МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчёт по лабораторной работе №4

**«Делегаты и события»**

Работу выполнил  
студент 41 группы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.C. Онянов

Направление подготовки 01.03.02 Прикладная математика и информатика

Проверил  
канд. техн. наук, доц.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Полупанов

Краснодар 2021

**Цель работы:** Знакомство с делегатами и событиями в C#.

**Практическая часть:** Использование делегата для вызова математических функций

using System;

using System.Text;

namespace Example1

{

class MathOprt

{

public static double Mul2(double val)

{

return val \* 2;

}

public static double Sqr(double val)

{

return val \* val;

}

}

delegate double DblOp(double x);

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

DblOp[] operation =

{

new DblOp(MathOprt.Mul2),

new DblOp(MathOprt.Sqr)

};

for (int j = 0; j < operation.Length; j++)

{

Console.WriteLine("Резльтаты операции[{0}]:", j);

Prc(operation[j], 4.0);

Prc(operation[j], 9.94);

Prc(operation[j], 3.143);

}

}

static void Prc(DblOp act, double val)

{

double rslt = act(val);

Console.WriteLine("Исходное значение {0}, результат {1}", val, rslt);

}

}

}

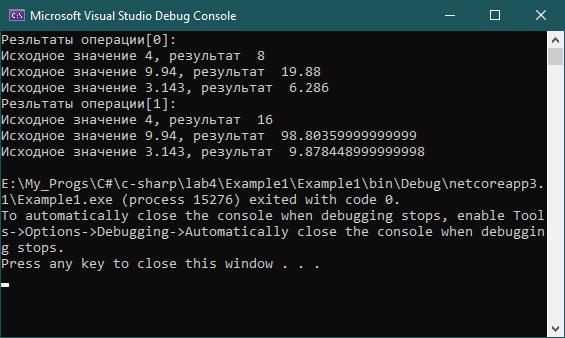


Рисунок 1.

В предыдущем примере был использован делегат, который ссылался только 1 область памяти. Однако делегаты могут ссылаться на несколько методов сразу. В данном случае методы будут вызываться друг за другом последовательно.

using System;

using System.Text;

namespace Example2

{

class Program

{

class MathOprt

{

public static void Mul2(double val)

{

double rslt = val \* 2;

Console.WriteLine("Mul2 bсходное значение {0},результат {1}",

val, rslt);

}

public static void Sqr(double val)

{

double rslt = val \* val;

Console.WriteLine("Sqr исходное значение {0}, результат {1}",

val, rslt);

}

}

delegate void DblOp(double x);

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

DblOp operations = new DblOp(MathOprt.Mul2);

operations += new DblOp(MathOprt.Sqr);

Prc(operations, 4.0);

Prc(operations, 9.94);

Prc(operations, 3.143);

}

static void Prc(DblOp act, double val)

{

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\n");

act(val);

}

}

}

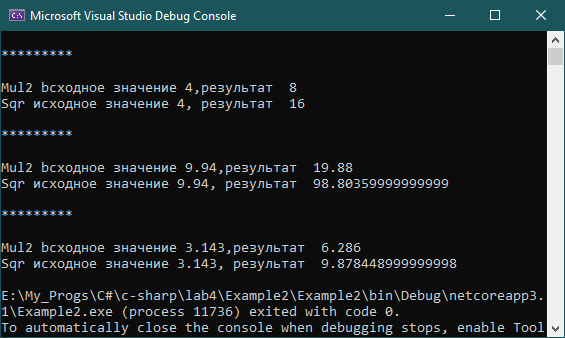


Рисунок 2.

Использование событий

using System;

using System.Text;

namespace Example3

{

class Program

{

class ChangeEventArgs : EventArgs

{

string str;

public string Str

{

get

{

return str;

}

}

int change;

public int Change

{

get

{

return change;

}

}

public ChangeEventArgs(string str, int change)

{

this.str = str;

this.change = change;

}

}

class GenEvent // Генератор событий - издатель

{

public delegate void ChangeEventHandler(object source, ChangeEventArgs e);

public event ChangeEventHandler OnChangeHandler;

public void UpdateEvent(string str, int change)

{

if (change == 0)

return;

ChangeEventArgs e = new ChangeEventArgs(str, change);

if (OnChangeHandler != null)

{

OnChangeHandler(this, e);

}

}

}

//Подписчик

class RecEvent

{

//Обработчик события

void OnRecChange(object source, ChangeEventArgs e)

{

int change = e.Change;

Console.WriteLine("Вес груза '{0}' был {1} на {2} тонны",

e.Str, change > 0 ? "увеличен" : "уменьшен",

Math.Abs(e.Change));

}

GenEvent gnEvent;

// в конструкторе класса осуществляется подписка

public RecEvent(GenEvent gnEvent)

{

this.gnEvent = gnEvent;

gnEvent.OnChangeHandler += //здесь осуществляется подписка

new GenEvent.ChangeEventHandler(OnRecChange);

}

}

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

GenEvent gnEvent = new GenEvent();

RecEvent inventoryWatch = new RecEvent(gnEvent);

gnEvent.UpdateEvent("грузовика", -2);

gnEvent.UpdateEvent("автопоезда", 4);

}

}

}

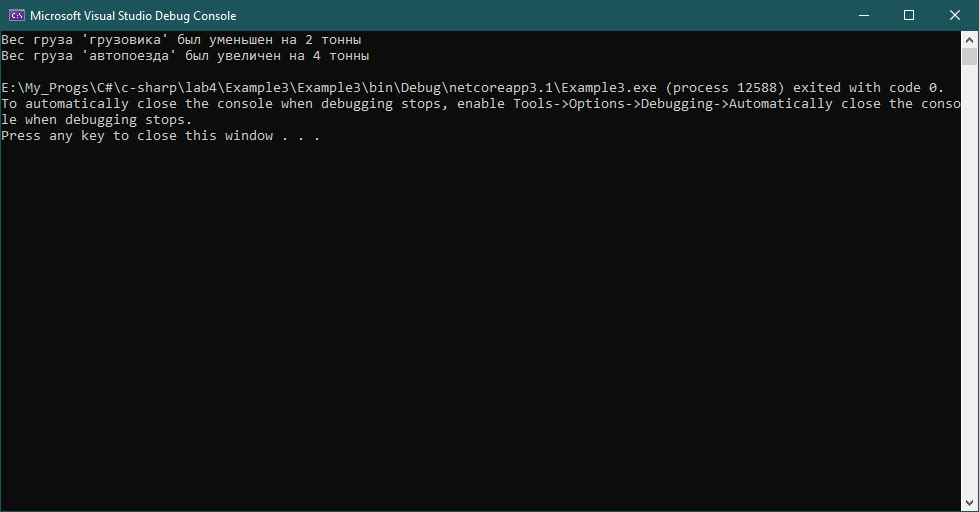


Рисунок 3.

**Вывод:** В ходе лабораторной работы были изучены структуры делегатов и событий. Изучены их потенциальные возможности.