Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет информационных технологий и управления

Кафедра вычислительных методов и программирования

Пояснительная записка

к курсовой работе по курсу

«Объектно-ориентированное программирование»

на тему

**«ПРОГРАММА ПО ОПРЕДЕЛЕНИЮ ВЕЛИЧИНЫ ТАМОЖЕННЫХ СБОРОВ»**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | Калеников И. А.  группа 024402 |  |
| Руководитель |  | Бич Н.А. |  |

Минск 2022

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc102574834)

[1 Обзор автоматизированной работы рассматриваемой сферы деятельности, программных аналогов, методов и алгоритмов решения поставленной задачи 6](#_Toc102574835)

[1.1 Обзор автоматизированной работы предприятия 6](#_Toc102574836)

[1.2 Программные аналоги 7](#_Toc102574837)

[1.3 Обзор методов и алгоритмов решения поставленной задачи 9](#_Toc102574838)

[2 Функциональное моделирование на основе стандарта IDEF0 12](#_Toc102574839)

[3 Структура данных 16](#_Toc102574840)

[4 Описание созданных программных конструкций 19](#_Toc102574841)

[5 Разработка и описание диаграммы классов 21](#_Toc102574842)

[6 Разработка и описание диаграммы вариантов использования приложения 22](#_Toc102574843)

[7 Блок-схема алгоритма работы всей программы и алгоритма работы нескольких основных методов 24](#_Toc102574844)

[8 Описание алгоритма запуска приложения, его использования, результаты работы программы, тестироания обработки ошибок 25](#_Toc102574845)

[Заключение 26](#_Toc102574846)

[Список использованных источников 27](#_Toc102574847)

[Приложение А (обязательное) Листинг программы 28](#_Toc102574848)

[Приложение Б (обязательное) Диаграмма классов 29](#_Toc102574849)

[Приложение В (обязательное) Диаграмма вариантов использования 31](#_Toc102574850)

[Приложение Г (обязательное) Схемы алгоритмов программы 32](#_Toc102574851)

# ВВEДЕНИЕ

В настоящее время проблема автоматизации стала актуальной практически для каждой отрасли. Стремительное [развитие компьютерной техники](https://studopedia.ru/3_55511_istoriya-razvitiya-kompyuternoy-tehniki.html) и роста объема данных привело к тому, что все большее распространение стали получать [информационные системы](https://studopedia.ru/18_61568_informatsionnie-tehnologii-it.html). Автоматизированные информационные системы представляют собой совокупность различных средств, предназначенных для сбора, подготовки, хранения, обработки, поиска и предоставления информации.

Главной целью автоматизации является повышение производительности и эффективности труда работников, улучшение качества информационной продукции и услуг, повышение сервиса и оперативности обслуживания пользователей.  С её помощью сокращается время выполнения определенных действий, преобразуются и изменяются технологические процессы, предоставляются новые виды информационных услуги продуктов.  Особенно это важно в сфере услуг. От качества предоставляемых услуг напрямую зависит рейтинг и имидж предприятия среди клиентов, а также авторитет среди конкурирующих организаций на рынке.

Одной из самых значимых сфер оказания услуг является сфера предоставления перевозочных услуг. Ежедневно в мире перевозятся сотни тысяч контейнеров с различными товарами. Поэтому очень важно наличия автоматизированной информационной системы, которая контролирует и управляет потоками перемещения товаров. При работе с товарами важно точно и быстро предоставить информацию об интересующих направлениях, о существующих перевозках, времени и месте выезда и прибытия, количества свободных мест и ценах перевозок. Кроме того, должна быть предоставлена возможность зарегистрировать перевозку заранее заранее, а также отменить регистрацию в случае необходимости.

Грамотное обеспечение перевозок поддерживает высокий уровень развития сферы импорта-экспорта, что является залогом комфорта и выгоды клиентов, ключевым моментом, определяющим динамику экономического развития страны.

Цель курсовой работы – разработать систему для автоматического определения таможенных сборов, которая позволит своевременно вносить и получать всю нужную информацию, что повысит качество и производительности работы таможенных структур.

Для достижения цели требуется реализовать следующие задачи:

* исследовать процесс формирования и управления операциями ввоза–вывоза, процесса регистрации сборов;
* описать процесс оформления операции с помощью IDEF0;
* организовать структуры необходимых данных для реализации приложения;

– описать созданные программные конструкции;

– реализовать разделение ролей: администратор, пользователь, а также хранение пароля в зашифрованном виде;

* разработать диаграммы классов;
* составить диаграмму вариантов использования, которая будет отражать возможности администратора и пользователей;
* разработать схемы алгоритмов работы приложения и основных ее методов;
* описать алгоритм запуска приложения, его использования, результаты работы программы;
* провести тестирование разработанной программы.

Объектом исследования является процесс введения онлайн-регистрации на рейс, а также процессы формирования и контроля грузоперевозок.

Ключевые слова: ВВОЗ, ВЫВОЗ, АЛГОРИТМ, ПОДПРОГРАММА, ФУНКЦИЯ, ГРУЗОПЕРЕВОЗКИ, КЛАСС, ФАЙЛ, ПОЛЬЗОВАТЕЛЬ, АДМИНИСТРАТОР, ПРИЛОЖЕНИЕ.

# ОБЗОР АВТОМАТИЗИРОВАННОЙ РАБОТЫ РАССМАТРИВАЕМОЙ СФЕРЫ ДЕЯТЕЛЬНОСТИ, ПРОГРАММНЫХ АНАЛОГОВ, МЕТОДОВ И АЛГОРИТМОВ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

## Обзор автоматизированной работы предприятия

Цель автоматизации – это способствование рациональному планированию расписания и суточного плана товарооборота, сокращения времени на обслуживание, эффективному контролю за работой персонала, своевременное предоставление максимального объема информации.

Информационная система связывает поставщика товаров и заказчика. Для оптимального взаимодействия этих сторон необходимо автоматизировать информационные процессы, это приведет к улучшению их качества и повышению быстродействия. Самое главное при перевозках правильно правильно составить их график и возможность акторам получать необходимую информацию в любой момент времени. Для каждого рейса необходимо выделить транспорт, составить список товаров и внести рейс в график. Вместе с этим необходимо обеспечить выполнение ряда других задач, таких регистрация перевозки и расчет стоимости товаров.

Управлением автоматизированной информационной системой занимаются администраторы, которые получают доступ посредством ввода пароля, установленным вышестоящими работниками. Цель администраторов заключается в добавлении, обработке и удалении необходимой информации. Они отвечают за полноту и актуальность хранимых данных.

В данной предметной области администратор должен выполнять следующие функции:

– добавление, изменение и удаление товаров из базы данных;

– формирование операций на основе доступных товаров;

– просмотр списка клиентов;

– отмена операции.

Вход в программу защищен паролем. Каждый желающий может зарегистрироваться в системе введя необходимую информацию. Пользователь информационной системы авторизуется посредством ввода логина и пароля. Авторизированные пользователи имею широкий набор возможностей, таких как:

– просмотр всех существующих товаров;

– возможность просмотра товаров в нужном порядке, а также организация операции;

– отмена операции;

– просмотр информации о операциях;

Профиль авторизированного пользователя должен содержать:

– логин;

– пароль.

