**Лабораторная работа №5**

**Задание 1.** Создайте проект, в котором опишите класс для решения задачи Вашего варианта. Каждый разрабатываемый класс должен, содержать следующие элементы: скрытые и открытые поля, конструкторы (один из них должен передавать параметром массив), перегруженные операции. В программе должна выполняться проверка всех разработанных элементов класса.

**9.** Описать класс для работы с одномерным массивом вещественных чисел.

Обеспечить следующие возможности: нахождение суммы элементов

массива (перегрузка операции +).

using System;

namespace ArrayOperations

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите количество элементов массива: ");

int size = int.Parse(Console.ReadLine());

// Создаем массив вещественных чисел

double[] myArray = new double[size];

// Вводим значения для массива

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

{

Console.Write($"Введите элемент {i + 1}: ");

myArray[i] = double.Parse(Console.ReadLine());

}

// Создаем объект класса для работы с массивом

ArrayCalculator calculator = new ArrayCalculator(myArray);

// Вычисляем сумму элементов массива

double sum = calculator.CalculateSum();

Console.WriteLine($"Сумма элементов массива: {sum}");

}

}

class ArrayCalculator

{

private double[] array;

// Конструктор, принимающий массив в качестве параметра

public ArrayCalculator(double[] inputArray)

{

array = inputArray;

}

// Перегрузка операции сложения для массивов

public static double operator +(ArrayCalculator calc, double value)

{

double sum = calc.CalculateSum();

return sum + value;

}

// Метод для вычисления суммы элементов массива

public double CalculateSum()

{

double sum = 0;

foreach (double element in array)

{

sum += element;

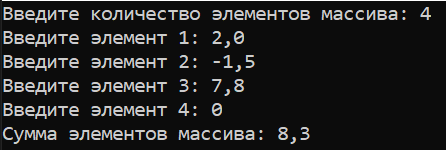
}

return sum;

}

}

}



Контрольные вопросы:

1. **Что представляет собой перегрузка методов?**

Перегрузка методов представляет собой возможность создания нескольких методов с одним и тем же именем, но разной сигнатурой. Сигнатура метода включает в себя имя метода, количество параметров, типы параметров и их порядок. Например, можно иметь несколько версий метода Sum, каждая из которых принимает разное количество аргументов или аргументы разных типов.

1. **Что представляет собой перегрузка операций**

Перегрузка операций позволяет определить пользовательскую реализацию стандартных операторов (например, сложение, вычитание, равенство) для пользовательских типов данных. Это делается с помощью ключевого слова operator. Например, можно перегрузить операторы +, -, \*, / для пользовательской структуры, представляющей рациональное число.

1. **Формат описания операции класса.**

Формат описания операции класса включает в себя объявление класса с ключевым словом class, за которым следует уникальный идентификатор. Внутри класса можно определить поля, свойства, методы и события.

1. **Какие операции нельзя перегружать?**

Операции, которые нельзя перегружать, включают условные логические операторы (например, &&, ||) и операторы присваивания (например, +=, -=).

1. **Что является результатом перегрузки унарных операций?**

Результатом перегрузки унарных операций может быть изменение состояния объекта или возврат нового объекта с измененными значениями.

1. **Какие параметры могут быть у бинарных операций класса?**

Параметры бинарных операций класса могут быть любого типа, но хотя бы один из параметров должен иметь тип, который содержит объявление оператора.

1. **Как выполняется перегрузка операций отношения?**

Перегрузка операций отношения позволяет определить пользовательское поведение для операторов сравнения (например, ==, <, >).

1. **Чем являются строки в С#?**

Строки в С# являются ссылочными типами данных и представляют последовательности символов.

1. **Какие операции определекны для строк?**

Для строк определены операции конкатенации (+), сравнения (==, !=), доступа к символам и другие.

1. **Как создаются строки?**

Строки создаются с помощью ключевого слова new и конструктора string.

1. **Можно ли изменять значение строки?**

Значение строки нельзя изменять, так как строки в C# являются неизменяемыми (immutable).