

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет инженерно-экономический
Кафедра экономической информатики
Дисциплина «Технологии разработки программного обеспечения»

«К ЗАЩИТЕ ДОПУСТИТЬ»
Руководитель курсового проекта
Ассистент кафедры ЭИ
_____ Г.П. Котковец
_____._____.2022

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему:
«РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ТОВАРОВ НА СКЛАДЕ»

БГУИР КР 1-40 05 01-02 006 ПЗ

Выполнил студент группы 172302
БОВКУН Максим Игоревич

(подпись студента)
Курсовая работа представлена на
проверку _____.2022

(подпись студента)

Минск 2022

РЕФЕРАТ

БГУИР КР 1-40 05 01-02 001 ПЗ

Бовкун М.И. Разработка программы учета товаров на складе / М.И. Бовкун. – Минск: БГУИР, 2022. –19 с.

Пояснительная записка 19 с., 13 рис., 0 табл., 6 источников, 1 приложения

СКЛАДСКОЙ УЧЕТ, ИНВЕНТАРИЗАЦИЯ ТОВАРНО-МАТЕРИАЛЬНЫХ ЦЕННОСТЕЙ, МОНИТОРИНГ СБЫТА, АНАЛИЗ СБЫТА, МОДЕЛИ *UML*, *IDEF0*, *BPMN*, СХЕМЫ АЛГОРИТМОВ, ПРОГРАММНОЕ СРЕДСТВО

Цель курсовой работы: повышение качества применяемых управленческих решений по оптимизации складского учета и ускорению оборачиваемости товарно-материальных ценностей на основе программной поддержки планирования, учета и анализа складских операций на предприятии.

Методология проведения работы: в процессе решения поставленных задач использованы принципы системного подхода, аналитические методы, методы компьютерной обработки экспериментальных данных и компьютерного моделирования.

Результаты работы: выполнена постановка задачи и определены основные методы ее решения; в ходе объектного моделирования системы построен ряд *UML*-диаграмм; разработаны модели бизнес-процессов предметной области на основе нотаций *IDEF0* и *BPMN*; описаны основные алгоритмы работы программного средства; разработано руководство пользователя; выполнено тестирование программного средства, показавшее его соответствие функциональным требованиям, поставленным в задании на разработку.

Программный продукт разработан на языке *C++* с применением *MS Visual Studio 2019*.

Область применения результатов: с помощью разработанного программного средства можно проанализировать складской учет на предприятии. Анализ оборачиваемости товарно-материальных ценностей производится за квартал, полугодие и произвольные периоды. В процессе анализа фактические данные сравниваются с плановыми, предшествующими периодами, рассчитывается процент выполнения плана.

Разработанное программное средство полностью отвечает всем функциональным требованиям, необходимым при планировании, учете и анализе складских операций на предприятии.

СОДЕРЖАНИЕ

Введение	4
1 Автоматизированная система учёта товаров на складе	6
2 Описание процесса автоматизированного системы учёта товаров на складе на основе стандартов IDEF0, IDEF3, DFD.....	8
3 Анализ требований к разрабатываемому программному средству. разработка BPMN-модели основного процесса предметной области	11
4 UML-Модели представления программного средства и их описание	14
4.1 Диаграмма вариантов использования	14
4.2 Диаграмма последовательности	14
4.3 Диаграмма состояния.....	16
4.4 Диаграмма компонентов.....	16
4.5 Диаграмма развёртывания.....	17
Список использованных источников	19

ВВЕДЕНИЕ

С конца XX века люди стали стремительно развивать информационные технологии. Если изначально эти технологии использовались в основном государством, военными и крупными корпорациями, то сейчас все сферы человека охвачены ими. Раньше предприятия спокойно обходились без современных технологиях, но в современное время избавиться от всего этого просто невозможно. Лишившись всех технологий, компания потеряет конкурентоспособность на рынке. Любое развитие требует увеличения нагрузки, повышения рисков и ответственности, а значит предприятию нужно двигаться вперёд, искать новые методы оптимизации и автоматизации различных работ.

Использование информационных систем и технологий упрощает труд человека, позволяя больше уделить время сущности проблемы. Благодаря компьютеру мы улучшаем наше взаимодействие с друг другом, делаем жизнь более комфортной. Современные технологии это уже не роскошь для предприятия, а просто необходимость для того, чтобы оставаться на плаву.

Технологии, конечно, дошли и до логистической сферы. Склады играют важную роль в логистике, это важная часть всех предприятий, для оптовой и розничной торговли они служат фундаментом. Поэтому необходимо максимально хорошо организовать работы на складе. Это позволит экономить время и людской ресурс, избегать ненужных ошибок, которые связаны с учётом различных товаров, а также быть востребованным на рынке. Практически каждое предприятие имеет в своём распоряжении склад, но многие из них не ввели и не планируют вводить автоматизацию работы на складе. Автоматизация производства – это применение комплекса средств, позволяющих осуществлять производственные процессы без непосредственного участия человека, но под его контролем.

Из основных преимуществ автоматизации складского учёта можно выделить следующие наиболее значимые аргументы:

- снижение затрат на бумажные документы, так их заменяют электронные копии;
- складской учёт напрямую связан с операциями по учёту продаж и покупки;
- моментальный доступ к информации о товаре;
- отсутствие необходимости в большом штате сотрудников.
- Целью данного проекта является оптимизация системы учёта товаров на складе.

Для реализации данной цели курсового проекта необходимо осуществить выполнение следующих задач:

- исследовать процесс учёта товаров на складе;
- разработать функциональную модель основного процесса;
- выполнить программную реализацию;
- разработать алгоритмы пользовательских функции приложения;
- разработать простой и удобный интерфейс приложения;
- описать программу;
- предусмотреть обработку исключительных ситуаций.

1 АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЁТА ТОВАРОВ НА СКЛАДЕ

Автоматизированная система учётов товаров на складе – это мгновенный поиск продукции, его удобное размещение на складе, уменьшение людского ресурса, высокая скорость перемещение товаров по складу и его удобное размещение. Каждый товар имеет собственный электронный документ, в котором отражена вся необходимая информация о нём. Также в нём отражается вся история перемещения продукта. Каждый продукт имеет свой уникальный код, который привязан только к нему. Собственно, по этому коду и следует осуществлять поиск товара в базе данных через программный продукт.

Прежде, чем автоматизировать складской учёт, необходимо разобраться зачем вообще он нужен. Тут можно выделить следующие задачи:

- оперативное снабжение клиентов ресурсами и товарами;
- оценка наличия товара, остатков, информация о движении товара;
- обеспечение сохранности товара и его комплектующих;
- переоценка товаров в зависимости от набора факторов, влияющих на стоимость.

Важным аспектом работы склада является перманентный контроль. Периодически проводится полная инвентаризация склада. В процессе инвентаризации происходит полная проверка фактического наличия товаров с учётным количеством. Если будут найдены несоответствия, то это станет сигналом руководству о хищении, либо о халатности сотрудников во время учёта товаров. Наиболее чётко функции складского учёта вы можете увидеть на рисунке 1.1.

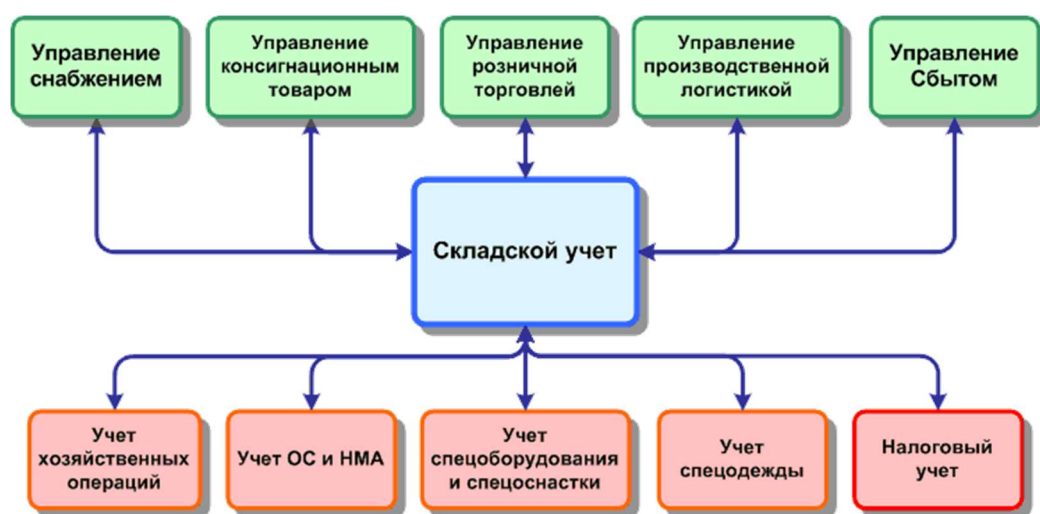


Рисунок 1.1 – Функции складского учёта

Теперь, когда мы определились с функциями складского учёта, то становится ясно, автоматизация – это ключ к решению многих проблем в складском учёте.

Автоматизированная система учётов товаров на складе работает за счёт следующих компонентов:

- программное обеспечение;
- оборудование для заполнения данных о продукте (подразумевается, что имеется аппарат для считывания штрих-кода, но в данной курсовой его роль будет играть ручной ввод данных);
- работник склада, который и ведёт учёт.

Теперь необходимо разобраться с этапами автоматизации склада, их всего три. Больше этапов и не нужно, так как эти сполна удовлетворяют процесс автоматизации.

Первый этап – это *подготовительный этап*. На данном этапе происходит разработка планов проекта, сбор информации от заказчика, описываются все бизнес процессы, составляется техническое задание, назначаются ответственные работники, выбор технического оборудования.

Второй этап – это *внедрение*. В это время происходит установка и настройка программы, проведение функциональных тестов, для выявления ошибок и их устранении либо для убеждения в исправности программного продукта, затем происходит обучения сотрудников и настройка информационной системы под организацию, и наконец само внедрение программы.

Третий (заключительный) этап – это *поддержка*. Этот этап включает в себя поддержку проекта, техническое сопровождение автоматизированного склада и последующие доработки.

Таким образом можно сделать вывод, что автоматизация складского учёта на предприятии помогает решить довольно много простых и сложных задач, выводит предприятие на новый уровень, делает её более конкурентоспособной.

2 ОПИСАНИЕ ПРОЦЕССА АВТОМАТИЗИРОВАННОГО СИСТЕМЫ УЧЁТА ТОВАРОВ НА СКЛАДЕ НА ОСНОВЕ СТАНДАРТОВ IDEF0, IDEF3, DFD

Для того, чтобы организовать автоматизированную систему учёта товаров на складе, нужно разобраться из чего состоит этот процесс, разбить его на подпроцессы, понять, чем руководствуются сотрудники склада, а также найти все нюансы при выполнении данной работы. Чтобы всё это выполнить, необходимо смоделировать бизнес-процесс. Для этого хорошо подходит нотация IDEF0, IDEF3, DFD. Последние 2 нужно использовать чтобы показать последовательность выполнения действий и поток данных, приходящих на систему, соответственно.

Функциональная модель автоматизированного учёта товаров на складе выполнена с помощью программы AllFusion Process Modeler.

На рисунке 2.1 изображена контекстная диаграмма верхнего уровня. На ней показаны входные данные, механизмы (то есть кто или с помощью кого выполняют процесс), требования и выходные данные. В диаграмму входят:

- а) Входные данные: учётная запись пользователя и информация о товарах;
- б) Механизмы: информационная система и персонал;
- в) Требования: база данных пользователей и нормативно-правовые документы;
- г) Выходные данные: база данных товаров и выданный/списанный товар.

