

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра информационных технологий

УТВЕРЖДАЮ

Заведующий кафедрой
информационных технологий
_____ Ю.Н. Орлов

«___» _____ 2021 г.

КУРСОВАЯ РАБОТА

на тему

«Разработка модели информационной системы Интернет-магазина на основе UML»

по дисциплине «Моделирование сложно структурированных систем»

Выполнил
Студент группы НПИбд-01-18
Студенческий билет №: 1032162043
Еременко Артем Геннадьевич
«___» _____ 2021 г.

Научный руководитель
Доцент кафедры
Информационных технологий,
к.ф.-м.н., М. В. Хачумов

Научный консультант
Доцент кафедры
Информационных технологий,
к.ф.-м.н., М. В. Хачумов

Москва 2021

Содержание

Аннотация	3
Введение	3
Постановка задачи	4
Техническое задание	4
Обзор языка UML	5
Строительные блоки UML	5
Сущности	5
Отношения	10
Диаграммы UML	12
Общие механизмы UML.....	14
Диаграмма вариантов использования.....	16
Диаграмма классов	17
Заключение.....	17
Список используемой литературы.....	17

Аннотация

В данной курсовой работе описывается техническое задание по разработке модели информационной системы Интернет-магазина на основе UML. Так же автор делает обзор на язык UML: изучает какие строительные блоки в нём присутствуют, и как их следует применять; анализирует общие механизмы действия. Далее автор приводит результат выполнения практической части в виде построения двух диаграмм по модели информационной системы Интернет-магазина: диаграмму использования и диаграмму классов.

Введение

UML (Unified Modeling Language) – это унифицированный графический язык моделирования для описания, визуализации, проектирования и документирования ОО (объектно-ориентированных) систем. UML призван поддерживать процесс моделирования ПС (программных систем) на основе ОО подхода, организовывать взаимосвязь концептуальных и программных понятий, отражать проблемы масштабирования сложных систем. Модели на UML используются на всех этапах жизненного цикла ПС, начиная с бизнес-анализа и заканчивая сопровождением системы. Разные организации могут применять UML по своему усмотрению в зависимости от своих проблемных областей и используемых технологий.

К середине 90-х годов различными авторами было предложено несколько десятков методов ОО моделирования, каждый из которых использовал свою графическую нотацию. При этом любой из этих методов имел свои сильные стороны, но не позволял построить достаточно полную модель ПС, показать ее «со всех сторон», то есть, все необходимые проекции. К тому же отсутствие стандарта ОО моделирования затрудняло для разработчиков выбор наиболее подходящего метода, что препятствовало широкому распространению ОО подхода к разработке ПС.

По запросу OMG (Object Management Group) - организации, ответственной за принятие стандартов в области объектных технологий и баз данных назревшая проблема унификации и стандартизации была решена авторами трех наиболее популярных ОО методов - Г.Бучем, Д.Рамбо и А.Джекобсоном, которые объединенными усилиями создали версию UML 1.1, утвержденную OMG в 1997 году в качестве стандарта.

Постановка задачи

Разработать модель информационной системы Интернет-магазина на основе UML.

Составить техническое задание.

Сделать обзор UML.

Техническое задание

Информационная система предназначена для продажи товаров, обработки информации о пользователях и предоставления информации о товарах.

Система должна предоставлять посетителю следующие возможности:

1. Управление корзиной
 - a. Поиск товара по каталогу
 - b. Добавление товара в корзину
 - c. Манипуляции с товаром в корзине
2. Просмотр информации об организации
3. Служба поддержки

Система должна предоставлять покупателю следующие возможности:

1. Регистрация пользователя
2. Выбор метода оплаты
 - a. Картой онлайн
 - b. Платёжный сервис
 - c. Наличными при получении
4. Применение бонусных баллов или промо-кодов
5. Согласовать детали заказа

Система должна предоставлять администратору следующие возможности:

1. Управление товаром
 - a. Добавить позицию
 - b. Удалить позицию
 - c. Изменить информацию

2. Управление пользователями
 - a. Добавить пользователя
 - b. Удалить пользователя
 - c. Редактировать информацию
3. Связь с клиентом
4. Связь с менеджером

Система должна предоставлять менеджеру следующие возможности:

1. Связь с клиентом
2. Связь с администратором
3. Управление заказом
 - a. Согласовать детали заказа
 - b. Сформировать документы

Обзор языка UML

Строительные блоки UML

Словарь языка UML включает три вида строительных блоков:

- сущности;
- отношения;
- диаграммы.