Автоматизированная система по своей сути является центральной оперативной базой данных предприятия. Она позволяет точно фиксировать все события, которые планируются и происходят в процессе товарооборота. Данное решение предоставит специалистам всю необходимую информацию как о планируемых перевозках, так и о потоке товаров. Система может использоваться в качестве источника оперативной информации для многих других систем, включая отображение информации о перевозках, управление отправками и регистрации клиентов.

Регулярно проводится инструктаж ответственных специалистов о работе с автоматизированной системой.

## Программные аналоги

Альта – ГДТ. Рабочий инструмент декларанта. Предназначен для максимально быстрого и удобного заполнения, печати и выгрузки в электронный вид всех документов, необходимых при таможенном оформлении. Содержит весь необходимый функционал для осуществления электронного декларирования. При работе с программой экономится время на формирование заявок от клиента и администратора, актов выполненных работ, отчетов таможенным заявкам. Программа дает статистику по работе таможни.

Основные функции программы:

* Составление документов, которые могут понадобиться для электронного декларирования (ЭД), предварительного информирования (ПИ), транзита, информационного обмена между СВХ и таможней, оформления реестров экспресс грузов и других возможных процедур системы ЭД и не только;
* Наличие необходимых справочников и классификаторов;
* Составление нормативных документов;
* Отчетность для таможенных органов и внутренней статистики;
* Встроенная проверка правильности заполнения и возможность подключения внешней проверки.

ВЭД – платежи. Программа позволяет производить расчеты таможенных платежей посредством нескольких простых операций: достаточно ввести в соответствующие графы наименование и код товара (ТН ВЭД ЕАЭС), страну происхождения, актуальную таможенную стоимость и курс валюты. При изменении любого из параметров платежи пересчитываются.

Результаты расчетов могут быть представлены в виде справки и распечатаны.

Основные функции программы:

* Расчет таможенных платежей с учетом всех особенностей начисления таможенных пошлин и налогов;
* выбор таможенной процедуры, курса валюты и пр.;
* авто обновление с помощью дополнительной программы «СТМ-Офис»; идущей в составе основного продукта (создается ярлык, который скачивает обновление с сайта при его наличии, обновляет и запускает программу);
* самостоятельное обновление с сайта;
* подписка на файлы обновлений, автоматически рассылаемые на e-mail;
* обновление посредством программного средства «СТМ-Сервис»;

## Обзор методов и алгоритмов решения поставленной задачи

Поставленная задача заключается в реализации автоматизированной системы учёта ввоза и вывоза. В программе существуют несколько типов доступа. В зависимости от предоставляемого доступа в программе предусмотрен определенный набор функций, которые отличаются для клиента и администратора. Клиент не должен иметь прав, таких, какие имеет администратор, и наоборот. Клиент и администратор должны иметь интуитивно понятный интерфейс для удобного использования программы.

В разрабатываемой информационной системе для администратора необходимо реализовать следующий набор методов:

* вход в систему путем ввода логина и пароля;
* добавление операции посредством ввода всей необходимой информации;
* редактирование информации об операциях;
* просмотр списка клиентов;
* удаление операции.

Для входа в систему необходимо ввести логин и пароль. При некорректном вводе логина и пароля выдается ошибка о неправильном вводе данных. Программа запросит повторный ввод данных.

Метод управление базой данных товаров позволяет: добавлять, редактировать и удалять товары из учета.

Метод формирования операции нужен для вноса администратором полной информации об операции.

Редактирование операции позволяет внести возникшие изменения в операции.

Метод просмотра клиентов выводит список данных о клиентах.

Метод удаления операции производит отмену операции и её исключения из списка существующих.

В разрабатываемой информационной системе для пользователя необходимо реализовать следующий набор методов:

– авторизация или вход в систему;

– просмотр личной информации;

– просмотр списка существующих товаров;

– регистрация операции;

– просмотр информации об операциях;

– отмена операции;

– удаление профиля.

Регистрация осуществляется вводом логина и пароля. При вводе логина, который уже числиться в базе данных, выдается ошибка с предложением повторный ввод данных либо возращение к предыдущему меню. После ввода данных, пользователь заносится в базу данных пользователей информационной системы.

Для входа в систему необходимо ввести логин и пароль пользователя. При некорректном вводе логина или пароля выдается ошибка о неправильном вводе данных. Программа предложит повторный ввод данных либо возращение к предыдущему меню.

При регистрации, пользователю необходимо личные данные: имя, фамилию, дату рождения, гражданство и паспортные данные. При попытке повторной регистрации человека, чьи паспортных данные уже зарегистрированы, выдается ошибка и запрашивается повторный ввод.

Просмотр личной информации выводит на экран все данные о клиенте, введенные пользователем. Поиск осуществляется по номеру паспорта.

Редактирование личной информации позволяет пользователю изменить любые из данных, либо отредактировать весь набор данных.

Метод просмотра существующих товаров дает возможность ознакомится со всеми товарами. Реализованы возможности сортировки, фильтрации и поиска рейсов по вводимым параметрам.

Регистрация операции позволяет произвести необходимую операцию.

Метод просмотра информации об операциях предоставляет возможность просмотра информации о всех операциях, которые зарегистрировал клиент.

Отмена регистрации операции дает возможность пользователю отменить регистрацию операции.

Метод удаления профиля исключает пользователя из базы данных информационной системы. Пользователя с таким логином и паролем больше не существуют, для входа нужна повторная регистрация.

# ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ НА ОСНОВЕ СТАНДАРТА IDEF0

Одними из самых известных и широко используемых методологий в области моделирования бизнес-процессов являются методологии семейства IDEF.

Один из стандартов, который она включает IDEF0. IDEF0 – методология функционального моделирования. Используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции.

Результатом применения IDEF0 к некоторой системе является модель этой системы, состоящая из иерархически упорядоченного набора диаграмм, текста документации и словарей, связанных друг с другом с помощью перекрестных ссылок. IDEF0 требует, чтобы в диаграмме было не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования.

Выделяют четыре типа диаграмм данной модели:

* контекстная;
* диаграмма декомпозиции;
* диаграмма дерева узлов;
* диаграмма только для экспозиции.

Контекстная диаграмма – ее принято считать главной диаграммой, поскольку нацелена на изображение основной функции и ее взаимодействие с внешней средой.

Диаграммами декомпозиции – считаются второстепенными или дочерними. Описывают составные части основной функции.

Диаграмма дерева узлов – выражает зависимость функций между собой.

Диаграммы для экспозиции – разработаны для изображения отдельных частей системы, создана для выражения оптимальной точки зрения на бизнес-процесс.

Построим функциональную модель на основе стандарта IDEF0 для разрабатываемого приложения. Создадим контекстную диаграмму (рисунок 2. 1), которая опишет создаваемую систему формирования сбора и ее взаимодействие с внешним миром.

