mockito3 cheat sheet

by Adrian Więch

CREATE

```
//mocks: default values returned by default
ClassName mock = mock(ClassName.class);
  or use: @Mock

//spies: class logic used by default
ClassName spy = spy(ClassName.class);
  or use: @Spy

//static mocks: use try-with-resources
MockedStatic<ClassName> mock =
      mockStatic(ClassName.class));
```

DEFINE BEHAVIOUR

//mocks

```
- when(mock.sampleMethod())
   .thenReturn(sampleVal);
- when(mock.sampleMethod())
   .thenThrow(SampleException.class);
- when(mock.sampleMethod())
   .thenAnswer(inv -> ...)
```

//spies or void methods

```
- doReturn(value)
  .when(spy).method();
- doThrow(new ExceptionClass())
  .when(spy).metohd();
- doNothing().when(spy).method();
```

//static mocks

```
staticMock.when(() ->
  StaticClass.method().thenReturn(val);
```

//argument matchers

```
when(mock.sampleMethod(any(),
  anyDouble())).thenReturn(sampleVal);
```

VERIFY BEHAVIOUR

//method() invoked once

```
- verify(mockClass, times(1)).method();
```

//method never invoked

```
- verify(mockClass, never()).method();
```

//no more methods invoked

```
- verifyNoMoreInteractions(mockClass);
```

ARGUMENT CAPTORS

1. Define:

```
@Captor
private ArgumentCaptor<Double>
  doubleCaptor;
```

2. Capture:

```
verify(mock, times(1)).method(eq(val),
  doubleCaptor.capture());
```

3. Get value:

```
double capturedArgument =
  doubleCaptor.getValue();
```

MOCKITO BDD

// when...thenReturn

```
given(mock.method()).willReturn(val);
```

// verify(class, times(1)).method()

```
then(mock).should(times(1)).method();
```

SAMPLE TEST WITH A MOCK

```
@ExtendWith(MockitoExtension.class)
class ForCheatSheet {

 @InjectMocks
 private BookingService bookingService;

 @Mock
 private RoomService roomServiceMock;

 @Test
 void sample() {
  // given
  when(this.roomServiceMock.getAvailableRooms())
    .thenReturn(Collections.singletonList(new
         Room("Room 1", 5)));
  int expected = 5;

  // when
  int actual =
     bookingService.getAvailablePlaceCount();

  // then
  assertEquals(expected, actual);

 }

}
```