Description

Your task is to create a Java program which simulates the manipulation of a Formula 1 racing car championship.

For the GUI part you are NOT allowed to use drag and drop tools (such as those found in Netbeans, etc.) to create the graphical user interface for any part of this coursework! All graphical code should be manually written in Java Swing and no tool which generates code automatically should be used. Any submission which uses for any part of this assignment drag and drop tools will not receive any marks for these parts.

Implement a class Formula1ChampionshipManager which extends interface ChampionshipManager. The Formula1ChampionshipManager class maintains a number of drivers and cars (constructors, i.e. manufacturer of the car) which compete in the Formula 1 car championship. Each driver belongs to exactly one constructor team (e.g. Ferrari) and each constructor team has a single driver (e.g. Mercedes has a single driver called Hamilton)

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

1. Design and implement classes Driver (abstract class), Formula1Driver. The classes should include appropriate methods and hold information about the name of the driver, their location, the team they belong to and various statistics about the drivers. Formula1Driver should include statistics such as how many frst position, second positions and third positions an instance of it has achieved in the season. The number of points that a driver currently has, and the number of races participated so far in the season should also be included.

The points awarded for each driver in a race (and for the all the calculations in this assignment) are according to the following scheme:

1:25 2:18 3:15 4:12 5:10 6:8 7:6 8:4 9:2 10:1

i.e. the driver who got the first position in the race wins 25 points, the second 18, the third 15 and so on. A driver must finish the race to be awarded points and also finish it in the first 10 positions. (5 marks).

2. Implement a class **Formula1ChampionshipManager** which extends interface **ChampionshipManager**. The **Formula1ChampionshipManager** class maintains a number of drivers who play in the Formula 1 championship. (5 marks).

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

Create a new driver (who is added in the championship). The driver should be associated with a unique team (car manufacturer) (4 marks).

Delete a driver and the team that the driver belongs to from the Formula 1 championship. (2 marks).

- Change the driver for an existing constructor team (e.g. change the driver for the Ferrari team). 2 marks

- Display the various statistics for a selected existing driver. (4 marks).

- Display the Formula 1 Driver Table, i.e. display all the drivers competing in the Formula 1 championship, the team they belong to and some of their statistics, in descending order, according to the points they have in the current season. Thus, the driver who has the maximum number of points should be displayed first, the driver being second in the championship should be displayed next, etc. In the case that two drivers have the same number of points the driver who has won the most first positions in races should appear first. (8 marks).

- Add a race completed with its date and the positions that all the drivers achieved. The statistics of all the drivers who participated and the Formula 1 championship table are updated automatically. (7 marks).

- Saving in a file of all the information entered by the user so far. (8 marks).

- 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. (9 marks).

3. Start a graphical user interface (GUI) based on Java Swing from the text menu (i.e. console) which is able to do the following:

- Display the list (table) of all the drivers and their statistics in descending order of points. (4 marks).

- Give the user the possibility of sorting the previous table according to points won by drivers (ascending order). (4 marks).

- Give the user the possibility of sorting the previous table according to the largest number of first positions won in races (descending order). (4 marks).

- Add a button which every time it is pressed it generates one random race with random positions achieved by the existing drivers. This automatically updates the Formula 1 championship table by adding the race (points, positions and other statistics). The positions should be entirely random and not hardcoded in your source code. The button should generate a different race with different driver positions every time it is clicked. The user should be able to see the randomly generated race with the driver positions (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. (8 marks).

- Add a button which is similar in functionality with the previous questions (i.e. it generates the results of a full race and adds them in the statistics) with the following modification. The drivers are starting the race in a randomly calculated position (e.g. Vettel in starting position 1, Hamilton in starting position 2 and so on). These starting positions should be random and not hardcoded. The results of the race are probabilistic and according to the starting position, more specifically: the driver starting in position 1, has 40% probability to win the race, the driver starting in position 2 has 30% probability to win the race, the drivers starting in positions 3 and 4 have a 10% each to win the race. All of the drivers in positions 5 to 9 each have a probability of 2% to win the race and all other drivers have a 0% chance to win the race. The rest of the positions (2–10) are determined completely randomly marks (8).

- Add a button which displays all the completed races 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 races that took place in the season, included races inserted and generated in previous runs of the application (assuming that the user saved the information entered using the text menu functionality above). (7 marks).

- Add a button and a textbox which can be used to search for all races that a given driver participated. The full details of the races should be displayed (i.e. both the dates and the positions of the driver in the matching races). (5 marks).