Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

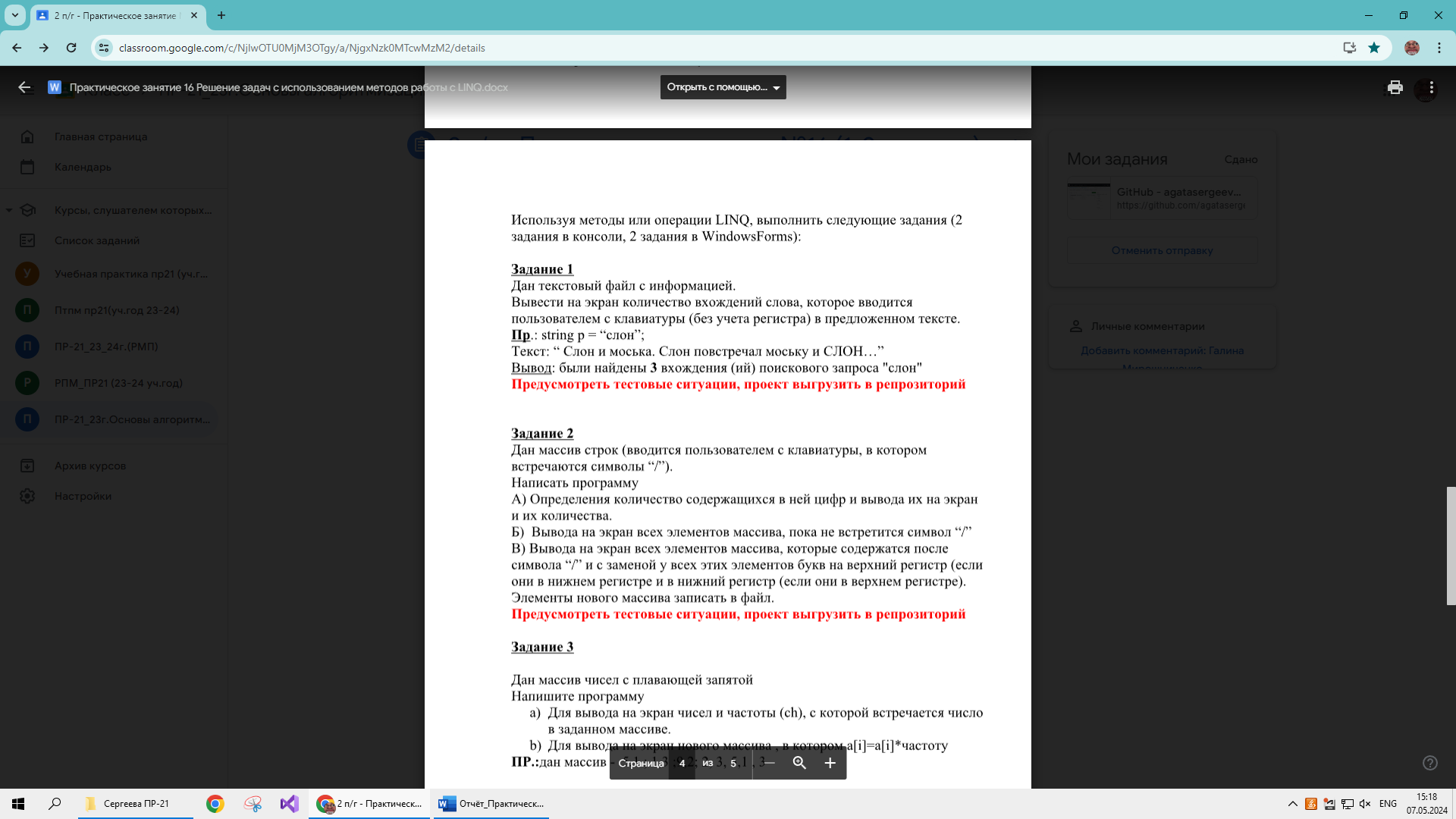
Отчёт по программе «**Практическое занятие 16**»

Выполнила: Рубцова Диана

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2024 г.

**Задание 1:**

**Входные данные:** stroka-текст из файла ‘file1.txt’, string[];

word – искомое слово, string.

