



# Set 인터페이스의 해시 기반 자료구조 설계 (3)

- 5가지 메소드를 통한 연산

## 학습내용

- 1 Set 인터페이스의 해시기반 자료구조 설계
- 2 Set 인터페이스의 해시기반 자료구조 구현

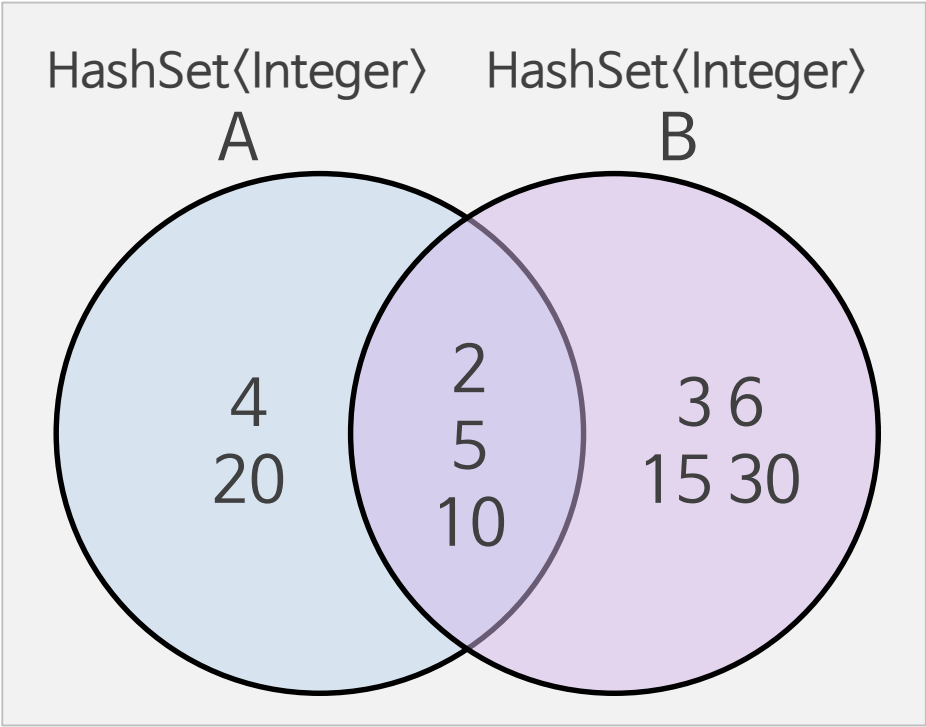
## 학습목표

- Set 인터페이스의 해시기반 자료구조를 설계할 수 있다.
- Set 인터페이스의 해시기반 자료구조를 구현할 수 있다.

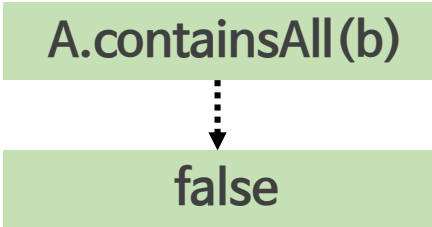
# set의 집합연산

Return Type	Method	Description
boolean	<code>containsAll(Collection&lt;?&gt; c)</code>	Set이 Collection c의 instance들을 모두 갖고있는지 확인 (부분집합)

B를 순회하며 각 요소 b에 대해 `A.contains(b)` 수행



부분집합

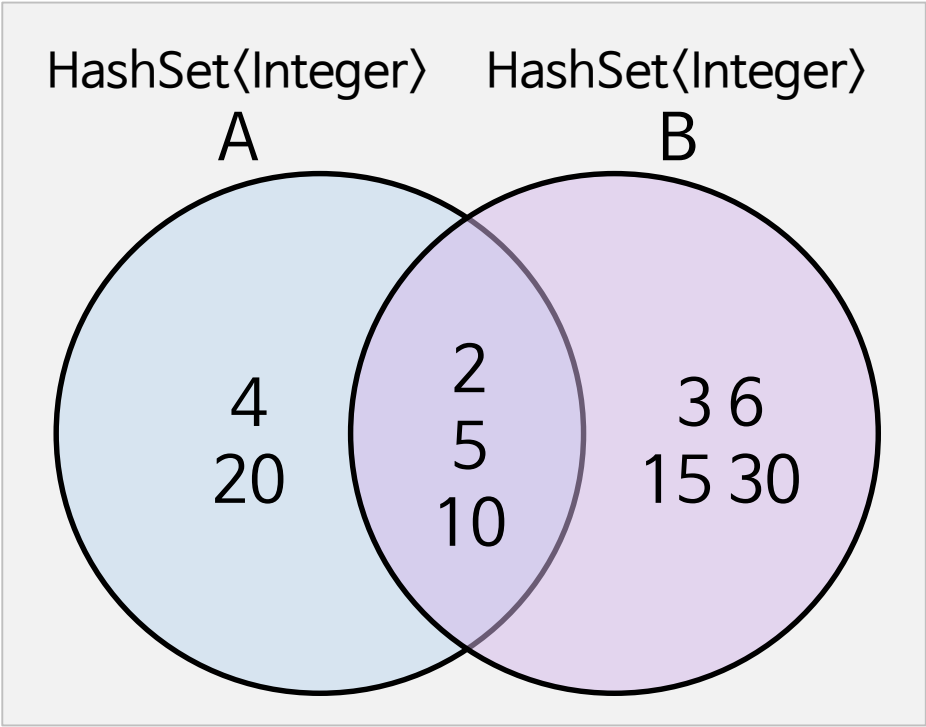


[MyHashSet 구현]

