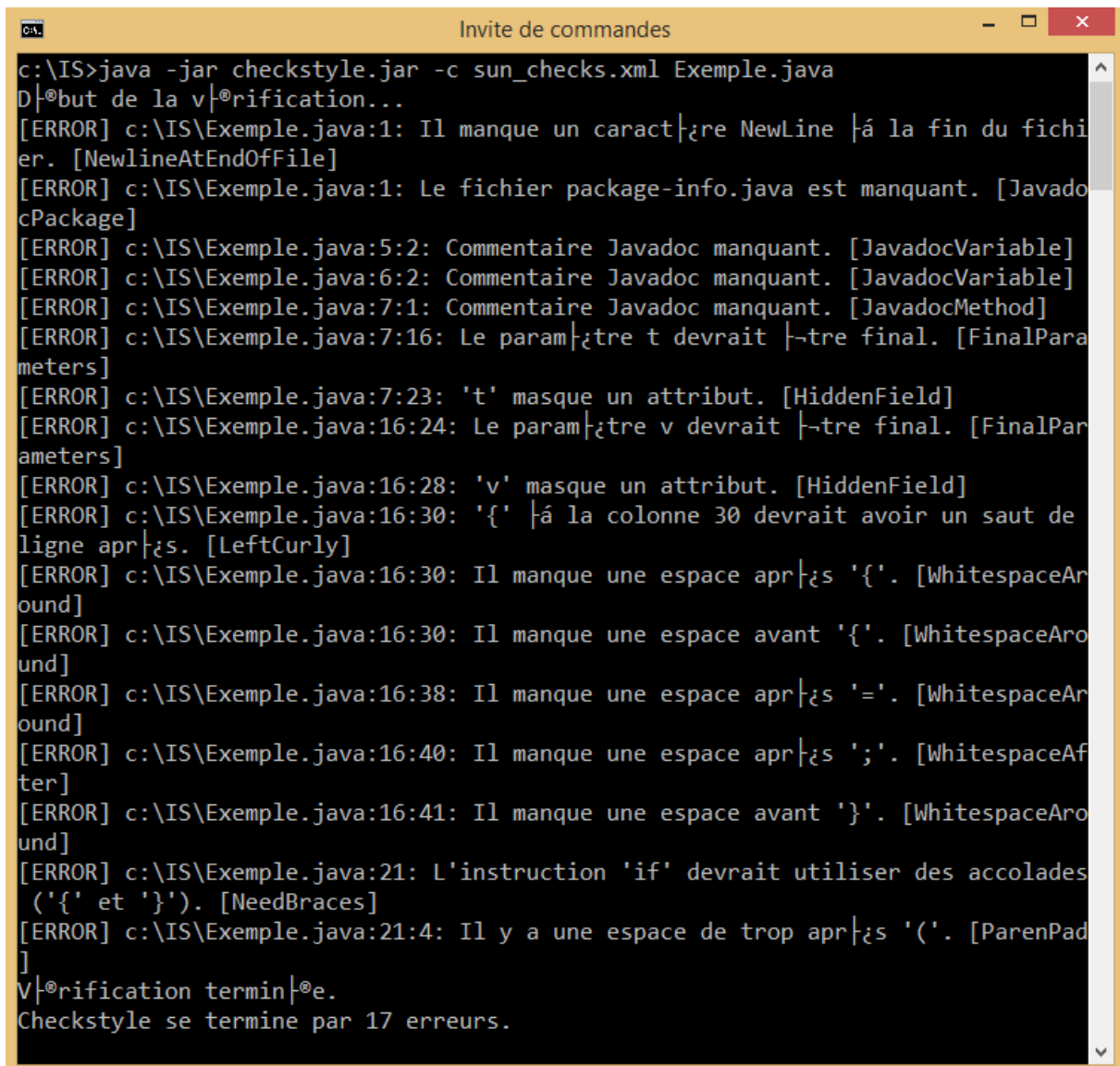


Exercice 1 : Ecriture de Code.

