|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования"МИРЭА - Российский технологический университет"РТУ МИРЭА | |
| Институт информационных технологий (ИТ) | |
| Кафедра вычислительной техники | |

|  |  |
| --- | --- |
| **Отчет по практической работе. Тема:**  **«Книжный магазин»** | |
| **по дисциплине** | |
| **«Проектирование баз данных»** | |
|  | |
| Выполнил студент группы ИВБО-02-19 | Зелякина А.И. |
| Принял | Лабузов Д. Ю. |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | «\_\_»\_\_\_\_\_\_\_2021\_ г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2021\_ г. |  |

2021 г.

Содержание

1. Моделирование в нотациях IDEF0, DFD, IDEF3

1.1Модель IDEF0

1.2Модель DFD

1.3Модель IDEF3

2. Проектирование информационной системы на языке UML

2.1 Диаграмма прецедентов

2.2 Диаграмма последовательности

2.3 Кооперативная диаграмма

2.4 Диаграмма состояний

2.5 Диаграмма активности

2.6 Диаграмма классов

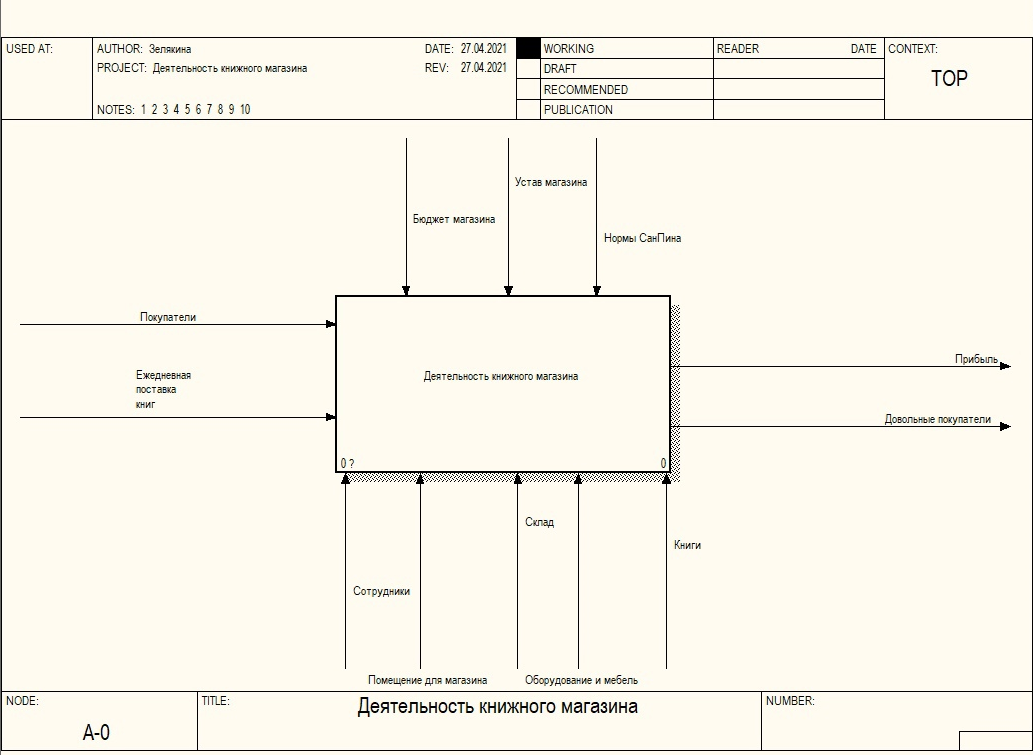
2.7 Диаграмма компонентов

2.8 Диаграмма развертывания

**Моделирование в нотациях IDEF0, DFD, IDEF3**

* 1. **Модель IDEF0**

*Модель IDEF0 по предметной области «*Книжный магазин*».*

**Предметная область «Книжный магазин»** — один из множества видов магазинов, для духовных потребностей человека

*Рисунок 1 —* *Контекстная диаграмма.*

Основной блок – Деятельность книжного магазина.

Входной информацией системы являются:

* **Покупатели**
* **Ежедневная поставка книг**

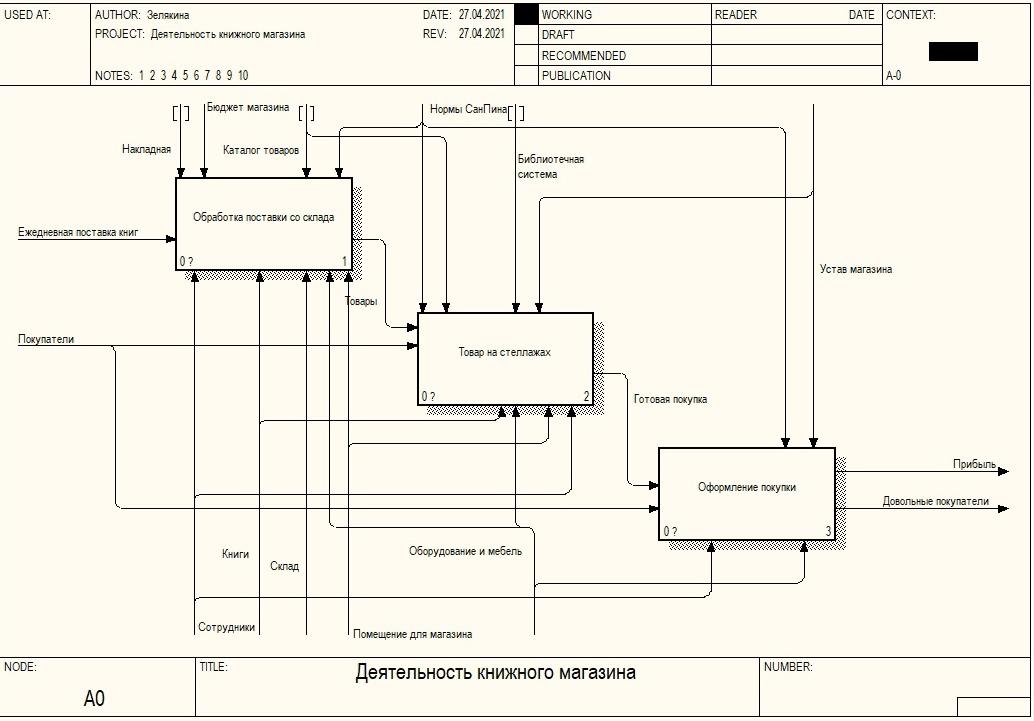
Выходной информацией системы являются:

* **Прибыль**
* **Довольные покупатели**

Механизмы информацией системы являются:

* **Сотрудники**
* **Помещение для магазина**
* **Склад**
* **Оборудование и мебель**
* **Книги**

Управляющие информацией системы являются:

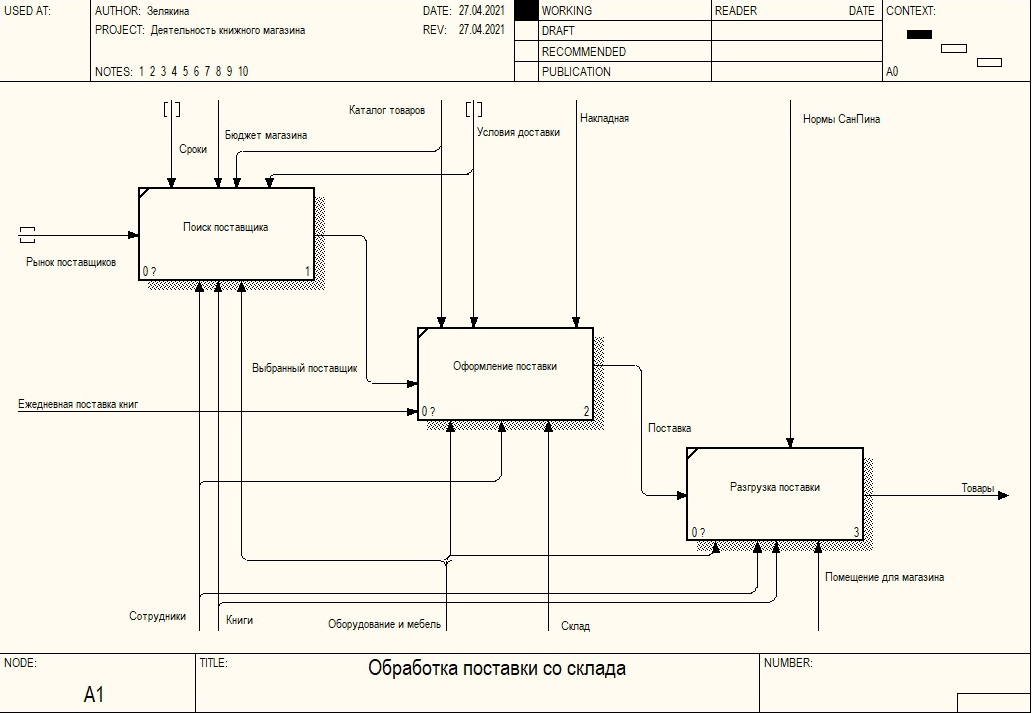
* **Бюджет магазина**
* **Устав магазина**
* **Нормы СанПина**

*Рисунок 2 —* *Декомпозиция контекстной диаграммы.*

Мы декомпозировали общий блок «Деятельность книжного магазина» на связанные между собой элементы. В нашем случае делится на 3 основных этапа:

* **Обработка поставки со склада;**
* **Товар на стеллажах;**
* **Оформление покупки.**