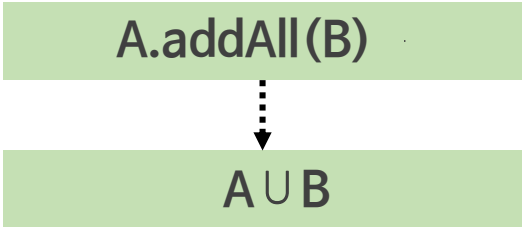
**containsAll**

Return Type	Method	Description
boolean	<code>addAll(Collection&lt;?&gt; c)</code>	Set에 Collection c의 instance들을 모두 추가함 (합집합)

B를 순회하며 각 요소 b에 대해 A.add(b) 수행



합집합

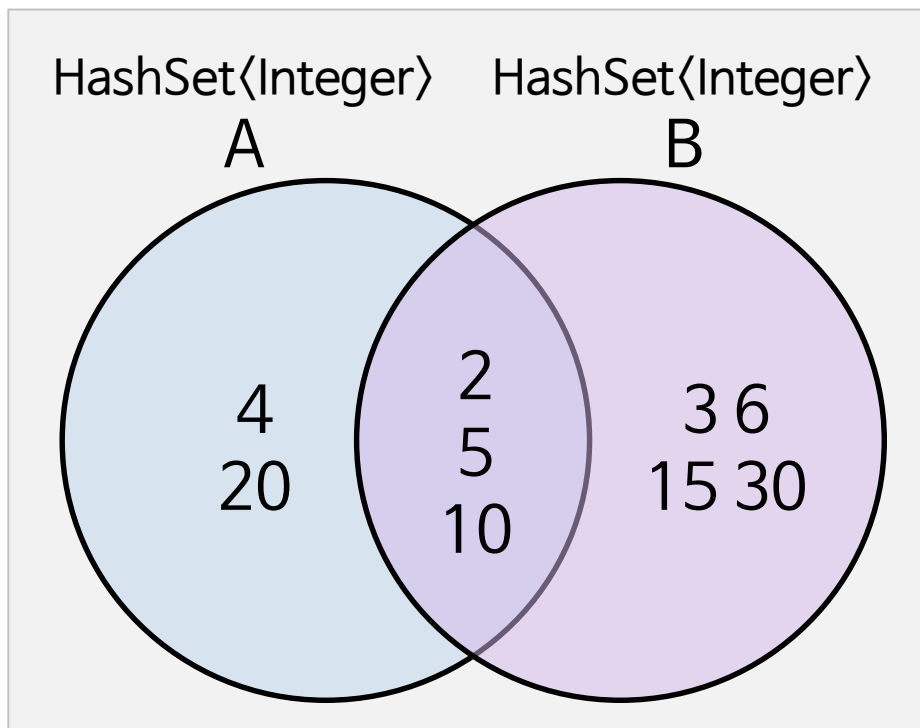


**[MyHashSet 구현]**

**addAll**

Return Type	Method	Description
boolean	<code>retainAll(Collection&lt;?&gt; c)</code>	Set에서 Collection c의 instance인 것만을 남김 (교집합)

현재의 Set을 순회하며, C와의 공통의 요소를 ArrayList에 삽입 및  
현재의 Set을 Clear 후 ArrayList를 순회하며 Add



