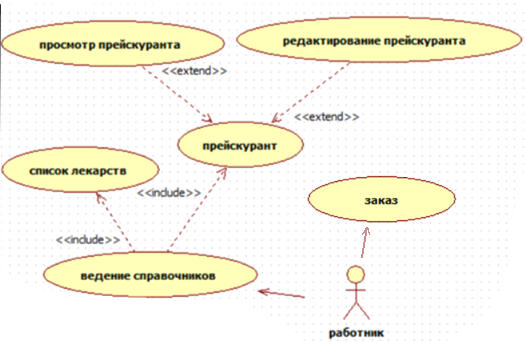


Рисунок 2.5 – Диаграмма прецедентов роли «Работник».

На рисунке 2.6 изображена диаграмма прецедентов взаимодействия работника и клиента, а именно заказ:

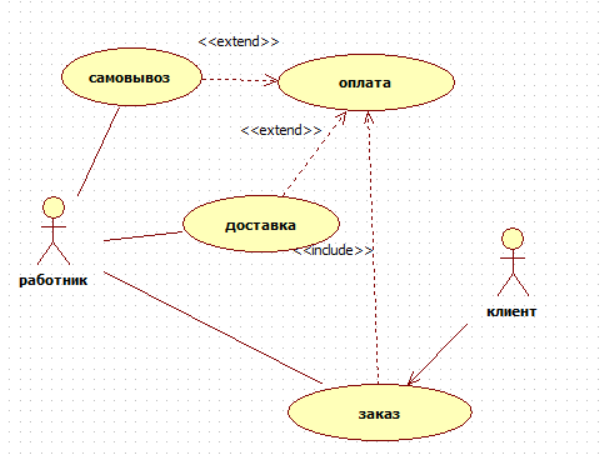


Рисунок 2.6 – Диаграмма прецедентов оформления заказа.

**2.1.6** Роль «Гость»

Для реализации модели гость доступны следующие возможности:

* просмотр прейскуранта;
* вход в программу;
* регистрация.

На рисунке 2.7 изображена диаграмма прецедентов взаимодействия гостя с программой:

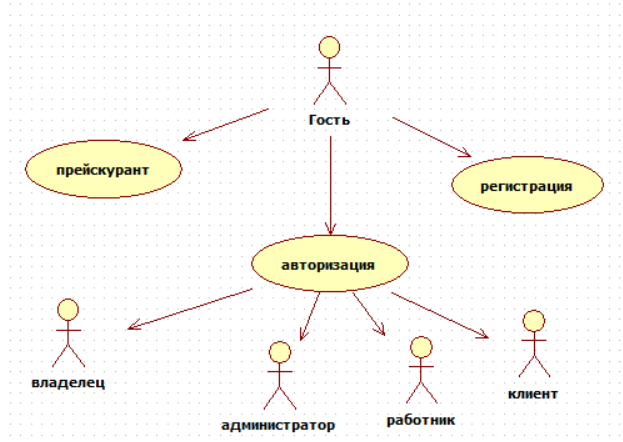
****

Рисунок 2.7 – Диаграмма прецедентов гостя.

# **2.2 Преценденты и их описание**

Неавторизированный пользователь заходит в приложение в качестве гостя, появляются доступные функции в главном меню:

1. Авторизация: UC1

2. Регистрация: UC1

US 1 Авторизация:

Действующие лица: Гость

Предыстория: Человек зашел в виде гостя

1. Появляются поля для ввода логина и пароля, а так же кнопка “Войти” , “Регистрация” и “Просмотр прейскуранта”

2. Гость вводит правильный логин и пароль и переходит в пункт US [2-4]

A1. Пользователь нажимает кнопку “Регистрация”

1. В случае успешной регистрации выводится соответствующее сообщение

E1 Введенные данные не прошли проверку

1. Система просит повторить попытку.

