

Correction des devoirs

Exercice 1 : Ecriture de code

1. Rapport de "checkstyle" appliqué à ce code.

```
1. Début de la vérification...
2. [ERROR] D:\\Exemple.java:1: Il manque un caractère NewLine à la fin du fichier. [NewLineAtEndOfFile]
3. [ERROR] D:\\Exemple.java:1: Le fichier package-info.java est manquant. [JavadocPackage]
4. [ERROR] D:\\Exemple.java:7:2: Commentaire Javadoc manquant. [JavadocVariable]
5. [ERROR] D:\\Exemple.java:8:2: Commentaire Javadoc manquant. [JavadocVariable]
6. [ERROR] D:\\Exemple.java:9:1: Commentaire Javadoc manquant. [MissingJavadocMethod]
7. [ERROR] D:\\Exemple.java:9:16: Le paramètre t devrait être final. [FinalParameters]

8. [ERROR] D:\\Exemple.java:9:23: 't' masque un attribut. [HiddenField]
9. [ERROR] D:\\Exemple.java:18:24: Le paramètre v devrait être final. [FinalParameters]

10. [ERROR] D:\\Exemple.java:18:28: 'v' masque un attribut. [HiddenField]
11. [ERROR] D:\\Exemple.java:18:30: '{' à la colonne 30 devrait avoir un saut de ligne a
    près. [LeftCurly]
12. [ERROR] D:\\Exemple.java:18:30: Il manque une espace après '{'. [WhitespaceAround]
13. [ERROR] D:\\Exemple.java:18:30: Il manque une espace avant '{'. [WhitespaceAround]
14. [ERROR] D:\\Exemple.java:18:38: Il manque une espace après '='. [WhitespaceAround]
15. [ERROR] D:\\Exemple.java:18:40: Il manque une espace après ';'. [WhitespaceAfter]
16. [ERROR] D:\\Exemple.java:18:41: Il manque une espace avant '}'. [WhitespaceAround]
17. [ERROR] D:\\Exemple.java:23: L'instruction 'if' devrait utiliser des accolades ('{
    et '}'). [NeedBraces]
18. [ERROR] D:\\Exemple.java:23:4: Il y a une espace de trop après '('. [ParenPad]
19. Vérification terminée.
20. Checkstyle se termine par 17 erreurs.
```

2. Réécriture en tenant compte des problèmes soulevés par "checkstyle".

```
1. package fr.u_picardie.active.c306;
2.
3. /**
4.  * Ma class d'exemple.
5.  */
6. public class Exemple {
7.     /**
8.      * Propriété t de la classe Exemple.
9.      */
10.    private final String t;
11.    /**
12.     * Propriété v de la classe Exemple.
13.     */
14.    private final int v;
15.    /**
16.     * Constructeur avec parametre.
17.     * @param t de type chaine de caractere
18.     */
19.    public Exemple(final String tParam) {
20.        t = tParam;
21.    }
22.    /**
```

```

23. * @return la valeur de v
24. */
25. public int getV() {
26.     return this.v;
27. }
28. public final void setV(final int vParam) { this.v = vParam;
29. }
30. /**
31. * @return t si v est positif
32. */
33. public final String getT() {
34.     if ( v > 0 ) { return t; }
35. }
36. }

```

3. Rapport de "checkstyle" appliqué au code réécrit en justifiant les éventuelles anomalies restantes.

```

1. Début de la vérification...
2. [ERROR] D:\\Exemple.java:17:3: Balise Javadoc @param inutilisée pour 't'. [JavadocMethod]
3. [ERROR] D:\\Exemple.java:19:29: Balise Javadoc @param manquante pour 'tParam'. [JavadocMethod]
4. [ERROR] D:\\Exemple.java:28:1: Commentaire Javadoc manquant. [MissingJavadocMethod]
5. [ERROR] D:\\Exemple.java:28:42: '{' à la colonne 42 devrait avoir un saut de ligne après. [LeftCurly]
6. [ERROR] D:\\Exemple.java:34:4: Il y a une espace de trop après '('. [ParenPad]
7. [ERROR] D:\\Exemple.java:34:13: '{' à la colonne 13 devrait avoir un saut de ligne après. [LeftCurly]
8. Vérification terminée.
9. Checkstyle se termine par 6 erreurs.

```

Explication des erreurs :

- ***[ERROR] D:\\Exemple.java:17:3: Balise Javadoc @param inutilisée pour 't'. [JavadocMethod]***

Suite à l'erreur HiddenField, j'ai renommé le paramètre t en tParam ce qui fait que le paramètre t n'existe plus. Pour pallier à ce problème nous devons remplacer dans la documentation de la méthode «@param t » par «@param tParam »

- ***[ERROR] D:\\Exemple.java:19:29: Balise Javadoc @param manquante pour 'tParam'. [JavadocMethod]***

Cette erreur justifie le fait que, le paramètre «tParam » ne figure pas dans la documentation de la méthode

- ***[ERROR] D:\\Exemple.java:28:1: Commentaire Javadoc manquant. [MissingJavadocMethod]***

Cette erreur justifie le fait qu'il n'existe aucune documentation de la méthode « setV »

- ***[ERROR] D:\\Exemple.java:28:42: '{' à la colonne 42 devrait avoir un saut de ligne après. [LeftCurly]***

Cette erreur justifie le fait que, tous ce qui vient après '{' se retrouve sur une même ligne alors que ces instructions devraient être écrites à la ligne suivante.

