

Formation Java 8

Méthode non abstraite dans une interface

Sommaire

- Du code dans des interfaces Java ?
- Les limites de l'implémentation
- Quid de l'héritage multiple ?
- Méthode statique



Du code dans des interfaces Java ?

Oui, il est à présent possible d'implémenter des méthodes dans une interface via le mot clé **default**.

```
default void myMethod() {  
    // ici une implémentation  
}
```



Pourquoi ?

Permettre d'enrichir les interfaces
“historiques” du JDK sans impacter la
planète.

Par exemple, une méthode **forEach** a pu
être ajoutée à l'**API Collection** sans
demandée une redéfinition de toutes les
classes filles.



Des limites ?

- L'implémentation étant dans une interface, elle ne peut utiliser que :
 - des constantes de l'interface
 - les autres méthodes de l'interface

```
interface MyInterface {  
  
    String CONSTANTE = "HELLO";  
  
    List<String> autreMethod();  
  
    default void myMethod() {  
        String result = CONSTANTE;  
        for(String str : autreMethod()) {  
            result += str;  
        }  
        System.out.println(result);  
    }  
}
```

Des limites ?

- La classe implémentation l'interface n'est pas dans l'obligation de redéfinir une méthode non abstraite.

```
interface interfaceA {  
    default void myMethod() {  
        System.out.println("A");  
    }  
}
```

```
class A implements interfaceA {  
    // pas besoin de redéfinir myMethod  
}
```


Et en cas d'ambiguïté ?

- En cas d'ambiguïté, **la redéfinition devient obligatoire.**

```
interface interfaceA {  
    default void myMethod() {  
        System.out.println("A");  
    }  
}
```

```
interface interfaceB {  
    void myMethod();  
}
```

```
class A implements interfaceA, interfaceB {  
    // Obligation de redéfinir  
    @Override public void myMethod() {  
        System.out.println("AB");  
    }  
}
```

Méthode statique

- Il est désormais également autorisé d'avoir des méthodes statiques dans une interface.

```
interface interfaceB {  
  
    static void print(String text) {  
        System.out.println(text);  
    }  
  
}
```


Travaux Pratiques