Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

«**Проектирование программного обеспечения**»

лабораторная работа №4

«Объектно-ориентированное моделирование. UML – диаграммы поведения»

«Интерфейс системы автоматического управления объектом»

Студент: Беласин Д. А.

ФИТ 3 курс 1 группа

Преподаватель: Якубенко К. Д

# 1.Постановка задачи

Функциональное назначение программы «Интерфейс туристического гида» заключается в предоставлении удобной и многофункциональной платформы для поиска, бронирования и управления экскурсиями. Программа обеспечивает систематизированное хранение информации о туристических направлениях и экскурсиях, что упрощает пользователям выбор нужного варианта отдыха.

Детализированные страницы с характеристиками экскурсий, фотографиями и актуальными ценами помогают пользователям делать осознанный выбор. Для повышения удобства предусмотрена система отзывов, где пользователи могут делиться опытом путешествий. Каждая экскурсия содержит всю необходимую информацию, включая условия бронирования и доступные дополнительные услуги.

Программа предоставляет возможность создания личных кабинетов, где пользователи могут управлять своими бронированиями, отслеживать историю поездок и сохранять избранные экскурсии.

Для гидов экскурсий программа предлагает расширенные возможности для самостоятельного управления предложениями, добавления и редактирования информации об экскурсиях, отслеживания статистики бронирований и взаимодействия с клиентами через личный кабинет.

Также предусмотрена гибкая система возврата и изменения бронирования в соответствии с условиями гидов, что делает процесс бронирования максимально прозрачным и удобным для всех сторон.

Описание объекта «Экскурсия»:

* Это туристическая услуга, доступная для бронирования через систему.
* Основные атрибуты экскурсии: название, описание, дата проведения, места посещения, цена.
* Связи: экскурсия добавляется гидом и может быть забронирована пользователем.

Описание объекта «Гид»:

* Это физическое лицо или компания, предоставляющие экскурсии через систему.
* Основные атрибуты гида: имя или название компании, контактная информация, список экскурсий.
* Связи: гид добавляет экскурсии в систему, управляет их доступностью, следит за бронированиями и взаимодействует с платежной системой.

Описание объекта «Пользователь»:

* Это физическое лицо или компания, которые могут просматривать экскурсии и бронировать их через систему.
* Основные атрибуты пользователя: имя, контактные данные, список забронированных экскурсий.
* Связи: пользователь просматривает экскурсии, делает бронирования и взаимодействует с платежной системой.

Описание объекта «Бронирование»:

* Это запись в системе, подтверждающая намерение пользователя забронировать экскурсию на определенный срок.
* Основные атрибуты бронирования: уникальный идентификатор, дата и время начала экскурсии, срок действия бронирования, статус (подтверждено, ожидает оплаты, отменено).
* Связи: бронирование связано с экскурсией и пользователем, который его создал.

Описание объекта «Платеж»:

* Это финансовая операция, которую осуществляет пользователь для завершения бронирования.
* Основные атрибуты платежа: сумма, способ оплаты (банковская карта, электронные деньги и т.д.), статус платежа (успешен, неудачен), дата проведения.
* Связи: платеж связан с пользователем, который его осуществляет, и с бронированием, которое пользователь оплачивает.

Требования к составу выполняемых функций включают:

* Регистрация и авторизация пользователей.
* Личный кабинет пользователя должен включать историю бронирований.
* Экскурсии должны разделяться на категории (например, по типу: культурные, приключенческие, гастрономические и т.д.). Также должен быть общий каталог всех экскурсий. Основные категории должны иметь подкатегории по параметрам (например, по месту проведения).
* Фильтрация экскурсий по типу, дате, цене и другим параметрам. Доступ к фильтрации экскурсий должен быть у всех ролей.
* Сортировка экскурсий по названию, цене и дате проведения. Доступ к сортировке должен быть у всех ролей.
* Поле поиска для поиска экскурсий по введённому названию или другим параметрам. Доступ к поиску должен быть у всех ролей.
* Отзывы и оценки к каждой экскурсии с возможностью оставлять комментарии только тем пользователям, которые бронировали экскурсию.
* Для каждой экскурсии должна быть отдельная страница с подробной информацией. Доступ к просмотру экскурсии должен быть у всех ролей.
* При бронировании экскурсии пользователь должен выбрать дату и время, а также иметь возможность просмотреть количество доступных мест. Доступ к этой информации должен быть у всех ролей.
* Во время добавления экскурсии в корзину гостю необходимо авторизоваться и работать через свой аккаунт.
* Возврат бронирования осуществляется через личный кабинет пользователя.
* Должна быть удобная система регистрации для гидов.
* Гиды также имеют личный кабинет, где могут управлять своими предложениями, добавлять и редактировать экскурсии.
* Гиды могут создавать новые предложения, изменять или удалять существующие. При создании предложения гид должен внести следующую информацию: название экскурсии, описание, дата, цена и места посещения.
* Менеджеры проводят верификацию гидов, проверяя их подлинность и соответствие критериям сервиса.
* Менеджеры могут блокировать и разблокировать гидов и пользователей системы.
* Менеджеры имеют доступ к отчётам о деятельности пользователей и гидов.
* Менеджеры наряду с гидами могут изменять каталог экскурсий (включая удаление и редактирование предложений).
* Менеджеры должны отвечать на запросы пользователей, гостей и гидов в системе поддержки.

