Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Проектирование программного обеспечения»

Отчёт по лабораторной работе №4

Студент: Дрозд А. И.

ФИТ 3 курс 2 группа

Преподаватель: Якунович А. В.

Минск 2023

**Лабораторная работа №4**

**Тема:** «Объектно-ориентированное моделирование UML-диаграммы поведения»

**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**

UML диаграмма прецедентов (вариантов использования) иллюстрирует возможные сценарии внешнего взаимодействия пользователей (Actors) с прецедентами (Use cases) системы, т.е. описывает функциональное назначение системы. Кроме того, могут быть показаны отношения между вариантами использования, группировка прецедентов через роли и приведены комментарии в случае необходимости.

**Actor** — это типовые пользователи (менеджер, операторы и т. д.). Актер представляет собой некоторую роль, которую играет пользователь (или другая система) по отношению к системе.

Возможные отношения между вариантами использования:

**-отношения Include (Включения)** — используется, когда имеется какой-либо фрагмент поведения системы, который повторяется более чем в одном варианте использования и необходимо, чтобы его описание копировалось в каждом из этих вариантов использования.

**-отношение Extend (Расширения)** — расширяющий вариант использования может дополнять (делать более точным) поведение базового варианта использования, определив в базовом варианте использования точки расширения.

Возможные отношения между актерами (группировка прецедентов):

**-отношение Generalization (Обобщения)** — если есть варианты использования, которые присущи нескольким ролям, то можно их обобщить и выделить общую абстрактную роль.

**Отношение Uses (Использование)** – основное отношение на диаграмме, которое отражает взаимодействие между актерами и прецедентами.

**Примечания (Комментарии)** предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.

В контексте сервиса продажи туристических путевок можно выделить несколько ролей пользователя (актеров):

* Пользователь – данная роль представляет собой обычного пользователя, который посещает данный интернет-сервис с целью просмотра и выбора путевки, удовлетворяющей всем его требованиям.
* Администратор сервиса – данная роль предоставляет возможности управления и изменения работоспособности сервиса, а также управление всеми пользователями, которые применяют платформу в своих целях.
* Туроператор – данная роль определяет какого-либо туристического оператора, предоставляющего свои услуги на сервисе (размещение туристических путевок).

**Иерархия актеров**

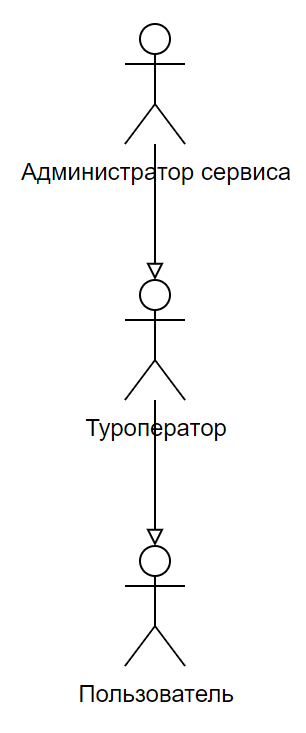
****

Рисунок 1 – Иерархия актеров

* Роль «Администратор сервиса» является расширением роли «Туроператор»
* Роль «Туроператор» является расширением роли «Пользователь»

**Выделенные преценденты подсистемы и схема их взаимодействия**

В контексте языка моделирования UML, "прецедент" обычно означает "use case" (случай использования). Use case (прецедент) представляет собой сценарий, описывающий взаимодействие между системой и внешними акторами (пользователями или другими системами) с целью достижения конкретной цели. Прецеденты (use cases) используются для определения функциональных требований к системе, а также для описания её функциональности на более высоком уровне абстракции.

В данном случае было выделено достаточно обширное количество прецедентов системы:

* «Управление путевками», включающий в себя «Добавление информации о турах», «Изменение информации о турах», «Удаление информации о турах»;
* «Управление сервисом», включающий в себя «Создание и управление схемами и пользователями», «Мониторинг и оптимизация производительности сервиса», «Резервное копирование и восстановление данных»;
* «Просмотр доступных туров и путевок»;
* «Бронирование», включающий в себя «Бронирование туров» и расширяющий «Отслеживание статуса забронированного тура»;
* «Платежи», включающий в себя «Внесение платежей» и «Просмотр истории платежей».

Все эти прецеденты системы и схема их взаимодействий отражены на рисунке 2.

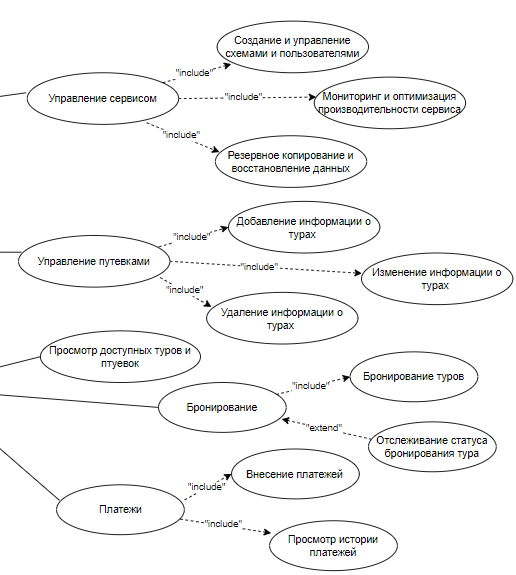


Рисунок 2 – Прецеденты системы и схема их взаимодействий

**Взаимодействие актеров с «публичными» прецедентами**

Данное взаимодействие отражает, как пользовательские роли (актеры) связаны с выделенными прецедентами сервиса продажи туристических путевок. Оно отражено на рисунке 3.

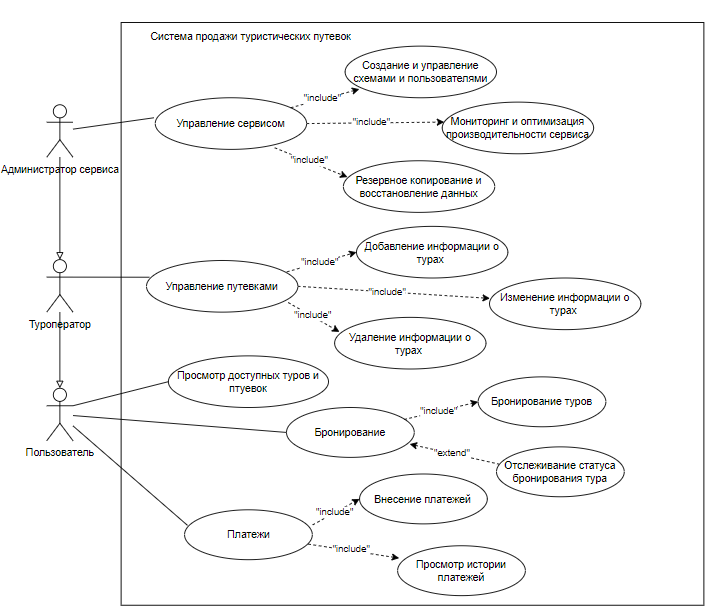


Рисунок 3 – Взаимодействие актеров с прецедентами

Рисунок, приведенный выше, показывает взаимодействие каждого из актеров, выделенных ранее. В данном случае видно, что для роли «Администратор сервиса» соответствует обобщенный прецедент «Управление сервисом», предоставляющий возможность полноценно управлять сервисом без каких-либо ограничений. Роли «Туроператор» также соответствует один обобщенный прецедент «Управление путевками», дающий возможность управлять всеми экземплярами, размещенными на сервисе от имени данного туристического оператора. Роли «Пользователь» соответствуют прецеденты «Просмотр доступных туров и путевок», «Бронирование» и «Платежи». Данные прецеденты позволяют просматривать все путевки, предложенные на сервисе, оформлять бронирование какой-либо из них и вносить оплату за выбранную путевку.

**Ответы на вопросы**

1. Дайте описание понятиям Unified process (UP) и UML.

Unified Process (UP) – это методология разработки программного обеспечения, представляющая собой итеративный и инкрементальный процесс управления проектом, который обеспечивает лучшие практики для разработки программного обеспечения. Она основана на четырёх основных компонентах: дисциплинах, артефактах, ролях и циклах разработки. UP предоставляет гибкий и адаптивный подход к разработке программного обеспечения, который позволяет адаптировать процесс к конкретным требованиям и характеристикам проекта.

UML (Unified Modeling Language) – это стандартизированный язык моделирования, который используется для визуализации, проектирования и документирования программных систем. UML предоставляет набор нотаций и диаграмм, которые позволяют разработчикам и аналитикам описывать структуру и поведение системы. UML включает различные виды диаграмм, такие как диаграммы классов, диаграммы прецедентов, диаграммы последовательности и многие другие.

2) Перечислите основные диаграммы UML 2.0.

Основные диаграммы UML 2.0 включают:

a) Диаграмма классов (Class Diagram) - используется для описания структуры классов и их отношений в системе.

b) Диаграмма прецедентов (Use Case Diagram) - описывает взаимодействие системы с акторами (пользователями или другими системами) и сценарии использования системы.

c) Диаграмма последовательности (Sequence Diagram) - показывает последовательность сообщений и взаимодействие между объектами во времени.

d) Диаграмма состояний (State Machine Diagram) - моделирует состояния объекта и переходы между ними.

e) Диаграмма активностей (Activity Diagram) - используется для описания потока действий и управления процессами в системе.

f) Диаграмма компонентов (Component Diagram) - показывает компоненты системы и их взаимосвязи.

g) Диаграмма развертывания (Deployment Diagram) - описывает физическое размещение компонентов системы на аппаратном обеспечении.

3) Назовите CASE-средства, поддерживающие создание UML диаграмм.

Несколько CASE-средств, которые поддерживают создание UML диаграмм, включают в себя:

* IBM Rational Rose
* Sparx Systems Enterprise Architect
* Visual Paradigm
* MagicDraw
* ArgoUML
* Papyrus UML
* Lucidchart

4) Укажите назначение диаграммы вариантов использования.

Диаграмма вариантов использования (Use-Case Diagram) используется для описания функциональных требований к системе и отображения взаимодействия между системой и её акторами (пользователями или другими системами). Она помогает определить, какие сценарии использования (прецеденты) существуют, какие акторы взаимодействуют с системой, и какие функции системы выполняются для удовлетворения потребностей акторов.

5) Опишите нотации, которые используются для построения Use-Case диаграммы.

Для построения диаграммы вариантов использования в UML используются следующие нотации:

* Фигура человека, представляющая акторов (пользователей или другие системы).
* Овалы, представляющие прецеденты (сценарии использования).
* Стрелки, обозначающие взаимодействие между акторами и прецедентами.
* Возможно, дополнительные артефакты и отношения для более детального описания прецедентов.