

## PROJECT-8

- Write a program that models a soccer league and keeps track of the season's statistics. Carefully consider what data should be stored in an array and what data should be stored in an ArrayList. Design classes with fields and methods based on the description of the league. You'll also need a test class that contains a main method. All fields must be private. Provide any necessary getters and setters.

### Teams

Each team has a name. The program should also keep track of each team's win-total, loss-total, tie-total, total goals scored, and total goals allowed. Create an array of teams that the scheduler will manage.

Print each team's statistics when the season ends.

```
package soccer;
```

```
public class Team {
```

```
    private String name;
    private int wins;
    private int losses;
    private int ties;
    private int totalGoalsScored;
    private int totalGoalsAllowed;
```

```
    public Team(String name) {
        this.name = name;
        this.wins = 0;
        this.losses = 0;
        this.ties = 0;
        this.totalGoalsScored = 0;
        this.totalGoalsAllowed = 0;
    }
```

```
// Getters and setters
```

```
    public String getName() { return name; }
    public int getWins() { return wins; }
    public int getLosses() { return losses; }
    public int getTies() { return ties; }
    public int getTotalGoalsScored() { return totalGoalsScored; }
    public int getTotalGoalsAllowed() { return totalGoalsAllowed; }
```

```
    public void addWin() { wins++; }
    public void addLoss() { losses++; }
    public void addTie() { ties++; }
    public void addGoalsScored(int goals) { totalGoalsScored += goals; }
    public void addGoalsAllowed(int goals) { totalGoalsAllowed += goals; }
```

```
@Override
```

```

public String toString() {
    return name + " - Wins: " + wins + ", Losses: " + losses + ", Ties: " + ties +
        ", Goals Scored: " + totalGoalsScored + ", Goals Allowed: " + totalGoalsAllowed;
}

```

## Games

In a game, it's important to note each team's name, each team's score, and the temperature that day. Number each game with integer ID number. This number increases as each game is played. Keep track of every game played this season. This class stores an ArrayList of all games as a field.

Your program should determine scores at random. The maximum number of goals any one team can score should increase proportionally with the temperature. But make sure these numbers are somewhat reasonable.

When the season ends, print the statistics of each game. Print the hottest temperature and average temperature for the season.

```
package soccer;
```

```

public class Game {

    private static int gameCount = 0;
    private int id;
    private Team team1;
    private Team team2;
    private int score1;
    private int score2;
    private int temperature;

    public Game(Team team1, Team team2, int temperature) {
        this.id = ++gameCount;
        this.team1 = team1;
        this.team2 = team2;
        this.temperature = temperature;
        playGame();
    }

    private void playGame() {
        int maxGoals = temperature / 10;
        score1 = (int) (Math.random() * (maxGoals + 1));
        score2 = (int) (Math.random() * (maxGoals + 1));
        team1.addGoalsScored(score1);
        team1.addGoalsAllowed(score2);
        team2.addGoalsScored(score2);
        team2.addGoalsAllowed(score1);

        if (score1 > score2) {
            team1.addWin();
            team2.addLoss();
        } else if (score1 < score2) {
            team2.addWin();

```

```

        team1.addLoss();
    } else {
        team1.addTie();
        team2.addTie();
    }
}

public int getTemperature() {
    return temperature;
}

@Override
public String toString() {
    return "Game " + id + ": " + team1.getName() + " " + score1 + " - " + score2 + " " +
team2.getName() + " | Temperature: " + temperature;
}
}

```

### Scheduler

Accept user input through a JOptionPane or Scanner. While the application is running, ask the user to input a temperature. The program should not crash because of user input. If it's warm enough to play, schedule 2 games. Opponents are chosen at random. Make sure teams aren't scheduled to play against themselves. If there are 3 consecutive weeks of freezing temperatures, the season is over.

```

package soccer;

import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;

public class scheduler {
    private Team[] teams;
    private ArrayList<Game> games;
    private int freezingWeeks;

    public scheduler() {
        teams = new Team[] {
            new Team("Team A"),
            new Team("Team B"),
            new Team("Team C"),
            new Team("Team D")
        };
        games = new ArrayList<>();
        freezingWeeks = 0;
    }

    public void startSeason() {
        Scanner scanner = new Scanner(System.in);
        Random random = new Random();
    }
}

```

```

boolean seasonOver = false;

while (!seasonOver) {
    System.out.print("Enter the temperature for this week: ");
    int temperature = scanner.nextInt();

    if (temperature <= 32) {
        System.out.println("Too cold to play.");
        freezingWeeks++;
        if (freezingWeeks >= 3) {
            seasonOver = true;
            System.out.println("Season is over.");
        }
    } else {
        freezingWeeks = 0;
        playGames(temperature, random);
    }
}

printSeasonResults();
}

private void playGames(int temperature, Random random) {
    ArrayList<Team> teamsList = new ArrayList<>();
    for (Team team : teams) {
        teamsList.add(team);
    }

    while (teamsList.size() > 1) {
        Team team1 = teamsList.remove(random.nextInt(teamsList.size()));
        Team team2 = teamsList.remove(random.nextInt(teamsList.size()));
        Game game = new Game(team1, team2, temperature);
        games.add(game);
        System.out.println(game);
    }
}

private void printSeasonResults() {
    System.out.println("RESULTS");
    for (Team team : teams) {
        System.out.println(team);
    }
    int hottestTemperature = games.stream().mapToInt(Game::getTemperature).max().orElse(0);
    double averageTemperature =
games.stream().mapToInt(Game::getTemperature).average().orElse(0);
    System.out.println("Hottest Temperature: " + hottestTemperature);
    System.out.println("Average Temperature: " + averageTemperature);
}

```

```
}  
}
```

