ArrayList trong Java

Đăng bởi: VanNguyen - Vào ngày: 31-10-2017 - View: 10358

Trong bài này, tôi sẽ hướng dẫn đến các bạn tìm hiểu loại Class Collection tiếp theo - đó là **ArrayList trong Java**. Nội dung của bài này sẽ mô tả đặc điểm, các phương thức thường dùng của Collection này. Với mỗi phương thức được liệt kê, tôi sẽ đưa ra ví dụ đơn giản để cho các bạn nắm bắt được. Cuối bài này tôi sẽ đưa ra một số bài tập cho các bạn luyện tập!

Nội dung chính

» 1. Đặc điểm
» 2. Các phương thức phổ biến
- Tạo mới một ArrayList
- Hiển thị các phần tử có trong ArrayList
- Thêm phần tử vào trong ArrayList
- Truy cập phần tử
- Cập nhật giá trị của phần tử
- Xóa phần tử
- Tìm kiếm một phần tử
- Chuyển ArrayList sang Array (mảng)

» 3. Lời kết

1. Đặc điểm

VÉ 0 ĐỒNG
Đặt Vé 0đ Từ Hôm Nay đến Hết 17/3/2019. Đăng Nhập Thành Viên Để Book Vé...
QC AirAsia
Đặt ngay

ArrayList là một lớp triển khai của List Interface trong Collections Framework nên nó sẽ có một vài đặc điểm và phương thức tương đồng với List . Như đã nói trong bài Tổng quan về Collection trong Java,

DANH SÁCH BÀI HỌC

TỔNG QUAN
JAVA core
Java là gì (1)
Cài đặt Java (2)
Cài đặt Eclipse (3)
Chương trình Hello World (4)
Cú pháp và quy tắc cơ bản (5)
Các kiểu dữ liệu (6)
Biến và hằng số (7)
Toán tử (8)

CẤU TRÚC ĐIỀU KHIỂN
Cấu trúc điều khiển if - else (9)
Cấu trúc rẽ nhánh switch - case. (10)

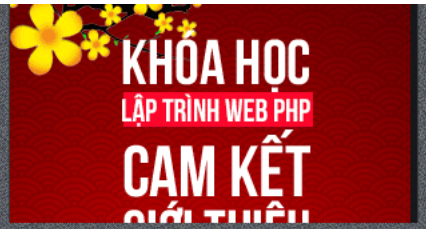
VÒNG LẶP

DANH SÁCH CHỦ ĐỀ
JAVA core
Câu hỏi phỏng vấn JAVA

ĐANG HOT!
» Giảm giá 40% tại học viện UNICA
» Mã giảm giá 20% HostingViet

ArrayList là một Collection kiểu danh sách sử dụng cấu trúc mảng để lưu trữ phần tử. Thứ tự các phần tử dựa theo thứ tự lúc thêm vào và giá trị của các phần tử này có thể trùng nhau.

Như vậy, cả ArrayList và LinkedList đều là lớp triển khai của List Interface. Vậy khi nào chúng ta sẽ sử dụng ArrayList và khi nào chúng ta sẽ sử dụng LinkedList ? Chúng ta sẽ sử dụng ArrayList khi ứng dụng của chúng ta cần truy xuất phần tử nhiều hơn cập nhật và xóa phần tử và chúng ta sẽ sử dụng LinkedList khi ứng dụng của chúng ta cần cập nhật và xóa phần tử nhiều hơn là truy cập phần tử.



2. Các phương thức phổ biến

Tạo mới một ArrayList

Để khai báo một ArrayList , chúng ta cần phải import gói thư viện java.util.ArrayList của Java. Cú pháp import như sau:

Cú pháp

```
// Khai báo ArrayList
// thì import gói thư viện java.util.ArrayList
import java.util.ArrayList;
public class TênClass {
    // ...
}
```

Sau đây là ví dụ cách tạo mới một ArrayList trong Java:

Ví dụ

```
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListString
    // có kiểu là String
    ArrayList<String> arrListString = new ArrayList<>();
}
```

Ngoài ra, nếu chúng ta đã biết trước số lượng phần tử thì chúng ta có thể khai báo kèm với số lượng phần tử của nó. Ví dụ dưới đây sẽ khai báo một ArrayList có tên là arrListName , kiểu là String và có 10 phần tử:

Ví dụ

```
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListName
    // có kiểu là String và có 10 phần tử
    ArrayList<String> arrListName = new ArrayList<>(10);
}
```

Hiển thị các phần tử có trong ArrayList

Để hiển thị các phần tử có trong `ArrayList`, chúng ta có các cách như sau:

Hiển thị theo tên của ArrayList.

