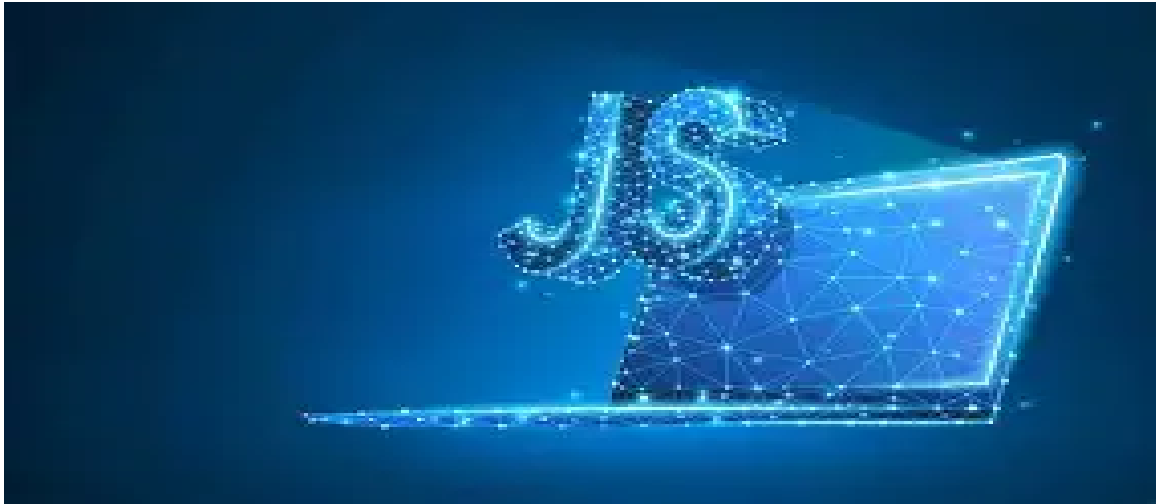


Mes premiers pas en Java : Compte Rendu



Mes premiers pas en Java : Compte Rendu

Introduction

1. Installation du Kit de Développement Java (JDK)
 - 1.1 Qu'est-ce que le JDK ?
 - 1.2 Téléchargement et Installation du JDK
2. Installation de l'IDE Eclipse
 - 2.1 Qu'est-ce qu'Eclipse ?
 - 2.2 Téléchargement et Installation d'Eclipse
3. Création du Premier Programme Java
 - 3.1 Création d'un Nouveau Projet
 - 3.2 Création d'une Classe Main
 - 3.3 Écriture du Code Java
 - 3.4 Utilisation de la Classe Scanner pour la Saisie Utilisateur
4. Exécution du Programme
 - 4.1 Lancer le Programme dans Eclipse
5. Synthèse des Concepts Appris
 - 5.1 Correspondance entre l'Algorithme et le Code Java
 - 5.2 Utilisation de Types Primitifs en Java

Conclusion

5. Synthèse des Concepts Appris
 - 5.1 Correspondance entre l'Algorithme et le Code Java
 - 5.2 Utilisation de Types Primitifs en Java

Conclusion

Introduction

L'apprentissage de la programmation en **Java** est une étape cruciale pour quiconque désire se lancer dans le développement logiciel. Java, en tant que langage de programmation orienté objet et multiplateforme, est largement utilisé pour le développement d'applications de grande envergure, qu'elles soient de bureau, mobiles, ou web. Ce compte rendu décrit les étapes essentielles pour démarrer avec Java, allant de l'installation des outils nécessaires à la création et à l'exécution d'un premier programme simple. Ce guide s'adresse à un débutant, visant à fournir une vue d'ensemble complète et structurée de la mise en place d'un environnement de développement et du processus de codage en Java.

1. Installation du Kit de Développement Java (JDK)

1.1 Qu'est-ce que le JDK ?

Le **JDK** (Java Development Kit) est une collection d'outils nécessaire pour développer des applications Java. Il contient un **compilateur Java**, une **Machine Virtuelle Java (JVM)** et des bibliothèques essentielles. Le JDK permet de compiler le code source Java en un fichier exécutable, qui peut ensuite être exécuté par la JVM sur différentes plateformes.

