**Activité 1 (1-2020) - C306 - Ingénierie du logiciel (ISI\_05)**

**Lien vers le repository :** <https://github.com/marckodjo/activite1>

**Exercice 1 :**

Ecriture de code

/\*\*

\* Ma class d'exemple.

\*/

public class Exemple {

private final String t;

private final int v;

public Exemple(String t) {

t = t;

}

/\*\*

\* @return la valeur de v

\*/

public int getV() {

return this.v;

}

public final void setV(int v){this.v =v;}

/\*\*

\* @return t si v est positif

\*/

public final String getT() {

if ( v > 0) return t;

}

}

1. Le rapport de « checkstyle » du code ci-dessus est le suivant

D├®but de la v├®rification...

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:1: Il manque un caract├¿re NewLine ├á la fin du fichier. [NewlineAtEndOfFile]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:5:2: Commentaire Javadoc manquant. [JavadocVariable]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:6:2: Commentaire Javadoc manquant. [JavadocVariable]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:7:1: Commentaire Javadoc manquant. [MissingJavadocMethod]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:7:16: Le param├¿tre t devrait ├¬tre final. [FinalParameters]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:7:23: 't' masque un attribut. [HiddenField]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:24: Le param├¿tre v devrait ├¬tre final. [FinalParameters]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:28: 'v' masque un attribut. [HiddenField]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:30: '{' ├á la colonne 30 devrait avoir un saut de ligne apr├¿s. [LeftCurly]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:30: Il manque une espace apr├¿s '{'. [WhitespaceAround]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:30: Il manque une espace avant '{'. [WhitespaceAround]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:38: Il manque une espace apr├¿s '='. [WhitespaceAround]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:40: Il manque une espace apr├¿s ';'. [WhitespaceAfter]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:16:41: Il manque une espace avant '}'. [WhitespaceAround]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:21: L'instruction 'if' devrait utiliser des accolades ('{' et '}'). [NeedBraces]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:21:4: Il y a une espace de trop apr├¿s '('. [ParenPad]

V├®rification termin├®e.

Checkstyle se termine par 16 erreurs.

1. Code après correction de certaines anomalies soulevées par checkstyle

/\*\*

\* Ma class d'exemple.

\*/

**public** **class** Exemple {

**private** **final** String t;

**private** **final** **int** v;

/\*\*

\***@param** t entier

\*/

**public** Exemple(String t) {

t = t;

}

/\*\*

\* **@return** la valeur de v

\*/

**public** **int** getV() {

**return** **this**.v;

}

/\*\*

\***@param** v int

\*/

**public** **final** **void** setV(**int** v) {

**this**.v = v;

}

/\*\*

\* **@return** t si v est positif

\*/

**public** **final** String getT() {

**if** (v > 0) {

**return** t;

}

}

}

1. Rapport de "checkstyle" appliqué au code réécrit :

D├®but de la v├®rification...

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:5:2: Commentaire Javadoc manquant. [JavadocVariable]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:6:2: Commentaire Javadoc manquant. [JavadocVariable]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:10:16: Le param├¿tre t devrait ├¬tre final. [FinalParameters]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:10:23: 't' masque un attribut. [HiddenField]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:22:24: Le param├¿tre v devrait ├¬tre final. [FinalParameters]

[ERROR] D:\MIAGE\Cours\C306 - Ing├®nierie du logiciel (ISI\_05)\C306-Exo\src\Exemple.java:22:28: 'v' masque un attribut. [HiddenField]

V├®rification termin├®e.

Checkstyle se termine par 6 erreurs.

* Justification :

Pour les attributs nous n’avons pas besoin d’annotations.

Les variables passées en paramètre sont les mêmes que les attributs de la classe Exemple.

