

Ewen CLÉMENT – MASTER MIAGE

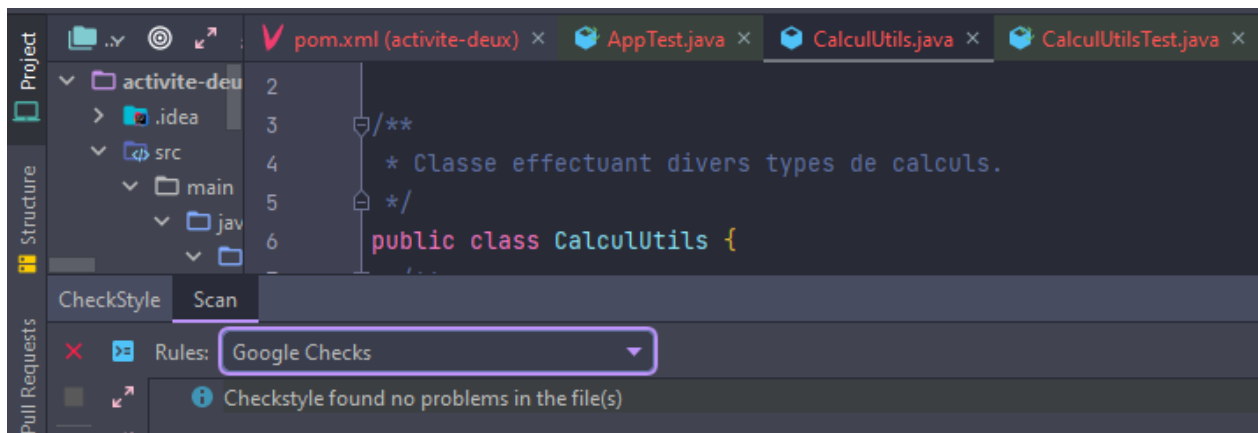
## Activité 2 C306 - Ingénierie du logiciel

### Exercice 1 : Couverture de code

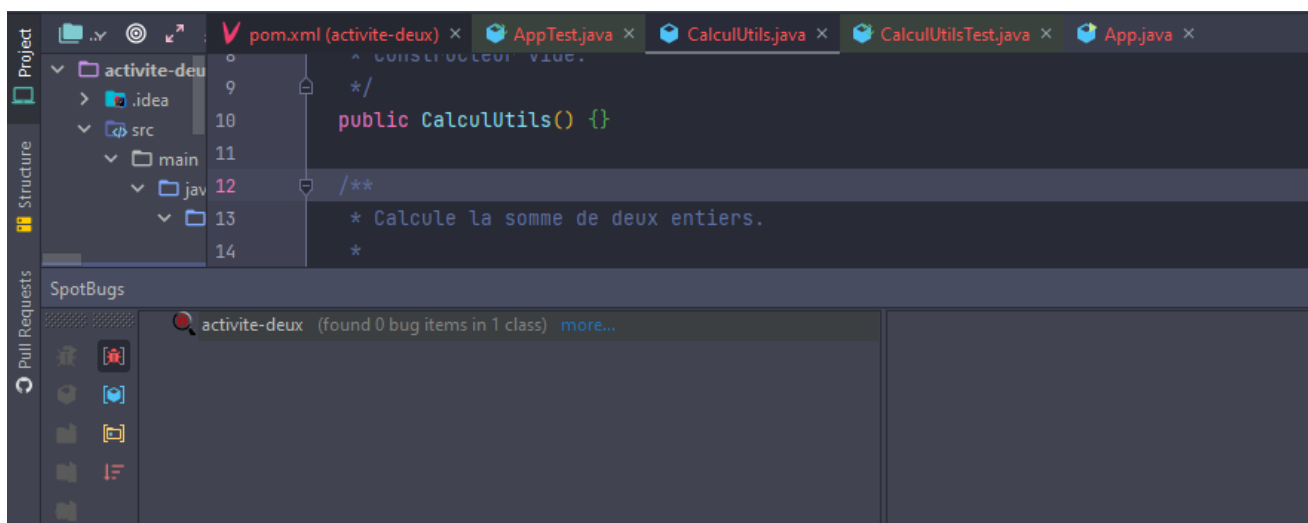
#### **1. Réécrire le code pour qu'il passe les vérifications de checkstyle, spotbugs et PMD.**

