

Description

Your task is to create an application using java which simulates the manipulation of a premier league championship.

Design and Implement a class PremierLeagueManager (for football https://en.wikipedia.org/wiki/Premier_League) which extends interface LeagueManager. The LeagueManager interface must be designed such that in the future it can be extended to maintain not only a number of football premier league but also academic clubs such as university sport club and school sport clubs (https://en.wikipedia.org/wiki/Professional_Development_League).

Design a solution:

The design of your system should be consistent with the Object Oriented principles and easy to understand by an independent programmer. You are required to design your program using UML diagrams. In particular you have to draw:

- class diagrams
- two or more use case diagrams for the system (one for the Console and one for GUI)

Problem description and requirement statement

The details for the implementation of the system are given in the steps below: It is important to follow exactly the specifications and your implementation must conform to these:

1. Design and implement a super class SportsClub that should include appropriate methods and hold information about the name of the club, its location and various statistics about the club and then create the sub class FootballClub, with appropriate attributes specific to each of them.

For example when it comes to football should include statistics such as how many wins, draws and defeats an instance of it has achieved in the season, and the number of goals received and scored. The number of points that a club currently has, and the number of matches played should also be included.

Further sub classes for UniversityFootballClub (Under 23) and SchoolFootballClubs (Under 18)

2. Implement a class PremierLeagueManager which extends interface LeagueManager. The PremierLeagueManager class maintains a number of football clubs which play in the premier league.

The class should create a menu based on command line interface based input (i.e. console and NOT graphical components) and give the user the choice of:

- Create a new football club and add it in the premier league.
- Delete (relegate) an existing club from the premier league.
- Display the various statistics for a selected club.
- Display the Premier League Table, i.e. display all the teams playing in the premier league and some of their statistics, in descending order, according to the points they have. Thus, the club which has the maximum number of points should be displayed first, the club being second in the league should be displayed next, etc. In the case which two clubs have the same number of points the club with the best goal difference should appear first.
- Add a played match with its score and its date, so that the statistics of the two clubs involved and the premier league table are updated automatically.
- Saving in a file of all the information entered by the user. **[Use of database is strictly prohibited]**
- The next time the application starts it should read all the information saved in the previous file (resume/recover the previous state of the program) and continue its operation based on that with the user being able to enter new information or change the existing information.

3. Design and implement a graphical user interface (GUI) which is able to do the following:

- Display the list (table) of all the teams and their statistics in descending order of points.
- Give the user the possibility of sorting the previous table according to goals scored (descending order).
- Give the user the possibility of sorting the previous table according to the largest number of wins (descending order).
- Add a button which every time it is pressed it generates one random played match between two randomly chosen clubs (teams) and it automatically updates the premier league table by adding the match (points, score and other statistics). The score and chosen teams should be entirely random and not hardcoded in your source code. The button should generate a different match and a different score every time it is clicked. The user should be able to see the randomly generated

match with the score (in addition to the table of standings), in order to be able to verify the correctness of your code for the updated information of the table.

- Add a button which displays all the played matches sorted in ascending order of date played (both randomly generated or manually entered using the text menu functionality described above). This should display all the matches played, included matches inserted and generated in previous runs of the application (assuming that the user saved the information entered using the text menu functionality above).
- Add a button and a textbox which can be used to search for all matches played in a given date. The full details of the matches should be displayed (i.e. both club names and the score).

Use same data source [No database only file handling is allowed] that is used in Console application for GUI as well.

Marking will also be allocated for code quality such as:

- Code readability (structure, comments, variable naming, etc.)
- Implementation (e.g. quality, efficiency, etc.)

The maximum for work which does not compile is 30% (i.e. a mark in the range 1 to 30% will be awarded).

Submission of assignments using a different method other than Blackboard will not be accepted and zero (0) marks will be awarded in such cases.

Deadline: Tuesday 4th of Jan 2021, 13:00.

Mark Allocation:

Design using UML diagram	10
POJO Classes	5
LeagueManager	2
FootballLeagueManager and CLI	40
GUI	35
Code Quality	8