



HashMap CRUD 연산(1)

- 생성자와 4가지 메소드를 통한 연산 -



학습내용

- 1 HashMap CRUD 연산 소개
- 2 HashMap CRUD 연산 실습(생성자와 4가지 메소드를 통한 연산)

학습목표

- HashMap CRUD 연산의 개념을 설명할 수 있다.
- HashMap CRUD 연산을 활용할 수 있다.



HashMap

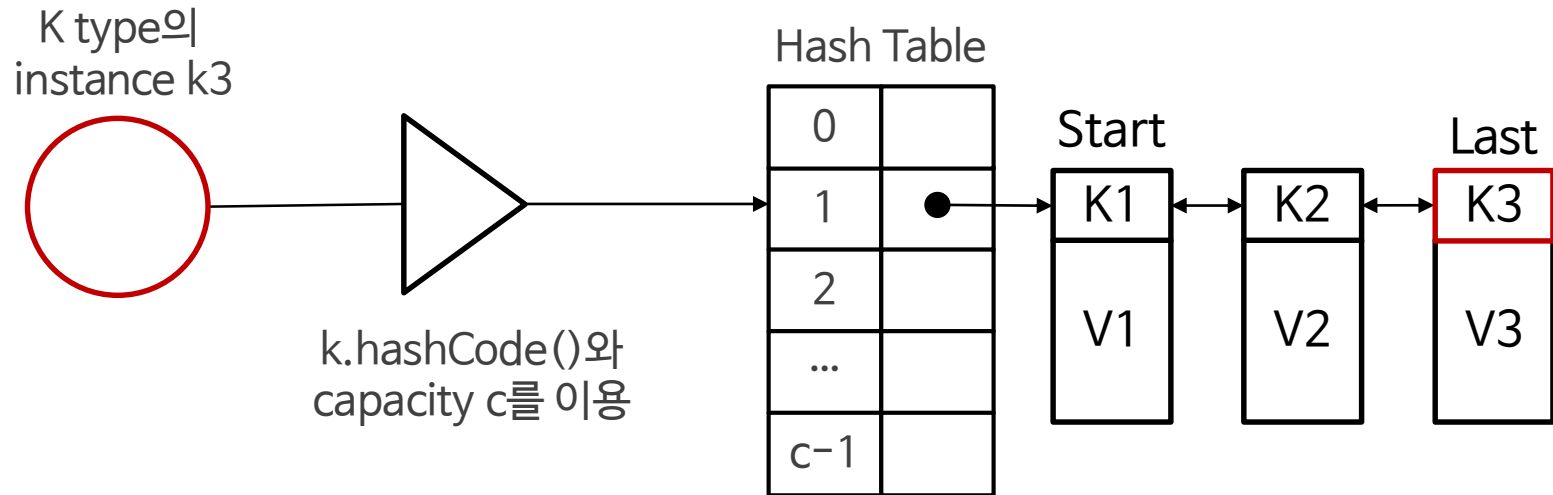
Map의 **Hash 기반** 구현

중복을 허용하지 않는 빠른 탐색이 가능한
Key에 대한 Value 유지

HashMap<K, V>

Key 값의 Type K

Value 값의 Type V



Map

META

C

R

D

T

| Return Type | Method | Description |
|-----------------|--|---|
| boolean | isEmpty() | Map이 비어 있는지 확인 |
| int | size() | Map의 크기를 반환 |
| V | put(K key, V value) | Map에 key와 value의 pair를 삽입 및 기존의 key에 대한 value 교체 |
| V | putIfAbsent(K key, V value) | Map에 주어진 key가 없을 때만 key-value 삽입 |
| boolean | containsKey(Object key) | Map에 key가 존재하는지 확인 |
| boolean | containsValue(Object value) | Map에 임의의 key에 연관된 value가 존재하는지 확인 |
| V | get(Object key) | Map에서 주어진 key에 대한 value를 반환 |
| V | getOrDefault(Object key, V defaultValue) | 주어진 key가 있다면 get(Object key), 없다면 defaultValue 반환 |
| V | remove(Object key) | Map에서 주어진 key와 연관된 value 제거 |
| void | clear() | Map을 비움 |
| Set<K> | keySet() | Map에 존재하는 모든 key들을 Set으로 반환 |
| Set<Entry<K,V>> | entrySet() | Map에 존재하는 모든 key-value pair (Entry)를 Set으로 반환 |
| Collection<V> | values() | Map에서 각각의 key에 연관된 모든 Value를 Collection으로 반환 |
| void | forEach(BiConsumer<? super K, ? super V action>) | Map의 각 Entry의 K, V에 대해 action을 수행함 |

[HashMap CRUD 연산]

| Return Type | Method | Description |
|-------------|-----------|---------------|
| 생성자 | HashMap() | 빈 HashMap을 생성 |

HashMap CRUD

| Return Type | Method | Description |
|-------------|--|--|
| boolean | <code>isEmpty()</code> | Map이 비어 있는지 확인 |
| int | <code>size()</code> | Map의 크기를 반환 |
| V | <code>put(K key, V value)</code> | Map에 key와 value의 pair를 삽입 및 기존의 key에 대한 value 교체 |
| V | <code>putIfAbsent(K key, V value)</code> | Map에 주어진 key가 없을 때만 key-value 삽입 |

HashMap CRUD

| Return Type | Method | Description |
|-------------|-----------------------------|--|
| boolean | isEmpty() | Map이 비어 있는지 확인 |
| int | size() | Map의 크기를 반환 |
| V | put(K key, V value) | Map에 key와 value의 pair를 삽입 및 기존의 key에 대한 value 교체 |
| V | putIfAbsent(K key, V value) | Map에 주어진 key가 없을 때만 key-value 삽입 |



[HashMap CRUD 연산]

put

custom class

HashMap CRUD

| Return Type | Method | Description |
|-------------|--|--|
| 생성자 | HashMap (Map<? extends K, ? extends V> m) | HashSet을 Map m의 요소를 가져와 생성 |
| 생성자 | HashMap (int capacity) | 빈 HashSet을 Hash Table의 초기 크기를 설정해 생성 |
| 생성자 | HashMap (int capacity, float loadFactor) | 빈 HashSet을 Hash Table의 초기 크기와 Rehashing 조건을 설정해 생성 |

Remind

HashMap CRUD 연산

| Return Type | Method | Description |
|-------------|--|--|
| 생성자 | HashMap() | 빈 HashMap을 생성 |
| boolean | isEmpty() | Map이 비어 있는지 확인 |
| int | size() | Map의 크기를 반환 |
| V | put(K key, V value) | Map에 key와 value의 pair를 삽입 및 기존의 key에 대한 value 교체 |
| V | putIfAbsent(K key, V value) | Map에 주어진 key가 없을 때만 key-value 삽입 |
| 생성자 | HashMap(Map<? extends K, ? extends V> m) | HashSet을 Map m의 요소를 가져와 생성 |
| 생성자 | HashMap(int capacity) | 빈 HashSet을 Hash Table의 초기 크기를 설정해 생성 |
| 생성자 | HashMap(int capacity, float loadFactor) | 빈 HashSet을 Hash Table의 초기 크기와 Rehashing 조건을 설정해 생성 |