Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Программирование на языках высокого уровня

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Басак

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

«ИНФОРМАЦИОННАЯ СИСТЕМА АВТОМОБИЛЬНОГО МАГАЗИНА»

БГУИР КП 1-40 02 01 502 ПЗ

Студент: Бейнар А.В.

Руководитель: Д. В. Басак

Минск 2022

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_Б. В. Никульшин

(подпись)

––"–––––––––––––––––2022г.

ЗАДАНИЕ

по курсовому проектированию

Студенту  *Бейнару Александру Владимировичу–––––––––––*

1. Тема проекта  «Информационная система автомобильного магазина»

2. Срок сдачи студентом законченного проекта  *26 декабря 2022 г.–––*

3. Исходные данные к проекту *Users.txt (хранятся данные о пользователях), Store.txt (хранится информация об автомобильном магазине), Cars.txt(хранятся данные о машинах), Orders.txt (хранится информация о заказах), Reports.txt (хранятся сформированные отчеты). Контейнер: свой – Container и STL – vector.*

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

*Введение. Содержание. 1. Анализ предметной области программного средства 2. Структура используемых данных. 3. Описание созданных программных конструкций. 4. Разработка и описание диаграммы классов. 5. Разработка и описание диаграммы вариантов использования приложения. 6. Разработка схем алгоритмов работы программы. 7. Описание программы. Заключение. Список использованных источников.*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*1. Диаграмма классов*

*2. Схема алгоритма метода addCar (); (Добавление новой машины в список)*

*3. Схема алгоритма метода delOrder(); (Удаление записи заказа)*

*4. Скриншоты работающей программы––––*

6. Консультант по проекту *Басак Д.В.*

7. Дата выдачи задания *14 сентября 2022 г.–––––– –––––––––*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

*разделы 1, 2 к 1 октября 2022 г. – 20 %;*

*разделы 3, 4 к 1 ноября 2022 г. – 30 %;*

*разделы 5, 6 ,7 к 1 декабря 2022 г. – 30 %;*

*оформление пояснительной записки и графического материала к 14 декабря 2022 г. – 20 %*

*Защита курсового проекта с 21 декабря 2022 г. по 28 декабря 2022 г.*

РУКОВОДИТЕЛЬ *Д. В. Басак*

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ *А. В. Бейнар*

(дата и подпись студента)

**Содержание**

[Введение 5](#_Toc123112361)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОГРАММНОГО СРЕДСТВА 6](#_Toc123112362)

[1.1 Обзор процесса функционирования автомобильного магазина 6](#_Toc123112363)

[1.2 Методы решения поставленной задачи 7](#_Toc123112364)

[2 Структура используемых данных 8](#_Toc123112365)

[3 Описание созданных программных конструкций 10](#_Toc123112366)

[4 Разработка и описание диаграммы классов 12](#_Toc123112367)

[5 Разработка и описание диаграммы вариантов использования приложения 13](#_Toc123112368)

[6 Разработка схем алгоритмов работы программы 15](#_Toc123112369)

[7 Описание программы 16](#_Toc123112370)

[7.1 Авторизация 16](#_Toc123112371)

[7.2 Меню администратора 18](#_Toc123112372)

[7.3 Меню менеджера 29](#_Toc123112373)

[7.4 Меню пользователя 29](#_Toc123112374)

[7.5 Тестирование и проверка работоспособности программного средства 30](#_Toc123112375)

[Заключение 34](#_Toc123112376)

[Список использованных источников 35](#_Toc123112377)

[Приложение А 36](#_Toc123112378)

[Приложение Б 38](#_Toc123112379)

[Приложение В 39](#_Toc123112380)

[Приложение Г 40](#_Toc123112381)

[Приложение Д 41](#_Toc123112382)

[ПРИЛОЖЕНИЕ Е 42](#_Toc123112383)

# **Введение**

Информационные технологии играют большую роль в современном мире. Благодаря их развитию существенно сократилось время на обработку и анализ информации. На данный момент созданы разнообразные программы для эффективной работы предприятий и организаций, которые автоматизирует однотипные процессы, производят вычисления и анализируют данные. Во многих видах деятельности без этих технологий невозможно осуществлять успешную работу.

Автомобильные магазины также нуждаются в информационных системах, которые позволяют автоматизировать и оптимизировать их деятельность. Специализированные информационные системы могут существенно снизить затраты на неэффективные и повторяющиеся процессы. Программа, разработанная в ходе выполнения курсовой работы, поможет улучшить качество обслуживания клиентов и облегчит монотонную работу бухгалтеров. Также разработанная информационная система позволит экономить время для проведения больших чеков. Программа может применяться всеми автомобильными магазинам. Учет информации, необходимой для функционирования магазина, требует больших ресурсов для обработки и хранения информации, что было бы трудоемким процессом при выполнении человеком, поэтому качественное программное обеспечение очень необходимо в данной сфере

Целью курсовой работы является повышение производительности труда сотрудников автомобильного магазина за счет автоматизации процессов, связанных с продажей автомобилей.

Для полного выполнения курсовой работы необходимо выполнить ряд задач:

* описать предметную область;
* проанализировать изучаемые процессы;
* выполнить программную реализацию;
* выполнить тестирование проектных решений.

Объектом исследования выступает процесс автоматизации системы функционирования автомобильного магазина.

# **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОГРАММНОГО СРЕДСТВА**

## **1.1 Обзор процесса функционирования автомобильного магазина**

Магазин продажи автомобилей должен быстро и качественно, как и любой другой магазин, обслуживать клиентов. Информационная система магазина будет использоваться как работниками, так и клиентами.

В процесс деятельности автомобильного магазина входит: обслуживание покупателей, прием заказов, оформление заказов, заказ и поставка автомобилей, предложение и выполнение сервисных услуг, оформление кредитов, сотрудничество с другими организациями, оптовая и розничная продажа, работа с поставщиками, прием товара на основном складе и выдача по складам отделов, гарантийное сервисное обслуживание и коммерческий ремонт, создание рекламных баннеров, листовок, акций и скидок.

Общие или главные цели деятельности автомобильного магазина устанавливаются высшим руководством и согласуются со всеми подразделениями. В отделе продаж, работа менеджера по продажам входит консультация и демонстрация товара, предложение действующих акций и сервиса, помощь в выборе, заказ автомобилей, далее выбор оплаты наличными, по карте и оформление кредита. Работа отдела закупок и продаж заключается: прием заказа автомобилей в магазин, прием и оформление заказа по предоплате, поиск клиентов для сотрудничества и предложение своих услуг, работа с организациями по безналичному расчету, консультирование и помощь в выборе машины, далее выписка счетов, отслеживание оплаченных счетов, закупка автомобилей на магазин, на организации по оплаченным счетам, закупка комплектующих в отдел сервиса, контроль отгрузки товара организациям и ведения документооборота. Работник основного склада принимает от отдела закупок и продаж список заказа и загружает в информационную систему, далее поступает товар на склад и по товарным накладным проверяется количество, комплектация и целостность товара. Товар отгружается по складам отделов, выдаются накладные и ценники. Отдел сервиса производит гарантийное и коммерческое обслуживание.

Большие потоки информации о заказах и поставках нового товара требует обработки данных и просто не могут быть оставлены без контроля. Именно поэтому так важно правильно вести учет продаж и товаров на складе с помощью информационных систем, которые существенно сокращают затраты время на обработку больших массивов данных.

## **1.2 Методы решения поставленной задачи**

Целью данного курсового проекта является создание информационной системы функционирования автомобильного магазина. Программа разрабатывалась с учетом объектно-ориентированного программирования. Система тестирования поддерживает разделение ролей, а именно: администратор, менеджер и пользователь. Основными используемыми объектами в системе являются машина, как товар, и заказ, представленные отдельными классами «Car» и «Order» соответственно.

Для хранения данных используются текстовые файлы. Также для обеспечения безопасности учетных записей используется шифрование с «солью».

При сортировке данных используется сортировка Шелла. Поиск по контейнерам, хранящим информацию о продукции, ведется в основном линейно. Фильтрация производится на основе сравнения характеристик, которые должны подходить под критерий фильтрации у объектов товара.

