Université de Mila
Institut des sciences et technologie
2ème année Licence Informatique
Matière : Programmation Orientée Objet
Responsable de la matière : Dr. Sadek BENHAMMADA

Série de TD N°3

Exercice 1 (*Packages+ méthodes statiques*)

Identifier les erreurs et proposer des corrections dans le code suivant :

```
package premier;
class A {
    private int q ;
    public static void f (int n)   { q = n ; }
}
```

```
package deuxieme;
public class TestA {
    public static void main (String args[])    {
        A a = new A() ;  int x = 5 ;
        a.f(x) ;
    }
}
```

Exercice 2 (*Surcharge+Passage de paramètres+ méthodes statiques*)

Soit le code suivant :

```
class Param {
    private int a ;
    public Param (int a) { this.a=a;}
    public static void incrementer (int n) { n=n+1;}
    public static void incrementer (int n) { n=n+2;}
    public static void incrementer (Param pm) { pm.a=pm.a+1;}
 }
```

1. Identifier les erreurs syntaxiques et proposer des corrections dans le code précédent.
2. Déclarer une classe `TestParam`, ne comportant pas d'attributs et comportant une unique méthode dans laquelle :
   - Déclarer un entier `x` et initialiser le avec la valeur 9, puis, créer un objet `p` de la classe `Param`, et initialiser son attribut a avec x.
   - Appeler les méthodes `incrementer()` de la classe `Param` avec comme paramètres respectivement la variable `x`, puis, l'objet `p`.
   - Afficher la valeur de la variable `x` et celle de l'attribut a de l'objet `p`
3. Qu'affichera l'exécution de la classe `TestParam` ? Justifier votre réponse.

Exercice 3 (*Attributs statiques +Surcharge+Tableaux*)

1. Déclarer une classe `Personne` contenant les attributs :
   - `numero` de type `int` affecté de façon auto-incrémentale à chaque création d'un nouvel objet de la classe `Personne` (Utilisation d'un attribut statique) ;
   - `nom` de type `String ;`
   - `prenom` de type `String ;`
   - `age` de type `byte ;`
   - `adresse` de type `String.`

Série de TD N°3

2. Ajouter, à la classe `Personne` un constructeur qui initialise tous les attributs.
3. Ajouter, à la classe `Personne`, une méthode `toString()`qui retourne une chaine de caractère similaire à la phrase :
   « `Numero:0, Prenom:Ali, Nom:Ahmed, Age:20 ans, Adresse:Mila`».

4. Ajouter une méthode de signature `String toString (boolean compact)` qui affiche selon l'un des formats suivants :
   - Si `compact` est égal à `false`, l'affichage est le même que celui de la méthode `toString()`.
   - Si `compact` est égal à `true` (affichage compacté), sera de la forme :
     « `1, Ahmed Ali, 20, Mila` ».
5. Créer une classe `TestPersonne` ne comportant pas d'attributs et comportant une unique méthode `public static void main (String[] args)`. Au sein de cette méthode :

   - Déclarer un tableau d'objets Personne `TabP` de 100 éléments.
   - Ecrire le code nécessaire pour remplir le tableau avec des objets de classe Personne. Le programme doit donner la possibilité à l'utilisateur de saisir les valeurs des attributs par le clavier.
   - Ecrire le code qui permettra à l'utilisateur d'afficher les objets du tableau `tabP` avec la méthode `toString(boolean compact).` Le programme doit permettre à l'utilisateur de choisir entre un affichage normal ou compacté.
6. Supposons que lors de l'exécution du programme, nous avons rempli le tableau avec des objets suivants :

Nom	**Prénom**	**Age**	**Adresse**
Ahmed	Ali	20	Mila
Benslimane	Aicha	19	Constantin
BenBrahim	Idir	22	Setif

Qu'affichera le programme ? Justifier votre réponse.

Annexe : Déclaration d'un tableau en Java

La syntaxe de déclaration d'un tableau à une dimension est la suivante :

```
<type du tableau> <nom du tableau>[] =new <type du tableau> [n]
```

Exemple:

```
int tab [] = new int[10]; // déclaration d'un tableau de 10 entiers
double[] tab1 = new double [20]; // déclaration d'un tableau de 20 doubles.
```