

1. Exercice 1 : Ecriture du Code**a. Pourquoi le code ne compile pas ?**

Le champ 'prix' n'est pas initialisé dans la creation de l'instance Produit.

La boucle 'if' dans la creation de la methode getReference() n'a pas d'accolades après la condition, de plus elle ne retourne pas de resultat

```
public final String getReference() {
String resultat = reference;
if ( prix > 0) return reference;
}
```

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

```
2. public class Produit {
3.
4.     private String reference;
5.     private double prix;
6.     double tva = 0.20;
7.
8.     public Produit(String reference, Double prix) {
9.         this.reference = reference;
10.        this.prix = prix;}
11.
12.    /** @return le prix */
13.    public double getPrix() {
14.        return this.prix;
15.    }
16.    /** modifie le prix */
17.    public final void setPrix(Double prix){this.prix = prix;}
18.    /**
19.     * @return la reference si le prix est positif, null sinon
20.     */
21.    public final String getReference() {
22.        String resultat = reference;
23.        if ( prix > 0)
24.            {return this.reference;}
25.        return resultat;
26.    }
27.    @Override
28.    public boolean equals(Object o) {
29.        return reference == ((Produit)o).reference;
30.    }
31.
32.
33. }
```

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

- CheckStyle (51 Problems)

Description	Path	Location	Type
'{' à la colonne 2 devrait avoir un saut de ligne après.	/samples/src/main/java/samples	line 25	
'{' à la colonne 2 devrait être sur la ligne précédente.	/samples/src/main/java/samples	line 25	
'{' à la colonne 33 devrait avoir un saut de ligne après.	/samples/src/main/java/samples	line 18	
'}' à la colonne 19 devrait être seul sur sa ligne.	/samples/src/main/java/samples	line 11	
'}' à la colonne 51 devrait être seul sur sa ligne.	/samples/src/main/java/samples	line 18	

'ctor def modifier' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 9

'if' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 4.
 /samples/src/main/java/samples line 24

'if lcurly' au niveau d'indentation 8 n'est pas indenté correctement, le niveau attendu est 6.
 /samples/src/main/java/samples line 25

'member def modifier' au niveau d'indentation 8 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 5

'member def modifier' au niveau d'indentation 8 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 6

'member def type' au niveau d'indentation 8 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 7

'METHOD_DEF' doit être séparé de la ligne précédente. /samples/src/main/java/samples line 18

'METHOD_DEF' doit être séparé de la ligne précédente. /samples/src/main/java/samples line 22

'METHOD_DEF' doit être séparé de la ligne précédente. /samples/src/main/java/samples line 28

'method def modifier' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 14

'method def modifier' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 18

'method def modifier' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 22

'method def modifier' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 28

'method def rcurly' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 16

'method def rcurly' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 27

'method def rcurly' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 2.
 /samples/src/main/java/samples line 31

Commentaire Javadoc manquant. /samples/src/main/java/samples line 3

Il manque une espace après ';'. /samples/src/main/java/samples line 11

Il manque une espace après ';'. /samples/src/main/java/samples line 18

Il manque une espace après ';'. /samples/src/main/java/samples line 26

Il manque une espace après 'conversion de type'. /samples/src/main/java/samples line 30

Il y a une espace de trop après '('. /samples/src/main/java/samples line 24

Javadoc tag '@return' doit être précédé d'une ligne vide. /samples/src/main/java/samples line 13

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 5

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 6

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 7

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 10

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 11

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 25

La ligne contient un caractère tabulation. /samples/src/main/java/samples line 33

La première phrase de la Javadoc doit se terminer avec un point. /samples/src/main/java/samples line 17

Le commentaire Javadoc doit être sur plusieurs lignes. /samples/src/main/java/samples line 13

Le fils de 'ctor def' au niveau d'indentation 8 n'est pas indenté correctement, le niveau attendu est 4.
 /samples/src/main/java/samples line 10

Le fils de 'ctor def' au niveau d'indentation 8 n'est pas indenté correctement, le niveau attendu est 4.
 /samples/src/main/java/samples line 11

Le fils de 'method def' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 4.
 /samples/src/main/java/samples line 15

