Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 12

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Файлы»

Выполнил: ст. гр. ТУУ-211

Тарасенков А.Д.

Вариант № 16

20.08.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2024 г.

1. **Цель работы**

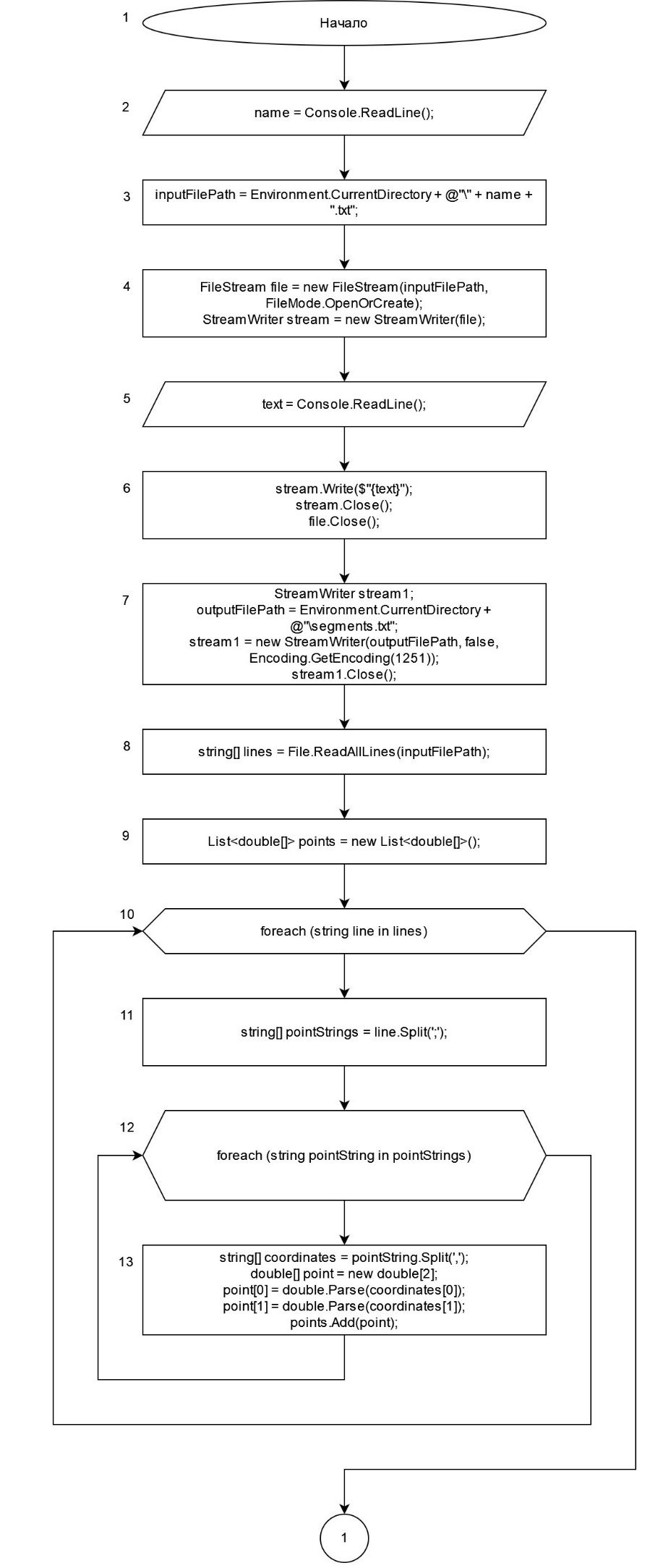
В среде Microsoft Visual Studio на языке Visual C# в консольном режиме составить программное обеспечение для решения типовых задач программирования по тематике «Файлы» («Files»)

