**ArrayList trong Java**

**1. Đặc điểm**

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](https://freetuts.net/tong-quan-ve-collections-trong-java-1070.html" \o "tong quan ve collections trong java 1070 html" \t "https://freetuts.net/_blank), 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**

### **2.1. 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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // 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**

|  |  |
| --- | --- |
| 1  2  3  4  5 | 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**

|  |  |
| --- | --- |
| 1  2  3  4  5 | 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);  } |

### **2.2. Hiển thị các phần tử có trong ArrayList**

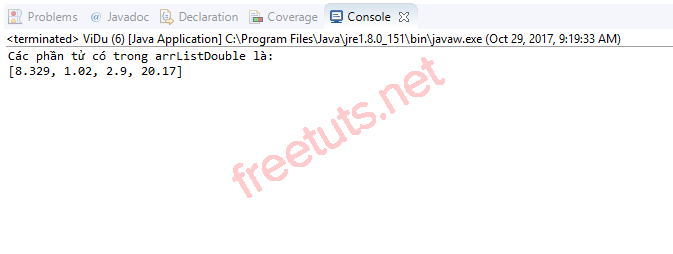
Để hiển thị các phần tử có trong ArrayList, chúng ta có các cách như sau:

**Hiển thị theo tên của ArrayList.**

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | 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:

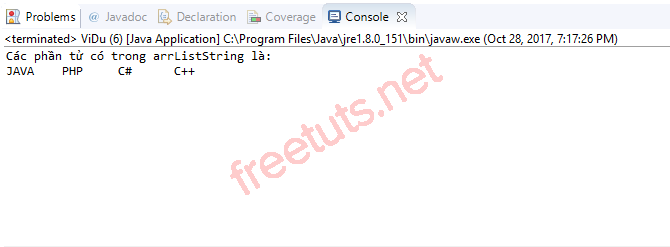


**Sử dụng vòng lặp**for**thông thường.**

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | 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++");        // 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:

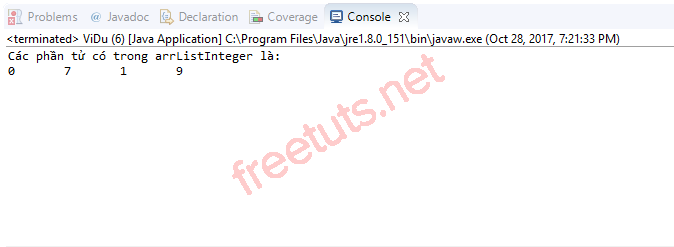


**Sử dụng vòng lặp**for**cải tiến duyệt theo đối tượng trong ArrayList.**

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | 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:



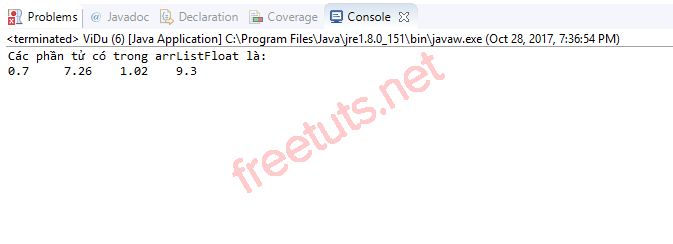
**Sử dụng Iterator.**

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

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | 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:



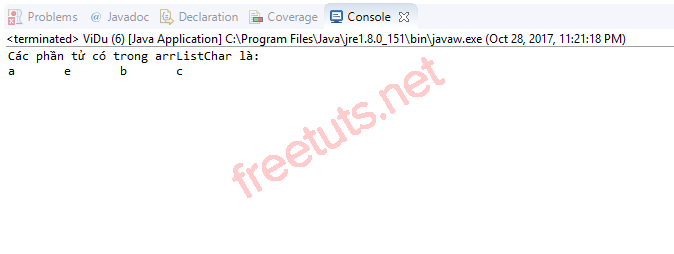
**Sử dụng ListIterator.**

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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | 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:



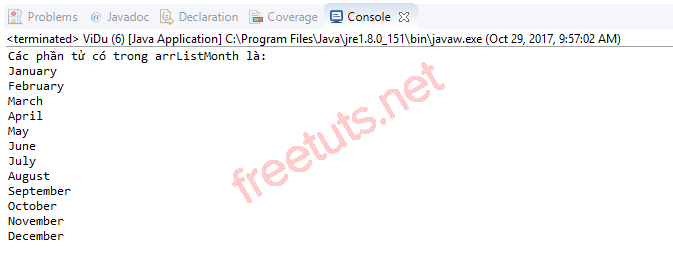
### **2.3. Thêm phần tử vào trong ArrayList**

**Thêm phần tử sử dụng phương thức add().**

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | 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:

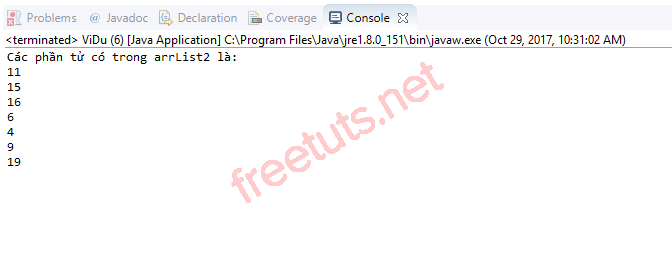


Bên cạnh đó, chúng ta có thể sử dụng phương thức addAll() để thêm tất cả các phần tử của một ArrayList khác vào 1 vị trí bất kỳ trong ArrayList đã tồn tại. Ví dụ dưới đây sẽ thêm các phần tử của arrList1 vào vị trí số 3 trong arrList2:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | 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:



### **2.4. 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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | public static void main(String[] args) {      ArrayList<Character> arrListChar = new ArrayList<>();        // 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:



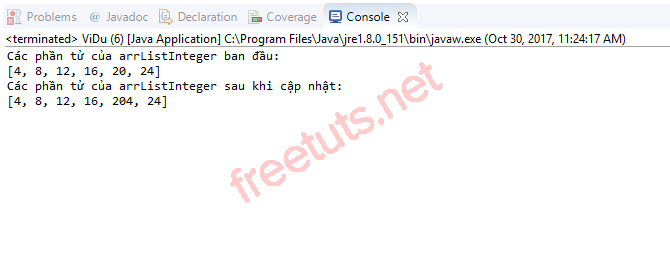
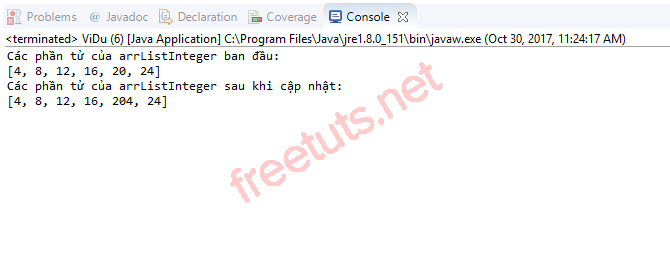
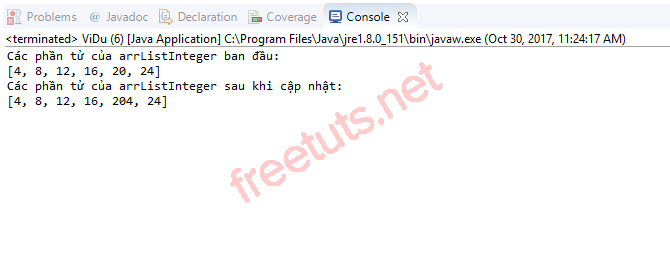
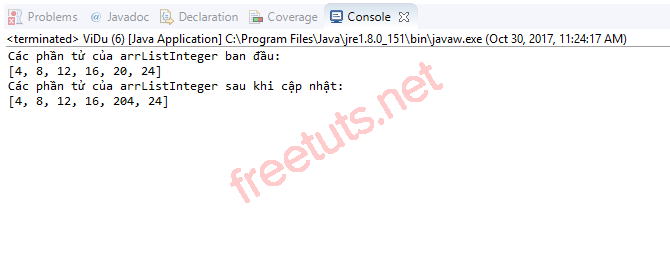
### **2.5. Cập nhật giá trị của phần tử**

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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | 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()      arrListInteger.set(4, 204);        System.out.println("Các phần tử của arrListInteger sau khi cập nhật: ");      System.out.println(arrListInteger);  } |

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:



### **2.6. 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().

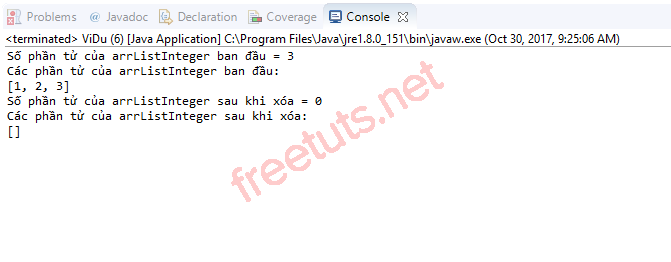
**Phương thức clear().**

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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | 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 = " + arrListInteger.size());      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:



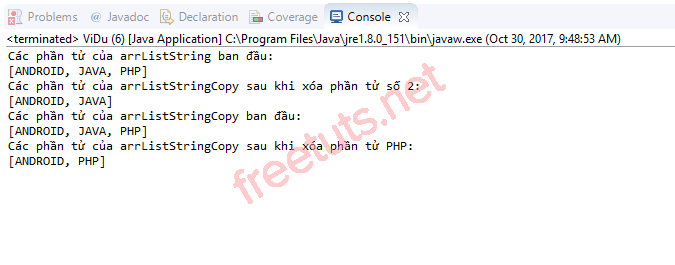
**Phương thức remove().**

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

Kết quả sau khi biên dịch chương trình:



### **2.7. 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**

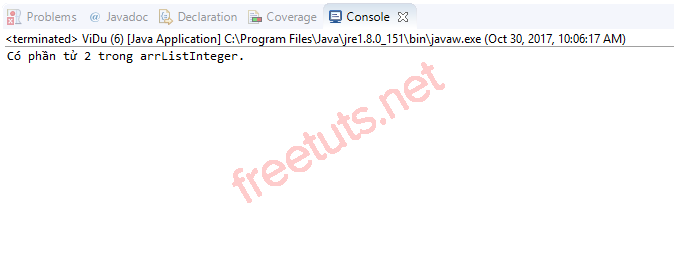
|  |  |
| --- | --- |
| 1 | boolean contains(giá\_trị\_phần\_tử\_cần\_tìm); |

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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | 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:



**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 sừ 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.

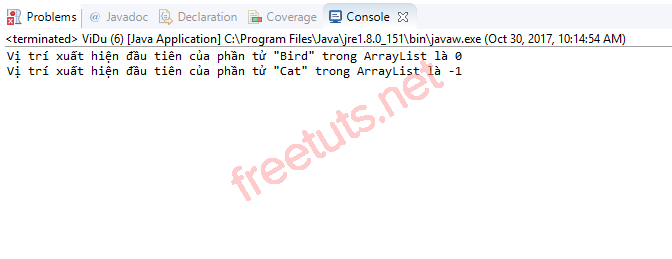
**Cú pháp**

|  |  |
| --- | --- |
| 1 | int indexOf(giá\_trị\_phần\_tử\_cần\_tìm); |

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | 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 arrListString      // 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 ArrayList là "          + 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 ArrayList là "          + position2);  } |

Kết quả sau khi biên dịch chương trình:



**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 sừ 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.

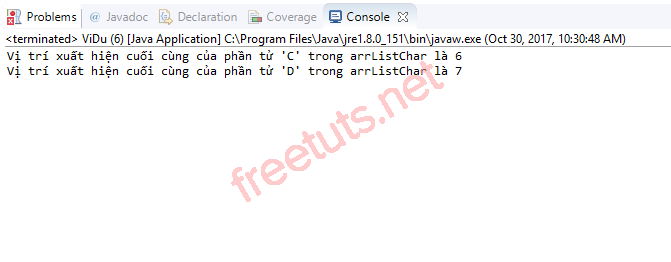
**Cú pháp**

|  |  |
| --- | --- |
| 1 | int lastIndexOf(giá\_trị\_phần\_tử\_cần\_tìm); |

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | 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 là 6      int position1 = arrListChar.lastIndexOf('C');      System.out.println("Vị trí xuất hiện cuối cùng của phần tử \'C\' trong arrListChar là "          + 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 là 7      int position2 = arrListChar.lastIndexOf('D');      System.out.println("Vị trí xuất hiện cuối cùng của phần tử \'D\' trong arrListChar là "          + position2);  } |

Kết quả sau khi biên dịch chương trình:



### **2.8. 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**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | 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:

