KH정보교육원

컬렉션(Collection)

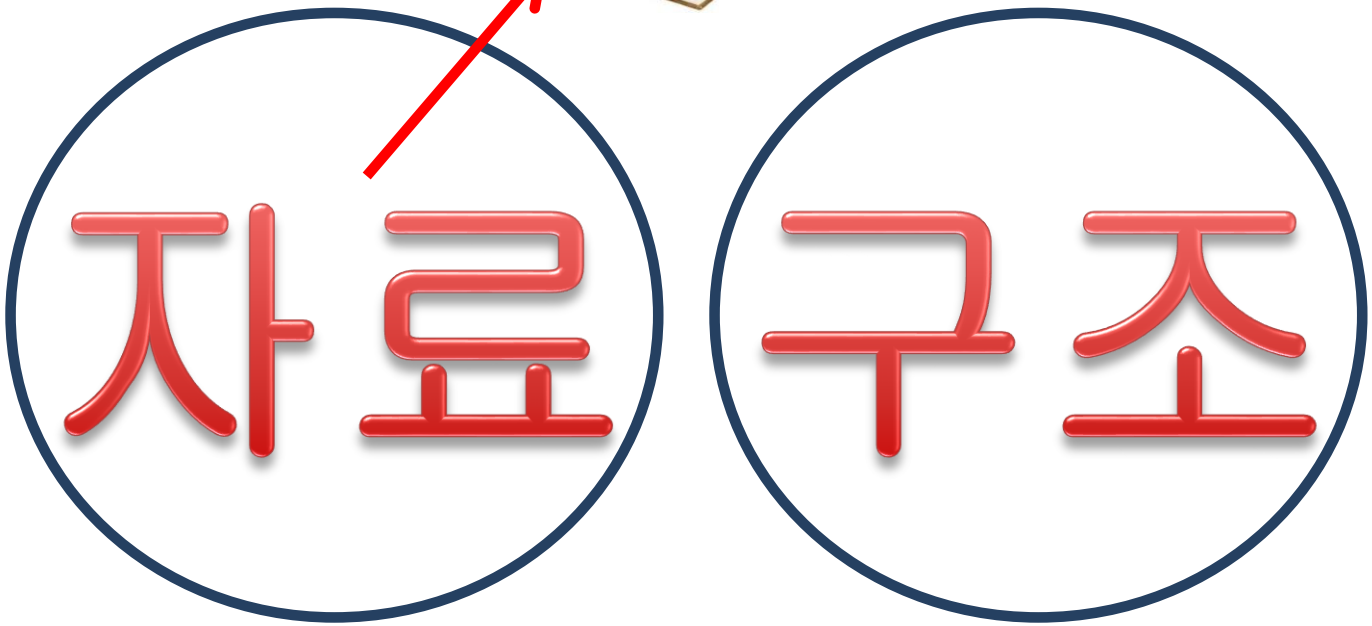
컬렉션(Collection)

KH정보교육원

메모리상에서 자료를 구조적으로 처리하는 방법을 **자료구조**라 일컫는다.

=> 자료를 쉽게 찾기 위해서 사용하는 방법

나열하는 방법



컬렉션(Collection)

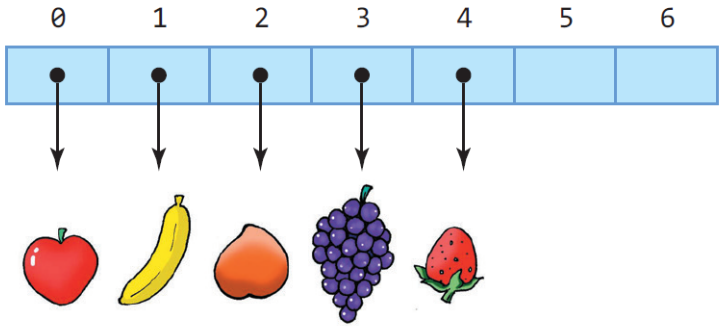
KH정보교육원

Collection이란?

- **컬렉션Collection**은 자바에서 제공하는 **자료구조**를 담당하는 **프레임워크**이다.
- 메모리상에서 자료를 구조적으로 처리하는 방법을 자료구조라 일컫는다.

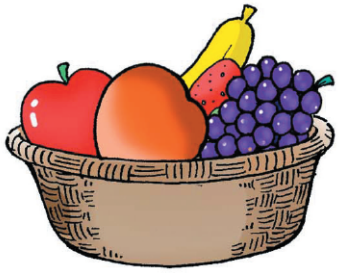
  => 자료를 쉽게 찾기 위해서 사용하는 방법
- 추가, 삭제, 정렬 등의 기능처리가 간단하게 해결 되어 자료구조적 알고리즘을 구현할 필요가 없다.