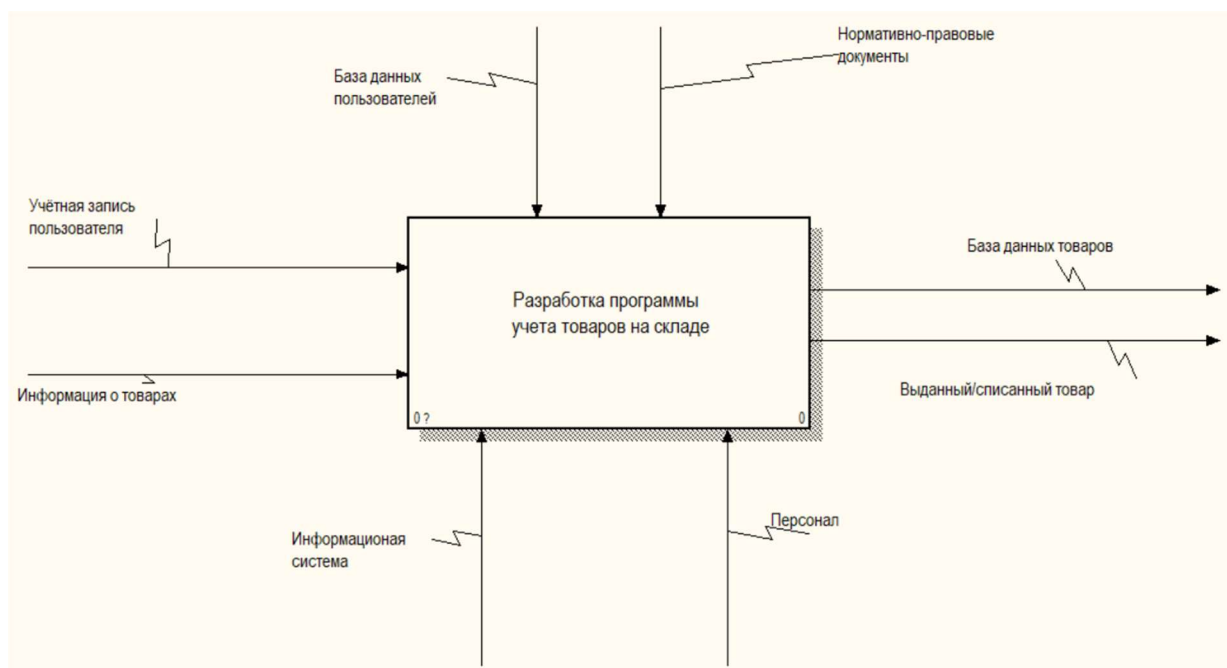


Рисунок 2.1 – Контекстная диаграмма верхнего уровня

На рисунке 2.2 представлена декомпозиция диаграммы верхнего уровня. Согласно данной схеме процесс состоит из 4 этапов: выполнить авторизацию пользователя, заполнение информации о товаре, работа с данными, удаление информации о товаре.

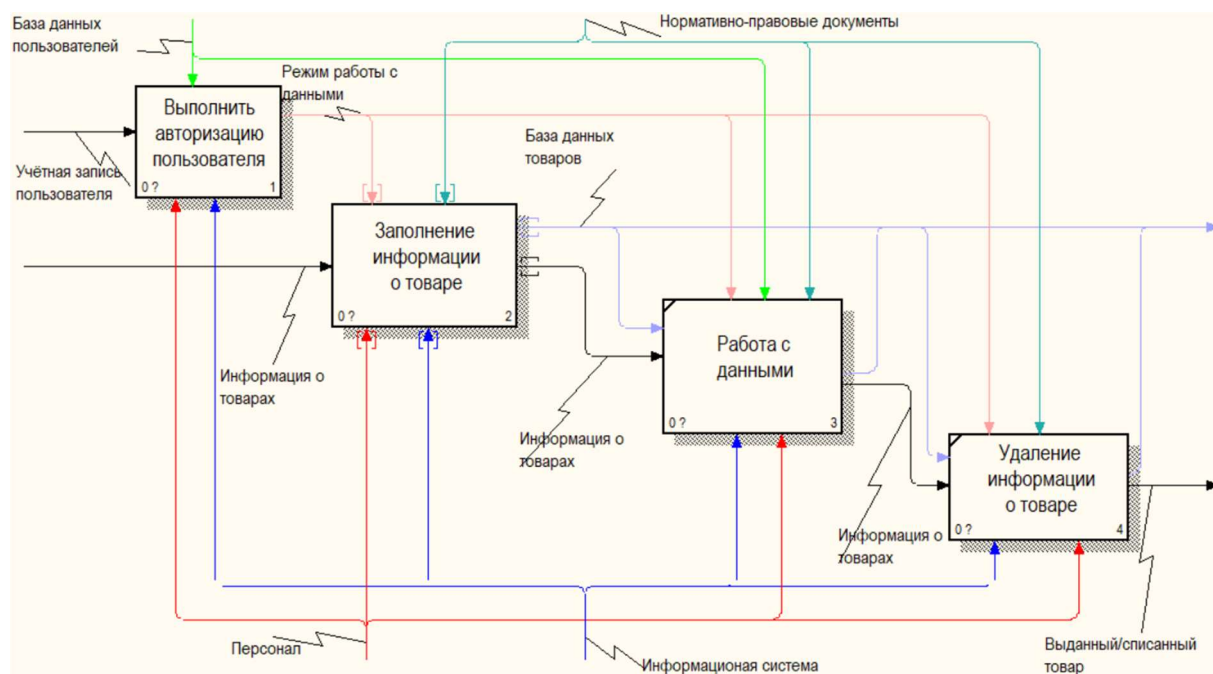


Рисунок 2.2 – Декомпозиция диаграммы верхнего уровня

Процесс выполнения авторизации пользователя можно декомпозировать. Для этого я выбрал уже нотацию IDEF3, для того чтобы более детальнее отразить процесс входа пользователя в аккаунт. Результат выполнения вы можете увидеть на рисунке 2.2.1.

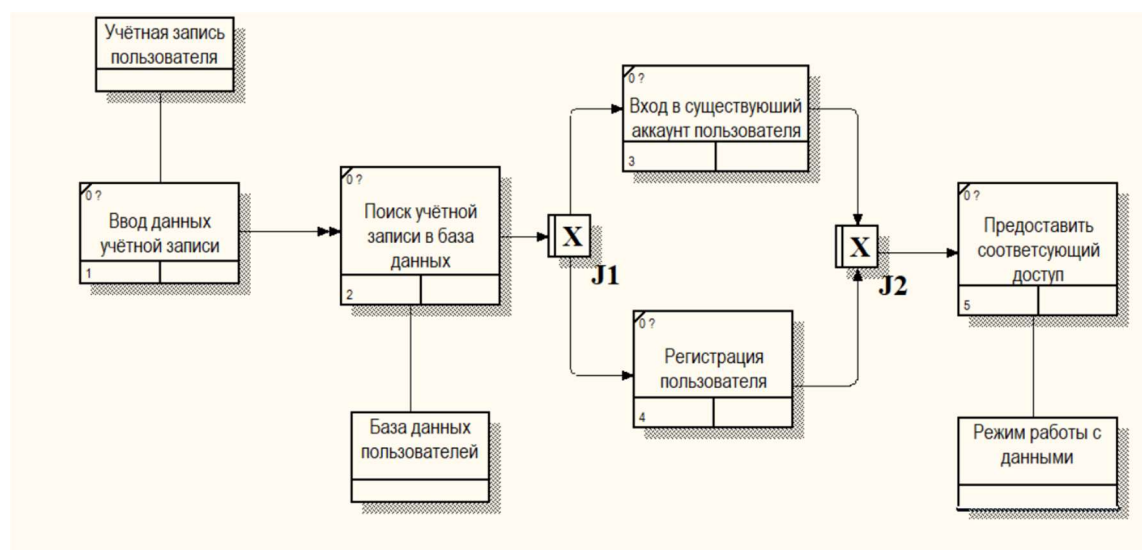


Рисунок 2.2.1 – Декомпозиция диаграммы «Выполнить авторизацию пользователя»

На данной декомпозиции видно 3 объекта ссылок: учётная запись пользователя, база данных пользователя и режим работы с данными. Они отражают какие-либо данные, к которым обращаются «работы».

