**ЗАДАНИЕ 19**

Задание 1. Описать класс для работы с двумерным массивом целых чисел.

Реализовать возможность уменьшения количества столбцов массива на

заданное число (перегрузка операции --) с удалением их с начала массива.

Листинг программы:

/// <summary>

/// Эта программа читает список чисел из текстового файла, находит минимальное число и считает количество положительных элементов.

/// </summary>

using System;

using System.IO;

using System.Collections.Generic;

class Program

{

/// <summary>

/// Точка входа в программу.

/// </summary>

/// <param name="args">Аргументы командной строки (не используются в этой программе).</param>

static void Main(string[] args)

{

try

{

/// <summary>

/// Имя файла по умолчанию.

/// </summary>

string filename = "lab16.txt";

/// <summary>

/// Список прочитанных чисел из файла.

/// </summary>

List<double> numbers = ReadNumbersFromFile(filename);

/// <summary>

/// Результат поиска минимального значения и количества положительных элементов.

/// </summary>

(double min, int positiveCount) = FindMinAndCountPositives(numbers);

Console.WriteLine($"Минимальное число в файле: {min}");

Console.WriteLine($"Количество положительных элементов: {positiveCount}");

}

catch (FileNotFoundException e)

{

Console.WriteLine($"Файл '{e.FileName}' не найден.");

}

catch (Exception e)

{

Console.WriteLine($"Ошибка: {e.Message}");

}

}

/// <summary>

/// Запрашивает у пользователя имя файла.

/// </summary>

/// <returns>Имя файла, введенное пользователем.</returns>

static string GetInputFilename()

{

Console.Write("Введите имя файла: ");

return Console.ReadLine();

}

/// <summary>

/// Читает список чисел из текстового файла.

/// </summary>

/// <param name="filename">Имя файла для чтения.</param>

/// <returns>Список прочитанных чисел из файла.</returns>

/// <exception cref="FileNotFoundException">Выбрасывается, если файл не найден.</exception>

static List<double> ReadNumbersFromFile(string filename)

{

List<double> numbers = new List<double>();

using (StreamReader reader = new StreamReader(filename))

{

string line;

while ((line = reader.ReadLine()) != null)

{

line = line.Trim();

if (line.Length > 0 && line.All(char.IsDigit))

{

numbers.Add(double.Parse(line));

}

}

}

return numbers;

}

/// <summary>

/// Находит минимальное число и считает количество положительных элементов в списке чисел.

/// </summary>

/// <param name="numbers">Список чисел для обработки.</param>

/// <returns>Кортеж, содержащий минимальное значение и количество положительных элементов.</returns>

/// <exception cref="ArgumentException">Выбрасывается, если список чисел пуст.</exception>

static (double min, int positiveCount) FindMinAndCountPositives(List<double> numbers)

{

if (numbers.Count == 0)

{

throw new ArgumentException("Список чисел пуст.");

}

double min = double.MaxValue;

int positiveCount = 0;

foreach (double number in numbers)

{

min = Math.Min(min, number);

if (number > 0)

{

positiveCount++;

}

}

return (min, positiveCount);

}

}

Задание 2. Используя Visual Studio, создайте проект по шаблону Console Application. Создайте анонимный метод, который принимает в качестве аргумента массив делегатов и возвращает среднее арифметическое возвращаемых значений методов сообщенных с делегатами в массиве. Методы, сообщенные с делегатами из массива, возвращают случайное значение типа int.

Листинг программы:

/// <summary>

/// Эта программа демонстрирует использование делегатов и анонимных методов для вычисления среднего значения случайных чисел, генерируемых несколькими методами.

/// </summary>

using System;

class Program

{

/// <summary>

/// Тип делегата, представляющий метод, генерирующий случайное целое число от 1 до 100 (включительно).

/// </summary>

delegate int RandomNumberMethod();

static void Main()

{

/// <summary>

/// Массив для хранения экземпляров делегатов, представляющих методы генерации случайных чисел.

/// </summary>

RandomNumberMethod[] methods = new RandomNumberMethod[5];

/// <summary>

/// Экземпляр класса Random для генерации случайных чисел.

/// </summary>

Random random = new Random();

/// <summary>

/// Инициализация массива методов анонимными методами, которые используют Random.Next() для генерации случайных чисел.

/// </summary>

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

{

methods[i] = delegate () { return random.Next(1, 100); };

}

/// <summary>

/// Вычисляет среднее значение случайных чисел, генерируемых методами в массиве.

/// </summary>

double average = CalculateAverage(methods);

Console.WriteLine("Среднее значение (Average value): " + average);

}

/// <summary>

/// Вычисляет среднее значение случайных чисел, генерируемых методами делегатов в предоставленном массиве.

/// </summary>

/// <param name="methods">Массив экземпляров делегатов, представляющих методы генерации случайных чисел.</param>

/// <returns>Среднее значение в виде double.</returns>

static double CalculateAverage(RandomNumberMethod[] methods)

{

int sum = 0;

foreach (var method in methods)

{

sum += method();

}

return (double)sum / methods.Length;

}

}