Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра Оборудования и автоматизации химических производств

**Отчет по лабораторной работе №1**

**Программная инженерия**

**Диаграммы прецедентов**

|  |  |
| --- | --- |
| Выполнил | |
| Студент гр. | АТП-22-1м |
|  | Щекалёв И. М. |
| (подпись) | (Ф.И.О.) |
|  | |
| Проверил  преподаватель | |
|  | Колыхматов А. О. |
| (подпись) | (Ф.И.О.) |

Пермь 2023

**Цель работы**

Изобразить и описать информационную систему с помощью диаграммы прецедентов.

**Теоретическая часть**

**Диаграмма прецедентов (Use Case)**

Диаграмма прецедентов (use case diagram) – Взаимодействие между системой и её окружением.

Диаграмма прецедентов (Use Case) - это единственная диаграмма UML, в которой большое внимание уделяется текстовым описаниям, а графическая нотация носит иллюстративный характер. В некоторых случаях допускается опускать текстовые описания, однако это не рекомендуется.

Основная идея прецедентов заключается в выявлении и описании ситуаций, когда пользователи или внешние системы взаимодействуют с системой. Основное внимание здесь обращается на то, что надо пользователю, и на то, как он это получает при помощи системы. Таким образом, выделив и описав все варианты использования системы, мы получаем описание функциональных требований. Построение модели прецедентов является первым шагом работы над проектом.

Прецедент - описание множества содержательно-близких сценариев взаимодействия акторов (внешних агентов) с системой, которое осуществляется с целью получения акторами некоторого полезного результата при помощи системы.

Прецедент может охватывать сразу несколько сценариев, следуя которым актор может достичь своей цели. Например, рассмотрим прецедент "Отправка e-mail" из некоторой почтовой системы. Основные шаги отправки письма хорошо известны: нужно написать текст письма, указать тему и адресата, и наконец отправить его. Однако в каждом случае возможны варианты.

Основные свойства прецедентов:

- Прецедент охватывает некоторую очевидную для пользователя функцию, т.е. пользователь должен представлять себе суть процесса и, самое главное, знать что он может получить в конце.

- Прецедент может быть как небольшим, так и достаточно крупным (например, составить список, сделать предметный указатель).

Прецедент решает некоторую дискретную и важную для пользователя задачу. За решение таких задач обычно платят деньги.

Источником прецедентов являются требования, которые пользователь предъявляет к системе, т.е. это описание функций, которые он хотел бы видеть в системе.

В процессе взаимодействия пользователя и системы можно выделить два аспекта: задачи пользователя и системные взаимодействия.

Задачи пользователя - это задачи из предметной области. Они важны для пользователя, от их выполнения зависит его душевное и материальное состояние.

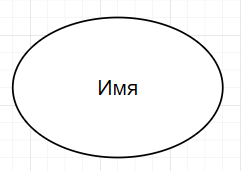
Системные взаимодействия - запросы к функциональности системы. С помощью системных взаимодействий пользователь решает свои задачи.

Прецеденты соответствуют задачам пользователя и описываются в терминах системных взаимодействий.

**Визуальное представление прецедента**

Прецедент-фрагмент поведения ИС без раскрытия его внутренней структуры.

Прецедент-сервис, который информационная система предоставляет пользователю (актеру).



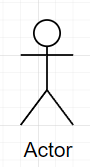
**Актер**

Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей.

Важно понимать, что актор - не конкретный человек, а роль, которую он играет по отношению к системе.

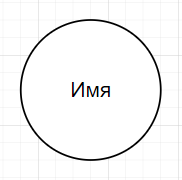
Процесс выделения акторов тесно связан с определением границ системы. Определив границы системы, можно идентифицировать ее внешние и внутренние свойства Внешняя среда представляется только акторами.

Выбор границ системы определяется целями моделирования.



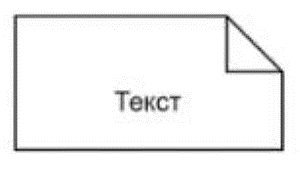
**Интерфейс**

Интерфейс определяет совокупность операций, которые обеспечивают необходимый набор сервисов для актера.



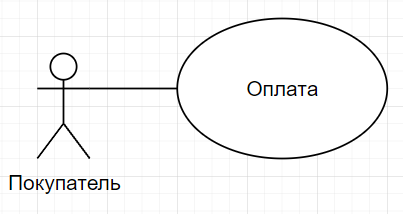
**Примечание**

Примечание предназначено для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.