배열(array)



- 고정 크기 이상의 객체를 관리할 수 없다.
- 배열의 중간에 객체가 삭제되면 응용프로그램에서 자리를 옮겨야 한다.

컬렉션(collection)



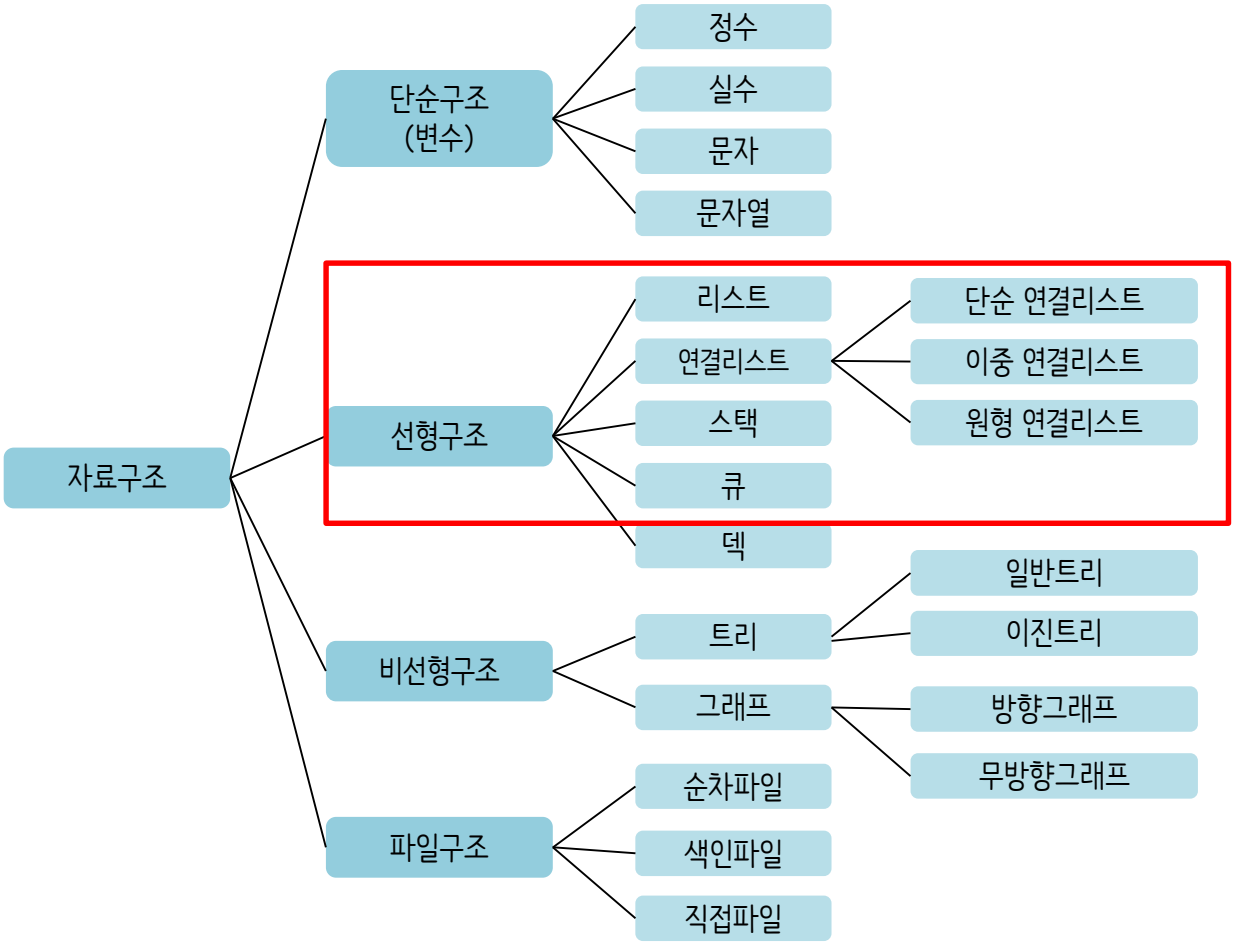
- 가변 크기로서 객체의 개수를 염려할 필요 없다.
- 컬렉션 내의 한 객체가 삭제되면 컬렉션이 자동으로 자리를 옮겨준다.