Для оптимизации работы большинства функций в качестве параметров передаются не сами объекты, а указатели, хранящие адрес на динамически выделенный участок память под объект.

# **2 Структура используемых данных**

Для хранения данных на накопителе используются файлы с расширением txt. Это наиболее часто используемые по умолчанию текстовые файлы, обеспеченные всем необходимым для взаимодействия c программой.

Для обеспечения безопасности учетных записей используется хеширование паролей. Данная операция является обязательной, так как при отсутствии шифрования паролей злоумышленники с легкостью смогут воспользоваться учетными записями при получении файла с ними.

В данной программе используется шифрование паролей алгоритмом SHA-1. На рисунке 2.1 представлен файл с учетными записями.

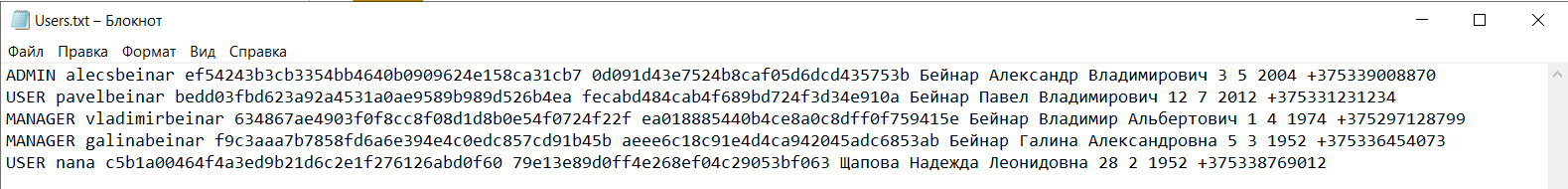


Рисунок 2.1 – Файл с учетными записями

На рисунке 2.2 представлен файл, который хранит в себе данные о машинах в автомобильном магазине, а именно: код, марка машины, модель машины, цена, количество в наличии.

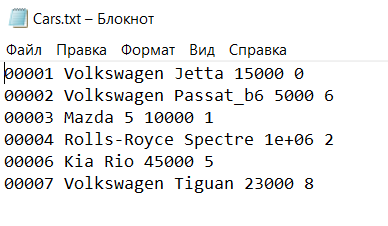


Рисунок 2.2 – Файл с данными о товарах

На рисунке 2.3 представлен файл, который хранит в себе данные о реализованных товарах, а именно: код добавления товара, код товара, продавец, дата.

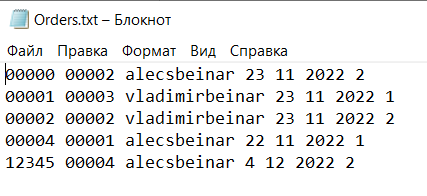


Рисунок 2.3 – Файл с данными о реализованном заказе

На рисунке 2.4 представлен файл, который хранит в себе данные о магазине, а именно: название магазина, город, адрес, директор, дата открытия.

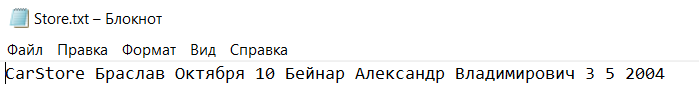


Рисунок 2.4 – Файл с данными магазине

На рисунке 2.5 показано содержание файла, который хранит в себе отчеты о реализованных заказа за период времени.

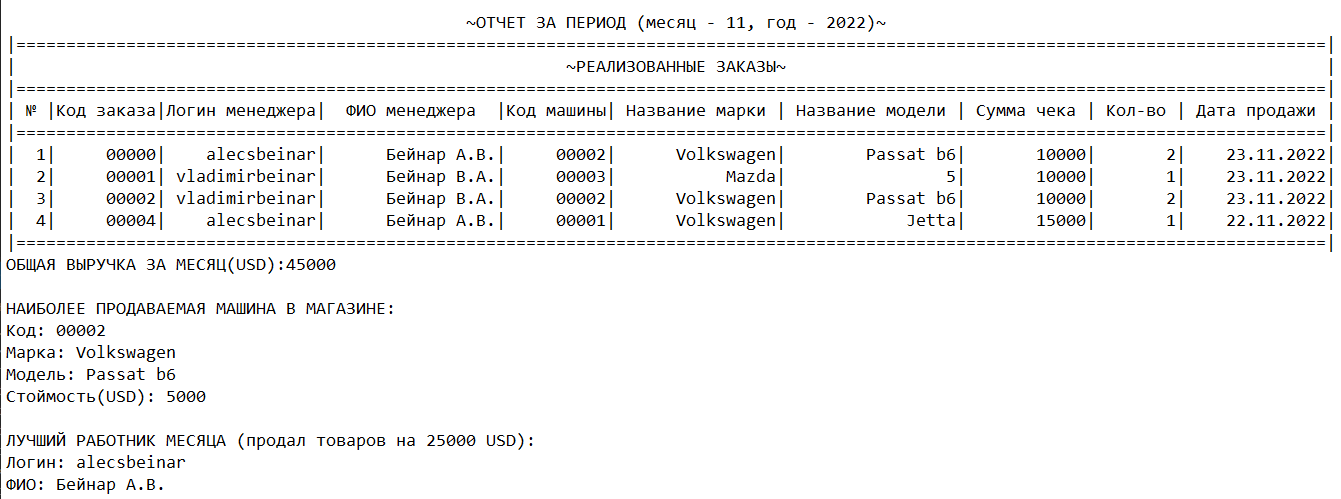


Рисунок 2.5 – Файл с отчетом

# **3 Описание созданных программных конструкций**

Для описания структуры данных используется объектно-ориентированная парадигма, которая позволяет разложить любую задачу на составные части - классы, каждый из которых становится самостоятельным объектом. Каждый класс содержит свой собственный код и данные, которые относятся к этому объекту. Ниже приводится описание классов и их полей «Address», «FullName», «Human», «Date», «Order», «Car», «Store», «Container», «User», «UseCases».

class FullName: public BasicInfo

{

string surname;

string name;

string patronymic;

string fullName;

}

class Address: public BasicInfo

{

string city;

string street;

string house;

string fullAddress;

}

class Date: public BasicInfo

{

int day;

int month;

int year;

string fullDate;

}

template <class T>

class Container

{

T\* Arr;

T\* head;

T\* tail;

int size;

}

class UseCases

{

Container<User> containerUser;

Container<Car> containerCar;

Container<Order> containerOrder;

Container<User> BackUpcontainerUser;

Container<Car> BackUpcontainerCar;

Container<Order> BackUpcontainerOrder;

map<string, int> mp = { {FILE\_USERS, 1}, {FILE\_Cars, 2}, {FILE\_ORDERS, 3} };

}

class Car

{

string brand;

string model;

string code;

double price;

int numberOf;

}

class Order

{

string code;

string deviceCode;

string loginManager;

Date date;

int numberOf;

}

class Store

{

string name;

Address address;

FullName directorFIO;

Date openingDate;

}

class Human

{

FullName FIO;

Date birthday;

string number\_telephone;

}

class User: public Human

{

string status;

string login;

string salted\_hash\_password;

string salt;

}

В качестве способа объединения входных данных используется контейнер.

# **4 Разработка и описание диаграммы классов**

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей (отношений) между ними. Она служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений.

Диаграммы классов процесса функционирования автомобильного магазина представлены в приложении Б.

Программная реализация представлена такими классами, как «Address», «FullName», «Date», «BasicInfo», «Order», «Store», «UseCases», «DataProcess», «Car», «User», «Human», «Container <T>».

# **5 Разработка и описание диаграммы вариантов использования приложения**

Диаграмма вариантов использования – это графическое изображение возможных взаимодействий пользователя с системой. Диаграмма вариантов использования показывает различные варианты использования и различные типы пользователей, которые есть в системе.