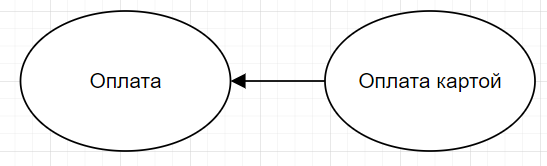


**Отношения**

Ассоциация - это устойчивое структурное отношение между сущностями



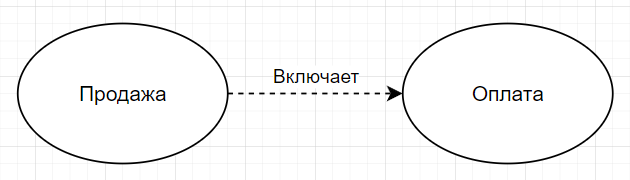
Обобщение - отношение между элементом-родителем и элементом-потомком, при котором потомок является частным случаем родителя и наследует его структуру, поведение и семантику



Зависимость - это семантическое отношение между двумя сущностями, которое означает, что одна сущность (зависимая) некоторым образом зависит от другой сущности (независимой) и изменения структуры, поведения или семантики независимой сущности могут повлечь за собой изменения в структуре, поведении или семантике зависимой сущности.

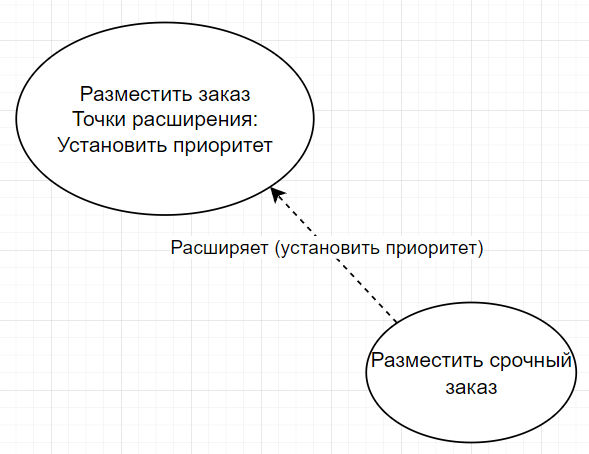
На диаграммах прецедентов используется два вида отношений зависимостей - это зависимость включения и зависимость расширения.

Зависимость включения - эта такая зависимость между прецедентами, в которой один прецедент (базовый) явно включается в другой прецедент.



Зависимость расширения - эта такая зависимость между прецедентами, в которой один прецедент (расширяющий) неявно включается в другой прецедент (расширяемый).

Зависимость расширения применяют для моделирования таких частей прецедента, которые пользователь воспринимает как необязательное поведение системы. Данная зависимость также используется для моделирования под потоков, которые выполняются только при определенных условиях или для больших альтернативных потоков.



Прецеденты

В простейшем случае прецедент представляет собой один или несколько абзацев текста, описывающих взаимодействие акторов и системы. Однако гораздо удобнее и эффективнее использовать шаблоны, состоящие из набора полей. UML не определяет стандартного способа описания прецедентов, и в каждом случае выбирается наиболее удобный формат. Как минимум обычно выделяют следующие поля: Название, Акторы и Описание.

Прецеденты можно разделить на высокоуровневые и развернутые.

Высокоуровневый прецедент-это прецедент, который кратко описывает основные моменты взаимодействия акторов с системой.

Развернутый прецедент - это прецедент, который детально описывает процесс взаимодействия акторов с системой.

При описании развернутых прецедентов обычно выделяют основной поток, подпотоки и альтернативные потоки.

Основной поток - это описание типичного, наиболее вероятного хода событий.

Подпоток - это логически законченная последовательность основного потока, которую имеет смысл описать отдельно (например, используется несколько раз).

Альтернативный поток - это отклонения от основного потока, связанные с возникновением различного рода исключений и нестандартных ситуаций. В альтернативном потоке могут пропускаться или изменяться шаги основного потока, а также добавляться новые.

При описании развернутых прецедентов обычно используют таблицы с двумя столбцами: действия акторов и отклик системы.

Виды прецедентов

Главные прецеденты - это прецеденты, без реализации которых невозможно функционирование системы.

Второстепенные прецеденты - это прецеденты без реализации которых возможно частичное функционирование системы.

Дополнительные прецеденты - прецеденты, которые не влияют на основные функционирование системы.

Идеальные прецеденты - это развернутые прецеденты, отражающие процесс взаимодействия акторов с системой без указания конкретных технологий ввода/вывода информации.

Реальные прецеденты - это развернутые прецеденты, отражающие процесс взаимодействия акторов с системой в терминах конкретных проектных решений и на основе конкретных технологий ввода-вывода. Такие прецеденты очень удобны при написании документации.

Порядок построения модели

Для построения модели прецедентов рекомендуется придерживаться следующей последовательности шагов:

- Определить границы системы, идентифицировать акторов и прецеденты.

- Записать все прецеденты в высокоуровневой форме и разбить их по

категориям: главные, второстепенные, дополнительные.

- Выделить наиболее важные и рискованные прецеденты и записать их в

развернутой идеальной форме.

- Определить отношения между прецедентами и проилюстрировать их на диаграмме.

Преимущества модели прецедентов

- Планирование итераций.

- Определение профилей пользователей.

- Оптимизация системы.

- Организация тестирования.

- Формирование документации пользователя.

**Практическая часть**

Информационная система продажи билетов в кинотеатр.

**Краткое описание системы**

*Акторы:*

* Администратор,
* Кассир,
* Клиент.

*Задачи:*

* Планирование сеансов (админ),
* Продажа билетов кассиром (кассир),
* Бронирование и покупка через интернет (клиент),
* Финансовые отчёты (админ),
* План зала (клиент при бронировании/покупки).

Кинотеатр должен предоставлять клиентам быстрый и удобный способ приобретения билетов для последующего посещения киносеанса. Работник может получить информацию обо всех проданных билетах на каждый сеанс фильмов, представленных к показу на текущий момент.

Приобрести билет на сеанс можно в кассах кинотеатров либо через сайт.

Администратор контролирует работу кассиров и занимается распределением сеансов на каждый день.

Кассир занимается продажей билетов на кассе.

Посетитель кинотеатра проходит на кассу, где выбирает желаемый фильм и сеанс на него.

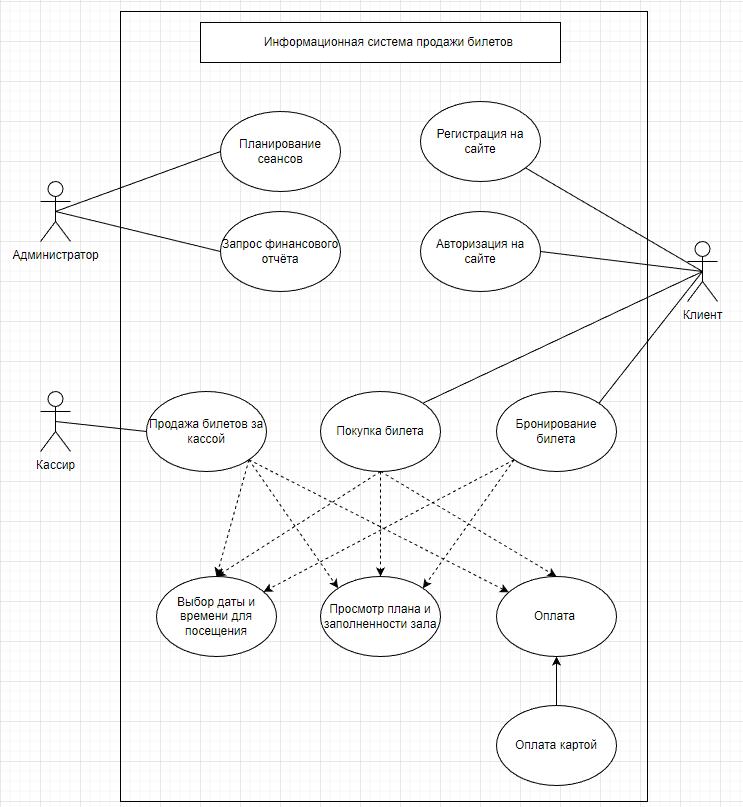
Информация о клиенте:

* ID клиента (создается автоматически)
* ФИО

Информация о сеансах:

* Название фильма
* Время сеанса
* Стоимость билета

Диаграмма прецедентов



Прецедент 1

**Название:** Регистрация.

**Акторы:** Клиент.

**Краткое описание:** Клиент проходит регистрацию для бронирования и покупки билетов в кинотеатр онлайн.

**Триггер:** Открытие бланка регистрации.

**Основной поток:**

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Покупатель открывает бланк регистрации и заполняет форму (ФИО, дату рождения, фотографию лица, данные банковской карты, доп. фильтры) | 2. Проверка на валидность (Е1). Система регистрирует покупателя. Отправка уведомления о регистрации. |

**Подпотоки:**

S1: Нет

**Альтернативные потоки:**

E1: Ошибка в заполненных данных. Информирование пользователя о допущенной ошибки в заполнении формы. Прецедент завершается.

**Прецедент 2**

**Название:** Авторизация

**Акторы:** Клиент

**Краткое описание:** Клиент заходит на сайт, система его авторизует и уведомляет о том, что человек может забронировать или купить билет.

**Триггер:** Открытие бланка авторизации.

**Основной поток:**

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Зарегистрированный покупатель открывает окно авторизации. | 2. Система авторизует покупателя. (Е1) Уведомление человека о прохождении авторизации и возможности . |

**Подпотоки:**

S1: Нет

**Альтернативные потоки:**

E1: Не прохождение авторизации. Система не смогла авторизовать (зарегистрированного/ не зарегистрированного) покупателя.

**Прецедент 3**

**Название:** Бронирование билета

**Акторы:** Клиент

**Краткое описание:** Зарегистрированный и авторизованный покупатель выбирает дату и время сеанса, а также место в зале. Система оформляет место на человека. У клиента есть 30 минут что бы купить билет на это место, дату и время.

**Триггер:** Авторизованный пользователь выбрал функцию бронирования.

**Основной поток:**

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Зарегистрированный и авторизованный пользователь выбирает дату, время и место. | 2. Система проверяет свободно ли выбранное место и оформляет введённые данных на покупателя. Подсчет цены и начало отсчёта 30 минут.(S1) (E2). |

**Подпотоки:**

S1: Прошло 30 минут и клиент не купил билет, система делает выбранное место свободным. Удаление товара из оформленного на него списка товаров. Прецедент продолжается.

E2: Выбранное место занято, система предлагает выбрать другое место.

**Прецедент 4**

**Название:** Покупка билета.

**Акторы:** Клиент.

**Краткое описание:** Зарегистрированный и авторизованный покупатель выбирает дату и время сеанса, а также место в зале. Система списывает с банковской карты сумму за товар.

**Триггер:** Авторизованный пользователь выбрал функцию покупки билета.

**Основной поток:**

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Зарегистрированный и авторизованный пользователь выбирает дату, время и место. | 2. Система проверяет, свободно ли выбранное место. (S1). |
| 3. Покупатель оплачивает билет. | 4. Списание, с карты покупателя, суммы за товар. Внесение купленного билета в специальный список. Внесение в отчеты купленного товара. (E2) |

**Подпотоки:**

S1: Выбранное место занято, система предлагает выбрать другое место.

**Альтернативные потоки:**

E2: Не хватка денег на счету. Система уведомляет покупателя о том, что на счету не хватает денег и нужно пополнить баланс

**Прецедент 5**

**Название:** Продажа билетов за кассой.

**Акторы:** Кассир.

**Краткое описание:** Кассир спрашивает дату, время и место посещения и вбивает полученные данные в систему. Далее клиент производит оплату.

**Триггер:** Запрос отчетов.

**Основной поток:**

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Кассир вбивает данные. | 2. Система проверяет, свободно ли выбранное место. (S1). |
| 3. Клиент производит оплату.(E1) | 4. Система оформляет выбранное место на клиента. |

**Подпотоки:**

S1: Выбранное место занято, система предлагает выбрать другое место.

**Альтернативные потоки:**

E1: Не хватка денег на счету. Система выдает отказ и просит оплатить ещё раз.

**Вывод**

В ходе лабораторной работы:

- Изучены диаграммы прецедентов.

- Приобретены навыки построения диаграмм.

- Изображена и описана информационная система продажи билетов в кинотеатр**.**

**Список литературы**

1. Колыхматов А. О. Презентация на тему «Моделирование ПО» по предмету Программная инженерия.