Le fils de 'method def' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 4.
 /samples/src/main/java/samples line 23

Le fils de 'method def' au niveau d'indentation 0 n'est pas indenté correctement, le niveau attendu est 4.
 /samples/src/main/java/samples line 30

Le résumé Javadoc est manquant. /samples/src/main/java/samples line 13

Le résumé Javadoc est manquant. /samples/src/main/java/samples line 19

WhitespaceAround: '{' is not followed by whitespace. Empty blocks may only be represented as {} when not part of a multi-block statement (4.1.3) /samples/src/main/java/samples line 18

WhitespaceAround: '{' is not followed by whitespace. Empty blocks may only be represented as {} when not part of a multi-block statement (4.1.3) /samples/src/main/java/samples line 25

WhitespaceAround: '{' is not preceded with whitespace. /samples/src/main/java/samples line 18

WhitespaceAround: '}' is not preceded with whitespace. /samples/src/main/java/samples line 11

WhitespaceAround: '}' is not preceded with whitespace. /samples/src/main/java/samples line 18

WhitespaceAround: '}' is not preceded with whitespace. /samples/src/main/java/samples line 26

WhitespaceAround: 'return' is not preceded with whitespace. /samples/src/main/java/samples line 25

- Spotbugs (3 Problem)

Description	Resource	Path	Location	Type
Comparaison d'objets String utilisant == ou != dans samples.Produit.equals(Object) [Troubling(11), Normal confidence]	Produit.java	/samples/src/main/java/samples	line 30	SpotBugs Problem (Troubling)
samples.Produit.equals(Object) ne vérifie pas la nullité d'un paramètre [Troubling(11), Normal confidence]	Produit.java	/samples/src/main/java/samples	line 30	SpotBugs Problem (Troubling)
samples.Produit définit equals() et utilise Object.hashCode() [Troubling(14), High confidence]	Produit.java	/samples/src/main/java/samples	line 30	SpotBugs Problem (Troubling)

- PMD

P	Ligne	Rule	Message
▶	3	CommentRequired	CommentRequired: Class comments are required
▶	5	ImmutableField	ImmutableField: Private field 'reference' could be made final; it is only initialized in the declaration or constructor.
▶	5	BeanMembersShouldSerialize	BeanMembersShouldSerialize: Found non-transient, non-static member. Please mark as transient or provide accessors.
▶	5	CommentRequired	CommentRequired: Field comments are required
▶	6	CommentRequired	CommentRequired: Field comments are required
▶	7	CommentDefaultAccessModifier	CommentDefaultAccessModifier: To avoid mistakes add a comment at the beginning of the tva field if you want a default access modifier
▶	7	DefaultPackage	DefaultPackage: Use explicit scoping instead of the default package private level
▶	7	CommentRequired	CommentRequired: Field comments are required
▶	9	CommentRequired	CommentRequired: Public method and constructor comments are required
▶	9	MethodArgumentCouldBeFinal	MethodArgumentCouldBeFinal: Parameter 'prix' is not assigned and could be declared final
▶	9	MethodArgumentCouldBeFinal	MethodArgumentCouldBeFinal: Parameter 'reference' is not assigned and could be declared final
▶	18	MethodArgumentCouldBeFinal	MethodArgumentCouldBeFinal: Parameter 'prix' is not assigned and could be declared final
▶	23	LocalVariableCouldBeFinal	LocalVariableCouldBeFinal: Local variable 'resultat' could be declared final
▶	23	PrematureDeclaration	PrematureDeclaration: Avoid declaring a variable if it is unreferenced before a possible exit point.
▶	23	DataflowAnomalyAnalysis	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'resultat' (lines '23'-'27').
▶	25	OnlyOneReturn	OnlyOneReturn: A method should have only one exit point, and that should be the last statement in the method
▶	29	OverrideBothEqualsAndHashcode	OverrideBothEqualsAndHashcode: Ensure you override both equals() and hashCode()
▶	29	ShortVariable	ShortVariable: Avoid variables with short names like o
▶	29	MethodArgumentCouldBeFinal	MethodArgumentCouldBeFinal: Parameter 'o' is not assigned and could be declared final
▶	30	CompareObjectsWithEquals	CompareObjectsWithEquals: Use equals() to compare object references.
▶	30	UseEqualsToCompareStrings	UseEqualsToCompareStrings: Use equals() to compare strings instead of '==' or '!='

d. Expliquez-en quoi les problèmes soulevés sont nuisibles.

