



Introduction

Lors de ce travail pratique, l'objectif principal était de nous familiariser avec les concepts fondamentaux de l'algorithmique et de découvrir leur traduction concrète dans le langage Java.

Algo B1 - exo: 4: la moyenne de 4 notes	2
Algo B2 Q.3 : les utilisateurs en catégories selon leur âge	3
QCM de TP JAVA	5
Exercice de programmation :	5
3. Classe ParoleChanson.java	7
Conclusion	13

Algo B1 - exo: 4: la moyenne de 4 notes

Description

Cet algorithme calcule la moyenne de 4 notes saisies par l'utilisateur en utilisant un tableau.

```
*MoyenneNotes.java X
1 package tp.java;
2
3 import java.util.Scanner;
4
5 public class MoyenneNotes {
6     public static void main(String[] args) {
7         Scanner clavier = new Scanner(System.in);
8
9         double[] note = new double[4];
10
11         System.out.print("Saisir la note 1 : ");
12         note[0] = clavier.nextDouble();
13
14         System.out.print("Saisir la note 2 : ");
15         note[1] = clavier.nextDouble();
16
17         System.out.print("Saisir la note 3 : ");
18         note[2] = clavier.nextDouble();
19
20         System.out.print("Saisir la note 4 : ");
21         note[3] = clavier.nextDouble();
22
23         double moyenne = (note[0] + note[1] + note[2] + note[3]) / 4;
24
25         System.out.println("La moyenne des 4 notes est : " + moyenne);
26
27         clavier.close();
28     }
29 }
```

Résultat :

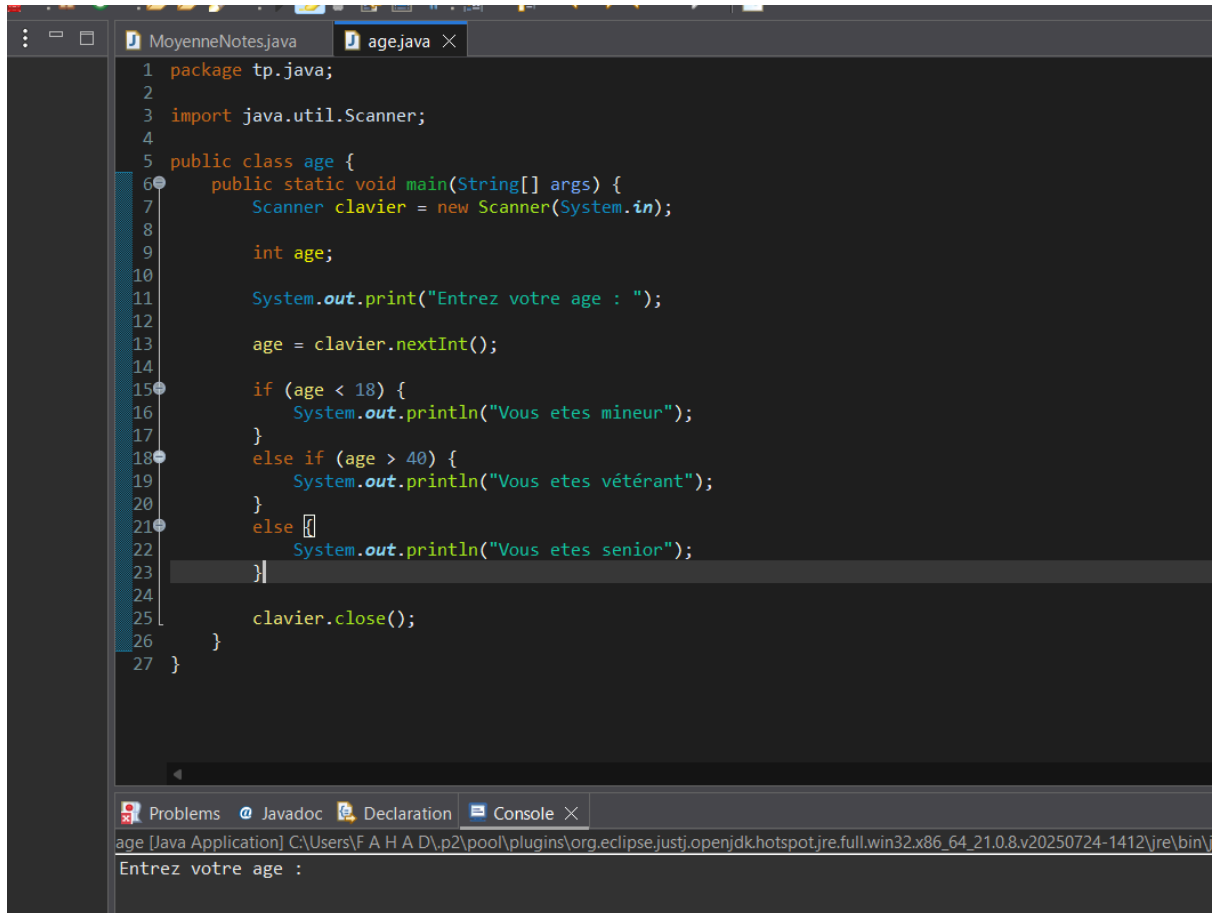
Lors de l'exécution, le programme demande à l'utilisateur de saisir les notes une par une. Par exemple, si l'on entre 15, 20, 35 et 45, le programme calcule et affiche la moyenne :

```
Problems @ Javadoc Declaration Console X
<terminated> MoyenneNotes [Java Application] C:\Users\F A H A D\p2\pool\
Saisir la note 1 : 15
Saisir la note 2 : 20
Saisir la note 3 : 35
Saisir la note 4 : 45
La moyenne des 4 notes est : 28.75
```