- ***[ERROR] D:\\Exemple.java:34:4: Il y a une espace de trop après '('. [ParenPad]***

Cette erreur justifie le fait qu'il ne devrait pas y avoir d'espace après l'instruction if.

- **[ERROR] D:\\Exemple.java:34:13: '{' à la colonne 13 devrait avoir un saut de ligne après.**
[LeftCurly]

Cette erreur justifie le fait que, tous ce qui vient après '{' se retrouve sur une même ligne alors que ces instructions devraient être écrites à la ligne suivante.

Exercice 2 : Tests unitaires

1. Ecriture des tests unitaires permettant de tester les méthodes à implémenter.

Code source

```
1. @Test
2. public void testPlusGrand() {
3.     assertEquals(5, TabAlgos.plusGrand(new int[] {1, 2, 3, 4, 5}));
4. }
5.
6. @Test
7. public void testMoyenne() {
8.     try {
9.         TabAlgos.moyenne(null);
10.        fail("tab est null");
11.    } catch (IllegalArgumentException e) {
12.
13.    }
14. }
15.
16. @SuppressWarnings("deprecation")
17. @Test
18. public void testMoyenneGood() {
19.     assertEquals(2, TabAlgos.moyenne(new int[] {2, 2, 2, 2}), 0.1);
20. }
21.
22. @Test
23. public void testEgaux() {
24.     assertEquals(true, TabAlgos.egaux(new int[] {1, 2, 3, 4, 5}, new int[] {1, 2, 3,
25. 4, 5}));
26. }
27. @Test
28. public void testSimilaires() {
29.     assertEquals(true, TabAlgos.similaires(new int[] {1, 2, 3, 4, 5}, new int[] {5,
30. 4, 3, 2, 1}));
31. }
```

