МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

кафедра программного обеспечения и администрирования

информационных систем

Отчёт

по лабораторной работе №7

«Делегаты и события»

по дисциплине

###### «Объектно-ориентированные языки и системы»

|  |  |
| --- | --- |
| Выполнил(а): | студент группы 313  Козявин Максим  Сергеевич |
| Проверил: | к.т.н., доцент  кафедры ПОиАИС  Макаров К.С. |

Курск

2023

**Цель работы:** овладение основными приемами событийного программирования и их программной реализацией.

**Индивидуальное задание (вариант 4):**

**Задание 1:**

Разработать проект использование делегата для:

– вызова разных методов одним экземпляром делегата;

– многоадресной передачи методов с одинаковой сигнатурой в классе 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Параметры делегата** | **Методы возвращают:** | | |
| **Класс 1** | | **Класс 2** |
| **Статический метод** | **Метод экземпляра** |
| 4. | string | строку, удалив два первых символа | строку, заменив первый символ символом ‘\_’ | строку, удалив два последних символа |

**Задание 2:**

Создать событие на основе делегата (задание 1). В тестирующем классе организовать цепочку вызовов.

**Основные теоретические положения**

Делегат – это вид класса, предназначенный для хранения ссылок на

методы. Делегат представляет собой тип данных. Объявление делегата можно размещать непосредственно в пространстве имен или внутри класса. Тип описывает возвращаемое значение методов, вызываемых с помощью делегата, а необязательными параметрами делегата являются параметры этих методов. На основе делегатов построены события. Объект, которому необходима информация о некотором событии, регистрирует обработчик для этого события, сигнатура которых соответствует типу делегата. Когда ожидаемое событие происходит, вызываются все зарегистрированные обработчики. Добавлять и удалять обработчики событий или методов в список делегата можно с помощью операторов “+=” и “-=”.

**Экспериментальные результаты**

Код программы для решения задачи 1-2:

class Class1

{

public static Del StaticDel = DeleteFirstTwo;

public Del ObjectDel;

public VoidDel EventDel;

public delegate string Del(string s);

public delegate void VoidDel(string s);

public event VoidDel Event;

public Class1() {

ObjectDel = ChangeFirst;

EventDel = EventHandler1;

Event = EventDel;

}

public static string DeleteFirstTwo(string s)

{

return s[2..];

}

public string ChangeFirst(string s)

{

return "\_"+s[1..];

}

public void EventHandler1(string s)

{

Console.WriteLine("Handler 1: " + DeleteFirstTwo(s));

}

public void EventHandler2(string s)

{

Console.WriteLine("Handler 2: " + ChangeFirst(s));

}

public void DoEvent(string s)

{

Event.Invoke(s);

}

}

class Class2

{

public Del ObjectDel;

public delegate string Del(string s);

public Class2() {

ObjectDel = DeleteLastTwo;

}

public string DeleteLastTwo(string s)

{

return s[0..^2];

}

}

internal class Program

{

public delegate void Del();

static Del StaticDel = Method1;

private static void Main(string[] args)

{

Class1 cl1 = new();

Class2 cl2 = new();

Console.WriteLine(cl1.ObjectDel("abcde"));

Console.WriteLine(Class1.StaticDel("abcde"));

Console.WriteLine(cl2.ObjectDel("abcde"));

cl1.ObjectDel += cl2.DeleteLastTwo;

Console.WriteLine(cl1.ObjectDel("abcde"));

StaticDel += Method2;

StaticDel();

cl1.Event += cl1.EventHandler2;

cl1.DoEvent("abcde");

}

public static void Method1() {

Console.WriteLine("Method 1");

}

public static void Method2() {

Console.WriteLine("Method 2");

}

}

Тестирование:

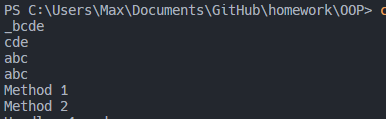


Рисунок 1 – Тест задачи 1



Рисунок 2 – Тест задачи 2

**Вывод:** овладел основными приемами событийного программирования и их программной реализацией на языке C#.