

Comparator와 Comparable

🕒 작성일시	@2023년 4월 18일 오전 2:38
📄 강의 번호	자바의정석 Chapter11
📄 유형	
📎 자료	
☑ 복습	<input type="checkbox"/>
☰ Spring Framework	

- Arrays.sort()를 호출만 하면 컴퓨터가 알아서 배열을 정렬하는 것처럼 보이지만 사실은 Character클래스의 Comparable의 구현에 의해 정렬된 것이다.
- Comparable 과 Comparator는 모두 인터페이스로 컬렉션을 정렬하는데 필요한 메서드를 정의하고 있으며, Comparable을 구현하고 있는 클래스들은 같은 타입의 인스턴스끼리 비교할 수 있는 클래스들, 주로 Integer와 같은 wrapper클래스와 String, Date, File과 같은 것들이다.
- 기본적으로 오름차순으로 구현되어 있다.
- 그래서 Comparable을 구현한 클래스는 정렬이 가능하다는 것을 의미한다.

▼ Comapre()와 CompareTo()

- 선언형태와 이름이 약간 다를 뿐 두 객체를 비교한다는 같은 기능을 목적으로 고안된 것이다.
-

```
public interface Comparator {  
    int compare(Object o1, Object o2);  
    boolean equals(Object obj);  
}  
  
public interface Comparable{  
    public int compareTo(Object o);  
}
```

▼ Comparable과 Comparator

- Comparable : 기본 정렬기준을 구현하는데 사용
- Comparator : 기본 정렬기준 외에 다른 기준으로 정렬하고자할 때 사용

▼ 예시 코드

```

package ch11;
import java.util.*;

public class ComparatorEx {
    public static void main(String[] args) {

        String[] strArr= {"cat","dog","lion","tiger"};

        Arrays.sort(strArr);
        System.out.println("strArr =" + Arrays.toString(strArr));

        Arrays.sort(strArr, String.CASE_INSENSITIVE_ORDER); //대소문자 구분안함
        System.out.println("strArr =" + Arrays.toString(strArr));

        Arrays.sort(strArr, new Descending());
        System.out.println("strArr =" + Arrays.toString(strArr));
    }
}

class Descending implements Comparator {
    public int compare(Object o1, Object o2) {
        if( o1 instanceof Comparable && o2 instanceof Comparable) {
            Comparable c1 = (Comparable)o1; //compare의 매개변수가Object이기 때문에 Comparable로 형변환해야 한다.
            Comparable c2 = (Comparable)o2;
            return c1.compareTo(c2) *-1 ; // 역순 정렬
        }
        return -1;
    }
}

```