A2. Пользователь нажимает кнопку “Войти”

E1. Попытка входа была завершена не успешна, система просит пользователя повторить попытку или зарегистрироваться, если аккаунт еще не создан

A3. Пользователь нажимает кнопку “ Просмотр прейскуранта ”

1. Появляется окно прейскуранта

US 2 Авторизация прошла под клиентом:

Действующие лица: Клиент

Предыстория: Гость ввел данные клиента во время авторизации

Доступные функции:

1. Просмотр лекарств в аптеке US 6

2. Изменение корзины US 7

3. Заказ US 8

4. Личный кабинет US 9

US 3 Авторизация прошла под работником:

Действующие лица: Работник

Предыстория: Гость ввел данные работника

Доступные функции:

1. Ведение справочников US 10

2. Исполнение заказа US 8

US 4 Авторизация прошла под админом

Действующие лица: Админ

Предыстория: Гость ввел данные админа

Доступные функции:

1. Работа с сотрудниками US 11

US 5 Авторизация прошла под владельцем

Действующие лица: Владелец

Предыстория: Гость ввел данные владельца

Доступные функции:

1. Просмотр лекарств в аптеке US 6

2. Статистика US 12

3. Прейскурант US 13

US 6 Просмотр лекарств в аптеке:

Действующие лица: владелец, клиент

Предыстория: владелец или клиент начали просмотр лекарств в аптеке

Доступные функции:

1. Просмотр аптек с прейскурантом

2. Сортировка лекарств по цене

3. Сортировка лекарств по стране производителю

4. Добавление выбранного лекарства в корзину (для клиента)

US 7 Работа с корзиной

Действующие лица: клиент

Предыстория: Клиент добавил товар корзину и перешел в нее

Доступные функции:

1. Удаление лекарств из корзины

2. Добавление лекарств в корзину

3. Просмотр корзины

4. Изменение количества лекарств в корзине

E1. Клиент попытался добавить лекарств больше , чем находится в аптеке

1. Система выводит сообщение об ошибке

5. После редактирования корзины клиент может нажать кнопку “Оплата” и перейти в пункт US 8

US 8 Заказ товара

Действующие лица: Работник, клиент

Предыстория: Клиент пытается оформить заказ по выбранным товарам в корзине

1. Система проверяет, привязал ли клиент свой банковскую карту к своему аккаунту

А1. Если нет, то просит пользователя привязать свою банковскую карту к своему аккаунту

2. Система проверяет, хватает ли денежных средств на карте для оплаты корзины

A2. Если нет, то оповещает пользователя о нехватке денежных средств

E1. Платежная система недоступна

1. Система выводит соответствующее сообщение

2. Завершает сеанс заказа

3. После успешной оплаты система предложит выбор оформления заказа Доставка/Самовывоз

А3 Если клиент выбирает доставку , то он должен указать адрес доставки и контактный телефон

А4 Если клиент выбирает самовывоз , то он должен указать телефон для и работнику требуется связаться с клиентом

4. После выбора типа оформления заказа в базу данных добавится информация о заказе: Код аптеки, код лекарства, сумма заказа , Доставка/Самовывоз , код заказа , код клиента.

5. Теперь работник может увидеть данные о заказе.

6. Работник должен обработать заказ.

A5. В случае, если пользователь передумал забирать заказ или ждать курьера, то он может сообщить об отмене заказа работнику, а тот в свою очередь может удалить заказ из базы данных

7. Если заказ не был отменен и наступил день окончания заказа, то заказ может считаться завершенным

US 9 Личный кабинет клиента

Действующие лица: Клиент

Предыстория: Клиент зашел в свой личный кабинет

Доступные функции:

1. Просмотр совершенных заказов

2. Возможность привязки банковской карты

E1 При вводе неверных данных просит повторить попытку.

US 10 Ведение справочников

Действующие лица: Работник

Предыстория: Работник перешел в пункт меню “Ведение справочников”

Доступные функции:

1. Просмотр и редактирование прейскуранта

2. Просмотр и редактирование лекарств в аптеке

US 11 Работа с сотрудниками

Действующие лица: Админ

Предыстория: Администратор перешел в пункт меню “Работа с сотрудниками”

Доступные функции:

1. Просмотр сотрудников

2. Добавление сотрудников в БД

3. Удаление сотрудников из БД

US 12 Статистика

Действующие лица: Владелец

Предыстория: Владелец перешел в пункт меню “Статистика ”

Доступные функции:

1. Просмотр стоимости проданных лекарств по всем аптекам

US 13 Прейскурант

Действующие лица: Владелец

Предыстория: Владелец перешел в пункт меню “Прейскурант ”

Доступные функции:

1. Просмотр стоимости всех лекарств по аптекам

# **2.3 Описание классов и предметной области**

В ходе работы были выделены следующие классы предметной области:

Класс *User* –класс, описывающий сущность «Пользователь», которая хранит в себе данные всех пользователей.