Rapport

```
1. Début de la vérification...
2. [ERROR] D:\\TabAlgos.java:1: Il manque un caractère NewLine à la fin du fichier. [NewlineAtEndOfFile]
3. [ERROR] D:\\TabAlgos.java:10: La ligne excède 80 caractères (trouvé 86). [LineLength]
4. [ERROR] D:\\TabAlgos.java:10: Line has trailing spaces. [RegexpSingleline]
5. [ERROR] D:\\TabAlgos.java:10:1: Le fichier contient des caractères de tabulation (ce n'est que la première occurrence). [FileTabCharacter]
6. [ERROR] D:\\TabAlgos.java:11: La ligne excède 80 caractères (trouvé 86). [LineLength]
```

```
7. [ERROR] D:\\TabAlgos.java:11: Line has trailing spaces. [RegexpSingleline]
8. [ERROR] D:\\TabAlgos.java:13:49: Balise Javadoc @param manquante pour 'tab'. [JavadocMethod]
9. [ERROR] D:\\TabAlgos.java:14: La ligne excède 80 caractères (trouvé 92). [LineLength]
10. [ERROR] D:\\TabAlgos.java:14: Line has trailing spaces. [RegexpSingleline]
11. [ERROR] D:\\TabAlgos.java:17: La ligne excède 80 caractères (trouvé 86). [LineLength]
12. [ERROR] D:\\TabAlgos.java:17: Line has trailing spaces. [RegexpSingleline]
13. [ERROR] D:\\TabAlgos.java:18: La ligne excède 80 caractères (trouvé 86). [LineLength]
14. [ERROR] D:\\TabAlgos.java:18: Line has trailing spaces. [RegexpSingleline]
15. [ERROR] D:\\TabAlgos.java:19: La ligne excède 80 caractères (trouvé 86). [LineLength]
16. [ERROR] D:\\TabAlgos.java:19: Line has trailing spaces. [RegexpSingleline]
17. [ERROR] D:\\TabAlgos.java:21:50: Balise Javadoc @param manquante pour 'tab'. [JavadocMethod]
18. [ERROR] D:\\TabAlgos.java:22: La ligne excède 80 caractères (trouvé 92). [LineLength]
19. [ERROR] D:\\TabAlgos.java:22: Line has trailing spaces. [RegexpSingleline]
20. [ERROR] D:\\TabAlgos.java:25: La ligne excède 80 caractères (trouvé 86). [LineLength]
21. [ERROR] D:\\TabAlgos.java:25: Line has trailing spaces. [RegexpSingleline]
22. [ERROR] D:\\TabAlgos.java:26: La ligne excède 80 caractères (trouvé 86). [LineLength]
23. [ERROR] D:\\TabAlgos.java:26: Line has trailing spaces. [RegexpSingleline]
24. [ERROR] D:\\TabAlgos.java:27: La ligne excède 80 caractères (trouvé 89). [LineLength]
25. [ERROR] D:\\TabAlgos.java:27: Line has trailing spaces. [RegexpSingleline]
26. [ERROR] D:\\TabAlgos.java:28: La ligne excède 80 caractères (trouvé 87). [LineLength]
27. [ERROR] D:\\TabAlgos.java:28: Line has trailing spaces. [RegexpSingleline]
28. [ERROR] D:\\TabAlgos.java:29: La ligne excède 80 caractères (trouvé 86). [LineLength]
29. [ERROR] D:\\TabAlgos.java:29: Line has trailing spaces. [RegexpSingleline]
30. [ERROR] D:\\TabAlgos.java:31:49: Balise Javadoc @param manquante pour 'tab1'. [JavadocMethod]
31. [ERROR] D:\\TabAlgos.java:31:53: Il manque une espace après ','. [WhitespaceAfter]
32. [ERROR] D:\\TabAlgos.java:31:66: Balise Javadoc @param manquante pour 'tab2'. [JavadocMethod]
33. [ERROR] D:\\TabAlgos.java:32: La ligne excède 80 caractères (trouvé 92). [LineLength]
34. [ERROR] D:\\TabAlgos.java:32: Line has trailing spaces. [RegexpSingleline]
35. [ERROR] D:\\TabAlgos.java:35: La ligne excède 80 caractères (trouvé 86). [LineLength]
36. [ERROR] D:\\TabAlgos.java:35: Line has trailing spaces. [RegexpSingleline]
37. [ERROR] D:\\TabAlgos.java:36: La ligne excède 80 caractères (trouvé 86). [LineLength]
38. [ERROR] D:\\TabAlgos.java:36: Line has trailing spaces. [RegexpSingleline]
39. [ERROR] D:\\TabAlgos.java:37: La ligne excède 80 caractères (trouvé 89). [LineLength]
40. [ERROR] D:\\TabAlgos.java:37: Line has trailing spaces. [RegexpSingleline]
41. [ERROR] D:\\TabAlgos.java:38: La ligne excède 80 caractères (trouvé 87). [LineLength]
42. [ERROR] D:\\TabAlgos.java:38: Line has trailing spaces. [RegexpSingleline]
43. [ERROR] D:\\TabAlgos.java:39: La ligne excède 80 caractères (trouvé 87). [LineLength]
44. [ERROR] D:\\TabAlgos.java:39: Line has trailing spaces. [RegexpSingleline]
45. [ERROR] D:\\TabAlgos.java:41:54: Balise Javadoc @param manquante pour 'tab1'. [JavadocMethod]
46. [ERROR] D:\\TabAlgos.java:41:58: Il manque une espace après ','. [WhitespaceAfter]
47. [ERROR] D:\\TabAlgos.java:41:71: Balise Javadoc @param manquante pour 'tab2'. [JavadocMethod]
48. [ERROR] D:\\TabAlgos.java:42: La ligne excède 80 caractères (trouvé 92). [LineLength]
49. [ERROR] D:\\TabAlgos.java:42: Line has trailing spaces. [RegexpSingleline]
```

```

50. [ERROR] D:\\TabAlgos.java:45:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
51. [ERROR] D:\\TabAlgos.java:47:30: '5' devrait être défini comme une constante. [Magic
    Number]
52. [ERROR] D:\\TabAlgos.java:47:69: '3' devrait être défini comme une constante. [Magic
    Number]
53. [ERROR] D:\\TabAlgos.java:47:72: '4' devrait être défini comme une constante. [Magic
    Number]
54. [ERROR] D:\\TabAlgos.java:47:75: '5' devrait être défini comme une constante. [Magic
    Number]
55. [ERROR] D:\\TabAlgos.java:50:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
56. [ERROR] D:\\TabAlgos.java:55:19: Il manque une espace après 'catch'. [WhitespaceArou
    nd]
57. [ERROR] D:\\TabAlgos.java:60:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
58. [ERROR] D:\\TabAlgos.java:66:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
59. [ERROR] D:\\TabAlgos.java:68: La ligne excède 80 caractères (trouvé 105). [LineLengt
    h]
60. [ERROR] D:\\TabAlgos.java:68:68: '3' devrait être défini comme une constante. [Magic
    Number]
61. [ERROR] D:\\TabAlgos.java:68:71: '4' devrait être défini comme une constante. [Magic
    Number]
62. [ERROR] D:\\TabAlgos.java:68:74: '5' devrait être défini comme une constante. [Magic
    Number]
63. [ERROR] D:\\TabAlgos.java:68:95: '3' devrait être défini comme une constante. [Magic
    Number]
64. [ERROR] D:\\TabAlgos.java:68:98: '4' devrait être défini comme une constante. [Magic
    Number]
65. [ERROR] D:\\TabAlgos.java:68:101: '5' devrait être défini comme une constante. [Magi
    cNumber]
66. [ERROR] D:\\TabAlgos.java:71:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
67. [ERROR] D:\\TabAlgos.java:73: La ligne excède 80 caractères (trouvé 110). [LineLengt
    h]
68. [ERROR] D:\\TabAlgos.java:73:73: '3' devrait être défini comme une constante. [Magic
    Number]
69. [ERROR] D:\\TabAlgos.java:73:76: '4' devrait être défini comme une constante. [Magic
    Number]
70. [ERROR] D:\\TabAlgos.java:73:79: '5' devrait être défini comme une constante. [Magic
    Number]
71. [ERROR] D:\\TabAlgos.java:73:94: '5' devrait être défini comme une constante. [Magic
    Number]
72. [ERROR] D:\\TabAlgos.java:73:97: '4' devrait être défini comme une constante. [Magic
    Number]
73. [ERROR] D:\\TabAlgos.java:73:100: '3' devrait être défini comme une constante. [Magi
    cNumber]
74. Vérification terminée.
75. Checkstyle se termine par 72 erreurs.

