**13 События**

Задание 1. При решении задач, предусмотреть обработку исключительных ситуаций. Создать 3 метода для подсчета значений трех разных функций. Создать делегат. В программе вызывать все три метода при помощи делегата. Создать метод, в котором делегат будет параметром.

Листинг программы:

delegate double MyDelegate(double x); static void Main(string[] args)

{try{MyDelegate myDelegate = Method1; myDelegate += Method2;

myDelegate += Method3;

Console.WriteLine("Результаты методов:");

foreach (MyDelegate method in myDelegate.GetInvocationList())

{try{double result = method(2); Console.WriteLine($"{method.Method.Name}: {result}");}

catch (Exception ex){Console.WriteLine($"Ошибка при выполнении метода {method.Method.Name}: {ex.Message}"); } }

DoSomething(myDelegate, 2); } catch (Exception ex) {

Console.WriteLine($"Ошибка: {ex.Message}"); } }

static double Method1(double x) { return x \* x;}

static double Method2(double x) { return Math.Pow(x,2); }

static double Method3(double x) { return Math.Exp(x); }

static void DoSomething(MyDelegate myDelegate, double x)

{try {foreach (MyDelegate method in myDelegate.GetInvocationList())

{double result = method(x); Console.WriteLine($"Результат метода: {result}");}} catch (Exception ex){ Console.WriteLine($"Ошибка при выполнении метода DoSomething: {ex.Message}");} }

Таблица 13.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | Результаты методов:  Method1: 4  Method2: 4  Method3: 7,38905609893065  Результат метода: 4  Результат метода: 4  Результат метода: 7,38905609893065 |

Источник: собственная разработка

Анализ результатов:

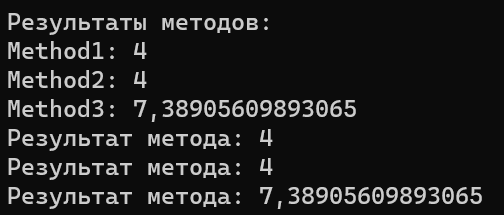


Рисунок 13.1 – Результат работы программы

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:

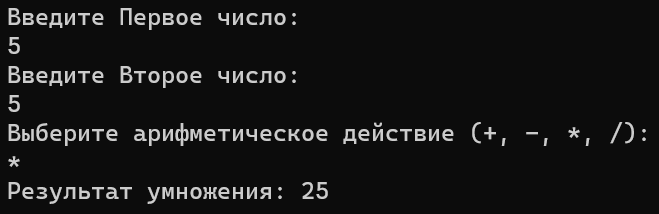


Рисунок 12.2 – Результат работы программы

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