MSc CS+ Java Test 2015 - 16

Assigning Referees to JavaBall Matches

Program Context

The favourite sport in the country of Hibernia is the team game JavaBall. You have been asked to produce a program to manage the assignment of referees to the JavaBall matches that come into the 'professional' category. Interviews and discussions have been carried out with the AJB (Association of JavaBall Referees) to discover the program context and the functionality required. These details are given below.

Details of matches

There are several professional JavaBall teams in the country, and once a week there is a scheduled match between two of these teams. There is one JavaBall stadium in each area of the country. Because of its popularity, the general public are prepared to travel the length of the country to watch a match. It has therefore been decided that there will only ever be one professional match played in a particular week.

There is a Sports Academy where young people under the age of 21 are trained in the game, and they have several teams which play in the "professional" category. A match containing at least one under-21 team is referred to a "junior" match. A match that does not contain an under-21 team is a "senior" match.

Referee qualifications

- There are two qualification awarding bodies, the National JavaBall Council, and the International JavaBall Council, represented by the letters NJB and IJB respectively.

- Qualifications are referred to as NJB1, NJB2, IJB1, IJB2, IJB3 etc., where the final digit refers to the level of the qualification.
- The highest possible qualification is level 4.
- A "level one" qualification enables the referee to attend junior matches only.
- A "level two" qualification or above enables the referee to attend all matches.

Areas

For administrative purposes, the country is divided into three areas: North, South and Central. As you might expect, the North area is adjacent to the Central area and not to the South area. The South area is adjacent to the Central area and not to the North area. The Central area is adjacent to both North and South areas.

Referee details

The following information is held about each referee:

- **ID.** This consists of two alphabetic characters (their initials) and a sequence number. The sequence number will be '1' in most cases, wherever the initials are unique. If the initials are not unique, there will be one ID with the sequence number '1', and another with sequence number '2', etc.
  E.g. KL1, KL2, JM1, …
- **Name.** This consists of a first name followed by a last name. The people in this country take great care to avoid repetition when naming their children, and therefore you can assume that the combination of first name and last name will be unique.

There should not be any spaces within either the first or last names. The names of the referee will not change (if they get married they keep their maiden name for professional purposes).
E.g., John Rankin, Anne-Marie Harrison-Smythe

- **Qualification**. See details above. E.g., NJB2.
- **Home locality.** This is "North", "South" or "Central". Each referee must be based in Hibernia, and will therefore have a home locality.
- **Localities that they are prepared to visit.** These are any of the areas that they are prepared to referee a match in. This *must* include their own home locality, and may include one or both of the other areas.
- **Number of matches allocated.** The number of matches to which the referee has been allocated so far this season.

Assigning the two referees
Each match requires *two* suitably qualified referees (refer to the "Referee qualifications" section above).

Preference is given to referees who live in the area of the stadium and have the least number of allocations compared to other suitable referees in this area.

After that, referees are considered who live in an adjacent area to the stadium, are prepared to travel there, and have the least number of allocations compared to other referees who come into this category. As described above, the North area is adjacent to the Central area and not to the South area. The South area is adjacent to the Central area and not to the North area. The Central area is adjacent to both North and South areas.

Finally, referees are considered who live in a non-adjacent area to the stadium, but who are still prepared to travel there, and have the least number of allocations compared to other referees in this category.

If there is more than one equally suitable referee (i.e., at equal distance, and having the same number of matches), it does not matter who is chosen.

Here are three examples to illustrate how these priorities affect the assignment of referees. Each of the examples should be considered in isolation: no attempt has been made to ensure consistency between the examples – they are merely there to demonstrate situations that might occur, rather than referring to any specific list of referees.

Example 1. Three suitably-qualified referees are available who live in the area: KT1 has been allocated 10 matches, PT1 7 matches, and TR1 5. Therefore the first referee chosen is TR1, and the second is PT1.

Example 2. Only one suitably qualified referee (GH1) is available in the area (which is Central). He has been allocated 8 matches. Two suitable referees are available from the North Area (RG2 has been allocated 5 matches and JW2 has been allocated 6 matches). One suitable referee is available from the South area (KL1 with 5 matches). Therefore GH1 is chosen, and either of KL1 and RG2.

Example 3. There are no suitable referees in the area, which is North. There are three suitable referees in Central (HG1, 3 matches; JK1, 10 matches; KL2: 13 matches). There is one suitable referee in South (RG2, 5 matches). HG1 and JK1 are chosen because they live in Central, even though RG2 has fewer matches than JK1.

Functionality

1. The functionality of the program should be accessed by a GUI. You are free to design this as you choose, using one or more windows.

2. A text file giving the details of the referees should be read in automatically at the start of the program, and the details stored into an array. The text file will be called `RefereesIn.txt`, ordered on the referee ID, and may contain up to 12 referees. (A sample file `RefereesIn.txt` is shown in Appendix 1.) The referee IDs are ordered lexicographically (i.e., in dictionary order) on the first two characters, and then numerically on the third character if the first two characters are equal. So for example AB3 comes before AY2, and DG2 comes before DG4.

   You may assume that the input file contains valid data and is correctly formatted. There is no requirement to validate the input data, and it is acceptable for your program to crash if the input file is corrupt. It is however always a good idea to use the trim() method to remove white space from any String object that might be passed to a parsing method for conversion to an integer.

   The details of each referee are supplied on a single line consisting of:
   - Referee ID;
   - First name followed by last name;
   - Qualification;
   - Number of allocations so far this season;
   - Home area;
   - Areas that the candidate is willing to travel to, in the order North, Central, South. 'Y' indicated that they will travel, 'N' that they will not. Their home area will always be Y.

