

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **«СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

.

**Отчет по лабораторной работе № 5**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **на тему** | |  | Функции и рекурсия | | | | |
|  | | | | | |  | (наименование темы) |
|  |  | |  | | | | |
| **по дисциплине** | | | |  | Проектирование классов | | |
|  | | | | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Бактияров Белек Бактиярович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | Дкип111 прог |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев И. В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва, 2025**

# Лабораторная работа № 5

Оглавление

[Лабораторная работа № 5 2](#_Toc198809636)

[Задание №1 2](#_Toc198809637)

[Листинг кода 2](#_Toc198809638)

[Скрин кода 4](#_Toc198809639)

[Задание №2 4](#_Toc198809640)

[Листинг кода 4](#_Toc198809641)

[Скрин кода 7](#_Toc198809642)

[Задание №3 7](#_Toc198809643)

[Листинг кода 7](#_Toc198809644)

[Скрин кода 9](#_Toc198809645)

[Задание №4 9](#_Toc198809646)

[Листинг кода 9](#_Toc198809647)

[Скрин кода 11](#_Toc198809648)

# Задание №1

# Листинг кода

**using System;**

**using System.Text.RegularExpressions;**

**class Program**

**{**

**static void Main()**

**{**

**// Эмуляция чтения из файла**

**string text = "Привет, как дела? Сегодня хорошая погода. Я люблю программирование. А ты любишь C#? Программирование — это весело!";**

**// Ввод слова**

**Console.Write("Введите слово для поиска: ");**

**string keyword = Console.ReadLine();**

**// Разделение на предложения**

**string[] sentences = Regex.Split(text, @"(?<=[\.!\?])\s+");**

**Console.WriteLine($"\nПредложения, содержащие слово \"{keyword}\":\n");**

**bool found = false;**

**foreach (string sentence in sentences)**

**{**

**if (Regex.IsMatch(sentence, $@"\b{Regex.Escape(keyword)}\b", RegexOptions.IgnoreCase))**

**{**

**Console.WriteLine(sentence.Trim());**

**found = true;**

**}**

**}**

**if (!found)**

**{**

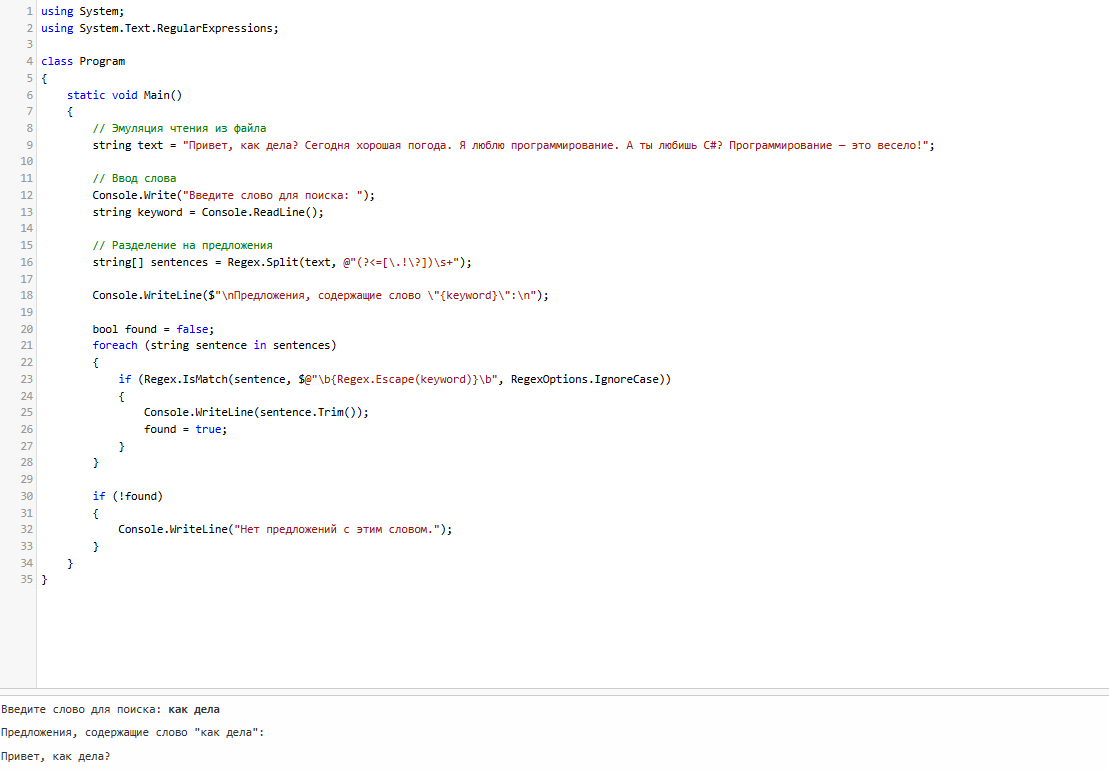
**Console.WriteLine("Нет предложений с этим словом.");**

