SCIENCES SORBONNE UNIVERSITÉ

2I002
HÉRITAGE &ABSTRACTION : INTERFACES JAVA

Vincent Guigue

Le cours est inspiré de sources diverses: L. Denoyer, F. Peschanski,...

LIP6

MLIA Machine Learning & Deep Learning for Information Access

... Et certaines situations posent problèmes

- Je fais du dessin... Et je veux pouvoir sauver le résultat.
- Je fais du dessin... Et je veux afficher facilement le résultat à l'écran

... Et certaines situations posent problèmes

- Je fais du dessin... Et je veux pouvoir sauver le résultat.
- Je fais du dessin... Et je veux afficher facilement le résultat à l'écran

$\Rightarrow$ limite de l'héritage : pas d'héritage multiple en JAVA

... Et certaines situations posent problèmes

- Je fais du dessin... Et je veux pouvoir sauver le résultat.
- Je fais du dessin... Et je veux afficher facilement le résultat à l'écran

$\Rightarrow$ limite de l'héritage : pas d'héritage multiple en JAVA

$\Rightarrow$ mais un autre outil: **les interfaces**

Usage

Une interface définit :

- un cahier des charges (e.g. Voiture, Circuit...)
- une propriété (e.g. Serializable, Clonable...)

Elle donne les fonctions à implémenter et leur signature.

≈ Classe abstraite pure

- Elle contient des signatures de fonctions (comme les fonctions abstraites d'une classe abstraite)
- Pas de code (<Java 1.8)
- Pas d'attribut (<Java 1.8)
- Toutes les fonctions sont `public`

Que doit faire une voiture? vue de l'extérieur de l'objet

- (Création)
- Accélérer, freiner, tourner
- Observation (Position, Direction, Vitesse, Dérapage)
- Observation des propriétés (capacités de braquage, vmax...)

```
1  public interface Voiture {
2      // pour le pilotage
3      public void accelerer(double d);
4      public void freiner(double d);
5      public void tourner(double d);
6
7      // pour l'observation
8      public double getVitesse();
9      public Vecteur getPosition();
10     public Vecteur getDirection();
11 }
```

Interface: exemples & syntaxe (2)

Pour une propriété

Qu'est ce qu'un objet Sauvable?

Pour une propriété

Qu'est ce qu'un objet Sauvable?

Réponse: c'est un objet qui contient la méthode suivante

- `void save(String filename)`

On définit donc l'interface suivante:

```
1 public interface Sauvable {
2     public void save(String filename);
3 }
```

Exemple d'objet `Sauvable`: la classe `Vecteur`.

- L'objet `Sauvable` implémente l'interface `Sauvable`.
- Comme pour les classes abstraites, Vecteur doit contenir la méthode (**contrat**): `public void save(String filename)`
- On est donc sûr de pouvoir sauver l'objet Vecteur, il y a donc bien une logique de cahier des charges

```
1 public class Vecteur implements Sauvable{
2 (...)
3     public void save(String filename){
4         (...) // contient le code
5     }
6
7 }
```

- Un objet ne peut hériter que d'**une classe mère**
- Mais un objet peut implémenter **plusieurs interfaces** (c'est à dire respecter plusieurs propriétés)
- Dans un logiciel de géométrie on peut imaginer plusieurs propriétés:
  - Dessinable (méthode `draw`), Deplacable (méthode `move`)...

```
1 public class Polygone extends Figure
2         implements Dessinable, Deplacable{
3 (...)
4   // Herite des methodes de Figure
5   // doit implementer les methodes draw et move
6 }
```

NB: si la classe mère implémente une interface, la classe fille aussi (elle hérite des méthodes de toutes façons)

Interfaces et variables

- Il est possible (et souvent souhaitable) de déclarer une variable d'un type interface
  - Pas d'instanciation d'une interface (comme pour les classes abstraites)
  - Evidemment, on peut faire des tableaux dont le type est celui de l'interface
  - Utilisation intensive dans la suite du cours
- Dans ce cas, seules les méthodes de l'interfaces sont accessibles

- Les interfaces peuvent hériter les unes des autres

```
1 public interface Positionnable{
2   public Vecteur getPosition();
3 }
4
5 public interface Deplacable extends Positionnable{
6   public void move(Vecteur v);
7   // Un objet Deplacable doit implementer getPosition
8 }
```