Сущности

Сущности – это абстракции, являющиеся основными элементами модели. Они являются наиболее важными строительными блоками UML. В UML имеется четыре типа сущностей:

- структурные;
- поведенческие;
- группирующие;
- аннотационные.

Структурные сущности

Структурные сущности определяют статическую часть модели. Они представляют физические и концептуальные элементы. Ниже приведены краткие описания структурных сущностей.

Класс — это описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Класс реализует один или несколько интерфейсов. Графически класс изображается в виде прямоугольника, в котором обычно записаны его имя, атрибуты и операции, как показано на рис. 1.

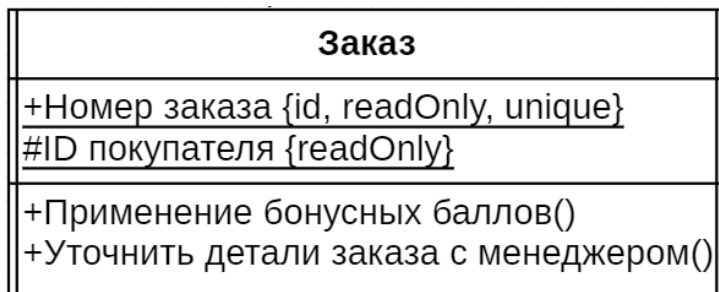


Рис. 1. Класс

Интерфейс — это совокупность операций, которые определяют сервис (набор услуг), предоставляемый классом или компонентом. Таким образом, интерфейс описывает видимое извне поведение элемента. Интерфейс может представлять поведение класса или компонента полностью или частично; он определяет только спецификации операций (сигнатуры), но никогда - их реализации. Графически интерфейс изображается в виде круга, под которым пишется его имя, как показано на рис. 2. Интерфейс редко существует сам по себе - обычно он присоединяется к реализующему его классу или компоненту.

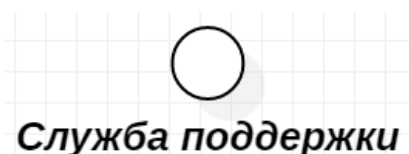


Рис. 2. Интерфейс

Кооперация — определяет взаимодействие; она представляет собой совокупность ролей и других элементов, которые, работая совместно, производят некоторый кооперативный эффект, не сводящийся к простой сумме слагаемых. Кооперация, следовательно, имеет как структурный, так и поведенческий аспект. Один и тот же класс может принимать участие в нескольких кооперациях; таким образом, они являются реализацией образцов поведения, формирующих систему. Графически кооперация изображается в виде эллипса, ограниченного пунктирной линией, в который обычно заключено только имя, как показано на рис. 3.

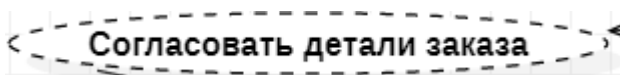


Рис. 3. Кооперация

Вариант использования — это описание последовательности выполняемых системой действий, которая производит наблюдаемый результат, значимый для какого-то определенного актера (Actor). Прецедент применяется для структурирования поведенческих сущностей модели. Прецеденты реализуются посредством кооперации. Графически прецедент изображается в виде ограниченного непрерывной линией эллипса, обычно содержащего только его имя как показано на рис. 4.

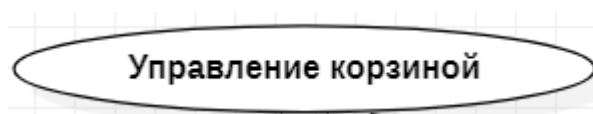


Рис. 4. Вариант использования

Компонент — это физическая заменяемая часть системы, которая соответствует некоторому набору интерфейсов и обеспечивает его реализацию. В системе можно встретить различные виды устанавливаемых компонентов, а также компоненты, являющиеся артефактами процесса разработки, например файлы исходного кода. Компонент, как правило, представляет собой физическую упаковку логических элементов, таких как классы, интерфейсы и кооперации. Графически компонент изображается в виде прямоугольника с вкладками, содержащего обычно только имя, как показано на рис. 5.