1.2 Téléchargement et Installation du JDK

Depuis 2019, Oracle a changé sa politique tarifaire et ne propose plus de JDK gratuit pour un usage commercial. Toutefois, des alternatives libres sont disponibles, notamment :

- **OpenJDK** : Le projet source de Java.
- **AdoptOpenJDK** (maintenant connu sous le nom d'**Adoptium**) : Une version compilée et prête à l'emploi, compatible avec tous les systèmes d'exploitation (Windows, Linux, Mac).

L'installation d'**AdoptOpenJDK** se fait via le site officiel, où l'on choisit la version souhaitée (OpenJDK 8 ou 11, par exemple) et suit les étapes d'installation en laissant les paramètres par défaut.

2. Installation de l'IDE Eclipse

2.1 Qu'est-ce qu'Eclipse ?

Eclipse est un environnement de développement intégré (IDE) très populaire pour le développement en Java. Il facilite l'écriture, la gestion et l'exécution des programmes Java grâce à ses fonctionnalités avancées (complétion automatique, débogage, gestion de projet, etc.).

2.2 Téléchargement et Installation d'Eclipse

Pour installer Eclipse, il suffit de télécharger la version **Eclipse IDE for Java Developers** depuis le site officiel. Une fois téléchargé, il faut suivre les instructions pour l'installation. Il est essentiel de choisir l'**option "Java for Developer"** lors de l'installation pour assurer une compatibilité maximale avec le développement Java.

3. Création du Premier Programme Java

3.1 Création d'un Nouveau Projet

Une fois Eclipse installé, voilà ce qu'il faut faire :

1. **File > New > Java Project.**
2. Choisir un nom pour le projet et cliquer sur **Finish**.

Cette étape crée un répertoire de travail où on pourra organiser les classes Java.

3.2 Création d'une Classe `Main`

En Java, chaque programme doit contenir une classe avec la méthode `main()` (point d'entrée du programme). Pour ce faire :

1. Clique droit sur ton projet dans l'Explorateur de projets.
2. Sélectionne **New > Class** et donne un nom à ta classe, généralement `Main`.
3. Coche la case **public static void main(String[] args)** pour générer automatiquement la méthode principale.

3.3 Écriture du Code Java

Voici un exemple de programme très simple :

```
public class Main {  
    public static void main(String[] args) {  
        // Affichage d'un message  
        System.out.println("Hello, World!");  
  
        // Déclaration d'une variable et calcul  
        int montant = 100;  
        float TTC = montant * 1.2f;  
    }  
}
```

```
        // Affichage du résultat
        System.out.println("Le montant TTC est de:
" + TTC);
    }
}
```

Explication :

- La méthode **main()** est le point d'entrée du programme.
- **System.out.println()** est utilisée pour afficher des messages dans la console.
- Les variables comme **montant** et **TTC** sont déclarées et utilisées dans des calculs.

3.4 Utilisation de la Classe **Scanner** pour la Saisie Utilisateur

En Java, pour permettre à l'utilisateur d'entrer des données, tu dois utiliser la classe **Scanner**. Voici comment :

1.Importer la bibliothèque :

```
import java.util.Scanner;
```

2.Créer un objet **Scanner** et récupérer une entrée de l'utilisateur :

```
Scanner sc = new Scanner(System.in);
int nombre = sc.nextInt(); // Saisie d'un entier
```

4. Exécution du Programme

4.1 Lancer le Programme dans Eclipse

Une fois ton code écrit, tu peux l'exécuter en cliquant sur l'icône Run (le triangle vert) située dans la barre d'outils. L'exécution du programme affichera les résultats dans la console d'Eclipse.

