Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М.Машерова

**Отчет**

по учебной практике

«Массивы, строки, переключатели, статические методы (методы классов)»

по предмету «Конструирование программ и языки программирования»

Выполнил учащийся Казаченко Н.С.

группы 3ПОИС23 30.09.2025 г.

Проверил Алейников М.А.

30.09.2025 г.

Орша, 2025

**Цель:** закрепить знания о статических методах (методы классов) в языке программирования С#.

**Модуль 1.3 СТАТИЧЕСКИЕ МЕТОДЫ (МЕТОДЫ КЛАССОВ)**

Задача 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

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

Задача 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.

**Ход работы**

**Выполнение задания 1**

**Листинг 1.**

using System;

using static System.Console;

class Program

{

static void Main()

{

// Ввод числителя и знаменателя

Write("Введите числитель (неотрицательное число): ");

int numerator = int.Parse(ReadLine());

Write("Введите знаменатель (положительное число): ");

int denominator = int.Parse(ReadLine());

// Проверка на корректность ввода

if (numerator < 0 || denominator <= 0)

{

WriteLine("Ошибка: числитель должен быть неотрицательным, а знаменатель положительным числом.");

return;

}

// Вычисление НОД числителя и знаменателя

int gcd = GCD(numerator, denominator);

// Сокращение дроби

int reducedNumerator = numerator / gcd;

int reducedDenominator = denominator / gcd;

// Вывод сокращенной дроби

WriteLine($"Сокращенная дробь: {reducedNumerator}/{reducedDenominator}");

}

// Статический метод для вычисления НОД двух чисел

static int GCD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return a;

}

}

На рисунке 1 представлено выполнение задания создания функции (статического метода) для вычисления наибольшего общего делителя двух

чисел.

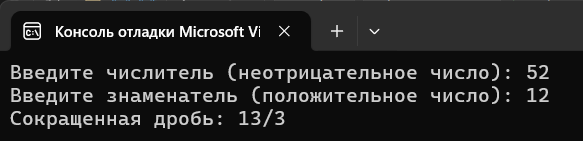


Рисунок 1 – Выполнение программы

**Выполнение задания 2**

**Листинг 2.**

using System;

using static System.Console;

internal class Program

{

static void Main(string[] args)

{

while (true)

{

// Ввод числа, которое будет являться суммой элементов массива

WriteLine("Введите число, которое будет являться суммой элементов массива:");

int targetSum = Convert.ToInt32(ReadLine());

// Проверка корректности ввода числа

if (targetSum <= 0)

{

WriteLine("Число должно быть больше 0");

WriteLine("Хотите выполнить еще одно преобразование? (Y для продолжения, любой другой символ для выхода):");

char Zerochoice = ReadKey().KeyChar;

WriteLine(); // Переход на новую строку

if (Zerochoice != 'Y' && Zerochoice != 'y')

break; // Прерываем цикл и завершаем программу, если пользователь не ввел 'Y'

continue; // Переход к следующей итерации цикла

}

// Массив для хранения элементов

var elements = new System.Collections.Generic.List<int>();

Random rand = new Random();

int currentSum = 0;

// Заполнение массива максимальными значениями

while (currentSum < targetSum)

{

int valueToAdd = Math.Min(9, targetSum - currentSum); // Определяем максимальное значение для добавления

elements.Add(valueToAdd);

currentSum += valueToAdd;

}

// Преобразуем список в массив

int[] resultArray = elements.ToArray();

// Вывод результата

WriteLine("Сгенерированный массив: " + string.Join(", ", resultArray)); // Вывод элементов массива

WriteLine("Сумма элементов: " + currentSum);

WriteLine("Хотите выполнить еще одно преобразование? (Y для продолжения, любой другой символ для выхода):");

char choice = ReadKey().KeyChar;

WriteLine(); // Переход на новую строку

if (choice != 'Y' && choice != 'y')

break; // Прерываем цикл и завершаем программу, если пользователь не ввел 'Y'

}

WriteLine("Программа завершена.");

