**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | МФПУ Синергия |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Программист |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе № 2.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Основные инструменты для создания, исполнения и управления информационной системой | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка ПО |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Панцулая Темур Генадьевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП\_205 ПРОГ |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев И.В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Лабораторная работа № 2. «**Основные инструменты для создания, исполнения и управления информационной системой**»**

**Цель работы:** изучение методологии моделирования информационных систем на основе языка UML, построение диаграммы вариантов использования и генерация кода.

**Задание 1.**

|  |  |
| --- | --- |
| @startuml  left to right direction  actor "Продавец" as Seller  actor "Управляющий магазином" as Manager  actor "Клерк магазина" as Clerk  actor "Бухгалтерская система" as AccountingSystem  rectangle "Система управления заказами" {  usecase "Ввести новый заказ" as NewOrder  usecase "Изменить существующий заказ" as ModifyOrder  usecase "Напечатать инвентарную опись" as PrintInventory  usecase "Обновить инвентарную опись" as UpdateInventory  usecase "Оформить заказ" as ProcessOrder  usecase "Отклонить заказ" as RejectOrder  usecase "Выполнить поставку заказа" as FulfillOrder  Seller -- NewOrder  Seller -- ModifyOrder  Clerk -- PrintInventory  Manager -- UpdateInventory  Clerk -- ProcessOrder  Clerk -- RejectOrder  Clerk -- FulfillOrder  Manager -- ProcessOrder  RejectOrder .> ProcessOrder : extends  }  @enduml |  |

**Задание 2.**

|  |  |
| --- | --- |
| @startuml  left to right direction  actor "Клиент банкомата" as Customer  actor "Банк" as Bank  rectangle "Банкомат" {  usecase "Снятие наличных по кредитной карте" as WithdrawCash  usecase "Проверка ПИН-кода" as CheckPIN  usecase "Получение справки о состоянии счета" as CheckBalance  usecase "Блокирование кредитной карточки" as BlockCard  Customer -- WithdrawCash  Customer -- CheckBalance  WithdrawCash ..> Bank  CheckBalance ..> Bank  WithdrawCash -- CheckPIN : include  CheckBalance -- CheckPIN : include  BlockCard ..> CheckPIN : extend  }  @enduml |  |

**Задание 3.**

|  |  |
| --- | --- |
| @startuml  left to right direction  actor "Клиент банка" as Client  actor "Кассир" as Teller  actor "Оператор" as Operator  actor "Банк" as Bank  usecase "Снять наличные" as WithdrawCash  usecase "Перевести деньги со счета" as TransferMoney  usecase "Положить деньги на счет" as DepositMoney  usecase "Пополнить запас денег" as ReplenishCash  usecase "Подтвердить пользователя" as AuthenticateUser  Client -- WithdrawCash  Client -- TransferMoney  Client -- DepositMoney  Teller -- ReplenishCash  Operator -- ReplenishCash  WithdrawCash ..> AuthenticateUser : include  TransferMoney ..> AuthenticateUser : include  DepositMoney ..> AuthenticateUser : include  @enduml |  |

**Контрольные вопросы**

1.Универсальный язык моделирования UML (Unified Modeling Language).

UML — это стандартизированный язык моделирования, используемый для визуального представления системы, ее структуры и поведения. Он не является языком программирования, а служит инструментом для спецификации, визуализации, конструирования и документирования артефактов программных систем. UML позволяет разработчикам создавать модели, понятные всем участникам проекта, независимо от их технических знаний. Он унифицирует различные подходы к моделированию, обеспечивая общее понимание и облегчая сотрудничество.

2. Понятие диаграммы UML.

Диаграмма UML — это графическое представление части модели системы. Она использует определённые символы и нотации для визуализации структуры, поведения или взаимодействия элементов системы. Диаграммы упрощают понимание сложных систем, делая их более доступными для анализа и обсуждения. Каждая диаграмма UML фокусируется на определённом аспекте модели.

3. Виды диаграмм UML UML включает множество видов диаграмм, которые группируются в две основные категории:

Диаграммы структуры: Они описывают статические аспекты системы, её структуру и отношения между компонентами.

Примеры:

- Диаграмма классов

- Диаграмма объектов

- Диаграмма компонентов

- Диаграмма развертывания

- Диаграмма пакетов

- Диаграмма профилей

- Диаграмма структуры составного компонента

- Диаграммы поведения: Они описывают динамические аспекты системы, её поведение и взаимодействие компонентов во времени.

