

## Exercice 1

1-1) Réécriture du code en respectant les recommandations de checkstyle.  
Calcul.java àè

```
Calcul.java x CalculTest.java
home > philemon > Documents > C306-Genie Logiciel > Calcul.java
1  package com.calcul.operations;
2  public class Calcul {
3      /**
4       * Definition d'une constante pour eliiminer nombre magique.
5       * {@value NOMBRES_REFERENCE} ce nombre vaut 10
6       */
7      private static final int NOMBRES_REFERENCE = 10;
8      /**constructor of the class. */
9      protected Calcul() {
10
11     }
12     /**
13     * Calcul la somme de deux nombres.
14     * @return a + b
15     * @param a premier parametre de somme est un entier
16     * @param b second parametre de somme est un entier
17     */
18     public static int somme(final int a, final int b) {
19         return a + b;
20     }
21     /**
22     * @return a / b si b >= 10.
23     * @param a premier parametere de maFonciton est un entier
24     * @param b second parametre de maFonction est un entier
25     */
26     public static int maFonction(final int a, final int b) {
27         if (b >= NOMBRES_REFERENCE) {
28             return a / b;
29         }
30         return b;
31     }
32     /**
33     * @return a / b si b != 0.
34     * @param a premier parametre de division est un entier
35     * @param b second parametre de division est un entier
36     * @throw IllegalArgumentException si b == 0;
37     */
```

```

38 | public static int division(final int a, final int b) {
39 |     if (b == 0) {
40 |         throw new IllegalArgumentException("b ne doit pas etre 0");
41 |     }
42 |     return a / b;
43 | }
44 | }
45 |

```

Le rapport checkstyle de la classe Calcul.java:

```

philemon@philemon:~/Documents/C306-Genie Logiciel$ java -jar checkstyle-8.41-all.jar
-r -c sun_checks.xml Calcul.java
Starting audit...
Audit done.

```

1- 2) Le rapport de couverture de code des tests unitaires avec Jacoco et Junit5.

```

philemon@philemon:~/Documents/C306-Genie Logiciel/Devoir2_C306$ java -javaagent
:jacoco-0.8.6/lib/jacocoagent.jar -jar junit-platform-console-standalone-1.8.0-
M1.jar -classpath out --scan-classpath

```

Thanks for using JUnit! Support its development at <https://junit.org/sponsoring>

```

├─ JUnit Jupiter ✓
│   └─ CalculTest ✓
│       ├── testSomme() ✓
│       ├── testConstructeur() ✓
│       └─ testDivision() ✓
└─ JUnit Vintage ✓

```

Test run finished after 828 ms

```

[ 3 containers found ]
[ 0 containers skipped ]
[ 3 containers started ]
[ 0 containers aborted ]
[ 3 containers successful ]
[ 0 containers failed ]
[ 3 tests found ]
[ 0 tests skipped ]
[ 3 tests started ]
[ 0 tests aborted ]
[ 3 tests successful ]
[ 0 tests failed ]

```

```

philemon@philemon:~/Documents/C306-Genie Logiciel/Devoir2_C306$

```

```

philemon@philemon:~/Documents/C306-Genie Logiciel/Devoir2_C306$ java -jar jacoco-0.8.6/lib/jacococli.jar report jacoco.exec --classfiles out --html out --sourcefiles ./

```

```

[INFO] Loading execution data file /home/philemon/Documents/C306-Genie Logiciel/Devoir2_C306/jacoco.exec.

```

```

[INFO] Analyzing 2 classes.

```

```

philemon@philemon:~/Documents/C306-Genie Logiciel/Devoir2_C306$

```

# JaCoCo Coverage Report

| Element | Missed Instructions    | Cov. | Missed Branches        | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|---------|------------------------|------|------------------------|------|-------------|--------------|----------------|----------------|
| calcul  | <div><div></div></div> | 70%  | <div><div></div></div> | 25%  | 310         | 416          | 18             | 02             |
| Total   | 14 of 47               | 70%  | 3 of 4                 | 25%  | 310         | 416          | 18             | 02             |

JaCoCo Coverage Report > calcul

## calcul

| Element    | Missed Instructions    | Cov. | Missed Branches        | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|------------|------------------------|------|------------------------|------|-------------|--------------|----------------|----------------|
| Calcul     | <div><div></div></div> | 48%  | <div><div></div></div> | 25%  | 36          | 49           | 14             | 01             |
| CalculTest | <div><div></div></div> | 100% | n/a                    |      | 04          | 07           | 04             | 01             |
| Total      | 14 of 47               | 70%  | 3 of 4                 | 25%  | 310         | 416          | 18             | 02             |