**Выходные данные:** count – количество вхождений слова в текст, int.

**Листинг программы:**

private void button1\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

string stroka = "";

if (File.Exists("file1.txt"))

{

StreamReader sr = File.OpenText("file1.txt");

while (!sr.EndOfStream)

{

stroka = sr.ReadLine();

listBox1.Items.Add("из файла:");

listBox1.Items.Add(stroka);

}

sr.Close();

string word = textBox1.Text.ToLower();

if (word == "" || word == " ")

{ MessageBox.Show("вы неправильно ввели слово", "error", MessageBoxButtons.OK, MessageBoxIcon.Error); }

else

{

int count = stroka.ToLower().Split(' ', ',', '.').Count(s => s == word);

listBox1.Items.Add($"были найдены {count} вхождения(ий) поискового запроса '{word}' ");

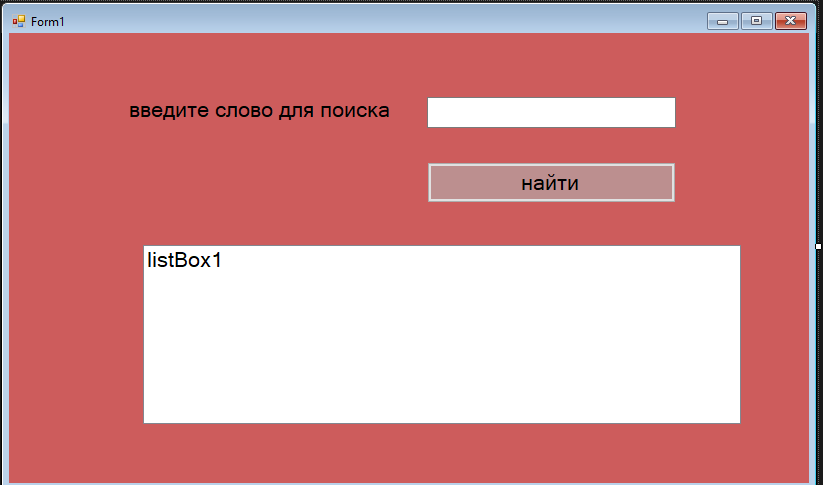
}

}

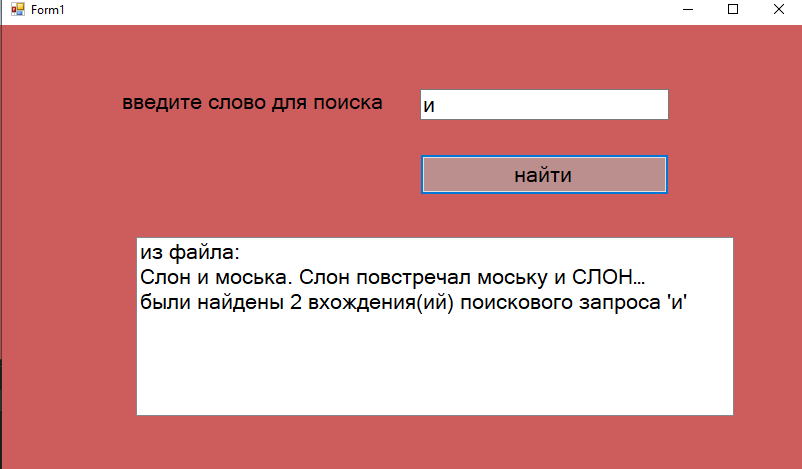
else MessageBox.Show("файл не найден", "error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

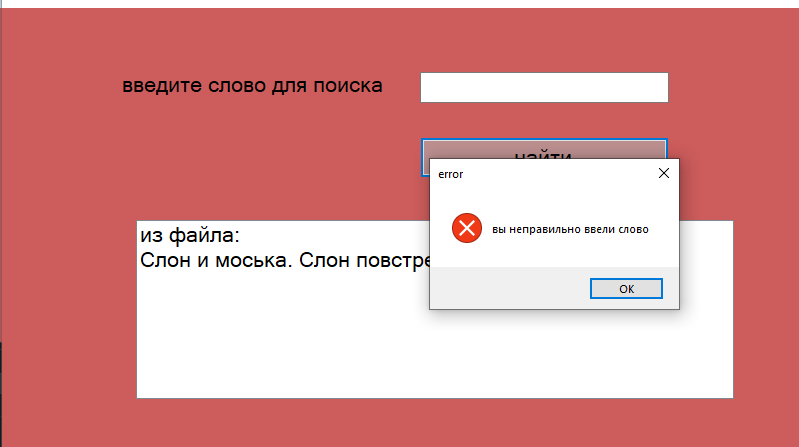
**Тестовые ситуации:**



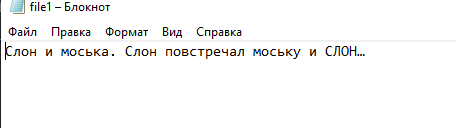
Так выглядит форма, при запуске программы