Таблица 1 –Класс User.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | int | IdUser | Cвойство для хранения уникального идентификатора пользователя |
| public | string | Login | Свойство для хранения логина пользователя |
| public | string | Password | Свойство для хранения пароля пользователя |
| public | string | Status | Свойство для хранения статуса пользователя в системе |

Класс *Client* **–** класс, описывающий сущность «Клиент», является основным классом предметной области и хранит в себе данные клиента и его счета.

Таблица 2 – Описание содержимого класса Client.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | string | Familyname | Cвойство для хранения фамилии клиента |
| public | string | Name | Свойство для хранения имени имени клиента |
| public | string | Аddress | Свойство для хранения адреса клиента |
| public | int | IdClient | Свойство для хранения уникального идентификатора клиента |
| public | int | IdBill | Свойство для хранения уникального идентификатора счета клиента |

Класс *BankAcount* **–** класс, описывающий сущность «Банковского счета», хранит в себе данные о счетах клиента.

Таблица 3 – Описание содержимого класса BankAcount.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | int | IdClient | Свойство для хранения уникального идентификатора клиента |
| public | int | IdBill | Свойство для хранения уникального идентификатора счета клиента |
| public | int | Deposit | Свойство для хранения суммы на счету |

Класс *Order* **–** класс, описывающий сущность «Заказа», хранит в себе данные о счетах клиента.

Таблица 4 – Описание содержимого класса Order.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | string | TypeDelivery | Cвойство для хранения типа доставки |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| public | int | IdPharmacy | Свойство для хранения уникального индификатора аптеки |
| public | int | Price | Свойство для хранения стоимости заказа |
| public | int | IdDrug | Свойство для хранения уникального индефикатора лекарства |
| public | int | IdOrder | Свойство для хранения уникального индефикатора заказа |
| public | int | IdClient | Свойство для хранения уникального индефикатора клиента |
| public | date | Date | Свойство для хранения даты заказа |
| public | int | DeliveryFee | Свойство для хранения платы за доставку заказа |

Класс *OrderLines***–** класс, описывающий вспомогательную сущность «Строки заказа» для формирования заказа и хранит в себе необходимые строки заказа.

Таблица 5 – Описание содержимого класса OrderLines.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | int | IdOrder | Свойство для хранения уникального индефикатора заказа |
| public | int | IdDrug | Свойство для хранения уникального индефикатора лекарства |
| public | int | CountMedicine | Свойство для хранения количества лекарства |
| public | int | Price | Свойство для хранения стоимости заданного оличества лекарств |

Класс *Pharmacy* **–** класс, описывающий сущность «Аптеки» для предоставления цен на лекарства и продажи лекарств.

Таблица 6 – Описание содержимого класса Pharmacy.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | int | IdPharamacy | Свойство для хранения уникального индефикатора аптеки |
| public | string | Street | Свойство для хранения улицы |
| public | string | Title | Свойство для хранения названия аптеки |
| public | string | PhoneNumber | Свойства для хранения телефона аптеки |
| public | int | Allowance | Свойство для хранения процента надбавки аптеки |

Класс *PriceList* **–** класс, описывающий вспомогательную сущность «Прейскуранта» для аптеки, хранящую в себе список всех лекарств с заданной ценой.

Таблица 7 – Описание содержимого класса PriceList.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | int | IdPharmacy | Свойство для хранения уникального индефикатора аптеки |
| public | int | IdMedicine | Свойство для хранения уникального индефикатора лекарства |
| public | int | Price | Свойство для хранения цены за единицу лекарства |

Класс *Medicine* **–** класс, описывающий сущность «Лекарств», хранящую в себе информацию о лекарстве.

