Министерство образования Республики Беларусь

Оршанский колледж учреждения образования

«Витебский государственный университет имени П. М. Машерова»

Специальность 5-04-0612-02

«Разработка и сопровождение программного обеспечения информационных систем»

Группа 3ПОИС23

УЧЕБНАЯ ПРАКТИКА

ПО ПРОГРАММИРОВАНИЮ

с 29 сентября 2025 года по 25 октября 2025 года

ОТЧЕТ

Выполнил Трубочкин И.А.

Руководитель практики Алейников М.А.

2025

**СОДЕРЖАНИЕ**

## Модуль 1.1 Основы C#

## Модуль 2. Объектно-ориентированное программирование (ООП)

## Модуль 3. Задания по делегатам

## Модуль 4. Задания по интерфейсам

## Модуль 5. Графический интерфейс

## Модуль 6. Базы данных

## Модуль 7. Проектная работа

## Модуль 8. Реализация проекта

## Постановка задачи

## Описание предметной области

## Характеристика решаемой задачи

## Разработка моделей решаемой задачи

## Проектирование программного модуля

## Сбор исходных материалов

## Проектирование информационной модели

## Описание входных и выходных данных

## Реализация программного модуля

## Описание диаграммы классов разрабатываемого проекта

## Описание структуры разрабатываемого проекта

## Проектирование и реализация интерфейса программы

## Модуль 9. Совершенствование навыков ООП

## ЗАКЛЮЧЕНИЕ

## Список использованных источников

## Приложение А Текст программы модуля 8

## Модуль 1.1 Основы C#

Вариант 1

Выполнение:

Задание 1. Реализуйте приложение для вычисления факториала числа.

Листинг 1. Код программы на C#

using System;

class SimpleFactorial

{

static void Main()

{

while (true)

{

Console.Write("\nВведите число: ");

string input = Console.ReadLine();

if (input == "exit")

break;

if (int.TryParse(input, out int number))

{

if (number < 0)

{

Console.WriteLine("Факториал отрицательного числа не существует!");

}

else if (number > 20)

{

Console.WriteLine("Попробуйте число до 20.");

}

else

{

long factorial = 1;

for (int i = 1; i <= number; i++)

{

factorial \*= i;

}

Console.WriteLine($"Факториал {number} = {factorial}");

}

}

else

{

Console.WriteLine("Пожалуйста, введите целое число!");

}

}

}

}

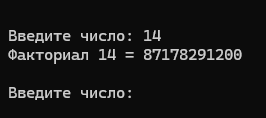


Рисунок 1.1.1 – Выполнение кода листинга 1

Задание 2. Напишите программу, которая запрашивает у пользователя два целых числа и выводит их сумму.

Листинг 2. Код программы на C#

using System;

class Program

{

static void Main()

{

Console.Write("Введите первое число: ");

string input1 = Console.ReadLine();

Console.Write("Введите второе число: ");

string input2 = Console.ReadLine();

if (int.TryParse(input1, out int number1) && int.TryParse(input2, out int number2))

{

int sum = number1 + number2;

Console.WriteLine($"\nРезультат: {number1} + {number2} = {sum}");

}

else

{

Console.WriteLine("\nОшибка: Пожалуйста, введите корректные целые числа!");

}

Console.WriteLine("\nНажмите любую клавишу для выхода...");

Console.ReadKey();

}

}

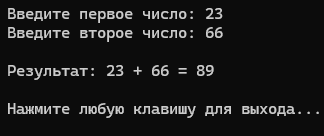


Рисунок 1.1.2 – Выполнение кода листинга 2

Задание 3. Напишите программу, которая принимает на вход строку и выводит ее в обратном порядке.

Листинг 3. Код программы на C#

using System;

class Program

{

static void Main()

{

Console.Write("Введите строку: ");

string input = Console.ReadLine();

if (string.IsNullOrEmpty(input))

{

Console.WriteLine("Вы ввели пустую строку!");

return;

}

string reversed = ReverseString(input);

Console.WriteLine($"\nОригинал: {input}");

Console.WriteLine($"Реверс: {reversed}");

}

static string ReverseString(string str)

{

char[] charArray = str.ToCharArray();

Array.Reverse(charArray);

return new string(charArray);

}

}

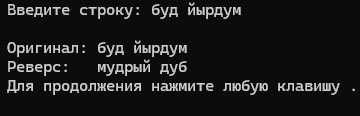


Рисунок 1.1.3 – Выполнение кода листинга 3

Задание 4. Напишите программу, которая создает массив из 15 случайных чисел и находит среднее значение всех положительных чисел в массиве.

