Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Проектирование программного обеспечения»

Отчёт по лабораторной работе №5

Студент: Дрозд А. И.

ФИТ 3 курс 2 группа

Преподаватель: Якунович А. В.

Минск 2023

**Лабораторная работа №5**

**Тема: «**Объектно-ориентированное моделирование. Структурные диаграммы UML.»

**Цель:** Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования функциональности информационной системы с применением UML.

**Постановка задачи (описание функциональных требований):**

* Управление турами и путевками: добавление новых туров с указанием места назначения, описания, цены, дат начала и окончания; редактирование и удаление существующих туров; просмотр списка доступных туров.
* Отчеты и аналитика: генерация отчетов о продажах, бронированиях и финансовых показателях; создание дашбордов и статистики продаж.
* Интерфейс для пользователей: создание учетных записей для сотрудников с разными уровнями доступа (администраторы, менеджеры склада и операторы).
* Выдача путевок: предоставление туристических путевок клиентам после завершения оплаты.
* Нормативы и правила: контроль соответствия этапов оформления путевки различным законодательным актам, ведомственным инструкциям, должностным нормативам и правилам.

**Описание программных средств:**

* Название приложения: Draw.io
* Версия: 21.8.2
* Разработчик: Jgraph
* Адрес загрузки: https://app.diagrams.net/
* Режим использования: веб-сервис
* Доступность на платформах: поддерживается на всех популярных веб-браузерах (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).

**Описание практического задания**

Диаграмма классов на унифицированном языке моделирования (UML) — это диаграмма статической структуры, которая описывает структуру системы, показывая ее классы, их атрибуты, операции (или методы) и отношения между объектами. Диаграмма классов — это план системы или подсистемы. Диаграммы классов можно использовать для моделирования объектов, составляющих систему, для демонстрации отношений между объектами и для описания ролей этих объектов и предоставляемых ими услуг.

Диаграммы классов полезны на многих этапах проектирования системы. На этапе анализа диаграммы классов могут помочь вам понять требования предметной области и определить ее компоненты. В проектах объектно-ориентированного программного обеспечения диаграмма классов, созданная на ранних стадиях проекта, содержит классы, которые часто преобразуются в реальные классы и объекты программного обеспечения при написании кода.

Позднее вы можете преобразовать ранний анализ и концептуальные модели в диаграммы классов, чтобы показать определенные части системы, пользовательские интерфейсы, логические реализации и т. д.

Диаграммы классов широко используются при моделировании объектно-ориентированных систем, поскольку они являются единственными диаграммами UML, которые могут быть отображены непосредственно на объектно-ориентированные языки. На этапе реализации цикла разработки программного обеспечения вы можете использовать диаграммы классов для преобразования моделей в код и кода в модели.

Диаграмма пакетов – это вид диаграммы в языке моделирования UML (Unified Modeling Language), который предназначен для визуализации организации и структуры системы или приложения. Эта диаграмма позволяет разделить систему на логические группы, называемые пакетами, и показать зависимости между этими пакетами.

Элементами диаграммы пакетов могут быть пакеты, классы, интерфейсы, а также зависимости между ними. Зависимости могут быть направленными или двунаправленными и могут представлять собой различные типы отношений, такие как использование, наследование, ассоциации и т. д.

Диаграммы пакетов часто используются на этапах архитектурного проектирования для визуализации и обсуждения структуры системы перед написанием кода.

Диаграмма классов и диаграмма пакетов изображены на рисунке 1:

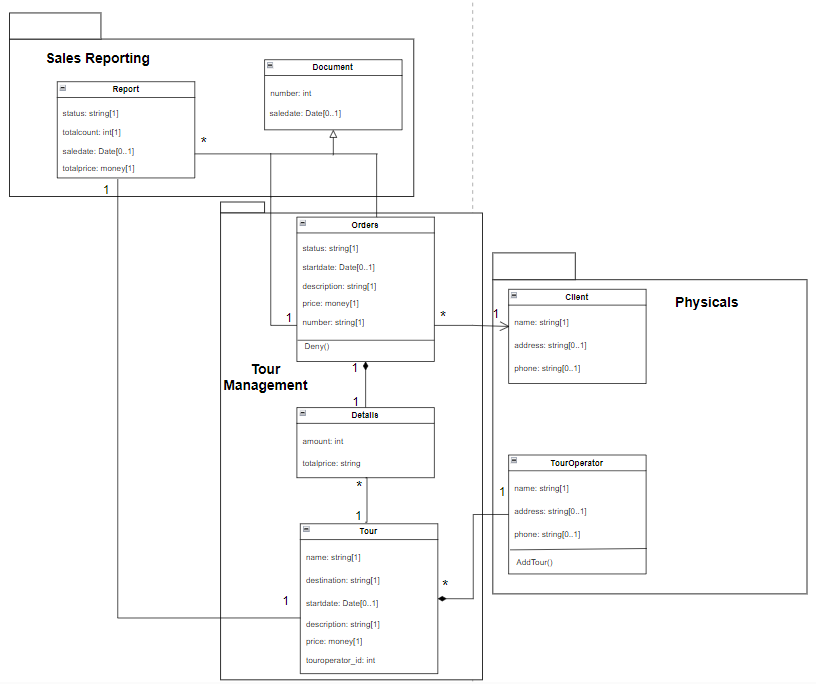


Рисунок 1 – Диаграмма пакетов/классов

Пакет «Sales Reporting» отвечает за запись транзакций путевок и включает в себя:

Класс «Report» состоит из полей: status (статус заказа), totalcount (общее количество путевок), saledate (дата продажи), totalprice (итоговая сумма заказа).

Класс «Document» состоит из полей: number (номер), saledate (дата оформления документа продажи). Является связующим для классов Orders и Report.

Пакет «Physicals» включает в себя информацию о клиентах и туристических операторах:

Класс «Client» хранит информацию о клиентах и состоит из полей: name (наименование туристического агентства), address (адрес главного офиса), phone (контактный номер телефона). Имеет связь один ко многим с классом Orders.

Класс «TourOperator» хранит информацию о туристических операторах и состоит из полей: name (ФИО клиента), address (адрес), phone (контактный номер телефона). Имеет связь один ко многим и композицию с классом «Tour». Это обусловлено тем, что без туристическим операторов некому будет предоставлять туры. Содержит метод «AddTour()» для добавления нового тура.

Пакет «Tour Management» отвечает за управление турами на сервисе:

Класс «Tour» хранит информацию о туре и содержит поля: name (наименование тура), destination (место назначения тура), startdate (дата начала тура), description (описание тура), price (стоимость тура). Установлена свзяь один ко многим с классом Details, а также связь один к одному с классом Report.

Класс «Details» хранит информацию о деталях заказа и имеет поля: amont (общее количество путевок) и totalprice (итоговая стоимость заказа). Имеет связь один к одному с классом Orders и композицию с тем же классом Orders. Композиция – это агрегация, при которой объекты-части не могут существовать сами по себе и уничтожаются при уничтожении объекта агрегирующего класса.

Класс «Orders» состоит из полей: status(статус заказа), startdate(дата тура), description (описание тура), price(стоимость за 1 тур), number(номер). Имеет свзяь один ко многим с классом Invoice. В связи с тем, что Orders и Report, в сущности, являются документами и имеют сходные атрибуты, они были объединены с помощью общего класса-предка Document. Содержит метод «Deny()» для отмены заказа.

**Ответы на теоретические вопросы**

1. **Перечислите структурные диаграммы, которые входят в UML 2.0.**

Новая версия UML 2.0, разработанная ассоциацией OMG в 2005 г., отличается от UML 1.0 большей степенью формализации в связи с необходимостью автоматизации перевода описания на UML в машинный код.

Все расширения языка основаны на получении новых модулей путем вложения базовых модулей более низкого уровня в создаваемые модули более высоких уровней.

В UML 2.0 введено 13 типов диаграмм, разделенных на два подмножества — структурных и поведенческих диаграмм.

Структурные модели, известные также как статические модели, описывают структуру сущностей или компонентов некоторой системы, включая их классы, интерфейсы, атрибуты и отношения. К структурным относятся следующие диаграммы:

* диаграммы пакетов или контейнеров (Package diagrams);
* диаграммы классов (Class or Structural diagrams);
* диаграммы объектов (Object diagrams);
* композитные диаграммы (Composite Structure);
* диаграммы компонентов (Component diagrams), показывающие образование структур из классов и интерфейсы между структурами;
* диаграммы развертывания (Deployment diagrams).

1. **Укажите назначение структурных диаграммы.**

Основное назначение структурных диаграмм заключается в графическом представлении состава статистических совокупностей, характеризующихся как соотношение различных частей каждой из совокупностей. Состав статистической совокупности графически может быть представлен с помощью как абсолютных, так и относительных показателей.

1. **Опишите нотации, которые используются для построения Classes диаграмм.**

Существует несколько обозначений диаграмм классов, которые используются при рисовании диаграмм классов UML. Мы перечислили ниже наиболее распространенные нотации диаграммы классов.

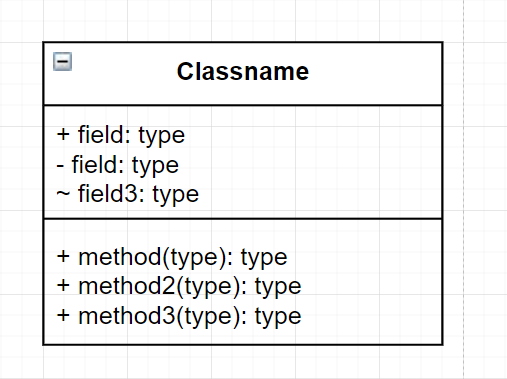


Рисунок 3 – Изображение класса

Классы представляют собой центральные объекты в системе. Он представлен прямоугольником с 3 отсеками.

Первый показывает имя класса, а средний – атрибуты класса, которые являются характеристиками объектов. В нижнем списке перечислены операции класса, которые представляют собой поведение класса.

Последние два отсека являются необязательными. Нотация класса без последних двух отделений называется простым классом и содержит только имя класса.

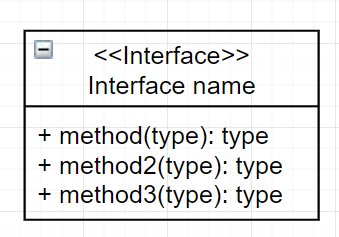


Рисунок 4 – Изображение интерфейса

Символ интерфейса на диаграммах классов обозначает набор операций, которые детализируют ответственность класса.

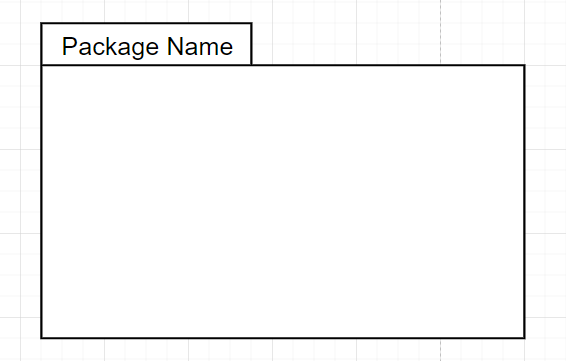


Рисунок 5 – Изображение пакета

Символ пакета используется для группировки классов или интерфейсов, которые либо похожи по своей природе, либо связаны. Группировка этих элементов дизайна с использованием символов упаковки улучшает читабельность диаграммы

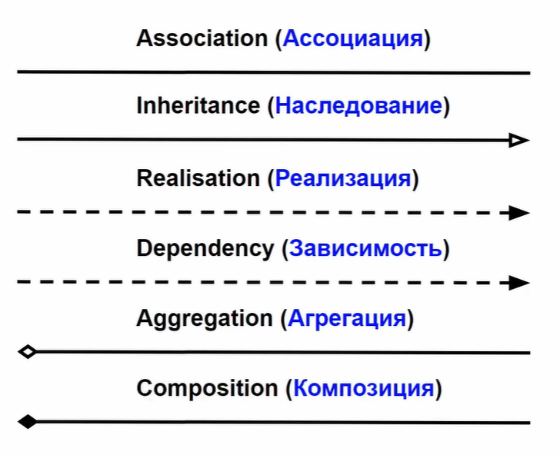


Рисунок 6 – Отношения в диаграмме классов

1. **Для чего применяются расширения диаграмм UML?**

Механизмы расширения UML включают: стереотипы (stereotype) - расширяют словарь UML, позволяя на основе существующих элементов языка создавать новые, ориентированные для решения конкретной проблемы; помеченные значения (tagged value) - расширяют свойства основных конструкций UML, позволяя включать дополнительную информацию в спецификацию элемента.

1. **Что означают понятия «стереотип» и «тегированное значение» в контексте расширенных диаграмм?**

Стереотипы: Позволяют добавить дополнительные метаданные и семантику к элементам диаграммы, превращая их в специализированные виды.

Тегированные значения: Это дополнительные атрибуты, которые могут быть присвоены элементам для уточнения их характеристик.