JaCoCo Coverage Report > calcul > Calcul

## Calcul

| Element              | Missed Instructions    | Cov. | Missed Branches        | Cov. | Missed Cxty | Missed Lines | Missed Methods |
|----------------------|------------------------|------|------------------------|------|-------------|--------------|----------------|
| maFonction(int, int) | <div><div></div></div> | 0%   | <div><div></div></div> | 0%   | 22          | 33           | 11             |
| division(int, int)   | <div><div></div></div> | 54%  | <div><div></div></div> | 50%  | 12          | 13           | 01             |
| somme(int, int)      | <div><div></div></div> | 100% | n/a                    |      | 01          | 01           | 01             |
| Calcul()             | <div><div></div></div> | 100% | n/a                    |      | 01          | 02           | 01             |
| Total                | 14 of 27               | 48%  | 3 of 4                 | 25%  | 36          | 49           | 14             |

```

1. package calcul;
2. public class Calcul {
3. /**
4.  * Definition d'une constante pour eliiminer nombre magique.
5.  * {@value NOMBRES_REFERENCE} ce nombre vaut 10
6.  */
7. private static final int NOMBRES_REFERENCE = 10;
8. /**constructor of the class. */
9. protected Calcul() {
10.
11. }
12. /**
13.  * Calcul la somme de deux nombres.
14.  * @return a + b
15.  * @param a premier parametre de somme est un entier
16.  * @param b second parametre de somme est un entier
17.  */
18. public static int somme(final int a, final int b) {
19. return a + b;
20. }
21. /**
22.  * @return a / b si b >= 10.
23.  * @param a premier parametere de maFonciton est un entier
24.  * @param b second parametre de maFonction est un entier
25.  */
26. public static int maFonction(final int a, final int b) {
27. if (b >= NOMBRES_REFERENCE) {
28. return a / b;
29. }
30. return b;
31. }
32. /**
33.  * @return a / b si b != 0.
34.  * @param a premier parametre de division est un entier
35.  * @param b second parametre de division est un entier
36.  * @throw IllegalArgumentException si b == 0;
37.  */
38. public static int division(final int a, final int b) {
39. if (b == 0) {
40. throw new IllegalArgumentException("b ne doit pas etre 0");
41. }
42. return a / b;
43. }
44. }

```

1-3) Les rapports laissent voir que la méthode **maFonction** n'a pas du tout été testée (0%), et division l'a été testée partiellement (50%).

le rapport de couverture de code pour **maFonction** est de 0% car aucun test n'a été proposé jusqu'ici pour le tester.

le rapport pour **division** est de 50% car le test propose a permis de tester une seule condition de la structure "If" (c'est une couverture en branche);

La méthode: **somme()** et le constructeur **Calcul()** ont eu une couverture lineaire de 100% chacuns.

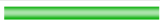
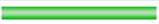




1-4) Pour une couverture de code complète, nous avons ajouté d'autres méthodes de test.

```
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;

public class CalculTest {
    @Test
    public void testConstructeur() {
        new Calcul();
    }
    @Test
    public void testSomme() {
        assertEquals(5, Calcul.somme(2,3));
    }
    @Test
    public void testMaFonction(){
        assertEquals(4, Calcul.maFonction(40,10));
        assertEquals(9, Calcul.maFonction(20,9));
    }
    @Test
    public void testDivision() {
        assertEquals(4, Calcul.division(8,2));
        assertEquals(2, Calcul.division(4,0));
    }
}
```

et nous obtenons une couverture de 100%.

## Calcul

| Element                                | Missed Instructions   | Cov. | Missed Branches   | Cov. | Missed Cxty | Missed Lines | Missed Methods |
|--|---|------|---|------|-------------|--------------|----------------|
| • <a href="#">division(int, int)</a>   |  | 100% |  | 100% | 0 2         | 0 3          | 0 1            |
| • <a href="#">maFonction(int, int)</a> |  | 100% |  | 100% | 0 2         | 0 3          | 0 1            |
| • <a href="#">somme(int, int)</a>      |  | 100% |   | n/a  | 0 1         | 0 1          | 0 1            |
| • <a href="#">Calcul()</a>             |  | 100% |   | n/a  | 0 1         | 0 2          | 0 1            |
| Total                                  | 0 of 27   | 100% | 0 of 4  | 100% | 0 6         | 0 9          | 0 4            |

```

1. package calcul;
2. public class Calcul {
3.     /**
4.      * Definition d'une constante pour eliminer nombre magique.
5.      * {@value NOMBRES_REFERENCE} ce nombre vaut 10
6.      */
7.     private static final int NOMBRES_REFERENCE = 10;
8.     /**constructor of the class. */
9.     protected Calcul() {
10.
11.     }
12.     /**
13.      * Calcul la somme de deux nombres.
14.      * @return a + b
15.      * @param a premier parametre de somme est un entier
16.      * @param b second parametre de somme est un entier
17.      */
18.     public static int somme(final int a, final int b) {
19.         return a + b;
20.     }
21.     /**
22.      * @return a / b si b >= 10.
23.      * @param a premier parametre de maFonction est un entier
24.      * @param b second parametre de maFonction est un entier
25.      */
26.     public static int maFonction(final int a, final int b) {
27.         if (b >= NOMBRES_REFERENCE) {
28.             return a / b;
29.         }
30.         return b;
31.     }
32.     /**
33.      * @return a / b si b != 0.
34.      * @param a premier parametre de division est un entier
35.      * @param b second parametre de division est un entier
36.      * @throws IllegalArgumentException si b == 0;
37.      */
38.     public static int division(final int a, final int b) {
39.         if (b == 0) {
40.             throw new IllegalArgumentException("b ne doit pas etre 0");
41.         }
42.         return a / b;
43.     }
44. }

```

## Exercice 2 - Compilation assistée

2-1) Les chemins du projet ont été adaptés pour permettre une utilisation de Maven:

Devoir2\_C306/src/main/java/Calcul.java

Devoir2\_C306/src/test/java/CalculTest.java

Devoir2\_C306/target/classes/Calcul.class

Devoir2\_C306/target/test-classes/CalculTest.class

2-2) Le fichier pom.xml utilisé a été copié du module C306 et complété avec les plugins de Spotbug, et Javadoc.

2-3) on utilise **mvn test** pour exécuter toutes les lignes de codes proposées en 2-2.

2-4) mvn site:site à générer un site html pour tous les rapports de Maven.

2-5) Codes sources soumis.

### Exercice 3 - Implémentation de SUDOKU.

#### 3-1) GrilleImpl.java

```
import java.util.ArrayList;
import java.util.List;

/** Implementation d'une grille */
public class GrilleImpl implements Grille {

    private int width = 0;
    private int height = 0;
    private char [][] lagrille = null;

    public GrilleImpl(final int wid, final int hei){
        this.width = wid;
        this.height = hei;
        this.lagrille = new char [wid][hei];
        for(int i = 0; i < this.width; i++){
            for(int j = 0; j < this.height; j++){
                lagrille[i][j] = EMPTY;
            }
        }
    }

    /** @return largeur/hauteur de la grille */
    public int getDimension(){
        return this.width/this.height;
    }

    /**
     * Affecte une valeur dans la grille
     * @param x position x dans la grille
     * @param y position y dans la grille
     * @param value a mettre dans la case
     */
}
```

```

* @throw IllegalArgumentException si x ou y sont hors bornes (0-8)
* @throw IllegalArgumentException si la valeur est interdite aux vues
des
* autres valeurs de la grille
* @throw IllegalArgumentException si value n'est pas un caractere
autorise ('1',..., '9')
*/
public int setValue(final int x, final int y, final char value) throws
IllegalArgumentException{
    if(x > this.width || y > this.height || x < 0 || y < 0) throw new
IllegalArgumentException("x or y hors bornes");
    if (!possibleContains(value)) throw new
IllegalArgumentException("value n'est pas un caractere autorise");
    this.lagrille[x-1][y-1] = value;
    return 0;
}
/**
* Recupere une valeur de la grille
* @param x position x dans la grille
* @param y position y dans la grille
* @return valeur dans la case x,y
* @throw IllegalArgumentException si x ou y sont hors bornes (0-8)
*/
public char getValue(final int x, final int y) throws
IllegalArgumentException{
    if(x > this.width || y > this.height || x < 0 || y < 0) throw new
IllegalArgumentException("x or y hors bornes");
    return this.lagrille[x-1][y-1];
}
/**
* Test si une grille est terminee
* @return true si la grille est complete
*/
public boolean complete(){
    for(int i = 0; i < this.width; i++){
        for(int j = 0; j < this.height; j++){
            if(lagrille[i][j] == EMPTY) return false;
        }
    }
    return true;
}
/**