```

2. Implémentation des méthodes en respectant les règles d'écriture contrôlées par l'outil "checkstyle".

Code source

```

1. /**
2.  * @return valeur la plus grande d'un tableau.

```

```

3.      */
4.      public static int plusGrand(final int[] tab) {
5.          int result = 0;
6.          if(tab.length == 1) {
7.              return tab[0];
8.          } else {
9.              for(int i = 0; i < tab.length; i++) {
10.                 if(tab[i] <= 0 ) {
11.                     result = tab[i];
12.                 } else {
13.                     if(tab[i] > result ) {
14.                         result = tab[i];
15.                     }
16.                 }
17.             }
18.         }
19.         return result;
20.     }
21.
22.     /**
23.      * @return moyenne des valeurs du tableau.
24.      * @throws IllegalArgumentException si tab est null ou vide.
25.      */
26.     public static double moyenne(final int[] tab) {
27.         double moyenne = 0;
28.         if(tab != null) {
29.             if ( tab.length > 0 ) {
30.                 int somme = 0;
31.                 for(int i = 0; i < tab.length; i++) {
32.                     somme += tab[i];
33.                 }
34.                 moyenne = (somme/tab.length);
35.                 return moyenne;
36.             }
37.         }
38.         throw new IllegalArgumentException("tab null ou vide");
39.     }
40.
41.     /**
42.      * Compare le contenu de 2 tableaux en tenant compte de l'ordre.
43.      * @return true si les 2 tableaux contiennent les mêmes éléments
44.      *         avec les mêmes nombres d'occurrences
45.      *         (avec les elements dans le meme ordre).
46.      */
47.     public static boolean egaux(final int[] tab1,final int[] tab2) {
48.         boolean result = false;
49.         if( (tab1 != null) && (tab2 != null) ) {
50.             if( tab1.length == tab2.length) {
51.                 for(int i = 0; i < tab1.length; i++) {
52.                     if( tab1[i] == tab2[i] ) {
53.                         result = true;
54.                     } else {
55.                         return false;
56.                     }
57.                 }
58.             }
59.         }
60.         return result;
61.     }
62.
63.     /**
64.      * Compare le contenu de 2 tableaux sans tenir compte de l'ordre.
65.      * @return true si les 2 tableaux contiennent les mêmes éléments
66.      *         avec les mêmes nombres d'occurrence
67.      *         (pas forcément dans le meme ordre).
68.      */

```

```

69.     public static boolean similaires(final int[] tab1,final int[] tab2) {
70.         boolean result = false;
71.         Map<Integer, Integer> mapOccurenceTab1 = new HashMap<Integer, Integer>();
72.         Map<Integer, Integer> mapOccurenceTab2 = new HashMap<Integer, Integer>();
73.         if( (tab1 != null) && (tab2 != null) ) {
74.             if( tab1.length == tab2.length) {
75.                 for(int i = 0; i < tab1.length; i++) {
76.                     if(!mapOccurenceTab1.containsKey(tab1[i])) {
77.                         int occurence = 1;
78.                         mapOccurenceTab1.put(tab1[i], occurence);
79.                     } else {
80.                         int occurence = mapOccurenceTab1.get(tab1[i]);
81.                         mapOccurenceTab1.replace(tab1[i], occurence++);
82.                     }
83.                 }
84.
85.                 for(int j = 0; j < tab2.length; j++) {
86.                     if(!mapOccurenceTab2.containsKey(tab2[j])) {
87.                         int occurence = 1;
88.                         mapOccurenceTab2.put(tab2[j], occurence);
89.                     } else {
90.                         int occurence = mapOccurenceTab2.get(tab2[j]);
91.                         mapOccurenceTab2.replace(tab2[j], occurence++);
92.                     }
93.                 }
94.             }
95.
96.             for (Map.Entry entry : mapOccurenceTab1.entrySet()) {
97.                 Integer nombre = (Integer) entry.getKey();
98.                 Integer occurence = (Integer) entry.getValue();
99.
100.                 if(mapOccurenceTab2.containsKey(nombre)) {
101.                     int nombreMap2 = mapOccurenceTab2.get(nombre);
102.
103.                     if(occurence == nombreMap2) {
104.                         result = true;
105.                     } else {
106.                         return false;
107.                     }
108.                 }
109.             }
110.         }
111.         return result;
112.     }

```

Rapport

1. Début de la vérification...
2. [ERROR] D:\\TabAlgos.java:1: Il manque un caractère NewLine à la fin du fichier. [NewlineAtEndOfFile]
3. [ERROR] D:\\TabAlgos.java:13: La ligne excède 80 caractères (trouvé 86). [LineLength]
4. [ERROR] D:\\TabAlgos.java:13: Line has trailing spaces. [RegexpSingleline]
5. [ERROR] D:\\TabAlgos.java:13:1: Le fichier contient des caractères de tabulation (ce n'est que la première occurrence). [FileTabCharacter]
6. [ERROR] D:\\TabAlgos.java:14: La ligne excède 80 caractères (trouvé 86). [LineLength]
7. [ERROR] D:\\TabAlgos.java:14: Line has trailing spaces. [RegexpSingleline]
8. [ERROR] D:\\TabAlgos.java:16:49: Balise Javadoc @param manquante pour 'tab'. [JavadocMethod]