На рисунке 5.1 представлена диаграмма вариантов использования приложения. В разрабатываемой программе предусмотрена авторизация под одной из трех ролей: администратор, менеджер и пользователь. Для каждой из ролей предусмотрен свой функционал. Администратор обладает неограниченными возможностями и может выполнять любые операции, предусмотренные программой: редактирование данных о магазине, управление пользователями, процедуры поиска и фильтрации, удаление, добавление и редактирование записей, авторизация, выход, формирование отчета, просмотр данных, отмена последнего действия. Менеджер обладает похожим функционалом, однако он не может редактировать данные о магазине и управлять учетными записями. Пользователь обладает ограниченным функционалом: поиск, фильтрация и просмотр данных, авторизация, выход, формирование отчета и редактирование персональных данных.

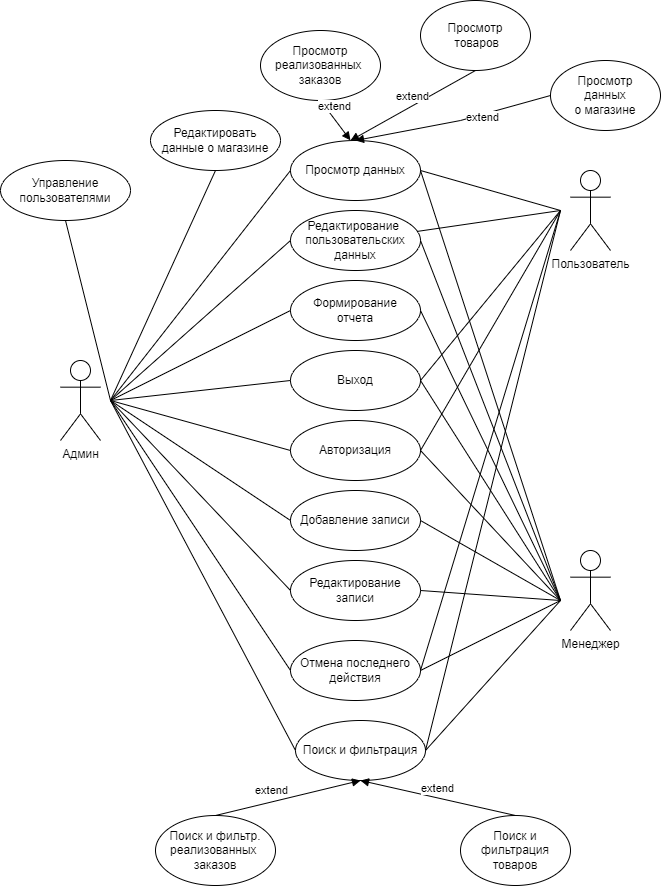


Рисунок 5.1 – Диаграмма вариантов использования

# **6 Разработка схем алгоритмов работы программы**

Разработка алгоритмов является важной частью создания программы. Они записываются в виде последовательности шагов решения задачи от ввода исходных данных до вывода результатов. Разработка алгоритма включает в себя выбор методов проектирования алгоритма, выбор формы записи алгоритма, выбор метода тестирования.

В приложении В показан алгоритм главной функции «main». При запуске программного продукта в первую очередь происходит считывание данных о магазине, пользователях, товарах и заказах. Далее пользователь может выбрать один из пунктов: регистрация, авторизация и выход. При авторизации в зависимости от роли человек попадает либо в меню пользователя, либо в меню администратора.

В приложения Г и Д показаны алгоритмы функций добавления новой машины «addCar» и удаления записи «delOrder» соответственно.

# **7 Описание программы**

## **7.1 Авторизация**

При первом запуске программы необходимо ввести данные об автомобильном магазине: название, адрес, ФИО директора и дату открытия. Представлено на рисунке 7.1.1.

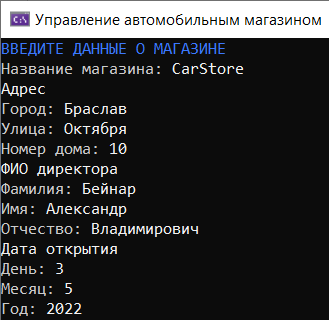


Рисунок 7.1.1 – Ввод данных о магазине электроники

Далее необходимо зарегистрировать главного администратора, где пользователь вводит необходимые данные (рис 7.1.2).

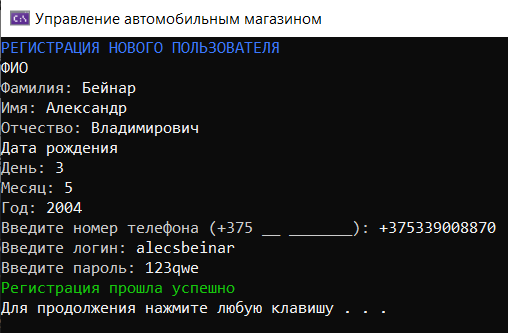


Рисунок 7.1.2 – Ввод данных главного администратора

После входа в программу отображается главное меню (рис. 7.1.3), которое состоит из 3 пунктов: «Авторизация», «Регистрация», «Выход из программы».

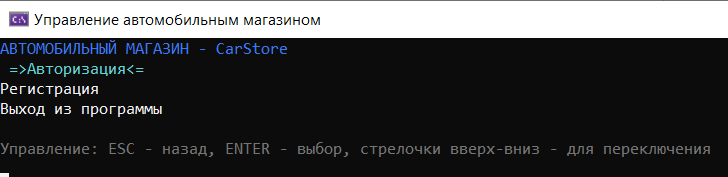


Рисунок 7.1.3 – Главное меню программы

При выборе пункта меню «Авторизация» пользователь должен ввести логин и пароль аккаунта (рис. 7.1.4), при вводе правильного логина и пароля он направляется в модуль пользователя, либо в модуль менеджера, либо в модуль администратора в зависимости от роли аккаунта.

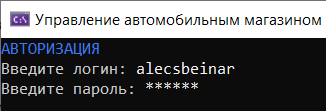


Рисунок 7.1.4 – Авторизация в программу

При отсутствии аккаунта пользователь может зарегистрировать новый, для этого необходимо выбрать пункт меню «Регистрация». После ввода несуществующего в базе данных логина и корректного пароля новый аккаунт пользователя заносится в базу данных (рис. 7.1.2).

Для обеспечения безопасности учетных записей используется хеширование паролей. Данная операция является обязательной, так как при отсутствии шифрования паролей злоумышленники с легкостью смогут воспользоваться учетными записями при получении файла с ними. Также стоит отметить, что при вводе пароля вводимые символы отображаются звездочками. Таким образом, данные меры повышают защищенность программы от несанкционированного доступа.

## **7.2 Меню администратора**

В меню администратора доступны следующие пункты (рис. 7.2.1): «Работа с данными», «Работа с учетными записями», «Изменение данных о магазине», «Редактирование персональных данных», «Отмена последнего шага», «Выйти из аккаунта».

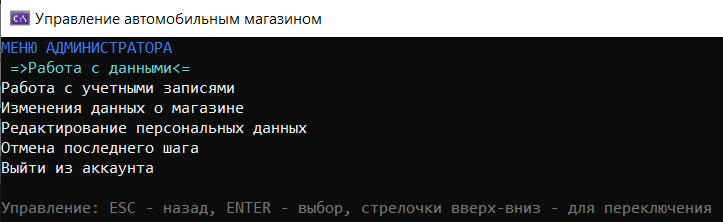


Рисунок 7.2.1 – Меню администратора

При выборе пункта «Работа с данными» выводится меню (рис. 7.2.2) со следующими пунктами: «Работа с продукцией», «Работа с заказами», «Формирование отчета за период».

