Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Проектирование программного обеспечения»

Лабораторная работа №5

«Интерфейс поиска и бронирования лекарств»

Студент: Немкович А. В.

ФИТ 3 курс 1 группа

Преподаватель: Якубенко К. Д

Минск 2024

# 1.Постановка задачи

## 1.1. Функциональное назначение программы

Программа предназначена для предоставления пользователям удобного сервиса по поиску, бронированию и покупке лекарственных средств в аптеках города. Функциональное назначение платформы заключается в упрощении процесса поиска необходимых медикаментов и повышении доступности лекарств для населения.

Для пользователя программа предоставляет возможность быстро находить нужные лекарства и медицинские товары, вводя их название или активное вещество, с доступом к информации о наличии в аптеках города и цене, что позволяет сравнивать предложения и выбирать наиболее выгодные. Программа также предоставляет возможность бронирования медикаментов для самовывоза из выбранной аптеки с отображением контактной информации, адреса и времени работы.

Для зарегистрированного пользователя предусмотрен личный кабинет с функциями управления заказами и бронированиями, просмотра истории покупок, настройки уведомлений и создания списков избранных товаров. Интеграция с картами предоставляет возможность просматривать расположение аптек на карте и строить маршруты до них, а также оставлять и просматривать отзывы о товарах и аптеках, что помогает в выборе.

Для партнёрских аптек программа предоставляет возможность обновлять информацию о товарах, их наличии и ценах, управлять заказами и бронированиями через административную панель. Интеграция с базами данных и системами управления запасами обеспечивает актуальность данных и безопасность транзакций.

## 1.2. Требования к составу выполняемых функций

Любой пользователь интернета может подключиться к сервису для поиска лекарств в аптеках города, набрав правильный URI в адресной строке браузера. По умолчанию он соединяется с сервисом как пользователь с ролью «Гость» (пользователь по умолчанию).

В режиме «Гость» пользователю доступен для ознакомления документ «Соглашение об уровне услуг» и следующие функции:

* Производить поиск лекарств по названию, активному веществу или другим параметрам для получения информации о доступных препаратах в городе.
* Просматривать подробную информацию о каждом найденном лекарстве, включая инструкцию по применению, показания, противопоказания и побочные эффекты.
* Узнавать о наличии лекарств в ближайших аптеках и сравнивать цены, чтобы выбрать наиболее выгодное предложение.
* Просматривать информацию о необходимой аптеке: местоположение, время работы, контактные номера (реквизиты, связь с отделами предприятий, адрес для электронных обращений).
* Получать персонализированную информацию о побочных эффектах, взаимодействии с другими лекарствами, необходимости приема на пустой желудок и других особенностях применения конкретного препарата.
* Получить доступ к справочной информация и отзывам.
* Возможность регистрации и аутентификации для получения роли «Пользователь», «Менеджер» или «Представитель аптеки».

Регистрация осуществляется путем заполнения формы, с указанием основной необходимой информации о пользователе.

После регистрации пользователь автоматически переключается в режим «Пользователь», соответствующий пользовательской роли «Пользователь» (зарегистрированный пользователь).

Зарегистрированному пользователю доступны все возможности пользователя роли «Гость» и дополнительные функции, позволяющие ему:

* Оставлять отзывы о лекарствах, делиться своим опытом и помогать другим пользователям принять обоснованное решение при выборе лекарства.
* Осуществлять бронь необходимого лекарства.
* Добавлять необходимые лекарства в «Избранное», что в будущем позволит пользователю осуществить быстрый поиск.

При подключении пользователя в режиме «Менеджер» ему доступна возможность:

* Управлять ресурсом сервиса, включая добавление, редактирование и удаление информации о лекарствах.
* Получать статистику использования сервиса и предоставлять отчеты администратору сервиса.
* Поддержание обратной связи с пользователем интернет-сервиса.

При подключении пользователя в режиме «Представитель аптеки» ему доступна возможность:

* Управлять ресурсом сервиса, включая добавление, редактирование и удаление информации о лекарствах.
* Вносить изменения в базу данных в связи с изменением количества лекарства в хранилище аптеки.
* Осуществлять бронь лекарства (пользователь оставляет на интернет-сервисе запрос, который в последствии будет обработан и внесен в базу данных).