9. [ERROR] D:\\TabAlgos.java:18:17: Il manque une espace après 'if'. [WhitespaceAround]

10. [ERROR] D:\\TabAlgos.java:19: Line has trailing spaces. [RegexpSingleline]

11. [ERROR] D:\\TabAlgos.java:21:25: Il manque une espace après 'for'. [WhitespaceAround]

12. [ERROR] D:\\TabAlgos.java:22:33: Il manque une espace après 'if'. [WhitespaceAround]

13. [ERROR] D:\\TabAlgos.java:22:48: Il y a une espace de trop avant ')'. [ParenPad]

14. [ERROR] D:\\TabAlgos.java:25:41: Il manque une espace après 'if'. [WhitespaceAround]

15. [ERROR] D:\\TabAlgos.java:25:60: Il y a une espace de trop avant ')'. [ParenPad]

16. [ERROR] D:\\TabAlgos.java:31: Line has trailing spaces. [RegexpSingleline]

17. [ERROR] D:\\TabAlgos.java:34: La ligne excède 80 caractères (trouvé 86). [LineLength]

18. [ERROR] D:\\TabAlgos.java:34: Line has trailing spaces. [RegexpSingleline]

19. [ERROR] D:\\TabAlgos.java:35: La ligne excède 80 caractères (trouvé 86). [LineLength]

20. [ERROR] D:\\TabAlgos.java:35: Line has trailing spaces. [RegexpSingleline]

21. [ERROR] D:\\TabAlgos.java:36: La ligne excède 80 caractères (trouvé 86). [LineLength]

22. [ERROR] D:\\TabAlgos.java:36: Line has trailing spaces. [RegexpSingleline]

23. [ERROR] D:\\TabAlgos.java:38:50: Balise Javadoc @param manquante pour 'tab'. [JavadocMethod]

24. [ERROR] D:\\TabAlgos.java:40:17: Il manque une espace après 'if'. [WhitespaceAround]

25. [ERROR] D:\\TabAlgos.java:41:25: Il manque une espace après 'if'. [WhitespaceAround]

26. [ERROR] D:\\TabAlgos.java:41:27: Il y a une espace de trop après '(' . [ParenPad]

27. [ERROR] D:\\TabAlgos.java:41:46: Il y a une espace de trop avant ')'. [ParenPad]

28. [ERROR] D:\\TabAlgos.java:43:33: Il manque une espace après 'for'. [WhitespaceAround]

29. [ERROR] D:\\TabAlgos.java:46:49: Il manque une espace après '/'. [WhitespaceAround]

30. [ERROR] D:\\TabAlgos.java:46:49: Il manque une espace avant '/'. [WhitespaceAround]

31. [ERROR] D:\\TabAlgos.java:53: La ligne excède 80 caractères (trouvé 86). [LineLength]

32. [ERROR] D:\\TabAlgos.java:53: Line has trailing spaces. [RegexpSingleline]

33. [ERROR] D:\\TabAlgos.java:54: La ligne excède 80 caractères (trouvé 86). [LineLength]

34. [ERROR] D:\\TabAlgos.java:54: Line has trailing spaces. [RegexpSingleline]

35. [ERROR] D:\\TabAlgos.java:55: La ligne excède 80 caractères (trouvé 89). [LineLength]

36. [ERROR] D:\\TabAlgos.java:55: Line has trailing spaces. [RegexpSingleline]

37. [ERROR] D:\\TabAlgos.java:56: La ligne excède 80 caractères (trouvé 87). [LineLength]

38. [ERROR] D:\\TabAlgos.java:56: Line has trailing spaces. [RegexpSingleline]

39. [ERROR] D:\\TabAlgos.java:57: La ligne excède 80 caractères (trouvé 86). [LineLength]

40. [ERROR] D:\\TabAlgos.java:57: Line has trailing spaces. [RegexpSingleline]

41. [ERROR] D:\\TabAlgos.java:59:49: Balise Javadoc @param manquante pour 'tab1'. [JavadocMethod]

42. [ERROR] D:\\TabAlgos.java:59:53: Il manque une espace après ','. [WhitespaceAfter]

43. [ERROR] D:\\TabAlgos.java:59:66: Balise Javadoc @param manquante pour 'tab2'. [JavadocMethod]

44. [ERROR] D:\\TabAlgos.java:61:17: Il manque une espace après 'if'. [WhitespaceAround]

45. [ERROR] D:\\TabAlgos.java:61:19: Il y a une espace de trop après '(' . [ParenPad]

46. [ERROR] D:\\TabAlgos.java:61:54: Il y a une espace de trop avant ')'. [ParenPad]

47. [ERROR] D:\\TabAlgos.java:62:25: Il manque une espace après 'if'. [WhitespaceAround]

