Lab2. Programació Orientada a Objectes

Jhonatan Barcos Gambaro - P201 G09

Índice

1. Introducción

Este proyecto se basa en el seminario 2 realizado previamente en el cual ampliamos el diseño anterior con las siguientes clases: **Match** y **League**
A partir de dicho diseño, hemos realizado un programa en el cual se permita la creación, gestión y seguimiento de equipos, jugadores y partidos de fútbol.
Vayamos clase por clase detallando sus componentes:

- **Match:**

La clase Match es uno de los componentes fundamentales de nuestro sistema de gestión de la liga de fútbol. En esta clase, modelamos un partido de fútbol entre dos equipos, representados por las instancias de la clase Team. Contiene atributos que almacenan información sobre el equipo local (*homeTeam*) y el equipo visitante (*awayTeam*), así como el número de goles marcados por cada equipo y los jugadores que marcaron esos goles.

El constructor de la clase Match toma dos equipos como parámetros y crea una instancia de partido con estos equipos. Utilizamos un generador de números aleatorios para simular el marcador del partido, generando un número aleatorio de goles para cada equipo. Además, registramos los jugadores que marcaron los goles mediante listas vinculadas (*homeScorers* y *awayScorers*).

```java
public class Match {
    //atributes
    private Team homeTeam;
    private Team awayTeam;
    private int homeGoals;
    private int awayGoals;
    private LinkedList<Player> homeScorers = new LinkedList<Player>();
    private LinkedList<Player> awayScorers = new LinkedList<Player>();

    //constructor
    public Match(Team h, Team a) {
        this.homeTeam = h;
        this.awayTeam = a;
    }
```

La clase Match también proporciona métodos para simular el partido (*simulateMatch()*)

```java
public void simulateMatch(){
    System.out.println("\n" + "Simulating match between " + this.homeTeam.getName() + " and " + this.awayTeam.getName());
    //Generate random goals
    int n = 6;
    Random random = new Random();
    this.homeGoals = random.nextInt(n);
    this.awayGoals = random.nextInt(n);

   //Generate random scorers
    int k = this.homeTeam.getPlayers().size();
    int j = this.awayTeam.getPlayers().size();
    for(int i=0; i<this.homeGoals; i++){
        int r = random.nextInt(k);
        this.homeScorers.add(this.homeTeam.getPlayers().get(r));
    }
    for(int i=0; i<this.awayGoals; i++){
        int r = random.nextInt(j);
        this.awayScorers.add(this.awayTeam.getPlayers().get(r));
```

y para imprimir las estadísticas del partido (*printMatch()*). Durante la simulación, se generan goles aleatorios para ambos equipos y se registran los jugadores que marcaron esos goles. Las estadísticas del partido incluyen el nombre de los equipos, el número de goles marcados por cada equipo y la lista de goleadores de ambos equipos.

```java
  public void printMatch(){
      System.out.println("\n" + " Match simulated statics");
      System.out.println("- Home team: " + this.homeTeam.getName());
      System.out.println("- Away team: " + this.awayTeam.getName());
      System.out.println("- noHomeGoals: " + this.homeGoals);
      System.out.println("- noAwayGoals: " + this.awayGoals);
  }
```

- **League**

La clase League constituye el núcleo de nuestro sistema de gestión de la liga de fútbol virtual. En esta clase, modelamos una liga que alberga varios equipos representados por instancias de la clase Team. La clase League tiene atributos que contienen información crucial sobre la liga, como el nombre de la liga (name), el país en el que se juega la liga (country), el género de los equipos permitidos en la liga (gender), una lista de equipos que participan en la liga (teams) y una lista de partidos que se jugarán en la liga (matches).

```java
public class League {
    //attributes

    private String name;
    private Country country;
    private LinkedList<Team> teams = new LinkedList<Team>();
    private Gender gender;
    private LinkedList<Match> matches = new LinkedList<Match>();

    // constructor
    public League(String n, Country c, Gender g) {
        this.name = n;
        this.country = c;
        this.gender = g;
    }
```

