

TD - Séance n° 6

Interfaces et classes abstraites

1 Questions de cours

Exercice 1 (*Interfaces vs classes abstraites*)

1. Peut-on instancier directement une interface ? une classe abstraite ?
2. Peut-on munir une interface d'un constructeur sans corps ? d'un constructeur avec un corps ? Mêmes questions pour une classe abstraite.
3. Considérons l'instruction suivante : `A a = new B();`
Quelles sont les deux conditions que doit vérifier la classe B pour que cette instruction soit valide :
 - lorsque A est une classe abstraite ?
 - lorsque A est une interface ?
4. Une interface peut-elle contenir des champs ? Avec quels modificateurs ? Doivent-ils être initialisés ? Mêmes questions pour une classe abstraite.
5. Une interface peut-elle contenir des méthodes statiques/non-statiques, abstraites/non-abstraites ? Même question pour une classe abstraite.
6. Une interface peut-elle hériter d'une autre interface ? d'une classe abstraite ?
7. Une classe abstraite peut-elle hériter d'une autre classe abstraite ? d'une interface ?

Exercice 2 (*Interfaces et héritage*) On considère les trois déclarations d'interfaces suivantes, que l'on supposera placées dans des fichiers distincts :

```
public interface I {  
    default void m() {  
        System.out.println("I, m");  
    }  
    void n();  
}  
  
public interface J {  
    void p();  
}  
  
public interface K {  
    void q();  
}
```

Soient à présent les trois déclarations de classes suivantes, que l'on supposera encore dans des fichiers distincts :

```
public abstract class A implements I, J {
    public void n() {
        System.out.println("A, n");
    }
}
public class B extends A {
    public void p() {
        System.out.println("B, p");
    }
}
public class C extends B implements K {
    public void m() {
        System.out.println("C, m");
    }
    public void q() {
        System.out.println("C, q");
    }
}
```

Dans le code ci-dessous, quelles sont les instructions qui déclencheront un message d'erreur, et pourquoi? En supposant ces instructions commentées, qu'affichera ce code?

```
A a = new A(); a.m(); a.n(); a.p();
B b = new B(); b.m(); b.n(); b.p();
C c = new C(); c.m(); c.n(); c.p(); c.q();

A u = b; u.m(); u.n(); u.p();
A v = c; v.m(); v.n(); v.p(); v.q();
```

2 Modélisation

Exercice 3 (*Instruments de musique*) Dans cet exercice, nous allons construire une hiérarchie de classes et d'interfaces permettant de modéliser des instruments de musique. Plusieurs solutions existent pour classer ces instruments.

Une première solution consiste à les différencier selon le procédé qui leur permet de produire un son. Certains instruments sont dits mécaniques, au sens où le son provient d'une vibration mécanique d'une pièce ou d'une masse d'air (tous les instruments traditionnels). Les instruments mécaniques se séparent en trois grandes familles :

- Les cordes, pincées (guitare), frappées (piano), frottées (violon).
- Les vents, divisés en bois (saxophone) et cuivres (trombone).
- Les percussions, à peau (timbale), en bois (xylophone), en métal (triangle).

D'autres instruments, dits électroniques, produisent un son à l'aide d'un générateur (tous les synthétiseurs).

Une seconde solution consiste à différencier les instruments selon la manière dont leur son est amplifié. L'amplification peut à nouveau être mécanique (par une caisse de résonance) ou bien électrique (à l'aide d'un microphone). Une guitare électrique, par exemple, est un instrument mécanique à amplification électrique : le son provient de la vibration d'une corde, puis il est amplifié à l'aide de microphones qui transforment cette vibration en signal électrique.

1. Trouver et représenter graphiquement une hiérarchie de classes et d'interfaces permettant de représenter les différentes sortes d'instruments, ainsi que la manière dont ils sont amplifiés. Insérer dans cette hiérarchie des classes représentant un piano, un saxophone, et une guitare électrique.
Inutile de spécifier l'implémentation des classes : indiquez seulement visuellement quelles sont leurs liens de parenté, quelles sont les interfaces implémentées, et quelles sont les classes qui doivent être abstraites.
2. Donner la déclaration (l'en-tête) des classes représentant un piano, un saxophone, et une guitare électrique.
3. On aimerait bien aussi pouvoir dire qu'un piano, un orgue, un piano synthétiseur, appartiennent tous à la famille des claviers. Comment faire ?

Si vous avez le temps

Exercice 4 Implémenter une ou plusieurs interfaces peut être vu comme la capacité d'un objet à remplir certains rôles, indépendamment de sa position dans la hiérarchie des classes. Cet exercice peut être résolu sans utiliser de `if` ni de `instanceof`.

(1) *Une école de samourais dispense deux types de cours : des cours de combat et des cours de méditation. Chaque cours est assuré par un unique orateur, qui doit avoir été formé dans la spécialité correspondante (c'est-à-dire, formé à la méditation pour un cours de méditation, formé au combat pour un cours de combat, un orateur pouvant être formé aux deux).*

Construire et implémenter deux hiérarchies modélisant la situation décrite ci-dessus : une hiérarchie de classes pour les cours et une hiérarchie d'interfaces (vides) pour les orateurs. Les constructeurs des cours devront, par simple application des règles de typage, interdire la création d'un cours par un orateur non formé à sa spécialité.

(2) *L'école comporte deux types de membres : les élèves et les maîtres. Les élèves sont divisés en deux groupes : apprentis et disciples. Les maîtres se divisent en deux groupes : initiés et grands maîtres.*

L'orateur d'un cours de méditation ne peut être qu'un initié ou un grand maître. L'orateur d'un cours de combat peut être un disciple, un initié ou un grand maître.

Construire et implémenter une hiérarchie de classes supplémentaires (vides) pour les membres de l'école, en respectant cette dernière contrainte.