컬렉션(Collection)

KH정보교육원

자료구조란?

데이터(자료)를 메모리에서 구조적으로 처리하는 방법론이다.



컬렉션(Collection)

KH정보교육원

배열의 문제점

1. 한번 크기를 지정하면 변경할 수 없다.

- 공간의 크기가 부족하면 에러가 발생하기 때문에, 할당시 넉넉한 크기로 할당을 하게 된다.(메모리 낭비)
- 필요에 따라 늘리거나 줄일 수 없다.

2. 배열에 기록된 데이터에 대한 중간 위치의 추가, 삭제가 불편하다.

- 추가, 삭제할 데이터부터 마지막 기록된 데이터까지 하나씩 뒤로 밀어내고 추가해야 함.(알고리즘이 복잡하다)

3. 한 타입의 데이터만 저장 가능하다.

컬렉션(Collection)

KH정보교육원

컬렉션의 장점

1. 저장하는 크기의 제약이 없다.

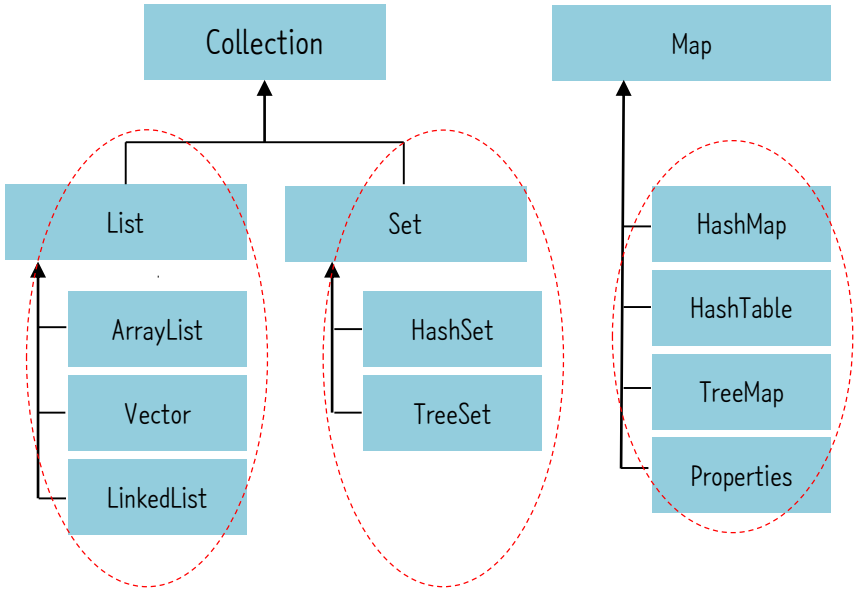
2. 추가, 삭제, 정렬 등의 기능처리가 간단하게 해결된다.
   - 자료를 구조적으로 처리하는 자료구조가 내장되어 알고리즘 구현이 필요 없다.

3. 여러 타입을 저장할 수 있다.
   - 객체만 저장할 수 있기 때문에, 필요에 따라 기본 자료형을 저장해야 하는 경우 Wrapper클래스를 사용한다.

컬렉션(Collection)

KH정보교육원

컬렉션의 주요 인터페이스



인터페이스 분류		특징	구현 클래스
Collection	List 계열	- 순서를 유지하고 저장 - 중복 저장 가능	ArrayList, Vector, LinkedList
	Set계열	-순서를 유지하지 않고 저장 - 중복 저장 안됨	HashSet, TreeSet
Map 계열		-키와 값의 쌍으로 저장 -키는 중복 저장 안됨(Set속성) -값은 중복저장가능(List속성)	HashMap, Hashtable, TreeMap, Properties