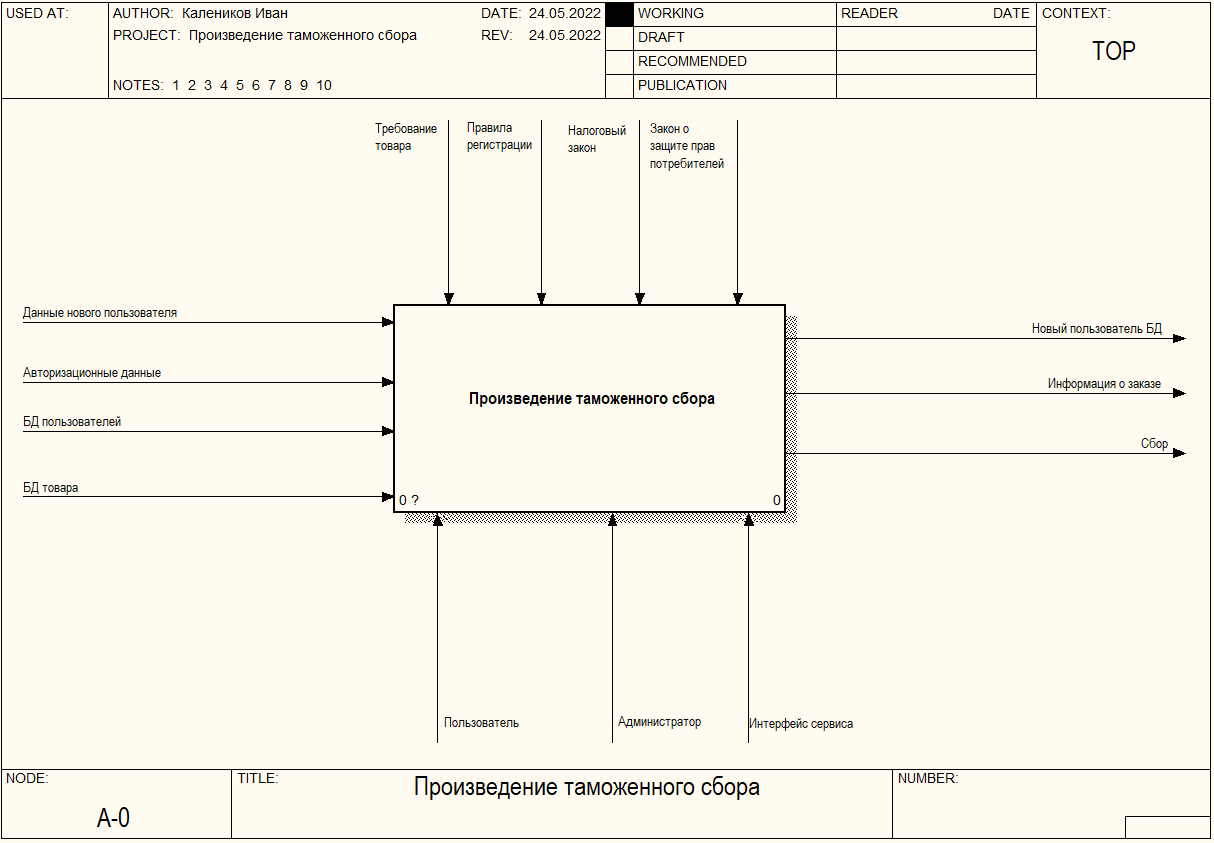


Рисунок 2.1 – Контекстная диаграмма верхнего уровня

Далее последовательно декомпозируем все блоки полученной диаграммы. Контекстная диаграмма детализируется на под­системы, каждая из которых подробно описывается отдельно (диаграммы декомпозиции). В результате декомпозиции получена структура, каждый элемент которой изображается на отдельной диаграмме декомпозиции.

Декомпозиция для диаграммы верхнего уровня представлена на рисунке 2.2. Разбиения диаграммы верхнего уровня изображены на блоки «регистрация пользователей» и «составление сбора» рисунках 2.3 и 2.4 соответственно.

