CS601: Principles of Software Development

Interfaces. Comparator. Iterator.

Olga A. Karpenko

Parts of this presentations are based on the Java Solftware Solutions book by Lewis&Loftus.

Announcements

- Lab0 is due tonight
- Must be submitted via github
- Tuo's office hours today: 2:15-4:15
- My office hours today: 5:40-6:40pm
- International Orientation today!

  3:30pm HR 136

Comparable

- Built in Interface in Java
- Any class can implement Comparable
  - to provide a mechanism for comparing objects of that type

```
MyClass obj1, obj2;
// TODO: ... initialize obj1, obj2

if (obj1.compareTo(obj2) < 0)
  System.out.println ("obj1 is less than obj2");
```

Comparable

- The value returned from compareTo should be:
  - negative if obj1 < obj2,
  - 0 if obj1 == obj2
  - positive if obj1 > obj2

Comparable

- It's up to the programmer to determine what makes one object less than another
- Examples:
  - Compare students based on GPA
  - Compare Employees based on salary
  - Compare books based on titles

Example

- See classes `Student` and `Driver`
- We can compare students based on the GPA or based on the name
- Can sort a list of Students if Student class implements Comparable

Comparator

Comparator: Motivation

```
public class Rectangle implements Comparable<Rectangle> {
    private int x, y, width, height;

    public int compareTo(Rectangle other) {
            // what to compare them based on?
    }
}
```

What is the "natural ordering" of rectangles?

- By x (and if x-s are equal, by y?)
- By width? By area? By perimeter?

What if we want to compare based on multiple different criteria?

Reference: https://courses.cs.washington.edu/.../slides/04a-compare.ppt

Comparator Interface

- `int compare(Object o1, Object o2)`

- Put comparison method in a separate class
- Can use different Comparators to compare objects using different criteria

Example

```
public class RectangleComparator1
             implements Comparator<Rectangle> {

  public int compare(Rectangle r1, Rectangle r2){
     // compare rectangles by area
  }

}
```

Example

```
public class RectangleComparator2
             implements Comparator<Rectangle> {

  public int compare(Rectangle r1, Rectangle r2){
     // compare rectangles by perimeter
  }

}
```

Using Comparator

```
Comparator<Rectangle> comp = new RectangleComparator1();

Set<Rectangle> set = new TreeSet<Rectangle>(comp);
```

Example

- Create two Comparators for class Student:
  - one for comparing based on the name
  - one based on the GAP

Exercise

- Write class Rectangle that has width and height
- Implement two Comparators: one for area, one for perimeter
- Create 2 TreeSets with two Rectangles: one for each Comparator
- Print the result