Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование программного обеспечения»

Индивидуальная работа

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ.

ДИАГРАММЫ ПОВЕДЕНИЯ UML.

Цель: Изучение методологии объектно-ориентированного моделирования средствами UML. Получение дополнительных навыков проектирования моделей информационной системы с применением возможностей UML диаграмм

Выполнил: Ободов П.А.

ФИТ 3 курс 2 группа

Преподаватель: Якунович А. В.

**Введение**

Целью данной работы является изучение методологии объектно-ориентированного моделирования с применением средств UML и применение этой методологии для создания двух ключевых диаграмм поведения. Рассмотренная в проекте тематика – интернет-магазин кроссовок - стала основой для разработки этих диаграмм.

Первая диаграмма поведения моделирует процесс сбора данных о клиентах а также доступность каталога моделей для пользователей, позволяя визуализировать последовательность шагов и действий, необходимых для этого процесса. Эта диаграмма акцентирует внимание на активностях и переходах между ними, что способствует лучшему пониманию логики сбора информации о клиентах и возможность этих клиентов видеть каталог.

Вторая диаграмма поведения моделирует процесс оформления заказа, предоставляя взгляд на последовательность шагов, которые совершает клиент при оплате модели. Эта диаграмма состояний позволяет увидеть изменения состояний системы в ответ на действия клиента, что полезно для оптимизации пользовательского опыта.

Обе диаграммы предоставляют ясное представление о том, как объекты взаимодействуют друг с другом в рамках этих действий. Результаты исследования могут быть использованы для более глубокого понимания процессов и оптимизации системы в целом.

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

**Управление товарами:**

* Добавление новых моделей кроссовок с указанием бренда, описания, цены, размеров и доступных цветов.
* Редактирование и удаление существующих моделей кроссовок.
* Просмотр списка доступных моделей.

**Отчеты и аналитика:**

* Генерация отчетов о продажах, остатках товаров на складе и финансовых показателях.
* Создание дашбордов и статистики продаж для анализа популярности определенных брендов или моделей.

**Интерфейс для пользователей:**

* Создание учетных записей для сотрудников с разными уровнями доступа (администраторы, менеджеры магазина и продавцы).
* Управление доступом к функциональным возможностям в зависимости от роли пользователя.

**Продажа кроссовок:**

* Оформление заказов на покупку кроссовок с указанием размера, цвета и количества.
* Подтверждение оплаты и формирование заказа для отправки.

**Выдача товара:**

* Подготовка заказанных кроссовок к выдаче после завершения оплаты.
* Уведомление клиента о готовности кроссовок к выдаче.

В данной лабораторной работе необходимо составить диаграмму развёртывания и компонентов, которая должна содержать: список компонент, из которых будет состоять подсистема, существующие связи между компонентами подсистемы, список физических устройств, на которых будет работать подсистема и определить соединения.

**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**

Первой диаграммой была выбрана диаграмма деятельности – рисунок 1. Она посвящена процессу извлечения данных моделей кроссовок пользователю.

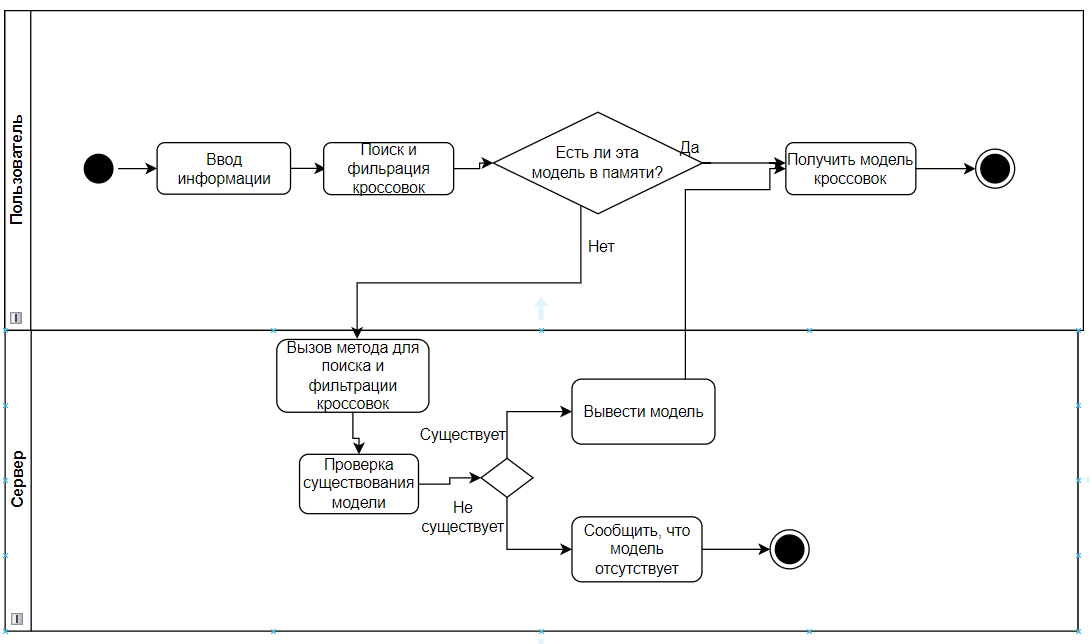


Рисунок 1 – Диаграмма деятельности

Основными элементами диаграммы выступают операции: «Ввод информации», «Поиск и фильтрация моделей кроссовок», «Вызов метода поиска и фильтрации «Вывод сообщения об ошибке», «Сообщить, что моделей нет». Использовано 2 условных блока – после проверки данных и поиска необходимой модели с условием.

Вторая диаграмма – диаграмма состояния. Отобразим на ней процесс оплаты заказа.

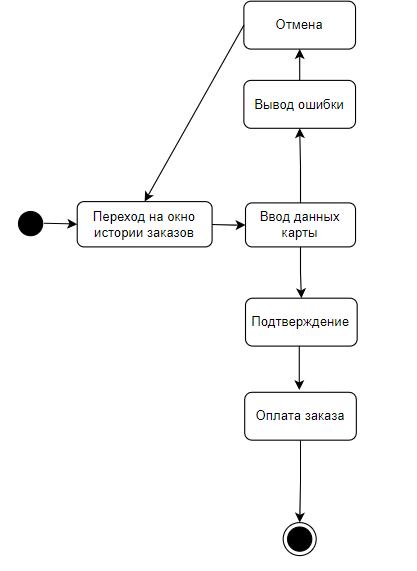


Рисунок 2 – Диаграмма состояния

Основными элементами диаграммы выступают следующие объекты: начальное псевдосостояние, состояния «Переход на окно истории заказов», «Ввод данных карты», «Подтверждение», «Отмена», а также конечное состояние.

Переходы определяют статус транзакции оплаты, а также статус взаимодействия пользователя с системой.

**Вывод:** Я изучил методологию объектно-ориентированного моделирования средствами UML. Получил дополнительные навыки проектирования моделей информационной системы с применением возможностей UML диаграмм поведения в контексте программного средства «МusApp».

**Ответы на теоретические вопросы**

1. **Укажите виды диаграмм поведения.**

Всего существует 3 вида диаграмм поведения:

* диаграмма деятельности;
* диаграмма состояний;
* диаграмма вариантов использования.

1. **Опишите назначение диаграммы деятельности.**

**Диаграмма активностей (видов деятельности)** - один из доступных [видов диаграмм](https://flexberry.github.io/ru/fd_editing-diagram.html), поддерживаемых [Flexberry Designer](https://flexberry.github.io/ru/fd_flexberry-designer.html). Она, как и [диаграмма состояний](https://flexberry.github.io/ru/fd_statechart-diagram.html), **отражает динамические аспекты поведения системы**. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

Активности на диаграмме “разбросаны” по беговым дорожкам, каждая из которых соответствует поведению одного из объектов (например, клиента, менеджера, веб-сервера, сервера БД и т.п.). Благодаря этому легко определить, каким из объектов выполняется каждая из активностей. Дорожка - часть области диаграммы деятельности, на которой отображаются только **те активности, за которые отвечает конкретный объект**. Предназначены дорожки для разбиения диаграммы в соответствии **с распределением ответственности за действия**. Имя дорожки может означать роль или объект, которому она соответствует.

**3. Опишите основные нотации, которые используются на диаграмме состояний.**

| **Элемент/Нотация** | **Предназначение** |
| --- | --- |
| Пример | Класс (Class) |
| Пример | Состояние (State) |
| Пример | Состояние (StateEx) |
| Пример | Составное состояние (Composite state) |
| Пример | Разделитель (Concurrent state) |
| Пример | История (History) |
| Пример | Глубокая история (Deep history) |
| Пример | Начальное состояние (Start state) |
| Пример | Конечное состояние (Final state) |
| ПримерПример | Синхронизатор/разветвитель (Complex transition) |
| Пример | Переход (Transition) |
| Пример | Сообщение (Event message) |
| Пример | Точка изгиба связей (Point) |
| Пример | Комментарий (Note) |
| Пример | Коннектор комментария (Note connector) |

**4. Укажите виды связей между объектами на диаграмме последовательностей.**

Синхронное сообщение — отправитель передаёт ход управления актёру-получателю, которому необходимо провести в прецеденте некоторое действие. Пока проводимое получателем действие не будет завершено (не будет получено ответное сообщение), отправитель теряет возможность производить какие-либо действия. Графически изображается как сплошная линия со стрелкой в виде закрашенного треугольника, после которой идёт прямоугольник, отражающий деятельность объекта, в конце которого находится ответное сообщение.

Ответное сообщение — данное сообщение является ответом на синхронное сообщение. Обычно, содержит какое-либо возвращаемое изначальному отправителю значение, также возвращающее ему управление (возможность действовать). Графически изображается пунктирной линией с открытой стрелкой.

Асинхронное сообщение — отправитель передаёт ход управления получателю, которому необходимо провести в прецеденте некоторое действие. Основное отличие от синхронного сообщения состоит в том, что отправитель не теряет возможности совершать другие действия. Графически изображается сплошной линией с открытой стрелкой.

Потерянное сообщение — сообщение без адресата.

Найденное сообщение — сообщение без отправителя.

Последние два вида стрелок (взаимодействий) используются крайне редко. В основном они используются для демонстрации взаимодействия имеющихся объектов в данном прецеденте с внешними системами.

**5. Какая диаграмма позволяет моделировать параллельные вычисления?**

Диаграммы деятельности используются для моделирования бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. или диаграмма прецедентов, описывает отношения между актёрами (действующими лицами) и вариантами использования моделируемой системы (ее возможностями).