Girija.B

192311044

Java Foundations Practices - Section 8:

The Soccer League

Overview

It's been a brutally cold and snowy winter. None of your friends have wanted to play soccer. But now that spring has arrived, another season of the league can begin. Your challenge is to write a program that models a soccer league and keeps track of the season's statistics. There are 4 teams in the league. Matchups are determined at random. 2 games are played every Tuesday, which allows every team to participate weekly. There is no set number of games per season. The season continues until winter arrives. The league is very temperature-sensitive. Defenses are sluggish on hot days. Hotter days allow for the possibility of more goals during a game. If the temperature is freezing, no games are played that week. If there are 3 consecutive weeks of freezing temperatures, then winter has arrived and the season is over.

Tasks

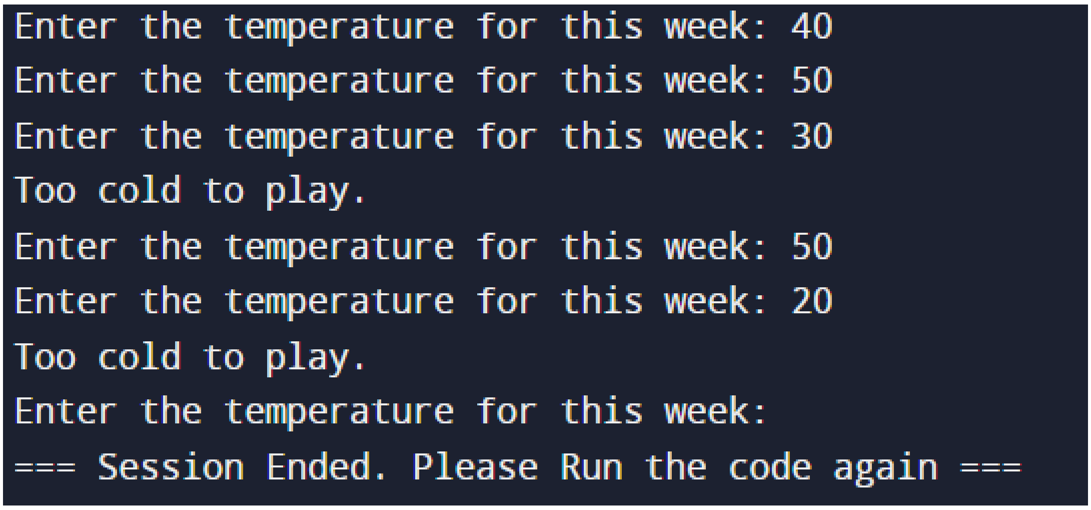
Write a program that models a soccer league and keeps track of the season's statistics. Carefully consider what data should be stored in an array and what data should be stored in an ArrayList. Design classes with fields and methods based on the description of the league. You'll also need a test class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