48. [ERROR] D:\\TabAlgos.java:62:27: Il y a une espace de trop après '(' . [ParenPad]

49. [ERROR] D:\\TabAlgos.java:63:33: Il manque une espace après 'for'. [WhitespaceAround]

50. [ERROR] D:\\TabAlgos.java:64:41: Il manque une espace après 'if'. [WhitespaceAround]


```

51. [ERROR] D:\\TabAlgos.java:64:43: Il y a une espace de trop après '('. [ParenPad]
52. [ERROR] D:\\TabAlgos.java:64:65: Il y a une espace de trop avant ')'. [ParenPad]
53. [ERROR] D:\\TabAlgos.java:75: La ligne excède 80 caractères (trouvé 86). [LineLength]
54. [ERROR] D:\\TabAlgos.java:75: Line has trailing spaces. [RegexpSingleline]
55. [ERROR] D:\\TabAlgos.java:76: La ligne excède 80 caractères (trouvé 86). [LineLength]
56. [ERROR] D:\\TabAlgos.java:76: Line has trailing spaces. [RegexpSingleline]
57. [ERROR] D:\\TabAlgos.java:77: La ligne excède 80 caractères (trouvé 89). [LineLength]
58. [ERROR] D:\\TabAlgos.java:77: Line has trailing spaces. [RegexpSingleline]
59. [ERROR] D:\\TabAlgos.java:78: La ligne excède 80 caractères (trouvé 87). [LineLength]
60. [ERROR] D:\\TabAlgos.java:78: Line has trailing spaces. [RegexpSingleline]
61. [ERROR] D:\\TabAlgos.java:79: La ligne excède 80 caractères (trouvé 87). [LineLength]
62. [ERROR] D:\\TabAlgos.java:79: Line has trailing spaces. [RegexpSingleline]
63. [ERROR] D:\\TabAlgos.java:81:54: Balise Javadoc @param manquante pour 'tab1'. [JavadocMethod]
64. [ERROR] D:\\TabAlgos.java:81:58: Il manque une espace après ','. [WhitespaceAfter]
65. [ERROR] D:\\TabAlgos.java:81:71: Balise Javadoc @param manquante pour 'tab2'. [JavadocMethod]
66. [ERROR] D:\\TabAlgos.java:83: La ligne excède 80 caractères (trouvé 89). [LineLength]
67. [ERROR] D:\\TabAlgos.java:84: La ligne excède 80 caractères (trouvé 89). [LineLength]
68. [ERROR] D:\\TabAlgos.java:85:17: Il manque une espace après 'if'. [WhitespaceAround]
69. [ERROR] D:\\TabAlgos.java:85:19: Il y a une espace de trop après '('. [ParenPad]
70. [ERROR] D:\\TabAlgos.java:85:54: Il y a une espace de trop avant ')'. [ParenPad]
71. [ERROR] D:\\TabAlgos.java:86:25: Il manque une espace après 'if'. [WhitespaceAround]
72. [ERROR] D:\\TabAlgos.java:86:27: Il y a une espace de trop après '('. [ParenPad]
73. [ERROR] D:\\TabAlgos.java:87:33: Il manque une espace après 'for'. [WhitespaceAround]
74. [ERROR] D:\\TabAlgos.java:88: La ligne excède 80 caractères (trouvé 85). [LineLength]
75. [ERROR] D:\\TabAlgos.java:88: Line has trailing spaces. [RegexpSingleline]
76. [ERROR] D:\\TabAlgos.java:88:41: Il manque une espace après 'if'. [WhitespaceAround]
77. [ERROR] D:\\TabAlgos.java:90: La ligne excède 80 caractères (trouvé 89). [LineLength]
78. [ERROR] D:\\TabAlgos.java:92: La ligne excède 80 caractères (trouvé 95). [LineLength]
79. [ERROR] D:\\TabAlgos.java:93: La ligne excède 80 caractères (trouvé 95). [LineLength]
80. [ERROR] D:\\TabAlgos.java:97:33: Il manque une espace après 'for'. [WhitespaceAround]
81. [ERROR] D:\\TabAlgos.java:98: La ligne excède 80 caractères (trouvé 85). [LineLength]
82. [ERROR] D:\\TabAlgos.java:98: Line has trailing spaces. [RegexpSingleline]
83. [ERROR] D:\\TabAlgos.java:98:41: Il manque une espace après 'if'. [WhitespaceAround]
84. [ERROR] D:\\TabAlgos.java:100: La ligne excède 80 caractères (trouvé 89). [LineLength]
85. [ERROR] D:\\TabAlgos.java:102: La ligne excède 80 caractères (trouvé 95). [LineLength]
86. [ERROR] D:\\TabAlgos.java:103: La ligne excède 80 caractères (trouvé 95). [LineLength]
87. [ERROR] D:\\TabAlgos.java:112:33: Il manque une espace après 'if'. [WhitespaceAround]
88. [ERROR] D:\\TabAlgos.java:113: La ligne excède 80 caractères (trouvé 87). [LineLength]
89. [ERROR] D:\\TabAlgos.java:115:41: Il manque une espace après 'if'. [WhitespaceAround]

