МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«Харківський політехнічний інститут»

Кафедра стратегічного управління

ЗВІТ

з лабораторної роботи №3

з дисципліни «Якість і тестування програмного забезпечення»

Перевірила

Виконав:

ст. гр. КН-27

Харків – 2019

**Тема:** Знакомство с методами и техниками модульного тестирования на реальных примерах программного кода, выполненного на языках C# и Java.

**Теоретическая информация:**

Модульное тестирование или как его еще называют UNIT-тестирование характеризуется следующими особенностями:

1. Модульное тестирование — это тестирование программы на уровне отдельно взятых модулей, функций или классов.
2. Цель модульного тестирования состоит в выявлении локализованных в модуле ошибок в реализации алгоритмов, а также в определении степени готовности системы к переходу на следующий уровень разработки и тестирования.
3. Модульное тестирование проводится по принципу "белого ящика“.
4. Модульное тестирование обычно подразумевает создание вокруг каждого модуля определенной среды.
5. Оно позволяет обнаруживать ошибки:
6. На уровне модульного тестирования проще всего обнаружить дефекты, связанные с алгоритмическими ошибками и ошибками кодирования алгоритмов.
7. Ошибки, связанные с неверной трактовкой данных, некорректной реализацией интерфейсов, совместимостью, производительностью и т.п. обычно выявляются на более поздних стадиях тестирования.

Для разных сред программирования применяются различные подходы к модульному тестированию:

1. Java – Junit;
2. C# - NUnit;
3. C++ - CppUnit;
4. PHP – PHPUnit
5. JavaScript – JQuery, Qunit.

Программный код на java (приложение):

**public class** Program {  
**public void** main(String[] args) {init(5); }  
**public double** init(**int**g) {  
**double** max = 0, y = 0;  
**try** {  
**for** (**int**i = 0; i<= g; i++) {  
 y = *func*(i);  
**if** (y > max) {  
 max = y;  
 } **else continue**;}}  
**catch** (NumberFormatException e1) {  
System.***out***.println(**"Error "** );  
 } **return** max; }  
**public static double** func(**int**x) {  
**return** Math.*pow*(5, x) / *factorial*(x);}  
**static int** factorial(**int** x) {  
**if** (x == 1 || x == 0) {  
**return** 1; }  
**return** x \* *factorial*(x - 1);}}

Программный код на C# (тестирование):

**import** junit.framework.TestCase;  
**import** org.junit.Test;  
**import static** org.junit.Assert.\*;  
  
**public class** ProgramTest {  
 @Test  
**public void** test1() {  
 Program program=**new** Program();  
**double** actual= program.init(5);  
**double** expected= 26.04167;  
*assertEquals*(expected,actual,0.0001); }  
  
 @Test  
**public void** test2() {  
 Program program = **new** Program();  
**double** expectedResult = program.init(5);  
*assertNotNull*(expectedResult); }  
  
 @Test  
**public void** test3() {  
 Program program = **new** Program();  
*assertTrue*(program.init(5) == 26.041666666666668);  
 }  
}

Результаты тестирования:

Результаты тестирования представлены на рисунке 1.

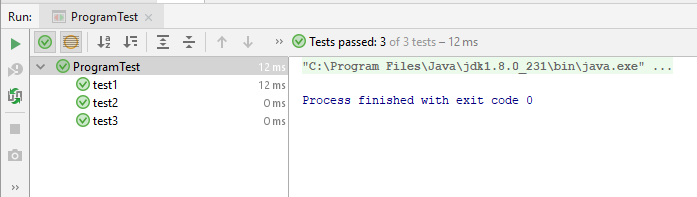
****

Рисунок 1 – Результаты тестирования (Java)

Программный код на Java (приложение):

usingSystem;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

namespaceProgram{

publicclass Program1{

public static void Main(string[] args){

Console.WriteLine("max = " + init(5));

Console.ReadLine();}

public static double init(int g) {

double max = 0, y = 0;

try {

for (inti = 0; i<= g; i++) {

y = func(i);

if (y > max) { max = y; } else continue;}}

catch (Exception e1) {Console.WriteLine("Error ");}

return max;}

public static double func(int x) {return Math.Pow(5, x) / factorial(x);}

static int factorial(int x) {if (x == 1 || x == 0) {return 1;}

returnx \* factorial(x - 1);}}}

Программный код на C# (тестирование):

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Program;using System;using System.Collections.Generic;using System.Linq;

using System.Text;using System.Threading.Tasks;

namespace Program.Tests{

[TestClass()]

public class Program1Tests {

[TestMethod]

public void Test1() {

Program.Program1 program = new Program.Program1();

double expected = Program.Program1.init(5);

double actual = 26.0416666666667;

Assert.AreEqual(expected, actual);

}

[TestMethod]

public void Test2() {

Program.Program1 program = new Program.Program1();

double expected = Program.Program1.init(5);

Assert.IsNotNull(expected, "Error");}}}

Результаты тестирования:

Результаты тестирования представлены на рисунке 2.

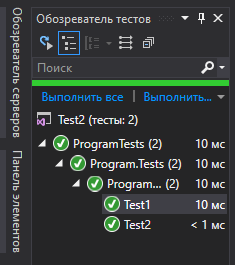
****

Рисунок 2 – Результаты тестирования (C#)

**Выводы:** в результате выполнения лабораторной работы были получены навыки работы с методами модульного тестирования на реальных примерах программного кода, разработанного в предыдущей лабораторной работе. По результатам тестирования в данной программе не было обнаружено ошибок, все тесты были выполнены удачно.