Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра информационных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №4.1

Технология разработки программного обеспечения

Выполнил: ст. гр. 35

Посевин Р. Р,

Проверил: доц. каф. ИТ

Полетайкин А.Н.

Краснодар,

2021

**Лабораторная работа №4.1**

Тема: Проектирование функциональной структуры программного продукта: объектно-ориентированный подход.

Цель: изучение методики объектно-ориентированного подхода программной инженерии для разработки и описания функциональности разрабатываемого программного обеспечения.

### Задание

1. Проанализировать описание функционирования программной системы, разработанного при выполнении лабораторной работы №4, на предмет выявления набора абстракций предметной области, проектируемой ПС. В качестве предварительных кандидатов в абстракции принять подлежащие, выделенные из текста анализируемого потока событий.
2. Разделить выделенные абстракции на три типа: абстракции сущности, абстракции поведения, абстракции интерфейсы. Результат представить в виде таблицы.
3. Провести классификацию обнаруженных абстракций по классическому подходу. Заполнить таблицу.
4. Проанализировать поведение выделенных абстракций. Выделить возможное поведение каждой абстракции в пределах рассматриваемой функции (потребности, табл. 1). Заполнить таблицу.
5. Построить диаграмму классов UML (class diagram), указывая при этом лишь имена классов без указания свойств и методов класса.
6. На основе анализа описания предметной области, разработанного при выполнении лабораторной работы №1, выявить атрибуты и операции классов. Заполнить секции атрибутов и операций классов.
7. Выбрать в модели классов такой класс, которых характеризуется наиболее частой сменой состояний, и построить для него диаграмму состояния (statechart diagram).
8. На основе анализа функциональных моделей, разработанных при выполнении лабораторной работы №4, для каждого из базовых вариантов использования построить диаграмму деятельности (activity diagram). Для вариантов использования, с которыми связаны несколько действующих лиц, диаграмму деятельности построить в виде дорожек с привязкой к исполнителям конкретных операций алгоритма
9. Для каждого варианта использования выделить список объектов, участвующих во взаимодействии в этом прецеденте, заполнить таблицу.
10. Создать диаграммы последовательности (sequence diagram) для перечисленных прецедентов.
11. Для наиболее сложных диаграмм последовательности создать кооперативные диаграммы (collaboration diagram) и доработать их, если это необходимо.

12. Сделать вывод.

**Тема проекта:** ПС анализа популярности и рейтингования фильмов.

**Ход работы**

При анализе описания функционирования программной системы были выявлены следующие абстракции:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Тип | Описание |
| 1 | Данные о фильмах | Абстракция сущности | Сущность, содержащая входные данные для работы системы |
| 2 | Аккаунт пользователя | Абстракция сущности | Сущность, необходимая для работы пользователя в системе |
| 3 | Отчет о фильмах | Абстракция интерфейса | Сущность, которая содержит информацию о фильмах, содержащихся в системе |
| 4 | Список отзывов | Абстракция сущности | Сущность, которая содержит набор данных, необходимых для формирования статистики |
| 5 | Рейтинг фильмов | Абстракция сущности | Сущность, которая содержит информацию о среднем балле каждого фильма |
| 6 | Пользователь | Абстракция интерфейса | Человек, который пишет отзывы |
| 7 | Администратор | Абстракция интерфейса | Человек, который контролирует работу системы |
| 8 | Добавление отзыва к фильму | Абстракция сущности | Получение отзыва от пользователей и привязка их к фильму |
| 9 | Формирование статистики | Абстракция сущности | Составление отчетов, полученных в ходе работы системы |

Таблица 1 – Абстракции подсистемы

Проведем классификацию обнаруженных абстракций по классическому подходу. Результат отображен в Таблице 2.

|  |  |  |
| --- | --- | --- |
| № | Класс | Список абстракций |
| 1 | Люди | Пользователь, администратор |
| 2 | Места | – |
| 3 | Предметы | Данные о фильмах, аккаунт пользователя, отчет о фильмах, список отзывов, рейтинг фильмов |
| 4 | Организации | – |
| 5 | Концепции | - |
| 6 | События | Добавление отзыва к фильму, формирование статистики |

Таблица 2 – Классификация абстракций.

