**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:

Console.Write("Enter r: ");

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

CalcFigure cf = GetLength;

cf += GetArea;

cf += GetVolume;

Console.WriteLine($"Result of delegate execution: {cf(r):F2}");

static double GetLength(double r)

{ Console.WriteLine("Circumference: ");

return 2 \* Math.PI \* r;

}

static double GetArea(double r)

{

Console.WriteLine("Area of a circle:");

return Math.PI\* r \*r;

}

static double GetVolume(double r)

{

Console.WriteLine("Ball volume:");

return 4 / 3 \* Math.PI \* r \* r \* r;

}

delegate double CalcFigure(double r);

Таблица 12.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| R = 5 | Результат выполнения делегата:392,70 |

Анализ результатов:

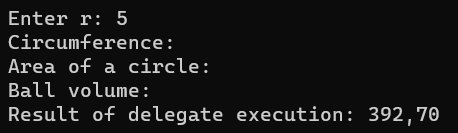


Рисунок 12.1 – Результат работы программы из задания №1

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:

delegate int Func(int a, int b);

class Program

{

static void Main()

{ Func<int, int, int> f = (a, b) => a + b;

Func g = (a, b) => a - b;

Func h = (a, b) => a \* b;

Func i = (a, b) => a / b;

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

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

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

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

Console.WriteLine("Выберите операцию: \n1.Add \n2.Sub \n3.Mul\n4.Div");

string c = Console.ReadLine();

switch (c)

{ case "1":

Console.WriteLine($"Выражение равно: {f(a, b)}");

break;

case "2":

Console.WriteLine($"Выражение равно: {g(a, b)}");

break;

case "3":

Console.WriteLine($"Выражение равно: {h(a, b)}");

break;

case "4":

Console.WriteLine($"Выражение равно: {i(a, b)}");

if (b == 0)

{

Console.WriteLine("На ноль делить нельзя, перезапустите программу");

}

break;

default:

Console.WriteLine("Такой операции нет");

break;

Таблица 12.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| A=5  B=6 | Сумма: 11 |

Анализ результатов:

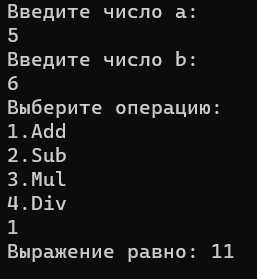


Рисунок 12.2 – Результат работы программы из задания №2

Задание №3. Создать 3 метода для работы со строкой. Создать делегат. В программе вызывать все три метода при помощи делегата.

Листинг программы:

Console.Write("Enter string: ");

string str = Console.ReadLine();

WorkerString ws = ToLower;

ws += ToUpper;

ws += Replace;

Console.WriteLine($"\r\nExecution result: {ws(str)}");

static string ToUpper(string str) => str.ToUpper();

static string ToLower(string str) => str.ToLower();

static string Replace(string str) => str.Replace("а","о");

delegate string WorkerString(string str);

Таблица 12.3 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Строка : Сумка | Результат выполнения:Сумко |

Анализ результатов:

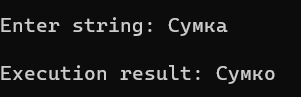


Рисунок 12.3 – Результат работы программы из задания №2

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 с делегатами из массива, возвращают случайное значение типа int.

Листинг программы:

public delegate int Number();

public delegate int Calc(Number[] arrayX);

class Program

{

public static int Randomizer()

{

Random rand = new Random();

return rand.Next(1, 10);

}

static void Main(string[] args)

{

Func<int>[] numArr = new Func<int>[5];

Console.Write("Массив: ");

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

{

numArr[i] = Randomizer;

Console.Write(" " + numArr[i].Invoke() + " ");

}

Console.WriteLine();

var mediumCalc = delegate (Func<int>[] array)

{

int sum = 0;

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

{

sum += array[i]();

}

return ((sum) / (array.Length));

};

Console.WriteLine($"Среднее арифметичекое: {mediumCalc(numArr)}");

Console.ReadKey();

}

Таблица 12.4 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Массив:5,4,3,4,7  Среднее арифметическое:5 |

Анализ результатов:



Рисунок 12.4 – Результат работы программы из задания №2