# 2.Описание программных средств

## 2.1. Описание программного средства Draw.io

Название: draw.io (diagrams.net)

Версия: Последняя доступная версия обновляется разработчиком автоматически, так как draw.io является веб-приложением. Пользователи всегда работают с самой актуальной версией, которая обновляется на сервере. Для локального использования доступны десктопные версии, которые необходимо обновлять вручную.

Разработчик: JGraph Ltd.

Адрес загрузки: https://app.diagrams.net

Режим использования: draw.io — это инструмент для создания диаграмм и схем, доступный как в режиме онлайн через браузер, так и оффлайн с помощью настольного приложения. Приложение часто используется для проектирования бизнес-процессов, архитектуры систем, блок-схем, UML-диаграмм и других визуальных моделей. Для работы онлайн не требуется регистрация, а проекты могут сохраняться на локальном устройстве, в облаке (Google Drive, OneDrive, Dropbox), или в корпоративных системах хранения.

Веб-приложение работает на всех устройствах с современными браузерами, такими как Google Chrome, Mozilla Firefox, Safari и Microsoft Edge. Десктопные версии поддерживаются на Windows, macOS и Linux, предоставляя пользователям гибкость для работы оффлайн. Приложения для мобильных платформ отсутствуют, но сайт может быть доступен через мобильный браузер для просмотра и базового редактирования.

Draw.io поддерживает создание различных типов диаграмм и моделей:

* 2D-векторные диаграммы. Позволяет проектировать блок-схемы, организационные структуры, карты процессов, схемы сетевой инфраструктуры и многое другое.
* UML и технические схемы. Поддерживаются готовые шаблоны и библиотеки для моделирования программного обеспечения, инженерных проектов и электронных схем.

Одной из ключевых особенностей draw.io является интеграция с популярными облачными платформами, такими как Google Drive, Microsoft OneDrive и GitHub, что позволяет совместно работать над схемами и сохранять их в привычной среде. Несмотря на отсутствие полноценного многопользовательского режима в реальном времени, файлы можно быстро обменивать между участниками проекта через совместный доступ в облаке.

# 3. Описание практического задания

Диаграмма классов для нашего проекта будет выглядеть следующим образом – Рисунок 3.1.

