

Thuật toán Collection trong Java

Collection Framework định nghĩa một số thuật toán mà có thể áp dụng tới các Collection và Map.

Những thuật toán này được định nghĩa như là các phương thức static bên trong lớp Collection. Một số phương thức có thể ném một **ClassCastException**, xảy ra khi cố gắng so sánh các kiểu không tương thích, hoặc ném một **UnsupportedOperationException**, xảy ra khi cố gắng sửa đổi một *Unmodifiable Collection*.

Để hiểu sâu hơn các khái niệm được trình bày trong chương này, mời bạn tham khảo loạt bài: [Ví dụ về Collection trong Java](#).

Dưới đây liệt kê các phương thức được định nghĩa trong thuật toán của Collection Framework trong Java:

STT	Phương thức và Miêu tả
1	static int binarySearch(List list, Object value, Comparator c) Tìm kiếm value trong list được sắp xếp theo c. Trả về vị trí của value trong list; hoặc trả về -1 nếu value không được tìm thấy
2	static int binarySearch(List list, Object value) Tìm kiếm value trong list. List phải được xếp thứ tự. Trả về vị trí của value trong list; trả về -1 nếu không tìm thấy value
3	static void copy(List list1, List list2) Sao chép các phần tử của list2 vào list1
4	static Enumeration enumeration(Collection c) Trả về một bản liệt kê qua c
5	static void fill(List list, Object obj)

	Gán obj tới mỗi phần tử của list
6	static int indexOfSubList(List list, List subList) Tìm kiếm trong list về sự xuất hiện đầu tiên của subList. Trả về chỉ mục của so khớp đầu tiên; trả về -1 nếu không có so khớp được tìm thấy
7	static int lastIndexOfSubList(List list, List subList) Tìm kiếm trong list về sự xuất hiện cuối cùng của subList. Trả về chỉ mục của so khớp đầu tiên; trả về -1 nếu không có so khớp được tìm thấy
8	static ArrayList list(Enumeration enum) Trả về một ArrayList mà chứa các phần tử của enum
9	static Object max(Collection c, Comparator comp) Trả về phần tử tối đa trong c như được xác định bởi comp
10	static Object max(Collection c) Trả về phần tử tối đa trong c như được xác định bởi thứ tự tự nhiên. Collection không cần xếp theo thứ tự
11	static Object min(Collection c, Comparator comp) Trả về phần tử tối thiểu trong c như được xác định bởi comp. Collection không cần xếp theo thứ tự
12	static Object min(Collection c) Trả về phần tử tối thiểu trong c như được xác định bởi thứ tự tự nhiên
13	static List nCopies(int num, Object obj) Trả về num bản sao của obj được chứa trong một list không đổi. num phải lớn hơn

	hoặc bằng 0
14	static boolean replaceAll(List list, Object old, Object new) Thay thế tất cả old bởi new trong list. Trả về true nếu có ít nhất một sự thay thế xảy ra. Nếu không là false
15	static void reverse(List list) Đảo ngược dãy trong list
16	static Comparator reverseOrder() Trả về một comparator đảo ngược
17	static void rotate(List list, int n) Quay list bởi n vị trí tới bên phải. Để quay sang bên trái, sử dụng một giá trị âm cho n
18	static void shuffle(List list, Random r) Xáo trộn các phần tử trong list bởi sử dụng r như là một nguồn của các số ngẫu nhiên
19	static void shuffle(List list) Xáo trộn các phần tử trong list
20	static Set singleton(Object obj) Trả về obj như là một set không thay đổi. Đây là một cách dễ dàng để biến đổi một đối tượng đơn vào trong một set
21	static List singletonList(Object obj) Trả về obj như là một list không thay đổi. Đây là một cách dễ dàng để biến đổi một đối tượng đơn vào trong một list

22	static Map singletonMap(Object k, Object v) Trả về cặp key/value (k/v) như là một map không thay đổi. Đây là một cách dễ dàng để biến đổi một cặp key/value đơn vào trong một map
23	static void sort(List list, Comparator comp) Xếp thứ tự các phần tử trong list như đã xác định bởi comp
24	static void sort(List list) Xếp thứ tự các phần tử trong list như đã xác định bởi thứ tự tự nhiên
25	static void swap(List list, int idx1, int idx2) Trao đổi các phần tử trong list tại các chỉ mục được xác định bởi idx1 và idx2
26	static Collection synchronizedCollection(Collection c) Trả về một Collection an toàn luồng (thread-safe) bởi c
27	static List synchronizedList(List list) Trả về một thread-safe list bởi list
28	static Map synchronizedMap(Map m) Trả về một thread-safe map bởi m
29	static Set synchronizedSet(Set s) Trả về một thread-safe set bởi s
30	static SortedMap synchronizedSortedMap(SortedMap sm) Trả về một thread-safe sorted set bởi sm
31	static SortedSet synchronizedSortedSet(SortedSet ss)

	Trả về một thread-safe set bởi ss
32	static Collection unmodifiableCollection(Collection c) Trả về một unmodifiable collection bởi c
33	static List unmodifiableList(List list) Trả về một unmodifiable list bởi list
34	static Map unmodifiableMap(Map m) Trả về một unmodifiable map bởi m
35	static Set unmodifiableSet(Set s) Trả về một unmodifiable set bởi s
36	static SortedMap unmodifiableSortedMap(SortedMap sm) Trả về một unmodifiable sorted map bởi sm
37	static SortedSet unmodifiableSortedSet(SortedSet ss) Trả về một unmodifiable sorted set bởi ss

Ví dụ

Ví dụ sau minh họa các thuật toán đa dạng của Collection trong Java:

```
import java.util.*;  
  
public class AlgorithmsDemo {  
  
    public static void main(String args[]) {  
        // Create and initialize linked list  
        LinkedList l1 = new LinkedList();  
        l1.add(new Integer(-8));  
    }  
}
```

```
l1.add(new Integer(20));
l1.add(new Integer(-20));
l1.add(new Integer(8));

// Create a reverse order comparator
Comparator r = Collections.reverseOrder();
// Sort list by using the comparator
Collections.sort(l1, r);
// Get iterator
Iterator li = l1.iterator();
System.out.print("List sorted in reverse: ");
while(li.hasNext()){
    System.out.print(li.next() + " ");
}
System.out.println();
Collections.shuffle(l1);
// display randomized list
li = l1.iterator();
System.out.print("List shuffled: ");
while(li.hasNext()){
    System.out.print(li.next() + " ");
}
System.out.println();
System.out.println("Minimum: " + Collections.min(l1));
System.out.println("Maximum: " + Collections.max(l1));
}
}
```

Nó sẽ cho kết quả sau:

```
List sorted in reverse: 20 8 -8 -20
List shuffled: 20 -20 8 -8
Minimum: -20
Maximum: 20
```

