

## Programmation orientée objets

Documents et appareils électroniques interdits. 1.5 heures.

**Exercice 1 (Questions)**

(10 points)

*Question 1* : En Java, qu'est-ce que le « boxing » et le « unboxing » ? Pourquoi est-ce nécessaire ?

*Question 2* : Pourquoi est-il préférable de ne dépendre que des interfaces ?

*Question 3* : Donner les critères permettant de garantir la substituabilité d'un sous-type.

*Question 4* : Quels sont les intérêts d'utiliser des accesseurs ?

*Question 5* : Quelles sont les caractéristiques principales du langage Java ?

*Question 6* : Qu'est-ce que l'héritage ? Quelles alternatives peut-on utiliser ?

*Question 7* : Expliquez la notion de sous-typage ? Quel en est l'intérêt ?

*Question 8* : Qu'est-ce que le polymorphisme ? Quels en sont les types supportés par Java.

*Question 9* : Que signifie le mot-clé Java `final` utilisé sur :

- un attribut,
- une méthode,
- une classe.

*Question 10* : Qu'est-ce que le JIT et à quoi sert-il ?

**Exercice 2**

(2 points)

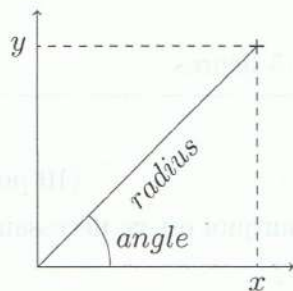
Soient les deux classes Java suivante :

```
class A {  
    private int a;  
    A(int v) {  
        a = v;  
    }  
    public int foo(A other) {  
        return a + other.a;  
    }  
}
```

```
class B extends A {  
    B(int v) {  
        super(v);  
    }  
    @Override  
    public int foo(B other) {  
        return 42;  
    }  
}
```

Que se passe-t-il si l'on compile ? Expliquez.

**Exercice 3** (Un point c'est tout !) On se propose ici de mettre en œuvre une classe représentant un point géométrique, dont voici une représentation UML *partielle* :



Point
name : String {readOnly} x : Double y : Double /angle : Double /radius : Double
move(deltaX : Double, deltaY : Double) : void scale(ratio : Double) : void rotate(angle : Double) : void

x et y représentent les coordonnées carthésiennes, et angle et radius les coordonnées polaires. La méthode rotate change l'angle, et scale multiplie x et y par le ratio donné. move déplace le point (en relatif).

**Question 1 (Implémentation) :** (2 pt) Donnez le code Java correspondant, en respectant les conventions du langage (attention, le diagramme n'est pas au niveau d'abstraction de l'implémentation, il faudra donc ajouter les éléments nécessaires).

**Question 2 (Construction) :** (1 pt) On veut pouvoir créer un point en spécifiant soit ses coordonnées carthésiennes, soit ses coordonnées polaires. Comment faire ? Donner le code correspondant.

**Question 3 (Égalité) :** (2 pt) On veut que deux points soient considérés égaux s'ils sont au même endroit. Comment faire ? Donner le code correspondant.

**Question 4 (De la couleur) :** (2 pt) On souhaite étendre cette classe en ajoutant de la couleur. Une couleur est représentée par trois valeurs entières dans  $[0, 255]$  (rouge, vert et bleu). Donner le diagramme UML et le code correspondants (la classe Point ne change pas).

**Question 5 (Forme) :** (1 pt) Une forme est un ensemble d'au moins deux points. Donnez le diagramme UML correspondant. Expliquez la mise en œuvre (pas de code nécessaire).

**Question 6 :** (2 pt) Comment gérer l'égalité des points en prenant en compte les points colorés ?