Capture d'écran des rapports après réécriture :

Checkstyle :

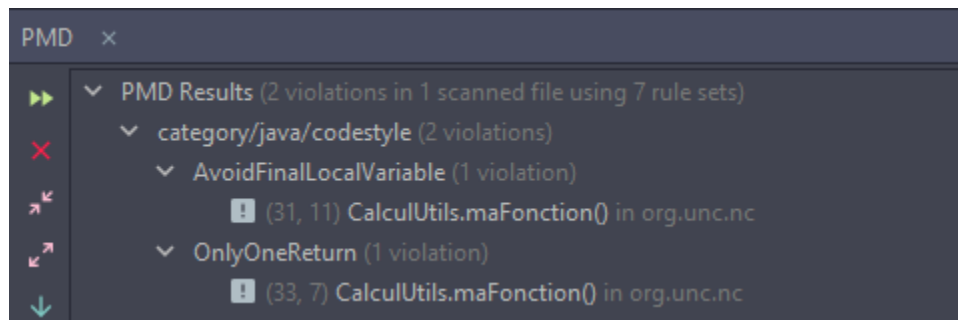


SpotBugs :



PMD :

Ici encore PMD sort des avertissements qui ne sont pas très pertinents, comme le fait de ne pas créer de variable locale finale, qui est une règle dépréciée qui rentre en contradiction avec la règle de `LocalVariableCouldBeFinal` qui elle n'est pas dépréciée. Le premier avertissement quant à lui est `OnlyOneReturn` que l'on a déjà pu rencontrer dans le devoir 1 et qui n'a pas vraiment d'intérêt à mon sens car on peut très bien faire plusieurs return dans une même fonction selon notre besoin.



## 2. Donner le rapport de couverture de code des tests unitaires de 'CalculTest' sur la classe 'Calcul'.

Coverage: CalculUtilsTest x					
50% classes, 44% lines covered in package 'org.unc.nc'					
Element	Class, %	Method, %	Line, %	Branch, %	
App	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)	
CalculUtils	100% (1/1)	66% (2/3)	50% (4/8)	0% (0/1)	

### 3. Expliquer pourquoi la couverture n'est pas complète (en couverture de ligne et en couverture de branche).

La couverture n'est pas complète pour plusieurs raisons. Concernant la couverture de ligne (50%), seule la moitié des lignes sont couvertes, en voici les raisons :

- Les tests ne couvrent pas l'ensemble des cas possibles que l'on peut rencontrer dans les fonctions appelées (somme et division).
- Seule deux des trois fonctions de la classe Calcul sont sollicités dans les tests (maFonction n'est jamais appelée).

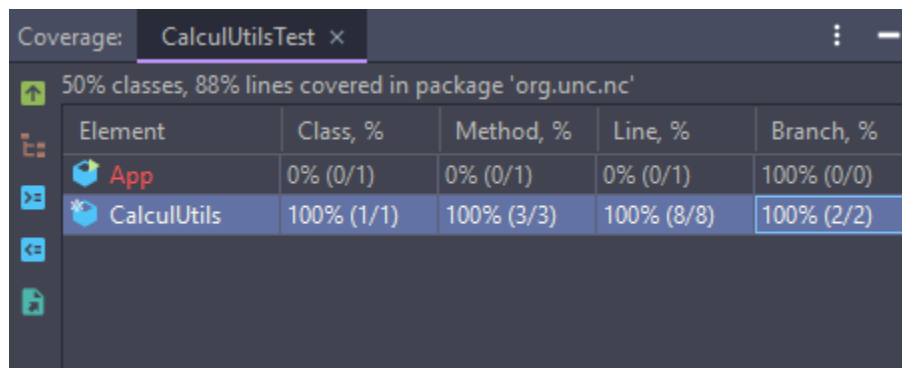
Pour la couverture en branche (0%), celle-ci considère les éléments suivants :

