

REPORT



과 목 명 : 디자인패턴

담당교수 : 박제호 교수님

소 속 : 소프트웨어학과

학 번 : 32151671

이 름 : 박민혁



단국대학교
Dankook University

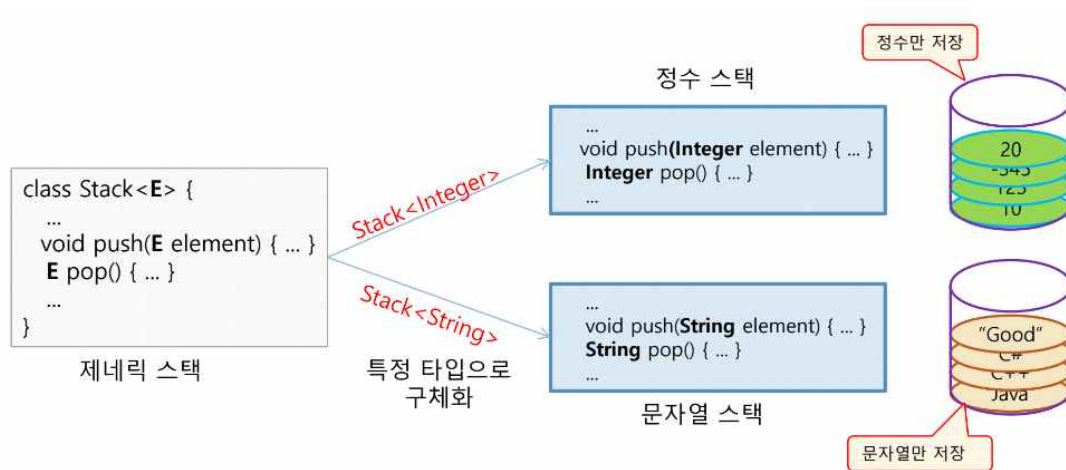
Chapter 7. 제네릭과 컬렉션

컬렉션

- 요소 객체들의 저장소
- 요소의 개수에 따라 크기 자동 조절
- 제네릭 기법으로 구현
- 컬렉션의 요소는 객체만 가능

제네릭

- 특정 타입만 다루지 않고, 여러 종류의 타입으로 변신할 수 있도록 클래스나 메소드를 일반화 시키는 기법
- <E>, <K>, <V> : 타입 매개 변수
- 제네릭 스택 : Stack<E>



Vector<E>

- 여러 객체들의 삽입, 삭제, 검색하는 컨테이너 클래스

메소드	설명
boolean add(E element)	벡터의 맨 뒤에 element 추가
void add(int index, E element)	인덱스 index에 element를 삽입
int capacity()	벡터의 현재 용량 리턴
boolean addAll(Collection<? extends E> c)	컬렉션 c의 모든 요소를 벡터의 맨 뒤에 추가
void clear()	벡터의 모든 요소 삭제
boolean contains(Object o)	벡터가 지정된 객체 o를 포함하고 있으면 true 리턴
E elementAt(int index)	인덱스 index의 요소 리턴
E get(int index)	인덱스 index의 요소 리턴
int indexOf(Object o)	o와 같은 첫 번째 요소의 인덱스 리턴, 없으면 -1 리턴
boolean isEmpty()	벡터가 비어 있으면 true 리턴
E remove(int index)	인덱스 index의 요소 삭제
boolean remove(Object o)	객체 o와 같은 첫 번째 요소를 벡터에서 삭제
void removeAllElements()	벡터의 모든 요소를 삭제하고 크기를 0으로 만들
int size()	벡터가 포함하는 요소의 개수 리턴
Object[] toArray()	벡터의 모든 요소를 포함하는 배열 리턴

ArrayList<E>

- java.util.ArrayList, 가변 크기 배열을 구현한 클래스
- 벡터와 달리 스레드 동기화 기능이 없음

메소드	설명
boolean add(E element)	ArrayList의 맨 뒤에 element 추가
void add(int index, E element)	인덱스 index 위치에 element 삽입
boolean addAll(Collection<? extends E> c)	컬렉션 c의 모든 요소를 ArrayList의 맨 뒤에 추가
void clear()	ArrayList의 모든 요소 삭제
boolean contains(Object o)	ArrayList가 지정된 객체를 포함하고 있으면 true 리턴
E elementAt(int index)	index 인덱스의 요소 리턴
E get(int index)	index 인덱스의 요소 리턴
int indexOf(Object o)	o와 같은 첫 번째 요소의 인덱스 리턴, 없으면 -1 리턴
boolean isEmpty()	ArrayList가 비어있으면 true 리턴
E remove(int index)	index 인덱스의 요소 삭제
boolean remove(Object o)	o와 같은 첫 번째 요소를 ArrayList에서 삭제
int size()	ArrayList가 포함하는 요소의 개수 리턴
Object[] toArray()	ArrayList의 모든 요소를 포함하는 배열 리턴

Iterator<E>

- Vector<E>, ArrayList<E>, LinkedList<E>가 상속받는 인터페이스

메소드	설명
boolean hasNext()	방문할 요소가 남아 있으면 true 리턴
E next()	다음 요소 리턴
void remove()	마지막으로 리턴된 요소 제거

HashMap<K,V>

- 키와 값의 쌍으로 구성되는 요소를 다루는 컬렉션
- K는 키로 사용할 요소의 타입, V는 값으로 사용할 요소의 타입 지정
- 키와 값이 한 쌍으로 삽입
- 키는 해시맵에 삽입되는 위치 결정에 사용
- 값을 검색하기 위해서는 반드시 키 이용
- 요소 삽입 : put() 메소드
- 요소 검색 : get() 메소드

메소드	설명
<code>void clear()</code>	해시맵의 모든 요소 삭제
<code>boolean containsKey(Object key)</code>	지정된 키(key)를 포함하고 있으면 true 리턴
<code>boolean containsValue(Object value)</code>	지정된 값(value)에 일치하는 키가 있으면 true 리턴
<code>V get(Object key)</code>	지정된 키(key)의 값 리턴, 키가 없으면 null 리턴
<code>boolean isEmpty()</code>	해시맵이 비어 있으면 true 리턴
<code>Set<K> keySet()</code>	해시맵의 모든 키를 담은 Set<K> 컬렉션 리턴
<code>V put(K key, V value)</code>	key와 value 쌍을 해시맵에 저장
<code>V remove(Object key)</code>	지정된 키(key)를 찾아 키와 값 모두 삭제
<code>int size()</code>	HashMap에 포함된 요소의 개수 리턴

LinkedList<E>

- E에 요소로 사용할 타입 지정하며 구체화
- List 인터페이스를 구현한 컬렉션 클래스
- 위의 두 개 클래스와 매우 유사하게 작동
- 요소 객체들은 양 방향으로 연결 되어 관리 됨
- 요소 객체는 맨 앞, 맨 뒤에 추가 가능
- 요소 객체는 인덱스를 이용하여 중간에 삽입 가능
- 맨 앞이나 맨 뒤에 요소를 추가하거나 삭제할 수 있어 스택이나 큐로 사용 가능

타입 매개 변수

- E : Element를 의미하며 컬렉션에서 요소를 표시할 때 많이 사용한다.
- T : Type를 의미한다.
- V : Value를 의미한다.
- K : Key를 의미한다.
- 타입 매개변수가 나타내는 타입의 객체 생성 불가
- 어떤 문자도 매개 변수로 사용 가능