

# TP2 Fiabilité logicielle - couverture de tests avec JaCoCo

ARTAUD Cyril

DRAOU Meryem

## 1. Premiers pas avec JaCoCo

### 1.1. Calcul de la couverture par JaCoCo

1) Après lancement des tests on en déduit que :

- La couleur verte indique que l'instruction est couverte.
- La couleur rouge indique que l'instruction n'est pas couverte.
- La couleur jaune indique que la condition n'est pas totalement couverte.

Si une condition est un jaune c'est que toutes les branches ne sont pas couvertes.

Il est possible de couvrir toutes les instructions de la classe Palindrome, il suffit de tester le cas où on a un mot de taille impaire. Le test qui est `@Disabled` permettra de couvrir cette instruction ce qui permettra de couvrir à 100 % les instructions de Palindrome.

Par contre pour PartialCoverage il est impossible de couvrir ses instructions à 100 %. Malgré que nous donnons des valeurs pour x et y en paramètre ces valeurs seront changées au début de la méthode x vaudra 1 et y vaudra -1, la boucle for qui utilisera i comme indice commencera à la valeur de x, donc 1, et nous ne passons pas dans la boucle car  $i \geq y$ . Donc l'instruction dans la boucle ne sera pas couverte car la condition de sortie de boucle sera vraie au premier parcours.

## 2. Etude de couverture de code

### 2.1. Couverture des classes de l'application Complex

Lors de l'analyse de couverture de la classe Complex nous remarquons que seulement 2 méthodes ne sont pas 100 % couvertes. La méthode `isZero()` est partiellement couverte car toutes les possibilités de sa décision ne sont pas couvertes et la méthode `toString()` n'était pas couverte.

Nous ajoutons ensuite le test `testZeroAndFalse()` qui permet que `isZero()` soit couvert à 100 %, et nous ajoutons le test `testToString()` pour couvrir `toString()`. Avec ces tests nous obtenons un taux de couverture de 100 % sur la classe Complex.