Листинг 4. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

int[] numbers = GenerateRandomArray(15, -100, 100);

PrintArray("Сгенерированный массив", numbers);

int[] positiveNumbers = numbers.Where(n => n > 0).ToArray();

PrintArray("Положительные числа", positiveNumbers);

if (positiveNumbers.Length > 0)

{

double average = positiveNumbers.Average();

Console.WriteLine($"\n Результаты:");

Console.WriteLine($"Количество положительных чисел: {positiveNumbers.Length}");

Console.WriteLine($"Сумма положительных чисел: {positiveNumbers.Sum()}");

Console.WriteLine($"Среднее значение: {average:F2}");

}

else

{

Console.WriteLine("\n В массиве нет положительных чисел!");

}

}

static int[] GenerateRandomArray(int size, int minValue, int maxValue)

{

Random random = new Random();

int[] array = new int[size];

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

{

array[i] = random.Next(minValue, maxValue + 1);

}

return array;

}

static void PrintArray(string title, int[] array)

{

Console.WriteLine($"\n{title}:");

Console.WriteLine(string.Join(", ", array));

}

}

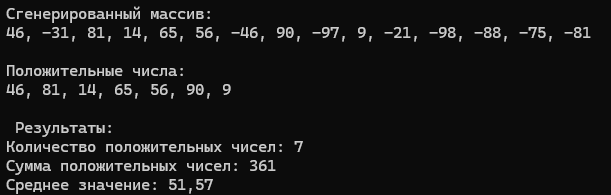


Рисунок 1.1.4 – Выполнение кода листинга 4

Задание 5. Напишите программу, которая проверяет, является ли введенное пользователем число простым (не имеет делителей, кроме 1 и самого себя).

Листинг 5. Код программы на C#

using System;

class Program

{

static void Main()

{

while (true)

{

Console.Write("Введите число: ");

string input = Console.ReadLine();

if (input?.ToLower() == "exit")

break;

if (int.TryParse(input, out int number))

{

if (number < 2)

{

Console.WriteLine($" Число {number} не является простым (простые числа ≥ 2)");

}

else if (IsPrime(number))

{

Console.WriteLine($" Число {number} - Простое");

}

else

{

Console.WriteLine($" Число {number} - Составное");

}

}

else

{

Console.WriteLine(" Ошибка: Введите целое число!");

}

Console.WriteLine();

}

}

static bool IsPrime(int n)

{

if (n == 2 || n == 3)

return true;

if (n % 2 == 0 || n % 3 == 0)

return false;

for (int i = 5; i \* i <= n; i += 6)

{

if (n % i == 0 || n % (i + 2) == 0)

return false;

}

return true;

}

}

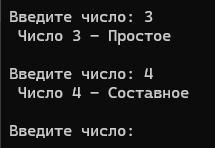


Рисунок 1.1.5 – Выполнение кода листинга 5

Модуль 1.2 Массивы, Строки, Переключатели

Задача 1. Ввести размер массива N и значения его элементов. Нормировать элементы массива, разделив их на значение максимального по модулю элемента. Вывести значения элементов измененного массива.

Выполнение:

Листинг 1. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

int n;

while (true)

{

Console.Write("Введите размер массива N: ");

if (int.TryParse(Console.ReadLine(), out n) && n > 0)

{

break;

}

Console.WriteLine("Ошибка: введите целое положительное число!");

}

double[] array = new double[n];

Console.WriteLine($"Введите {n} элементов массива:");

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

{

while (true)

{

Console.Write($"Элемент [{i}]: ");

if (double.TryParse(Console.ReadLine(), out array[i]))

{

break;

}

Console.WriteLine("Ошибка: введите число!");

}

}

Console.WriteLine("\nИсходный массив:");