Il s'agit des erreurs de syntaxe qui ne respectent pas le standard dans l'écriture d'un code.

L'écriture d'un code suit un standard qui permet de faciliter la lecture du code, la verification de la syntaxe et de la documentation, la verification des erreurs.

- e. Ecriture du code qui tient en compte les problèmes soulevés par CheckStyle, PMD et SpotBugs

```
package samples;

/** Classe Produit.*/
public class Produit {
    /** @reference est le nom propre du produit.*/
    private final String reference;
    /**@prix est le prix du produit.*/
    private double prix;
    /** @tva est la taxe sur la valeur ajoutée.*/
    double tva = 0.20;

    public Produit(String reference, Double prix) {
        this.reference = reference;
        this.prix = prix;
    }

    /** retourne le prix. */
    public double getPrix() {
        return this.prix;
    }

    /** modifie le prix. */
    public final void setPrix(Double prix) {
        this.prix = prix;
    }

    /**retourne la reference si le prix est positif, null sinon. */
    public final String getReference() {
        String resultat = reference;
        if (prix > 0) {
            return this.reference;
        }
        return resultat;
    }

    @Override
    public boolean equals(Object obj) {
        //return reference == ((Produit)o).reference;
        return obj.equals(this.reference);
    }
}
```

f. Rapport de contrôle appliqués au code

- Tous les problèmes Checkstyle corrigés
- 2 Problèmes SpotBugs restants
- 13 Problèmes PMD Restants

```
package samples;

/** Classe Produit.*/
public class Produit {
    /** @reference est le nom propre du produit.*/
    private final String reference;
    /**@prix est le prix du produit.*/
    private double prix;
    /** @tva est la taxe sur la valeur ajoutée.*/
    double tva = 0.20;

    public Produit(String reference, Double prix) {
        this.reference = reference;
        this.prix = prix;
    }
}
```

Markers ×

rs, 14 warnings, 1 other

Description	Location
PMD Markers (13 items)	
CommentDefaultAccessModifier: To avoid mistakes add a comment at the beginning of the tva field if you want a default access modifier	line 10
DefaultPackage: Use explicit scoping instead of the default package private level	line 10
CommentRequired: Public method and constructor comments are required	line 11
MethodArgumentCouldBeFinal: Parameter 'prix' is not assigned and could be declared final	line 11
MethodArgumentCouldBeFinal: Parameter 'reference' is not assigned and could be declared final	line 11
MethodArgumentCouldBeFinal: Parameter 'prix' is not assigned and could be declared final	line 25
LocalVariableCouldBeFinal: Local variable 'resultat' could be declared final	line 32
PrematureDeclaration: Avoid declaring a variable if it is unreferenced before a possible exit point.	line 32
DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'resultat' (lines '32'-'37').	line 32
OnlyOneReturn: A method should have only one exit point, and that should be the last statement in the method	line 34
MethodArgumentCouldBeFinal: Parameter 'o' is not assigned and could be declared final	line 40
OverrideBothEqualsAndHashCode: Ensure you override both equals() and hashCode()	line 40
ShortVariable: Avoid variables with short names like o	line 40
SpotBugs Problem (Troubling) (2 items)	
samples.Produit.equals(Object) ne vérifie pas la nullité d'un paramètre [Troubling(11), Normal confidence]	line 41
samples.Produit définit equals() et utilise Object.hashCode() [Troubling(14), High confidence]	line 41

2. Exercice 2 :

1. Ecrire les tests unitaires

```
package samples;

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

public class TestTabAlgos {

    TabAlgos Tab = new TabAlgos();

    @Test
    public void TestplusGrand () {
        assertEquals(12, Tab.plusGrand({10,9,8,5,12,4}));
    }

    @Test
    public void Testmoyenne () {
        assertEquals(8, Tab.moyenne({10,9,8,5,12,4}));
    }

}
```