1- Rapport de "checkstyle" appliqué au code du fichier joint Exemple.java :



```
c:\IS>java -jar checkstyle.jar -c sun_checks.xml Exemple.java
Démarrage de la vérification...
[ERROR] c:\IS\Exemple.java:1: Il manque un caractère NewLine à la fin du fichier. [NewlineAtEndOfFile]
[ERROR] c:\IS\Exemple.java:1: Le fichier package-info.java est manquant. [JavadocPackage]
[ERROR] c:\IS\Exemple.java:5:2: Commentaire Javadoc manquant. [JavadocVariable]
[ERROR] c:\IS\Exemple.java:6:2: Commentaire Javadoc manquant. [JavadocVariable]
[ERROR] c:\IS\Exemple.java:7:1: Commentaire Javadoc manquant. [JavadocMethod]
[ERROR] c:\IS\Exemple.java:7:16: Le paramètre t devrait être final. [FinalParameters]
[ERROR] c:\IS\Exemple.java:7:23: 't' masque un attribut. [HiddenField]
[ERROR] c:\IS\Exemple.java:16:24: Le paramètre v devrait être final. [FinalParameters]
[ERROR] c:\IS\Exemple.java:16:28: 'v' masque un attribut. [HiddenField]
[ERROR] c:\IS\Exemple.java:16:30: '{' à la colonne 30 devrait avoir un saut de ligne après. [LeftCurly]
[ERROR] c:\IS\Exemple.java:16:30: Il manque une espace après '{'. [WhitespaceAround]
[ERROR] c:\IS\Exemple.java:16:30: Il manque une espace avant '{'. [WhitespaceAround]
[ERROR] c:\IS\Exemple.java:16:38: Il manque une espace après '='. [WhitespaceAround]
[ERROR] c:\IS\Exemple.java:16:40: Il manque une espace après ';'. [WhitespaceAfter]
[ERROR] c:\IS\Exemple.java:16:41: Il manque une espace avant '}'. [WhitespaceAround]
[ERROR] c:\IS\Exemple.java:21: L'instruction 'if' devrait utiliser des accolades ('{' et '}'). [NeedBraces]
[ERROR] c:\IS\Exemple.java:21:4: Il y a une espace de trop après '('. [ParenPad]
Vérification terminée.
Checkstyle se termine par 17 erreurs.
```

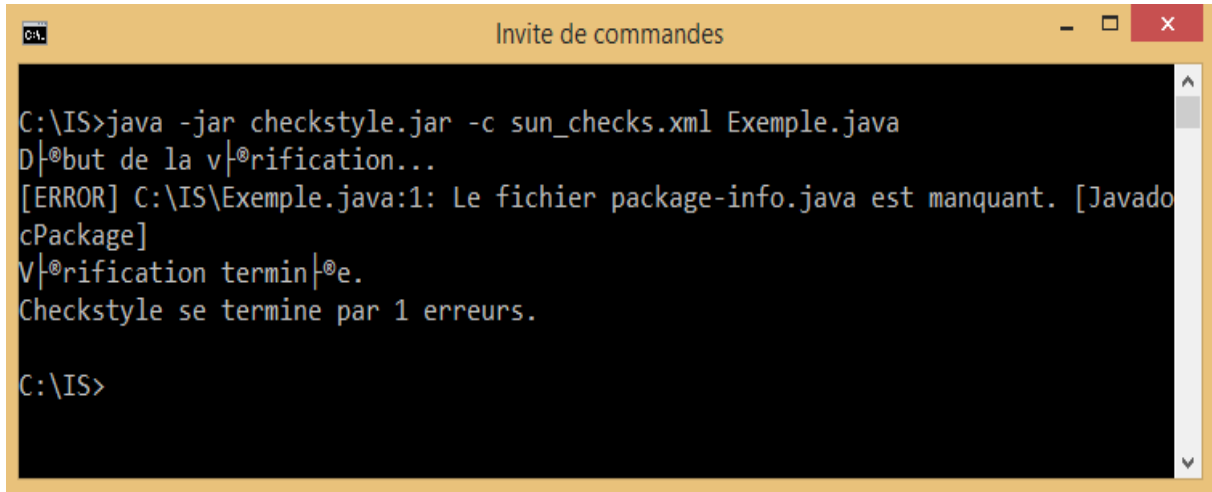
Fig. 01 : Capture d'écran du rapport de checkstyle appliqué au code.

L'audit du fichier s'est terminé avec 17 erreurs.

2- Ci-dessous le nouveau code réécrit :

```
/**
 * Ma class d'exemple.
 */
public class Exemple {
    /**
     * t est une chaine de caractère.
     */
    private final String t;
    /**
     * v est un entier.
     */
    private final int v;
    /**
     * Constructeur.
     * @param a est une chaine de caractère
     */
    public Exemple(final String a) {
        t = a;
    }
    /**
     * getter.
     * @return la valeur de v
     */
    public int getV() {
        return this.v;
    }
    /**
     * setter.
     * @param b est un entier
     */
    public final void setV(final int b) {
        this.v = b;
    }
    /**
     * @return t si v est positif.
     */
    public final String getT() {
        if (v > 0) {
            return t;
        }
    }
}
```

3- Après la réécriture du code, l'exécution du "checkstyle" s'est terminée avec une erreur que nous justifierons dans les lignes qui suivent.



```
C:\IS>java -jar checkstyle.jar -c sun_checks.xml Exemple.java
D|@but de la v|rification...
[ERROR] C:\IS\Exemple.java:1: Le fichier package-info.java est manquant. [Javadoc
cPackage]
V|rification termin|e.
Checkstyle se termine par 1 erreurs.

C:\IS>
```

Fig. 02 : Capture d'écran du rapport de checkstyle appliqué au code proposé ci-dessus.

- [ERROR] c:\IS\Exemple.java:1: Le fichier package-info.java est manquant.
[JavadocPackage]

L'erreur est due au fait que le fichier javadoc qui doit contenir toute la description de la classe n'a pas été déclaré.

Exercice 2 : Tests Unitaires

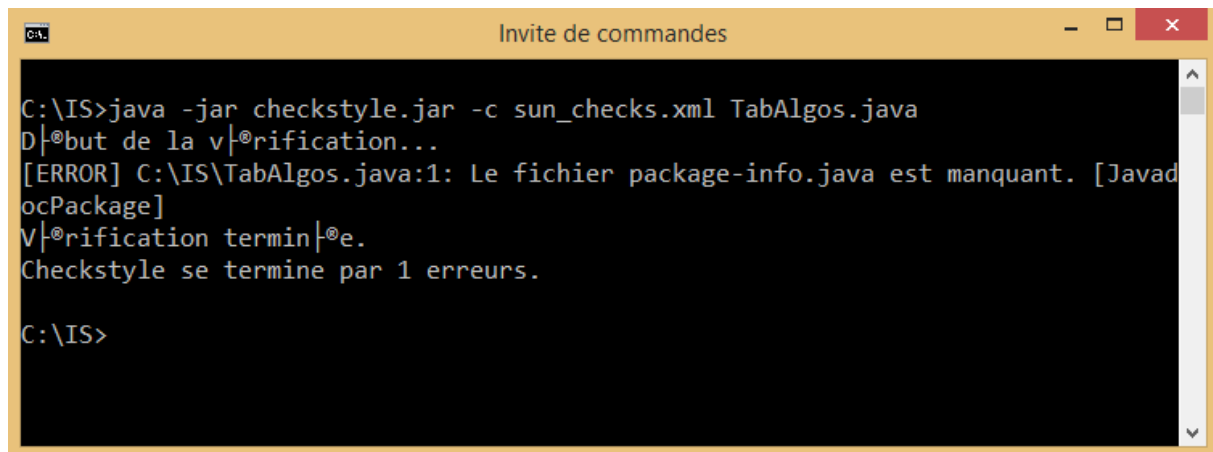
1- Veuillez trouver ci-dessous le code des tests unitaires :

```
import static org.junit.Assert.*;
import org.junit.Test;
/**
 * Test pour TabAlgos.
 */
public final class TabAlgosTest {
    /** tab est un tableau static private.*/
    private final int[] tab = {1, 13, 20, 9, 30, 18, 7};
    /** tab est un tableau static private.*/
    private final int[] tab1 = {1, 13, 20, 9, 30, 18, 7};
    /** a est une constante.*/
    private final int a = 30;
    /** b est une constante.*/
    private final double b = 14;
    /** Teste la methode plusGrand() . */
    @Test
    public void plusGrandTest() {
        int pgrd = TabAlgos.plusGrand(tab);
        assertEquals(a, pgrd);
    }
    /** Teste la methode moyenne() . */
    @Test
    public void moyenneTest() {
        double avg = TabAlgos.moyenne(tab);
        assertEquals(b, avg, b);
    }
    /** Teste la methode egaux() . */
    @Test
    public void egauxTest() {
        assertTrue(TabAlgos.egaux(tab, tab1));
    }
}
```

