

Lab3. Programació Orientada a Objectes

Jhonatan Barcos Gambaro - P201 G09

Índice

1. Introducción
2. Descripción soluciones alternativas
3. Conclusión

1. Introducción

El presente informe se centra en el análisis detallado de un sistema de gestión de fútbol virtual, implementado mediante la programación orientada a objetos (POO). Dos conceptos fundamentales de la POO que juegan un papel central en este sistema son la herencia y el polimorfismo. La herencia permite la creación de nuevas clases basadas en clases existentes, lo que fomenta la reutilización del código y establece relaciones entre las clases. Por otro lado, el polimorfismo permite a los objetos de diferentes clases ser tratados como objetos de una clase común, lo que facilita la flexibilidad y extensibilidad del código. Para ello, las clases más relevantes que hemos definido han sido las siguientes::

Clases extendidas de Player

➤ Clase Outfielder

Atributos:

- noTackles (int): Número de tackles realizados por el jugador.
- noPasses (int): Número de pases realizados por el jugador.
- noShots (int): Número de tiros realizados por el jugador.
- noAssists (int): Número de asistencias realizadas por el jugador.
- noGoals (int): Número de goles marcados por el jugador.

Métodos:

- updatePlayerStats(Match m): Método heredado de la clase Player que actualiza las estadísticas del jugador después de un partido. Incrementa noTackles, noPasses, noShots, noAssists y noGoals según el desempeño del jugador en el partido.

➤ Clase Goalkeeper

Atributos:

- noSaves (int): Número de tiros atajados por el portero.
- noGoalsLet (int): Número de goles permitidos por el portero.

Métodos:

- `updatePlayerStats(Match m)`: Método heredado de la clase `Player` que actualiza las estadísticas del portero después de un partido. Incrementa `noSaves` y `noGoalsLet` según el desempeño del portero en el partido.

Clases extendidas de `Competition`

➤ Clase `Cup` :

Atributos:

- `tr (LinkedList<Team>)`: Lista de equipos para la ronda actual del torneo.
- `mr (LinkedList<Match>)`: Lista de partidos para la ronda actual del torneo.

Métodos:

- `generateMatches()`: Genera los partidos para cada ronda del torneo y simula los partidos hasta que se obtiene un ganador. Actualiza la lista de equipos y partidos en cada ronda.
- `simulateMatches()`: Simula los partidos de la ronda actual y determina los equipos ganadores para la siguiente ronda.

➤ Clase `League`:

Atributos:

- Ninguno adicional a los heredados de la clase base `Competition`.

Métodos:

- `generateMatches()`: Genera los partidos de la liga. Crea un partido entre cada par de equipos en la liga y agrega esos partidos a la lista de partidos de la liga.
- Métodos heredados de la clase base `Competition`: `addTeam(Team team)`, `removeTeam(Team team)`, `generateMatches()`, `simulateMatches()`, `printMatches()`.

➤ Clase `GroupPlay`:

Atributos:

- `noGroups (int)`: Número de grupos en el torneo.
- `groups (LinkedList<League>)`: Lista de ligas que representan los grupos en el torneo.

Métodos:

- `generateGroups()`: Genera grupos a partir de los equipos disponibles y asigna equipos a cada grupo de manera aleatoria. Cada grupo tiene al menos dos equipos.
- `printGroups()`: Imprime los equipos que están en cada grupo.
- Métodos heredados de la clase base `Competition`: `addTeam(Team team)`, `removeTeam(Team team)`, `generateMatches()`, `simulateMatches()`, `printMatches()`.

2. Descripción soluciones alternativas

A raíz de ampliar nuestros conocimientos en clases posteriores nos hemos dado cuenta de posibles alternativas a nuestro código como por ejemplo:

- Implementación de Interfaces: En lugar de utilizar herencia directa, podríamos haber optado por implementar interfaces para definir ciertos comportamientos comunes entre las clases. Por ejemplo, podríamos haber tenido una interfaz llamada Scorer que define métodos para gestionar goles y que es implementada tanto por Outfielder como por Goalkeeper.
-
- Uso de Clases Abstractas: Podríamos haber introducido una clase abstracta ScoringPlayer que representa jugadores que pueden marcar goles. Las clases Outfielder y Goalkeeper podrían heredar de esta clase abstracta, permitiendo una mayor cohesión entre las clases relacionadas con el marcador.

