

Arrays用法总结

原创

DaydreamerZz

2018-03-23 17:29:17


15928

收藏 95

版权

分类专栏：

java

 java 专栏收录该内容

0 订阅 9 篇文章

订阅专栏

数组转字符串

```
1 int[] array = new int[]{1, 2, 3};
2 out.println(Arrays.toString(array)); //[1, 2, 3]
```

如果是一维数组，toString方法可以很好的适用。但遇到多维数组时，需要使用deepToString把数组完全转成字符串。

```
1 int[][] deepArray = new int[][]{{1, 3},{2, 4}};
2 out.println(Arrays.toString(deepArray)); [[[I@1540e19d, [I@677327b6]
3 out.println(Arrays.deepToString(deepArray)); [[[1, 3], [2, 4]]
```

填充数组

```
1 array = new int[5];
2 Arrays.fill(array, 2);
3 out.println(Arrays.toString(array)); //[2, 2, 2, 2, 2]
4
5 array = new int[5];
6 Arrays.fill(array, 1, 4, 2); //部分填充
7 out.println(Arrays.toString(array));//[0, 2, 2, 2, 0]
```

数组元素排序

```
1 array = new int[]{3, 10, 4, 0, 2};
2 Arrays.sort(array);
3 out.println(Arrays.toString(array)); //[0, 2, 3, 4, 10]
4
5 array = new int[]{3, 10, 4, 0, 2};
6 Arrays.parallelSort(array); //和sort相比是这个是并行的
7 out.println(Arrays.toString(array)); //[0, 2, 3, 4, 10]
8
9 array = new int[]{3, 10, 4, 0, 2};
10 Arrays.sort(array, 0, 4); //部分排序
11 out.println(Arrays.toString(array)); //[0, 3, 4, 10, 2]
```

数组的比较

```
1 array = new int[]{1, 2, 3};
2 int[] array2 = new int[]{1, 2, 3};
3 out.println(Arrays.equals(array, array2)); //true
```

和toString方法一样，equals方法遇到多维数组时也会出现问题。

```
1 int[][] deepArray1 = new int[][]{{1, 3},{2, 4}};
2 int[][] deepArray2 = new int[][]{{1, 3},{2, 4}};
3 out.println(Arrays.equals(deepArray1, deepArray2)); //false
4 out.println(Arrays.deepEquals(deepArray1, deepArray2)); //true
```

deepEquals用于判定两个指定数组彼此是否深层相等，此方法适用于任意深度的嵌套数组。

equals用于判定两个数组是否相等，如果两个数组以相同顺序包含相同元素，则返回true。

如果两个数组使用equals返回true，则使用deepEquals必定也返回true，也就是说在比较的两个数组均为一维数组的前提下，equals和deepEquals的比较结果没有差别。


数组复制

```
1 array = new int[]{3, 10, 4, 0, 2};
2 int[] arrayCopy = Arrays.copyOf(array, 3);
3 out.println(Arrays.toString(arrayCopy)); //[3, 10, 4]
4
5 arrayCopy = Arrays.copyOf(array, 7);
6 out.println(Arrays.toString(arrayCopy)); //[3, 10, 4, 0, 2, 0, 0], 多出的长度补0
7
8 arrayCopy = Arrays.copyOfRange(array, 1, 4);
9 out.println(Arrays.toString(arrayCopy)); //[10, 4, 0]
```

二分查找返回下标

```
1 array = new int[]{0, 3, 4, 10, 20};
2 out.println(Arrays.binarySearch(array, 10)); //3, array必须是排序的，否则得到的是错误的结果
3 out.println(Arrays.binarySearch(array, 6)); //-4, 找不到的值，从-1开始，6如果存在下标是3， 所以返回-4
4 out.println(Arrays.binarySearch(array, 2, 5, 10)); //3, 返回的还是全局的下标值。
```

数组转List

DaydreamerZz

关注

13

3

95

专栏目录







举报

```
1 int array = new int[]{3, 10, 4, 0, 2};
2 out.println(Arrays.asList(array).contains(3)); //false
3
4 Integer arr[] = new Integer[]{3, 10, 4, 0, 2};
5 out.println(Arrays.asList(arr).contains(3)); //true
```

这里是比较有意思的地方，实际上拆开来看是这样的

```
1 int array = new int[]{3, 10, 4, 0, 2};
2 List<int[]> ints = Arrays.asList(array);

1 Integer arr[] = new Integer[]{3, 10, 4, 0, 2};
2 List<Integer> integers = Arrays.asList(arr);
```

现在就知道区别了，原始数据类型int的数组调用asList之后得到的List只有一个元素，这个元素就是元素类型的数组。而封装类Integer数组调用asList是把数组中每个元素加到了List中。

