# 1：集合（1）

1. 调用ArrayList集合对象的subList()方法，返回SubList是ArrayList的内部类（SubList继承AbstractList）的对象，并不是java.util.ArrayList对象，因此不能把该对象强制为ArrayList类型对象否则会报java.lang.ClassCastException。
2. ArrayList集合对象调用subList()方法后，不能再对原集合进行增，删操作改变集合的元素个数，否则子集合subList进行增删改操作会报ConcurrentModificationException异常。

并且子子集合的增删改操作得到结果会作用在原集合上，比如在子集合增加一个元素，那么原集合也会增加该元素。

ArrayList<String> arrList = **new** ArrayList<String>();

arrList.add("dd");

arrList.add("ee");

arrList.add("mm");

List<String> list=arrList.subList(0, 2);// ArrayList.modCount=subList.modCount,arrayList.subList返回是一个List类型（继成AbstractList），并不是ArrayList类型的，不可以强转为arrayList

arrList.add("ff");// arrayList.modCount++,arrayList集合对象调用subList方法后，不能对该集合进行增，删操作

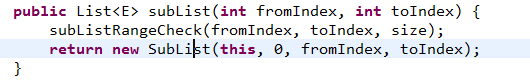
//list.remove(1);

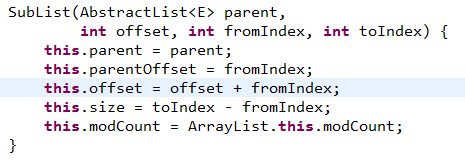
list.get(1);

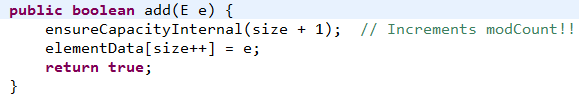
由于ArrayList的subList方法生成的内部类SubList与ArrayList是操作同一个Object数组elementData，所以在原集合arrayList中修改元素个数后，会导致子集合subList的元素有可能会变化。与实际不符，实际子集合不应随原集合的变化而变化，

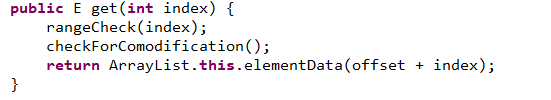
这样会造成数据的不安全，因此规定当生成一个subList对象时，如果ArrayList通过增删操作改变集合元素个数的话，那么subList操作操作子集合时会报ConcurrentModificationException异常，是通过它们的modCount变量来控制的，modCount变量表示arrayList集合元素个数改变次数，在，然arrayList调用subList（）方法会创建一个SubList对象，在SubList的构造方法中，使得arrayList和sublist的modCount相等，在arrayList的add(),remove（）方法中会对modCount增加1，而在sublist的get(),set()，add(),remove()等方法中首先判断它们的modCount是否相等，如果不等于则报ConcurrentModificationException异常。

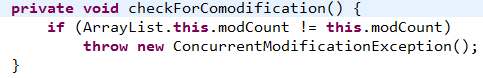
相关源码如下：











1. ：将集合转换成数组的方法，最好使用集合的toArray(T[] arrary),传入的数组类型跟集合的类型要一致，数组的大小跟集合的大小要一致，不要使用无参的toArray（）方法，此方法只能返回一个 Objet[] 数组，不能强转为其他类型数组，否则会报ClassCastExecption异常，案例如下：

 ArrayList<String> arrList = **new** ArrayList<String>();

arrList.add("dd");

arrList.add("ee");

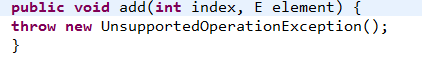
arrList.add("mm");

**int** size=arrList.size();

java.lang.UnsupportedOperationException ary=arrList.toArray(ary);

System.*out*.println(Arrays.*toString*(ary));

1. ：使用工具类Arrays.aList()把数组转换成集合时，得到的集合不能调用其add（），remove（） ，clear（）。否则会报UnsupportedOperationException异常，这个因为Arrays.alist()生成的ArrayList是Arrays的内部类，跟java.util.ArrayList实现继承相同的类和接口，由于继承AbstractList,但是没有重写add（），remove（），clear（），所以这些方法内容跟父类AbstractList这些方法一样，AbstractList的这些方法会生成UnsupportedOperationException异常，所以会报UnsupportedOperationException异常，并且对数组元素进行修改，会导致得到的集合元素也进行相应的修改。



上图abstractList的add方法

案例如下：String str [] =**new** String [10];

str[0]="jack";

str[1]="mary";

List<String> strList=Arrays.*asList*(str);

System.*out*.println(strList);

str [3]="maryt";

//strList.add("ddrr"); // UnsupportedOperationException异常

System.*out*.println(strList);

1. :不要在循环中（迭代器循环）用List对象进行元素的remove/add操作，否则会报conCurrentModificationException异常，如果foreach循环则不能删除第一个元素，删除第一个报conCurrentModificationException异常。remove元素请Iterator对象的remove.

ArrayList<String> arrList = **new** ArrayList<String>();

arrList.add("dd");

arrList.add("ee");

arrList.add("mm");

/\*for (String string : arrList) {

if(string.equals("dd")){

arrList.remove(string);

}

}\*/

Iterator<String> iterator= arrList.iterator();

**while** (iterator.hasNext()) {

**if**(iterator.next().equals("dd")){

arrList.remove("dd"); }

}