На этом скриншоте показано, как работает поиск слов

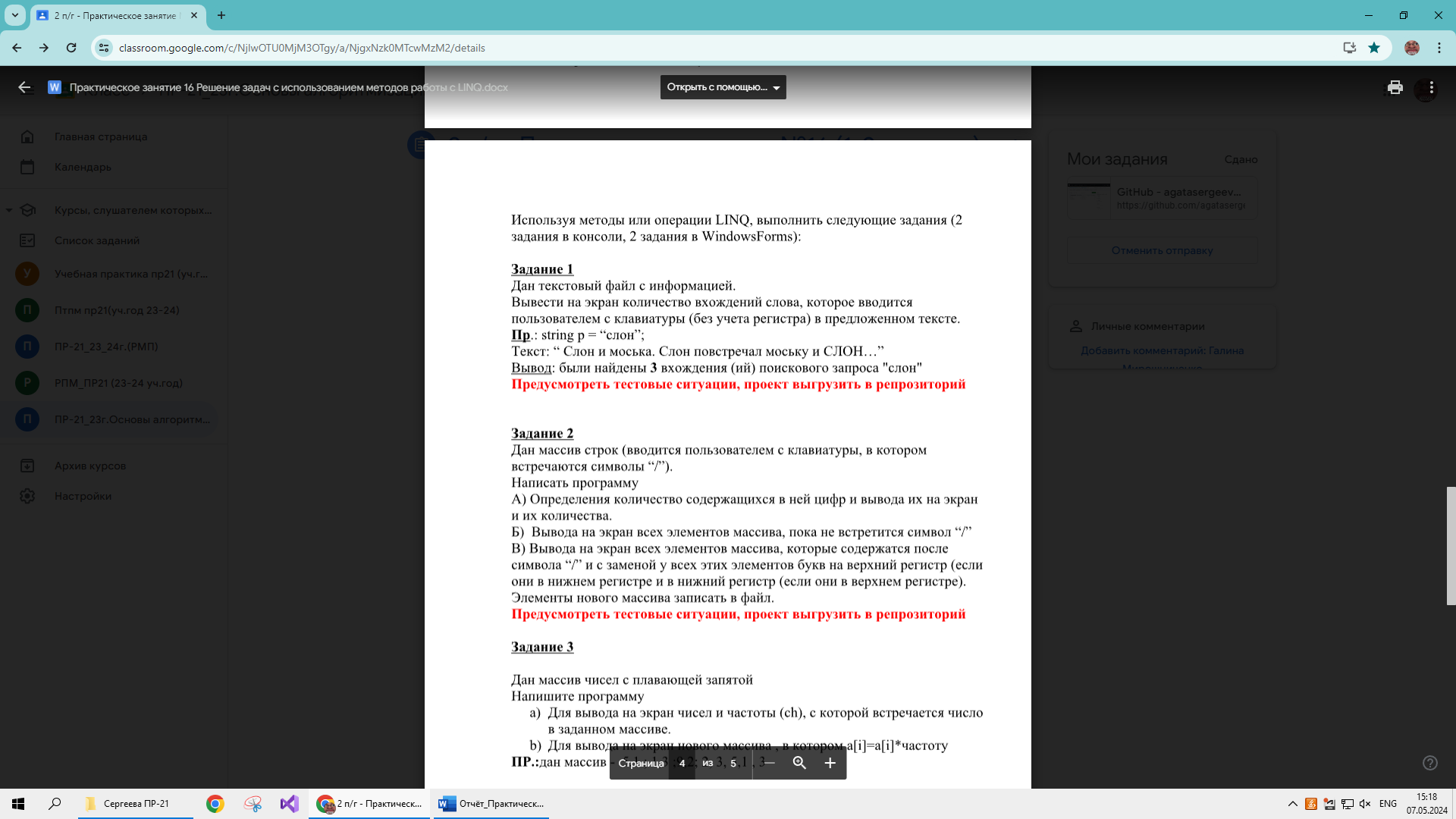


Ошибка на пустой ввод



Так выглядит сам файл

**Задание 2:**



**Входные данные:** array – введённая пользователем строка, string[].

**Выходные данные:** количество цифр в строке, var; элементы до первого ‘/’, var; элементы после первого ‘/’, var.

**Листинг программы:**

if (File.Exists("file1.txt"))