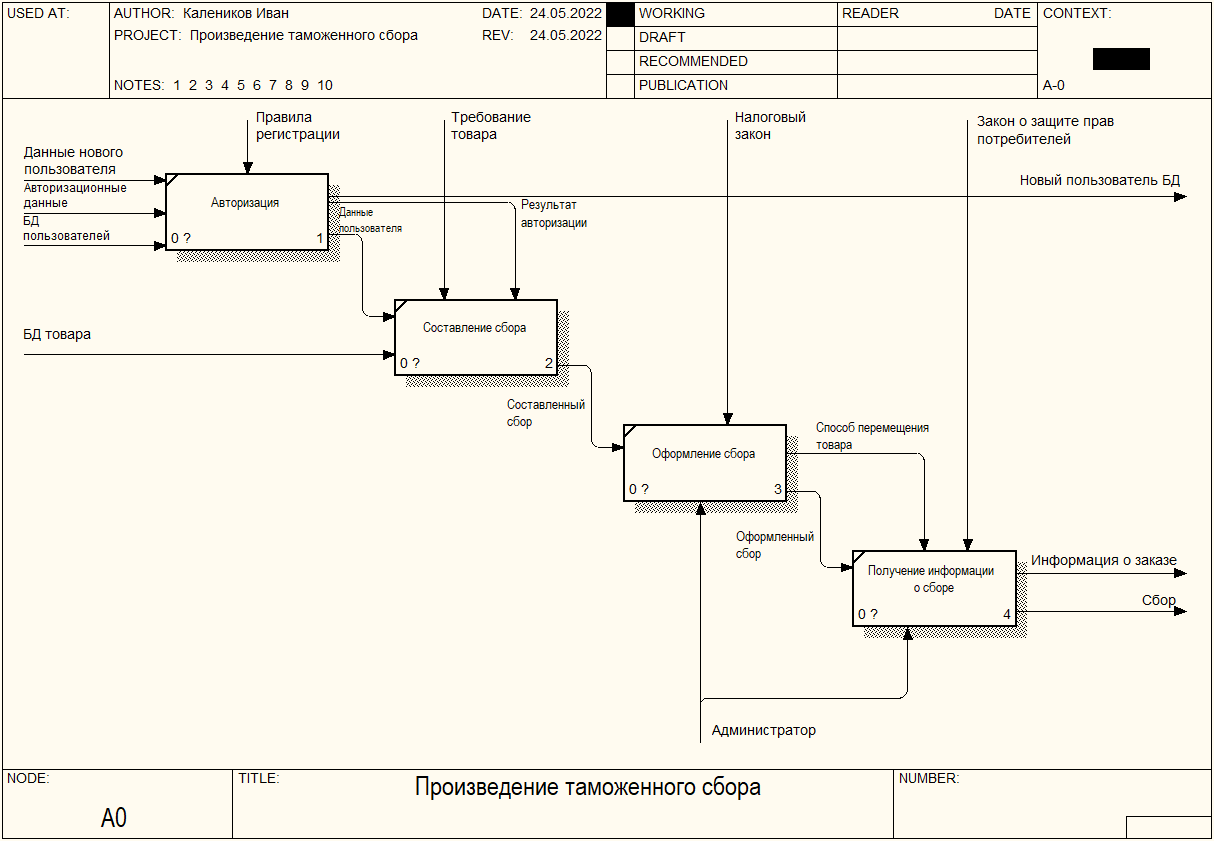


Рисунок 2.2 – Декомпозиция диаграммы верхнего уровня

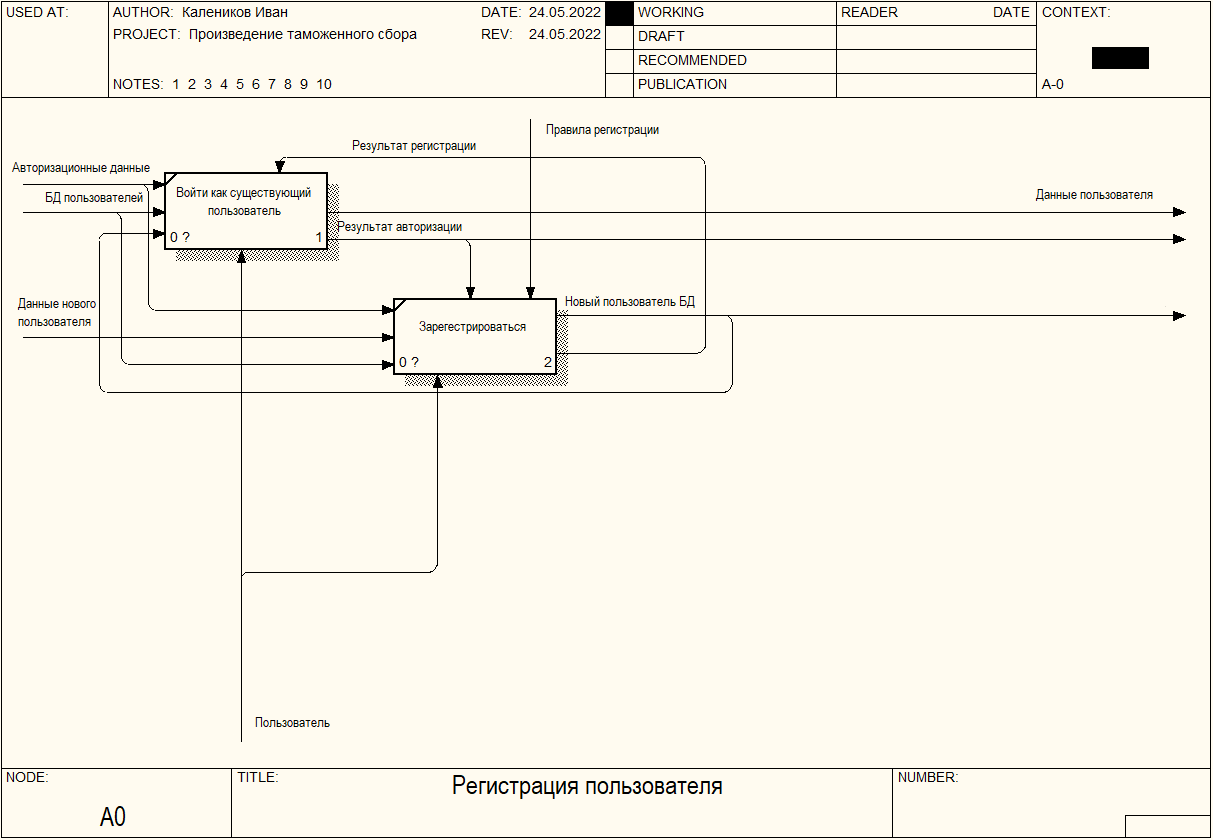


Рисунок 2.3 – Декомпозиция блока «Регистрация пользователей»

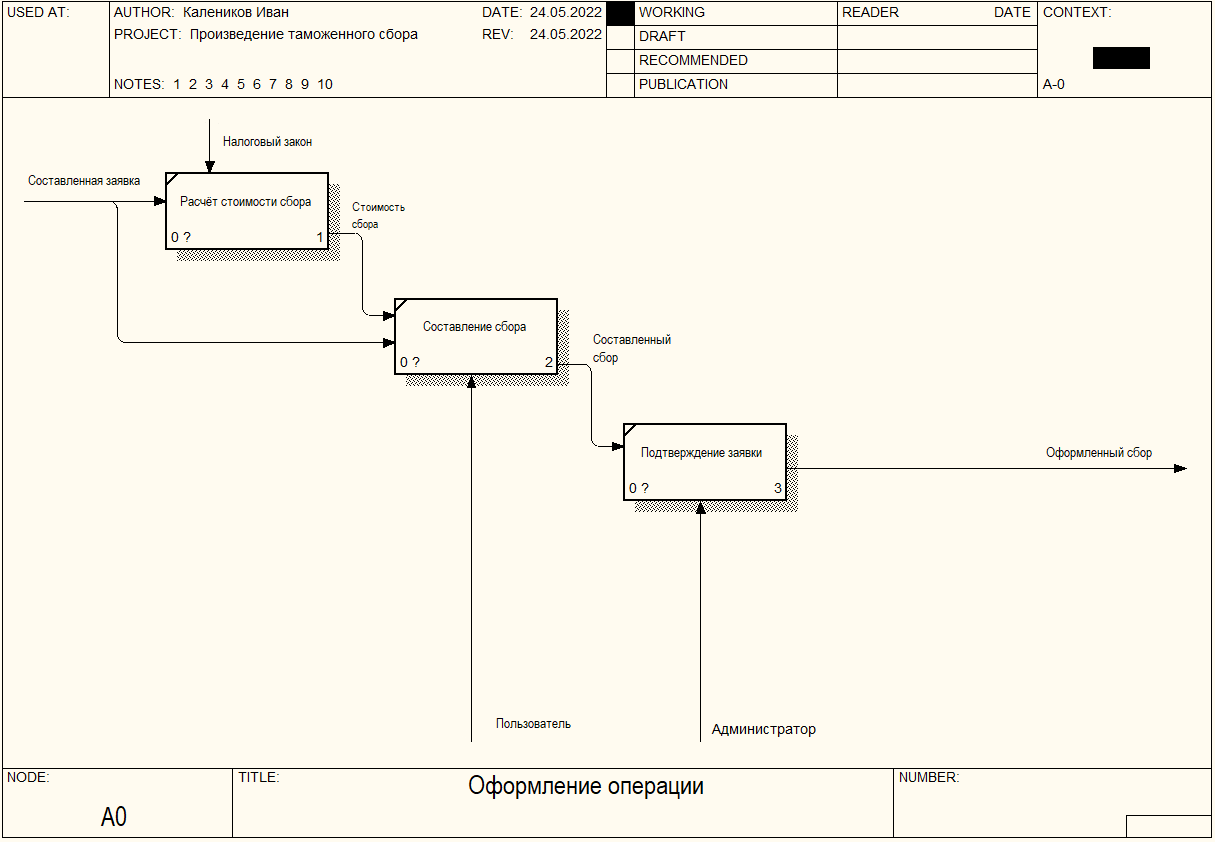


Рисунок 2.4 – Декомпозиция блока «Составление сбора»

# СТРУКТУРА ДАННЫХ

Вся необходимая информация хранится в текстовых файлах, сгруппированных в папки. На рисунке 3.1 изображена общая структура данных программы.

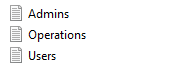


Рисунок 3.1 – Общая структура данных

В файле Operations.txt находиться информация о всех заявках, оставленных пользователями, что можно увидеть на рисунке 3.2.

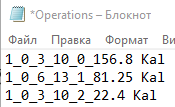


Рисунок 3.2 – Файл с информацией об операциях

Информация о товарах включает в себя:

* Направление товара относительно таможни;
* Статут рассмотрения
* Идентификационный номер товара;
* Количество;
* Идентификационный номер заявки;
* Итоговая цена;
* Логин пользователя, оставившего заявку.

В файле Admins.txt хранятся логин и пароль в зашифрованном виде, что можно увидеть на рисунке 3.3.

