1. **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 · 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:

namespace Task\_1

{

class Program

{

public static void Main(string[] args)

{

CalkFigur cf = new CalkFigur(GetLength);

cf += GetArea;

cf += GetVolume;

Console.Write("Введите радиус: ");

cf(double.Parse(Console.ReadLine()));

}

public delegate double CalkFigur(double radius);

static double GetLength(double r)

{

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

Console.WriteLine($"Длинна окружности: {D}");

return D;

}

static double GetArea(double r)

{

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

Console.WriteLine($"Площадь круга: {S}");

return S;

}

static double GetVolume(double r)

{

double V = (4 / 3) \* Math.PI \* Math.Pow(r, 3);

Console.WriteLine($"Обьем шара: {V}");

return V;

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | 12,56… 12,56… 25,13… |

Анализ результатов:

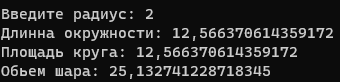


Рисунок 1.1 – Результат работы программы

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:

namespace Task\_2

{

delegate void DelegatCalculator();

class Program

{

static void Main()

{

try

{

MyEvent evt = new MyEvent();

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

string a = Console.ReadLine();

switch (a)

{

case "+":

evt.activate += new DelegatCalculator(Add);

evt.fire();

break;

case "-":

evt.activate += new DelegatCalculator(Sub);

evt.fire();

break;

case "\*":

evt.activate += new DelegatCalculator(Mul);

evt.fire();

break;

case "/":

evt.activate += new DelegatCalculator(Div);

evt.fire();

break;

default:

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

break;

}

}

catch (Exception e)

{

Console.WriteLine("Ошибка \n{0}", e);

}

}

static (int, int) Input()

{

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!");

}

}

}

class MyEvent

{

public event DelegatCalculator activate;

public void fire()

{

if (activate != null)

{

activate();

}

}

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 5 | 7 |

Анализ результатов:

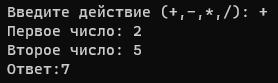


Рисунок 2.1 – Результат работы программы

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:

namespace Task\_3

{

public delegate void StringOperation(string[] str);

class Program

{

public static void Main(string[] args)

{

try

{

Console.Write("Введите слово затем что заменить в слове на что: ");

string[] array = Console.ReadLine().Split(" ").ToArray();

StringOperation oper = new StringOperation(StringReplace);

oper(array);

Console.Write("Введите слово затем индекс и то что надо вставить: ");

array = Console.ReadLine().Split(" ").ToArray();

oper = StringInsert;

oper(array);

Console.Write("Введите слово затем индекс с какого момента произвести удаление\nи сколько удалить: ");

array = Console.ReadLine().Split(" ").ToArray();

oper = StringDelete;

oper(array);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

public static void StringReplace(string[] str)

{

Console.WriteLine(str[0].Replace(str[1], str[2]));

}

public static void StringInsert(string[] str)

{

Console.WriteLine(str[0].Insert(int.Parse(str[1]), str[2]));

}

public static void StringDelete(string[] str)

{

Console.WriteLine(str[0].Remove(int.Parse(str[1]), int.Parse(str[2])));

}

}

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Пастух ух ила  Пожар 1 р  Пастух 3 2 | Пастила  Прожар  Пасх |

Анализ результатов:

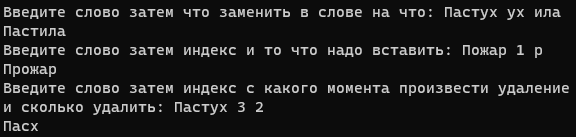


Рисунок 3.1 – Результат работы программы

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:

namespace Task\_4

{

public delegate int DelegatCalculator();

public delegate int MediumCalcc(DelegatCalculator[] array);

class Program

{

private static readonly Random rnd = new Random();

public static int Randomizer()

{

return rnd.Next(10);

}

static void Main()

{

try

{

Console.WriteLine("Числа для выполнения последовательных орераций с ними (+,-,\*,/)");

DelegatCalculator[] delCalc = new DelegatCalculator[4];

delCalc[0] = new DelegatCalculator(Add);

delCalc[1] = new DelegatCalculator(Sub);

delCalc[2] = new DelegatCalculator(Mul);

delCalc[3] = new DelegatCalculator(Div);

MediumCalcc mediumCalc = delegate (DelegatCalculator[] array)

{

int sum = 0;

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

{

sum += array[i]();

}

Console.WriteLine($"Сумма: {sum}");

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

};

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

}

catch (Exception e)

{

Console.WriteLine("Ошибка \n{0}", e);

}

}

static (int, int) Input()

{

int x = Randomizer();

int y = Randomizer();

return (x, y);

}

static int Add()

{

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

Console.WriteLine($"Ответ:{c + b}");

int z = c + b;

return z;

}

static int Sub()

{

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

Console.WriteLine($"Ответ:{c - b}");

int z = c - b;

return z;

}

static int Mul()

{

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

Console.WriteLine($"Ответ:{c \* b}");

int z = c \* b;

return z;

}

static int Div()

{

try

{

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

if (b == 0)

Console.WriteLine($"Ответ:{c / b}");

int z = c / b;

return z;

}

catch (DivideByZeroException e)

{

Console.WriteLine(e.Message);

return 0;

}

}

}

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 6 | 8 -2 36 42 10 |

Анализ результатов:

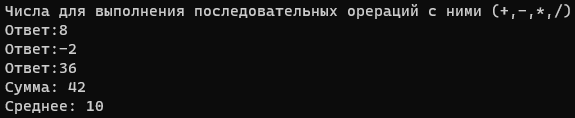


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка