Exercise 1: Mocking and Stubbing

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.mockito.Mockito.when;

import org.junit.jupiter.api.Test;

import org.mockito.Mock;

import org.mockito.MockitoAnnotations;

class DataService {

private final ExternalApi apiClient;

public DataService(ExternalApi apiClient) {

this.apiClient = apiClient;

}

public String fetchData() {

return apiClient.getData();

}

public boolean validateData() {

String data = apiClient.getData();

return data != null && !data.isEmpty();

}

}

interface ExternalApi {

String getData();

void refreshConnection();

}

class DataServiceTest {

@Test

void testFetchDataWithMock() {

ExternalApi mockApi = org.mockito.Mockito.mock(ExternalApi.class);

when(mockApi.getData()).thenReturn("Sample test data");

DataService service = new DataService(mockApi);

String result = service.fetchData();

assertEquals("Sample test data", result);

}

@Test

void testValidateDataWithDifferentResponses() {

ExternalApi mockApi = org.mockito.Mockito.mock(ExternalApi.class);

DataService service = new DataService(mockApi);

when(mockApi.getData()).thenReturn("Valid response");

assertTrue(service.validateData());

when(mockApi.getData()).thenReturn("");

assertFalse(service.validateData());

when(mockApi.getData()).thenReturn(null);

assertFalse(service.validateData());

}

@Test

void testApiMethodCalledExactlyOnce() {

ExternalApi mockApi = org.mockito.Mockito.mock(ExternalApi.class);

when(mockApi.getData()).thenReturn("Test data");

DataService service = new DataService(mockApi);

service.fetchData();

org.mockito.Mockito.verify(mockApi, org.mockito.Mockito.times(1)).getData();

org.mockito.Mockito.verifyNoMoreInteractions(mockApi);

}

private void assertTrue(boolean condition) {

assertEquals(true, condition);

}

private void assertFalse(boolean condition) {

assertEquals(false, condition);

}

}