Возвращаясь на декомпозицию вверх, мы можем понять, что процессу к «Заполнения информации о товаре» возникают вопросы о более детальной реализации, поэтому необходимо выполнить его декомпозицию. Но, на мой взгляд, для этого лучше подходит DFD, чем IDEF0. Благодаря этой замене мы можем более детально описать поток работы с данными при выполнении данного процесса. Результат вы можете увидеть на рисунке 2.3.

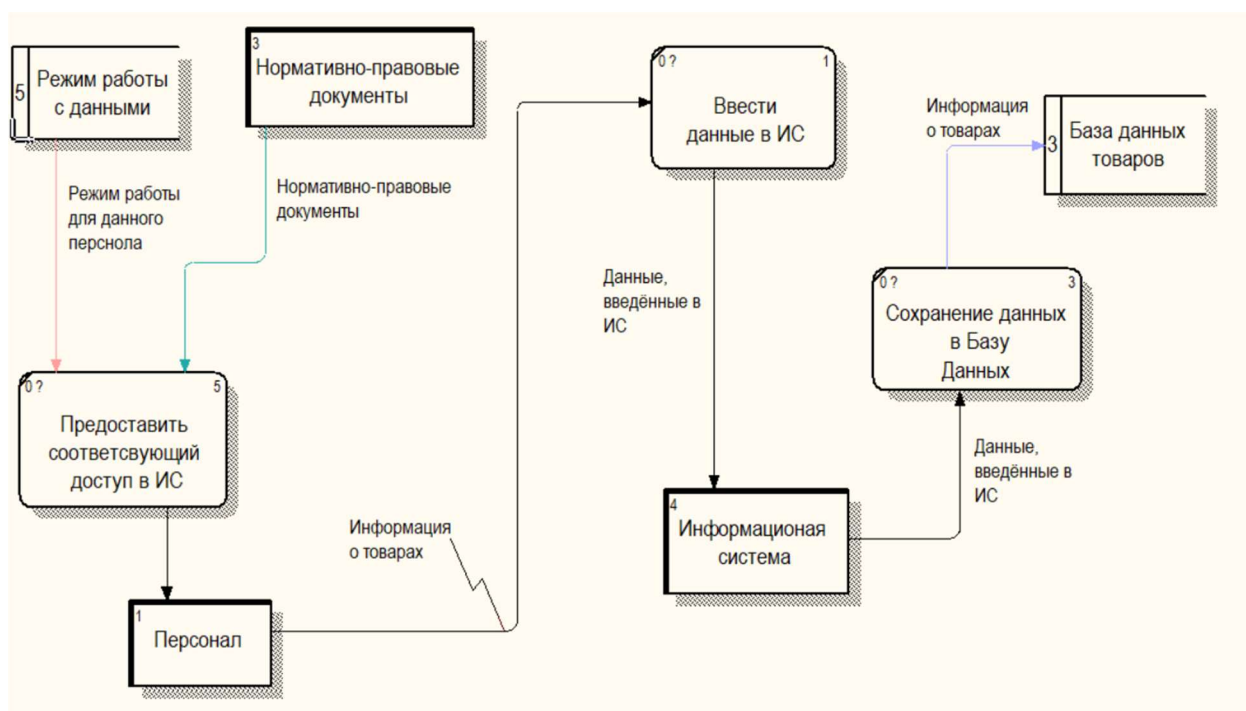


Рисунок 2.3 – Декомпозиция диаграммы «Заполнение информации о товаре»

Данная декомпозиция помимо процессов содержит 2 внешние сущности: персонал и информационная система; это нужно, чтобы показатель участников данных процессов. А также здесь присутствуют 2 области хранения данных: режим работы с данными и база данных товаров; так мы показываем какие-то внешние источники данных, с которыми взаимодействует процесс.

Таким образом в данной главе был рассмотрен процесс автоматизированного учёта товаров на складе путём моделирования бизнес-процессов на основе нотаций IDEF0, IDEF3 и DFD.

3 АНАЛИЗ ТРЕБОВАНИЙ К РАЗРАБАТЫВАЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ. РАЗРАБОТКА VRMN-МОДЕЛИ ОСНОВНОГО ПРОЦЕССА ПРЕДМЕТНОЙ ОБЛАСТИ

Любой процесс начинается с проектирования программного продукта, необходимо определить требования к программному продукту и его исходные данные. Поэтому в этой главе будет произведён анализ требований к программному продукту, а также будет реализована разработка основного процесса.

Прежде тем перейти к моделированию основного процесса, необходимо понять, что любая информационная система начинается с авторизации пользователя. Это очень важный процесс будущего и действующего проекта. Ведь чрезвычайно важно разграничивать возможности пользователей, а также закрыть доступ лицам, которые не зарегистрирован в информационном средстве. Поэтому нужно описать как будет происходить авторизация в нашем программном продукте.

На рисунке 3.1 вы можете увидеть моделирование процесса авторизации.

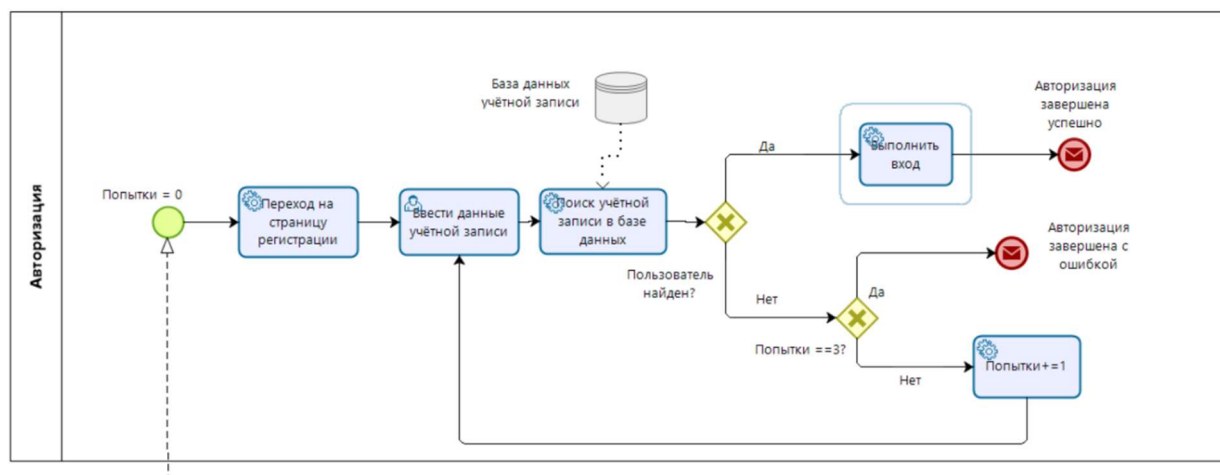


Рисунок 3.1 – Авторизация

На рисунке 3.1 вы можете увидеть переменную «попытки». Это нужно для того, чтобы не позволить пользователю делать более 3 попыток авторизации в процессе выполнения программы. При большем количестве попыток программа посчитает действия подозрительными и закончится выполнение программы. В процессе авторизации будут задействоваться некий файл, который будет играть роль базы данных, в котором будут храниться различные учётные записи. Авторизации в данном случае имеет один начало, но несколько сценариев окончания с разными сообщениями: «авторизация

завершена успешно» и «авторизация завершена с ошибкой». Следует дополнить, что вход происходит согласно выбранному режиму работы.