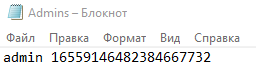


Рисунок 3.3 – Учётные данные администратора

В файле Users.txt так же хранятся логин и пароль в зашифрованном виде, что показано на рисунке 3.4

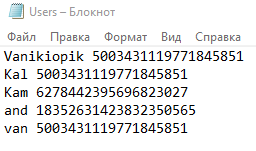


Рисунок 3.4 – Учётные данные клиентов

Исходная и итоговая информация храниться в текстовых файлах. Для записи и считывания данных из файлов использовался заголовочный файл fstream  из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют возможность для чтения/записи [данных](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)) из/в [файл](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB).

Для шифрования используется класс стандартной библиотеки C++ std::hash. С помощью оператора круглых скобок объект этого класса принимает на вход строку и возвращает хэш-значение.

Для удаления данных из файла, отмены операций и сортировки были использованы встроенные контейнеры языка С++: list, map, multimap.

# ОПИСАНИЕ СОЗДАННЫХ ПРОГРАММНЫХ КОНСТРУКЦИЙ

В разрабатываемой информационной системе были созданы программные конструкции, которые представляют собой пользовательские функции, выполняющие ряд определенных задач и объединенные логические группы.

Функция проверки целочисленного ввода данных в нужном диапазоне использует обработку ошибок программы средствами языка С++ (try, throw, catch). Алгоритм программы запрашивает ввод данных, пока не будет получено целочисленное значение в нужном диапазоне, передаваемом в качестве параметра функции. При некорректном вводе срабатывает исключение. В случае корректного ввода данных блоки обработки исключения пропускаются, а полученной значение возвращается в точку вызова функции. Данная конструкция предотвращает возникновения ошибок в программе и оптимизирует процессы проверки ввода данных.

Функция проверки правильности ввода логина и пароля вызывается при авторизации как администратора, так и пользователя. В качестве параметров функции передаются введённые логин и пароль. Данная программная конструкция использует класс ifstream заголовочного файла f**stream.h**. С файла, в котором в зашифрованном виде хранятся логин и пароль, построчно считываются данные. Полученная информация дешифруется с помощью пользовательской функции и сравнивается с введёнными значениями. При совпадении данных программа возвращает значение true. Если достигнут конец файла, а совпадения не найдены, функция возвращает false.

Функция деления строки позволяет разделить строку с пробелами, подаваемую в качестве параметра, на слова. Это программная конструкция полезна в случаи считывания набора данных из файла в виде строки. В функции создается буферная строка типа char. Алгоритм данной функции посимвольно проходит по полученной строке, добавляя на каждом шаге в созданную буферную строку текущий символ, пока не достигнут пробел. При равенстве символа пробелу или достижении конца исходной строки, буферная строка возвращает значение.

В пользовательском пространстве имен работы с файлами определены следующие функции: создание файла, запись в файл, дозапись в конец файла, очистка файла. Все функции реализованы с помощи классов и методов заголовочного файла f**stream.h.** Функция создания файла получает в качестве параметра путь к создаваемому файлу. Внутри функции **создается новый файл, готовый для дальнейшей работы с ним. Шаблонная функция записи открывает файл, путь которого передается в качестве параметра, в режиме записи и записывает данные любого формата, подаваемая в качестве параметра. Шаблонная функция дозаписи в конец файла выполняет схожие с предыдущей функцией действия, отличия заключаются в режиме открыто файла. Функция очистки файла получает в качестве параметра путь к файлу, который необходимо очистить. После вызова данной функции файл становится пустым, вся информация стерта.**

# РАЗРАБОТКА И ОПИСАНИЕ ДИАГРАММЫ КЛАССОВ

Целью курсового проекта является разработка объектной модели и ее программная реализация на языке C++. При разработке объектной модели требуется определить ее составные части – классы и установить связи между ними. Для наглядного представления была построена диаграмма классов, которая изображена на рисунке Б.1 приложения Б.

Диаграмма классов построена с учетом того, что объявлено стандартное пространство имен std.

Абстрактный класс User содержит учётные данные пользователя вместе с функциями формирования зашифрованного пароля и авторизации. Также он содержит защищённые константы с названиями файлов и папок.

Класс Admin наследуется от класса User. Он содержит методы для работы администраторского меню.

Класс Client наследуется от класса User. Он содержит персональную информацию о клиенте, а также методы работы клиентского меню.

Все классы содержат конструкторы по умолчанию и перегруженные конструкторы с параметрами, а также деструкторы.

# РАЗРАБОТКА И ОПИСАНИЕ ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ ПРИЛОЖЕНИЯ

Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Проектируемая система представляется в виде множества сущностей или акторов, взаимодействующих с системой с помощью так называемых вариантов использования.

Актором (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик.

В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актору. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актором. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

Диаграмма вариантов использования представляет собой граф специального вида, который является графической нотацией для представления конкретных вариантов использования, акторов, возможно некоторых интерфейсов, и отношений между этими элементами.

В разрабатываемой информационной системе можно выделить следующие группы акторов:

* администраторы;
* пользователи.

Каждая из групп имеет свои возможности использования системы. Некоторые варианты использования для них совпадают. При входе в систему для каждого происходит авторизация. Администраторам и пользователям доступен интуитивно понятный интерфейс и определенный функционал.

В информационной системе администраторы отвечают за:

* обновление статуса заявки;
* просмотр клиентов;
* рассмотрение клиентских операций (отклонение или одобрение).

В разрабатываемой системе пользователи могут:

* просматривать содержимое своих сборов;
* сформировать новою операцию;
* отменить операцию.

Диаграмма вариантов использования представлена на рисунке В.1 приложения В.

# БЛОК-СХЕМА АЛГОРИТМА РАБОТЫ ВСЕЙ ПРОГРАММЫ И АЛГОРИТМА РАБОТЫ НЕСКОЛЬКИХ ОСНОВНЫХ МЕТОДОВ

Алгоритм работы написанной программы представлен на рисунке Г.1 приложения Г.1. В главном меню необходимо выбрать режим входа в программу: в качестве администратор или пользователя. Завершить программу можно выбором пункта «Выход» главного меню. Пользователь может либо зарегистрироваться в программе, если у него нет созданного аккаунта, либо войти в существующий. Для доступа в программу, администратору и пользователю необходимо правильно ввести логин и пароль. При неправильном вводе данных будет предложен повторный ввод или возращение в начальное меню. После успешного входа пользователям и администратору предоставляется меню с определенным набором функций, доступное пока они выйдут из личного аккаунта в главное меню.

На рисунке Г.2 изображён алгоритм считывания операций из файла. Его суть состоит в том, что мы проходимся по всем строкам в файле операци. Потом мы по очереди считываем из файла операции, после каждой из которых проверяя следующий символ в потоке с помощью методы peek(). Если следующим символом оказался символ конца файла, то устанавливается флаг ошибки, а файл закрывается.

# ОПИСАНИЕ АЛГОРИТМА ЗАПУСКА ПРИЛОЖЕНИЯ, ЕГО ИСПОЛЬЗОВАНИЯ, РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ, ТЕСТИРОВАНИЯ ОБРАБОТКИ ОШИБОК

## Главное меню программы

На рисунке 8.1 изображено главное меню программы, которое открывается после запуска программы. Данное меню позволяет авторизоваться как пользователь или как администратор. Также оно позволяет завершить работу программы.

