

Exercice 1 : Ecriture de code

1. Pourquoi ce code ne compile-t-il pas ?

Le code ne compile pas, parceque :

- La variable Prix ne peut pas être final car sa valeur devrait être modifiable par la méthode setPrix
- La variable reference n'a pas été initialisée
- Une instruction return manquante dans la méthode getReference()

2. Corrigez le code pour qu'il compile (en respectant les consignes de la documentation).

Code Corrigé

```
/** Classe Produit représentant un produit avec un prix et une référence */
public class Produit {
    private final String reference;
    private double prix;
    double tva = 0.20;
    public Produit(String reference) {this.reference = reference;}
    /** @return le prix */
    public final double getPrix() {
        return this.prix;
    }
    /** modifie le prix */
    public final void setPrix(double prix){this.prix = prix;}
    /**
     * @return la reference si le prix est positif, null sinon
     */
    public final String getReference() {
        String resultat = reference;
        if ( prix > 0) return reference;
        return null;}
    @Override
    public boolean equals(Object o) {
        return reference == ((Produit)o).reference;
    }
}
```

3. Donnez le rapport de *checkstyle*, *spotbugs* et *PMD* appliqués à votre code.

Rapport checkstyle :

```
C:\devoir1\exo1>java -jar checkstyle-10.8.0-all.jar -c sun_checks.xml src/Produit.java
```

Starting audit...

```
[ERROR] C:\devoir1\exo1\src\Produit.java:1: First sentence should end with a period.
```

```
[JavadocStyle]
```

```
[ERROR] C:\devoir1\exo1\src\Produit.java:1: Missing package-info.java file. [JavadocPackage]
```

```
[ERROR] C:\devoir1\exo1\src\Produit.java:3:1: Missing a Javadoc comment. [JavadocVariable]
```

```
[ERROR] C:\devoir1\exo1\src\Produit.java:4:1: Missing a Javadoc comment. [JavadocVariable]
```

```
[ERROR] C:\devoir1\exo1\src\Produit.java:5:1: Missing a Javadoc comment. [JavadocVariable]
```

[ERROR] C:\devoir1\exo1\src\Produit.java:5:8: Variable 'tva' must be private and have accessor methods. [VisibilityModifier]
[ERROR] C:\devoir1\exo1\src\Produit.java:5:14: '0.20' is a magic number. [MagicNumber]
[ERROR] C:\devoir1\exo1\src\Produit.java:6:16: Parameter reference should be final. [FinalParameters]
[ERROR] C:\devoir1\exo1\src\Produit.java:6:23: 'reference' hides a field. [HiddenField]
[ERROR] C:\devoir1\exo1\src\Produit.java:6:34: '{' at column 34 should have line break after. [LeftCurly]
[ERROR] C:\devoir1\exo1\src\Produit.java:6:34: '{' is not followed by whitespace. [WhitespaceAround]
[ERROR] C:\devoir1\exo1\src\Produit.java:6:61: ';' is not followed by whitespace. [WhitespaceAfter]
[ERROR] C:\devoir1\exo1\src\Produit.java:6:62: '}' is not preceded with whitespace. [WhitespaceAround]
[ERROR] C:\devoir1\exo1\src\Produit.java:11: First sentence should end with a period. [JavadocStyle]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:27: Parameter prix should be final. [FinalParameters]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:34: 'prix' hides a field. [HiddenField]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:34: Expected @param tag for 'prix'. [JavadocMethod]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:39: '{' at column 39 should have line break after. [LeftCurly]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:39: '{' is not followed by whitespace. [WhitespaceAround]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:39: '{' is not preceded with whitespace. [WhitespaceAround]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:56: ';' is not followed by whitespace. [WhitespaceAfter]
[ERROR] C:\devoir1\exo1\src\Produit.java:12:57: '}' is not preceded with whitespace. [WhitespaceAround]
[ERROR] C:\devoir1\exo1\src\Produit.java:18:1: 'if' construct must use '{}'. [NeedBraces]
[ERROR] C:\devoir1\exo1\src\Produit.java:18:4: '(' is followed by whitespace. [ParenPad]
[ERROR] C:\devoir1\exo1\src\Produit.java:19:12: ';' is not followed by whitespace. [WhitespaceAfter]
[ERROR] C:\devoir1\exo1\src\Produit.java:19:13: '}' is not preceded with whitespace. [WhitespaceAround]
[ERROR] C:\devoir1\exo1\src\Produit.java:20:1: Class 'Produit' looks like designed for extension (can be subclassed), but the method 'equals' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'Produit' final or making the method 'equals' static/final/abstract/empty, or adding allowed annotation for the method. [DesignForExtension]
[ERROR] C:\devoir1\exo1\src\Produit.java:20:1: Definition of 'equals()' without corresponding definition of 'hashCode()'. [EqualsHashCode]
[ERROR] C:\devoir1\exo1\src\Produit.java:21:23: Parameter o should be final. [FinalParameters]
[ERROR] C:\devoir1\exo1\src\Produit.java:22:30: 'typecast' is not followed by whitespace. [WhitespaceAfter]
Audit done.
Checkstyle ends with 30 errors.

Rapport Spotbug:

Voir fichier 'Produit rapport-spotbugs avant.html'

Rapport PMD:

```
C:\Users\JEAN>pmd.bat -d C:\devoir1\exo1\src\Produit.java -f text -R
rulesets/java/quickstart.xml
May 27, 2023 9:18:30 PM net.sourceforge.pmd.PMD encourageToUseIncrementalAnalysis
WARNING: This analysis could be faster, please consider using Incremental Analysis:
https://pmd.github.io/pmd-6.55.0/pmd_userdocs_incremental_analysis.html
C:\devoir1\exo1\src\Produit.java:2:  NoPackage:  All classes, interfaces, enums and
annotations must belong to a named package
C:\devoir1\exo1\src\Produit.java:17:  UnusedLocalVariable:  Avoid unused local variables
such as 'resultat'.
C:\devoir1\exo1\src\Produit.java:18:  ControlStatementBraces: This statement should have
braces
C:\devoir1\exo1\src\Produit.java:21:  OverrideBothEqualsAndHashCode: Ensure you override
both equals() and hashCode()
```

4. Expliquez en quoi les problèmes soulevés sont nuisibles.

Ces problèmes sont nuisibles car elles peuvent contribuer à la détection de Fausse erreurs ,ils imposent une rigidité de codage qui n'est pas forcément bénéfique pour le projet ,peut diminuer la performance d'un projet en augmentant le temps de compilation.

5. Proposez une réécriture en tenant compte des problèmes soulevés par *checkstyle*, *spotbugs* et *PMD*.

Reecriture du code :

```
package info.ht;
/**
 *Classe Produit représentant un produit avec un prix et une référence.
 *@author jean garino.
 *@version 1.1
 */

public class Produit {
    /**
     *reference.
     */
    private final String reference;
    /**
     *prix.
     */
    private double prix;
    /**
     *TVA.
     */
    public static final double TVA = 0.20;

    /**
     *Constructor of the class.
```

```

    *@param reference2
    */
    public Produit(final String reference2) {
        this.reference = reference2;
    }

    /**
    *return the price.
    *@return le prix
    */
    public final double getPrix() {
        return this.prix;
    }

    /**
    *modifie le prix.
    *@param prix2
    */
    public final void setPrix(final double prix2) {
        this.prix = prix2;
    }

    /**
    *return the reference of the class.
    * @return la reference si le prix est positif, null sinon
    */
    public final String getReference() {
        String resultat;

        if (prix > 0) {
            resultat = this.reference;
        } else {
            resultat = null;
        }
        return resultat;
    }

    /**
    *methode equals.
    *@param o
    *@return t;
    */

    @Override
    public boolean equals(final Object o) {
        boolean t = false;
        if (o instanceof Produit) {
            Produit p = (Produit) o;
            t = this.reference.equals(p.reference);
        }

        return t;
    }

    /**

```

```

*methode equals.
*@return t;
*/
@Override
public int hashCode() {
    final int vAL = 42;
    assert false : "hashCode not designed";
    return vAL; // any arbitrary constant will do
}
}

```

6. Donnez les rapports appliqués au code réécrit montrant que tous les problèmes sont corrigés.

Rapport checkstyle :

```

C:\devoir1\exo1>java -jar checkstyle-10.8.0-all.jar -c sun_checks.xml
src/Produit.java
Starting audit...
Audit done.

```

Rapport spotbugs :

Voir fichier 'Produit rapport-spotbugs avant.html'

Rapport PMD :

```

C:\devoir1\exo1>pmd.bat -d C:\devoir1\exo1\src\Produit.java -f text -R
rulesets/java/quickstart.xml

```

May 27, 2023 9:30:53 PM net.sourceforge.pmd.PMD
encourageToUseIncrementalAnalysis

WARNING: This analysis could be faster, please consider using Incremental
Analysis: [https://pmd.github.io/pmd-](https://pmd.github.io/pmd-6.55.0/pmd_userdocs_incremental_analysis.html)
6.55.0/pmd_userdocs_incremental_analysis.html

Exercice 2 : Tests unitaires

1. Ecrivez des tests unitaires (en utilisant JUnit 5) permettant de tester les méthodes à implémenter.

Implementation des tests unitaires

```
package info.net;

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

```
public class TestTabAlgos {

    @Test
    public void testPlusGrand1() {
        final int resultat = 7;
        final int[] tab1 = {1, 2, 4, 6, 7, 7};
        assertEquals(resultat, TabAlgos.plusGrand(tab1));
    }

    @Test
    public void testPlusGrand2() {
        final int resultat = 9;
        final int[] tab1 = {9, 2, 4, 6, 5, 0};
        assertEquals(resultat, TabAlgos.plusGrand(tab1));
    }
}
```

@Test

```
public void testPlusGrand3() {  
    final int resultat = 11;  
    final int[] tab1 = {1, 2, 4, 11, 5, 3};  
    assertEquals(resultat, TabAlgos.plusGrand(tab1));  
}
```

@Test

```
public void testMoyenne1() {  
    final int resultat = 5;  
    final int[] tab = {9, 3, 4, 6, 7, 1};  
    assertEquals(resultat, TabAlgos.moyenne(tab));  
}
```

@Test

```
public void testMoyenne2() {  
    final int[] tab1 = {};  
  
    try {  
        TabAlgos.moyenne(tab1);  
        fail("should not reach here");  
    } catch (IllegalArgumentException e) {  
        System.out.println("exception attendue " + e.getMessage());  
    }  
}
```

@Test

```
public void testMoyenne3() {  
    final int[] tab1 = null;  
  
    try {
```

```

        TabAlgos.moyenne(tab1);
        fail("should not reach here");
    } catch (IllegalArgumentException e) {

        System.out.println("exception attendue " + e.getMessage());

    }

}

```

```

@Test
public void testEgaux1() {
    final int[] tab1 = {1, 2, 5, 8, 9};
    final int[] tab2 = {9, 8, 5, 2, 1};
    assertFalse(TabAlgos.egaux(tab1, tab2));
}

```

```

@Test
public void testEgaux2() {
    final int[] tab1 = {1, 2, 5, 8, 9};
    final int[] tab2 = {1, 2, 5, 8, 9};
    assertTrue(TabAlgos.egaux(tab1, tab2));
}

```

```

@Test
public void testSimilaires1() {
    final int[] tab1 = {5, 2, 1, 8, 9};
    final int[] tab2 = {1, 2, 5, 8, 9};
    assertTrue(TabAlgos.similaires(tab1, tab2));
}

```



```
@Test
```

```
public void testSimilaires2() {  
    final int[] tab1 = {5, 2, 1, 8, 9};  
    final int[] tab2 = {1, 2, 4, 8, 9};  
    assertFalse(TabAlgos.similaires(tab1, tab2));  
}  
}
```

2. Implémentez les méthodes en respectant les règles d'écriture contrôlées par les outils checkstyle, spotbugs et PMD.

Implementation des méthodes

```
/**  
 *@version 1.0  
 *@author Jean Garino  
 */  
package info.net;  
  
public final class TabAlgos {  
  
    /**  
     *constructor of the class.  
     */  
    private TabAlgos() {  
  
    }  
  
    /** @return valeur la plus grande d'un tableau.  
     *@param tab of int.  
     */  
    public static int plusGrand(final int[] tab) {  
        int a = tab[0];
```

```

    for (int i: tab) {
        if (a < i) {
            a = i;
        }
    }

    return a;
}

/**
 * @return moyenne des valeurs du tableau.
 * @param tab of int.
 * @throw IllegalArgumentException si tab est null ou vide.
 */
public static double moyenne(final int[] tab) {
    double a = 0;
    int b = 0;
    if (tab != null) {
        for (int i:tab) {
            a = a + i;
            b++;
        }
    }
    if (b == 0 || tab == null) {
        throw new IllegalArgumentException();
    }

    return a / b;
}

/** 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'occurrences (avec les éléments dans le même ordre).

```

```

@param tab1 of int.
@param tab2 of int.
** /

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

    int j = 0;
    boolean result = true;
    for (int i: tab1) {
        if (i != tab2[j]) {
            result = false;
        }
        j++;
    }

    return result;
}

/** 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).
 * @param tab1 of int
 * @param tab2 of int
 ** /

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

    int a1 = 0;
    int a2 = 0;
    for (int i: tab1) {
        for (int j:tab1) {
            if (i == j) {
                a1++;
            }
        }
    }

```

```

    }
}

for (int k:tab2) {
    if (i == k) {
        a2++;
    }
}

if (a1 != a2) {
    return false;

}

}

return true;
}

}

```

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

```
C:\devoir1\exo2>java -jar junit-platform-console-standalone-1.8.1.jar -cp out --scan-
classpath
```

exception attendue null

exception attendue null

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

```

←[36m?←[0m
←[36m |←[0m ←[36mJUnit Jupiter←[0m ←[32m?←[0m
←[36m |   └←[0m ←[36mTestTabAlgos←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestSimilaires1()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestSimilaires2()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestPlusGrand1()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestPlusGrand2()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestPlusGrand3()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestMoyenne1()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestMoyenne2()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestMoyenne3()←[0m ←[32m?←[0m
←[36m |     |←[0m ←[34mtestEgaux1()←[0m ←[32m?←[0m
←[36m |     └←[0m ←[34mtestEgaux2()←[0m ←[32m?←[0m
←[36m └←[0m ←[36mJUnit Vintage←[0m ←[32m?←[0m

```

Test run finished after 176 ms

```

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

```

4. Fournissez les rapports des tests unitaires, de checkstyle, spotbugs et PMD. Il ne doit rester aucune anomalie.

Rapport des tests unitaires :

```
C:\devoir1\exo2>java -jar junit-platform-console-standalone-1.8.1.jar -cp out --scan-classpath
```

```
exception attendue null
```

```
exception attendue null
```

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

```
←[36m?←[0m
←[36m |←[0m ←[36mJUnit Jupiter←[0m ←[32m?←[0m
←[36m |  └←[0m ←[36mTestTabAlgos←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestSimilaires1()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestSimilaires2()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestPlusGrand1()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestPlusGrand2()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestPlusGrand3()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestMoyenne1()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestMoyenne2()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestMoyenne3()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestEgaux1()←[0m ←[32m?←[0m
←[36m |    └←[0m ←[34mtestEgaux2()←[0m ←[32m?←[0m
←[36m └←[0m ←[36mJUnit Vintage←[0m ←[32m?←[0m
```

Test run finished after 176 ms

```
[      3 containers found      ]
```

```
[      0 containers skipped    ]
```

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

Rapport checkstyle :

```
java -jar checkstyle-10.8.0-all.jar -c sun_checks.xml src/TabAlgos.java
```

Starting audit...

Audit done.

Rapport spotbugs :

Voir fichier 'TabAlgos rapport-spotbugs.html'

Rapport PMD :

```
C:\devoir1\exo2>pmd.bat -d C:\devoir1\exo2\src\TabAlgos.java -f text -R
rulesets/java/quickstart.xml
```

```
May 27, 2023 11:34:58 PM net.sourceforge.pmd.PMD
encourageToUseIncrementalAnalysis
```

WARNING: This analysis could be faster, please consider using Incremental Analysis:
https://pmd.github.io/pmd-6.55.0/pmd_userdocs_incremental_analysis.html

Exercice 3 : Dépôt sur serveur de Versionning

Repositoty Github : [JeanGarino/Activit-s-C306: Devoirs-C306 \(github.com\)](https://github.com/JeanGarino/Activit-s-C306: Devoirs-C306)