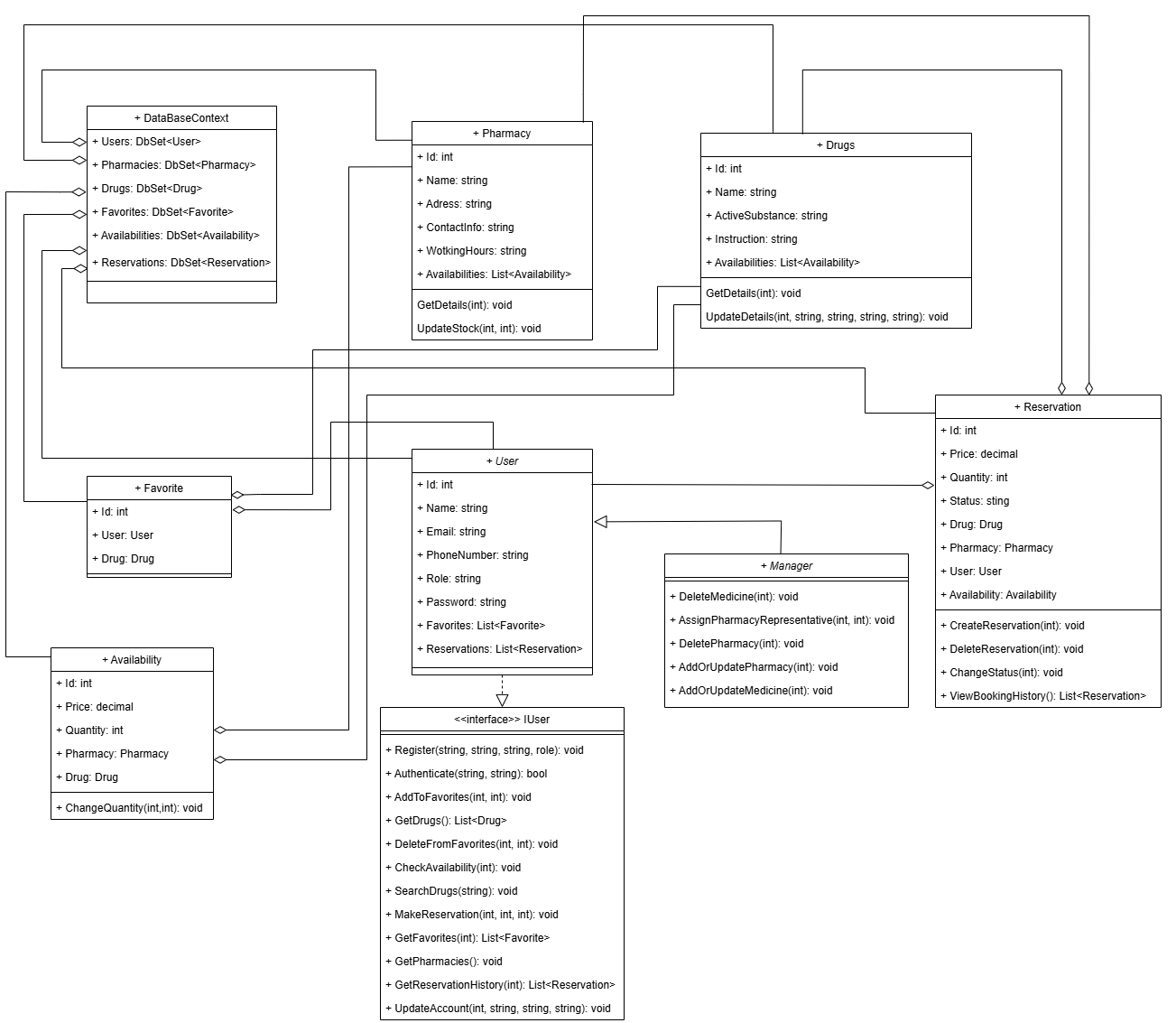


Рисунок 3.1 – Диаграмма классов

Сперва разберём сущности приложения:

* DataBaseContext.
* User.
* Manager.
* Pharmacy.
* Drugs.
* Reservation.
* Favorite.
* Availability.

DataBaseContext представляет собой центральный элемент, обеспечивающий взаимодействие между приложением и базой данных. Этот класс используется для управления всеми основными сущностями системы, предоставляя удобный способ доступа к данным.

Каждое из свойств представляет собой объект DbSet, который используется Entity Framework для выполнения операций CRUD (создание, чтение, обновление и удаление) над соответствующими таблицами базы данных.

"User" (Пользователь). содержит несколько атрибутов, среди которых уникальный идентификатор пользователя (Id), имя (Name), адрес электронной почты (Email), номер телефона (PhoneNumber), роль в системе (Role) и пароль (Password). Также для пользователя предусмотрены два списка: Favorites, который содержит избранные элементы пользователя, такие как товары или услуги, и Reservations, в котором хранится информация о резервациях, сделанных пользователем.

Сущность Manager расширяет функционал пользователя, добавляя возможности управления системой. Включает методы для редактирования данных о лекарствах, мониторинга бронирований и управления статистикой.

Объект системы "Pharmacy" (Аптека) включает несколько атрибутов и методов. Он имеет уникальный идентификатор (Id), название аптеки (Name), адрес (Address), контактную информацию (ContactInfo), а также рабочие часы (Working Hours). Дополнительно в классе хранится список "Availabilities" — доступности товаров, который представляет собой перечень объектов типа "Availability", описывающих наличие товаров в аптеке. В рамках методов предусмотрены следующие действия: метод "GetDetails", который позволяет получить подробную информацию о конкретной аптеке по её идентификатору, и метод "Update Stock", который обновляет данные о запасах товаров, принимая в качестве параметров идентификатор товара и новое количество на складе.

Объект системы "Drugs" (Лекарства) включает несколько атрибутов и методов. У него есть уникальный идентификатор (Id), название лекарства (Name), активное вещество (Active Substance), а также инструкция по применению (Instruction). Класс также содержит список "Availabilities", который отображает доступность данного лекарства, представляя собой перечень объектов типа "Availability", которые содержат информацию о наличии товара в аптеках или других точках продажи. Метод "GetDetails" позволяет получить подробную информацию о конкретном лекарстве по его идентификатору. Метод "Update Details" предназначен для обновления данных лекарства, принимая в качестве параметров идентификатор лекарства и новые значения для его названия, активного вещества, инструкции и других характеристик.