}

}

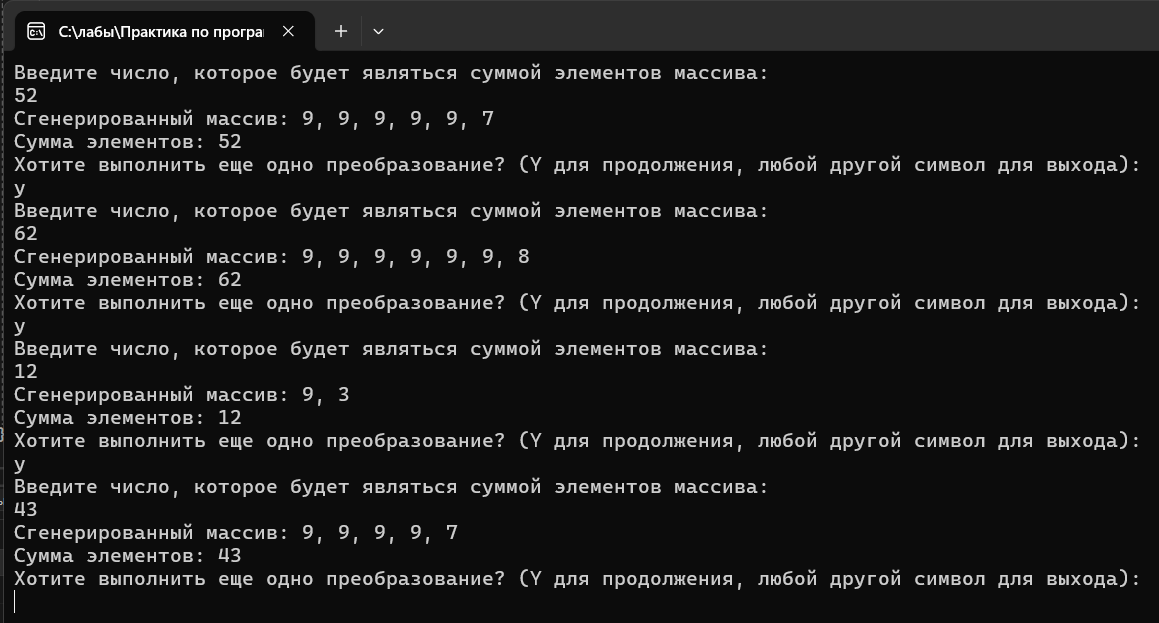
На рисунке 2 представлено выполнение задания создания массива и заполнения его случайными числами из диапазона от 1 до 9.

Рисунок 2 – Выполнение программы

**Выполнение задания 3**

**Листинг 3.**

using System;

using static System.Console;

class Program

{

static void Main()

{

int n = 5; // Размер матрицы

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

Random rand = new Random();

// Заполнение матрицы случайными значениями

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

{

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

matrix[i, j] = rand.Next(-50, 51);

}

// Вывод исходной матрицы

WriteLine("Исходная матрица:");

PrintMatrix(matrix);

// Сортировка строк по суммам

int[] rowSums = new int[n];

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

rowSums[i] = GetRowSum(matrix, i);

// Создание массива индексов для сортировки

int[] indices = new int[n];

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

indices[i] = i;

// Сортировка индексов на основе сумм строк

Array.Sort(rowSums, indices);

// Создание новой матрицы для отсортированных строк

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

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

{

int rowIndex = indices[i];

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

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

}

// Вывод отсортированной матрицы

WriteLine("\nМатрица после сортировки по суммам строк:");

PrintMatrix(sortedMatrix);

}

static void PrintMatrix(int[,] matrix)

{

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

Write(matrix[i, j] + "\t");

WriteLine();

}

}

static int GetRowSum(int[,] matrix, int rowIndex)

{

int sum = 0;

for (int j = 0; j < matrix.GetLength(1); j++)

sum += matrix[rowIndex, j];

return sum;

}

}

На рисунке 3 представлено выполнение задания создания квадратной целочисленной матрицы и заполнения её случайными значениями из диапазона [–50, 50].

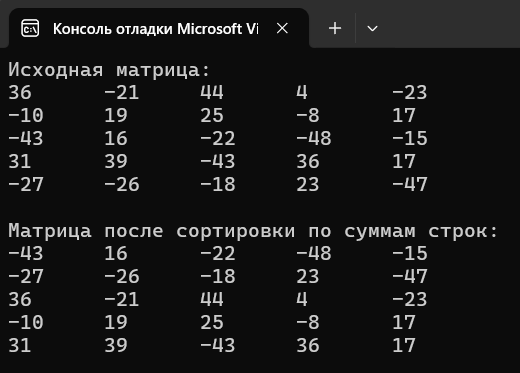


Рисунок 3 – Выполнение программы

**Ссылка на репозиторий:** [Asevzz/-1.3: Практика КПиЯП (github.com)](https://github.com/Asevzz/-1.3)

**Вывод:** в ходе выполнения модуля 1.3 были закреплены знания о статических методах (методы классов) в языке программирования С#.