Также в будущем проекте будут присутствовать возможность регистрации, которая будет подтверждена администратором, но на диаграмме это не отражено, так как это сильно затруднило бы её чтение.

Теперь необходимо определиться с требованиями по авторизации, которые будут в будущем программном продукте:

- а) наличие двух режимов доступа: администратор и пользователь;
- б) наличие у администратора нескольких режимов работы с данными: работа с данными, с аккаунтами и с редактирование данных;
- в) возможность ввод новых данных;
- г) сохранение и чтение данных с файла;
- д) сортировка данных по какому-либо полю;
- е) выполнение индивидуального задания: вывести в алфавитном порядке список товаров, хранящихся более х месяцев, стоимость которых превышает у рублей (х, у вводятся с клавиатуры).
- ж) Изменение данных в процессе выполнения программы и перезапись в файл соответственно.

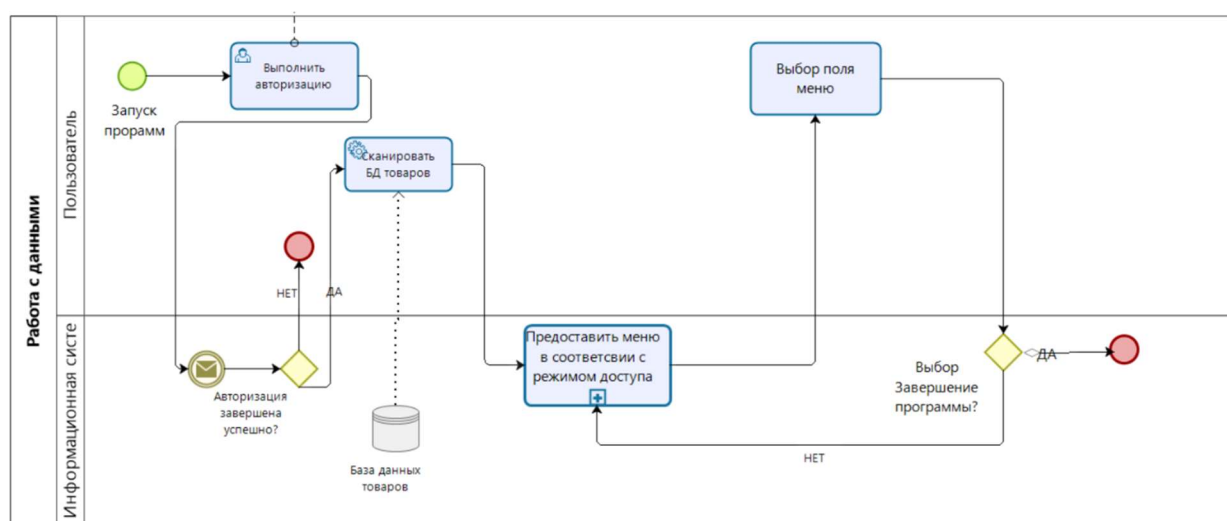


Рисунок 3.2 – Работа с данными

На рисунке 3.2 вы можете увидеть, что процесс работы с данными начинается с выполнения авторизации. Её начинает процесс «Выполнить авторизацию», затем информационное система ожидает получить сообщение с успешной или неуспешной автоматизацией, от которого и зависит дальнейшее выполнение программы. Вы можете обратить внимание, что на

диаграмме имеется предпроцесс, поэтому на рисунке 3.3.1 вы можете увидеть его реализацию.

В пункте меню должен быть выбор окончания работы. Если выбран не он, то программа выполняет выбранный пункт меню и заново показывает меню, до тех пор, пока не будет выбрано окончания программы. Также будет присутствовать пункт выход из аккаунта, на схеме он не изображен, так как это будет перегружать модель различными стрелками.

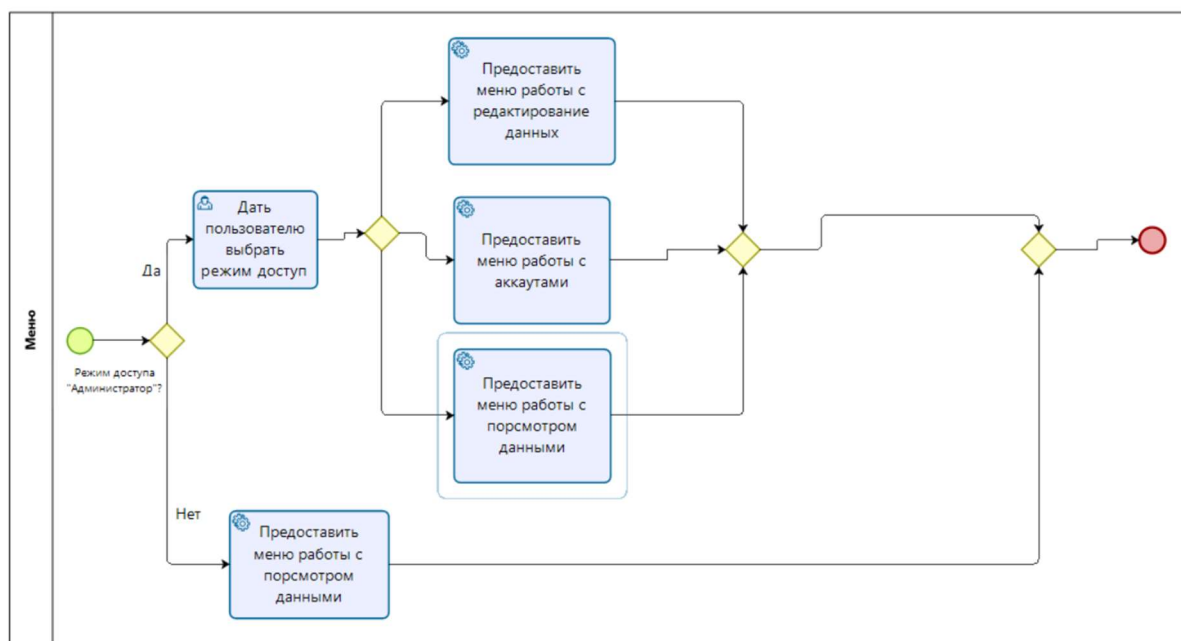


Рисунок 3.2.1 – Предоставить меню в соответствии с режимом доступа

Таким образом в этой главе был смоделирован основной процесс автоматизации складского учёта с использование нотации BPMN, а также описаны некоторые нюансы при будущей разработки.

