



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра практической и прикладной информатики (ППИ)

ИТОГОВЫЙ ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ
по дисциплине
Анализ и концептуальное моделирование систем»

Выполнил студент группы ИНБО-10-21

Четырин Б.П.

Приняла ассистент

Свищёва И.В.

Москва 2023

СОДЕРЖАНИЕ

Практическая работа 1	3
Практическая работа 2	6
Практическая работа 3	11
Практическая работа 4	13
Практическая работа 5	18
Практическая работа 6	24
Практическая работа 7	26
Доклад.....	29
ЗАКЛЮЧЕНИЕ	32

Практическая работа 1

Сбор предварительной информации

Кинотеатр - коммерческое предприятие со зрительными залами, оборудованными для показа фильмов. В зале располагается экран и зрительные места. С точки зрения функционирования или структуры кинотеатра, можно сказать, что он располагает зрительными местами с разным уровнем сервиса, комфортности и, соответственно, оплаты.

В настоящее время люди часто ходят в различные заведения развлекательного характера, в такие как кинотеатр. До появления информационных технологий, людям часто приходилось долгое время стоять в очередях из-за медленной работы кассира. Сейчас благодаря информационным системам процесс работы кассира стал намного быстрее и удобнее.

Описание объекта автоматизации

Сайты по оплате, бронированию, возврату билетов работают оперативно и в срок, поскольку имеют множество автоматизированных процессов для удобного и быстрого взаимодействия пользователя с функциями сайта кинотеатра.

Таблица 1 – описание видов кинотеатров

Наименование	Описание
Формула кино	Кинотеатр, специализирующийся на продаже билетов онлайн, внутри кинотеатра. Просмотра кино на большом экране в помещении. Также на продаже еды и напитков на сеанс. Имеет внутри здания развлекательные места и chill зоны для времяпрепровождения перед сеансом
Кинопоиск	Онлайн кинотеатр, специализирующийся на продаже подписок для просмотра фильмов в сети интернет

Описание объекта автоматизации

Таблица 2 – сравнение функционала сервисов других компаний

№	Название	Просмотр сеансов	Возможность возврата билета/подписки	Возможность ночного просмотра фильма	Стоимость просмотра фильмов	Удобство Покупки услуги просмотра фильма
	Формула кино					
	Кинопоиск					

Таблица 3 – функционал проектируемой системы

Наименование	Краткое описание	Роли
Просмотр сеанса	Позволяет пользователю ознакомиться с текущей информацией сеанса	Пользователь/Клиент
Продажа билета/подписки онлайн	Позволяет пользователю/клиенту быстро купить услугу кинотеатра	Кассир, Сайт, Банк
Электронное использование билета	Электронный билет уменьшает время на проход к сеансу фильма	Клиент, Проверяющий
Онлайн возврат билета	Позволяет из дома вернуть купленный билет по какой-то причине	Клиент, Банк, Сайт
Просмотр зоны посадки онлайн	Клиент может увидеть модель зала и занятые в нём места	Сайт, Пользователь

Ожидаемые результаты моделируемой системы

Результатом моделируемой системой получится сервис удовлетворяющий потребность любого клиента/пользователя, позволяющий комфортно и быстро оформлять услугу покупки и возврата билета, а также позволяет ознакомиться с информацией о репертуаре кинотеатра.

Практическая работа 2

Описание функций системы через диаграмму вариантов использования.

Цель работы: изучить основные элементы и правила построения диаграммы вариантов использования.

Задачи: описать функции моделирования организации продажи билетов в кинотеатр с помощью диаграммы использования.

Нотация: UML (Use case diagram).

ПО: Visual Paradigm, Draw.io, Rational Rose. Можно использовать Draw.io веб-версию: <https://app.diagrams.net/#>

Теоретический материал:

Разработка диаграммы вариантов использования преследует цели:

- определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

1. Построить диаграмму вариантов использования по следующему описанию: «Клиент банка может пополнить счет, в случае отсутствия счета предварительно открыв его, или снять деньги со счета, с возможностью его закрытия. В каждом из описанных действий участвует операционист банка и кассир.» Заполнить таблицу на основе полученной диаграммы:

Таблица 4 — Описание взаимодействий актеров и вариантов использования

Актер/ ВИ	Тип связи	Вариант использования
Клиент Банка	Простая ассоциация	Пополнить счет
Клиент Банка	Простая ассоциация	Снять деньги со счета
Клиент Банка	Простая ассоциация	Заккрыть счет
Клиент Банка	Простая ассоциация	Открыть счет
Операционист Банка	Направленная ассоциация	Открыть счет клиента
Операционист Банка	Направленная ассоциация	Заккрыть счет клиента
Кассир	Простая ассоциация	Подтверждение открытие счета клиента
Кассир	Простая ассоциация	Снятие денег со счета клиента
Кассир	Простая ассоциация	Заккрытие счета клиента
Открыть счет	Расширение	Пополнить счет
Пополнить счет	Расширение	Заккрыть счет
Пополнить счет	Расширение	Снять деньги со счета
Заккрыть счет	Расширение	Снять деньги со счета

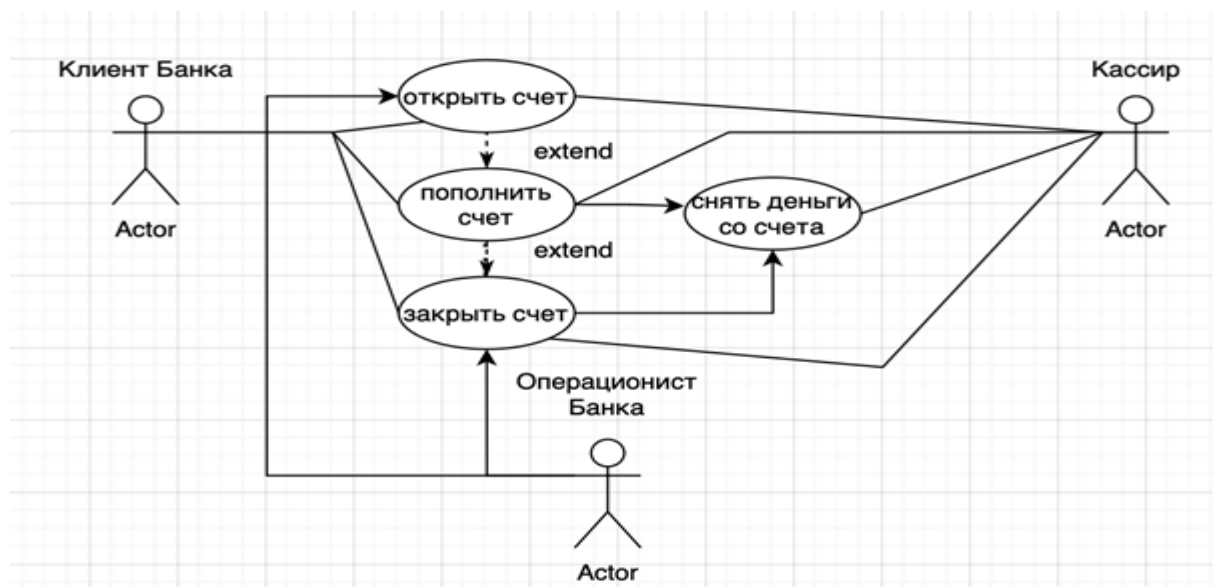


Рисунок 1 – Диаграмма вариантов использования

2. Описать спецификацию функций рассматриваемой системы с учетом индивидуального варианта учебного проекта.

Перед построением диаграммы необходимо задокументировать потоки событий в системе.

Поток событий – процесс обработки данных, реализуемый в рамках одного или нескольких вариантов использования. Описание потока включает информацию о том, какие обязанности возлагаются на актеров, а какие на систему.

При получении услуги клиент кинотеатра проходит сквозь этапы: выборочный этап, этап оформления заказа, этап выдачи услуги.

3. Изобразить спецификацию функций системы, описанной в п.2 через диаграмму вариантов использования + заполнить таблицу «Описание взаимодействий актеров и вариантов использования» по построенной диаграмме.