Примеры:

- Диаграмма вариантов использования (Use Case Diagram)

- Диаграмма состояний (State Machine Diagram)

- Диаграмма деятельности (Activity Diagram)

- Диаграмма взаимодействия:

- Диаграмма последовательности (Sequence Diagram)

- Диаграмма сотрудничества (Collaboration Diagram) \* Диаграмма временной последовательности (Timing Diagram)

4. Основные элементы диаграммы вариантов использования.

Основные элементы диаграммы вариантов использования:

- Актеры (Actors): Представляют роли пользователей или внешних систем, взаимодействующих с системой. Обычно изображаются в виде человечка.

- Варианты использования (Use Cases): Представляют собой функции или сервисы, предоставляемые системой. Обычно изображаются в виде эллипсов.

- Ассоциации (Associations): Линии, соединяющие актеров с вариантами использования, показывающие, как актеры взаимодействуют с функциями системы.

- Зависимости (Dependencies): Линии со стрелками, показывающие зависимости между вариантами использования (например, `include` или `extend`).

- Системная граница (System Boundary): (Необязательно) Прямоугольник, окружающий все варианты использования, чтобы обозначить границы системы.

5. Что такое диаграмма вариантов использования (Use Case Diagram)?

Диаграмма вариантов использования (Use Case Diagram) — это диаграмма UML, которая моделирует функциональность системы с точки зрения внешних актеров, взаимодействующих с ней. Она показывает, какие функции система предоставляет и кто эти функции использует. Диаграмма вариантов использования полезна на ранних этапах разработки для определения требований к системе и ее взаимодействия с пользователями.

6. Что такое диаграмма вариантов (Variant Diagram)?

Термин "диаграмма вариантов" не является стандартным термином в UML (Unified Modeling Language) или других распространенных методологиях моделирования.

- Диаграмма состояний (State Machine Diagram): Эта диаграмма показывает различные состояния объекта и переходы между ними. Различные пути переходов могут представлять варианты поведения системы в зависимости от условий.

- Диаграмма активности (Activity Diagram): Показывает поток управления и действия в процессе. Разветвления и слияния потоков могут представлять варианты выполнения процесса.

- Диаграмма классов (Class Diagram) с использованием вариантов: В диаграмме классов можно использовать различные модификаторы (например, обобщения, реализацию интерфейсов) для отображения вариантов реализации классов.

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: Основные инструменты для создания, исполнения и управления информации.

Цель: изучение методологии моделирования информационных систем на основе

языка UML, построение диаграммы вариантов использования и генерация кода.

**Задание 1.**

|  |  |
| --- | --- |
| @startuml  left to right direction  actor Клиент  actor Поставщик  actor Администратор  rectangle Магазин\_видеопродукции {  usecase "Покупка товара"  usecase "Прокат видеоносителя"  usecase "Возврат видеоносителя"  usecase "Регистрация в видеоклубе"  usecase "Предварительный заказ"  usecase "Управление товаром"  usecase "Управление поставщиками"  usecase "Управление клиентами"  usecase "Списание видеоносителей"  usecase "Выписка квитанции"  usecase "Оплата счета поставщику"  usecase "Получение отчета"  }  Клиент -- "Покупка товара"  Клиент -- "Прокат видеоносителя"  Клиент -- "Возврат видеоносителя"  Клиент -- "Регистрация в видеоклубе"  Клиент -- "Предварительный заказ"  Поставщик -- "Оплата счета поставщику"  Администратор -- "Управление товаром"  Администратор -- "Управление поставщиками"  Администратор -- "Управление клиентами"  Администратор -- "Списание видеоносителей"  Администратор -- "Выписка квитанции"  Администратор -- "Получение отчета"  "Прокат видеоносителя" -- "Выписка квитанции"  "Возврат видеоносителя" -- "Выписка квитанции"  "Покупка товара" -- "Выписка квитанции"  @enduml |  |

**Задание 2.**

|  |  |
| --- | --- |
| @startuml  actor Пассажир  actor Агент  usecase Заказать\_билет  usecase Купить\_билет  usecase Бронирование  usecase Продажа  Пассажир -- Заказать\_билет  Пассажир -- Купить\_билет  Агент -- Бронирование  Агент -- Продажа  Заказать\_билет .> Бронирование : includes  Купить\_билет .> Бронирование : includes  Купить\_билет .> Продажа : includes  @enduml |  |