**}**

**}**

**}**

# Скрин кода



# Задание №2

# Листинг кода

**using System;**

**using System.Collections.Generic;**

**class StringArray**

**{**

**private string[] array;**

**private int lowerBound;**

**private int upperBound;**

**public StringArray(int lowerBound, int upperBound)**

**{**

**if (upperBound < lowerBound)**

**throw new ArgumentException("Верхняя граница должна быть больше или равна нижней.");**

**this.lowerBound = lowerBound;**

**this.upperBound = upperBound;**

**array = new string[upperBound - lowerBound + 1];**

**}**

**public string this[int index]**

**{**

**get**

**{**

**if (index < lowerBound || index > upperBound)**

**throw new IndexOutOfRangeException("Индекс вне диапазона.");**

**return array[index - lowerBound];**

**}**

**set**

**{**

**if (index < lowerBound || index > upperBound)**

**throw new IndexOutOfRangeException("Индекс вне диапазона.");**

**array[index - lowerBound] = value;**

**}**

**}**

**public static StringArray Concat(StringArray a, StringArray b)**

**{**

**int newLower = 0;**

**int newUpper = a.Length + b.Length - 1;**

**StringArray result = new StringArray(newLower, newUpper);**

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

**result[i] = a[a.lowerBound + i];**

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

**result[a.Length + i] = b[b.lowerBound + i];**

**return result;**

**}**

**public static StringArray MergeUnique(StringArray a, StringArray b)**

**{**

**HashSet<string> set = new HashSet<string>();**

**foreach (var item in a.array)**

**set.Add(item);**

**foreach (var item in b.array)**

**set.Add(item);**

**StringArray result = new StringArray(0, set.Count - 1);**

**int i = 0;**

**foreach (var item in set)**

**result[i++] = item;**

**return result;**

**}**

**public int Length => array.Length;**

**public void PrintElement(int index)**

**{**

**Console.WriteLine(this[index]);**

**}**

**public void PrintAll()**

**{**

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

