

Laboratorio 1

1. ¿Qué debe hacer el programa y cuáles son las clases y métodos que debes definir?

Para la práctica el objetivo era empezar a añadir jugadores y equipos siguiendo las clases creadas en el seminario, seguidamente el programa debe de hacer print de los stats de los equipos y jugadores. Lo primero que hemos hecho ha sido crear 4 archivos, 3 por las clases y uno por el main.

La primera clase ha sido country, permite crear objetos que representan países y proporciona métodos para obtener y establecer su nombre. La segunda clase ha sido player con atributos como género, nombre, edad, nacionalidad, y estadísticas de rendimiento (ya vistas en el seminario). El constructor inicializa estos atributos y el método updateStatistics actualiza las estadísticas del jugador después de un partido. La clase también proporciona varios métodos para acceder a los atributos y estadísticas del jugador, y un método PrintStats para imprimir información sobre las estadísticas del jugador. Finalmente la última de las clases creadas ha sido equipo los atributos incluyen el nombre del equipo, el país al que pertenece, el género (masculino, femenino o mixto), una lista de jugadores, estadísticas de partidos (partidos jugados, victorias, empates, derrotas, goles a favor, goles en contra y puntos), y un constructor que inicializa estos atributos. La clase proporciona métodos para agregar y eliminar jugadores del equipo comprobando su género, actualizar estadísticas del equipo después de un partido y métodos para acceder a los atributos y estadísticas del equipo. Además, como en la clase jugador existe la opción de hacer print de las estadísticas.

Finalmente en el main creamos jugadores, paises y equipos pasando directamente los atributos del constructor manualmente.

El programa se encarga de gestionar equipos y jugadores relacionados con el fútbol. Para ello, se definen las siguientes clases: Country (para representar países), Player (para representar jugadores), y Team (para representar equipos). Los métodos que se deben implementar incluyen la creación de instancias de jugadores, países y equipos, la asignación de jugadores a equipos, el seguimiento de estadísticas de equipos y jugadores, y la presentación de información sobre los equipos.

2. ¿Qué soluciones alternativas se consideraron y por qué se eligió la solución actual?

En el código proporcionado, se ha optado por utilizar una única clase Player con un atributo para indicar el género (masculino o femenino). Otra posible solución podría haber sido crear una jerarquía de clases con una clase base "Persona" y subclases "JugadorMasculino" y "JugadorFemenino". La decisión de elegir la solución actual se basó por la simpleza y facilidad del código, ya que no requería la creación de subclases adicionales y era más fácil de entender. Además, esta solución permite una mayor flexibilidad si se desea extender el programa en el futuro para manejar otros géneros o roles.

3. ¿Cómo funcionó la solución en la práctica? ¿Hubo dificultades en la implementación o dudas?

La solución funcionó en general en la práctica, permitiendo la creación de instancias de países, jugadores y equipos, y el seguimiento de estadísticas. Sin embargo, se identificó un problema al agregar jugadores a equipos. La dificultad inicial se debió a que el constructor de la clase Team estaba configurado para asignar automáticamente el género del equipo como femenino (Team.Gender.FEMENINO), lo que causaba problemas al intentar agregar jugadores masculinos a un equipo femenino. La solución a este problema fue ajustar el constructor de la clase Team para que el género del equipo se asignará de acuerdo con el género del jugador que se agregara al equipo, en lugar de asignar automáticamente un género femenino.

La solución implementada en el programa funcionó en la práctica de manera efectiva en la mayoría de los casos. Se logró la creación de instancias de países, jugadores y equipos, y se llevaron a cabo las operaciones de seguimiento de estadísticas de manera exitosa. Además, se logró resolver el problema inicial que se mencionó, relacionado con la asignación del género al equipo en el constructor de la clase Team.

En cuanto a las dificultades, el único problema que se encontró fue el mencionado anteriormente, donde se asignaba incorrectamente el género de los equipos en el constructor de la clase Team. Una vez identificado este problema, se corrigió de manera efectiva cambiando la lógica de asignación de género en el constructor del equipo para que se ajustara según el género del jugador agregado al equipo. No hubo otras dificultades significativas en la implementación del programa, y este funcionó de manera satisfactoria.