Отчет по практической работе №19

Наименование работы: коллекции, классы-прототипы.

Индивидуальные задания (вариант №9):

1. Создать класс Point (точка) . На его основе создать классы ColoredPoint и Line(линия) . На основе класса Line создать классы в коллекции(списке) ColoredLine и PolyLine (многоугольник). В классах описать следующие элементы:

• конструкторы с параметрами и конструкторы по умолчанию;

• свойства для установки и получения значений всех координат, а также для изменения цвета и получения текущего цвета;

• для линий — методы изменения угла поворота линий относительно первой точки;

• для многоугольника — метод масштабирования.

2. Описать структуру с именем TRAIN, содержащую следующие поля:

• название пункта назначения;

• номер поезда;

• время отправления.

Написать программу, выполняющую следующие действия:

• ввод с клавиатуры данных в коллекцию(список), состоящий из восьми элементов типа TRAIN (записи должны быть упорядочены по номерам поездов);

• вывод на экран информации о поезде, номер которого введен с клавиатуры (если таких поездов нет, вывести соответствующее сообщение).

Ход работы:

1. Разработал программу в соответствии с заданием №1. Были реализованы методы, классы, массивы объектов, наследование, перегрузка.

Листинг программы:

using System;

using System.Collections.Generic;

namespace Lab19.\_1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите кол-во сторон многоугольника: ");

int n = int.Parse(Console.ReadLine());

Totchka(n);

}

public static void Totchka(int n)

{

List<Point> point = new List<Point>();

for (int i = 0; i < n; i++)

{

point.Add(new Point());

Console.Write($"Введите координату Х{i + 1}: ");

point[i].X = int.Parse(Console.ReadLine());

Console.Write($"Введите координату Y{i + 1} : ");

point[i].Y = int.Parse(Console.ReadLine());

}

for (int i = 0; i < n; i++)

{

point[i].GetInfo();

}

}

}

public class Point

{

public int X { get; set; }

public int Y { get; set; }

public Point() { }

public Point(int x, int y)

{

X = x;

Y = y;

}

public virtual void GetInfo()

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"Координаты вершин: {X}, {Y}");

Console.ResetColor();

}

}

public class ColoredPoint:Point

{

public string ColorP { get; set; }

public ColoredPoint(int x, int y, string colorp):base(x,y)

{

ColorP = colorp;

}

public override void GetInfo()

{

Console.WriteLine($"Координаты точки: {X}, {Y}; Цвет: {ColorP}");

}

}

public class Line:Point

{

public int Xl { get; set; }

public int Yl { get; set; }

public Line(int xl, int yl, int x, int y) : base(x, y)

{

Xl = xl;

Yl = yl;

}

public override void GetInfo()

{

Console.WriteLine($"Координаты точки №1: {X}, {Y}; Координаты точки №2: {Xl}, {Yl}");

}

}

public class ColoredLine:Line

{

public string ColorL { get; set; }

public ColoredLine(int xl, int yl, int x, int y, string colorl):base(xl, yl, x, y)

{

ColorL = colorl;

}

public override void GetInfo()

{

Console.WriteLine($"Координаты точки №1: {X}, {Y}; Координаты точки №2: {Xl}, {Yl}; Цвет: {ColorL}");

}

}

public class PolyLine:Line

{

public List<string> PolL;

public PolyLine(List<string> poll, int x, int y, int xl, int yl) : base(x, y, xl, yl)

{

PolL = poll;

}

public override void GetInfo()

