Отчет по практической работе №17

Наименование работы: работа с файлами.

Индивидуальные задания (вариант №9):

1. Для участия в конкурсе исполнителей необходимо заполнить анкету с данными (

— Ф.И.О и год рождения;

— название страны;

— класс музыкального инструмента (гитара, фортепиано, скрипка).

Отсортировать по ФИО и записать в файл.

1. В файле хранится список студентов расположенных в случайном порядке и их номер группы. Прочитать данные из файла. Отсортировать студентов по группам и фамилиям и записать данные в новый файл.

Ход работы:

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

Листинг программы:

using System;

using System.Collections.Generic;

using System.IO;

using System.Text;

namespace Lab17

{

delegate void Files(string message);

class Program

{

public static event Files Messag;

static string path = @"output.txt";

static void Main(string[] args)

{

Uan();

}

static void Uan()

{

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

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

List<Musik> mus = new List<Musik>();

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

{

mus.Add(new Musik());

Console.Write("Введите ФИО: ");

mus[i].Fio = Console.ReadLine();

Console.Write("Введите страну: ");

mus[i].Strana = Console.ReadLine();

Console.Write("Инструмент: ");

mus[i].Instrument = Console.ReadLine();

}

Console.WriteLine("Ваши объекты: ");

mus.Sort((left, right) => left.Fio.CompareTo(right.Fio));

string[] last\_arr = new string[n];

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

{

Musik ms = mus[i];

last\_arr[i] = ms.ToString();

mus[i].GetInfo();

}

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

{

File.WriteAllLines(path, last\_arr, Encoding.UTF8);

}

FileInfo fi = new FileInfo(path);

if (!fi.Exists)

{

Messag += Mess2;

Messag?.Invoke("Файла не существует! \n Создаем!");

File.Create(path);

Messag -= Mess2;

}

else

{

Messag += Mess1;

Messag?.Invoke("Файл существует! \nНачинаем сортировку!");

}

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

{

}

}

public static void Mess1(string message)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine(message);

Console.ResetColor();

}

public static void Mess2(string message)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(message);

Console.ResetColor();

}

}

public class Musik

{

public string Fio { get; set; }

public string Strana { get; set; }

public string Instrument { get; set; }

public Musik() { }

public Musik(string fio, string strana, string instrument)

{

Fio = fio;

Strana = strana;

Instrument = instrument;

}

public void GetInfo()

{

Console.WriteLine($"ФИО: {Fio}, Страна: {Strana}, Инструмент: {Instrument}");

}

public override string ToString()

{

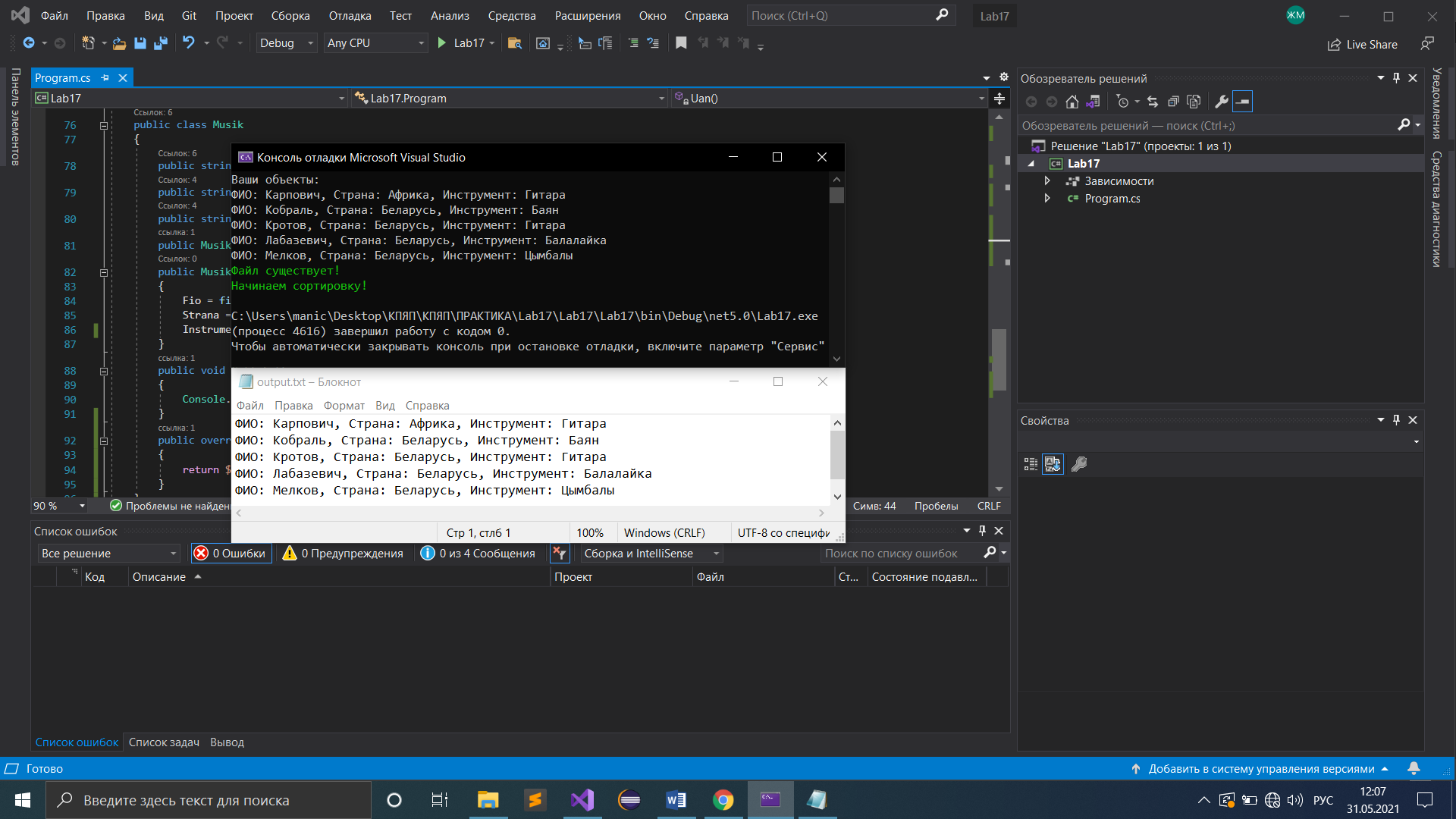
return $"ФИО: {Fio}, Страна: {Strana}, Инструмент: {Instrument}";

}

}

}

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, классы, массивы.

Листинг программы:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

namespace ConsoleApplicationTest

{

public static class Program

{

static string input = @"users.txt";

static string output = @"output.txt";

private static void Main()

{

FileInfo fi = new FileInfo(output);

Auth();

string[] allLines = File.ReadAllLines(input);

List<Student> st = new List<Student>();

string[] fields = new string[5];

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

fields = line.Split(';');

Student student = new Student(fields[0], fields[1], fields[2], fields[3], fields[4]);

st.Add(student);

}

foreach(Student student in st)

{

//Console.WriteLine(student);

}

List<string> isx = new List<string>();

var sortCors = from u in st orderby u.Name select u;

sortCors = from u in sortCors orderby u.Cours select u;

foreach (Student u in sortCors)

{

Console.WriteLine(u);

isx.Add(Convert.ToString(u));

}

// Сохраняем в файл

if (fi.Exists)

{

File.WriteAllLines(output, isx, Encoding.UTF8);

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"Отсортированные данные записаны в файл {output}");

Console.ResetColor();

}

else

{

File.Create(output);

File.WriteAllLines(output, isx, Encoding.UTF8);

Console.ReadKey();

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"Отсортированные данные записаны в файл {output}");

Console.ResetColor();

}

}

public static void Auth()

{

FileInfo fn = new FileInfo(input);

if (fn.Exists)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"Файл {input} существует! Продолжаем работу!");

Console.ResetColor();

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine($"Файл {input} не найден! Проверьте расположение файла!");

Console.ResetColor();

}

}

}

public struct Student : IComparable

{

public Student(string name, string fac, string group, string cours, string nums)

: this()

{

Name = name;

Fac = fac;

Group = group;

Cours = cours;

Nums = nums;

string[] str = nums.Split(' ');

Ocen = new int[5];

for (int i = 0; i < 5; i++)

{

Ocen[i] = Convert.ToInt32(str[i]);

}

}

public string Name { get; set; }

public string Fac { get; set; }

public string Group { get; set; }

public string Cours { get; set; }

public string Nums { get; set; }

public int[] Ocen { get; set; }

public int CompareTo(object obj)

{

return Fac.CompareTo(((Student)obj).Fac);

}

public override string ToString()

{

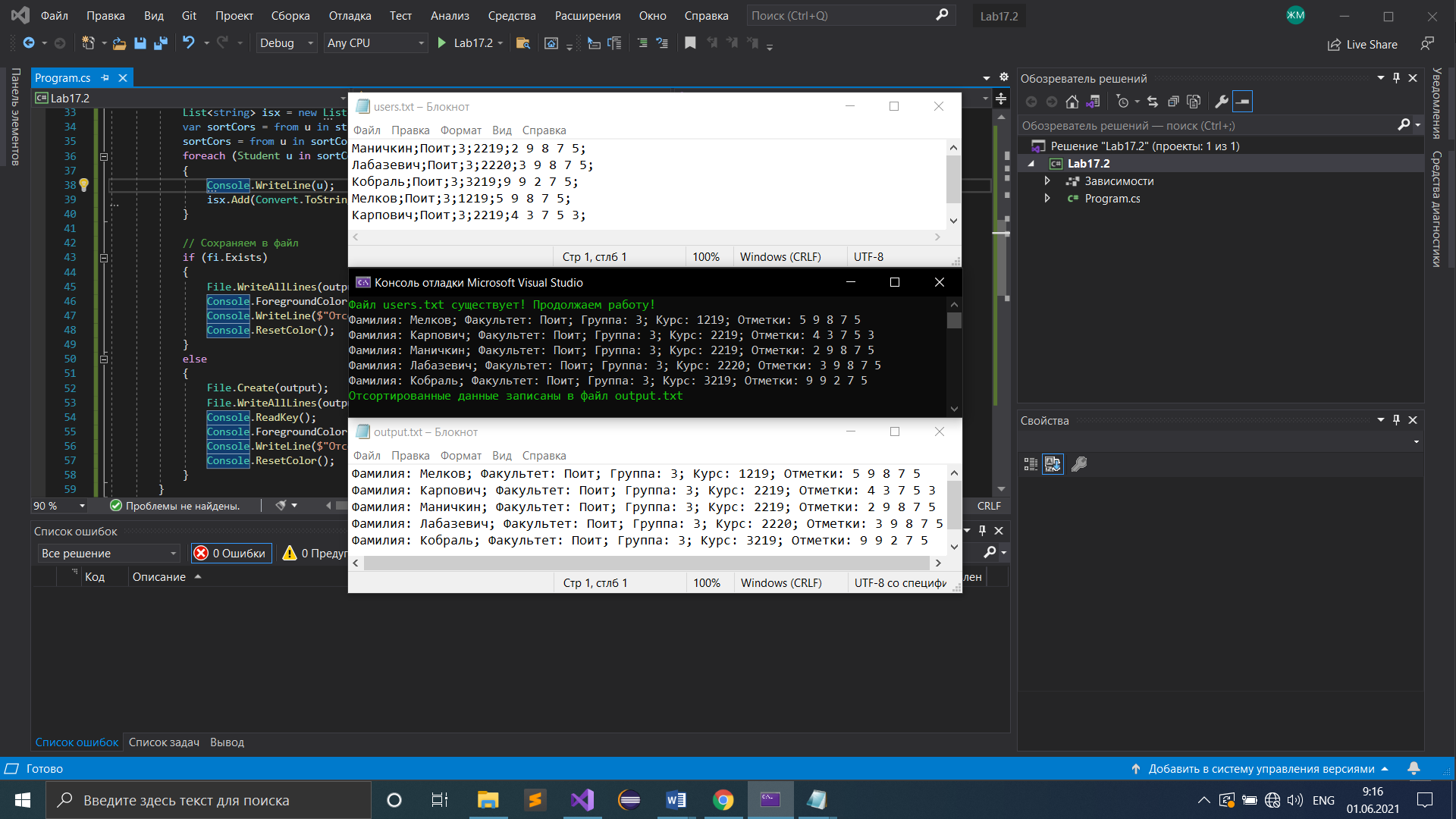
return string.Format($"Фамилия: {Name}; Факультет: {Fac}; Группа: {Group}; Курс: {Cours}; Отметки: {Nums}");

}

}

}

Результат выполнения программы Рис 2.1.



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

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

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

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

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

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

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