

INFORME LAB01

Introducció

En aquesta primera pràctica de l'assignatura, se'ns ha demanat crear un programa on s'afegeixin jugadors (amb unes certes característiques) a equips. Per a dur a terme aquest objectiu, hem creat tres classes (sense incloure el Main):

- La primera és "Country", que permet obtenir l'origen (el país) del jugador i de l'equip.
- La segona, "Player", conté totes les característiques del jugador: edat, país d'origen i sexe, a més un seguit d'estadístiques que s'actualitzen quan aquest està inclòs en algun equip.
 - o Els mètodes emprats en aquesta classe són: el constructor, on s'inicien les variables del país, sexe i nacionalitat, "update(...)", on s'actualitzen les dades del jugador cada partit) i "printStats()", on s'imprimeixen les estadístiques del jugador.
- La tercera i última classe és "Team", que defineix les característiques d'un equip. De fet, els atributs que defineixen a aquesta classe són els mateixos que els de la classe "Player" però afegint una *Linkedlist*, on s'inclouen els jugadors que té cada equip.
 - o Els mètodes que trobem en aquesta classe són: el constructor, on s'inicien les mateixes variables que a "Player" i la *Linkedlist*, addPlayer(Player p) i removePlayer(Player p), que afegeixen i eliminen jugadors d'un equip respectivament, playMatch(...), que actualitza les victòries i derrotes d'un equip i finalment, printStats(), que imprimeix les estadístiques de l'equip i (ens permetrà comprovar que s'han afegit correctament).

Descripció de les solucions emprades

A l'hora d'elaborar el codi en conjunt, s'ha seguit el següent ordre: Primerament s'ha creat la classe "Player" i s'han definit els seus atributs i mètodes i després, s'ha creat la última classe, "Team", definit els seus atributs i mètodes. Cal remarcar que la classe "Country" ja estava creada, si no hagués sigut la primera en haver-se creat. Aquest ordre de creació de classes és l'únic possible a seguir, donat que "Team" és el contenidor de "Player", és a dir que segueixen una relació d'agregació.



En aquest primer treball, a nivell general no hi havien moltes solucions distintes possibles, degut a que era un codi relativament senzill amb una relació entre classes molt visible. Tot i que, a nivell dels mètodes pròpiament, sempre hi ha petits matisos que cada programador fa diferent. N'és un exemple el següent mètode.

```
//mètode per a afegir jugador a l'equip
public void addPlayer(Player p){
    if (gender==Gender.MIXTED){//si el gènere es mixte, qualsevol jugador podrà ser afegit a l'equip
        players.add(p);
        System.out.println("Si que es pot afegir el jugador/a: "+p.getName()+" a l'equip mixte."+" amb edat "+p.getAge());
    }
    else if (gender == Gender.MALE && !p.isFemale()) {
        //si no és mixte, haurem de mirar si el gènere de l'equip "gender" coincideix amb el genere MALE.
        //Si coincideixen voldrem mirar si a més el genere del jugador es també masculí, fet que ho obtindrem si la funció isFemale()
        //Per això utilitzarem les portes lògiques AND
        players.add(p);
        System.out.println("Si que es pot afegir el jugador/a: " + p.getName()+" amb edat "+p.getAge());
    }
    else if (gender == Gender.FEMALE && p.isFemale()) {
        ///la mateixa condició que abans, però ara en comptes de què isFemale() sigui false mirarem si és true
        players.add(p);
        System.out.println("Si que es pot afegir la jugador/a: " + p.getName()+" amb edat "+p.getAge());
    }
    else {//si no es compleixen les condicions anteriors voldrà dir que no podrà ser afegit a l'equip perquè són generes diferent
        System.out.println(x:"No es pot afegir ja que són generes diferents");
    }
}
```

Aquest mètode forma part de la classe “Team” i afegeix un jugador nou a l’equip. Primer, però, s’ha de comprovar si es pot afegir a l’equip o no. En comptes de fer la comprovació directament en aquest mètode, es va contemplar la possibilitat de fer-la en un altre. Finalment es va decidir que es fes tot a la mateixa funció perquè aquesta comprovació només s’utilitza en aquesta part del codi. Si fos més recurrent, segurament s’hagués fet un mètode que servís només per comprovar si el gènere de l’equip coincideix amb el del jugador.

Conclusió

Un cop definida la classe “Player” amb els seus atributs i mètodes, ens hem assegurat de que no hi havia cap error de compilació. Per això, simplement hem comprovat que les estadístiques, donat un jugador en específic, s’imprimien correctament. Després d’haver creat la classe “Team” hem fet el mateix, primer donat un equip en específic i després per tots els equips definits al Main.

Com s'ha mencionat abans, no era un codi gaire complex, sinó que era una pràctica orientada a la familiarització de Java i de l'assignatura en sí. Per tant, no hem tingut gaires errors a l'hora de programar.