**Exercise 7: Handling Void Methods with Exceptions**

1. ExternalApi.java

package com.exercise7;

public interface ExternalApi {

void sendNotification(String message);

}

1. MyService.java

package com.exercise7;

public class MyService {

private ExternalApi notifier;

public MyService(ExternalApi notifier) {

this.notifier = notifier;

}

public void triggerAlert(String msg) {

notifier.sendNotification(msg);

}

}

1. MyServiceTest.java

package com.exercise7;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

public class MyServiceTest {

*@Test*

public void testVoidMethodThrowsException() {

ExternalApi mockNotifier = *mock*(ExternalApi.class);

*doThrow*(new RuntimeException("Notification failed"))

.when(mockNotifier).sendNotification("ALERT");

MyService service = new MyService(mockNotifier);

Exception exception = *assertThrows*(RuntimeException.class, () -> {

service.triggerAlert("ALERT");

});

*assertEquals*("Notification failed", exception.getMessage());

*verify*(mockNotifier).sendNotification("ALERT");

}

}

Output:

A computer screen shot of a program

AI-generated content may be incorrect.