

## • 11651번 : 좌표 정렬하기 2

Comparator를 사용한 정렬

```
Arrays.sort(ps, new Comparator<Position>() {  
    @Override  
    public int compare(Position o1, Position o2) {  
        if(o1.y == o2.y)  
            return Integer.compare(o1.x, o2.x);  
        return Integer.compare(o1.y, o2.y);  
    }  
});
```

**Comparable** : 객체 간의 일반적인 정렬이 필요할 때,  
Comparable 인터페이스를 확장해서 정렬의 기준을 정의  
하는 compareTo() 메서드를 구현한다.

**Comparator** : 객체 간의 특정한 정렬이 필요할 때,  
Comparator 인터페이스를 확장해서 특정 기준을 정의하  
는 compare() 메서드를 구현한다.

공통점은 정렬의 기준을 정의한다는 것이고,

차이점은 정렬 기준이 일반적이냐 일반적이지 않냐와  
compareTo(Object o) 메서드를 구현하느냐  
compare(Object o1, Object o2) 메서드를 구현하느냐가  
전부다.



# Comparable의 예

```
@Override
public int compareTo(Employee o) {
    return this.name.compareTo(o.name);
    //return this.id - o.id;
    //return this.department.compareTo(
}
```

주요하게 볼 것은 일반적인 기준이 되기 때문에 클래스에 implements로 Comparable을 확장했다는 것이고,

compareTo() 메서드를 알아보면 되는 것이다.

compareTo()는 int 타입을 반환하며, 파라미터로 비교 대상이 되는 객체가 들어오게 된다.

return하는 int 값은 기준이 있다.

A.compareTo(B) 일 때  $A < B$  인 경우는 음수를 리턴하고,  $A = B$  일 때는 0을 리턴하고,  $A > B$  일 때 양수를 리턴한다.

즉, 정렬에 있어서 앞에 오고 싶을 때 음수를 내보내면 된다.

\* 주의해야할 점은 아무 음수, 아무 양수를 내보내면 되는 것이 아니다. 마찬가지로 -1, 0, 1만 사용하는 것이 아니다.

위의 예시 클래스 파일에서는 Emplyoee들간에 정렬할 때 기본적으로 이름순으로 정했다.

물론 String 클래스가 compareTo를 구현하고 있어서 마찬가지로 사용했지만 만약 사번순으로 기준을 잡는다면 아래 주석처리한 것 처럼 this.id - o.id 로 하면 된다.



# Comparator의 예

Comparator는 일반적이지 않은 문자열의 길이 순으로 보고 싶다든지, Comparable로 구현한 것 말고 기준으로 정렬하고 싶다든지 할 때 사용한다.

예를 들어서 위의 예시에서 Comparable로 이름순으로 해놨다고 치자. 그런데 나는 연봉순으로 보고싶다든지 부서명으로 보고싶다든지할 수가 있다.

이럴때마다 compareTo를 수정할 수 없고 런타임중에는 심지어 불가능한 일이다.

대신 이럴 때 Comparator를 사용하면 정렬이 필요할 때 특수한 기준을 줄 수 있다.

```
//익명의 Comparator를 만듦
Comparator<Employee> salesComparator = new Comparator<Employee>() {
    @Override
    public int compare(Employee o1, Employee o2) {
        return o2.getSales().intValue() - o1.getSales().intValue();
    }
};
Collections.sort(list, salesComparator);
```

\* 일반적으로 Comparator를 만들 때 위의 예제처럼 익명 클래스를 만들어서 사용한다.

왜냐하면 Comparator를 사용하는것 자체가 그 때 그 때마다 정렬 기준이 바뀔 수 있는 것이기 때문이다.

머릿속에 정리하면 Comparable은 일반적인 정렬,  
Comparator는 커스터마이즈(customize) 정렬에 사용되  
고 "비교"가 아니다. 끝.