

TD N°1 : Classe et Objet

Exercice 1 : Classe Personne.

1. Constructeur exhaustif

Créer une classe, nommée Personne, non publique (sans le mot clé public devant class), contenant 2 champs :

- un champ **nom** de type **String**,
 - un champ **age** de type primitif **int**.
- a. Créer un constructeur exhaustif (constructeur paramétré initialisant tous les champs), de signature **Personne(String leNom, int lAge)** initialisant les deux champs de la classe.
 - b. Ajouter les sélecteurs et modificateurs pour tous les attributs.
 - c. Créer la méthode **afficher()** de signature **String afficher()** qui affiche le nom et l'âge au format décrit ci-après. Pour un objet de type Personne dont la variable d'instance nom vaut "Coupdetrix" et la variable d'instance age vaut 25, la méthode afficher() retournera : **Mon nom est : Coupdetrix et mon Age est : 25 ans**
 - d. Créer la méthode main() de signature **public static void main(String[] args)**. Au sein de cette méthode, créer, à l'aide de l'opérateur new une instance d'objet de type Personne et appeler sa méthode afficher().

2. Champ de type tableau

- a. Ajouter un champ à la classe Personne, de type tableau de réels **double[] comptes** ; susceptibles de représenter les divers soldes des comptes en banque de la personne. Adapter le constructeur exhaustif ainsi que la méthode afficher() (qui doit afficher, outre le nom et l'âge de la personne les différents éléments du tableau comptes).
- b. Créer, au sein de la classe Personne, la méthode **void diviserParDeux()**, qui modifie le tableau comptes en divisant chaque élément par deux.
- c. Dans la méthode main, appeler diviserParDeux() sur le premier des objets de type Personne et la méthode afficher() de ces deux objet.

3. Constructeurs par défaut et de copie

- a. Au sein de la classe Personne, créer un constructeur par défaut, de signature **Personne()**.
- b. Créer un constructeur de copie, de signature **Personne(Personne p)** qui initialise les champs de la classe à ceux de p.

- c. Modifier la méthode `main()` de `PersonneMain` en créant 3 personnes, l'une à l'aide du constructeur exhaustif, une autre avec le constructeur par défaut et la dernière avec le constructeur de copie. Les afficher toutes les trois en appelant la méthode `afficher()`.

Exercice 2 : Jeu de combat

Soit un jeu de combat dans lequel peut s'affronter deux personnages. Chaque personnage possède un nom et une force de frappe. En plus de ces informations, on souhaite gérer son niveau d'expérience et ses dégâts au cours du jeu. Comme comportement, on distingue leur capacité à frapper un autre personnage et par de ce fait gagner en expérience.

Comme règles de gestion on distingue :

- Chaque personnage a par défaut un niveau de dégât à 0 et un niveau d'expérience à 1.
- Quand un personnage frappe un autre, il lui inflige des dégâts dont la valeur est égale au 10 % de sa force.
- Quand un personnage assomme un autre, c'est-à-dire lui inflige des dégâts équivalents à 100 au total, son niveau d'expérience augmente de la valeur du niveau d'expérience du personnage assommé.

TAF :

1. A l'intérieur d'une classe `Personnage`, écrivez les constructeurs permettant de créer respectivement des personnages avec les valeurs par défaut et des personnages avec une force et un nom bien défini.
2. Vous ajouterez les sélecteurs et modificateurs pour chaque attribut identifié.
3. En plus de ces méthodes, vous écrierez les méthodes `frapper()` et `gagnerExperience()` conformément aux règles de gestion.
4. Dans la méthode `main`, créer deux personnages et simuler les méthodes écrites.

Exercice 3 : Compte bancaire

1. Créer une classe, nommée `Compte`, non publique (sans le mot clé `public` devant `class`), contenant le champ : **solde** de type primitif **double**.

- a. Créer un constructeur par défaut, un constructeur paramétré et un constructeur de copie.

- b. Créer la méthode **afficher()** qui affiche la situation du compte au format décrit ci-après :
 - Pour un objet de type Compte dont la variable d'instance solde vaut **25 000**, la méthode **afficher()** affichera à l'écran : **Le solde est créditeur de : 25 000 FCFA**.
 - Pour un objet de type Compte dont la variable d'instance solde vaut **-10 000**, la méthode **afficher()** affichera à l'écran : **Le solde est débiteur de : 10 000 FCFA**
- c. Créer les méthodes **deposer()** et **retirer()** qui permettent respectivement de créditer et débiter un compte.
- d. Créer la méthode **virerVers()** qui permet de retirer un montant d'un compte afin de créditer un autre compte.

2. Créez un compte vide et un autre compte initialisé à **15 000** que vous affecterez à deux variables. Ecrivez le code correspondant aux opérations suivantes :

- a. Dépôt de **5 000** sur le premier compte.
- b. Dépôt de **15 000** sur le deuxième compte.
- c. Retrait de **20 000** sur le deuxième compte.
- d. Virement de **25 000** du deuxième compte vers le premier.
- e. Affichage des soldes des deux comptes.

3.

- a. Créez un tableau de dix comptes. Pour cela, notez bien qu'il faut d'abord créer le tableau puis créer successivement les dix comptes à mettre dans les dix cases de ce tableau.
- b. Dans chaque case, faites un dépôt de 15 000 plus une somme égale à 1000 fois l'indice du compte dans le tableau.
- c. Ensuite, vous ferez un virement de 1700 de chaque compte vers chacun des comptes qui le suivent dans le tableau (par exemple, du compte d'indice 5, il faut faire des virements vers les comptes d'indice 6, 7, 8 et 9).