

# TP3 : Compte rendu

## Algorithme Java

# Sommaire

Quizz Java.....	3
Exercice de programmation :.....	4
Exercice 1.....	4
d. invoice# : illégal.....	4
Exercice 2.....	4
d. invoice#() : illégal.....	4
Exercice 3.....	5
Exercice 4.....	6
Exercice 5.....	7
Exercice correction de bugs.....	8
Fichier debug1 corrigé.....	8
Fichier debug2 corrigé.....	8
Fichier debug3 corrigé.....	8
Fichier debug4 corrigé.....	9
Cas pratique.....	10
1. Importation de classe JOptionPane.....	10
2. Déclaration de la classe et de la méthode main.....	10
3. Demander un nombre à l'utilisateur.....	10
4. Convertir la saisie en entier.....	11
5. Générer un nombre aléatoire.....	11
6. Afficher le nombre généré.....	11
7. Comparer et donner le résultat.....	11
7. Test du programme.....	12
Le cas problème.....	14
Yummy.....	14
Yummy2.....	15
YummyFun.....	15

## Quizz Java

Question 1 : Le langage machine le plus basique niveau circuit est :

Réponse : Le langage machine

Question 2 : Les langages qui permettent d'utiliser un vocabulaire qui utilise les termes : read, write ou add sont :

Réponse : Haut niveau

Question 3 : Les règles du langage de programmation constituent :

Réponse : La syntaxe

Question 4 : Un \_\_\_\_\_ traduit les instructions de langage de haut niveau en code machine

Réponse : Un compilateur

Question 5 : Les emplacements de mémoire nommés de l'ordinateur sont appelés :

Réponse : Adresses

Question 6 : Les opérations individuelles utilisées dans un programme informatique sont souvent regroupés en unités logiques appelées :

Réponse : Procédures

Question 7 : Une instance de classe est :

Réponse : Un objet

Question 8 : Java a une architecture

Réponse : Neutre

Question 9 : Vous devez compiler les classes écrites en Java dans :

Réponse : Un bytecode

Question 10 : Toutes les instructions de programmation Java doivent se terminer par : Un point-Virgule

## Exercice de programmation :

### Exercice 1

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

### Exercice 2

2. Chacun des identifiants de méthode suivant est-il légal, 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 et conventionnel
  - h. 911() : illégal
  - i. EcranTotal() : légal
  - j. Acompte\_Recevable() : légal

## Exercice 3

Nous créons un fichier intitulé ParoleChanson.java.

Ensuite pour afficher nos quatres lignes, nous allons utiliser une instruction println()

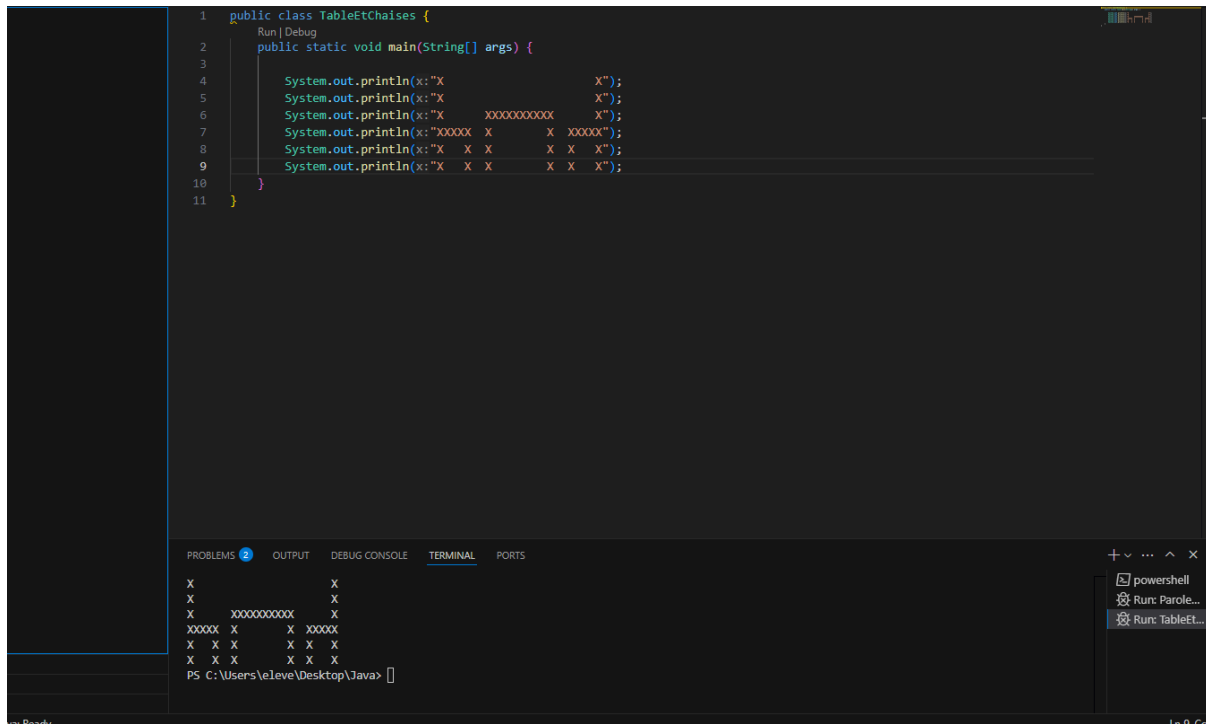
```
1 public class ParoleChanson {  
    Run | Debug  
2     public static void main(String[] args) {  
3         System.out.println(x:"Au travers de ça on decouvre de nouveaux chemins");  
4         System.out.println(x:"Un souffle leger s'envole");  
5         System.out.println(x:"Le marché s'anime de bon matin");  
6         System.out.println(x:"Il regarde sa vieille boussole");  
7     }  
8 }  
9
```

Voici le résultat dans le terminal dans VSCode.

```
Au travers de ça on decouvre de nouveaux chemins  
Un souffle leger s'envole  
Le marché s'anime de bon matin  
Il regarde sa vieille boussole
```

## Exercice 4

Ecrivez, compilez et testez une classe qui affiche le modèle illustré :



The screenshot shows an IDE with a Java class named `TableEtChaises`. The code uses `System.out.println` to print a pattern of 'X' characters. The output in the terminal window matches the pattern shown in the code.

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

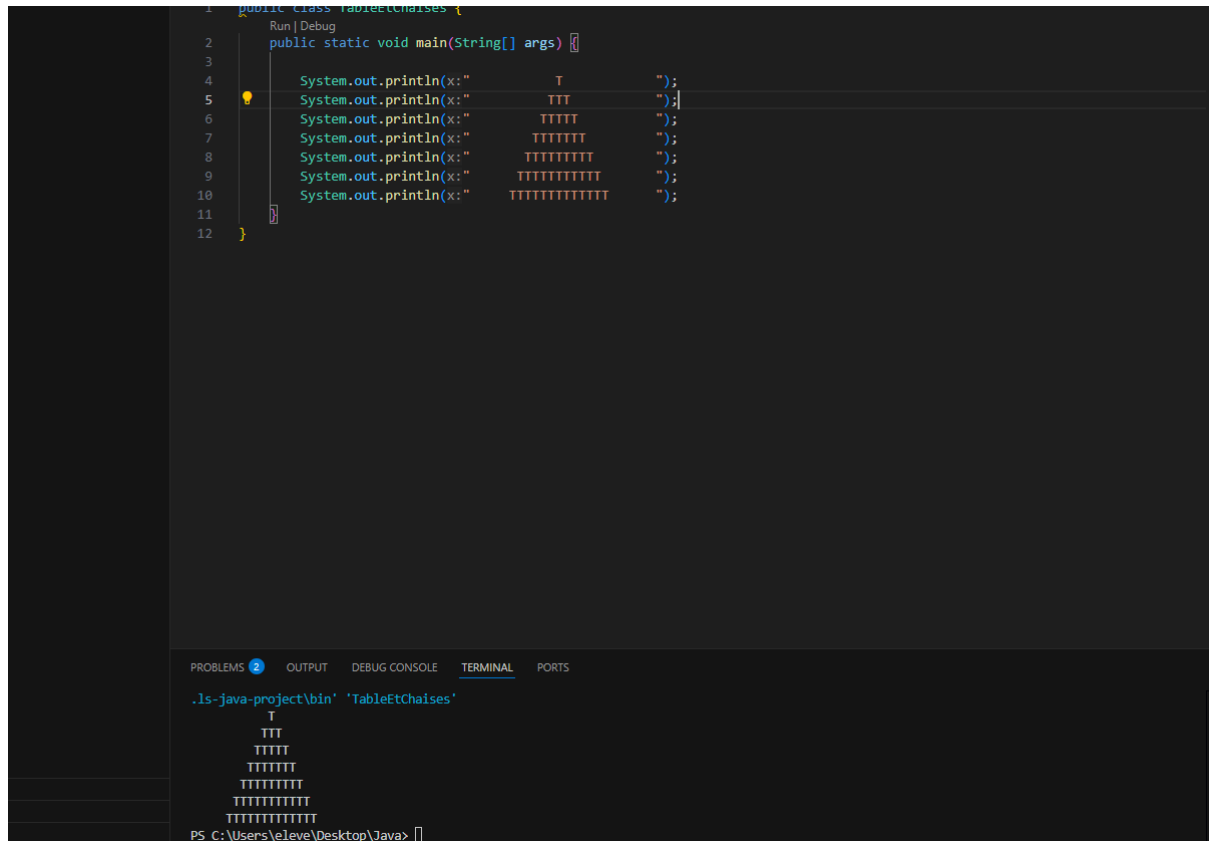
Terminal Output:

```
X               X  
X               X  
X      XXXXXXXXXXXX      X  
XXXXXX X      X XXXXXX  
X  X X      X X X  X  
X  X X      X X X
```

PS C:\Users\eleve\Desktop\Java>

## Exercice 5

5. Ecrivez, compilez et testez une classe qui affiche le modèle illustré :



The screenshot shows an IDE with a Java class named `TabletChaises`. The `main` method contains six `System.out.println` statements that print a pattern of 'T' characters. The pattern consists of six lines: the first line has one 'T', the second has three, the third has five, the fourth has seven, the fifth has nine, and the sixth has eleven. Below the code editor, the `TERMINAL` tab is active, showing the command `.\ls-java-project\bin\ 'TabletChaises'` and the resulting output of the program, which matches the pattern described in the code.

```
1 public class TabletChaises {
2     Run | Debug
3     public static void main(String[] args) {
4
5         System.out.println(x:"      T      ");
6         System.out.println(x:"     TTT     ");
7         System.out.println(x:"    TTTT    ");
8         System.out.println(x:"   TTTTT   ");
9         System.out.println(x:"  TTTTTT  ");
10        System.out.println(x:" TTTTTTT ");
11        System.out.println(x:"TTTTTTTTT");
12    }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
.\ls-java-project\bin\ 'TabletChaises'
```

```
      T
     TTT
    TTTT
   TTTTT
  TTTTTT
 TTTTTTT
TTTTTTTTT
```

PS C:\Users\eleve\Desktop\Java>

## Exercice correction de bugs

### Fichier debug1 corrigé

```
1 public class FixBug1 {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Salut");  
5     }  
6 }  
7
```

### Fichier debug2 corrigé

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

### Fichier debug3 corrigé

```
1 public class FixBug3  
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 }  
10
```



## Fichier debug4 corrigé

```
1  import javax.swing.JOptionPane;
2
3  public class FixBug4
4  {
5      public static void main(String[] args) {
6          JOptionPane.showMessageDialog(null, "1er GUI program");
7      }
8  }
```

## Cas pratique

## Objectif :

Nous devons créer une petite application Java qui demande à l'utilisateur de penser à un nombre entre 1 et 10, génère un nombre aléatoire dans le même intervalle, affiche ce nombre puis indique si l'utilisateur a deviné.

### 1. Importation de classe JOptionPane

Nous importons JOptionPane pour pouvoir afficher des boîtes de dialogue (saisie et messages)

```
import javax.swing.JOptionPane;
```

### 2. Déclaration de la classe et de la méthode main

Nous définissons une classe MiniJeu et la méthode main, point d'entrée du programme : c'est là que l'exécution commence.

```
import javax.swing.JOptionPane;

public class MiniJeu {
    public static void main(String[] args) {
```

### 3. Demander un nombre à l'utilisateur

Nous affichons une boîte de dialogue pour que l'utilisateur saisisse un nombre. La valeur que nous récupérons est de type String.

```
String reponse = JOptionPane.showInputDialog("Pensez à un nombre entre 1 et 10");
```

## 4. Convertir la saisie en entier

Nous transformons la chaîne saisie (String) en un nombre entier (int) afin de pouvoir comparer deux nombres.

```
int nombreUtilisateur = Integer.parseInt(reponse);
```

## 5. Générer un nombre aléatoire

Nous générons un nombre aléatoire entre 1 et 10 grâce à Math.random().  
Le résultat est d'abord un nombre décimal, que nous transformons en entier avec (int).  
En ajoutant 1, nous nous assurons que le résultat est bien compris entre 1 et 10 inclus.

```
int nombreAleatoire = 1 + (int)(Math.random() * 10);
```

## 6. Afficher le nombre généré

Nous montrons à l'utilisateur le nombre aléatoire produit par le programme.

```
JOptionPane.showMessageDialog(null, "Le nombre généré est : " + nombreAleatoire);
```

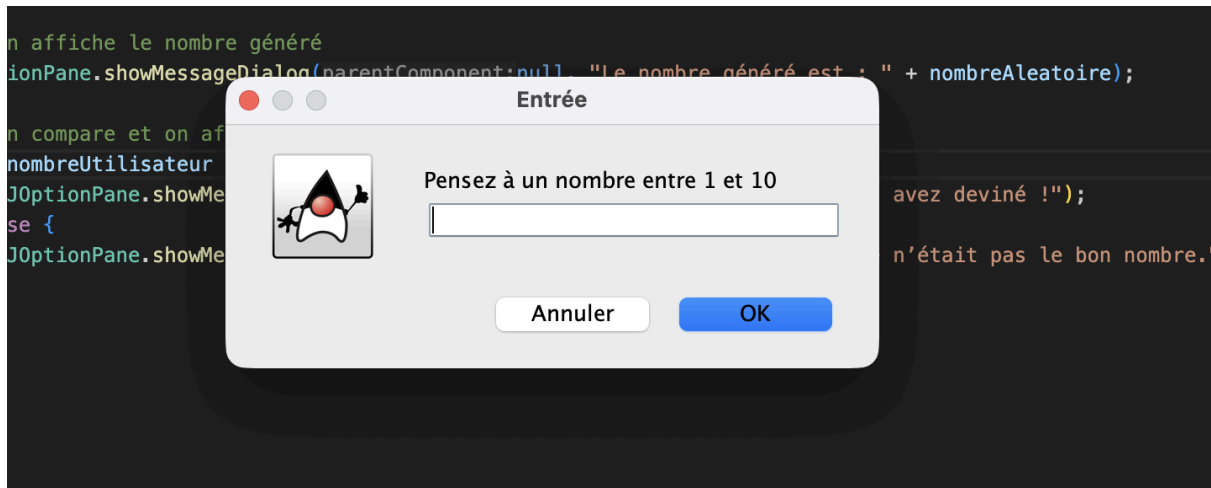
## 7. Comparer et donner le résultat

Nous comparons le nombre entré par l'utilisateur avec celui généré par le programme.  
Si les deux nombres sont identiques, nous affichons un message de réussite.  
Sinon, nous affichons un message d'échec.

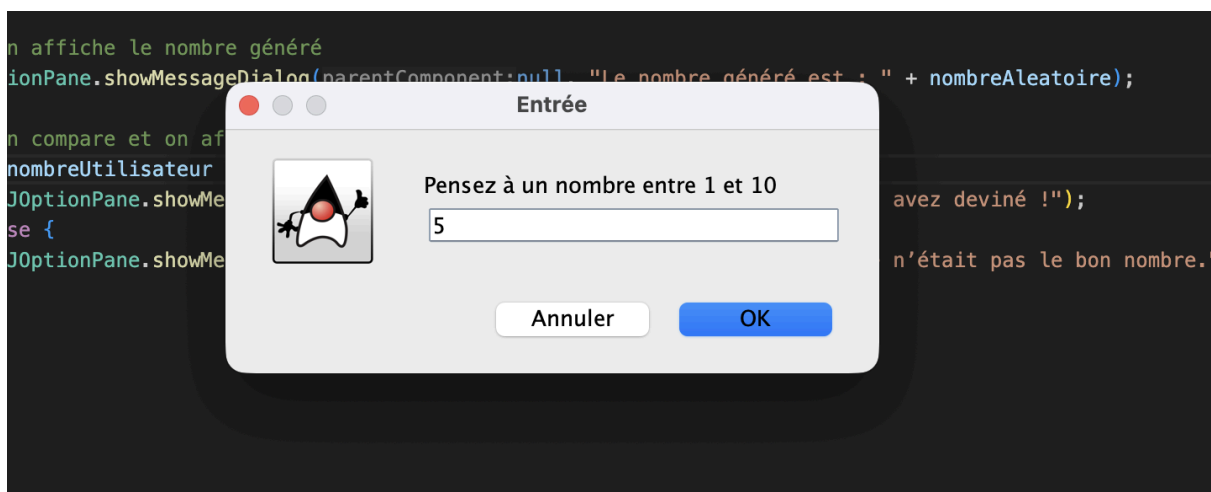
```
if (nombreUtilisateur == nombreAleatoire) {  
    JOptionPane.showMessageDialog(null, "Bravo, vous avez deviné !");  
} else {  
    JOptionPane.showMessageDialog(null, "Dommage, ce n'était pas le bon  
nombre.");  
}
```

## 7. Test du programme

Le programme me demande d'entrer un nombre entre 1 et 10.



Je rentre 5

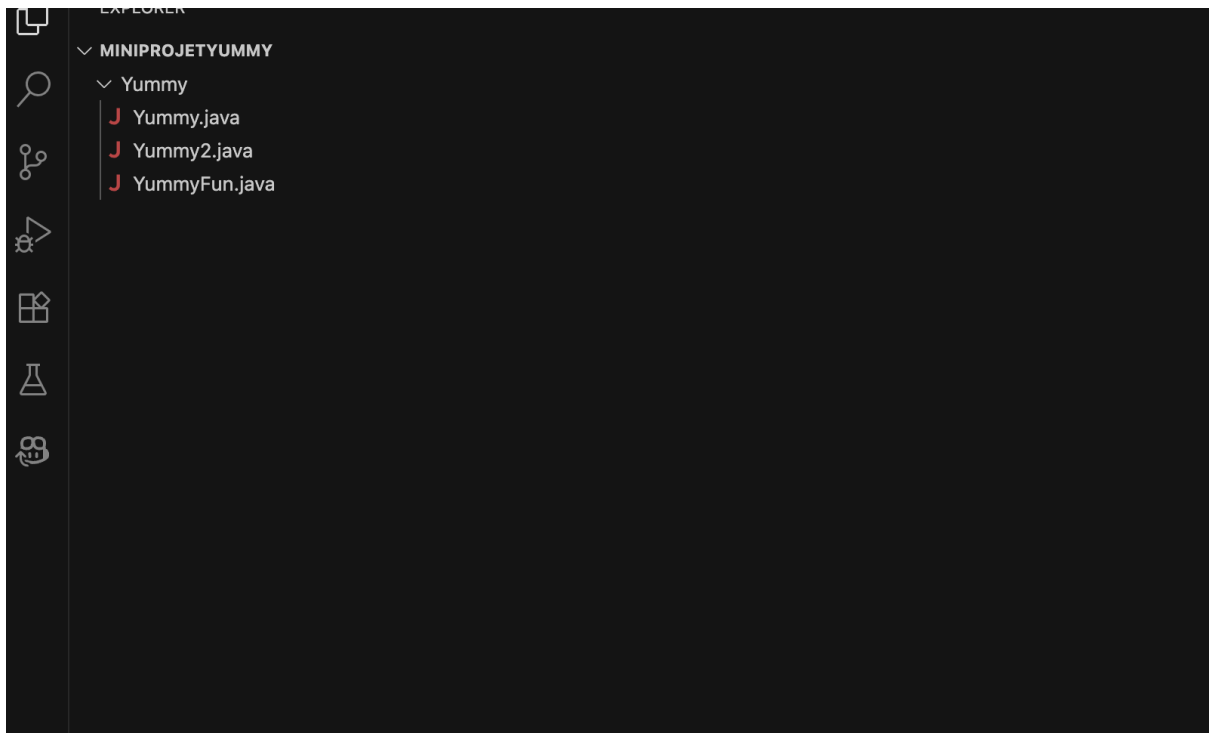


Le nombre généré était 2



## Le cas problème

Nous allons créer un package (des fichiers dans un même dossier) appelé Yummy et y placer trois programmes (Yummy.java, Yummy2.java, YummyFun.java).



### Yummy

Ensuite nous devons écrire une classe Yummy qui affiche la devise :  
“Yummy prépare les meilleurs plats pour vos fêtes”

```
System.out.println("Yummy prépare les meilleurs plats pour vos fêtes");
```

```
Yummy prépare les meilleurs plats pour vos fêtes
```

## Yummy2

Nous avons écrit une classe Yummy2 qui affiche la même devise mais entourée d'une bordure faite d'astérisques.

```
public static void main(String[] args) {  
    String message = "Yummy prépare les meilleurs plats pour vos fêtes";
```

\*\*\*\*\*  
 \* Yummy prépare les meilleurs plats pour vos fêtes \*  
 \*\*\*\*\*

YummyFun

Et enfin nous avons écrit une classe YummyFun qui affiche une devise d'été entourée d'une bordure faite de lettres S.

```
System.out.println(bordure);  
        System.out.println("S " + message + " S");  
        System.out.println(bordure);
```

```

SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
S Yummy, c'est fun au soleil S
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

```

Fin.