Таблица 8 – Описание содержимого класса *Medicine*.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор доступа** | **Тип** | **Имя** | **Описание** |
| public | int | IdMedicine | Свойство для хранения уникального индефикатора лекарства |
| public | int | Price | Свойство для хранения цены лекарства |
| public | string | Title | Свойство для хранения названия лекарства |
| public | string | Country | Свойство для хранения страны производителя лекарства |

На рисунке 2.8 представлена диаграмма классов предметной области.

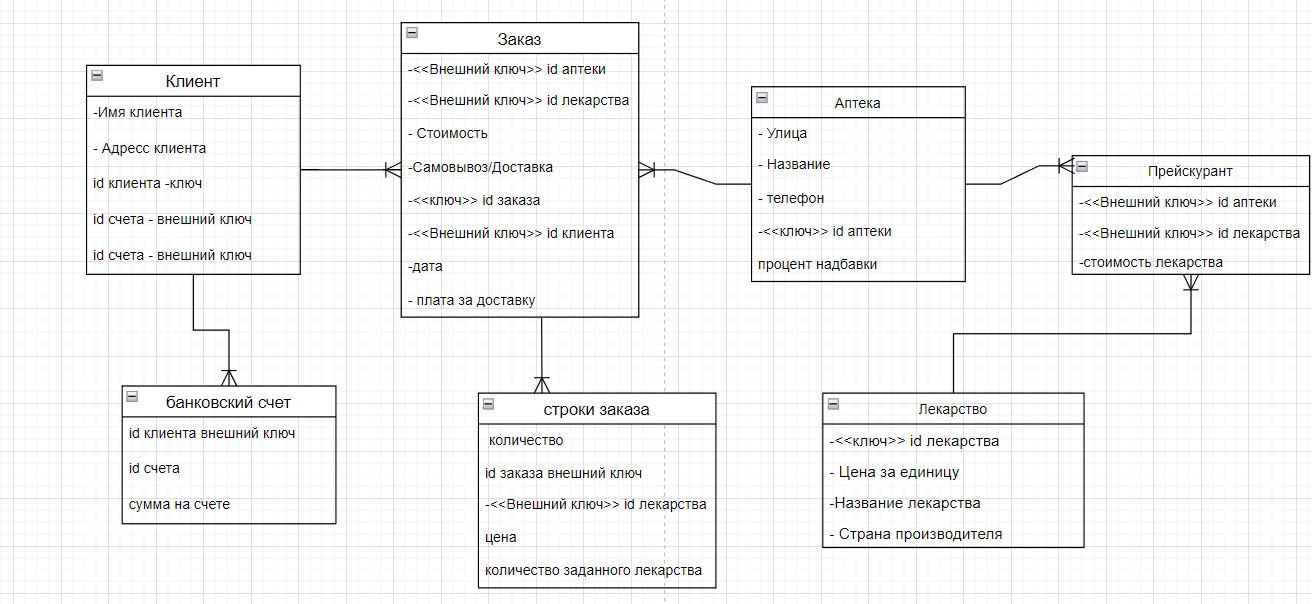


Рисунок 2.8 – Диаграмма классов предметной области

# **2.4 Отношения между классами предметной области**

Рассмотрим виды связей между сущностями в приведенной диаграмме классов предметной области (рисунок 2.8).

Класс Client соединен связью «одни ко многим» с классом BankAcount, так как у клиента могут быть резервные счета.

Класс Client соединен связью «одни ко многим» с классом Order, так как один клиент может совершать много заказов.

Класс Order соединен связью «одни ко многим» с классом OrderLines, так как один заказ может включать много строк заказа.

Класс Pharmacy соединен связью «одни ко многим» с классом Order, так как одна аптека может оформлять много заказов.

Класс Pharmacy соединен связью «одни ко многим» с классом PriceList, так как одна аптека может содержать много лекарств и их стоимость.

Класс Medicine соединен связью «одни ко многим» с классом PriceList, так как одно лекарство может находиться в различных прейскурантах.

# **3 СТРУКТУРА И ОСНОВНЫЕ АЛГОРИТМЫ СОЗДАННОГО ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ**

# **3.1 Структурная схема приложения и ее описание**