Отчет по практической работе №10

Наименование работы: интерфейсы.

Индивидуальные задания (вариант №9):

1. В индивидуальном задании №2 из практической работы №9 реализуйте метод CompareTo интерфейса IComparable, перегрузите метод ToString базового класса object и необходимые операции отношения.

На основе данных входного файла составить список студентов, включающий фамилию, факультет, курс, группу, 5 оценок. Вывести в новый файл информацию о тех студентах, которые имеют хотя бы одну двойку, отсортировав их по курсу.

2) Описать интерфейс ISequence, соответствующий числовым по- следовательностям. Он должен содержать метод GetElement, возвращаю- щий элемент последовательности по его номеру. Описать классы ArithmeticProgression (арифметическая прогрессия) и GeometricProgression (геометрическая прогрессия), реализующие этот интерфейс. Параметры прогрессий (первый элемент и разность или знаменатель) должны задаваться при создании экземпляра. Написать функцию Sum которая возвращают сумму указанного количества элементов последовательности начиная с первого. Написать программу, использующую эту функцию. Пример вызова функции: var s = Sum (new ArithmeticProgression ( 3 , 5 ) , 10 ). Построить UML-диаграмму.

Ход работы:

1. Разработал программу в соответствии с заданием №1. Были реализованы методы, классы, работа с файлами, интерфейсы.

Листинг программы:

using System;

using System.IO;

using System.Text;

namespace ConsoleApplicationTest

{

public static class Program

{

static string path = @"users.txt";

static string outs = @"output.txt";

private static void Main()

{

Auth();

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

Student[] students = new Student[allLines.Length];

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]);

students[i] = student;

}

Array.Sort(students);

foreach (Student student in students)

{

Console.WriteLine(student);

}

string[] linesToSave = new string[students.Length];

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

{

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

{

if (students[i].Ocen[j] == 2)

{

Student student = students[i];

linesToSave[i] = student.ToString();

}

}

}

// Сохраняем в файл

FileInfo fi = new FileInfo(outs);

if (fi.Exists)

{

File.WriteAllLines(outs, linesToSave, Encoding.UTF8);

Console.ForegroundColor = ConsoleColor.Green;

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

Console.ResetColor();

}

else

{

File.Create(outs);

File.WriteAllLines(outs, linesToSave);

Console.ReadKey();

Console.ForegroundColor = ConsoleColor.Green;

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

Console.ResetColor();

}

}

public static void Auth()

{

FileInfo fn = new FileInfo(path);

if (fn.Exists)

{

Console.ForegroundColor = ConsoleColor.Green;

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

Console.ResetColor();

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine($"Файл {path} 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()