Объект системы "Reservation" включает несколько атрибутов и методов. У него есть уникальный идентификатор (Id), цена (Price), количество (Quantity), а также статус (Status), который описывает текущее состояние резервации (например, активна, отменена, выполнена). Также присутствуют связи с другими объектами системы: "Drug" (лекарство), "Pharmacy" (аптека), "User" (пользователь), и "Availability" (доступность товара), которые связывают резервацию с конкретным лекарством, аптекой, пользователем и данными о наличии товара в аптеке. Методы включают "Create Reservation", который позволяет создать новую резервацию, принимая идентификатор лекарства, "Delete Reservation", предназначенный для удаления резервации по идентификатору, и "ChangeStatus", который изменяет статус резервации. Кроме того, есть метод "View Booking History", который позволяет пользователю просматривать историю своих резерваций, возвращая список всех сделанных резерваций.

Сущность Favorite позволяет пользователям сохранять список избранных лекарств. Поля включают ID (int), UserID (int), и DrugID (int). Это упрощает доступ к часто используемым препаратам и ускоряет процесс поиска.

Объект системы "Availability" (Доступность) описывает информацию о наличии конкретного лекарства в аптеке. Он включает несколько атрибутов: уникальный идентификатор (Id), цену (Price) лекарства в аптеке, количество (Quantity) данного лекарства, доступное для продажи, а также связь с объектами "Pharmacy" (аптека) и "Drug" (лекарство), что позволяет определить, в какой аптеке и какое именно лекарство доступно. Метод "Change Quantity" позволяет изменять количество доступного товара в аптеке, принимая два параметра: идентификатор лекарства и новое количество.

Использовался тип связи — агрегация. Это означает, что один объект может содержать другие объекты как части, но эти части могут существовать независимо от целого. В такой связи объект «целое» состоит из объектов «частей», однако эти части могут существовать вне зависимости от целого, и их жизнь не зависит от существования целого объекта.

