Exercise 1: Mocking Databases and Repositories

@class

package org.example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

public class ServiceTest {

@Test

public void testServiceWithMockRepository() {

Repository mockRepository = mock(Repository.class);

when(mockRepository.getData()).thenReturn("Mock Data");

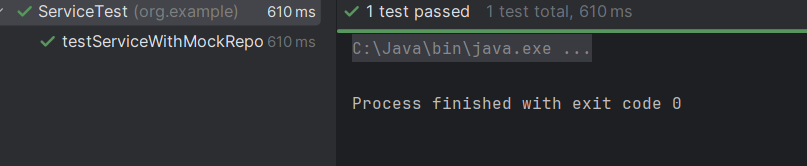
Service service = new Service(mockRepository);

String result = service.processData();

assertEquals("Processed Mock Data", result);

}

}



Exercise 2: Mocking External Services (RESTful APIs)

@class

package org.example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

public class ApiServiceTest {

@Test

public void testServiceWithMockRestClient() {

RestClient mockClient = mock(RestClient.class);

when(mockClient.getResponse()).thenReturn("Mock Response");

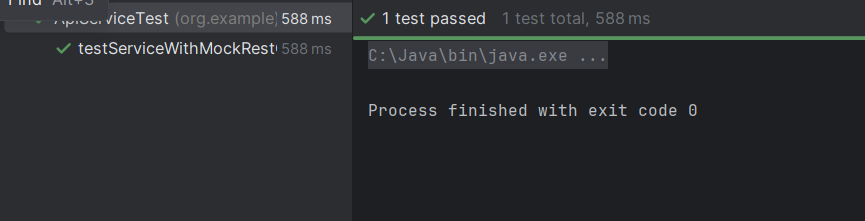
ApiService service = new ApiService(mockClient);

String result = service.fetchData();

assertEquals("Fetched Mock Response", result);

}

}



Exercise 3: Mocking File I/O

@class

package org.example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

public class FileServiceTest {

@Test

public void testServiceWithMockFileIO() {

FileReader mockReader = mock(FileReader.class);

FileWriter mockWriter = mock(FileWriter.class);

when(mockReader.read()).thenReturn("Mock File Content");

FileService service = new FileService(mockReader, mockWriter);

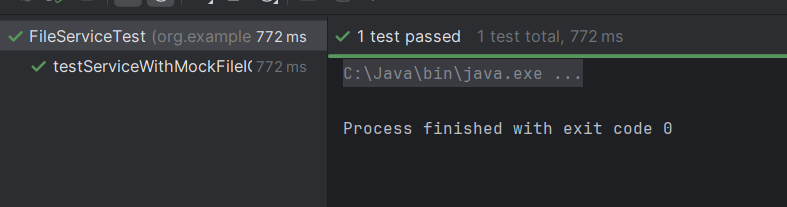
String result = service.processFile();

assertEquals("Processed Mock File Content", result);

verify(mockWriter).write("Processed Mock File Content");

}

}



Exercise 4: Mocking Network Interactions

@class

package org.example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

public class NetworkServiceTest {

@Test

public void testServiceWithMockNetworkClient() {

NetworkClient mockClient = mock(NetworkClient.class);

when(mockClient.connect()).thenReturn("Mock Connection");

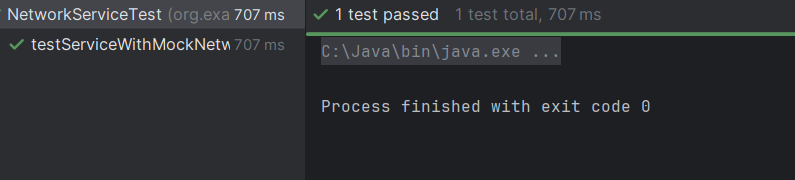
NetworkService service = new NetworkService(mockClient);

String result = service.connectToServer();

assertEquals("Connected to Mock Connection", result);

}

}



Exercise 5: Mocking Multiple Return Values

@class

package org.example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

public class MultiReturnServiceTest {

@Test

public void testServiceWithMultipleReturnValues() {

Repository mockRepository = mock(Repository.class);

when(mockRepository.getData())

.thenReturn("First Mock Data")

.thenReturn("Second Mock Data");

Service service = new Service(mockRepository);

String firstResult = service.processData();

String secondResult = service.processData();

assertEquals("Processed First Mock Data", firstResult);

assertEquals("Processed Second Mock Data", secondResult);

}

}

