

# Вызовы mock-методов с параметрами

JSP & Servlets  
4 уровень, 3 лекция

ОТКРЫТА

## 4.1 Параметры методов

Предыдущие правила добавленные мок-объекту касались методов без параметров. А как создавать правила для методов с параметрами? Более того, хотелось бы создавать правила, чтобы при одних значениях параметров был один результат, а при других – другой.

Так тоже можно делать. Если вы хотите, чтобы при определенном параметре метод возвращал что-то определенное, то правило можно записать так:

```
Mockito.doReturn(результат).when(объект).имяМетода(параметр);
```

И сразу рассмотрим пример, чтобы лучше все понять. Пусть наш List возвращает имя Иван при запросе 10-го элемента, и имя Марья при запросе 500-го. Сказано – сделано.

```
1  @ExtendWith(MockitoExtension.class)
2  class ParamsTest {
3      @Mock
4      List mockList;
5
6      @Test
7      public void whenMockAnnotation() {
8          //добавление первого правила
9          Mockito.doReturn("Иван").when(mockList).get(10);
10         //добавление второго правила
11         Mockito.doReturn("Марья").when(mockList).get(500);
12
13         assertEquals("Иван", mockList.get(10));
14         assertEquals("Марья", mockList.get(500));
15
16     }
17 }
```

## 4.2 Шаблоны параметров

И сразу хитрые коллеги зададут мне вопрос: “А как быть, если метод требует аргументы, но при любых значениях должен возвращать один и тот же результат?”. Не будем же мы писать:

```
1  Mockito.doReturn("Иван").when(mockList).get(1);
2  Mockito.doReturn("Иван").when(mockList).get(2);
3  Mockito.doReturn("Иван").when(mockList).get(99);
```

Нет, никто тебя так писать не заставляет. Если ты хочешь добавить правило мок-объекту, которое действует для метода с любыми аргументами, то для этого есть специальный объект:

```
Mockito.any()
```

Наш пример с его помощью будет записан так:

```
Mockito.doReturn("Иван").when(mockList).get(any(int.class));
```

Есть тут пара нюансов. Объект `Mockito.any()` имеет тип `Object`, поэтому для параметров разных типов есть его аналоги:

	Метод	Тип параметра
1	<code>any()</code>	Object, включая null
2	<code>any(ClassName.class)</code>	ClassName
3	<code>anyInt()</code>	int
4	<code>anyBoolean()</code>	boolean
5	<code>anyDouble()</code>	double
6	<code>anyList()</code>	List

Более корректно наш пример будет выглядеть так:

```
Mockito.doReturn("Иван").when(mockList).get(anyInt());
```

### 4.3 Метод doAnswer()

Мы добрались до сложного поведения виртуальных методов. Рано или поздно наступит ситуация, когда ты захочешь, чтобы этот виртуальный метод имел сложное поведение. Например, он должен возвращать значения в зависимости от параметров, преобразовывать строку в верхний регистр.

Для этого есть специальный метод – `doAnswer()`, в который передается функция, которая делает то, что тебе нужно:

```
Mockito.doAnswer(функция).when(объект).имяМетода(параметр);
```

Давай заставим метод `get()` класса `List` возвращать квадрат переданного ему аргумента. И напишем такую программу:

```
1  @ExtendWith(MockitoExtension.class)
2  class DoAnswerTest {
3      @Mock
4      List mockList;
5
6      @Test
7      public void whenMockAnnotation() {
8          Mockito.doAnswer(invocation -> {
9              int parameter = invocation.getArgument(0);
10             return parameter * parameter;
11         }).when(mockList).get(anyInt());
```

```
12
13         assertEquals(100, mockList.get(10));
14         assertEquals(25, mockList.get(5));
15     }
16 }
```

Функцию мы задали с помощью объект класса Answer.

< Предыдущая лекция

Следующая лекция >

-

+16

+

Комментарии (5)

популярные новые старые

JavaCoder

Введите текст комментария

Сергей

Уровень 65, Воронеж, Россия

1 октября, 19:21

...

Вместо

1 Mockito.doReturn("Иван").when(mockList.get(any(int.class)));

2 Mockito.doReturn("Иван").when(mockList.get(anyInt()));

Нужно писать так:

1 Mockito.doReturn("Иван").when(mockList).get(any(int.class));

2 Mockito.doReturn("Иван").when(mockList).get(anyInt());

Иначе будет:

1 org.mockito.exceptions.misusing.UnfinishedStubbingException:

2 Unfinished stubbing detected here:

3 -> at MockitoTest.doReturnStringListWhenAnySizeList(MockitoTest.java:101)

4

5 E.g. thenReturn() may be missing.

6 Examples of correct stubbing:

7 when(mock.isOk()).thenReturn(true);

8 when(mock.isOk()).thenThrow(exception);

9 doThrow(exception).when(mock).someVoidMethod();

10 Hints:

11 1. missing thenReturn()

12 2. you are trying to stub a final method, which is not supported

13 3. you are stubbing the behaviour of another mock inside before 'thenReturn' inst

Ответить

-

0


+

Stas S

Уровень 76, Гродно, Беларусь

8 сентября, 11:15

...



Оба примера упадут с UnfinishedStubbingException и не потому, что там не хватает круглой скобки :)  
Правильно писать так :

1 Mockito.doAnswer(функция).when(объект).имяМетода(параметр);

Ответить

-

0

+

Никита

Уровень 76, Киев, Украина

25 июля, 22:12

...

getArgument - это метод с рефлексии?

Ответить

Тимофей

Уровень 51, Москва, Россия

21 июля, 14:43

...

в примере с `any()` пропустили скобку.

Ответить

0

Dolly

Уровень 1, Belarus

14 июля, 03:38

...

Сытно -\_-

Ответить

0

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

▼

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