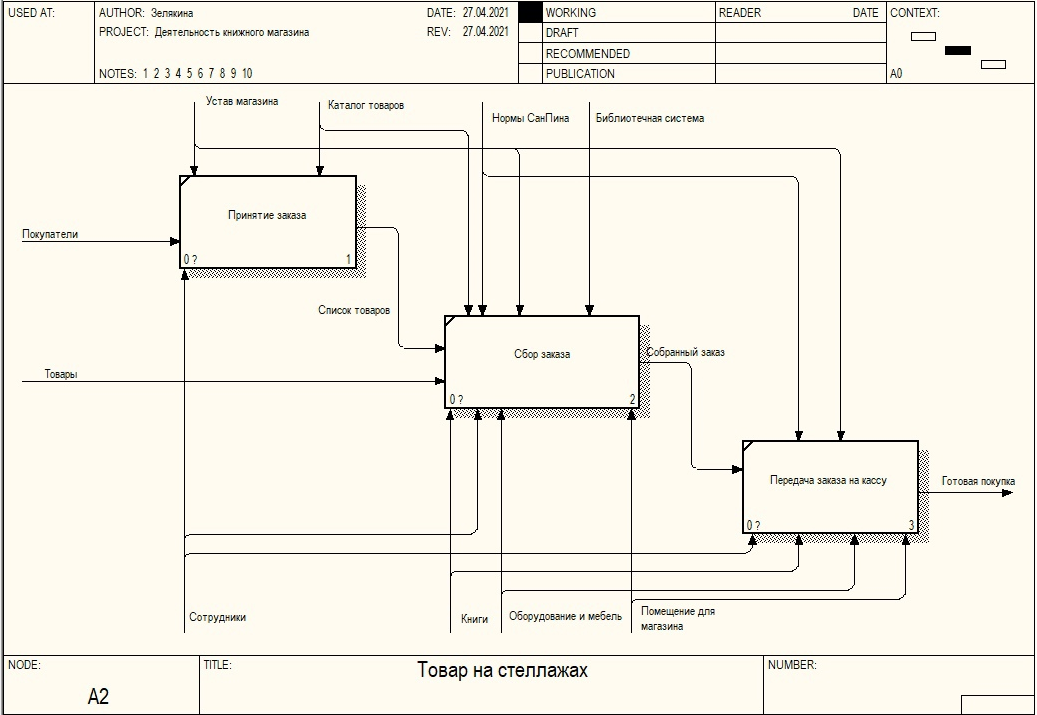
В начале дня сотрудники получают новую поставку книг, а после размещают новый товар на стеллажах магазина, где покупатели выбирают нужный товар и оплачивают на кассе.



*Рисунок 3 —* *Декомпозиция блока “Обработка поставки со склада”*

Блок «Обработка поставки со склада» мы декомпозируем еще на 3 этапов:

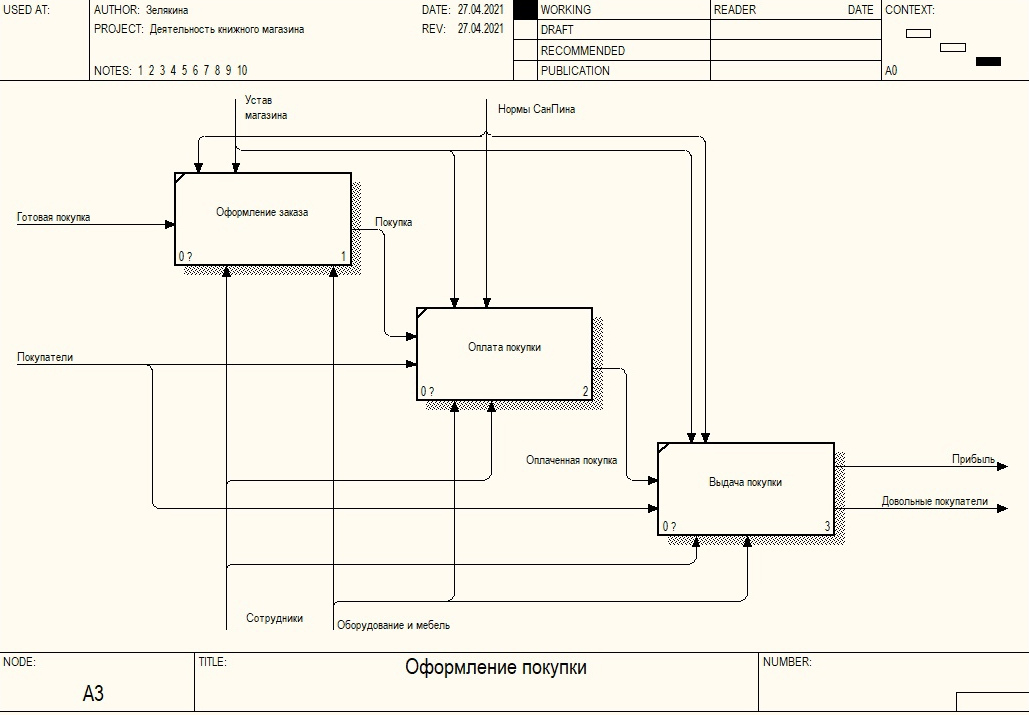
* **Поиск поставщика;**
* **Оформление поставки;**
* **Разгрузка поставки;**

