

AJ séance 6

Objectifs :

- Être capable d'implémenter correctement les associations
- Savoir écrire des classes énumérées et les utiliser correctement.

Thèmes abordés :

- Association
- Enumérés

Afin de faciliter l'organisation des interclubs, la fédération échiquéenne désire se munir d'un logiciel de gestion des divisions.

L'analyse a déterminé les classes `Division`, `Equipe`, `Club`, `Joueur` :

- Une division stocke un `nom` (1, 2A, 2B,...) et un nombre de joueurs par `Equipe`.
- Une équipe stocke son `numero`.
- Un club stocke son `nom`.
- Un joueur stocke son `nom`, son `prenom`, sa cote ELO¹ (`elo`) et son niveau. La cote ELO d'un joueur classé doit être supérieure ou égale à 1000. Le niveau du joueur se trouve sur base de sa cote ELO. On implémentera les différents niveaux existant à l'aide d'un énuméré `Niveau`. Vous trouverez plus de détails à ce sujet dans la suite de cet énoncé.

À cela s'ajoutent un certain nombre de champs liés aux associations :

- Une `Equipe` doit faire partie d'une et une seule `Division` qui elle garde la liste de ses 12 `Equipes`.
- Une `Equipe` doit faire partie d'un et d'un seul `Club` qui lui garde la liste de ses `Equipes`. Cette association est simplifiée par le fait que le `Club` est passé dans le constructeur d'`Equipe` et qu'il ne peut pas être modifié. Le numéro de l'`Equipe` sera attribué en fonction du nombre d'`Equipes` déjà présentes dans le club (s'il y a déjà 3 équipes inscrites pour ce club, l'équipe recevra le numéro 4). Les équipes doivent être gardées dans l'ordre dans lequel elles ont été créées.
- Un `Joueur` doit faire partie d'un et un seul `Club` qui lui garde la liste de ses `Joueurs`. C'est une association tout-à-fait classique.
- Un `Joueur` peut faire partie d'une `Equipe` qui elle garde la liste de ses `Joueurs`.
- Dessinez avant toute chose, le **diagramme de classes** reprenant les 4 classes citées plus haut en y indiquant les attributs et les associations. Pour ces dernières, précisez les multiplicités. Dans l'énoncé, à chaque fois qu'il est indiqué une liste, c'est à vous de choisir le type de collection qui convient le mieux pour faire ce qui est demandé.

Faites valider votre diagramme avant de commencer à programmer.

Quelques précisions nécessaires à l'implémentation des diverses associations :

- L'association `Equipe` – `Division` est tout-à-fait classique. Elle correspond exactement à la présentation théorique.
- L'association `Equipe` – `Club` est simplifiée par le fait que le `Club` est passé dans le constructeur d'`Equipe` et qu'il ne peut pas être modifié. Le numéro de l'`Equipe` ne sera pas passé au constructeur mais sera attribué en fonction du nombre d'`Equipes` déjà présentes dans le `Club` (s'il y a déjà 3 équipes inscrites pour ce club, l'équipe recevra le numéro 4).
- L'association `Club` – `Joueur` est classique. Toutefois la suppression d'un joueur qui ferait partie d'une équipe doit entraîner la suppression de ce joueur de cette équipe.
- L'association `Equipe` – `Joueur` est soumise à la contrainte supplémentaire que les `Joueurs` de l'`Equipe` doivent appartenir au `Club`. Une particularité supplémentaire est que le nombre minimum

¹ La cote ELO est une note attribuée à un joueur d'échec après qu'il ait joué plusieurs parties officielles et permettant ainsi d'estimer la force du joueur.

de `Joueurs` dans l'`Equipe` dépend de la `Division` dans laquelle elle se trouve. Par conséquent, les méthodes `minimumJoueurGaranti` et `joueurs` (renvoyant un itérateur) lanceront une `RenseignementInsuffisantException` si la `Division` n'est pas connue quand on les appelle.

On vous demande de compléter les classes `Division`, `Equipe`, `Club` et `Joueur` qui vous sont fournies. Plus précisément, vous devez :

- Créer, dans la classe `Joueur`, un énuméré pour les niveaux (en classe interne publique). L'énuméré `Niveau` déclare les constantes `NON_CLASSE`, `DEBUTANT`, `JOUEUR_CONFIRME`, `MAITRE_FIDE`, `MAITRE_INTERNATIONAL`, `GRAND_MAITRE_INTERNATIONAL`. Elle contient de plus deux champs : `coteMinimale` correspondant à la cote minimale à avoir pour être à ce niveau et `abreviation` correspondant à ce qui est affiché à côté du nom du joueur dans le `toString`. Ces deux champs sont initialisés dans le constructeur. Les cotes minimales sont dans l'ordre : 0, 1000, 1600, 2300, 2400, 2500 et les abréviations sont dans l'ordre "NC", "D", "JC", "FM", "IM", "IGM". On y redéfinit la méthode `toString` afin qu'elle renvoie l'abréviation.
- Ajouter, dans chaque classe, les champs liés aux associations.
- Compléter les constructeurs et les méthodes business qui se trouvent dans les classes fournies ainsi que la méthode `setElo` de la classe `Joueur`.
- Ajouter les méthodes de gestion des associations. Quand il y a moyen de simplifier une association, vous devez le faire !