

ЛАБОРАТОРНА РОБОТА № 12

Інструменти тестування Junit та Mockito

Мета: набути навичок модульних та інтеграційних тестів використовуючи Junit та Mockito.

Хід роботи:

Завдання 1. Створити maven Java проект java_lab_12 з пакетом com.education.ztu.

Додайте необхідні залежності до проекту (в pom.xml): junit 4.12, mockito-core

1.9.5, assertj-core 3.11.1, JUnitParams 1.1.0 Посилання: <https://mvnrepository.com/>

Завдання 2. Додати до створеного проекту код, який треба покрити тестами. Можете використати код з лабораторних робіт 2 чи 3 або код якогось іншого вашого java проекту.

Завдання 3. Тестування за допомогою бібліотеки Junit. Створіть відповідні тестові класи та напишіть тест-кейси. Ваші тест-кейси посинні використовувати такі анотації: @BeforeClass, @AfterClass, @Before, @After, @Test, @Ignore Ваші тест-кейси посинні використовувати такі методи: fail, assertTrue, assertEquals, assertNull, assertNotNull, assertSame, assertNotSame, Перевірте позитивні, негативні випадки та випадки, які генерують виключні ситуації. Використайте Rules

(TemporaryFolder, ExpectedException, TestName, Timeout, ErrorCollector, Verifier)

Використайте групування тестів @Suite.SuiteClasses та параметризацію

@Parameters

Завдання 4. Тестування за допомогою фреймворка Mockito. Створіть відповідні тестові класи та проанотуйте їх анотацією @RunWith(MockitoJUnitRunner.class)

Напишіть тест-кейси. Ваші тест-кейси посинні використовувати такі методи та

анотації: • @Mock, @Spy, @Captor, @InjectMocks або аналогічні їм статичні ме-

тоди класу Mockito • when(mock).thenReturn(value), when(mock).thenThrow(),

doReturn(value).when(mock).methodName(params) • verify() в комбінації з atLeast,

atLeastOnce, atMost, times, never

| | | | | | | | |
|-----------|--------------------|----------|--------|------|---|------|---------|
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.00.000 – Пр12 | | |
| Розроб. | Iцук Ол.С. | | | | Звіт з лабораторної роботи | | |
| Перевір. | Піонтківський В.І. | | | | | | |
| Керівник | | | | | | | |
| Н. контр. | | | | | | | |
| Зав. каф. | | | | | ФІКТ Гр. ІПЗ-23-1 | | |
| | | | | | Lіт. | Арк. | Аркушів |
| | | | | | | 1 | 13 |

Лістинг програми:

OperationsTestMockito.java:

```
package com.education.ztu;

import static org.mockito.Mockito.*;
import static org.junit.Assert.*;

import Classes.OperationsForMock;
import org.junit.*;
import org.junit.runner.RunWith;
import org.mockito.*;
import org.mockito.junit.MockitoJUnitRunner;

@RunWith(MockitoJUnitRunner.class)
public class OperationsTestMockito {

    @Mock
    private OperationsForMock operations;

    @Captor
    private ArgumentCaptor<Double> captor;

    @Test
    public void testAddition() {
        when(operations.addition(2.0, 3.0)).thenReturn(5.0);
        double result = operations.addition(2.0, 3.0);
        assertEquals(5.0, result, 0.001);
        verify(operations, times(1)).addition(2.0, 3.0);
    }

    @Test
    public void testSubtraction() {
        when(operations.subtraction(5.0, 3.0)).thenReturn(2.0);
        double result = operations.subtraction(5.0, 3.0);
        assertEquals(2.0, result, 0.001);
        verify(operations, times(1)).subtraction(5.0, 3.0);
    }

    @Test
    public void testMultiplication() {
        when(operations.multiplication(2.0, 3.0)).thenReturn(6.0);
        double result = operations.multiplication(2.0, 3.0);
        assertEquals(6.0, result, 0.001);
        verify(operations, times(1)).multiplication(2.0, 3.0);
    }

    @Test
    public void testDivision() {
        when(operations.division(6.0, 2.0)).thenReturn(3.0);
        double result = operations.division(6.0, 2.0);
        assertEquals(3.0, result, 0.001);
        verify(operations, times(1)).division(6.0, 2.0);
    }

    @Test
    public void testCaptor() {
        when(operations.addition(anyDouble(), anyDouble())).thenReturn(5.0);
        operations.addition(2.0, 3.0);
        verify(operations).addition(captor.capture(), captor.capture());
    }
}
```

| | | | | |
|------|------|--------------------|--------|------|
| | | Iлук Ол.С. | | |
| | | Плюнгієвський В.І. | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

ДУ «Житомирська політехніка».25.121.00.000 – Пр12

Арк.