Ví dụ

```
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListDouble
    // có kiểu là Double và có 10 phần tử
    ArrayList<Double> arrListDouble = new ArrayList<>(10);

    arrListDouble.add(8.329d);
    arrListDouble.add(1.02d);
    arrListDouble.add(2.9d);
    arrListDouble.add(20.17d);

    System.out.println("Các phần tử có trong arrListDouble là: ");
    System.out.println(arrListDouble);
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 29, 2017, 9:19:33 AM)
Các phần tử có trong arrListDouble là:
[8.329, 1.02, 2.9, 20.17]
```

freetuts.net

Sử dụng vòng lặp `for` thông thường.

Ví dụ

```
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListString
    // có kiểu là String
    ArrayList<String> arrListString = new ArrayList<>();

    // thêm các phần tử sử dụng phương thức add()
    arrListString.add("JAVA");
    arrListString.add("PHP");
    arrListString.add("C#");
    arrListString.add("C++");

```

```java
    // duyệt theo kích thước của arrListString
    // sử dụng vòng lặp for thông thường
    // phương thức arrListString.size() sẽ trả về số phần tử của arrListString
    // và sau đó lấy phần tử tại vị trí thứ i thông qua hàm get()
    // sau đó hiển thị giá trị phần tử đó ra
    System.out.println("Các phần tử có trong arrListString là: ");
    for (int i = 0; i < arrListString.size(); i++) {
        System.out.print(arrListString.get(i) + "\t");
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems | @ Javadoc | Declaration | Coverage | Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 28, 2017, 7:17:26 PM)
Các phần tử có trong arrListString là:
JAVA    PHP    C#    C++
```

freetuts.net

Sử dụng vòng lặp `for` cải tiến duyệt theo đối tượng trong ArrayList.

Ví dụ

```java
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListString
    // có kiểu là String
    ArrayList<Integer> arrListInteger = new ArrayList<>();

    // thêm các phần tử sử dụng phương thức add()
    arrListInteger.add(0);
    arrListInteger.add(7);
    arrListInteger.add(1);
    arrListInteger.add(9);

    // duyệt theo kích thước của arrListInteger
    // sử dụng vòng lặp for duyệt theo đối tượng
    // trong đó kiểu dữ liệu của biến number
    // phải trùng với kiểu dữ liệu của arrListInteger
    System.out.println("Các phần tử có trong arrListInteger là: ");
    for (int number : arrListInteger) {
        System.out.print(number + "\t");
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 28, 2017, 7:21:33 PM)
Các phần tử có trong arrListInteger là:
0       7       1       9
```

freetuts.net

Sử dụng Iterator.

Để sử dụng được `Iterator` chúng ta cần phải import gói thư viện `java.util.Iterator` của Java

Ví dụ

```java
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListString
    // có kiểu là String
    ArrayList<Float> arrListFloat = new ArrayList<>();

    // thêm các phần tử sử dụng phương thức add()
    // chữ f cho biết các số thêm vào là số thực
    // nếu không có chữ này thì trình biên dịch sẽ hiểu đây là số double
    // và sẽ báo lỗi dữ liệu
    arrListFloat.add(0.7f);
    arrListFloat.add(7.26f);
    arrListFloat.add(1.02f);
    arrListFloat.add(9.3f);

    // khai báo một Iterator có kiểu là Float
    Iterator<Float> iterator = arrListFloat.iterator();

    System.out.println("Các phần tử có trong arrListFloat là: ");
    while (iterator.hasNext()) {
        System.out.print(iterator.next() + "\t");
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 28, 2017, 7:36:54 PM)
Các phần tử có trong arrListFloat là:
0.7     7.26    1.02    9.3
```

freetuts.net

Sử dụng ListIterator.