```

```

90. [ERROR] D:\\TabAlgos.java:126:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
91. [ERROR] D:\\TabAlgos.java:128:30: '5' devrait être défini comme une constante. [MagicNumber]
92. [ERROR] D:\\TabAlgos.java:128:69: '3' devrait être défini comme une constante. [MagicNumber]
93. [ERROR] D:\\TabAlgos.java:128:72: '4' devrait être défini comme une constante. [MagicNumber]
94. [ERROR] D:\\TabAlgos.java:128:75: '5' devrait être défini comme une constante. [MagicNumber]
95. [ERROR] D:\\TabAlgos.java:131:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
96. [ERROR] D:\\TabAlgos.java:136:19: Il manque une espace après 'catch'. [WhitespaceAround]
97. [ERROR] D:\\TabAlgos.java:141:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
98. [ERROR] D:\\TabAlgos.java:147:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
99. [ERROR] D:\\TabAlgos.java:149: La ligne excède 80 caractères (trouvé 105). [LineLength]
100. [ERROR] D:\\TabAlgos.java:149:68: '3' devrait être défini comme une constante. [MagicNumber]
101. [ERROR] D:\\TabAlgos.java:149:71: '4' devrait être défini comme une constante. [MagicNumber]
102. [ERROR] D:\\TabAlgos.java:149:74: '5' devrait être défini comme une constante. [MagicNumber]
103. [ERROR] D:\\TabAlgos.java:149:95: '3' devrait être défini comme une constante. [MagicNumber]
104. [ERROR] D:\\TabAlgos.java:149:98: '4' devrait être défini comme une constante. [MagicNumber]
105. [ERROR] D:\\TabAlgos.java:149:101: '5' devrait être défini comme une constante. [MagicNumber]
106. [ERROR] D:\\TabAlgos.java:152:9: Commentaire Javadoc manquant. [MissingJavadocMethod]
107. [ERROR] D:\\TabAlgos.java:154: La ligne excède 80 caractères (trouvé 110). [LineLength]
108. [ERROR] D:\\TabAlgos.java:154:73: '3' devrait être défini comme une constante. [MagicNumber]
109. [ERROR] D:\\TabAlgos.java:154:76: '4' devrait être défini comme une constante. [MagicNumber]
110. [ERROR] D:\\TabAlgos.java:154:79: '5' devrait être défini comme une constante. [MagicNumber]
111. [ERROR] D:\\TabAlgos.java:154:94: '5' devrait être défini comme une constante. [MagicNumber]
112. [ERROR] D:\\TabAlgos.java:154:97: '4' devrait être défini comme une constante. [MagicNumber]
113. [ERROR] D:\\TabAlgos.java:154:100: '3' devrait être défini comme une constante. [MagicNumber]
114. Vérification terminée.
115. Checkstyle se termine par 112 erreurs.

```

3. Validité des tests

Source

```

1. package fr.u_picardie.active.c306;
2.
3. import static org.junit.jupiter.api.Assertions.assertEquals;
4. import static org.junit.jupiter.api.Assertions.fail;
5. import java.util.HashMap;

```

```

6. import java.util.Map;
7. import org.junit.jupiter.api.Test;
8.
9. public final class TabAlgos {
10.    /**
11.     * Constante pour représenter le chiffre 3 imposée par checkstyle.
12.     */
13.     static final int VAL_3 = 3;
14.    /**
15.     * Constante pour représenter le chiffre 4 imposée par checkstyle.
16.     */
17.     static final int VAL_4 = 4;
18.    /**
19.     * Constante pour représenter le chiffre 5 imposée par checkstyle.
20.     */
21.     static final int VAL_5 = 5;
22.    /**
23.     * Constante pour représenter la marge.
24.     */
25.     static final double MARGE = 0.1;
26.    /**
27.     * @return valeur la plus grande d'un tableau.
28.     * @param tab tableau d'entiers
29.     */
30.     public static int plusGrand(final int[] tab) {
31.         int result = 0;
32.         if (tab.length == 1) {
33.             return tab[0];
34.         } else {
35.             for (int i = 0; i < tab.length; i++) {
36.                 if (tab[i] <= 0) {
37.                     result = tab[i];
38.                 } else {
39.                     if (tab[i] > result) {
40.                         result = tab[i];
41.                     }
42.                 }
43.             }
44.         }
45.         return result;
46.     }
47.    /**
48.     * @return moyenne des valeurs du tableau.
49.     * @throws IllegalArgumentException si tab est null ou vide.
50.     * @param tab tableau d'entiers
51.     */
52.     public static double moyenne(final int[] tab) {
53.         double moyenne = 0;
54.         if (tab != null) {
55.             if ((tab.length > 0)) {
56.                 int somme = 0;
57.                 for (int i = 0; i < tab.length; i++) {
58.                     somme += tab[i];
59.                 }
60.                 moyenne = (somme / tab.length);
61.                 return moyenne;
62.             }
63.         }
64.         throw new IllegalArgumentException("tab null ou vide");
65.     }
66.
67.    /**
68.     * Compare le contenu de 2 tableaux en tenant compte de l'ordre.
69.     * @return true si les 2 tableaux contiennent les mêmes éléments
70.     *         avec les mêmes nombres d'occurrences
71.     *         (avec les éléments dans le même ordre).