2

```

        assertEquals(Double.valueOf(2.0), captor.getAllValues().get(0));
        assertEquals(Double.valueOf(3.0), captor.getAllValues().get(1));
    }
}

```

OperationTest.java:

```

package com.education.ztu;

import static org.assertj.core.api.Assertions.*;
import static org.junit.Assert.*;

import Classes.Operations;
import org.junit.Assert;
import org.junit.Test;
import org.junit.runner.RunWith;
import junitparams.JUnitParamsRunner;
import junitparams.Parameters;

@RunWith(JUnitParamsRunner.class)
public class OperationTest {

    @Test
    public void testAddition() {
        Assert.assertEquals(10.0, Operations.addition(1, 2, 3, 4), 0.001);
    }

    @Test
    public void testSubtraction() {
        assertEquals(0.0, Operations.subtraction(10, 5, 5), 0.001);
    }

    @Test
    public void testMultiplication() {
        assertEquals(24.0, Operations.multiplication(2, 3, 4), 0.001);
    }

    @Test
    public void testDivision() {
        assertEquals(2.0, Operations.division(8, 2, 2), 0.001);
    }

    @Test(expected = ArithmeticException.class)
    public void testDivisionByZero() {
        Operations.division(1, 0);
    }

    @Test
    public void testAverage() {
        assertEquals(2.5, Operations.average(1, 2, 3, 4), 0.001);
    }

    @Test
    public void testMaximum() {
        assertEquals(4.0, Operations.maximum(1, 2, 3, 4), 0.001);
    }

    @Test(expected = IllegalArgumentException.class)
    public void testMaximumNoNumbers() {
        Operations.maximum();
    }
}

```

| | | | | |
|------|------|--------------------|--------|------|
| | | Ilyuk Ol.C. | | |
| | | Піоніківський В.І. | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

ДУ «Житомирська політехніка».25.121.00.000 – Пр12

Арк.

3

```

    @Test
    public void testMinimum() {
        assertEquals(1.0, Operations.minimum(1, 2, 3, 4), 0.001);
    }

    @Test(expected = IllegalArgumentException.class)
    public void testMinimumNoNumbers() {
        Operations.minimum();
    }

    @Test
    @Parameters({"1, 2, 3, 6", "2, 3, 4, 9"})
    public void testAdditionWithParams(double a, double b, double c, double
expected) {
        assertThat(Operations.addition(a, b, c)).isEqualTo(expected);
    }
}

```

OperationTestJunit.java:

```

package com.education.ztu;

import static org.junit.Assert.*;
import static org.assertj.core.api.Assertions.*;

import java.io.File;
import java.io.IOException;

import Classes.Operations;
import junitparams.JUnitParamsRunner;
import junitparams.Parameters;
import org.junit.*;
import org.junit.rules.*;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(JUnitParamsRunner.class)
@Suite.SuiteClasses({OperationTestJunit.class})
public class OperationTestJunit {

    @Rule
    public TemporaryFolder folder = new TemporaryFolder();

    @Rule
    public ExpectedException thrown = ExpectedException.none();

    @Rule
    public TestName testName = new TestName();

    @Rule
    public Timeout globalTimeout = Timeout.seconds(10);

    @Rule
    public ErrorCollector errorCollector = new ErrorCollector();

    @Rule
    public Verifier verifier = new Verifier() {
        @Override
        public void verify() {
        }
    };

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {

```

| | | | | | | | | |
|------|------|-------------------|--------|------|--|---|------|---|
| | | Ilyuk Ol.C. | | | | | | |
| | | Плюнгівський В.І. | | | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | ДУ «Житомирська політехніка».25.121.00.000 – Пр12 | Арк. | 4 |

| | | | | | | |
|------|------|---------------------------|--------|------|--|------|
| | | <i>Iцук Ол.С.</i> | | | | Арк. |
| | | <i>Піонтківський В.І.</i> | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.00.000 – Пр 12 | 5 |

```

    @Test
    public void testMinimumNoNumbers() {
        thrown.expect(IllegalArgumentException.class);
        thrown.expectMessage("No numbers provided.");
        Operations.minimum();
    }

    @Test
    @Parameters({ "1, 2, 3, 6", "2, 3, 4, 9" })
    public void testAdditionWithParams(double a, double b, double c, double expected) {
        assertThat(Operations.addition(a, b, c)).isEqualTo(expected);
    }

    @Test
    public void testFail() {
        org.junit.Assert.fail("Цей тест навмисно провалений");
    }

    @Test
    @Ignore("Тест ігнорується")
    public void testIgnored() {
    }

    @Test
    public void testRules() throws IOException {
        File file = folder.newFile("test.txt");
        assertTrue(file.exists());

        thrown.expect(IOException.class);
        throw new IOException("Test IOException");

    }

    @Test
    public void testAssertions() {
        assertNull(null);
        assertNotNull(new Object());

        Object obj = new Object();
        assertSame(obj, obj);
        assertNotSame(obj, new Object());
    }
}

```

Результат тесту програми:

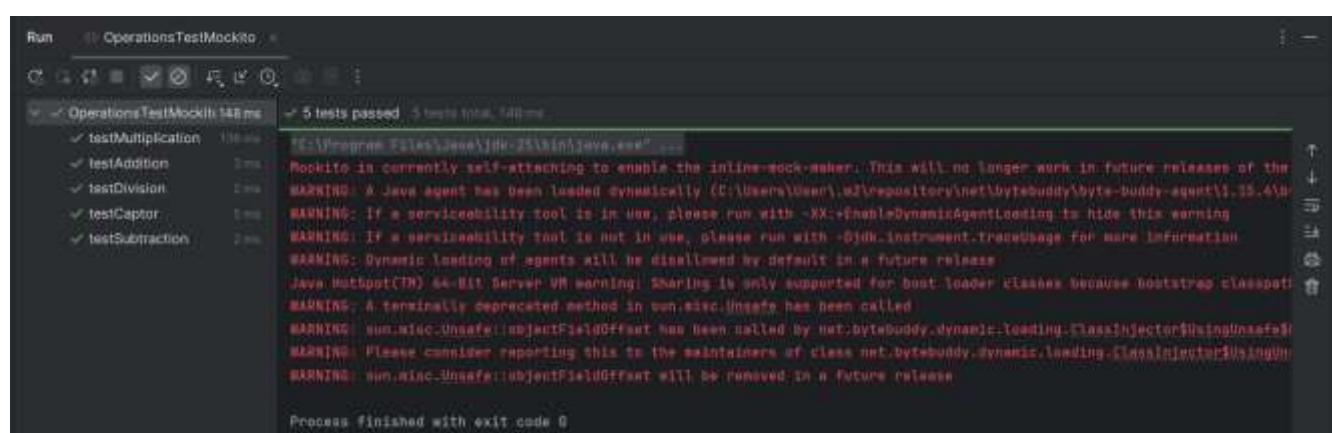


Рис.1 Результат первого тесту

| | | | | | | | | |
|------|------|---------------------|--------|------|--|--|--|---|
| | | Ilyuk Ol.C. | | | | | | |
| | | Піонірківський В.І. | | | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | ДУ «Житомирська політехніка».25.121.00.000 – Пр12 |

