

Лабораторна робота №12

Інструменти тестування JUnit та Mockito

Мета роботи: 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).

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000 – Пр12		
Розроб.	Iлуцьк О.С.						
Перевір.	Плюнгівський В.І.						
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Лім.	Арк.	Аркушів
						1	9
					ФІКТ Гр. ІПЗ-23-1		

- Напишіть тест-кейси.
- Ваші тест-кейси посинні використовувати такі методи та анотації:
- @Mock, @Spy, @Captor, @InjectMocks або аналогічні їм статичні методи класу Mockito
- when(mock).thenReturn(value), when(mock).thenThrow(), doReturn(value), when(mock).methodName(params)
- verify() в комбінації з atLeast, atLeastOnce, atMost, times, never

Лістинг програми:

JUnit

AllTestsSuite.java

```
package com.education.ztu.junit;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    ProductJUnitTest.class,
    ProductRulesTest.class,
    ProductParameterizedTest.class
})
public class AllTestsSuite {
}
```

ProductJUnitTest.java

```
package com.education.ztu.junit;

import com.education.ztu.Product;
import org.junit.*;

import static org.junit.Assert.*;

public class ProductJUnitTest {

    private Product product;

    @BeforeClass
    public static void beforeAll() {
        System.out.println("===== @BeforeClass: START TESTING =====");
    }

    @AfterClass
    public static void afterAll() {
        System.out.println("===== @AfterClass: FINISH TESTING =====");
    }

    @Before
    public void setUp() {
        System.out.println("---- @Before executed");
        product = new Product("Apple", "iPhone 14", 2023, 999.99);
    }
}
```

		Iлуць О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр12	Арк.
		Піонірківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

@After
public void tearDown() {
    System.out.println("---- @After executed");
    product = null;
}

@Test
public void positiveTest_getters() {
    assertEquals("Apple", product.getBrand());
    assertEquals("iPhone 14", product.getModel());
    assertEquals(2023, product.getYear());
    assertEquals(999.99, product.getPrice(), 0.01);
}

@Test
public void positiveTest_setters() {
    product.setBrand("Samsung");
    product.setModel("S23");
    product.setYear(2024);
    product.setPrice(1200);

    assertEquals("Samsung", product.getBrand());
    assertEquals("S23", product.getModel());
    assertEquals(2024, product.getYear());
    assertEquals(1200, product.getPrice(), 0.01);
}

@Test
public void testNotNull() {
    assertNotNull(product);
}

@Test
public void testNull() {
    Product p = null;
    assertNull(p);
}

@Test
public void testSame() {
    Product p2 = product;
    assertEquals(product, p2);
}

@Test
public void testNotSame() {
    Product p2 = new Product("Apple", "iPhone 14", 2023, 999.99);
    assertNotSame(product, p2);
}

@Test
public void testEquals() {
    Product p2 = new Product("Apple", "iPhone 14", 2023, 999.99);
    assertTrue(product.equals(p2));
}

@Test
public void negativeTest_equals() {
    Product p2 = new Product("Samsung", "S23", 2023, 999.99);
    assertTrue(!product.equals(p2));
}

@Test

```

		Ilyuk O.C.					Арк.
		Піонійківський В.І.					
Змн.	Арк.	№ докум.	Підпис	Дата			3

```

public void failExample() {
    if (product.getPrice() < 0) {
        fail("Price cannot be negative!");
    }
}

@Ignore("Demonstration of @Ignore")
@Test
public void ignoredTest() {
    fail("This test is ignored");
}
}

```

ProductParameterizedTest.java

```

package com.education.ztu.junit;

import com.education.ztu.Product;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

import java.util.Arrays;
import java.util.Collection;

import static org.junit.Assert.*;

@RunWith(Parameterized.class)
public class ProductParameterizedTest {

    private double price;

    public ProductParameterizedTest(double price) {
        this.price = price;
    }

    @Parameterized.Parameters
    public static Collection<Object[]> data() {
        return Arrays.asList(new Object[][] {
            {100.0},
            {200.5},
            {999.99},
            {0.1}
        });
    }

    @Test
    public void testDifferentPrices() {
        Product product = new Product("Brand", "Model", 2024, price);
        assertEquals(price, product.getPrice(), 0.01);
    }
}

```

ProductRulesTest.java

```

package com.education.ztu.junit;

import com.education.ztu.Product;
import org.junit.*;
import org.junit.rules.*;

import java.io.File;

import static org.junit.Assert.*;

```

		Ilyuk O.C.			Арк.
		Піоніївський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000 – Пр12 4

```

public class ProductRulesTest {

    private Product product;

    @Rule
    public TemporaryFolder folder = new TemporaryFolder();

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

    @Rule
    public TestName testName = new TestName();

    @Rule
    public Timeout timeout = Timeout.seconds(2);

    @Rule
    public ErrorCollector collector = new ErrorCollector();

