**ЗАДАНИЕ 12**

Задание 1. Объявить тип делегата, который ссылается на метод.

Требования к сигнатуре метода следующие:

* метод получает входным параметром переменную типа double;
* метод возвращает значение типа double, которое есть результатом вычисления.

Реализовать вызов методов с помощью делегата, которые получают

радиус R и вычисляют:

* длину окружности по формуле D = 2 · π · R;
* площадь круга по формуле S = π · R 2 ;
* объем шара. Формула: V = 4/3 \* π · R 3 .

Методы должны быть объявлены как статические (с использованием

ключевого слова static). Для работы программы выбираем такие имена:

* название типа делегата – CalcFigure;
* название делегата (экземпляра объекта) – CF;
* название метода, который вычисляет длину окружности
* – Get\_Length();
* название метода, который вычисляет площадь круга – Get\_Area();
* название метода, который вычисляет объем шара – Get\_Volume().

Листинг программы:

using System;

public delegate double CalcFigure(double R);

class Program

{

static void Main()

{

CalcFigure CF = null;

CF = Get\_Length;

Console.WriteLine($"Длина окружности: {CF(5)}");

CF = Get\_Area;

Console.WriteLine($"Площадь круга: {CF(5)}");

CF = Get\_Volume;

Console.WriteLine($"Объем шара: {CF(5)}");

}

static double Get\_Length(double R)

{

return 2 \* Math.PI \* R;

}

static double Get\_Area(double R)

{

return Math.PI \* Math.Pow(R, 2);

}

static double Get\_Volume(double R)

{

return 4.0 / 3 \* Math.PI \* Math.Pow(R, 3);

}

}

Таблица 12.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| - | Длина окружности: 31,41592653589793  Площадь круга: 78,53981633974483  Объем шара: 523,5987755982989 |

Анализ результатов:

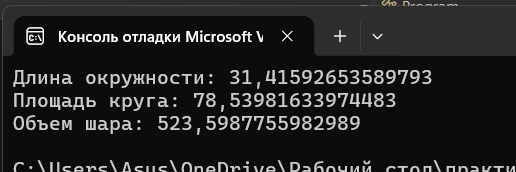


Рисунок 12.1 – Результат работы программы

Задание 2. Используя Visual Studio, создайте проект по шаблону Console Application. Создайте четыре лямбда оператора для выполнения арифметических действий: (Add – сложение, Sub – вычитание, Mul – умножение, Div – деление). Каждый лямбда оператор должен принимать два аргумента и возвращать результат вычисления. Лямбда оператор деления должен делать проверку деления на ноль. Написать программу, которая будет выполнять арифметические действия указанные пользователем.

Листинг программы:

using System;

class Program

{

static void Main()

{

Func<double, double, double> Add = (a, b) => a + b;

Func<double, double, double> Sub = (a, b) => a - b;

Func<double, double, double> Mul = (a, b) => a \* b;

Func<double, double, double> Div = (a, b) =>

{

if (b != 0)

{

return a / b;

}

else

{

Console.WriteLine("Ошибка: деление на ноль.");

return double.NaN; // возвращаем NaN для обозначения ошибки

}

};

Console.WriteLine("Введите два числа:");

double num1 = Convert.ToDouble(Console.ReadLine());

double num2 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Сумма: {Add(num1, num2)}");

Console.WriteLine($"Разность: {Sub(num1, num2)}");

Console.WriteLine($"Умножение: {Mul(num1, num2)}");

Console.WriteLine($"Деление: {Div(num1, num2)}");

}

}

Таблица 12.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Два числа | Сумма: 9  Разность: -1  Умножение: 20  Деление: 0,8 |

Анализ результатов:

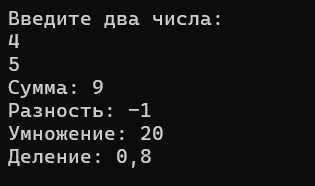


Рисунок 12.2 – Результат работы программы

Задание 3. Создать 3 метода для работы со строкой. Создать делегат. В программе вызывать все три метода при помощи делегата.

Листинг программы:

using System;

class Program

{

delegate string StringOperation(string str);

static void Main()

{

string input = "Hello, World!";

StringOperation strOp = null;

strOp = ToUpperCase;

Console.WriteLine(strOp(input));

strOp = ToLowerCase;

Console.WriteLine(strOp(input));

strOp = ReverseString;

Console.WriteLine(strOp(input));

}

static string ToUpperCase(string str)

{

return str.ToUpper();

}

static string ToLowerCase(string str)

{

return str.ToLower();

}

static string ReverseString(string str)

{

char[] charArray = str.ToCharArray();

Array.Reverse(charArray);

return new string(charArray);

}

}

Таблица 12.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| HELLO, WORLD! | hello, world!  !dlroW ,olleH |

Анализ результатов:

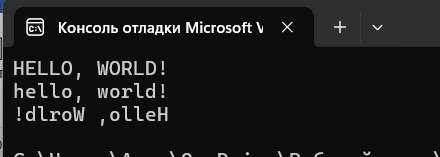


Рисунок 13.3 – Результат работы программы

Задание 4. Используя Visual Studio, создайте проект по шаблону Console Application. Создайте анонимный метод, который принимает в качестве аргумента массив делегатов и возвращает среднее арифметическое возвращаемых значений методов сообщенных с делегатами в массиве. Методы, сообщенные с делегатами из массива, возвращают случайное значение типа int.

Листинг программы:

using System;

class Program

{

delegate int RandomNumberGenerator();

static void Main()

{

RandomNumberGenerator[] delegatesArray = new RandomNumberGenerator[]

{

delegate { return new Random().Next(1, 100); },

delegate { return new Random().Next(1, 100); },

delegate { return new Random().Next(1, 100); }

};

Func<RandomNumberGenerator[], double> GetAverage = delegate (RandomNumberGenerator[] delegates)

{

int sum = 0;

foreach (var del in delegates)

{

sum += del();

}

return (double)sum / delegates.Length;

};

double average = GetAverage(delegatesArray);

Console.WriteLine($"Среднее арифметическое случайных чисел: {average}");

}

}

Таблица 12.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| - | 27,666666666666668 |

Анализ результатов:

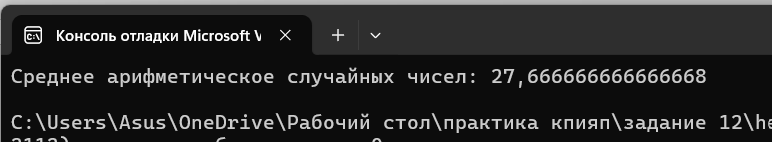


Рисунок 12.4 – Результат работы программы