**Exercice 2**

1. Tests unitaires permettant de tester les méthodes

**class** TestTabAlgos {

@Test

**public** **void** testPlusGrand() {

**int** tableauEntier[] = {0,5,8,12,2,30};

*assertEquals*(30, TabAlgos.*plusGrand*(tableauEntier));

}

@Test

**public** **void** testCalculMoyenne() {

**int** tableauEntier[] = {10,5,5,5};

**try** {

*assertEquals*(6.0,TabAlgos.*moyenne*(tableauEntier));

} **catch** (Exception e) {

// **TODO**: handle exception

}

}

@Test

**public** **void** testEgaux() {

**int** tableauEntier1[] = {0,5,8,12,2,30};

**int** tableauEntier2[] = {0,5,8,12,2,30};

*assertTrue*(TabAlgos.*egaux*(tableauEntier1, tableauEntier2));

}

@Test

**public** **void** testSimilaire() {

**int** tableauEntier1[] = {0,5,8,12,2,30};

**int** tableauEntier2[] = {12,5,30,0,2,8};

*assertTrue*(TabAlgos.*similaires*(tableauEntier1, tableauEntier2));

}

}

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

**package** testunitaire;

**public** **final** **class** TabAlgos {

**protected** TabAlgos() {

}

/\*\*

\***@param** tab est un tableau d'entier

\***@return** valeur la plus grande d'un tableau

\*/

**public** **static** **int** plusGrand(**final** **int**[] tab) {

//**TODO**

**int** value = tab[0];

**for** (**int** i = 0; i < tab.length; i++) {

**if** (value < tab[i]) {

value = tab[i];

}

}

**return** value;

}

/\*\*

\***@param** tab est un tableau de double

\***@return** moyenne des valeurs du tableau

\***@throw** IllegalArgumentException si tab et null ou vide

\*\*/

**public** **static** **double** moyenne(**final** **int**[] tab) {

//**TODO**

**int** value = 0;

**if** (tab.length < 0 | tab == **null**) {

**throw** **new** IllegalArgumentException("Le tableau est vide ou null");

} **else** {

**for** (**int** i = 0; i < tab.length; i++) {

value += tab[i];

}

}

**return** value / tab.length;

}

/\*\*

\***@param** tab1 est un tableau d'entier

\***@param** tab2 est un tableau d'entier

\*Compare le contenu de 2 tableaux en tenant compte de l'ordre.

\***@return** true si les 2 tableaux contiennent les mêmes éléments avec

\*les mêmes nombres d'occurences (avec les elements dans le meme ordre)

\*\*/

**public** **static** **boolean** egaux(**final** **int**[] tab1, **final** **int**[] tab2) {

//**TODO**

**boolean** b = **false**;

**int** cpt = 0;

**for** (**int** j = 0; j < tab1.length; j++) {

**if** (tab1[j] == tab2[j]) {

cpt++;

}

}

**if** (cpt == tab1.length) {

b = **true**;

}

**return** b;

}

/\*\*

\***@param** tab1 est un tableau d'entier

\***@param** tab2 est un tableau d'entier

\*Compare le contenu de 2 tableaux sans tenir compte de l'ordre.

\***@return** true si les 2 tableaux contiennent les mêmes éléments

\*\*avec les mêmes nombres d'occurrence

\*(pas forcément dans le meme ordre)

\*\*/

**public** **static** **boolean** similaires(**final** **int**[] tab1, **final** **int**[] tab2) {

//**TODO**

**boolean** b = **false**;

**int** cpt = 0;

**for** (**int** i = 0; i < tab1.length; i++) {

b = **false**;

**for** (**int** j = 0; j < tab1.length; j++) {

**if** (tab1[i] == tab2[j]) {

cpt++;

}

}

}

**if** (cpt == tab1.length) {

b = **true**;

}

**return** b;

}

}

1. Rapport de checkstyle de la classe TestTabAlgos.java

PS D:\MIAGE\Cours\C306 - Ingénierie du logiciel (ISI\_05)\Projets\C306-Devoir 1\src\test\java\testunitaire> java -jar checkstyle-8.30-all.jar -c sun\_checks.xml TestTabAlgos.java D├®but de la v├®rification...

V├®rification termin├®e.

Rapport checkstyle de la classe TabAlgos.java

PS D:\MIAGE\Cours\C306 - Ingénierie du logiciel (ISI\_05)\Projets\C306-Devoir 1\src\main\java\testunitaire> java -jar checkstyle-8.30-all.jar -c sun\_checks.xml TabAlgos.java D├®but de la v├®rification...

V├®rification termin├®e.

**Exercice 3**

Clé publique :

ssh-rsa  Marc AMEDONOU@hp