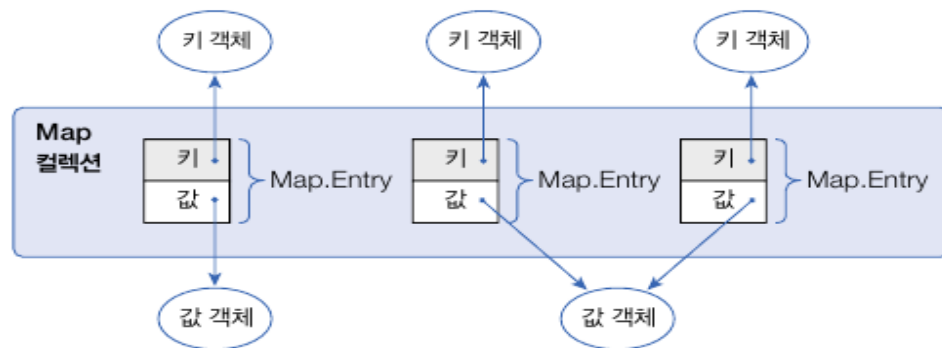
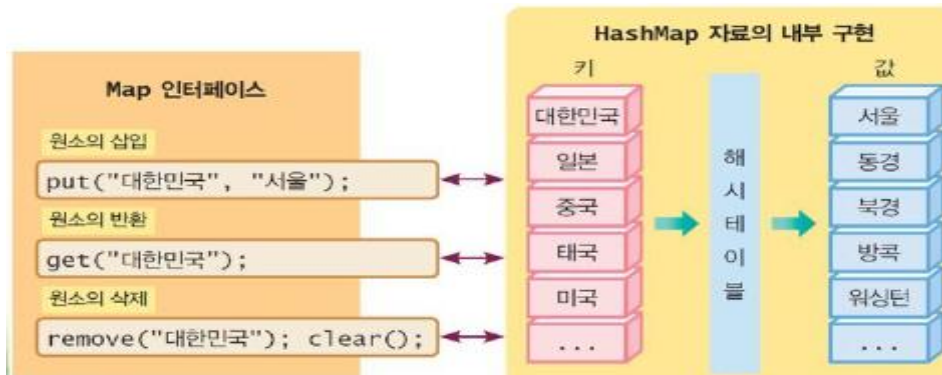


Map 컬렉션

■ Map 컬렉션

- Map 컬렉션은 키(key)와 값(value)으로 구성된 엔트리(Entity)객체를 저장한다. 여기서 키와 값은 모두 객체이다.
- 키는 중복 저장할 수 없지만, 값은 중복 저장할 수 있다.
- 기존에 저장된 키와 동일한 키로 값을 저장하면 기존의 값은 없어지고 새로운 값으로 대체된다.



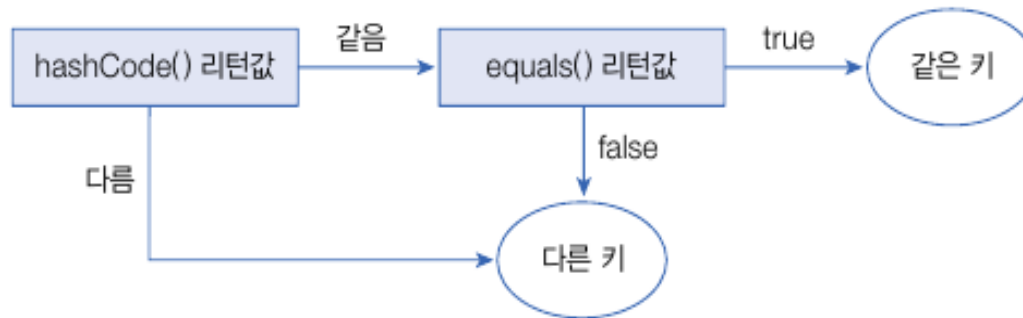
- Map 컬렉션에는 HashMap, Hashtable, LinkedHashMap, Properties, TreeMap 등이 있다.
- Map 컬렉션에서 공통적으로 사용 가능한 Map 인터페이스 메소드들은 키로 객체들을 관리하기 때문에 키를 매개값으로 갖는 메소드가 많다.