{

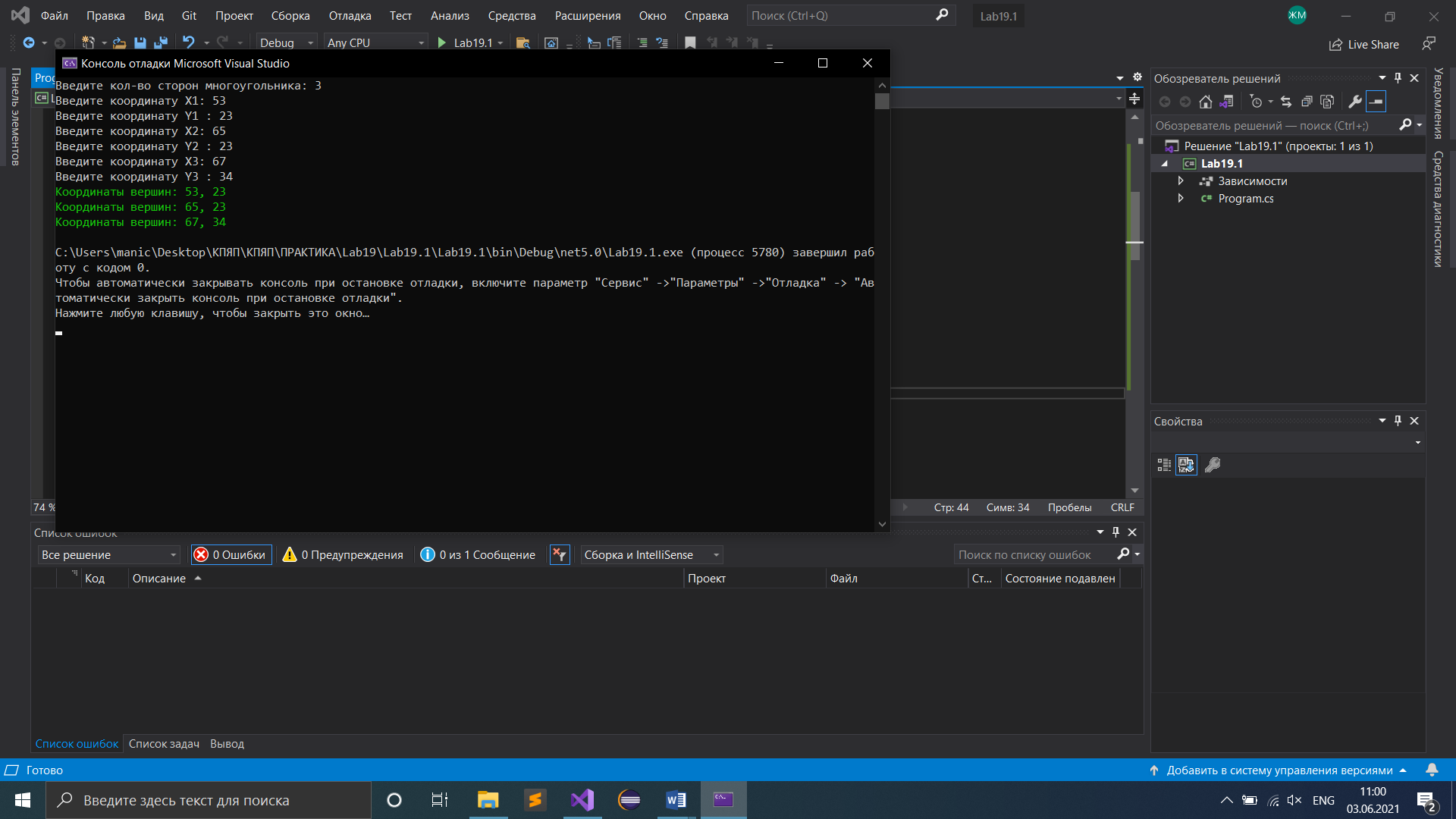
Console.WriteLine($"Координаты точки №1: {X}, {Y}; Координаты точки №2: {Xl}, {Yl};");

}

}

}

Результат выполнения программы Рис 1.1.



**Рисунок 1.1 – Пример работы программы c файлами**

1. Разработал программу в соответствии с заданием №2. Были реализованы методы, структуры, списки.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

namespace Lab19.\_2

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите номер поезда: ");

string nam = Console.ReadLine();

Trains(nam);

}

public static void Trains(string nam)

{

List<Train> tr = new List<Train>();

for (int i = 0; i < 8; i++)

{

Console.Write($"Введите номер поезда №{i + 1}: ");

string NumP = Console.ReadLine();

Console.Write("Введите пункт назначения: ");

string PunktN = Console.ReadLine();

Console.Write("Введите время отправления: ");

string Time = Console.ReadLine();

tr.Add(new Train(PunktN, NumP, Time));

}

var sortTrain = from t in tr orderby t.NumP select t;

Console.WriteLine("Отсортированные поезда: ");

foreach(Train t in tr)

{

t.GetInfo();

}

Console.WriteLine("Найденные поезда: ");

bool val = false;

foreach (Train t in sortTrain)

{

if (t.NumP == nam)

{

Console.ForegroundColor = ConsoleColor.Green;

t.GetInfo();

Console.ResetColor();

val = true;

}

}

if(val==false)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine($"Поездов с номером {nam} не существует!");

Console.ResetColor();

}

}

}

public struct Train

{

public string PunktN { get; set; }

public string NumP { get; set; }

public string Time { get; set; }

public Train(string punktn, string nump, string time)

{

PunktN = punktn;

NumP = nump;

Time = time;

}

public void GetInfo()

{

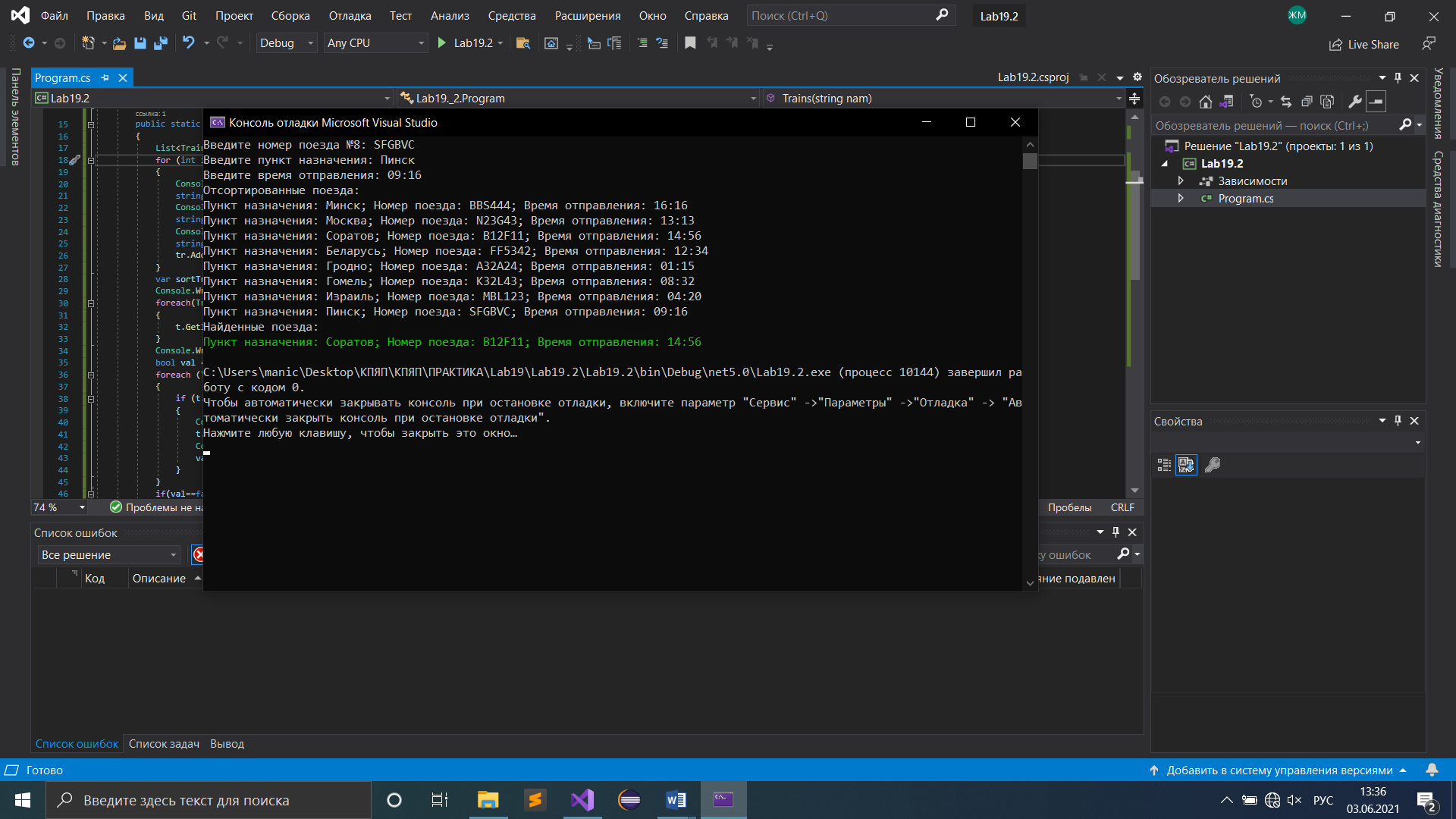
Console.WriteLine($"Пункт назначения: {PunktN}; Номер поезда: {NumP}; Время отправления: {Time}");

}

}

}

Результат выполнения программы Рис 2.1.



**Рисунок 2.1 – Пример работы программы c файлами**

Отметка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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