

Programmation avancée

TP 5 – La Pierre et le Sabre – Interfaces

V. Padovani, PPS - IRIF

Exercice 1 : Pénible, mais interactif

On considère les déclarations de classes suivantes :

```

class A {
    void m () { System.out.println ("A : m()"); n (); }
    void n () { System.out.println ("A : n()"); p (); }
    void p () { System.out.println ("A : p()");      }
}

interface I { void q(); }

class B extends A implements I {
    void p () { System.out.println ("B : p()"); q (); }
    public void q () { System.out.println ("B : q()"); }
}

class C extends B {
    void n () { System.out.println ("C : n()"); p (); }
    public void q () { System.out.println ("C : q()"); }
}

interface J { void r (); }

class D extends C implements J {
    public void n () { System.out.println ("D : n()"); r (); }
    public void r () { System.out.println ("D : r()"); super.n (); }
}
  
```

- Pour chaque fragment ci-dessous, quel est l'affichage résultant ?
- Indiquer pour chaque fragment ci-dessous s'il est compilable ou s'il déclencherà une erreur d'exécution.

A a = new A (); a.m ();

B b = new B (); b.m ();

C c = new C (); c.m ();

D d = new D (); d.m ();

A a = new D (); a.r ();

C c = new D (); J j = c;

I i = new D (); C c = (C) i;

I i = new C (); J j = (J) i; C c = (C) j;

Exercice 2 :

Une école de samouraïs contient deux sortes de membres : les élèves, et les maîtres. Les élèves sont divisés en trois groupes : novices, apprentis et disciples. Les maîtres se divisent en deux groupes : initiés, et grands maîtres.

Chaque cours de l'école est assuré par un unique orateur. L'école donne deux sortes de cours : combat, et méditation. L'orateur d'un cours de méditation ne peut être qu'un initié ou un grand maître. L'orateur d'un cours de combat ne peut être qu'un disciple, un initié ou un grand maître.

1. Construire, d'abord sur le papier, une hiérarchie de classes et d'interfaces modélisant la structure de l'école. Vos choix devront permettre dans l'implémentation de respecter la contrainte suivante : il devra être impossible de construire un cours sans que son orateur ne soit habilité à le faire, le tout sans `if ... else`, sans `instanceof`, sans marquage par des chaînes ou des booléens dans les classes, etc.
Indication : Une interface peut avoir une ou plusieurs extensions.
2. Ecrire une implémentation minimale de votre modélisation.