Поиск экскурсий на сайте будет осуществляться с помощью полнотекстового поиска. Пользователь сможет вводить ключевые слова, связанные с названием экскурсии или местом проведения, а система найдет релевантные результаты в базе данных.

Фильтрация позволит пользователям сузить результаты по различным параметрам. Они смогут указать желаемые даты, тип экскурсии, ценовой диапазон и другие фильтры, как по отдельности, так и комбинируя их между собой для более точного поиска.

Сортировка предоставит возможность упорядочить результаты поиска по дате проведения, цене или другим параметрам.

Отчеты для менеджеров и гидов будут включать информацию о количестве зарегистрированных пользователей, количестве бронирований, общей выручке и предпочтениях пользователей. Также будут доступны данные о популярных экскурсиях и активности гидов.

Процесс бронирования экскурсии должен быть простым и удобным. Пользователь сможет выбрать интересующую экскурсию, дату и время, а затем оформить бронирование онлайн. После выбора система зарезервирует место на определённое время, в течение которого пользователь сможет завершить оформление и оплатить бронирование.

Возврат бронирований будет осуществляться в соответствии с правилами и условиями сервиса. Пользователь сможет подать заявку на возврат через личный кабинет, указав причину. Система автоматически рассчитает сумму возврата с учётомсроков и комиссий.

# 2.Описание программных средств

Отчёт по проектированию выполнялся в приложение MS Word. Диаграммы и модели были построены в онлайн сервисе Figma, имеющим открытый доступ.

Описание программного средства Figma:

* Название: Figma.
* Версия: актуальная версия (по состоянию на 2024 год). Figma обновляется регулярно.
* Разработчик: Figma, Inc.
* Адрес загрузки: https://www.figma.com/. Приложение доступно как через браузер (онлайн), так и в виде настольного клиента для различных операционных систем.
* Режим использования: Figma предоставляет бесплатный тарифный план с ограничениями на количество активных файлов и проекты, а также платные планы для команд и профессионалов с расширенными возможностями.
* Платформы: Figma работает на различных платформах. Основная версия Figma — это веб-приложение, которое работает в любом современном браузере. Доступны настольные приложения для Windows и macOS. Существует приложение Figma Mirror для iOS и Android для просмотра прототипов на мобильных устройствах.
* Типы моделей, с которыми работает Figma: Figma предназначена для работы с векторными и растровыми графическими элементами. Она широко используется для проектирования пользовательских интерфейсов (UI/UX), создания интерактивных прототипов, коллаборативной работы над дизайном, создания диаграмм, схем и визуальных моделей (например, диаграмм IDEF0, UML и других).
* Основные функции: создание и редактирование векторных элементов, кликабельных прототипов с анимацией для демонстрации работы интерфейсов, поддержка одновременной работы нескольких пользователей над одним проектом, возможность оставлять комментарии и обсуждать дизайн прямо в приложении.

Figma — это мощный кроссплатформенный инструмент для проектирования и прототипирования интерфейсов, отличающийся возможностями для совместной работы и широким спектром инструментов для создания графических и интерактивных моделей, что делает его идеальным для разработки приложений в любой сфере.

# 3.Описание практического задание

Actor — это типовые пользователи системы. Актер представляет собой некоторую роль, которую играет пользователь (или другая система) в отношении к системе. Основные роли:

Гость — это неавторизованный посетитель сайта. Он может просматривать каталог экскурсий, фильтровать их по различным параметрам, таким как тип экскурсии, дата, цена, и искать по ключевым словам. Гость имеет возможность открыть страницу понравившейся экскурсии, чтобы изучить подробную информацию о ней, посмотреть фотографии, описание и актуальные цены. Для бронирования экскурсии и использования расширенного функционала гостю необходимо зарегистрироваться.

