Лабораторная работа №14

**Тема:** Разработка классов с событиями и использование их в программах.

**Цель:** Научиться разрабатывать классы с событиями и использовать их в программах.

**Техническое оснащение:** MS Windows 10, MS Word 2019, MS VS 2019, MS Visio 2019.

Выполнение работы

**Пример.** Применение события

Выполнение примера (рисунок 1).

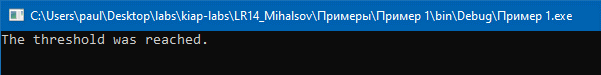


Рисунок 1 – Проверка работы примера 1

**Задание.** Используя среду разработки Microsoft Visual Studio, создать консольное приложение на языке программирования С#. На основе результата лабораторной работы №11 реализовать обработку событий, реагирующих на изменение данных в реализованных структурах. Реакции на событие должны быть в виде вывода сообщения на экран. Ввод данных должен производиться с клавиатуры.

Листинг программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Задание

{

class Program

{

interface IPersona

{

string getName();

string getId();

}

interface IWorker

{

string getWorkRang();

string getActivity();

}

interface IEngineer

{

string getActivityProject();

string getSecrecyLevel();

}

interface IAdmin

{

string getDeportament();

string getadminLevel();

}

static public void PrintAction(string action)

{

Console.WriteLine(action);

}

class Persona : IPersona

{

string id;

public string name;

public string getId()

{

return id;

}

public string getName()

{

return name;

}

public Persona(string name, string id)

{

this.id = id;

this.name = name;

}

public Persona() { }

}

class Worker : Persona, IWorker

{

public delegate void Log(string message);

public event Log Loging;

public string getActivity()

{

return activity;

}

public string getWorkRang()

{

return workrang;

}

string activity;

string workrang;

public Worker(string name, string id) : base(name, id)

{

activity = "nowork";

workrang = "C9";

Loging += PrintAction;

}

public void setRang(string rang)

{

workrang = rang;

Loging.Invoke($"Работкник {name}, теперь имееет разряд {rang}");

}

public void setActivity(string activity)

{

this.activity = activity;

Loging.Invoke($"Работкник {name}, теперь {activity}");

}

}

class Engener : Worker, IEngineer

{

public string getActivityProject()

{

return activityproject;

}

public string getSecrecyLevel()

{

return secrecy;

}

string secrecy;

string activityproject;

public Engener(string name, string id, string secrecy, string activityproject) : base(name, id)

{

this.secrecy = secrecy;

this.activityproject = activityproject;

}

}

class Admin : Persona

{

string deportament;

string adminlevel;

public Admin(string name, string id, string adminlevel) : base(name, id)

{

this.adminlevel = adminlevel;

}

public void setDeportament(string deportament)

{

this.deportament = deportament;

}

}

static int lastid = -1;

public static string getNewId()

{

lastid++;

return lastid.ToString();

}

static void Main(string[] args)

{

List<Admin> Admins = new List<Admin>

{

new Admin("Артёмов К.А",getNewId(),"S0"),

new Admin("Андреев Г.А",getNewId(),"S1"),

new Admin("Кучинский Д.С",getNewId(),"S1"),

};

List<Engener> Engeners = new List<Engener>

{

new Engener("Кукуш З.С",getNewId(),"S3","KOLIMA3.0"),

new Engener("Генадьев Т.З",getNewId(),"S3","KOLIMA4.0"),

};

List<Worker> workers = new List<Worker>

{

new Worker("Мякишь П.А.",getNewId()),

new Worker("Масляков Н.А.",getNewId()),

new Worker("Цаль В.И.",getNewId()),

new Worker("Цой В.Р.",getNewId()),

};

List<Persona> deportam = new List<Persona>();

foreach (var item in Admins)

{

item.setDeportament("депортамент 1");

deportam.Add(item);

}

Console.WriteLine("Найм работкников:");

foreach (var item in Engeners)

{

item.setActivity("наботает над проектом " + item.getActivityProject());

deportam.Add(item);