{

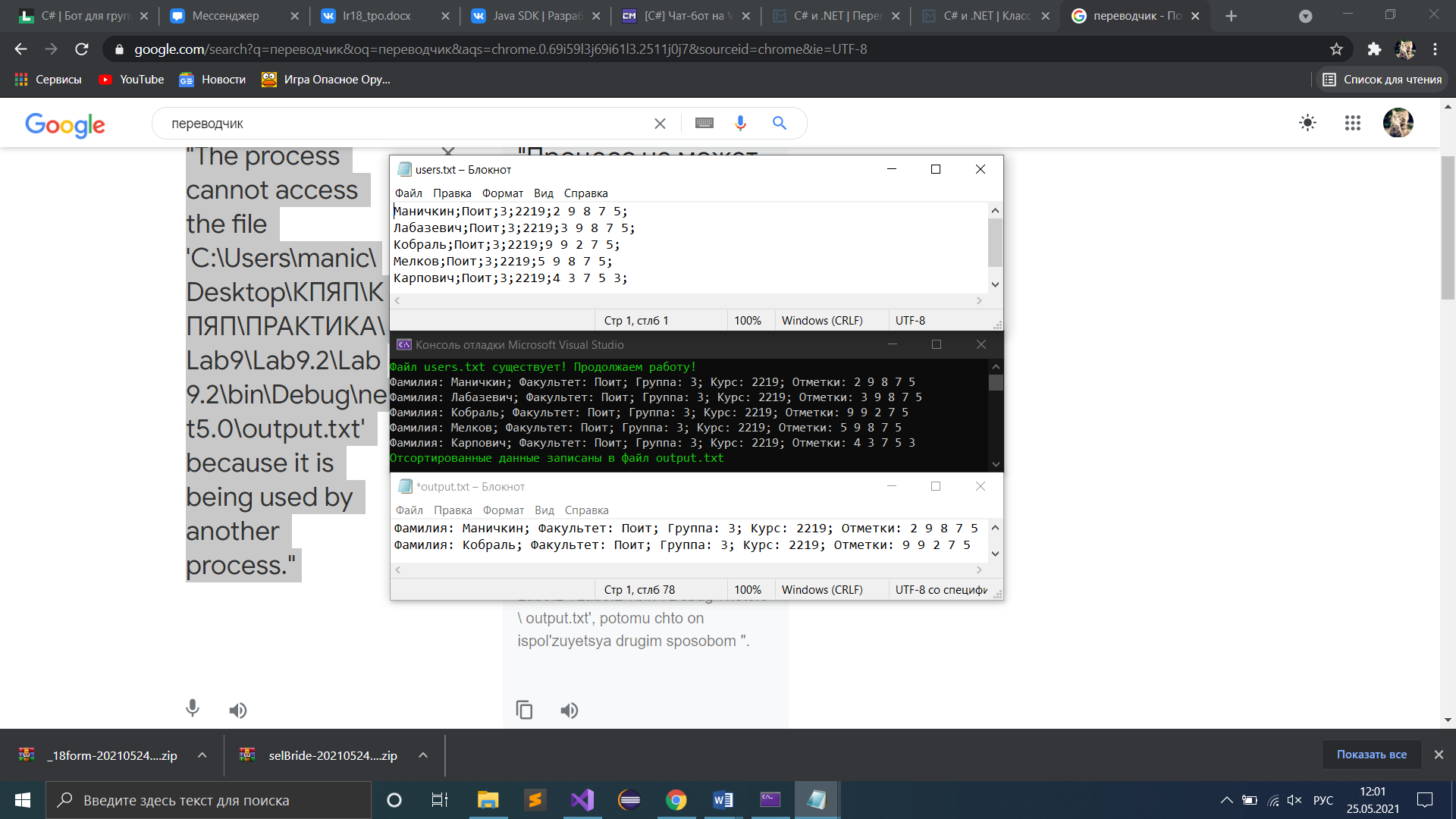
return string.Format($"Фамилия: {Name}; Факультет: {Fac}; Группа: {Group}; Курс: {Cours}; Отметки: {Nums}");

}

}

}

Результат выполнения программы Рис 1.1.



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

1. Разработал программу в соответствии с заданием №2. Были реализованы методы, классы, наследование.

Листинг программы:

using System;

namespace Lab10.\_2

{

interface ISequence

{

void GetElement(int nam);

void Sum();

}

class Program

{

static void Main(string[] args)

{

Console.Write("Введите элемент: ");

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

GeometricProgression gm = new GeometricProgression();

gm.Geometric(2, 2, 10);

gm.GetElement(s);

gm.Sum();

ArithmeticProgression am = new ArithmeticProgression();

am.Arithmetic(2, 3, 30);

am.Sum();

am.GetElement(s);

}

}

public class GeometricProgression : ISequence

{

int i;

public int sum;

public int[] arr\_a;

public void Geometric(int uan, int tm, int n)

{

Console.Write("Ваши элементы геометрической прогресси: ");

arr\_a = new int[n];

for (i = 0; i < n; i++,uan\*=tm)

{

arr\_a[i] = uan \* tm;

Console.Write(arr\_a[i] + " ");

}

Console.WriteLine();

}

public void Sum()

{

Console.Write("До какого элемента вывести сумму? ");

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

int sum = 0;

for(i = 0; i < kon; i++)

{

sum += arr\_a[i];

}

Console.WriteLine($"GF: Сумма элементов до элемента под №{kon} равна: {sum}");

}

public void GetElement(int nam)

{

int nams = arr\_a[nam];

Console.WriteLine($"GF: Под номером {nam}, лежит {nams} элемент");

}

}

public class ArithmeticProgression : ISequence

{

public int[] arr\_a;

public void Arithmetic(int uan, int razn, int n)

{

Console.Write("Ваши элементы арифметической прогресси: ");

arr\_a = new int[n];

for (int i = 0; i < n; i++, uan += razn)

