



## TD N° 3

### Exercice 1

Trouvez et corrigez les sept (07) erreurs pour que ce programme fonctionne.

<pre>package a; class A {     private int x;     public A(int px){         px=x ;     } }</pre>	<pre>package a; class B extends A {     private int y;     public B(int px, int py){         x=px ;         y=py;     }     public int somme (){         return x+y;     } }</pre>	<pre>package b; class Test{     public static void main (String         [] args){         B b=new B(8,7);         System.out.print(somme());     } }</pre>
---	--	--

### Exercice 2

Examinez le code Java suivant (constitué de 2 fichiers) :

```
/* fichier Schtroumpfable.java */
public interface Schtroumpfable {
    void Schtroumpfer() {
        System.out.println("Je suis schtroumpfé");
    }
}
```

```
public abstract class Truc implements Schtroumpfable {
    private String nom;
    public Truc(String unNom){
        nom = unNom;
    }
}
```

- 1) L'interface **Schtroumpfable** est-elle correcte ? Si non expliquez pourquoi et réécrivez-la correctement.
- 2) Dans le code du fichier **Truc.java**, peut-on enlever le mot-clé **abstract** ? Donnez une réponse détaillée.
- 3) Ecrivez le code Java minimal d'une classe nommée **Machin concrète** (pas abstraite) et qui hérite de la classe **Truc**.
- 4) On ajoute à notre programme Java la classe **Chose** suivante :



```
1. /* fichier Chose.java */
2. public class Chose extends Machin {
3.     public Chose(String nom) {
4.         super(nom);
5.     }
6.     public void Schtroumpfer(){
7.         System.out.println(nom + " est une Chose Schtroumpfée");
8.     }
9. }
```

5) Le compilateur signale une **erreur** à la **ligne 7**. Laquelle ? Expliquez comment l'éviter.

6) Soit le programme principal :

```
1. public class Pp {
2.     public static void main(String[] args){
3.         Schtroumpfable[] tab = new Schtroumpfable[6];
4.         tab[0] = new Chose("c");
5.         tab[1] = new Machin("m");
6.         tab[2] = new Truc("t");
7.         tab[3] = new Schtroumpfable("s");
8.         tab[4] = (Machin) tab[0];
9.         tab[5] = (Chose) tab[1];
10.        for (Schtroumpfable x : tab)
11.            if (x != null)
12.                x.Schtroumpfer();
13.    }
14. }
```

a) Corrigez ce programme

b) Donnez le résultat affiché après correction.



## Solution

### Exercice 1

<pre>package a; class A {     private int x;     public A(int px){         x=px; // au lieu de px=x     }     int getX(){         return x;     } }</pre>	<pre>package a; public class B extends A {     private int y;     public B(int px, int py){         super(px);         y=py;     }     public int somme (){         return getX()+y;     } }</pre>	<pre>package b; import a.B; class Test{     public static void main         (String [] args){         B b=new B(8,7);         System.out.print(b.somme());     } }</pre>
---	--	--

1) Non. Une interface ne peut pas avoir de méthode concrète. La méthode Schtroumpfer() doit donc être déclarée sans corps :

```
public interface Schtroumpfable3 {
    void Schtroumpfer(); }
```

2) Non. Le mot-clé abstract ne peut pas être enlevé car pour que la classe Truc soit concrète et implémente l'interface Schtroumpfable, il faut qu'elle contienne la définition de la méthode Schtroumpfer().

3) La classe Machin doit définir la méthode Schtroumpfer() pour pouvoir être concrète :

```
public class Machin extends Truc {
    public Machin(String nom){ super(nom); }

    public void Schtroumpfer(){
        System.out.println("Je suis schtroumpfé");
    }
}
```

4) La ligne 7 contient une référence au champ nom, qui bien qu'il existe dans Chose car il a été hérité de Truc (par l'intermédiaire de Machin), ne peut pas être accédé car déclaré comme privé (private). Le compilateur donnera le message « error: nom has private access in Truc ».

- Une solution simple est de rendre l'accès à nom protégé (protected) au lieu de privé.
- On peut aussi laisser l'accès à nom privé et ajouter un accesseur protégé (protected String getNom()) que l'on utilisera dans Chose.

5) Classe Pp

a) Instructions incorrectes :

Ligne 6 : tente d'instancier une classe abstraite

Ligne 7 : tente d'instancier une interface.

Ligne 9 : tente de transtyper un objet du type dynamique Machin en Chose. Ne provoque pas d'erreur à la compilation car le type statique (déclaré) de l'objet est bien un supertype (Schtroumpfable) de Chose. L'erreur apparaîtra à l'exécution : « ClassCastException:

b) Resultat de l'exécution :

```
c est une Chose Schtroumpfée
Je suis schtroumpfé
c est une Chose Schtroumpfée
```