교집합

`A.retainAll(B)`



$A \cap B$

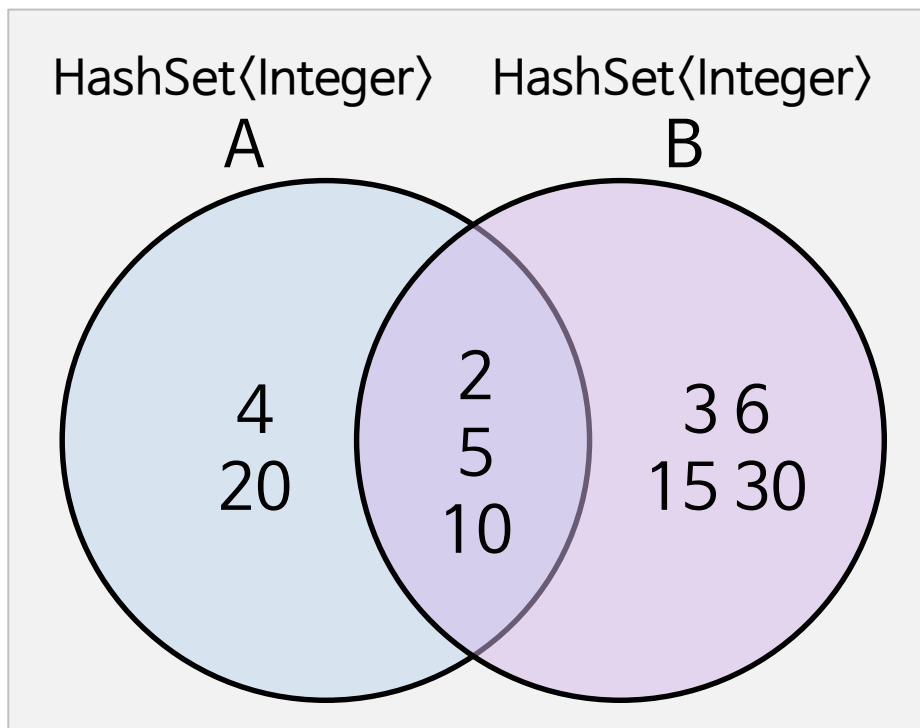


[MyHashSet 구현]

**retainAll**

Return Type	Method	Description
boolean	<code>removeAll(Collection&lt;?&gt; c)</code>	Set에서 Collection c의 instance인 것은 지움 (차집합)

B를 순회하며 각 요소 b에 대해 **A.contains(b)가 true**라면 → A.remove(b) 수행



차집합

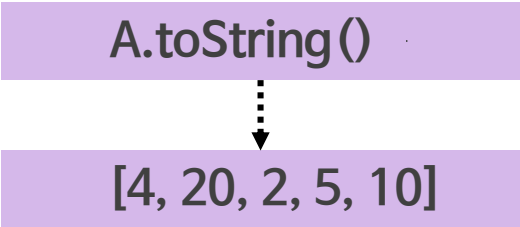
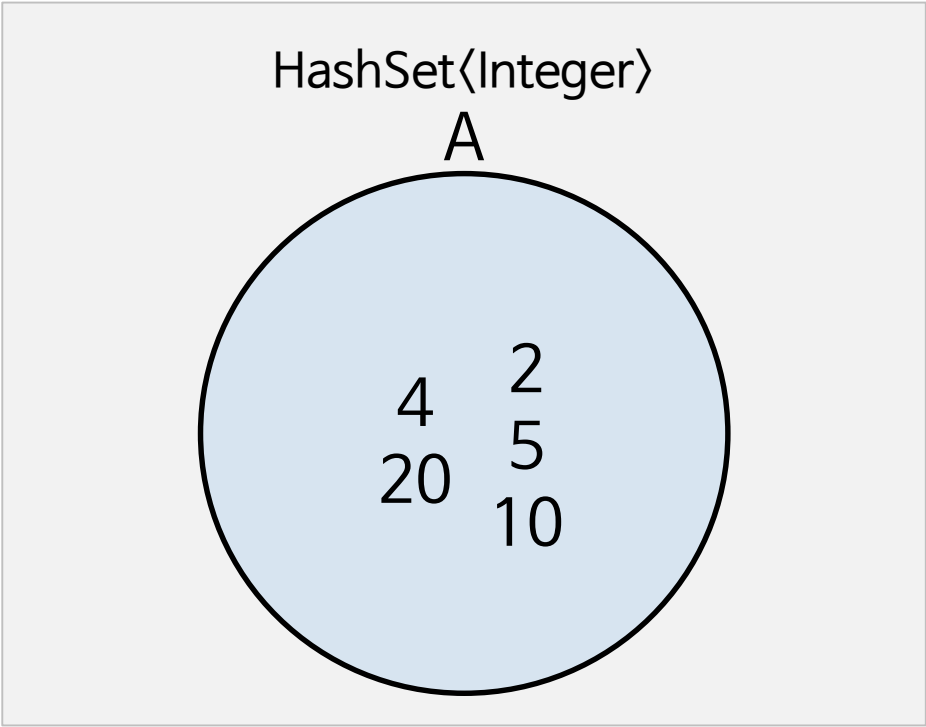
`A.removeAll(B)`

$A - B$

**[MyHashSet 구현]**

**removeAll**

Return Type	Method	Description
boolean	toString()	Set을 잘 표현할 수 있는 문자열을 반환



[MyHashSet 구현]

toString

# Remind

## Set 인터페이스의 해시 기반 자료구조 구현

Return Type	Method	Description
boolean	containsAll (Collection<?> c)	Set이 Collection c의 instance들을 모두 갖고있는지 확인 (부분집합)
boolean	addAll (Collection<?> c)	Set에 Collection c의 instance들을 모두 추가함 (합집합)
boolean	retainAll (Collection<?> c)	Set에서 Collection c의 instance인 것만을 남김 (교집합)
boolean	removeAll (Collection<?> c)	Set에서 Collection c의 instance인 것은 지움 (차집합)
boolean	toString()	Set을 잘 표현할 수 있는 문자열을 반환