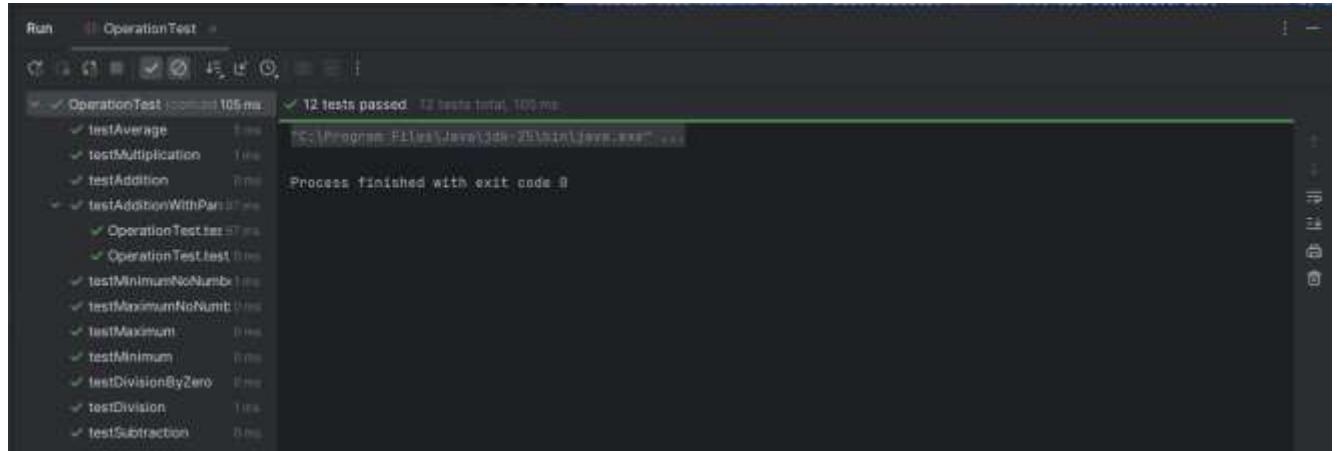


Рис.2 Результат другого тесту

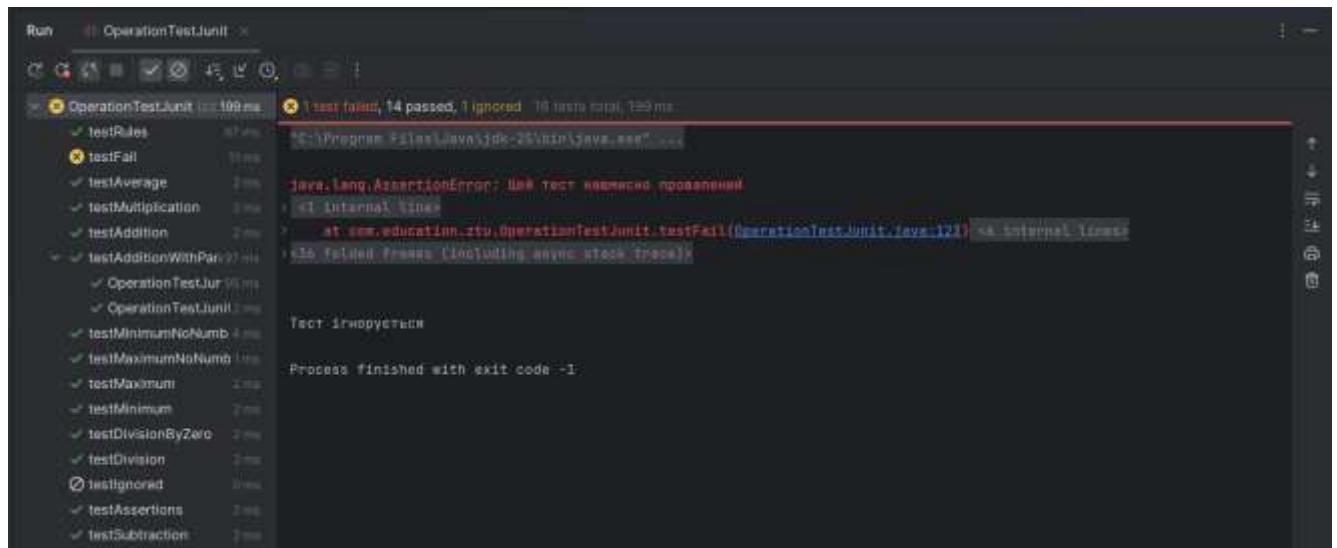


Рис.3 Результат третього тесту

Посилання на репозиторій: <https://github.com/Sasha1845/Java>

Висновок: Я набув навичок модульних та інтеграційних тестів використовуючи Junit та Mockito.

| | | | | | |
|------|------|---------------------|--------|------|---|
| | | Iлуку Ол.С. | | | ДУ «Житомирська політехніка».25.121.00.000 – Пр12 |
| | | Плюнгіківський В.І. | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | |