

# 07/07 Collection

2020년 7월 7일 화요일    오후 9:17

도대체 **Collection**이란 무엇인가?!!

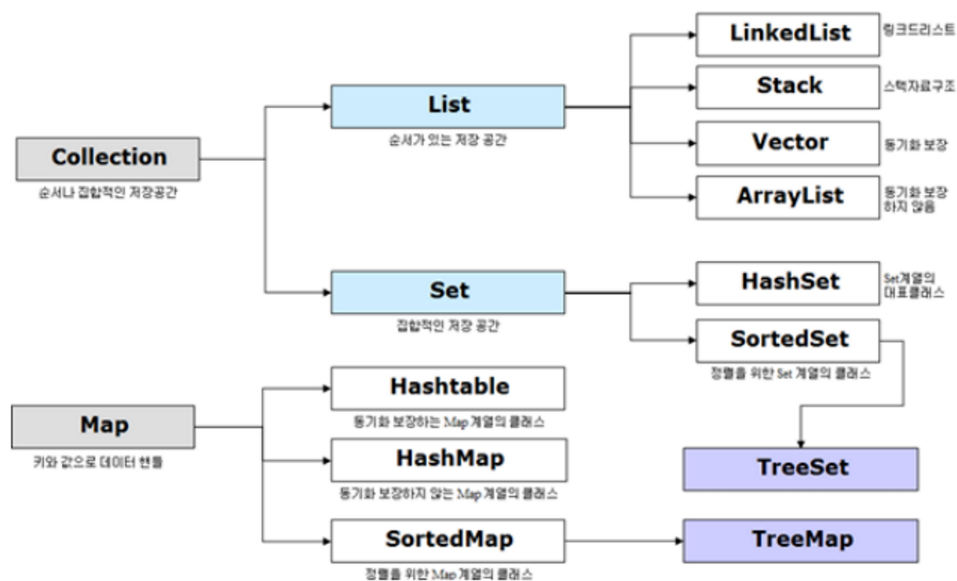
너무 자주나오는데 잘 모름! 헛갈림!

Java에서 Collection이란 **데이터의 집합, 그룹**을 의미한다!!!

다음은 상속구조를 나타낸다!!

Collection이 다 상속했다!!

Map의 경우에는 Collection으로 상속받고 있지 않지만 Collection으로 분류된다.



- Collection 인터페이스 공통 메소드 중, **iterator() 메소드**

Iterator<E> iterator() - 현재 컬렉션에 포함되어 있는 요소를 반복적으로 가져오기 위해 Iterator 인스턴스를 반환한다.

Iterator <string> itr = list.iterator(); // 모든 컬렉션 안에는 iterator메소드가 있다!!

```
while(itr.hasNext()){
    String str = itr.next();
    sysout(str);
}
```

- Vector**

배열 사용 시 배열의 크기를 벗어나는 인덱스를 접근하면 에러가 발생한다.

자바에서 동적인 길이로 여러 데이터형을 저장하기 위해 Vector클래스를 제공한다.

Vector 클래스는 **가변 길이의 배열**이라 할 수 있다!!

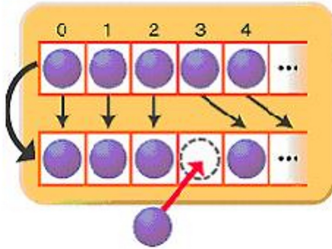
- ArrayList와 LinkedList의 차이?!

➢ ArrayList는 내부 배열에 기반을 둔 리스트 구현을 제공하며 리스트 요소에 대한 접근이 빠르다!!

데이터의 추가, 삭제를 위해 데이터를 복사하는 방법을 사용한다.

데이터를 복사하는 방법을 사용하므로 대량의 데이터를 추가/삭제하는 경우 그만큼 데이터 복사가 많이 일어나 성능이 저하된다.

반면 각 데이터는 인덱스를 가지고 있어 한번에 참조가 가능해 데이터의 검색에 유리하다!!!



- LinkedList는 연결된 노드들을 기반으로 구현된 리스트!  
 리스트 요소에 대한 접근은 느린 반면에 추가, 삭제에 대한 작업은 빠르다!!!  
 ArrayList와 같이 데이터의 추가, 삭제 시 불필요한 데이터 복사가 없어 데이터 추가, 삭제가 빠르다!!!  
 LinkedList는 데이터를 저장하는 각 노드가 이전 노드와 다음 노드의 상태만 알고있다고 생각하면 된다!!!

