Name: SASHMITHA R

Reg No: 727722EUCD042

Exercise 1: Mocking and Stubbing

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 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>com.example</groupId>
 <artifactId>Mock</artifactId>
 <version>0.0.1-SNAPSHOT</version>
 <name>Mock</name>
 <dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <version>5.12.0</version>
    <scope>test</scope>
  </dependency>
 </dependencies>
 <build>
  <plugins>
    <plugin>
```

```
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.1.2</version>
        <configuration>
          <includes>
            <include>**/*Test.java</include>
          </includes>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

MyService.java

```
package com.example;
public class MyService
 {
    private ExternalApi api;
    public MyService(ExternalApi api)
    {
        this.api = api;
    }
    public String fetchData()
    {
        return api.getData();
    }
}
```

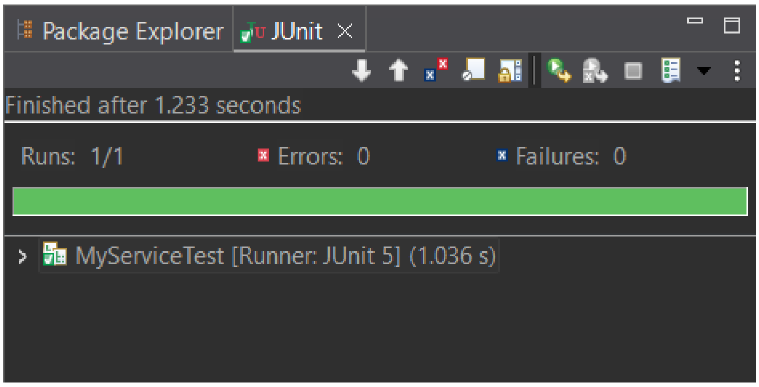
ExternalApi.java

```
package com.example;
public interface ExternalApi
{
        String getData();
}
```

MyServiceTest.java

```
package com.example.test;
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
import org.mockito.Mockito;
import com.example.ExternalApi;
import com.example.MyService;

public class MyServiceTest
{
   @Test
   public void testExternalApi()
   {
      ExternalApi mockApi = Mockito.mock(ExternalApi.class);
      when(mockApi.getData()).thenReturn("Mock Data");
      MyService service = new MyService(mockApi);
      String result = service.fetchData();
      assertEquals("Mock Data", result);
   }
}
```

Output :



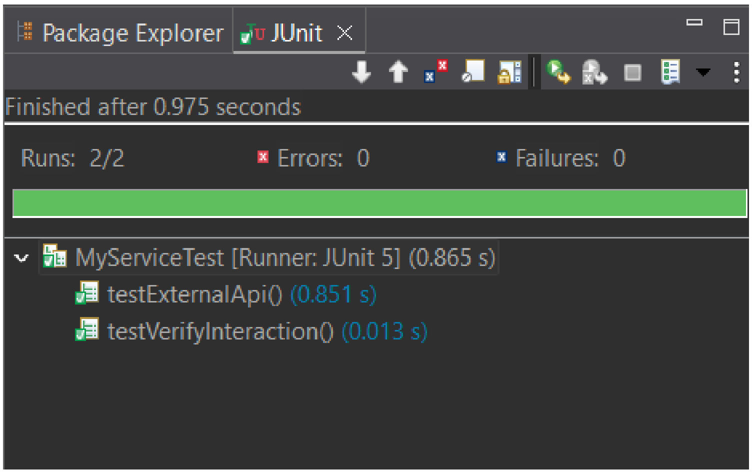
```
Finished after 1.233 seconds

Runs:  1/1          Errors:  0          Failures:  0

> MyServiceTest [Runner: JUnit 5] (1.036 s)
```

Exercise 2: Verifying Interactions

MyServiceTest.java

```
package com.example.test;

import static org.mockito.Mockito.*;

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


import org.junit.jupiter.api.Test;

import org.mockito.Mockito;


import com.example.ExternalApi;

import com.example.MyService;


public class MyServiceTest

{

    @Test

    public void testExternalApi()

    {

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

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


        MyService service = new MyService(mockApi);

        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }


    @Test

    public void testVerifyInteraction()

    {

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

        MyService service = new MyService(mockApi);
        service.fetchData();

        verify(mockApi).getData();

    }

}
```

Output :



```
Finished after 0.975 seconds

Runs:  2/2          Errors:  0          Failures:  0

MyServiceTest [Runner: JUnit 5] (0.865 s)
    testExternalApi() (0.851 s)
    testVerifyInteraction() (0.013 s)
```