

TP 3 – 1^{ère} Partie - Héritage

Exercice 1/

Question 1/ Créez une classe `Sport` avec `code` et `libelle` comme variables d'instance. Définissez un constructeur dont les valeurs initiales sont passées en paramètres. Écrivez les accesseurs, une méthode `toString()` et une méthode `affiche()` pour l'affichage, utilisant `toString()`.

Question 2/ Un sport collectif est caractérisé en plus par un nombre de joueurs. Créez une classe `SportCo` qui hérite de la classe `Sport` et qui possède une variable d'instance `nbJoueurs`. Gérez l'affichage d'une instance de `SportCo`.

Question 3/ Créez une classe `LesSports` ayant en variable d'instance une `ArrayList` de `Sport` et des méthodes d'instance permettant :

- d'ajouter un sport dans cette `ArrayList`,
- d'afficher les sports enregistrés dans cette `ArrayList`.

Question 4/ Dans une classe `ProgSports`, écrivez un `main()` qui crée une instance de `LesSports` comportant MAX sports, ajoute 5 sports, dont 3 sports collectifs, et les affiche.

Exercice 2/

On souhaite informatiser la gestion de courses de chevaux.

- Un cheval est caractérisé par un numéro, un nom, un sexe et une race.
- Un cheval de course possède en plus un entraîneur, un jockey et un montant des gains.
- L'entraîneur est une personne qui dispose d'un numéro de licence.
- Un jockey est une personne dont on conserve le poids et le salaire.
- Une personne est décrite par son nom, son prénom et son adresse.
- Une course est caractérisée par un nom, une dotation (en €), un numéro et une liste de partants.

Question 1/ Définir une classe `Personne` ayant comme variables d'instance le nom, le prénom et l'adresse de la personne. Prévoir le constructeur, dont les valeurs initiales sont passées en paramètres, les accesseurs et la méthode `toString()`.

Définir les classes `Entraîneur` et `Jockey`.

Écrire un `main()` testant ces classes.

Question 2/ Définir une classe `Cheval` ayant comme variables d'instance un numéro d'inscription au service d'identification des équidés, qui devra être géré automatiquement (à l'aide d'une variable de classe), un nom, un sexe et une race.

Prévoir le constructeur, les accesseurs et la méthode `toString()`.

Définir également la classe `ChevalDeCourse`.

Prévoir le constructeur permettant de créer une instance de `ChevalDeCourse` possédant un numéro, un nom, un sexe, une race, un montant des gains et un entraîneur. Le jockey n'est pas connu lors de la création.

Définir les méthodes :

- `affiche()` qui visualise les informations caractérisant un cheval de course,
- `attribueJockey(Jockey j)` qui attribue un jockey au cheval de course.

Tester ces classes.

Question 3/ Définir une classe `Course` ayant en variables d'instance le nom de l'épreuve, une dotation et une `ArrayList` de `ChevalDeCourse`.

Prévoir le constructeur qui crée initialement une course sans chevaux ainsi que les méthodes :

- `affiche()` qui visualise les renseignements sur les chevaux de la course (avec son nom et sa dotation),
- `chevalPresent(String nom)` qui retourne **vrai** si le cheval identifié par son nom est présent dans la course, **faux** sinon (utilisation de la méthode `equals` de la classe `String`),

Le programme de test de cette méthode doit afficher la valeur attendue et la valeur obtenue, pour chacun des 4 cas suivants : un cheval absent, un cheval situé en milieu ou aux 2 extrémités de l'`ArrayList`.

- `enregistre(ChevalDeCourse c)` qui enregistre un cheval de course dans la course (on utilisera la méthode précédente pour empêcher l'enregistrement d'homonymes).

Tester cette classe.

Question 4/ Dans la classe `Course`, ajouter une méthode `rechercheCheval(String nom)` qui affiche les renseignements sur le cheval identifié par son nom, s'il est présent, ou un message « *absent* » sinon.

Tester cette méthode : Afficher la valeur attendue et la valeur obtenue, pour chacun des 4 cas suivants : un cheval absent, un cheval situé en milieu ou aux 2 extrémités de l'`ArrayList`.

Question 5/ Dans la classe `Course`, ajouter une méthode `meilleureCote()` qui retourne le cheval ayant remporté le plus de gains parmi tous les participants d'une course.

Tester cette méthode.