PrintArray(array);

double[] normalizedArray = NormalizeArray(array);

Console.WriteLine("\nНормированный массив:");

PrintArray(normalizedArray);

Console.WriteLine($"\nМаксимальный по модулю элемент: {FindMaxAbsolute(array)}");

}

static double[] NormalizeArray(double[] array)

{

double maxAbsolute = FindMaxAbsolute(array);

if (maxAbsolute == 0)

return (double[])array.Clone();

double[] result = new double[array.Length];

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

{

result[i] = array[i] / maxAbsolute;

}

return result;

}

static double FindMaxAbsolute(double[] array)

{

double max = 0;

foreach (double element in array)

{

double absolute = Math.Abs(element);

if (absolute > max)

max = absolute;

}

return max;

}

static void PrintArray(double[] array)

{

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

{

Console.WriteLine($"[{i}] = {array[i]:F6}");

}

}

}

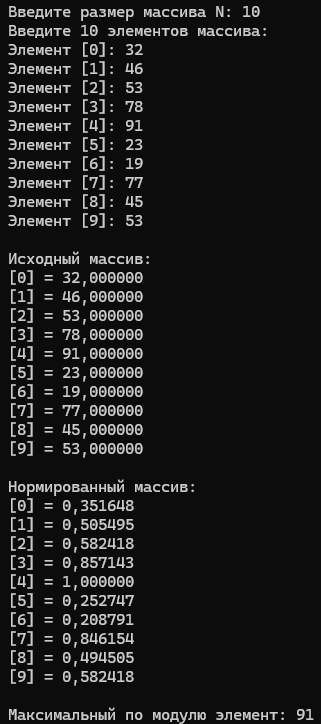


Рисунок 1.2.1 – Выполнение кода листинга 1

Задание 2. Определить и инициализировать целочисленный массив из 10-ти элементов. Ввести целое число и заменить им значение максимального элемента в массиве.

Выполнение:

Листинг 2. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

int[] array = { 15, 7, 23, 9, 42, 18, 31, 5, 27, 12 };

Console.WriteLine("Исходный массив:");

PrintArray(array);

int maxValue = array.Max();

int maxIndex = Array.IndexOf(array, maxValue);

Console.WriteLine($"\nМаксимальный элемент: array[{maxIndex}] = {maxValue}");

int newNumber;

while (true)

{

Console.Write("\nВведите целое число для замены: ");

if (int.TryParse(Console.ReadLine(), out newNumber))

{

break;

}

Console.WriteLine("Ошибка: введите целое число!");

}

array[maxIndex] = newNumber;

Console.WriteLine("\nИзмененный массив:");

PrintArray(array);

Console.WriteLine($"\nМаксимальный элемент заменен на: {newNumber}");

}

static void PrintArray(int[] array)

{

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

{

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

}

}

}

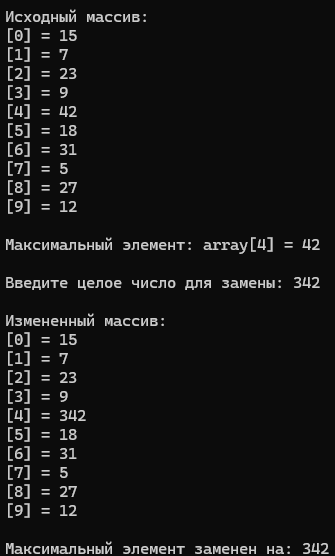


Рисунок 1.2.2 – Выполнение кода листинга 2

Задание 3. Вычислить К простых чисел. Значение К ввести с клавиатуры. Вывести значения чисел, размещая их по 10 на строке.

