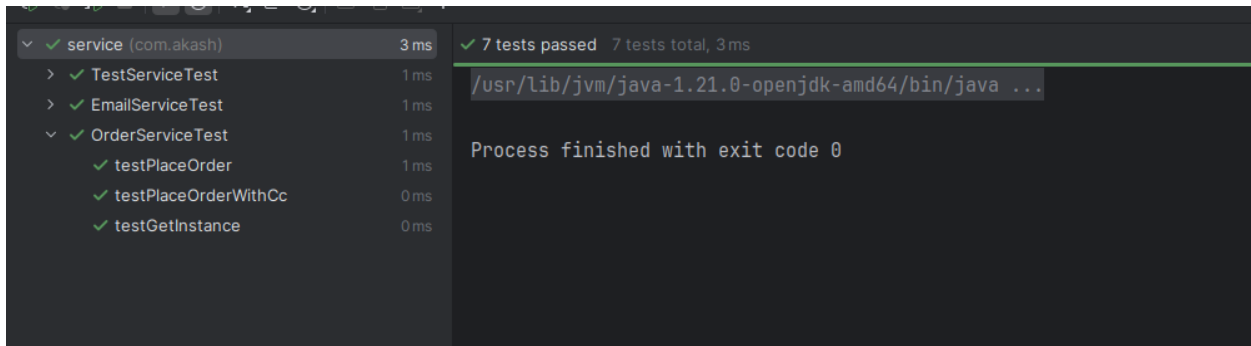


Assignment - JUNIT - Basics

Q1) Write the unit tests for following: OrderService.java EmailService.java
Find the code base here. <https://github.com/nitin-singh-0208/unit-testing>

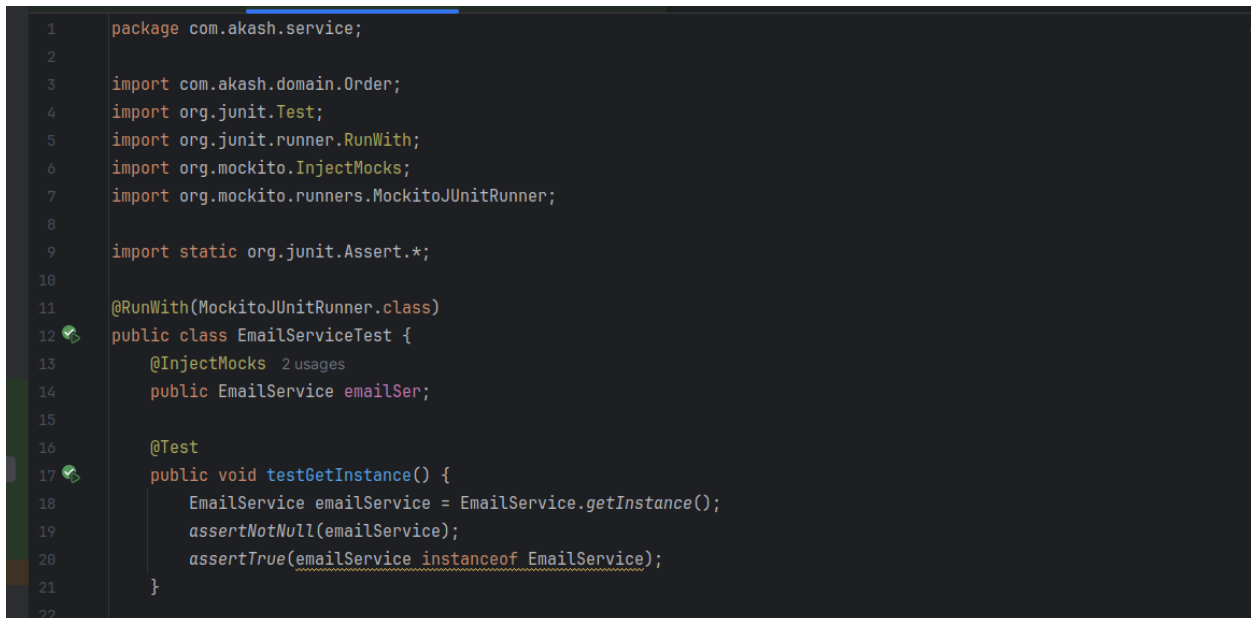


```
✓ service (com.akash) 3 ms
  > ✓ TestServiceTest 1 ms
  > ✓ EmailServiceTest 1 ms
  ✓ OrderServiceTest 1 ms
    ✓ testPlaceOrder 1 ms
    ✓ testPlaceOrderWithCc 0 ms
    ✓ testGetInstance 0 ms

✓ 7 tests passed 7 tests total, 3 ms

/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java ...

Process finished with exit code 0
```



```
1 package com.akash.service;
2
3 import com.akash.domain.Order;
4 import org.junit.Test;
5 import org.junit.runner.RunWith;
6 import org.mockito.InjectMocks;
7 import org.mockito.runners.MockitoJUnitRunner;
8
9 import static org.junit.Assert.*;
10
11 @RunWith(MockitoJUnitRunner.class)
12 public class EmailServiceTest {
13     @InjectMocks 2 usages
14     public EmailService emailSer;
15
16     @Test
17     public void testGetInstance() {
18         EmailService emailService = EmailService.getInstance();
19         assertNotNull(emailService);
20         assertTrue(emailService instanceof EmailService);
21     }
22 }
```

```

@Test
public void testSendEmail() {
    Order order = new Order( quantity: 1, itemName: "Test Item", price: 100.0);
    order.setCustomerNotified(true);
    try {
        emailSer.sendEmail(order);
        assertFalse(order.isCustomerNotified());
    } catch (RuntimeException e) {
        assertTrue(e instanceof RuntimeException);
    }
}

@Test
public void testSendEmailWithAttachment() {
    Order order = new Order( quantity: 1, itemName: "Test Item", price: 100.0);
    order.setCustomerNotified(false);
    boolean result = emailSer.sendEmail(order, cc: "123");
    assertTrue(order.isCustomerNotified());
    assertTrue(result);
}
}

