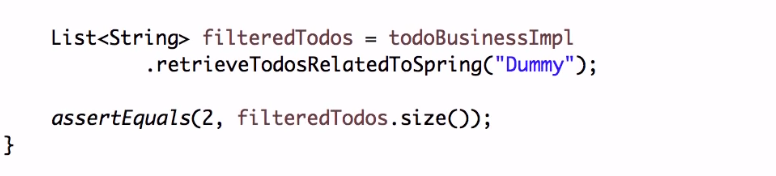
**Mocking: It is creating objects that simulate the behavior of real objects at run time**. We can also use stubbing, where we need to create separate classes for each test with many hard-coded conditions. Mocks also offer more functionality than stubbing and can verify method calls.

We use Mockito framework when we application class uses a web service or dependent on database. It will impact the performance of running test cases if it hits the web service for each change in the test classes.so we use mocking of those services at development stage using Mockito framework.

A screenshot of a cell phone

Description automatically generated 

A screenshot of a cell phone

Description automatically generated

**Nice Mock Behavior:** assumes default functionality when there nothing is defined. **\**

**Argument Matchers:** if you want to get same result for any input parameter. It has methods like “anyInt()” to specify this.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**BDD (Behavior driven development):**

* Splitting user story in to multiple scenarios of given, when, and, then conditions.
* Mockito provides BDD methods to perform those.
* Here, we split the code in to three parts Given, When and Then which makes code readable.

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Mockito Annotations:**

**@Mock:** when we use on the property, it will create a Mock of that specific type and make it available to use. For this to happen we need to use Runner( MockitoJUnitRunner).

**@RunWith :** applied at class level.Here we need to give runner class to work with Mockito annotations.

**@InjectMocks:** Instead of creating the objects, it will inject all dependencies required for that object ( Similar to Auto Wiring). It will check for the dependency mocks in test class and inject them.

**@Captor:** it creates argument captors.

A screenshot of a cell phone

Description automatically generated

Hamcrest Matchers:

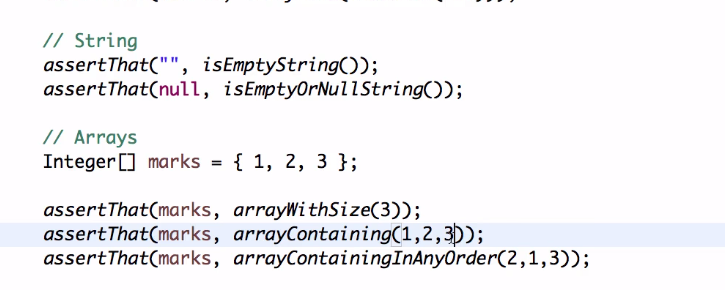
* Make writing asserts easy and readable.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated



* **Mockito doesn’t allow to mock Static methods, so we use powermock to mock them.**

**PowerMock:**

* Using powerMock with Mockito, we can mock static methods.
* Need to use poweMock runner.

**A screenshot of a cell phone

Description automatically generated**

**SPY: Partial Mock**

* By using mock if we do any changes to the methods, it wont effect for next checks. It means we add a element to list it won’t change the value of size of list.
* In order to retain and override the functionalities we need to use SPY.
* It is like almost using the real arraylist but we can stub some methods.
* Also, can verify invocation of methods.
* **Letting the real behavior to continue and stub some methods if needed.**

**A screenshot of a cell phone

Description automatically generated**