1. 泛型的概念

所謂泛型，就是允許在定義類、接口時通過一個標識表示類中某個屬性的類型或者是某個方法的返回值及參數類型。這個類型參數將在使用時（例如，繼承或實現這個接口，用這個類型聲明變量、創建對象時）確定（即傳入實際的類型參數，也稱為類型實參）。

1. 泛型的引入背景

集合容器類在設計階段/聲明階段不能確定這個容器到底實際存的是什麽類型的對象，所以在JDK1.5之前只能把元素類型設計為Object，JDK1.5之後使用泛型來解決。因為這個時候除了元素的類型不確定，其他的部分是確定的，例如關於這個元素如何保存，如何管理等是確定的，因此此時把元素的類型設計成一個參數，這個類型參數叫做泛型。Collection<E>，List<E>，ArrayList<E>這個<E>就是類型參數，即泛型。

1. 在集合中使用泛型之前的例子

@Test  
 public void test1(){  
 ArrayList list = new ArrayList();  
 *//需求：存放学生的成绩* list.add(78);  
 list.add(76);  
 list.add(89);  
 list.add(88);  
 *//问题一：类型不安全  
// list.add("Tom");* for(Object score : list){  
 *//问题二：强转时，可能出现ClassCastException* int stuScore = (Integer) score;  
 System.*out*.println(stuScore);  
 }  
 }

圖示：



1. 在集合中使用泛型例子1  
   *//在集合中使用泛型的情况：以ArrayList为例*

@Test  
 public void test2(){  
 ArrayList<Integer> list = new ArrayList<Integer>();  
  
 list.add(78);  
 list.add(87);  
 list.add(99);  
 list.add(65);  
 *//编译时，就会进行类型检查，保证数据的安全  
// list.add("Tom");  
 //方式一：  
// for(Integer score : list){  
// //避免了强转操作  
// int stuScore = score;  
//  
// System.out.println(stuScore);  
// }  
 //方式二：* Iterator<Integer> iterator = list.iterator();  
 while(iterator.hasNext()){  
 int stuScore = iterator.next();  
 System.*out*.println(stuScore);  
 }  
 }

圖示：



1. 在集合中使用泛型例子2

*//在集合中使用泛型的情况：以HashMap为例* @Test  
 public void test3(){  
*// Map<String,Integer> map = new HashMap<String,Integer>();  
 //jdk7新特性：类型推断* HashMap<String, Integer> map = new HashMap<>();  
  
 map.put("Tom",87);  
 map.put("Jerry",87);  
 map.put("Jack",67);  
  
*// map.put(123,"ABC");  
 //泛型的嵌套* Set<Map.Entry<String,Integer>> entry = map.entrySet();  
 Iterator<Map.Entry<String, Integer>> iterator = entry.iterator();  
  
 while(iterator.hasNext()){  
 Map.Entry<String, Integer> e = iterator.next();  
 String key = e.getKey();  
 Integer value = e.getValue();  
 System.*out*.println(key + "----" + value);  
 }  
 }

1. 集合中使用泛型總結：

*① 集合接口或集合类在jdk5.0时都修改为带泛型的结构。  
② 在实例化集合类时，可以指明具体的泛型类型  
③ 指明完以后，在集合类或接口中凡是定义类或接口时，内部结构（比如：方法、构造器、属性*

*等）使用到类的泛型的位置，都指定为实例化的泛型类型。  
 比如：add(E e) --->实例化以后：add(Integer e)  
④ 注意点：泛型的类型必须是类，不能是基本数据类型。需要用到基本数据类型的位置，拿包装*

*类替换*

*⑤ 如果实例化时，没有指明泛型的类型。默认类型为java.lang.Object类型。*