对数组元素采用指定的方法计算

```
1 array = new int[]{3, 10, 4, 0, 2};
2 Arrays.parallelPrefix(array, (x,y)->(x+y)); //[3, 13, 17, 17, 19]
3 out.println(Arrays.toString(array));
4
5 array = new int[]{3, 10, 4, 0, 2};
6 Arrays.parallelSetAll(array, (x)->(x*x)); //[0, 1, 4, 9, 16]
7 out.println(Arrays.toString(array));
8
9 Arrays.setAll(array, (x)->(x%3));
10 out.println(Arrays.toString(array)); //[0, 1, 2, 0, 1], 与parallelSetAll相比只是不并行
```

对其他对象数组进行排序

一个对象数组，排序算法需要重复比较数组中的元素。不同的类比较元素的规则是不同的，但是排序算法只应该调用类提供的比较方法，只要所有的类就比较的时候提供的方法达成一致，那么排序算法就能开始工作。这个在排序时对象之间进行比较方法就可以是一个接口，所有需要比较的对象继承这个接口并且实现比较的方法，就可以对这些对象进行排序。

如果一个类想启用对象排序，那么就应该实现Comparable接口。

```
1 public class Test{
2     public static void main(String[] args){
3         Employee[] employees = new Employee[3];
4         employees[0] = new Employee(20);
5         employees[1] = new Employee(10);
6         employees[2] = new Employee(30);
7         Arrays.sort(employees);
8         for(Employee e : employees){
9             System.out.println(e); //Employee{id=10} Employee{id=20} Employee{id=30}
10        }
11    }
12 }
13 static class Employee implements Comparable<Employee>{
14     private int id;
15     public Employee(int id){this.id = id;}
16 @Override
17     public int compareTo(Employee o) {
18         return this.id - o.id;
19     }
20 @Override
21     public String toString() {
22         return "Employee{" + "id=" + id + '}';
23     }
24 }
25 }
```

自定义排序规则

```
1 String[] names = {"tom", "alice", "fred"};
2 Arrays.sort(names);
3 out.println(Arrays.toString(names));
```

假如想根据字符串的长度而不是根据字典顺序对字符串排序，但是String类我们是无法修改的。上面的代码对String数组进行排序，只能按照字典顺序对String数组进行排序。

Arrays.sort 方法和 Collections.sort 方法都提供了一个可以接收 Comparator 实例作为第二个参数的版本。

要按照长度比较字符串，定义一个实现 Comparator<String> 的类。

```
1 public class LengthComparator implements Comparator<String> {
2     @Override
3     public int compare(String o1, String o2) {
4         return o1.length() - o2.length();
5     }
6     public static void main(String[] args){
7         String[] names = {"tom", "alice", "fred"};
8         Arrays.sort(names, new LengthComparator());
9         out.println(Arrays.toString(names));
10    }
11 }
```

像Comparator、Runnable等这一些接口有一个特点就是只有一个抽象方法（其他的都是static或者default的方法），比如继承Comparator接口只需要重写compareTo方法，做了类似的事情，这跟自己重写toString方法，这种类型的接口方法，函数式接口，可以被lambda表达式所代替。



举报

像Comparator、Runnable等这一些接口有一个特点就是只有一个抽象方法（其他的都是static或者default的方法），比如继承Comparator接口只需要重写compareTo方法，做了类似的事情，这跟自己重写toString方法，这种类型的接口方法，函数式接口，可以被lambda表达式所代替。



DaydreamerZz

关注

👍 13

💬 3

🌟 95



专栏目录

比如上面根据字符串的长度进行排序的代码，`Arrays.sort` 的第二个参数是需要实现了Comparator接口的实例，用lambda表达是就可以写成这样：

```
1 String[] names = {"tom", "alice", "fred"};
2 Comparator<String> comp = (first, second) -> first.length() - second.length();
3 Arrays.sort(names, comp);
```

或者更加简单点

```
1 String[] names = {"tom", "alice", "fred"};
2 Arrays.sort(names, (first, second) -> first.length() - second.length());
```

`Arrays.sort`方法的第二个参数变量接受一个实现了Comparator接口的类的实例，调用该对象的compare方法会执行lambda表达式中的代码，所以这也就是为什么接口只有一个抽象方法的时候可以用lambda表达式代替。

Java `Arrays`工具类用法详解 08-26
主要介绍了Java `Arrays`工具类用法,结合实例形式分析了java `Arrays`工具类针对数组元素修改、复制、排序等操作使用技巧与相关...

深入理解`Arrays.sort()` 热门推荐 github_38838414的博客 5万+
深入理解Java 中的`Arrays.sort()` 方法 Java的`Arrays`类中有一个sort()方法，该方法是`Arrays`类的静态方法，在需要对数组进行排序...

zzuhai: `setAll`是用的下标来计算的，怎么用数组的元素来计算呢 9 月前 回复 ...

weixin_43578748: `asList`处，因为List中只能存储引用数据类型，所以它不会将int类型数组中的每个元素自动转化为Integer类型进行添加。但是Integer类型的数组就可以进行自动转化。 1 年前 回复 ...

weixin_47428776 回复： 原来如此 1 月前 回复 ...

`Arrays`使用详解_大灰狼的专栏 7-18
System.out.println(`Arrays.binarySearch`(b,3));//使用二分查找法查找3的位置 }} 此外,在System类里也包含了一个static void arrayco...

Array与`Arrays`的区别(数组转list)_gm13508613743的博客 7-16
`Arrays.asList`确实会返回一个ArrayList对象,但是该类是`Arrays`类 中一个私有静态内部类,而不是常见的java.util.ArrayList类。这个jav...

`Arrays`类的用法详述 买米当卡的博客 2203
java.util.`Arrays` 类能方便地操作数组，它提供的所有方法都是静态的。 例子如下： import java.util.`Arrays`; public class Test3 { publi...

`Arrays`的用法 `arrays`是Sun公司就在JDK中提供了一个数组的工具类，此类包含用来操作数组（比如排序和搜索）的各种方法。此... 你好，严宝辉 348

Java中`Arrays`详解_LY破晓的博客 8-18
一、`Arrays`类的定义 `Arrays`类位于 java.util 包中,主要包含了操纵数组的各种方法 使用时导包:import java.util.`Arrays` 二、`Arrays`常...

`Arrays`的常用方法详解_alankin的博客 8-14
`Arrays`的常用方法详解 sort() 待续... parallelSort() 待续... binarySearch() 待续... equals() 待续... fill() 待续... copyOf() @since 1.6 ...

Array和`Arrays`的区别 pengzhisen123的博客 6815
Array的api如下： public final class Array extends Object The Array class provides static methods to dynamically create and acces...

java `arrays`_Java中`Arrays`详解 weixin_33783283的博客 201
一、`Arrays`类的定义`Arrays`类位于java.util包中，主要包含了操纵数组的各种方法使用时导包:import java.util.`Arrays`二、`Arrays`常用...

数组`Arrays`_四魑的码的博客 8-15
`Arrays`是针对数组的工具类,可以进行 排序,查找,复制填充等功能。 大大提高了开发人员的工作效率。 数组复制 与使用System arra...

`Arrays`的简单使用_@XL的博客 9-1
java.util.`Arrays`简单使用(int) binarySearch(int[] a, int key) a->数组名,key->查找的值 publicclass`ArraysTest`{publicstaticvoidmain(Str...

`Arrays`用法整理 剑凌天下 497
原文 http://mouselearnjava.iteye.com/blog/1985990 本文将整理 java.util.`Arrays` 工具类比较常用的方法： 本文介绍的方法基于JD...

Java Arrys的用法 u010041998的博客 448
java.util.`Arrays`类能方便地操作数组，它提供的所有方法都是静态的。具有以下功能:请看测试代码：package cn.feng.method.arrys...

`Arrays`_我要变成万人迷的博客 7-23
import java.util.`Arrays`; public class TestDemo01 { /* * 数组工具类`Arrays`:是在java.util.`Arrays`包下 * 此包提供了大量的和数组相关你...

`Arrays`总结_征途尔的博客 8-8
`Arrays.sort`(names); 另外,`Arrays`的sort方法也可以结合比较器,完成更加复杂的排序。 Java代码 publicstatic<T>voidsort(T[] a, Com...

Java `Arrays` 用法 assiduous_me的博客 3400
目录 1. 将数组转成集合对象 2. 在数组中二分查找指定元素key 3. 将数组的一部分拷贝出来（ newLength 这个是从被拷贝数组头元...

`Arrays.sort()`的用法 Lin的博客 2607
一、介绍1、sort(T[] a) 的使用2、sort(T[] a,int formIndex, int toIndex) 的使用3、sort(T[] a, Comparator<? supre T> c)的使用补充：...

`Arrays.asList`的用法 菜鸟程序员成长路 4163
使用工具类`Arrays.asList()`把数组转换成集合时，不能使用其修改集合相关的方法，它的add/remove/clear方法会抛出UnsupportOp...

`Arrays`类的常见用法 高数老师的博客 188
`Arrays`类包含用来操作数组（比如排序和搜索）的各种方法。 sort方法，用来对指定数组中的元素进行排序（元素值从小到大进行...

java中数组缩容的方法_Java中Arrys数组常用的方法 最新发布 weixin_29071809的博客 119
Arrys常用方法1.toString()`Arrays.toString`(arr) //打印数组的内容，观察数组元素的值2.sort()`Arrays.sort`(arr); //升序3.copyOf(int[] a1...

Java中的`Arrays`类使用详解 言壘 1万+
首先先创建一个打印数组的方法，方便后面直接使用public static void output(int []a) { for(int i=0;i<a.length;i++) { System.out.printf(...

qq发文件大小上限_微信又出新功能！网友：终于不用转QQ了 weixin_39841136的博客 489
刚上新了表情包的微信，又双叟发出新功能了，这次的功能有点实用。20日，腾讯微信团队宣布，微信现已支持向朋友发送高清视...

JAVA基于`Arrays.sort()`实现数组升序和降序 08.10
主要介绍了JAVA基于`Arrays.sort()`实现数组升序和降



DaydreamerZz

关注

13

3

95



专栏目录