5. Synthèse des Concepts Appris

5.1 Correspondance entre l'Algorithme et le Code Java

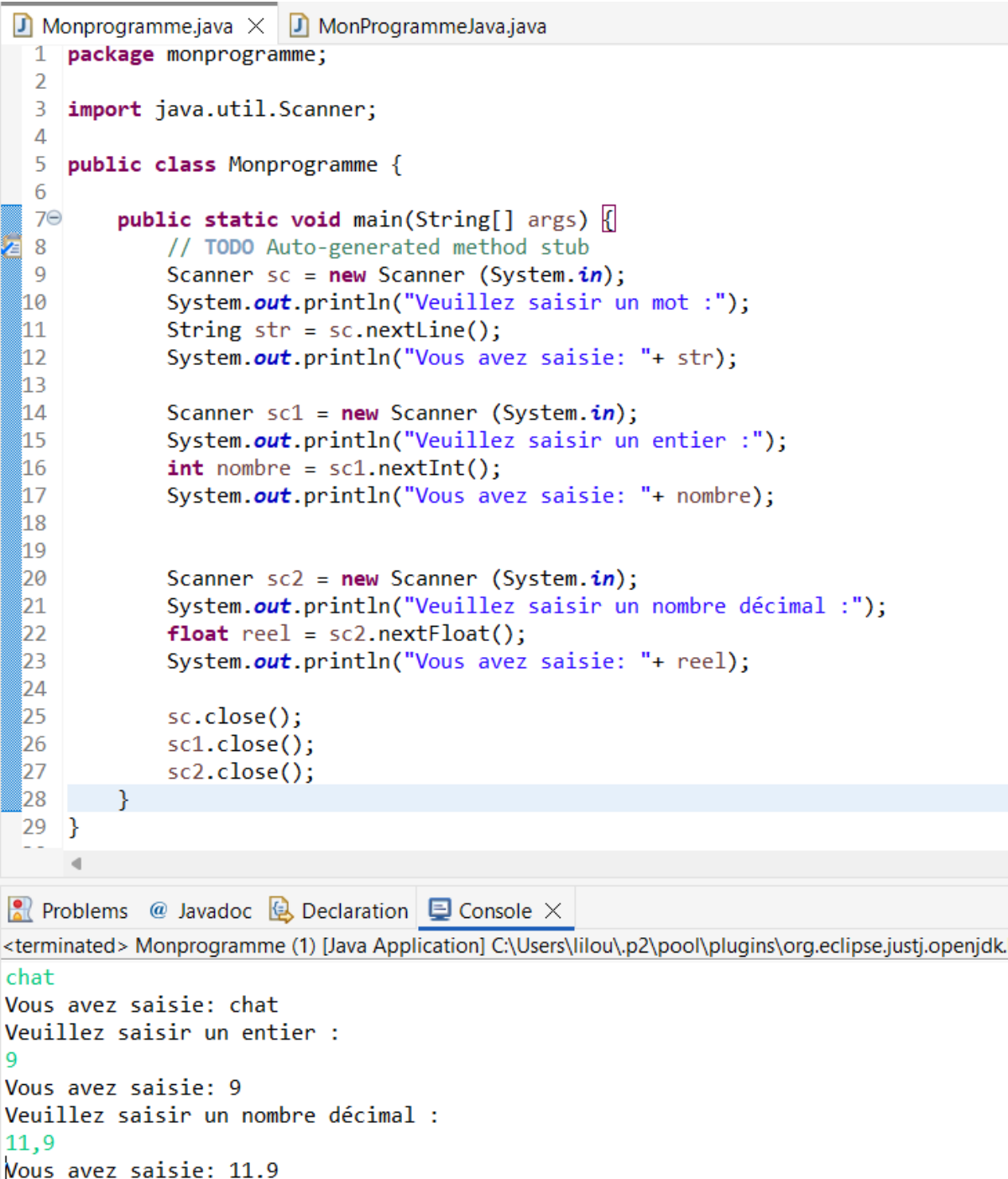
| Algorithme | Java |
|----------------------------|----------------------------------------------------------------|
| Déclaration d'une variable | <code>int montant;</code> |
| Affectation | <code>montant = 100;</code> |
| Affichage | <code>System.out.println("Le montant est: " + montant);</code> |
| Saisie utilisateur | <code>Scanner sc = new Scanner(System.in);</code> |

5.2 Utilisation de Types Primitifs en Java

En Java, les types primitifs les plus courants sont :

- **int** pour les entiers.
- **float** pour les décimaux.
- **String** pour les chaînes de caractères.

