



Activité 1 C306 - Ingénierie du logiciel

Taoufiq BENMESSAOUD

Exercice 1

1. Le code ne compile pas parce que

- On ne peut assigner une valeur à un variable final
- Il faut un return dans la méthode getT()

2. Corrigez le code pour qu'il compile.

```
/**
 * Ma class d'exemple.
 */
public class ExempleOld {
    private String t;
    private int v;

    public ExempleOld(String t) {
        t = t;
    }

    /**
     * @return la valeur de v
     */
    public int getV(){
        return this.v;
    }

    /**
     * modifie la valeur de v
     */
    public final void setV(int v){
        this.v=v;
    }

    /**
     * @return t si v est positif, null sinon
     */
    public final String getT()
    {
        if(v>0) return t;
        return null;
    }
}
```

3. Donnez le rapport de "checkstyle" appliqué à votre code :

Exemple.java:1: File does not end with a newline. [NewlineAtEndOfFile]
Exemple.java:1: Missing package-info.java file. [JavadocPackage]
Exemple.java:4:21: '{' is not preceded with whitespace. [WhitespaceAround]
Exemple.java:5:5: Missing a [javadoc](#) comment. [JavadocVariable]
Exemple.java:6:5: Missing a [javadoc](#) comment. [JavadocVariable]
Exemple.java:7: Line has trailing spaces. [RegexpSingleline]
Exemple.java:8:5: Missing a [javadoc](#) comment. [MissingJavadocMethod]
Exemple.java:8:20: Parameter t should be final. [FinalParameters]
Exemple.java:8:27: 't' hides a field. [HiddenField]
Exemple.java:15:22: '{' is not preceded with whitespace. [WhitespaceAround]
Exemple.java:19: First sentence should end with a period. [JavadocStyle]
Exemple.java:22:28: Parameter v should be final. [FinalParameters]
Exemple.java:22:32: 'v' hides a field. [HiddenField]

Exemple.java:22:32: Expected @param tag for 'v'. [JavadocMethod]
 Exemple.java:22:34: '{' is not preceded with whitespace. [WhitespaceAround]
 Exemple.java:23:15: '=' is not followed by whitespace. [WhitespaceAround]
 Exemple.java:23:15: '=' is not preceded with whitespace. [WhitespaceAround]
 Exemple.java:30:5: '{' at column 5 should be on the previous line. [LeftCurly]
 Exemple.java:31:9: 'if' construct must use '{}'s. [NeedBraces]
 Exemple.java:31:9: 'if' is not followed by whitespace. [WhitespaceAfter]
 Exemple.java:31:13: '>' is not followed by whitespace. [WhitespaceAround]
 Exemple.java:31:13: '>' is not preceded with whitespace. [WhitespaceAround]
 Exemple.java:31:16: 'return' is not preceded with whitespace. [WhitespaceAround]

4. Proposez une réécriture en tenant compte des problèmes soulevés par "checkstyle".

```
/**
 * main package.
 *
 * @author taoufiq.benmessaoud@etud.u-picardie.fr (Taoufiq BENMESSAoud)
 * @version 1.0
 */
package activite.exercice1;

/**
 * Class Exemple.java.
 */
public class Exemple {
    /**
     * variable t.
     */
    private String t;

    /**
     * variable v.
     */
    private int v;

    /**
     * @param newT
     */
    public Exemple(final String newT) {
        this.t = newT;
    }

    /**
     * @return int
     */
    public int getV() {
        return this.v;
    }

    /**
     * @param newV
     */
    public void setV(final int newV) {
        this.v = newV;
    }

    /**
     * @return t si v est positif, null sinon
     */
    public final String getT() {
        if (v > 0) {
            return t;
        }

        return null;
    }
}
```

5. Donnez le rapport de "checkstyle" appliqué au code réécrit. Audit done.

Exercice 2

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

```
import static org.junit.jupiter.api.Assertions.assertEquals;
import activite.exercice2.TabAlgos;
import org.junit.jupiter.api.Test;

/**
 * class pour tester les methodes de la classe TabAlgos.
 */
public class TestTabAlgos {

    /**
     * activite.exercice2.test de la methode plusGrand.
     */
    @Test
    public void testPlusGrand() {
        final int[] tab = {5, 10, 15};
        final int result = 15;
        assertEquals(result, TabAlgos.plusGrand(tab));
    }

    /**
     * activite.exercice2.test de la methode moyenne.
     */
    @Test
    public void testMoyenne() {
        final int[] tab = {5, 10, 15};
        final int result = 10;
        assertEquals(result, TabAlgos.moyenne(tab));
    }

    /**
     * activite.exercice2.test de la methode egaux.
     */
    @Test
    public void testEgaux() {
        final int[] tab1 = {5, 10, 15};
        final int[] tab2 = {5, 10, 15};
        assertEquals(true, TabAlgos.egaux(tab1, tab2));

        final int[] tab3 = {57, 10, 15};
        final int[] tab4 = {5, 10, 15};
        assertEquals(false, TabAlgos.egaux(tab3, tab4));
    }

    /**
     * activite.exercice2.test de la methode similaire.
     */
    @Test
    public void testSimilaires() {
        final int[] tab1 = {5, 10, 15};
        final int[] tab2 = {10, 5, 15};
        assertEquals(true, TabAlgos.similaires(tab1, tab2));

        final int[] tab3 = {57, 10, 15};
        final int[] tab4 = {5, 10, 15};
        assertEquals(false, TabAlgos.similaires(tab3, tab4));
    }
}
```

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

```
/**
 * exercice2 package.
 *
 * @author taoufiq.benmessaoud@etud.u-picardie.fr (Taoufiq BENMESSAOUD)
 * @version 1.0
 */
package activite.exercice2;

/**
 * class qui contient des fonctions pour les tableaux.
 */
public final class TabAlgos {
    private TabAlgos() {
    }

    /**
     * la valeur la plus grande d'un tableau.
     * @param tab
     * @return int
     */
    public static int plusGrand(final int[] tab) {
        int lePlusGrand = tab[0];
        for (int i = 0; i < tab.length; i++) {
            if (tab[i] > lePlusGrand) {
                lePlusGrand = tab[i];
            }
        }
        return lePlusGrand;
    }

    /**
     * moyenne des valeurs du tableau.
     * @param tab
     * @return float
     */
    public static float moyenne(final int[] tab) {
        int somme = 0;
        for (int i = 0; i < tab.length; i++) {
            somme += tab[i];
        }
        return somme / tab.length;
    }

    /**
     * Compare le contenu de 2 tableaux en tenant compte de l'ordre.
     * true si les 2 tableaux contiennent les mêmes éléments
     * avec les mêmes nombres d'occurrences
     * (avec les elements dans le meme ordre).
     * @param tab1
     * @param tab2
     * @return boolean
     */
    public static boolean egaux(final int[] tab1, final int[] tab2) {
        boolean siEgaux = true;
        if (tab1.length != tab2.length) {
            siEgaux = false;
        } else {
            for (int i = 0; i < tab1.length; i++) {
                if (tab1[i] != tab2[i]) {
                    siEgaux = false;
                    break;
                }
            }
        }
        return siEgaux;
    }
}
```

```

/**
 * Compare le contenu de 2 tableaux sans tenir compte de l'ordre.
 * true si les 2 tableaux contiennent les mêmes éléments
 * avec les mêmes nombres d'occurrence
 * (pas forcément dans le meme ordre).
 * @param tab1
 * @param tab2
 * @return boolean
 */
public static boolean similaires(final int[] tab1, final int[] tab2) {
    return egaux(orderTab(tab1), orderTab(tab2));
}

/**
 * trie d'un tableau du plus petit au plus grand.
 * @param tab
 * @return int[]
 */
private static int[] orderTab(final int[] tab) {
    for (int i = 0; i < tab.length; i++) {
        for (int j = 0; j < tab.length; j++) {
            if (tab[i] < tab[j]) {
                int temp = tab[i];
                tab[i] = tab[j];
                tab[j] = temp;
            }
        }
    }

    return tab;
}
}

```

3. Vérifiez la validité des tests avec le code implémenté.

TestTabAlgos		30 ms
testEgaux()	passed	26 ms
testPlusGrand()	passed	1 ms
testSimilaires()	passed	1 ms
testMoyenne()	passed	2 ms

Exercice 3

1. Clé publique :

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAQgQDemUcxvHyOXXNsiGwMjwAJWxpbHMglUzFARRtfz
RB+PvcxmNuCkitEzuat9OMW1zfLcTeXbxfPhg3lqZeQB5QbVWf1vO6G705BM2bnABKAkyF7OF
2u1gbjQj0jQxLeVSh0dyq/T/TvFGPgZRTvFl0X9KI9c1zux8qH3A8jRT0yeC3t9r4p6DJ1zQvs/5I
WtQoIRAES64Lbup6mwYfEDbBsLvWjahPRJgBrZymjNf8z7ch3J35xafCNH4Dhv7EWFy/zlmG8
QPGM/tvIkFiV0XVf4vUvB3e57uQWpN6of8+Sf4B7Abb0pAkA6uDq0YvSvEc5XjLeRJyT+z6g9lZ
saxc6WgMp2B9r+F2JhrP6k2De89MnoyFmLmjlQ/tyZ3+bu2mkOd/z/MWh+OKA2Etduv76NrZ
e5/tPyR2gAc1wRBISpCcqa9of/tnn/2HrI34N9P+3L3rCqMdhwVxskGd/siD8TFm3WTFkLsQy
d/VDmq1k5oSq1Slj8uYXpTHk7sPgQM= taoufiq@taoufiq