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](#_Toc70868457)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc70868458)

[**1.1. Общие положения 4**](#_Toc70868459)

[**1.2. Постановка задачи 4**](#_Toc70868460)

[**1.3. Спецификация входных и выходных данных 4**](#_Toc70868461)

[**1.4. Выбор средства 5**](#_Toc70868462)

[**1.5.Описывается требуемое оборудование и программное обеспечение 6**](#_Toc70868463)

[2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ 7](#_Toc70868464)

[**2.1. Алгоритм решения 7**](#_Toc70868465)

[**2.2. Описание программы 8**](#_Toc70868466)

[**2.3. Результаты работы программы 11**](#_Toc70868467)

[3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 12](#_Toc70868468)

[ЗАКЛЮЧЕНИЕ 13](#_Toc70868469)

[СПИСОК ЛИТЕРАТУРЫ 14](#_Toc70868470)

[ПРИЛОЖЕНИЕ 15](#_Toc70868471)

# ВВЕДЕНИЕ

Была поставлена задача разработать программный продукт «Информационные процессы подразделения дефектоскопии рельсов», предназначенный для использования работниками подразделения дефектоскопии.

Программа создана для автоматизации процесса ремонта и профилактике ЖД-путей, а так же для удобного слежения за состоянием ЖД-путей. Что позволит сильно сократить время переговоров.

Цель разработки обеспечить работу программного продукта быстро и максимально эффективно, удовлетворять работников, а также должен быть возможность пополнять базу данных, для учёта состояния путей. Основная же задача стоит в высокой оптимизация, чтобы пользователь мог запустить даже на самом слабом устройстве.

Задачи данного программного продукта: 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

Последнее десятилетие ЖД-перевозки стремительно развиваются. Увеличивается количества путей, а также увеличивается частота пользования путями. При развитии данной отросли возникает и ряд проблем, такие как: усложнение процесса контроля за состоянием железнодорожных путей и последующим их ремонтов. Именно для решения данных задачи был реализован данный программный продукт.

Была поставлена задача разработать программный продукт «Информационные процессы подразделения дефектоскопии рельсов», предназначенный для использования работниками подразделения дефектоскопии.

Существуют инструкции, которые указывают периодичность контроля участков рельсов, в зависимости от классификации участков пути. Каждый участок пути имеет свой паспорт, в котором указаны его начальные параметры при построении участка, загруженность (тоннаж состава, проходящий по участку), результаты предыдущих контролей. У подразделения имеется набор средств дефектоскопии (переносные и вагоны - дефектоскопы). Это средства подлежат (по инструкциям) периодическому контролю. Существует несколько уровней контроля: профилактический и капитальный, когда средство увозят в специальную лабораторию. Необходимо выполнять учет контроля средств дефектоскопии.

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не хранить данные путей, но и просматривать историю ремонтов .

## 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 присутствуют пользовательские типы данных: Developer, Team-Developer, Customer, Contract.

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 данным название удобного пользователя). Во втором же случае данный тип данных используется для вывода всех данных на экран.

Int – используется для отсаживание состояние железнодорожного пути и его дальнейшее

В данном проекте реализован единственный класс Path. Вокруг него и строиться весь функционал данного программного продукта. В себе он имеет 4 переменных типа int и одну переменную типа srting.

Так же в данном классе реализован конструктор присваивания и есть метод.

## 

## 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

Алгоритм данного программного продукта строиться на условном операторе switch. Данный оператор имеет ряд плюсов и минусов. При его использовании значительно ухудшается читаемость кода, но зато он крайне удобен с точки зрения построения алгоритма.

Вначале идёт инициализация коллекции pathes. Она хранит все железнодорожные пути. Данная коллекция является основной в данном программном продукте.

Далее эти создаться две вспомогательные коллекции overhaul и redecorating. Эти две коллекции необходимы для реализации разделения ремонта железнодорожных путей на 2 вида: косметический и копательный. Так как их ремонт производиться отдельно друг от друга, а значит необходима и в программном продукте их разделить.

Весь алгоритм описываемый далее помещён в бесконечный цикл, чтобы иметь возможность пользоваться приложением неоднократно, без его перезапуска.

Далее пользователем программного продукта производиться выбор желаемой операции путём ввода числового значения (рис 2.1.1)

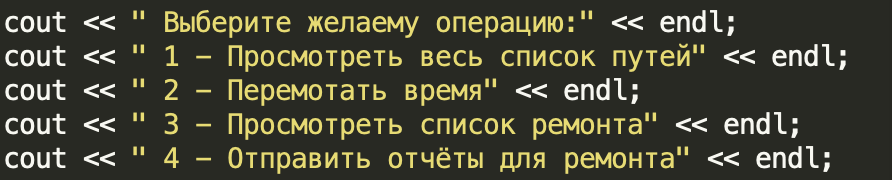


рис 2.1.1

1. Происходит перебор коллекции paths и при каждой инерции происходит вызов метода Print (рис 2.1.2).

