Класс ObjectUtils из Apache Commons

JSP & Servlets
20 уровень, 3 лекция

ОТКРЫТА

Знакомство с классом ObjectUtils

Методы:

`allNotNull(Object... values)`	Проверяет, что все объекты не null
`allNull(Object... values)`	Проверяет, что все объекты равны null
`anyNotNull(Object... values)`	Проверяет, что хотя бы один объект не null
`anyNull(Object... values)`	Проверяет, что хотя бы один объект null
`clone(T obj)`	Клонирует объект
`cloneIfPossible(T obj)`	Клонирует объект или возвращает оригинал
`compare(T c1, T c2)`	Сравнивает объекты
`defaultIfNull(T object, T defaultValue)`	Возвращает default-объект, если object равен null
`equals(Object object1, Object object2)`	Сравнивает два объекта
`notEqual(Object object1, Object object2)`	Проверят, что два объекта не равны
`firstNonNull(T... values)`	Возвращает первый объект, который не null
`getFirstNonNull(Supplier... suppliers)`	Возвращает первый объект, который не null
`getIfNull(T object, Supplier defaultSupplier)`	Возвращает данный объект, если он не равен null, иначе возвращает значение Supplier.get() переданного Supplier
`hashCode(Object obj)`	Вычисляет hashCode для объекта
`hashCodeMulti(Object... objects)`	Вычисляет hashCode для группы объектов
`isEmpty(Object object)`	Проверяет, что объект empty или null
`isNotEmpty(Object object)`	Проверяет, что объект не empty или null
`requireNonEmpty(T obj)`	Проверяет, что объект не null, иначе кидает исключение

`requireNonEmpty(T obj, String message)`	Проверяет, что объект не null, иначе кидает исключение
`identityToString(Object object)`	Возвращает строку для объекта
`toString(Object obj)`	Возвращает строку для объекта
`toString(Object obj, String nullStr)`	Возвращает строку для объекта
`toString(Object obj, Supplier supplier)`	Возвращает строку для объекта

Давай рассмотрим по одному методу из каждой группы. Я надеюсь, ты будешь часто ими пользоваться, ведь они очень удобные и позволяют избегать лишнего кода.

ObjectUtils.compare()

Метод сравнивает объекты по такому же принципу, как и comparator: больше, меньше или равно. Его можно использовать для сортировки объектов.

Сигнатура метода имеет вид:

```
public static <T extends Comparable<? super T>> int compare(final T c1, final T c2);
public static <T extends Comparable<? super T>> int compare(final T c1, final T c2, final boolean nullGr
```

Если третий параметр (nullGreater) равен *true*, то ***null*** будет всегда считаться больше, чем не ***null***. Метод возвращает позитивное значение, если c1> c2, негативное, если c1<c2, и 0, если c1 == c2.

Пример:

```
String firstValue = "javaRush";
String secondValue = "javaRush";
System.out.print(ObjectUtils.compare(firstValue, secondValue));
System.out.println();

firstValue = "javaRush";
secondValue = null;
System.out.print(ObjectUtils.compare(firstValue, secondValue));
System.out.println();

firstValue = "";
secondValue = "javaRush";
System.out.print(ObjectUtils.compare(firstValue, secondValue));
System.out.println();
```

Программа выведет на экран результат:

```
0
1
-8
```

ObjectUtils.isNotEmpty()

Метод `isNotEmpty()` проверяет, что переданный в него объект не пустой и не ***null***.

Сигнатура метода:

```
1  public static boolean isNotEmpty(final Object object)
```

Пример:

```
1  List<String> values = new ArrayList<>();
2  System.out.println(ObjectUtils.isNotEmpty(values));
3
4  values.add("javaRush");
5  System.out.println(ObjectUtils.isNotEmpty(values));
6
7  values = null;
8  System.out.println(ObjectUtils.isNotEmpty(values));
```

На экран будет выведен результат:

```
false
true
false
```

java.util.Objects

Разработчикам Java очень понравилась идея с `ObjectUtils`, так что в JDK 7 они добавили свой вариант:

`isNull(Object obj)`	Проверяет, что объект равен null
`nonNull(Object obj)`	Проверяет, что объект не равен null
`toString(Object o)`	Преобразовывает объект в строку
`toString(Object o, String nullDefault)`	Преобразовывает объект в строку
`boolean equals(Object a,Object b)`	Сравнивает объекты
`boolean deepEquals(Object a,Object b)`	Сравнивает объекты
`T requireNonNull(T obj)`	Проверяет, что переданный параметр не null
`T requireNonNull(T obj,String message)`	Проверяет, что переданный параметр не null
`int hashCode(Object o)`	Вычисляет hashCode для объекта
`int hash(Object… values)`	Вычисляет hashCode для группы объектов
`int compare(T a,T b,Comparator c)`	Сравнивает объекты

Так как класс `java.util.Objects` входит в состав JDK, рекомендуется использовать его в коде.

Важно заметить, что когда ты будешь читать чужой код, то скорее всего встретишь варианты из `ObjectUtils`, такое часто бывает в open-source. Тут ты можешь посмотреть, чем они отличаются.

‹ Предыдущая лекция

Следующая лекция ›

+7

Комментарии (1) популярные новые старые

JavaCoder

Введите текст комментария

Иван Голубев Уровень 84 19 сентября 2022, 12:59

Во второй задаче массив array2 в методе main не указан. Тест ругался, что изменен метод main. В готовом решении массив array2 указан.

Ответить +6

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