```

```

72.      *      @param tab1 premier tableau d'entiers
73.      *      @param tab2 second tableau d'entiers
74.      */
75.      public static boolean egaux(final int[] tab1, final int[] tab2) {
76.          boolean result = false;
77.          if ((tab1 != null) && (tab2 != null)) {
78.              if (tab1.length == tab2.length) {
79.                  for (int i = 0; i < tab1.length; i++) {
80.                      if (tab1[i] == tab2[i]) {
81.                          result = true;
82.                      } else {
83.                          return false;
84.                      }
85.                  }
86.              }
87.          }
88.          return result;
89.      }
90.
91.      /**
92.       * Compare le contenu de 2 tableaux sans tenir compte de l'ordre.
93.       * @return true si les 2 tableaux contiennent les mêmes éléments
94.       * avec les mêmes nombres d'occurrence
95.       * (pas forcément dans le même ordre).
96.       * @param tab1 premier tableau d'entiers
97.       * @param tab2 second tableau d'entiers
98.       */
99.      public static boolean similaires(final int[] tab1, final int[] tab2) {
100.          boolean result = false;
101.          Map<Integer, Integer> mapOccurrenceTab1 =
102.              new HashMap<Integer, Integer>();
103.          Map<Integer, Integer> mapOccurrenceTab2 =
104.              new HashMap<Integer, Integer>();
105.          if ((tab1 != null) && (tab2 != null)) {
106.              if (tab1.length == tab2.length) {
107.                  for (int i = 0; i < tab1.length; i++) {
108.                      if (!mapOccurrenceTab1
109.                          .containsKey(tab1[i])) {
110.                          int occurrence = 1;
111.                          mapOccurrenceTab1
112.                              .put(tab1[i], occurrence);
113.                      } else {
114.                          int occurrence = mapOccurrenceTab1
115.                              .get(tab1[i]);
116.                          mapOccurrenceTab1
117.                              .replace(tab1[i], occurrence++);
118.                      }
119.                  }
120.
121.                  for (int j = 0; j < tab2.length; j++) {
122.                      if (!mapOccurrenceTab2
123.                          .containsKey(tab2[j])) {
124.                          int occurrence = 1;
125.                          mapOccurrenceTab2
126.                              .put(tab2[j], occurrence);
127.                      } else {
128.                          int occurrence = mapOccurrenceTab2
129.                              .get(tab2[j]);
130.                          mapOccurrenceTab2
131.                              .replace(tab2[j], occurrence++);
132.                      }
133.                  }
134.              }
135.
136.              for (Map.Entry entry : mapOccurrenceTab1.entrySet()) {
137.                  Integer nombre = (Integer) entry.getKey();

```

```

138.             Integer occurrence = (Integer) entry.getValue();
139.
140.             if (mapOccurenceTab2
141.                 .containsKey(nombre)) {
142.                 int nombreMap2 = mapOccurenceTab2
143.                     .get(nombre);
144.
145.                 if (occurrence == nombreMap2) {
146.                     result = true;
147.                 } else {
148.                     return false;
149.                 }
150.             }
151.         }
152.     }
153.     return result;
154. }
155.
156. /**
157.  * Test de la fonction plusGrand().
158.  */
159. @Test
160. public void testPlusGrand() {
161.     int[] tableau = new int[] {2, 2, 2, 2};
162.     assertEquals(VAL_5, TabAlgos.plusGrand(tableau));
163. }
164.
165. /**
166.  * Test de la fonction moyenne().
167.  */
168. @Test
169. public void testMoyenne() {
170.     try {
171.         TabAlgos.moyenne(null);
172.         fail("tab est null");
173.     } catch (IllegalArgumentException e) {
174.
175.     }
176. }
177.
178. /**
179.  * Test de la fonction moyenne cas nominal.
180.  */
181. @SuppressWarnings("deprecation")
182. @Test
183. public void testMoyenneGood() {
184.     assertEquals(2, TabAlgos.moyenne(
185.         new int[] {2, 2, 2, 2}), MARGE);
186. }
187.
188. /**
189.  * Test de la fonction egaux().
190.  */
191. @Test
192. public void testEgaux() {
193.     assertEquals(true, TabAlgos.egaux(
194.         new int[] {1, 2, VAL_3, VAL_4, VAL_5},
195.         new int[] {1, 2, VAL_3, VAL_4, VAL_5}));
196. }
197.
198. /**
199.  * Test de la fonction similaires().
200.  */
201. @Test
202. public void testSimilaires() {
203.     assertEquals(true, TabAlgos.similaires(

```

```

204.         new int[] {1, 2, VAL_3, VAL_4, VAL_5},
205.         new int[] {VAL_5, VAL_4, VAL_3, 2, 1}));
206.     }
207. }

```

Rapports de checkstyle

```

1. Début de la vérification...
2. [ERROR] D:\\TabAlgos.java:11:1: Le fichier contient des caractères de tabulation (ce
   n'est que la première occurrence). [FileTabCharacter]
3. Vérification terminée.
4. Checkstyle se termine par 1 erreurs.

```

Cependant, rien n'explique la présence de cette erreur.

Exercice 3 : Dépôt sur serveur de Versionning

Lien du dépôt : <https://github.com/abakachiali/activite1>