

Arrays 类

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：

- 给数组赋值：通过fill方法。
- 对数组排序：通过sort方法,按升序。
- 比较数组：通过equals方法比较数组中元素值是否相等。
- 查找数组元素：通过binarySearch方法能对排序好的数组进行二分查找法操作。

具体说明请查看下表：

序号	方法和说明
1	public static int binarySearch(Object[] a, Object key) 用二分查找算法在给定数组中搜索给定值的对象(Byte,Int,double等)。... 含在数组中，则返回搜索键的索引；否则返回 (-(插入点) - 1)。
2	public static boolean equals(long[] a, long[] a2) 如果两个指定的 long 型数组彼此相等，则返回 true。如果两个数组包 有相应元素对都是相等的，则认为这两个数组是相等的。换句话说，如 则两个数组是相等的。同样的方法适用于所有的其他基本数据类型 (By
3	public static void fill(int[] a, int val) 将指定的 int 值分配给指定 int 型数组指定范围中的每个元素。同样的 (Byte , short , Int等)。
4	public static void sort(Object[] a) 对指定对象数组根据其元素的自然顺序进行升序排列。同样的方法适用 short , Int等)。

java.util.Arrays类

```
int[] a = {1, 2, 323, 23, 543, 12, 59};
```

打印：

```
System.out.println(Arrays.toString(a));
```

 数组元素的值

Arrays.toString()方法是Arrays类的静态方法，不是前面讲的Object的toString()方法

数组元素的排序：

升序 Arrays.sort(a) ;

二分法查找：

```
Arrays.sort(a);        //使用二分法查找，必须先对数组进行排序;
```

```
Arrays.binarySearch(a, 12); //返回排序后新的索引位置,若未找到返回负数。
```

数组填充:

```
Arrays.fill(a, 2, 4, 100); //将2到4索引的元素替换为100;
```

数组元素是引用类型的排序(Comparable接口的应用)

引用类型实现Comparable接口来排序:

```
import java.util.Arrays;

public class Test {

    public static void main(String[] args) {

        Man[] msMans = { new Man(3, "a"), new Man(60, "b"), new Man(2, "c") };
        Arrays.sort(msMans);
        System.out.println(Arrays.toString(msMans));
    }
}

class Man implements Comparable {

    int age;
    int id;
    String name;

    public Man(int age, String name) {
        super();
        this.age = age;
        this.name = name;
    }

    public String toString() {
        return this.name;
    }

    public int compareTo(Object o) {
        Man man = (Man) o;
        if (this.age < man.age) {
            return -1;
        }
        if (this.age > man.age) {
            return 1;
        }
        return 0;
    }
}
```

