
Like & Share & Subscribe
thuydung ProIT4All



Lớp tiện ích Collections



Fan page: <http://facebook.com/Proit4All>



Lớp tiện ích Collections

❑ Lớp Collections cung cấp: tập các hàm tiện ích mạnh mẽ nhằm hỗ trợ xử lý List

Phương thức	Mô tả
<code>int binarySearch (List list, Object key)</code>	Tìm kiếm nhị phân
<code>void fill (List list, Object obj)</code>	Gán giá trị cho các phần tử
<code>void shuffle (List list)</code>	Hoán vị ngẫu nhiên
<code>void sort (List list)</code>	Sắp xếp tăng dần
<code>void reverse (List list)</code>	Đảo ngược
<code>void rotate (List list, int distance)</code>	Xoay vòng
<code>void swap(List list, int i, int j)</code>	Tráo đổi



Sắp xếp tập đối tượng

- ❑ Để sắp xếp tập các đối tượng cần tiêu chí so sánh các đối tượng.
- ❑ Có 2 cách cung cấp tiêu chí so sánh các đối tượng
 - ❖ Cách 1: Định nghĩa tiêu chí so sánh trong class bằng cách implements interface Comparable sau đó viết mã so sánh 2 đối tượng trong phương thức compareTo(). Cách này ít được sử dụng vì khó thay đổi tiêu chí so sánh.
 - ❖ Cách 2: Tạo đối tượng từ interface Comparator sau đó cung cấp cho phương thức Collections.sort(). Cách này được sử dụng nhiều vì tính linh hoạt về tiêu chí so sánh.



Cách 1: Sắp xếp tập đối tượng

```
public class Student implements Comparable<Student>{  
    public String fullname;  
    public Double marks;  
  
    public Student(String fullname, Double marks) {  
        this.fullname = fullname;  
        this.marks = marks;  
    }  
    @Override  
    public int compareTo(Student other) {  
        return marks.compareTo(other.marks);  
    }  
}
```

```
List<Student> list = new ArrayList<>();  
list.add(new Student("Tuấn", 5.0));  
list.add(new Student("Cường", 7.0));  
list.add(new Student("Phương", 6.0));  
Collections.sort(list);
```



Cách 2: Sắp xếp tập đối tượng

```
public class Student {  
    public String fullname;  
    public Double marks;  
  
    public Student(String fullname, Double marks) {  
        this.fullname = fullname;  
        this.marks = marks;  
    }  
}
```

```
List<Student> list = new ArrayList<>();  
list.add(new Student("Tuấn", 5.0));  
list.add(new Student("Cường", 7.0));  
list.add(new Student("Phương", 6.0));  
  
Comparator<Student> com = new Comparator<Student>() {  
    @Override  
    public int compare(Student o1, Student o2) {  
        return o1.marks.compareTo(o2.marks);  
    }  
};  
Collections.sort(list, com);
```



Like & Share & Subscribe

Fan page: <http://facebook.com/Proit4All>

DEMO

Collections



Java™





Demo

```
public static void main(String[] args) {  
    ArrayList<Integer> a = new ArrayList<>();  
    a.add(2);  
    a.add(5);  
    a.add(7);  
    a.add(1);  
    System.out.println(a.toString());  
    Collections.shuffle(a);  
    System.out.println("List sau khi hoan vi ngau nhien: \n"+a.toString());  
    Collections.sort(a);  
    System.out.println("list sau khi sx tang dan: \n"+a.toString());  
    Collections.reverse(a);  
    System.out.println("list sau khi dao nguoc: \n"+a.toString());  
    Collections.swap(a, 0, 2);  
    System.out.println("List sau khi doi vi tri 0 va 2 \n"+a.toString());  
}
```

```
[2, 5, 7, 1]  
List sau khi hoan vi ngau nhien:  
[2, 7, 1, 5]  
list sau khi sx tang dan:  
[1, 2, 5, 7]  
list sau khi dao nguoc:  
[7, 5, 2, 1]  
List sau khi doi vi tri 0 va 2  
[2, 5, 7, 1]
```

thuydung

ProIT4All

Fan page: <http://facebook.com/Proit4All>



THANK YOU

<http://youtube.com/@AnhNguyenNgoc>



Java™