{

arr\_a[i] = uan + razn;

Console.Write(arr\_a[i] + " ");

}

Console.WriteLine();

}

public void Sum()

{

Console.Write("До какого элемента вывести сумму? ");

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

int sum = 0;

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

{

sum += arr\_a[i];

}

Console.WriteLine($"AF: Сумма элементов до элемента под №{kon} равна: {sum}");

}

public void GetElement(int nam)

{

int nams = arr\_a[nam];

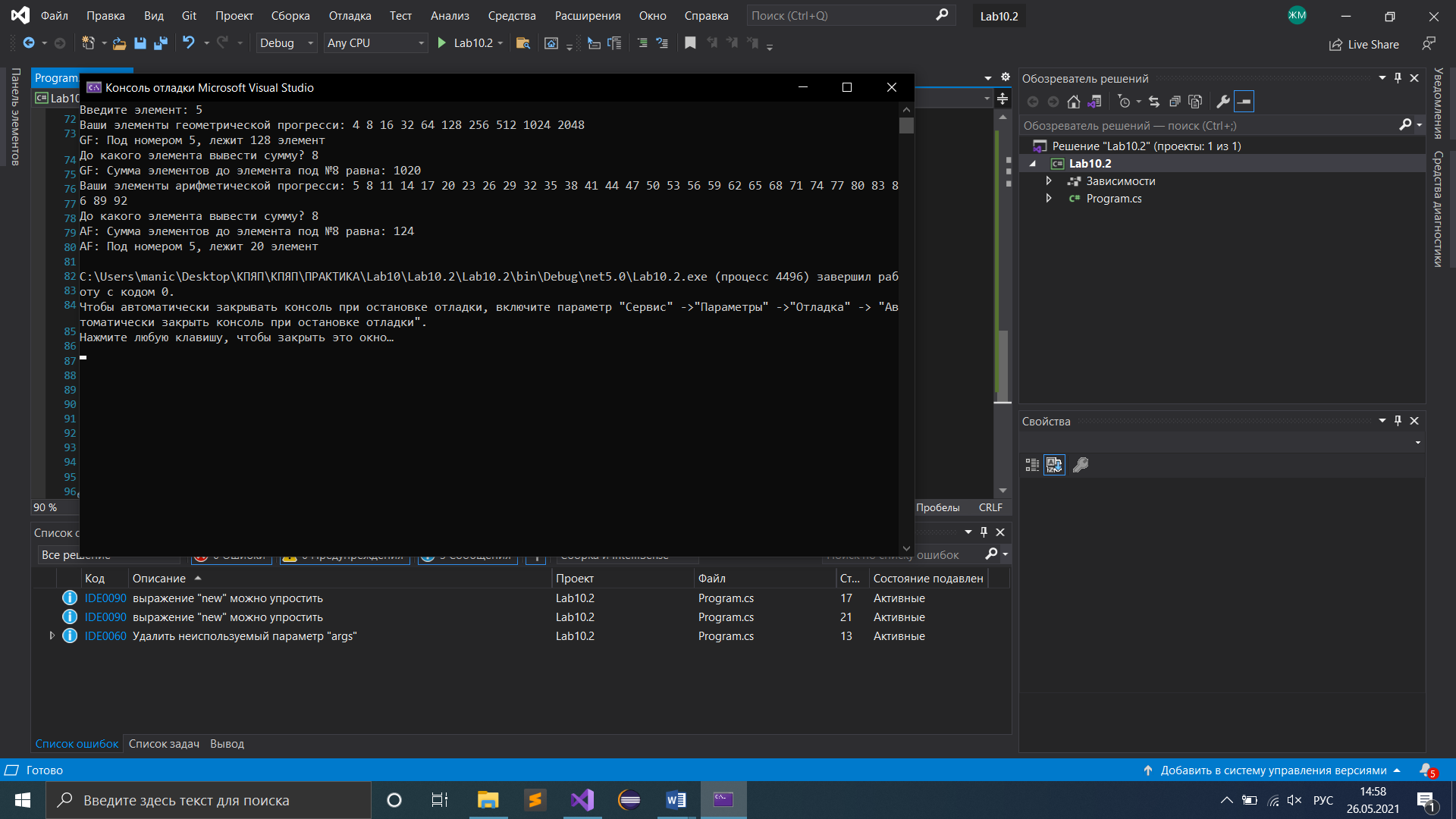
Console.WriteLine($"AF: Под номером {nam}, лежит {nams} элемент");

