Турсунов Баходурхон

#9. Methods. Delegates. Lambda expressions. Default Values for Lambda Parameters. Static modifier for Lambda Expression. Input parameters: in, out, ref.

**1. Создайте консольное приложение C#, демонстрирующее использование метода.**

**метод должен принимать два целых числа в качестве входных параметров и возвращать их**

**сумму. Вызовите этот метод из метода Main и отобразите результат на**

**экран.**

class Program

{

static void Main(string[] args)

{

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

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

int res = Sum(firstNumber, secondNumber);

Console.WriteLine(res);

}

static int Sum(int firstNumber, int secondNumber)

{

return firstNumber + secondNumber;

}

}

**2. Реализуйте три метода, каждый из которых выполняет разные операции (например,**

**сложение, вычитание, умножение) с использованием делегата. Проверьте эти методы**

**вызвав их через делегата и отобразив результаты на экране.**

public delegate int MathOperation(int a, int b);

class Program

{

static void Main(string[] args)

{

MathOperation substraction = new MathOperation(Subtraction);

MathOperation multiplication = new MathOperation(Multiplication);

MathOperation division = new MathOperation(Division);

int resSubstraction = PerformOperation(10, 2, Subtraction);

Console.WriteLine($"Вычитание: {resSubstraction}");

int resMultiplication = PerformOperation(10, 2, Multiplication);

Console.WriteLine($"Умножение: {resMultiplication}");

int resDivision = PerformOperation(10, 0, Division);

Console.WriteLine($"Деление: {resDivision}");

}

static int PerformOperation(int a, int b, MathOperation operation)

{

return operation(a, b);

}

static int Subtraction(int a, int b) => a - b;

static int Multiplication(int a, int b)

{

if (a == 0 || b == 0)

{

return 0;

}

return a \* b;

}

static int Division(int a, int b)

{

if (a == 0 || b == 0)

{

Console.WriteLine("На ноль делить нельзя");

}

return a / b;

}

}

3. Создайте лямбда-выражение, включающее модификатор static. Обсудите и

задокументируйте, как этот модификатор влияет на способность лямбды захватывать переменные из области применения.

namespace HW9\_3

{

internal class Program

{

static void Main(string[] args)

{

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

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

int res = Sum(first, second);

Console.WriteLine(res);

}

static int Sum(int a, int b) => a + b;

}

}

// Лямбда выражения представляют с собой учпрощенную запись анонимных методов. Лямбда выражения позволяют создать емкие лаконичные

//методы, которые могу возвращать некоторое значение и которые можно передать в качестве параметров в другие методы.

// а ключево слово static при объявлении методов или функций можно использовать только в статических классах.

4. Создайте еще один метод, который принимает целочисленный параметр с «in».

модификатор. Попытайтесь изменить параметр «in» внутри метода. Обсудите

результат и объясните, почему модификатор «in» предотвращает модификацию, в комментариях.

namespace HW9\_4

{

internal class Program

{

static void Main(string[] args)

{

int value = 5;

int res = Sum(value);

}

static void Sum(in int value) // in как и ref и out позволяет передать параметр в метод по ссылке, но в отличие от ref и out.

{

// value = 5; // внутри методы в которой мы данный параметр передаем мы его изменить не можем и по сути он становится readonly, мы можем извлечь оттуда данные, но записать в нее мы ничего не можем

}

}

}