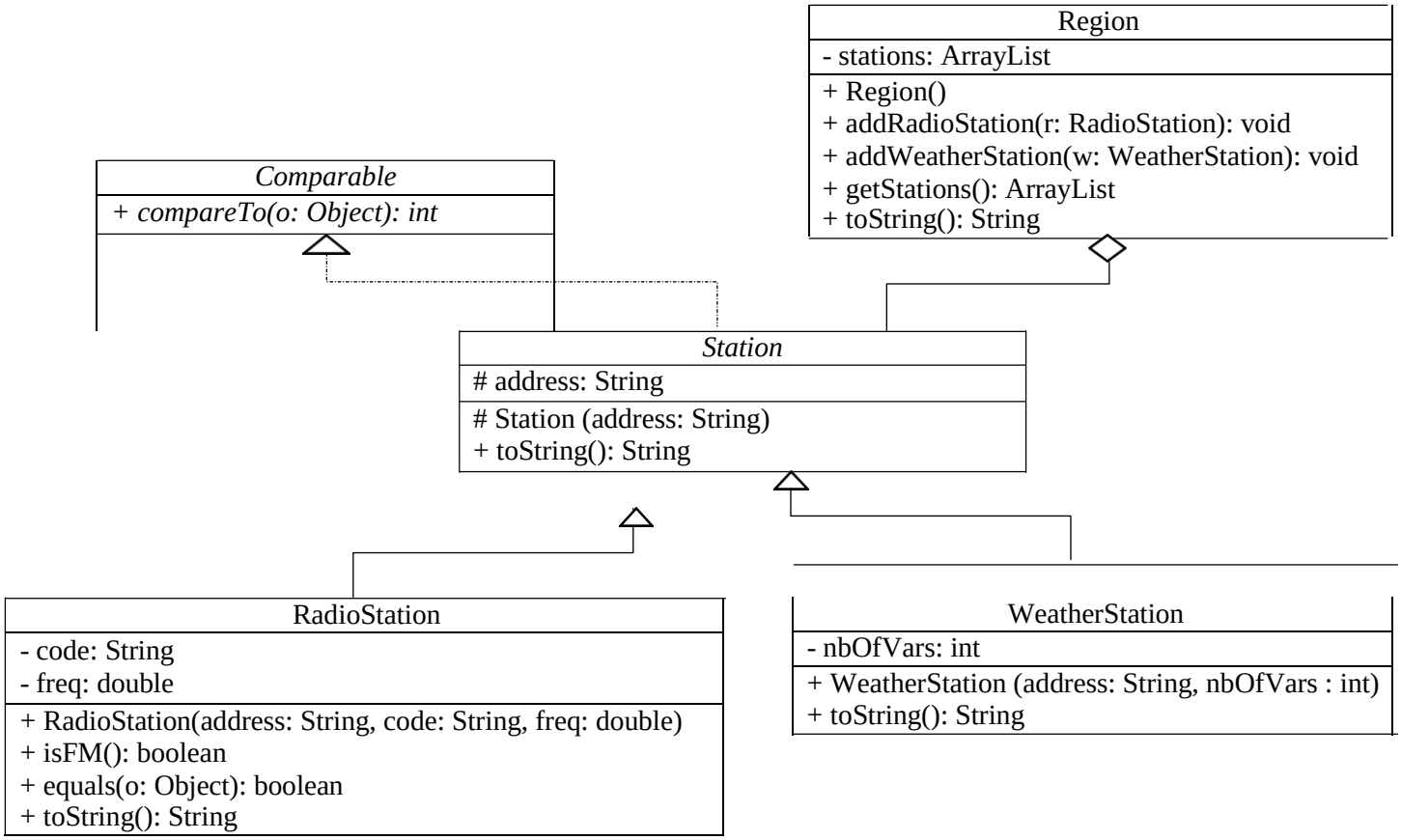
Part 1

Implement the following classes. Pay attention to the details depicted in the UML diagram below.



RadioStation:

1) The ***code*** is formed of 3 or 4 capital case alphabets.
2) The frequency, referred to as ***freq,*** is a number between 87.8 to 108.0 for FM stations, and a number between 520.0 and 1610.0 for AM stations.
3) The constructor throws an IllegalArgumentException if the code and freq do not meet the requirements explained in parts 1 and 2.
4) Two RadioStation objects are equal if they have the same code and frequency.
5) Two RadioStation objects are compared according to their frequency.
6) The toString method should return the description of the object as in the following example: [KCSB, FM 91.9]

WeatherStation:

1) Two WeatherStation objects are compared according to their nbOfVars.

Region:

1) In the addRadioStation method, make sure not to add a radio station that already exists in the ArrayList; in this case, you should display to the user a message "Already Exists!".

Part 2.

Using the classes in Part 1, develop an application as follows:

a) Create a region object
b) Add some Station Objects to it as long as the user wishes. Make sure you handle at least two Exceptions; one of them is the IllegalArgumentException that might arise in the constructor of RadioStation.
c) Write a method that takes as parameter an ArrayList and displays all FM radio stations. Invoke it in the main method.
d) Write a method that takes as parameter an ArrayList and returns an array of type WeatherStation that contains all WeatherStation objects. Invoke this method in the main method. Then display the WeatherStation object that has the highest nbOfVars value.