CÂU HỎI PHỎNG VẤN COLLECTION FRAMEWORK JAVA CORE LẬP TRÌNH JAVA

LinkedList trong Java có gì hay? (LinkedList in Java)

IT Phú Trần 13 Tháng Ba 2018

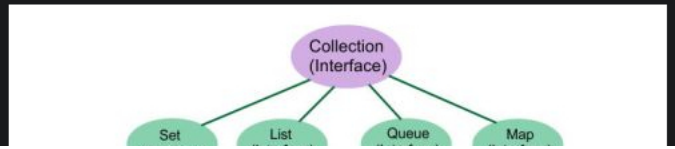
cách sử dụng collection câu hỏi phỏng vấn về collection collection là gì collection trong java java core linkedlist linkedlist trong java linkedlist trong java là gì phân biệt arraylist và linkedlist trong java

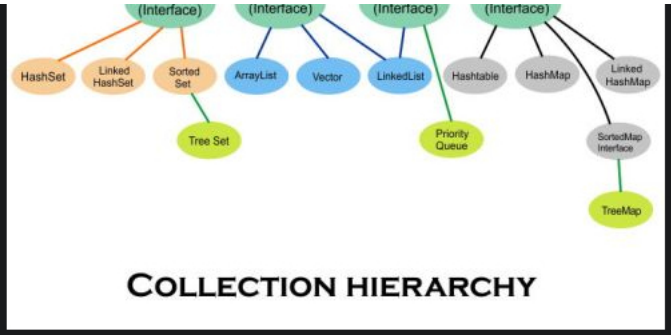
Trong **Java**, một trong những **collection** đáng được nhắc đến để sử dụng trong những chương trình không ai khác đó chính là LinkedList.Bài viết hôm nay mang hương vị chia sẻ về những điều cần biết **LinkedList trong java**. Đây cũng là một trong những kiến thức thường được hỏi trong những kỳ **phỏng vấn** và có hiệu suất khá cao nếu bạn là một lập trình viên có kinh nghiệm. Vậy **LinkedList trong java là gì?** Quan trọng nhất là khi nào dùng LinkedList? Và **dùng LinkedList như thế nào**?

LinkedList là gì?

LinkedList implementation của List interface, vì vậy LinkedList mang đặc điểm chung của List. Và tất nhiên nó sẽ có những phương thức của List.

Các bạn có thể nhìn thấy mô hình tổng quan về collection:





Mô hình tổng quan về collection framework

Hiểu về LinkedList như thế nào cho dễ?

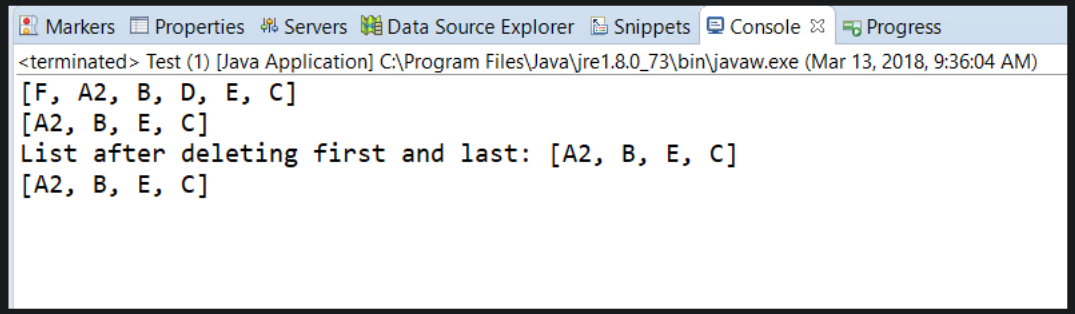
Lý thuyết là như vậy, nhưng chúng ta hiểu đơn giản về LinkedList như thế nào? Và cần lưu ý cái gì?