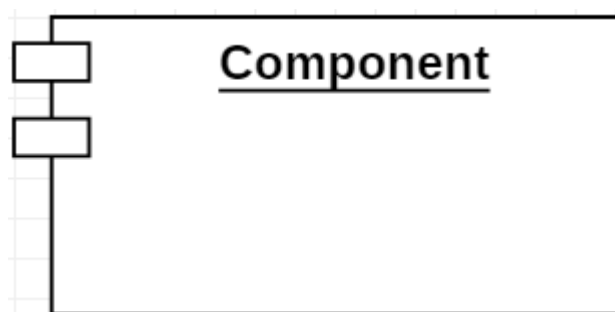


Рис. 5. Компонент

Узел — это элемент реальной (физической) системы, который существует во время функционирования программного комплекса и представляет собой вычислительный ресурс, обычно обладающий как минимум некоторым объемом памяти, а часто еще и способностью обработки. Совокупность компонентов может размещаться в узле, а также мигрировать с

одного узла на другой. Графически узел изображается в виде куба, обычно содержащего только имя, как показано на рис. 6.

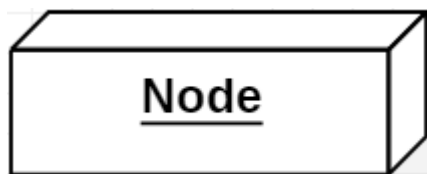


Рис. 6. Узел

Поведенческие сущности

Поведенческие сущности состоят из динамических частей моделей UML. Это глаголы языка: они описывают поведение модели во времени и пространстве. Существует всего два основных типа поведенческих сущностей.

Взаимодействие — это поведение, суть которого заключается в обмене сообщениями (Messages) между объектами в рамках конкретного контекста для достижения определенной цели. С помощью взаимодействия можно описать как отдельную операцию, так и поведение совокупности объектов. Взаимодействие предполагает ряд других элементов, таких как сообщения, последовательности действий (поведение, инициированное сообщением) и связи (между объектами). Графически сообщения изображаются в виде стрелки, над которой почти всегда пишется имя соответствующей операции, как показано на рис. 7.

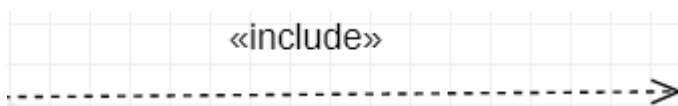


Рис. 7. Взаимодействие

Конечный автомат — это алгоритм поведения, определяющий последовательность состояний, через которые объект или взаимодействие проходят на протяжении своего жизненного цикла в ответ на различные события, а также реакции на эти события. С помощью автомата можно описать поведение сдельного класса или кооперации классов. С автоматом связан ряд других элементов: состояния, переходы (из одного состояния в другое), события (сущности, инициирующие переходы) и виды действий (реакция на переход). Графически состояние изображается в виде прямоугольника с закругленными углами, содержащего имя и, возможно, подсостояния (см. рис. 8).



Рис. 8. Состояние

Группирующие сущности

Группирующие сущности являются организующими частями модели UML. Это блоки, на которые можно разложить модель. Есть только одна первичная группирующая сущность, а именно пакет.

Пакеты — представляют собой универсальный механизм организации элементов в группы. В пакет можно поместить структурные, поведенческие и даже другие группирующие сущности. В отличие от компонентов, существующих во время работы программы, пакеты носят чисто концептуальный характер, то есть существуют только во время разработки. Изображается пакет в виде папки с закладкой, содержащей, как правило, только имя и иногда - содержимое (см. рис. 9).



Рис. 9. Пакет

Аннотационные сущности

Аннотационные сущности могут быть определены как механизм для сбора замечаний, описаний и комментариев элементов модели UML. Имеется только один базовый тип аннотационных элементов – примечание.

Примечание — это просто символ для изображения комментариев или ограничений, присоединенных к элементу или группе элементов. Графически примечание изображается в виде прямоугольника с загнутым краем, содержащим текстовый или графический комментарий, как показано на рис. 10.

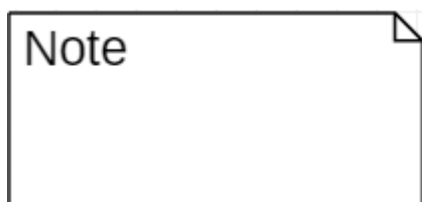


Рис. 10. Примечание

Отношения

Отношения — это еще один важнейший строительный блок UML. Он показывает, как элементы связаны друг с другом, и эта связь описывает функциональность приложения.

Есть четыре вида доступных отношений:

- зависимость;
- ассоциация;
- обобщение;
- реализация.

Зависимость

Зависимость — это семантическое отношение между двумя сущностями, при котором изменение одной из них, независимой, может повлиять на семантику другой, зависимой. Графически зависимость изображается в виде прямой пунктирной линии, часто со стрелкой, которая может содержать метку (см. рис. 11).



Рис. 11. Зависимость

Ассоциация

Ассоциация — структурное отношение, описывающее совокупность связей; связь - это соединение между объектами. Разновидностью ассоциации является агрегирование (Aggregation) - так называют структурное отношение между целым и его частями. Графически ассоциация изображается в виде прямой линии (иногда завершающейся стрелкой или содержащей метку), рядом с которой могут присутствовать дополнительные обозначения, на пример кратность и имена ролей. На рис. 12 показан пример отношений этого типа.



Рис. 12. Ассоциация

Обобщение

Обобщение — это отношение "специализация/обобщение", при котором объект специализированного элемента (потомок) может быть подставлен вместо объекта обобщенного элемента (родителя или предка). Таким образом, потомок (Child) наследует структуру и поведение своего родителя (Parent). Графически отношение обобщения изображается в виде линии с не закрашенной стрелкой, указывающей на родителя, как показано на рис. 13.



Рис. 13. Обобщение

Реализация

Реализация — это семантическое отношение между классификаторами, при котором один классификатор определяет "контракт", а другой гарантирует его выполнение. Отношения реализации встречаются в двух случаях: во-первых, между интерфейсами и реализующими их классами или компонентами, а во-вторых, между прецедентами и реализующими их кооперациями. Отношение реализации изображается в виде пунктирной линии с не закрашенной стрелкой, как нечто среднее между отношениями обобщения и зависимости (см. рис. 14).

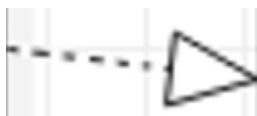


Рис. 14. Реализация

Отношения связывают различные сущности; диаграммы группируют представляющие интерес совокупности сущностей.

Диаграммы UML

Диаграмма — это графическое представление набора элементов, изображаемое чаще всего в виде связанного графа с вершинами (сущностями) и ребрами (отношениями). Диаграммы рисуют для визуализации системы с разных точек зрения. Диаграмма - в некотором смысле одна из проекций системы. Как правило, за исключением наиболее тривиальных случаев, диаграммы дают свернутое представление элементов, из которых составлена система. Один и тот же элемент может присутствовать во всех диаграммах, или только в нескольких (самый распространенный вариант), или не присутствовать ни в одной (очень редко). Теоретически диаграммы могут содержать любые комбинации сущностей и отношений. На практике, однако, применяется сравнительно небольшое количество типовых комбинаций, соответствующих пяти наиболее употребительным видам, которые составляют архитектуру программной системы. Таким образом, в UML выделяют восемь типов диаграмм, которые разделяются на два глобальных типа:

- Структурные диаграммы;
- Диаграммы поведения.

Структурные диаграммы

- **Классов** — показывают классы, интерфейсы, объекты и кооперации, а также их отношения. При моделировании объектно-ориентированных систем этот тип диаграмм используют чаще всего. Диаграммы классов соответствуют статическому виду системы с точки зрения проектирования. Диаграммы классов, которые включают активные классы, соответствуют статическому виду системы с точки зрения процессов.
- **Объектов** — представлены объекты и отношения между ними. Они являются статическими "фотографиями" экземпляров сущностей, показанных на диаграммах классов. Диаграммы объектов, как и диаграммы классов, относятся к статическому виду системы с точки зрения проектирования или процессов, но с расчетом на настоящую или макетную реализацию.
- **Компонентов** — представлена организация совокупности компонентов и существующие между ними зависимости. Диаграммы компонентов относятся к статическому виду системы с точки зрения реализации. Они могут быть соотнесены с

диаграммами классов, так как компонент обычно отображается на один или несколько классов, интерфейсов или коопераций.