Пользователь обладает всем функционалом гостя, а также может создавать учетную запись, авторизовываться в системе, бронировать экскурсии, оставлять отзывы об экскурсиях и оценивать их. В личном кабинете пользователю доступна история его бронирований и информация о предстоящих экскурсиях, а также возможность отмены или изменения бронирования.

Гид — это пользователь, предоставляющий экскурсии через систему. После прохождения верификации менеджером, гид получает доступ к личному кабинету с расширенным функционалом. Через личный кабинет гид может добавлять новые экскурсии в систему, редактировать информацию о текущих предложениях и удалять устаревшие экскурсии. Ему также доступны отчеты по бронированиям и отзывы клиентов.

Менеджер следит за порядком на платформе и взаимодействует с пользователями и гидами. Он проверяет достоверность данных новых гидов, может блокировать или разблокировать пользователей и гидов при нарушении ими правил сервиса. Менеджеру доступны отчеты о действиях пользователей и гидов для анализа активности и выявления подозрительных действий. Также менеджер занимается модерацией контента, редактирует и удаляет информацию о предложениях экскурсий, а также отвечает на вопросы пользователей через систему поддержки.

Иерархия актёров подсистемы:

* Гость: Начальный уровень пользователя без авторизации.
* Пользователь: Гость, который зарегистрировался в системе, получает дополнительные функции.
* Оператор: Пользователь, который получил роль от администратора, имеет доступ к управлению объектов.
* Администратор: Суперпользователь системы, который имеет все права, а также функции администрирования системы.

Отношения Include (Включения) и Extend (Расширения) применяются для описания взаимодействий между вариантами использования в моделировании систем, особенно в языке UML. Эти отношения помогают сделать модель более гибкой, упорядоченной и позволяют повторно использовать описанные элементы поведения:

* Отношения Include (Включения) — используется, когда есть общий фрагмент поведения, который должен повторяться в нескольких вариантах использования системы. Этот подход позволяет избежать дублирования информации, так как один и тот же функционал может быть необходим для разных действий пользователей.
* Отношение Extend (Расширения) — используется, когда базовый вариант использования может быть дополнен дополнительным поведением при определённых условиях. Это позволяет указать, что определенный вариант использования может быть расширен в зависимости от контекста или других факторов.

Диаграмма вариантов использования предоставлена на рисунке 3.1.



Рисунок 3.1 – Диаграмма вариантов использования

Взаимодействие гостя с прецедентами:

* Посмотреть объекты. Включает: применить фильтры, применить поиск по названию, применить сортировки, посмотреть подробную информацию об объекте.
* Регистрация.
* Авторизация.

Взаимодействие пользователя с прецедентами. Перед этими взаимодействиями необходима авторизация:

* Просмотреть состояние добавленного объекта. Расширяется: изменения функции получения уведомлений и запросить изменить параметров.
* Оставить заявку на добавление объекта. Включает: ввести подробные данные об объекте.
* Отменить или изменить заявку на добавление.
* Просмотреть профиль.
* Редактировать профиль.
* Просмотреть историю добавления объектов.

Взаимодействие оператора с прецедентами. Перед этими взаимодействиями необходима авторизация и повышение от администратора:

* Управление объектами. Расширяется: добавить объект, редактировать объект, удалить объект. Включает: отслеживание состояния объектов в реальном времени.

Взаимодействие администратора с прецедентами. Перед этими взаимодействиями необходима авторизация:

* Управление пользователями. Расширяется: выдача прав пользователям, лишение прав пользователей удаление пользователей.
* Конфигурация параметров системы оборудовния.

Диаграмма вариантов использования (use case diagram) в UML является важным инструментом для визуализации взаимодействия системы с ее пользователями и другими системами. Она помогает разработчикам и заинтересованным сторонам понять, какие функции система должна выполнять, чтобы удовлетворить потребности пользователей. На диаграмме отображаются роли пользователей, называемые акторами, и конкретные функции или задачи, которые система должна выполнять. Эти функции называются вариантами использования (use cases) и описывают отдельные сценарии взаимодействия. Например, если рассматривать систему онлайн-магазина, акторами могут быть покупатели, администраторы и курьеры. Варианты использования для покупателя могут включать такие действия, как просмотр товаров, добавление товаров в корзину и оформление заказа. Для администратора варианты использования могут включать управление товарами и обработку заказов. Таким образом, диаграммы вариантов использования не только предоставляют наглядное представление о функциональности системы, но и помогают выявить требования, которые необходимо учесть при разработке. Они служат основой для дальнейшего проектирования и реализации системы, позволяя команде сосредоточиться на потребностях пользователей.