El constructor de la clase League toma el nombre de la liga, el país y el género como parámetros, creando así una instancia de liga con estos detalles. Uno de los aspectos más fundamentales de esta clase es el método *generateMatches()*, el cual crea una serie de partidos para la liga, asegurando que cada equipo juegue contra todos los demás equipos al menos una vez.

```java
public void generateMatches(){
    for (int i = 0; i < teams.size(); i++) {
        for (int j = i+1; j < teams.size(); j++) {
            Match match1 = new Match(teams.get(i), teams.get(j));
            Match match2 = new Match(teams.get(j), teams.get(i));
            this.matches.add(match1);
            this.matches.add(match2);
```

Además, la clase League proporciona métodos para agregar (*addTeam()*) y eliminar *(removeTeam()*) equipos de la liga. Estos métodos aplican condiciones estrictas basadas en el país y el género de los equipos para garantizar una competición justa y equilibrada. La liga también puede simular los partidos

(simulateMatches()) y mostrar los resultados de los partidos (*printMatches()*) utilizando instancias de la clase Match.

```java
public void simulateMatches(){
    System.out.println(x:"\n==================================");
    System.out.println("Simulating matches of " + this.name);
    System.out.println(x:"==================================");
    for (Match match : this.matches) {
        match.simulateMatch();
        match.getHomeTeam().updateTeamStats(match);
        match.getAwayTeam().updateTeamStats(match);
    }
}

public void printMatches(){
    for (Match match: this.matches){
        match.printMatch();
```

2. *Descripción soluciones alternativas*

En dicha práctica no me he encontrado con posibles alternativas al código.
Hechos a destacar serían la aparición de un error "This 'method' is not defined to the class X" el cual a pesar de haberse implementado el código de forma correcta no conseguía evitar. Para solucionar dicho problema he tenido que duplicar la carpeta y volver a ejecutarla sin ningún tipo de error.

Por otra parte, me he encontrado con un pequeño error de código el cual he solucionado con rapidez pues al realizar un *updateStats()* tras simular los partidos llamaba en dos funciones diferentes al *updatePlayerStats()* haciendo así que las estadísticas de los jugadores estuvieran mal actualizadas.

3. *Conclusión*

En este experimento, hemos creado un emocionante escenario de fútbol virtual que encapsula la esencia de una verdadera liga. Utilizando la programación orientada a objetos, hemos modelado jugadores, equipos y ligas con gran detalle y realismo.

Primero, hemos establecido las bases con la creación de jugadores, representando tanto a estrellas veteranas como a jóvenes promesas, cada uno con sus habilidades y atributos únicos. Estos jugadores, provenientes de países icónicos como España y Francia, han sido la fuerza impulsora detrás de nuestros equipos.

```
//Create country instances
Country spain = new Country(n:"Spain");
Country france = new Country(n:"France");
```

```
//Create player instances
Player p1 = new Player(g:false, n:"Cristiano Ronaldo", a:35, spain);
Player p2 = new Player(g:false, n:"Lionel Messi", a:33, spain);
Player p3 = new Player(g:false, n:"Sergio Ramos", a:34, spain);
Player p4 = new Player(g:false, n:"Sergio Busquets", a:33, spain);
```

Luego, hemos estructurado estos jugadores en equipos tanto masculinos como femeninos. Esta inclusividad y diversidad no solo añaden complejidad a nuestra simulación, sino que también reflejan la realidad del deporte moderno.

```
Player p35 = new Player(g:true, n:"Wendie Renard", a:30, france);
Player p36 = new Player(g:true, n:"Amandine Henry", a:31, france);
Player p37 = new Player(g:true, n:"Eugénie Le Sommer", a:32, france);
```

El componente central de nuestro experimento ha sido la creación de ligas, cada una con sus reglas y restricciones. Hemos explorado las dinámicas de equipos puramente masculinos y femeninos, así como ligas mixtas. Estos escenarios han demostrado que la diversidad de género no solo es posible, sino también emocionante y competitiva.

```
//Create Male France Team
Team franceMaleTeam = new Team(n:"France Male Team", france, Team.Gender.Male);
franceMaleTeam.addPlayer(p24);
franceMaleTeam.addPlayer(p25);
franceMaleTeam.addPlayer(p26);
franceMaleTeam.addPlayer(p27);
franceMaleTeam.addPlayer(p28);
franceMaleTeam.addPlayer(p29);
franceMaleTeam.addPlayer(p30);
franceMaleTeam.addPlayer(p31);
franceMaleTeam.addPlayer(p32);
franceMaleTeam.addPlayer(p33);
franceMaleTeam.addPlayer(p34);
```

```java
// Create the France Female Team
Team franceFemaleTeam1 = new Team(n:"France Female Team 1", france, Team.Gender.Female);
franceFemaleTeam1.addPlayer(p35);
franceFemaleTeam1.addPlayer(p36);
franceFemaleTeam1.addPlayer(p37);
franceFemaleTeam1.addPlayer(p38);
franceFemaleTeam1.addPlayer(p39);
franceFemaleTeam1.addPlayer(p40);
franceFemaleTeam1.addPlayer(p41);
franceFemaleTeam1.addPlayer(p42);
franceFemaleTeam1.addPlayer(p43);
franceFemaleTeam1.addPlayer(p44);
franceFemaleTeam1.addPlayer(p45);

Team franceFemaleTeam2 = new Team(n:"France Female Team 2", france, Team.Gender.Female);
franceFemaleTeam2.addPlayer(p46);
franceFemaleTeam2.addPlayer(p47);
franceFemaleTeam2.addPlayer(p48);
franceFemaleTeam2.addPlayer(p49);
franceFemaleTeam2.addPlayer(p50);
franceFemaleTeam2.addPlayer(p51);
franceFemaleTeam2.addPlayer(p52);
franceFemaleTeam2.addPlayer(p53);
franceFemaleTeam2.addPlayer(p54);
franceFemaleTeam2.addPlayer(p55);
franceFemaleTeam2.addPlayer(p56);
```

```java
System.out.println(x:"\n================================");
System.out.println(x:"Create Mixed France League");
System.out.println(x:"================================");
League mxFrLg = new League(n:"Mixed France League", france, League.Gender.Mixed);
//Add teams to the league
mxFrLg.addTeam(franceMaleTeam);
mxFrLg.addTeam(franceFemaleTeam1);
mxFrLg.addTeam(franceFemaleTeam2);
```

```
================================
Create Mixed France League
================================
France Male Team added to the league Mixed France League
France Female Team 1 added to the league Mixed France League
France Female Team 2 added to the league Mixed France League
```

Durante la simulación, hemos generado partidos y evaluado el rendimiento de los equipos y jugadores.

```java
System.out.println(x:"\n==================================");
System.out.println(x:"Stats of 'france Male Team' and his players before Mixed France League");
System.out.println(x:"==================================");

franceMaleTeam.printStats();
franceMaleTeam.printPlayers();
for (Player p : franceMaleTeam.getPlayers()) {
    p.printStats();
}

mxFrLg.generateMatches();
mxFrLg.simulateMatches();
mxFrLg.printMatches();

    System.out.println(x:"\n==================================");
System.out.println(x:"Stats of 'france Male Team' and his players after Mixed France League");
System.out.println(x:"==================================");
franceMaleTeam.printStats();
franceMaleTeam.printPlayers();
for (Player p : franceMaleTeam.getPlayers()) {
    p.printStats();
}
```

```
==================================
Stats of 'france Male Team' and his players before Mixed France League
==================================

France Male Team statics
- noMatches: 0
- noWins: 0
- noTies: 0
- noLosses: 0
- noGoalsScored: 0
- noGoalsReceived: 0

Printing players of France Male Team
- Kylian Mbappé
- Antoine Griezmann
- Paul Pogba
- Hugo Lloris
- Raphaël Varane
- Benjamin Pavard
- Lucas Hernández
- N'Golo Kanté
- Olivier Giroud
- Adrien Rabiot
- Kingsley Coman

Kylian Mbappé stats
- noMatches: 0
- noTrackles: 0
- noPasses: 0
- noShots: 0
- noAssists: 0
- noGoals: 0
```

```
==================================
Stats of 'france Male Team' and his players after Mixed France League
==================================

France Male Team statics
- noMatches: 4
- noWins: 3
- noTies: 2
- noLosses: 0
- noGoalsScored: 8
- noGoalsReceived: 9

Printing players of France Male Team
- Kylian Mbappé
- Antoine Griezmann
- Paul Pogba
- Hugo Lloris
- Raphaël Varane
- Benjamin Pavard
- Lucas Hernández
- N'Golo Kanté
- Olivier Giroud
- Adrien Rabiot
- Kingsley Coman

Kylian Mbappé stats
- noMatches: 4
- noTrackles: 0
- noPasses: 0
- noShots: 0
- noAssists: 0
- noGoals: 2
```