**Exercise 1: Mocking and Stubbing**

**WeatherApi.java**

public interface WeatherApi {  
 String getCurrentTemperature(String city);  
}

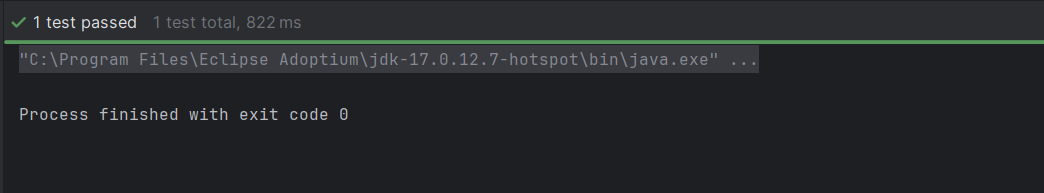
**WeatherService.java**

public class WeatherService {  
  
 private WeatherApi api;  
  
 public WeatherService(WeatherApi api) {  
 this.api = api;  
 }  
  
 public String getCityWeather(String city) {  
 return "Temperature in " + city + ": " + api.getCurrentTemperature(city);  
 }  
}

**WeatherServiceTest.java**

import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class WeatherServiceTest {  
  
 @Test  
 public void testGetCityWeather() {  
  
 WeatherApi mockApi = *mock*(WeatherApi.class);  
  
 *when*(mockApi.getCurrentTemperature("Chennai")).thenReturn("32°C");  
  
 WeatherService service = new WeatherService(mockApi);  
 String result = service.getCityWeather("Chennai");  
  
   
 *assertEquals*("Temperature in Chennai: 32°C", result);  
 }  
}

**Output :**

****

**Exercise 2 : Verifying Interactions**

**Printer.java**

public interface Printer {

void print(String message);

}

**DocumentService.java**

public class DocumentService {

private Printer printer;

public DocumentService(Printer printer) {

this.printer = printer;

}

public void printDocument(String doc) {

printer.print("Printing: " + doc);

}

}

**DocumentServiceTest.java**

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

public class DocumentServiceTest {

@Test

public void testPrinterInteraction() {

Printer mockPrinter = mock(Printer.class);

DocumentService service = new DocumentService(mockPrinter);

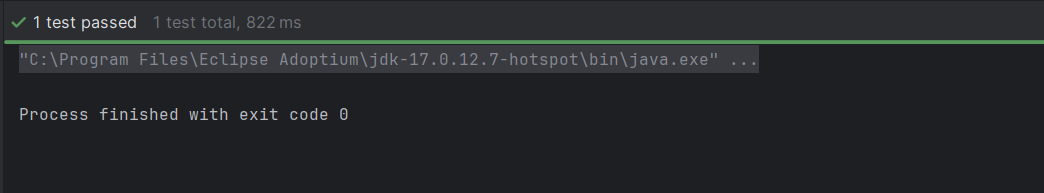
service.printDocument("Invoice.pdf");

verify(mockPrinter).print("Printing: Invoice.pdf");

}

}

**Output :**

****