Formation Java 17

Méthodes non abstraites dans une interface

Sommaire

Du code dans des interfaces Java ?

Les limites de l'implémentation

Quid de l'héritage multiple

Méthode statique

Java 17

Du code dans des interfaces Java ?

Oui il est à présent possible d'implémenter des méthodes dans une interface !

```
private void myPrivateMethod()
{

}

public default void myMethod()
{

}
```

Pourquoi ?

Permettre d'enrichir les interfaces "historiques" du JDK dans impacter les projets du monde entier.

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

Méthodes par défaut – exemple

Méthodes publiques par défaut dans les interfaces.

Une méthode par défaut est obligatoirement public.

```
public interface MyInterface {

    String RESULTAT = "Résultat";

    List<String> findAll();

    public default void myMethod(){
          String result = null;
          for (String str: findAll()){
              result+=str;
          }
          System.out.println(RESULTAT + ":" result);
    }
}
```

Méthodes privées

Il est à présent possible d'implémenter des méthodes **privées** dans une interface !

Permet de mettre en place des méthodes réutilisables pour les méthodes par défaut.

```
public interface MyInterface {

   private void myPrivateMethod(){

   }
}
```

Des limites ?

La classe implémentant l'interface n'est pas dans l'obligation de redéfinir une **méthode concrète**.

```
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** est **obligatoire**.

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

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

```
class A implements InterfaceA, InterfaceB {

   @Override
   public void myMethod() {
      System.out.println("AB");
   }
}
```

Méthodes statiques

Il est désormais également autorisé d'avoir des méthodes **statiques** dans les interfaces.

```
interface interfaceA {

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

Atelier (TP)

OBJECTIFS : Savoir mettre en œuvre des méthodes par défaut dans les interfaces

DESCRIPTION :

- Dans le TP n°1 vous allez compléter des méthodes par défaut dans des interfaces afin que des classes de tests unitaires déjà développées soient passantes.