Lab1

INTRODUCCION

En este laboratorio nos hemos encontrado frente a la problemática en la que necesitábamos un programa capaz de crear equipos de futbol en función del género, así como de otros aspectos como el país del equipo. Al mismo tiempo debíamos implementar de alguna manera el elemento del jugador con sus respectivos atributos como la edad, nombre, género, etc.

Nuestro programa se define en esencia por tres objectos de los cuales uno se nos viene dado (“country”).

En primer lugar, tenemos el objecto jugador el cual tiene diversos atributos como el nombre, edad, país de residencia, sexo (en formato booleano) y un seguido de enteros como en número de partidos jugados, así como número de pases, chutes, asistencias, goles i faltas. En el objecto “Player” describimos cuatro “getters”, concretamente el de género, el nombre i la nacionalidad. El método “getGender” funciona de una manera mas peculiar que los demás. Esta preguntará a la instancia de “Player” si es hembra en cullo caso devolverá un valor booleano. También declaramos su constructor como método y otro el cual será nombrado “printStats” e imprimirá en consola todos los atributos del jugador.

En otro lado tenemos al objecto “Team” que describe en esencia una lista de jugadores. Este objecto tiene diversos atributos como el “name” i un conjunto de enteros que definen en número de partidos jugados, así como los ganados, perdidos, empatados i número de goles hechos contra el oponente i recibidos por estos. En este objecto declaramos diferentes métodos como su constructor i los “getters” principales, “getName”, “getName”, “getGender”. La clase declara el género como un tipo de dato “enum GENDER” el cual implicará que solo existirán “MALE”, “FEMALE” y “MIXED”. De esta forma ya se puede describir la que por diferencia nos ha sido el atributo más difícil de implementar, el “addPlayer” que recibe como parámetro un “Player”, compara de forma booleana si el género del jugador es el mismo que el del equipo. Por eso mismo primero preguntamos al objecto “Team” si género es femenino para así traducirlo en forma booleana i poder trabajar con él. Como antónimo a este atributo tenemos también el “removePlayer” el cual elimina un jugador en cuestión de la lista de jugadores. Finalmente, el método “printTeam” imprime por consola la lista de todos los jugadores mediante un bucle “for”.

Finalmente, lo hemos enlazado todo mediante un “package” y en otro documento .java hemos implementado el main. En este laboratorio el main solo nos sirve como herramienta para testear y demostrar por consola todas las implementaciones realizadas en nuestro código, así como mostrar que hemos logrado conseguir esos objetivos planteados en un principio

METODOLOGIA

En un primer momento planteamos el laboratorio de una forma poco eficaz ya que al ser nuestra primera implementación de un código de problema resoluble en programación orientada a objectos la complejidad ha sido mayor. El principal desafío fue la comunicación entre objectos, una vez definida la estructura básica de cada objeto.

Imatge que conté text, captura de pantalla, Font, nombre

Descripció generada automàticamentEl principal reto fue en la función “addPlayer”. Planteamos muchas formas resolutivas, aunque incorrectas de diseñar el algoritmo. Ya que del objeto “Player” teníamos en dato booleano si era hembra o no lo era. Y por parte del objecto “Team” se definió un "enum GENDER" para identificar los géneros “MALE”, “FEMALE” y “MIXED”. Así que decidimos “convertir” en formato booleano si el género del equipo era femenino o no lo era con las siguientes líneas.

Ilustración 1

Después hacemos una comparación entre el booleano del género del equipo i el género del jugador para saber si podemos asignar o no el jugador con el equipo solicitado.

