



链滴

Java 学习之第 4 章 集合类

作者: [xiaoqian](#)

原文链接: <https://ld246.com/article/1642571379686>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

第4章 集合类

4.1 初识集合

4.1.1 集合概述

也叫容器类，这些类可以存储任意类型的对象，而且长度可变

可分为两大类:单列集合和双列集合

Collection:单列集合类的根接口,用于存储一系列符合某种规则的元素,它有两个重要的子接口,分别是 List 和 Set。其中,List的特点是元素有序,元素可重复;Set的特点是元素无序,而且不可重复。List接口的主要实现类有ArrayList和LinkedList,Set接口的主要实现类有HashSet和 TreeSet。

Collection:单列集合类的根接口,用于**存储一系列符合某种规则的元素**,它有两个重要的子接口,分别是 List 和 Set。List的特点是**元素有序,元素可重复**;Set的特点是**元素无序,而且不可重复**。List接口的主要实现类有**ArrayList**和**LinkedList**,Set接口的主要实现类有**HashSet**和 **TreeSet**。

Map:双列集合类的根接口,用于存储具有**键(Key),值(Value)**映射关系的元素,每个元素都包含一对键值,使用Map集合时可以**通过指定的Key找到对应的Value**,例如根据一个员工的工号就可以找到对应的员工。Map接口的主要实现类有HashMap和 TreeMap。(这样子看描述有点像python的元组)

4.1.2 Collection接口简介

方法声明	功能描述
boolean add(Object o)	向集合中添加一个元素
boolean addAll(Collection c)	将指定 Collection
中的所有元素添加到该集合中	
void clear()	删除该集合中的所有元素
boolean remove(Object o)	删除该集合中指定
元素	
boolean removeAll(Collection c)	删除指定集
中的所有元素	
boolean isEmpty()	判断该集合是否为空
boolean contains(Object o)	判断该集合中是否
含某个元素	
boolean containsAll(Collection c)	判断该集合
是否包含指定集合中的所有元素	
Iterator iterator()	返回在该集合的元素上进行迭
的迭代器(Iterator),用于遍历该集合所有元素	
int size()	获取该集合元素的个数

4.2 List接口

4.2.1 List接口简介

是线性的

元素有序

方法声明

功能描述

void add(int index, Object element)
ent插入到List集合的index处

将元素ele

boolean addAll(int index, Collection c)
所包含的所有元素插入到 List集合的index处

将集合

Object get(int index)
素

返回集合索引 index处的

Object remove(int index)
素

删除 index索引处的

Object set(int index, Object element)
dex处元素替换成element对象,并将替换后的元素返回

将索引i

int indexOf(Object o)
现的位置索引

返回对象o在 List集合中

int lastIndexOf(Object o)
最后一次出现的位置索引

返回对象o在 List集合

List subList(int fromIndex, int toIndex)
引fromIndex(包括)到toIndex(不包括)处所有元素集合组成的子集合

返回从

4.2.2 ArrayList 集合

```
package com.itheima.example;
```

```
import java.util.ArrayList;
```

```
public class Example01 {  
    public static void main(String[] args) {  
        ArrayList list = new ArrayList();  
        list.add("zhangsan");  
        list.add("lisi");  
        list.add("wangwu");  
        list.add("zhaoliu");  
        System.out.println("集合的长度是:" + list.size());  
        System.out.println("第2个元素是:" + list.get(2));  
    }  
}
```

4.2.3 Iterator接口

迭代器用于遍历元素

hasNext()方法是判断是否存在下一个元素，存在调用next()方法将元素取出

```
package com.itheima.example;
```

```
import java.util.*;
```

```
public class Example02 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add("zhangsan");
        list.add("lisi");
        list.add("wangwu");
        list.add("zhaoliu");
        Iterator it = list.iterator();
        while (it.hasNext()){
            Object obj = it.next();
            System.out.println(obj);
        }
    }
}
```

4.2.4 foreach循环

(增强for循环)

```
package com.itheima.example;

import java.util.*;

public class Example03 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add("zhangsan");
        list.add("lisi");
        list.add("wangwu");
        list.add("zhaoliu");
        for(Object obj : list){
            System.out.println(obj);
        }
    }
}
```

4.3 接口

4.3.2 HashSet集合

```
package com.itheima.example;

import java.util.*;

public class Example05 {
    public static void main(String[] args) {
        HashSet hashSet = new HashSet();
        hashSet.add("zhangsan");
        hashSet.add("lisi");
        hashSet.add("wangwu");
        hashSet.add("lisi");
    }
}
```

```

    Iterator it =hashSet.iterator();
    while(it.hasNext()){
        Object obj = it.next();
        System.out.println(obj);
    }
}
}
}

```

4.4 Map接口（映射）

4.4.1 Map接口简介

方法声明	功能描述
void add(int index,Object element) ent插入到List集合的index处	将元素ele
boolean addAll(int index,Collection c) 所包含的所有元素插入到List集合的index处	将集合
Object get(int index) 素	返回集合索引 index处的
Object remove(int index) 素	删除 index索引处的
Object set(int index, Object element) dex处元素替换成element对象,并将替换后的元素	将索引 i
int indexOf(Object o) 现的位置索引	返回对象o在 List集合中
int lastIndexOf(Object o) 最后一次出现的位置索引	返回对象o在List集合
List subList(int fromIndex, int toIndex) 索引fromIndex(包括)到 toIndex(不包括)处所有元素集合组成的子集合	返回

4.4.2 HashMap集合

```

package com.itheima.example;

import java.util.*;

public class Example06 {
    public static void main(String[] args) {
        HashMap hashMap = new HashMap();
        hashMap.put("001","zhangsan");
        hashMap.put("002","lisi");
        hashMap.put("003","wnagwu");
        Set keySet=hashMap.keySet();
        Iterator iterator = keySet.iterator();
        while (iterator.hasNext()){
            Object key = iterator.next();
            Object value = hashMap.get(key);
            System.out.println(key+"="+value);
        }
    }
}

```

```
}  
}  
}
```

4.4.3 Properties集合

主要用于存储字符串类型的键和值

```
package com.itheima.example;  
  
import java.util.Properties;  
import java.util.Set;  
  
public class Example08 {  
    public static void main(String[] args) {  
        Properties p = new Properties();  
        p.setProperty("face", "微软雅黑");  
        p.setProperty("size", "20px");  
        p.setProperty("color", "green");  
        Set<String> names = p.stringPropertyNames();  
        for(String key : names){  
            String value = p.getProperty(key);  
            System.out.println(key+"="+value);  
        }  
    }  
}
```

setProperty():将配置项的键和值添加到集合中

getProperty():根据键获取对应的值

stringPropertyNames(): 得到一个所有键的集合