**

*Рисунок 4 —* *Декомпозиция блока “Товар на стеллажах”*

Блок «Товар на стеллажах» мы декомпозируем еще на 3 этапов:

* **Принятие заказа;**
* **Сбор заказа;**
* **Подача заказа на кассу;**

******

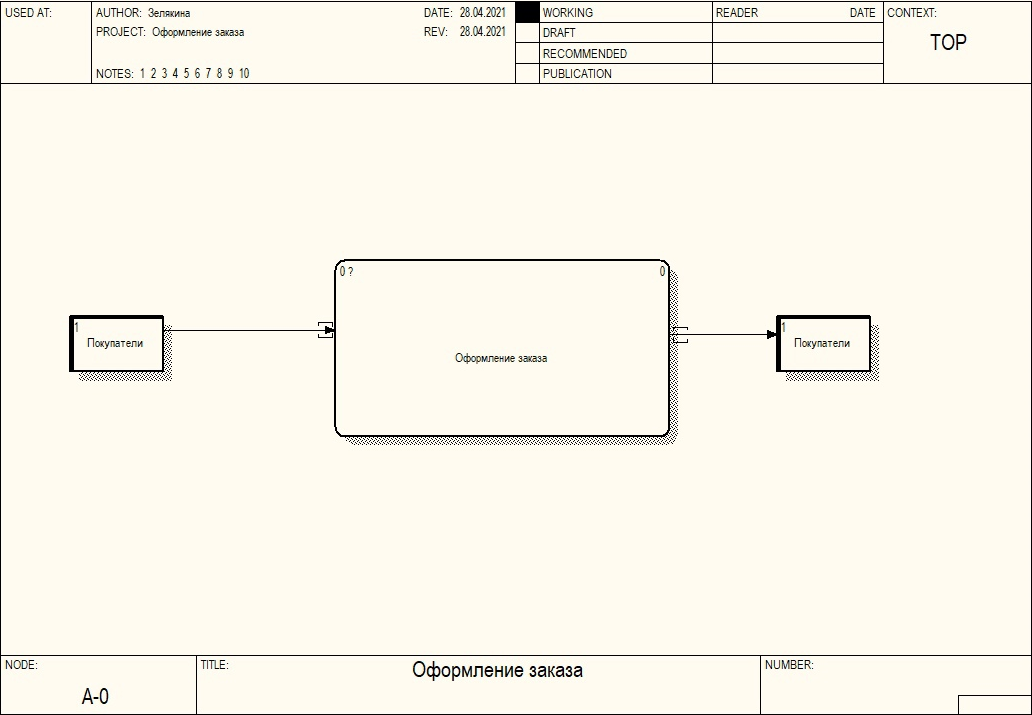
*Рисунок 5 —* *Декомпозиция блока “Оформление покупки”*

Блок «Оформление покупки» мы декомпозируем еще на 3 этапов:

* **Оформление заказа;**
* **Оплата покупки;**
* **Выдача покупки;**

**1.2 Модель DFD**

*Модель DFD по предметной области «*Оформление покупки в книжном магазине*».*

**

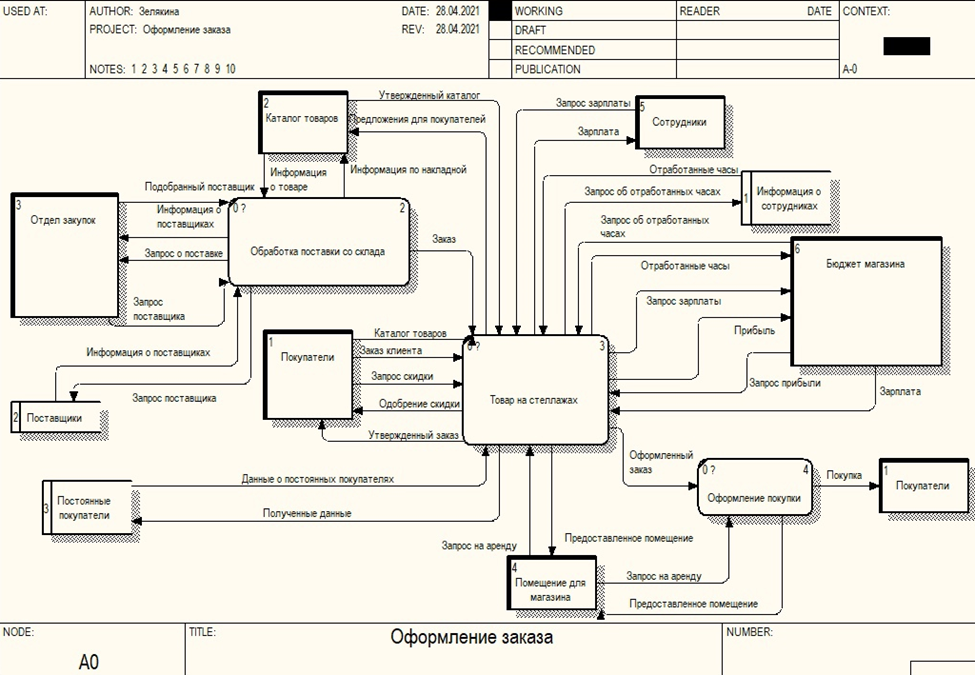
*Рисунок 6—* *DFD. Контекстная диаграмма.*

На данной модели отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

**Внешние сущности** изображают входы в систему и/или выходы из нее.

**Стрелки (потоки данных).** Стрелки описывают движение объектов из одной части системы в другую.

**Хранилище данных.** В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое.

Мы декомпозировали общий блок «Оформление заказа» на связанные между собой элементы.

