Тема: Классы

Задание 1

Создайте проект, в котором опишите класс для решения задачи.

Указания. Разрабатываемый класс должен содержать следующие элементы: скрытое поле, статическое поле, константа, конструкторы без параметров и с параметрами (имена некоторых полей должны совпадать с идентификаторами параметров), свойство, метод вывода полей и указанный в таблице метод.

Создал проект, в котором описал класс для решения задачи моего варианта (2, 3, 4)

Вариант 2

using System;

using static System.Console;

class Program

{

static void Main(string[] args)

{

Flat f = new Flat(5, 8);

WriteLine(f.GetInfo());

}

}

class Flat

{

public int Number { get; set; }

private static int Cost = 900;

public int Square { get; set; }

public Flat()

{

Number = 1;

Square = 60;

}

public Flat(int number, int square)

{

Number = number;

Square = square;

}

public string GetInfo()

{

return $"Номер кв. {Number}, результат {Cost \* Square}";

}}

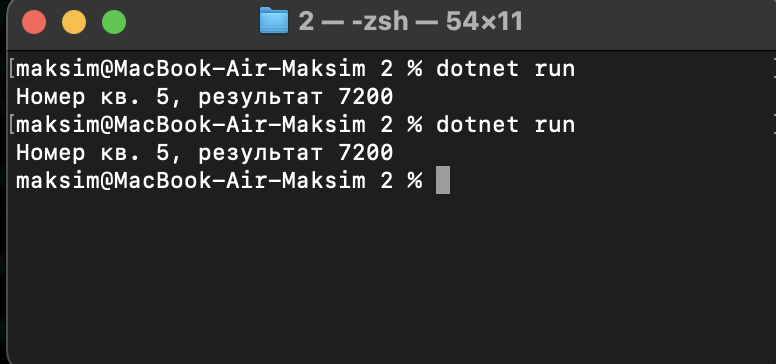


Рисунок 1. Результат 1 задачи.

Вариант 3

using System;

using static System.Console;

class Program

{

static void Main(string[] args)

{

Agropole a = new Agropole("Майский", 6000);

WriteLine(a.GetInfo());

}

}

class Agropole

{

public string name { get;}

const double weight = 0.5;

private double square;

public Agropole()

{

name = "Майский";

square = 60;

}

public Agropole(string Name, double Square)

{

name = Name;

square = Square;

}

public string GetInfo()

{

return $"Название - {name}, нужный вес {weight \* square} gram";

}

}

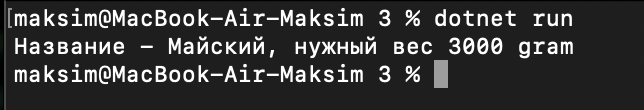


Рисунок 2. Результат 2 задачи.

Вариант 4

using System;

class Program

{

static void Main(string[] args)

{

Table table = new Table();

table.DisplayInfo();

Console.WriteLine("Cost: " + table.GetCost());

}

}

class Table

{

private string name;

private double area;

const double additionalCost = 500000;

public Table()

{

name = "Table";

area = 210;

}

public Table(string name, double area)

{

this.name = name;

this.area = area;

}

public string Name

{

get { return name; }

set { name = value; }

}

public double Area

{

get { return area; }

set { area = value; }

}

public double GetCost()

{

return area \* area / 3 + additionalCost;

}

public void DisplayInfo()

{

Console.WriteLine("Name: " + name);

Console.WriteLine("Area: " + area + " sq.cm");

}}

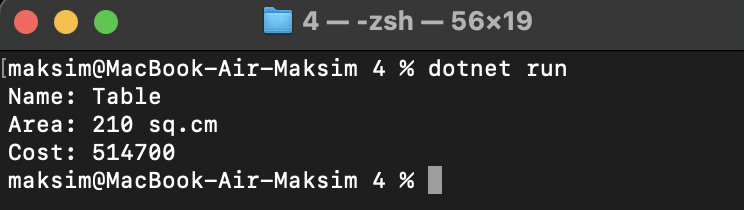


Рисунок 3. Результат 3 задачи.

