**Practices - Section 8: The Soccer League**

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.

**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.

**PROGRAM :**

public class Team

{

private String name;

private int wins;

private int losses;

private int ties;

private int goalsScored;

private int goalsAllowed;

public Team(String name)

{

this.name = name;

this.wins = 0;

this.losses = 0;

this.ties = 0;

this.goalsScored = 0;

this.goalsAllowed = 0;

}

public String getName()

{

return name;

}

public int getWins()

{

return wins;

}

public int getLosses()

{

return losses;

}

public int getTies()

{

return ties;

}

public int getGoalsScored()

{

return goalsScored;

}

public int getGoalsAllowed()

{

return goalsAllowed;

}

public void recordWin(int scored, int allowed)

{

wins++;

goalsScored += scored;

goalsAllowed += allowed;

}

public void recordLoss(int scored, int allowed)

{

losses++;

goalsScored += scored;

goalsAllowed += allowed;

}

public void recordTie(int scored, int allowed)

{

ties++;

goalsScored += scored;

goalsAllowed += allowed;

}

public String toString()

{

return name + "\nWins: " + wins + ", Losses: " + losses + ", Ties: " + ties + "\nPoints Scored: " + goalsScored + ", Points Allowed: " + goalsAllowed;

}

}

**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.

**PROGRAM :**

public class Game

{

private static int gameCounter = 0;

private int gameId;

private Team homeTeam;

private Team awayTeam;

private int homeScore;

private int awayScore;

private int temperature;

public Game(Team homeTeam, Team awayTeam, int temperature)

{

this.gameId = ++gameCounter;

this.homeTeam = homeTeam;

this.awayTeam = awayTeam;

this.temperature = temperature;

playGame();

}

private void playGame()

{

int maxGoals = temperature / 10; // Higher temperatures lead to higher scores

homeScore = (int) (Math.random() \* (maxGoals + 1));

awayScore = (int) (Math.random() \* (maxGoals + 1));

if (homeScore > awayScore)

{

homeTeam.recordWin(homeScore, awayScore);

awayTeam.recordLoss(awayScore, homeScore);

}

else if (homeScore < awayScore)

{

awayTeam.recordWin(awayScore, homeScore);

homeTeam.recordLoss(homeScore, awayScore);

}

else

{

homeTeam.recordTie(homeScore, awayScore);

awayTeam.recordTie(awayScore, homeScore);

}

}

public int getTemperature()

{

return temperature;

}

public String toString()

{

return "Game #" + gameId + "\nTemperature: " + temperature + "\nAway Team: " + awayTeam.getName() + ", " + awayScore +

"\nHome Team: " + homeTeam.getName() + ", " + homeScore;

}

}

**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.

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. 2  
Sample Output :  
run:  
Too cold to play.  
Too cold to play.  
Too cold to play.  
Season is over

**PROGRAM :**

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import java.util.Scanner;

public class Scheduler {

private List<Team> teams;

private List<Game> games;

private int consecutiveFreezingWeeks;

private int hottestTemperature;

private int totalTemperature;

private int totalGamesPlayed;

public Scheduler() {

teams = new ArrayList<>();

games = new ArrayList<>();

consecutiveFreezingWeeks = 0;

hottestTemperature = Integer.MIN\_VALUE;

totalTemperature = 0;

totalGamesPlayed = 0;

// Initialize teams

teams.add(new Team("Team 1"));

teams.add(new Team("Team 2"));

teams.add(new Team("Team 3"));

teams.add(new Team("Team 4"));

}

public void startSeason() {

Scanner scanner = new Scanner(System.in);

Random random = new Random();

while (true) {

System.out.print("Enter temperature for this week: ");

int temperature = scanner.nextInt();

if (temperature <= 32) {

consecutiveFreezingWeeks++;

System.out.println("Too cold to play.");

} else {

consecutiveFreezingWeeks = 0;

playGames(temperature, random);

}

if (consecutiveFreezingWeeks >= 3) {

System.out.println("Season is over due to freezing temperatures.");

break;

}

}

printSeasonResults();

}

private void playGames(int temperature, Random random) {

totalGamesPlayed++;

totalTemperature += temperature;

if (temperature > hottestTemperature) {

hottestTemperature = temperature;

}

// Randomly choose teams for 2 games, ensuring no team plays against itself

for (int i = 0; i < 2; i++) {

Team homeTeam, awayTeam;

do {

homeTeam = teams.get(random.nextInt(teams.size()));

awayTeam = teams.get(random.nextInt(teams.size()));

} while (homeTeam == awayTeam);

Game game = new Game(homeTeam, awayTeam, temperature);

games.add(game);

System.out.println(game);

}

}

private void printSeasonResults() {

System.out.println("\*\*\*\*\*\*\*\*\*RESULTS\*\*\*\*\*\*\*\*\*");

for (Team team : teams) {

System.out.println(team);

}

System.out.println("\nGames Played:");

for (Game game : games) {

System.out.println(game);

}

System.out.println("Hottest Temperature: " + hottestTemperature);

System.out.println("Average Temperature: " + (totalGamesPlayed == 0 ? 0 : (double) totalTemperature / totalGamesPlayed));

}

}

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import java.util.Scanner;

public class Scheduler {

private List<Team> teams;

private List<Game> games;

private int consecutiveFreezingWeeks;

private int hottestTemperature;

private int totalTemperature;

private int totalGamesPlayed;

public Scheduler() {

teams = new ArrayList<>();

games = new ArrayList<>();

consecutiveFreezingWeeks = 0;

hottestTemperature = Integer.MIN\_VALUE;

totalTemperature = 0;

totalGamesPlayed = 0;

// Initialize teams

teams.add(new Team("Team 1"));

teams.add(new Team("Team 2"));

teams.add(new Team("Team 3"));

teams.add(new Team("Team 4"));

}

public void startSeason() {

Scanner scanner = new Scanner(System.in);

Random random = new Random();

while (true) {

System.out.print("Enter temperature for this week: ");

int temperature = scanner.nextInt();

if (temperature <= 32) {

consecutiveFreezingWeeks++;

System.out.println("Too cold to play.");

} else {

consecutiveFreezingWeeks = 0;

playGames(temperature, random);

}

if (consecutiveFreezingWeeks >= 3) {

System.out.println("Season is over due to freezing temperatures.");

break;

}

}

printSeasonResults();

}

private void playGames(int temperature, Random random) {

totalGamesPlayed++;

totalTemperature += temperature;

if (temperature > hottestTemperature) {

hottestTemperature = temperature;

}

// Randomly choose teams for 2 games, ensuring no team plays against itself

for (int i = 0; i < 2; i++) {

Team homeTeam, awayTeam;

do {

homeTeam = teams.get(random.nextInt(teams.size()));

awayTeam = teams.get(random.nextInt(teams.size()));

} while (homeTeam == awayTeam);

Game game = new Game(homeTeam, awayTeam, temperature);

games.add(game);

System.out.println(game);

}

}

private void printSeasonResults() {

System.out.println("\*\*\*\*\*\*\*\*\*RESULTS\*\*\*\*\*\*\*\*\*");

for (Team team : teams) {

System.out.println(team);

}

System.out.println("\nGames Played:");

for (Game game : games) {

System.out.println(game);

}

System.out.println("Hottest Temperature: " + hottestTemperature);

System.out.println("Average Temperature: " + (totalGamesPlayed == 0 ? 0 : (double) totalTemperature / totalGamesPlayed));

}

}