

**HAROON ABID****Compte rendu TP03****Table des matières**

PARTIE 1 :.....	2
Partie 2 : QCM Java .....	4
Identifiants – Classes .....	4
Identifiants – Méthodes .....	5
println() .....	5
Correction Des Bugs.....	7
CONCLUSION.....	8

## PARTIE 1 :

Algo BP1 :

```
1 package Yoann;
2
3 import java.util.Scanner;
4
5 public class moyenne_notes {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         // Déclaration du tableau de 4 notes
10        double[] notes = new double[4];
11        double somme = 0;
12
13        // Saisie des 4 notes
14        for (int i = 0; i < notes.length; i++) {
15            System.out.print("Saisir la note " + (i + 1) + " : ");
16            notes[i] = sc.nextDouble();
17            somme += notes[i];
18        }
19
20        // Calcul de la moyenne
21        double moyenne = somme / notes.length;
22
23        // Affichage de la moyenne
24        System.out.println("La moyenne des notes est : " + moyenne);
25
26        sc.close();
27    }
28 }
29
30
```

```
Saisir la note 1 : 13
Saisir la note 2 : 18
Saisir la note 3 : 09
Saisir la note 4 : 11
La moyenne des notes est : 12.75
```

## ALGO BP2

```
1 package Yoann;
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner clavier = new Scanner(System.in);
8
9         // Variables
10        int age;
11
12        // Ecrire "entrez votre age"
13        System.out.print("Entrez votre age : ");
14
15        // Lire age
16        age = clavier.nextInt();
17
18        // Si age < 18 alors
19        if (age < 18) {
20            // Afficher "vous etes mineur"
21            System.out.println("Vous etes mineur");
22        }
23        // Sinon si age > 40
24        else if (age > 40) {
25            // Afficher "vous etes vétéran"
26            System.out.println("Vous etes vétéran");
27        }
28        // Sinon
29        else {
30            // Afficher "vous etes senior"
```

```
Entrez votre age : 88
Vous etes vétéran
```

```
Entrez votre age : 5
Vous etes mineur
```

```
Entrez votre age : 25
Vous etes senior
```

## Partie 2 : QCM Java

Le langage le plus proche du circuit est : le langage machine.

Les langages permettant d'utiliser des mots comme *read*, *write*, *add* appartiennent au niveau : haut niveau.

Les règles d'écriture d'un langage correspondent à : la syntaxe.

Le programme qui traduit du haut niveau vers le langage machine est : le compilateur.

Une variable est : un emplacement mémoire nommé.

Les instructions d'un programme regroupées logiquement forment : une procédure.

Une instance de classe correspond à : un objet.

L'architecture de Java est dite : neutre.

Les classes Java sont transformées en : bytecode.

Chaque instruction Java doit se terminer par : un point-virgule.

## Identifiants – Classes

maClasse → légal, pas conventionnel

b. void → illégal

c. Golden Retriever → illégal

d. invoice# → illégal

e. 36535CodePostal → illégal

f. Appartement → légal et conventionnel

g. Fruit → légal et conventionnel

h. 8888 → illégal

i. EcranTotal() → illégal

j. Acompte\_recevable → légal, pas conventionnel

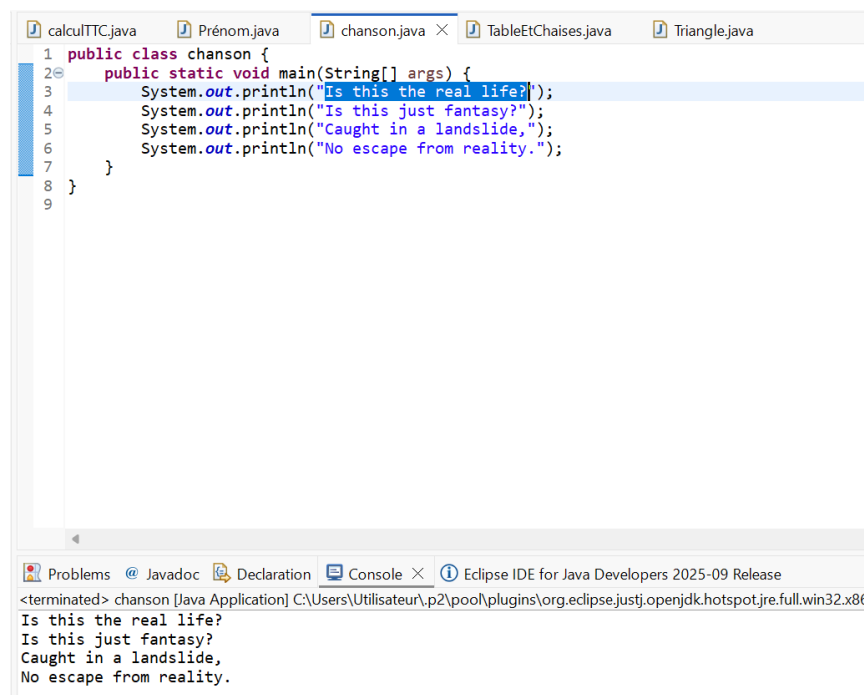
## Identifiants – Méthodes

- associationRoles() → légal et conventionnel
- b. void() → illégal
- c. Golden Retriever() → illégal
- d. invoice#() → illégal
- e. 24500CodePostal() → illégal
- f. PayrollApp() → légal, pas conventionnel
- g. getReady() → légal et conventionnel
- h. 911() → illégal
- i. EcranTotal() → légal, pas conventionnel
- j. Acompte\_Recevable() → légal, pas conventionnel

## println()

Affichage de 4 lignes avec System.out.println().

- On doit afficher 4 lignes avec ce code.



```
1 public class chanson {
2     public static void main(String[] args) {
3         System.out.println("Is this the real life?");
4         System.out.println("Is this just fantasy?");
5         System.out.println("Caught in a landslide,");
6         System.out.println("No escape from reality.");
7     }
8 }
9
```

Problems @ Javadoc Declaration Console Eclipse IDE for Java Developers 2025-09 Release

<terminated> chanson [Java Application] C:\Users\Utilisateur\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64.jre\bin\java.exe -Xmx1024m -Dorg.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64.jre=C:\Users\Utilisateur\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64.jre\bin\java.exe -jar C:\Users\Utilisateur\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64.jre\bin\java.exe

Is this the real life?  
Is this just fantasy?  
Caught in a landslide,  
No escape from reality.

- Ça nous donne un résultat de chanson qu'on a choisit sur 4 lignes vous pouvez afficher ce que vous voulez.

- On va créer une table et 2 chaises il faut mettre les étoiles dans le code avec une précision pour bien transformer le code.

```

1 public class TableEtChaises {
2     public static void main(String[] args) {
3         System.out.println("X                X");
4         System.out.println("X                X");
5         System.out.println("Xxxx  0  0  xxxX");
6         System.out.println("X  x  0  0  x  X");
7         System.out.println("X  x  0  0  x  X");
8     }
9 }
10

```

```

X      00000000      X
Xxxxx  0  0  xxxX
X  x  0  0  x  X
X  x  0  0  x  X

```

- Voilà le résultat qu'on obtient ce n'est pas propre mais ça nous permet de voir comment on peut utiliser les codes pour pouvoir faire ça et franchement ça m'a beaucoup aidé.
- Après ça on doit faire un triangle ou on peut appeler ça un arbre aussi avec le même code je vais mettre le code et le résultat.

```

1 public class Triangle {
2     public static void main(String[] args) {
3         System.out.println(" * ");
4         System.out.println(" *  * ");
5         System.out.println(" *  *  * ");
6         System.out.println(" *  *  *  * ");
7         System.out.println(" *  *  *  *  * ");
8     }
9 }

```

```

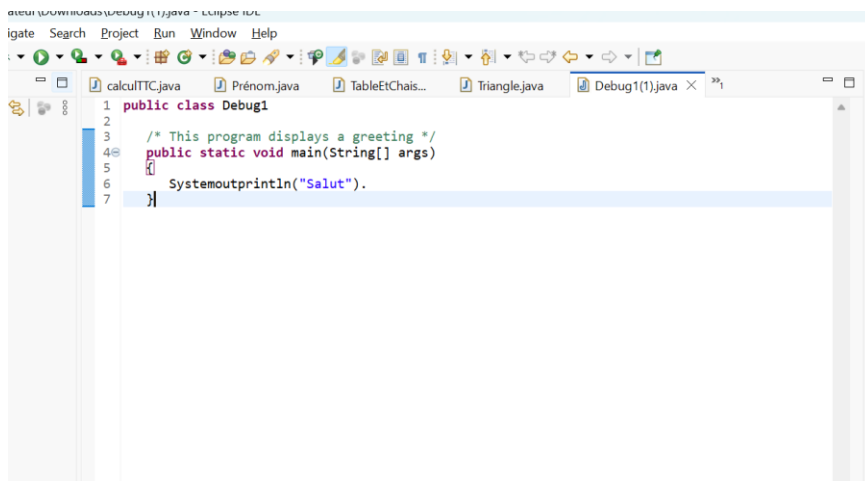
 * 
 *  * 
 *  *  * 
 *  *  *  * 
 *  *  *  *  * 

```

- Là on peut voir les Etoiles entre les () et un résultat en bas qui nous donne un triangle bien structuré.

## Correction Des Bugs

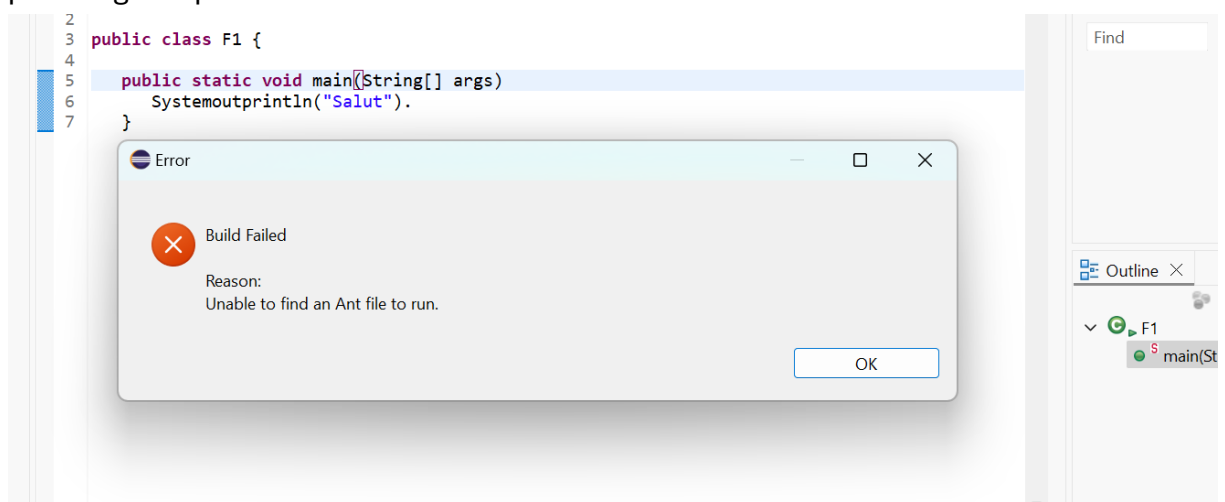
- Dans cet exercice on doit corriger les 4 fichier il y a des petites erreur.
- **Debug 1 :**
- J'ai bien corriger les erreurs dans ce fichier et on a un code avec 0 erreurs.



```

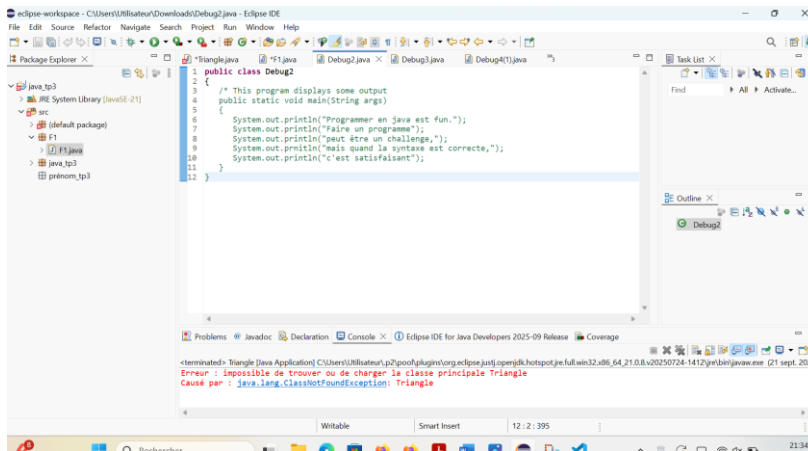
1 public class Debug1
2 {
3     /* This program displays a greeting */
4     public static void main(String[] args)
5     {
6         System.out.println("Salut");
7     }
  
```

- Mais pour lancer j'ai un petit problème ca m'affiche ca j'ai essayer mais j'ai pas pu corriger le problème.

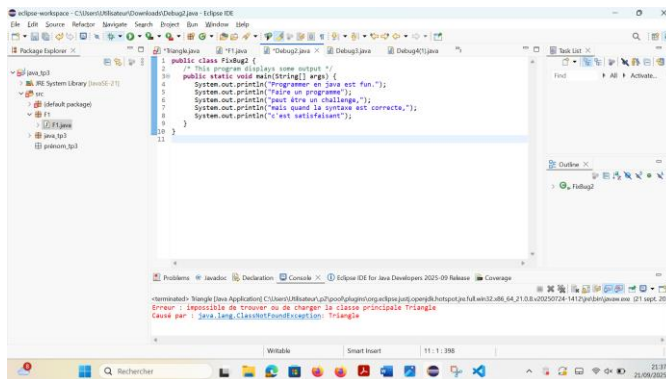


- **FICHIER 2 :**

- J'ai corrigé le code mais pareil ca m'affiche le même message voici le code dans le fichier.



- Et la version corriger



- Le problème était dans la 2ème ligne on doit mettre \*/ pour un commentaire et pas /\*.
- Il manquait les [] entre string et args sur la 3ème ligne.
- Et aussi une faute de frappe sur println il y'avait écrit prnitln.

## CONCLUSION

Au cours de ce TP, j'ai travaillé sur différents exercices en Java :

- Les algorithmes BP1 et BP2, qui m'ont permis de comprendre comment résoudre des problèmes et afficher les résultats d'un programme.
- Le quiz Java et les exercices sur les identifiants, qui m'ont aidé à maîtriser les règles de syntaxe, les conventions de nommage pour les classes et les méthodes.
- L'utilisation de System.out.println() pour afficher du texte et la création de programmes simples comme l'affichage de tables, triangles ou messages personnalisés.



- La correction de bugs, qui m'a appris à repérer et corriger des erreurs classiques : fautes de frappe, majuscules/minuscules, points-virgules manquants, mauvais noms de classe ou erreurs dans le main.

Ce TP m'a donc permis de renforcer mes bases en Java, d'améliorer ma rigueur dans l'écriture du code, et de mieux comprendre l'organisation d'un projet Java dans Eclipse.