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

Corrigé type du TD N°3

Exercice 1

1. La classe A n'est pas déclaré public, donc, elle n'est pas accessible en dehors du package premier, ainsi, la classe `TestA` ne peut pas y accéder. La solution consiste à déclarer la A comme classe publique

```
public class A{…}
```

2. La méthode statique `f` de A ne peut pas agir sur un attribut non statique ; l'affectation `q=n` est incorrecte. Une solution possible consiste à passer l'objet dont on veut modifier l'attribut `q` en paramètres :

```
public static void f (int n, a A)   { a.q = n ; }
```

3. La classe `TestA` se réfère à la classe `A` qui appartient à un autre package. Il est nécessaire donc d'importer la classe A dans la classe `TestA` :

```
package deuxieme;
import premier.A;
public class TestA {…}
```

4. L'appel `a.f(x)` se réfère à un objet, ce qui est toléré. Cependant, il est préférable que la méthode statique soit appelées par sa classe :

```
A.f(x,a)
```

Exercice 2

1. Les 2 méthodes incrémenter (int n) ont la même signature (même nom et mess paramètres), ceci provoque une erreur, car les méthodes peuvent avoir le même nom mais ils doivent avoir des paramètres différents (la surcharge). Une correction possible consiste à changer le nom de la deuxième méthode :

   ```
   public static void incrementer2 (int n) { n=n+2;} //2
   ```

   La methode `incrementer (Param pm)`est correcte, car sa signature est different de celle de la méthode `incrementer (int n).`

```
public class TestParam
{
    public static void main(String arg[])
    {
        int x=9;
        Param p=new Param(x);
        Param.incrementer(x);
        Param.incrementer(p);
        System.out.println( "La variable x="+x);
        System.out.println( "L'attribut a="+x);
    }
}
```

2. L'exécution de classe `TestParam` affichera:

```
La variable x=9
L'attribut a=10
```

Explication :

Le passage de paramètres en java se fait toujours par valeur, ainsi, la valeur de la variable x ne sera pas modifiée par l'instruction incrementer(x), car la valeur de x sera copié dans le paramètre formelle, et la modification sera effectué sur cette copie et non sur x.

Par contre, Quand on passe un objet en paramètres, c'est la référence sur cet objet qui est passée et

Corrigé type du TD N°3

copiée comme paramètre formel. Donc, cette copie référence le même objet passé en paramètres, ainsi, la modification des attributs dans la méthode modifiera effectivement leurs valeurs. Par conséquent, l'instruction incrementer (p) ; incrémente effectivement la valeur de l'attribut a de l'objet p.

Exercice 3

```
//1,2,3,4
public class Personne
{
    static int nombre=0;
    private int numero;
    private String prenom;
    private String nom;
    private byte age;
    private String adresse;
    void setNom(String n){this.nom=n;}
   public Personne(String nom, String prenom, byte age, String adresse)
    {
       this.numero=nombre;
       nombre++;
       this.nom=nom;
       this.prenom=prenom;
       this.age=age;
       this.adresse=adresse;
    }

   public String toString()
    {
        return "Numero:"+ this.numero+", Prenom:"+this.prenom+",
Nom:"+this.nom+", Age:"+this.age+" ans, Adresse:" +this.adresse;
    }

   public String toString(boolean compact)
    {
        if (compact)
            return this.numero+", "+this.prenom+", "+this.nom +",
"+this.adresse;
        else
            return this.toString();

    }
 }
```

Corrigé type du TD N°3

5.

```
      import java.util.Scanner;
      public class TestPersonne
      {
          public static void main(String arg[])
          {
              Scanner sc = new Scanner(System.in);
              String reponse;
              boolean compact;
              Personne tabP[]=new Personne[100];
              do
              {
              System.out.print("Nom:");
              String nom = sc.next();


              System.out.print("Prenom:");
              String prenom = sc.next();

              System.out.print("Age:");
              byte age = sc.nextByte();

              System.out.print("Adresse:");
              String adresse = sc.next();

              tabP[Personne.nombre]=new Personne(nom, prenom,age,adresse);

              System.out.println("Un objet de classe Personne a ete cree
avec succes.");

              System.out.println("Voulez vous céer un autre objet?
(O/N):");
              reponse = sc.next();
              } while (reponse.equals("O"));

              System.out.println("Voulez vous un affichage compacté (C) ou
détaillé (N)");
              reponse = sc.next();
              compact=reponse.equals("C");

              for (int i=0;i<Personne.nombre;i++)
              {
                  System.out.println(tabP[i].toString(compact));
              }

          }
```

Corrigé type du TD N°3

```
    }
```

6. Le programme affichera :

```
0, Ahmed   Ali   20,   Mila
1, Benslimane   Aicha,     19,   Constantin
2, BenBrahim    Idir, 22,   Setif
```

Explication :

L'attribut nbr de classe Personne est un attribut statique, i.e, il n'y a qu'une seule copie de nbr pour tous les objets de la classe Personne. L'affectation de nbr à numero et l'incrémentation de sa valeur dans le constructeur permettra de générer des valeurs auto-incrémentales pour l'attribut numéro, ainsi, les objets de la classe Personne auront des valeurs différentes pour l'attribut numero.