1. 遍歷Collection的兩種方式：
2. 使用疊代器Iterator (2) foreach循環
3. java.utils包下定義的疊代器接口:Iterator
   1. 說明：

* 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。
* GOF给迭代器模式的定义为：提供一种方法访问一个容器(container)对象中各个元素，而又不需暴露该对象的内部细节。迭代器模式，就是为容器而生。
  1. 作用：遍歷集合Collection元素
  2. 如何獲取實例： coll.iterator()返回一個疊代器實例
  3. 遍歷的代碼實現：

Iterator iterator = coll.iterator();

*//hasNext():判断是否还有下一个元素*while(iterator.hasNext()){  
*//next():①指针下移 ②将下移以后集合位置上的元素返回* System.*out*.println(iterator.next());

* 1. 圖示說明：



* 1. remove()的使用：

1. *//测试Iterator中的remove():使用前須再遍歷集合中寫equals()表明要刪除的東西  
   //(情況一)如果还未调用next()或 (情況二)在上一次调用 next 方法之后已经调用了 remove 方法，  
    再调用remove都会报IllegalStateException。*

*//内部定义了remove(),可以在遍历的时候，删除集合中的元素。此方法不同于集合直接调用remove()*

@Test  
 public void test3(){  
 Collection coll = new ArrayList();  
 coll.add(123);  
 coll.add(456);  
 coll.add(new Person("Jerry",20));  
 coll.add(new String("Tom"));  
 coll.add(false);  
 *//删除集合中"Tom"* Iterator iterator = coll.iterator();  
 while (iterator.hasNext()){  
*// iterator.remove();(情況一)* Object obj = iterator.next();  
 if("Tom".equals(obj)){  
 iterator.remove();  
*// iterator.remove();(情況二)* }  
 }  
 *//遍历集合* iterator = coll.iterator();  
 while (iterator.hasNext()){  
 System.*out*.println(iterator.next());  
 }  
 }

1. jdk5.0新特性—foreach循環
   1. 遍歷集合舉例：

public void test1(){  
 Collection coll = new ArrayList();  
 coll.add(123);  
 coll.add(456);  
 coll.add(new Person("Jerry",20));  
 coll.add(new String("Tom"));  
 coll.add(false);  
 *//for(集合元素的类型 局部变量 : 集合对象)* for(Object obj : coll){  
 System.*out*.println(obj);  
 }  
}

說明：*内部仍然调用了迭代器。*

* 1. 遍歷數組舉例：

public void test2(){  
 int[] arr = new int[]{1,2,3,4,5,6};  
 *//for(数组元素的类型 局部变量 : 数组对象)* for(int i : arr){  
 System.*out*.println(i);  
 }  
}