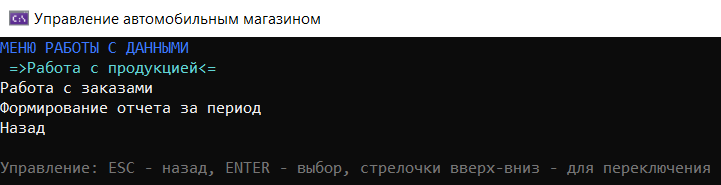


Рисунок 7.2.2 – Меню работы с данными

В меню «Работа с продукцией» доступны такие пункты, как «Добавление новой машины», «Редактирование информации о машине», «Просмотр машин», «Удаление машин», «Поиск и фильтрация», «Отмена последнего шага», «Назад». Представлено на рисунке 7.2.3

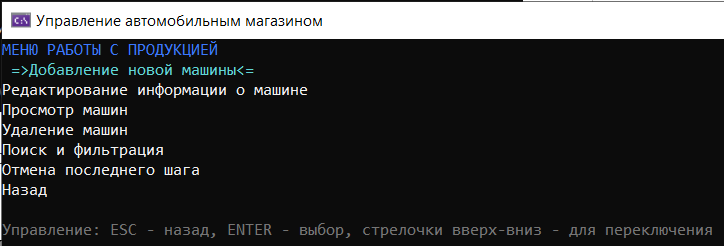


Рисунок 7.2.3 – Меню работы с продукцией

При выборе «Добавление новой машины» администратор может добавить новую машину в базу данных, для этого необходимо ввести уникальный код товара, который не повторяется в базе данных, марку, модель, цену товара, а также количество машин в наличии. На рисунке 7.2.4 представлено добавление новой машины.

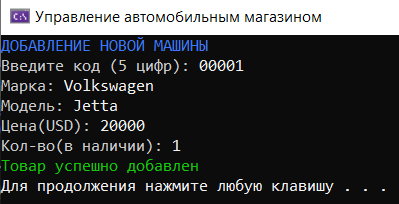


Рисунок 7.2.4 – Добавление новой машины

При выборе «Редактирование информации о машине» необходимо выбрать у какой машины будут изменены данные (рис 7.2.5). Далее выбирается какой параметр будет редактироваться (рис 7.2.6). На рисунке 7.2.7 показано изменение цены машины номер 2.

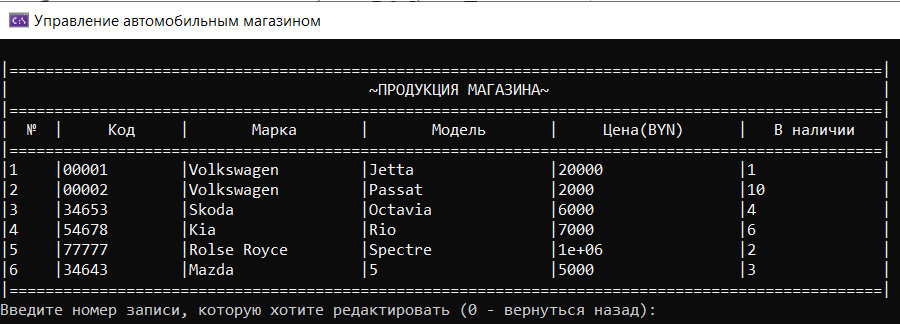


Рисунок 7.2.5 – Выбор машины для редактирования данных

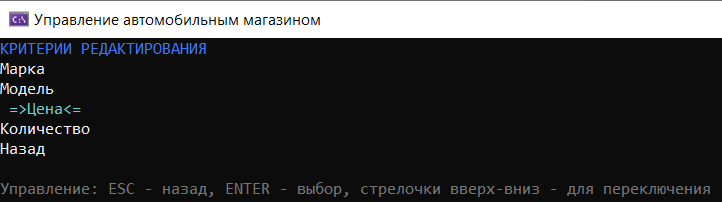


Рисунок 7.2.6 – Выбор параметра для редактирования данных

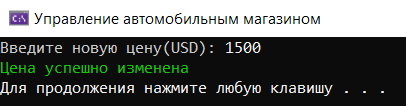


Рисунок 7.2.7 – Изменение цены машины

Для просмотра всех машин добавленных в базу данных необходимо выбрать «Просмотр машин» (рис 7.2.8).

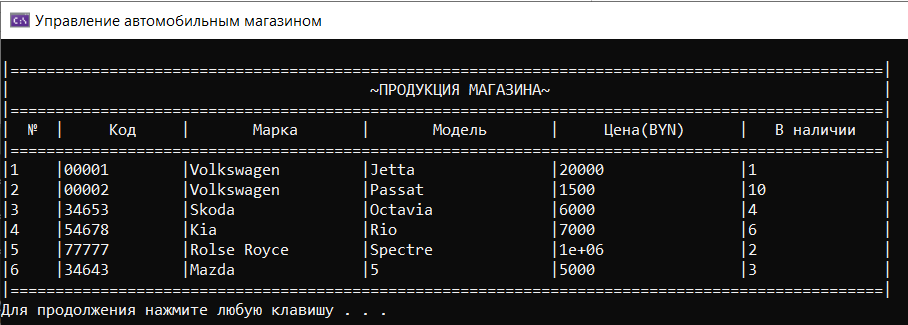


Рисунок 7.2.8 – Вывод всех машин автомобильного магазина

При необходимости удалить какую-либо машину администратор может выбрать пункт меню «Удаление машин», на экран выводится база данных с машинами, далее необходимо выбрать машину для удаления и подтвердить действие. На рисунке 7.2.9 показано удаление машины.

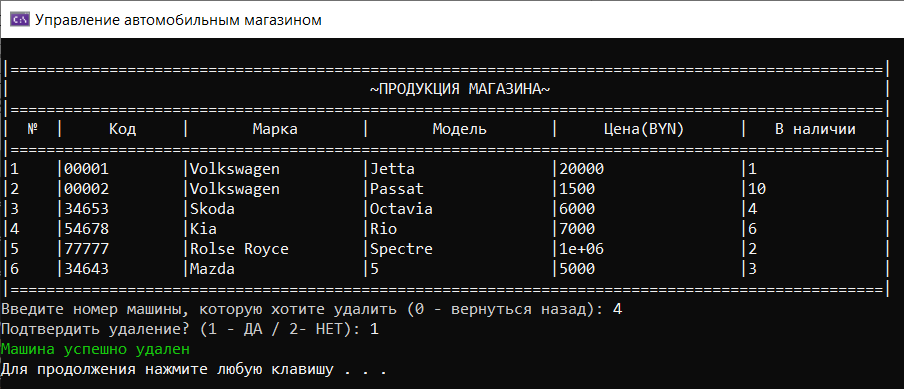


Рисунок 7.2.9 – Удаление машины

При выборе «Поиск и фильтрация» на консоли появляется меню поиска и фильтрации (рис. 7.2.10). В меню можно выбрать один из пунктов: «Поиск по коду машины», «Поиск по названию машины», «Фильтрация по цене», «Сортировка по цене», «Назад». На рисунке 7.2.11 показана сортировка товаров по цене.

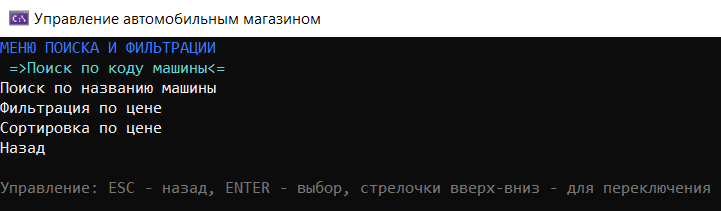


Рисунок 7.2.10 – Меню поиска и фильтрации

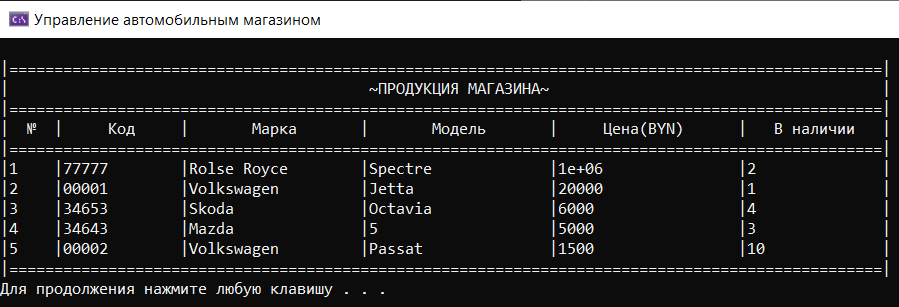


Рисунок 7.2.11 – Сортировка по цене

При выборе «Отмена последнего шага» надо подтвердить отмену последнего действия (рис. 7.2.12). После этого восстанавливаются прошлое состояние файла и контейнера с данными.

