

在集合中使用泛型：jdk的新特性

- 1, 解决元素存储的安全性问题
- 2, 解决获取数据元素时需要强制类型转换的问题

```
Iterator<Integer> it = list.iterator();
while(it.hasNext()){
    System.out.println(it.next());
}
```

没有使用泛型，任何object及其子类都能添加到collection中

强转时，可能报ClassCastException的异常

```
Set set2 = map.entrySet();
for (Object obj : set2) {
    Map.Entry entry = (Map.Entry) obj;
    // System.out.println(entry.getKey() + "---->" + entry.getValue());
    System.out.println(entry);
}
```

```
map.put("AA", 78);
map.put("BB", 87);
map.put("DD", 98);

Set<Map.Entry<String,Integer>> set = map.entrySet();
for(Map.Entry<String,Integer> o : set){
    System.out.println(o.getKey() + "---->" + o.getValue());
}
```