## Background:

You have been tasked with creating an application to manage the points and costs of a racing championship. The championship contains multiple teams. Each team has 2 drivers, and the engines of a team comes from different manufacturers. The manufacturer has an influence on the costs.

## Task Description:

### Implement the Team as a private type:

Each Team stores the following data:

- Team Name of type Team_Name_T.
- Engine Type.
- Name of Driver 1.
- Name of Driver 2.
- Points of the team of type Natural -> initially 0.
- Season Cost of type Float.

Use the following enumerator for Team_Name_T:

**type Team_Name_T is (Mclaren, Red_Bull_Racing, Ferrari, Mercedes, Aston_Martin, RB, Haas, Williams, Alpine, Kick_Sauber);**

Use the following enumerator for Engine Type:

**type Engine_T is (Renault, Ferrari, Mercedes, Honda_RBPT);**
**for Engine_T use (1,2,3,4);**

To get the defined value of an enumerator you can use:

**EnumType'Enum_Rep(Enum)**

Implement the following functions and procedures as an interface for the Team:

- Getters for all 6 attributes.
- Getter for the value of the Engine Type.
- Procedure to print all data of the Team in JSON format (Create a string which looks like JSON).
- Procedure to increase the points of the team by an integer value.
- Procedure to increase the Season Cost of the team by a Float value.
- Procedure to create a New Team.

If you want to implement any other functions of procedures, they must be private.

You may add extra attributes to the Team to keep track of extra data.

## Implement the Championship as generic and private:

The Championship should store the following data:

- Championship Name.
- List of Teams in the championship where both the Index and Elem are generic.
- Current Team which is not allowed to Race.
- Number of Teams allowed to race.
- Number of points given to a team if one of their drivers finishes first -> initialized to 25
- Number of points given to a team if one of their drivers finishes first -> initialized to 18
- Number of points given to a team if one of their drivers finishes first -> initialized to 15

The Championship should have a parameter passed to it which is the maximum number of teams in the championship.

Implement the following functions and procedures as an interface for the Championship package:

- Operation which receives an array of teams and adds them to the Championship. If no more teams can be added, then the user should be informed.
- Procedure which prints all data of the Championship in JSON format (Create a string which looks like JSON).
- Procedure which creates a new Championship with the Name of the Championship received as a parameter.
- A generic operation which records a race and takes a generic function or procedure which does the following:
    - Receives the name of the Top 3 drivers who finished and adds the corresponding number of points to the team for which the driver races.
    - The team of the driver who came in first place will not be allowed to race in the next race:
        - This team must be moved to the  variable of the Current Team which cannot race and removed from the List of teams.
        - When a next race occur, the new team which may not race must be swapped with the previous team which was not allowed to race. The previous team must be added back to the List of teams.
    - A generic operation which increases the season cost of each team
        - The increase is 10,000,000 * by the Value of the Engine Type
        - Print all data of the team.

You may add extra attributes to the Championship to keep track of extra data.

Do intensive testing within the Main program to show all features working as intended as well as error checking where needed (Such as adding too many teams to the Championship).

Submit your program in a zip folder which contains the src folder, the main test file and the project file. Do not submit the obj folder.