Выполнение:

Листинг 3. Код программы на C#

using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

int k;

while (true)

{

Console.Write("Введите количество простых чисел K: ");

if (int.TryParse(Console.ReadLine(), out k) && k > 0)

{

break;

}

Console.WriteLine("Ошибка: введите целое положительное число!");

}

List<int> primes = GeneratePrimes(k);

Console.WriteLine($"\nПервые {k} простых чисел:");

PrintPrimesFormatted(primes);

Console.WriteLine($"\n Статистика:");

Console.WriteLine($"Последнее простое число: {primes[primes.Count - 1]}");

Console.WriteLine($"Сумма всех чисел: {SumPrimes(primes):N0}");

}

static List<int> GeneratePrimes(int k)

{

List<int> primes = new List<int>();

int number = 2;

while (primes.Count < k)

{

if (IsPrime(number))

{

primes.Add(number);

}

number++;

}

return primes;

}

static bool IsPrime(int n)

{

if (n < 2) return false;

if (n == 2) return true;

if (n % 2 == 0) return false;

for (int i = 3; i \* i <= n; i += 2)

{

if (n % i == 0)

return false;

}

return true;

}

static void PrintPrimesFormatted(List<int> primes)

{

for (int i = 0; i < primes.Count; i++)

{

Console.Write($"{primes[i],6}");

if ((i + 1) % 10 == 0)

Console.WriteLine();

}

if (primes.Count % 10 != 0)

Console.WriteLine();

}

static long SumPrimes(List<int> primes)

{

long sum = 0;

foreach (int prime in primes)

{

sum += prime;

}

return sum;

}

}

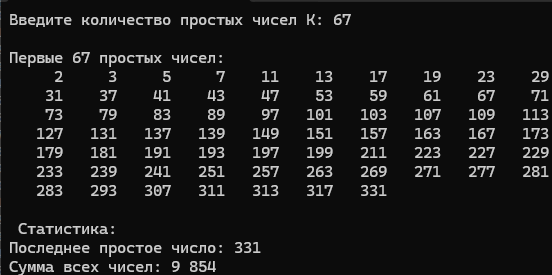


Рисунок 1.2.3 – Выполнение кода листинга 3

Задание 4. Определить целочисленный массив из К элементов. Присвоить элементам случайные значения из диапазона [А,В). Найти индексы минимального и максимального элементов массива. Вывести значения элементов, расположенных между найденными (включая найденные)

Выполнение:

Листинг 4. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

int k = GetPositiveInteger("Введите размер массива K: ");

int a = GetInteger("Введите начало диапазона A: ");

int b = GetInteger("Введите конец диапазона B: ");

if (b <= a)

{

Console.WriteLine("Ошибка: B должно быть больше A!");

return;

}

int[] array = GenerateRandomArray(k, a, b);

Console.WriteLine("\n Исходный массив:");

PrintArray(array);

int minIndex = FindMinIndex(array);

int maxIndex = FindMaxIndex(array);

Console.WriteLine($"\n Минимальный элемент: array[{minIndex}] = {array[minIndex]}");

Console.WriteLine($" Максимальный элемент: array[{maxIndex}] = {array[maxIndex]}");

PrintElementsBetween(array, minIndex, maxIndex);

}

static int GetPositiveInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int n) && n > 0)

return n;

Console.WriteLine(" Ошибка: введите целое положительное число!");

}

}

static int GetInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int n))

return n;

Console.WriteLine(" Ошибка: введите целое число!");

}

}

static int[] GenerateRandomArray(int size, int min, int max)

{

Random random = new Random();

int[] array = new int[size];

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

{

array[i] = random.Next(min, max);

}

return array;

}

static int FindMinIndex(int[] array)

{

int minIndex = 0;

for (int i = 1; i < array.Length; i++)

{

if (array[i] < array[minIndex])

minIndex = i;

}

return minIndex;

}

static int FindMaxIndex(int[] array)

{

int maxIndex = 0;

for (int i = 1; i < array.Length; i++)

{

if (array[i] > array[maxIndex])

maxIndex = i;

}

return maxIndex;

}

static void PrintArray(int[] array)

{

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

{

Console.WriteLine($"[{i,2}] = {array[i],4}");

}

}

static void PrintElementsBetween(int[] array, int index1, int index2)

{

int start = Math.Min(index1, index2);

int end = Math.Max(index1, index2);

Console.WriteLine($"\n Элементы между индексами {start} и {end}:");

for (int i = start; i <= end; i++)

{

Console.WriteLine($"[{i,2}] = {array[i],4}");

}

Console.WriteLine($"\n Количество элементов: {end - start + 1}");

Console.WriteLine($" Диапазон значений: от {array[start]} до {array[end]}");