2- Implémentation des méthodes :

```
/**
 * Opération sur les tableaux.
 * @author KLEVA KOUGBLENOU
 */
public final class TabAlgos {
    /** Constructeur pour empêcher l'instanciation.*/
    private TabAlgos() { }
    /**
     * Renvoie la valeur la plus grande d'un tableau.
     * @param tab tableau d'entier
     * @return grand la valeur la plus grand
     */
    public static int plusGrand(final int[] tab) {
        int tampon = tab[0];
        for (int i = 1; i < tab.length; i++) {
            if (tab[i] > tampon) {
                tampon = tab[i];
            }
        }
        return tampon;
    }
    /** Calcul la moyenne des valeurs d'un tableau.
     * @param tab tableau d'entier
     * @return la moyenne
     */
    public static double moyenne(final int[] tab) {
        int somme = 0;
        double moy;
        for (int i = 0; i < tab.length; i++) {
            somme += tab[i];
        }
        moy = somme / tab.length;
        return moy;
    }
    /** Teste si 2 tableaux sont identiques.
     * (avec les elements dans le meme ordre)
     * @param tab1 est un tableau d'entier
     * @param tab2 est un tableau d'entier
     * @return true / false
     */
    public static boolean egaux(final int[] tab1, final int[] tab2) {
        if (tab1.length == tab2.length) {
            for (int i = 0; i < tab1.length; i++) {
                if (tab1[i] != tab2[i]) {
                    i = tab1.length;
                    return false;
                }
            }
            return true;
        } else {
            return false;
        }
    }
}
```

- 3- Vérifiez la validité des tests avec le code implémenté et la conformité avec les règles d'écritures en utilisant "checkstyle" :
- ✓ La vérification de la conformité du code avec les règles de "checkstyle" est terminée avec une erreur due au fait que le fichier javadoc contenant la description de la classe n'a pas été déclaré.



```
C:\IS>java -jar checkstyle.jar -c sun_checks.xml TabAlgos.java
D\@but de la v\@rification...
[ERROR] C:\IS\TabAlgos.java:1: Le fichier package-info.java est manquant. [JavadocPackage]
V\@rification termin\@e.
Checkstyle se termine par 1 erreurs.

C:\IS>
```

Fig. 03 : Capture d'écran du rapport de checkstyle appliqué à la classe TabAlgos.java