```



```

* Test si une valeur est possible dans la grille par rapport a ce
qu'elle contient deja
* contient deja
* @param x position x dans la grille
* @param y position y dans la grille
* @param value valeur a mettre dans la case
* @throw IllegalArgumentException si x ou y sont hors bornes (0-8)
* @throw IllegalArgumentException si value n'est pas un caractere
autorise ('1',..., '9')
*/
public boolean possible(final int x, final int y, final char value)
throws IllegalArgumentException{
    if(x > this.width || y > this.height || x < 0 || y < 0) throw new
IllegalArgumentException("x or y hors bornes");
    if (!possibleContains(value)) throw new
IllegalArgumentException("value n'est pas un caractere autorise");

    for(int i = 0; i < this.width; i++){
        if((lagrille[i][y-1] == value) && ((x-1) != i)) return false;
    }

    for(int j = 0; j < this.height; j++){
        if((lagrille[x-1][j] == value) && ((y-1) != j)) return false;
    }

    return true;
}

/**
* verifie si @param value est un caractere autorise
*/
public boolean possibleContains(char value){
    List possibleAL = new ArrayList();
    for(int i = 0; i < possible.length; i++){
        possibleAL.add(possible[i]);
    }

    return (possibleAL.contains(value) || value == EMPTY);
}
}

```

## GrilleImplTest.java

```
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;

public class GrilleImplTest {

    @Test
    public void testGetDimension() {
        GrilleImpl test = new GrilleImpl(4,4);
        assertEquals(1, test.getDimension());
    }

    @Test
    public void testSetValue() {
        GrilleImpl test = new GrilleImpl(4,4);
        assertEquals(0, test.setValue(2,4, '6'));
    }

    @Test
    public void testGetValue() {
        GrilleImpl test = new GrilleImpl(4,4);
        test.setValue(2,4, '6');
        assertEquals('6', test.getValue(2,4));
    }

    @Test
    public void testComplete() {
        GrilleImpl test = new GrilleImpl(4,4);
        test.setValue(1,1, '1');
        test.setValue(1,2, '2');
        test.setValue(1,3, '3');
        test.setValue(1,4, '4');
        test.setValue(2,1, '5');
        test.setValue(2,2, '6');
        test.setValue(2,3, '7');
        test.setValue(2,4, '8');
        test.setValue(3,1, '9');
        test.setValue(3,2, 'a');
        test.setValue(3,3, 'b');
        test.setValue(3,4, 'c');
        test.setValue(4,1, 'd');
        test.setValue(4,2, 'e');
        test.setValue(4,3, 'f');
        test.setValue(4,4, '0');
        assertEquals(true, test.complete());
    }
}
```

```

}
@Test
public void testPossible() {
    GrilleImpl test = new GrilleImpl(4,4);
    test.setValue(3,1,'6');
    assertEquals(true, test.possible(2,4,'6'));
}
}

```

3-3) les rapports Jacoco sont listés ci-dessous. Ceux de checkstyle et PMD en **ANNEXE**.

#### résultat des tests:

```

[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running GrilleImplTest
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.028 s -
in GrilleImplTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] <<< maven-surefire-report-plugin:3.0.0-M3:report < [surefire]test @ Sudoku
U <<<

```

#### default

| Element    | Missed Instructions | Cov.  | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|------------|---------------------|-------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| GrilleImpl |                     | 82 %  |                 | 57 % | 21     | 35   | 8      | 41    | 0      | 7       | 0      | 1       |
| Grille     |                     | 100 % |                 | n/a  | 0      | 1    | 0      | 1     | 0      | 1       | 0      | 1       |
| Total      | 45 of 324           | 86 %  | 24 of 56        | 57 % | 21     | 36   | 8      | 42    | 0      | 8       | 0      | 2       |

DevoirC306 > default > GrilleImpl

#### GrilleImpl

| Element                  | Missed Instructions | Cov.  | Missed Branches | Cov.  | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------------------|---------------------|-------|-----------------|-------|--------|------|--------|-------|--------|---------|
| possible(int, int, char) |                     | 69 %  |                 | 50 %  | 9      | 12   | 4      | 11    | 0      | 1       |
| setValue(int, int, char) |                     | 74 %  |                 | 50 %  | 5      | 6    | 2      | 6     | 0      | 1       |
| getValue(int, int)       |                     | 82 %  |                 | 50 %  | 4      | 5    | 1      | 3     | 0      | 1       |
| possibleContains(char)   |                     | 87 %  |                 | 50 %  | 2      | 4    | 0      | 4     | 0      | 1       |
| complete()               |                     | 92 %  |                 | 83 %  | 1      | 4    | 1      | 5     | 0      | 1       |
| GrilleImpl(int, int)     |                     | 100 % |                 | 100 % | 0      | 3    | 0      | 11    | 0      | 1       |
| getDimension()           |                     | 100 % |                 | n/a   | 0      | 1    | 0      | 1     | 0      | 1       |
| Total                    | 45 of 256           | 82 %  | 24 of 56        | 57 %  | 21     | 35   | 8      | 41    | 0      | 7       |

