

TP Classes et Objets

Exercice 0 Donner une définition concise des termes : objet, classe, constructeur.

Exercice 1 Définir une classe `Compte`. Le constructeur doit avoir un seul argument qui initialise un champ `titulaire`. La classe `compte` devrait permettre le dépôt et le retrait d'une somme d'argent passée comme argument. L'objectif final est de pouvoir faire tourner un code celui listé ci-après.

```
1 .....
2 void setup(){
3     Compte c1 = new Compte("Alice");
4     Compte c2 = new Compte("Nathalie");
5     c1.depot(20);
6     c1.retrait(10);
7     println(c2);
8 }
```

Exercice 2 Reprendre l'exercice 1 de l'ED6 en introduisant une classe `Motif` qui contient une méthode `dessiner()` spécifique. La méthode `draw` du programme principale doit créer un objet `Motif` et appeler `dessiner` sur cet objet.

Exercice 3

1.

a) Créer une classe, nommée `Personne` contenant 2 champs :

-- un champ `nom` de type `String`, et

-- un champ `age` de type primitif `int`.

Créer un constructeur exhaustif (c.à.d. initialisant tous les champs), de signature:

```
Personne(String leNom, int lAge)
```

initialisant les deux champs de la classe à `leNom` et `lAge`.

b) Créer la méthode `afficher()` de signature

```
String afficher()
```

qui renvoie le nom et l'age au format décrit ci-après. Pour un objet de type `Personne` dont la variable d'instance `nom` vaut "Anne" et la variable d'instance `age` vaut 25, la méthode `afficher()` devrait renvoyer

```
Nom : Anne, Age : 25 ans
```

c) Tester ces classes en affichant le nom d'une personne dans une fenêtre graphique

2. Champ de type tableau

a) Ajouter un champ à la classe `Personne`, de type tableau de réels

```
float[] 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'age de la personne les différents éléments du tableau `comptes`. Créer, au sein de la classe `Personne`, la méthode `void diviserParDeux()`, qui modifie le tableau `comptes` en divisant chaque élément par deux.

b) Afin de tester que le constructeur exhaustif précédent est bien conforme à ce qu'il doit être, créer deux objets de type `Personne`, de même nom, même age et à partir du même objet nombres comportant deux éléments. Appeler

diviserParDeux() sur le premier des objets de type Personne. Appeler la méthode afficher() de ces deux objets. Le code utilisé sera par exemple le suivant :

```
int[] mesNombres = new int[2] ;
mesNombres[0] = 100 ;
mesNombres[1] = 92 ;
Personne sage = new Personne("Agecanonix", 80,
mesNombres) ;
Personne sageBis = new Personne("Agecanonix", 80,
mesNombres) ;
sage.diviserParDeux() ;
```

```
text(sage.afficher(),... ) ;
text(sageBis.afficher(), .., ...)
```

Si l'affichage est identique pour les deux objets, le présent constructeur exhaustif, ainsi que celui que vous avez réalisé à la question précédente ne sont **pas** corrects. Les initialisations des champs nom et nombres doivent alors avoir été faites avec ce que l'on nomme de la copie superficielle (copie des références uniquement), alors qu'elles auraient dues être faites avec de la copie profonde (copie du contenu des objets). Corriger le constructeur exhaustif si nécessaire.