{

using (StreamWriter sw = new StreamWriter("file1.txt"))

{

Console.WriteLine("Введите строку:");

string[] array = Console.ReadLine().Split(' ');

bool containSlash = array.Any(x => x.Contains("/"));

if (containSlash == true)

{

int digitcount = array.SelectMany(x => x).Count(char.IsDigit);

Console.WriteLine($"Количество цифр в массиве: {digitcount}");

Console.WriteLine("Элементы массива до символа '/':");

bool slash = false;

foreach (string s in array)

{

if (s.Contains("/"))

{

slash = true;

break;

}

Console.Write($"{s}");

}

Console.WriteLine("\nЭлементы массива после символа '/' с заменой регистра:");

var r = array.SkipWhile(x => x != "/")

.Select(x => string.Concat(x.Select(c => char.IsUpper(c) ? char.ToLower(c) : char.ToUpper(c))))

.ToArray();

foreach (var item in r)

{

Console.Write($"{item}");

}

}

else Console.WriteLine("В строке нет '/'!");

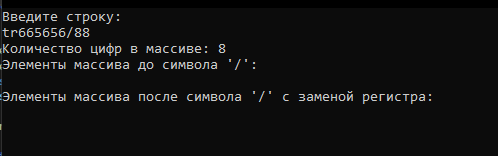
}

}

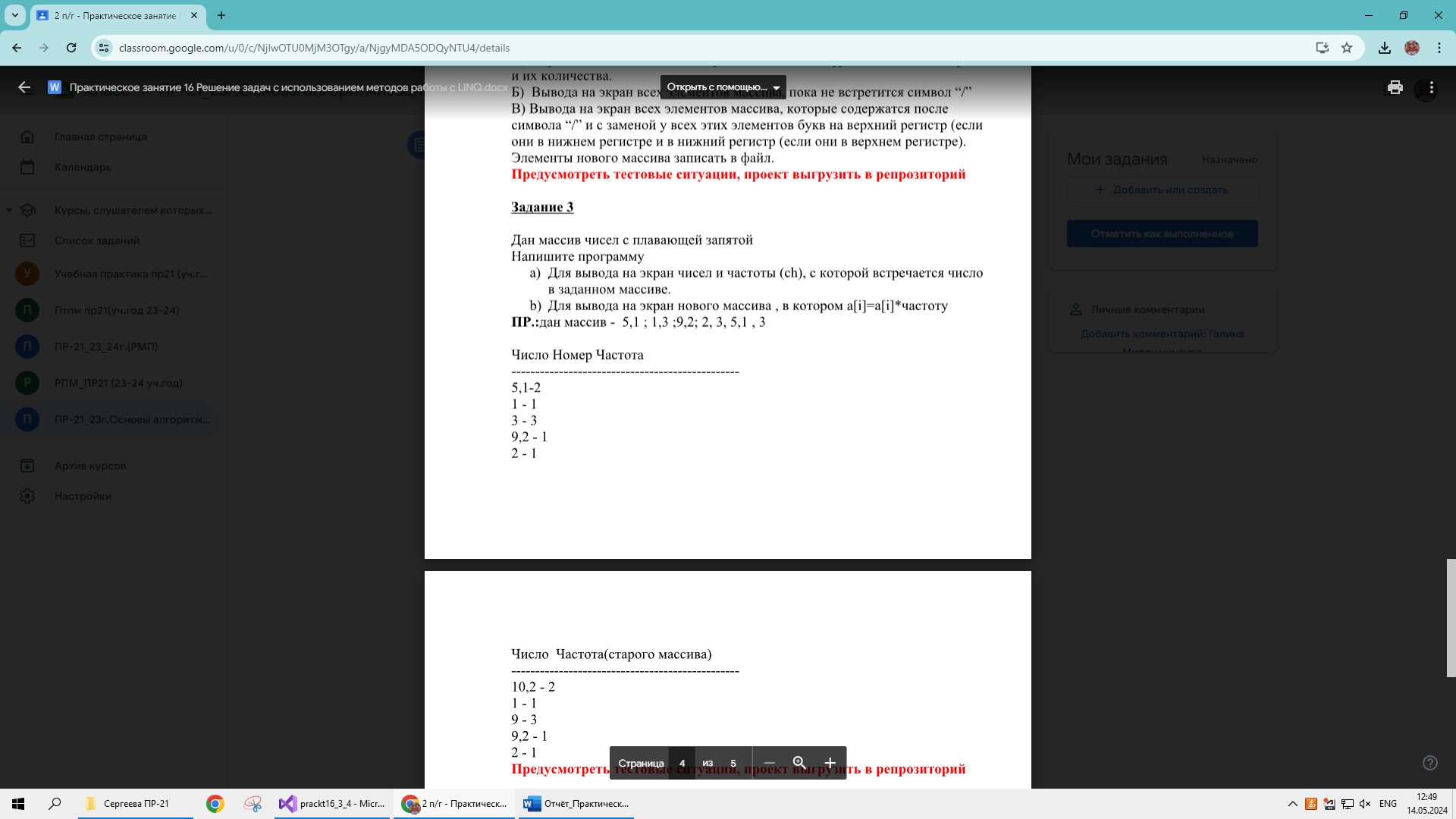
else Console.WriteLine("файл для записи не найден");

