

Lớp ArrayList trong Java

Lớp ArrayList trong Java kế thừa AbstractList và triển khai List Interface. Lớp ArrayList hỗ trợ các mảng động mà có thể tăng kích cỡ nếu cần.

Các mảng Java chuẩn là có độ dài cố định. Sau khi các mảng được tạo, chúng không thể tăng hoặc giảm kích cỡ, nghĩa là bạn phải có bao nhiêu phần tử mà một mảng sẽ giữ.

Danh sách mảng được tạo với một kích cỡ ban đầu. Khi kích cỡ này bị vượt, collection tự động được tăng. Khi các đối tượng bị gỡ bỏ, mảng có thể bị giảm kích cỡ.

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](#).

Lớp ArrayList hỗ trợ 3 constructor. Constructor đầu tiên xây dựng một danh sách mảng trống.

```
ArrayList( )
```

Constructor thứ hai xây dựng một Array List mà được khởi tạo với các phần tử của collection c.

```
ArrayList(Collection c)
```

Constructor tiếp theo xây dựng một Array List mà có dung lượng ban đầu được xác định. Dung lượng này là kích cỡ của mảng mà được sử dụng để lưu các phần tử.

Dung lượng tự động tăng khi các phần tử được thêm vào Array List này.

```
ArrayList(int capacity)
```

Ngoài các phương thức được kế thừa từ các lớp cha, lớp ArrayList cũng định nghĩa một số phương thức sau:

STT	Phương thức và Miêu tả
1	void add(int index, Object element) Chèn phần tử đã cho tại index đã xác định trong list này. Ném IndexOutOfBoundsException nếu index này ở bên ngoài dãy (index < 0 index >

	<code>size()</code>
2	boolean add(Object o) Phụ thêm phần tử đã xác định tới cuối list này
3	boolean addAll(Collection c) Phụ thêm tất cả phần tử trong collection đã xác định tới cuối list này, trong thứ tự mà chúng được trả về bởi Iterator của collection đã cho. Ném NullPointerException nếu collection đã cho là null
4	boolean addAll(int index, Collection c) Chèn tất cả phần tử trong collection đã cho vào trong list này, bắt đầu tại vị trí đã cho. Ném NullPointerException nếu collection đã cho là null
5	void clear() Gỡ bỏ tất cả phần tử từ list này
6	Object clone() Trả về một shallow copy của ArrayList này
7	boolean contains(Object o) Trả về true nếu list này chứa phần tử đã cho. Chính thức hơn, trả về true nếu và chỉ nếu list này chứa ít nhất một phần tử e như $(o==null ? e==null : o.equals(e))$
8	void ensureCapacity(int minCapacity) Tăng dung lượng của ArrayList này, nếu cần thiết, để bảo đảm rằng nó có thể giữ tối thiểu số phần tử đã được xác định bởi tham số minCapacity
9	Object get(int index) Trả về phần tử tại index đã cho. Ném IndexOutOfBoundsException nếu index đã cho

	là ở bên ngoài dãy (<code>index < 0 index >= size()</code>)
10	int indexOf(Object o) Trả về index trong list này của sự xuất hiện đầu tiên của phần tử đã cho, hoặc -1 nếu List không chứa phần tử này
11	int lastIndexOf(Object o) Trả về index trong list này của sự xuất hiện cuối của phần tử đã cho, hoặc -1 nếu List không chứa phần tử này
12	Object remove(int index) Gỡ bỏ phần tử tại index đã cho. Ném <code>IndexOutOfBoundsException</code> nếu index ở ngoài dãy (<code>index < 0 index >= size()</code>)
13	protected void removeRange(int fromIndex, int toIndex) Gỡ bỏ từ list này tất cả phần tử mà có index ở giữa <code>fromIndex</code> và <code>toIndex</code>
14	Object set(int index, Object element) Thay thế phần tử tại vị trí đã cho trong list này với phần tử đã xác định. Ném <code>IndexOutOfBoundsException</code> nếu index ở ngoài dãy (<code>index < 0 index >= size()</code>)
15	int size() Trả về số phần tử trong list này
16	Object[] toArray() Trả về mảng chứa tất cả phần tử trong list này trong thứ tự chính xác. Ném <code>NullPointerException</code> nếu mảng này là null
17	Object[] toArray(Object[] a) Trả về một mảng chứa tất cả phần tử trong list này trong thứ tự chính xác; kiểu

	runtime là của mảng trả về là như của mảng đã xác định
18	void trimToSize() Trim dung lượng của ArrayList này về kích cỡ hiện tại của list đó

Ví dụ

Chương trình sau minh họa các phương thức được hỗ trợ bởi lớp ArrayList trong Java:

```
import java.util.*;

public class ArrayListDemo {
    public static void main(String args[]) {
        // create an array list
        ArrayList al = new ArrayList();
        System.out.println("Initial size of al: " + al.size());

        // add elements to the array list
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add("D");
        al.add("F");
        al.add(1, "A2");
        System.out.println("Size of al after additions: " + al.size());

        // display the array list
        System.out.println("Contents of al: " + al);
        // Remove elements from the array list
        al.remove("F");
        al.remove(2);
        System.out.println("Size of al after deletions: " + al.size());
    }
}
```

```
        System.out.println("Contents of a1: " + a1);  
    }  
}
```

Nó sẽ cho kết quả sau:

```
Initial size of a1: 0  
Size of a1 after additions: 7  
Contents of a1: [C, A2, A, E, B, D, F]  
Size of a1 after deletions: 5  
Contents of a1: [C, A2, E, B, D]
```