

5 Stars

Purpose

To review interfaces

Directions

Your task for this lab is to implement and test a class to store ratings of items. Your Rating class should have the following fields and methods:

private double sumOfRatings - the sum of all of the ratings

private int raters - the number of people who have rated an item

public Rating() - initialize both fields to zero

public Rating(double sumOfRatings, int raters) - initialize the fields to the parameter values

public void addRating(double newRating) - increase the sumOfRatings by newRating and increase the number of raters by one

public double getAverageRating() - return the average rating; if there are no raters so far, return 0.0

public String toString() - display the average rating and how many reviews it is based on; for example, if the sum of the ratings is 32 and that is based on ratings from ten people, the toString method will return "3.2 based on 10 reviews"

Your Rating class should also implement the Comparable interface - Ratings should be sorted such that the highest average rating appears first. If two Rating objects have the same average, the one with the higher number of reviews should appear first.

Example

If your class is used with this driver code:

```
Rating r1 = new Rating();
System.out.println("r1: " + r1);
r1.addRating(5);
System.out.println("r1: " + r1);
r1.addRating(3.5);
System.out.println("r1: " + r1);

Rating r2 = new Rating(40, 10);
```

```
System.out.println("r2: " + r2);

Rating r3 = new Rating(12.75, 3);
System.out.println("r3: " + r3);

ArrayList<Rating> ratings = new ArrayList<>();
ratings.add(r1);
ratings.add(r2);
ratings.add(r3);
Collections.sort(ratings);

for (Rating r: ratings) {
    System.out.println(r);
}
```

it should produce this output:

```
r1: 0.0 based on 0 reviews
r1: 5.0 based on 1 reviews
r1: 4.25 based on 2 reviews
r2: 4.0 based on 10 reviews
r3: 4.25 based on 3 reviews
4.25 based on 3 reviews
4.25 based on 2 reviews
4.0 based on 10 reviews
```