**Тема Работа с коллекциями. Работа с Linq запросами**

Задание 1

class Program

{

*static* void Main(string[] args)

{

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

Random rnd = new Random();

int a = rnd.Next(1, 10);

int[] b = new int[a];

int[] c = new int[a];

int sum1 = 0;

int sum2 = 0;

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

{

b[i] = rnd.Next(1, 30);

}

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

{

c[j] = b[j] \* 78;

}

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

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

{

Console.Write(b[j] + " ");

sum1 += b[j];

}

Console.WriteLine();

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

{

Console.Write(c[j] + " ");

sum2 += c[j];

}

Console.WriteLine();

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

Console.WriteLine(sum1);

Console.WriteLine(sum2);

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

Console.WriteLine($"max first: {b.Max()}, min first: {b.Min()}");

Console.WriteLine($"max second: {c.Max()}, min second: {c.Min()}");

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

int p = Convert.ToInt32(Console.ReadLine());

if (b.Contains(p))

{

Console.WriteLine("First true");

}

else

{

Console.WriteLine("First false");

}

if (c.Contains(p))

{

Console.WriteLine("Second true");

}

else

{

Console.WriteLine("Second false");

}

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

var b50 = from elem in b

where elem > 1 && elem < 50

select elem;

var c40 = from elem in c

where elem > 70 && elem < 500

select elem;

Console.WriteLine("Элементы 1 массива, которые меньше 50 и больше 1");

foreach (var element in b50)

{

Console.Write(element + " ");

}

Console.WriteLine("");

Console.WriteLine("Элементы 2 массива, которые меньше 500 и больше 70");

foreach (var element in b50)

{

Console.Write(element + " ");

}

Console.WriteLine("");

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

int[] all = b.Where(element => c.Contains(element)).ToArray();

if (all.Length == 0)

{

Console.WriteLine("Таких элементов нет");

}

else

{

foreach (var element in all)

{

Console.WriteLine(element);

}

}

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

int[] combinedArray = b.Concat(c).ToArray();

Console.WriteLine("Объединенный массив:");

foreach (var element in combinedArray)

{

Console.Write(element + " ");

}

Console.WriteLine("");

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

if (all.Length == 0)

{

Console.WriteLine("Пересечения нет");

}

else

{

foreach (var element in all)

{

Console.WriteLine(element);

}

}

Console.WriteLine("");

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

var result1 = combinedArray.OrderBy(t => t);

Console.WriteLine("По возрастанию");

foreach (int t in result1)

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

Console.WriteLine("");

Console.WriteLine("к.1-------------------------");

var result2 = combinedArray.OrderByDescending(t => t);

Console.WriteLine("По убыванию");

foreach (int t in result2)

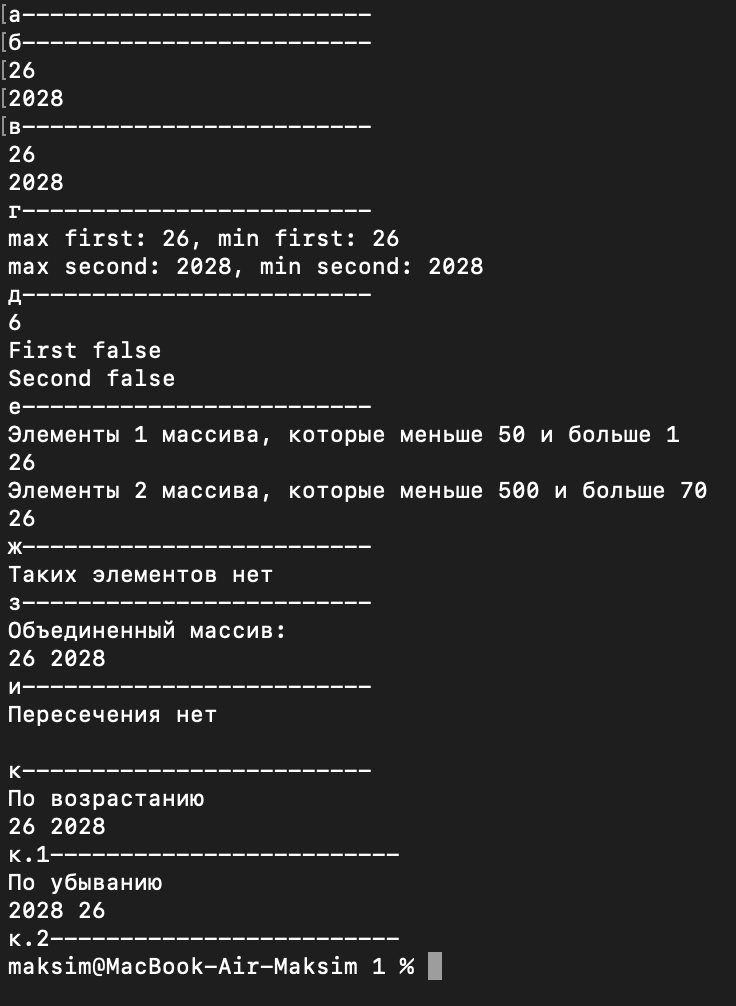
Console.Write($"{t} ");

Console.WriteLine("");

Console.WriteLine("к.2-------------------------");

}

}



Задание 2

using System;

using System.Collections.Generic;

class Program

{

*static* void Main(string[] args)

{

List<Student> students = new List<Student>

{

new Student { studentnumber = 1, Surname = "Sluchaev", name = "Maxim", patronymic = "Konstantinovich", yearbirth = "2005", studentclass = "9V", averagescore = 5.0},

new Student { studentnumber = 2, Surname = "Spiridonov", name = "Evgeniy", patronymic = "Andreevich", yearbirth = "2005", studentclass = "9V", averagescore = 4.99},

new Student { studentnumber = 3, Surname = "Patkov", name = "Dmitry", patronymic = "Andreevich", yearbirth = "2009", studentclass = "9A", averagescore = 4.0},

new Student { studentnumber = 4, Surname = "Shavaliev", name = "Timur", patronymic = "Maratovich", yearbirth = "2005", studentclass = "9V", averagescore = 4.0},

new Student { studentnumber = 5, Surname = "Funtova", name = "Sofi", patronymic = "Pavlovna", yearbirth = "2005", studentclass = "9V", averagescore = 4.4}

};

getinfo(students);

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var ListMax = students.Max(p => p.averagescore);

var ListMin = students.Min(p => p.averagescore);

System.Console.WriteLine($"Максимальный балл {ListMax}");

System.Console.WriteLine($"Минимальный балл {ListMin}");

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var result9V = students

.Where(p => p.studentclass == "9A");

foreach (var student in result9V)

{

Console.WriteLine($"Номер: {student.studentnumber}");

Console.WriteLine($"Фамилия: {student.Surname}");

Console.WriteLine($"Имя: {student.name}");

Console.WriteLine($"Отчество: {student.patronymic}");

Console.WriteLine($"Год рождения: {student.yearbirth}");

Console.WriteLine($"Класс: {student.studentclass}");

Console.WriteLine($"Средний балл: {student.averagescore}");

Console.WriteLine();

}

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var result2009 = students

.Where(p => p.yearbirth == "2009");

foreach (var student in result2009)

{

Console.WriteLine($"Номер: {student.studentnumber}");

Console.WriteLine($"Фамилия: {student.Surname}");

Console.WriteLine($"Имя: {student.name}");

Console.WriteLine($"Отчество: {student.patronymic}");

Console.WriteLine($"Год рождения: {student.yearbirth}");

Console.WriteLine($"Класс: {student.studentclass}");

Console.WriteLine($"Средний балл: {student.averagescore}");

Console.WriteLine();

}

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var gr = students.GroupBy(args => args.studentclass);

foreach (var student in gr)

{

Console.WriteLine($"{student.Key} {student.Average(x => x.averagescore)}");

}

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var FIO = students.Select(p => new { p.Surname, p.studentnumber });

foreach (var student in FIO)

{

Console.WriteLine($"Фамилия: {student.Surname}");

Console.WriteLine($"Номер: {student.studentnumber}");

Console.WriteLine();

}

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var sort = students.OrderBy(args => args.averagescore).ToList();

foreach (var student in sort)

{

Console.WriteLine($"Номер: {student.studentnumber}");

Console.WriteLine($"Фамилия: {student.Surname}");

Console.WriteLine($"Имя: {student.name}");

Console.WriteLine($"Отчество: {student.patronymic}");

Console.WriteLine($"Год рождения: {student.yearbirth}");

Console.WriteLine($"Класс: {student.studentclass}");

Console.WriteLine($"Средний балл: {student.averagescore}");

Console.WriteLine();

}

System.Console.WriteLine($"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

*static* void getinfo(List<Student> students)

{

foreach (Student student in students)

{

Console.WriteLine($"Номер: {student.studentnumber}");

Console.WriteLine($"Фамилия: {student.Surname}");

Console.WriteLine($"Имя: {student.name}");

Console.WriteLine($"Отчество: {student.patronymic}");

Console.WriteLine($"Год рождения: {student.yearbirth}");

Console.WriteLine($"Класс: {student.studentclass}");

Console.WriteLine($"Средний балл: {student.averagescore}");

Console.WriteLine();

}

}

}

class Student

{

*public* int studentnumber { get; set; }

*public* string Surname { get; set; }

*public* string name { get; set; }

*public* string patronymic { get; set; }

*public* string yearbirth { get; set; }

*public* string studentclass { get; set; }

*public* double averagescore { get; set; }

}