Ces types sont utilisés pour effectuer des calculs ou afficher des résultats.



The screenshot displays the Eclipse IDE interface. The top editor shows a Java file named `MonProgrammeJava.java` with the following code:

```
1 package monprogramme;
2
3 import java.util.Scanner;
4
5 public class Monprogramme {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Scanner sc = new Scanner (System.in);
10        System.out.println("Veuillez saisir un mot :");
11        String str = sc.nextLine();
12        System.out.println("Vous avez saisi: " + str);
13
14        Scanner sc1 = new Scanner (System.in);
15        System.out.println("Veuillez saisir un entier :");
16        int nombre = sc1.nextInt();
17        System.out.println("Vous avez saisi: " + nombre);
18
19
20        Scanner sc2 = new Scanner (System.in);
21        System.out.println("Veuillez saisir un nombre décimal :");
22        float reel = sc2.nextFloat();
23        System.out.println("Vous avez saisi: " + reel);
24
25        sc.close();
26        sc1.close();
27        sc2.close();
28    }
29 }
```

The bottom console window shows the execution output:

```
<terminated> Monprogramme (1) [Java Application] C:\Users\lilou\p2\pool\plugins\org.eclipse.justj.openjdk.l
chat
Vous avez saisi: chat
Veuillez saisir un entier :
9
Vous avez saisi: 9
Veuillez saisir un nombre décimal :
11,9
Vous avez saisi: 11.9
```

Conclusion

L'apprentissage des bases de Java est une étape importante pour tout développeur débutant. Ce compte rendu a permis de comprendre comment installer le JDK et Eclipse, créer un projet Java, et écrire un premier programme fonctionnel en utilisant des concepts de base tels que la déclaration de variables, les opérations arithmétiques et l'affichage à la console. La classe Scanner permet en outre d'interagir avec l'utilisateur en saisissant des données. Ces premiers pas en Java ouvrent la voie à l'exploration de concepts plus avancés, tels que la programmation orientée objet, les bibliothèques Java et la gestion des exceptions. À travers cette expérience, il est possible de saisir les fondamentaux nécessaires pour le développement d'applications robustes et efficaces.

5. Synthèse des Concepts Appris

5.1 Correspondance entre l'Algorithme et le Code Java

| Algorithme | Java |
|----------------------------|----------------------------------------------------------------|
| Déclaration d'une variable | <code>int montant;</code> |
| Affectation | <code>montant = 100;</code> |
| Affichage | <code>System.out.println("Le montant est: " + montant);</code> |
| Saisie utilisateur | <code>Scanner sc = new Scanner(System.in);</code> |

5.2 Utilisation de Types Primitifs en Java

En Java, les types primitifs les plus courants sont :

- **int** pour les entiers.

- **float** pour les décimaux.
- **String** pour les chaînes de caractères.

Ces types sont utilisés pour effectuer des calculs ou afficher des résultats.

Conclusion

L'apprentissage des **bases de Java** est une étape importante pour tout développeur débutant. Ce compte rendu a permis de comprendre comment installer le JDK et Eclipse, créer un projet Java, et écrire un premier programme fonctionnel en utilisant des concepts de base tels que la déclaration de variables, les opérations arithmétiques et l'affichage à la console. La **classe Scanner** permet en outre d'interagir avec l'utilisateur en saisissant des données. Ces premiers pas en Java ouvrent la voie à l'exploration de concepts plus avancés, tels que la programmation orientée objet, les bibliothèques Java et la gestion des exceptions. À travers cette expérience, il est possible de saisir les fondamentaux nécessaires pour le développement d'applications robustes et efficaces.
