Java ArrayList類

上一篇 下一篇

ArrayList 類擴展AbstractList，並實現了List接口。支持 ArrayList 動態數組根據需要可以增長。

標準的Java數組是一個固定長度的。創建數組後，他們不能生長或縮小，這意味著必須事先知道數組將容納多少元素。

數組列表是用初始大小創建。當超出該大小時，該集合會自動放大。當被刪除的對象，數組可以被縮小。

ArrayList類支持三種構造函數。第一個構造函數建立一個空的數組列表。

```
ArrayList( )
```

下麵的構造函數建立一個與集合c 的元素初始化一個數組列表 。

```
ArrayList(Collection c)
```

下麵的構造函數建立一個數組列表，具有指定的初始容量。容量是用於存儲元素的底層數組的大小。

元素添加到數組列表的容量會自動增加。

```
ArrayList(int capacity)
```

除了從它的父類繼承的方法，ArrayList中定義了以下方法：

SN	方法及描述
1	**void add(int index, Object element)** 插入指定位置的索引在此列表中的指定元素。如果指定的索引超出 range (index < 0 || index > size())，拋出IndexOutOfBoundsException異常。
2	**boolean add(Object o)** 將指定的元素添加到此列表的末尾。
3	**boolean addAll(Collection c)** 所有追加指定collection中的元素添加到此列表的結尾，因為它們是由指定collection的迭代器返回的順序。拋出NullPointerException異常，如果指定集合為null。
4	**boolean addAll(int index, Collection c)** 插入所有指定集合中的元素插入此列表，開始在指定的位置。拋出NullPointerException異常，如果指定集合為null。
5	**void clear()** 移除此列表中的元素。
6	**Object clone()** 返回此ArrayList淺表副本。

7	**boolean contains(Object o)** 如果此列表包含指定的元素返回true。更正式地說，當且僅當此列表包含至少一個元素e，使得返回true (o==null ? e==null : o.equals(e)).
8	**void ensureCapacity(int minCapacity)** 增加此ArrayList實例的容量，如果需要，以確保其能容納至少由最小容量參數指定的元素數。
9	**Object get(int index)** 返回此列表中指定位置的元素。拋出IndexOutOfBoundsException異常，如果指定的索引超出range (index < 0 || index >= size()).
10	**int indexOf(Object o)** 返回索引中的指定元素第一次出現的這個名單，或者-1，如果列表中不包含該元素。
11	**int lastIndexOf(Object o)** 返回索引中的指定元素中最後出現的這個名單，或者-1，如果列表中不包含該元素。
12	**Object remove(int index)** 移除元素在此列表中的指定位置。如果索引超出 range (index < 0 || index >= size()) 拋出一個IndexOutOfBoundsException。
13	**protected void removeRange(int fromIndex, int toIndex)** 從這個列表中刪除所有索引為fromIndex（包括）和toIndex，獨占的元素。
14	**Object set(int index, Object element)** 替換元素在與指定元素在此列表中的指定位置。拋出IndexOutOfBoundsException異常，如果指定的索引超出 range (index < 0 || index >= size()).
15	**int size()** 返回此列表中的元素數。
16	**Object[] toArray()** 返回包含所有在此列表中正確的順序元素的數組。拋出NullPointerException異常如果指定數組為null。
17	**Object[] toArray(Object[] a)** 返回包含所有在此列表中正確的順序元素的數組;返回數組的運行時類型是指定數組。
18	**void trimToSize()** 這個微調，ArrayList實例的是列表的當前大小容量。

例子：

下麵的程序說明了幾個ArrayList中所支持的方法：

```
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 al: " + al);
   }
}
```

這將產生以下結果：

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