

Lab4. 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:

Player

Atributos:

- female (boolean): Indica si el jugador es femenino o masculino.
- name (String): Nombre del jugador.
- age (int): Edad del jugador.
- nationality (Country): País de origen del jugador.
- noMatches (int): Número de partidos jugados por el jugador.
- stats (HashMap<Competition, PlayerStats>): Estadísticas del jugador en las competiciones.

Métodos:

- Constructor para inicializar un jugador con su información.
- Métodos getters para obtener la información del jugador.
- update(Competition c, Match m): Actualiza las estadísticas del jugador en una competición.

Goalkeeper (extiende de Player)

Representa a un jugador de fútbol portero.

Implementa métodos específicos para el portero.

Outfielder (extiende de Player)

Representa a un jugador de fútbol de campo.

Implementa métodos específicos para jugadores de campo.

(abstract) PlayerStats

Clase abstracta que representa las estadísticas de un jugador.

Define métodos abstractos para actualizar y mostrar estadísticas.

GoalkeeperStats (extiende de PlayerStats)

Implementa estadísticas específicas para el portero.

Actualiza y muestra las estadísticas del portero.

OutfielderStats (extiende de PlayerStats)

Implementa estadísticas específicas para jugadores de campo.

Actualiza y muestra las estadísticas de los jugadores de campo.

Team

Atributos:

- name (String): Nombre del equipo.
- country (Country): País del equipo.
- genderClass (Gender): Clasificación del equipo (Masculino, Femenino o Mixto).
- players (LinkedList<Player>): Lista de jugadores en el equipo.
- stats (HashMap<Competition, TeamStats>): Estadísticas del equipo en las competiciones.

Métodos:

- Métodos para agregar, eliminar, y obtener jugadores.
- update(Competition c, Match m): Actualiza las estadísticas del equipo en una competición.
- Métodos para imprimir los jugadores y validar adiciones o eliminaciones.

TeamStats

Representa las estadísticas de un equipo en una competición.

Actualiza y muestra las estadísticas del equipo.

Competition

Clase que representa una competición genérica.

Define métodos abstractos para generar partidos y simularlos.

League (extiende de Competition)

Representa una competición de liga.
Genera y simula partidos de una liga.
Imprime la tabla de clasificación y los goleadores.

(interface) Comparable<TeamStats> / Comparable<PlayerStats>

Implementación de la interfaz Comparable para poder ordenar listas.
Utilizado para los métodos printTable() y printGoalScorers()

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 realizaremos una simulación de una liga:

```
//Create diferent types of competitions
//1. League
System.out.println(x:"\n=====");
System.out.println(x:"Create League based on Competition");
System.out.println(x:"=====");
Competition league = new League(cl:true, n:"League Competition", france, Competition.Gender.Mixed);
league.addTeam(spainMaleTeam1);
league.addTeam(franceMaleTeam);
league.addTeam(franceFemaleTeam1);
league.addTeam(franceFemaleTeam2);
league.addTeam(franceFemaleTeam3);
```

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

Imprimimos la tabla de estadísticas de la liga basada en la implementación de la interfaz Comparable.

```
league.printTable();
```

```
=====
Printing table of League Competition
=====
```

Printing stats of France Male Team

Matches: 6
Wins: 1
Ties: 0
Losses: 5
Goals Scored: 10
Goals Received: 22

Printing stats of France Female Team 3

Matches: 6
Wins: 3
Ties: 0
Losses: 3
Goals Scored: 18
Goals Received: 14

Printing stats of France Female Team 2

Matches: 6
Wins: 3
Ties: 2
Losses: 1
Goals Scored: 18
Goals Received: 14

Printing stats of France Female Team 1

Matches: 6
Wins: 3
Ties: 2
Losses: 1
Goals Scored: 22
Goals Received: 18

Finalmente, imprimimos el top de goleadores de dicha liga.

```
league.printGoalScorers(k:3);
```

```
=====
Printing top 3 scorers of League Competition
=====

Printing stats of player: Marion Torrent from France
Number of goals: 4
Number of assists: 0
Number of shots: 0
Number of passes: 0
Number of trackles: 0

Printing stats of player: Elise Bussaglia from France
Number of goals: 4
Number of assists: 0
Number of shots: 0
Number of passes: 0
Number of trackles: 0

Printing stats of player: Grace Geyoro from France
Number of goals: 4
Number of assists: 0
Number of shots: 0
Number of passes: 0
Number of trackles: 0
```