Таблица 5 — описание взаимодействий актеров и вариантов использования по теме варианта учебного проекта

Актер/ ВИ	Тип связи	Вариант использования
Клиент	Направленная ассоциация	Выбор фильма
Клиент	Направленная ассоциация	Выбор свободного места
Клиент	Направленная ассоциация	Выбор времени сеанса
Клиент	Направленная ассоциация	Выбор ценовой категории
Клиент/Зависимость	Направленная ассоциация	Оплата билета
Кассир/Зависимость	Направленная ассоциация	Принятие денежных средств за билет
Кассир/Зависимость	Направленная ассоциация	Отметить информацию о продаже билета в БД

Продолжение таблицы 5

Выбор фильма	Расширение	Выбор свободного места
Выбор свободного места	Расширение	Выбор времени сеанса
Выбор времени сеанса	Расширение	Выбор ценовой категории
Кассир/Зависимость	Направленная ассоциация	Выдача билета клиенту
Клиент/Зависимость	Направленная ассоциация	Принятие билета

Практическая работа 3

Построение UML – модели системы. Диаграмма классов анализа.

Цель работы: изучить структуру иерархии классов системы.

Задачи: научиться выстраивать структуру основных элементов диаграммы классов анализа с определением видов классов и типов отношений.

ПО: Visual Paradigm, Draw.io, Rational Rose.

Вариант: №26 «Моделирование продажи билета в кинотеатре». **Порядок выполнения работы:** Построить диаграмму классов анализа рассматриваемой системы с учетом индивидуального варианта. **Описание:** На следующем рисунке показан фрагмент диаграммы классов анализа. В правой части диаграммы (начиная с класса «Дорога») расположены классы, описывающие структуру БД на сервере. Несмотря на то, что исходный программный код не будет содержать этих классов, их детальная разработка не менее важна, чем разработка остальных классов. В данном случае правая часть диаграммы представляет концептуальную модель БД.

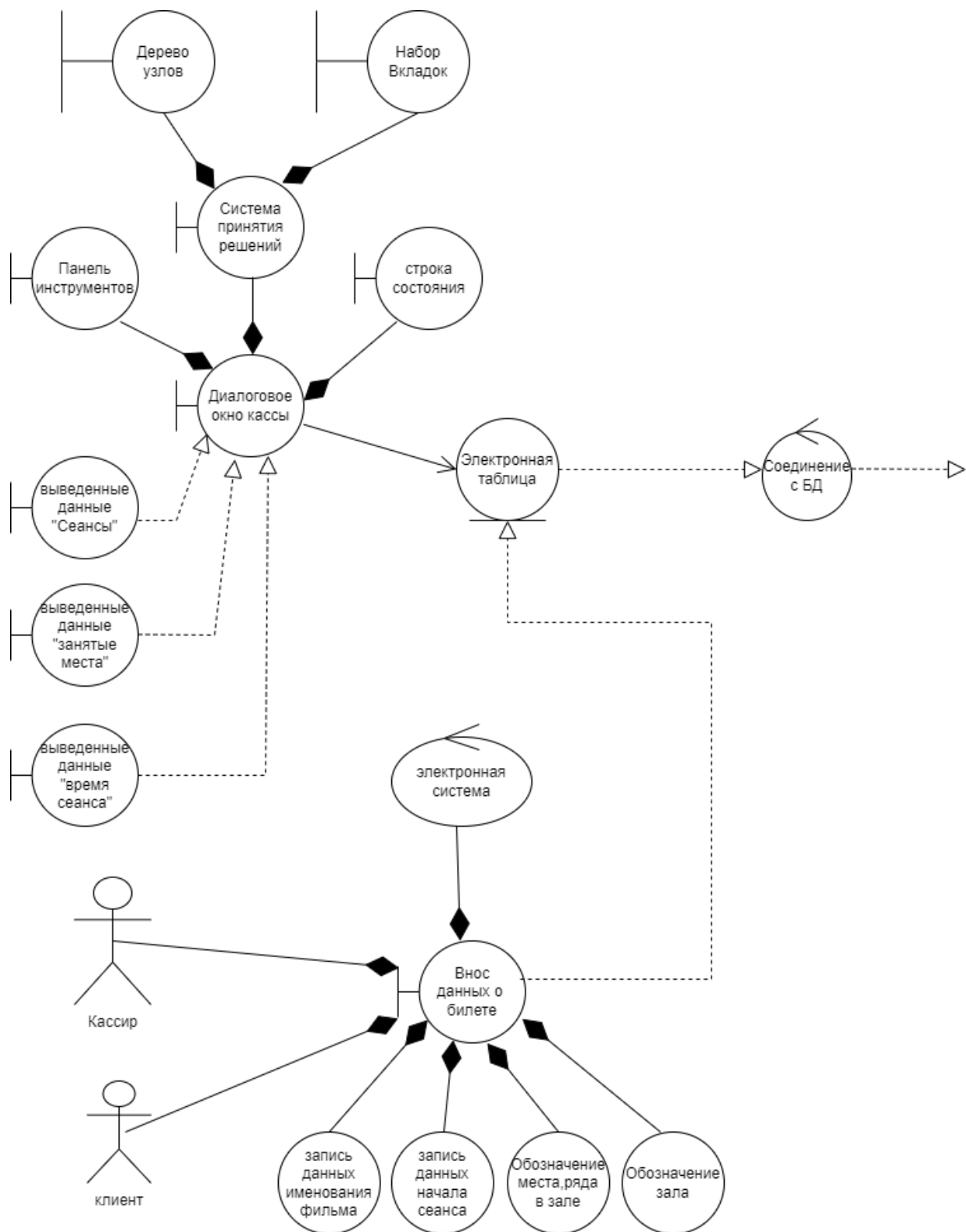


Рисунок 2 – Диаграмма класса анализа продажи билета в кинотеатре

Практическая работа 4

Цель работы: изучить структуру модели анализа, правила построения диаграмм последовательности, кооперации.

Задачи: научиться отображать взаимодействие объектов в динамике. ПО: Visual Paradigm, Draw.io, Rational Rose.

Вариант: №26 Моделирование учёта посетителей в поликлинике

Порядок выполнения работы:

1. Построить диаграмму последовательности по описанию приведенного варианта использования: «Студент хочет записаться на некий семинар, предлагаемый в рамках некоторого учебного курса. С этой целью проводится проверка подготовленности студента, для чего запрашивается список (история) семинаров курса, уже пройденных студентом (перейти к следующему семинару можно, лишь проработав материал предыдущих занятий). После получения истории семинаров объект класса "Слушатель" получает статус подготовленности, на основе которой студенту сообщается результат (статус) его попытки записи на семинар» Заполнить таблицу на основе полученной диаграммы:

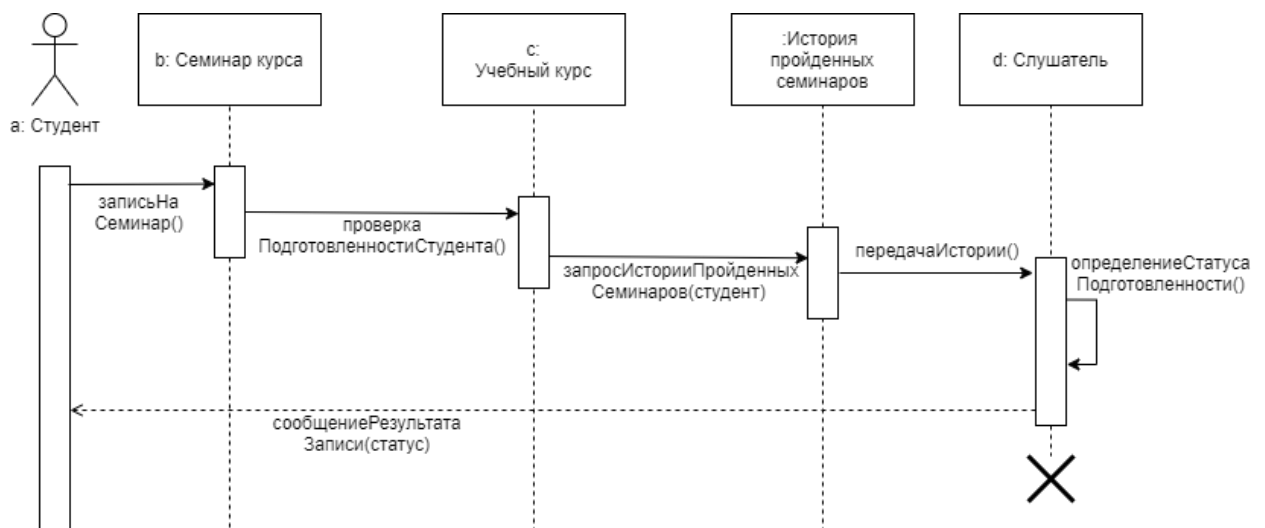


Рисунок 3 – Диаграмма последовательности по приведённому описанию

Таблица 6 — Взаимодействие элементов диаграммы

Отправитель	Тип сообщения	Наименование	Получатель
a: Студент	Синхронное	записьНаСеминар()	b: Семинар курса
b: Семинар курса	Синхронное	проверкаПодготовленностиСтудента()	c: Учебный курс
c: Учебный курс	Синхронное	запросИсторииПройденныхСеминаров(студент)	:История пройденных семинаров
:История пройденных семинаров	Синхронное	передачаИстории()	d: Слушатель
d: Слушатель	Самовывоз	определениеСтатусаПодготовленности()	d: Слушатель
d: Слушатель	Возврат	сообщениеРезультатаЗаписи()	a

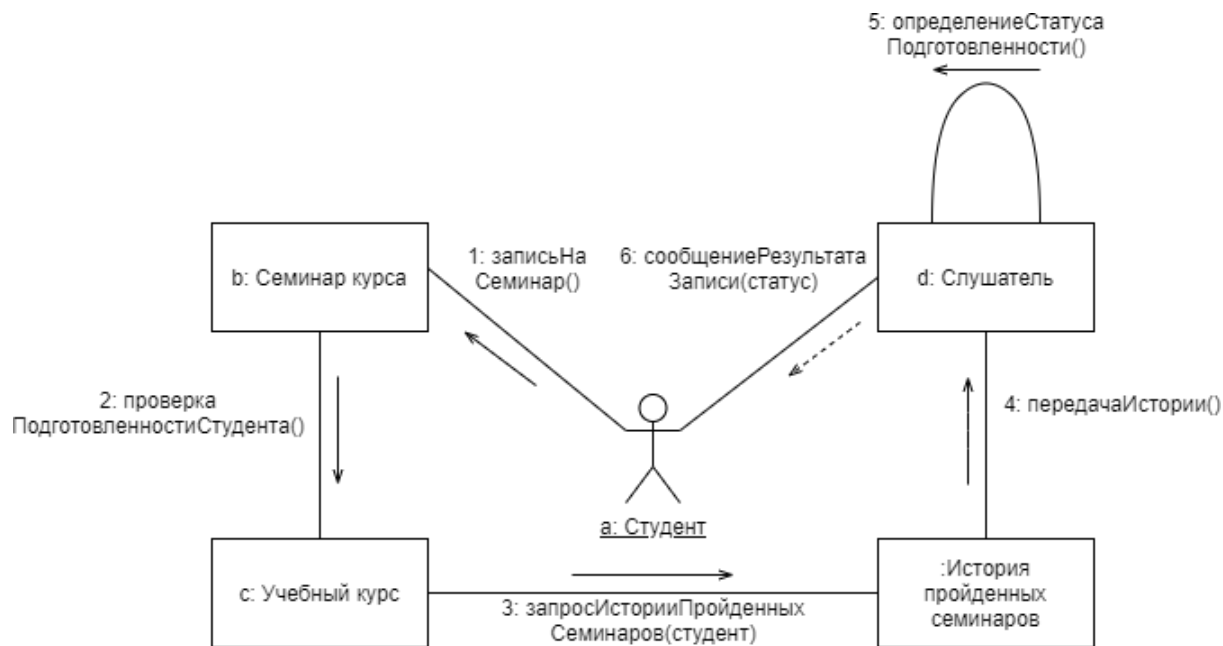


Рисунок 4 – Диаграмма кооперации по приведённому описанию

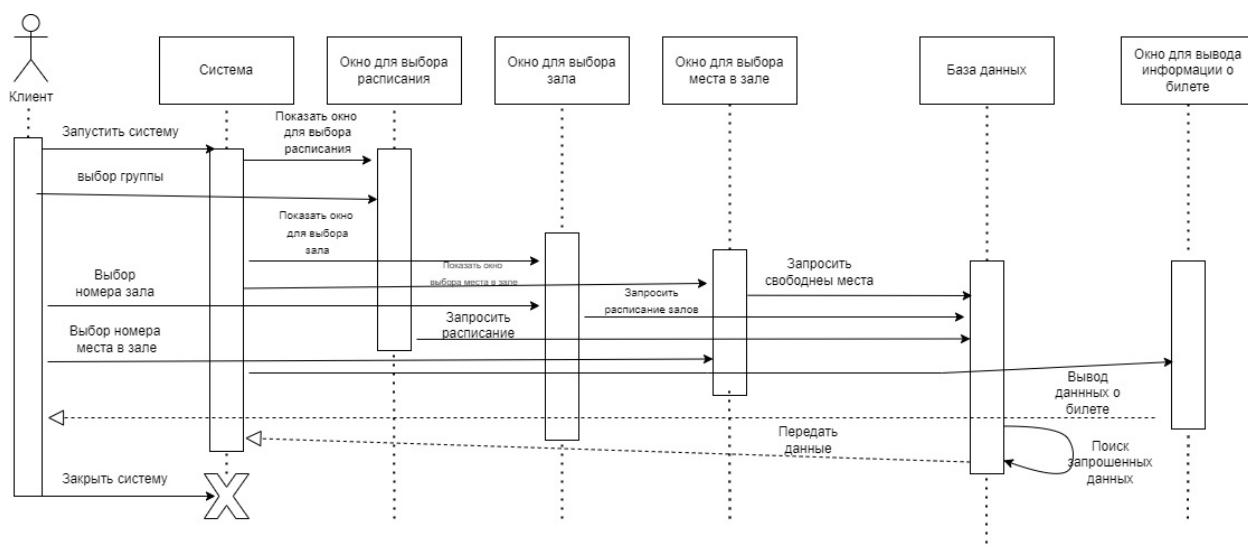


Рисунок 5 – Диаграмма последовательности организации продажи билетов в кинотеатре

Таблица 7 — Взаимодействие элементов диаграммы

Отправитель	Тип сообщения	Наименование	Получатель
Клиент	Синхронное	Запустить Систему	Система
Система	Синхронное	Показать окно для выбора расписания	Окно для выбора расписания
Пользователь	Синхронное	Выбор Группы	Окно для выбора расписания
Система	Синхронное	Показать Окно для Выбора Зала	Окно для выбора зала
Пользователь	Синхронное	Выбор номер зала	Окно для выбора зала
Пользователь	Синхронное	Выбор номера места в зале	Окно для выбора места в зале
Окно для выбора расписания	Синхронное	Запросить расписание	База данных
Окно для выбора зала	Синхронное	Запросить расписание залов	База данных
Окно для выбора места в зале	Синхронное	Запросить свободные места	База данных
База данных	Самовывоз	Поиск запрошенных данных	База данных
:База данных	Возврат	Передать данные(данные)	Система
Система	Синхронное	Вывод данных о билете	Окно для вывода Информации о билете
Пользователь	Синхронное	Заккрыть систему()	Система