Algo B2 Q.3 : les utilisateurs en catégories selon leur âge

Description

Cet algorithme classe les utilisateurs en catégories selon leur âge avec des structures conditionnelles. Ou classer une personne dans une catégorie en fonction de son âge.

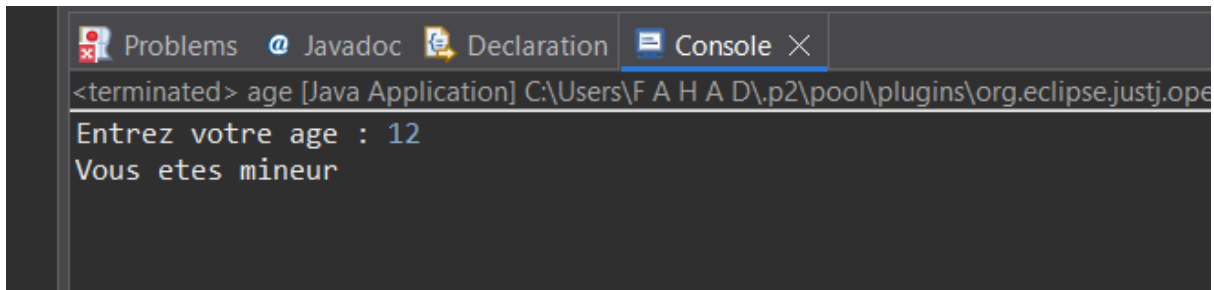


```
1 package tp.java;
2
3 import java.util.Scanner;
4
5 public class age {
6     public static void main(String[] args) {
7         Scanner clavier = new Scanner(System.in);
8
9         int age;
10
11         System.out.print("Entrez votre age : ");
12
13         age = clavier.nextInt();
14
15     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

The screenshot shows the Eclipse IDE with two tabs: 'MoyenneNotes.java' and 'age.java'. The 'age.java' tab is active, displaying the Java code. The code uses a Scanner to read an integer age from the user. It then uses a series of if-else statements to categorize the user: 'mineur' for ages less than 18, 'vétérant' for ages greater than 40, and 'senior' for all other ages. The Scanner is closed at the end of the main method. The bottom of the IDE shows the 'Console' tab with the prompt 'Entrez votre age : '.

Si l'âge est inférieur à 18, la personne est mineure. Si l'âge est supérieur à 40, elle est considérée comme vétéran. Sinon, entre 18 et 40, elle est senior.

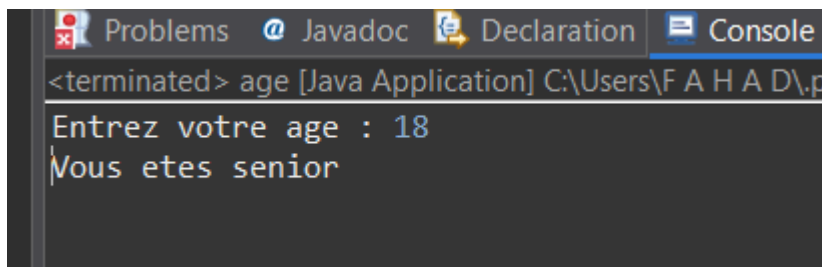
Si l'utilisateur saisit **12** :

A screenshot of the Eclipse IDE's console window. The console shows the output of a Java application. The first line is "<terminated> age [Java Application] C:\Users\F A H A D\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi". The second line is "Entrez votre age : 12". The third line is "Vous etes mineur". The console window has tabs for "Problems", "Javadoc", "Declaration", and "Console".

```
<terminated> age [Java Application] C:\Users\F A H A D\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
Entrez votre age : 12
Vous etes mineur
```

Vous etes mineur

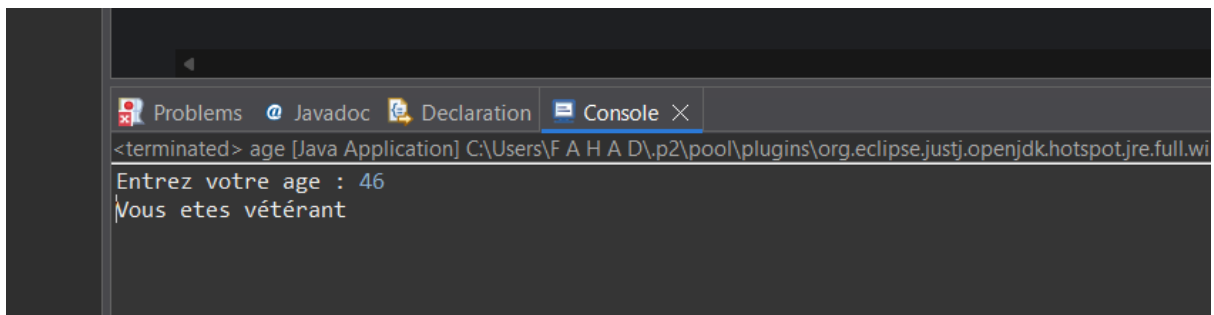
Si l'utilisateur saisit **18** :

A screenshot of the Eclipse IDE's console window. The console shows the output of a Java application. The first line is "<terminated> age [Java Application] C:\Users\F A H A D\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi". The second line is "Entrez votre age : 18". The third line is "Vous etes senior". The console window has tabs for "Problems", "Javadoc", "Declaration", and "Console".

```
<terminated> age [Java Application] C:\Users\F A H A D\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
Entrez votre age : 18
Vous etes senior
```

Vous etes senior

Si l'utilisateur saisit **46** :

A screenshot of the Eclipse IDE's console window. The console shows the output of a Java application. The first line is "<terminated> age [Java Application] C:\Users\F A H A D\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi". The second line is "Entrez votre age : 46". The third line is "Vous etes vétéran". The console window has tabs for "Problems", "Javadoc", "Declaration", and "Console".

```
<terminated> age [Java Application] C:\Users\F A H A D\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
Entrez votre age : 46
Vous etes vétéran
```

Vous êtes vétéran

QCM de TP JAVA

1. Le langage machine le plus basique niveau circuit est?
 - **a. Le langage machine**
2. Les langages qui permettent d'utiliser un vocabulaire qui utilise les termes : read, write ou add sont :
 - **b. Haut niveau**
3. les règles du langage de programmation constituent :
 - **a. la syntaxe**
4. Un _____ traduit les instructions de langage de haut niveau en code machine
 - **c. un compilateur**
5. Les emplacements de mémoire nommés de l'ordinateur sont appelés :
 - **b. variables**
6. les opérations individuelles utilisées dans un programme informatique sont souvent regroupés en unités logiques appelées :
 - **a. une procédures**
7. Une instance de classe est :
 - **c. un objet**
8. Java a une architecture
 - **a. neutre**
9. Vous devez compiler les classes écrites en Java dans
 - **a. un bytecode**
10. Toutes les instructions de programmation Java doivent se terminer par :
 - **c. un point-virgule**

Exercice de programmation :

1. Chacun des identifiants de classe suivant est-il légal, légal et conventionnel ou illégal ?
 - a. maClasse - **Légal**
 - b. Void - **Illégal**
 - c. Golden Retriever - **Illégal**
 - d. invoice# - **Illégal**
 - e. 36535CodePostal - **Illégal**
 - f. Appartement - **Légal**
 - g. Fruit - **Légal**
 - h. 8888 - **Illégal**
 - i. EcranTotal() - **Illégal**
 - j. Acompte Recevable - **Légal**

2. Chacun des identifiants de la méthode suivante est-il légal, légal et conventionnel ou illégal?

a. AssociationRoles () - **Légal**

b. Void () - **Illégal**

c. Golden Retriever () - **Illégal**

d. Invoice# () - **Illégal**

e. 24500CodePostal () - **Illégal**

f. PayrollApp () - **Légal**

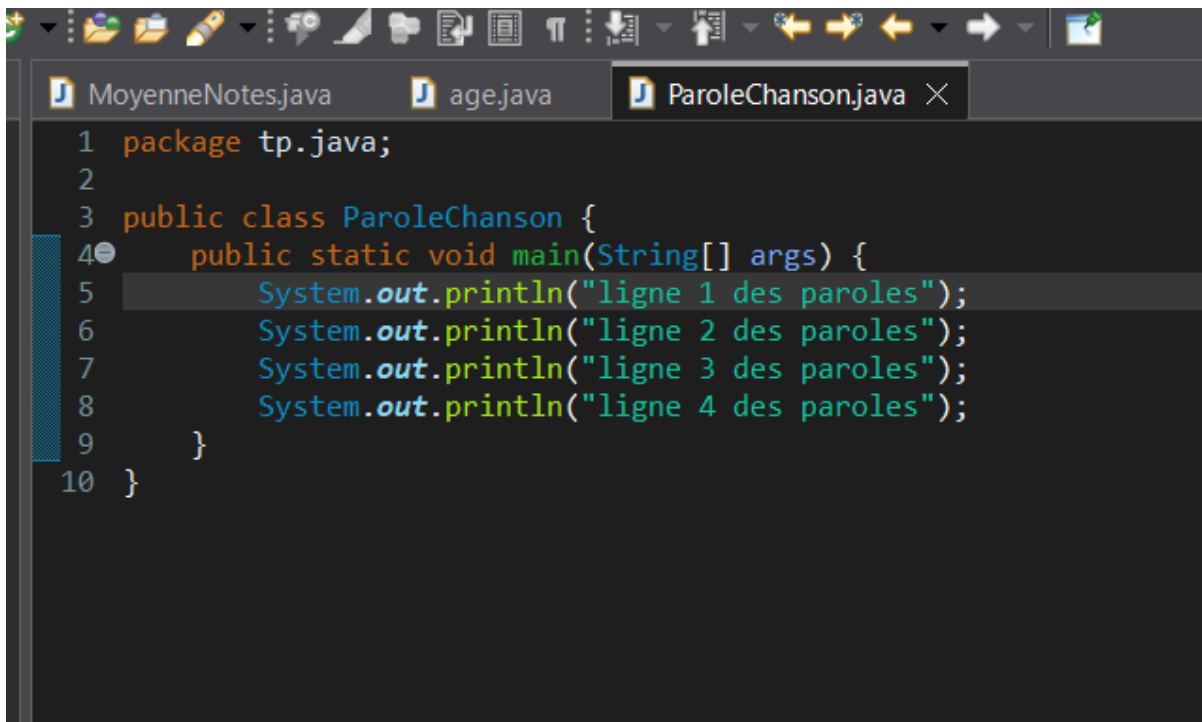
g. getReady () - **Légal**

h. 911 () - **Légal**

i. EcranTotal () - **Légal**

j. Acompte_Recevable () - **Légal**

3. Classe ParoleChanson.java

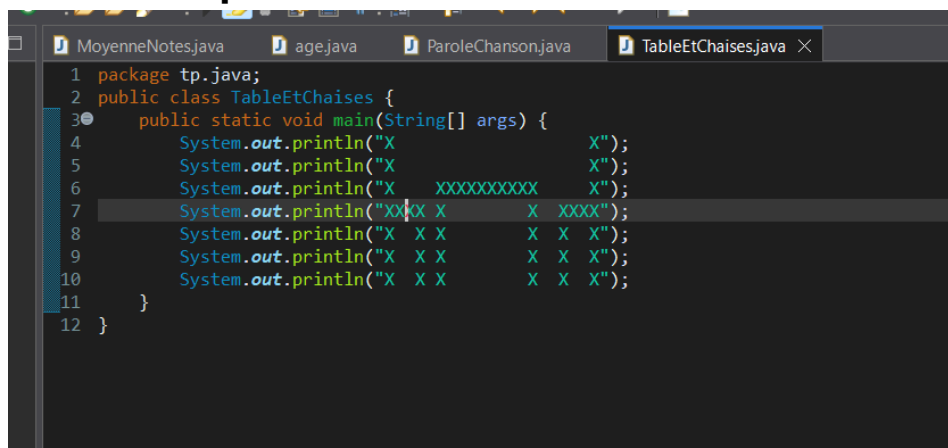


```
1 package tp.java;
2
3 public class ParoleChanson {
4     public static void main(String[] args) {
5         System.out.println("ligne 1 des paroles");
6         System.out.println("ligne 2 des paroles");
7         System.out.println("ligne 3 des paroles");
8         System.out.println("ligne 4 des paroles");
9     }
10 }
```

Voici le resultat,

```
ligne 1 des paroles
ligne 2 des paroles
ligne 3 des paroles
ligne 4 des paroles
```

4. compilez et testez une classe

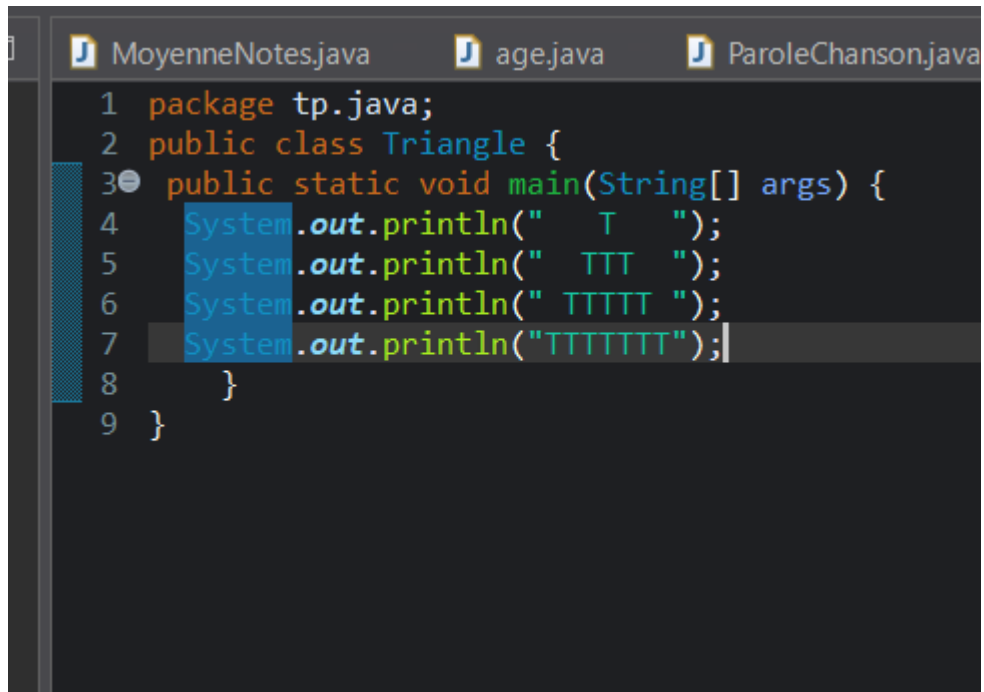


```
1 package tp.java;
2 public class TableEtChaises {
3     public static void main(String[] args) {
4         System.out.println("X X X X X");
5         System.out.println("X X X X X");
6         System.out.println("X XXXXXXXXXX X");
7         System.out.println("XXXX X X X XXXX");
8         System.out.println("X X X X X X X");
9         System.out.println("X X X X X X X");
10        System.out.println("X X X X X X X");
11    }
12 }
```

Voici le resultat,

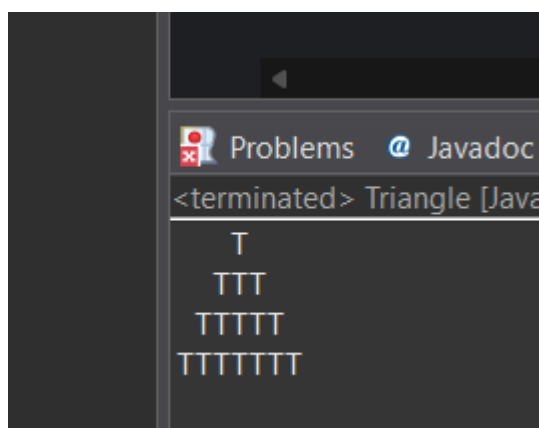
```
X X X X X
X X X X X
X XXXXXXXXXX X
XXXX X X X XXXX
X X X X X X X
X X X X X X X
X X X X X X X
```

5. compilez et testez une classe



```
1 package tp.java;
2 public class Triangle {
3     public static void main(String[] args) {
4         System.out.println("  T  ");
5         System.out.println("   TTT  ");
6         System.out.println("  TTTTT  ");
7         System.out.println("TTTTTTT");
8     }
9 }
```

Voici le resultat,



```
<terminated> Triangle [Java]
T
TTT
TTTTT
TTTTTTT
```


Installer les 4 debug.java pour continuer le tp

TP3 Java

[Quizz et TP Java.pdf](#)

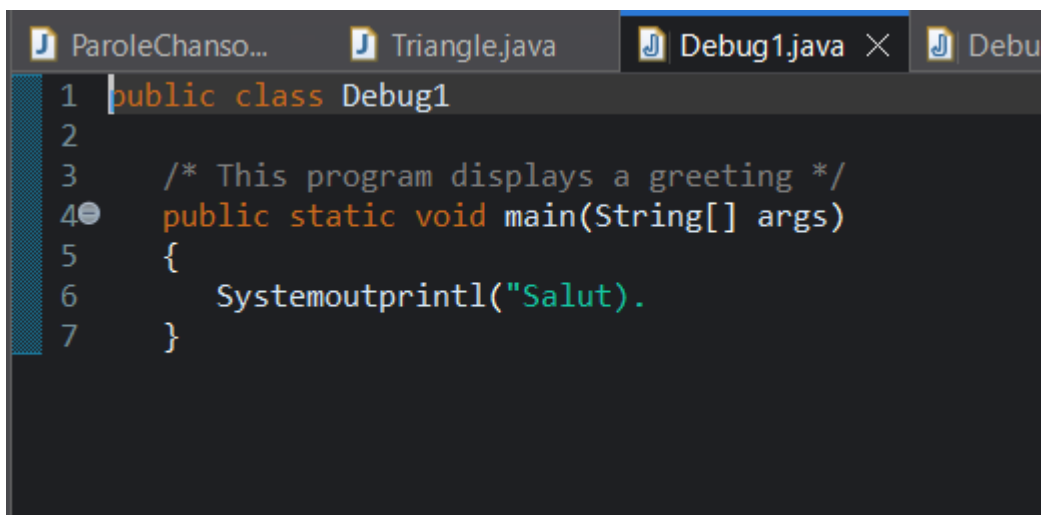
[Debug1.java](#)

[Debug2.java](#)

[Debug3.java](#)

[Debug4.java](#)

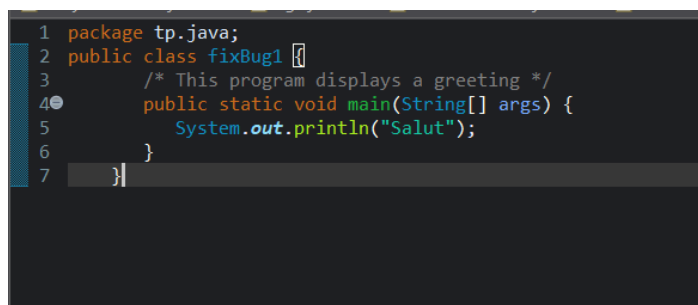
Et puis continuer a repéré les problème et corriger,



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

1. Accolade { manquante après le nom de la classe.
2. SystemOutPrintln n'est pas une méthode valide.
3. Guillemet fermant manque dans la chaîne de caractères
4. Manque point-virgule à la fin de l'instruction.

Voici la correction,