- **Размещения** — представлена конфигурация обрабатывающих узлов системы и размещенных в них компонентов. Диаграммы развертывания относятся к статическому виду архитектуры системы с точки зрения развертывания. Они связаны с диаграммами компонентов, поскольку в узле обычно размещаются один или несколько компонентов.

Диаграммы поведения

- **Последовательности** — отражают временную упорядоченность сообщений. Является частным случаем диаграммы взаимодействия, на которой представлены связи между объектами; показаны, в частности, сообщения, которыми объекты могут обмениваться. Может быть преобразована в диаграмму коммуникации.
- **Коммуникации** — отражает структурную организацию обменивающихся сообщениями объектов. Также является частным случаем диаграммы взаимодействия. Может быть преобразована в диаграмму последовательности.
- **Автомата / состояний** — на диаграммах состояний представлен автомат, включающий в себя состояния, переходы, события и виды действий. Диаграммы состояний относятся к динамическому виду системы; особенно они важны при моделировании поведения интерфейса, класса или кооперации. Они акцентируют внимание на поведении объекта, зависящем от последовательности событий, что очень полезно для моделирования реактивных систем.
- **Деятельности** — это частный случай диаграммы состояний; на ней представлены переходы потока управления от одной деятельности к другой внутри системы. Диаграммы деятельности относятся к динамическому виду системы; они наиболее важны при моделировании ее функционирования и отражают поток управления между объектами.

Общие механизмы UML

Общие механизмы UML последовательно применяются по всему языку моделирования. Всего выделяют четыре общих механизма:

- спецификации (описание «заднего плана» модели);
- дополнения (возможности дополненного описания любого символа UML);
- принятые деления (описывают конкретные способы представления объектов реального мира в модели);
- механизмы расширения (применяются в случае, когда базовые возможности UML не удовлетворяют выдвигаемым требованиям).

Спецификации — это текстовое описание модели, которое отражает ее суть. Любая модель может быть представлена как графически, так и в текстовом виде — в виде спецификаций. Спецификации также используются для удобства моделирования, поскольку ряд элементов модели может присутствовать в спецификации и отсутствовать на диаграмме.

UML не сводится к диаграммам. Напротив, диаграммы служат для визуального отражения модели интеллектуальной системы. Соответственно, одних только графических диаграмм недостаточно для описания модели. Любая диаграмма должна содержать спецификации, благодаря которым она и оказывается вплетенной в общую модель. Только диаграмма, обладающая исчерпывающей спецификацией, может рассматриваться как по-настоящему полезная.

Дополнения используются в UML для того, чтобы представить какой-либо элемент диаграмм с необходимой полнотой. Визуализация как таковая не всегда достаточна, и в этом случае востребованными оказываются дополнения.

Дополнения нужны, чтобы отразить на диаграмме важные характеристики модели, которые не представляется возможным показать при помощи визуальных методов (например, из-за перегруженности диаграммы). Как правило, дополнения изображаются в форме прямоугольника, который содержит информацию об элементе модели.

Другой важный общий механизм UML — это принятые деления. Во-первых, следует рассмотреть деление «Классификатор — Экземпляр». В качестве классификатора понимается

абстрактное понятие (например, сотрудник предприятия). Соответственно, экземпляром является конкретный объект (например, «экземпляр» маркетолог Иванов Иван Иванович).

Второе принятое деление — это «Интерфейс — Реализация». Это деление позволяет отделить что выполняется, от того, как это выполняется. Например, в случае самолета приборная панель и штурвал будут рассматриваться в качестве интерфейса, тогда как устройство самолета будет реализацией. Как правило, конкретные виды реализации обеспечивают существование того или иного интерфейса, но возможна и обратная ситуация.

И, наконец, следует сказать несколько слов о механизмах расширения. Выделяют три основных механизма:

- ограничения (расширяют семантику элемента, позволяя тем самым добавлять новые правила или изменять старые);
- стереотипы (возможность определять новые элементы модели на основании существующих);
- теговые величины (возможность добавлять новую специальную информацию в спецификации элемента).

