

Bài 8.10: Sắp xếp mảng với lớp Arrays

- ✓ Tổng quan
- ✓ Các phương thức và mô tả
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Tổng quan

- Lớp Arrays là một lớp tiện ích triển khai nhiều phương thức hỗ trợ thao tác trên mảng.
- Ví dụ: sắp xếp, tìm kiếm, so sánh...
- Lớp Arrays nằm trong gói `java.util`.
- Các phương thức của lớp này là các phương thức static nên khi sử dụng ta không cần tạo đối tượng của lớp Arrays.
- Trong nội dung này ta sẽ tập trung vào các phiên bản nạp chồng của các phương thức sắp xếp.

Các phương thức và mô tả

Phương thức	Mô tả
<code>void sort(t[] arr)</code>	Sắp xếp mảng theo thứ tự tăng dần, trong đó t là kiểu nguyên thủy: int, float, double... Phương thức này sử dụng thuật toán sắp xếp trộn để sắp xếp mảng.
<code>void sort(t[] arr, int fromIndex, int toIndex)</code>	Sắp xếp các phần tử mảng trong nửa khoảng [fromIndex, toIndex) theo thứ tự tăng dần, trong đó t là kiểu nguyên thủy: int, float, double... Phương thức này sử dụng thuật toán sắp xếp trộn để sắp xếp mảng. Vẫn ngoại lệ <code>IllegalArgumentException</code> nếu <code>from > to</code> hoặc thứ tự tự nhiên xung đột với chuẩn sắp xếp trong <code>Comparable</code> . Vẫn ngoại lệ <code>ArrayIndexOutOfBoundsException</code> nếu <code>from < 0</code> hoặc <code>to > arr.length</code> .
<code>void sort(Object[] arr)</code>	Sắp xếp mảng các đối tượng theo thứ tự tăng dần. Phương thức này sử dụng thuật toán sắp xếp trộn để sắp xếp mảng. Các đối tượng trong mảng phải triển khai interface <code>Comparable</code> để chúng có thể so sánh với nhau theo cặp. Có thể xảy ra ngoại lệ <code>ClassCastException</code> nếu hai đối tượng trong mảng không tương thích để so sánh.

Các phương thức và mô tả

<pre>void sort(Object[] arr, int fromIndex, int toIndex)</pre>	<p>Sắp xếp các đối tượng trong nửa khoảng [fromIndex, toIndex) của mảng theo thứ tự tăng dần. Phương thức này sử dụng thuật toán sắp xếp trộn để sắp xếp mảng. Các đối tượng trong mảng phải triển khai interface Comparable để chúng có thể so sánh với nhau theo cặp. Có thể xảy ra ngoại lệ ClassCastException nếu hai đối tượng trong mảng không tương thích để so sánh.</p> <p>Văng ngoại lệ IllegalArgumentException nếu from > to hoặc thứ tự tự nhiên xung đột với chuẩn sắp xếp trong Comparable.</p> <p>Văng ngoại lệ ArrayIndexOutOfBoundsException nếu from < 0 hoặc to > arr.length.</p>
<pre><E> void sort(E[] arr, Comparator<? Super E> comparator)</pre>	<p>Sắp xếp mảng các đối tượng theo thứ tự được chỉ định trong comparator. Phương thức này sử dụng thuật toán sắp xếp trộn để sắp xếp mảng. Có thể xảy ra ngoại lệ ClassCastException nếu hai đối tượng trong mảng không tương thích để so sánh.</p>

Các phương thức và mô tả

<code><E> void sort(E[] arr, int fromIndex, int toIndex, Comparator<? Super E> comparator)</code>	<p>Sắp xếp các đối tượng trong nửa khoảng [fromIndex, toIndex) của mảng theo thứ tự được chỉ định trong comparator. Phương thức này sử dụng thuật toán sắp xếp trộn để sắp xếp mảng. Có thể xảy ra ngoại lệ ClassCastException nếu hai đối tượng trong mảng không tương thích để so sánh.</p> <p>Văng ngoại lệ IllegalArgumentException nếu from > to hoặc thứ tự tự nhiên xung đột với chuẩn sắp xếp trong Comparable.</p> <p>Văng ngoại lệ ArrayIndexOutOfBoundsException nếu from < 0 hoặc to > arr.length.</p>
<code>void parallelsort(...)</code>	<p>Phương thức sắp xếp có các biến thể tương tự như sort() đã mô tả ở trên. Phương thức này thực hiện chia mảng gốc thành các mảng con, tiến hành sắp xếp sau đó trộn lại. Quá trình sắp xếp nó gọi phương thức sort() tương ứng.</p>
<code>String deepToString(Object[] inputArr)</code>	<p>Trả về một String đại diện chứa tất cả các phần tử của mảng.</p>

Nội dung tiếp theo

Sắp xếp tập hợp trong Collection Framework