<http://www.cnblogs.com/epeter/p/5648026.html>

**目录：**

1. list中添加，获取，删除元素；
2. list中是否包含某个元素；
3. list中根据索引将元素数值改变(替换)；
4. list中查看（判断）元素的索引；
5. 根据元素索引位置进行的判断；
6. 利用list中索引位置重新生成一个新的list（截取集合）；
7. 对比两个list中的所有元素；
8. 判断list是否为空；
9. 返回Iterator集合对象；
10. 将集合转换为字符串；
11. 将集合转换为数组；
12. 集合类型转换；
13. 去重复；

备注：内容中代码具有关联性。

**1.list中添加，获取，删除元素；**

　　添加方法是：.add(e)；　　获取方法是：.get(index)；　　删除方法是：.remove(index)； 按照索引删除；　　.remove(Object o)； 按照元素内容删除；

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | List<String> person=new ArrayList<>();              person.add("jackie");   //索引为0  //.add(e)              person.add("peter");    //索引为1              person.add("annie");    //索引为2              person.add("martin");   //索引为3              person.add("marry");    //索引为4                person.remove(3);   //.remove(index)              person.remove("marry");     //.remove(Object o)                String per="";              per=person.get(1);              System.out.println(per);    ////.get(index)                for (int i = 0; i < person.size(); i++) {                  System.out.println(person.get(i));  //.get(index)              } |

**2.list中是否包含某个元素；**

　　方法：.contains（Object o）； 返回true或者false

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | List<String> fruits=new ArrayList<>();              fruits.add("苹果");              fruits.add("香蕉");              fruits.add("桃子");              //for循环遍历list              for (int i = 0; i < fruits.size(); i++) {                  System.out.println(fruits.get(i));              }              String appleString="苹果";              //true or false              System.out.println("fruits中是否包含苹果："+fruits.contains(appleString));                if (fruits.contains(appleString)) {                  System.out.println("我喜欢吃苹果");              }else {                  System.out.println("我不开心");              } |

**3.list中根据索引将元素数值改变(替换)；**

　　注意 .set(index, element); 和 .add(index, element); 的不同；

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | String a="白龙马", b="沙和尚", c="八戒", d="唐僧", e="悟空";              List<String> people=new ArrayList<>();              people.add(a);              people.add(b);              people.add(c);              people.set(0, d);   //.set(index, element);     //将d唐僧放到list中索引为0的位置，替换a白龙马              people.add(1, e);   //.add(index, element);     //将e悟空放到list中索引为1的位置,原来位置的b沙和尚后移一位                //增强for循环遍历list              for(String str:people){                  System.out.println(str);              } |

**4.list中查看（判断）元素的索引；**

　　注意：.indexOf（）； 和  lastIndexOf（）的不同；

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | List<String> names=new ArrayList<>();              names.add("刘备");    //索引为0              names.add("关羽");    //索引为1              names.add("张飞");    //索引为2              names.add("刘备");    //索引为3              names.add("张飞");    //索引为4              System.out.println(names.indexOf("刘备"));              System.out.println(names.lastIndexOf("刘备"));              System.out.println(names.indexOf("张飞"));              System.out.println(names.lastIndexOf("张飞")); |

**5.根据元素索引位置进行的判断;**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | if (names.indexOf("刘备")==0) {      System.out.println("刘备在这里");  }else if (names.lastIndexOf("刘备")==3) {      System.out.println("刘备在那里");  }else {      System.out.println("刘备到底在哪里？");  } |

**6.利用list中索引位置重新生成一个新的list（截取集合）；**

　　方法： .subList(fromIndex, toIndex)；　　.size() ； 该方法得到list中的元素数的和

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | List<String> phone=new ArrayList<>();              phone.add("三星");    //索引为0              phone.add("苹果");    //索引为1              phone.add("锤子");    //索引为2              phone.add("华为");    //索引为3              phone.add("小米");    //索引为4              //原list进行遍历              for(String pho:phone){                  System.out.println(pho);              }              //生成新list              phone=phone.subList(1, 4);  //.subList(fromIndex, toIndex)      //利用索引1-4的对象重新生成一个list，但是不包含索引为4的元素，4-1=3              for (int i = 0; i < phone.size(); i++) { // phone.size() 该方法得到list中的元素数的和                  System.out.println("新的list包含的元素是"+phone.get(i));              } |

**7.对比两个list中的所有元素；**

　　//两个相等对象的equals方法一定为true, 但两个hashcode相等的对象不一定是相等的对象

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | //1.<br>if (person.equals(fruits)) {      System.out.println("两个list中的所有元素相同");  }else {      System.out.println("两个list中的所有元素不一样");  }  //2.  if (person.hashCode()==fruits.hashCode()) {      System.out.println("我们相同");  }else {      System.out.println("我们不一样");  } |

**8.判断list是否为空；**

　　//空则返回true，非空则返回false

|  |  |
| --- | --- |
| 1  2  3  4  5 | if (person.isEmpty()) {      System.out.println("空的");  }else {      System.out.println("不是空的");  } |

**9.返回Iterator集合对象；**

|  |  |
| --- | --- |
| 1 | System.out.println("返回Iterator集合对象:"+person.iterator()); |

**1+0.将集合转换为字符串；**

|  |  |
| --- | --- |
| 1  2  3 | String liString="";  liString=person.toString();  System.out.println("将集合转换为字符串:"+liString); |

**11.将集合转换为数组；**

|  |  |
| --- | --- |
| 1 | System.out.println("将集合转换为数组:"+person.toArray()); |

**12.集合类型转换；**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | //1.默认类型  List<Object> listsStrings=new ArrayList<>();  　　for (int i = 0; i < person.size(); i++) {      listsStrings.add(person.get(i));  }  //2.指定类型  List<StringBuffer> lst=new ArrayList<>();  　　for(String string:person){  　　lst.add(StringBuffer(string));  } |

**13.去重复；**

|  |  |
| --- | --- |
| 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 | List<String> lst1=new ArrayList<>();              lst1.add("aa");              lst1.add("dd");              lst1.add("ss");              lst1.add("aa");              lst1.add("ss");                       //方法 1.              for (int i = 0; i <lst1.size()-1; i++) {                  for (int j = lst1.size()-1; j >i; j--) {                      if (lst1.get(j).equals(lst1.get(i))) {                          lst1.remove(j);                      }                  }              }              System.out.println(lst1);                       //方法 2.              List<String> lst2=new ArrayList<>();              for (String s:lst1) {                  if (Collections.frequency(lst2, s)<1) {                      lst2.add(s);                  }              }              System.out.println(lst2); |