Проанализируем поведение выделенных абстракций. Выделим возможное поведение каждой абстракции в пределах рассматриваемой функции и составим Таблицу 3.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Поведение | Описание поведения |
| 1 | Данные о фильмах | Элементарное | Информация, которую получают пользователь |
| 2 | Аккаунт пользователя | Локальное | Информация, которая позволяет пользователю работать в системе |
| 3 | Отчет о фильмах | Локальное | Предоставление данных о фильмах, содержащихся в системе |
| 4 | Список отзывов | Локальное | Предоставление данных об отзывах, содержащихся в системе |
| 5 | Формирование статистики | Целевое | Составление отчетов, полученных в ходе работы системы |
| 6 | Администратор | Локальное | Контроль работы системы |

Таблица 3 – Абстракции подсистемы и их поведение.

Построим диаграмму классов UML. Результат приведен на Рисунке 1.

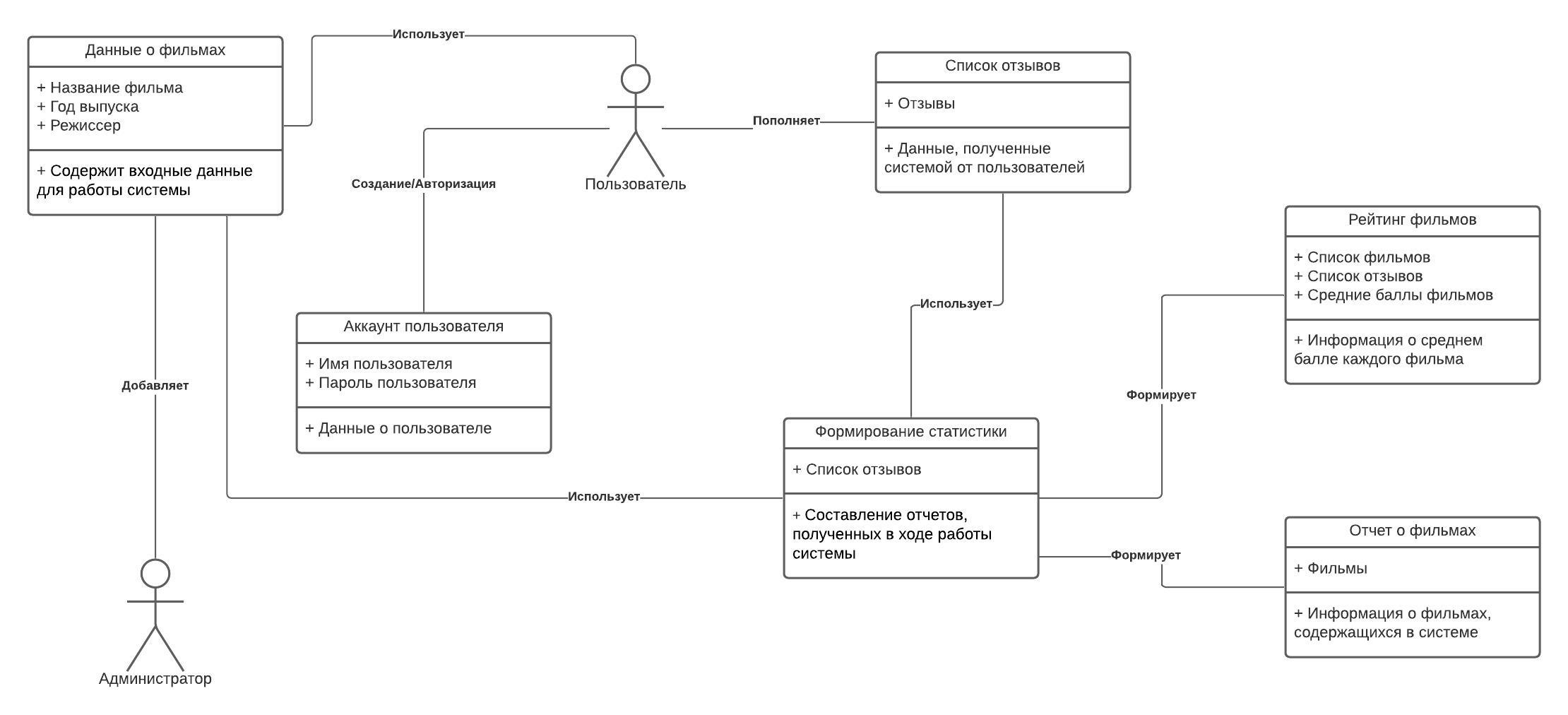


Рисунок 1 – Расширенная диаграмма классов UML.

Выберем класс «Пользовательский интерфейс», который характеризуется наиболее частой сменой состояний, и построим для него диаграмму состояния. Результат представлен на Рисунках 3 и 4.

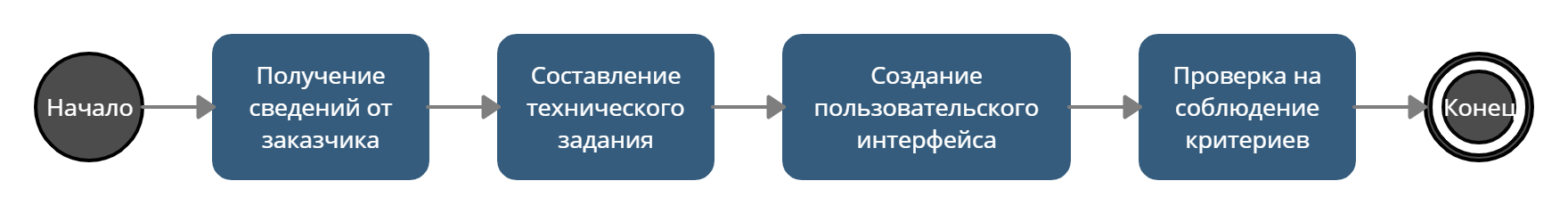


Рисунок 3 – Диаграмма состояний для класса «Пользовательский интерфейс»

На основе анализа функциональных моделей, разработанных при выполнении лабораторной работы №4, для каждого из базовых вариантов использования построим диаграмму деятельности.



Рисунок 4 – Диаграмма деятельности для класса «Пользовательский интерфейс»

Для каждого варианта использования выделим список объектов, участвующих во взаимодействии в этом прецеденте, заполним таблицу 4.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Прецедент | Объект | Описание объекта |
| 1.1 | Поступление данных от заказчика | Заказчик | Предоставляет данные |
| 1.2 | Поступление данных от заказчика | Специалист | Принимает данные |
| 2.1 | Составление технического задания | Специалист | Специалист составляет техническое задание на основе пожеланий заказчика |
| 2.2 | Составление технического задания | Данные | Данные полученные от заказчика |
| 3.1 | Создание пользовательского интерфейса | Специалист | Специалист создаёт пользовательский интерфейс |
| 3.2 | Создание пользовательского интерфейса | Техническое задание | Специалист использует данные из технического задания для создания ПИ |
| 4.1 | Проверка на соответствие критериям заказчика и расположения структурных элементов | Специалист | Специалист сверяет ПИ на соответствие с ТЗ и критериями размещения структурных элементов |
| 4.2 | Проверка на соответствие критериям заказчика и расположения структурных элементов | Данные | Данные полученные из ТЗ и о правилах размещения структурных элементов |

