Spying

Understanding Spies

Creating a Spy

```java
// Simple ArrayList
List<String> list = new ArrayList<>();

// Creating a spy of the list
List<String> listSpy = Mockito.spy(list);

// Now any interaction with listSpy is monitored from this point forward
listSpy.add("one");
listSpy.add("two");

// You can verify the interactions
Mockito.verify(listSpy).add("one");
Mockito.verify(listSpy).add("two");

// And you can also manipulate its behavior if needed
// e.g., overriding the size method
Mockito.doReturn(100).when(listSpy).size();
```

Real Method Invocations

```java
// Continue using the listSpy from previous code snippet

// When you don't stub a method, real methods are called
int size = listSpy.size(); // This will call the actual size method

// Output will be 3 because we have stubbed the size method earlier
System.out.println("Size of list: " + size);
```

Deep Dive into Spying

How to create a spy

```java
List<String> list = new ArrayList<>();
List<String> spyList = Mockito.spy(list);



// use @ExtendWith(MockitoExtension.class) or
// MockitoAnnotations.openMocks(this);

@Spy
List<String> spyList = new ArrayList<>();
```

Stubbing a Spy

```java
doReturn("42").when(spyList).get(0);



doThrow(new RuntimeException()).when(spyList).clear();
```