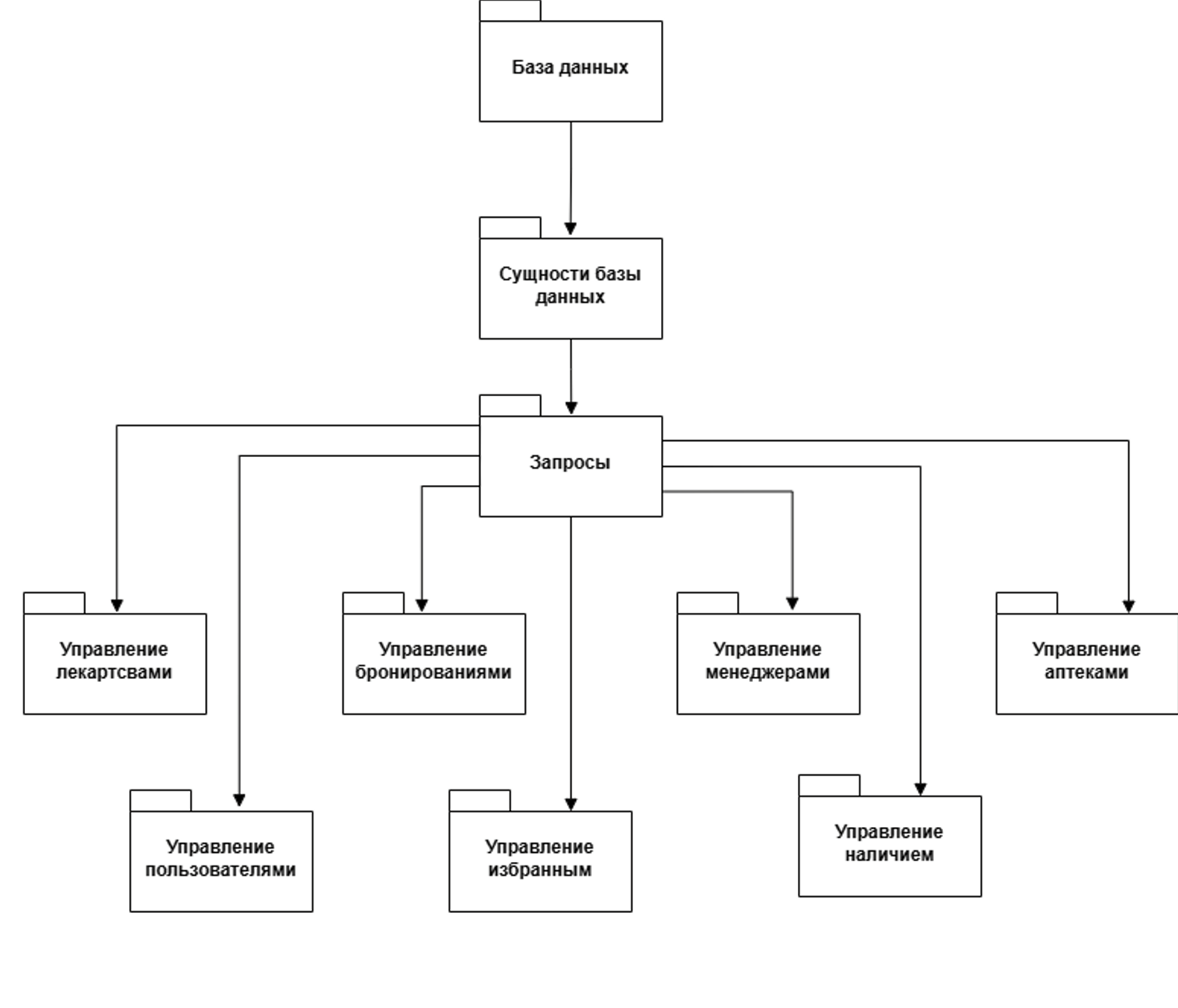


Рисунок 3.2 – Диаграмма пакетов

В рамках разработки системы используются различные пакеты, которые обеспечивают функциональность взаимодействия с базой данных и управления различными аспектами системы. Описание пакетов и их сущностей приведено ниже:

База данных – этот пакет включает взаимодействие с основной базой данных системы. Он отвечает за подключение к базе данных, выполнение запросов и управление данными, такими как добавление, удаление и обновление записей. В нем могут быть реализованы функции для создания, изменения и удаления таблиц и индексов, а также для управления транзакциями и обеспечивания целостности данных.

Сущности базы данных – этот пакет включает классы или объекты, которые соответствуют таблицам в базе данных. Каждая сущность представляет собой модель для хранения информации о пользователях, лекарствах, аптеках, резервациях и других данных.

Запросы – этот пакет содержит логику для создания, выполнения и обработки SQL-запросов, которые необходимы для извлечения данных из базы данных или внесения изменений в неё. Здесь могут быть реализованы функции для поиска, фильтрации и сортировки данных, а также для выполнения агрегированных операций, таких как подсчет количества товаров или суммирование цен.

Управление лекарствами – пакет, отвечающий за все операции, связанные с лекарствами. Он включает функции для добавления новых лекарств в систему, их обновления, удаления и поиска.

Управление бронированиями – этот пакет предназначен для работы с резервациями. Он включает в себя логику для создания новых бронирований, их изменения, удаления и просмотра.

Управление менеджерами – пакет, который управляет данными о менеджерах системы. Он включает функции для добавления новых менеджеров, их обновления, удаления и авторизации.

Управление аптеками: Пакет, который отвечает за работу с данными о аптеках. Включает функции для добавления новых аптек, их обновления, удаления, а также управление информацией об их рабочем времени, контактных данных и расположении.

Управление пользователями – пакет, который управляет всеми операциями, связанными с пользователями. Он включает функции для регистрации, авторизации, обновления данных пользователей, таких как имя, email, телефон и пароль.

Управление избранным – пакет, отвечающий за операции с избранными товарами пользователей. Он позволяет добавлять товары в список избранного, удалять их, а также управлять доступом к этому списку. Этот пакет может включать функциональность для отображения списка избранных товаров пользователю и для их удаления.

Управление наличием – пакет, который отвечает за управление информацией о наличии товаров в аптеках. Он включает в себя логику для обновления данных о запасах лекарств, отображения текущей доступности и управления количеством товаров в аптеке.