※ K = Key, V = Value

기능	메소드	설명
객체 추가	V put(K key, V value)	주어진 키와 값을 추가, 저장되면 값을 리턴
	boolean containsKey(Object key)	주어진 키가 있는지 여부
객체 검색	boolean containsValue(Object value)	주어진 값이 있는지 여부
	Set<Map.Entry<K,V>> entrySet()	키와 값의 쌍으로 구성된 모든 Map.Entry 객체를 Set에 담아서 리턴
	V get(Object key)	주어진 키의 값을 리턴
	boolean isEmpty()	컬렉션이 비어있는지 여부
	Set<K> keySet()	모든 키를 Set 객체에 담아서 리턴
	int size()	저장된 키의 총 수를 리턴
	Collection<V> values()	저장된 모든 값 Collection에 담아서 리턴
	void clear()	모든 Map.Entry(키와 값)를 삭제
객체 삭제	V remove(Object key)	주어진 키와 일치하는 Map.Entry 삭제, 삭제가 되면 값을 리턴

◎ HashMap

- HashMap은 **키**로 사용할 객체가 `hashCode()` 메소드의 리턴값이 같고, `equals()` 메소드가 `true`를 리턴할 경우 동일 키로 보고 중복 저장을 허용하지 않는다.



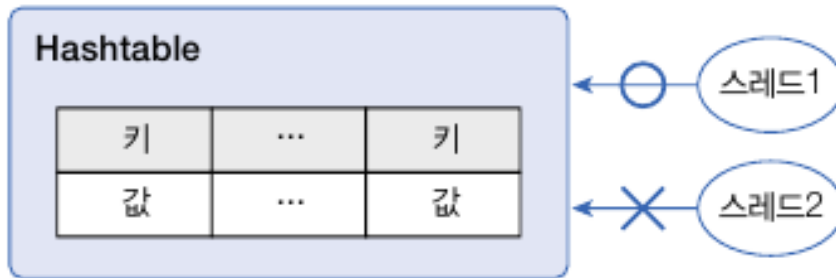
```
Map<K, V> map = new HashMap<K, V>();
```

↑ ↑ ↑ ↑
키 타입 값 타입 키 타입 값 타입

```
Map<String, Integer> map = new HashMap<String, Integer>();  
Map<String, Integer> map = new HashMap<>();  
Map map = new HashMap();
```

◎ Hashtable

- Hashtable은 HashMap과 동일한 내부 구조를 가지고 있다.
- 차이점은 Hashtable은 동기화된(synchronized) 메소드로 구성되어 있기 때문에 멀티 스레드가 동시에 Hashtable의 메소드들을 실행할 수 없다는 것이다.
- 따라서 멀티 스레드 환경에서도 안전하게 객체를 추가, 삭제할 수 있다.



```
Map<String, Integer> map = new Hashtable<String, Integer>();  
Map<String, Integer> map = new Hashtable<>();  
Map map = new Hashtable();
```

◎ Properties

- Properties는 Hashtable의 자식 클래스이기 때문에 Hashtable의 특징을 그대로 가지고 있다.
- Properties는 키와 값을 String 타입으로 제한한 컬렉션이다.
- Properties는 주로 확장자가 .properties인 프로퍼티 파일을 읽을 때 사용한다.
- 프로퍼티 파일은 키와 값이 = 기호로 연결되어 있는 텍스트 파일이다.
- 일반 텍스트 파일과는 다르게 ISO 8859-1 문자셋으로 저장되며, 한글일 경우에는 Wu+유니코드로 표현되어 저장된다.
- Properties 객체를 생성하고, load() 메소드로 프로퍼티 파일의 내용을 메모리로 로드한다.
- 일반적으로 프로퍼티 파일은 클래스 파일(~.class)들과 함께 저장된다.
- Class 객체의 getResourceAsStream() 메소드는 주어진 상대 경로의 리소스 파일을 읽는 InputStream을 리턴한다.

```
Properties properties = new Properties();  
properties.load(Xxx.class.getResourceAsStream("database.properties"));
```