**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №6(часть 1)

«Работа с делегатами»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32 |  | преподаватель каф. ИУ5 |
| Барышников Михаил |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

**Задача**

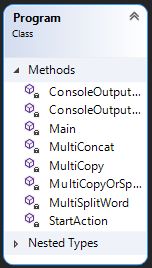
Разработать программу демонстрирующую работу с делегатами. Программа должна быть разработана в виде консольного приложения на языке C#. Определите делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа. Напишите метод, соответствующий данному делегату. Напишите метод, принимающий разработанный вами делегат, в качестве одного из входным параметров. Осуществите вызов метода, передавая в качестве параметра-делегата:

• только что разработанный метод;

• лямбда-выражение.

Повторите пункт те же действия, используя вместо разработанного вами делегата, обобщенный делегат Func< > или Action< >, соответствующий сигнатуре разработанного вами делегата.

**Диаграмма классов**



**Текст программы**

* Program.cs

using System;

namespace RealizedDelegates

{

class Program

{

delegate T FirstDelegate<T, Q, W>(Q firstParam, W secondParam);

delegate T SecondDelegate<T, Q>(params Q[] param);

static void Main(string[] args)

{

Console.WriteLine("Для теста используется 3 функции:\n1) Множественное копирование строки\n2) Многочисленное разделение символов строки\n3) Множественное копирование символов строки\n");

// Delegate

Console.BackgroundColor = ConsoleColor.DarkBlue;

Console.WriteLine("\nTest Delegate:\n");

Console.BackgroundColor = ConsoleColor.Black;

FirstDelegate<string, string, int> multiAction;

multiAction = MultiCopy;

Console.WriteLine("{0}\n", MultiCopyOrSplit(multiAction, "qwerty ", 22));

multiAction = MultiSplitWord;

Console.WriteLine("{0}\n", MultiCopyOrSplit(multiAction, "qwerty", 1));

multiAction = (str, countSymCopy) => {

for (int i = str.Length - 1; i != -1; --i)

str = str.Insert(i, MultiCopy(str[i].ToString(), countSymCopy - 1));

return str;

};

Console.WriteLine("{0}\n", MultiCopyOrSplit(multiAction, "qwerty", 4));

// Action

Console.BackgroundColor = ConsoleColor.DarkBlue;

Console.WriteLine("\nTest Action:\n");

Console.BackgroundColor = ConsoleColor.Black;

Action<string, int> actionDelegate = ConsoleOutputMultiCopy;

actionDelegate += ConsoleOutputMultiSplitWord;

actionDelegate += (str, countSymCopy) => {

for (int i = str.Length - 1; i != -1; --i)

str = str.Insert(i, MultiCopy(str[i].ToString(), countSymCopy - 1));

Console.WriteLine(str);

};

StartAction(actionDelegate, "Hello!", 4);

//Func

Console.BackgroundColor = ConsoleColor.DarkBlue;

Console.WriteLine("\nTest Func:\n");

Console.BackgroundColor = ConsoleColor.Black;

Func<string, int, string> funcDelegate;

funcDelegate = MultiCopy;

Console.WriteLine("{0}\n", MultiCopyOrSplit(funcDelegate, "qwerty ", 22));

funcDelegate = MultiSplitWord;

Console.WriteLine("{0}\n", MultiCopyOrSplit(funcDelegate, "qwerty", 1));

funcDelegate = (str, countSymCopy) => {

for (int i = str.Length - 1; i != -1; --i)

str = str.Insert(i, MultiCopy(str[i].ToString(), countSymCopy - 1));

return str;

};

Console.WriteLine("{0}\n", MultiCopyOrSplit(funcDelegate, "qwerty", 4));

Console.ReadKey();

}

static void StartAction(Action<string, int> method, string str, int copyCount)

{

method(str, copyCount);

}

static void ConsoleOutputMultiCopy(string str, int count)

{

Console.WriteLine(MultiCopy(str, count));

}

static void ConsoleOutputMultiSplitWord(string str, int count)

{

Console.WriteLine(MultiSplitWord(str, count));

}

static string MultiCopyOrSplit(FirstDelegate<string, string, int> method, string str, int count)

{

return method(str, count);

}

static string MultiCopyOrSplit(Func<string, int, string> method, string str, int count)

{

return method(str, count);

}

static string MultiSplitWord(string str, int countOfSpaces)

{

for (int i = str.Length - 1; i != 0; --i)

str = str.Insert(i, MultiCopy(" ", countOfSpaces));

return str;

}

static string MultiCopy(string strF, int count)

{

string strResult = "";

for (int i = 0; i != count; ++i)

strResult += strF;

return strResult;

}

static string MultiConcat(params string[] strArray)

{

string strResult = "";

foreach (string i in strArray)

strResult += i;

return strResult;

}

}

}

**Результаты выполнения программы**