#### Resources

- [https://www.youtube.com/watch?v=xND\\_1YBtMNo](https://www.youtube.com/watch?v=xND_1YBtMNo)
- le module de cours C306
- <https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-javadoc-plugin/3.2.0>
- <https://spotbugs.readthedocs.io/en/stable/maven.html>

## Résultats Checkstyle

Le document suivant contient les résultats de [Checkstyle](#) 6.18 avec les règles [sun\\_checks.xml](#). 

## Résumé

| Fichiers |  Infos |  Avertissements |  Erreurs |
|----------|---|--|---|
| 2        | 0   | 0  | 11  |

## Fichiers


| Fichier                         |  I |  A |  E |
|---------------------------------|---|--|---|
| <a href="#">Grille.java</a>     | 0   | 0  | 1   |
| <a href="#">GrilleImpl.java</a> | 0   | 0  | 10  |

## Règles




| Catégorie | Règle                              | Violations | Sévérité   |
|-----------|------------------------------------|------------|--|
| design    | <a href="#">DesignForExtension</a> | 6          |  Erreur |
| javadoc   | <a href="#">JavadocPackage</a>     | 1          |  Erreur |
| sizes     | <a href="#">LineLength</a>         | 4          |  Erreur |








## Détails

### Grille.java

| Sévérité   | Catégorie | Règle          | Message                             | Ligne |
|--|-----------|----------------|-------------------------------------|-------|
|  Erreur | javadoc   | JavadocPackage | Manquant fichier package-info.java. |       |

### GrilleImpl.java

| Sévérité   | Catégorie | Règle              | Message  | Ligne              |
|--|-----------|--------------------|--|--------------------|
|  Erreur | design    | DesignForExtension | La méthode 'getDimension' n'est pas conçue pour être dérivée - il faut la déclarer abstraite, finale ou la laisser vide. | <a href="#">32</a> |
|  Erreur | sizes     | LineLength         | La ligne excède 80 caractères.   | <a href="#">47</a> |
|  Erreur | design    | DesignForExtension | La méthode 'setValue' n'est pas conçue pour être dérivée - il faut la déclarer abstraite, finale ou la laisser vide.     | <a href="#">47</a> |

|  |        |                    |  |     |
|--|--------|--------------------|--|-----|
|  Erreur | sizes  | LineLength         | La ligne excède 80 caractères.   | 52  |
|  Erreur | design | DesignForExtension | La méthode 'getValue' n'est pas conçue pour être dérivée - il faut la déclarer abstraite, finale ou la laisser vide.         | 63  |
|  Erreur | design | DesignForExtension | La méthode 'complete' n'est pas conçue pour être dérivée - il faut la déclarer abstraite, finale ou la laisser vide.         | 73  |
|  Erreur | sizes  | LineLength         | La ligne excède 80 caractères.   | 94  |
|  Erreur | design | DesignForExtension | La méthode 'possible' n'est pas conçue pour être dérivée - il faut la déclarer abstraite, finale ou la laisser vide.         | 94  |
|  Erreur | sizes  | LineLength         | La ligne excède 80 caractères.   | 99  |
|  Erreur | design | DesignForExtension | La méthode 'possibleContains' n'est pas conçue pour être dérivée - il faut la déclarer abstraite, finale ou la laisser vide. | 119 |

## Résultats PMD

Le document suivant contient les résultats de [PMD](#) 6.13.0.

## Violations By Priority

### Priority 4

#### GrilleImpl.java

| Rule                               | Violation            | Ligne               |
|------------------------------------|----------------------|---------------------|
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">103</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">103</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">103</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">109</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">109</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">109</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | <a href="#">124</a> |

## Fichiers

#### GrilleImpl.java

| Rule                               | Violation            | Priority | Ligne               |
|------------------------------------|----------------------|----------|---------------------|
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">103</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">103</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">103</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">109</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">109</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">109</a> |
| <a href="#">UselessParentheses</a> | Useless parentheses. | 4        | <a href="#">124</a> |