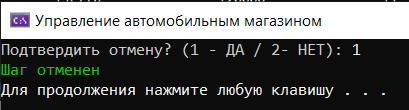


Рисунок 7.2.12 – Отмена последнего шага

При выборе в меню работы с данными пункта «Работа с заказами» администратор попадает в соответствующее меню (рис. 7.2.13).

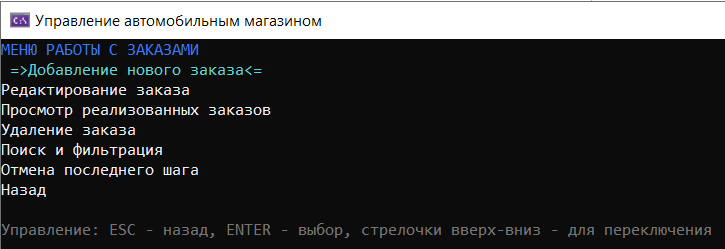


Рисунок 7.2.13 – Меню работы с заказами

Для добавления нового заказа необходимо выбрать 1 пункт. Далее нужно ввести уникальный код, выбрать машину из базы данных, ввести количество и дату проведения. На рисунке 7.2.15 показано добавление нового заказа.

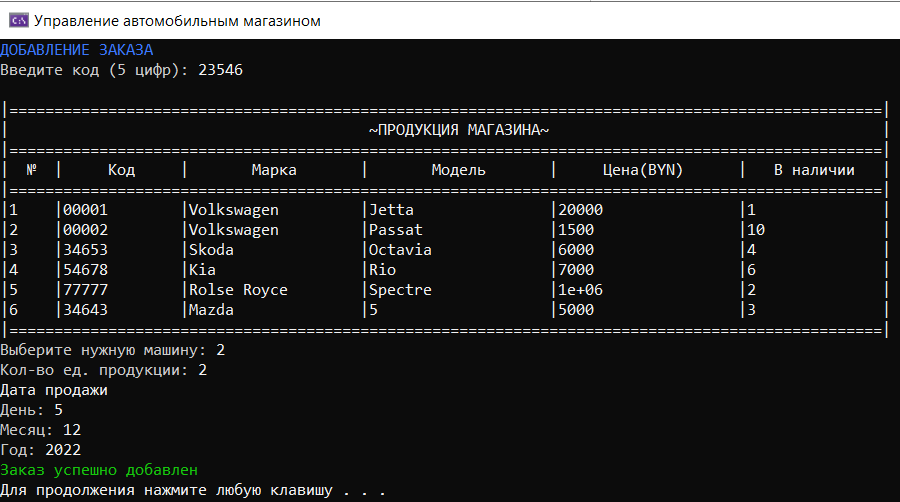


Рисунок 7.2.15– Добавление нового заказа

При необходимости редактирования заказа нужно выбрать второй пункт меню, далее выбирается заказ (рис 7.2.16) и параметр изменения данных (рис 7.2.17). На рисунке 7.2.18 показано изменение даты заказа.

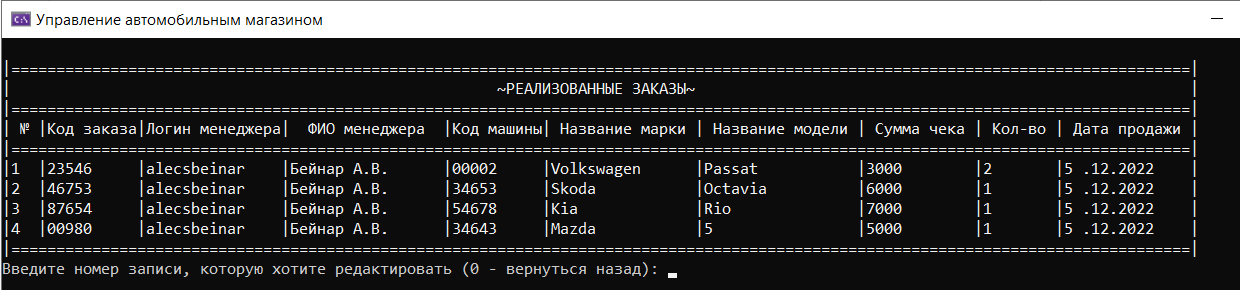


Рисунок 7.2.16 – Выбор заказа

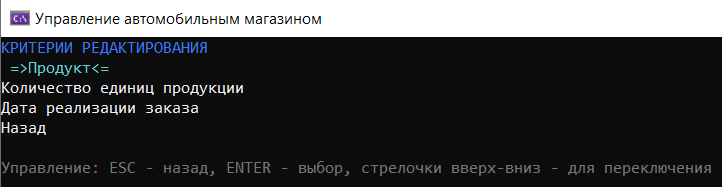


Рисунок 7.2.17 – Выбор параметра для изменения

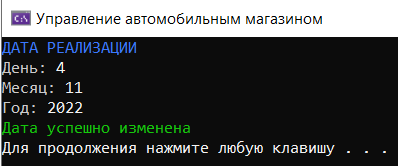


Рисунок 7.2.18 – Изменения даты заказа

При выборе «Просмотр реализованных заказов» на консоль выводятся все проведенные заказы (рис. 7.2.19).

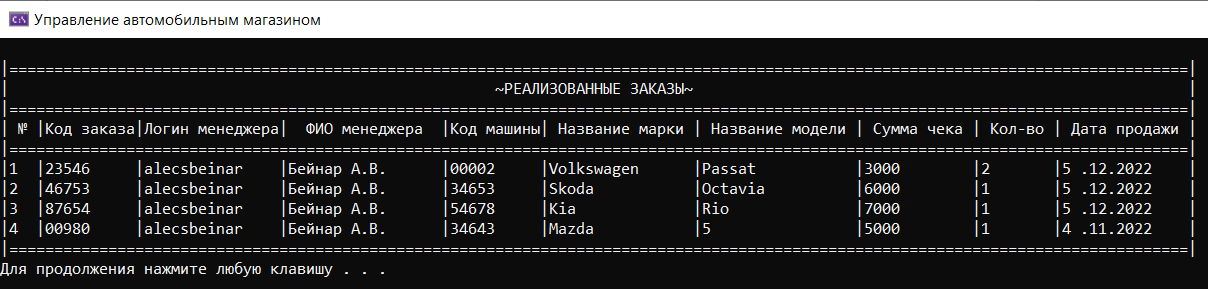


Рисунок 7.2.19 – Просмотр реализованных заказов

Для удаления заказа из базы данных нужно выбрать «Удаление заказа», где администратор выбирает какой из заказов будет удален. Представлено на рисунке 7.2.20.

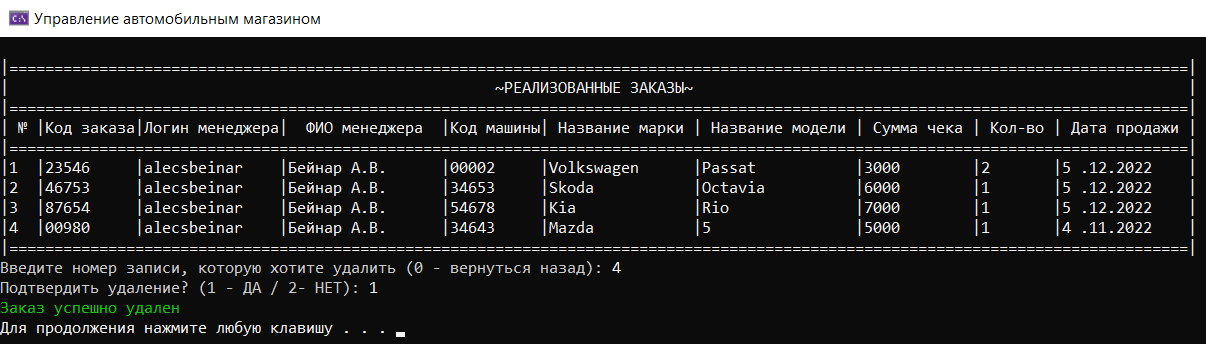


Рисунок 7.2.20 – Просмотр реализованных заказов

При выборе «Поиск и фильтрация» на консоли появляется меню поиска и фильтрации (рис 7.2.21). В меню можно выбрать один из пунктов: «Поиск по коду заказа», «Поиск по коду продукта», «Поиск по фамилии менеджера», «Фильтрация по дате реализации заказа», «Сортировка по дате реализации заказа», «Назад». На рисунке 7.2.22 показан поиск по коду заказа.

