Exercise 2: Verifying Interactions in Mockito

Scenario

We have to make sure that when we call a method from our service class, it actualy calls the method from the API (the mocked one). For that we use `verify()` to check if the method was realy called or not.

Step 1: Setup Dependencies (again)

Same as before I added these to my `pom.xml`:

```
<dependency>
   <groupId>org.mockito</groupId>
   <artifactId>mockito-core</artifactId>
   <version>5.10.0</version>
   <scope>test</scope>
</dependency>

<dependency>
   <groupId>org.junit.jupiter</groupId>
   <artifactId>junit-jupiter</artifactId>
   <version>5.10.0</version>
   <scope>test</scope>
</dependency>
```

Step 2: Interface `ExternalApi.java`

```
public interface ExternalApi {
   String getData();
}
```

Step 3: Service Class `MyService.java`

```
public class MyService {
    private ExternalApi api;

    public MyService(ExternalApi api) {
        this.api = api;
    }

    public String fetchData() {
        return api.getData();
    }
}
```

Step 4: Test Class `MyServiceTest.java`

```
import static org.mockito.Mockito.*;
import org.junit.jupiter.api.Test;
import org.mockito.Mockito;

public class MyServiceTest {

    @Test
    public void testVerifyInteraction() {
        // create the mock api
        ExternalApi mockApi = Mockito.mock(ExternalApi.class);

        // create service with mock api
        MyService service = new MyService(mockApi);

        // call the method
        service.fetchData();

        // verify that getData was called
        verify(mockApi).getData();
    }
}
```

Output from `mvn test`

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running MyServiceTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
```