Console.ReadKey();

**Тестовые ситуации:**



**Задание 3:**



**Входные данные:** numbers – массив чисел из файла, double[]

**Выходные данные:** v – вывод числа и частоты, var.

**Листинг программы:**

if (File.Exists("file.txt"))

{

string file = File.ReadAllText("file.txt");

if (!string.IsNullOrEmpty(file))

{

string[] fromfFile = file.Split(' ', (char)StringSplitOptions.RemoveEmptyEntries);

double[] numbers = fromfFile.Select(double.Parse).ToArray();

var f = numbers.GroupBy(n => n).Select(g => new { num = g.Key, period = g.Count() });

Console.WriteLine("Число Номер Частота");

Console.WriteLine("---------------------------");

foreach (var v in f)

{

Console.WriteLine($"{v.num}\t{v.period}");

}

Console.WriteLine("\nЧисло\tЧастота(старого массива)");

Console.WriteLine("---------------------------");

foreach (var v in f)

{

Console.WriteLine($"{v.num \* v.period}\t{v.period}");

}

}

else

{

Console.WriteLine("Файл пустой!");

}

}

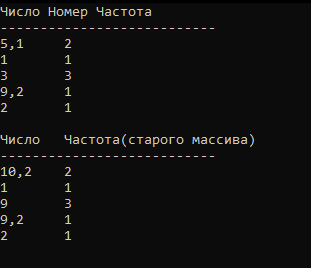
else

{

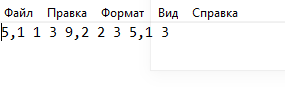
Console.WriteLine("Файл с данными не найден!");

}

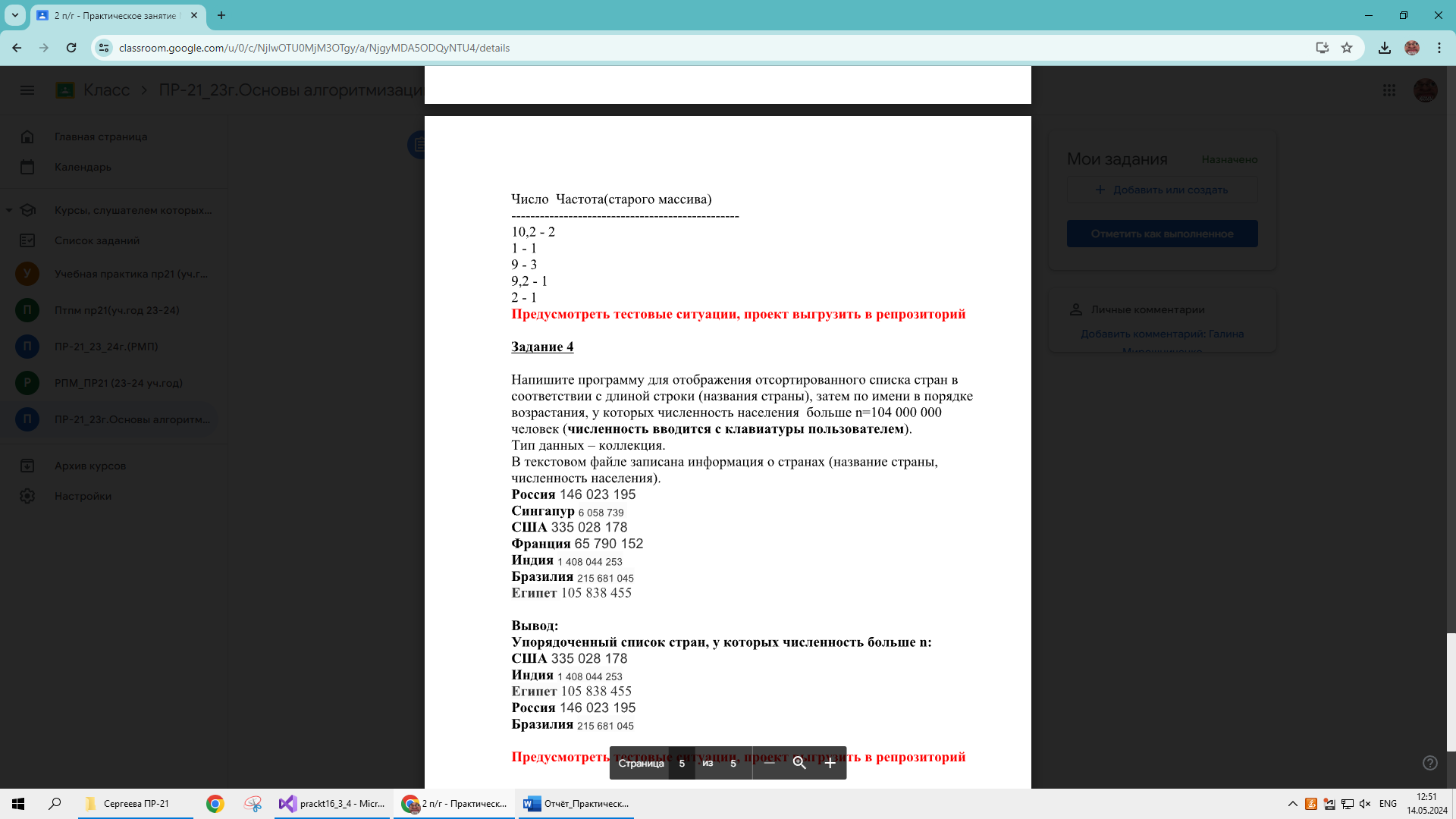
**Тестовые ситуации:**



Файл с данными



**Задание 4:**



**Входные данные:** c, List.

**Выходные данные:** sort– отсортированные страны по длине строки, var; sortt– страны, численность которых больше введённой пользователем, var

**Листинг программы:**

List<Class1> cc = new List<Class1>();

private void Form1\_Load(object sender, EventArgs e)

{

if (File.Exists("f.txt"))

{

string[] ffile = File.ReadAllLines("f.txt");

foreach (string ff in ffile)

{

char d = ff.First(c => char.IsDigit(c));

string[] part = ff.Split(d);

string name = part[0];

string popul = d + part[1];

int population = int.Parse(popul.Replace(" ", ""));

cc.Add(new Class1(name, population));

}

var sort = cc.OrderBy(c => c.Name.Length);

listBox1.Items.Clear();

foreach (Class1 c in sort)

{

listBox1.Items.Add(c.Name + " " + c.Population);

}

}

else MessageBox.Show("Файл с данными не найден!");

}

private void button1\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox1.Text.Replace(" ", ""), out int rez))

{

long n = long.Parse(textBox1.Text.Replace(" ", ""));

var sortt = cc.Where(c => c.Population > n)

.OrderBy(c => c.Name.Length)

.ToList();

listBox2.Items.Clear();

foreach (Class1 c in sortt)

