

# Extension d'interfaces

Alexis Nasr (d'après les slides de Arnaud Labourel)



Supposons que nous ayons l'interface suivante :

```
public interface List<E> {  
    public int size();  
    public E get(int index);  
}
```

En Java, on peut aussi étendre une interface :

```
public interface ModifiableList<E> extends List<E> {  
    public void add(E value);  
    public void remove(int index);  
}
```

Une classe qui implémente l'interface `ModifiableList` doit implémenter les méthodes `size`, `get`, `add` et `remove`.

# Extension d'interface

Supposons que la classe `ArrayModifiableList` implémente l'interface `ModifiableList`. Dans ce cas, nous pouvons écrire :

```
ModifiableList<Integer> modifiableList;  
modifiableList = new ArrayModifiableList<>();  
modifiableList.add(2);  
modifiableList.add(5);  
modifiableList.remove(0);  
List list = modifiableList;  
System.out.println(list.size());
```

En revanche, il n'est pas possible d'écrire :

```
list.remove(0);  
/* → Cette méthode n'existe pas */  
/* dans l'interface List. */
```

# Extension de plusieurs interfaces

Supposons que nous avons l'interface suivante :

```
public interface Printable { public void print(); }
```

En Java, une interface peut étendre plusieurs interfaces :

```
public interface ModifiablePrintableList  
    extends ModifiableList, Printable {  
}
```

## Remarques

- Nous ne définissons pas de nouvelles méthodes dans l'interface `ModifiablePrintableList`.
- Cette interface ne représente que l'union des interfaces `ModifiableList` et `Printable`.
- De nouvelles méthodes auraient pu être définies dans `ModifiablePrintableList`.