**12. Делегаты**

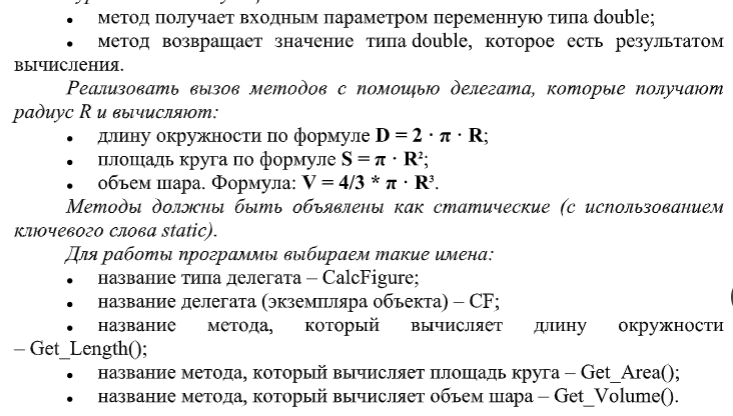
Задание 1. Объявить тип делегата, который ссылается на метод. Требования к сигнатуре метода следующие:

Рисунок 12.1 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

namespace zad1

{

class Program

{

public delegate double CalcFigure(double R);

static double G\_Lenght(double r)

{

double D;

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

Console.WriteLine("Длина окружности={0}", Math.Round(D, 3));

return D;

}

static double G\_Area(double r)

{

double S;

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

Console.WriteLine("Площадь круга={0}", Math.Round(S, 3));

return S;

}

static double G\_Volume(double r)

{

double V;

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

Console.WriteLine("Объем шара={0}", Math.Round(V, 3));

return V;

}

public static void Zad1(CalcFigure a)

{

Console.Write("Введите радиус = ");

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

a(rd);

}

static void Main(string[] args)

{

Console.Write("Введите радиус = ");

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

CalcFigure CF = new CalcFigure(G\_Lenght);

CF += G\_Area;

CF += G\_Volume;

CF(rad);

Console.WriteLine("Выполнение");

Zad1(CF);

Console.ReadLine();

}

}

}

Таблица 12.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | 12.566, 12.566, 25.133 |

Анализ результатов:

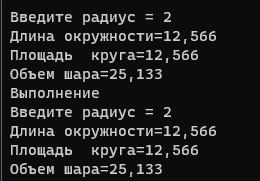


Рисунок 12.2 – Результат работы программы

Источник: собственная разработка

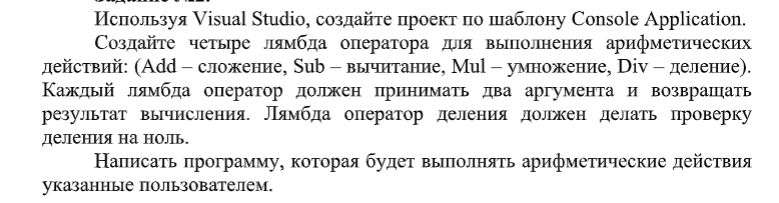
Задание 2.

Рисунок 12.3 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Security.Cryptography.X509Certificates;

namespace MyNamespace

{

class Program

{

delegate double Operation(double x, double y);

public static void Main()

{

Console.WriteLine("Введите число х: ");

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

Console.WriteLine("Введите число y: ");

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

Console.WriteLine("Выберите операцию:");

Console.WriteLine("1 - Сумма, 2 - Разность, 3 - Умножение, 4 - Деление");

double Operation = Convert.ToInt32(Console.ReadLine());

double result;

switch (Operation)

{

case 1:

Operation sum = Add;

result = sum(xNUM, yNUM);

Console.WriteLine(result);

break;

case 2:

Operation sub = Sub;

result = sub(xNUM, yNUM);

Console.WriteLine(result);

break;

case 3:

Operation mul = Mul;

result = mul(xNUM, yNUM);

Console.WriteLine(result);

break;

case 4:

Operation div = Div;

result = Div(xNUM, yNUM);

Console.WriteLine(result);

break;

}

}

public static double Add(double xNUM, double yNUM) => xNUM + yNUM;

public static double Sub(double xNUM, double yNUM) => xNUM - yNUM;

public static double Mul(double xNUM, double yNUM) => xNUM \* yNUM;

public static double Div(double xNUM, double yNUM)

{

if (xNUM == 0)

{

throw new DivideByZeroException();

Console.WriteLine("Ошибка деления на ноль");

}

else if (yNUM == 0)

{

throw new DivideByZeroException();

Console.WriteLine("Ошибка деления на ноль");

}

else

{

return xNUM / yNUM;

}

}

}

}

Таблица 12.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| x = 3, y = 5 | -2 |

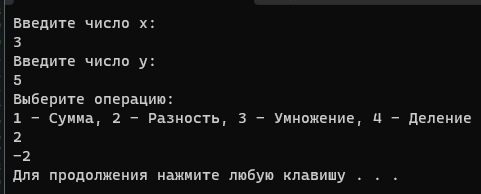
Анализ результатов:

Рисунок 12.4 – Результат работы программы

Источник: собственная разработка

Задание 3.

Рисунок 12.5 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Text;

using System.Threading.Tasks;

namespace zad3

{

internal class Program

{

delegate string JustOperation(string message);

static void Main(string[] args)

{

Console.WriteLine("Введите сообщение");

string message = Console.ReadLine();

Console.WriteLine("Выбор действия: ");

Console.WriteLine("1 - Перевести текст в нижний регистр, 2 - Перевести текст в верхний регистр, 3 - Удаление");

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

string res;

switch (Operation)

{

case 1:

JustOperation lower = toLower;

res = lower(message);

Console.WriteLine(res);

break;

case 2:

JustOperation upper = toUpper;

res = upper(message);

Console.WriteLine(res);

break;

case 3:

JustOperation Trim = Tr;

res = Trim(message);

Console.WriteLine(res);

break;

}

}

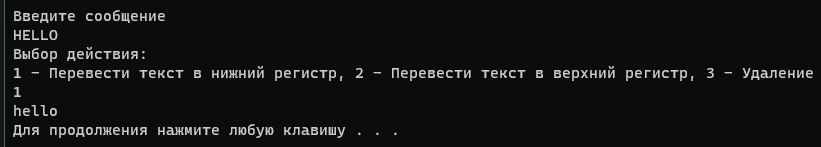
public static string toLower(string message) => message.ToLower();

public static string toUpper(string message) => message.ToUpper();

public static string Tr(string message) => message.Trim();

}

}

Таблица 12.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| HELLO | Hello |

Анализ результатов:

Рисунок 12.6 – Результат работы программы

Источник: собственная разработка