```

```

1 package com.akash.service;
2
3
4 import com.akash.domain.Order;
5 import org.junit.Test;
6 import org.junit.runner.RunWith;
7 import org.mockito.runners.MockitoJUnitRunner;
8
9 import static org.junit.Assert.*;
10
11 @RunWith(MockitoJUnitRunner.class)
12 public class OrderServiceTest {
13
14     @Test
15     public void testGetInstance() {
16         OrderService orderService = OrderService.getInstance();
17         assertNotNull(orderService);
18         assertTrue(orderService instanceof OrderService);
19     }
20 }

```

```

2 public class OrderServiceTest {
3
4     @Test
5     public void testPlaceOrderWithCc() {
6         OrderService orderService = OrderService.getInstance();
7         Order order = new Order( quantity: 1, itemName: "Test Item", price: 100.0);
8         order.setCustomerNotified(false);
9         boolean testCc = orderService.placeOrder(order, cc: "Test CC");
10        assertTrue(testCc);
11        assertTrue(order.isCustomerNotified());
12        assertEquals( expected: 120.0, order.getPriceWithTax(), delta: 0.01);
13    }
14
15    @Test
16    public void testPlaceOrder() {
17        OrderService orderService = OrderService.getInstance();
18        Order order = new Order( quantity: 1, itemName: "Test Item", price: 100.0);
19        order.setCustomerNotified(true);
20        try {
21            orderService.placeOrder(order);
22            assertFalse(order.isCustomerNotified());
23            assertEquals( expected: 120.0, order.getPriceWithTax(), delta: 0.01);
24        } catch (Exception e) {
25            assertTrue(e instanceof RuntimeException);
26        }
27    }
28 }

```

```

1 package com.akash.service;
2
3 import org.junit.Test;
4 import org.junit.runner.RunWith;
5 import org.mockito.InjectMocks;
6 import org.mockito.runners.MockitoJUnitRunner;
7
8 import static org.junit.Assert.assertEquals;
9
10 @RunWith(MockitoJUnitRunner.class)
11 public class TestServiceTest {
12     @InjectMocks 1 usage
13     TestService testService;
14
15     @Test
16     public void testFirstMethod() {
17         assertEquals( expected: 1, testService.test());
18     }
19 }

```