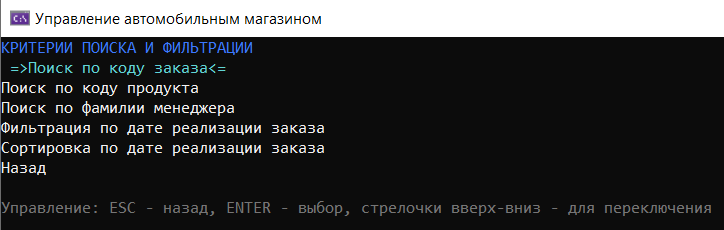


Рисунок 7.2.21 – Меню поиска и фильтрации

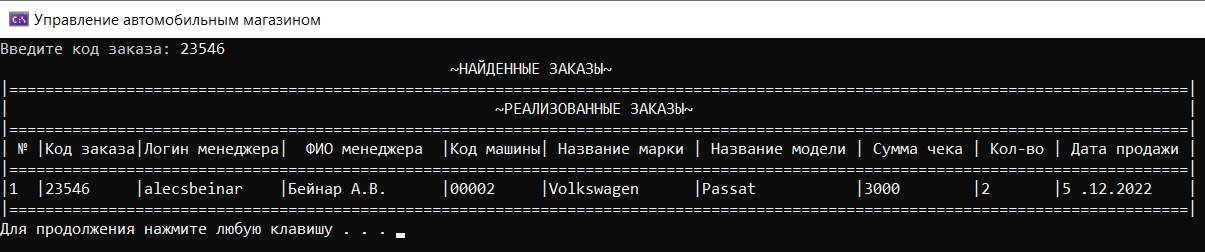


Рисунок 7.2.22 – Поиск реализованных заказов по коду

В программе предусмотрена такая функция, как «Формирование отчета за период», где администратор может сформировать отчет за определенный месяц, для этого необходимо ввести месяц и год. Далее формируется отчет, где показаны реализованные заказы за период, общая выручка за месяц, а также проводится аналитика: вычисляется самый продаваемый товар в магазине и находится лучший работник месяца. При необходимости отчет можно записать в файл. На рисунке 7.2.23 показан сформированный отчет за 12 месяц 2022 года.

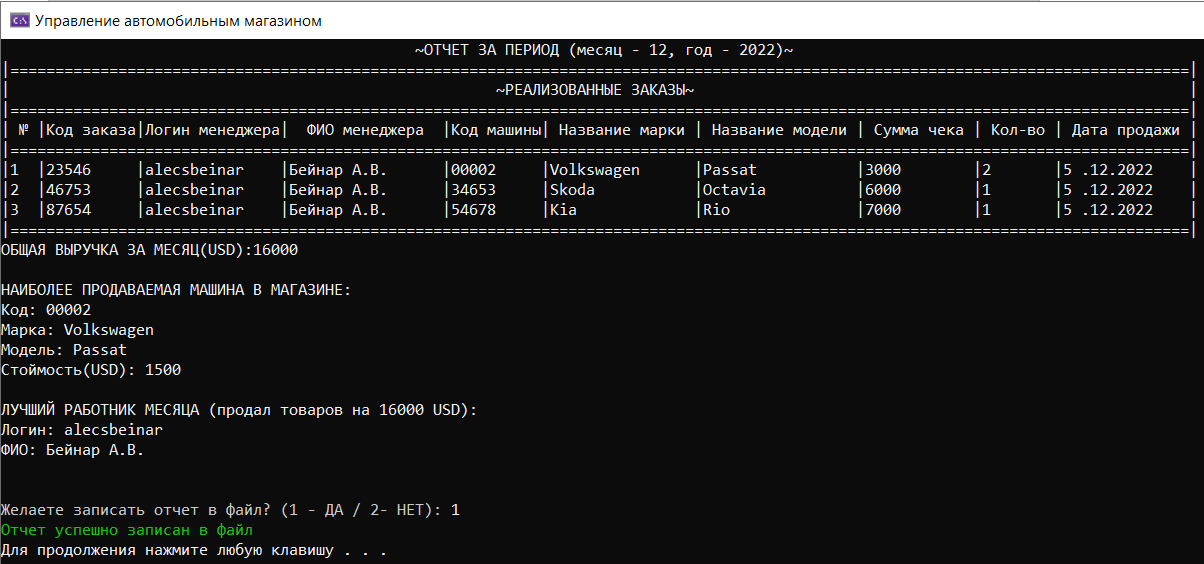


Рисунок 7.2.23 – Сформированный отчет за период

Управление учетными записями является одной из основных задач администратора, для этого необходимо в меню администратора выбрать второй пункт. На рисунке 7.2.24 показано меню работы с учетными записями.

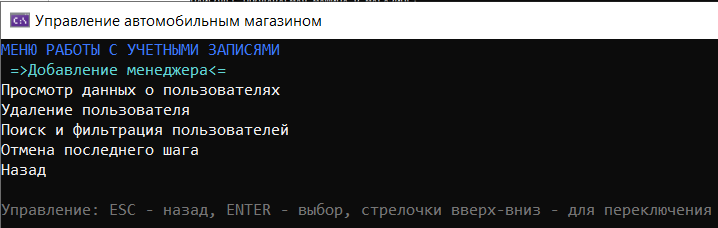


Рисунок 7.2.24 – Меню работы с учетными записями

При необходимости добавления нового менеджера нужно выбрать первый пункт, данный функционал был показан на рисунке 7.1.2

На рисунке 7.2.25 показан вывод всех учетных записей программы.

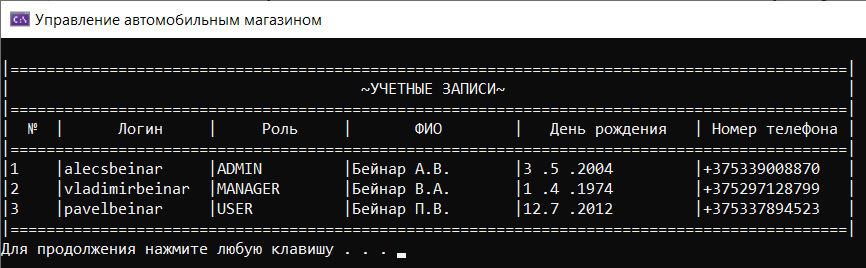


Рисунок 7.2.25 – Вывод учетных записей

Также в программе реализован функционал для удаления учетных записей из базы данных. Для безопасной эксплуатации приложения было запрещено удаление собственной учетной записи администратора. Также при необходимости удаления менеджера сначала необходимо удалить все реализованные им заказы. На рисунке 7.2.26 показана попытка удаления учетной записи.

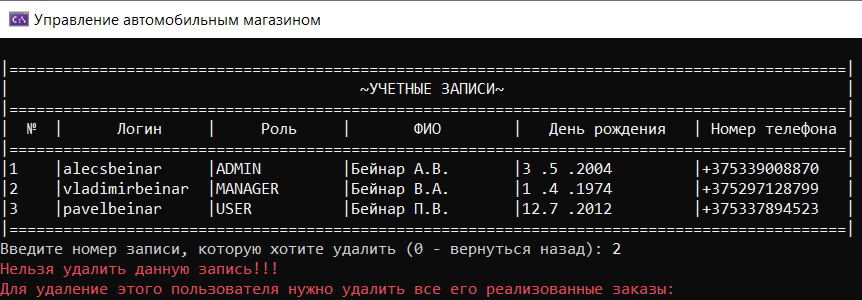


Рисунок 7.2.26 – Попытка удаления учетной записи

Также реализован функционал для поиска и фильтрации пользователей. Разработан ряд параметров, по которым можно осуществить поиск и сортировку (рис 7.2.27): поиск по статусу, поиск по фамилии, поиск по телефону и дате, а также сортировка по дате.

