

## Мокирование объектов

JSP & Servlets  
4 уровень, 2 лекция

ОТКРЫТА

### 3.1 Метод doReturn()

А теперь будет магия...

Допустим, ты создал фейковый мок-объект, но ведь нужно чтобы он как-то работал. При вызове определенных методов делалось что-то важное или методы возвращали определенный результат. Что делать?

Библиотека Mockito позволяет добавить мок-объекту нужное поведение.

Если ты хочешь, чтобы при вызове определенного метода, мок-объект вернул определенный результат, то это “правило” можно добавить объекту с помощью кода:

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

Видишь, в конце вызова метода `имяМетода?` На самом деле никакого вызова тут не происходит. Метод `doReturn()` возвращает специальный проху-объект с помощью которого следит за вызовами методов объекта и, таким образом, идет запись правила.

Еще раз. Это просто такой хитрый способ записать правило, которое нужно добавить к мок-объекту. Нужна определенная сноровка, чтобы правильно интерпретировать такой код в своей голове, когда его видишь. С опытом приходит.

Думаю, нужен конкретный пример. Давайте создадим мок-объект класса `ArrayList` и попросим его метод `size()` вернуть число 10. Полный вариант кода будет выглядеть так:

```
1  @ExtendWith(MockitoExtension.class)
2  class DoReturnTest {
3      @Mock
4      List mockList;
5
6      @Test
7      public void whenMockAnnotation () {
8          //создаем правило: вернуть 10 при вызове метода size
9          Mockito.doReturn(10).when(mockList).size();
10
11         //тут вызывается метод и вернет 10!!
12         assertEquals(10, mockList.size());
13     }
14 }
```

Да, этот код будет работать, тест не упадет.

### 3.2 Метод when()

Есть еще один способ добавить правило поведения к мок-объекту – через вызов метода `Mockito.when()`. Выглядит вот так:

```
Mockito.when(объект.имяМетода()).thenReturn(результат);
```

Это такой же способ записи правила поведения мок-объекта, как и предыдущий. Сравните:

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

Тут происходит абсолютно одно и то же – конструирование нового правила.

Правда первый пример имеет два минуса:

- вызов `объект.имяМетода()` сильно сбивает с толку.
- не будет работать, если метод `имяМетода()` возвращает `void`.

Ну и давайте запишем полюбившийся нам пример с помощью `Mockito.when()`

```
1  @ExtendWith(MockitoExtension.class)
2  class WhenTest {
3      @Mock
4      List mockList;
5
6      @Test
7      public void whenMockAnnotation() {
8          //создаем правило: вернуть 10 при вызове метода size
9          Mockito.when(mockList.size()).thenReturn(10);
10
11         //тут вызывается метод и вернет 10!!
12         assertEquals(10, mockList.size());
13     }
14 }
```

### 3.3 Метод doThrow()

Мы разобрались, как сделать так, чтобы метод мок-объекта вернул определенный результат. А как сделать так, чтобы он кинул определенное исключение? Передать его в `doReturn()` ?

Чтобы метод не вернул, а именно выбросил (throw) исключение, нужно задать правило с помощью метода `doThrow()` .

```
Mockito.doThrow(исключение.class).when(объект).имяМетода();
```

И сразу второй вариант:

```
Mockito.when(объект.имяМетода).thenThrow(исключение.class);
```

Немного ожидаемо, да?

Ну вот видишь, ты уже начинаешь разбираться. Закрепим примером:

```
1  @ExtendWith(MockitoExtension.class)
2  class DoThrowTest {
3      @Mock
```

```
4      List mockList;
5
6      @Test
7      public void whenMockAnnotation() {
8          Mockito.when(mockList.size() ).thenThrow(IllegalStateException.class);
9          mockList.size(); //тут кинется исключение
10     }
11 }
```

Если нужно выкинуть определенный объект-исключение, то воспользуйся конструкцией вида:

```
Mockito.doThrow(new Исключение()).when(объект).имяМетода();
```

Просто передай в метод `doThrow()` объект исключения и он будет выброшен во время вызова метода.

[← Предыдущая лекция](#)

[Следующая лекция →](#)

+14

Комментарии (3)

популярные    новые    старые

JavaCoder

Введите текст комментария

Нурбек Абдыканов    Уровень 2, Kazakhstan    14 октября, 12:50   

Толку от этого? если мы задаем сами, то что должен метод возвращать (не смотря на внутреннюю бизнес логику метода). Мы можем просто сделать  
**when(Scanner.nextLine()).thenReturn(15);**  
Хоть nextLine не возвращает Int, мы пройдем тест прописав  
**assertEquals( 15 , Scanner.nextLine() );**

Это то же самое что сказать, A=15 и спросить Равен ли a==15?

Ответить    0

Stas S    Уровень 76, Гродно, Беларусь    5 сентября, 11:51   

Так в чем разница между "исключение.class" и "new Исключение()" , переданных в doThrow/thenThrow?

Метод не будет вызван, отработает правило, заданное для него.

Ответить    +1

Владимир    Уровень 68, Тель-Авив, Израиль    24 августа, 21:33   

"Но вот видите, ты уже начинаешь разбираться." Давайте уже на "ты", чо мы как не родные?

Ответить    +7

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

