**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:

delegate double CalcFigure(double R);static void Main(string[] args) {CalcFigure CF = GetLength; CF += GetArea; CF += GetVolume; Console.Write("Введите радиус: ");double radius = Convert.ToDouble(Console.ReadLine()); Console.WriteLine("Радиус: {0}", radius); Console.WriteLine("Длина окружности: {0}", CF(radius)); Console.WriteLine("Площадь круга: {0}", CF(radius)); Console.WriteLine("Объем шара: {0}", CF(radius));} static double GetLength(double R) { return 2.0 \* Math.PI \* R; } static double GetArea(double R) { return Math.PI \* R \* R; }

static double GetVolume(double R) {return 4.0 / 3.0 \* Math.PI \* R \* R \* R; }

Таблица 12.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | Радиус: 5  Длина окружности: 523,5987755982989  Площадь круга: 523,5987755982989  Объем шара: 523,5987755982989 |

Источник: собственная разработка

Анализ результатов:

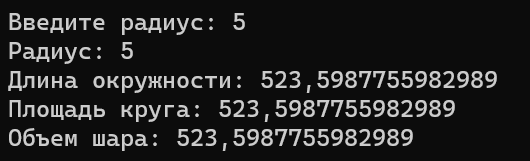


Рисунок 12.1 – Результат работы программы

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.

Листинг программы:

Func<double, double, double> Add = (x, y) => x + y;

Func<double, double, double> Sub = (x, y) => x - y;

Func<double, double, double> Mul = (x, y) => x \* y;

Func<double, double, double> Div = (x, y) => (y != 0) ? (x / y) : double.NaN;

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

double num1 = double.Parse(Console.ReadLine());

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

double num2 = double.Parse(Console.ReadLine());

Console.WriteLine("Выберите арифметическое действие (+, -, \*, /): ");

char operation = char.Parse(Console.ReadLine());

switch (operation){case '+': Console.WriteLine("Результат сложения: {0}", Add(num1, num2)); break; case '-':Console.WriteLine("Результат вычитания: {0}", Sub(num1, num2));break; case '\*':Console.WriteLine("Результат умножения: {0}", Mul(num1, num2)); break; case '/': Console.WriteLine("Результат деления: {0}", Div(num1, num2)); break;

default: Console.WriteLine("Некорректный оператор"); break; }

Таблица 12.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5; 5; \* | Результат умножения: 25 |

Источник: собственная разработка

Анализ результатов:

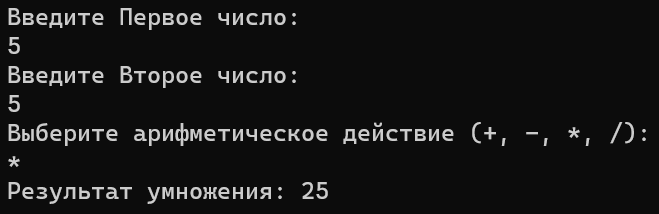


Рисунок 12.2 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

delegate string StringDelegate(string str); static void Main(string[] args)

{ StringDelegate stringDelegate = null;

stringDelegate += RemoveSpaces; stringDelegate += ToUpperCase;

stringDelegate += AddExclamationMark; string input = Console.ReadLine();

string result = AddExclamationMark(ToUpperCase(RemoveSpaces(input)));

Console.WriteLine(result);} static string RemoveSpaces(string str)

{return str.Replace(" ", "");}static string ToUpperCase(string str)

{return str.ToUpper();}static string AddExclamationMark(string str)

{ return str + "!";}

Таблица 12.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| мишка съел мёд свой | МИШКАСЪЕЛМЁДСВОЙ! |

Источник: собственная разработка

Анализ результатов:



Рисунок 12.3 – Результат работы программы

Источник: собственная разработка

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:

Func<int>[] delegates = new Func<int>[]

{ () => new Random().Next(1,100), () => new Random().Next(1,100),

() => new Random().Next(1,100),};

Func<Func<int>[], double> average = delegate (Func<int>[] funcs)

{ double sum = 0;foreach (var func in funcs)

{ sum += func();} return (double)sum / funcs.Length;};

Console.WriteLine("Среднее арифметическое: " + average(delegates));

Таблица 12.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Среднее арифметическое: 49,666666666666664 |

Источник: собственная разработка

Анализ результатов:



Рисунок 12.4 – Результат работы программы

Источник: собственная разработка