

Collection





Collection



Collection

Collection

- 메모리상에서 자료를 구조적으로 처리하는 방법을 자료구조라 함
- Collection은 자바에서 제공하는 자료구조를 담당하는 Framework
- 추가, 삭제, 정렬 등의 기능처리가 간단하게 해결되어 자료구조적 알고리즘을 사용자가 별도로 구현할 필요가 없음



Collection

배열의 문제점

1. 한번 크기를 지정하면 변경할 수 없다.
 - 공간이 부족하면 에러가 발생하기 때문에, 할당 시 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 데이터 부터 마지막 데이터까지 하나씩 뒤로밀어내거나 당겨오는 작업을 진행하고 추가 및 삭제가 가능하다.
3. 한 타입의 데이터만 저장 가능하다.



Collection

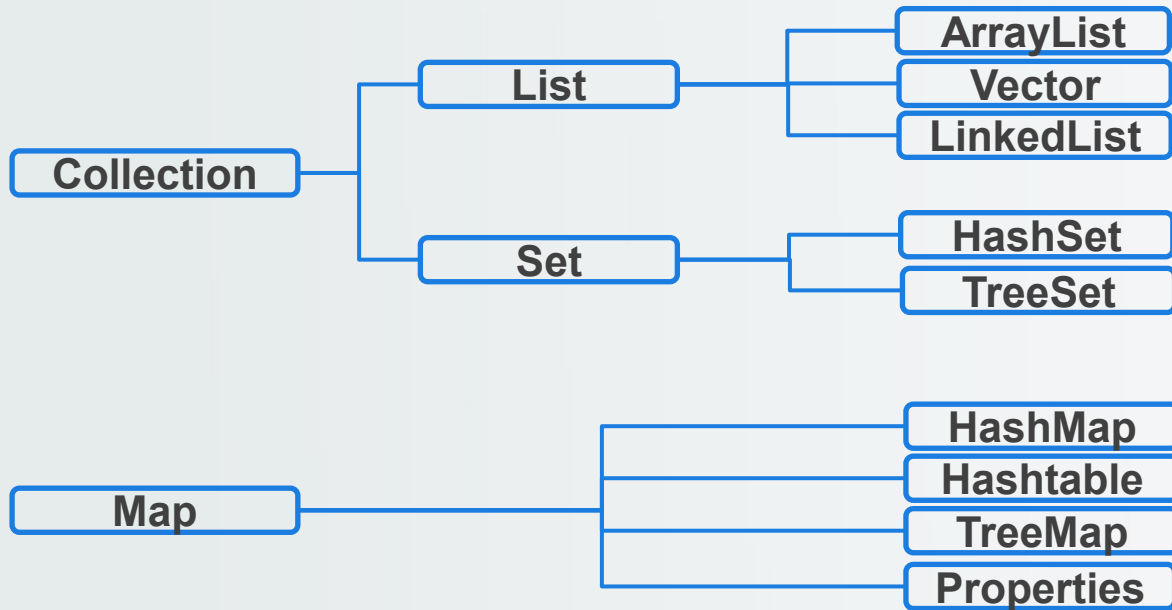
Collection의 장점

1. 저장하는 크기의 제약이 없다.
2. 추가, 삭제, 정렬 등의 기능처리가 간단하게 해결된다.
 - 자료를 구조적으로 처리하는 자료구조가 내장되어있다.
3. 여러 타입을 저장할 수 있다.
 - 객체타입만 저장 할 수 있기때문에, 상속과 다형성을 이용하여 여러타입을 저장 할 수 있다.
 - 기본자료형의 경우 Wrapper클래스를 사용한다.



Collection

Collection의 주요 인터페이스



Collection

Collection의 주요 인터페이스

인터페이스 분류		특징	구현 클래스
Collection	List	<ul style="list-style-type: none">- 순서를 유지하고 저장- 중복 저장 가능	<ul style="list-style-type: none">- ArrayList- Vector- LinkedList
	Set	<ul style="list-style-type: none">- 순서를 유지하지 않고 저장- 중복 저장 불가	<ul style="list-style-type: none">- HashSet- TreeSet
Map 계열		<ul style="list-style-type: none">- 키와 값을 쌍으로 저장- 키는 중복 저장 불가- 값은 중복 저장 가능	<ul style="list-style-type: none">- HashMap- Hashtable- TreeMap- Properties





List



Collection

List

- 자료들을 순차적으로 나열한 자료구조
- 인덱스로 관리되며, 중복해서 객체 저장이 가능
- 배열과 가장 유사한 형태를 띄고 있다.
- 구현클래스 ArrayList, Vector, LinkedList가 존재