}

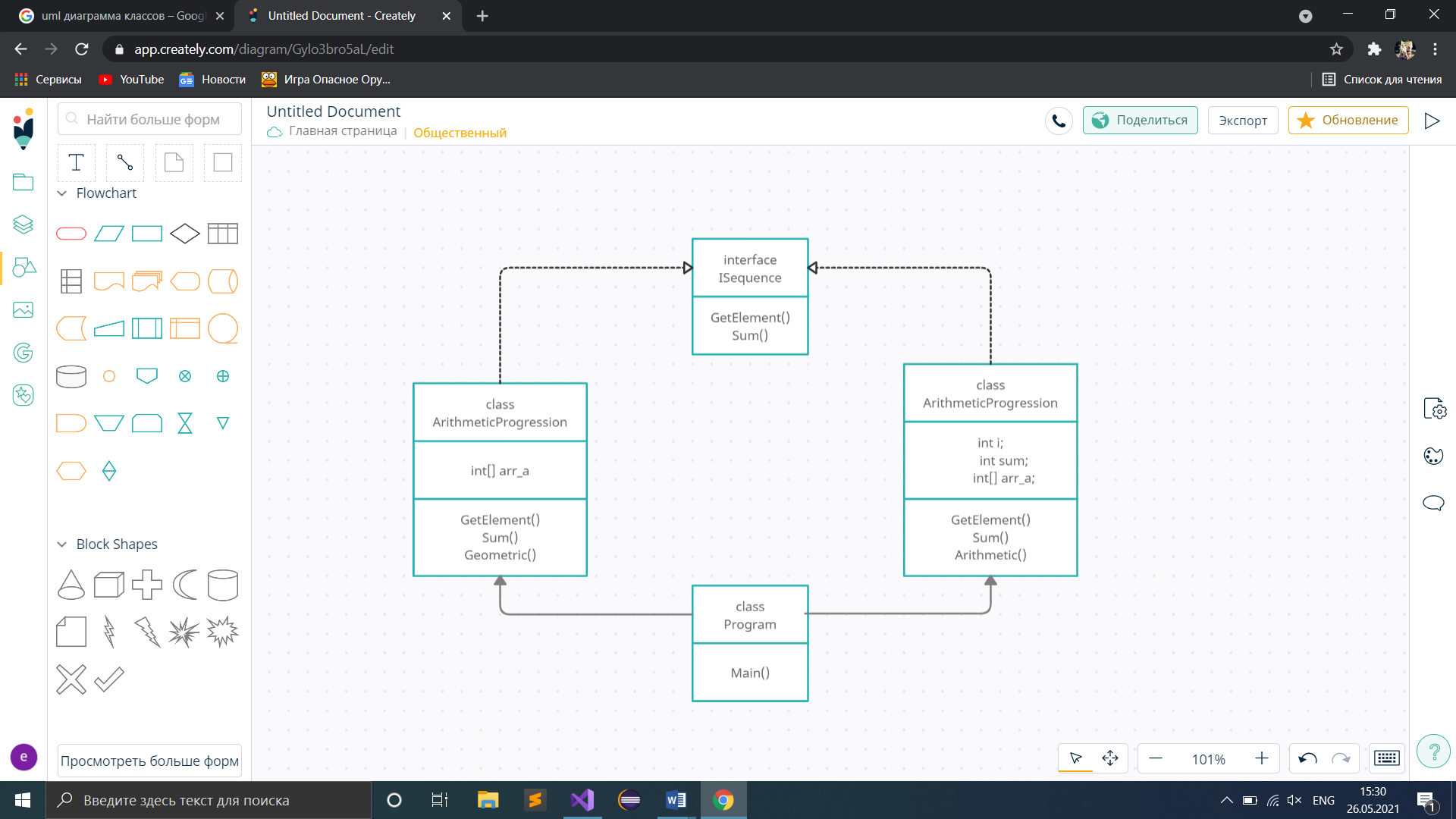
}

}

Результаты выполнения программы Рис 2.1.



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



**Рисунок 2.2 – UML диаграмма класов.**

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

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

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

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

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

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