Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

Рег. № \_\_\_\_\_\_\_\_\_\_ Кафедра информационных систем и технологий

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_ Учет клиентов музыкальной студии

по дисциплине \_Название\_\_Основы конструирования программ\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_2020\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc71676381)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc71676382)

[2.ТРЕБОВАНИЕ К ПРОГРАММЕ 5](#_Toc71676383)

[3.КОНСТРУИРОВАНИЕ ПРОГРАММЫ 6](#_Toc71676384)

[4.ОПИСАНИЕ ИНТЕРФЕЙСА 7](#_Toc71676385)

[5.ВЫБОР СПОСОБА ОРГАНИЗАЦИИ ДАННЫХ 9](#_Toc71676386)

[6.РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ МЕТОДОВ ПРОГРАММЫ 10](#_Toc71676387)

[7.ПРОГРАММНАЯ ЧАСТЬ 11](#_Toc71676388)

[ЗАКЛЮЧЕНИЕ 13](#_Toc71676389)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 14](#_Toc71676390)

[ПРИЛОЖЕНИЕ 15](#_Toc71676391)

# **ВВЕДЕНИЕ**

Была поставлена задача разработать программный модуль «Учёт автозапчастей», предназначенный для использования работниками магазина автозапчастей. В базе содержатся сведения о запчастях (Название, цена, пополнение). При возникновения новой запчасти её данные заноситься в базу.

Программа создана для систематизации склада автозапчастей, быстрого просмотра/изменения и добавления новых запчастей.

Цель разработки обеспечить работы магазина автозапчастей быстро и максимально эффективно, удовлетворять запросы работников магазина, а также должен быть возможность пополнять базу, для оптимизации процессов магазина. И неотъемлемой частью данной программы должна высокая оптимизация, чтобы пользователь мог запустить даже на самом слабом устройстве.

Задачи данного программного продукта: быстро и гибко редактироваться со стороны владельцев музыкальной студии. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было владельцу приятно пользоваться программой. Так же требуется предусмотреть возможность добавления новых функций в программе.

Для разработки программы был использован текстовый файл и язык программирования C++ разработанный в среде Visual Studio компаний Microsoft.

Visual Studio-линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой Windows Forms, а также веб-сайты, веб-приложения, веб-службы, как в родном, так и в управляемом кодах для всех платформ, поддерживающих Windows, Windows Mobile, Windows Ce, .Net Framework, Xbox, Windows Phone .Net Compact Framework и Silverling.

# **1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Была поставлена задача разработать программный модуль «Учёт автозапчастей», предназначенный для использования работниками магазина автозапчастей. В базе содержатся сведения о запчастях (Название, цена, пополнение). При возникновения новой запчасти её данные заноситься в базу.

Количество автомобилей на дорогах стремительно растёт. С увеличением количества автомобилей и растёт спрос на их обслуживания, что привело к росту количества магазинов автозапчастей. Поэтому разрабатываемая программа должна выдержать весьма серьёзную конкуренцию.

Функции программного продукта: содержатся сведения о запчастях (Название, цена, пополнение) должны иметь понятные данные и возможность обрабатывать их (добавлять, просматривать, изменять).

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не только просматривать запчасти, но и производить различные взаимодействия с самими запчастями.

