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 в C#»**

Работу выполнил  
студент 42 группы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Б.М. Ибрагимов

Направление подготовки 01.03.02 Прикладная математика и информатика

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

Краснодар 2020

**Цель работы:** является изучение делегатов и событий в C#.

**Задача 1:** ввести и выполнить приложения из примеров в лабораторной. Проанализировать и объяснить работу приложений по примерам.

**ЛИСТИНГ ПРОГРАММЫ**

**Main Class**

using System;

namespace lab04 {

class Program {

public static void Ex1() {

DblOp[] operation = // создание экземпляров делегата

{

new DblOp(MathOprt.Mul2),

new DblOp(MathOprt.Sqr)

};

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

Console.WriteLine("Результаты 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);

}

}

static void Main(string[] args) {

System.Console.WriteLine("---------------Ex1----------------");

Ex1();

System.Console.WriteLine("---------------Ex2----------------");

Class1.Ex2();

System.Console.WriteLine("---------------Ex3----------------");

Class2.Ex3();

}

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);//объявление делегата

}

}

**Class1**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab04 {

class Class1 {

public static void Ex2() {

//

// TODO: Add code to start application here

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);

}

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);//объявление делегата

}

}

**Class2**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab04 {

class Class2 {

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);

}

}

public static void Ex3() {

GenEvent gnEvent = new GenEvent();

RecEvent inventoryWatch = new RecEvent(gnEvent);

gnEvent.UpdateEvent("грузовика", -2);

gnEvent.UpdateEvent("автопоезда", 4);

}

}

}

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

Изображение выглядит как черный

Автоматически созданное описание

Рисунок 1 – результат выполнения работы программы

**Вывод:** язык С# имеет делегаты и события. Делегаты предназначены для передачи в качестве параметров одних методов другим методам, могут хранить несколько адресов областей памяти (многоадресность делегатов). Работа с событиями осуществляется в C# согласно модели «издатель-подписчик». Класс, ответственный за инициализацию событий публикует событие, и любые классы могут подписаться на это событие. При возникновении события исполняющая среда уведомляет всех подписчиков о произошедшем событии, при этом вызываются соответствующие методы- обработчики событий подписчиков. Какой обработчик события будет вызван – определяется делегатом.