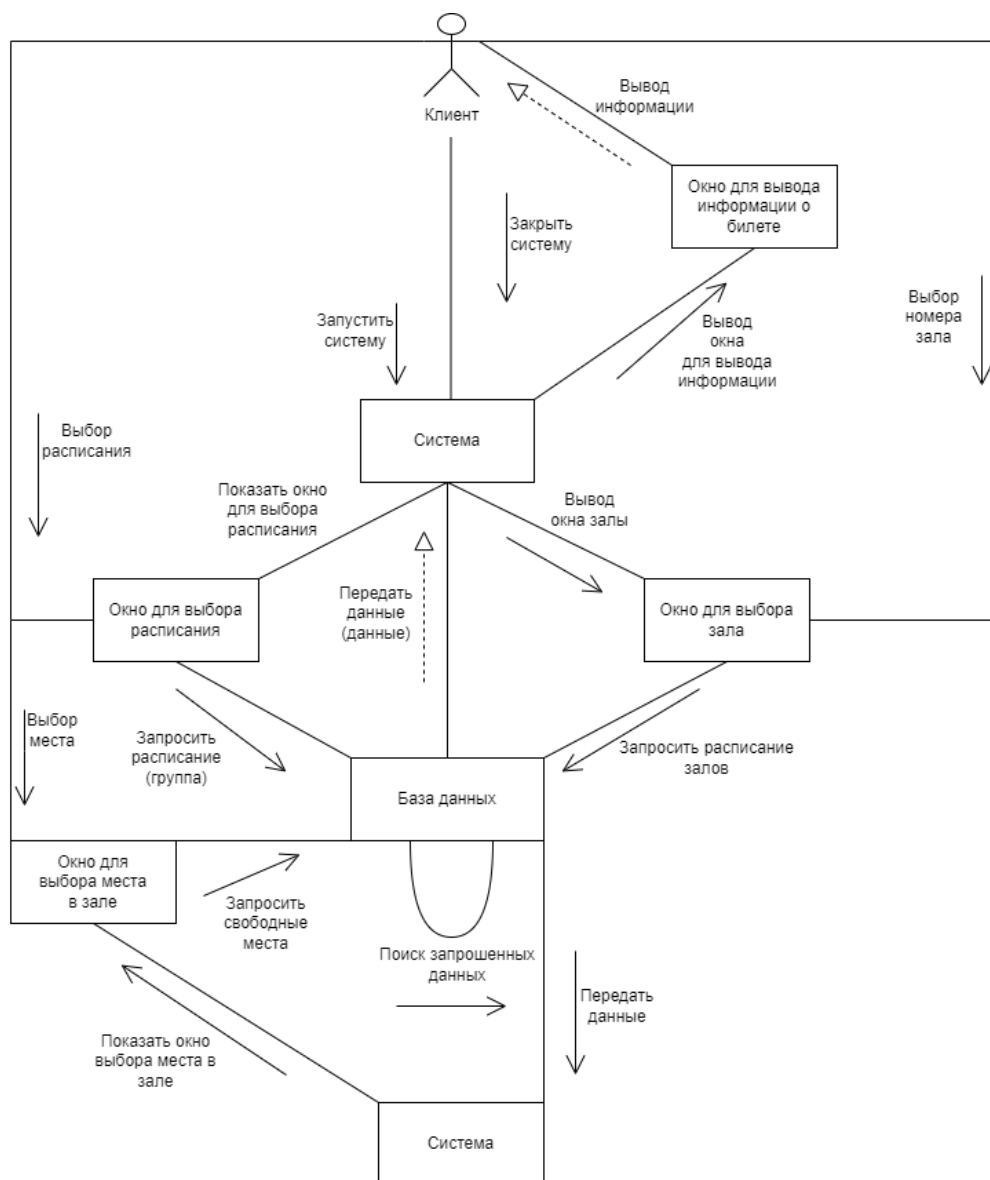


Рисунок 6 – Диаграмма кооперации организации продажи билетов в кинотеатре

Практическая работа 5

Построение UML – модели системы. Диаграмма классов.

Цель работы: изучить структуру модели проектирования, правила построения диаграммы классов.

Задачи: описать сервисные функции исследуемой системы.

ПО: Visual Paradigm, Draw.io, Rational Rose.

Теоретический материал:

Диаграмма классов представляет собой логическую модель статического представления моделируемой системы. Задача заключается в том, чтобы представить поведение более детально на логическом уровне.

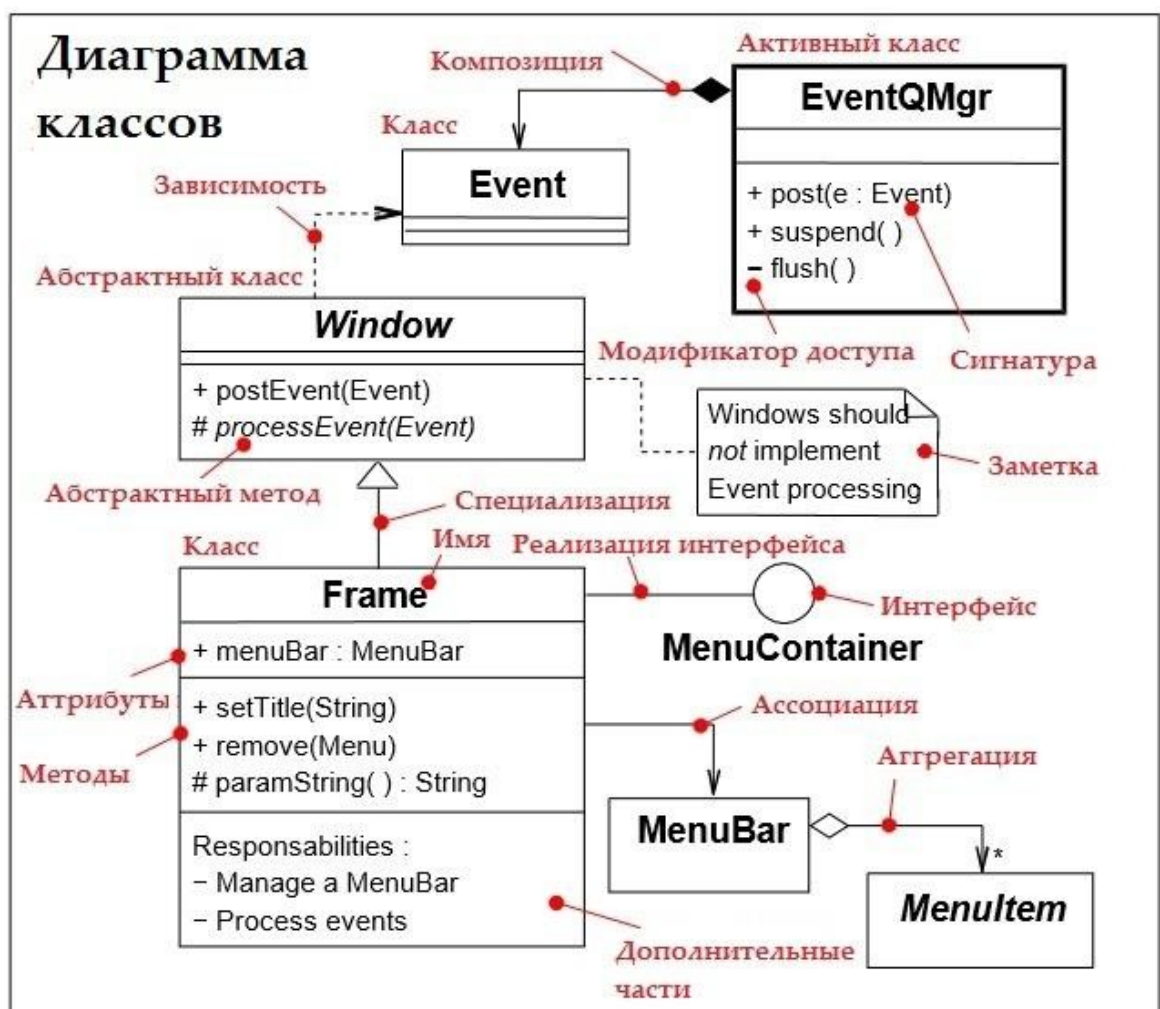


Рисунок 7 – Диаграмма классов

Описание методологии моделирования классов в языке UML.

Объект представляет собой экземпляр класса – особую сущность, которая имеет заданные значения атрибутов и операций.