*Рисунок 7 —* *Декомпозиция контекстной диаграммы*

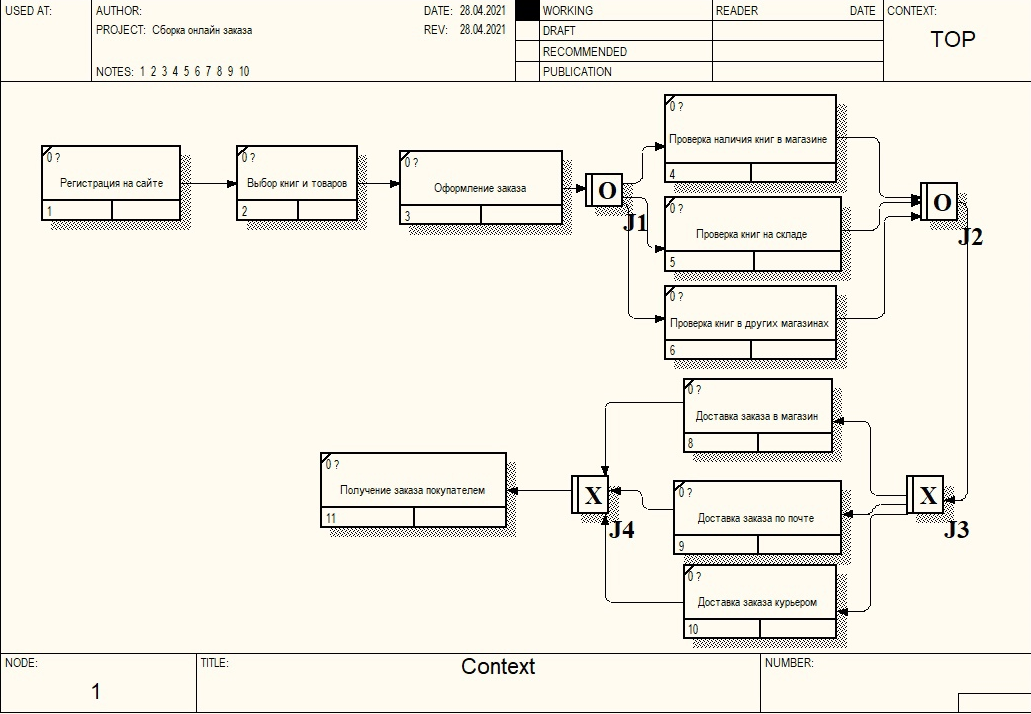
* 1. **Модель IDEF3**

*Модель IDEF3 по предметной области «*Интернет-заказа в книжном магазине*».*

Методология IDEF3 позволяет декомпозировать работу многократно, т. е. работа может иметь множество дочерних работ. Возможность множественной декомпозиции отражается в нумерации работ: номер работы состоит из номера родительской работы, номера декомпозиции и номера работы на текущей диаграмме.

**Слабые связи переходов** изображаются сплошными одинарными стрелками.

**Сильные связи переходов** изображаются двойными однонаправленными стрелками



*Рисунок 8 —* *Декомпозиция контекстной диаграммы*

**2. Проектирование информационной системы на языке UML**

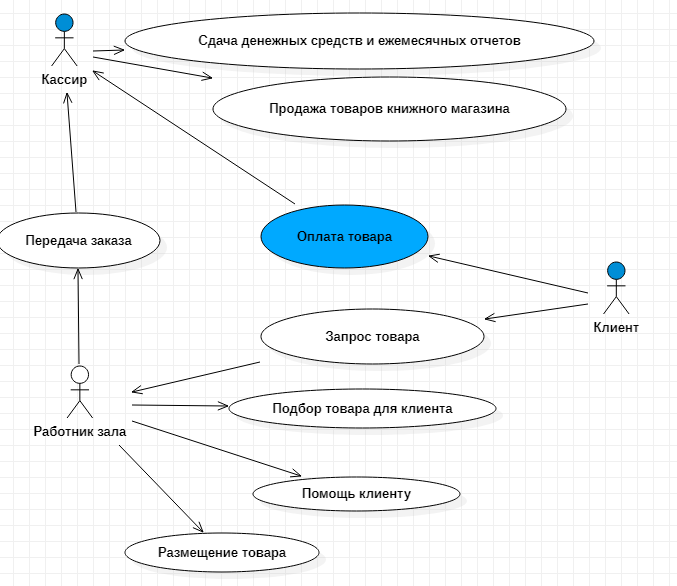
Целью данной работы является освоение технологии проектирования информационных систем с позиции объектно-ориентированного проектирования на основе языка UML.

