МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем и цифровых технологий

**ОТЧЕТ**

по лабораторной работе №6

на тему: «Автоматизированное тестирование программного обеспечения»

по дисциплине: «Качество и тестирование программного обеспечения»

Вариант 10

Выполнил: Музалевский Н.С.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 92ПГ

Проверил: Олькина Е.В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

Орел, 2022 г.

**Код программы:**

using System.Reflection;

using System.Text;

class TestDriver

{

static void Main(string[] args)

{

string text = string.Empty;

if (args.Length > 1)

{

if (File.Exists(args[0]))

{

if (File.Exists(args[1]))

{

StreamReader sr = new StreamReader(args[1]);

int numberTest = 1;

while (!sr.EndOfStream)

{

string? line = sr.ReadLine();

if (!string.IsNullOrEmpty(line))

{

object[] str = line.Split(' ');

if (str.Length > 3)

{

Object[] arguments = new Object[str.Length - 3];

for (int i = 0; i < arguments.Length; i++)

{

arguments[i] = str[i + 2];

}

text += "Тест №" + numberTest + ": (" + Test(args[0], str[0].ToString(), str[1].ToString(), arguments, str[^1].ToString()) + ")\n";

numberTest++;

}

}

}

sr.Close();

}

}

else

{

text = "Файл " + args[0] + " не найден";

}

}

else

{

text = "Неверное количество аргументов на входе в программу";

}

using (FileStream fstream = new FileStream("Testing report.txt", FileMode.Create))

{

byte[] input = Encoding.Default.GetBytes(text);

fstream.Write(input, 0, input.Length);

Console.WriteLine("Отчёт тестирования сформирован");

}

}

public static string Test(string path, string className, string funcName, Object[] arguments, string expectedOutput)

{

Assembly modul = Assembly.LoadFile(path);

Type? classNameT = modul.GetTypes().FirstOrDefault(x => x.Name == className);

if (classNameT != null)

{

object? classNameO = Activator.CreateInstance(classNameT);

if (classNameO != null)

{

MethodInfo? method = classNameT.GetMethod(funcName);

if (method != null)

{

ParameterInfo[] param = method.GetParameters();

if (param.Length == arguments.Length)

{

for (int i = 0; i < param.Length; i++)

{

arguments[i] = Convert.ChangeType(arguments[i], param[i].ParameterType);

}

object? output = method.Invoke(classNameO, arguments);

if (output != null)

{

if (expectedOutput == output.ToString())

{

return ReportConstructor(expectedOutput, output.ToString(), "success");

}

else

{

return ReportConstructor(expectedOutput, output.ToString(), "Функция вывела другой результат faile");

}

}

else

{

return ReportConstructor(expectedOutput, "Функция ничего не вывела faile");

}

}

else

{

return ReportConstructor(expectedOutput, "Функция имеет другое количество параметров faile");

}

}

else

{

return ReportConstructor(expectedOutput, "Функция для тестирования не найдена faile");

}

}

else

{

return ReportConstructor(expectedOutput, "Не удалось создать объет класса для тестирования faile");

}

}

else

{

return ReportConstructor(expectedOutput, "Класс для тестирования не найден faile");

}

}

static string ReportConstructor(string expectedOutput, string output, string TotalResult)

{

if (string.IsNullOrEmpty(output))

{

output = "-";

}

return "Ожидаемый вывод: " + expectedOutput + " Вывод: " + output + " Итог: " + TotalResult;

}

static string ReportConstructor(string expectedOutput, string TotalResult)

{

return "Ожидаемый вывод: " + expectedOutput + " Вывод: - Итог: " + TotalResult;

}

}

Код файла для тестирования:

namespace Library1

{

public class Test

{

public int Test1(int x, int y)

{

return x + y;

}

public int Test2(int x, int y, int z)

{

return x + y + z;

}

public float Test3(float x, float y)

{

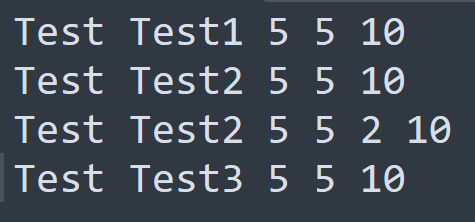
return x \* y;

}

}

}

Файл с тестами:



Отчёт проверки тестов:

