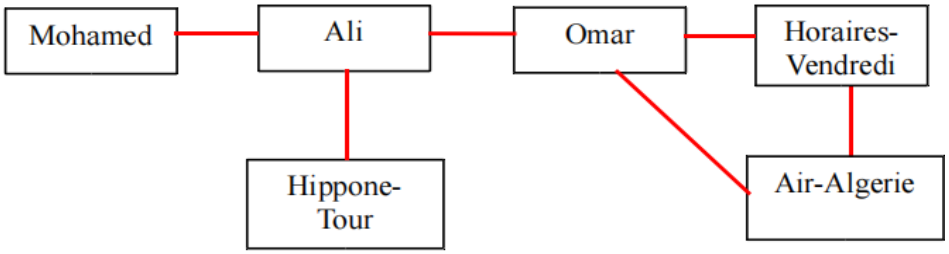
Université Badji Mokhtar Annaba
Département d'informatique
2ème Année LMD
Module POO
Année 2023/2024

Corrigé type Série TD1: Notions de base POO

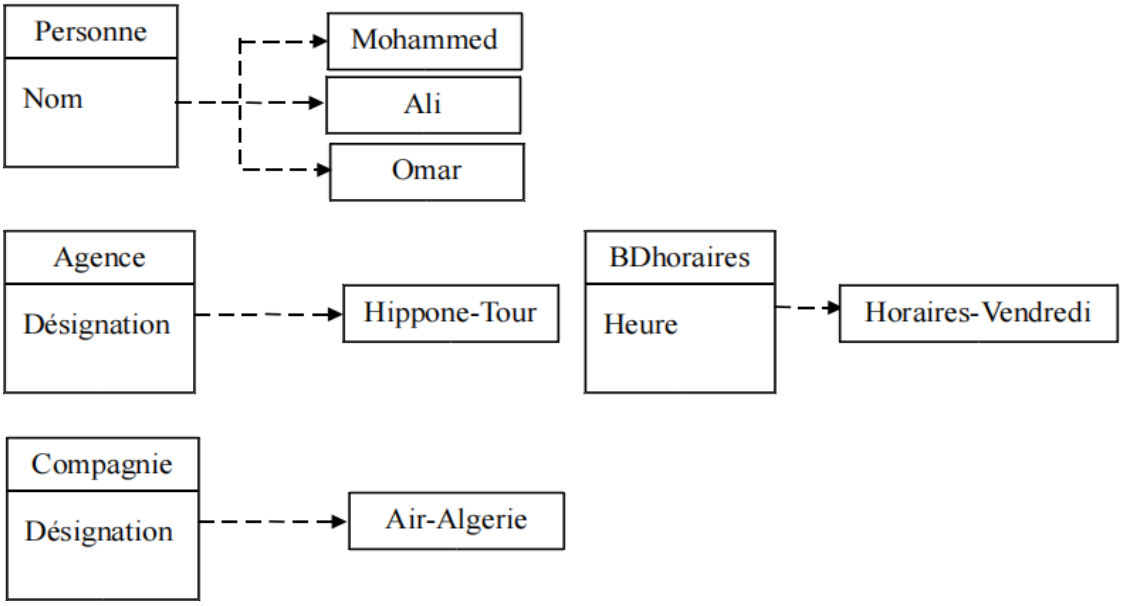
Exercice 01:

a- Les Objets sont : Mohamed, Ali, Omar, Hippone-Tour, Air-Algérie, Horaires-Vendredi

b-



c- Classes :



Exercice 02:

`System.out.println(f.i);` → `valide`
`System.out.println(f.s);` → `valide`
`f.imethod();` → `valide`
`f.smethod();` → `valide`
`System.out.println(F.i);` → `non valide`
`System.out.println(F.s);` → `valide`
`F.imethod();` → `non valide`
`F.smethod();` → `valide`

On ne peut pas faire un accès statique à des membres non statiques.

Exercice 03:

```
i = 2
j = 2
i = 2
j = 3
```

Une variable d'instance sera réinitialisée avec chaque nouvel objet. En revanche, une variable statique(de classe) ne sera initialisée que lors de la création du premier objet de la classe.

Exercice 04:

`box1 == box2?` → **`true`** (Les deux variables pointent sur le même objet)

`box1 == box3?` → **`false`** (Les variables pointent sur deux objets différents)

Exercice 05:

Le compilateur choisit la méthode la plus adaptée en fonction du nombre et type des paramètres effectifs (la loi du moindre effort)

1. `double z = m(4, 5);` → m (2) avec casting implicite (promotion) du premier paramètre vers le type double
2. `double z = m(4, 5.4);` → m (2)
3. `double z = m(4.5, 5.4);` → m (1)

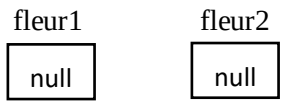
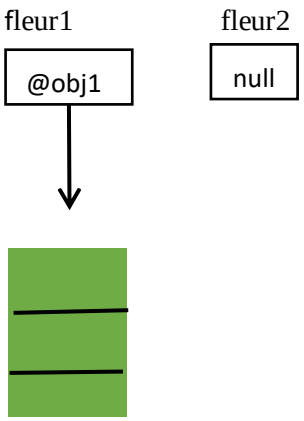
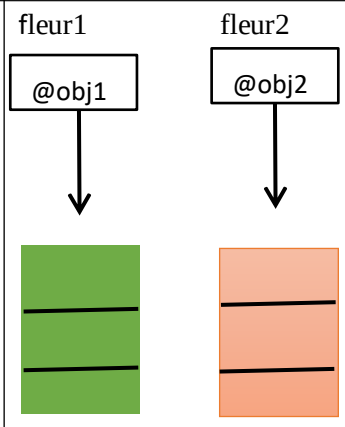
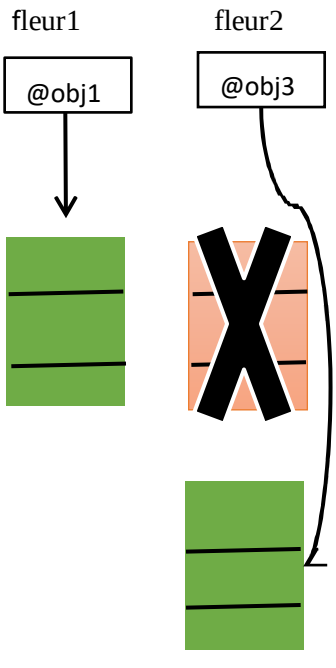
Exercice 06:

1. - Les variables de classe : pollen
   - Les variables d'instance : petales, tige, pedoncule, etamines
   - Le nombre total de variable pour 3 objet = nombre de variable d'instances + nombre de variable de classe
     = 3x4 + 1 = 13
2. `getEtamine()` est une fonction accesseur.

```
int getEtamine(){ return etamine ;}
```

3. La surcharge du constructeur est un exemple de polymorphisme; le constructeur est surchargé car il a deux formes différentes l'une paramétrée et l'autre sans paramètres.

4. Schéma de l'état de mémoire :

Fleur fleur1=null, fleur2=null;	fleur1=new Fleur();	fleur2=new Fleur();	fleur2=new Fleur();
			

Exercice 07:

(1)
```
class Point {
    char  nom;
    double abscisse;
    Point(char nom, double abscisse){
    this.nom = nom;
    this.abscisse = abscisse;
    }
    void affiche() {
    System.out.println("nom: "+nom+" abscisse: "+abscisse);
    }
    void translate(double val) {
    abscisse += val;
    }
    }
```

(2)
```
public class TestPoint {
    public static void main(String[] args) {
    Point p = new Point('x',3.2);
    p.affiche(); // affiche nom: x abscisse: 3.2
    p.translate(2.1);
    p.affiche(); // affiche nom: x abscisse: 5.3
    }
    }
```

(3)
```
public class Point {
    char  nom;
    double abscisse;
    double ordonne;
    Point(char nom, double abscisse){
    this.nom = nom;
    this.abscisse = abscisse;
    }
    Point(char nom, double abscisse, double ordonne){
    this.nom = nom;
    this.abscisse = abscisse;
    this.ordonne = ordonne;
    }
    void affiche() {
    System.out.println("nom: "+nom+" abscisse: "+abscisse + "ordonnée" + ordonnee);
    }
    void translate(double val) {
    abscisse += val;
    }
    void translate(double dx, double dy) {
    abscisse += dx;
    ordonne += dy;
    }
    }
```

Class point
+char nom +double abscisse +double ordonne
+point(char nom, double abscisse, double ordonne) +void affiche() +void translate(double dx,double dy)

Exercice 08:

1.

```
public class FeuDeSignalisation {
    int  couleur;
    int position ;
    double hauteur ;
    {
    public void change  () {   this.couleur (this.couleur % 3) +
    1;
              }
```

2.

```
public  class Voiture {
int num ;
String marque ;
char couleur ;
 int vitesse ;
 public void changevitesse(int nv) {vitesse=nv ;}
```

3.

```
public class FeuDeSignalisation {
    int  couleur;
    int position ;
    double hauteur ;
    Voiture voitureDevant;
    public void change() {
     this.couleur = (this.couleur % 3) + 1;
     if (this.couleur== 1) voitureDevant.changevitesse(50);
}
}
```

4.

```
FeuDeSignalisation(int positionInit, double hauteurInit) {
// pas de retour pourleconstructeur
position = positionInit;
hauteur = hauteurInit;
couleur = 1;
}
```

5.

```
FeuDeSignalisation NouveauFeu = new FeuDeSignalisation(1, 4);
```