Collection

List 계열 주요 메소드 - 1

기능	메소드	리턴타입	설명
객체 추가	add(E e)	boolean	주어진 객체를 맨 끝에 추가
	add(int index, E element)	void	주어진 인덱스에 객체 추가
	addAll(Collection c)	boolean	주어진 Collection 타입 객체 추가
	addAll(int index, Collection c)	boolean	주어진 Collection 타입 객체를 주어진 인덱스에 추가
	set(int index, E element)	Generic	주어진 인덱스에 저장된 객체를 변경하고 변경 전 값 리턴



Collection

List 계열 주요 메소드 - 2

기능	메소드	리턴타입	설명
객체 검색	contains(Object o)	boolean	주어진 객체가 저장되어 있는지 확인
	containsAll(Collection c)	boolean	주어진 Collection타입 객체가 있는지 확인
	get(int index)	Generic	주어진 인덱스에 저장된 객체 리턴
	isEmpty()	boolean	컬렉션이 비어 있는지 확인
	size()	int	저장 되어 있는 객체 수 리턴
	listIterator	ListIterator	앞 또는 뒤부터 양방향 조회가 가능한 객체 리턴



Collection

List 계열 주요 메소드 - 3

기능	메소드	리턴타입	설명
객체 삭제	clear()	void	저장된 모든 객체 삭제
	remove(int index)	Generic	주어진 인덱스의 객체를 삭제하며 삭제된 객체 리턴
	remove(Object o)	boolean	주어진 객체를 삭제





Set



Collection

Set

- 저장 순서가 유지되지 않고, 중복 객체도 저장하지 못하는 자료구조
- 수학으로 비유하자면 집합
- 구현클래스 HashSet, TreeSet, LinkedSet이 존재



Collection

Set 계열 주요 메소드 - 1

기능	메소드	리턴타입	설명
객체 추가	add(E e)	boolean	주어진 객체 저장하고 결과를 리턴 중복인 경우 저장실패 후 false 리턴
	addAll(Collection c)	boolean	주어진 Collection 타입 객체 추가



Collection

Set 계열 주요 메소드 - 2

기능	메소드	리턴타입	설명
객체 검색	contains(Object o)	boolean	주어진 객체가 저장되어 있는지 확인
	containsAll(Collection c)	boolean	주어진 Collection타입 객체가 있는지 확인
	iterator()	iterator	저장된 객체를 한번씩 가져오는 반복자 리턴
	isEmpty()	boolean	컬렉션이 비어 있는지 확인
	size()	int	저장 되어 있는 객체 수 리턴

※ 인덱스가 없어 넣어둔 데이터에 대한 접근 방법이 없으므로, 전체 객체를 순회하는 반복자를 이용



Collection

Set 계열 주요 메소드 - 3

기능	메소드	리턴타입	설명
객체 삭제	clear()	void	저장된 모든 객체 삭제
	remove(Object o)	boolean	주어진 객체를 삭제





Map



Collection

Map

- 키(key)와 값(value)으로 구성되어 있으며 키와 값 모두 객체타입
- 키는 중복 저장 불가(Set의 특성)
- 값은 중복 저장 가능(List의 특성)
- 중복된 키가 들어오는 경우 기존의 키에 해당하는 값에 덮어 쓴다.
- 구현 클래스는 HashMap, Hashtable, LinkedHashMap, Properties, TreeMap이 존재

※ 키를 중복해서 넣으면 덮어쓰기가 되므로 이전데이터가 사라짐



Collection

Map 계열 주요 메소드 - 1

기능	메소드	리턴타입	설명
객체 추가	put(K key, V value)	V	주어진 키와 값을 추가, 저장되면 값을 리턴
	putAll(Map)	void	주어진 Map 타입 객체 추가



Collection

Map 계열 주요 메소드 - 2

기능	메소드	리턴타입	설명
객체 검색	containsKey(Object key)	boolean	주어진 키가 있는지 확인하고 결과 리턴
	containsValue(Object value)	boolean	주어진 값이 있는지 확인하고 결과 리턴
	isEmpty()	boolean	컬렉션이 비어 있는지 확인
	get(Object key)	Generic V	주어진 키에 해당하는 값을 리턴
	size()	int	저장 되어 있는 키의 수 리턴
	keyset()	Set<K>	모든 키를 Set에 담아 리턴
	Values()	Collection<V>	저장된 모든 값을 Collection에 담아 리턴

Collection

Map 계열 주요 메소드 - 3

기능	메소드	리턴타입	설명
객체 삭제	clear()	void	저장된 모든 객체 삭제
	remove(Object key)	Generic V	주어진 키와 일치하는 Map Entry를 삭제하고 삭제되면 해당 값 리턴