Атрибут – это свойство класса. Атрибуты описывают перечень значений, в рамках которых указываются свойства объектов (т.е. экземпляров) этого класса. Класс может не иметь атрибутов или содержать любое их количество. Имена атрибутов, состоящие из одного слова, принято обозначать строчными буквами. Если имя состоит из нескольких слов, то эти слова объединяются, и каждое слово, за исключением первого, начинается с прописной буквы. UML позволяет отображать дополнительную информацию об атрибутах. В изображении класса можно указать тип для каждого значения атрибута. Перечень возможных типов включает строку, число с плавающей точкой, целое число, логическое значение и другие перечислимые типы. Для отображения типа используется двоеточие, которое отделяет имя атрибута от его типа. Здесь же можно указать значение атрибута по умолчанию.

Операция – это то, что может выполнять класс, либо то, что вы (или другой класс) можете выполнять над данным классом. Подобно имени атрибута, имя операции записывается строчными буквами, если это однослово. Если имя состоит из нескольких слов, они соединяются, и все слова, кроме первого, пишутся с прописной буквы. Список операций начинается ниже линии, отделяющей операции от атрибутов.

Помимо дополнительной информации об атрибутах, можно отобразить дополнительную информацию об операциях. В скобках, следующих за именем операции, можно указать параметр операции и его тип. Один из типов операций, *функция*, по окончании работы возвращает значение. В этом случае можно указать возвращаемое значение и его тип.

Для ассоциации, агрегации и композиции может указываться

кратность (англ. multiplicity), характеризующая общее количество экземпляров сущностей, участвующих в отношении. Она, как правило, указывается с каждой стороны отношения около соответствующей сущности.

Кратность может указываться следующими способами:

* – любое количество экземпляров, в том числе и ни одного;

целое неотрицательное число – кратность строго фиксирована и равна указанному числу (например: 1, 2 или 5);

диапазон целых неотрицательных чисел "первое число .. второе число" (например: 1..5, 2..10 или 0..5);

диапазон чисел от конкретного начального значения до произвольного конечного "первое число .. *" (например: 1..*, 5..* или 0..*); перечисление целых неотрицательных чисел и диапазонов через запятую (например: 1, 3..5, 10, 15..*).

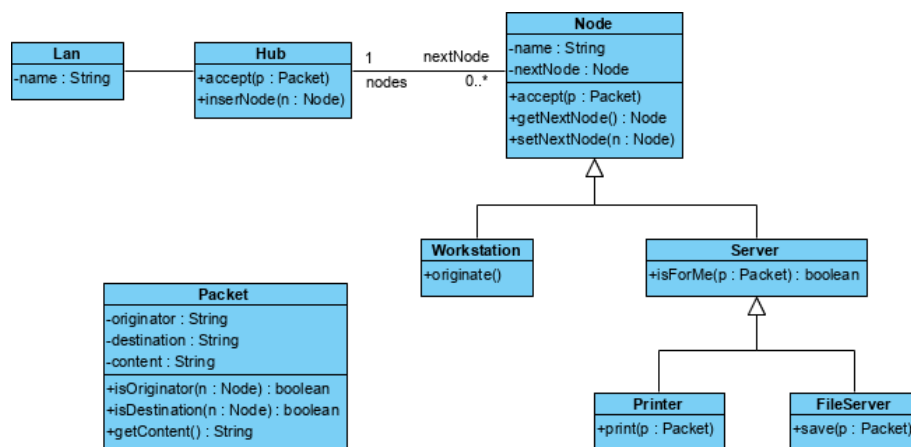


Рисунок 8 – Диаграмма классов

При разработке диаграммы следует придерживаться следующих правил:

1. За основу диаграммы классов при ее разработке берется диаграмма классов анализа.

2. Для классов должны быть определены и специфицированы все атрибуты и методы. Их спецификация, как правило, выполняется с учетом выбранного языка программирования.

3. При определении методов рекомендуется использовать сообщения с ранее разработанных диаграмм последовательности и коммуникации.

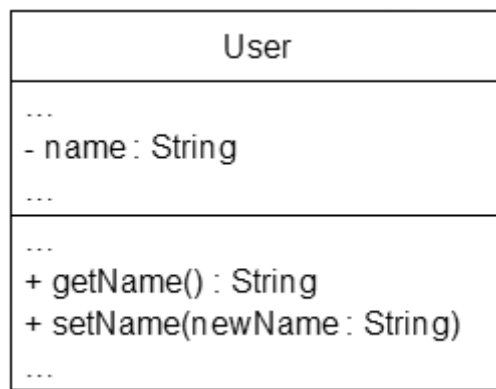
4. Детальное проектирование граничных классов, как правило, не требуется. Большинство современных средств разработки поддерживает визуальную разработку интерфейса системы – меню, диалоговых форм, элементов диалоговых окон, панелей инструментов и т. д. В качестве исходных данных для их проектирования служат прототипы пользовательских интерфейсов. В связи с этим при проектировании таких классов основное внимание следует уделять особенностям отображения информации и специфичным операциям, которые возникают при диалоге пользователя с системой. Граничные классы, определяющие интерфейс взаимодействия с другими системами, требуют детального проектирования.

5. Для проектирования классов-сущностей можно применять подходы, используемые при проектировании БД, особенно в том случае, если данные будут храниться в таблицах БД. Если представление данных в БД и классах отличается друг от друга и в качестве хранилища информации будет применяться реляционная база данных, то рекомендуется разработать отдельную диаграмму классов, описывающую состав и структуру БД.

6. Несмотря на то, что каждому объекту при выполнении программы автоматически назначается уникальный идентификатор, рекомендуется для классов-сущностей явно определять атрибуты, хранящие значения первичного ключа.

7. В отличие от реляционных БД поощряется использование в классах многозначных атрибутов в виде массивов, множеств, списков и т. д.

8. Управляющие классы следует проектировать только в случаях крайней необходимости – управления сложным взаимодействием объектов, реализации сложной бизнес-логики и вычислений, контроля целостности объектов и т. п. В противном случае функциональность этого класса лучше распределить между соответствующими граничными классами и классами-сущностями.



9. Ввиду большого количества классов в системе рекомендуется диаграммы классов разрабатывать отдельно для каждого пакета.

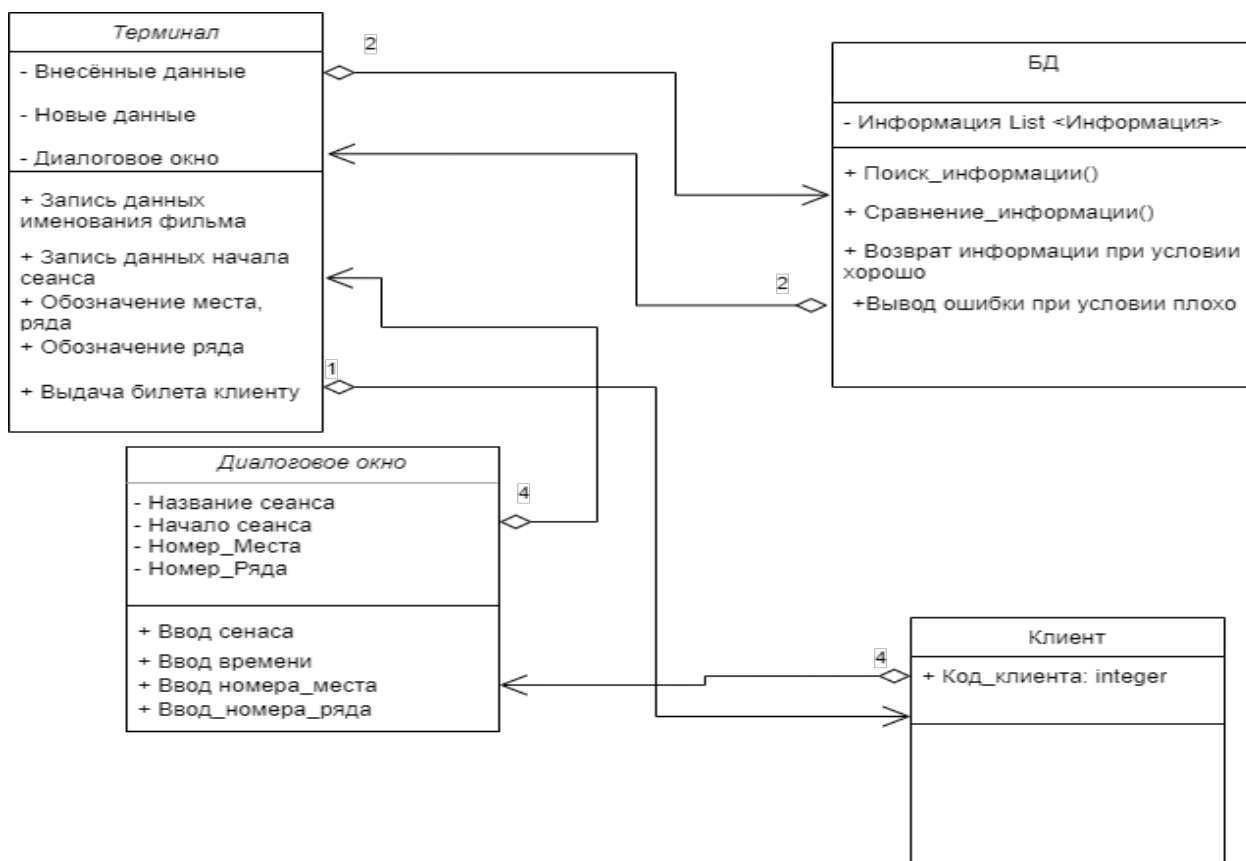


Таблица 8 – Описания классов диаграммы

Название класса	Описание
Клиент	Класс клиентов с одним атрибутом – код клиента
Диалоговое окно	Класс функций, который предоставляет информацию клиенту
Терминал	Класс, обрабатывающий введённые данные, хранит их
База данных	Класс, ищущий сходства с введёнными данными и данными из таблицы

Таблица 9 – Взаимодействия между классами

Класс	Кратность	Тип отношения	Класс
Клиент	4	Агрегирование	Диалоговое окно
Диалоговое окно	4	Агрегирование	Терминал
Терминал	2	Агрегирование	БД
БД	2	Агрегирование	Терминал
Терминал	1	Агрегирование	Клиент

Практическая работа 6

Построение UML – модели системы. Диаграмма деятельности.

Цель работы: научиться строить усовершенствованные блок-схемы с параллельными процессами.

Задачи: описать все системные операции и последовательность состояний и переходов в рассматриваемой системе.

ПО: Visual Paradigm, Draw.io, Rational Rose.

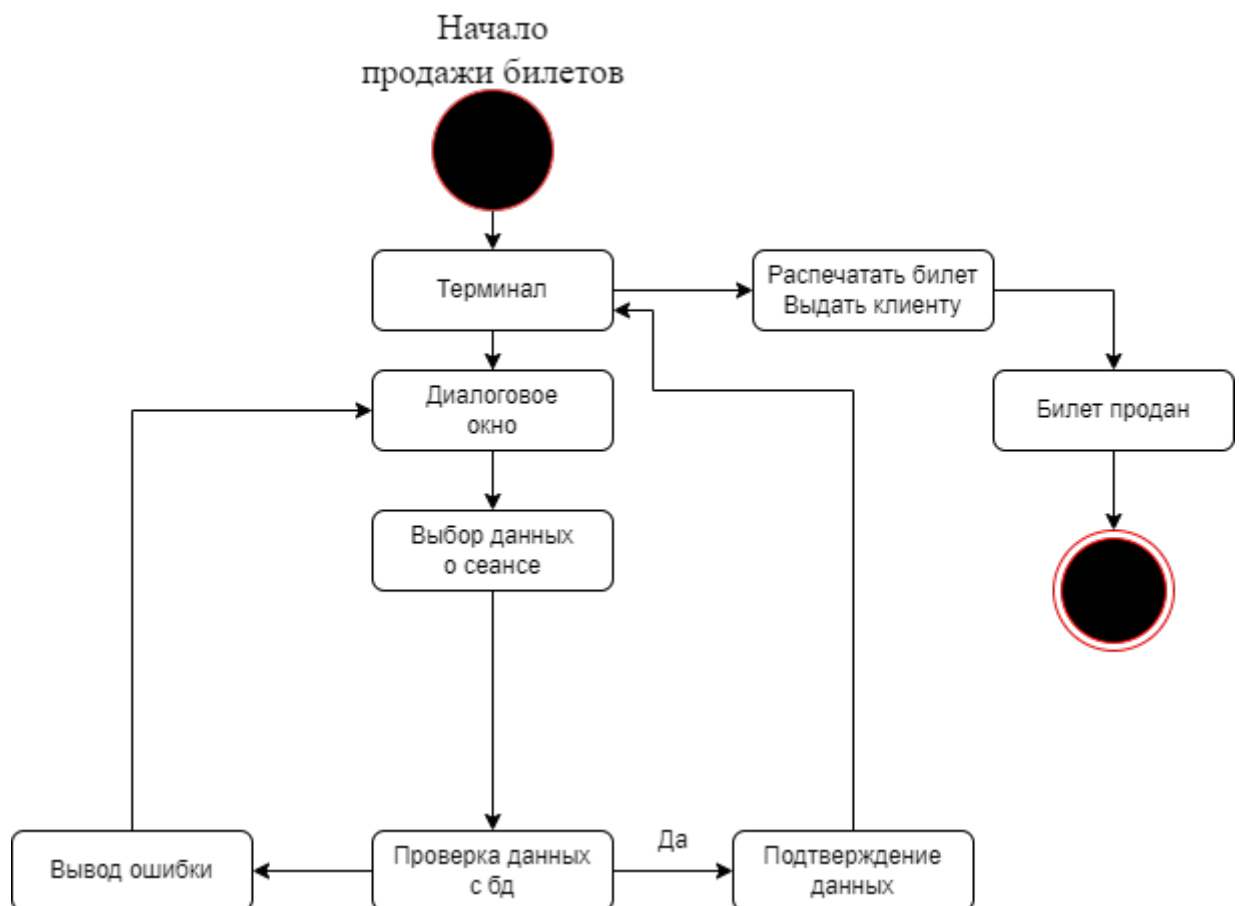


Рисунок 11 – Диаграмма состояний продажи билета в кинотеатре

С начала клиент должен подойти к терминалу, после активации терминала всплывает диалоговое окно для пользователя. В этом диалоговом окне клиент выбирает нужные ему данные для формирования билета. После этого данные проверяются с основной базой данных о сеансах. Если возникли

ошибки в сравнении полученных данных, клиенту отображается диалоговое окно об ошибке. Если при проверке не возникло ошибок, эти данные вносятся в билет, отправляются в терминал, печатающий билет.

