

* Collection接口是List、Set和Queue接口的父接口，该接口里定义的方法既可用于操作Set集合，也可用于操作List和Queue集合。
* JDK不提供此接口的任何直接实现，而是提供更具体的子接口(如：Set和List)实现。
* 在JDK5之前，Java集合会丢失容器中所有对象的数据类型，把所有对象都当成Object类型处理；从Java5增加了**泛型**以后，Java集合可以记住容器中对象的数据类型。



Collection coll = new ArrayList();

* size(); 返回集合中元素的个数

System.***out***.println(coll.size());

* add(Object obj); 向集合中添加一个元素

coll.add(123);  
coll.add("AA");  
coll.add(new Date());  
coll.add("BB");

* addAll(Collection coll); 将形参coll中包含的所有元素添加到当前集合

Collection coll1 = Arrays.asList(1,2,3);  
coll.addAll(coll1);

* isEmpty(); 判断集合是否为空

System.***out***.println(coll.isEmpty());

* clear(); 清空集合元素

coll.clear();

* contains(Object obj); 判断集合中是否包含指定的obj元素，包含则返回true。

//判断的依据，根据元素所在的类的equals()方法进行判断。自定义类要重写equals()方法。  
boolean b1 = coll.contains(123);  
b1 = coll.contains("AA");  
System.***out***.println(b1);  
boolean b2 = coll.contains(new Person("MM",23));  
System.***out***.println(b2);

* retainAll(Collection coll); 当前集合与coll的共有的元素返回给当前集合。

coll.retainAll(coll1);

* remove(Object obj); 删除集合中的obj元素，若删除成功，则返回true。

boolean b4 = coll.remove("BB");

* removeAll(Collection coll); 从当前集合中删除包含在coll中的元素。

coll.removeAll(coll1);

* equals(); 判断集合中的所有元素是否完全相同。

Collection coll2 = new ArrayList();  
coll2.add(123);  
coll2.add(new String("AA"));  
System.***out***.println(coll1.equals(coll2));

* hashCode();

System.***out***.println(coll.hashCode());

* toArray(); 将集合转化为数组

Object[] obj = coll.toArray();  
for (int i = 0;i < obj.length;i++) {  
 System.***out***.println(obj[i]);  
}

* iterator(); 返回一个iterator接口实现类的对象，进而实现集合的遍历。

Iterator iterator = coll.iterator();  
while (iterator.hasNext()) {  
 System.***out***.println(iterator.next());  
}