# **2.ТРЕБОВАНИЕ К ПРОГРАММЕ**

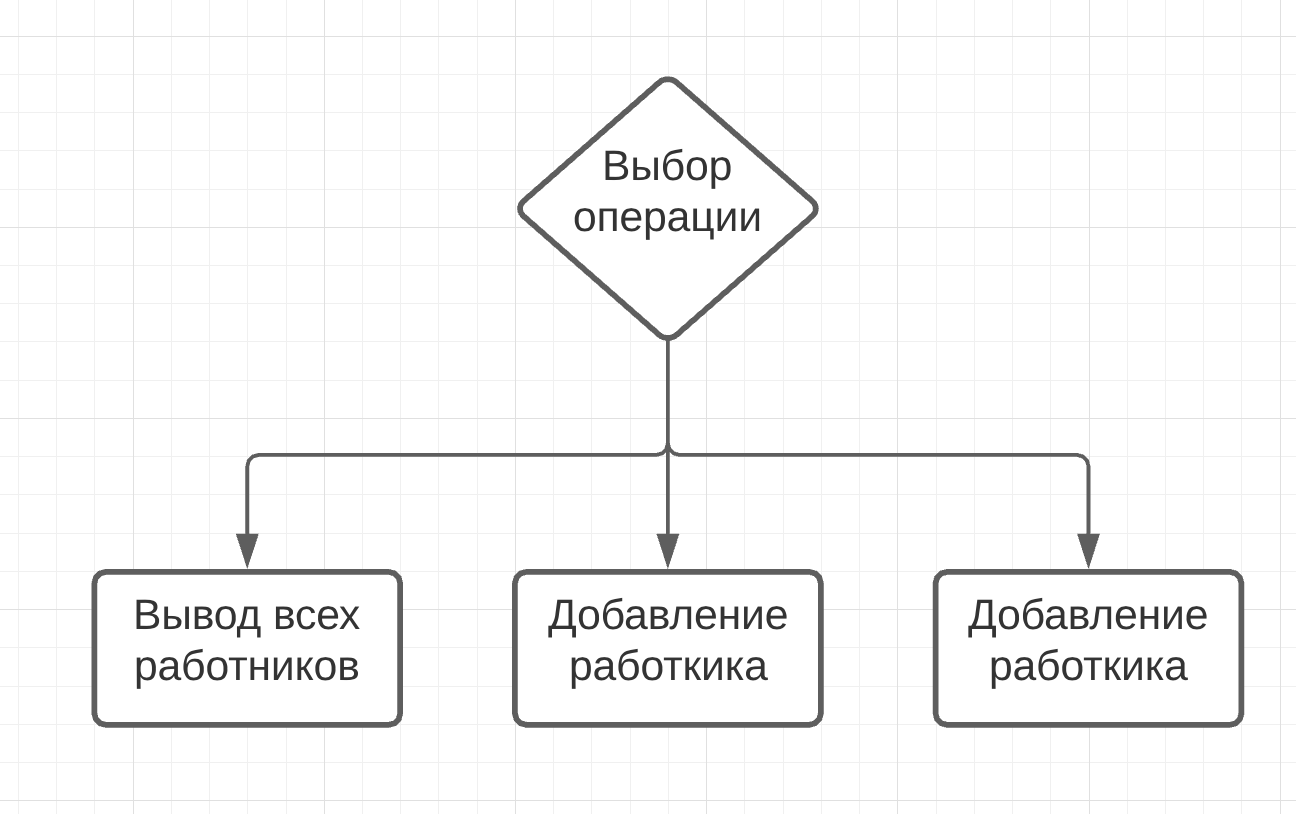
Исходя из задач, описанных выше главными требованиями к языку, на котором будет написана программы — это тонкая настройка и возможность использованием минимум вычислительных ресурсов добиваться максимум результата.

Для выполнения данных задач очень подходит такой язык программирования как с++. Так как его из-за того, что он является весьма низкоуровневым языком программирования, а значит он больше других приближен к машинному коду, из-за этого с++ дает возможность оптимизировать проект так что даже на очень старых устройствах он будет работать.

Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 – Windows Server 2016. Система призвана стать единой для разных устройств, таких как персональные компьютеры, планшеты, смартфоны, консоли Xbox One и пр. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами. Windows 10 поставляется в качестве услуги с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий Windows 7, Windows 8.1 и Windows Phone 8.1. Среди значимых нововведений – голосовая помощница Кортана, возможность создания и переключения нескольких рабочих столов и др. Windows 10 – последняя «коробочная» версия Windows, все последующие версии будут распространяться исключительно в цифровом виде.

Среда разработки Microsoft Visual Studio. Вид приложения – консольное. Программа построена на базе объектно-ориентированного программирования. Способ организации данных поля классов. Способ хранения данных – текстовый фалй/xml. Каждая логическая завершенная данных подзадача программы реализована в виде методов. К защите курсовой предоставляется: консольное приложение и пояснительная записка.

# **3.КОНСТРУИРОВАНИЕ ПРОГРАММЫ**



В текущей главе представлено описание алгоритма, который имеет начало алгоритма и конец, внутри алгоритма имеются циклы и иные операций.

Первым пунктом идёт выбор операции, за счет введение цифры от 1 до 3 и последующей за вводом проверки.

Следующий шаг — это работа непосредственно с программой в зависимости от выбранных данных. Добавление данных, вывод данных,изменение данных.

# **4.ОПИСАНИЕ ИНТЕРФЕЙСА**

При старте программы происходит выбор желаемой операции (рис.4.1)

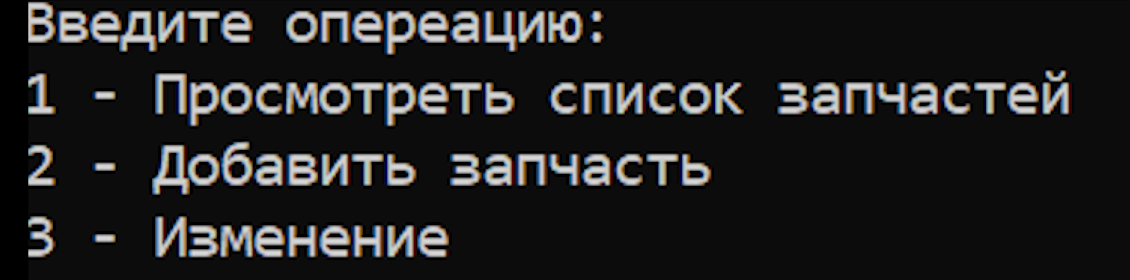


Рис. 4.1

При выборе первого пункта на консоль выводиться все данные запчастей (рис. 4.2).

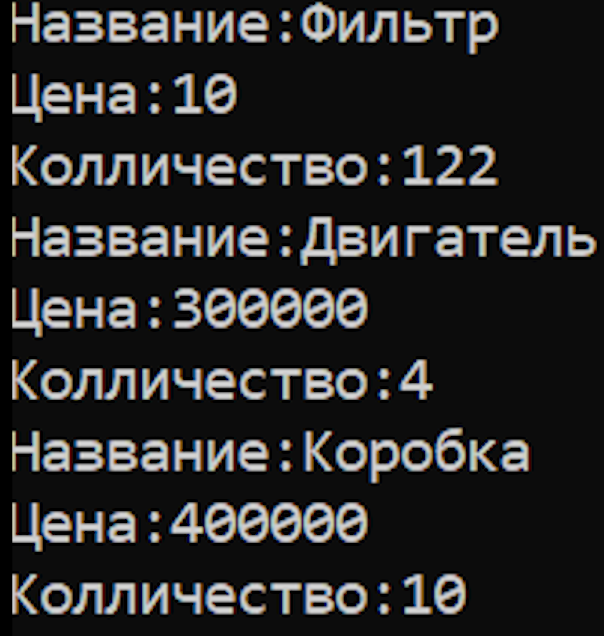


Рис.4.2

При выборе второго пункта поочерёдно вводить данные для изменения (рис 4.3).