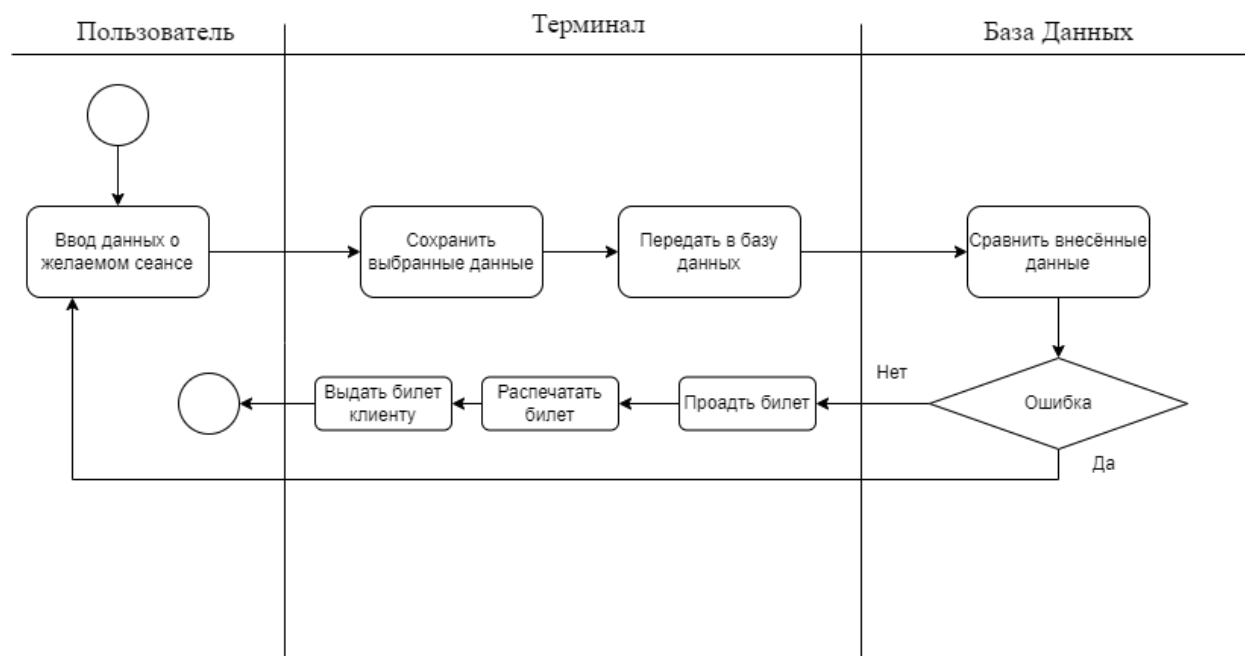


Рисунок 12 – Диаграмма деятельности продажи билета в кинотеатре

С начала клиент должен подойти к терминалу, после активации терминала всплывает диалоговое окно для пользователя. В этом диалоговом окне клиент выбирает нужные ему данные для формирования билета. После этого данные проверяются с основной базой данных о сеансах. Если возникли ошибки в сравнении полученных данных, клиенту отображается диалоговое окно об ошибке. Если при проверке не возникло ошибок, эти данные вносятся в билет, отправляются в терминал, печатающий билет.

Практическая работа 7

Построение UML – модели системы. Диаграмма компонентов, развертывания.

Цель работы: научиться строить модель реализации.

Задачи: построить модель реализации с помощью диаграмм компонентов и развертывания с рассмотрением основных элементов и правил построения.

Нотация: UML (Component diagram; Deployment diagram).

ПО: Visual Paradigm, Draw.io, Rational Rose.

1. Построить диаграмму компонентов продажи билетов в кинотеатре (Рисунок 1):

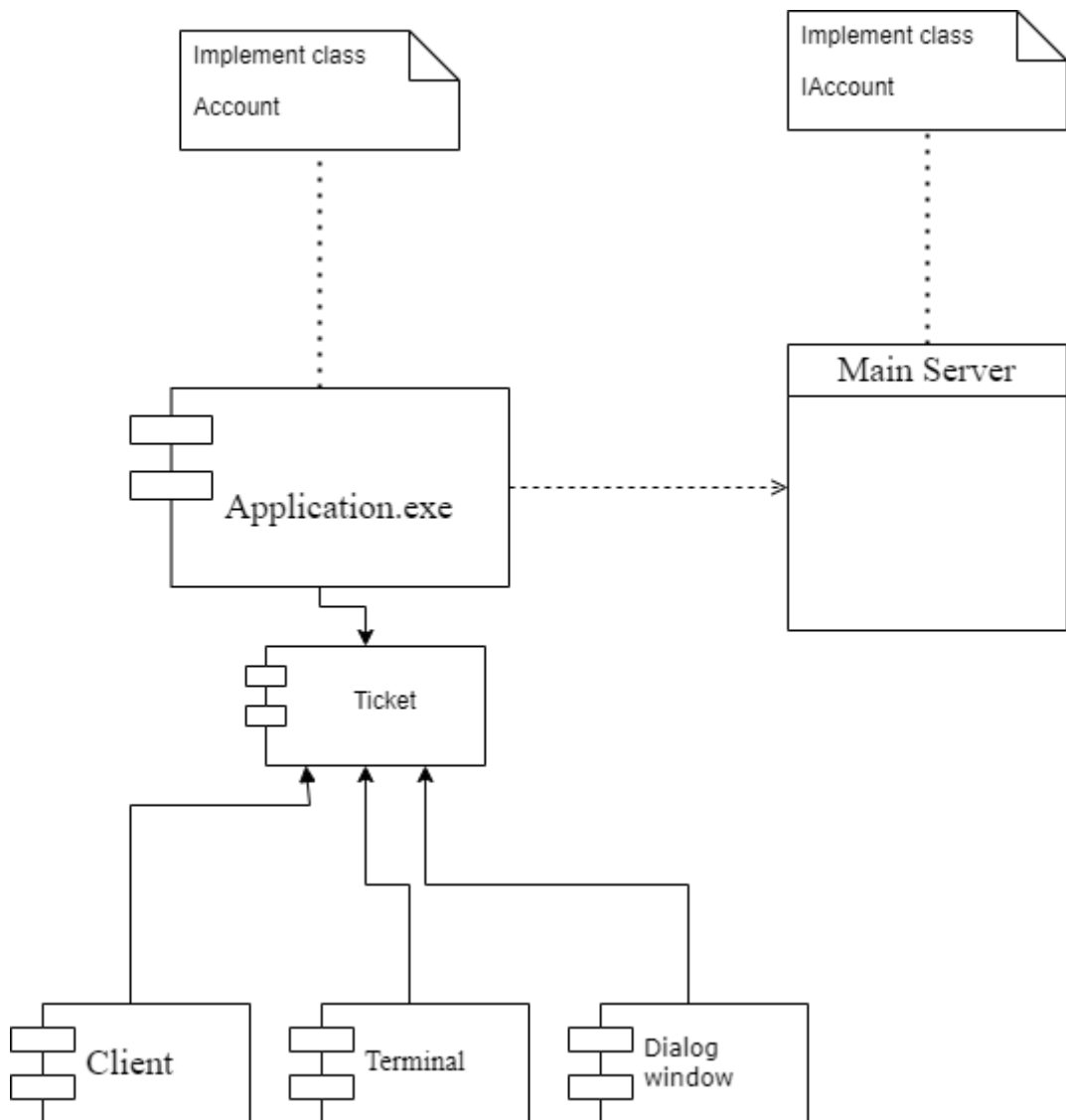


Рисунок 13 – Диаграмма компонентов

Диаграмма компонентов описывает особенности физического представления системы. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Application.exe – это главная программа. Реализует базовую логику работы программного приложения и содержит ссылки на другие компоненты модели.

2. Построить диаграмму развертывания рассматриваемой системы с учётом индивидуального варианта учебного проекта (Рисунок 2):