}

}

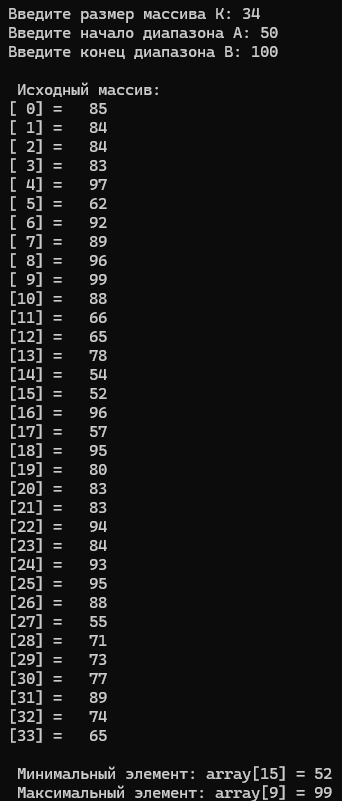


Рисунок 1.2.5 – Выполнение кода листинга 4

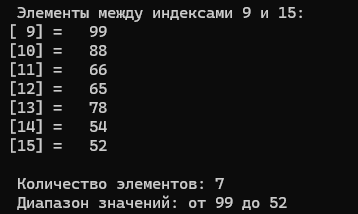


Рисунок 1.2.6 – Выполнение кода листинга 4

Задание 5. Определить символьный массив из К элементов. Присвоить элементам случайные значения букв русского алфавита. Создать новый массив, поместив в него только согласные буквы из первого массива. Значение К ввести с клавиатуры. Вывести элементы обоих массивов.

Выполнение:

Листинг 5. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

int k = GetPositiveInteger("Введите размер массива K: ");

char[] originalArray = GRussianArray(k);

char[] consonantsArray = FilterConsonants(originalArray);

Console.WriteLine("\n Исходный массив:");

PrintCharArray(originalArray);

Console.WriteLine("\n Массив согласных букв:");

PrintCharArray(consonantsArray);

Console.WriteLine($"\n Статистика:");

Console.WriteLine($"Всего элементов: {originalArray.Length}");

Console.WriteLine($"Согласных букв: {consonantsArray.Length}");

Console.WriteLine($"Гласных букв: {CountVowels(originalArray)}");

}

static int GetPositiveInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int n) && n > 0)

return n;

Console.WriteLine(" Ошибка: введите целое положительное число!");

}

}

static char[] GRussianArray(int size)

{

char[] russianAlphabet = {

'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М',

'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ',

'Ы', 'Ь', 'Э', 'Ю', 'Я',

'а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м',

'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ',

'ы', 'ь', 'э', 'ю', 'я'

};

Random random = new Random();

char[] array = new char[size];

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

{

array[i] = russianAlphabet[random.Next(russianAlphabet.Length)];

}

return array;

}

static char[] FilterConsonants(char[] array)

{

char[] vowels = { 'А', 'Е', 'Ё', 'И', 'О', 'У', 'Ы', 'Э', 'Ю', 'Я',

'а', 'е', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю', 'я' };

return array.Where(c => !vowels.Contains(c)).ToArray();

}

static void PrintCharArray(char[] array)

{

if (array.Length == 0)

{

Console.WriteLine("Массив пуст");

return;

}

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

{

Console.Write($"{array[i]} ");

if ((i + 1) % 15 == 0)

Console.WriteLine();

}

if (array.Length % 15 != 0)

Console.WriteLine();

}

static int CountVowels(char[] array)

{

char[] vowels = { 'А', 'Е', 'Ё', 'И', 'О', 'У', 'Ы', 'Э', 'Ю', 'Я',

'а', 'е', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю', 'я' };

return array.Count(c => vowels.Contains(c));

}

}

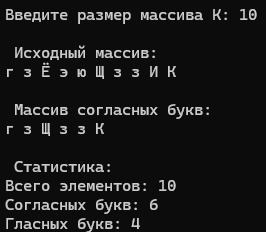


Рисунок 1.2.7 – Выполнение кода листинга 5

Задание 6. Определить вещественный массив из 10-ти элементов. Присвоить элементам случайные значения из диапазона [–10, 10). Сформировать массив индексов, которые нумеруют элементы первого массива в порядке возрастания их значений.