Последовательность работ следующая, строятся диаграммы логического проектирования, не имеющие прямого отношения к языку программирования. Это диаграммы концептуального моделирования.

**2.1 Диаграмма прецедентов**

### Use case diagram (диаграммы прецедентов) - этот вид диаграмм позволяет создать список операций, которые выполняет система. Каждая такая диаграмма – это описание сценария поведения, которому следуют действующие лица (Actors).

Данный тип диаграмм используется при описании бизнес процессов предметной области, определении требований к будущей программной системе. Отражает объекты как системы, так и предметной области и задачи, ими выполняемые. Окончательный вид диаграммы показан на рисунке 9. На диаграмме находятся Use Case (Вариант использования), Actor (Действующее лицо). Так же были добавлены ассоциации и описания ко всем элементам диаграммы.

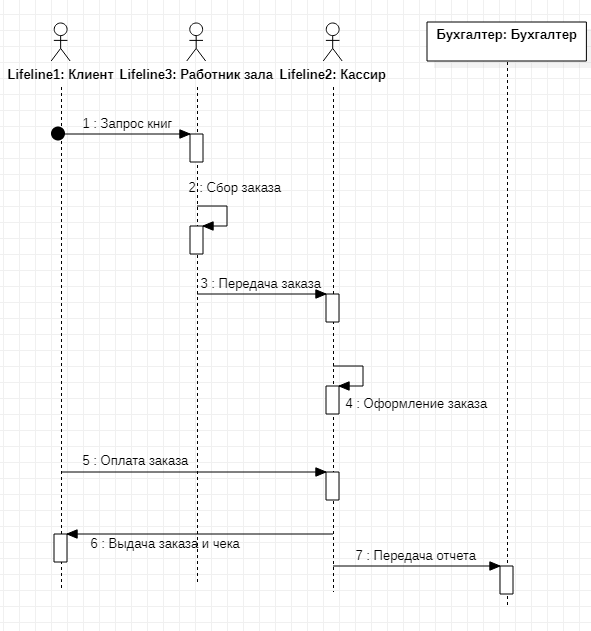
На диаграмме прецедентов, представленной на рисунке 9, описаны возможные взаимодействия между тремя действующими лицами: кассиром, клиентом и работником зала.

*Рисунок 9 — Диаграмма прецедентов*

**2.2 Диаграммы п****оследовательности**

Sequence diagram (диаграммы последовательностей действий). Данный тип диаграмм позволяет отразить последовательность передачи сообщений между объектами. Этот тип диаграммы не акцентирует внимание на конкретном взаимодействии, главный акцент уделяется последовательности приема/передачи сообщений. Окончательный вид диаграммы показан на рисунке 14. Диаграмма содержит действующие лица и объекты, так же добавлены такие объекты, как Object Message (Новое сообщение между объектами) и Message to Self (Рефлексивное сообщение самому себе).

На диаграмме последовательности, представленной на рисунке 10, описан процесс покупки в магазине.

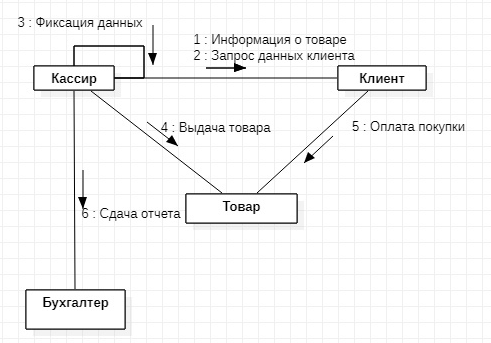
**

*Рисунок 10 — Диаграмма последовательности*

**2.3 Кооперативная диаграмма**

Collaboration diagram (диаграммы сотрудничества). Этот тип диаграмм позволяет описать взаимодействия объектов, абстрагируясь от последовательности передачи сообщений. На этом типе диаграмм в компактном виде отражаются все принимаемые и передаваемые сообщения конкретного объекта и типы этих сообщений. По причине того, что диаграммы Sequence и Collaboration являются разными взглядами на одни и те же процессы, Rational Rose позволяет создавать из Sequence диаграммы диаграмму Collaboration и наоборот, а также производит автоматическую синхронизацию этих диаграмм. Окончательный вид диаграммы показан на рисунке 15. Диаграмма содержит Link To Self (Связь объекта самого с собой), Object Link (Связь между объектами), объекты и Link Message (сообщение, передаваеме между двумя объектами или объектом самому себе).

На кооперативной диаграмме, представленной на рисунке 11, описан процесс покупки.