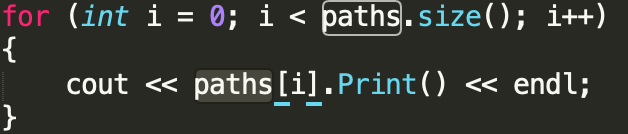


рис 2.1.2

1. На данном участке кода реализован механизм искусственного износа железнодорожных путей. Как как типов поверхности 3 то и изнашиваются они по-разному. Для реализации вариативности износа железнодорожных путей используется условный оператор switch (рис 2.1.3)

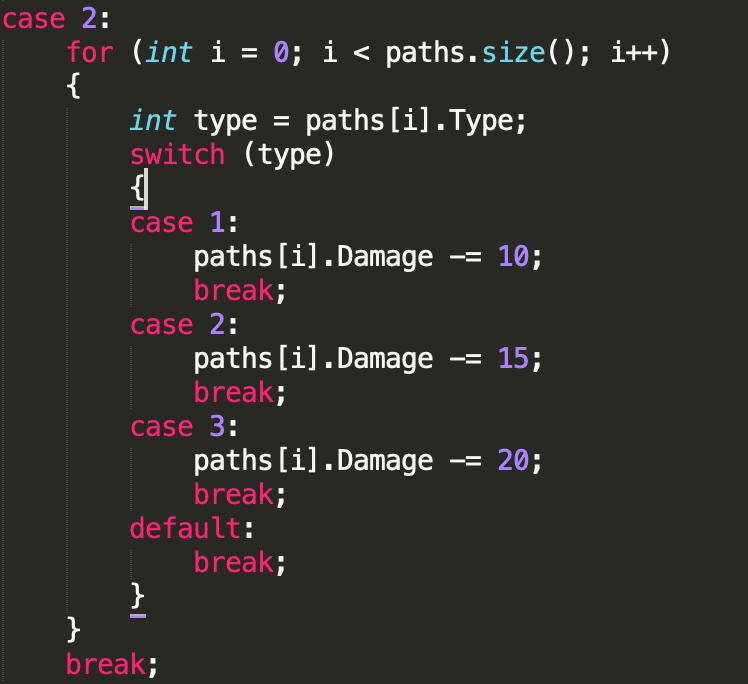


Рис 2.1.3

1. В данном пункте реализован механизм определения какой ремонт необходим (капитальный или косметический). Так как каждый тип поверхности, согласно инструкции, требует обслуживание если показатели его износа падают ниже определённого значения, то необходимо производить ремонт. У каждого типа путей эти показатели разные, из-за этого есть необходимость работать с конкретным типом. Для выбора типом и дальнейших действий с ним потребовалось воспользоваться оператором Switch.
2. В четвертом пункте реализован алгоритм починки железнодорожных путей. За счёт перебора коллекции железнодорожных путей и при каждой итерации присвоение им значение целостности равное 100 (рис 2.1.4).

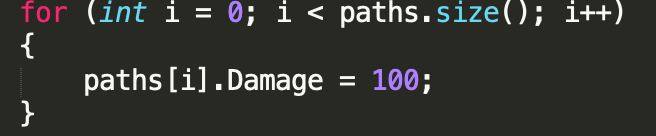


рис 2.1.4

## 2.2. Описание программы

В данной программы используются следующие библиотеки:

#include <iostream> - основная библиотека, которая реализует основные возможность языка С++.

#include <string> - библиотека для возможности работы с типом данных string.

#include <random> - данная библиотека необходима для реализации работы с генератором случайных числах.

#include <ctime> - данная библиотека необходима для реализации работы с датой и времени.

#include <deque> - данная библиотека необходима для реализации работы с двухсторонней очередью.

В данном проекте реализован единственный класс Patch**.** Вокруг него и строиться весь функционал данного программного продукта. В себе он имеет 4 переменных типа int и одну переменную типа srting.(рис 2.2.1)

## 

Рис 2.2.1

В данном классе реализован конструктор присваивание. Он содержит 5 переменных. С помощью его реализуется алгоритм присвоение данных к конкретному объекта класса.

Метод string **Print:**

Данный метод возвращает строковое значение, в котором содержаться все данные, содержащиеся в объекте класса, для последующего вывода данных на консоль.

## 2.3. Результаты работы программы

Результат работы данной программы являться либо вывод ранее введённых данных на консоль. Либо вывод тех же данных только в изменённом виде.

# 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Вначале пользователю необходимо выбрать пункт, который будет выполняться. Выбор происходит за счёт ввода пользователем в консоль значения от 1 до 4(рис 3.1).