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);
```

```
// 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);
```

Seguidamente hemos creado una competición base a partir de la cual realizar las distintas pruebas de cada uno de los tipos de competición existente:

```
//Create competition instances
System.out.println(x:"\n=====");
System.out.println(x:"Create Mixed National France Competition");
System.out.println(x:"=====");
Competition mxFrNatCmt = new Competition(cl:true, n:"Mixed National France Competition", france, Competition.Gender.Mixed);
mxFrNatCmt.addTeam(spainMaleTeam1);
mxFrNatCmt.addTeam(franceMaleTeam);
mxFrNatCmt.addTeam(franceFemaleTeam1);
mxFrNatCmt.addTeam(franceFemaleTeam2);
mxFrNatCmt.addTeam(franceFemaleTeam3);
```

```
=====
Create Mixed National France Competition
=====
Spain Male Team 1 can't be added to the competition Mixed National France Competition cause of country issues
France Male Team added to the competition Mixed National France Competition
France Female Team 1 added to the competition Mixed National France Competition
France Female Team 2 added to the competition Mixed National France Competition
France Female Team 3 added to the competition Mixed National France Competition
```

En primer lugar, hemos simulado una Liga

```
//Create different types of competitions
//1. League
System.out.println(x:"\n=====");
System.out.println(x:"Create League based on Competition");
System.out.println(x:"=====");
League league = new League(mxFrNatCmt.clubs,n:"League",mxFrNatCmt.country,mxFrNatCmt.gender);
for(Team team : mxFrNatCmt.getTeams()){
    league.addTeam(team);
}

league.generateMatches();
league.simulateMatches();
league.printMatches();
```

```
=====
Create League based on Competition
=====
France Male Team added to the competition League
France Female Team 1 added to the competition League
France Female Team 2 added to the competition League
France Female Team 3 added to the competition League

=====
Simulating matches of League
=====

Simulating match between France Male Team and France Female Team 1
The winner of the match is France Female Team 1

Simulating match between France Female Team 1 and France Male Team
The winner of the match is France Male Team
```

(...)

Seguidamente, hemos simulado un Group Play

```
//2. GroupPlay
System.out.println(x:"\n=====");
System.out.println(x:"Create GroupPlay based on Competition");
System.out.println(x:"=====");
GroupPlay groupPlay = new GroupPlay(mxFrNatCmt.clubs,n:"Group Play",mxFrNatCmt.country,mxFrNatCmt.gender);
for(Team team : mxFrNatCmt.getTeams()){
    groupPlay.addTeam(team);
}
groupPlay.generateGroups();
groupPlay.printGroups();
groupPlay.generateMatches();
groupPlay.simulateMatches();
groupPlay.printMatches();
```

```
=====
Create GroupPlay based on Competition
=====
France Male Team added to the competition Group Play
France Female Team 1 added to the competition Group Play
France Female Team 2 added to the competition Group Play
France Female Team 3 added to the competition Group Play

=====
Printing Group Play groups:
=====

League 1:
France Female Team 2
France Female Team 3

League 2:
France Male Team
France Female Team 1
```

(...)

Finalmente, una copa

```
//3. Cup
System.out.println(x:"\n=====");
System.out.println(x:"Create Cup based on Competition");
System.out.println(x:"=====");

Cup cup = new Cup(mxFrNatCmt.clubs,n:"World Cup",mxFrNatCmt.country,mxFrNatCmt.gender);
for(Team team : mxFrNatCmt.getTeams()){
    cup.addTeam(team);
}
cup.generateMatches();
```

=====

Create Cup based on Competition

=====

France Male Team added to the competition World Cup

France Female Team 1 added to the competition World Cup

France Female Team 2 added to the competition World Cup

France Female Team 3 added to the competition World Cup

Simulating match between France Female Team 2 and France Female Team 1

The winner of the match is France Female Team 2

Simulating match between France Male Team and France Female Team 3

The winner of the match is France Male Team

Simulating match between France Female Team 2 and France Male Team

The winner of the match is France Male Team

The winner of World Cup is France Male Team

=====