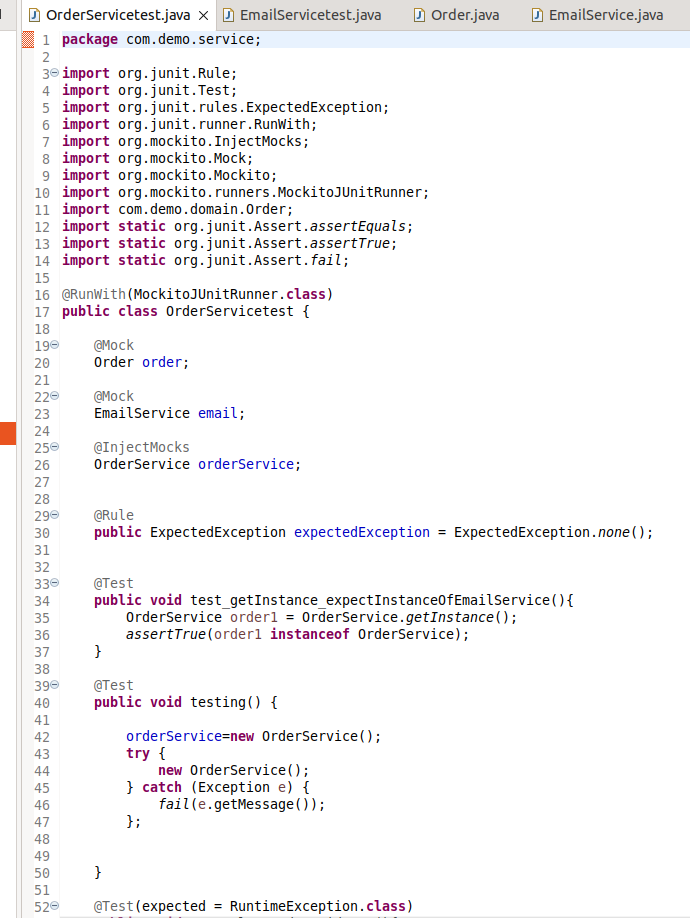
Java: Unit testing HONEY ARORA

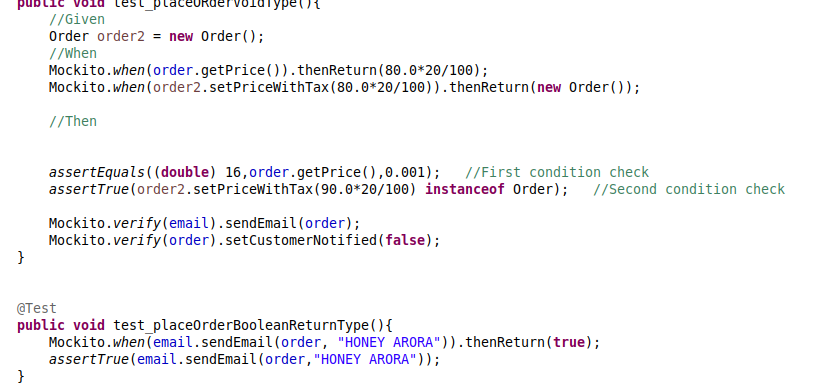
Trainee

**QUES: Write the unit tests for following:**

* **OrderService.java**
* **EmailService.java**

**OrderServicetest:**

****

****

**(below pasted code)**

package com.demo.service;

import org.junit.Rule;

import org.junit.Test;

import org.junit.rules.ExpectedException;

import org.junit.runner.RunWith;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.Mockito;

import org.mockito.runners.MockitoJUnitRunner;

import com.demo.domain.Order;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertTrue;

import static org.junit.Assert.fail;

@RunWith(MockitoJUnitRunner.class)

public class OrderServicetest {

@Mock

Order order;

@Mock

EmailService email;

@InjectMocks

OrderService orderService;

@Rule

public ExpectedException expectedException = ExpectedException.none();

@Test

public void test\_getInstance\_expectInstanceOfEmailService(){

OrderService order1 = OrderService.getInstance();

assertTrue(order1 instanceof OrderService);

}

@Test

public void testing() {

orderService=new OrderService();

try {

new OrderService();

} catch (Exception e) {

fail(e.getMessage());

};

**}**

@Test(expected = RuntimeException.class)

public void test\_placeORderVoidType(){

//Given

Order order2 = new Order();

//When

Mockito.when(order.getPrice()).thenReturn(80.0\*20/100);

Mockito.when(order2.setPriceWithTax(80.0\*20/100)).thenReturn(new Order());

//Then

assertEquals((double) 16,order.getPrice(),0.001); //First condition check

assertTrue(order2.setPriceWithTax(90.0\*20/100) instanceof Order); //Second condition check

Mockito.verify(email).sendEmail(order);

Mockito.verify(order).setCustomerNotified(false);

}

@Test

public void test\_placeOrderBooleanReturnType(){

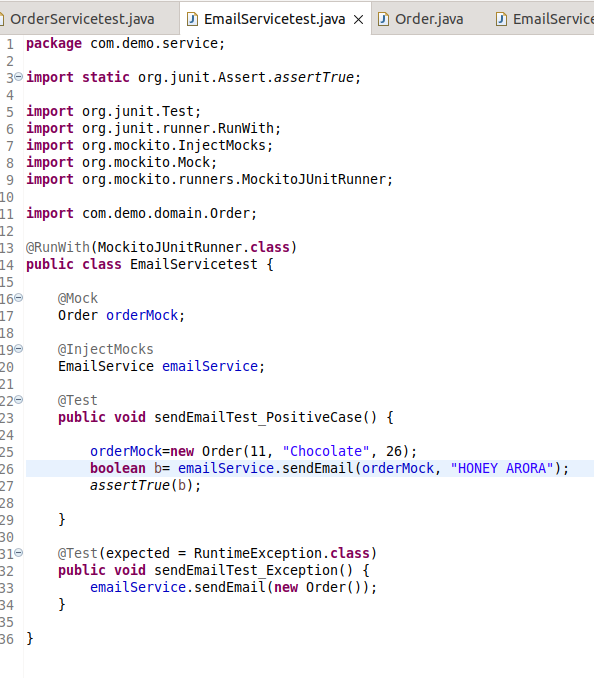
Mockito.when(email.sendEmail(order, "HONEY ARORA")).thenReturn(true);

assertTrue(email.sendEmail(order,"HONEY ARORA"));

}

}

**EmailServiceTest**

****