



Contrôle (Module : Java)
(2heures)

Exercice 1 : (2 points) :

Compléter le tableau suivant :

<u>Vocabulaire</u>	<u>Signification</u>
Polymorphisme
Protected
Surcharge plusieurs méthodes peuvent avoir le même nom, mais des paramètres différents.
Constructeur appeler au moment de la création de l'objet.

Exercice 2 : (18 points) :

La coupe du monde de football est une compétition qui s'organise par FIFA, elle réunit un nombre d'équipes représentant les différents continents. Ces équipes sont réparties en groupes. Chaque groupe contient quatre équipes. L'objectif de ce sujet est la gestion automatique de cette compétition en utilisant une application java.

Une équipe est caractérisée par les informations suivantes : le symbole de l'équipe (**sym**), le nom de l'équipe (**nom**), l'entraîneur de l'équipe (**ent**), le groupe de l'équipe (**gr**), le nombre de points (**pts**), le nombre de buts (les buts marqués (**mrq**) et les buts reçus (**rec**)). Cette gestion est assurée à l'aide d'un ensemble de classes.

On suppose qu'on a la classe suivante :

```
public class Personne{
    private String nom, prenom, cin, fonction;
    Date naissance ;
    Personne(String n, String p, String c, Date d, String f)
    { nom=n ; prenom=p ; cin=c ; naissance=d ; }
    public String toString()
    { String ch=" Nom : "+nom+" Prénom : "+prenom+" CIN : "+cin+"
      Date de naissance : "+date+" Fonction : "+fonction ; return ch ; }
}
```

(Remarque : les attributs et les méthodes doivent être définis avec les modificateurs suivants: **private**, **public** ou **static**.)

1) Ecrire une classe **Joueur** qui hérite de la classe **Personne** contenant:

(3pts)

- Le nom du joueur (**nom** : String)
- Le prénom du joueur (**prenom** : String).
- Le numéro de la carte d'identité nationale du joueur (**cin** : String).
- La date de naissance du joueur (**naissance** : date).
- Le poids (**pds** : double).
- La position occupée dans l'équipe (**pos** : String).
- Un constructeur **Joueur** pour initialiser toutes les informations du joueur.



- La méthode **toString()** qui retourne une chaîne de caractères contenant toutes les informations du joueur.

2) Ecrire une classe **Equipe** contenant :

(4pts)

- Le groupe de l'équipe (**gr** : char).
- Le symbole de l'équipe (**sym** : String).
- Le nom de l'équipe (**nom** : String).
- L'entraîneur de l'équipe (**ent** : Personne).
- Le nombre de points (**pts** : int).
- Le nombre de buts marqués (**mrq** : int).
- Le nombre de buts reçus (**rec** : int).
- Le nombre de joueurs de l'équipe (**nbj** : int).
- Les joueurs de l'équipe (**eqp[]** : Joueur).
- Un constructeur **Equipe()** qui prend en paramètre le symbole de l'équipe.
- Une méthode **clavier()** pour saisir et retourner une chaîne de caractères.
- Une méthode **lecture()** permettant la saisie du reste des informations (**gr**, **nom**, **ent**, **pts**, **mrq**, **rec**, **nbj** est **eqp**).
- Une méthode **affichage()** pour afficher toutes les informations de l'équipe.



3) Ecrire une classe **Coupe** répondant au cahier des charges suivant :

(3pts)

- Le nombre maximum des équipes (**max** : int).
- Un tableau des équipes (**table[]** : **Equipe**). (Les équipes d'un groupe donné seront enregistrées dans le tableau de façon adjacente : les quatre premières cases seront réservées au premier groupe, les quatre suivantes seront réservées au deuxième groupe, ...)
- Le nombre des équipes enregistrées (**nbe** : int).
- Un constructeur qui prend en paramètre un entier pour initialiser le nombre maximum des équipes qui peuvent participer à la compétition.
- Une méthode **boolean ajouteEquipe(Equipe eq)** qui ajoute une équipe au tableau « **table** », la méthode renvoie **true** si l'ajout s'est correctement effectué, **false** sinon.
- Une méthode **afficher(Equipe[] tab, int d)** qui permettra d'afficher l'ensemble des équipes enregistrées dans un tableau de type **Equipe**.

4) Réaliser une application JAVA pour tester la classe **Coupe**.

(2pts)

5) Enrichir la classe **Coupe** d'une méthode **void triGrp(int idDeb, int idFin)** qui permet de Trier les équipes d'un groupe délimité par les deux indices (**idDeb** et **idFin**) par ordre décroissant selon le nombre de points. Dans le cas d'égalité (Même nombre de points), on compare la différence entre les buts marqués et les buts reçus (Nombre de buts marqués - Nombre de buts reçus). Aussi dans le cas d'égalité (Même différence de buts).

(2pts)

6) Ajouter à la classe **Coupe** une méthode **void premTour()** qui Affiche la liste des équipes de chaque groupe par ordre décroissant (Résultats du premier tour de la compétition).

(2pts)

7) Ajouter à la classe **Coupe** une méthode **void deuxTour()** qui supprime les équipes non qualifiées au deuxième tour (les deux dernières équipes de chaque groupe). Après la suppression, il faut afficher les équipes supprimées.

(2pts)