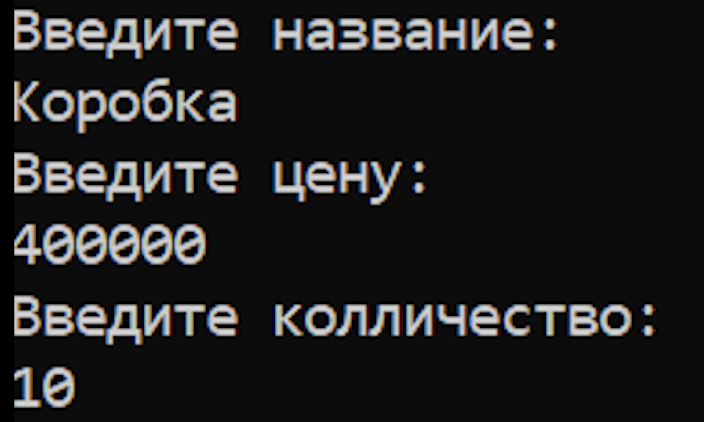


Рис. 4.3

При выборе второго пункта пользователю необходимо вначале выбрать изменяемый пункт. А далее необходимо поочерёдно вводить данные (рис 4.4).

(рис 4)

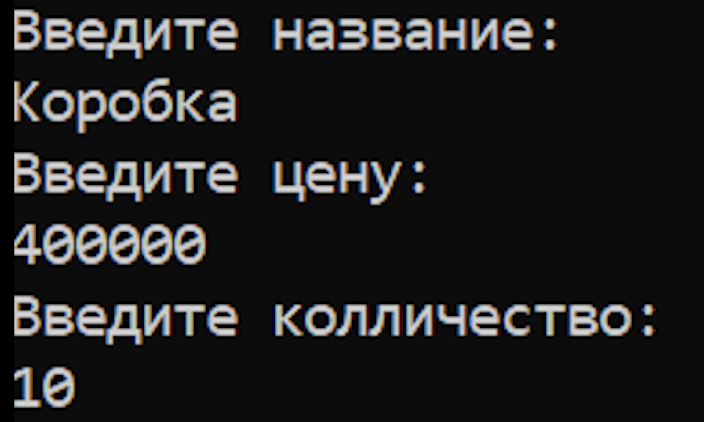


Рис.4

# 

# **5.ВЫБОР СПОСОБА ОРГАНИЗАЦИИ ДАННЫХ**

Программа состоит одного класса.

Класс **Spares** (рис 5):

Переменные: string Name, string Prise, string Size; – публичные переменные, которые служат в качестве информации о запчастях.

Методы:

AddSpares ()– данный метод является частью класса Spares. Он необходим для заполнения или редактирование данных запчастей. Работа данного метода заключается в заполнении полей класса данными с консоли (Name, Prise, Size). Данный метод не возвращает значений.

Print() – данный метод необходим для вывода данных с консоли. Данный метод выводит на консоль следующие переменные: Name, Prise, Size. Данный метод не возвращает значения.

В данном классе не разлизано конструкторов.



Рис 5

# **6.РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ МЕТОДОВ ПРОГРАММЫ**

**void** AddSpares (рис 6.1)– данный метод является частью класса Spares. Он необходим для заполнения или редактирование данных запчастей. Работа данного метода заключается в заполнении полей класса данными с консоли (Name, Prise, Size). Данный метод не возвращает значений.

**void** Print (рис 6.2) – данный метод необходим для вывода данных с консоли. Данный метод выводит на консоль следующие переменные: Name, Prise, Size. Данный метод не возвращает значения.

