

Lab4

INTRODUCCIÓN

En este laboratorio hemos actualizado nuestro código enfocándonos en la eficiencia del mismo mediante la implementación de nuevos objetos. Hemos diseñado una serie de objetos los cuales realizan la funcionalidad de almacenar i gestionar las estadísticas de diferentes objetos ya declarados des de un principio de nuestro programa. En nuestro código encontraremos el objeto los objetos GoalKeeperStats, OutfielderStats, PlayerStats y TeamStats. Todos cumplen el mismo propósito, pero cada uno con sus singularidades.

En primer lugar, tenemos el PlayerStats, padre de GoalKeeperStats y OutfielderStats y en este caso clase abstracta. Definimos PlayerStats como una clase abstracta ya que consta de un método abstracto, printStats, el cual se acabará de definir en cada hijo. Por otro lado, se implementa un nuevo método que se repite en TeamStats, y los hijos de PlayerStats, que consiste en actualizar las estadísticas. Este método se llama updateStats y siempre recibirá por parámetro un objeto de clase Match.

En siguiente lugar se implementa tanto en Team como en Player un nuevo atributo llamado "stats". Este será un diccionario que contendrá vectores de dos elementos, el primero un objeto Competition i el segundo un objeto del tipo "TeamStats" o "PlayerStats" en función de si estamos hablando del Team o Player respectivamente. Este atributo nos permitirá almacenar todas las competiciones de un equipo o jugador. Estos dos objetos también contemplan un nuevo método que sería el "update" el cual añade nuevas competiciones a las estadísticas. En el caso del Player también se le define el "getStats" que no es más complejo que un "getter" de las mismas estadísticas.

Todos aquellos atributos que antes permanecían en los objetos Team o Player que almacenaban las estadísticas como "numMatches", "numWins", etc. Ahora pertenecen a sus respectivos objetos de estadísticas.

Finalmente terminamos de definir los métodos "printTables" de las competiciones que aún no se habían definido.

METODOLOGÍA

En el transcurso de este laboratorio, nos hemos sumergido en diversos caminos, no tanto por nuestra propia versatilidad como programadores, sino más bien debido a ciertas lagunas en nuestra comprensión conceptual. En una primera instancia, nos hemos adentrado en la implementación de interfaces, en este caso, la interfaz "Comparable".

Esta interfaz nos brinda la capacidad de incorporar un método altamente especializado, el `compareTo`, el cual ha sido definido en las clases `TeamStats`, `GoalKeeperStats` y `OutfielderStats`. Dicho método desempeña un papel crucial al posibilitar la comparación entre dos objetos, permitiéndonos determinar clasificaciones en términos generales. La implementación de una interfaz, en última instancia, se traduce en trabajar con una clase abstracta que únicamente contiene métodos abstractos, ofreciendo así una estructura sólida y coherente para el desarrollo de nuestro sistema orientado a objetos.

CONCLUSIÓN

En conclusión, el laboratorio 4 ha representado un paso significativo en la evolución de nuestro código, centrándonos en la eficiencia y la gestión de estadísticas a través de la implementación de nuevos objetos. La introducción de los objetos `GoalKeeperStats`, `OutfielderStats`, `PlayerStats` y `TeamStats` ha permitido una mayor modularidad y claridad en la organización de nuestro programa.

La jerarquía de clases, con `PlayerStats` como clase abstracta y padre de `GoalKeeperStats` y `OutfielderStats`, ha proporcionado una estructura coherente para la gestión de estadísticas de jugadores. La implementación de métodos abstractos, como `printStats`, asegura la consistencia en la manipulación de datos entre las diferentes subclases.

A pesar de los avances, enfrentamos desafíos conceptuales que nos llevaron a explorar diversas soluciones y ajustar nuestro enfoque. La implementación de la interfaz "Comparable" ha sido un ejemplo destacado.

En resumen, el laboratorio 4 ha sido una experiencia enriquecedora que no solo ha mejorado la eficiencia de nuestro código, sino que también ha fortalecido nuestra comprensión de los principios fundamentales de la programación orientada a objetos. Estamos mejor equipados para abordar desafíos futuros y continuar refinando nuestro sistema.