Выполнение:

Листинг 6. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

double[] array = GenerateRandomArray(10, -10, 10);

Console.WriteLine("\n Исходный массив:");

PrintArray(array);

int[] sortedIndices = GetSortedIndices(array);

Console.WriteLine("\n Отсортированные индексы по возрастанию значений:");

PrintSortedResults(array, sortedIndices);

}

static double[] GenerateRandomArray(int size, double min, double max)

{

Random random = new Random();

double[] array = new double[size];

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

{

array[i] = min + (random.NextDouble() \* (max - min));

}

return array;

}

static int[] GetSortedIndices(double[] array)

{

int[] indices = new int[array.Length];

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

{

indices[i] = i;

}

Array.Sort(indices, (a, b) => array[a].CompareTo(array[b]));

return indices;

}

static void PrintArray(double[] array)

{

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

{

Console.WriteLine($"[{i,2}] = {array[i],8:F3}");

}

}

static void PrintSortedResults(double[] array, int[] sortedIndices)

{

Console.WriteLine("Индекс | Значение");

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

foreach (int index in sortedIndices)

{

Console.WriteLine($"{index,6} | {array[index],8:F3}");

}

}

}

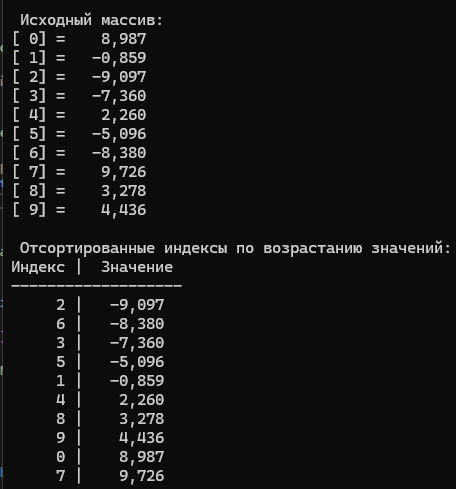


Рисунок 1.2.8 – Выполнение кода листинга 6

Модуль 1.3 Статические методы (Методы классов)

Задание 1. Определить функцию (статический метод) для вычисления наибольшего общего делителя двух целых натуральных чисел (Greatest Common Measure). В основной программе, используя функцию, сократить неотрицательную обыкновенную дробь. Дробь вводится с клавиатуры в виде неотрицательного числителя и положительного знаменателя.

Листинг 1. Код программы на C#

using System;

class Program

{

static void Main()

{

int numerator = GetNonNegativeInteger("Введите неотрицательный числитель: ");

int denominator = GetPositiveInteger("Введите положительный знаменатель: ");

Console.WriteLine($"\nИсходная дробь: {numerator}/{denominator}");

(int simplifiedNumerator, int simplifiedDenominator) = SimplifyFraction(numerator, denominator);

Console.WriteLine($"Сокращенная дробь: {simplifiedNumerator}/{simplifiedDenominator}");

if (simplifiedNumerator >= simplifiedDenominator && simplifiedDenominator != 1)

{

int wholePart = simplifiedNumerator / simplifiedDenominator;

int remainder = simplifiedNumerator % simplifiedDenominator;

if (remainder != 0)

{

Console.WriteLine($"В виде смешанного числа: {wholePart} {remainder}/{simplifiedDenominator}");

}

else

{

Console.WriteLine($"Целое число: {wholePart}");

}

}

double decimalValue = (double)numerator / denominator;

Console.WriteLine($"Десятичное представление: {decimalValue:F6}");

}

static int GCD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return a;

}

static (int numerator, int denominator) SimplifyFraction(int numerator, int denominator)

{

if (numerator == 0)

return (0, 1);

int gcd = GCD(numerator, denominator);

int simplifiedNumerator = numerator / gcd;

int simplifiedDenominator = denominator / gcd;

return (simplifiedNumerator, simplifiedDenominator);

}

static int GetNonNegativeInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int number) && number >= 0)

return number;

Console.WriteLine("Ошибка: введите неотрицательное целое число!");

}

}

static int GetPositiveInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int number) && number > 0)

return number;

Console.WriteLine("Ошибка: введите положительное целое число!");

}

}

}

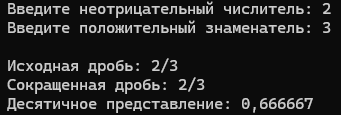


Рисунок 1.3.1 – Выполнение кода листинга 1

Задание 2. Присваивая последовательным элементам массива случайные значения от 1 до 9, создать массив с минимальным количеством элементов, сумма которых не превышает заданного пользователем числа.

