using System; // Подключение пространства имен для работы с основными классами

using System.Linq; // Подключение пространства имен для работы с LINQ

namespace OneDimensionalArray // Объявление пространства имен

{

// Задание 1: Определение класса "Одномерный массив"

public class OneDimensionalArray

{

private int[] array; // Поле для хранения массива целых чисел

// Конструктор, принимающий массив целых чисел

public OneDimensionalArray(int[] array)

{

this.array = array; // Инициализация поля

}

// Индексатор для доступа к элементам массива

public int this[int index]

{

get => array[index]; // Получение элемента по индексу

set => array[index] = value; // Установка значения элемента по индексу

}

// Перегрузка оператора \* (умножение массивов)

public static OneDimensionalArray operator \*(OneDimensionalArray a1, OneDimensionalArray a2)

{

// Умножение соответствующих элементов массивов и создание нового массива

int[] result = a1.array.Zip(a2.array, (x, y) => x \* y).ToArray();

return new OneDimensionalArray(result); // Возврат нового объекта OneDimensionalArray

}

// Перегрузка оператора true (если массив не содержит отрицательных элементов)

public static bool operator true(OneDimensionalArray a)

{

return a.array.All(x => x >= 0); // Проверка, что все элементы неотрицательные

}

// Перегрузка оператора false

public static bool operator false(OneDimensionalArray a)

{

return a.array.Any(x => x < 0); // Проверка, есть ли отрицательные элементы

}

// Перегрузка оператора приведения к int (возвращает размер массива)

public static explicit operator int(OneDimensionalArray a)

{

return a.array.Length; // Возврат длины массива

}

// Перегрузка оператора == (проверка на равенство)

public static bool operator ==(OneDimensionalArray a1, OneDimensionalArray a2)

{

return a1.array.SequenceEqual(a2.array); // Проверка на равенство массивов

}

// Перегрузка оператора !=

public static bool operator !=(OneDimensionalArray a1, OneDimensionalArray a2)

{

return !(a1 == a2); // Логическое отрицание проверки на равенство

}

// Перегрузка оператора < (сравнение)

public static bool operator <(OneDimensionalArray a1, OneDimensionalArray a2)

{

return a1.array.Sum() < a2.array.Sum(); // Сравнение суммы элементов массивов

}

public static bool operator >(OneDimensionalArray a1, OneDimensionalArray a2)

{

return a1.array.Sum() > a2.array.Sum(); // Сравнение суммы элементов массивов

}

// Печать массива в виде строки

public override string ToString()

{

return string.Join(", ", array); // Объединение элементов массива в строку

}

public override bool Equals(object obj)

{

return obj is OneDimensionalArray array && this == array; // Проверка равенства объектов

}

public override int GetHashCode()

{

return array.GetHashCode(); // Получение хэш-кода массива

}

// Задание 2: Вложенный объект Production

public class Production

{

public int Id { get; set; } // Идентификатор организации

public string OrganizationName { get; set; } // Имя организации

// Конструктор для инициализации полей

public Production(int id, string organizationName)

{

Id = id; // Инициализация идентификатора

OrganizationName = organizationName; // Инициализация имени организации

}

}

public Production ProductionInfo { get; set; } // Свойство для хранения информации о производстве

// Задание 3: Вложенный класс Developer

public class Developer

{

public string FullName { get; set; } // Полное имя разработчика

public int Id { get; set; } // Идентификатор разработчика

public string Department { get; set; } // Отдел разработчика

// Конструктор для инициализации полей

public Developer(string fullName, int id, string department)

{

FullName = fullName; // Инициализация полного имени

Id = id; // Инициализация идентификатора

Department = department; // Инициализация отдела

}

}

public Developer DeveloperInfo { get; set; } // Свойство для хранения информации о разработчике

// Метод для получения массива

public int[] GetArray()

{

return array; // Возврат массива

}

}

// Задание 4: Статический класс StatisticOperation

public static class StatisticOperation

{

// Сумма элементов

public static int Sum(OneDimensionalArray array)

{

return array.GetArray().Sum(); // Возврат суммы элементов массива

}

// Разница между максимальным и минимальным

public static int DifferenceMaxMin(OneDimensionalArray array)

{

var arr = array.GetArray(); // Получение массива

return arr.Max() - arr.Min(); // Разница между максимальным и минимальным элементами

}

// Подсчёт количества элементов

public static int CountElements(OneDimensionalArray array)

{

return array.GetArray().Length; // Возврат количества элементов в массиве

}

// Задание 5: Методы расширения

// Проверка на содержание определённого символа в строке

public static bool ContainsSymbol(this string str, char symbol)

{

return str.Contains(symbol); // Проверка наличия символа в строке

}

// Удаление отрицательных элементов

public static OneDimensionalArray RemoveNegative(this OneDimensionalArray array)

{

// Создание нового массива без отрицательных элементов

return new OneDimensionalArray(array.GetArray().Where(x => x >= 0).ToArray());

}

}

// Основной класс программы

class Program

{

static void Main(string[] args)