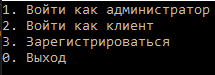


Рисунок 8.1 – Главное меню программы

## Вход в режиме администратора

Для входа в программу в режиме администратора необходимо ввести логин и пароль. Если неправильно введены данные, появится сообщение об ошибке (рисунок 8.2 и 8.3).



Рисунок 8.2 – Ввод неправильного логина

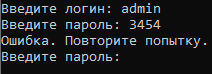


Рисунок 8.3 – Ввод неправильного пароля

При правильном вводе данных администратор успешно входит в информационную систему. Он получает доступ к меню администратора (рисунок 8.4).

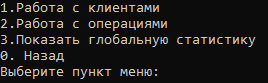


Рисунок 8.4 – Меню администратора

Пункт меню «Работа с клиентами» представляет собой подменю с функциями для работы с клиентами, которое изображено на рисунке 8.5.

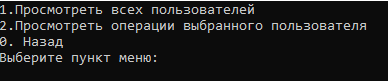


Рисунок 8.5 – Меню работы с клиентами

Первый пункт подменю работы с клиентами показывает информацию обо всех клиентах, совершавших транзакции(рисунок 8.6).



Рисунок 8.6 – Информация о клиентах

Второй пункт меню показывает операций конкретного клиента для этого необходимо ввести логин клиента. Если клиента с таким логином нет, выведется соответствующее сообщение (рисунок 8.7).



Рисунок 8.7 – Ввод неправильного логина

При вводе правильного логина на экран выведется история операций клиента (рисунок 8.8).

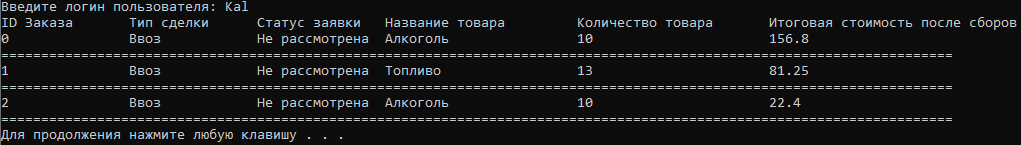


Рисунок 8.8 – Операции

Пункт меню «Работа с операциями» представляет собой подменю с функциями для работы с операциями, которое изображено на рисунке 8.9.

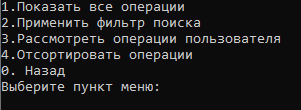


Рисунок 8.9 – Меню работы с операциями

Первый пункт подменю работы с операциями показывает информацию обо всех операциях (рисунок 8.10).

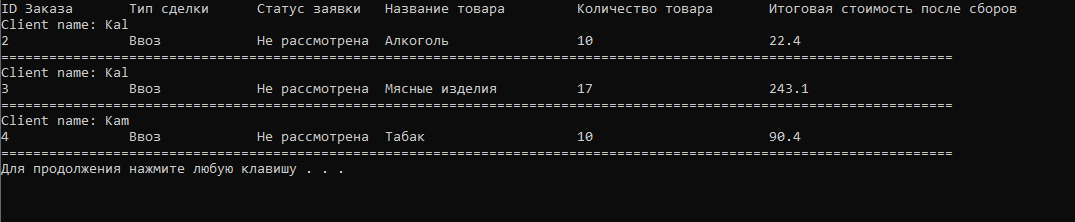


Рисунок 8.10 – Информация об операциях

Второй пункт открывает подменю фильтрации поиска, который позволяет отобрать необходимую информацию (рисунок 8.11).

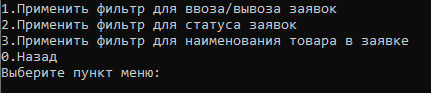


Рисунок 8.11 – Меню фильтрации

Третий пункт открывает меню для ввода логина пользователя, заявки которого нужно рассмотреть. (рисунок 8.12 ).

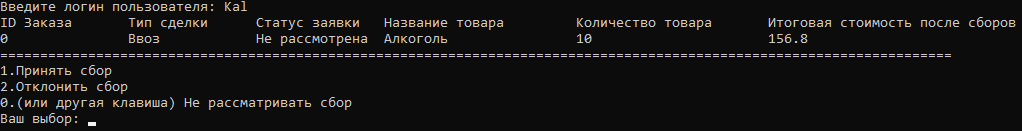


Рисунок 8.12 – Меню рассмотрения сборов

Четвертый пункт открывает подменю сортировки, которое позволяет отсортировать записи операций в файле по необходимым полям (рисунок 8.13).

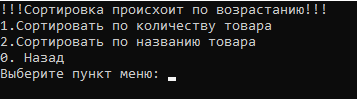


Рисунок 8.13 – Меню сортировки

Третий пункт в главном меню администратора показывает полную статистику по программе. (рисунок 8.14 ).

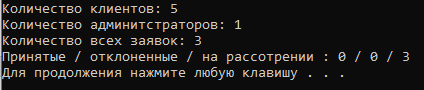


Рисунок 8.14 – Полная статистика программы

## Вход в режиме клиента

При входе в режиме клиента откроется меню клиента (рисунок 8.15).

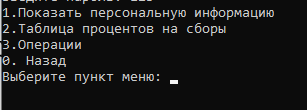


Рисунок 8.15 – Меню клиента

В данном меню можно просмотреть свою персональную информацию (рисунок 8.16).



Рисунок 8.16 – Персональная информация

Пункт «Таблица процентов на сборы» открывает подменю для просмотра сборов на ввоз и вывоз товаров (рисунок 8.17).

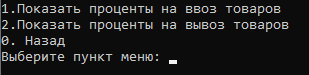


Рисунок 8.17 – Меню таблиц со сборами

Первый и второй пункт идентичны, но выводят на экран разные проценты от сборов из-за разницы в выборе ввоза или вывоза.

Пункт «Операции» открывает меню для работы с операциями (рисунок 8.18).

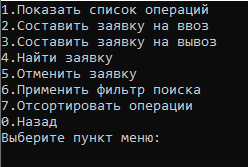


Рисунок 8.18 – Меню работы с операциями

Первый пункт выводит на экран информацию об операциях клиента.

Второй и третий пункт позволяет оформить соответственно ввоз или вывоз. Для этого необходимо ввести идентификационный номер товара, количество и цену за единицу товара. При вводе происходит проверка на правильный ввод данных.

При составлении сбора выводиться соответствующая таблица и просьба ввести данные (рисунок 8.19).

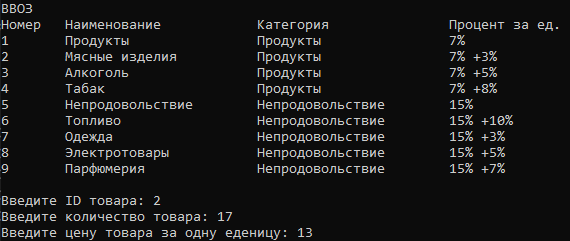


Рисунок 8.19 – Оформление операции

Четвёртый пункт позволяет найти операцию. Для этого необходимо ввести идентификационный номер операции. При вводе неправильного номера операции или этот номер принадлежит другому пользователю то выведется сообщение (рисунок 8.20).



Рисунок 8.20 – Ввод неправильного номера

Пятый пункт отменяет операцию пользователя. Для этого необходимо ввести идентификационный номер операции (рисунок 8.21).