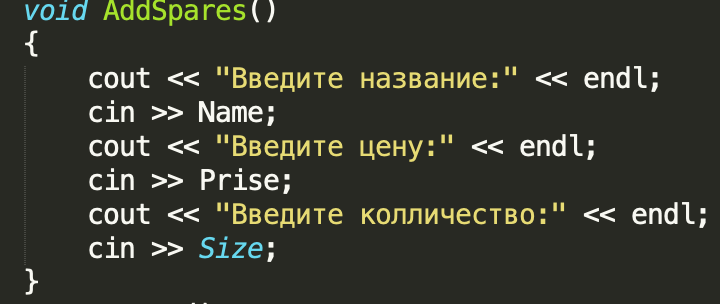


Рис 6.1

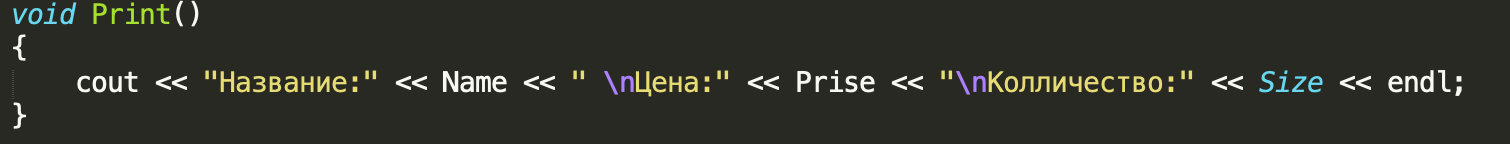
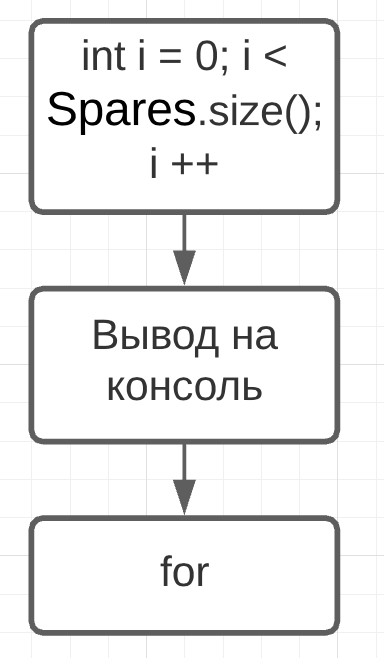


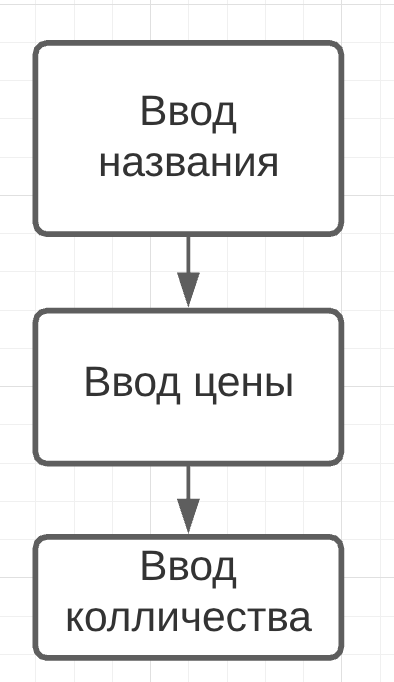
Рис 6.2

# **7.ПРОГРАММНАЯ ЧАСТЬ**

void Print()



Void add Spares()



Вначале происходит выбор операции:

1. Просмотреть список запчастей
2. Добавить запасть
3. Изменение

Далее при помощи конструкции switch происходит переход к выбранной операции.

1: Циклом for происходит итерация по списку. При каждой итерации происходит вызов метода Print. Который выводит в консоль данные каждого cсотрудника (Название, цена, количество).

2: Вначале создаётся новая(пустая) запчасть. Далее вызывается метод класса Person, AddSpares.

Его суть заключается в том, что пользователь поочерёдно вводит данные с консоли. Далее происходит добавление клиента в список.

3: Вначале происходит выбор изменяемого элемента. Далее происходит проверка на существование данного элемента. Если элемент отсутствует, то на консоль выводиться надпись: “Данного элемента не существует”.

В обратном же случае происходит вызов метода AddSpares для изменения элемента. Его суть заключается в том, что пользователь поочерёдно вводит данные с консоли. Далее происходит добавление клиента в список.

# **ЗАКЛЮЧЕНИЕ**

Исходя из поставленных целей и задач было сконструировано, разработана и протестирован программный модуль «Учёт автозапчастей», предназначенный для использования работниками магазина автозапчастей. В базе содержатся сведения о запчастях (Название, цена, пополнение). При возникновения новой запчасти её данные заноситься в базу.