Диаграмма вариантов использования

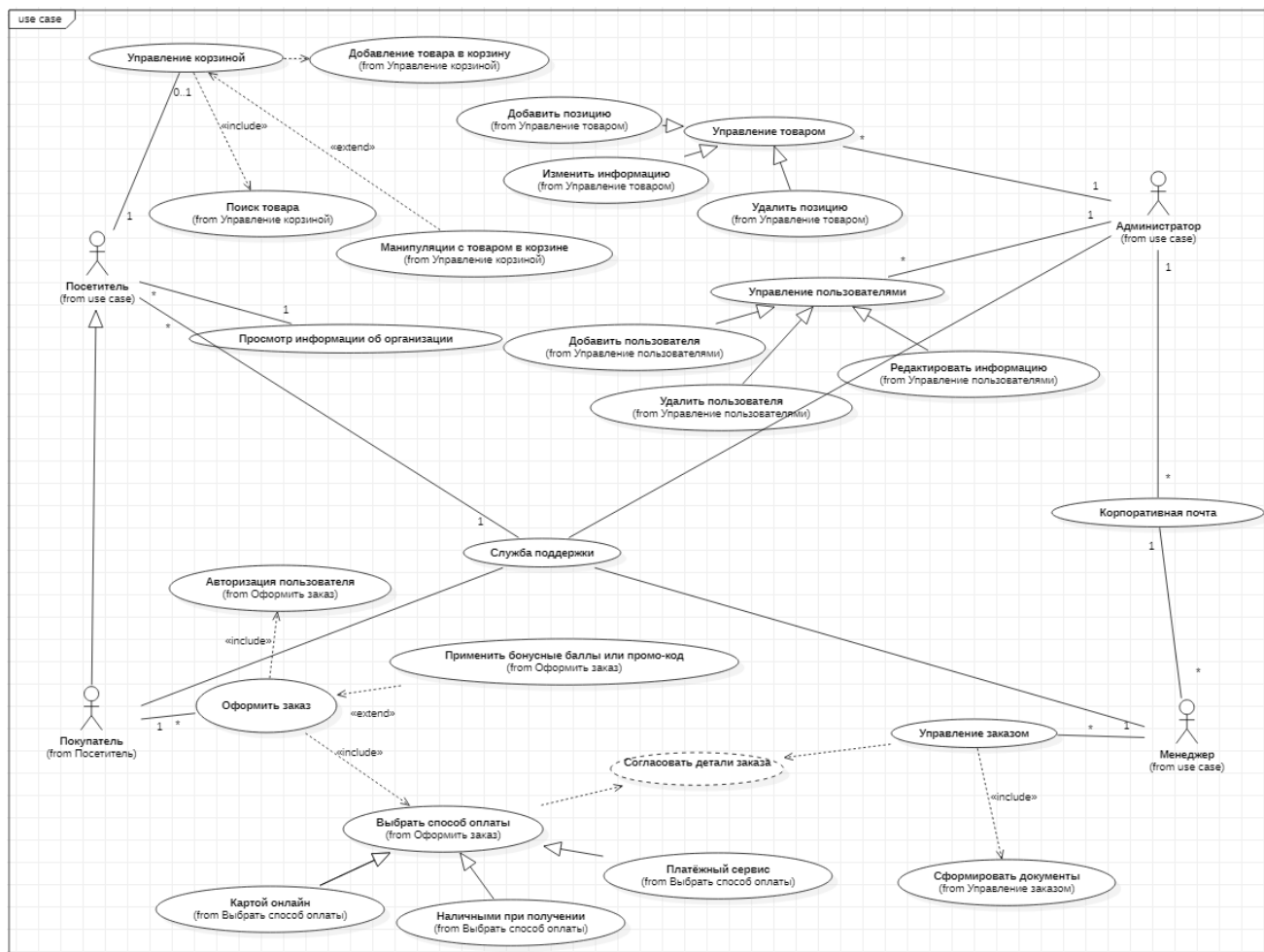
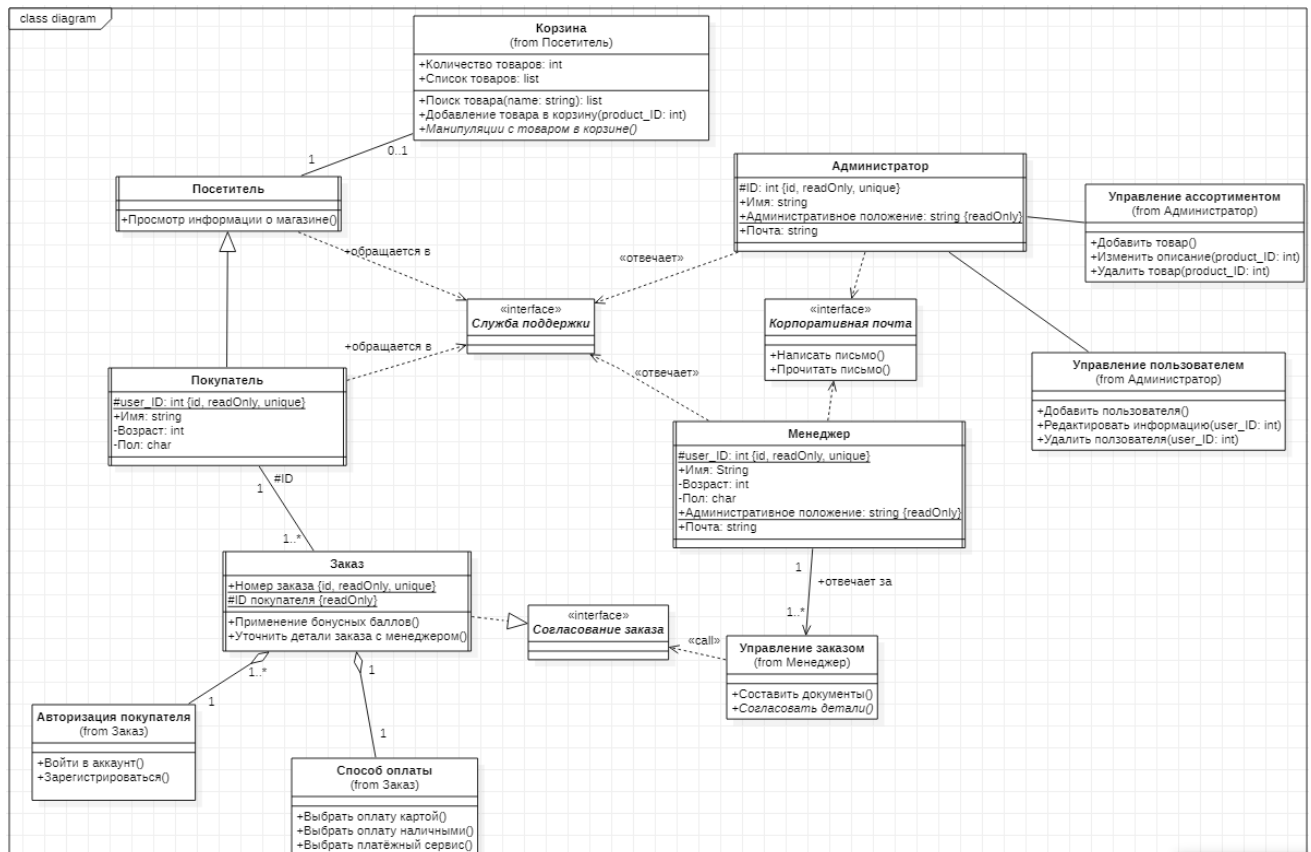


Диаграмма классов



Заключение

В ходе проделанной работы была разработана модель информационной системы Интернет-магазина на основе UML. Было составлено техническое задание. Также был проведён обзор строительных блоков UML и проанализированы общие механизмы их действия. Была выполнена практическая часть в виде построения двух диаграмм по модели информационной системы Интернет-магазина. В неё входит диаграмма использования и диаграмма классов.

Список используемой литературы

1. Мартин Фаулер. UML. Основы. Краткое руководство по стандартному языку объектного моделирования, третье издание. – М.: Символ-Плюс, 2013. – 192 с.
2. Леоненков, А. Нотация и семантика языка UML / А. Леоненков. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 205 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-94774-408-2; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429143>