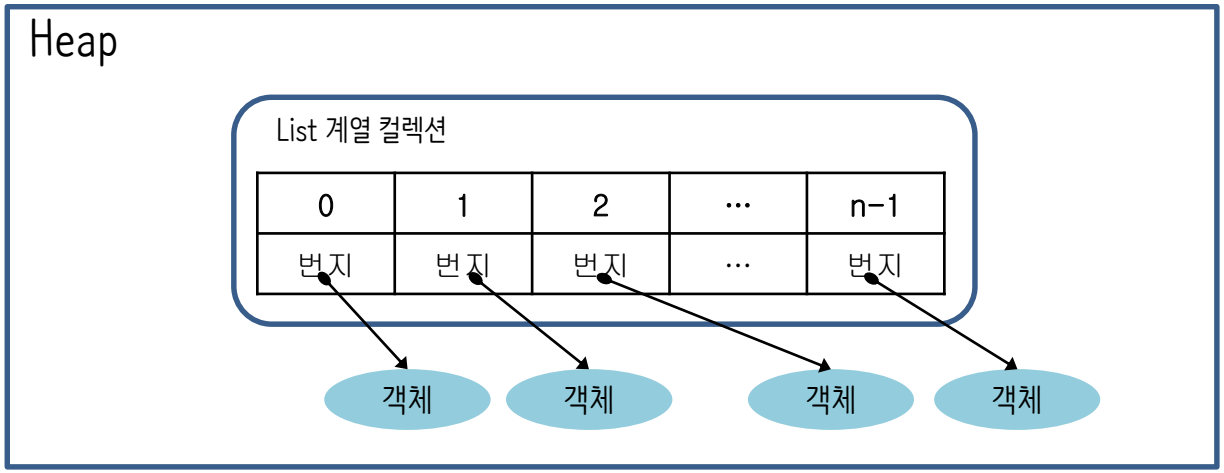
컬렉션(Collection)

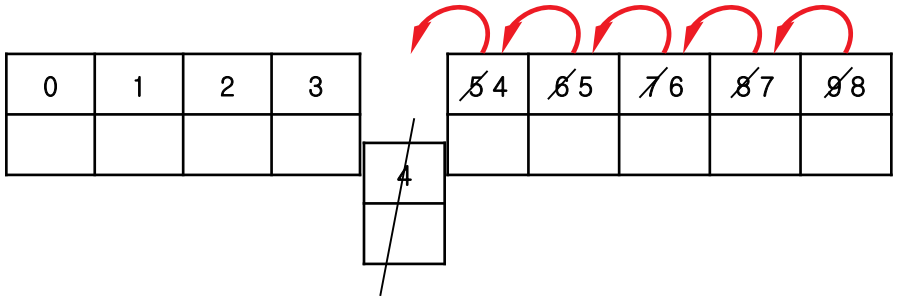
KH정보교육원

List

자료들을 순차적으로 나열한 자료 구조이다. 인덱스로 관리되며, 중복해서 객체 저장이 가능하다.

구현 클래스는 ArrayList와 Vector, LinkedList가 있다.





컬렉션(Collection)

KH정보교육원

List계열 주요 메소드

기능	메소드	설명
객체 추가	boolean add(E e)	주어진 객체를 맨 끝에 추가
	void add(int index, E element)	주어진 인덱스에 객체를 추가
	boolean addAll(Collection<? extends E> c)	주어진 Collection타입 객체를 리스트에 추가
	boolean addAll(int index, Collection<? extends E> c)	주어진 Collection타입 객체를 리스트 특정 위치에 추가
	set(int index, E element)	주어진 인덱스에 저장된 객체를 주어진 객체로 바꿈
객체 검색	boolean contains(Object o)	주어진 객체가 저장되어 있는지 여부
	boolean containsAll(Collection c)	주어진 Collection타입 객체가 리스트에 포함되는지 여부
	E get(int index)	주어진 인덱스에 저장된 객체를 리턴
	isEmpty()	컬렉션이 비어 있는지 조사
	int size()	저장되어 있는 전체 객체수를 리턴
	boolean equals(Object o)	주어진 객체가 리스트이고, 같은 사이즈, 상응하는 요소가 모두 동일하다면, true를 리턴.
	ListItrator<E> listIterator()	앞 또는 뒤부너 양방향 조회가 가능한 ListIterator리턴
객체 삭제	void clear()	저장된 모든 객체를 삭제
	E remove(int index)	주어진 인덱스에 저장된 객체를 삭제
	boolean remove(Object o)	주어진 객체를 삭제

컬렉션(Collection)

KH정보교육원

ArrayList

List의 후손으로, 초기 저장 용량은 10으로 자동 설정 되며 따로 지정도 가능하다. 저장 용량을 초과한 객체들이 들어오면 자동적으로 늘어나며, 고정도 가능하다. 동기화(Synchronization)를 제공하지 않는다.