1)Быстро и гибко редактироваться со стороны сотрудников магазина.

2)Программный продукт имеет красивый, понятный и интуитивный интерфейс и владельцу приятно пользоваться программой.

3)Так же предусмотрено возможность добавления новых функций в программе.

Цель была выполнены, картотека имеет следующие преимущества:

1)Работа магазина автозапчастей происходит быстро и максимально эффективно.

2)Удовлетворяет запросы магазина автозапчастей.

2)Также реализована возможность пополнять базу, для оптимизации процессов магазина автозапчастей.

3)Реализована высокая оптимизация, что-то привело к возможности пользователем запускать приложение даже на самом слабом устройстве.

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не только просматривать запчамти, но и производить различные взаимодействия с самими запчастями.

Таким образом программа работает без сбоев и все задачи были выполнены.

# **СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирование / Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. – СПб: Питер, 2015. – 368с.
2. Коплиен Дж. Мультипарадигменное проектирование для С++ / Коплиен Дж. – Питер, 2005.
3. Обзор обновлений и новых функций Windows 10 [Электронный ресурс]. – Microsoft, 2020. – Режим доступа: https://www.microsoft.com/ru-ru/windows/features. – Дата доступа: 22.05.2020.
4. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
5. Стивен Прата Язык программирования C++ / Стивен Прата – Вильямс, 2012.
6. Страуструп Б. Программирование. Принципы и практика с использованием C++ / Страуструп Б. – 2-е изд. – Вильямс, 2016.
7. Model-View-Presenter [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Presenter. – Дата доступа: 22.04.2020.
8. SFML [Электронный ресурс]. – SFML, 2020. – Режим доступа: https://www.sfml-dev.org. – Дата доступа: 22.04.2020.
9. SFML [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/SFML. – Дата доступа: 22.04.2020.

# **ПРИЛОЖЕНИЕ**

#include <iostream>

#include <string>

#include <deque>

#include <Windows.h>

using namespace std;

class Person

{

public:

string Name;

string Surname;

string MiddleName;

string MaritalStatus;

string children;

string position;

string AcademicDegree;

void AddPerson()

{

cout << "Введите имя:" << endl;

cin >> Name;

cout << "Введите фамилия:" << endl;

cin >> Surname;

cout << "Введите отсчество:" << endl;

cin >> MiddleName;

cout << "Симейное положение:" << endl;

cin >> MaritalStatus;

cout << "Колличество детей:" << endl;

cin >> children;

cout << "Должность:" << endl;

cin >> position;

cout << "Учёная степень:" << endl;

cin >> AcademicDegree;

}

void Print()

{

cout << "Введите имя:" << Name << " \nВведите фамилия:" << Surname << "\nВведите отсчество:" << MiddleName << "\nСимейное положение:" << MaritalStatus << "\nКолличество детей:" << children << "\nДолжность:" << position << "\nУчёная степень:" << AcademicDegree << "\n"<< endl;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Russian");

deque<Person> Persons;

while (true)

{

int Number;

bool selection = false;

Person person;

cout << "Введите опереацию:" << endl;

cout << "1 - Просмотреть всех работников" << endl;

cout << "2 - Добавить работника" << endl;

cout << "3 - Изменение" << endl;

cin >> Number;

int number;

switch (Number)

{

case 1:

for (int i = 0; i < Persons.size(); i++)

{

Persons[i].Print();

}

break;

case 2:

person.AddPerson();

Persons.push\_back(person);

break;

case 3:

cout<< "Введите номер изменяемого эллемента:" <<endl;

cout << "Колличество эллентов: " << Persons.size() <<endl;

cin >> number;

for (int i = 0; i < Persons.size(); i++)

{

if (i = number-1)

{

Persons[i].AddPerson();

selection = true;

}

if (selection)

{

cout<< "Данного номера не существует" <<endl;

}

}

break;

default:

cout << "Неверно выедена операция";

break;

}

}

}