AP Computer Science A@Beijing National Day School

Lab 13: `ScoreInfo`

Due date: Friday, March 29, 2019
Instructor: Mr. Alwin Tareen
Total Points: 15

Task Overview

- Implement a program that maintains a database of student test scores.

Background

- A school district would like to get some statistics on its students' standardized test scores. These scores will be represented as objects of the following `ScoreInfo` class. Each `ScoreInfo` object contains a score value and the number of students who earned that score.

```
1  public class ScoreInfo
2  {
3      private int score;
4      private int numStudents;
5
6      public ScoreInfo(int aScore)
7      {
8          score = aScore;
9          numStudents = 1;
10     }
11
12     /** Adds 1 to the number of students who earned this score.
13      */
14     public void increment()
15     {
16         numStudents++;
17     }
18
19     /** @return this score
20      */
21     public int getScore()
22     {
23         return score;
24     }
25
26     /** @return the number of students who earned this score.
27      */
28     public int getFrequency()
29     {
30         return numStudents;
31     }
32 }
```

- The following `Stats` class creates and maintains a database of student score information. The scores are stored in sorted order in the database.

```
1  public class Stats
2  {
3      /** Listed in increasing score order; no two ScoreInfo objects contain the same score.
4       */
5      private ArrayList<ScoreInfo> scoreList;
6
7      /** Records a score in the database, keeping the database in increasing score order.
8       * If no other ScoreInfo object represents score, a new ScoreInfo object representing
9       * score is added to the database; otherwise, the frequency in the ScoreInfo object
10      * representing score is incremented.
11      * @param score a score to be recorded in the list.
12      * @return true if a new ScoreInfo object representing score was added to the list;
13      * false otherwise.
14      */
15     public boolean record(int score)
16     {
17         /* to be implemented in part (a) */
18     }
19
20     /** Records all scores in stuScores in the database, keeping the database in
21      * increasing score order.
22      * @param stuScores an array of student test scores.
23      */
24     public void recordScores(int[] stuScores)
25     {
26         /* to be implemented in part (b) */
27     }
28
29     // There may be instance variables, constructors, and methods that are not shown.
30 }
```

(a) Write the `Stats` method `record` that takes a test score and records that score in the database. If the score already exists in the database, then the frequency of that score is updated. If the score does not exist in the database, then a new `ScoreInfo` object is created, and inserted into the appropriate position, so that the database is maintained in increasing score order. The method returns `true` if a new `ScoreInfo` object was added to the database; otherwise, it returns `false`.

(b) Write the `Stats` method `recordScores` that takes an array of test scores and records them in the database. The database contains at most one `ScoreInfo` object per unique score value. Each `ScoreInfo` object contains a score and an associated frequency. The database is maintained in increasing order based on the score.

In writing `recordScores`, assume that `record` works as specified, regardless of what you wrote in part (a).

Specification

The Information Box Which Includes Your Name[5 points]

- Type your English and Pinyin name into the `Author` field, where it says: `YOUR NAME HERE`

Maintain a Database of Student Test Scores [10 points]

- Write a `Java` program in the file `Stats.java` that maintains a database of student test scores.
- You will write your solution in a class called: `public class Stats` right below the place where it says: `YOUR CODE HERE`.
- Make sure that you run your `Java` program, and ensure that it is free of errors.

Testing

- The file `StatsJUnitTest.java` contains the `JUnit` test cases which verify the correct functionality of the program.

Submission

- Submit your `Java` program by uploading it to the `Web-CAT` automated grading platform: `http://ec2-54-65-207-33.ap-northeast-1.compute.amazonaws.com:8080/Web-CAT/WebObjects/Web-CAT.woa`