    @Rule
    public Verifier verifier = new Verifier() {
        @Override
        protected void verify() {
            System.out.println("Verifier check after test: " +
testName.getMethodName());
        }
    };

    @Before
    public void setUp() {
        product = new Product("Apple", "iPhone 14", 2023, 999.99);
    }

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

    @Test
    public void testExpectedException() {
        exception.expect(IllegalArgumentException.class);
        exception.expectMessage("Year cannot be negative");

        if (product.getYear() < 0) {
            throw new IllegalArgumentException("Year cannot be negative");
        }

        throw new IllegalArgumentException("Year cannot be negative");
    }

    @Test
    public void testErrorCollector() {
        try {
            assertTrue(product.getPrice() > 0);
            assertEquals("Apple", product.getBrand());
        } catch (Throwable t) {
            collector.addError(t);
        }
    }

    @Test
    public void testNameRule() {

```

		Iлуць О.С.					Арк.
		Піонійківський В.І.					
Змн.	Арк.	№ докум.	Підпис	Дата		ДУ «Житомирська політехніка».25.121.00.000 – Пр12	5

```

        System.out.println("Running test: " + testName.getMethodName());
        assertNotNull(product);
    }
}

```

Mockito

MockitoTestSuite.java

```

package com.education.ztu.mockito;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    ProductServiceMockitoTest.class
})
public class MockitoTestSuite {
}

```

ProductRepository.java

```

package com.education.ztu.mockito;

import com.education.ztu.Product;
import java.util.List;

public interface ProductRepository {
    Product findById(String id);
    List<Product> findAll();
    void save(Product product);
}

```

ProductService.java

```

package com.education.ztu.mockito;

import com.education.ztu.Product;
import java.util.List;

public class ProductService {

    private final ProductRepository repository;

    public ProductService(ProductRepository repository) {
        this.repository = repository;
    }

    public Product getProductById(String id) {
        if (id == null || id.isEmpty()) {
            throw new IllegalArgumentException("ID cannot be empty");
        }
        return repository.findById(id);
    }

    public List<Product> getAllProducts() {
        return repository.findAll();
    }

    public boolean updateProductPrice(String id, double newPrice) {
        Product product = repository.findById(id);

        if (product != null) {
            product.setPrice(newPrice);
            repository.save(product);
        }
    }
}

```

		Ilyuk O.C.			Арк.
		Піоніківський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000 – Пр12 6

```

        return true;
    }

    return false;
}

public void createNewProduct(Product product) {
    if (product == null) {
        throw new NullPointerException("Product cannot be null");
    }
    repository.save(product);
}

public boolean isProductAvailable(String id) {
    Product product = repository.findById(id);
    return product != null;
}
}

```

ProductServiceMockitoTest.java

```

package com.education.ztu.mockito;

import com.education.ztu.Product;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.*;
import org.mockito.junit.MockitoJUnitRunner;

import java.util.Arrays;
import java.util.List;

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

@RunWith(MockitoJUnitRunner.class)
public class ProductServiceMockitoTest {

    @Mock
    private ProductRepository mockRepository;

    @InjectMocks
    private ProductService productService;

    @Captor
    private ArgumentCaptor<Product> productCaptor;

    private final Product testProduct =
        new Product("Apple", "iPhone 14", 2023, 999.99);

    @Test
    public void testGetProductById_Found() {
        when(mockRepository.findById("1")).thenReturn(testProduct);

        Product result = productService.getProductId("1");

        assertNotNull(result);
        assertEquals("iPhone 14", result.getModel());
        verify(mockRepository, times(1)).findById("1");
    }

    @Test
    public void testGetProductById_NotFound() {
        when(mockRepository.findById("2")).thenReturn(null);
    }
}

```

		Iлуцк О.С.					Арк.
		Піонійківський В.І.					
Змн.	Арк.	№ докум.	Підпис	Дата		ДУ «Житомирська політехніка».25.121.00.000 – Пр12	7

```

        Product result = productService.getProductById("2");

        assertNull(result);
        verify(mockRepository, times(1)).findById("2");
    }

    @Test(expected = IllegalArgumentException.class)
    public void testGetProductById_InvalidId() {
        productService.getProductById("");
    }

    @Test
    public void test GetAllProducts() {
        List<Product> products = Arrays.asList(
            testProduct,
            new Product("Samsung", "S23", 2023, 899.99)
        );

        when(mockRepository.findAll()).thenReturn(products);

        List<Product> result = productService.getAllProducts();

        assertEquals(2, result.size());
        verify(mockRepository, atLeastOnce()).findAll();
    }

    @Test
    public void testUpdateProductPrice_Success() {
        when(mockRepository.findById("1")).thenReturn(testProduct);
        doNothing().when(mockRepository).save(any(Product.class));

        boolean result = productService.updateProductPrice("1", 1100);

        assertTrue(result);
        assertEquals(1100, testProduct.getPrice(), 0.01);

        verify(mockRepository).findById("1");
        verify(mockRepository).save(testProduct);
    }