{

listBox2.Items.Add(c.Name + " " + c.Population);

}

}

else MessageBox.Show("Неправильный формат ввода!");

}

internal class Class1

{

public string Name { get; set; }

public int Population { get; set; }

public Class1(string name, int population)

{

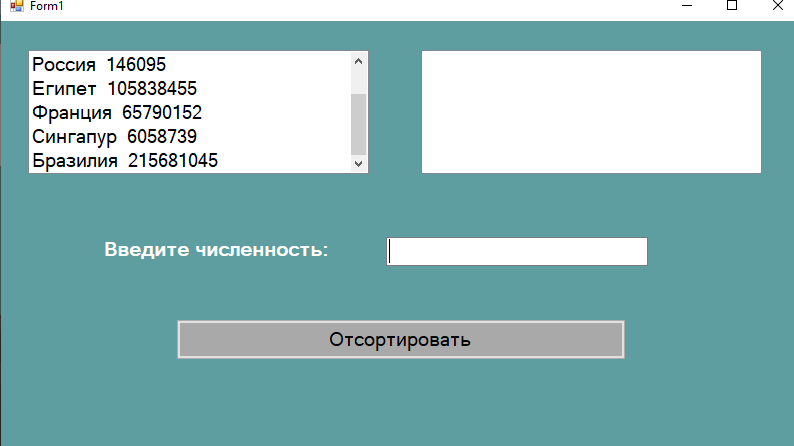
Name = name;

Population = population;

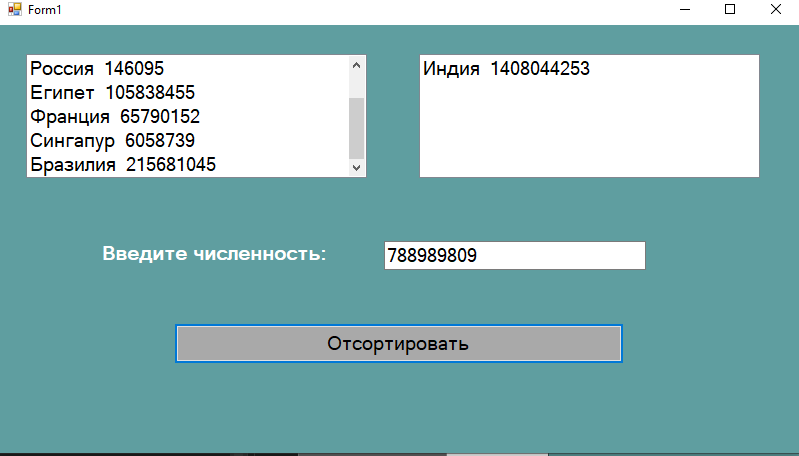
}

}

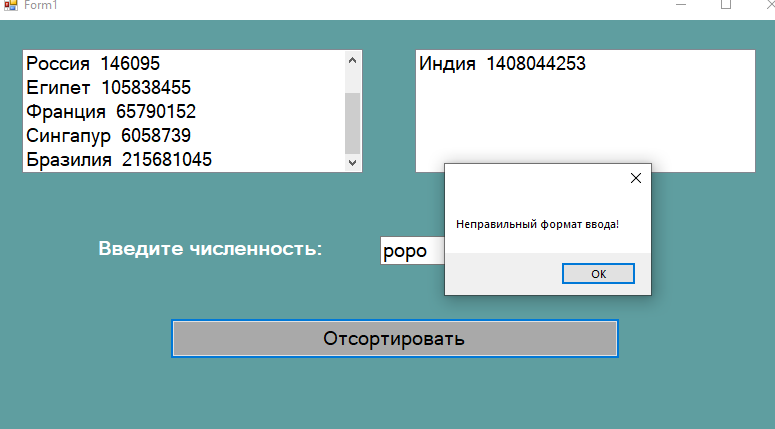
**Тестовые ситуации:**



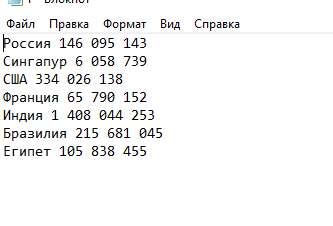
Так выглядит форма, при запуске программы



На этом скриншоте показано как работает сортировка



Ошибка при неправильном вводе



Файл с данными