**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().

namespace task1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите R");

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

CalcFigure CF;

CF = Get\_Length;

Console.WriteLine($"Длина {CF(R):f3}");

CF = Get\_Area;

Console.WriteLine($"Площадь {CF(R):f3}");

CF = Get\_Volume;

Console.WriteLine($"Объём {CF(R):f3}");

class Class1

{

public delegate double CalcFigure(double R);

public static double Get\_Length(double R)

{

double D = 2 \* Math.PI \* R;

return D;

}

public static double Get\_Area(double R)

{

double S = Math.PI \* R \* R;

return S;

}

public static double Get\_Volume(double R)

{

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

return V;

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | Длина 31,416  Площадь 78,540  Объём 392,699 |

Анализ результатов:

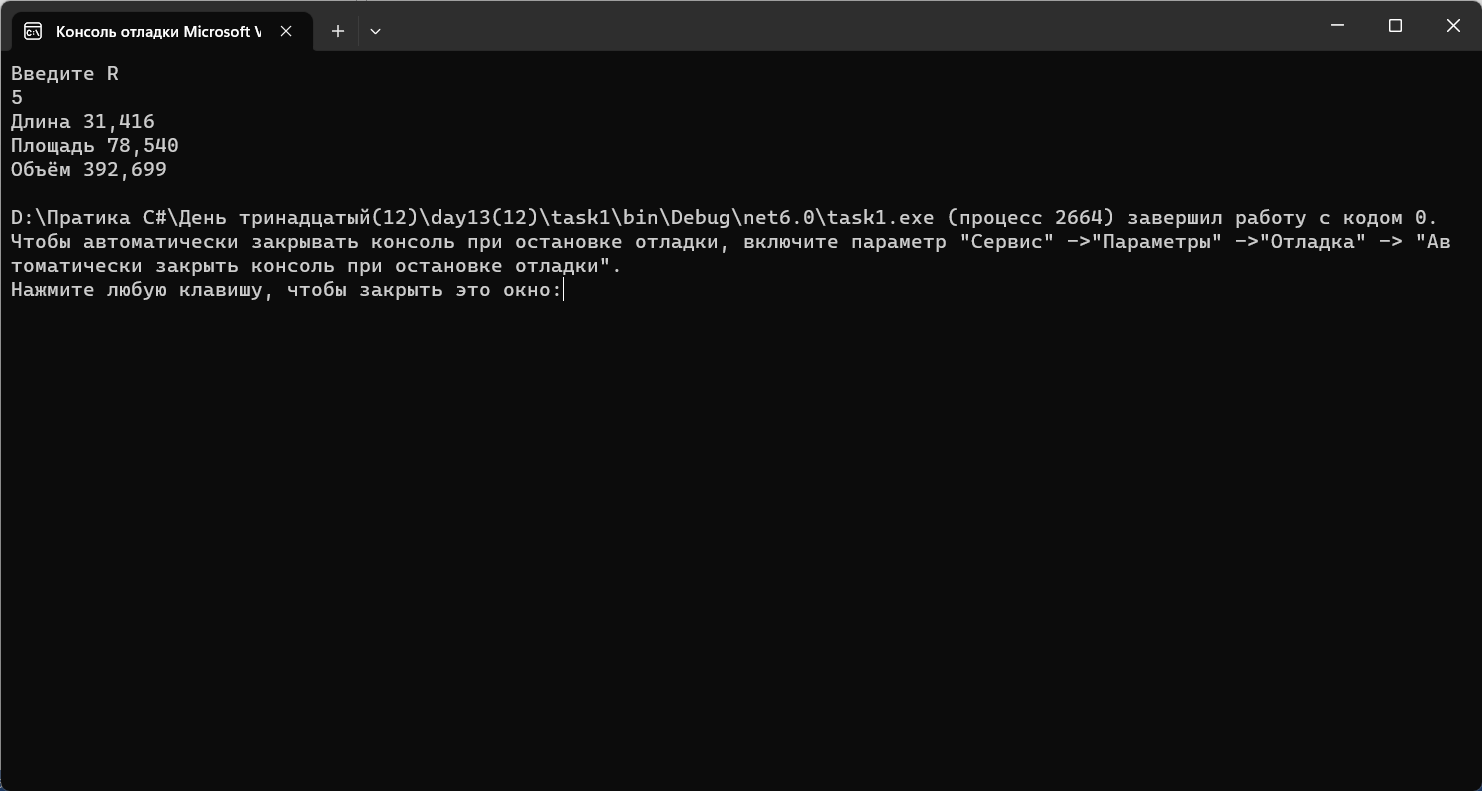


Рисунок 1.1 –работы программы

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.

namespace task2

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите х");

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

Console.WriteLine("Введите y");

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

MathOperators MO;

