

Параметризованные тесты в JUnit

JSP & Servlets
3 уровень, 4 лекция

ОТКРЫТА

5.1 Аннотация @ParameterizedTest

Иногда просто хочется вызвать тест несколько раз с различными параметрами: разные значения, разные входные параметры, разные имена пользователей. JUnit стремится упростить тебе жизнь, поэтому на этот случай у него есть такая вещь как параметризованные тесты.

Для использования параметризованных тестов надо добавить еще одну зависимость в ваш `pom.xml`:

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-params</artifactId>
  <version>5.8.2</version>
  <scope>test</scope>
</dependency>
```

После чего можем рассмотреть пример:

```
1  @ParameterizedTest
2  @ValueSource(ints = { 1, 2, 3 })
3  void testMethod(int argument) {
4      //test code
5  }
6
7  @ParameterizedTest
8  @ValueSource(ints = { 1, 2, 3 })
9  void testMethodWithAutoboxing(Integer argument) {
10     //test code
11 }
```

Каждый тестовый метод вызовется по 3 раза и при каждом вызове в него будет передан очередной параметр. Значения задаются с помощью другой аннотации — **@ValueSource**. Но о ней нужно рассказать подробнее.

5.2 Аннотация @ValueSource

Аннотация **@ValueSource** отлично подходит для работы с примитивами и литералами. Просто перечисли значения параметра через запятую и тест будет вызван по одному разу для каждого значения.

```
1  @ParameterizedTest
2  @ValueSource(ints = { 1, 2, 3 })
3  void testMethodWithAutoboxing(Integer argument) {
4      //test code
5  }
```

Но есть и минус — эта аннотация не поддерживает null, так что для него нужно будет использовать специальный костыль — **@NullSource**. Выглядит это так:

```
1  @ParameterizedTest
2  @NullSource
3  void testMethodNullSource(Integer argument) {
4      assertTrue(argument == null);
5  }
```

5.3 Аннотация @EnumSource

Также есть специальная аннотация **@EnumSource**, которая передает в метод все значения определенного Enum’a. Ее применение выглядит так:

```
1  enum Direction {
2      EAST, WEST, NORTH, SOUTH
3  }
4
5  @ParameterizedTest
6  @EnumSource(Direction.class)
7  void testWithEnumSource(Direction d) {
8      assertNotNull(d);
9  }
```

5.4 Аннотация @MethodSource

А как же передавать в качестве параметров объекты? Особенно, если они имеют сложный алгоритм построения. Для этого можно просто указать специальный вспомогательный метод, который будет возвращать список (Stream) из таких значений.

Пример:

```
1  @ParameterizedTest
2  @MethodSource("argsProviderFactory")
3  void testWithMethodSource(String argument) {
4      assertNotNull(argument);
5  }
6
7  static Stream<String> argsProviderFactory() {
8      return Stream.of("один", "два", "три");
9  }
```

Параметризованные тесты с несколькими аргументами

Конечно вы уже задались вопросом, а что делать, если в метод хочется передать несколько параметров. Для этого есть очень классная аннотация **@CSVSource**. Она позволяет перечислить значения параметров метода просто через запятую.

Пример:

```
1  @ParameterizedTest
2  @CsvSource({
3      "alex, 30, Программист, Работает",
4      "brian, 35, Тестировщик, Работает",
5      "charles, 40, Менеджер, Пинает"
6  })
7  void testWithCsvSource(String name, int age, String occupation, String status) {
8      //код метода
9  }
```

−

+25

+

Комментарии (6)

популярные

новые

старые

JavaCoder

Введите текст комментария

Антон Уровень 68, Иваново, Россия

23 августа, 22:01

...

"Пинает"

Ответить

−

0

+

Иван Голубев Уровень 79, Москва, Россия

10 июля, 16:44

...

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-params</artifactId>
  <version>5.8.2</version>
  <scope>test</scope>
</dependency> дополнительно для параметризированных тестов в maven
```

Ответить

−

+1

+

Jh-007 Уровень 47

22 июня, 06:49

...

```
1  @ParameterizedTest
2  @ValueSource(ints = { 1, 2, 3 })
3  void testMethod(int argument) {
4      //test code
5  }
```

А как результат проверить? Как писать assert для нескольких значений?

Ответить

−

0

+

Anonymous #2957882 Уровень 41, Днепр, Ukraine

12 августа, 16:13

...

На примере калькулятора

Ответить

−

+6

+

Kate Уровень 14

2 июня, 12:56

...

При использовании @MethodSource также можно передавать несколько параметров в тестовый метод через запятую:

```
1  @ParameterizedTest
2  @MethodSource("argsProviderFactory")
3  void testWithMethodSource(String argument1, String argument2, String argument3) {
4      assertNotNull(argument);
5  }
6
7  static Stream<String> argsProviderFactory() {
8      return Stream.of("один", "два", "три");
9  }
```

Ответить

−

+1

+

Модератор Безработный с 24.12.2021

29 мая, 21:24

...

лайк

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

 **ДОСТУПНО В**
Google Play

 Загрузите в
App Store