- Chaque branche d'un switch-statement et chaque condition d'un if/else a été prise ;
- Chaque saut fonctionnel (goto, break, continue, return, throw) a été pris ;
- Chaque cas d'exception ("exception handler") a été testé.

Et dans le cas de notre classe, les seuls cas qui correspondent à ces éléments est une exception dans la fonction de division et le if dans la méthode maFonction. Mais comme l'exception n'est jamais levée, elle n'est pas couverte, donc on est à 0% de couverture dans le cas de la couverture en branche, et comme maFonction n'est jamais appelée, elle n'est pas présente dans la statistique de couverture.

### 4. Proposer une amélioration des tests pour obtenir une couverture complète (donner le résultat de couverture obtenu)

En écrivant des tests couvrant tous les cas pour qu'ils puissent passer dans chacune des conditions, on en arrive à une couverture de code atteignant les 100% dans chaque type de couverture :



Coverage: CalculUtilsTest x					
50% classes, 88% lines covered in package 'org.unc.nc'					
Element	Class, %	Method, %	Line, %	Branch, %	
App	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)	
CalculUtils	100% (1/1)	100% (3/3)	100% (8/8)	100% (2/2)	

## Exercice 2 : Compilation assistée

3. Donner la commande permettant d'exécuter toutes les lignes de la question 2 en une fois.

On peut faire : `mvn compile test site` afin de compiler, tester et générer tous les rapports en une fois.

4. Donner les rapports HTML générés







Ils sont présents dans le dossier « rapports Exercice 2 ».

## Exercice 3 : Sudoku

4. Donner le résultat des tests unitaires, la couverture de code sur l'exécution de ces tests unitaires, les rapports checkstyle, spotbugs et PMD.

Rapports présents dans le dossier « rapports Exercice 3 ».

Tests unitaires :

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
App		0%		n/a	2	2	3	3	2	2	1	1
GrilleImpl		100%		100%	0	21	0	35	0	6	0	1
Grille		100%		n/a	0	1	0	1	0	1	0	1
CalculUtils		100%		100%	0	5	0	8	0	3	0	1
Total	7 of 344	97%	0 of 34	100%	2	29	3	47	2	12	1	4

Checkstyle :

### Checkstyle Results

The following document contains the results of Checkstyle 8.29 with google\_checks.xml ruleset. [XML](#)

### Summary

Files	Info	Warnings	Errors
0	0	0	0

### Files

File	I	W	E
------	---	---	---

### Rules

Category	Rule	Violations	Severity
----------	------	------------	----------

### Details

Spotbugs :**SpotBugs Bug Detector Report**

The following document contains the results of [SpotBugs](#) ☞

SpotBugs Version is 4.2.2

Threshold is *medium*

Effort is *default*

**Summary**

Classes	Bugs	Errors	Missing Classes
7	1	0	0

**Files**

Class	Bugs
<a href="#">org.unc.nc.Grille</a>	1

[org.unc.nc.Grille](#)

Bug	Category	Details	Line	Priority
org.unc.nc.Grille.POSSIBLE devrait être sorti de l'interface et mis en package protected	MALICIOUS_CODE	<a href="#">MS_OOI_PKGPROTECT</a> ☞	23	Medium

PMD :**PMD Results**

The following document contains the results of [PMD](#) ☞ 6.38.0.

**Violations By Priority**

Priority 2

[org/unc/nc/App.java](#)

Rule	Violation	Line
<a href="#">SystemPrintIn</a> ☞	System.out.println is used	13