Console.WriteLine("Введите: \n 1 - Add\n 2 - Sub\n 3 - Mul\n 4 - Div");

int number = int.Parse(Console.ReadLine());

try

{

switch (number)

{

case 1:

MO = Add;

break;

case 2:

MO = Sub;

break;

case 3:

MO = Mul;

break;

case 4:

MO = Div;

break;

default: throw new Exception("Нет такого");

}

double rezult = MO(x, y);

Console.WriteLine($"Ответ {rezult}");

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

class Zad2

{

public delegate double MathOperators(double x, double y);

public static MathOperators Add = (double x, double y) => { return x + y; };

public static MathOperators Sub = (double x, double y) => { return x - y; };

public static MathOperators Mul = (double x, double y) => { return x \* y; };

public static MathOperators Div = (double x, double y) => { if (y != 0) return x / y; else throw new Exception("Деление на ноль"); };

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3,10,2 | -7 |

Анализ результатов:

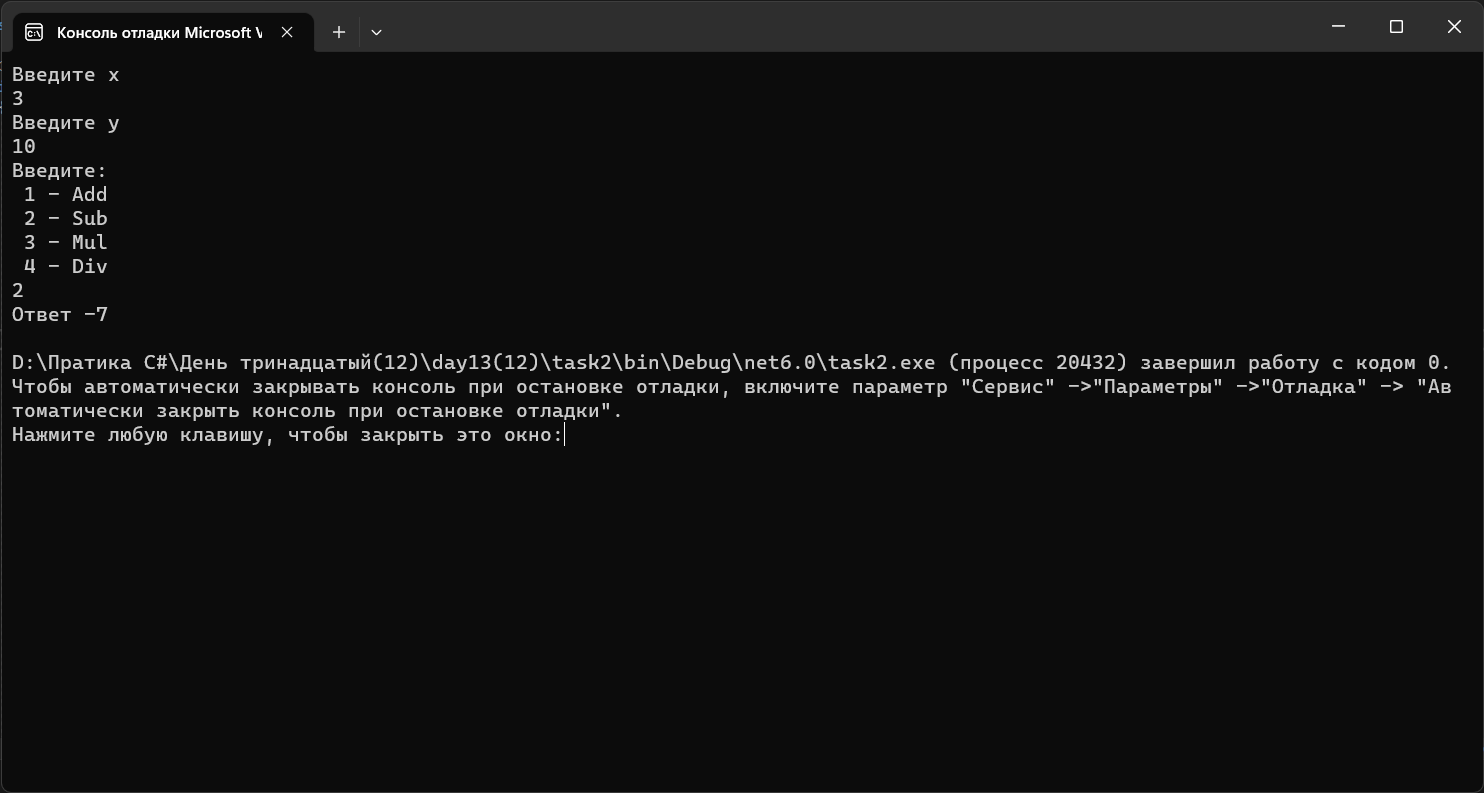
**

Рисунок 1.2 – Результат работы программы

работы программы