Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

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

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_\_\_\_Название\_\_\_Разработать программный модуль «Картотека недвижимости»\_\_\_\_\_\_\_\_

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

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Урбанович Владислав Сергеевич (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_1920\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc70793473)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc70793474)

[**1.1. Общие положения 4**](#_Toc70793475)

[**1.2. Постановка задачи 4**](#_Toc70793476)

[**1.3. Спецификация входных и выходных данных 4**](#_Toc70793477)

[**1.4. Выбор средства 5**](#_Toc70793478)

[**1.5.Описывается требуемое оборудование и программное обеспечение 6**](#_Toc70793479)

[2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ 7](#_Toc70793480)

[**2.1. Алгоритм решения 7**](#_Toc70793481)

[**2.2. Описание программы 7**](#_Toc70793482)

[**2.3. Результаты работы программы 9**](#_Toc70793483)

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 10](#_Toc70793484)

[ЗАКЛЮЧЕНИЕ 10](#_Toc70793485)

[ПРИЛОЖЕНИЕ 11](#_Toc70793486)

# ВВЕДЕНИЕ

Была поставлена задача разработать программный продукт «Информационные процессы подразделения, отвечающего за сбор заявок на материалы», предназначенный для использования сотруднимами подразделения. В программном продукте необходимо реализовать учёт заявок на материалы, сбор заявок, подготовка отчетов, контроль выполнения заявок. Материалы имеют следующие показатели: материал, количество, дата поставки.

Программа создана для сбора заявок, консолидации материалов заявок, подготовка отчетов, контроль выполнения заявок.

Цель разработки обеспечить работу программного продукта быстро и максимально эффективно, удовлетворять запросы подразделения, а также должен быть возможность пополнять базу данных, для работы с материалами. Основная же задача стоит в высокой оптимизация, чтобы работники подразделения могли запустить даже на самом слабом устройстве.

Задачи данного программного подукта: быстро и гибко редактироваться со стороны владельцев компании. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было клиенту 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

## 1.1. Общие положения

С ростом любой компании расширяются и подразделения компании, что приводит к необходимости оптимизации процессов определённых операций. Для обеспечение подобного рода залач необходимо использовать технологии, которые будут обеспечивать возможность подобной работы.

Была поставлена задача разработать программный продукт «Информационные процессы подразделения, отвечающего за сбор заявок на материалы», предназначенный для использования сотруднимами подразделения. В программном продукте необходимо реализовать учёт заявок на материалы, сбор заявок, подготовка отчетов, контроль выполнения заявок. Материалы имеют следующие показатели: материал, количество, дата поставки.

Структурные подразделения (например, железной̆ дороги) каким-то образом планируют потребность в материалах. Эти потребности оформляются в виде документа «Заявка»

Далее заявки концентрируются в подразделении, составляются суммарные поквартальные заявки (на последнее число квартала) и утверждаются у начальника службы. Затем, утверждённый̆ план поставок передается в договорной̆ отдел. Из договорного отдела потом поступает документ, о тех позициях консолидированной̆ заявки, которые получены (на склад). Инженер должен на основе этого документа отметить, какие заявки удовлетворены, для каких сроки сорваны.

Данный программный продукт выгодно отличается от других подобных тем что пользуясь им, пользователь может не только структурированно хранить данные об материалов, но и производить различные взаимодействия с ними.

## 1.2. Постановка задачи

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.

## 1.3. Спецификация входных и выходных данных

В данном программном продукте используются следующие типы данных типа данных: sting, double, int bool. А также присутствует пользовательский тип данных Request.

boll – используется только внутри программы, для проверки, подсчёта, иннерваций и так далее. Пользователь на данный этого типа влиять и пользоваться не может.

String – используется в двух случаях. Либо для наименования данных (присвоение данных материалу). Во втором же случае данный тип данных используется для вывода всех данных на экран.

Int и double – используется для подсчёта внутри программы и для наименования данных.

Request – представляет собой заявку.

## 1.4. Выбор средства

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.

Среда разработки Microsoft Visual Studio. Вид приложения – консольное. Программа построена на базе объектно-ориентированного программирования. Способ организации данных поля классов. Способ хранения данных – динамические массивы. Каждая логическая завершенная данных подзадача программы реализована в виде методов. Построение программного кода соответствует с <<C++ Code Convention>>. К защите курсовой предоставляется: консольное приложение и пояснительная записка

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.

## 1.5.Описывается требуемое оборудование и программное обеспечение

Для полноценного функционирования разработанного программного приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор Intel Core i3;
* оперативная память DDR3 512 МБ или больше;
* минимально свободного места на жёстком диске 512 МБ;
* мышь;
* операционная система Windows 7/8/10.

# 2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ

## 2.1. Алгоритм решения

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

Вначале программы происходит инициализация данных материально ответственных лиц и основанных средств. Далее происходит бесконечный цикд while. Он необходим для реализации повторного выбора операции.

Далее в программе происходит выбор операции по нажатию необходимой цифры.

1 - Просмотр базы данных от лица начальника службы.

2 - Просмотр базы данных от лица складского менеджера.

1. - Просмотр базы данных от лица инженера.
2. – Поиск заявки по всей базе.
3. – Удаление заявки.
4. - Добавление заявки.

0 - Выход из программы.

Пункты №1-3 – происходит вывод всей базы данных. Отличие заключается только в выводе разных столбцов.

4 - Производиться поиск определённой заявки по названию и даёт возможность её редактировать.

5 - Удалении определённой заявки по номеру.

6 - Добавление заявки путём добавление нового экземпляра класса Request в вектор.

0 - Производиться окончание работы программы (выход из программы).

Представление алгоритма в виде блок-схемы(рис 2,2)

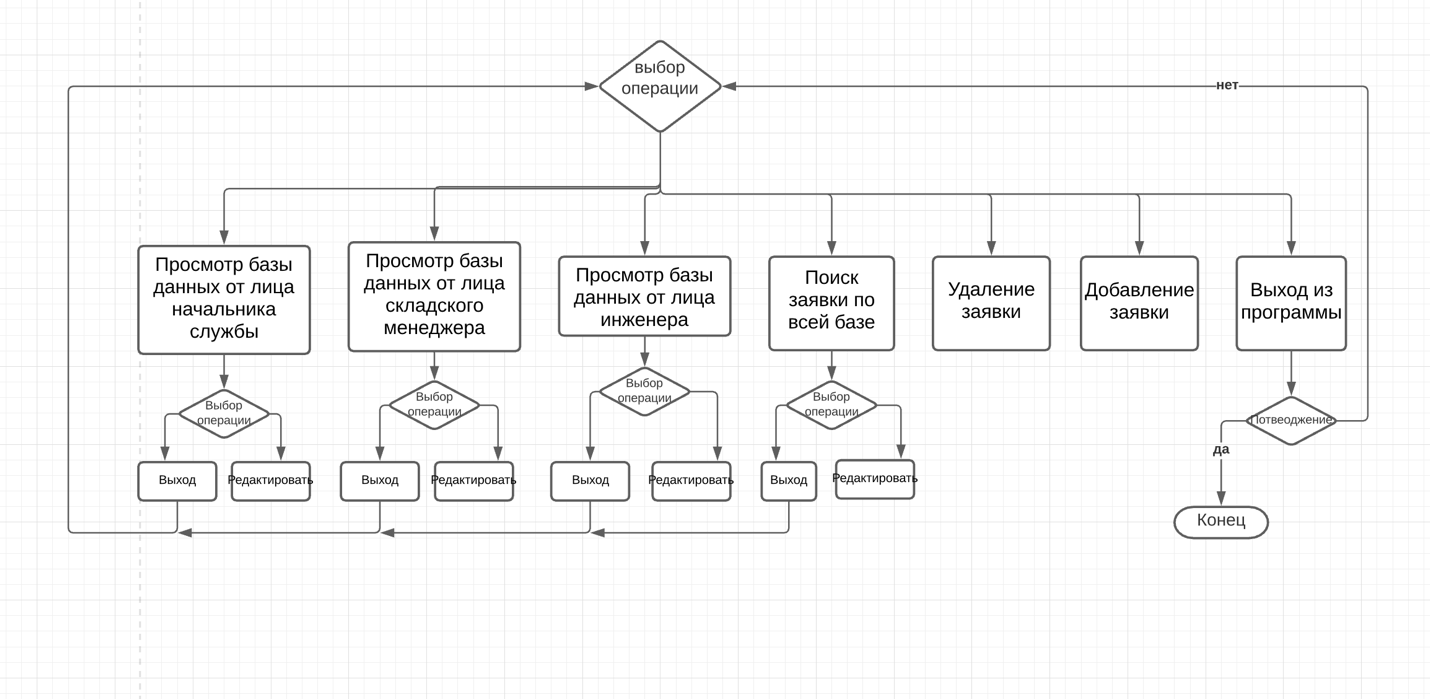


Рис 2.2

## 2.2. Описание программы

В данной программы используются следующие библиотеки:

**#include <iostream>** - заголовочный файл с классами, функциями и переменными для организации ввода-вывода в языке программирования C++. Он включён в стандартную библиотеку C++. Название образовано от Input/Output Stream («поток ввода-вывода»). В языке C++ и его предшественнике, языке программирования Си, нет встроенной поддержки ввода-вывода, вместо этого используется библиотека функций. iostream управляет вводом-выводом, как и stdio.h в Си. iostream использует объекты cin, cout, cerr и clog для передачи информации и из стандартных потоков ввода, вывода, ошибок без буферизации и ошибок с буферизацией соответственно. Являясь частью стандартной библиотеки C++, эти объекты также являются частью стандартного пространства имён — std.

**#include <string>** - класс с методами и переменными для организации работы со строками в языке программирования C++. Он включён в стандартную библиотеку C++. Название образовано от имени строчного типа данных (англ. string; с англ. — «строка»). В языке C++ и его предшественнике, языке программирования Си, нет встроенной поддержки строкового типа данных, вместо этого используется массив символов. string управляет строками, как и string.h в Си. string использует единственный объект string для организации работы со строками. Являясь частью стандартной библиотеки C++, эти объекты также являются частью стандартного пространства имён — std. #include <random> - данная библиотека необходима для реализации работы с генератором случайных числах.

**#include <fstream>** — заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл. Для манипуляции с данными файлов используются объекты, называемые потоками («stream»).

Функции, включенные в данный файл, позволяют производить чтение из файлов как побайтово, так и блоками, и записывать так же. В комплект включены все необходимые функции для управления последовательностью доступа к данным файлов, а также множество вспомогательных функций.

**#include <vector>** - Шаблон vector расположен в заголовочном файле <vector>. Как и все стандартные компоненты, он расположен в пространстве имён std. Данный интерфейс эмулирует работу стандартного массива C (например, быстрый произвольный доступ к элементам), а также некоторые дополнительные возможности, вроде автоматического изменения размера вектора при вставке или удалении элементов.

**#include <conio.h>** - заголовочный файл, используемый в старых компиляторах, работающих в операционных системах MS-DOS, для создания текстового интерфейса пользователя. Тем не менее, он не является частью языка программирования Си, стандартной библиотеки языка Си, ISO C или требуемой стандартом POSIX**.**

**#include <Windows.h>** - является Windows-конкретный файл заголовков для языка C программирования, который содержит заявления для всех функций в Windows API, все общие макросы, которые используются программистами окон, и все типы данных, используемых различными функциями и подсистем.

В данном программном продукте используется Request.

Струтура **Request** (рис. 2.2.1):



рис. 2.2.1

Данный класс состоит из следующих переменных: NameThing (название основного средства), Prise (цена товара), PriseResult (цена с учётом износа).

Переменные:

string **requestName** – название заявки.

int **requestSize** – размер заявки

string **requestSizeType** – единицы измерения материала.

string **dateOfDelivery** – дата доставки материала.

string **subdivisionName** – название подразделения который оформляет запрос на материал.

Так же в данной структуре реализован конструктор по умолчанию и конструктор копирования

## 2.3. Результаты работы программы

Результат работы программы представлен структурой Request, а именно следующими переменами:

string **requestName** – название заявки.

int **requestSize** – размер заявки

string **requestSizeType** – единицы измерения материала.

string **dateOfDelivery** – дата доставки материала.

string **subdivisionName** – название подразделения который оформляет запрос на материал.

# 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Так как была поставлена задача разработать консольное приложение, а значит при разработке интерфейса возникают все проблемы для консольных приложений. Этот факт необходимо учитывать при разработке интерфейса.

При запуске программы на консоль выводиться выбор операции. Для осуществления выбора необходимо ввести соответствующую цифру в консоль(рис 3.1).