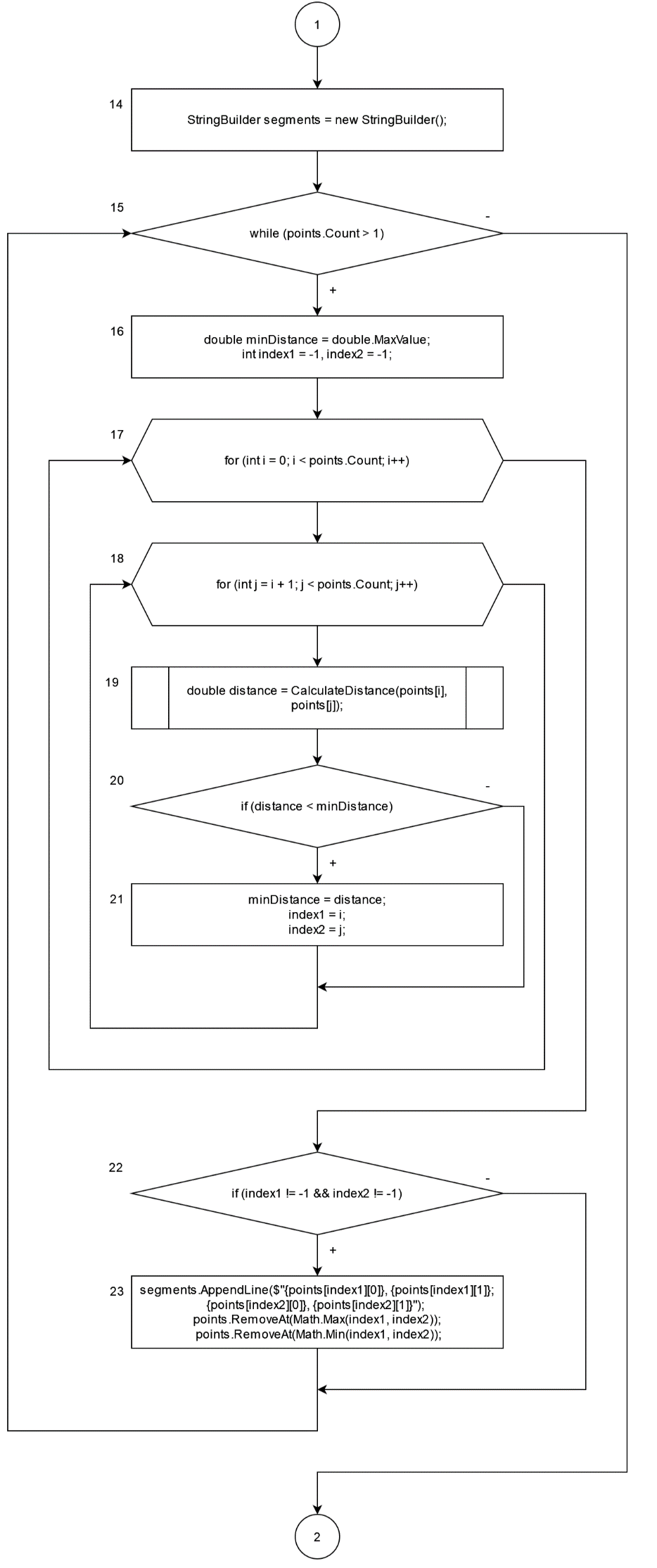
1. **Формулировка задачи**

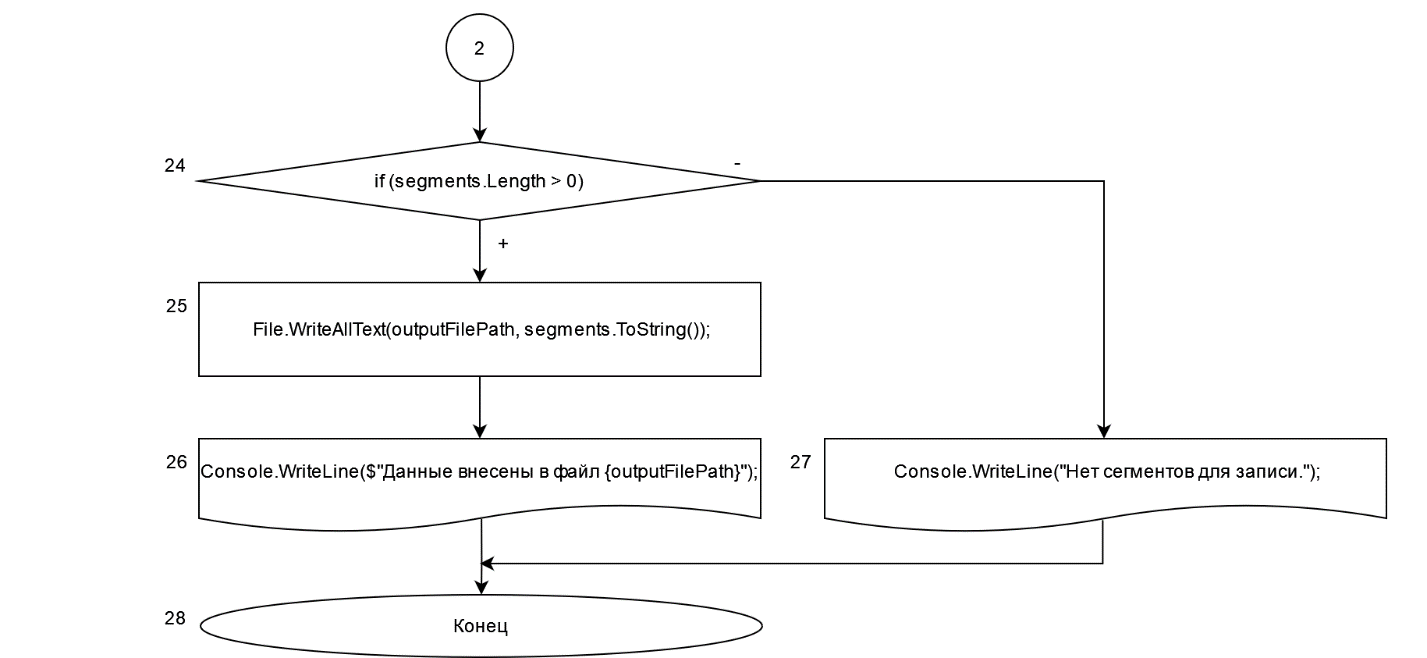
**Вариант №16**

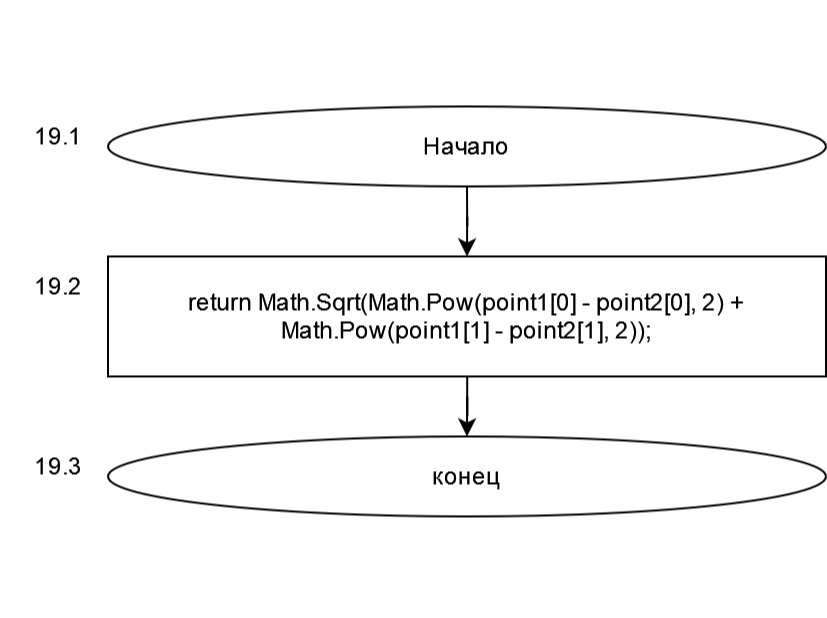
В интегрированной среде разработки (IDE) Microsoft Visual Studio составить учебную консольную программу. Убедиться, что IDE создала решение (solution) с классической структурой каталогов и файлов программного продукта, разрабатываемого на языке Visual C# (csproj). Подготовить исходный текстовый файл «dots.txt», содержащий координаты точек, разделённых между собой «;». Координаты перечислены через «,». Точки произвольные и не принадлежат никакой функциональной зависимости. Файл разместить по пути «../bin/Debug/» решения (solution). Считать исходный файл и записать точки в новый текстовый файл «segments.txt» построчно и попарно в порядке увеличения расстояний между парой точек, без каких-либо разделителей. Разделитель координат сохранить тем же. Задан алгоритм поиска: фиксируется первая точка в наборе; перебираются все последующие точки; для получающихся пар из зафиксированной и каждой последующей вычисляется длина отрезка; при обнаружении подходящей пары, дающей минимум длины отрезка, координаты записываются в файл; обе точки исключаются из дальнейшего рассмотрения; алгоритм повторяется. Алгоритм повторяется до тех пор, пока в наборе не останется ни одной точки или точка, к которой уже невозможно подобрать пару.

1. **Блок-схема алгоритма**







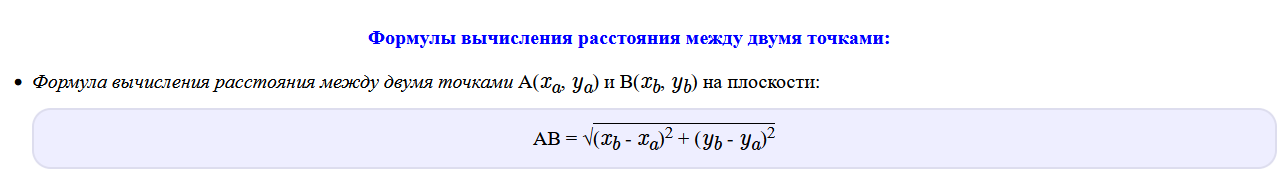


1. **Подбор тестовых примеров**

Возьмём 4 случайные точки с координатами:

1. -100,-100; (A)
2. -50,-50; (B)
3. 10,10; (C)
4. 40,40; (D)