Листинг 2. Код программы на C#

using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main()

{

int targetSum = GetPositiveInteger("Введите максимальную сумму элементов: ");

var result = GenerateArrayWithSumLimit(targetSum);

int[] array = result.Item1;

int actualSum = result.Item2;

int count = result.Item3;

Console.WriteLine("\nРезультат:");

Console.WriteLine("Количество элементов: " + count);

Console.WriteLine("Сумма элементов: " + actualSum);

Console.WriteLine("Целевая сумма: " + targetSum);

Console.WriteLine("Разница: " + (targetSum - actualSum));

PrintArray(array, count);

PrintStatistics(array, count, targetSum);

}

static Tuple<int[], int, int> GenerateArrayWithSumLimit(int targetSum)

{

Random random = new Random();

List<int> elements = new List<int>();

int currentSum = 0;

int count = 0;

while (currentSum < targetSum)

{

int randomValue = random.Next(1, 10);

int potentialSum = currentSum + randomValue;

if (potentialSum > targetSum)

break;

elements.Add(randomValue);

currentSum = potentialSum;

count++;

}

return Tuple.Create(elements.ToArray(), currentSum, count);

}

static void PrintArray(int[] array, int count)

{

Console.WriteLine("\nМассив элементов:");

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

{

Console.Write(array[i] + " ");

if ((i + 1) % 15 == 0)

Console.WriteLine();

}

Console.WriteLine();

var subArray = new List<int>();

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

{

subArray.Add(array[i]);

}

Console.Write("Сумма: " + string.Join(" + ", subArray) + " = " + subArray.Sum());

Console.WriteLine();

}

static void PrintStatistics(int[] array, int count, int targetSum)

{

if (count == 0)

{

Console.WriteLine("Не удалось добавить ни одного элемента!");

return;

}

int[] usedArray = new int[count];

Array.Copy(array, usedArray, count);

int sum = usedArray.Sum();

double average = (double)sum / count;

int min = usedArray.Min();

int max = usedArray.Max();

Console.WriteLine("\nСтатистика:");

Console.WriteLine("Эффективность использования суммы: " + ((double)sum / targetSum \* 100).ToString("F1") + "%");

Console.WriteLine("Среднее значение элемента: " + average.ToString("F2"));

Console.WriteLine("Минимальный элемент: " + min);

Console.WriteLine("Максимальный элемент: " + max);

// Распределение значений

var valueCounts = new Dictionary<int, int>();

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

{

if (valueCounts.ContainsKey(array[i]))

valueCounts[array[i]]++;

else

valueCounts[array[i]] = 1;

}

Console.WriteLine("\nРаспределение значений:");

for (int i = 1; i <= 9; i++)

{

int freq = valueCounts.ContainsKey(i) ? valueCounts[i] : 0;

double percentage = (double)freq / count \* 100;

Console.WriteLine(" " + i + ": " + freq + " раз (" + percentage.ToString("F1") + "%)");

}

}

static int GetPositiveInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int number) && number > 0)

return number;

Console.WriteLine("Ошибка: введите положительное целое число!");

}

}

}

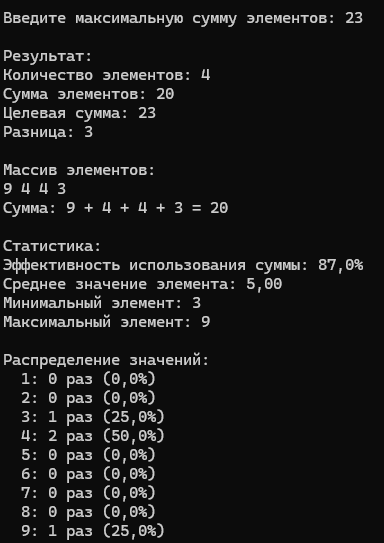


Рисунок 1.3.2 – Выполнение кода листинга 2

Задание 3. Сформировав квадратную целочисленную матрицу со случайными значениями элементов, упорядочить ее строки по возрастанию сумм их элементов. Пусть элементы принимают значения в диапазоне от –50 до +50.