예) List<E> list = new ArrayList<E>();

ArrayList

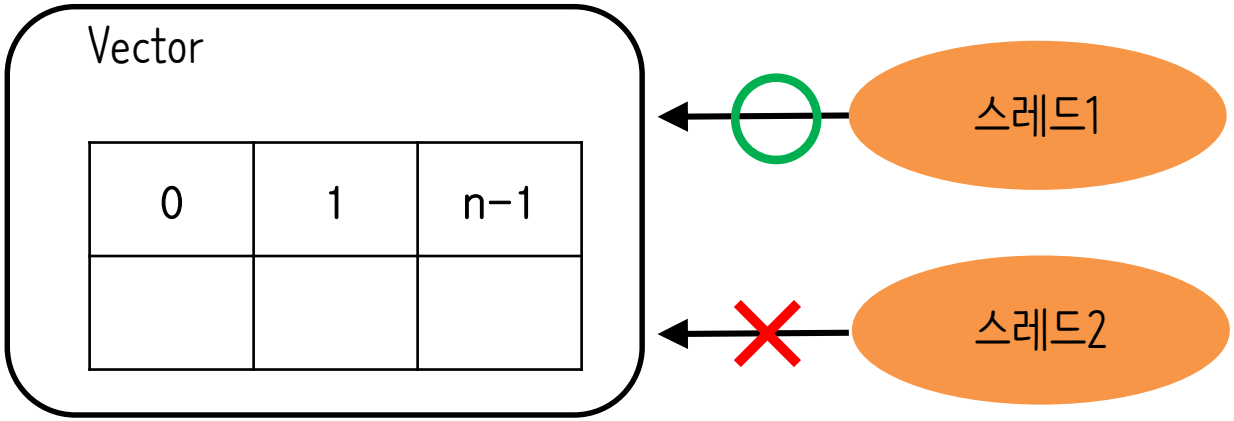
컬렉션(Collection)

KH정보교육원

Vector

List의 후손으로, 기본적으로 ArrayList와 동등하지만 동기화(Synchronize)를 제공한다는 점이 ArrayList와 차이점이다.

따라서 List 객체들 중에서 가장 성능이 좋지 않다.



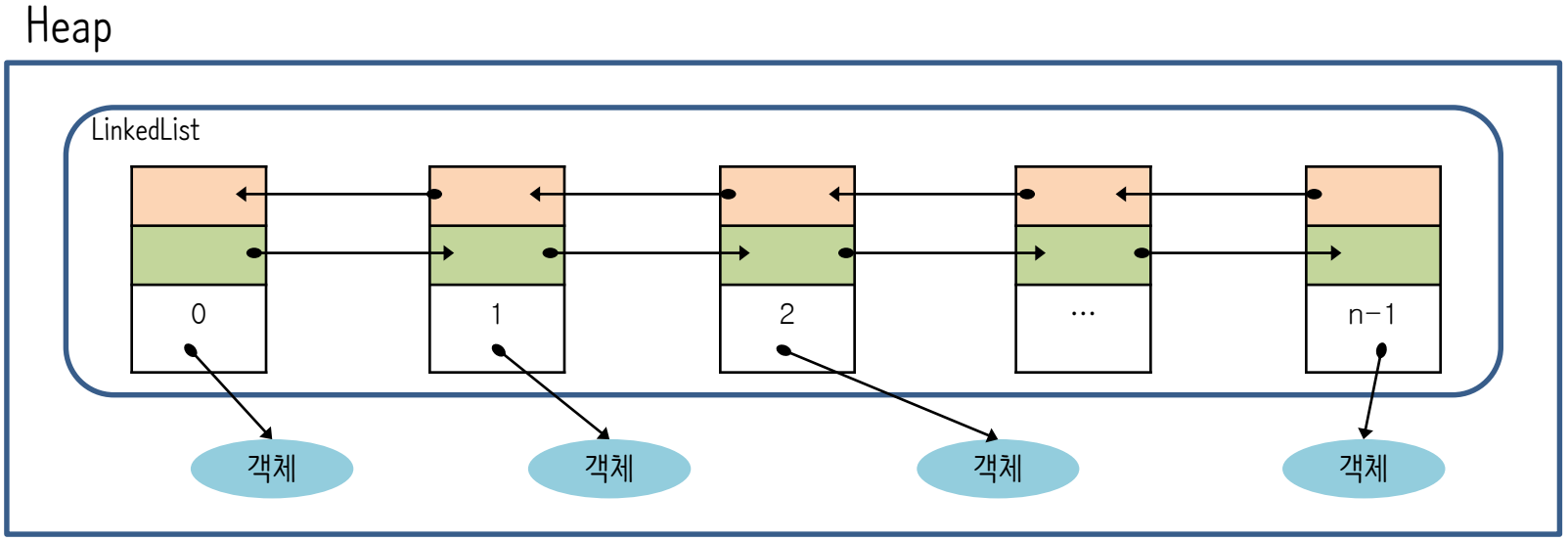
컬렉션(Collection)

KH정보교육원

LinkedList

List의 후손으로, 인접 참조를 링크해서 체인처럼 관리한다.