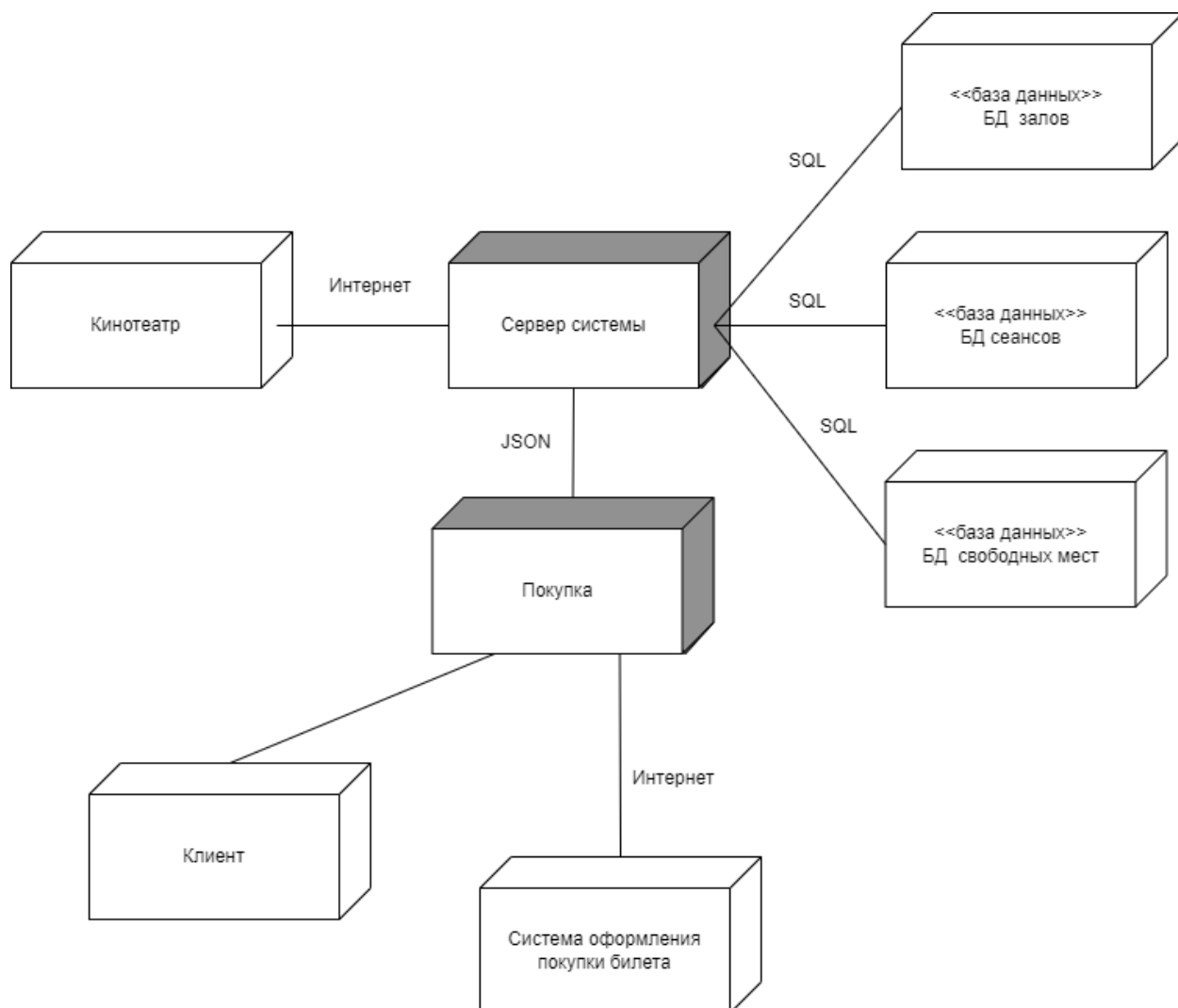


Рисунок 14 – Диаграмма развертывания

Application.exe – это компонент, который представляет физическую часть системы. Компоненты являются файлами с исходным кодом классов, библиотеки, исполняемые модули.

Продажа билета – спецификация задачи. Может содержать определение класса, его атрибутов и операций, которые предполагается использовать в независимом потоке управления.

Все остальные элементы – это тела задач. Могут содержать реализацию операций класса, которые имеют независимый поток управления.

Доклад

Токенизация относится к процессу разбиения текста на более мелкие токены. Это может включать разбиение абзацев на предложения, а предложений на слова, части слов или даже символы. Это фундаментальный шаг к созданию словаря, необходимого для выполнения любой задачи в НейроЛингвистическогоПрограммирования.

Одним из наиболее важных соображений при токенизации является определение границ. Например, в английском языке слова разделяются пробелами, но это может быть неверно в других языках. В зависимости от уровня, на который мы хотим разбить текст, мы можем выбрать токенизацию слов, токенизацию символов или токенизацию вложенных слов.

В естественных языках слова могут принимать различные формы, которые изменяют их грамматическое употребление, но не их семантическое значение. Например, travel, travelling, travels и travelled имеют разное употребление, но схожие значения. В области НЛП эти формы слова известны как словоизменяемые формы. Для IR часто желательно привести все эти слова к их базовой форме.

Целью как стеммирования, так и лемматизации является генерация базовой формы изменяемых слов в тексте. Использование эвристического подхода с использованием стемминга является более грубым путем отсечения окончания слова для получения его базовой формы. Лемматизация более сложна и для достижения того же результата использует словарный запас и морфологический анализ слов.

Часть речи, или просто PoS, - это категория слов со схожими грамматическими свойствами. В английском языке мы обычно выделяем девять частей речи, таких как существительное, глагол, артикль, прилагательное и другие. Мы можем использовать одно и то же слово в

предложении как существительное или как глагол. Например, посмотрите на использование слова парк в этом предложении

По сути, пометка частью речи - это процесс выделения слова в тексте как соответствующего определенной части речи в зависимости от его использования. Обычно мы используем основанный на правилах или стохастический алгоритм тегирования для достижения этого автоматически. PoS-теги имеют несколько применений в NLP

Именованный объект относится к объекту реального мира, такому как города, люди или организации. Например, Baeldung, London, Jack Daniel, все они могут быть названы именованными объектами в любом тексте. В некоторых случаях именованные объекты могут также включать временные и числовые выражения, например 400 морских миль или 2020 год. Например, давайте проанализируем следующее предложение:

Распознавание именованных объектов - это процесс идентификации именованных объектов в неструктурированных текстах. Существует несколько основанных на правилах и статистических алгоритмов для автоматического распознавания именованных объектов. Статистическое распознавание именованных объектов обычно включает контролируемые и полууправляемые модели машинного обучения. Это имеет множество применений в IR, таких как классификация контента, индексирование и рекомендации.

Коллокация в основном относится к последовательности токенов, которые встречаются вместе в корпусе чаще, чем то, что мы можем считать совпадением. Это имеет скорее культурное значение, чем грамматическую ориентацию. Например, в обычной практике мы довольно часто используем такие фразы, как “крепкий чай” и “мощный компьютер”. Существует шесть основных типов словосочетаний:

Извлечение словосочетаний - это задача автоматического определения всех словосочетаний в тексте с использованием алгоритма. Один из простейших алгоритмов для определения словосочетаний использует их частоту в корпусе. Более сложные алгоритмы могут использовать методы, основанные на средних значениях и отклонениях, а также точечные взаимные информационные показатели. Это может иметь полезные применения в задачах IR.

Методы извлечения ключевых фраз под наблюдением обычно работают, переформулируя проблему извлечения ключевых фраз в задачи классификации или ранжирования. При использовании подхода классификации нам интересно знать, подходит ли ключевая фраза-кандидат для представления текста или нет. Однако, поскольку это непростая задача, подход ранжирования пытается ранжировать кандидатов попарно на основе их релевантности

Существует несколько традиционных алгоритмов машинного обучения для обучения под наблюдением, которые мы можем использовать здесь. Например, наивный Байесов, деревья принятия решений, машины опорных векторов и многое другое. Однако некоторые конкретные реализации соответствуют всем требованиям больше, чем другие. Например, KEA - это метод бинарной классификации, который использует TF-IDF (t) и должность (d) первое появление для выбора ключевых фраз.

В этой статье мы обсудили фундаментальные концепции IR и NLP. В первую очередь мы сосредоточились на методах и алгоритмах, которые наиболее часто используются для автоматического извлечения ключевых фраз. Наконец, мы рассмотрели процесс и алгоритмы, которые мы можем использовать для идентификации и выбора ключевых фраз.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы была собрана предварительная информация, составлено описание объекта автоматизации, описаны основные функции системы и описаны ожидаемые результаты реализации моделируемой системы.

Изучены основные элементы и правила построения диаграммы вариантов использования. Описаны функции моделирования организации продажи билетов с помощью диаграммы вариантов использования.

Изучена структура иерархии классов системы выстроена структура основных элементов диаграммы классов анализа с определением видов классов и типов отношений.

При выполнении данной практической работы была изучена структура анализа, правила построения диаграмм последовательности, кооперации.

При выполнении данной практической работы была изучена структура модели проектирования, правила построения диаграммы классов, а также были описаны сервисные функции исследуемой системы.

В данной работе я научился строить усовершенствованные блок-схемы с параллельными процессами, а также описал все системные операции и последовательность состояний и переходов в рассматриваемой системе.

В данной работе я научился строить модель реализации, а также построил модель реализации с помощью диаграмм компонентов и развертывания с рассмотрением основных элементов и правил построения.