HAUTE ÉCOLE DE
NAMUR-LIÈGE-LUXEMBOURG
Catégorie technique
Année académique : 2020-2021

TI215 : programmation orientée objet 1
Exercices : Série 2

2.1. Objectif: Etablir des liens entre classes et manipuler des réseaux d'objets

Reprenez l'énoncé de la série 1. Retravaillez-le et étendez-le pour qu'il soit encore plus orienté objet. Scindez le concept de rencontre en plusieurs classes, à savoir, les classes *Coach, Club, Equipe, Arbitre* et *Rencontre*. Créez ces classes dans le même package que Rencontre et prévoyez pour chacune de ces classes *au moins un* ***constructeur***.

N.B. Dès que vous avez défini une classe, créez au moins un objet de cette classe dans la classe Principal et testez-le en :
- appelant des méthodes sur cet objet ;
- affichant la valeur de certaines de ses variables d'instances ;
- modifiant la valeur de certaines de ses variables d'instance.

a. Créez la classe intitulée **Coach.** Un coach est le responsable d'une équipe de joueurs (entrainement, inscription aux tournois, …). La classe **Coach** contient les ***variables d'instances*** suivantes :
de type String :
- nom
- prenom

de type int :
- anneeDebut (année durant laquelle a débuté le coach).

Ajoutez la ***méthode*** appelée *nbAnneesExperience* qui retourne le nombre d'années d'expérience du coach.
N.B. Pour récupérer l'année correspondant à la date système : (import java.util.)*

```
GregorianCalendar now = new GregorianCalendar( ) ;
int  anneeEnCours = now.get(GregorianCalendar.YEAR) ;
```

b. Créez la classe intitulée **Club** qui contient les ***variables d'instances*** suivantes :

de type String :

- nom
- adrClub (adresse du club)
- secretaire (nom et prénom du secrétaire du club)
- noTel (téléphone du secrétaire du club)

c. Créez la classe intitulée **Equipe** qui contient les ***variables d'instances*** suivantes :

de type String :

- nom
- categorie (catégorie de l'équipe: débutant, confirmé, expert, …)

*de type **Club** (cf. classe Club) :*

- club (club auquel appartient l'équipe)

*de type **Coach** (cf. classe Coach) :*

- coach

d. Créez la classe intitulée **Arbitre** qui contient les ***variables d'instances*** suivantes :

de type String :

- nom
- prenom
- code (chaque arbitre se voit attribuer un code de 3 lettres)

de type int :

- anneeNaissance (en 4 chiffres)

Ajoutez la ***méthode*** appelée ***matricule*** qui retourne le matricule de l'arbitre formé par la concaténation de l'année de naissance et du code de l'arbitre (ex: si l'année de naissance de l'arbitre est **1973** et son code est **abc**, son matricule est **1973abc**).

e. Créez la classe intitulée **Rencontre** qui contient les ***variables d'instances*** suivantes :

*de type **Equipe** (cf. classe Equipe) :*

- locaux (équipe locale)
- visiteurs (équipe des visiteurs)

de type int :

- pointsLocaux (points de l'équipe locale)
- pointsVisiteurs (points de l'équipe des visiteurs)
- fautesLocaux (nombre de fautes de l'équipe locale)
- fautesVisiteurs (nombre de fautes de l'équipe des visiteurs)

de type ***Arbitre*** *(cf. classe Arbitre) :*
- juge (arbitre de la rencontre)

Reprenez (cf. série 1) les **méthodes** *exAequo*, *ajouterPointsLocaux*, *ajouterPointsVisiteurs*, *ajouterFauteLocaux*, *ajouterFauteVisiteurs* et *ajouterPoints*. Vérifiez si elles doivent être adaptées ou non.
Attention, arrangez-vous pour que les méthodes *vainqueur* et *equipeFairPlay* retournent bien respectivement le **nom** de l'équipe victorieuse ou la plus fair play (ou « ex æquo ») et ***non pas la description*** *complète de l'équipe.*

2.2. Prévoyez la méthode toString( ) afin de créer la chaîne de caractères « décrivant » chacune de ces classes.

Attention : La « description » d'une équipe **doit contenir** la « description » du coach et du club correspondants. De même, la « description » d'une rencontre doit contenir la « description » de chacune des deux équipes ainsi que la « description » de l'arbitre.

2.3 Testez dans la classe Principal les appels implicites aux méthodes toString( ) des différentes classes.

Soit la variable ***renc1 de type Rencontre***.
Après avoir testé l'instruction : System.out.println(renc1), essayez d'affichez *au minimum* ce qui suit :
Attention, toute instruction doit débuter par :
System.out.println(renc1. …) ;

- le nombre de points de l'équipe locale ;
- l'équipe la plus fair-play ;
- le prénom de l'arbitre;
- la catégorie de l'équipe locale ;
- le nom des deux équipes ;
- le matricule de l'arbitre ;
- les nom et prénom du coach de l'équipe locale ;
- l'adresse du club de l'équipe des visiteurs ;
- les coordonnées du secrétaire du club de l'équipe des visiteurs ;
- le nombre d'années d'expérience du coach de l'équipe des visiteurs.

A partir de l'objet *renc1*, modifiez les informations suivantes :

- Il y a erreur d'encodage des points de l'équipe des visiteurs ; après réclamation de ces derniers, leurs points sont modifiés : ils ont désormais 56 points.
- L'arbitre tombe malade, il faut le remplacer par un nouvel arbitre.

Attention, pour ces deux derniers points, vous ne devez pas créer un nouvel objet de type Rencontre, mais bien partir de l'objet ***renc1*** *existant.*

2.4. Amélioration

Essayez d'étendre encore le réseau des objets en créant une classe **Secrétaire** qui reprendrait les coordonnées (nom, prénom et téléphone) des personnes secrétaires d'un club. Puis modifiez le *type* de la variable d'instance appelée ***secrétaire*** de la classe ***Club*** comme ci-dessous :

ancienne déclaration : **String** secrétaire, noTel
nouvelle déclaration : **Secrétaire** secrétaire

Affichez ensuite les coordonnées (nom, prénom et téléphone) du secrétaire du club de l'équipe des visiteurs d'une rencontre donnée.