

Objects 클래스

Objects 클래스

❖ Objects 클래스

○ Object의 유틸리티 클래스

리턴타입	메소드(매개변수)	설명
int	compare(T a, T b, Comparator<T> c)	두 객체 a 와 b 를 Comparator 를 사용해서 비교
boolean	deepEquals(Object a, Object b)	두 객체의 깊은 비교(필드도 비교)
boolean	equals(Object a, Object b)	두 객체의 얕은 비교(번지만 비교)
int	hash(Object... values)	매개값이 저장된 배열의 해시코드 생성
int	hashCode(Object o)	객체의 해시코드 생성
boolean	isNull(Object obj)	객체가 널 인지 조사
boolean	nonNull(Object obj)	객체가 널이 아닌지 조사
T	requireNonNull(T obj)	객체가 널인 경우 예외 발생
T	requireNonNull(T obj, String message)	객체가 널인 경우 예외 발생(주어진 예외 메시지 포함)
T	requireNonNull(T obj, Supplier<String> messageSupplier)	객체가 널인 경우 예외 발생(람다식이 만든 예외 메시지 포함)
String	toString(Object o)	객체의 toString() 리턴값 리턴
String	toString(Object o, String nullDefault)	객체의 toString() 리턴값 리턴, 첫번째 매개값이 null 일 경우 두번째 매개값 리턴

Objects 클래스

❖ 객체 비교

- `Objects.compare(T a, T b, Comparator<T> c)`
 - `a, b` 두 객체를 비교자(`c`)로 비교해 `int`값 리턴
 - 논리적으로 `a-b`의 결과 리턴
 - 음수 : `a < b`
 - 0 : `a == b`
 - 양수 : `a > b`
- `Comparator<T>` 인터페이스
 - 제너릭 인터페이스 타입
 - `T` 타입의 객체를 비교하는 `compare(T a, T b)` 메소드 가짐

```
public interface Comparator<T> {  
    int compare(T a, T b)  
}
```

Objects 클래스

❖ 동등 비교(equals())와 deepEquals()

- 두 객체의 동등 비교
- `Objects.equals(Object a, Object b)`

a	b	<code>Objects.equals(a, b)</code>
not null	not null	<code>a.equals(b)</code>
null	not null	false
not null	null	false
null	null	true

- `deepEquals(Object a, Object b)`
 - 비교할 객체가 배열일 경우 항목 값까지도 비교

a	b	<code>Objects.deepEquals(a, b)</code>
not null (not array)	not null (not array)	<code>a.equals(b)</code>
not null (array)	not null (array)	<code>Arrays.deepEquals(a, b)</code>
not null	null	false
null	not null	false
null	null	true

❖ 해시코드 생성(hash(), hashCode())

○ Objects.hash(Object... values)

- 매개값으로 주어진 값들 이용해 해시 코드 생성하는 역할
- Arrays.hashCode(Object[]) 호출해 해시코드 얻어 리턴
- 클래스의 hashCode()의 리턴값 생성할 때 유용하게 사용

```
@Override  
public int hashCode() {  
    return Objects.hash(field1, field2, field3);  
}
```

○ Objects.hashCode(Object o)

- o.hashCode() 호출하고 받은 값 리턴
- 매개값이 null 이면 0 리턴

❖ 널 여부 조사(isNull(), nonNull(), requireNonNull())

- `Objects.isNull(Object obj)`
 - `obj`가 `null`일 경우 `true`
- `Objects.nonNull(Object obj)`
 - `obj`가 `not null`일 경우 `true`
- `requireNonNull()`

리턴타입	메소드(매개변수)	설명
T	<code>requireNonNull(T obj)</code>	<code>not null</code> → <code>obj</code> <code>null</code> → <code>NullPointerException</code>
T	<code>requireNonNull(T obj, String message)</code>	<code>not null</code> → <code>obj</code> <code>null</code> → <code>NullPointerException(message)</code>
T	<code>requireNonNull(T obj, Supplier<String> msgSupplier)</code>	<code>not null</code> → <code>obj</code> <code>null</code> → <code>NullPointerException(msgSupplier.get())</code>

Objects 클래스

❖ 객체 문자정보(toString())

- 객체의 문자정보 리턴

```
toString(Object obj);  
toString(Object obj, String nullDefault);
```

- 첫 번째 매개값이 not null
 - toString()으로 얻은 값을 리턴
- null이면
 - "null" 또는 두 번째 매개값인 nullDefault 리턴