**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**КАФЕДРА САПР**

**ОТЧЕТ**

**по лабораторной работе №10**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Использование делегатов**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 9301 |  | Примакова Е.Е.. |
|  |  | Русанова К.В. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2021

# Цель работы

Ознакомиться и получить практические навыки по созданию делегатов и работе с ними в языке C#.

# Анализ задачи

Необходимо:

1. Написать программу, объявить, реализовать и использовать делегат.

2. Создать анонимный делегат, реализующий вывод сообщения на консоль.

3. Изменит тип делегата на тип Action<T>.

# Формальная постановка задачи

## Исходные данные

В качестве исходных данных во всех трех программах выступает заданное сообщение.

## Результирующие (выходные) данные

1 упражнение:

Выходные данные записываются в файл log.txt.

2 и 3 упражнения:

Выходные данные записываются в файл log.txt, а также выводятся на экран пользователя.

# Текст программы

Упражнение 1:

using System;

using System.IO;

namespace OopLabs.Delegates

{

class Program

{

private delegate void Log(string message);

static void DoSomething(Log log)

{

log(DateTime.Now + ": log message");//имитация операции, требующей журналирования

}

static void LogToFile(string message)

{

string myDocsPath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

string logFilePath = Path.Combine(myDocsPath, "log.txt"); //путь до файла

File.AppendAllText(logFilePath, message + Environment.NewLine); //запись в файл

}

static void Main()

{

DoSomething(LogToFile);

Console.ReadKey(); //ожидание нажатия клавиши

}

}

}

Упражнение 2:

using System;

using System.IO;

namespace OopLabs.Delegates

{

class Program

{

private delegate void Log(string message);

static void DoSomething(Log log)

{

log(DateTime.Now + ": log message"); //имитация операции, требующей журналирования

}

static void LogToFile(string message)

{

string myDocsPath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

string logFilePath = Path.Combine(myDocsPath, "log.txt"); //путь до файла

File.AppendAllText(logFilePath, message + Environment.NewLine); //запись в файл

}

static void Main()

{

DoSomething(LogToFile);

DoSomething(delegate (string message) { Console.WriteLine(message); }); //явное определение

DoSomething(message => Console.WriteLine(message)); //лямбда выражение

DoSomething(Console.WriteLine); //третий вывод в консоль

Console.ReadKey(); //ожидание нажатия клавиши

}

}

}

Упражнение 3:

using System;

using System.IO;

namespace OopLabs.Delegates

{

class Program

{

//private delegate void Log(string message);

static void DoSomething(Action<string> log) //замена на более общий тип

{

log(DateTime.Now + ": DoSomething()");

}

static void LogToFile(string message)

{

string myDocsPath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

string logFilePath = Path.Combine(myDocsPath, "log.txt"); //путь до файла

File.AppendAllText(logFilePath, message + Environment.NewLine); //запись в файл

}

static void Main()

{

DoSomething(LogToFile);

DoSomething(delegate (string message) { Console.WriteLine(message); }); //явное определение

DoSomething(message => Console.WriteLine(message)); //лямбда выражение

DoSomething(Console.WriteLine); //третий вывод в консоль

Console.ReadKey(); //ожидание нажатия клавиши

}

}

}

# Контрольный пример

Упражнение 1:

На рисунке 1 представлено содержимое файла с выходными данными программы из первого упражнения

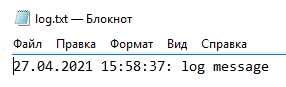


Рисунок 1. Вывод упражнения 1.

Упражнение 2:

На рисунке 2 представлено содержимое консоли с выходными данными программы из второго упражнения

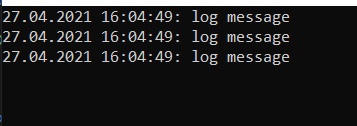


Рисунок 2. Вывод упражнения 2 в консоль.

Упражнение 3:

На рисунке 3 представлено содержимое консоли с выходными данными программы из третьего упражнения

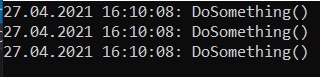


Рисунок 3. Вывод упражнения 3 в консоль.

# Полученные результаты

В ходе работы были получены в консоли и в файлах выводы программ с имитацией журналирования.

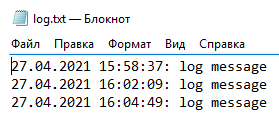


Рисунок 4. Содержимое файла вывода работы программы с имитацией журналирования

# Выводы

В ходе выполнения данной лабораторной работы были изучены способы объявлять, реализовывать и использовать именованные делегаты, анонимные делегаты и лямбда-выражения, делегаты с типом Action<T>.