}

string[] works = new string[] { "слесарь", "сварщик", "бригадир", "станочник" };

Random rnd = new Random();

foreach (var item in workers)

{

item.setRang("B1");

item.setActivity("работает по спецальности " + works[rnd.Next(0, works.Length)]);

deportam.Add(item);

}

List<Worker> workersandengeners = new List<Worker>();

foreach (var item in Engeners)

workersandengeners.Add(item);

foreach (var item in workers)

workersandengeners.Add(item);

Console.WriteLine("Отдел:");

foreach (var item in deportam)

Console.WriteLine("Списочный номер {0}, имя {1}", item.getId(), item.getName());

Console.WriteLine("\nДеятельность рабочих и инженеров:");

foreach (var item in workersandengeners)

{

Console.WriteLine("{0}, {1}", item.name, item.getActivity());

}

int iinput;

do

{

Console.WriteLine("0. Выход");

Console.WriteLine("1. Изменить деятельность рабочего");

int.TryParse(Console.ReadLine(), out iinput);

switch (iinput)

{

case 1:

Console.WriteLine("Выберете рабочего 3-8");

int.TryParse(Console.ReadLine(), out iinput);

if(3 <= iinput && iinput <= 8)

{

Console.WriteLine("Введите деятельность");

string activity = Console.ReadLine();

workersandengeners[iinput].setActivity(activity);

} else

{

Console.WriteLine("Неправельный номер ;(");

iinput = -1;

}

break;

default:

iinput = 0;

break;

}

} while (iinput != 0);

Console.ReadKey(true);

}

}

}

Проверка запуска программы (рисунок 2).

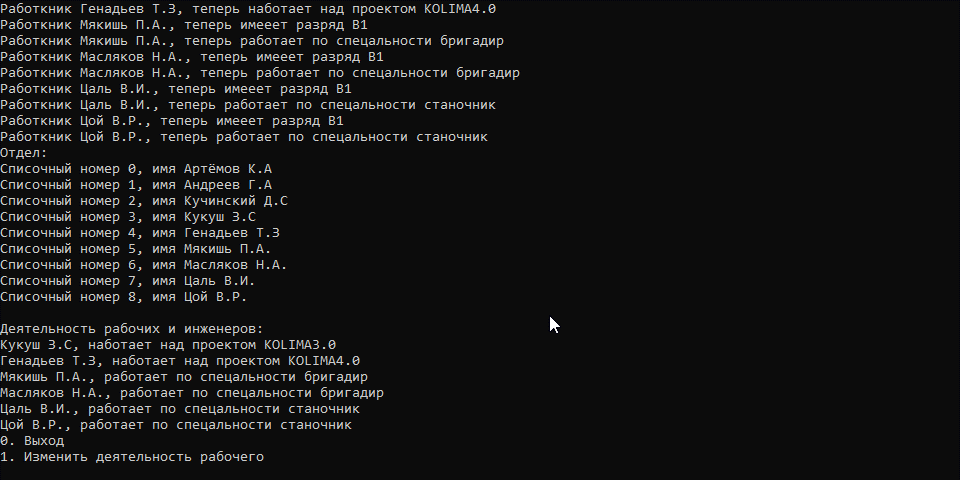


Рисунок 2 – Запуск программы

Проверка изменения деятельности рабочего и проверка обработки события (рисунок 3).

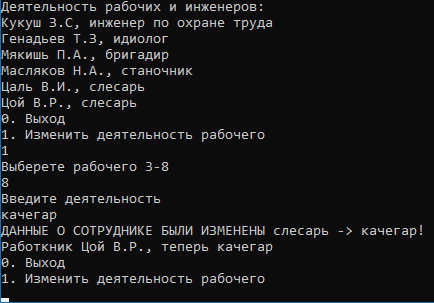


Рисунок 3 – Работа обработчика события

**Вывод:** В результате выполнения лабораторной работы были получены практические навыки по работе с событиями в С#.