특정 인덱스에서 객체를 제거하거나 추가하게 되면 바로 앞/뒤 링크만 변경하면 되기 때문에 객체 삭제와 삽입이 빈번하게 일어나는 곳에서는 ArrayList보다 성능이 좋다.



컬렉션(Collection)

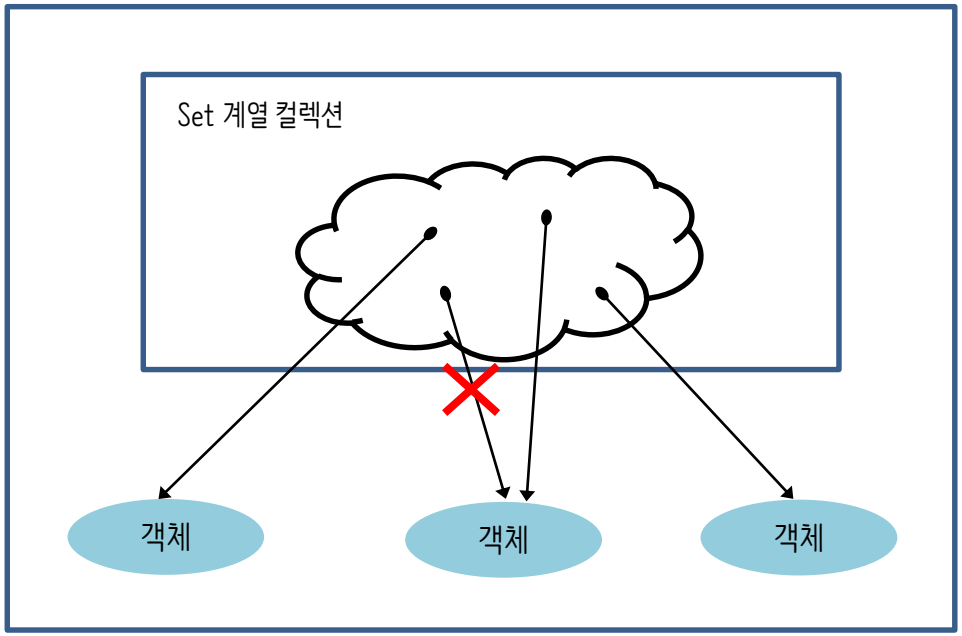
KH정보교육원

Set

저장 순서가 유지되지 않고, 중복 객체도 저장하지 못하게 하는 자료 구조이다. 수학적으로 비유하면 집합에 비유된다.

Null도 중복을 허용하지 않기 때문에 1개의 null만 저장이 된다.

구현 클래스는 HashSet, LinkedSet, TreeSet이 있다.



컬렉션(Collection)

KH정보교육원

Set계열 주요 메소드

기능	메소드	설명
객체 추가	boolean add(E e)	주어진 객체를 저장, 객체가 성공적으로 저장되면 true를 리턴하고 중복 객체이면 false를 리턴함
	boolean addAll(Collection<? extends E> c)	주어진 Collection객체를 현재 Set에 추가.
객체 검색	boolean contains(Object o)	주어진 객체가 저장되어 있는지 여부
	boolean containsAll(Collection<?> c)	주어진 Collection 객체가 현재 Set에 모두 포함되는지 여부
	boolean equals(Object o)	주어진 객체가 Set객체이고, 같은 크기, 가지고 있는 요소가 모두 같다면 ture를 리턴.
	boolean isEmpty()	컬렉션이 비어 있는지 조사
	Iterator<E> iterator()	저장된 객체를 한번씩 가져오는 반복자 리턴
	int size()	저장되어 있는 전체 객체수를 리턴
객체 삭제	void clear()	저장된 모든 객체를 삭제
	boolean remove(Object o)	주어진 객체를 삭제

전체 객체 대상으로 한번씩 반복해서 가져오는 반복자(Iterator)를 제공
인덱스로 객체에 접근할 수 없음

컬렉션(Collection)

KH정보교육원

HashSet

Set에 객체를 저장할 때 hash를 사용하여 처리 속도가 빠르다. 동일 객체 뿐 아니라 동등 객체도 중복하여 저장하지 않는다.

LinkedHashSet

hashSet과 거의 동일하지만 set에 추가되는 순서를 유지한다는 점이 다르다.

