JAVA

Interfaces

Interfaces

Une interface est une notion correspondant à une classe abstraite où aucune méthode n'est implémentée.

```
public interface I
{
  void f(int n); //public et abstract sont facultatifs
  void g();
}

public class A implements I
{
  //A doit redéfinir f et g
}
```

- Une même classe peut implémenter plusieurs interfaces.

```
public interface I2
{
  int h();
}

class B implements I,I2
{
  //redéfinition de f et g de I et h de I2
}
```

Interfaces

- Les interfaces peuvent contenir des constantes de type `static final`. Ces constantes seront donc accessibles en dehors d'une classe implémentant l'interface.
- Les interfaces peuvent se dériver, mais les classes dérivées obtenues sont aussi des interfaces.

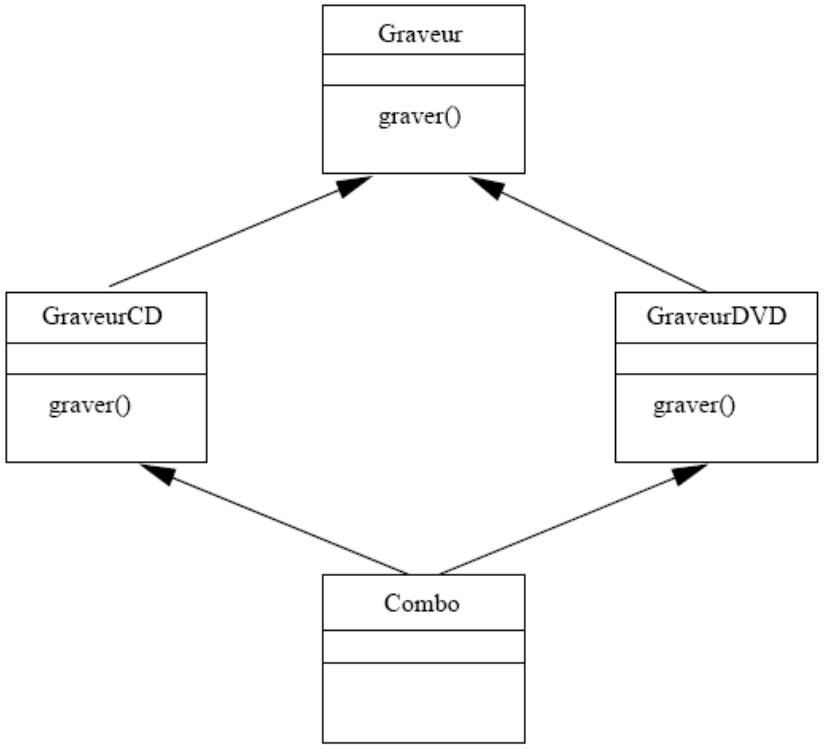
```
interface I1
{
  static final int MAXI = 20;
  void f(int n);
}

interface I2 extends I1
{
  void g();
}

class A implements I1
{
  //redéfinit f, on a accès à MAXI directement
  //par exemple if(i < MAXI) ...
}

class B extends A implements I2
{
  //redéfinition de g
}
```

Interfaces

On peut ainsi écrire les choses suivante d'où on veut :

```
I1 i = new A();
I1 i2 = new B();
I2 b = new B();
System.out.println("Constante "+I1.MAXI);
```

Interfaces



Problème de l'héritage multiple : quelle méthode `graver` s'exécute sur le combo ?

En Java il n'y a pas d'héritage multiple : on ne peut hériter que d'une seule classe. On contourne alors le problème en utilisant des interfaces. Dans l'exemple du dessus, une solution pourrait être de créer une classe `Combo` qui hérite de `Graveur` et qui implémente l'interface `GraveurCD` possédant une méthode `graverCD` et l'interface `graverDVD` possédant une méthode `graverDVD`.

Tester