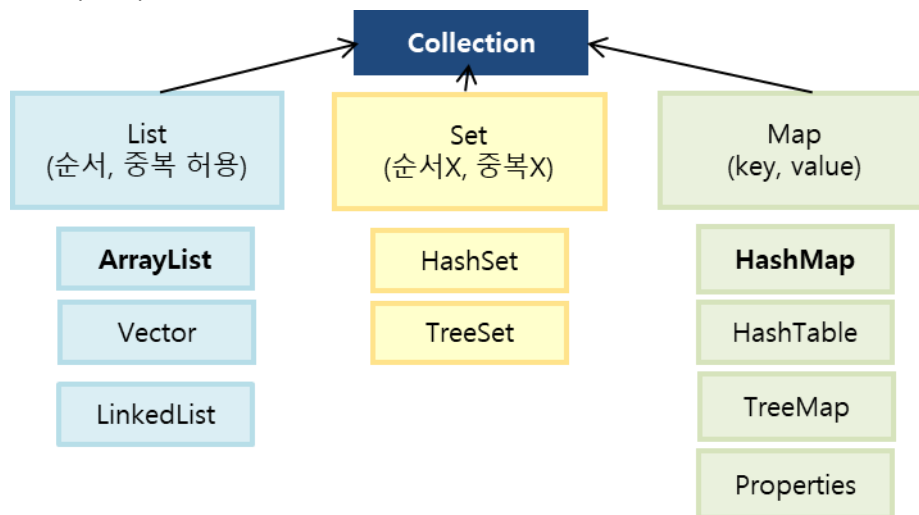


13. 자료 구조(Collection) //src: ch16_collection 참고

1) 자료 구조(Collection)

- 다수의 데이터, 데이터 그룹을 의미
- 데이터의 성질에 따라 데이터를 관리(정리)해야 하는 방식이 다름
- 자료 구조형 안에서는 객체의 레퍼런스 만을 관리함(`int` 불가능, `Integer` 가능)

2) Collection 자료 구조



(1) List계열; 배열과 유사(순서O, 중복O), 크기 유동적(배열 단점 보완)

`add(객체)`, `add(index, 객체)`, `set(index, 객체)`, `get(index)`, `size()`, `remove(index)`, `remove(객체)`, `contains(객체)`, `isEmpty()`, `clear()` 등

※데이터를 순차적으로 추가/삭제할 경우 `ArrayList`, 비순차적일 경우 `LinkedList` 적합(빠름)

(2) Set계열; 순서X, 중복 데이터X

//equal()메소드 override

(중복 데이터 여부는 `hashCode()`값이 같거나 `equal()` 메소드 결과 값에 의해 해석)

(3) Map계열; 인덱스 대신 key 값으로 데이터(value)에 액세스. 키는 중복될 수 없음.

`put(key, value)`, `remove(key)` 등

3) **Iterator(반복자)**; 데이터를 반복 검색하기 유용한 인터페이스

```
ArrayList<String> list = new ArrayList<String>();
```

```
Iterator<String> iterator = list.iterator();
```

```
while(iterator.hasNext()) {
```

```
    System.out.println(iterator.next());
```

```
}
```