**Exercise 1: Mocking and Stubbing**

**#XML**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.example</groupId>

<artifactId>DifferentTestDoubles</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-all</artifactId>

<version>2.0.2-beta</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**# MessagingEngine.java**

import java.util.List;

public class MessagingEngine {

public void sendEmail(String subject, List<String> emails) {

//set email subject

//set to: with the emails list

//send email

}

}

# NewsletterSender

import java.util.List;

public class NewsletterSender {

private SubscribersDatabase subscribersDatabase;

private MessagingEngine messagingEngine;

public NewsletterSender(SubscribersDatabase subscribersDatabase, MessagingEngine messagingEngine) {

this.subscribersDatabase = subscribersDatabase;

this.messagingEngine = messagingEngine;

}

public void sendNewsletter(String subject){

List<String> emails = subscribersDatabase.getSubscribers();

if(numberOfSubscribers() == 0){

throw new ZeroSubscribersException();

}

messagingEngine.sendEmail(subject, emails);

}

public int numberOfSubscribers(){

return subscribersDatabase.getSubscribers().size();

}

public SubscribersDatabase getSubscribersDatabase() {

return subscribersDatabase;

}

public MessagingEngine getMessagingEngine() {

return messagingEngine;

}

}

# SubscribersDatabase

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class SubscribersDatabase {

public List<String> getSubscribers() {

List<String> subscribers = new ArrayList<String>();

return subscribers;

}

}

# ZeroSubscribersException

public class ZeroSubscribersException extends RuntimeException{

}

# NewsletterSenderTest

import org.junit.Test;

import java.util.Arrays;

import java.util.List;

import static org.junit.Assert.assertEquals;

import static org.mockito.Mockito.\*;

public class NewsletterSenderTest {

@Test

public void constructorAssignsDatabase(){

MessagingEngine messagingEngine = mock(MessagingEngine.class);

SubscribersDatabase subscribersDatabase = new SubscribersDatabase();

NewsletterSender newsletterSender = new NewsletterSender(subscribersDatabase, messagingEngine);

assertEquals(subscribersDatabase, newsletterSender.getSubscribersDatabase());

}

@Test

public void numberOfSubscribers(){

SubscribersDatabase subscribersDatabaseMock = mock(SubscribersDatabase.class);

MessagingEngine messagingEngineMock = mock(MessagingEngine.class);

NewsletterSender sender = new NewsletterSender(subscribersDatabaseMock, messagingEngineMock);

List<String> subscribersList = Arrays.asList("email1", "email2", "email3");

when(subscribersDatabaseMock.getSubscribers()).thenReturn(subscribersList);

assertEquals(3, sender.numberOfSubscribers());

}

@Test(expected = ZeroSubscribersException.class)

public void zeroSubscribersThrown(){

NewsletterSender newsletterSender = new NewsletterSender(new SubscribersDatabase(), new MessagingEngine());

NewsletterSender newsletterSenderSpy = spy(newsletterSender);

when(newsletterSenderSpy.numberOfSubscribers()).thenReturn(0);

newsletterSenderSpy.sendNewsletter("SUBJECT");

}

}

OutPut :