- ✓ L'exécution des tests unitaires s'est terminée avec succès, le rapport se présente comme suit.



```
C:\IS>javac -cp junit.jar TabAlgos.java TabAlgosTest.java

C:\IS>java -cp junit.jar;hamcrest-core.jar; org.junit.runner.JUnitCore TabAlgosTest
JUnit version 4.11
...
Time: 0,014

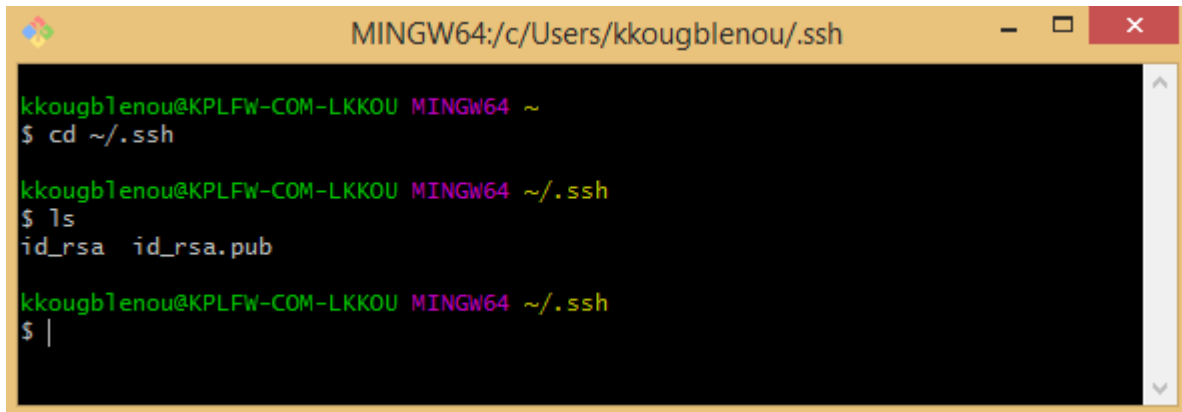
OK (3 tests)

C:\IS>
```

Fig. 04 : Capture d'écran du rapport d'exécution des tests unitaires

Exercice 3 : Dépôt sur serveur de Versionning GIT

- ✓ Git-2.21.0-64-bit a été téléchargé et installé sur Windows.
- ✓ L'e-mail koffi.kougblenou@etud.u-picardie.fr et le nom d'utilisateur [3k-kleva](#) ont été utilisés pour le paramétrage.
- ✓ La paire de clef RSA a été générée :



```
kkougblenou@KPLFW-COM-LKKOU MINGW64 ~
$ cd ~/.ssh

kkougblenou@KPLFW-COM-LKKOU MINGW64 ~/.ssh
$ ls
id_rsa  id_rsa.pub

kkougblenou@KPLFW-COM-LKKOU MINGW64 ~/.ssh
$ |
```

Fig. 05 : Capture d'écran montrant le répertoire ~/.ssh

- ✓ Veuillez trouver ci-dessous la clef publique :

```
$ cat ~/.ssh/id_rsa.pub
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCDomdnoFtrfcD1RebNeJl wWzco8VcyyoAT95sWSz8DuY+O
RTj KOB8m+dvORuMarJzfADI SALQv6PSrpRXq5QKA0Y7okHecBEfvcP6AKj DFzc+HDqFDNX04mTy
+BaVKfyN6gBaFH1j /49fJW16QKgAXboFOPpB94frNehx0GKpqxAJHhs0zl fa0eSzfn6YTVtyi gG
82Wni cTU7oezmQ0xhWnhkJUBBBbLi pt0ZKI 9+3kWA5rvxda/i 0sGXPLtfxbAgNm2q5mR8XUHi Vi
oJTArDqrYz0ti hWQr45Kou5NcbKj /fagCnFk4FEYI gQPeJl eEzLfj 5qNhuzGof2Cu20CsyLeoGG
tDW1n8Meyl j Uki QxXD5Efl 8P2AL21btvLE+8H00Fpl t8bt8X3L7ECJ069uuz800ai 3mHF6/MkTF
dEAH1hLbdpoce0df7d+0eVy+UuyEoBEI 005PhFrXDoF/sh+VhZ57Naj j 0+6sX2pGaVJHI HFQmnb
CczAXGDQmka1FW1aA8Fkj ZAAwBb0Q4shj NvnQmuaG8Qr/cR1Jmzj yPb/DyYki 92Syz4/cYF4Xsm
pl wURARz6D16auXW3UFh86bNXJ8RuyH17l yL9qNRT/cPaXfrvCvEekxSPdQYgPCmxzn6SA05k2w
XoEzl XSrAi p7m7rz2RDVi 40Xo9M43i 13zzyu0i w== koffi . kougbl enou@u-pi cardi e. fr
```

- ✓ Le travail de l'activité sera déposé dans le dossier activite1 de notre dépôt git.