    @Test
    public void testUpdateProductPrice_NotFound() {
        when(mockRepository.findById("5")).thenReturn(null);

        boolean result = productService.updateProductPrice("5", 1200);

        assertFalse(result);

        verify(mockRepository, times(1)).findById("5");
        verify(mockRepository, never()).save(any(Product.class));
    }

    @Test(expected = RuntimeException.class)
    public void testCreateNewProduct_Exception() {
        doThrow(new RuntimeException("Database error"))
            .when(mockRepository).save(testProduct);

        productService.createNewProduct(testProduct);
    }

    @Test
    public void testCreateNewProduct_Captor() {

```

		Ilyuk O.C.					Арк.
		Плоніцківський В.І.					
Змн.	Арк.	№ докум.	Підпис	Дата		ДУ «Житомирська політехніка».25.121.00.000 – Пр12	8

```

        doNothing().when(mockRepository).save(any(Product.class));

        productService.createNewProduct(testProduct);

        verify(mockRepository).save(productCaptor.capture());

        Product savedProduct = productCaptor.getValue();

        assertEquals("Apple", savedProduct.getBrand());
        assertEquals("iPhone 14", savedProduct.getModel());
    }

    @Test
    public void testAvailability_VerifyModes() {
        when(mockRepository.findById("A")).thenReturn(testProduct);
        when(mockRepository.findById("B")).thenReturn(null);

        assertTrue(productService.isProductAvailable("A"));
        assertFalse(productService.isProductAvailable("B"));
        assertTrue(productService.isProductAvailable("A"));

        verify(mockRepository, atLeast(2)).findById("A");
        verify(mockRepository, times(1)).findById("B");
        verify(mockRepository, never()).findById("C");
        verify(mockRepository, atMost(3)).findById(anyString());
    }
}

```

Виконання програми:

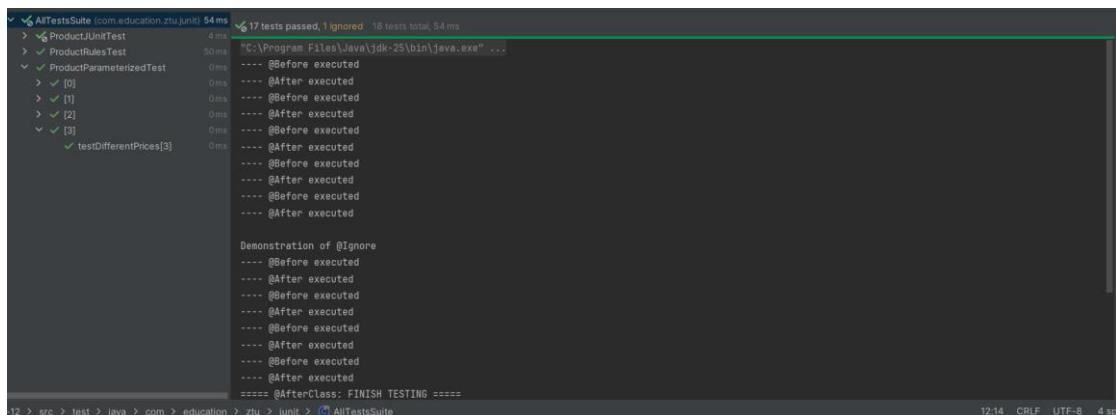


Рис. 1 Результат виконання тестів за допомогою Junit

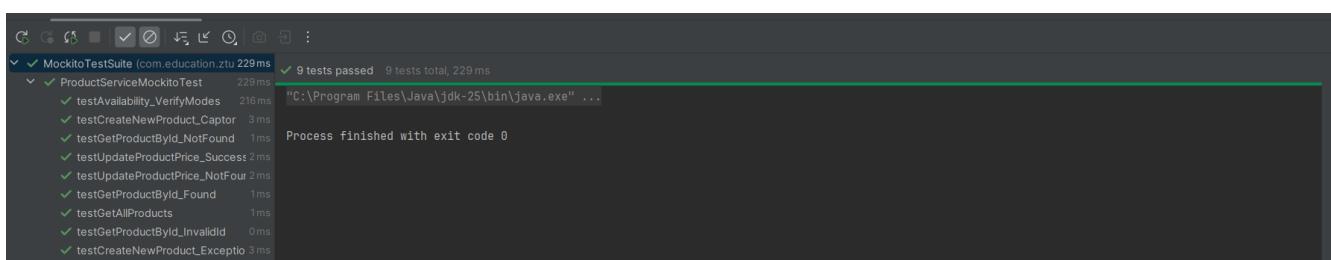


Рис. 2 Результат виконання тестів за допомогою Mockito

Посилання на репозиторій: <https://github.com/Oleg-Ischuk/Java-IPZ-23-1>

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

		Iуцук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр12	Арк.
		Плюнгіківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		9