Делегаты

Задание №1. 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.

Листинг программы:

namespace \_12

{

delegate void DelegatCalculator(); //объявление делегата

class Program

{

static void Main()

{

Console.Write("Введите действие (+,-,\*,/): ");

string a = Console.ReadLine();

switch (a) //условынй оператор

{

case "+":

DelegatCalculator calcA = new DelegatCalculator(Add); //создание переменной в которой хранится делегат сложения

calcA(); //вызов переменной

break;

case "-":

DelegatCalculator calcS = new DelegatCalculator(Sub);

calcS();

break;

case "\*":

DelegatCalculator calcM = new DelegatCalculator(Mul);

calcM();

break;

case "/":

DelegatCalculator calcD = new DelegatCalculator(Div);

calcD();

break;

default:

Console.WriteLine("Ошибка ввода операции!");

break;

}

static (int, int) Input() //ввод значений переменных

{

Console.Write("Первое число: ");

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

Console.Write("Второе число: ");

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

return (x, y);

}

static void Add()

{

(int c, int b) = Input();

Console.WriteLine("Ответ:{0}", c + b);

}

static void Sub()

{

(int c, int b) = Input();

Console.WriteLine("Ответ:{0}", c - b);

}

static void Mul()

{

(int c, int b) = Input();

Console.WriteLine("Ответ:{0}", c \* b);

}

static void Div()

{

try

{

(int c, int b) = Input();

if (b == 0) throw new DivideByZeroException();

Console.WriteLine("Ответ:{0}", c / b);

}

catch

{

Console.WriteLine("Деление на ноль!");

}

}

}

}

}

Анализ программы:

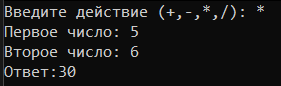


Рисунок 1.1 - Анализ программы

Источник: собственная разработка

Задание №2. Создать 3 метода для работы со строкой. Создать делегат. В программе вызывать все три метода при помощи делегата.

Листинг программы:

using System;

namespace StringMethods

{

class Program

{

delegate void StringDelegate(string str);

static void Main(string[] args)

{

StringDelegate stringDelegate = new StringDelegate(ReverseString);

stringDelegate += UppercaseString;

stringDelegate += LowercaseString;

stringDelegate("Hello World!");

}

static void ReverseString(string str)

{

char[] charArray = str.ToCharArray();

Array.Reverse(charArray);

Console.WriteLine(new string(charArray));

}

static void UppercaseString(string str)

{

Console.WriteLine(str.ToUpper());

}

static void LowercaseString(string str)

{

Console.WriteLine(str.ToLower());

}

}

}

Анализ программы:

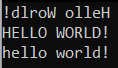


Рисунок 1.2 - Анализ программы

Источник: собственная разработка

Задание №3. 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().

Листинг программы:

using System;

namespace FigureCalculations

{

class Program

{

delegate double CalcFigure(double R);

static void Main(string[] args)

{

CalcFigure CF = null;

CF += Get\_Length;

CF += Get\_Area;

CF += Get\_Volume;

double radius = 5.0;

foreach (CalcFigure method in CF.GetInvocationList())

{

double result = method(radius);

Console.WriteLine($"Результат: {result}");

}

}

static double Get\_Length(double R)

{

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

return D;

}

static double Get\_Area(double R)

{

double S = Math.PI \* Math.Pow(R, 2);

return S;

}

static double Get\_Volume(double R)

{

double V = (4.0 / 3.0) \* Math.PI \* Math.Pow(R, 3);

return V;

}

}

}

Анализ программы:



Рисунок 1.3 - Анализ программы

Источник: собственная разработка