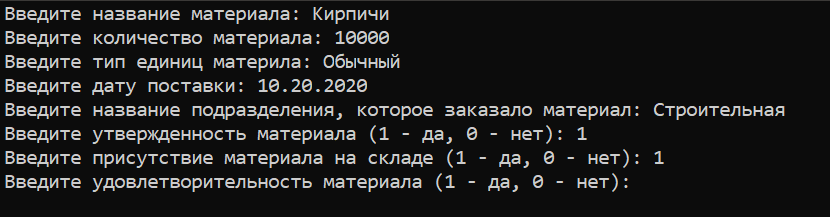


Рис 3.1

При выборе пункта № 1 на консоль выводиться вся база данных от лица начальника службы (рис 3.2)

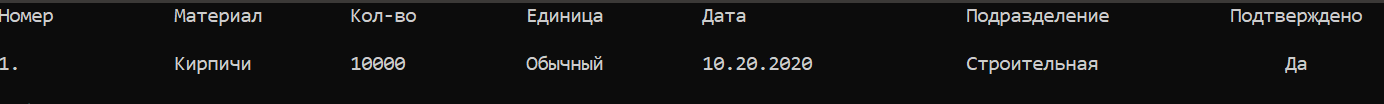


Рис 3.2

При выборе пункта № 2 на консоль выводиться вся база данных от лица складского менеджера (рис 3.3).

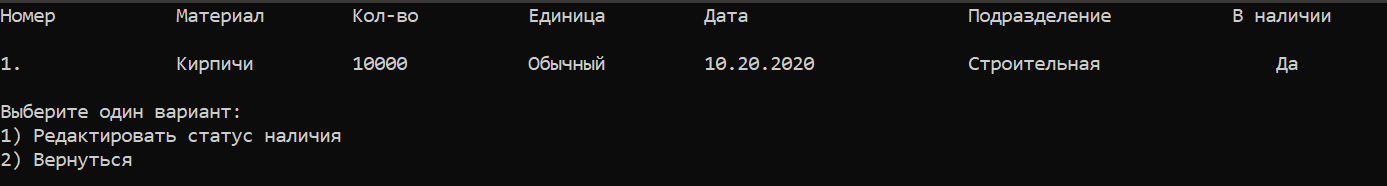


Рис 3.3

При выборе пункта № 3 на консоль выводиться вся база данных от лица инженера (рис 3.4).

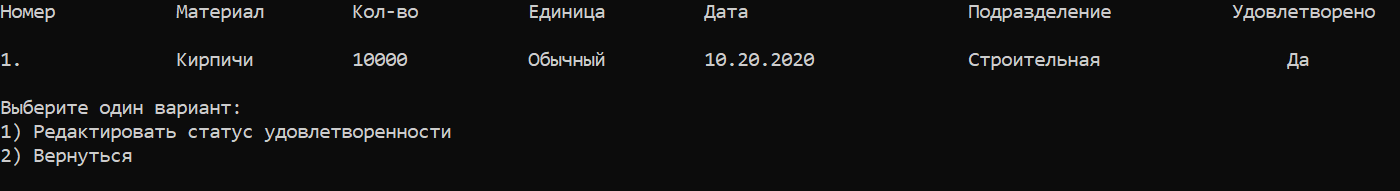


Рис 3.4

При выборе пункта № 4 происходит поиск заявки происходит путём ввода номера заявки. Если заявка существует, то данных заявки выводиться на консоль.

При выборе пункта № 5 происходит удаление заявки происходит путём ввода номера заявки. Если заявка существует, то заявка удаляется.

При выборе пункта № 5 происходит добавление заявки путём поочерёдного ввода данных на консоль (рис 3.5).

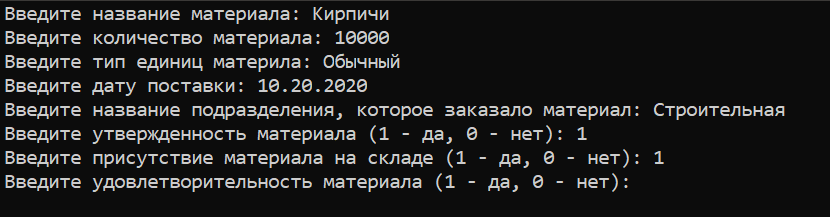


Рис 3.5

# ЗАКЛЮЧЕНИЕ

Исходя из поставленных целей и задач было сконструировано, разработана и протестирован программный модуль ««Информационные процессы подразделения, отвечающего за сбор заявок на материалы», предназначенный для использования сотруднимами подразделения.Были реализованы задачи:

1. Сбор данных.
2. Консолидация материальных заявок.
3. Подготовка отчетов.
4. Контроль выполнения заявок.

Цель была выполнены, программный продукт имеет следующие преимущества:

1. Работа программного продукта происходит быстро и максимально эффективно. Программа удовлетворяет запросам пользователя.
2. Имеет возможность пополнять базу данных, для расширения заявок.
3. Так же программы имеет высокую оптимизацию. Из-за этого программа работает весьма быстра и запускается даже на самом слабом устройстве.