4 UML-МОДЕЛИ ПРЕДСТАВЛЕНИЯ ПРОГРАММНОГО СРЕДСТВА И ИХ ОПИСАНИЕ

4.1 Диаграмма вариантов использования

Диаграмма вариантов использования в UML – это диаграмма, которая отражает отношения между актёрами и прецедентами. Это позволяет описать систему на концептуальном уровне. Суть данной диаграммы состоит в том, чтобы представить в виде множества актёров, которые взаимодействуют с системой с помощью так называемых вариантов использования.

Рассмотрим на рисунке 4.1.1 диаграмму вариантов использования для реализуемой программы.

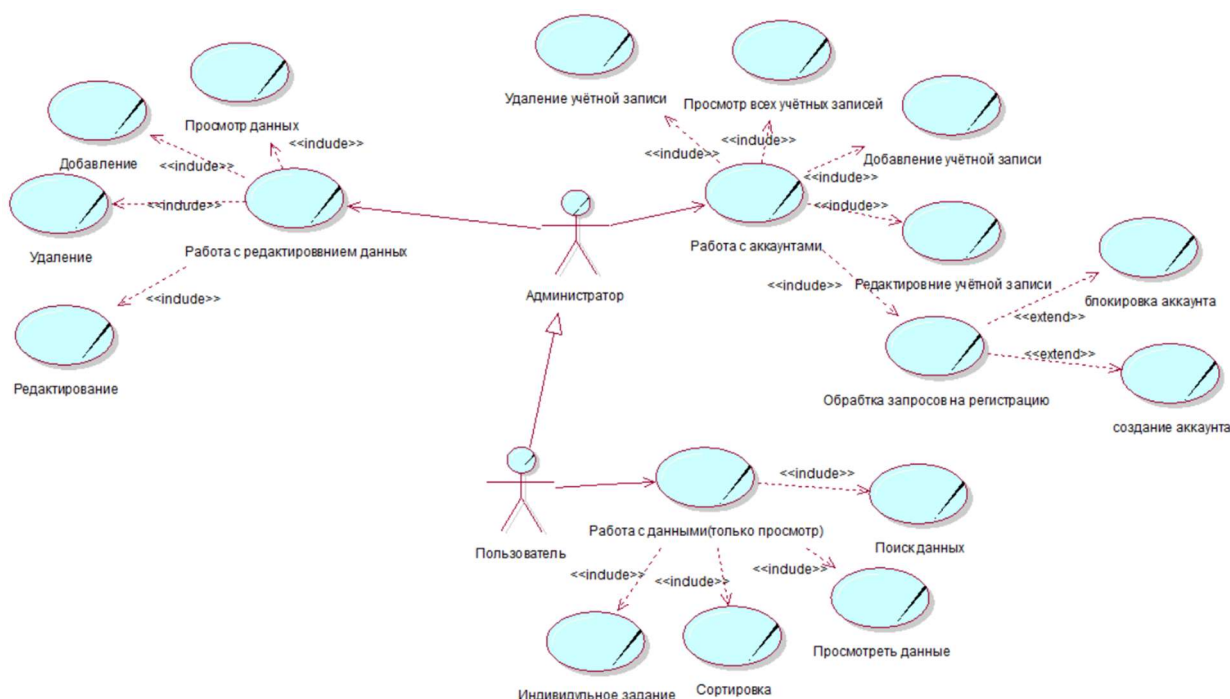


Рисунок 4.1.1 – Диаграмма вариантов использования

Актёрами на данной являются пользователь и администратор. Так как администратор обладает всеми возможностями пользователя, то было бы логично наследовать возможности пользователя и администратора. Поэтому для этого используем стрелку обобщения.

4.2 Диаграмма последовательности

Диаграммы последовательностей используются для уточнения диаграмм прецедентов, более детального описания логики сценариев

использования. Это отличное средство документирования проекта с точки зрения сценариев использования. Диаграммы последовательностей обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями. Впрочем, часто возвращаемые результаты обозначают лишь в том случае, если это не очевидно из контекста. Поэтому данная диаграмма идеально подходит, чтобы изобразить последовательность действий в реализуемом программном продукте.

На рисунке 4.2.1 изображена диаграмма последовательности для автоматизации складского учёта.