Thứ nhất, chúng ta cần phải nhớ vì LinkedList kế thừa từ interface Litst nên nó là một danh sách mà số phần tử có thể thay đổi, không bị giới hạn như mảng. Về cách sử dụng hoàn toàn tương tự như ArrayList mà tôi đã có một bài viết hướng dẫn sử dụng, bạn có thể đọc nó nếu chưa biết gì về ArrayList. Dưới đây là ví dụ:

```
1  package com.itphutran;
2
3  import java.util.LinkedList;
4
5  public class DemoLinkedList {
6
7      public static void main(String[] args) {
8   // Tạo một đối tượng LinkedList.
9   List<String> list = new LinkedList<String>();
10
11  // Thêm một số phần tử vào danh sách.
12  list.add("F");
13  list.add("B");
14  list.add("D");
15  list.add("E");
16  list.add("C");
17
18  // Trên một phần tử vào vị trí có chỉ số 1.
19  list.add(1, "A2");
20
21  // Ghi ra tất cả các phần tử của danh sách:
22  System.out.println(list);
23
24  // Loại bỏ một phần tử khỏi danh sách
25  list.remove("F");
26
27  // Loại bỏ phần tử tại vị trí có chỉ số 2.
28  list.remove(2);
29
30  // In ra danh sách sau khi đã xóa 2 phần tử.
31  System.out.println(list);
32  // In ra danh sách sau khi đã xóa.
33  System.out.println("List after deleting first and last: " + list);
34
35  // Lấy ra phần tử tại chỉ số 2.
36  String str = list.get(2);
37
38  // Sét đặt lại phần tử tại vị trí có chỉ số 2.
39  list.set(2, (String) str );
40  System.out.println(list);
41  }
```

```
42
43 }
```

Kết quả :

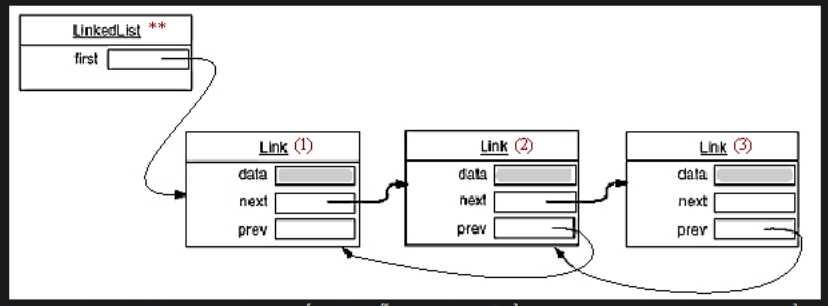


TOP

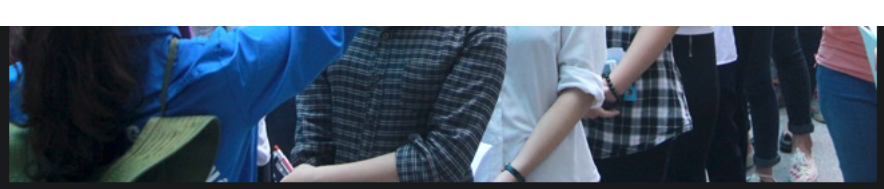
Tiếp theo, nhìn sơ đồ tổng quan ở đầu bài. Cac bạn thấy LinkedList ngoài implement interface là List thì nó còn implement Queue. Do đó nó có khác một vài cách sử dụng cần lưu ý so với ArrayList.

LinkedList có double Linked List. Vì LinkedList trong Java là double Linked nên khi mình nhắc đến Linked List các bạn cứ hiểu là double LinkedList.

Cấu trúc dữ liệu của LinkedList bao gồm các node, tuy nhiên cần lưu ý ở đây là LinkedList chỉ lưu địa chỉ của node đầu tiên (head) và node cuối cùng (tail) mà thôi. (Hình ảnh minh họa bên dưới)



Dựa vào hình minh họa các bạn thấy, với mỗi node nó chỉ cần quan tâm tới node trước và đằng sau nó thôi, và chả quan tâm đến bố con thằng nào nữa. Chúng ta có thể hình dùng và hiểu đơn giản là: "Trong khi xếp hàng điểm danh, chúng ta chỉ cần quan tâm tới thằng trước ta và thằng sau mình là thằng nào, và tương tự cho các thằng khác cũng vậy".



Cần chú ý các node này sẽ chiếm giữ các ô nhớ không liên tục trên bộ nhớ, khác với Array và ArrayList sẽ chiếm giữ các ô nhớ liên tục. Chính vì điều này mà LinkedList sẽ quyết định đến tốc độ khi thao tác trên LinkedList.Cụ thể là như sau:

Khi chúng ta thêm vào LinkedList (INSERT)

Để một phần tử vào LinkedList chúng ta sử dụng bằng lệnh add(), phần tử đó sẽ được thêm vào sau tail, con trỏ trỏ đến tail chỉ cần trỏ đến phần tử được thêm vào trước và sau nó. Tương tự khi thêm phần tử vào head. Khi thêm phần tử vào vị trí khác head và tail, LinkedList cũng chỉ việc thay đổi các con trỏ kề trước và sau. Do đó độ phức tạp là **O(1)**

Liên hệ với ví dụ về trường hợp điểm danh, khi thêm bất kỳ một phần tử nào vào trong linkedlist, nó chỉ cần cập nhật thông tin trước và sau nó.