# 4. Теоретические вопросы

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

Unified Process — методология для построения процессов [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), позволяющая команде разработки преобразовывать требования заказчика в работоспособный продукт. В зависимости от требований и доступных ресурсов, процесс разработки может быть адаптирован путём включения или исключения определённых проектных активностей.

Unified Process активно использует [унифицированный язык моделирования](https://ru.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B8%D1%84%D0%B8%D1%86%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (UML). В ядре UML лежит модель, которая позволяет команде разработки в упрощённом виде понять многообразие сложных процессов, необходимых для разработки программного обеспечения. Различные модели, используемые в Unified Process, позволяют визуализировать систему, описать её структуру и поведение, задокументировать принимаемые в процессе разработки решения.

UML — язык [графического](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) описания для [объектного моделирования](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) в области [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), для [моделирования бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2), [системного проектирования](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и отображения [организационных структур](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0).

1. Перечислите основные диаграммы UML 2.0.

* Диаграмма классов – описывает классы, их атрибуты, методы и отношения между ними.
* Диаграмма объектов – конкретные экземпляры классов и их связи в момент времени.
* Диаграмма компонентов – структура системы в виде компонентов и их взаимодействий.
* Диаграмма развертывания – отображает физическое размещение компонентов на узлах (серверы, оборудование).
* Диаграмма пакетов – показывает группировку классов и других элементов системы в пакеты.
* Диаграмма композитных структур – моделирует внутреннюю структуру классов или компонентов.

1. Назовите CASE-средства, поддерживающие создание UML диаграмм.

IBM Rational Rose – одно из самых известных средств для моделирования на основе UML. Enterprise Architect – мощное средство для моделирования систем, поддержки UML. Visual Paradigm – широко используемый инструмент для моделирования UML, поддерживающий множество типов диаграмм. StarUML – бесплатное и открытое средство для создания UML-диаграмм. ArgoUML – еще одно бесплатное и открытое средство для UML-моделирования. MagicDraw –средство для моделирования, поддерживающее SysML, BPMN. Microsoft Visio – популярный инструмент, который поддерживает создание UML-диаграмм. Lucidchart – веб-приложение для создания диаграмм, включая UML.

1. Укажите назначение диаграммы вариантов использования.

Диаграмма вариантов использования (Use Case Diagram) служит для:

* Моделирования функциональных требований системы. Она описывает, как различные пользователи (акторы) взаимодействуют с системой через различные сценарии (варианты использования).
* Определения ролей (акторов) и взаимодействий между этими ролями и системой.
* Коммуникации с заинтересованными сторонами. Диаграмма помогает объяснить функциональность системы пользователям, разработчикам и другим участникам проекта на высоком уровне.
* Планирования системы. Она позволяет определить границы системы и ключевые функциональности, что помогает в проектировании и управлении требованиями.

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

Для построения диаграммы вариантов использования (Use-Case Diagram) в UML используются следующие нотации:

* Актор (Actor): обозначается человечком (stick figure), представляющим собой роль, которая взаимодействует с системой. Это может быть человек, другая система или устройство. Акторы могут быть внешними по отношению к системе, но участвуют в её работе.
* Вариант использования (Use Case): овал, в котором указано название действия (сценария), выполняемого системой в ответ на запрос акторов. Каждый вариант использования представляет собой законченную последовательность действий.
* Система (System Boundary): прямоугольник, который ограничивает область системы. Все варианты использования помещаются внутрь этого прямоугольника, а акторы располагаются снаружи, демонстрируя взаимодействие с системой.
* Ассоциации: линии, соединяющие акторов с вариантами использования. Они показывают, кто и с какими сценариями взаимодействует.
* Обобщение (Generalization): стрелка с пустым треугольником, которая указывает на обобщение или наследование как между акторами, так и между сценариями. Например, актёр Администратор может быть частным случаем актёра Пользователь.
* Включение (Include): пунктирная стрелка с надписью include, которая указывает на обязательное использование одного сценария другим.
* Расширение (Extend): пунктирная стрелка с надписью extend, которая указывает на дополнительное или альтернативное поведение, возникающее в определённых условиях.