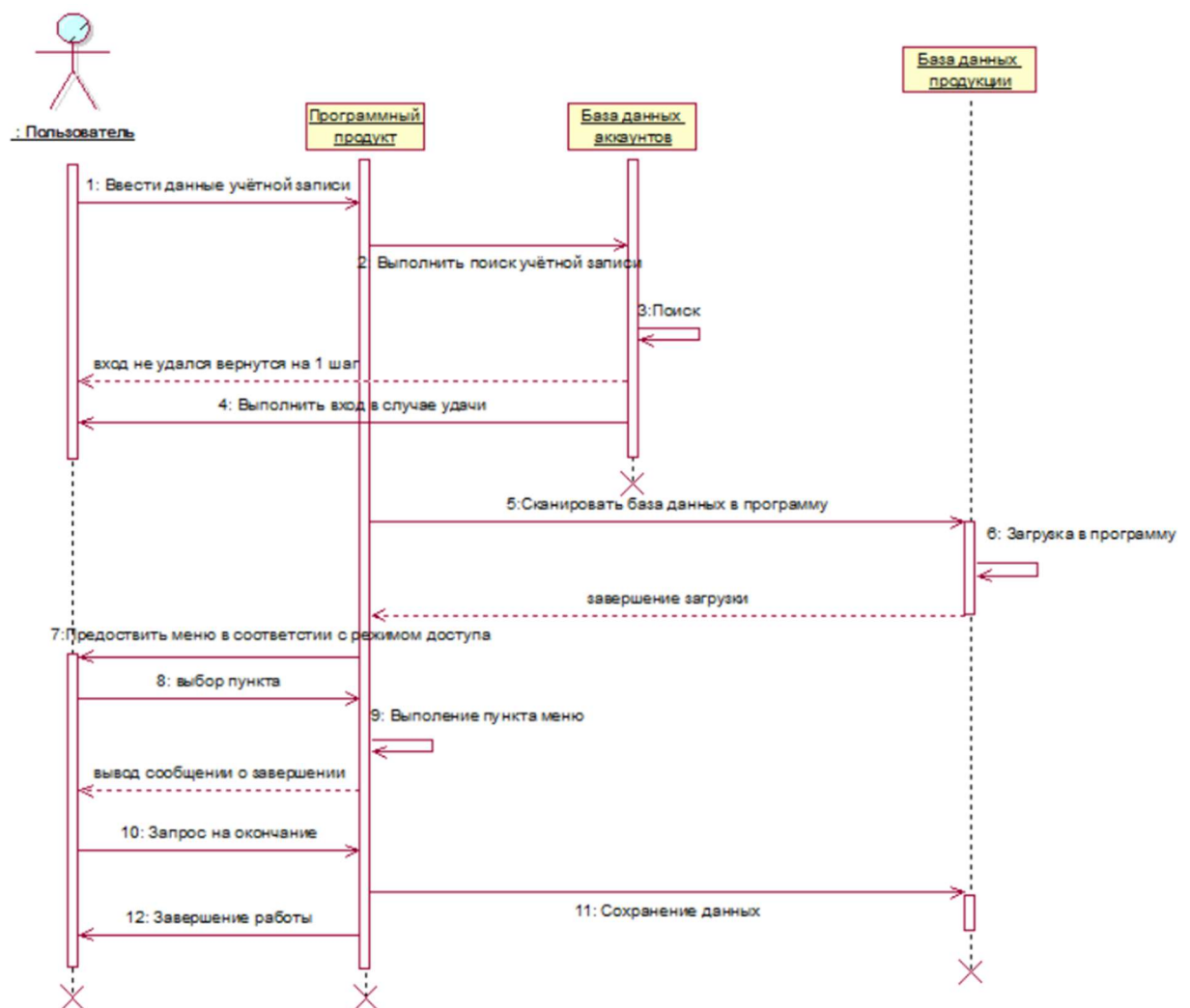


Рисунок 4.2.1 – Диаграмма последовательности

Из рисунка 4.2.1 видно, что мы имеем 4 объекта: пользователь, программный продукт, база данных аккаунтов и база данных продукции, которые между собой взаимодействуют в разные моменты времени выполнения программы.

4.3 Диаграмма состояния

Для того, чтобы показать условия процесс работы системы используют диаграмма состояния. Это позволят детально смоделировать жизненный цикл объекта. Диаграмма состояний покажет нам все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния и динамическое поведение сущности, показывая, как эта сущность в зависимости от своего текущего состояния реагирует на различные события. Основными элементами диаграммы состояний являются «Состояние» и «Переход».

Созданную диаграмму состояний конкретно данного будущего программного продукта изображена на рисунке 4.3.1.

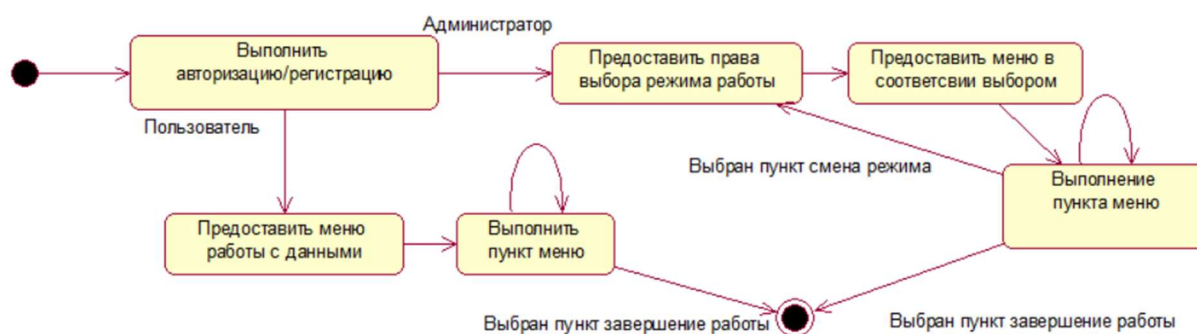


Рисунок 4.3.1 – Диаграмма состояния

Процесс «Выполнить авторизацию/регистрацию» имеет 2 ветвления. От этого зависит дальнейшее выполнение программы, так как администратор должен иметь больше прав. Ведь пользователь должен иметь права только на просмотр данных и небольшую работу с ними, а администратор имеет права на заполнение информации о товаре, а также на редактирования его свойств.

4.4 Диаграмма компонентов

Диаграмма компонентов называется диаграмма UML, на которой показаны компоненты системы и зависимости между ними.

С её помощью отвечающий за компиляцию и размещение системы персонал выясняет, какие библиотеки кода существуют и какие исполняемые файлы будут созданы при его компиляции. Разработчики узнают, какие библиотеки кода существуют и какова связь между ними и каковы связи

между ними. Зависимость между компонентами отражают порядок их компиляции.

На рисунке 4.4.1 изображена диаграмма компонентов, на которой показана связь между библиотеками и определение где находится их реализация.