```
1 package tp.java;
2 public class fixBug1 {
3     /* This program displays a greeting */
4     public static void main(String[] args) {
5         System.out.println("Salut");
6     }
7 }
```

Debug 2,

```
public class Debug2
{
    /* This program displays some output
    public static void main(String args)
    {
        System.out.println("Programmer en java est fun.");
        System.out.println("Faire un programme");
        System.out.println("peut être un challenge,");
        System.out.prnitln("mais quand la syntaxe est correcte,");
        System.out.println("c'est satisfaisant");
    }
}
```

La correction,

```
1 package tp.java;
2 public class FixBug2 {
3     /* This program displays some output */
4     public static void main(String[] args) {
5         System.out.println("Programmer en java est fun.");
6         System.out.println("Faire un programme");
7         System.out.println("peut être un challenge,");
8         System.out.println("mais quand la syntaxe est correcte,")
9         System.out.println("c'est satisfaisant");
10    }
11 }
```

Resultat,

```
Programmer en java est fun.
Faire un programme
peut être un challenge,
mais quand la syntaxe est correcte,
c'est satisfaisant
```

Debug3,

```
1 public class Debug33
2 {
3     public static void main(String[] args)
4     {
5         System.Out.println("Derrière la rivière");
6         system.out.println("et au dela du bois");
7         SysTem.Out.println("à la maison du garde nous irons");
8     }
9 }
```

Correction,

```
1 package tp.java;
2 public class FixBug3 {
3     public static void main(String[] args) {
4         System.out.println("Derrière la rivière");
5         System.out.println("et au dela du bois");
6         System.out.println("à la maison du garde nous irons");
7     }
8 }
```

Resultat,

```
Derrière la rivière
et au dela du bois
à la maison du garde nous irons
```

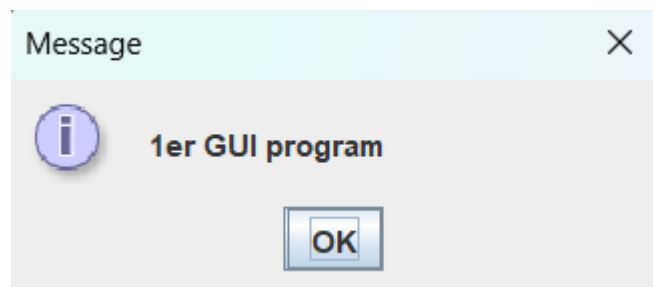
Debug 4,

```
1 package tp.java;
2 import javax.swing.JOptionPane;
3 public class Debug4
4 {
5     public static main(String[] args)
6     {
7         JOptionPane.showMessageDialog(null, 1er GUI program)!
8     }
9 }
```

Correction ,

```
1 package tp.java;
2 import javax.swing.JOptionPane;
3 public class FixBug4 {
4     public static void main(String[] args) {
5         JOptionPane.showMessageDialog(null, "1er GUI program");
6     }
7 }
```

Resulat,



Conclusion

Ce TP de Java était vraiment intéressant et m'a permis d'apprendre les bases de la programmation.

D'abord, j'ai fait un programme pour calculer la moyenne de 4 notes. C'était assez simple et ça m'a montré comment utiliser les tableaux pour stocker des données. Ensuite, j'ai créé un algorithme pour classer les personnes selon leur âge avec les conditions if et else. C'était plus compliqué mais j'ai bien compris comment faire prendre des décisions à un programme. Le QCM était utile pour comprendre la théorie. J'ai découvert les différences entre les langages de programmation, le rôle d'un compilateur, et les concepts de base sur les objets et les classes en Java. Ça m'a donné de bonnes bases théoriques.

Les exercices de debug étaient vraiment formateurs. Au début je trouvais ça galère mais finalement c'était super important. J'ai appris à repérer et corriger les erreurs classiques comme les accolades manquantes, les fautes dans les noms de méthodes, ou les points-virgules oubliés. Maintenant je fais moins d'erreurs et je sais mieux les corriger.