Khi xóa (Delete)

Nó hoàn toàn tương tự khi chúng ta insert.

Hai người đứng gần người này phải cập nhập lại thông tin người đứng trước, đứng sau họ là ai. Ok? Như vậy độ phức tạp của nó là O(1)

Khi lấy phần tử (Get element)

Do đặc điểm LinkedList chỉ lưu giá trị của 2 node head và tail, do đó để tìm một phần tử trong LinkedList, các bạn lưu ý nó có một số trường hợp sau:

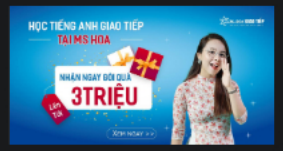
1. **TH1**: nó lấy phần tử đầu và phần tử cuối thì về tốc độ xử lý chẳng thua gì ArrayList.
2. **TH2**: Nếu nó lấy bất kỳ phần tử nào trong danh sách, chẳng hạn phần tử ở giữa hay sau giữa, trước giữa phần tử trong danh sách thì tùy thuộc vào phần tử nó đang nằm ở vị trí nào để quyết định đến thời gian xử lý. Và tất nhiên nó phải duyệt danh sách để tìm đến phần tử cần lấy ra. Do đó độ phức tạp của nó là O(n)

Bài viết mang quan điểm chia sẻ về LinkedList, theo kinh nghiệm của mình và thực tế. Khi dùng ta ít và không thay đổi danh sách, thì chúng ta nên sử dụng ArrayList, nhưng nếu chúng ta thường xuyên thay đổi danh sách của mảng, cụ thể là trường hợp, add, delete số lượng lớn thì nên sử dụng LinkedList, tốc độ mình đảm bảo một điều là sẻ nhanh hơn nhiều khi chúng ta sử dụng ArrayList! Vì mỗi lần ArrayList thực hiện việc thêm, hay xóa các phần tử nó phải cập nhật lại tất cả chỉ số cho nên việc nó tốn bộ nhớ và quá trình cập nhật mất thời gian. Với LinkedList nó chỉ quan tâm đằng trước và đăng sau nó, nghĩa là cập nhật lại thông tin phần tử trước và sau nó.

Sự khác nhau giữa ArrayList và LinkedList là gì?

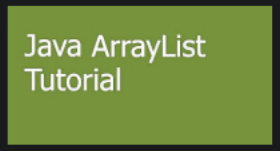
NO.	ARRAYLIST	LINKEDLIST
1)	ArrayList sử dụng một mảng động.	LinkedList sử dụng danh sách liên kết doubly.
2)	ArrayList không hiệu quả với thao tác vì cần nhiều chuyển đổi.	LinkedList là hiệu quả cho thao tác.
3)	ArrayList là tốt hơn để lưu trữ và lấy dữ liệu.	LinkedList là tốt hơn để thao tác dữ liệu.

Các bài viết cùng chủ đề:



Học Tiếng Anh Giao Tiếp

QC Ms Hoa Giao tiếp



Ví dụ và cách sử dụng ArrayList trong Java

itphutran.com

Hướng dẫn tạo menu trong Java

itphutran.com

JAVA CORE

itphutran.com

HỌC JAVA WEB VỚI JSP/SERVLET

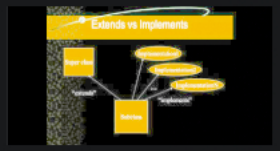
itphutran.com

nhập mảng 2 chiều trong java Archives

itphutran.com

Non static inner class trong java - Page 2 of 14

itphutran.com



Phân biệt extends và implements trong java

itphutran.com

BẢN QUYỀN BÀI VIẾT BỞI DMCA

DMCA

LIÊN KẾT WEBSITE CÙNG CHỦ ĐỀ

Vinaenter EDU

Kênh Lập Trình

Team Việt Dev

Hướng dẫn lập trình Java

GP Coder

Java study and share

VỀ BẢN QUYỀN BÀI VIẾT

Bài viết dựa trên kinh nghiệm cũng như mong muốn chia sẻ kiến thức lập trình mong đóng góp một phần nào đó để xây dựng cộng đồng **Lập Trình Việt Nam**, nên mọi bài viết nếu được chia sẻ, hay copy đều phải được:

✅ Phải được trích dẫn nguồn

✅ Không sử dụng vào mục đích thương mại

✅ Không được sửa đổi hay cố ý thay đổi nội dung của bài viết.

NGƯỜI DÙNG ĐANG ONLINE

3