# 4.Ответы на вопросы

## 4.1. Перечислите структурные диаграммы, которые входят в UML 2.0.

Следующие диаграммы входят в UML 2.0:

* **Диаграмма классов (Class Diagram).** Диаграмма классов используется для моделирования структуры системы, отображая классы, их атрибуты, методы и связи между ними, помогая в проектировании архитектуры системы.
* **Диаграмма компонентов (Component Diagram).** Диаграмма компонентов моделирует высокоуровневые архитектурные модули системы и их взаимодействия, отображая, как компоненты системы взаимодействуют друг с другом.
* **Диаграмма объектов (Object Diagram).** Диаграмма объектов отображает конкретные экземпляры классов и их взаимосвязи на определенный момент времени, демонстрируя фактическое состояние системы.
* **Диаграмма развертывания (Deployment Diagram).** Диаграмма развертывания отображает физическую архитектуру системы, показывая, как программные компоненты развертываются на физическом оборудовании и связаны между собой.
* **Диаграмма пакетов (Package Diagram).** Диаграмма пакетов используется для группировки связанных классов или компонентов в логические единицы (пакеты), облегчая понимание структуры и взаимодействия больших систем.
* **Диаграмма структуры (Composite Structure Diagram).** Диаграмма структуры используется для отображения внутренней структуры класса или компонента, показывая его части и взаимодействия между ними в рамках более сложной структуры.

## 4.2. Укажите назначение структурных диаграммы.

Структурные диаграммы в UML 2.0 используются для отображения статической структуры системы. Эти диаграммы помогают моделировать и визуализировать основные компоненты системы, их взаимосвязи и зависимость друг от друга. Они показывают, как различные части системы (классы, компоненты, объекты и пакеты) взаимодействуют и как они организованы на высоком уровне. Структурные диаграммы полезны для проектирования архитектуры системы, разработки баз данных и других статичных аспектов системы.

## 4.3. Опишите нотации, которые используются для построения Classes диаграмм.

Классы изображаются прямоугольниками, разделенными на три части.

Первая часть: название класса, обычно с использованием верхнего регистра. Вторая часть: атрибуты класса, которые могут включать их тип и видимость (например, + для публичных, - для приватных). Третья часть: методы класса с их параметрами и видимостью.

Ассоциации между классами показываются как линии, соединяющие их, с указанием кратности (например, 1..\* для одного ко многим), а также стрелки, если ассоциация направленная.

Наследование или обобщение отображается с помощью линии с треугольной стрелкой, направленной от подкласса к суперклассу.

Интерфейсы обозначаются как круги или полукруги, а связь с классами, которые реализуют интерфейс, — как линию с пустым треугольником.

Агрегация и композиция показываются линией с ромбом на конце, где ромб указывает на целый объект (например, объект, в котором содержится другой объект).

## 4.4. Для чего применяются расширения диаграмм UML?

Расширения диаграмм UML используются для добавления дополнительной функциональности и возможностей в стандартный набор диаграмм UML. Они позволяют детализировать элементы диаграмм, добавлять метки, стереотипы и другие конструкты, которые не предусмотрены в базовой спецификации UML, но важны для конкретных типов систем или областей применения. Расширения позволяют адаптировать UML под специфические нужды пользователей, таких как бизнес-процесс, управление проектами, моделирование баз данных, интеграция с другими методологиями и инструментами.

## 4.5. Что означают понятия «стереотип» и «тегированное значение» в контексте расширенных диаграмм?

Стереотип в UML — это механизм расширения, который позволяет создавать новые элементы или классы элементов, основываясь на существующих. Стереотипы обозначаются на диаграммах с помощью угловых скобок, например, «<<interface>>» или «<<entity>>». Они добавляют дополнительное значение или контекст для элементов диаграммы, предоставляя пользователям возможность специфицировать их роль или характер в системе.

Тегированное значение — это пара "тег-значение", которая используется для добавления дополнительной информации к элементу диаграммы. Теги позволяют задавать значения для атрибутов или параметров в элементах UML, например, указывая свойства или настройки объекта, которые могут быть уникальными для конкретного случая. Тегированные значения могут быть присоединены к стереотипам для еще большей кастомизации и детализированного описания элементов диаграмм.