   Here is a sample line in the text file for a referee in the North area who is prepared to travel to Central but not South:

   ```
   JL1  Jennifer Liddell  NJB2  5 North  YYN
   ```

3. All the referees should be displayed in ID order, one line per referee, showing all the details and using the same format as the input text file, with the information aligned in columns. This display should appear automatically when the GUI is launched, and should be updated whenever necessary throughout the duration of the program.

4. On request, all details of a particular referee can be displayed individually.
   - The referee is identified by the user supplying their first and last names.
   - If a referee with these names cannot be found in the list, a suitable message should be displayed.

- The referee details should be searched for using text fields / combo boxes etc. as appropriate.
- You should use a button to initiate the search.

5. On request, any of the referee details except the ID, first name and last name and the number of match allocations can be updated.
    - The referee should be identified, searched for and displayed exactly as in (4) above.
    - The user can now alter any of the details except the ID, first name, last name and the number of match allocations. Note that a referee's qualification can go down as well as up (the former perhaps corresponds to demotion after too many poor performances).
    - When a button is pressed to request the update, validation should be carried out on the qualification, home area, and other areas to travel to.
        i. The exact details of validation will depend on the instance variables that you are using to describe the data.
        ii. Remember that good use of combo boxes etc. can reduce the need for validation.
        iii. It is not necessary to keep track of which fields have been altered – all can be validated.
    - If any fields are invalid, a suitable message should be displayed to the user.
    - The referee details should only be updated in the array if *all* fields are valid. This updated array will be written to an output text file at the close of the program.
    - Any displays showing details of referees should be updated.

6. On request, a referee can be deleted from the array.
    - The referee should be identified, searched for and displayed exactly as in (4) above.
    - The user can then request that the referee is deleted.
    - Any displays showing details of referees should be updated.

7. On request, a new referee can be added to the array.
    - Use the text fields etc. that have been used in (4 - 6) above.
    - The new ID is not input by the user but is calculated automatically by the program.
        i. The initials are obtained from the first and last name.
        ii. A referee with initials that are not already in the list is allocated an ID consisting of their initials and the sequence number '1'.
        iii. A referee with initials that are already in the list should be allocated the next available sequence number.
        E.g., Kenneth Lawrence is entered but KL1 and KL2 are already in the list, so Kenneth Lawrence is given the ID KL3.
    - You should allow for the possibility that a new referee is to be given a number of match allocations that is larger than 0 (corresponding to him/her moving to Hibernia, having refereed somewhere else in the same season).
    - All input should be validated, and must be valid and complete before the record can be accepted.
    - Feedback of success or validation errors should be displayed to the user.
    - Any displays showing details of referees should be updated.

The maximum number of referees allowed throughout the season is 12, and you should check that there is room to add a new referee before allowing this operation to proceed.

8. On request, the program should display a bar chart showing the number of allocations for each referee above each bar. The referees should be identified in the chart by their ID, and the bars should be displayed in ID order.

9. On request, allocate two referees to a match.
   - The details of this match are supplied via the GUI.
   - Match details consist of:
     i. The week number that the match is held in (1 - 52). There is at most one match per week. The allocations of referees to matches need not occur in sequential week numbers (for example the referees for a match in week 3 may be allocated before those for a match in week 1).
     ii. The area that the match is held in.
     iii. The level of the match ("Junior" or "Senior").
   - A list of all suitable referees (containing at least their name and the number of allocations so far) should be displayed, in order of suitability. Also, the two most suitable referees are selected automatically, by the program:
     i. Their names are displayed to the user.
     ii. The number of allocations for each of these two referees is increased by one.
     iii. The match report for the list of matches must be updated (see below).
   - If there are not two suitable referees, a message should be displayed to this effect.
   - Note that there is no need to keep details of the particular matches that a given referee has been allocated to. We just need the number of allocations. It may however be useful to keep track internally of which referees have been allocated to a particular match.

10. By the end of the program, a text file should be produced giving details of allocations to each match. The text file should be called `MatchAllocs.txt`.
    - This text file should consist of a suitable heading, followed by one line per match, giving the details of the match (week number, level and area) and the names of the referees assigned.
    - These lines should be in order of allocation. (For example, if the referees for a match in week 7 are allocated before those for a match in week 3, then the details for the match in week 7 will appear in `MatchAllocs.txt` before the details for the match in week 3.) They could be appended to the text file at the time of allocation, or held in a list and written at the close of the program. Here is a sample line for one match:

      ```
      23 Senior  North     Derek Riordan     Jane Gray
      ```

11. There should be an 'exit' button. When this button is clicked, the updated referee list is written to an output text file, in exactly the same format as the input text file (and ordered on referee ID). The text file should be called `RefereesOut.txt`. The list of allocations, `MatchAllocs.txt`, in order of allocation, is also produced (or closed). Then the program exits.

12. Create a jar file and a batch file so that by placing a shortcut icon on the desk top the program can be run by double clicking the icon.
    The jar file should be called `TeamProj.jar`. (Eclipse can be used to create the jar file – see Part II, Lecture 18.)
    The batch file should be called `TeamProj.bat`.

After submission your program will be tested, so it is particularly important that the file-handling within your program does not refer to the absolute location of your text files. When testing your program within Eclipse, store your data files in the current working directory. Please do not use JFileChooser: it is too time-consuming for us when we test your programs!

Appendix 1 – format of the input file

The following is an example of a possible `RefereesIn.txt` file:

```
DG1 Dave   Gray NJB2 3 Central NYY
DM1 Denis Montague IJB1 3 North YYY
JL1 Jack Low IJB2 2 South NYY
KL1 Kenneth Lorrimer IJB2 6 South YYY
KL2 Kenneth Lindsay NJB3 12 Central YYY
TT1 Tim Toms IJB1 2 Central NYN
```