HVIT CLAN JSC

Java Collection

Phần 2 : Danh sách – Set trong Java

Thực hiện slide: Nguyễn Đức Toàn

Thuyết minh: Nguyễn Đức Toàn

HVIT CLAN 2021

NỘI DUNG CHÍNH

Giới thiệu

ArrayList

LinkedList

Set

Phần 1

Giới thiệu về danh sách

Giới thiệu về danh sách

List (danh sách) là gì?

List là một interface trong java. Nó chứa các phương thức để chèn và xóa các phần tử dựa trên chỉ số index.

```
public interface List<E> extends Collection<E>
```

Danh sách và mảng

Mảng (Array) là một cấu trúc dữ liệu có kích thước cố định, trong khi **List** là một lớp Collection có thể thay đổi được kích thước. Nghĩa là chúng ta không thể thay đổi kích thước của mảng khi đã tạo, nhưng ArrayList có thể được thay đổi.

List không thể lưu giữ dữ liệu nguyên thủy, nó chỉ có thể chứa các đối tượng. Nhưng **Array** có thể chứa *cả hai* kiểu dữ liệu nguyên thủy và các đối tượng trong Java. Kể từ Java 5, kiểu nguyên thủy được tự động chuyển đổi trong các đối tượng được gọi là auto-boxing.

Phần 2

ArrayList trong Java

ArrayList trong Java

Lớp ArrayList là gì?

Lớp ArrayList trong java là một lớp kế thừa lớp *AbstractList* và triển khai của *List Interface* trong Collections Framework nên nó sẽ có một vài đặc điểm và phương thức tương đồng với List. **ArrayList được sử dụng như một mảng động để lưu trữ các phần tử**.

Đặc điểm của Lớp ArrayList

- Lớp ArrayList trong java có thể chứa các phần tử trùng lặp.
- Lớp ArrayList duy trì thứ tự của phần tử được thêm vào.
- Lớp ArrayList là không đồng bộ (non-synchronized).
- Lớp ArrayList cho phép truy cập ngẫu nhiên vì nó lưu dữ liệu theo chỉ mục.
- Lớp ArrayList trong java, thao tác chậm vì cần nhiều sự dịch chuyển nếu bất kỳ phần tử nào bị xoá khỏi danh sách.

Khởi tạo ArrayList

Hàm khởi tạo

ArrayList()	Nó được sử dụng để khởi tạo một danh sách mảng trống.
ArrayList(Collection c)	Nó được sử dụng để xây dựng một danh sách mảng được khởi tạo với các phần tử của collection c.
ArrayList(int capacity)	Nó được sử dụng để xây dựng một danh sách mảng mà có dung lượng ban đầu được chỉ định.

Cú pháp

ArrayList <Kiểu Dữ Liệu> ten_danh_sach = new **ArrayList<Kiểu Dữ Liệu>()**;

VD:

```
//Tạo 1 danh sách trống
ArrayList lst= new ArrayList<>();

//Tạo 1 danh sách CHUỖI
ArrayList<String> lst1 = new ArrayList<String>();

//Tạo 1 danh sách CHUỖI có 20 phần tử
ArrayList<String> lst2 = new ArrayList<String>(20);
```

Phương thức trong ArrayList

Phương thức	Cách dùng
add(Object o)	Nối thêm phần tử được chỉ định vào cuối một danh sách.
add(int index, Object element)	Chèn phần tử element tại vị trí index vào danh sách.
get(int index)	Lấy ra phần tử ở vị trí truyền vào
indexOf(Object o)	Trả về chỉ mục trong danh sách với sự xuất hiện đầu tiên của phần tử được chỉ định, hoặc -1 nếu danh sách không chứa phần tử này.
lastIndexOf(Object o)	Nó được sử dụng để trả về chỉ mục trong danh sách với sự xuất hiện cuối cùng của phần tử được chỉ định, hoặc -1 nếu danh sách không chứa phần tử này.
contains(element)	Trả về là true nếu tìm thấy element trong danh sách, ngược lại trả về false.
retainAll(Collection c)	Xóa những phần tử không thuộc collection c ra khỏi danh sách.
removeAll(Collection c)	Xóa những phần tử thuộc collection c ra khỏi danh sách.
clone()	Tạo một bản sao của ArrayList.
clear()	Xóa tất cả các phần tử từ danh sách.

Phần 3

LinkedList trong Java

LinkedList trong Java

Lớp LinkedList là gì?

Lớp LinkedList trong java là một lớp kế thừa lớp AbstractSequentialList và triển khai của List, Queue Interface trong Collections Framework nên nó sẽ có một vài đặc điểm và phương thức tương đồng với List, Queue. Lớp LinkedList trong java sử dụng cấu trúc danh sách liên kết kép Doubly để lưu trữ các phần tử.

Đặc điểm của Lớp LinkedList

- Lớp LinkedList trong java có thể chứa các phần tử trùng lặp.
- Lớp LinkedList duy trì thứ tự của phần tử được thêm vào.
- Lớp LinkedList là không đồng bộ (non-synchronized).
- Trong java lớp LinkList, thao tác nhanh vì không cần phải dịch chuyển nếu bất kỳ phần tử nào bị xoá khỏi danh sách.
- Lớp LinkedList trong java có thể được sử dụng như list (danh sách), stack (ngăn xếp) hoặc queue (hàng đợi).

Khởi tạo LinkedList

Hàm khởi tạo

LinkedList()	Nó được sử dụng để khởi tạo một danh sách trống.
LinkedList(Collection c)	Nó được sử dụng để xây dựng một danh sách chứa các phần tử của collection được chỉ định, theo thứ tự chúng được trả về bởi iterator của collection.
LinkedList(int capacity)	Nó được sử dụng để xây dựng một danh sách mà có dung lượng ban đầu được chỉ định.

