

# Iterator

🕒 작성일시	@2023년 4월 15일 오후 10:52
📄 강의 번호	자바의정석 Chapter11
📄 유형	
📎 자료	
☑️ 복습	<input type="checkbox"/>
☰ Spring Framework	

## ▼ Iterator, ListIterator, Enumeration

- 모두 컬렉션에 저장된 요소를 접근하는데 사용하는 인터페이스
- Enumeration은 Iterator의 구버전
- ListIterator는 Iterator의 기능을 향상 시킨 것

## ▼ Iterator

- 컬렉션에 저장된 각 요소에 접근하는 기능을 가진 Iterator인터페이스를 정의하고, Collection인터페이스에서는 Iterator를 구현한 클래스의 인스턴스를 반환하는 iterator()를 정의하고 있다.

```
public interface Iterator {  
    boolean hasNext();  
    Object next();  
    void remove();  
}  
public interface Collection {  
    ...  
    public Iterator iterator();  
    ...  
}
```

- iterator()는 Collection인터페이스에 정의된 메서드이므로 자손인 List와 Set에도 포함되어 있다.
- 컬렉션 클래스에 대해 iterator()를 호출하여 Iterator를 얻은 다음 반복문, 주로 while문을 사용해서 컬렉션 클래스의 요소들을 불러올 수 있다.

```
boolean hasNext() : 읽어 올 요소가 남아있으면 true  
Object next() : 다음 요소를 읽어온다. next()를 하기전 hasNext()를 호출해서 읽어올 요소가 있는지 확인해야한다.
```

`void remove()` : `next()`로 읽어 온 요소를 삭제한다. `next()`를 호출한 다음에 `remove()`를 호출해야 한다. (선택적 기능)

- ArrayList에 저장된 요소들을 출력하기 위한 코드

```
Collection c = new ArrayList();
Iterator it = c.iterator();

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

- Map인터페이스를 구현한 컬렉션 클래스는 키와 값을 갖고 있기 때문에 각각 `keySet()`, `entrySet()`과 같은 메서드를 통해서 키와 값을 각각 따로 Set의 형태로 얻어 온 후에 다시 `iterator()`를 호출해야 Iterator를 얻을 수 있다.

## ▼ Map에서 iterator()호출하기

```
Map map = new HashMap();
...
Iterator it =map.entrySet().iterator();

Set eSet= map.entrySet();
Iterator it = eSet.iterator();

//결국 iterator가 다시 iterator를 참조
//이는 StringBuffer를 사용할 때와 유사하다
StringBuffer sb = new StringBuffer();
sb.append("A");
sb.append("B");
sb.append("C");

StringBuffer append(String str)이기 때문에 StringBuffer메서드를 호출 가
```