

Map<K, V> interface

Map<K,V> 인터페이스를 구현하는 컬렉션즈 클래스

- Map<K,V> 인터페이스를 구현하는 컬렉션즈 클래스들은 **Key - Value** 방식으로 인스턴스를 등록한다!
- **Value : 인스턴스 (중복가능!)**
- **Key는 Value를 찾는 열쇠 (중복불가!)**
- Map<K,V>를 구현하는 대표적인 컬렉션즈 클래스에는 **HashMap<K,V>**, **TreeMap<K,V>**가 있다.
- **TreeMap<K,V>**는 정렬방식으로 등록한다.

- Map<K,V>의 대표적인 메소드

put(K,V)	Key - Value 형태로 인스턴스를 등록한다.
get(K key)	Key 값에 대응되는 Value를 꺼낸다.
remove(K key)	Key 값에 대응되는 Value를 지운다.
int size()	Map<K,V>에 저장된 Key-Value의 개수 반환
Set<K> keySet()	Set<K> 형태의 keySet(Key들의 모임) 반환

i. HashMap<K,V>

- 해시 알고리즘이 적용된 Map
- **Key의 구분과 key를 이용한 Value의 탐색에 해시 알고리즘이 적용 (Key의 동등비교와 중복저장을 막기 위해서)**

```
import java.util.HashMap;

public class HashMapDemo {
    public static void main(String[] args) {
        HashMap<Integer, String> hMap = new HashMap<Integer, String>();
        hMap.put(new Integer(3), "수지");
        hMap.put(5, "아이유");
        hMap.put(8, "혜리");

        // Key 들의 모임인 KetSet을 가져온다.
        Set<Integer> keySet = hMap.keySet();

        // ketSet은 Set<E> interface이므로 iterator를 가지고 있다.
        Iterator<Integer> iterator = keySet.iterator();

        while(iterator.hasNext()){
            System.out.println(hMap.get(iterator.next()));
        }

        System.out.println("6-3반 8번학생 : "+hMap.get(new Integer(8)));
        System.out.println("6-3반 5번학생 : "+hMap.get(5));
        System.out.println("6-3반 3번학생 : "+hMap.get(3));

        hMap.remove(5);
        System.out.println("6-3반 5번학생 : "+hMap.get(5));
    }
}
```

ii. TreeMap<K,V>

- Key를 기준으로 정렬한다.
- 트리 자료구조를 이용한 데이터 관리

```
import java.util.Iterator;
import java.util.NavigableSet;
import java.util.TreeMap;

public class TreeMapDemo {
    public static void main(String[] args) {
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        // 순서가 앞서는 key부터 정렬
        tMap.put(1, "data1");
        tMap.put(3, "data3");
        tMap.put(5, "data5");
        tMap.put(2, "data2");
        tMap.put(4, "data4");

        // NavigableSet<E> : 오름차순, 내림차순의 정렬기준을 가지고 있는 iterator를 소유한 컬렉션즈 클래스
        NavigableSet<Integer> navi = tMap.navigableKeySet();

        System.out.println("오름차순 정렬...");

        Iterator<Integer> iterator = navi.iterator();
        while(iterator.hasNext())
            System.out.println(tMap.get(iterator.next()));

        System.out.println("내림차순 정렬...");

        iterator = navi.descendingIterator();
        while(iterator.hasNext())
            System.out.println(tMap.get(iterator.next()));
    }
}
```