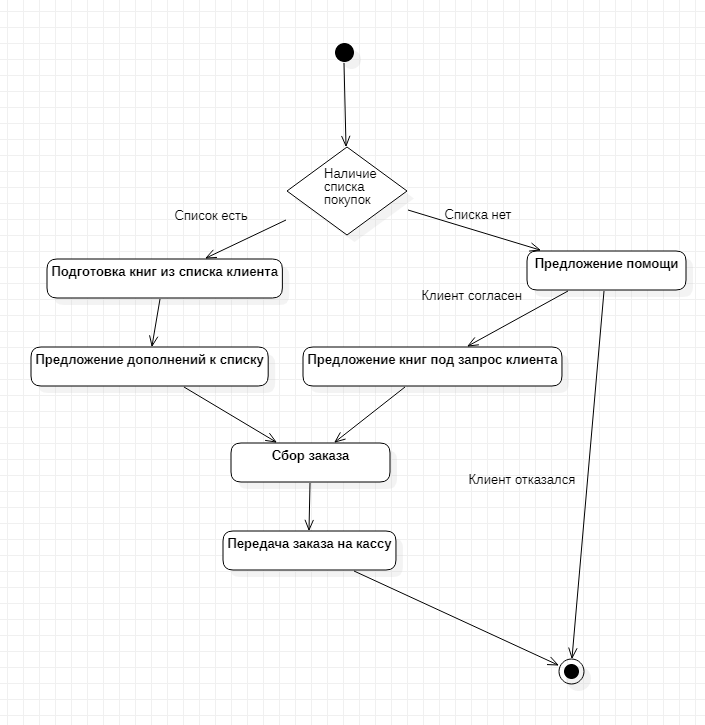
*Рисунок 11 — Кооперативная диаграмма*

**2.4 Диаграмма состояний**

Каждый объект системы, обладающий определенным поведением, может находится в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Поведение большинства объектов реальных систем можно представить с точки зрения теории конечных автоматов, то есть поведение объекта отражается в его состояниях, и данный тип диаграмм позволяет отразить это графически. Для этого используется два вида диаграмм: Statechart diagram (диаграмма состояний) и Activity diagram (диаграмма активности) .

Диаграмма состояний предназначена для отображения состояний объектов системы, имеющих сложную модель поведения. Конечный вид диаграммы представлен на рисунке 12. Диаграмма содержит в себе state (суперсостояние), состояния, state transition (переходы), начальное (Start state) и конечное состояние (End State).

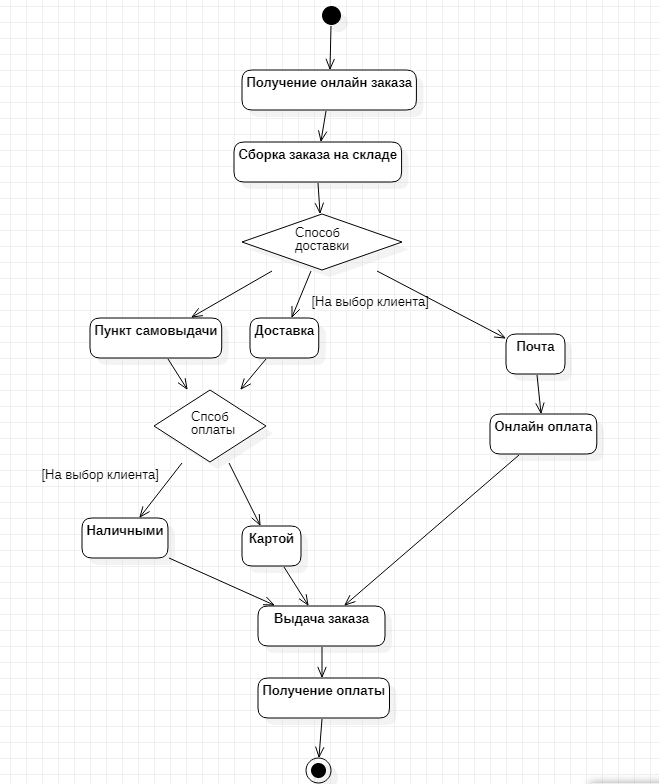
На диаграмме состояний, представленной на рисунке 12, описано поведение работника зала.



*Рисунок 12 — Диаграмма состояний*

**2.5 Диаграмма активности**

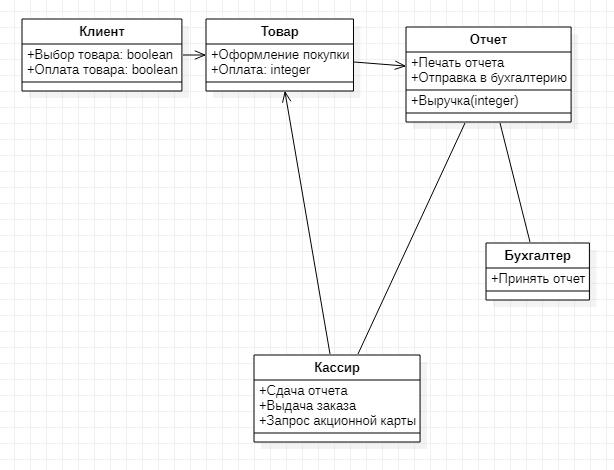
Activity diagram (диаграммы активности). Это дальнейшее развитие диаграммы состояний. Фактически данный тип диаграмм может использоваться и для отражения состояний моделируемого объекта, однако, основное назначение Activity diagram в том, чтобы отражать бизнес-процессы объекта. Этот тип диаграмм позволяет показать не только последовательность процессов, но и ветвление и даже синхронизацию процессов [7]. Окончательный вид диаграммы показан на рисунке 13. На диаграмме присутствуют дорожки (Swimlane), объекты состояния(Activity), начальное (Start state) и конечное состояние (End State).

