|  |
| --- |
| Devoir Surveillé poo 2017-2018 |
| Matière: Programmation orientée Objet Calculatrice Scientifique: **Non autorisée** Documents: **Non autorisés** Enseignants: Mhafdhi H., Hedhli J,Sghaier I, Daghfous.N, Chelbi.S. Durée: 1h00'Classes :DSI 21,22,23, RSI21, SEM21 Date: 09/11/2017 |

**Exercice 2 : 16 *Pts***

On se propose de créer une application qui permet de gérer des équipes de football. Pour se faire on a besoin des classes suivantes : Acteur, Joueur et équipe : écrire le code nécessaire pour créer les classes suivantes

**Classe Acteur**

Définir une classe **Acteur**, possédant des attributs **nom** et **prenom** de type **String , age** de type **int** et **Montant(** qui représente le salaire initiale d’un acteur**)** de type **double.** Ces attributs ne sont visibles qu’au sein de la classe.

1. Déclarer une méthode **getSalaire** sans paramètre retournant une valeur numérique de type **double.**
2. Ecrire un **constructeur** à **4** paramètres permettant d'initialiser les attributs de la classe **Acteur.**
3. Ecrire la méthode **toString** permettant de retourner toute l'information portée par les 4 attributs et le salaire de l'acteur.
4. Ecrire le méthode **getAge()** et **getMontant()** qui retourne respectivement la valeur de l’âge et le Montant.

**Classe Joueur**

Sachant qu'un joueur est un acteur particulier, définir, une classe **Joueur** possédant un attribut **club** de type **String.**

* Ecrire un **constructeur** à **5** paramètres permettant d'initialiser une instance de la classe **Joueur**.
* Ecrire un **constructeur** à **4** paramètres permettant d'initialiser une instance de la classe **Joueur** dont le club est le « *Club Africain »*. Ce constructeur devra utiliser celui à **5** paramètres.
* Ecrire l'accesseur en lecture pour l'attribut club.

Le salaire d’un joueur est calculé suivant l’âge du joueur : Si l’âge du joueur est inférieur ou égale à **20** son salaire est égal à **150%** de l’attribut **Montant,** si l’âgeest compris entre **20 et 30** son salaire est égal à **100%** de l’attribut **Montant** et si l’âge est compris entre **30 et 40** son salaire est égal à **75%** de l’attribut **Montant**

* Redéfinir la **méthode getSalaire()**
* Redéfinir la méthode **toString** permettant de retourner toute l'information portée par tous les attributs et le salaire d'un joueur.

**Classe Equipe**

Dans le cadre de gestion des équipes de football on se propose de créer une classe **Equipe**. Cette classe contient les attributs suivants

* Une constante de classe de nom **NOMBREJOUEURS** de type **int** et de valeur **11.**

Cette constante est de visibilité publique.

* Un attribut **club** de type **String** représentant le nom du club de l’équipe. Cet attribut n’est visible que dans la classe Equipe.
* Un **tableau Joueurs** comme attribut afin de ranger uniquement les joueurs de l’équipe.

Dans la classe joueur, on va ajouter un attribut de type **Equipe** ainsi que son **accesseur en lecture** (Revenir dans le code de la classe **Joueur**).

Ecrie la classe **Equipe** qui définit les méthodes suivantes :

4) Un constructeur à 1 paramètre de type String effectuant les traitements nécessaires sur les 2 attributs.

5) la méthode **toString** permettant l'affichage du club et de tous les joueurs de l'équipe

6) Une méthode ***AjoutJoueur()*** qui permet d'ajouter un joueur dans le tableau.

Cette méthode devra renseigner l'attribut **equipe** du joueur.

Elle affiche un message d’erreur **si** la composition de l'équipe dépasse les 11 joueurs autorisés sur un terrain de football. Le message affiché sera '*'Le nombre de joueurs dans un équipe de football ne doit pas dépasser les 11 '*'.

Comme elle affiche un message d’erreur **si** le joueur n'est pas dans le même club que celui de l'équipe. Le message affiché sera « ce joueur ne fais pas partie de notre club ! »

**.. /..**