Vì `ArrayList` là một lớp triển khai của List Interface nên nó cũng có thể sử dụng `ListIterator` để duyệt qua các phần tử của nó. Để sử dụng được `ListIterator` chúng ta cần phải import gói thư viện `java.util.ListIterator` của Java.

Ví dụ

```java
public static void main(String[] args) {
    // khai báo 1 ArrayList có tên là arrListChar
    // có kiểu là Character
    ArrayList<Character> arrListChar = new ArrayList<>();

    arrListChar.add('a');
    arrListChar.add('e');
    arrListChar.add('b');
    arrListChar.add('c');

    // khai báo một ListIterator có kiểu là Character
    ListIterator<Character> listIterator = arrListChar.listIterator();

    // hiển thị các phần tử có trong linkedList
    // bằng cách sử dụng ListIterator
    System.out.println("Các phần tử có trong arrListChar là: ");
    while (listIterator.hasNext()) {
        System.out.print(listIterator.next() + "\t");
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 28, 2017, 11:21:18 PM)
Các phần tử có trong arrListChar là:
a       e       b       c
```

freetuts.net

Thêm phần tử vào trong ArrayList

Thêm phần tử sử dụng phương thức add().

Ví dụ

```java
public static void main(String[] args) {
    ArrayList<String> arrListMonth = new ArrayList<>(12);

    // thêm phần tử vào trong ArrayList
    // sử dụng phương thức add()
    arrListMonth.add("January");
    arrListMonth.add("February");
    arrListMonth.add("March");
    arrListMonth.add("May");
    arrListMonth.add("June");
    arrListMonth.add("July");
    arrListMonth.add("August");
    arrListMonth.add("September");
    arrListMonth.add("October");
    arrListMonth.add("November");
    arrListMonth.add("December");

    // thêm vào trong ArrayList
    // phần tử ở vị trí thứ 3 và có giá trị là "April"
    arrListMonth.add(3, "April");

    System.out.println("Các phần tử có trong arrListMonth là: ");
    for (String month : arrListMonth) {
        System.out.println(month);
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 29, 2017, 9:57:02 AM)
Các phần tử có trong arrListMonth là:
January
February
March
April
May
June
July
August
September
October
November
December
```

freetuts.net

Bên cạnh đó, chúng ta có thể sử dụng phương thức `addAll()` để thêm tất cả các phần tử của một `ArrayList` khác vào 1 vị trí bất kỳ trong `ArrayList` đã tồn tại. Ví dụ dưới đây sẽ thêm các phần tử của `arrList1` vào vị trí số 3 trong `arrList2` :

Ví dụ

```java
public static void main(String[] args) {
    ArrayList<Integer> arrList1 = new ArrayList<>();

    // thêm phần tử vào trong arrList1
    arrList1.add(6);
    arrList1.add(4);
    arrList1.add(9);

    ArrayList<Integer> arrList2 = new ArrayList<>();

    // thêm phần tử vào trong arrList2
    arrList2.add(11);
    arrList2.add(15);
    arrList2.add(16);
    arrList2.add(19);

    // thêm các phần tử của arrList1
    // vào vị trí số 3 của arrList2
    arrList2.addAll(3, arrList1);

    System.out.println("Các phần tử có trong arrList2 là: ");
    for (int number : arrList2) {
        System.out.println(number);
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 29, 2017, 10:31:02 AM)
Các phần tử có trong arrList2 là:
11
15
16
6
4
9
19
```

freetuts.net

Truy cập phần tử

Java cung cấp cho chúng ta phương thức `get()` để truy cập đến 1 phần tử bất kỳ trong `ArrayList` thông qua chỉ số của phần tử đó. Ví dụ dưới đây sẽ truy cập phần tử có chỉ số là `2` trong 1 `ArrayList` có tên là `arrListChar` :

Ví dụ

```java
public static void main(String[] args) {
    ArrayList<Character> arrListChar = new ArrayList<>();
```

```java
    // thêm phần tử vào trong arrListChar
    arrListChar.add('l');
    arrListChar.add('a');
    arrListChar.add('b');

    // truy cập phần tử có chỉ số 2 trong arrListChar
    // tương ứng với ký tự 'b'
    // vì arrListChar có kiểu là Character
    // nên các phần tử con của nó cũng có kiểu dữ liệu là Character.
    char ch = arrListChar.get(2);
    System.out.print("Phần tử có chỉ số 2 trong arrListChar là " + ch);
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 29, 2017, 10:52:22 AM)
Phần tử có chỉ số 2 trong arrListChar là b
```

freetuts.net

Cập nhật giá trị của phần tử

Để cập nhật giá trị của phần tử trong `ArrayList`, Java cung cấp cho chúng ta phương thức `set(index, element)`, trong đó `index` là chỉ số của phần tử cần cập nhật và `element` là phần tử mới để thay thế. Để hiểu kỹ hơn về phương thức này các bạn hãy xem ví dụ sau:

Ví dụ

```java
public static void main(String[] args) {
    ArrayList<Integer> arrListInteger = new ArrayList<>();

    // thêm phần tử vào trong arrListInteger
    arrListInteger.add(4);
    arrListInteger.add(8);
    arrListInteger.add(12);
    arrListInteger.add(16);
    arrListInteger.add(20);
    arrListInteger.add(24);

    System.out.println("Các phần tử của arrListInteger ban đầu: ");
    System.out.println(arrListInteger);

    // cập nhật giá trị của phần tử thứ 4
    // trong arrListInteger bằng phương thức set()
```

```
17        arrListInteger.set(4, 204);
18
19        System.out.println("Các phần tử của arrListInteger sau khi cập nhật: ");
20        System.out.println(arrListInteger);
21    }
```

Các bạn thấy trong ví dụ trên, tôi đã tiến hành thay thế giá trị của phần tử có chỉ số là `4` trong danh sách. Kết quả là chương trình sẽ thay thế giá trị 20 của phần tử này thành 204. Kết quả sau khi biên dịch chương trình như sau:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 11:24:17 AM)
Các phần tử của arrListInteger ban đầu:
[4, 8, 12, 16, 20, 24]
Các phần tử của arrListInteger sau khi cập nhật:
[4, 8, 12, 16, 204, 24]
```

freetuts.net


```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 11:24:17 AM)
Các phần tử của arrListInteger ban đầu:
[4, 8, 12, 16, 20, 24]
Các phần tử của arrListInteger sau khi cập nhật:
[4, 8, 12, 16, 204, 24]
```

freetuts.net

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 11:24:17 AM)
Các phần tử của arrListInteger ban đầu:
[4, 8, 12, 16, 20, 24]
Các phần tử của arrListInteger sau khi cập nhật:
[4, 8, 12, 16, 204, 24]
```

freetuts.net

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 11:24:17 AM)
```

```
Các phần tử của arrListInteger ban đầu:
[4, 8, 12, 16, 20, 24]
Các phần tử của arrListInteger sau khi cập nhật:
[4, 8, 12, 16, 204, 24]
```

freetuts.net

Xóa phần tử

Để xóa phần tử trong `ArrayList`, Java cung cấp cho chúng ta 2 phương thức có sẵn đó là phương thức `clear()` và phương thức `remove()`.

Phương thức clear().

Phương thức `clear()` sẽ xóa tất cả các phần tử có trong `ArrayList`. Sau đây là ví dụ minh họa phương thức này.

Ví dụ

```java
public static void main(String[] args) {
    ArrayList<Integer> arrListInteger = new ArrayList<>();

    // thêm phần tử vào trong arrListInteger
    arrListInteger.add(1);
    arrListInteger.add(2);
    arrListInteger.add(3);

    // hiển thị số phần tử của arrListInteger ban đầu
    // sử dụng phương thức size()
    System.out.print("Số phần tử của arrListInteger ban đầu = " + arrListIntege
    System.out.println("\nCác phần tử của arrListInteger ban đầu: ");
    System.out.println(arrListInteger);

    // xóa tất cả các phần tử trong arrListInteger
    // sử dụng phương thức clear()
    arrListInteger.clear();

    System.out.print("Số phần tử của arrListInteger sau khi xóa = " +
        arrListInteger.size());
    System.out.println("\nCác phần tử của arrListInteger sau khi xóa: ");
    System.out.println(arrListInteger);
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 9:25:06 AM)
```

```
Số phần tử của arrListInteger ban đầu = 3
Các phần tử của arrListInteger ban đầu:
[1, 2, 3]
Số phần tử của arrListInteger sau khi xóa = 0
Các phần tử của arrListInteger sau khi xóa:
[]
```

freetuts.net

Phương thức remove().

Phương thức `remove()` sẽ xóa phần tử ra khỏi `ArrayList` theo 2 cách đó là **xóa dựa vào chỉ số của phần tử** và **xóa trực tiếp phần tử đó** (*không cần biết đến chỉ số của nó*). Ví dụ dưới đây sẽ minh họa 2 cách xóa này:

Ví dụ

```
1  public static void main(String[] args) {
2      ArrayList<String> arrListString = new ArrayList<>();
3      arrListString.add("ANDROID");
4      arrListString.add("JAVA");
5      arrListString.add("PHP");
6
7      // tạo bản sao của arrListString
8      // sử dụng phương thức clone()
9      ArrayList<String> arrListStringCopy = (ArrayList<String>) arrListString.clo
10
11     System.out.println("Các phần tử của arrListString ban đầu: ");
12     System.out.println(arrListString);
13
14     // xóa phần tử có chỉ số = 2 trong arrListString
15     // sử dụng phương thức remove()
16     // trong đó giá trị truyền vào là chỉ số của phần tử cần xóa
17     arrListString.remove(2);
18
19     System.out.println("Các phần tử của arrListStringCopy sau khi xóa phần tử s
20     System.out.println(arrListString);
21
22     System.out.println("Các phần tử của arrListStringCopy ban đầu: ");
23     System.out.println(arrListStringCopy);
24
25     // xóa phần tử "JAVA" ra khỏi arrListStringCopy
26     // sử dụng phương thức remove()
27     // trong đó giá trị truyền vào là giá trị của phần tử cần xóa
28     arrListStringCopy.remove("JAVA");
29
30     System.out.println("Các phần tử của arrListStringCopy sau khi xóa phần tử PH
31     System.out.println(arrListStringCopy);
32 }
```

Kết quả sau khi biên dịch chương trình:

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 9:48:53 AM)
Các phần tử của arrListString ban đầu:
[ANDROID, JAVA, PHP]
Các phần tử của arrListStringCopy sau khi xóa phần tử số 2:
[ANDROID, JAVA]
Các phần tử của arrListStringCopy ban đầu:
[ANDROID, JAVA, PHP]
Các phần tử của arrListStringCopy sau khi xóa phần tử PHP:
[ANDROID, PHP]
```

freetuts.net

Tìm kiếm một phần tử

Để tìm kiếm một phần tử trong `ArrayList` thì chúng ta có 3 phương pháp tìm kiếm như sau:

Tìm kiếm trực tiếp phần tử.

Để tìm kiếm trực tiếp phần tử, chúng ta sẽ sử dụng phương thức `contains()`. Kết quả trả về là `true` nếu tìm thấy, ngược lại trả về `false`.

Cú pháp

```
boolean contains(giá_trị_phần_tử_cần_tìm);
```

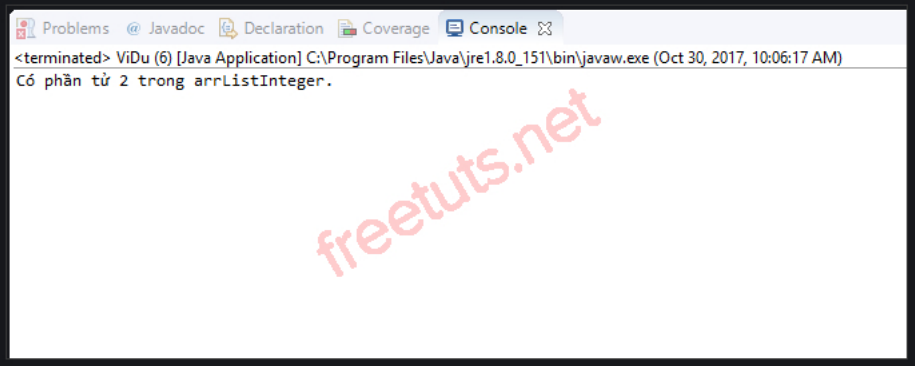
Ví dụ dưới đây sẽ tìm kiếm phần tử có giá trị = 2 trong `ArrayList` `arrListInteger`. Nếu tìm thấy thì thông báo "*Có phần tử 2 trong arrListInteger.*", ngược lại hiển thị thông báo "*Không tìm thấy phần tử 2.*".

Ví dụ

```
public static void main(String[] args) {
    ArrayList<Integer> arrListInteger = new ArrayList<>();
    arrListInteger.add(7);
    arrListInteger.add(2);
    arrListInteger.add(9);

    // kiểm tra arrListInteger có chứa phần tử có giá trị bằng 2 hay không
    boolean resultCheck = arrListInteger.contains(2);

    // nếu resultCheck == true
    // thì hiển thị thông báo "Có phần tử 2 trong arrListInteger."
    // ngược lại hiển thị thông báo "Không tìm thấy phần tử 2."
    if (resultCheck == true) {
        System.out.println("Có phần tử 2 trong arrListInteger.");
    } else {
        System.out.println("Không tìm thấy phần tử 2.");
    }
```

```
}
```

Kết quả sau khi biên dịch chương trình:



freetuts.net

Tìm kiếm vị trí xuất hiện đầu tiên của 1 phần tử trong ArrayList.

Để tìm kiếm vị trí xuất hiện đầu tiên của 1 phần tử trong `ArrayList`, chúng ta sẽ sử dụng phương thức `indexOf()`. Kết quả của phương thức này sẽ trả về chỉ số xuất hiện đầu tiên của phần tử đó trong `ArrayList`, ngược lại nếu không tìm thấy trả về -1.

Cú pháp

```
int indexOf(giá_trị_phần_tử_cần_tìm);
```

Ví dụ

```
public static void main(String[] args) {
    ArrayList<String> arrListString = new ArrayList<>();
    arrListString.add("Bird");
    arrListString.add("Rabbit");
    arrListString.add("Tiger");
    arrListString.add("Bird");

    // Tìm kiếm vị trí xuất hiện đầu tiên của phần tử "Bird" trong arrListStrin
    // sử dụng phương thức indexOf()
    // kết quả của phương thức này sẽ trả về 1 biến có kiểu số nguyên
    int position1 = arrListString.indexOf("Bird");
    System.out.println("Vị trí xuất hiện đầu tiên của phần tử \"Bird\" trong Ar
        + position1);

    // Tìm kiếm vị trí xuất hiện đầu tiên của phần tử "Cat" trong arrListString
    // vì phần tử này không có trong arrListString
    // nên kết quả sẽ bằng -1
    int position2 = arrListString.indexOf("Cat");
    System.out.println("Vị trí xuất hiện đầu tiên của phần tử \"Cat\" trong Arr
        + position2);
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems | @ Javadoc | Declaration | Coverage | Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 10:14:54 AM)
Vị trí xuất hiện đầu tiên của phần tử "Bird" trong ArrayList là 0
Vị trí xuất hiện đầu tiên của phần tử "Cat" trong ArrayList là -1
```

freetuts.net

Tìm kiếm vị trí xuất hiện cuối cùng của 1 phần tử trong List.

Để tìm kiếm vị trí xuất hiện cuối cùng của 1 phần tử trong `ArrayList`, chúng ta sẽ sử dụng phương thức `lastIndexOf()`. Kết quả của phương thức này sẽ trả về chỉ số xuất hiện cuối cùng của phần tử đó trong `ArrayList`, ngược lại nếu không tìm thấy trả về -1.

Cú pháp

```
int lastIndexOf(giá_trị_phần_tử_cần_tìm);
```

Ví dụ

```
public static void main(String[] args) {
    ArrayList<Character> arrListChar = new ArrayList<>();
    arrListChar.add('A');
    arrListChar.add('B');
    arrListChar.add('C');
    arrListChar.add('D');
    arrListChar.add('E');
    arrListChar.add('F');
    arrListChar.add('C');
    arrListChar.add('D');

    // Tìm kiếm vị trí xuất hiện cuối cùng của phần tử 'C' trong arrListChar
    // sử dụng phương thức lastIndexOf()
    // trong arrListChar có 2 phần tử 'C'
    // nên kết quả của chương trình sẽ trả về chỉ số của phần tử 'C' cuối cùng
    int position1 = arrListChar.lastIndexOf('C');
    System.out.println("Vị trí xuất hiện cuối cùng của phần tử \'C\' trong arrL
        + position1);

    // Tìm kiếm vị trí xuất hiện cuối cùng của phần tử 'D' trong arrListChar
    // sử dụng phương thức lastIndexOf()
```

```
    // trong arrListChar có 2 phần tử 'D'
    // nên kết quả của chương trình sẽ trả về chỉ số của phần tử 'D' cuối cùng
    int position2 = arrListChar.lastIndexOf('D');
    System.out.println("Vị trí xuất hiện cuối cùng của phần tử \'D\' trong arrL
        + position2);
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems | @ Javadoc | Declaration | Coverage | Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 10:30:48 AM)
Vị trí xuất hiện cuối cùng của phần tử 'C' trong arrListChar là 6
Vị trí xuất hiện cuối cùng của phần tử 'D' trong arrListChar là 7
```

freetuts.net

Chuyển ArrayList sang Array (*mảng*)

Java cung cấp cho chúng ta phương thức `toArray()` để chuyển đổi một ArrayList sang mảng tương ứng. Sau đây là ví dụ minh họa phương thức này:

Ví dụ

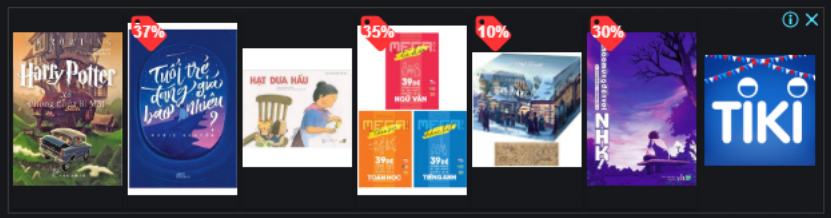
```
public static void main(String[] args) {
    ArrayList<Integer> arrListInteger = new ArrayList<>();
    arrListInteger.add(2);
    arrListInteger.add(4);
    arrListInteger.add(6);
    arrListInteger.add(8);

    // chuyển đổi arrListInteger sang mảng
    // sử dụng phương thức toArray()
    // kết quả của phương thức này sẽ trả về mảng arr
    // có kiểu là Object (chi tiết về Object chúng ta sẽ gặp trong các bài sau)
    Object[] arr = arrListInteger.toArray();

    // sử dụng vòng lặp for để hiển thị các phần tử có trong mảng arr
    for (int i = 0; i < arr.length; i++) {
        System.out.println("Phần tử tại vị trí " + i + " trong arr là " + arr[i
    }
}
```

Kết quả sau khi biên dịch chương trình:

```
Problems  @ Javadoc  Declaration  Coverage  Console
<terminated> ViDu (6) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (Oct 30, 2017, 11:00:42 AM)
Phần tử tại vị trí 0 trong arr là 2
Phần tử tại vị trí 1 trong arr là 4
Phần tử tại vị trí 2 trong arr là 6
Phần tử tại vị trí 3 trong arr là 8
```

freetuts.net

3. Lời kết



Trong bài này, tôi đã giới thiệu cho các bạn đặc điểm, các phương thức thường dùng của `ArrayList`. Sang bài sau chúng ta sẽ tiếp tục tìm hiểu 1 loại Class Collection tiếp theo đó là `HashSet` trong Java. Các bạn theo dõi nhé!

Câu hỏi và bài tập

- Java - Tìm và hiển thị phần tử lớn nhất trong 1 ArrayList.
- Java - Viết chương trình tạo 1 ArrayList mới từ 1 ArrayList đã tồn tại.
- Java - Viết chương trình loại bỏ các phần tử trùng nhau trong 1 ArrayList.
- Java - Tính và hiển thị tổng các phần tử của ArrayList.
- Java - Viết chương trình tương tác với 2 ArrayList.

Khóa học nên xem

- Lập trình Java trong 4 tuần
- Nền tảng lập trình Java cho người mới bắt đầu
- JAVA CORE - Lập trình hướng đối tượng từ Zero
- Nền Tảng Lập Trình JAVA Cho Người Mới Bắt Đầu
- Lập trình Java trong 4 tuần

BÀI KẾ SAU

BÀI KẾ TIẾP

Nguồn: freetuts.net

Những việc làm hấp dẫn

TOPDev

Java Developer (J2EE)

HansenCX *Ho Chi Minh* *Up to $1,700*

J2EE | Java

Java Developer

VIETTEL *Ha Noi* *Negotiable*

Java

Software Developer (Java, JavaScript, NodeJS)

CÔNG TY TNHH CYBOZU VIỆT NAM *Ho Chi Minh* *Negotiable*

Java | JavaScript | NodeJS

BÀI GẦN ĐÂY

- Câu hỏi phỏng vấn Java Core - phần 1
- Câu hỏi phỏng vấn về String trong Java
- Tổng quan về Exception trong Java
- Sử dụng CountDownLatch trong Java
- Hướng dẫn tạo và sử dụng ThreadPool
- Hướng dẫn cách sử dụng sleep() và join()
- Đồng bộ hóa đa luồng trong Java
- Tạo và quản lý luồng trong Java
- Lập trình đa tiến trình (Multithreading) trong Java
- Interface trong Java

ĐĂNG BÌNH LUẬN: Thông tin của bạn sẽ được giữ bí mật.

Lưu ý:: Sử dụng {Code} để đăng code, và bình luận của bạn sẽ được kiểm duyệt cẩn thận.

B I

Name (required) | Email (required) | Captcha

Đăng bình luận

Review khóa học

Mã giảm giá 40% khóa học UNICA, dùng suốt đời

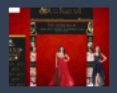
Đồng loạt giảm thêm 100k cho mỗi khóa học trên FEDU

Thiết kế đồ họa và Lập trình web - 2 kỹ năng HOT bạn nên học trong thời kỳ CMCN 4.0

Đăng ký thành viên khóa học bước trở thành PHP Developer

Lộ trình học lập trình PHP từ đầu cho người mới

Combo khóa học


Khóa học làm chủ kỹ năng thiết kế đồ họa Level 1

Combo 2 khóa học NodeJS và ReactJS + Redux

Combo 8 khóa học từng bước trở thành Full-stack Developer

Combo 5 khóa học lập trình Frontend từ căn bản đến nâng cao theo lộ trình 1

Combo 2 khóa học lập trình PHP và MySQL MVC từ căn bản đến nâng cao

Liên kết bạn bè

- Online Video Downloader
- bcdonline.net
- o7planning.org
- Nam Ngô
- Lập Trình VB
- Dịch vụ thiết kế web chuyên nghiệp
- Lời bài hát

DMCA PROTECTED

Giới thiệu

Freetuts.net là một blog cá nhân được thành lập vào năm 2014, tại đây bạn sẽ tìm thấy các bài viết về lập trình nói chung và lĩnh vực lập trình web nói riêng.

Học lập trình online đang là xu hướng, hiện nay có rất nhiều khóa học online chất lượng với chi phí rất rẻ giúp bạn tiết kiệm tiền và có thể học mọi lúc mọi nơi.