**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 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 · R2;
* объем шара. Формула: V = 4/3 \* π · R3.

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

delegate double CalculateFigure(double r);

public static double Get\_Length(double r)

{

double result = 2 \* Math.PI \* r;

Console.WriteLine("Circle length: {0:f2}", result);

return result;

}

public static double Get\_Area(double r)

{

double result = Math.PI \* Math.Pow(r, 2);

Console.WriteLine("Circle square: {0:f2}", result);

return result;

}

public static double Get\_Volume(double r)

{

double result = (4 / 3) \* Math.PI \* Math.Pow(r, 3);

Console.WriteLine("Circle volume: {0:f2}", result);

return result;

}

static void Main(string[] args)

{

Console.Write("Radius = ");

CalculateFigure CF = new CalculateFigure(Get\_Length);

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

CF += Get\_Area;

CF += Get\_Volume;

CF(r);

Console.ReadLine();

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Radius = 5 | Circle length: 31,42  Circle square: 78,54  Circle volume: 392,70 |

Анализ результатов:

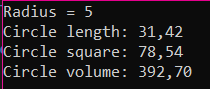


Рисунок 1.1 – Результат работы программы

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:

public delegate int Del(int first\_number, int second\_number);

class Program

{

static Func<int, int, double> Sum = (int fn, int sn) => fn + sn;

static Func<int, int, double> Sub = (int fn, int sn) => fn - sn;

static Func<int, int, double> Mul = (int fn, int sn) => fn \* sn;

static Func<int, int, double> Div = (int fn, int sn) => { if (fn == 0) return -1; else return fn / sn; };

static void Main(string[] args)

{

Get\_Result();

}

static void Get\_Result()

{

Console.Write("Input first number = ");

int fn = Convert.ToInt32(Console.ReadLine());

Console.Write("Input second number =");

int sn = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"Adding result: {fn + sn}");

Console.WriteLine($"Substract result:{fn - sn} ");

Console.WriteLine($"Mul result: {fn \* sn}");

string message = "Not a number!";

Console.WriteLine($"Div result: {fn/sn}");

if (fn == 0 || sn == 0)

{

Console.WriteLine($"Div by zero result: {message}");

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| first number = 4  second number =1 | Adding result: 5  Substract result:3  Mul result: 4  Div result: 4 |

Анализ результатов:

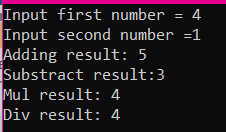


Рисунок 1.2 – Результат работы программы

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

Листинг программы:

delegate string Operations(string str);

static void Main(string[] args)

{

Console.Write("Введите строку: ");

string str = Console.ReadLine();

Operations op = upper;

Console.WriteLine("Высокий регистр: " + op(str));

op += lower;

Console.WriteLine("Низкий регистр: " + op(str));

op += letter3;

Console.WriteLine("Начальная буква стала последней: " + op(str));

Console.Read();

}

private static string lower(string str)

{

return str.ToLower();

}

private static string upper(string str)

{

char ch = 'o';

int indexOfChar = str.IndexOf(ch);

return str.ToUpper();

}

private static string letter3(string str)

{

return str.Replace(str[0], str[str.Length - 1]);

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| п Полина | Высокий регистр: П ПОЛИНА  Низкий регистр: п полина  Начальная буква стала последней: а Полина |

Анализ результатов:

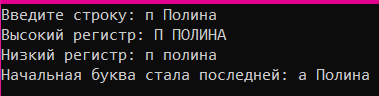


Рисунок 1.3 – Результат работы программы

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 делегатами из массива, возвращают случайное значение типа int.

Листинг программы:

try

{

Random rand = new Random();

int Randomizer() //функция рандома

{

return rand.Next(1, 10);

}

Number[] numArr = new Number[5];

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

{

numArr[i] = Randomizer;

}

MediumCalc mediumCalc = delegate (Number[] arrayX)

{

var arrayT = new int[5];

int sum = 0;

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

{

arrayT[i] = arrayX[i]();

sum += arrayT[i];

Console.Write(arrayT[i] + " ");

}

return ((double)(sum) / (arrayX.Length));

};

Console.WriteLine($"\nResult = {mediumCalc(numArr)}");

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

public delegate int Number();

public delegate double MediumCalc(Number[] arrayX);

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 1 4 2 9 6  4,4 |

Анализ результатов:

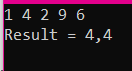


Рисунок 1.4 – Результат работы программы