Так же данный программный продукт выгодно отличается от других представленных на рынке, возможность не только хранить данные заявок, но и производить различные взаимодействие с базой данных.

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

# СПИСОК ЛИТЕРАТУРЫ

1. Коплиен Дж. Мультипарадигменное проектирование для С++ / Коплиен Дж. – Питер, 2005.
2. 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.
3. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
4. Стивен Прата Язык программирования C++ / Стивен Прата – Вильямс, 2012.
5. Страуструп Б. Программирование. Принципы и практика с использованием C++ / Страуструп Б. – 2-е изд. – Вильямс, 2016.
6. Model-View-Presenter [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Presenter. – Дата доступа: 22.04.2020.
7. SFML [Электронный ресурс]. – SFML, 2020. – Режим доступа: https://www.sfml-dev.org. – Дата доступа: 22.04.2020.

# ПРИЛОЖЕНИЕ

#include <iostream>

#include <string>

#include <random>

#include <ctime>

#include <deque>

using namespace std;

class Main\_Thing

{

string NameThing;

double Prise;

double PriseResult;

Main\_Thing() {}

string Print()

{

return "Название товара: " + NameThing + " Цена: " + to\_string(Prise) + " Остаточная цена: " + to\_string(PriseResult);

}

void Reduce()

{

PriseResult =PriseResult - 30;

}

void Add()

{

cout << "Введите название товара:";

cin >> NameThing;

cout << "Введите цену :";

cin >> Prise;

PriseResult = Prise;

}

};

class MOL

{

public:

string Name;

string Surname;

deque<Main\_Thing> Resources;

string Print() {

string result;

result += "Name: " + Name + " Surname: " + Surname;

if (Resources.empty())

{

result += "\nЗа данным материально ответственным лицом не чего не закреплено";

}

else

{

for (auto item: Resources)

result += "\n" + item.Print();

}

return result;

}

void SetName(string value)

{

Name = value;

}

void SetSurname(string value)

{

Surname = value;

}

};

int main()

{

setlocale(LC\_ALL, "Russian");

deque<MOL> Persons;

MOL person1;

person1.SetName("Peregrine");

person1.SetSurname("Dean");

Persons.push\_back(person1);

MOL person2;

person2.SetName("Charles");

person2.SetSurname("Simmons");

Persons.push\_back(person2);

MOL person3;

person3.SetName("Branden");

person3.SetSurname("Jordan");

Persons.push\_back(person3);

Main\_Thing accounting;

deque<Main\_Thing> client;

accounting.NameThing = "Компьютер";

accounting.Prise = 1000.0;

accounting.PriseResult = 1000.0;

client.push\_back(accounting);

accounting.NameThing = "телефон";

accounting.Prise = 400;

accounting.PriseResult = 400;

client.push\_back(accounting);

accounting.NameThing = "Шифратор";

accounting.Prise = 6000;

accounting.PriseResult = 6000;

client.push\_back(accounting);

while (true)

{

cout << " Выберите желаему операцию:" << endl;

cout << " 1 - просмотреть весь список" << endl;

cout << " 2 - перейти на следующий день" << endl;

cout << " 3 - добавить средство" << endl;

cout << " 4 - Просмотреть ответственных лиц" << endl;

cout << " 5 - выдать средство" << endl;

int selection;

cin >> selection;

double value;

MOL temp;

switch (selection)

{

case 1:

for (auto item : client)

{

cout << item.Print() << endl;

}

break;

case 2:

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

{

client[i].Reduce();

}

break;

case 3:

accounting.Add();

client.push\_back(accounting);

break;

case 4:

for (MOL item : )

{

cout << item.Print() << endl;

}

break;

case 5:

cout << "Выберите индекс лица получившего средство" <<endl;

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

int number;

cin >> number;

int i ;

i = 0;

for (MOL item : Persons)

{

i++;

if (i == number)

{

cout<< "Выберите индекс лица получвемого средство" <<endl;

cout << "Колличество средств: " << client.size() << endl;

cin >> number;

int i1 = 0;

for (auto itemm : client)

{

i1++;

if (i1 == number)

{

item.Resources.push\_back(itemm);

temp = item;

Persons[i-1] = temp;

break;

}

}

break;

}

}

break;

default:

cout << "Неверно введены данные";

break;

}

}

}