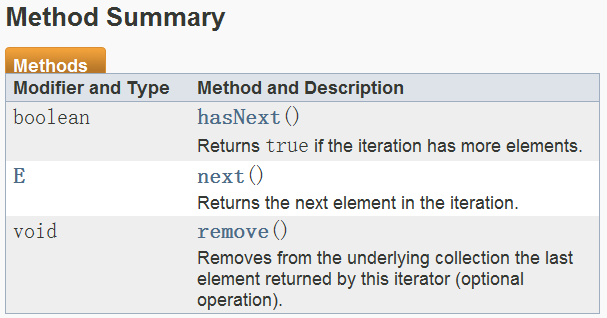
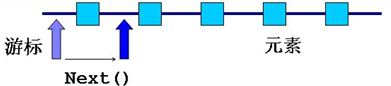
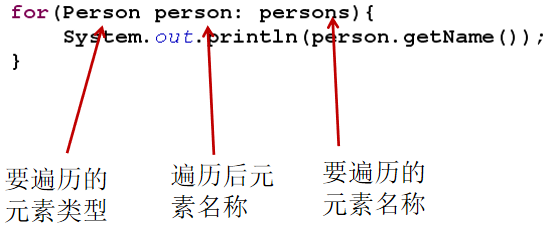
* 使用Iterator接口遍历集合元素
* **Iterator对象称为迭代器**(设计模式的一种)，主要用于遍历 Collection 集合中的元素。
* 所有实现了Collection接口的集合类都有一个iterator()方法，用以**返回一个实现了Iterator接口的对象**。
* **Iterator仅用于遍历集合**，Iterator本身并不提供承装对象的能力。如果需要创建Iterator对象，则必须有一个被迭代的集合。
* Iterator接口的方法





在调用it.next()方法之前必须要调用it.hasNext()进行检测。若不调用，且下一条记录无效，直接调用it.next()会抛出NoSuchElmentException异常。

* 使用foreach循环遍历集合元素
* Java5提供了foreach循环迭代访问Collection



@Test  
**public void** test1() {  
 Collection coll = **new** ArrayList();  
 coll.add(123);  
 coll.add(**new** String(**"AA"**));  
 coll.add(**new** Date());  
 coll.add(**"BB"**);  
 coll.add(**new** Person(**"MM"**, 23));  
  
 Iterator i = coll.iterator();  
 **while** (i.hasNext()) {  
 System.***out***.println(i.next());  
 }  
}  
  
*/\*\*  
 \* 使用增强for循环实现集合的遍历  
 \*/*@Test  
**public void** test3() {  
 Collection coll = **new** ArrayList();  
 coll.add(123);  
 coll.add(**new** String(**"AA"**));  
 coll.add(**new** Date());  
 coll.add(**"BB"**);  
 coll.add(**new** Person(**"MM"**, 23));  
  
 **for** (Object i:coll) {  
 System.***out***.println(i);  
 }  
}  
  
*/\*\*  
 \* 使用增强for循环实现数组的遍历  
 \*/*@Test  
**public void** test4() {  
 String[] str = **new** String[]{**"AA"**,**"BB"**,**"DD"**};  
  
 **for** (String s:str) {  
 System.***out***.println(s);  
 }  
}