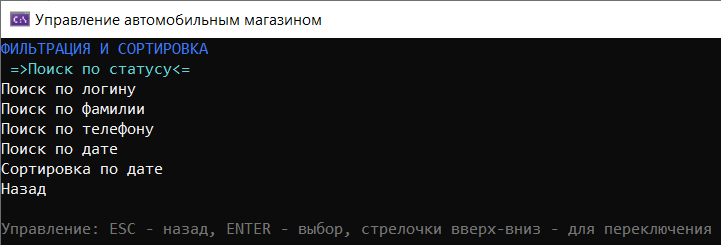


Рисунок 7.2.27 – Меню поиска и фильтрации пользователей

Для администратора предусмотрена возможность изменения данных о магазине. На рисунке 7.2.28 показано изменение адреса магазина.

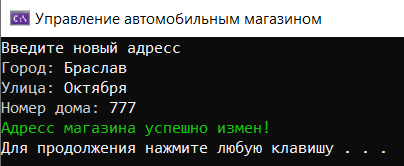


Рисунок 7.2.28 – Изменение данных о магазине

Каждый пользователь приложения может изменять данные собственной учетной записи. Для этого необходимо выбрать в главном меню пункт «Редактирование персональных данных». После этого появляется выпадающее меню с параметрами для редактирования (рис 7.2.29). На рисунке 7.2.30 показано изменение пароля собственной учетной записи.

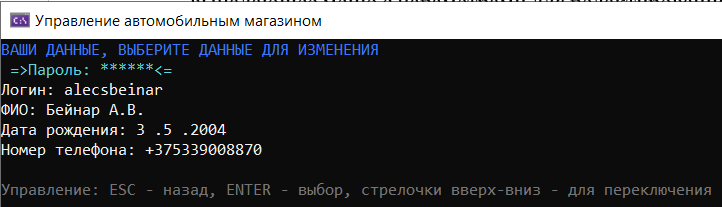


Рисунок 7.2.29 – Меню с параметрами учетной записи

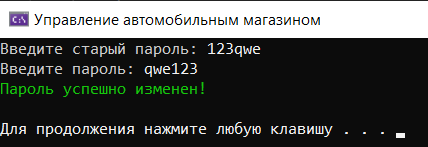


Рисунок 7.2.30 – Изменение пароля собственной учетной записи

## **7.3 Меню менеджера**

Менеджер обладает ограниченными возможностями по сравнению с администратором приложения. Он не имеет доступа к такому функционалу, как управление учетными записями и изменение данных о магазине. Менеджер автомобильного магазина может добавлять новые товары и изменять их данные, редактировать персональные данные, а также проводить заказы и создавать отчеты.

Весь функционал доступный как для менеджера, так и для администратора был подробно рассмотрен в пункте 7.2.

На рисунке 7.3.1 показано меню менеджера.

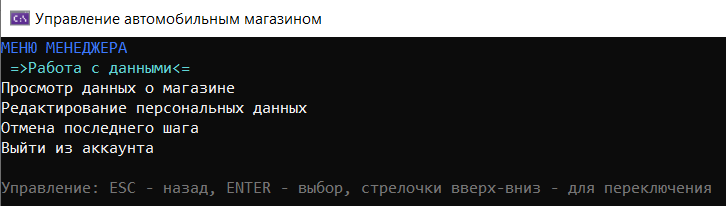


Рисунок 7.3.1 – Меню менеджера

## **7.4 Меню пользователя**

Функционал для пользователей приложением значительно урезан по сравнению с функционалом администратора или менеджера. Пользователь не имеет возможности добавлять либо изменять данные о товарах и заказах. Для него доступны такие функции, как просмотр продукции, поиск и фильтрации продукции, редактирование персональных данных, просмотр данных о магазине. Также весь функционал был рассмотрен в пункте 7.2. На рисунке 7.4.1 показано меню пользователя.

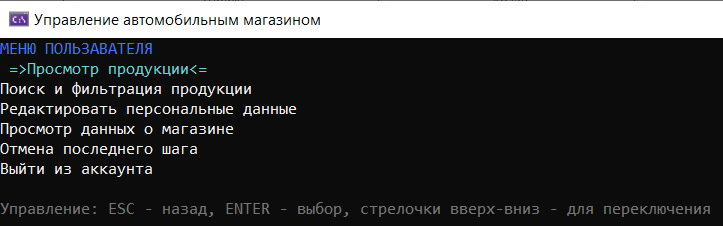


Рисунок 7.4.1 – Меню пользователя

## **7.5 Тестирование и проверка работоспособности программного средства**

Исключительные ситуации – ситуации, возникающие во время работы системы, выходящие за пределы, предусмотренные спецификацией. Для того, чтобы предотвратить такие случаи необходимо в коде программы предусмотреть защиту от неправильных действий.

В первую очередь стоит уделить внимание на неправильный ввод отрицательных чисел, выбор несуществующего пункта, отмена несовершенного шага, ошибку открытия файла и/или чтения и записи данных "в никуда", выход за размерность вектора. Также стоит учесть такие ошибки, как некорректный ввод логина, ввод некорректной даты и номера телефона. Такие моменты стоит обязательно обрабатывать для получения программы устойчивой к неправильным действиям.

На рисунке 7.5.1 представлена обработка ситуации ввода некорректной стоимости.

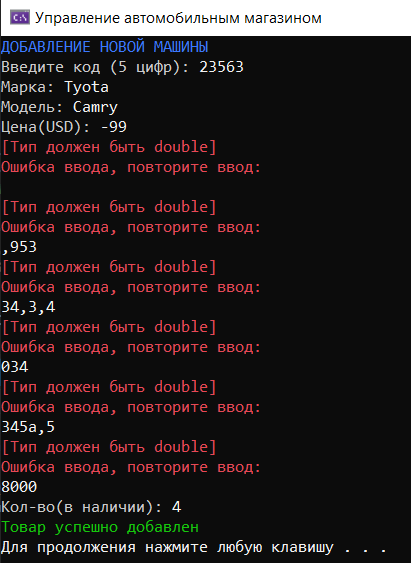


Рисунок 7.5.1 – Некорректный ввод стоимости

На рисунке 7.5.2 показана обработка ситуации некорректного ввода даты.

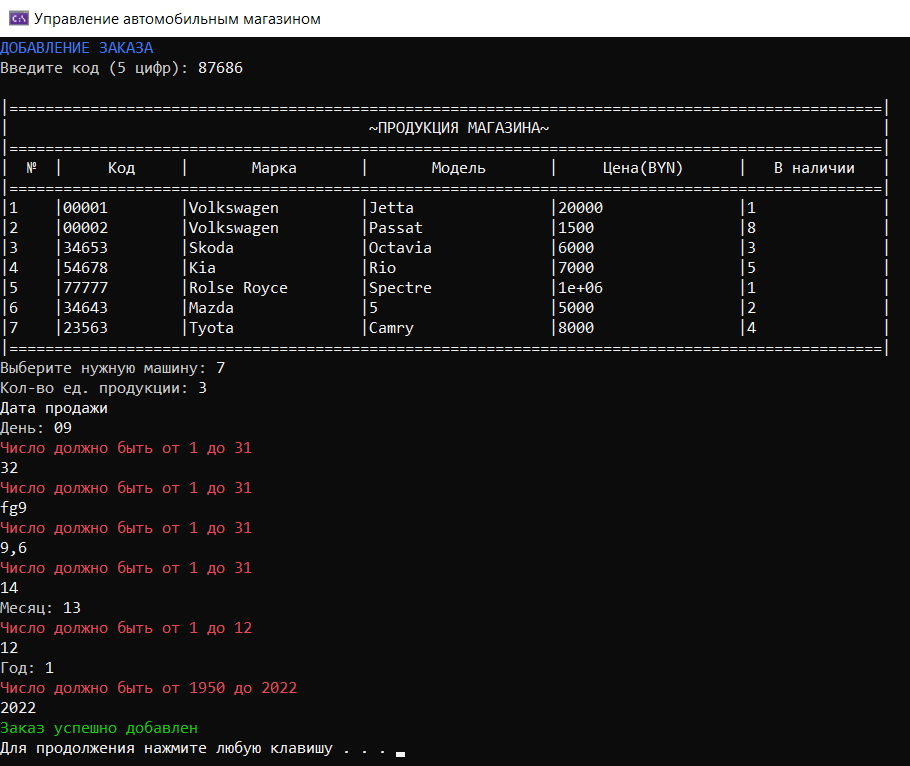


Рисунок 7.5.2 – Некорректный ввод даты

На рисунке 7.5.3 представлена обработка ситуации некорректного ввода номера телефона.