Рисунок 8.21 – Отмена операции

Шестой пункт аналогичен пункту в администраторской панели.

Седьмой пункт открывает подменю сортировки, аналогичное в администраторском меню. Оно выводит информацию только об операциях пользователя, находящегося в системе (рисунок 8.22).

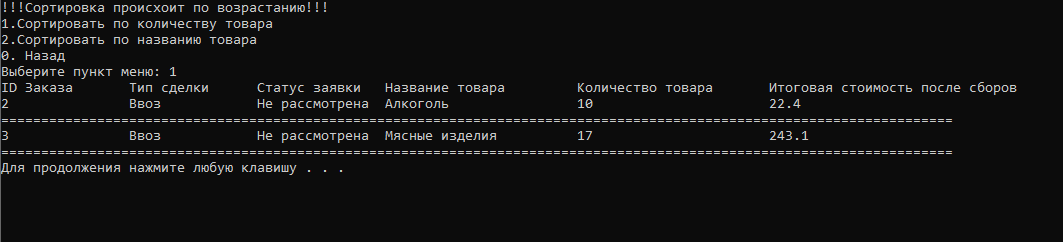


Рисунок 8.22 – Отсортированный вывод пользователю

## Регистрация

Чтобы зарегистрироваться в системе, нужно выбрать пункт «Зарегистрироваться» из главного меню. Затем нужно ввести новый логин и пароль. Если логин занят, выводится сообщение (рисунок 8.23).

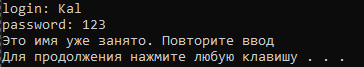


Рисунок 8.23 – Ввод занятого логина

При успешной регистрации выводится сообщение (рисунок 8.24).



Рисунок 8.24 – Успешная регистрация

# ЗАКЛЮЧЕНИЕ

В курсовом проекте были исследованы процессы составления и управления операциями импорта и экспорта. Реализована процедура разделение ролей, авторизации и регистрации пользователя, хранение пароля в зашифрованном виде. Для работы с данными были рассмотрены такие процедуры, как сортировка, поиск и фильтрация записей по определенным параметрам, добавление, просмотр, редактирование и удаление информации. Исходные и итоговые данные сохранялись в файлы.

Для достижения поставленных целей в курсовой работе была разработана объектная модель и ее программная реализация на языке C++. При разработке объектной модели были определены ее составные части – классы и установлены связи между ними. Для наглядного представления была построена диаграмма классов. Также были построены блок-схемы алгоритма работы всего приложения и основных ее функций. Была разработана и описана диаграмма вариантов использования приложения, показывающая доступный администратору и пользователя функционал.

Также в разработанной программе реализованы базовые принципы объектно-ориентированного программирования: абстракции, инкапсуляции, наследования и полиморфизма, использованы стандартные и пользовательские функции, дружественные и виртуальные функции, пространство имён. Были использование перегрузки методов и операторов, ­ использовались шаблонные классы и методы, динамическое выделения памяти и умные указателей.

В программе были предусмотрены обработки исключительных ситуаций посредством языка С++.

Программа была детально описана, что поможет в будущем пользователю понять все особенности работы с ней.

# Список использованных источников

1. Программные аналоги [Электронный ресурс]. – Режим доступа: <https://www.ekam.ru/blogs/pos/ekspertnyy-reyting-programm-skladskogo-ucheta>
2. Методология IDEF0 [Электронный ресурс]. – Режим доступа: <https://itteach.ru/bpwin/metodologiya-idef0>
3. Основы методологии IDEF1X [Электронный ресурс]. – Режим доступа: <https://www.cfin.ru/vernikov/idef/idef1x.shtml>
4. Текстовые файлы [Электронный ресурс]. – Режим доступа: <http://fileext.ru/txt>
5. Фомичев В. М. Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010. – 198 с.
6. Диаграмма классов [Электронный ресурс]. – Режим доступа: <https://flexberry.github.io/ru/gpg_class-diagram.html>
7. Диаграмма вариантов использования [Электронный ресурс]. – Режим доступа: <https://it.wikireading.ru/5107>

# Приложение А (обязательное) Листинг программы

//Использание собственных пространств имён

namespace cast

{

void LoginMenu(shared\_ptr<User> user);

};

//Использание встроенных пространств имён

using namespace std;

//Абстрактный класс

class User

{

//Инкапсуляция

protected:

//Статические поля

inline const static string OperationsFileName = "Operations.txt";

inline const static string AdminsFileName = "Admins.txt";

inline const static string UsersFileName = "Users.txt";

public:

//Шаблонный метод

template <typename T>

T IsUserExist(string login) {

ifstream file;

string line;

file.open(UsersFileName, ios::app);

while (getline(file, line)) {

if (login == FindName(line.c\_str())) {

return true;

}

}

return false;

file.close();

}

//Полиморфизм. Абстрактный класс

virtual void UserMenu() = 0;

};

//Наследование

class Client : public User

{

public:

//Переопределние метода

void UserMenu() override;

//Перегрузка методов

void ShowFeesTable();

void ShowFeesTable(int table);

};

Продолжение приложения Б

{

int choice;

//Использование умных указателей

shared\_ptr<User> user = nullptr;

while (true)

{

system("cls");

cout <<

"1.Зайти как пользователь\n"

"2.Зайти как администратор\n"

"3.Регистрация\n"

"0.Выход\n";

choice = IntInput("Выберите пункт меню: ");

system("cls");

switch (choice)

{

case 1:

//Полиморфизм. Переопределение объекта

user = shared\_ptr<User>(new Client());

LoginMenu(user);

user = nullptr;

break;

case 2:

user = shared\_ptr<User>(new Admin());

LoginMenu(user);

user = nullptr;

break;

}

{

//Использование встроенных контенеров C++

list <string> ListOfOperations;

multimap <int, string> mapOper;

string line;

ifstream file;

fstream out;

file.open(OperationsFileName, ios::app);

if (choice == 1) {

//Использование флагов

for (int i = 0; file.peek() != EOF; i++) {

getline(file, line);

//Помещение данных в контейнер

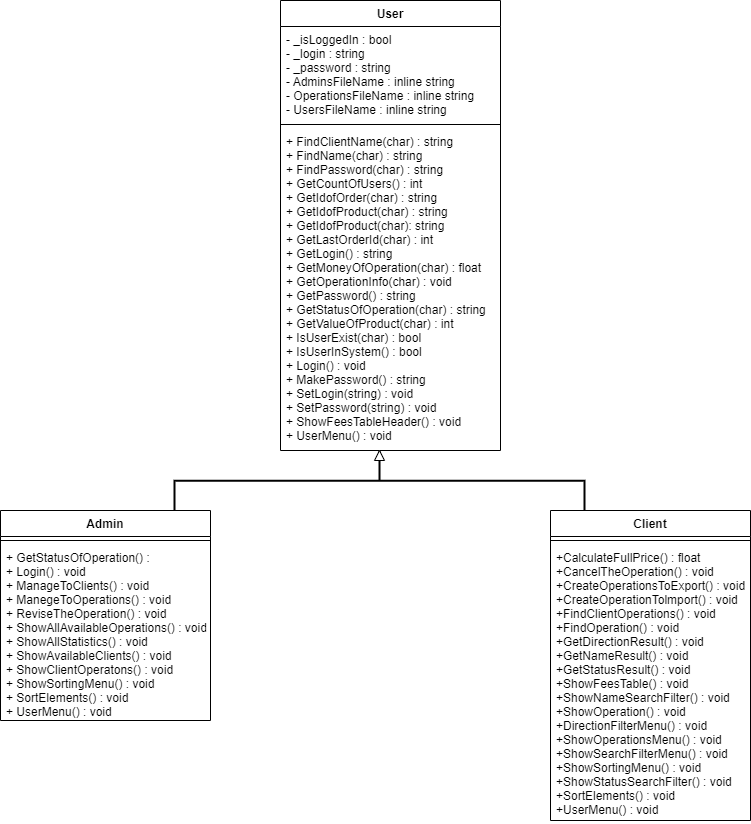
mapOper.emplace(GetValueOfProducts(line.c\_str()), line);