Значения 2 и 4 – не чего не возвращают на консоль, так как являются обслуживающими алгоритмами.

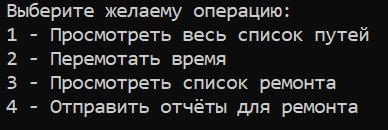


Рис 3.1

При выборе пункта №1 на консоль выводиться все железнодорожные пути и соответствующая информация о них (рис 3.2).

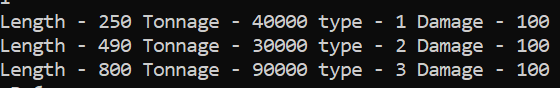


Рис 3.2

При выборе пункта №1 на консоль выводиться все железнодорожные пути которые нуждаться в ремонте, капитальном или косметическом.

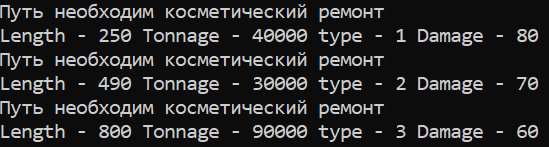


Рис 3.3

# ЗАКЛЮЧЕНИЕ

Была поставлена задача разработать программный продукт «Информационные процессы подразделения дефектоскопии рельсов», предназначенный для использования работниками подразделения дефектоскопии были реализованы задачи:

1. Создание классификации путей.
2. Каждый путь имеет свой паспорт.
3. Реализована два варианта ремонта (косметический и капитальный)
4. Учёт контроля ремонта

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 ассортимента товаров.
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, возможностью не только покупать, но и продавать недвижимость, ещё она снабжена интуитивным интерфейс.

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 <list>

#include <deque>

using namespace std;

class Path

{

public:

int Length;

int Tonnage;

int Type;

int Damage;

string Name;

Path(int length, int tonnage, int type, int damage, string name)

{

Length = length;

Tonnage = tonnage;

Type = type;

Damage = damage;

Name = name;

}

string Print()

{

return "Length - " + to\_string(Length) + " Tonnage - " + to\_string(Tonnage) + " type - " + to\_string(Type) + " Damage - " + to\_string(Damage);

}

void TimeNext() {}

};

int main()

{

setlocale(LC\_ALL, "Russian");

deque<Path> paths;

Path value1(250 ,40000, 1, 100, "Минск-Орша");

paths.push\_back(value1);

Path value2(490, 30000, 2, 100, "Минск-Гомель");

paths.push\_back(value2);

Path value3(800, 90000, 3, 100, "Минск-Москва");

paths.push\_back(value3);

deque<Path> overhaul;

deque<Path> redecorating;

while (true)

{

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

cout << " 1 - Просмотреть весь список путей" << endl;

cout << " 2 - Перемотать время" << endl;

cout << " 3 - Просмотреть список ремонта" << endl;

cout << " 4 - Отправить отчёты для ремонта" << endl;

int selection;

cin >> selection;

switch (selection)

{

case 1:

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

{

cout << paths[i].Print() << endl;

}

break;

case 2:

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

{

int type = paths[i].Type;

switch (type)

{

case 1:

paths[i].Damage -= 10;

break;

case 2:

paths[i].Damage -= 15;

break;

case 3:

paths[i].Damage -= 20;

break;

default:

break;

}

}

break;

case 3:

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

{

int type = paths[i].Type;

switch (type)

{

case 1:

if (paths[i].Damage > 40 && paths[i].Damage < 90)

{

cout<< "Путь необходим косметический ремонт\n" + paths[i].Print() <<endl;

redecorating.push\_back(paths[i]);

}

else if (paths[i].Damage < 40 )

{

cout << "Путь необходим капитальный ремонт\n" + paths[i].Print() << endl;

overhaul.push\_back(paths[i]);

}

break;

case 2:

if (paths[i].Damage > 30 && paths[i].Damage < 90)

{

cout << "Путь необходим косметический ремонт\n" + paths[i].Print() << endl;

redecorating.push\_back(paths[i]);

}

if (paths[i].Damage <= 30)

{

cout << "Путь необходим капитальный ремонт\n" + paths[i].Print() << endl;

overhaul.push\_back(paths[i]);

}

break;

case 3:

if (paths[i].Damage > 20 && paths[i].Damage < 90)

{

cout << "Путь необходим косметический ремонт\n" + paths[i].Print() << endl;

redecorating.push\_back(paths[i]);

}

if (paths[i].Damage <= 20)

{

cout << "Путь необходим капитальный ремонт\n" + paths[i].Print() << endl;

overhaul.push\_back(paths[i]);

}

break;

default:

cout << "Неверно введены данные";

break;

}

}

break;

case 4:

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

{

paths[i].Damage = 100;

}

break;

default:

cout << "Неверно введены данные";

break;

}

}

}