**Задание 3.**

|  |  |
| --- | --- |
| @startuml  left to right direction  actor Клиент  actor Администратор  rectangle Магазин\_косметики\_бытовой\_техники {  usecase Покупка\_товара  usecase Просмотр\_каталога  usecase Добавление\_товара\_в\_корзину  usecase Оформление\_заказа  usecase Оплата\_заказа  usecase Управление\_запасами  usecase Управление\_товарами  usecase Генерация\_отчетов  }  Клиент -- Покупка\_товара  Клиент -- Просмотр\_каталога  Клиент -- Добавление\_товара\_в\_корзину  Клиент -- Оформление\_заказа  Клиент -- Оплата\_заказа  Администратор -- Управление\_запасами  Администратор -- Управление\_товарами  Администратор -- Генерация\_отчетов  @enduml |  |

Контрольные вопросы

1.Универсальный язык моделирования UML:

Ответ: UML (Unified Modeling Language) — это универсальный язык моделирования, используемый для спецификации, визуализации, конструирования и документирования артефактов программного обеспечения, а также других не программных систем. Он предоставляет стандартный набор графических обозначений и понятий для описания различных аспектов системы.

2. Понятие диаграммы:

Ответ: В UML диаграмма — это графическое представление некоторой части модели системы. Диаграммы визуализируют различные аспекты системы, помогая разработчикам, аналитикам и другим заинтересованным сторонам понять ее структуру, поведение и взаимодействие компонентов. Они являются средством общения и документирования.

3. Виды диаграмм:

Ответ: UML включает множество типов диаграмм, группируемых по назначению:

\* Диаграммы поведения: описывают динамические аспекты системы.

\* Диаграммы вариантов использования (Use Case Diagrams): показывают взаимодействие акторов с системой.

\* Диаграммы состояний (State Machine Diagrams): моделируют переходы состояний объекта.

\* Диаграммы активности (Activity Diagrams): описывают поток управления в системе.

\* Диаграммы взаимодействия (Interaction Diagrams):

\* Диаграммы последовательности (Sequence Diagrams): показывают взаимодействие объектов во времени.

\* Диаграммы сотрудничества (Collaboration Diagrams): показывают взаимодействие объектов с указанием связей.

\* Диаграммы структуры: описывают статическую структуру системы.

\* Диаграммы классов (Class Diagrams): описывают классы, атрибуты и методы.

\* Диаграммы объектов (Object Diagrams): показывают экземпляры классов в конкретный момент времени.

\* Диаграммы компонентов (Component Diagrams): моделируют физическую структуру системы.

\* Диаграммы развертывания (Deployment Diagrams): показывают физическое размещение компонентов.

\* Диаграммы пакетов (Package Diagrams): организуют элементы модели в пакеты.

4. Основные элементы диаграммы вариантов использования:

Ответ: Основные элементы диаграммы вариантов использования:

\* Акторы (Actors): субъекты, взаимодействующие с системой (люди, системы, устройства). Обозначаются человечком.

\* Варианты использования (Use Cases): описывают функциональность системы с точки зрения акторов. Обозначаются овалом.

\* Связи (Relationships): показывают взаимодействие между акторами и вариантами использования, а также между самими вариантами использования.

Основные типы связей:

\* Ассоциация (Association): связь между актором и вариантом использования.

\* Обобщение (Generalization): наследование вариантов использования (более общий случай и его специализации).

\* Включение (Include): один вариант использования включает другой (как подзадача).

\* Расширение (Extend): один вариант использования расширяет другой в определенных условиях.

5. Что такое диаграмма использования?

Ответ: Термин "диаграмма использования" часто используется как синоним для диаграммы вариантов использования. Она показывает, \*как\* акторы взаимодействуют с системой, какие функции (варианты использования) они выполняют. 6. Что такое диаграмма вариантов?

Термин "диаграмма вариантов" не является стандартным термином в UML. Скорее всего, под этим подразумевается диаграмма вариантов использования (Use Case Diagram), которая описывает различные варианты взаимодействия акторов с системой. Поэтому это просто альтернативное (неформальное) название.