**12 «ДЕЛЕГАТЫ».**

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;

delegate double CalcFigure(double R);

class Program

{

static double Get\_Length(double R)

{

return 2 \* Math.PI \* R;

}

static double Get\_Area(double R)

{

return Math.PI \* R \* R;

}

static double Get\_Volume(double R)

{

return (4.0 / 3.0) \* Math.PI \* R \* R \* R;

}

static void Main(string[] args)

{

CalcFigure CF;

CF = Get\_Length;

double circumference = CF(5);

Console.WriteLine("Длина окружности: " + circumference);

CF = Get\_Area;

double area = CF(5);

Console.WriteLine("Площадь круга: " + area);

CF = Get\_Volume;

double volume = CF(5);

Console.WriteLine("Объем шара: " + volume);

}

}

Таблица 1.1 – Входные и выходные данные

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

Анализ результатов:

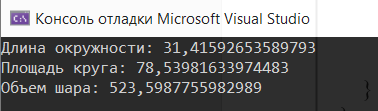


Рисунок 1.1 – Результат работы программы

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;

public class StringManipulation

{

public static string ToUpper(string input)

{

return input.ToUpper();

}

public static string ToLower(string input)

{

return input.ToLower();

}

public static int CountCharacters(string input)

{

return input.Length;

}

public delegate void StringManipulationDelegate(string input);

static void Main(string[] args)

{

StringManipulationDelegate delegateInstance;

string testString = "Hello, World!";

delegateInstance = ToUpper;

delegateInstance(testString);

delegateInstance = ToLower;

delegateInstance(testString);

Func<string, int> countDelegate = CountCharacters;

int charCount = countDelegate(testString);

Console.WriteLine("Количество символов в строке: " + charCount);

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 2 3 | Выберите операцию:  1. Сложение  2. Вычитание  3. Умножение  4. Деление  1  Введите два числа:  2  3  Результат: 5 |

Анализ результатов:

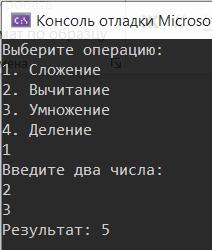


Рисунок 1.2 – Результат работы программы

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

Листинг программы:

using System;

public class StringManipulation

{

// Метод для преобразования строки в верхний регистр

public static void ToUpper(string input)

{

Console.WriteLine(input.ToUpper());

}

// Метод для преобразования строки в нижний регистр

public static void ToLower(string input)

{

Console.WriteLine(input.ToLower());

}

// Метод для подсчета количества символов в строке

public static void CountCharacters(string input)

{

Console.WriteLine("Количество символов в строке: " + input.Length);

}

// Объявление делегата

public delegate void StringManipulationDelegate(string input);

static void Main(string[] args)

{

// Создание экземпляра делегата

StringManipulationDelegate delegateInstance;

string testString = "Hello, World!";

// Присваивание методов делегату и вызов через делегат

delegateInstance = ToUpper;

delegateInstance(testString);

delegateInstance = ToLower;

delegateInstance(testString);

delegateInstance = CountCharacters;

delegateInstance(testString);

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | HELLO, WORLD!  hello, world!  Количество символов в строке: 13 |

Анализ результатов:

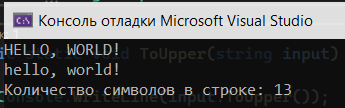


Рисунок 1.3 – Результат работы программы

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;

namespace DelegateArrayExample

{

class Program

{

delegate int RandomNumberDelegate();

static void Main(string[] args)

{

RandomNumberDelegate[] delegatesArray = new RandomNumberDelegate[5];

for (int i = 0; i < delegatesArray.Length; i++)

{

delegatesArray[i] = () =>

{

Random random = new Random();

return random.Next(1, 101);

};

}

double average = CalculateAverage(delegatesArray);

Console.WriteLine("Среднее арифметическое случайных чисел: " + average);

}

static double CalculateAverage(RandomNumberDelegate[] delegates)

{

int sum = 0;

foreach (var del in delegates)

{

sum += del();

}

return (double)sum / delegates.Length;

}

}

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Среднее арифметическое случайных чисел: 56 |

Анализ результатов:

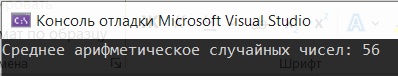


Рисунок 1.4 – Результат работы программы