**Exercise 5: Mocking and Stubbing with Multiple Returns**

**Scenario:** You need to test a service that depends on an external API with multiple return values.

**WeatherService.java :-**

package org.example1;  
  
public class WeatherService {  
  
 private final WeatherApiClient weatherApiClient;  
  
 public WeatherService(WeatherApiClient weatherApiClient) {  
 this.weatherApiClient = weatherApiClient;  
 }  
  
 public String getWeatherReport() {  
 // Calls the external API twice and combines responses  
 String morningWeather = weatherApiClient.getCurrentWeather();  
 String eveningWeather = weatherApiClient.getCurrentWeather();  
 return "Morning: " + morningWeather + ", Evening: " + eveningWeather;  
 }  
}

**WeatherApiClient.java :-**

package org.example1;  
  
public interface WeatherApiClient {  
 String getCurrentWeather();  
}

**WeatherServiceTest :-**

package org.example1;  
  
import org.junit.jupiter.api.Test; // JUnit 5  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
import static org.mockito.Mockito.\*;  
  
public class WeatherServiceTest {  
  
 @Test  
 public void testGetWeatherReport\_WithConsecutiveResponses() {  
 WeatherApiClient mockApiClient = *mock*(WeatherApiClient.class);  
  
 *when*(mockApiClient.getCurrentWeather())  
 .thenReturn("Sunny")  
 .thenReturn("Rainy");  
  
 WeatherService service = new WeatherService(mockApiClient);  
 String report = service.getWeatherReport();  
  
 *assertEquals*("Morning: Sunny, Evening: Rainy", report);  
 *verify*(mockApiClient, *times*(2)).getCurrentWeather();  
 }  
}

**pom.xml :-**

<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example1</groupId>  
 <artifactId>multi-response-mock</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <dependencies>  
 <!-- JUnit 5 -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- Mockito -->  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>5.11.0</version>  
 <scope>test</scope>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- Test runner plugin -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.1.2</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

Output :-