{

Console.WriteLine("--- Задание 1: Работа с классом и перегрузками --- \n");

var array1 = new OneDimensionalArray(new[] { -1, 2, 3 }); // Создание первого массива

var array2 = new OneDimensionalArray(new[] { 4, 5, 6 }); // Создание второго массива

Console.WriteLine("Первый массив: " + array1); // Вывод первого массива

Console.WriteLine("Второй массив: " + array2); // Вывод второго массива

var multipliedArray = array1 \* array2; // Умножение массивов

Console.WriteLine("Результат умножения массивов: " + multipliedArray); // Вывод результата умножения

if (array1) // Проверка на отсутствие отрицательных элементов

Console.WriteLine("Массив 1 не содержит отрицательных элементов");

int size = (int)array1; // Получение размера первого массива

Console.WriteLine("Размер массива 1: " + size); // Вывод размера массива

Console.WriteLine("Массивы равны: " + (array1 == array2)); // Проверка на равенство массивов

Console.WriteLine("Массив 1 меньше массива 2: " + (array1 < array2)); // Сравнение массивов по сумме

Console.WriteLine("\n--- Задание 2: Работа с вложенным объектом Production ---\n");

array1.ProductionInfo = new OneDimensionalArray.Production(1, "Организация ООП"); // Инициализация информации о производстве

Console.WriteLine("Id: " + array1.ProductionInfo.Id + ", Имя организации: " + array1.ProductionInfo.OrganizationName); // Вывод информации о производстве

Console.WriteLine("\n--- Задание 3: Работа с вложенным классом Developer ---\n");

array1.DeveloperInfo = new OneDimensionalArray.Developer("Адамович Карианна", 101, "IT"); // Инициализация информации о разработчике

Console.WriteLine("Разработчик: " + array1.DeveloperInfo.FullName + " Id: " + array1.DeveloperInfo.Id + " Отдел: " + array1.DeveloperInfo.Department); // Вывод информации о разработчике

Console.WriteLine("\n--- Задание 4: Статические методы ---\n");

Console.WriteLine("Сумма элементов: " + StatisticOperation.Sum(array1)); // Вывод суммы элементов массива

Console.WriteLine("Разница между макс и мин: " + StatisticOperation.DifferenceMaxMin(array1)); // Вывод разницы между максимальным и минимальным элементами

Console.WriteLine("Количество элементов: " + StatisticOperation.CountElements(array1)); // Вывод количества элементов

Console.WriteLine("\n--- Задание 5: Методы расширения ---\n");

string str = "Hello, world!"; // Инициализация строки

Console.WriteLine("Содержит 'o': " + str.ContainsSymbol('o')); // Проверка на наличие символа 'o'

var arrayWithoutNegatives = array1.RemoveNegative(); // Удаление отрицательных элементов

Console.WriteLine("Массив без отрицательных элементов: " + arrayWithoutNegatives); // Вывод массива без отрицательных элементов

}

}

}

# Описание программы "Одномерный массив"

## Общая информация

Данная программа реализует класс `OneDimensionalArray`, который представляет собой одномерный массив целых чисел и включает в себя различные операции и функционал для работы с массивами. Программа демонстрирует использование операторов перегрузки, статических методов, вложенных объектов и классов, а также методов расширения.

## Основные функции класса `OneDimensionalArray`:

1. \*\*Инициализация массива\*\*: Класс позволяет создавать массив целых чисел через конструктор.

2. \*\*Индексация\*\*: Использование индексатора для доступа к элементам массива.

3. \*\*Перегрузка операторов\*\*:

- Умножение массивов.

- Проверка на наличие отрицательных значений.

- Сравнение массивов на равенство и неравенство.

- Сравнение массивов по сумме элементов.

4. \*\*Методы для получения информации о массиве\*\*: Размер массива, сумма элементов, максимальное и минимальное значения.

## Вложенные классы:

- \*\*Production\*\*: Хранит информацию о производственной организации с идентификатором и именем.

- \*\*Developer\*\*: Хранит информацию о разработчике, включая полное имя, идентификатор и отдел.

## Статический класс `StatisticOperation`:

Содержит методы для выполнения статистических операций над объектами `OneDimensionalArray`, таких как:

- Подсчет суммы элементов.

- Вычисление разницы между максимальным и минимальным элементами.

- Подсчет количества элементов в массиве.

## Методы расширения:

- Метод `ContainsSymbol` для проверки наличия определенного символа в строке.

- Метод `RemoveNegative` для удаления отрицательных элементов из массива.

## Основной класс `Program`:

Содержит метод `Main`, который демонстрирует функционал класса `OneDimensionalArray`:

- Создание и вывод массивов.

- Демонстрация перегруженных операторов.

- Вывод информации о вложенных объектах `Production` и `Developer`.

- Использование статических методов для статистических операций.

- Применение методов расширения.

## Пример работы программы:

При запуске программы выводится информация о двух массивах, результат их умножения, статистические данные, и информация о разработчике и организации, что позволяет увидеть весь функционал класса `OneDimensionalArray` в действии.