

CSE 143: Computer Programming II

QuickCheck: Comparable Solutions (due Tuesday 17, February)

0. RadioStation

Suppose that a class `RadioStation` has been defined for storing information about radio stations. Each station object keeps track of its name (a `String`), its broadcast band (a `String`) and its station number (a `double`). For example, there is a local station called `KUOW` that is an FM station broadcast on 94.9. The class looks like:

```
1 public class RadioStation implements Comparable<RadioStation> {
2     private String name; // name such as "KUOW"
3     private String band; // band such as "AM" or "FM"
4     private double station; // station such as 94.9
5
6     //Constructor that takes name, band, and station
7     //Getters for all fields
8
9 }
```

Your task is to modify the class to be `Comparable` by adding an appropriate `compareTo` method. Radio stations should be grouped together by band (e.g., all AM stations grouped together and all FM stations grouped together). Within a given band, the stations should be sorted by station number (e.g., FM 94.9 less than FM 96.5).

The broadcast band can be any arbitrary `String`. For example, it might be "AM" versus "FM" or might include subdivisions like "FM 1" and "FM 2" or might include other text like "XM" for satellite radio.

Solution:

```
1 public int compareTo(RadioStation other) {
2     if (band.compareTo(other.band) != 0) {
3         return band.compareTo(other.band);
4     } else {
5         return ((Double) this.station).compareTo(other.station);
6         /* An alternative solution:
7          return (int) Math.signum(this.station - other.station);
8          */
9     }
10 }
```