**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №3**

**по дисциплине «Алгоритмы и структуры данных»**

**Вариант 3**

Студент гр. 9309 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Серов А.В.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тутуева А.В.

Санкт-Петербург

2021

Оглавление

[Постановка задачи 56](#_Toc74135793)

[Описание реализуемого классов и методов. 56](#_Toc74135794)

[Результаты 56](#_Toc74135795)

[UnitTest3.cpp 57](#_Toc74135796)

[Вывод 58](#_Toc74135797)

# Постановка задачи

 Дан список возможных авиарейсов в текстовом файле в формате:

Город отправления 1; Город прибытия 1; цена прямого перелета 1; цена обратного перелета 1

Город отправления 2; Город прибытия 2; цена перелета 2; цена обратного перелета 1

…

Город отправления N; Город прибытия N; цена перелета N; цена обратного перелета N

 В случае, если нет прямого или обратного рейса, его цена будет указана как N/A (not available)

Задание*:* найти наиболее эффективный по стоимости перелет из города ***i*** в город ***j*** С помощью алгоритма Флойда-Уоршелла и матрицы смежности.

# Описание реализуемого классов и методов.

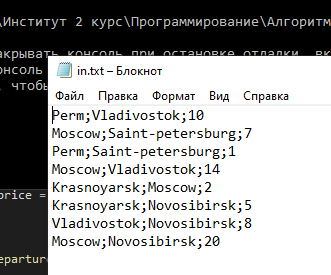
Class Flights – класс, который хранит все методы обработки авиарейсов.

Методы:

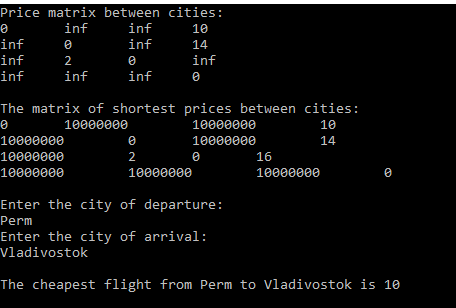
1. Method Departure – метод считывает города отправления. (O(log n))
2. Method Arrival – метод считывает города прибытия. (O(log n))
3. Method Prices– метод считывает цены перелетов. (O(log n))
4. Method start – метод открывает файл для чтения и создает двумерный массив структур для заполнения её полей. (O(n))
5. Method working – заполнение матрицы, а так же применение алгоритма Флойда-Уоршелла, Вывод результата после применения алгоритма. (O(logN^3))

# Результаты

Ввод:



Вывод:



# UnitTest3.cpp

#include "pch.h"

#include "CppUnitTest.h"

#include "../ALG\_4\_3/Func.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest43

{

TEST\_CLASS(UnitTest43)

{

public:

TEST\_METHOD(data\_set)

{

string a = "";

string expected = "Could not open file";

ifstream in;

if (!in.is\_open()) a = "Could not open file";

Assert::AreEqual(expected, a);

}

TEST\_METHOD(dataprice)

{

int msg=0; string a,b;

int expected = 50;

ifstream in;

Flights T;

msg = T.start3(a,b,msg);

Assert::AreEqual(expected, msg);

}

TEST\_METHOD(datadep)

{

string a;

string expected = "MOSCOW";

Flights T;

a = T.start2(a);

Assert::AreEqual(expected, a);

}

};

}

# Вывод

В ходе выполнения данной лабораторной работы был изучен алгоритма Флойда-Уоршелла и применен на практике.