}

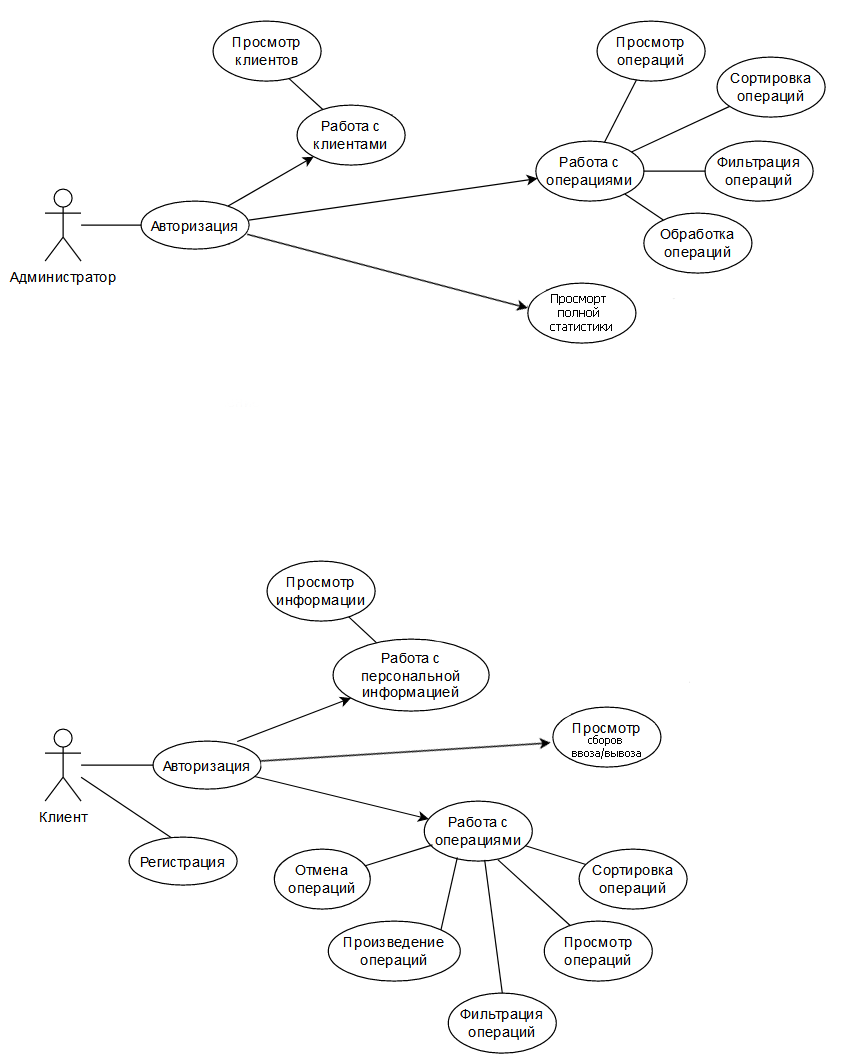
file.close();

}

# Приложение Б (обязательное) Диаграмма классов



# Приложение В (обязательное) Диаграмма вариантов использования



# Приложение Г (обязательное) Схемы алгоритмов программы

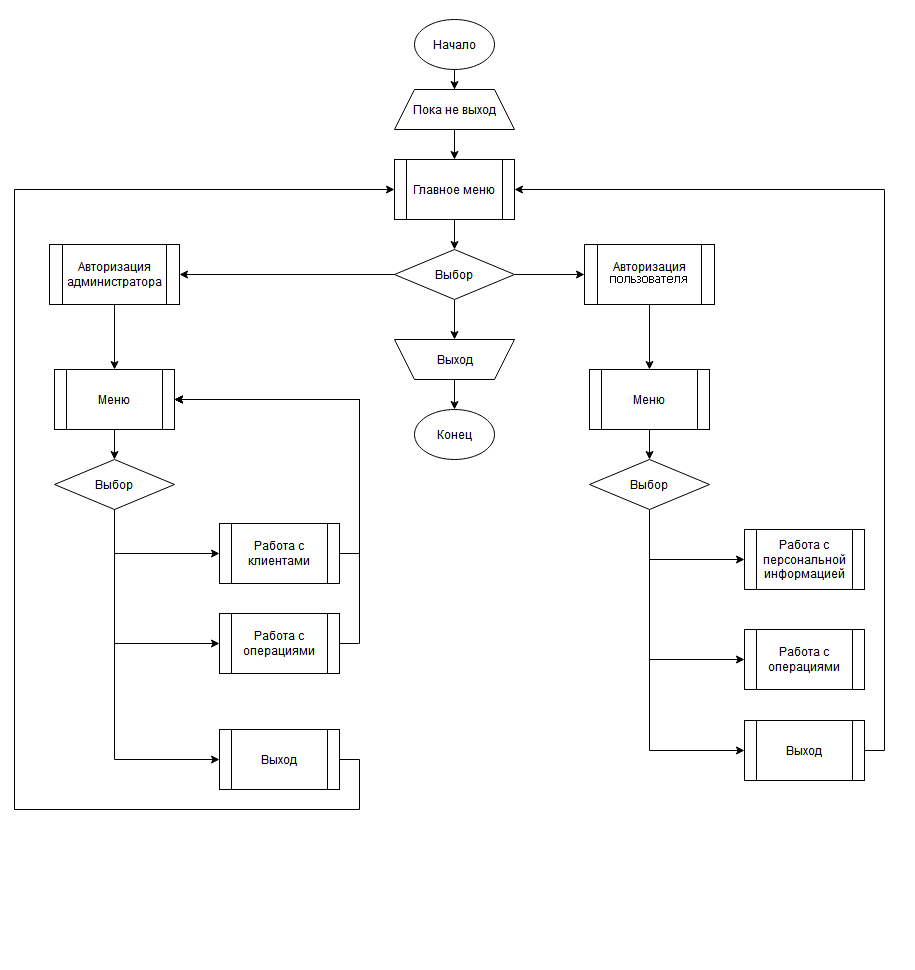


Рисунок Г.1 – Схема работы всей программы

Продолжение приложения Б

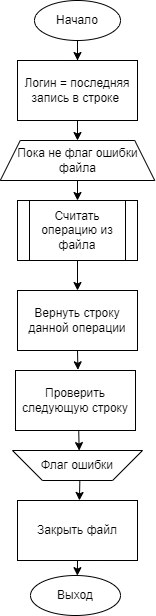


Рисунок Г.2 – Алгоритм считывания операций из файлов