**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:

class Program

{

static double Get\_Length(double r)

{

double D;

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

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

return V;

}

public delegate double CalcFigure(double R);

static void Main(string[] args)

{

Console.Write("Введите R ");

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

CalcFigure CF = new CalcFigure(Get\_Length);

CF += Get\_Area;

CF += Get\_Volume;

if (CF != null) CF(rad);

}}

Таблица 11.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| R 3,5 | Длина окружности 21,991  Площадь круга 38,485  Объём шара 134,696 |

Анализ результатов:

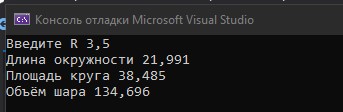


Рисунок 11.1 – Результат работы программы

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.

Листинг программы:

class Program

{

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

static void Main(string[] args)

{

Arif Add = (x, y) =>

{

Console.WriteLine("Результат={0}", x + y);

return x + y;

};

Arif Sub = (x, y) =>

{

Console.WriteLine("Результат={0}", x - y);

return x - y;

};

Arif Mul = (x, y) =>

{

Console.WriteLine("Результат={0}", x \* y);

return x \* y;

};

Arif Div = (x, y) =>

{

if (y == 0)

{

Console.WriteLine("Ошибка деление на 0!");

}

else

Console.WriteLine("Результат={0}", x / y);

return x / y;

};

Console.Write("Введите значение х- ");

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

Console.Write("Введите значение у- ");

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

Console.WriteLine("ВЫБЕРИТЕ ДЕЙСТВИЕ:\n1.Сложение \n2.Вычитание \n3.Умножение \n4.Деление");

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

switch (i)

{

case 1: if (Add != null) Add(X, Y); break;

case 2: if (Sub != null) Sub(X, Y); break;

case 3: if (Mul != null) Mul(X, Y); break;

case 4: if (Div != null) Div(X, Y); break;

default: Console.WriteLine("Нет такого пункта!"); break;

}

Console.ReadLine();

}

Таблица 12.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| х- 23,4  у- 55,3  3.Умножение | Результат=1294,0199999999998 |

Анализ результатов:

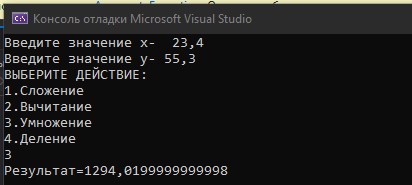


Рисунок 12.2 – Результат работы программы

Источник: собственная разработка