컬렉션(Collection)

KH정보교육원

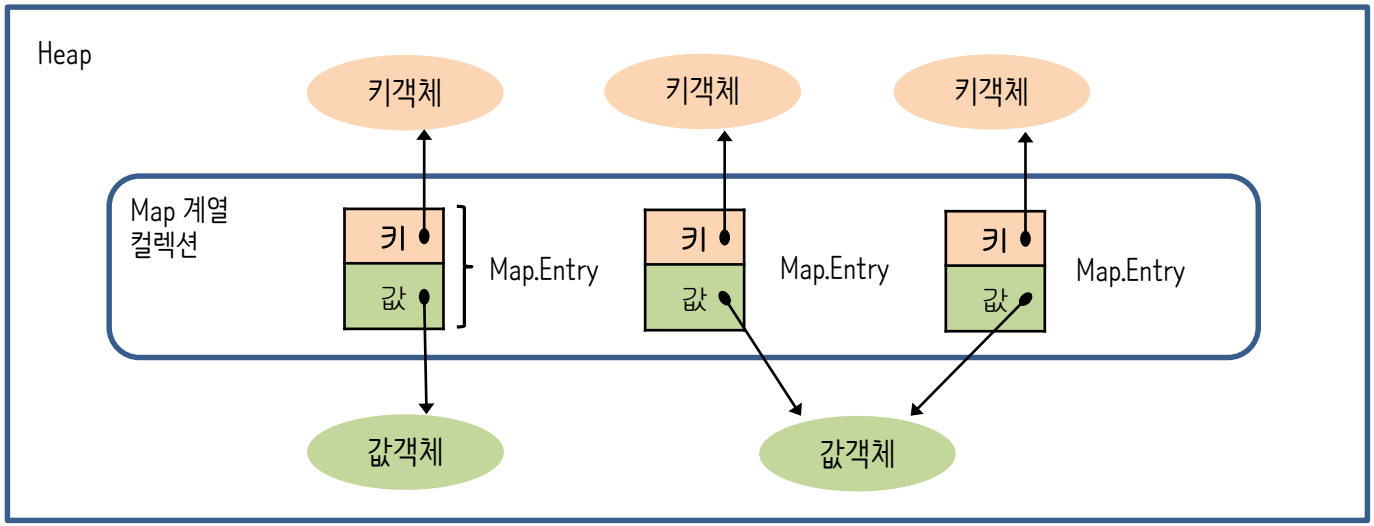
Map

키(key)와 값(value)으로 구성되어 있으며, 키와 값은 모두 객체이다.

키는 중복 저장을 허용하지 않고(set방식), 값은 중복 저장이 가능하다. (list방식)

키가 중복되는 경우에는 기존에 있는 키에 해당하는 값을 덮어 쓴다.

구현 클래스는 HashMap, HashTable, LinkedHashMap, Properties, TreeMap이 있다.



컬렉션(Collection)

KH정보교육원

Map계열 주요 메소드

기능	메소드	설명
객체 추가	V put(K key, V value)	주어진 키와 값을 추가, 저장이 되면 값을 리턴
	void putAll(Map<? extends K,? extends V> m)	주어진 맵객체를 모두 추가.
객체 검색	boolean containsKey(Object key)	주어진 키가 있는지 확인하여 결과 리턴
	boolean containsValue(Ovject value)	주어진 값이 있는지 확인하여 결과 리턴
	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 값을 Collection에 담아서 리턴
객체 삭제	void clear()	모든 Map.Entry를 삭제함
	V remove(Object key)	주어진 키와 일치하는 Map.Entry 삭제, 삭제가 되면 값을 리턴한다.

컬렉션(Collection)

KH정보교육원

HashMap

키 객체는 hachCode()와 equals()를 재정의해 동등 객체가 될 조건을 정해야 한다. 때문에 키 타입은 hashCode()와 equals()메소드가 재 정의 되어 있는 String타입을 사용한다.

