

Iteratori (Java)

Astrazione che permette di estrarre uno alla volta gli elementi di una collezione senza esporne la rappresentazione, generalizzando la scansione lineare di un array a oggetti arbitrari.

```
interface Iterable<T> {  
    Iterator<T> iterator()  
    ...  
}
```

```
interface Iterator<E> {  
    boolean hasNext();  
    E next();  
    void remove();  
    ...  
}
```

Utilizzo:

- **remove** cancella dalla collezione l'ultimo elemento restituito, ed è opzionale (`UnsupportedOperationException`);
- la collezione non deve essere modificata se non tramite **remove** finché l'iteratore è in uso;
- **next** lancia `NoSuchElementException` se sono finiti gli elementi, **remove** `IllegalStateException` prima del primo **next**.

Pattern:

- le collezioni possono avere più iteratori che utilizzano ordini diversi (e.g. visita anticipata/simmetrica/posticipata in un albero), restituiscono un iteratore di default con `iterator()`;
- i metodi possono prendere un `Iterable<T>` o `Iterator<E>` come parametri per implementare algoritmi su sequenze generiche (fissando il parametro di tipo se necessario);
- gli iteratori sono implementati solitamente come classe interna/annidata;
- nei linguaggi funzionali si preferiscono `map/reduce/filter/...` come meccanismi di iterazione, resi possibili da funzioni (anonime) di prima classe.

Esempio

Utilizzo:

```
Iterator<Integer> it = intlist.iterator();  
while (it.hasNext())  
    int x = it.next()  
  
// anche con array (che non implementano Iterable)  
for (int x : intlist) // intlist deve implementare Iterable  
    ...
```

Implementazione:

```
public class IntSet implements Iterable<Integer> {  
    ...  
  
    // iteratore di default, con classe interna anonima  
  
    public Iterator<Integer> iterator() {  
        return new Iterator<Integer>() {  
            private int curr = 0;  
            public boolean hasNext() { ... };  
            public Integer next() { ... };  
            public void remove() { ... };  
        };  
    }  
  
    // altro iteratore con classe interna static  
  
    private static class OrderedIterator implements Iterator<Integer> {  
        public Iterator<Integer> ordered_iterator() {  
            return new OrderedIterator();  
        }  
    }  
}
```