

Популярные аннотации в JUnit

JSP & Servlets
3 уровень, 6 лекция

ОТКРЫТА

7.1 Аннотация @Suite

И еще несколько часто используемых аннотаций. Даже если ты не будешь их использовать, то точно увидишь тесты с ними. Поэтому нужно хотя бы в общих чертах понимать, что там написано.

Тесты можно объединять в группы. Для этого есть специальная аннотация `@Suite`. Пример:

```
1  @Suite
2  @SuiteDisplayName("JUnit Platform Suite Demo")
3  @SelectPackages("com.javarush.test")
4  @IncludeClassNamePatterns(".*Tests")
5  class SuiteDemo {
6  }
```

В данном случае аннотация `@Suite` работает в паре с другими аннотациями.

- `SuiteDisplayName` – задает имя группы тестов в логе;
- `SelectPackages` – задает список пакетов, где искать классы-тесты;
- `IncludeClassNamePatterns` – задает шаблон имен классов-тестов.

Зачем такие сложности? Ну представь, что все тесты проекта выполняются, скажем, 50 часов. И выполнять их после каждого коммита очень накладно. В таком случае полезно создать некоторые отдельные сценарии тестирования и настроить тестирование гораздо гибче.

7.2 Аннотация @Order

Еще одна часто встречающаяся аннотация – `@TestMethodOrder`. Она позволяет задать порядок вызова тестовых методов в тестовом классе. Бывает очень полезна, когда вы знаете, что вызовы методов влияют друг на друга, но при определенном порядке все должно работать, как нужно. Используется довольно часто.

Во-первых, можно задать **вызов методов в алфавитном порядке**:

```
1  @TestMethodOrder(MethodOrderer.MethodName.class)
2  public class AlphanumericOrderUnitTest {
3
4  }
```

Во-вторых, у каждого метода можно расставить специальную аннотацию с его порядковым номером.

```
1  @TestMethodOrder(OrderAnnotation.class)
2  public class OrderAnnotationUnitTest {
3
4      @Test
5      @Order(1)
6      public void firstTest() {
7
8      }
```

```
9      @Test
10      @Order(2)
11      public void secondTest() {
12      }
13  }
```

Или даже создать отдельный класс, который будет указывать порядок вызова тестов:

```
1  @TestMethodOrder(CustomOrder.class)
2  public class CustomOrderUnitTest {
3
4  }
5
6  //сортируем имена методов по алфавитному порядку, но игнорируя регистр
7  public class CustomOrder implements MethodOrderer {
8      public void orderMethods(MethodOrdererContext context) {
9          context.getMethodDescriptors().sort(
10              (MethodDescriptor m1, MethodDescriptor m2)->
11                  m1.getMethod().getName().compareToIgnoreCase(m2.getMethod().getName()));
12      }
13  }
```

7.3 Аннотация @DisplayName

И наконец, каждому тесту можно задавать его имя. Бывает удобно, если тестов очень много и вы создаете специальные сценарии (подмножества) тестов. Для этого есть специальная аннотация `@DisplayName`.

Пример:

```
1  @DisplayName("Понятное имя для теста")
2  public class DisplayNameCustomTest {
3
4      @Test
5      @DisplayName("Проверка входных данных")
6      void inputData() {
7      }
8
9      @DisplayName("Проверка критических ситуаций")
10     @Test
11     void criticalCases() {
12     }
13 }
```

Как и в случае с заданием порядка тестов, тут можно создать специальный метод, который будет генерировать имена тестов и тестовых методов. Пример:

```
1  @DisplayNameGeneration(DisplayNameGeneratorUnitTest.ReplaceCamelCase.class)
2  class DisplayNameGeneratorUnitTest {
3
4      @Test
5      void camelCaseName() {
6      }
7
8      static class ReplaceCamelCase extends DisplayNameGenerator.Standard {
9          @Override
10         public String generateDisplayNameForClass(Class testClass) {
```

```
11         return super.generateDisplayNameForClass(testClass).toUpperCase();
12     }
13 }
14 }
```

Видите, как все просто :)

< Предыдущая лекция

Следующая лекция >

-

+10

+

Комментарии (2)

популярные

новые

старые

JavaCoder

Введите текст комментария

Анраник Мамиконян

Уровень 69, Москва, Россия

23 июня, 10:49

...

Как и в случае с заданием порядка тестов, тут можно создать специальный метод, который будет генерировать имена тесто* и тестовых методов
* - букву в пропустили
И либо я не заметил, либо реально не было ни слова сказано за весь модуль про то, что значит сама по себе аннотация @Test

Ответить

-

0

+

(delta)Roman_Sula

Уровень 78, Пинск, Беларусь

17 июля, 13:32

...

https://javarush.ru/quests/lectures/questservlets.level03.lecture01

Ответить

-

0

+

[Помощь по задачам](#)

[Подписки](#)

[Задачи-игры](#)

[Форум](#)

[Чат](#)

[Истории успеха](#)

[Активности](#)

[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