Таблица 4 – Список объектов для каждого потока событий

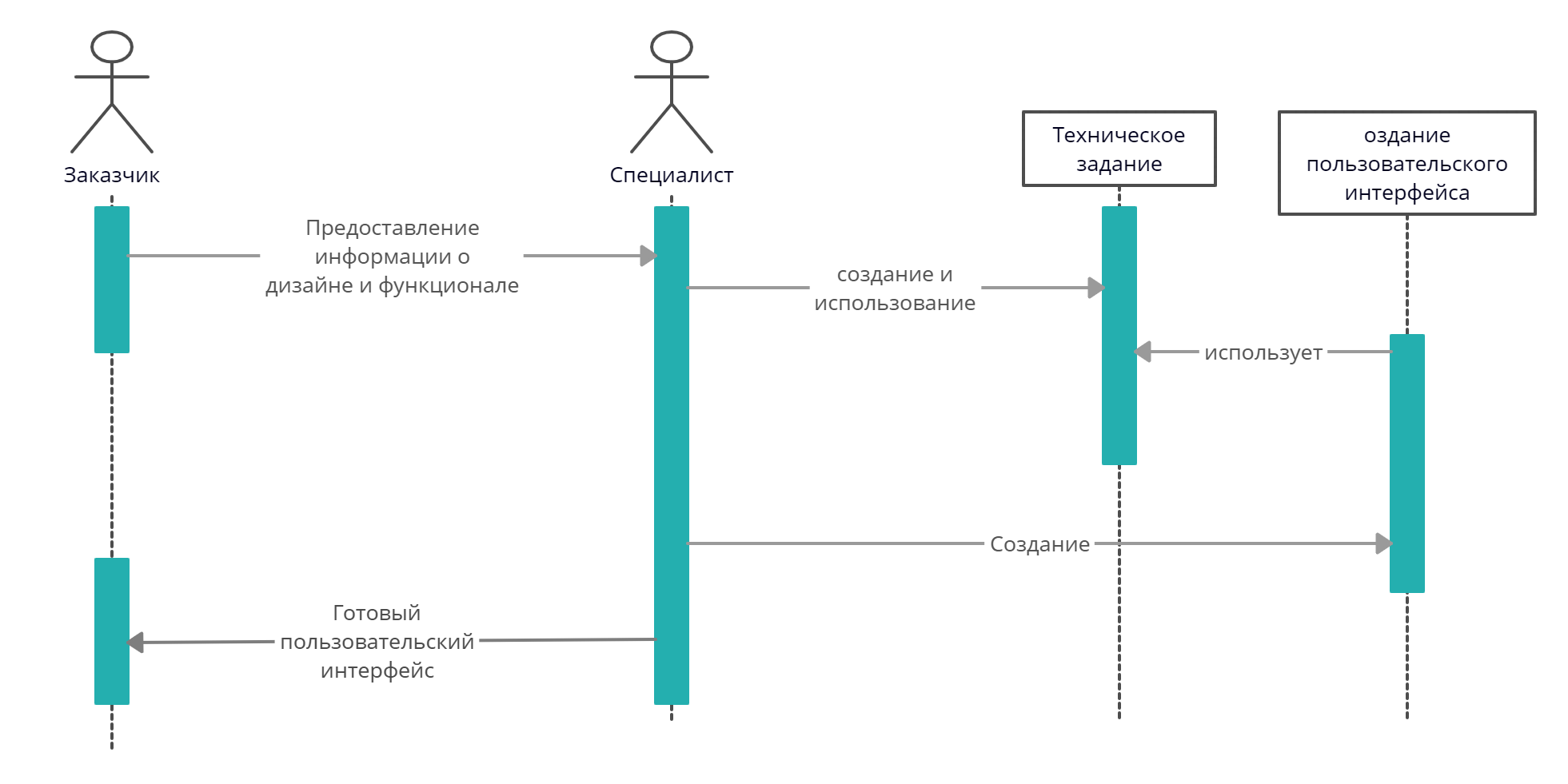


Рисунок 5 – Диаграмма последовательности для прецедента «Создание пользовательского интерфейса»

Вывод: я изучил методики объектно-ориентированного подхода программной инженерии для разработки и описания функциональности разрабатываемого программного обеспечения на основе создания пользовательских интерфейсов по описанию заказчика.