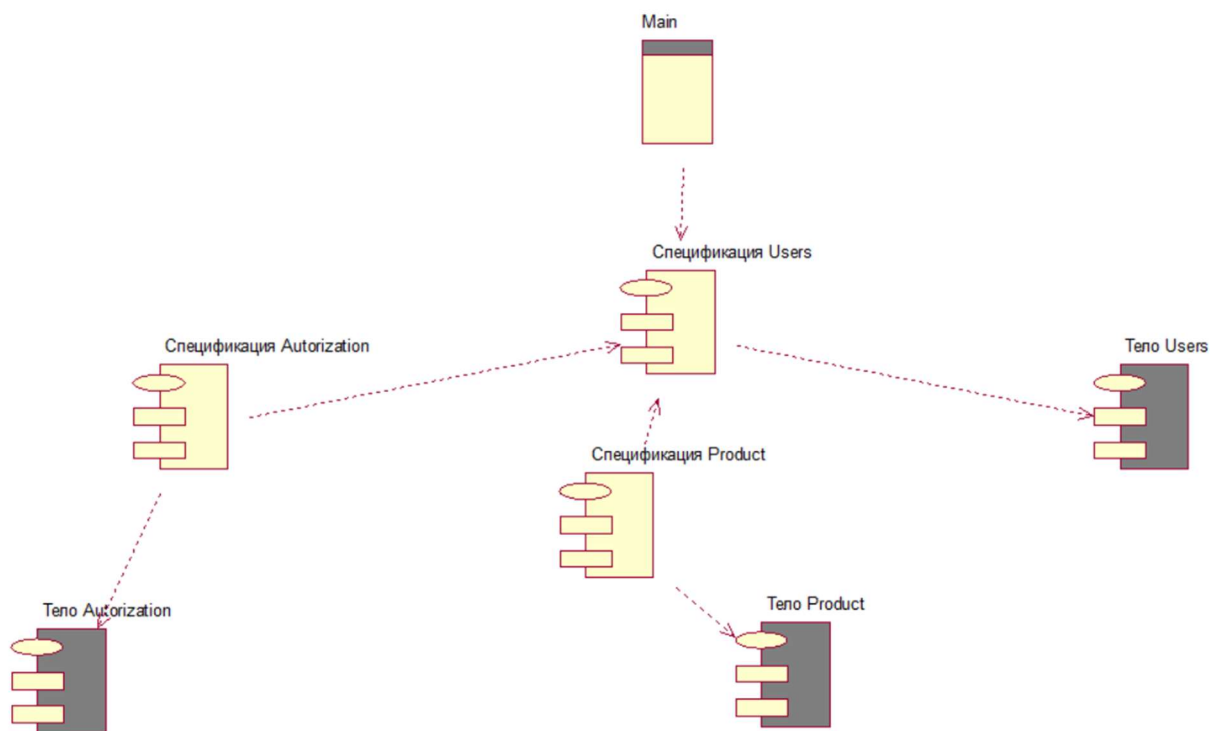


Рисунок 4.4.1 – Диаграмма компонентов

Итак, можно увидеть, что в будущем программном продукте будет несколько библиотек: Authorization, Product, Users, а также их реализация в других исходных файлах. Authorization и Product будут подключены в библиотеку Users и поэтому в главный исходный файл Main будет подключаться только библиотека Users.

4.5 Диаграмма развёртывания

Для того, чтобы отобразить все узлы сети, связи между ними и процессы, выполняющиеся на каждом узле, а также промежуточное программное обеспечение, соединяющее их и используют диаграмму развёртывания.

На рисунке 4.5.1 изображена диаграмма развёртывания для автоматизированного учёта товаров на складе.

Основными компонентами данной диаграммы является «процесс» и «устройство». «Процесс» называют любую машину, имеющую вычислительную мощность, а «устройство» – это аппаратура, не имеющая вычислительной способности. Из рисунка 4.5.1 видно, что на данной диаграмме есть 3 блока «процесс»: база данных продукции, база данных учётных записей и программа по учёту товаров на складе. Соответственно терминал ввода/вывода относится к блоку «устройство».

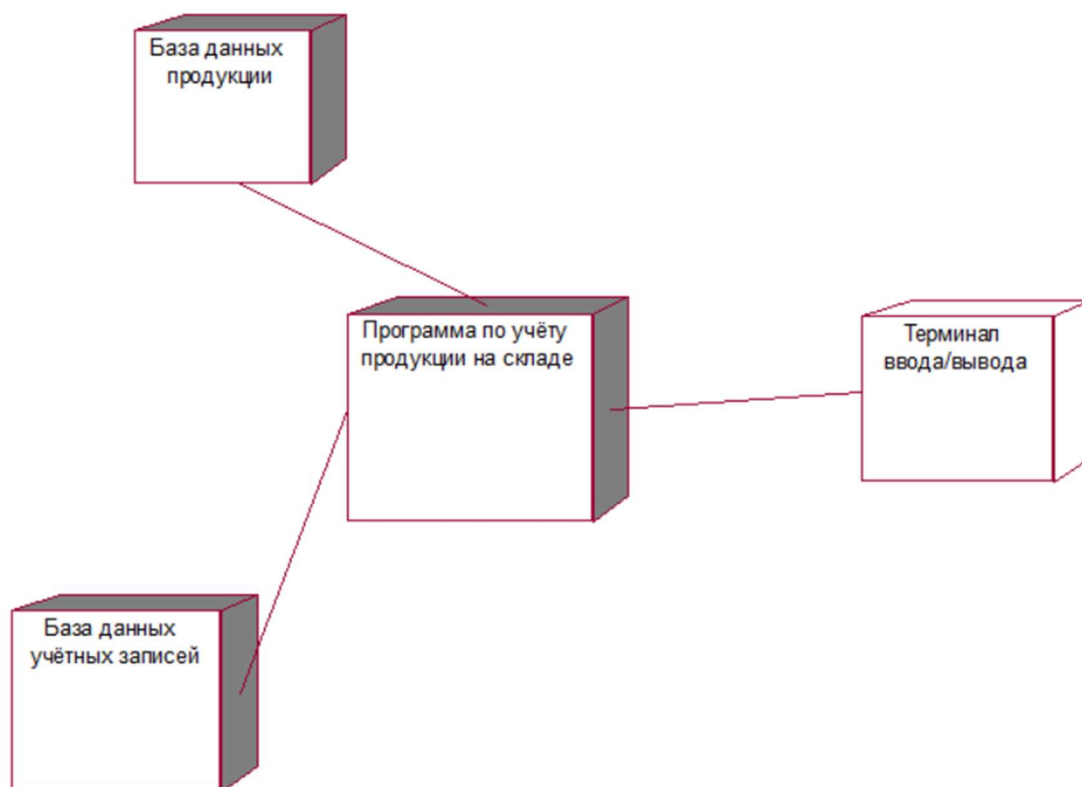


Рисунок 4.5.1 – Диаграмма развёртывания

Таким образом в данной главе был произведён анализ будущего программного продукта и были построены и описаны различные диаграммы UML.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Бритов, Г.А. Моделирование бизнес-процессов / Г.А. Бритов, Т.В. Осипова. – М.: LAP, 2014 г. – 124 с.
- [2] Гамма Э. Приёмы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. — СПб.: Питер, 2015 г. — 368 с.
- [3] Грекул В.И. Проектирование информационных систем / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. – М.: Интернет-университет информационных технологий, 2018 г. - 321 с.
- [4] Дубейковский В.И. Эффективное моделирование с СА ErwinProcessModeler (BPwin; AllfusionProcessModeler). 2-е изд., исправленное и дополненное / В.И. Дубейковский. – М.: Диалог-МИФИ, 2015. – 384 с.
- [5] Гагарина, Л.Г., Кокорева, Е.В., Виснадул, Б.Д. Технология разработки программного обеспечения: учебное пособие / под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2013. — 400 с.: ил. — (Высшее образование).
- [6] Золотухина, Е.Б., Красникова, С.А., Вишня, А.С. Управление жизненным циклом информационных систем (продвинутый курс): Краткий конспект лекций: — М.: КУРС: ИНФРА-М, 2017. — 117 с.