**

*Рисунок 13 — Диаграмма активности*

**2.6 Диаграмма классов**

На данном этапе создается уже готовая диаграмма классов, а именно добавляются новые атрибуты и операции к классу. Происходит подробное описание операций и атрибутов, а именно указываются их типы.

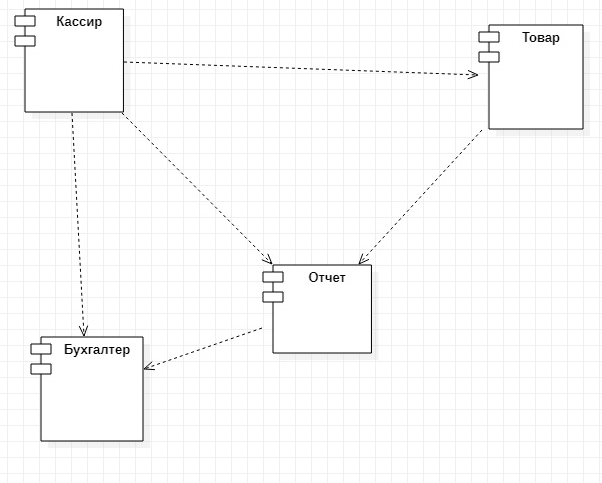


*Рисунок 14 — Диаграмма классов*

**2.7 Диаграмма компонентов**

Component diagram (диаграммы компонентов). Этот тип диаграмм предназначен для распределения классов и объектов по компонентам при физическом проектировании системы. Часто данный тип диаграмм называют диаграммами модулей [7]. Диаграмма компонентов представлена на рисунке 15.

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

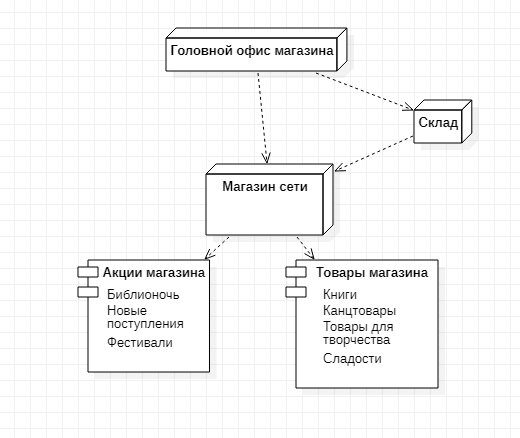


*Рисунок 15 — Диаграмма компонентов системы*

**2.8 Диаграмма развертывания**

Диаграмма развертывания (deployment diagram) - диаграмма, на которой представлены узлы выполнения программных компонентов реального времени, а также процессов и объектов.

Диаграмма развертывания применяется для представления общей конфигурации и топологии распределенной программной системы и содержит изображение размещения компонентов по отдельным узлам системы.

****

*Рисунок 16 — Диаграмма развертывания*

**Заключение**

В результате выполнения практической работы была достигнута поставленная цель, а именно практическое усвоение основных разделов проектирования информационных систем, развитие практических навыков решения задач исследования и проектирования систем на языке UML.

Была выполнена основная цель – разработка и внедрение информационной системы про книжный магазин.

Далее была построена соответствующая модель.

**Список литературы**

1. Боггс У. Боггс М. UML и Rational Rose [Текст]: учеб. Пособие/У.Боггс, М. Боггс:Лори, 2004, - 510с.
2. Кватрани, Т. Rational Rose 2000 и UML [Текст]: учеб. Пособие/ Т. Кватрани: ДМК Пресс, 2001, - 176с.
3. Ларман, К. Применение UML и шаблонов проектирования [Текст]: учеб. Пособие/ К. Ларман: Вильямс, 2002. – 624с.
4. Рамбо Дж., Блаха М.. UML 2.0. Объектно-ориентированное моделирование и разработка. [Текст]: учеб. Пособие/ Дж. Рамбо, М. Блаха: Питер, 2004. – 544с.
5. Трофимов, С. А. - CASE-технологии. Практическая работа в Rational Rose [Текст]/ С. А. Трофимов: Бином-Пресс, 2002. – 288с.
6. Статья С. А. Трофимов UML диаграммы в Rational Rose [Электронный ресурс]. - http://www.caseclub.ru/articles/rose2.html

Размещено н===