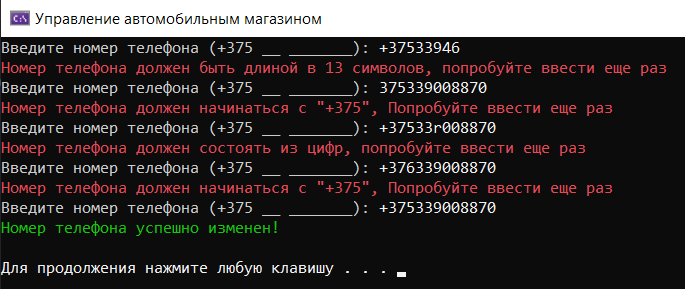


Рисунок 7.5.3 – Некорректный ввод номера телефона

Для реализации уникальности логинов пользователей в программе была реализована проверка на ввод логина. Показано на рисунке 7.5.4.

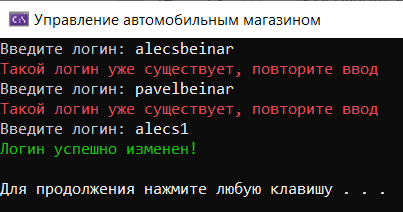


Рисунок 7.5.4 – Некорректный ввод логина

Также для повышения безопасности учетных записей были добавлены проверки на ввод пароля (рис. 7.5.5).

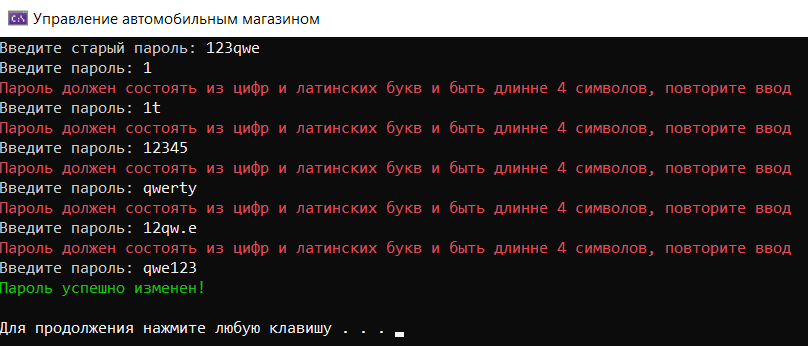


Рисунок 7.5.5 – Некорректный ввод пароля

Для реализации отмены последнего шага в программе была реализована проверка на внесение изменений. Показано на рисунке 7.5.6.

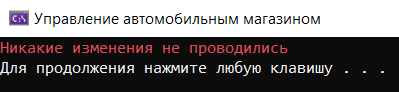


Рисунок 7.5.6 – Попытка отмены несовершенного шага

Таким образом для обеспечения стабильной работы программы, а также безопасности пользователей был разработан ряд функций обрабатывающих неправильные действия при работе с приложением.

## **Заключение**

В данной курсовой работе была разработана информационная система для автоматизации функционирования автомобильного магазина. В процессе разработки программного средства было проделано следующее:

* изучена литература по теме проекта;
* разработаны схемы алгоритмов работы программы;
* разработан интерфейс приложения;
* разработаны структуры и модули приложения;
* проведено тестирование программного продукта.

Данное программное средство обладает следующими достоинствами:

* использует мало ресурсов операционной системы;
* быстродействие программы;
* обеспечивает ввод корректных данных.

Для улучшения функционала программы можно реализовать следующее:

* более быстрые алгоритмы сортировки;
* функция сохранения данных в случае непредвиденного завершения работы программного средства.

## **Список использованных источников**

[1] Блок-схемы алгоритмов [Электронный ресурс]. – Режим доступа: <https://pro-prof.com> – Дата доступа: 05.12.2022.

[2] Бизнес-план автомобильного магазина [Электронный ресурс]. – Режим доступа: https://www.openbusiness.ru/biz/business/biznes-plan-magazina-avtomobilnykh-tovarov/ – Дата доступа: 05.12.2022.

[3] Герберт, Ш. Самоучитель C++/Ш. Герберт. Санкт-Петербург 2003г.

[4] Дейтел, Х.М. Как программировать на С++ / Х.М. Дейтел, П.Д. Дейтел; пер. с англ. – М. : Бином, 2007. – 1152 с..

[5] Страуструп, Б. Язык программирования С++ / Б. Страуструп; специальное издание. Пер. с англ. – СПб. : BHV, 2008. – 1098 с.

[6] Элджер, Дж. C++: библиотека программиста / Дж. Элджер. – СПб. : Питер, 2001. – с.

[7] Объектно-ориентированное программирование в С++/ Роберт Лафоре г. пер. с англ. – Санкт-Петербург, 2019. – 1152 с.

**Приложение А**

**Листинг кода алгоритмов, реализующих основную логику программы**

class Order

{

private:

string code;

string CarCode;

string loginManager;

Date date;

int numberOf;

public:

Order() {

code = "";

CarCode = "";

loginManager = "";

numberOf = 0;

}

void SetCode(Container<Order>& obj);

void SetCarCode(const string code) { this->CarCode = code; }

void SetLoginManager(const string login) { this->loginManager = login;}

void SetDate() { this->date.SetInfo(); }

void SetNumberOf(int max) { this->numberOf = DataProcessing::checkInterval(1, max); }

string GetCode() { return this->code; }

string GetCarCode() { return this->CarCode; }

string getLoginManager() { return this->loginManager; }

Date& GetDate() { return this->date; }

int GetNumberOf() { return this->numberOf; }

void SetInfo(const string code, const string CarCode, const string loginManager, const Date& date, const int numberOf);

friend istream& operator>>(istream& stream, Order& obj);

friend ostream& operator<<(ostream& stream, const Order& obj);

void operator=(const Order& obj) { SetInfo(obj.code, obj.CarCode, obj.loginManager, obj.date, obj.numberOf); }

bool operator<(const Order& obj);

};

class Car : public WheeledVehicle

{

private:

string brand;

string model;

string code;

double price;

int numberOf;

public:

Car() {

brand = "";

model = "";

code = "";

price = 0;

numberOf = 0;

}

void SetBrand() { getline(cin, this->brand); }

void SetModel() { getline(cin, this->model); }

void SetCode(Container<Car>& obj);

void SetPrice() { this->price = DataProcessing::checkInterval(0.0, 1000000.00); }

void SetNumberOf() { this->numberOf = DataProcessing::checkInterval(0, 1000); }

string GetBrand() { return this->brand; }

string GetModel() { return this->model; }

string GetCode() { return this->code; }

double GetPrice() { return this->price; }

int GetNumberOf() { return this->numberOf; }

void SetInfo(const string code, const string brand, const string model, const double price, const int numberOf);

void OutputInfo();

void Decrease(int number) { this->numberOf -= number; }

void Increase(int number) { this->numberOf += number; }

friend istream& operator>>(istream& stream, Car& obj);

friend ostream& operator<<(ostream& stream, const Car& obj);

void operator=(const Car& obj) { SetInfo(obj.code, obj.brand, obj.model, obj.price, obj.numberOf); }

bool operator<(const Car& obj);

};

**Приложение Б**

**Диаграмма классов**

**Приложение В**

**Алгоритм главной функции «main»**

**Приложение Г**

**Алгоритм функции добавления новой парковки «addCars»**

**Приложение Д**

**Алгоритм функции удаления парковки «delOrder»**

**ПРИЛОЖЕНИЕ Е**

*(обязательное)*

Ведомость документов