Writing BDD Style Tests

Understanding BDDMockito

BDDMockito in Action

Basic BDDMockito Usage

```java
@ExtendWith(MockitoExtension.class)
public class OrderServiceTest {

    @Mock
    PaymentProcessor paymentProcessor;

    private OrderService orderService;

    @BeforeEach
    void setUp() {
        orderService = new OrderService(paymentProcessor);
    }

    @Test
    void shouldProcessPaymentSuccessfully() {
        // Given
        double amount = 100.0;
        BDDMockito.given(paymentProcessor.processPayment(amount)).willReturn(true);

        // When
        boolean result = orderService.processOrder(amount);

        // Then
        assertTrue(result, "Order should be processed successfully");

        // Verify the interaction with the mock
        verify(paymentProcessor).should().processPayment(amount);
    }
}
```

Testing Exception Handling

```
import static org.mockito.BDDMockito.*;

// Inside your test method
PaymentService paymentService = mock(PaymentService.class);
given(paymentService.processPayment(anyString())).willThrow(new PaymentException("Invalid payment"));

// When & Then
assertThrows(PaymentException.class, () -> {
    paymentService.processPayment("invalid_payment_id");
});
```

Using Argument Matchers in BDDMockito

```
import static org.mockito.BDDMockito.*;
import static org.mockito.ArgumentMatchers.*;

// Inside your test method
Logger logger = mock(Logger.class);
given(logger.logMessage(anyString())).willReturn(true);

// When & Then
assertTrue(logger.logMessage("Info: User logged in"));
```

Dynamic Responses with willAnswer()

```
import static org.mockito.BDDMockito.*;
import org.mockito.invocation.InvocationOnMock;
import org.mockito.stubbing.Answer;

// Inside your test method
MessageFormatter formatter = mock(MessageFormatter.class);
given(formatter.format(anyString(), anyString())).willAnswer(new Answer<String>() {
    public String answer(InvocationOnMock invocation) {
        Object[] args = invocation.getArguments();
        return "Formatted: " + args[0] + ", " + args[1];
    }
});

// When & Then
assertEquals("Formatted: Hello, World", formatter.format("Hello", "World"));
```

Verifying Interactions in BDDMockito

```
import static org.mockito.BDDMockito.*;
import static org.mockito.Mockito.times;

// Inside your test method
OrderService orderService = mock(OrderService.class);
orderService.placeOrder("123", 5);

// Verification
then(orderService).should(times(1)).placeOrder("123", 5);
then(orderService).shouldHaveNoMoreInteractions();
```