

Iterator en Java



ÍNDICE

1. Introducción.....	2
2. Creación de un iterador.....	3
3. Recorrer colecciones.....	3
4. Eliminar elementos de la colección.....	4

1. Introducción

Una vez que ya conocemos las colecciones y los diccionarios vamos a aprender a trabajar con la interfaz `Iterator`. Un iterador es un objeto que se puede usar para **recorrer colecciones**, como `ArrayList` y `HashSet`. Se llama "iterador" porque "iterar" es el término técnico para los bucles. Para usar un iterador, debemos importarlo desde el `java.util`

2. Creación de un iterador

Para obtener un iterador podemos usar el método `iterator()` del objeto colección deseado. Por ejemplo, si tenemos una lista (`ArrayList`) podemos implementar lo siguiente:

```
List<String> nombres = new ArrayList<String>();
nombres.add("Ana");
nombres.add("Pedro");
nombres.add("Clara");
nombres.add("Ismael");

Iterator<String> it = nombres.iterator();
```

Como vemos, la forma de declararlo es:

```
Iterator<tipoDato> nombreIterador = objetoColeccion.iterator();
```

3. Recorrer colecciones

Una vez que tenemos un iterador sobre una colección, podemos recorrer dicha colección haciendo uso de los dos siguientes métodos:

- `next()`: devuelve el siguiente elemento de la colección.
- `hasNext()`: nos indica si la colección contiene más elementos.

Por ejemplo, para sacar todos los anteriores nombres podemos implementar el siguiente código fuente:

```
while(it.hasNext()) {  
    System.out.println(it.next());  
}
```

```
Ana  
Pedro  
Clara  
Ismael
```

4. Eliminar elementos de la colección

Los iteradores están diseñados para cambiar fácilmente las colecciones que recorren. Por ejemplo, si tenemos una colección de números y queremos eliminar los impares podemos implementar el siguiente código:

```
ArrayList<Integer> numeros = new ArrayList<Integer>();  
numeros.add(10);  
numeros.add(8);  
numeros.add(7);  
numeros.add(4);  
  
Iterator<Integer> it = numeros.iterator();  
  
while(it.hasNext()) {  
    if(it.next() % 2 != 0) {  
        it.remove();  
    }  
}
```

Otro ejemplo, siguiendo con el listado de nombres del apartado anterior podrías ser eliminar todos aquellos nombres que tienen una longitud superior a 8 caracteres. Para ello, podríamos hacer uso de un `Iterator` de la siguiente forma:

```
List<String> nombres = new ArrayList<String>();
nombres.add("Ana");
nombres.add("Pedro");
nombres.add("Clara");
nombres.add("Ismael");

Iterator<String> it = nombres.iterator();

while(it.hasNext()) {
    if(it.next().length() > 8) {
        it.remove();
    }
}
```

¡Importante! Intentar eliminar elementos usando un bucle `foreach` no funciona correctamente porque la colección cambia de tamaño al mismo tiempo que el código intenta iterar en el bucle.