Листинг 3. Код программы на C#

using System;

using System.Linq;

class Program

{

static void Main()

{

int size = GetPositiveInteger("Введите размер квадратной матрицы: ");

int[,] matrix = GenerateMatrix(size, -50, 50);

Console.WriteLine("\nИсходная матрица:");

PrintMatrix(matrix);

int[] rowSums = CalculateRowSums(matrix);

Console.WriteLine("\nСуммы строк исходной матрицы:");

PrintRowSums(rowSums);

int[,] sortedMatrix = SortMatrixByRowSums(matrix);

Console.WriteLine("\nОтсортированная матрица:");

PrintMatrix(sortedMatrix);

int[] sortedRowSums = CalculateRowSums(sortedMatrix);

Console.WriteLine("\nСуммы строк отсортированной матрицы:");

PrintRowSums(sortedRowSums);

PrintStatistics(matrix, sortedMatrix);

}

static int[,] GenerateMatrix(int size, int minValue, int maxValue)

{

Random random = new Random();

int[,] matrix = new int[size, size];

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

{

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

{

matrix[i, j] = random.Next(minValue, maxValue + 1);

}

}

return matrix;

}

static int[] CalculateRowSums(int[,] matrix)

{

int size = matrix.GetLength(0);

int[] sums = new int[size];

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

{

int sum = 0;

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

{

sum += matrix[i, j];

}

sums[i] = sum;

}

return sums;

}

static int[,] SortMatrixByRowSums(int[,] matrix)

{

int size = matrix.GetLength(0);

int[,] sortedMatrix = new int[size, size];

int[] rowSums = CalculateRowSums(matrix);

int[] indices = new int[size];

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

{

indices[i] = i;

}

for (int i = 0; i < size - 1; i++)

{

for (int j = 0; j < size - i - 1; j++)

{

if (rowSums[indices[j]] > rowSums[indices[j + 1]])

{

int temp = indices[j];

indices[j] = indices[j + 1];

indices[j + 1] = temp;

}

}

}

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

{

int originalRow = indices[i];

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

{

sortedMatrix[i, j] = matrix[originalRow, j];

}

}

return sortedMatrix;

}

static void PrintMatrix(int[,] matrix)

{

int size = matrix.GetLength(0);

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

{

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

{

Console.Write($"{matrix[i, j],5}");

}

Console.WriteLine();

}

}

static void PrintRowSums(int[] rowSums)

{

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

{

Console.WriteLine($"Строка {i}: {rowSums[i],6}");

}

}

static void PrintStatistics(int[,] originalMatrix, int[,] sortedMatrix)

{

int size = originalMatrix.GetLength(0);

int[] originalSums = CalculateRowSums(originalMatrix);

int[] sortedSums = CalculateRowSums(sortedMatrix);

int positiveSums = sortedSums.Count(sum => sum > 0);

int negativeSums = sortedSums.Count(sum => sum < 0);

int zeroSums = sortedSums.Count(sum => sum == 0);

}

static int GetPositiveInteger(string message)

{

while (true)

{

Console.Write(message);

if (int.TryParse(Console.ReadLine(), out int number) && number > 0)

return number;

Console.WriteLine("Ошибка: введите положительное целое число!");

}

}

}

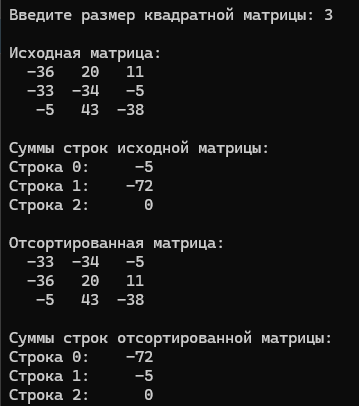


Рисунок 1.3.3 – Выполнение кода листинга 3