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

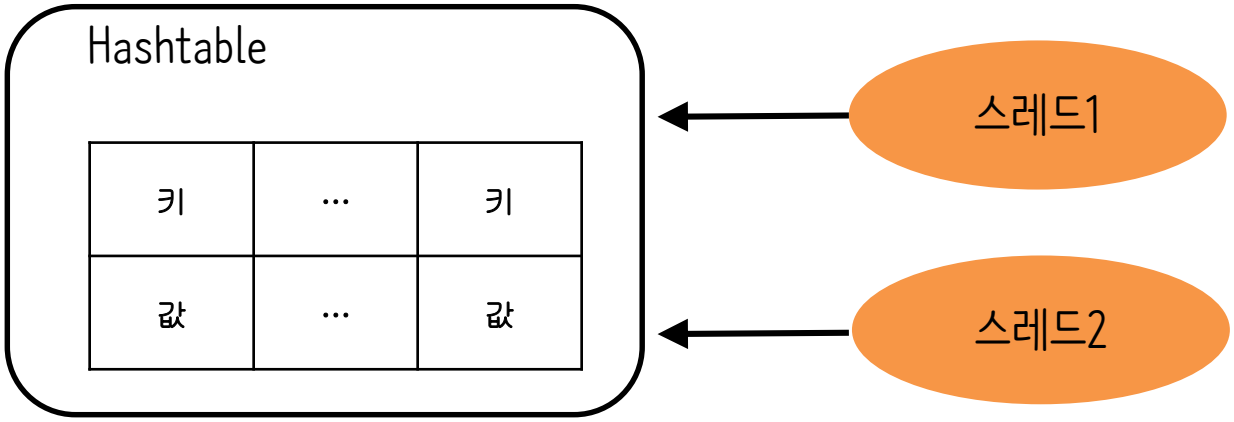
컬렉션(Collection)

KH정보교육원

HashTable

키 객체 만드는 법은 HashMap과 동일하다. 하지만 HashTable은 스레드 동기화(Synchronization)가 된 상태이기 때문에, 복수의 스레드가 동시에 HashTable에 접근해서 객체를 추가, 삭제 하더라도 스레드에 안전하다.
(Thread safe)

예) Map<K, V> map = new Hashtable<K, V>();



스레드 동기화 적용됨

컬렉션(Collection)

KH정보교육원

Properties

키와 값을 String 타입으로 제한한 Map 컬렉션이다.

주로 Properties는 프로퍼티(*.properties)파일을 읽어 들일 때 주로 사용된다.

프로퍼티(*.properties) 파일이란?

- 옵션정보, 데이터베이스 연결정보, 국제화(다국어)정보를 기록하여 텍스트 파일로 활용한다.
- 애플리케이션에서 주로 변경이 잦은 문자열을 저장하여 관리하기 때문에 유지보수를 편리하게 만들어 준다.
- 키와 값이 '='기호로 연결되어 있는 텍스트 파일이며, ISO 8859-1 문자셋으로 저장되고, 한글은 유니코드(Unicode)로 변환되어 저장된다.

컬렉션과 Wrapper 클래스

KH정보교육원

기본 자료형을 객체화 해주는 클래스이다.

기본 자료형	Wrapper Class	기본형으로 래퍼클래스 생성
boolean	Boolean	Boolean bool = new Boolean(true);
byte	Byte	Byte b = new Byte((byte)1);
char	Character	Character c = new Character('A');
short	Short	Short s = new Short((short)2);
int	Integer	Integer age = new Integer(30);
long	Long	Long l = new Long(100L);
float	Float	Float f =new Float(0.7f);
double	Double	Double d = new Double(0.75);

Wrapper 클래스 형변환

KH정보교육원

- 오토박싱(AutoBoxing)

  기본자료형 → Wrapper클래스 변환

  Integer num = new Integer(3);

  Interger num = 3;

  Double dnum = new Double(3.14);

  Double dnum = 3.14;

- 오토언박싱(Auto Unboxing)

  Wrapper클래스 → 기본자료형변환

  int n = num.intValue();

  int n = num;

  Double d = dnum.doubleValue();

  Double d = dnum;

String을 기본 자료형으로 바꾸기

KH정보교육원

```
byte b = Byte.parseByte("1");
short s = Short.parseShort("2");
int i = Integer.parseInt("3");
long l = Long.parseLong("4");
float f = Float.parseFloat("0.1");
double d = Double.parseDouble("0.2");
boolean b = Boolean.parseBoolean("true");

char c = "abc".charAt(0);
```

기본자료형을 String으로 바꾸기

KH정보교육원

```
String b = Byte.valueOf((byte)1).toString();
String s = Short.valueOf((short)2).toString();
String i = Integer.valueOf(3).toString();
String l = Long.valueOf(4L).toString();
String f = Float.valueOf(0.1f).toString();
String d = Double.valueOf(0.2).toString();
String b = Boolean.valueOf(true).toString();
String c = Character.valueOf('a').toString();
```

이밖에도 String.valueOf(1234), ""+1234 등 여러가지 방법이 있다.