JAVA INSIDE
TP 05 - Switch on String et Tests paramétrés

- Test JUnit Paramétré
- Inlining Cache

Lien GitHub : https://github.com/MelissaDaCosta/java-inside

Mélissa DA COSTA

Test Junit Paramétré

Pour appeler plusieurs fois une même méthode de test :

Annotations @ParameterizedTest

Effectue les tests de assertAll pour chacune des fonctions de la Stream renoyée par `testWithMultipleMethods` :

```
@ParameterizedTest
@MethodSource("testWithMultipleMethods")  // Méthode à appeler
public void multipleMethod(ToIntFunction<String> function) {
assertAll(
()->assertEquals(0, function.applyAsInt("foo")),
()->assertEquals(1, function.applyAsInt("bar")),
()->assertEquals(2, function.applyAsInt("bazz")),
()->assertEquals(-1, function.applyAsInt("autre"))
);}
```

@MethodSource

La méthode définit dans l'annotation @MethodSource est appelée par le test @ParameterizedTest.

Elle renvoie la Stream de plusieurs interfaces fonctionnelles qui correspondent aux méthodes de test à appeler.

ToIntFunction car les fonctions à tester renvoient des int.

```
static Stream<ToIntFunction<String>> testWithMultipleMethods()
{
return Stream.of(StringSwitchExample::stringSwitch,
StringSwitchExample::stringSwitch2,
StringSwitchExample::stringSwitch3); // Test plusieurs
méthodes
}
```

Refactoring (Réusinage de code)

Le refactoring est l'étape qui consiste à modifier du code dans le but de plus tard introduire une nouvelle fonctionnalité.

Cela permet d'améliorer la lisibilité et la maintenance.

Ici, nous avons modifier nos tests pour tester plusieurs méthodes à la fois avant d'ajouter une nouvelle méthode a tester.

StringSwitch

Le switch de base sur les String fait un switch sur le hashcode des string.

StringSwitch2 : renvoie un méthod handle qui donne l'index de chaîne passée en argument

StringSwitch3 : Utilise l'inlining cache.

Inlining Cache

L’inlining Cache est une technique consistant à optimiser l’exécution du code en mettant en cache le précédant lookUP.

Ici, cela permet de créer notre arbre d’exécution au fur et à mesure que les méthodes soient appelées avec des Strings différentes.

Résultats stringSwitch

3 Benchmark pour tester les 3 stringSwitch sur un tableau d'un million de valeurs :

StringSwitch1 : 0,002 ± 0,001 s/op

StringSwitch2 : 1,738 ± 0,190 s/op

StringSwitch3 : 15,718 ± 4,282 s/op