**Console.WriteLine($"[{i + lowerBound}] = {array[i]}");**

**}**

**}**

**class Program**

**{**

**static void Main()**

**{**

**StringArray arr1 = new StringArray(0, 2);**

**arr1[0] = "apple";**

**arr1[1] = "banana";**

**arr1[2] = "cherry";**

**StringArray arr2 = new StringArray(3, 5);**

**arr2[3] = "banana";**

**arr2[4] = "date";**

**arr2[5] = "fig";**

**Console.WriteLine("Массив 1:");**

**arr1.PrintAll();**

**Console.WriteLine("\nМассив 2:");**

**arr2.PrintAll();**

**var concatArray = StringArray.Concat(arr1, arr2);**

**Console.WriteLine("\nКонкатенация:");**

**concatArray.PrintAll();**

**var mergedArray = StringArray.MergeUnique(arr1, arr2);**

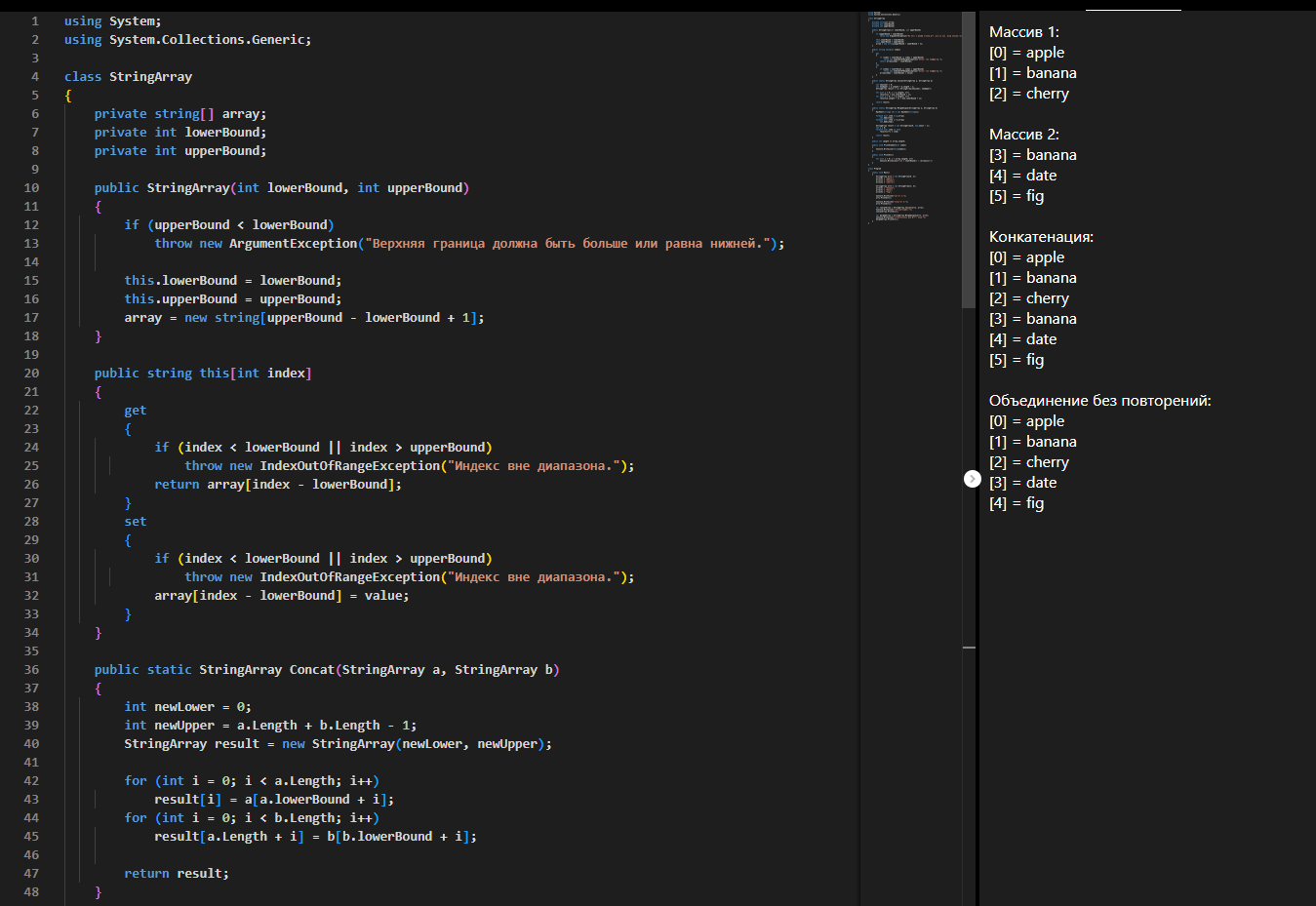
**Console.WriteLine("\nОбъединение без повторений:");**

**mergedArray.PrintAll();**

**}**

**}**

# Скрин кода



# Задание №3

# Листинг кода

**using System;**

**abstract class Vehicle**

**{**

**public double X { get; set; }**

**public double Y { get; set; }**

**public decimal Price { get; set; }**

**public double Speed { get; set; }**

**public int Year { get; set; }**

**public Vehicle(double x, double y, decimal price, double speed, int year)**

**{**

**X = x;**

**Y = y;**

**Price = price;**

**Speed = speed;**

**Year = year;**

**}**

**public abstract void ShowInfo();**

**}**

**class Plane : Vehicle**

**{**

**public double Altitude { get; set; }**

**public int Passengers { get; set; }**

**public Plane(double x, double y, decimal price, double speed, int year, double altitude, int passengers)**

**: base(x, y, price, speed, year)**

**{**

**Altitude = altitude;**

**Passengers = passengers;**

**}**

**public override void ShowInfo()**

**{**

**Console.WriteLine($"Самолет: координаты ({X}, {Y}), цена {Price}, скорость {Speed}, год {Year}, высота {Altitude}, пассажиров {Passengers}");**

**}**

**}**

**class Car : Vehicle**

**{**

**public Car(double x, double y, decimal price, double speed, int year)**

**: base(x, y, price, speed, year) { }**

**public override void ShowInfo()**

**{**

**Console.WriteLine($"Автомобиль: координаты ({X}, {Y}), цена {Price}, скорость {Speed}, год {Year}");**

**}**

**}**

**class Ship : Vehicle**

**{**

**public string Port { get; set; }**

**public int Passengers { get; set; }**

**public Ship(double x, double y, decimal price, double speed, int year, string port, int passengers)**

**: base(x, y, price, speed, year)**

**{**

**Port = port;**

**Passengers = passengers;**

**}**

**public override void ShowInfo()**

**{**

**Console.WriteLine($"Корабль: координаты ({X}, {Y}), цена {Price}, скорость {Speed}, год {Year}, порт {Port}, пассажиров {Passengers}");**

**}**

**}**

**class Program**

**{**

**static void Main()**

**{**

**Plane plane = new Plane(10, 20, 1000000m, 900, 2020, 12000, 150);**

**Car car = new Car(5, 5, 30000m, 200, 2022);**

**Ship ship = new Ship(0, 0, 500000m, 100, 2018, "Одесса", 300);**

**plane.ShowInfo();**

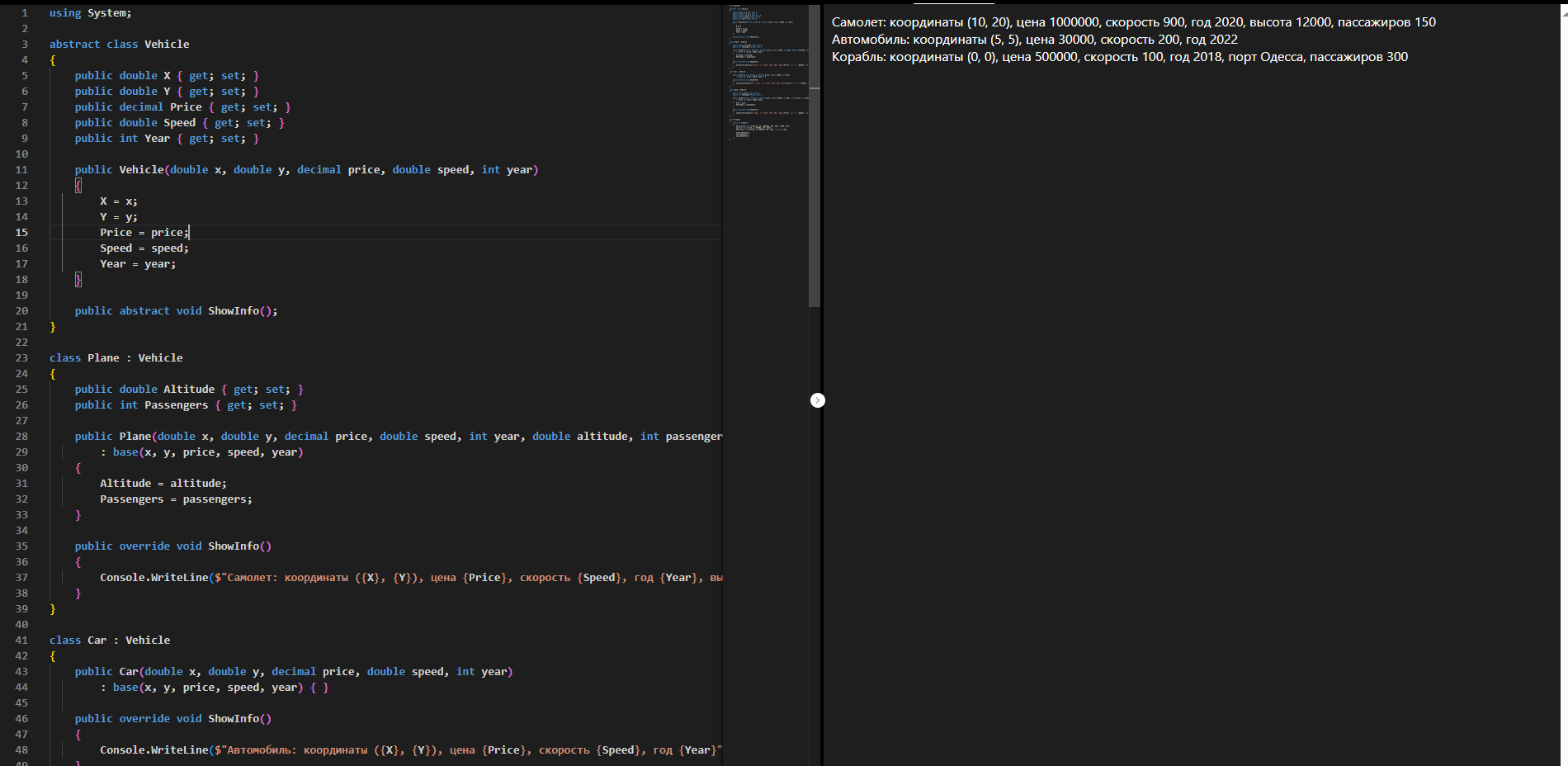
**car.ShowInfo();**

**ship.ShowInfo();**

**}**

**}**

# Скрин кода



# Задание №4

# Листинг кода

**using System;**

**using System.Linq;**

**struct STUDENT**

**{**

**public string Name;**

**public string Group;**

**public int[] Grades;**

**public double GetAverage()**

**{**

**return Grades.Average();**

**}**

**public bool HasOnlyGoodGrades()ыы**

**{**

**return Grades.All(g => g == 4 || g == 5);**

**}**

**}**

**class Program**

**{**

**static void Main()**

**{**

**STUDENT[] students = new STUDENT[]**

**{**

**new STUDENT { Name = "Иванов И.И.", Group = "A1", Grades = new int[] {5, 4, 5, 4, 5} },**

**new STUDENT { Name = "Петров П.П.", Group = "B2", Grades = new int[] {3, 4, 5, 3, 4} },**

**new STUDENT { Name = "Сидоров С.С.", Group = "A1", Grades = new int[] {4, 4, 4, 4, 4} },**

**new STUDENT { Name = "Кузнецов К.К.", Group = "C3", Grades = new int[] {2, 3, 4, 3, 3} },**

**new STUDENT { Name = "Николаев Н.Н.", Group = "A1", Grades = new int[] {5, 5, 5, 5, 5} },**

**new STUDENT { Name = "Федоров Ф.Ф.", Group = "B2", Grades = new int[] {4, 4, 5, 4, 5} },**

**new STUDENT { Name = "Андреев А.А.", Group = "C3", Grades = new int[] {3, 4, 3, 3, 4} },**

**new STUDENT { Name = "Морозов М.М.", Group = "A1", Grades = new int[] {4, 4, 4, 5, 4} },**

**new STUDENT { Name = "Волков В.В.", Group = "B2", Grades = new int[] {2, 2, 3, 2, 3} },**

**new STUDENT { Name = "Григорьев Г.Г.", Group = "C3", Grades = new int[] {5, 5, 4, 5, 5} }**

**};**

**// Сортировка по среднему баллу**

**var sorted = students.OrderBy(s => s.GetAverage()).ToArray();**

**Console.WriteLine("Студенты с оценками только 4 и 5:");**

**bool found = false;**

**foreach (var student in sorted)**

**{**

**if (student.HasOnlyGoodGrades())**

**{**

**Console.WriteLine($"Фамилия: {student.Name}, Группа: {student.Group}");**

**found = true;**

**}**

**}**

**if (!found)**

**{**

**Console.WriteLine("Нет студентов с оценками только 4 и 5.");**

**}**

**}**

**}**

# Скрин кода