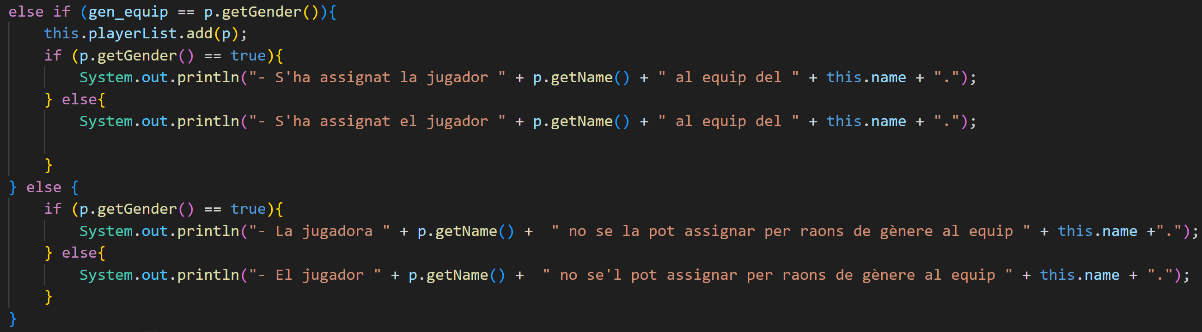


Ilustración 2

Otro desafío fue la función “playMatch”, en la cual se implementó un enfoque eficiente para contabilizar los partidos ganados y perdidos en función de los goles marcados y recibidos.

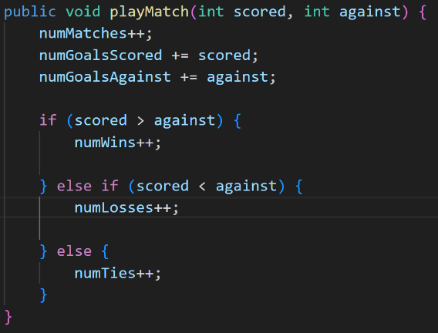


Ilustración 3

En cuanto a la metodología utilizada, se reconoce que la implementación inicial del laboratorio presentó dificultades debido a la falta de experiencia previa en programación orientada a objetos. Destacamos la importancia de establecer una estructura clara para cada objeto y definir una comunicación efectiva entre ellos. En general, el código logró cumplir con los objetivos planteados inicialmente y se utilizó el método "main" para probar y demostrar las implementaciones realizadas. Aunque se enfrentaron desafíos significativos

CONCLUSIÓN

**El** **propósito** **de** este **laboratorio** **fue** **crear** un programa orientado a objetos para **administrar** **un** **equipo** de fútbol y sus respectivos jugadores. La implementación inicial presentó **grandes** desafíos debido a la falta de **conocimientos** **previos** en programación orientada a objetos. La principal dificultad **se** **produjo** **a** **la** **hora** **de** establecer comunicación entre **diferentes** objetos, especialmente al **utilizar** la función “**addPlayer”** y **la** correspondiente lógica de comparación de **género** entre jugadores y equipos.

En particular, la estructura básica de los objetos **“Jugador”** y **“Equipo”** **se** **ha** **implementado** **exitosamente** con métodos y atributos que permiten la gestión de jugadores y equiposen **un** contexto **de** fútbol. La función “**addPlayer”** fue uno de los elementos más **difíciles** de implementar, ya que **requiere** una **consideración** cuidadosa **del** **género** **del** **jugador** y **del** **equipo,** **y** **utiliza** conversiones entre valores booleanos y enumeraciones.

Además, **encontramos** dificultades **al** **implementar** la función “**playMatch”,** **ya** que requería un enfoque eficiente para **calcular el** **resultado** **de** **un** **partido** **basándose** en los goles marcados y **concedidos.**

**Mediante la clase “main” hemos ido testeando el programa. En primer lugar, ejecutamos hasta 19 constructores. A continuación, actualizamos los valores de tres de los jugadores e imprimimos las stats de estos para comprobar que funcione correctamente. Finalmente, creamos cada equipo y al mismo tiempo probamos de meter jugadores en equipos de sexos diferentes para tal de comprobar si nuestro programa lo rechaza. Para comprobar la buena implementación de los equipos los imprimimos a todos.**

Aunque el proceso inicial presentó **desafíos,** el programa final logró **alcanzar** **las** **metas** **establecidas** **y** **demostró** la capacidad de gestionar **eficazmente** equipos y jugadores de **fútbol.** **Es** **bien** **sabido** **lo** **importante** **que** **es** establecer una estructura clara para cada objeto y asegurar una comunicación efectiva entre ellos para **asegurar** **el** **buen** **funcionamiento** **de** un programa.