Cú pháp

LinkedList <Kiểu Dữ Liệu> ten_danh_sach = new **LinkedList<Kiểu Dữ Liệu>()**;

VD:

```
//Tạo 1 danh sách trống
LinkedList lst = new LinkedList<>();

//Tạo 1 danh sách CHUỖI
LinkedList<String> lst1 = new LinkedList<String>();

//Tạo 1 danh sách CHUỖI có 10 phần tử
LinkedList<String> lst2 = new LinkedList<String>();
```

Phương thức trong LinkedList

Phương thức	Cách dùng
add(Object o)	Nối thêm phần tử được chỉ định vào cuối một danh sách.
add(int index, Object element)	Chèn phần tử element tại vị trí index vào danh sách.
addFirst(Object o) / addLast(Object o)	Nó được sử dụng để chèn phần tử được chỉ định vào đầu - cuối danh sách
get(int index)	Lấy ra phần tử ở vị trí truyền vào
getFirst() / getLast()	Nó được sử dụng để trả về phần tử đầu tiên – cuối cùng trong một danh sách.
indexOf(Object o)	Trả về chỉ mục trong danh sách với sự xuất hiện đầu tiên của phần tử được chỉ định, hoặc -1 nếu danh sách không chứa phần tử này.
lastIndexOf(Object o)	Nó được sử dụng để trả về chỉ mục trong danh sách với sự xuất hiện cuối cùng của phần tử được chỉ định, hoặc -1 nếu danh sách không chứa phần tử này.
contains(element)	Trả về là true nếu tìm thấy element trong danh sách, ngược lại trả về false.
removeAll(Collection c)	Xóa những phần tử thuộc collection c ra khỏi danh sách.
size()	Nó được sử dụng để trả lại số lượng các phần tử trong một danh sách

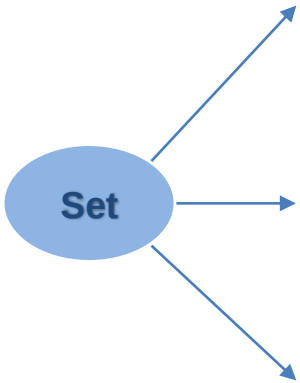
Phần 4

Set trong Java

Set trong Java

Set là gì?

- Set là một interface kế thừa Collection interface trong java.
- Set trong java là một Collection **không thể chứa các phần tử trùng lặp**.



HashSet lưu trữ các phần tử của nó trong bảng băm, là cách thực hiện tốt nhất, tuy nhiên nó ***không*** *đảm bảo* về thứ tự các phần tử được chèn vào.

TreeSet lưu trữ các phần tử của nó trong một cây, sắp xếp các phần tử của nó dựa trên các giá trị của chúng, về cơ bản là **chậm hơn** HashSet.

LinkedHashSet được triển khai dưới dạng bảng băm với có cấu trúc dữ liệu danh sách liên kết, sắp xếp các phần tử của nó dựa trên thứ tự chúng được chèn vào tập hợp (thứ tự chèn).

Khởi tạo Set

Cú pháp

Set <Kiểu dữ liệu > set_1 = **new** HashSet <Kiểu dữ liệu > ();

Set <Kiểu dữ liệu > set_2 = **new** LinkedHashSet <Kiểu dữ liệu > ();

Set <Kiểu dữ liệu > set_3 = **new** TreeSet <Kiểu dữ liệu > ();

VD:

```
Set<Integer> set = new LinkedHashSet<Integer>();
set.add(2);
set.add(5);
set.add(3);

System.out.println(set);
```

Phương thức trong Set

Phương thức	Cách dùng
add(Object o)	Nối thêm phần tử được chỉ định vào cuối một set.
addAll(Collection c)	Nó được sử dụng để chèn tất cả các phần tử của c vào set
clear()	Xóa tất cả các phần tử khỏi set
equals(Object o)	So sánh các đối tượng được chỉ định với set
remove(Object o)	Xóa phần tử đã chỉ định khỏi set.
removeAll(Collection c)	Xóa khỏi set tất cả các phần tử của nó được chứa trong collection c đã chỉ định.
retainAll(Collection c)	Chỉ giữ lại các phần tử trong set được chứa trong collection c đã chỉ định.
contains(element)	Trả về là true nếu tìm thấy element trong set, ngược lại trả về false.
Object[] toArray()	Trả về một mảng chứa tất cả các phần tử trong set.
size()	Nó được sử dụng để trả lại số lượng các phần tử trong một danh sách
isEmpty()	Trả về true nếu set không chứa bất kì phần tử nào. Ngược lại trả về false

Kết luận

Array	List	Set
Kích thước **cố định**, không thể thay đổi được	Kích thước **không cố định**, có thể thay đổi được	Kích thước **không cố định**, có thể thay đổi được
Có thể lưu trữ dữ liệu kiểu **nguyên thủy** và **đối tượng**.	Chỉ có thể lưu trữ dữ liệu kiểu **đối tượng.** Kiểu nguyên thủy được tự động chuyển đổi trong các đối tượng được gọi là **auto-boxing**.	Chỉ có thể lưu trữ dữ liệu kiểu **đối tượng**
Tốc độ lưu trữ và thao tác **nhanh**.	Tốc độ lưu trữ vào thao tác **chậm hơn Array**.	Tốc độ lưu trữ vào thao tác **chậm hơn Array**.
Có thể lưu trữ các phần tử trùng lặp	Có thể lưu trữ các phần tử trùng lặp	Không thể lưu trữ các phần tử trùng lặp

TRÂN TRỌNG
CẢM ƠN