CODE:

```
import java.util.ArrayList;

import java.util.Random;

import java.util.Scanner;

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;
}
public String getName() {
    return name;
}
public int getWins() {
    return wins;
}
public void addWin() {
    this.wins++;
}
public int getLosses() {
    return losses;
}
public void addLoss() {
    this.losses++;
}
public int getTies() {
    return ties;
}
public void addTie() {
    this.ties++;
}
public int getTotalGoalsScored() {
    return totalGoalsScored;
}
public void addGoalsScored(int goals) {
    this.totalGoalsScored += goals;
}
public int getTotalGoalsAllowed() {
    return totalGoalsAllowed;
    }
    public void addGoalsAllowed(int goals) {
        this.totalGoalsAllowed += goals;
    }
    @Override
    public String toString() {
        return name + "\nWins: " + wins + ", Losses: " + losses + ", Ties: " + ties
                + "\nPoints Scored: " + totalGoalsScored + ", Points Allowed: " + totalGoalsAllowed;
    }
}
class Game {
    private static int gameCount = 0;
    private int gameId;
    private Team homeTeam;
    private Team awayTeam;
    private int homeTeamScore;
    private int awayTeamScore;
    private int temperature;
    public Game(Team homeTeam, Team awayTeam, int temperature) {
        this.gameId = ++gameCount;
        this.homeTeam = homeTeam;
        this.awayTeam = awayTeam;
        this.temperature = temperature;
        playGame();
    }
    private void playGame() {

        homeTeamScore = (int) (Math.random() * (temperature / 10 + 1));
        awayTeamScore = (int) (Math.random() * (temperature / 10 + 1));
        homeTeam.addGoalsScored(homeTeamScore);
        homeTeam.addGoalsAllowed(awayTeamScore);
        awayTeam.addGoalsScored(awayTeamScore);
        awayTeam.addGoalsAllowed(homeTeamScore);
        if (homeTeamScore > awayTeamScore) {
            homeTeam.addWin();
            awayTeam.addLoss();
        } else if (homeTeamScore < awayTeamScore) {
            homeTeam.addLoss();
            awayTeam.addWin();
        } else {
            homeTeam.addTie();
            awayTeam.addTie();
        }
    }
    @Override
    public String toString() {
        return "Game #" + gameId + "\nTemperature: " + temperature
                + "\nHome Team: " + homeTeam.getName() + ", " + homeTeamScore
                + "\nAway Team: " + awayTeam.getName() + ", " + awayTeamScore;
    }
}
class Scheduler {
    private Team[] teams;
    private ArrayList<Game> games;
    private ArrayList<Integer> temperatures;
    public Scheduler() {
        teams = new Team[]{
                new Team("Team 1"),
                new Team("Team 2"),
                new Team("Team 3"),
                new Team("Team 4")
        };
        games = new ArrayList<>();
        temperatures = new ArrayList<>();
```

```
}
public void startSeason() {
    Scanner scanner = new Scanner(System.in);
    int freezingWeeks = 0;
    while (true) {
        System.out.print("Enter the temperature for this week: ");
        int temperature = scanner.nextInt();
        temperatures.add(temperature);
        if (temperature <= 32) {
            System.out.println("Too cold to play.");
            freezingWeeks++;
            if (freezingWeeks == 3) {
                System.out.println("Season is over");
                break;
            }
        } else {
            freezingWeeks = 0;
            scheduleGames(temperature);
        }
    }
    printResults();
}
private void scheduleGames(int temperature) {
    Random random = new Random();
    ArrayList<Team> availableTeams = new ArrayList<>();
    for (Team team : teams) {
        availableTeams.add(team);
    }
    for (int i = 0; i < 2; i++) {
        Team team1 = availableTeams.remove(random.nextInt(availableTeams.size()));
        Team team2 = availableTeams.remove(random.nextInt(availableTeams.size()));
        games.add(new Game(team1, team2, temperature));
        }
    }
    private void printResults() {
        System.out.println("\n*********RESULTS*********\n");
for (Team team : teams) {
            System.out.println(team);
            System.out.println();
        }
        for (Game game : games) {
            System.out.println(game);
            System.out.println();
        }
        int hottestTemp = temperatures.stream().max(Integer::compare).orElse(0);
        double averageTemp = temperatures.stream().mapToInt(Integer::intValue).average().orElse(0);
        System.out.println("Hottest Temp: " + hottestTemp);
        System.out.println("Average Temp: " + averageTemp);
    }
}
public class SoccerLeague {
    public static void main(String[] args) {
        Scheduler scheduler = new Scheduler();
        scheduler.startSeason();
    }
}
```

OUTPUT:



```
Enter the temperature for this week: 40
Enter the temperature for this week: 50
Enter the temperature for this week: 30
Too cold to play.
Enter the temperature for this week: 50
Enter the temperature for this week: 20
Too cold to play.
Enter the temperature for this week:
=== Session Ended. Please Run the code again ===
```