И найдем расстояние между ними о формуле:

****

Расстояния между точками будут:

(Расчёты проведены с помощью онлайн калькулятора)

AB= 70.7

AC= 155.5

AD= 197.9

BC= 84.8

BD= 127.2

CD= 42.4

Отрезок CD – минимальный по длине. Соответственно, точки С (10,10) и D(40,40) должны быть первые в файле segments.txt

1. **Листинг (код) программы**

using System;

using System.Collections.Generic;

using System.IO;

using System.Text;

class Program

{

static void Main(string[] args)

{

string inputFilePath, outputFilePath, name, text;

Console.Write("Введите имя создаваемого файла с координатами: ");

name = Console.ReadLine();

inputFilePath = Environment.CurrentDirectory + @"\" + name + ".txt";

FileStream file = new FileStream(inputFilePath, FileMode.OpenOrCreate);

StreamWriter stream = new StreamWriter(file);

Console.Write("Введите точки: ");

text = Console.ReadLine();

stream.Write($"{text}");

stream.Close();

file.Close();

StreamWriter stream1;

outputFilePath = Environment.CurrentDirectory + @"\segments.txt";

stream1 = new StreamWriter(outputFilePath, false, Encoding.GetEncoding(1251));

stream1.Close();

string[] lines = File.ReadAllLines(inputFilePath);

List<double[]> points = new List<double[]>();

foreach (string line in lines)

{

string[] pointStrings = line.Split(';');

foreach (string pointString in pointStrings)

{

string[] coordinates = pointString.Split(',');

double[] point = new double[2];

point[0] = double.Parse(coordinates[0]);

point[1] = double.Parse(coordinates[1]);

points.Add(point);

}

}

StringBuilder segments = new StringBuilder();

while (points.Count > 1)

{

double minDistance = double.MaxValue;

int index1 = -1, index2 = -1;

for (int i = 0; i < points.Count; i++)

{

for (int j = i + 1; j < points.Count; j++)

{

double distance = CalculateDistance(points[i], points[j]);

if (distance < minDistance)

{

minDistance = distance;

index1 = i;

index2 = j;

}

}

}

if (index1 != -1 && index2 != -1)

{

segments.AppendLine($"{points[index1][0]}, {points[index1][1]}; {points[index2][0]}, {points[index2][1]}");

points.RemoveAt(Math.Max(index1, index2));

points.RemoveAt(Math.Min(index1, index2));

}

}

if (segments.Length > 0)

{

File.WriteAllText(outputFilePath, segments.ToString());

Console.WriteLine($"Данные внесены в файл {outputFilePath}");

Console.ReadKey(true);

}

else

{

Console.WriteLine("Нет сегментов для записи.");

Console.ReadKey(true);

}

}

static double CalculateDistance(double[] point1, double[] point2)

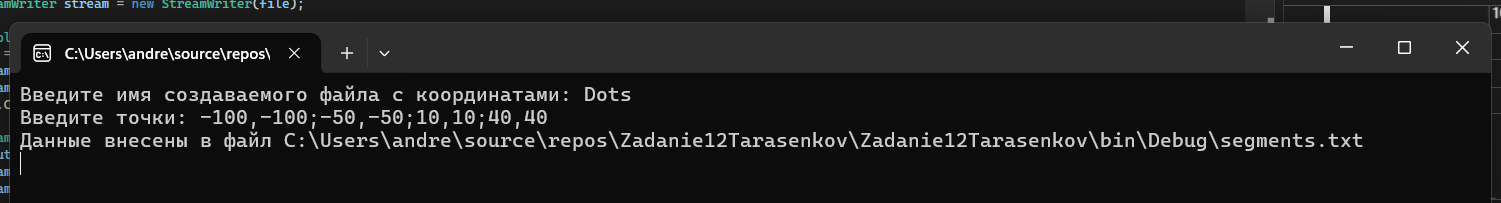
{

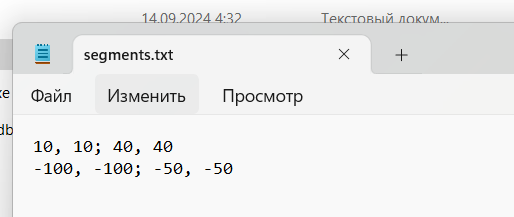
return Math.Sqrt(Math.Pow(point1[0] - point2[0], 2) + Math.Pow(point1[1] - point2[1], 2));

}

}

1. **Расчет тестовых примеров на ПК**

****

****

1. **Вывод по работе**

В среде Microsoft Visual Studio на языке Visual C# в консольном режиме составлено программное обеспечение для решения типовых задач программирования по тематике «Файлы» («Files»).