Лекции

Поиск

Карта квестов Лекции CS50 Android

Вызовы mock-методов с параметрами

```
JSP & Servlets
4 уровень, 3 лекция
```

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
              Mockito.doReturn("Марья").when(mockList).get(500);
11
12
              assertEquals("Иван", mockList.get(10));
13
14
              assertEquals("Марья", mockList.get(500));
15
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:

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

Нет, никто тебя так писать не заставляет. Если ты хочешь добавить правило mock-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	any()	Object, включая null
2	any(ClassName.class)	ClassName
3	anyInt()	int
4	anyBoolean()	boolean
5	anyDouble()	double
6	anyList()	List

Более корректно наш пример будет выглядеть так:

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

4.3 Meтод 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:

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

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

Функцию мы задали с помощью объект класса Answer.

< Предыдущая лекция</p>

Следующая лекция >



Комментарии (5) популярные новые старые **JavaCoder** Введите текст комментария Сергей Уровень 65, Воронеж, Россия 1 октября, 19:21 ••• Вместо Mockito.doReturn("Иван").when(mockList.get(any(int.class))); 1 2 Mockito.doReturn("Иван").when(mockList.get(anyInt()); Нужно писать так: Mockito.doReturn("Иван").when(mockList).get(any(int.class)); Mockito.doReturn("Иван").when(mockList).get(anyInt()); Иначе будет: $\verb"org.mockito.exceptions.misusing.Unfinished Stubbing Exception:$ 1 Unfinished stubbing detected here: 2 3 -> at MockitoTest.doReturnStringListWhenAnySizeList(MockitoTest.java:101) 4 E.g. thenReturn() may be missing. 5 Examples of correct stubbing: 6 7 when(mock.isOk()).thenReturn(true); when(mock.isOk()).thenThrow(exception); 8 doThrow(exception).when(mock).someVoidMethod(); 9 10 Hints: 1. missing thenReturn() 11 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 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:

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

Ответить



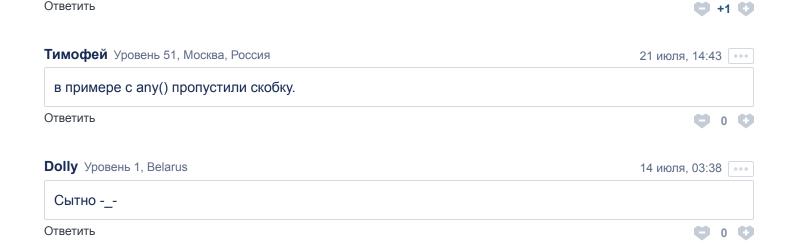




getArgument - это метод с рефлексии?

Никита Уровень 76, Киев, Украина

25 июля, 22:12



ОБУЧЕНИЕ сообщество КОМПАНИЯ Курсы программирования Пользователи Онас Статьи Контакты Kypc Java Форум Отзывы Помощь по задачам Чат FAQ Подписки Задачи-игры Истории успеха Поддержка Активности



RUSH

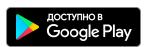
JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном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

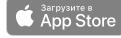
ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА



СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ







"Программистами не рождаются" © 2022 JavaRush