



Interfaces III

Comparator



If I want to sort objects which are not comparable?

Luckily for us, there is a second version of `sort()` in Arrays. This one takes two arguments. The first is a reference to an array again, and the second argument is an object which knows how to compare two objects in the array.

The second argument must be of type `Comparator` (which is an interface).

The only method *you need to* implement is...

```
public interface Comparator<T>
```

```
...
```

```
public int compare(T o1, T o2)
```

Implementing a Comparator for CDs

Let's implement a comparator for a class representing CDs (you know, these plastic discs from the 90s which can contain music for instance).

The CD class

```
public class CD{
    private String title;
    private String year;
    public CD(String title, String year){
        this.title=title;
        this.year=year;
    }
    public String title(){return title;}
    public String year(){return year;}

    @Override
    public String toString(){
        return String.format("%s, %s",title, year);
    }
}
```

CDYearComparator

Let's write a comparator which only cares about the year of the CD, using the Comparator interface (which has one method you must override):

```
public class CDYearComparator implements
    java.util.Comparator<CD>{
    @Override
    public int compare(CD first, CD other){
        return first.year().compareTo(other.year());
        // we know that Strings are comparable ;- )
    }
}
```

A closer look at CDYearComparator

```
public class CDYearComparator implements
        java.util.Comparator<CD>{
    @Override
    public int compare(CD first, CD other){
        return first.year().compareTo(other.year());
        // we know that Strings are comparable ;- )
    }
}
```

Now we can sort CDs

The CD class does not implement the Comparable interface, so we can't use the first version of Arrays.sort() but since we created a "cd comparator" class, we can use the second version of Arrays.sort():

```
CD first= new CD("Violator", "1990");
CD second = new CD("On the beach", "1969");
CD[] cds = new CD[2];
cds[0]=first;
cds[1]=second;
System.out.println(java.util.Arrays.toString(cds));
java.util.Arrays.sort(cds,new CDYearComparator());
System.out.println(java.util.Arrays.toString(cds));
// Will print:
[Violator, 1990, On the beach, 1969] # not sorted
[On the beach, 1969, Violator, 1990] # sorted on year
```