**13. Событие**

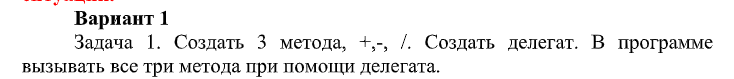
****Задание 1. Объявить тип делегата, который ссылается на метод. Требования к сигнатуре метода следующие:

Рисунок 13.1 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Schema;

namespace zad1

{

internal class Program

{

delegate double JustOperation(double x , double y);

static void Main(string[] args)

{

Console.WriteLine("Введите x");

double xNum = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите x");

double yNum = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Выбор действия: ");

Console.WriteLine("1 - ADD, 2 - SUB, 3 - DIV");

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

double res;

switch (Operation)

{

case 1:

JustOperation addder = Add;

res = addder(xNum, yNum);

Console.WriteLine(res);

break;

case 2:

JustOperation subliciton = Sub;

res = subliciton(xNum, yNum);

Console.WriteLine(res);

break;

case 3:

JustOperation divizion = Div;

res = divizion(xNum, yNum);

Console.WriteLine(res);

break;

default:

Console.WriteLine("Операции не существует! ");

break;

}

}

public static double Add(double x, double y) => x + y;

public static double Sub(double x, double y) => x - y;

public static double Div(double x, double y) => x / y;

}

}

Таблица 13.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| x = 5, y = 3 | 2 |

Анализ результатов:

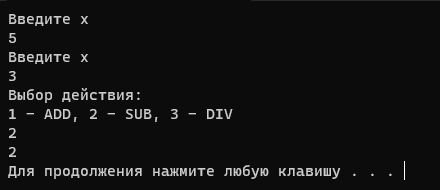


Рисунок 13.2 – Результат работы программы

Источник: собственная разработка

Задание 2.

Рисунок 13.3 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Schema;

namespace zad2

{

delegate double JustOperation(double x, double y);

internal class Program

{

public static void Main(string[] args)

{

Console.WriteLine("Введите x");

double xNum = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите x");

double yNum = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Выбор действия: ");

Console.WriteLine("1 - ADD, 2 - SUB, 3 - DIV");

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

double res;

switch (Operation)

{

case 1:

JustOperation addder = Add;

res = DelegateWatch(xNum, yNum, addder);

Console.WriteLine(res);

break;

case 2:

JustOperation subliciton = Sub;

res = DelegateWatch(xNum, yNum, subliciton);

Console.WriteLine(res);

break;

case 3:

JustOperation divizion = Div;

res = DelegateWatch(xNum, yNum, divizion);

Console.WriteLine(res);

break;

default:

Console.WriteLine("Операции не существует! ");

break;

}

}

public static double DelegateWatch(double xNum, double yNum, JustOperation oper) {

double res = oper(xNum, yNum);

return res;

}

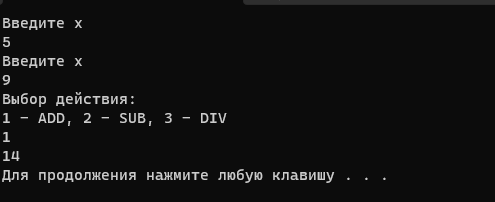
public static double Add(double x, double y) => x + y;

public static double Sub(double x, double y) => x - y;

public static double Div(double x, double y) => x / y;

}

}

Таблица 13.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| x = 5, y = 9 | 14 |

Анализ результатов:

Рисунок 13.4 – Результат работы программы

Источник: собственная разработка

Задание 3.

Рисунок 13.5 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.IO;

namespace zad3.MyInfo

{

internal class Program

{

public static void Main(string[] args)

{

Console.Write("Введите имя: ");

string name = Console.ReadLine();

MyInfo myinfo = new MyInfo(name);

myinfo.Act += Inf\_Act;

Console.WriteLine($"{myinfo.Name} будет изменено на: ");

myinfo.Name = Console.ReadLine();

string end = Console.ReadLine();

while (end != "end")

{

myinfo.Name = name;

Console.WriteLine("Введите имя");

name = Console.ReadLine();

string ends = Console.ReadLine();

}

Console.WriteLine();

}

private static void Inf\_Act(string obj)

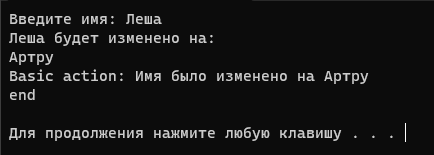
{

Console.WriteLine($"Basic action: {obj}");

}

}

}

Таблица 13.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Леша | Артру |

Анализ результатов:

Рисунок 13.6 – Результат работы программы

Источник: собственная разработка