**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().

Листининг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace task1

{

class Program

{

public delegate double CalcFigure(double R);

static double Get\_Length(double r)

{

double D;

D = 2 \* Math.PI \* r;

Console.WriteLine($"Длина окружности = {Math.Round(D, 3)}");

return D;

}

static double Get\_Area(double r)

{

double S;

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

Console.WriteLine($"Площадь круга = {Math.Round(S, 3)}");

return S;

}

static double Get\_Volume(double r)

{

double V;

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

Console.WriteLine($"Объём шара = {Math.Round(V, 3)}");

return V;

}

static void Main(string[] args)

{

Console.Write("Введите радиус-");

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

CalcFigure CF = new CalcFigure(Get\_Length);

CF += Get\_Area;

CF += Get\_Volume;

CF(rad);

}

}

}

Таблица 12.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 7 | Длина окружности = 43,982  Площадь круга = 153,938  Объём шара = 1077,566 |

Анализ результатов:

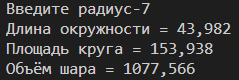


Рисунок 12.1 – Результат работы программы

Источник – собственная разработка

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.

Листининг программы:

using System;

namespace task2

{

class Program

{

public delegate double MyDelegat(double a, double b);

static void Main(string[] args)

{

Console.WriteLine("Введите действие--> +,-,\*,/");

string a = Console.ReadLine();

Console.WriteLine("Первое число -->");

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

Console.WriteLine("Второе число -->");

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

switch (a)

{

case "+":

MyDelegat myDelegat = (c, b) => { return c + b; };

double del = myDelegat(x, y);

Console.WriteLine("Ответ:{0}", del);

break;

case "-":

MyDelegat myDelegat1 = (c, b) => { return c - b; };

double del1 = myDelegat1(x, y);

Console.WriteLine("Ответ:{0}", del1);

break;

case "\*":

MyDelegat myDelegat2 = (c, b) => { return c \* b; };

double del2 = myDelegat2(x, y);

Console.WriteLine("Ответ:{0}", del2);

break;

case "/":

MyDelegat myDelegat3 = (c, b) => {

if ((c != 0) && (b != 0))

return c / b;

else

Console.WriteLine("операция деления на 0 не допускается");

return 0;

};

double del3 = myDelegat3(x, y);

if (del3!=0)

Console.WriteLine("Ответ:{0}", del3);

break;

default:

Console.WriteLine("Выберите из предложанного");

break;

}

Console.ReadKey();

}

}

}

Таблица 12.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| /, 10, 2 | 5 |

Анализ результатов:

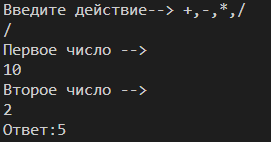


Рисунок 12.2 – Результат работы программы

Источник – собственная разработка

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

Листининг программы:

using System;

namespace task3

{

class Program

{

public delegate string StringDelegate (string line);

static string Increase(string line)

{

Console.WriteLine($"строка с увеличенным регистром: {line.ToUpper()}");

return line.ToUpper();

}

static string TurnOver(string line)

{

char[] sReverse = line.ToCharArray();

Array.Reverse(sReverse);

line = new string(sReverse);

Console.WriteLine($"строка задом наперед: {line}");

return line;

}

static string Decrease(string line)

{

Console.WriteLine($"строка с уменьшенным регистром: {line.ToLower()}");

return line.ToLower();

}

static void Main(string[] args)

{

Console.Write("Введите строку:");

string line = Console.ReadLine();

StringDelegate SD = new StringDelegate(Increase);

SD += TurnOver;

SD += Decrease;

SD(line);

}

}

}

Таблица 12.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Привет, пионер, как твоё настроение, отдохни и приступай к работе! | ПРИВЕТ, ПИОНЕР, КАК ТВОЁ НАСТРОЕНИЕ, ОТДОХНИ И ПРИСТУПАЙ К РАБОТЕ!;  !етобар к йапутсирп и инходто ,еинеортсан ёовт как ,реноип ,тевирП;  привет, пионер, как твоё настроение, отдохни и приступай к работе! |

Анализ результатов:

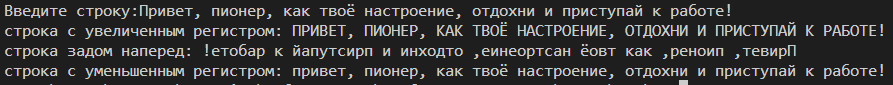


Рисунок 12.3 – Результат работы программы

Источник – собственная разработка