

les interfaces

1. **Définition des interfaces**
2. Utilité des interfaces
3. Implémentation des interfaces



03 - Manipuler les interfaces

Définition des interfaces

- Comme une classe et une classe abstraite, une interface permet de définir un nouveau type (référence).
- Une interface est une forme particulière de classe où **toutes les méthodes sont abstraites**.

```
<<interface>>  
imprimable
```

```
Imprimer()
```



1. Définition des interfaces
- 2. Utilité des interfaces**
3. Implémentation des interfaces

04 - Manipuler les interfaces

Utilité des interfaces

Utilité des interfaces

Les interfaces permettent de :

- Spécifier des propriétés qui peuvent être utilisées par les classes qui implémentent ces interfaces.
- Obliger les classes qui les implémentent de définir les méthodes abstraites déclarées dans les interfaces.
- Tirer profit du polymorphisme avec des instances dont les classes ne font pas partie de la même hiérarchie d'héritage.



1. Définition des interfaces
2. Utilité des interfaces
- 3. Implémentation des interfaces**

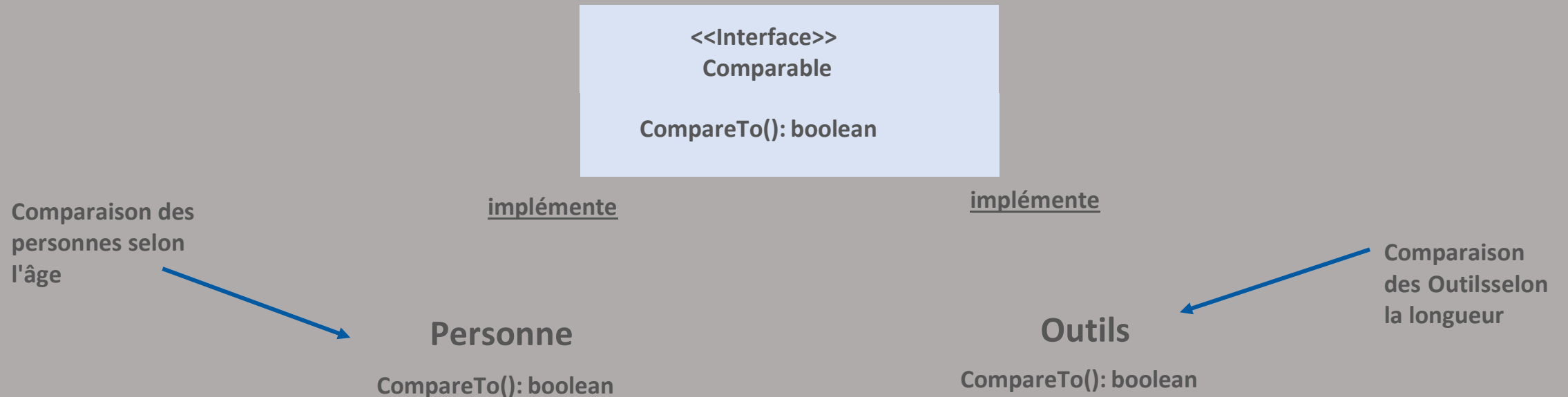
04 - Manipuler les interfaces

Implémentation des interfaces

- On dit qu'une **classe implémente une interface**, si elle fournit une implémentation (c'est-à-dire un corps) pour chacune des méthodes abstraites de cette interface.
- Si une classe implémente plus d'une interface, elle doit implémenter toutes les méthodes abstraites de chacune des interfaces.

Exemple1 :

- Si l'on souhaite caractériser la fonctionnalité de comparaison qui est commune à tous les objets qui ont une relation d'ordre (plus petit, égal, plus grand), on peut définir l'interface Comparable.
- Les classes Personne et Outils qui implémentent l'interface Comparable **doivent présenter une implémentation de la méthode CompareTo()** sinon elles seront abstraites.



Exemple 2 :

- Supposons que nous voulions que les classes dérivées de la classe Forme disposent toutes d'une méthode imprimer() permettant d'imprimer les formes géométriques.

Solution 1:

- Ajouter une méthode abstraite Imprimer() à la classe Forme et ainsi chacune des sous-classes concrètes devrait implémenter cette méthode.

→ Si d'autres classes (qui n'héritent pas de Forme souhaitaient également disposer des fonctions d'impression, elles devraient à nouveau déclarer des méthodes abstraites d'impression dans leur arborescence.

Solution 2 :

La classe forme implémente l'interface imprimable

- ayant la méthode Imprimer()

