SOEN 345 - Software Testing, Verification, and Quality Assurance
Assignment 3

Ali Hanni - 40157164

Gina Cody School of Computer Science and Software Engineering
Concordia University, Montreal, QC, Canada

Winter 2022

Question 1

Since no stubbing is set for in the case the method `.get` with argument value 1, the call `list.get(1)` on the mock object `list` will return the null value. The `assertNull()` method verifies if the parameter's value is null which is indeed the case.
For example, trying the test `assertNull(list.get(0))` will fail since the `list.get(0)` will return the string *"0th element"*, which is not null.

Question 2

Mockito allows to use "ArgumentMatchers" in order to build more flexible stubbs for better testing.
Using one of the different methods of the ArgumentMatchers class allows, for example, to dictate a behaviour for the stubbed object, without having to hard code any parameter value.
Let's look at the example in method `letsMockListGetWithAny`. Using `Mockito.anyInt()` method as a parameter for the `get()` method of the mocked `list` object, allows to stubb a common behaviour no matter the integer given as a parameter. That is why, not matter the integer (0, 1, or 22), the call `list.get(int)` will return *"this is an int"*.
Various other methods like `anyString()`, `anyList()`, or `any(Class <T> type)` (which let you specify what class to match), are very powerful when it comes to stubbing and verifying.

Question 3

First, we mock a `TodoService` object using the `TodoService` interface. Not that the `retrieveTodos()` method of the `TodoService` interface does not have an implementation, all we know is that it must returns a List<String>. We stubb the implementation so that when `todoService.retrieveTodos("Ranga")` is called, a list (allTodos) is returned. Note that the list `allTodos` is a real List<String> object, not a mock. Then we create a real `TodoBusinessImpl` object, giving the mocked `TodoService` object as a parameter. Finally we call `retrieveTodosRelatedToSpring` method of the `TodoBusinessImpl` class on the `todoBusinessImpl` object previously created. For clarity, I joined bellow the implementation of `retrieveTodosRelatedToSpring`. Inside of it is a call to the `retrieveTodos(user)` method. Since the passed parameter is *"Ranga"*, the stubbed method `retrieveTodos` will return the list `allTodos` declared earlier. In that list, two elements have *"Spring"* as a substring. The returned element *filteredTodos* will hence be a list containing two elements. Asserting that the size of that list is equal to 2 will be true.

```java
public List<String> retrieveTodosRelatedToSpring(String user) {
    List<String> filteredTodos = new ArrayList<String>();
    List<String> allTodos = todoService.retrieveTodos(user);
    for (String todo : allTodos) {
        if (todo.contains("Spring")) {
            filteredTodos.add(todo);
        }
    }
    return filteredTodos;
}
```

Implementation of the `retrieveTodosRelatedToSpring` method

Question 5

Bellow is a screen shot that shows that all tests have passed. Note that the source code of the whole project is included with this submission in the folder `./a3_ConcordiaAppDepot`