



INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL

DEPARTEMENT TECHNOLOGIE DE L'INFORMATIQUE

TPO : INSTALLATION DE L'ENVIRONNEMENT DE DEVELOPPEMENT EN JAVA

Matière : Atelier POO

Enseignante : Sonia

Guerbouj Classes : DSI2, MDW2, SEM2

Durée : 30min

Objectif

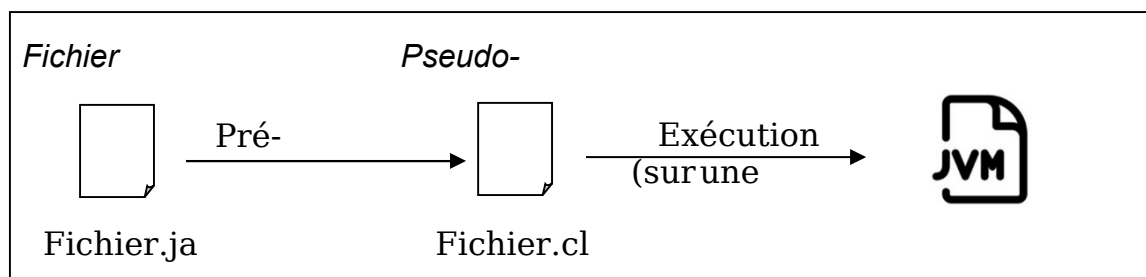
Le but de ce TP est de présenter le langage Java aux étudiants et leur apprendre l'installation de l'environnement nécessaire au développement en Java : Java Kit Development ainsi que la configuration des variables d'environnement.

Introduction

Java est un langage de programmation orienté objet. Il a été créé en 1995 par James Gosling un ingénieur dans Sun Microsystems qui a été rachetée par la société Oracle en 2009.

Java possède plusieurs avantages :

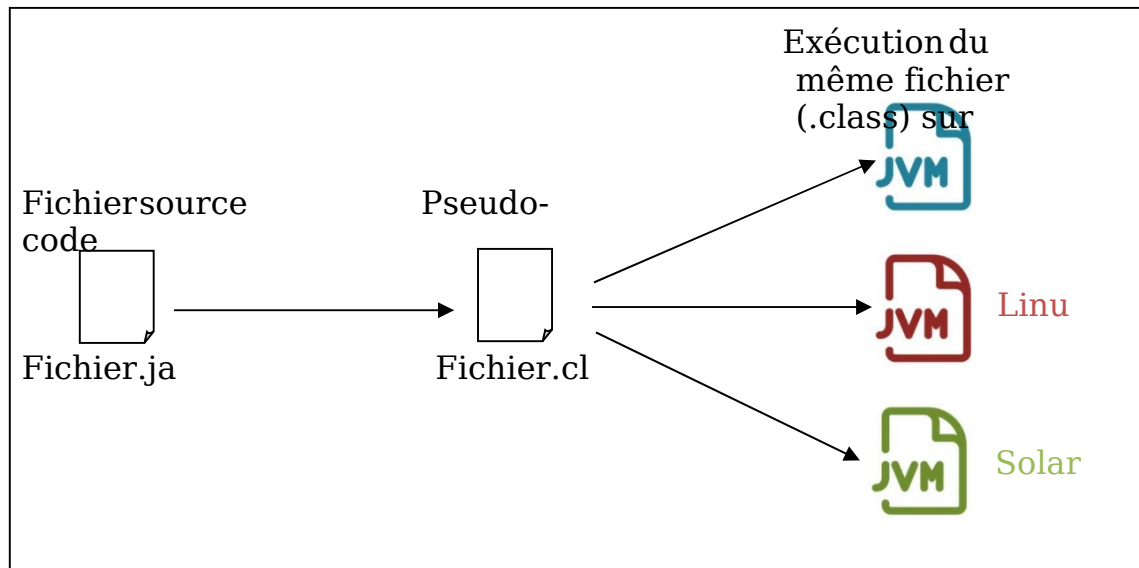
- **Simple** : Sa syntaxe est très proche des langages C et C++.
- **Orienté objet** : Il utilise les notions de classes et objets.
- **Interprété** : Un fichier source est simplement précompilé pour générer un pseudo code qui sera interprété par une machine virtuelle



lors de l'exécution.

- **Dynamique et performant** : Le chargement des classes se fait en cours d'exécution et à la demande. Sans oublier que Java n'est pas gourmand en mémoire, la JVM a besoin de quelques centaines de kilo octets pour s'exécuter.
- **Portable** : Java est indépendant du matériel ou de la plateforme car dès qu'on a un pseudo code (.class) il est exécutable sur n'importe


quel système d'exploitation disposant d'une JVM (Java Virtual Machine)



- **Robuste et sécurisé** : Il intègre plusieurs mécanismes de sécurité et d'autres qui rendent les applications plus fiables tels qu'un gestionnaire d'exception et un ramasse miette. En plus, il ne permet pas d'accès direct à la mémoire par des pointeurs (donc moins d'erreurs).
- **Riche et distribué** : Il contient plusieurs bibliothèques qui couvrent tous les domaines d'utilisation (Réseau, Système, Graphique, ...) et en particulier, il a été conçu pour être exploité dans des environnements serveur et distribués (Java RMI, Socket, Servlet, ...).
- **Multithread** : Java a la capacité d'effectuer plusieurs tâches simultanément. La programmation multithread est bien intégrée.
- **Intégré au Web** : Il est possible d'ajouter des applets Java dans des pages HTML.

Il existe de nombreux frameworks et API permettant l'utilisation de Java dans des domaines diversifiés. On distingue essentiellement 3 grandes éditions :

- **Java SE (Standard Edition)** : Destiné aux applications pour poste de travail.
- **Java EE (Entreprise Edition)** : Spécialisé dans les applications serveurs. Il contient un plus grand nombre d'API et d'extensions.
- **Java ME (Mobile Edition)** : Dédié pour les applications mobiles.

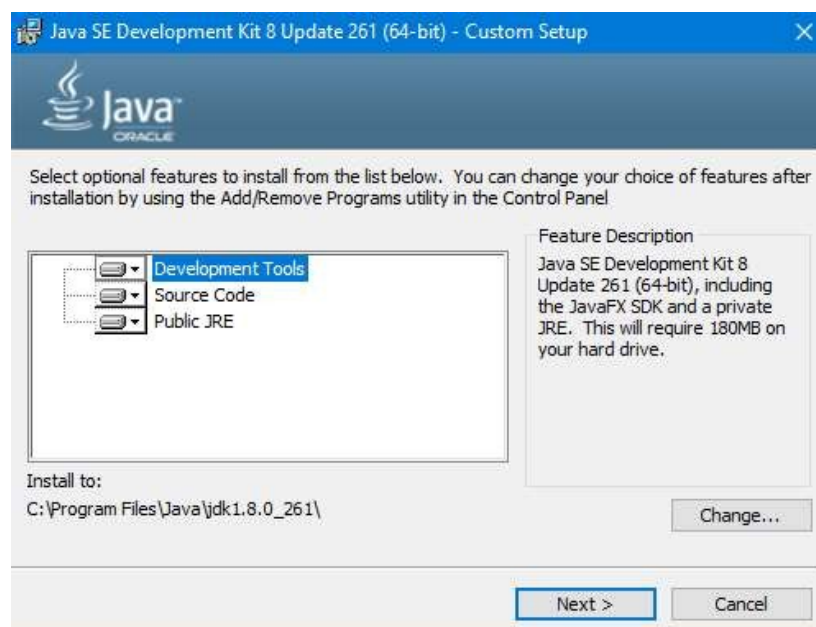
 Pour coder en Java, il faut disposer d'un environnement de développement composé d'un **JDK** (Java Development Kit) et d'un **JRE** (Java Runtime Environment). Le JDK offre un ensemble d'outils y compris le compilateur et interpréteur Java. Le JRE ou environnement d'exécution contient la **JVM** ou *machine virtuelle Java*.

Etape 1 : Téléchargement du JDK

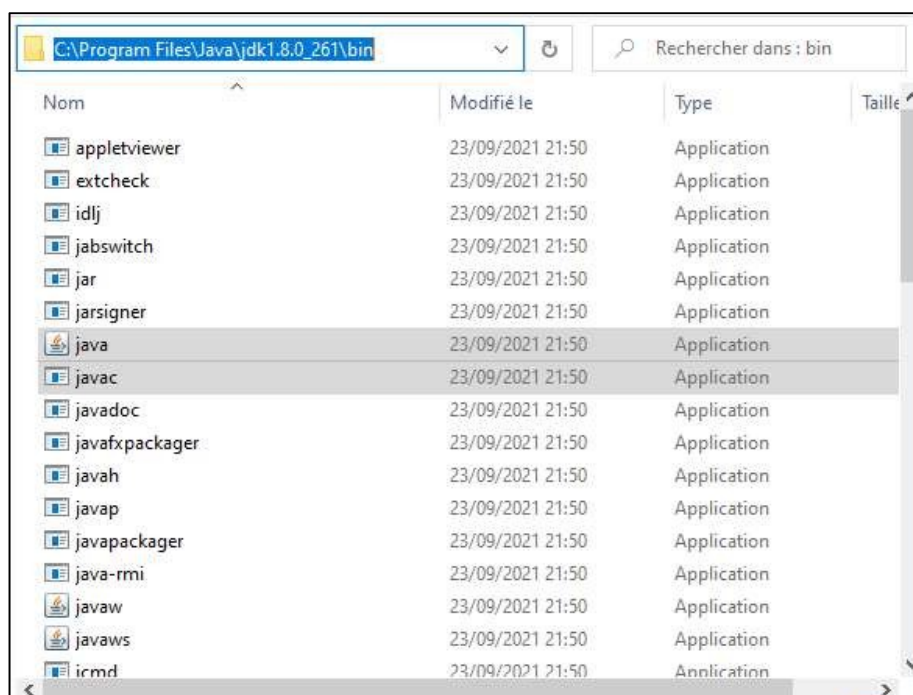
Le JDK est téléchargeable gratuitement sur le site officiel d'Oracle. La dernière version en date est le JDK 17 qui est disponible sur diverses plateformes (Linux, MacOS, Solaris et Windows). Lien : <https://www.oracle.com/java/technologies/downloads/>

Etape 2 : Installation du JDK

Nous avons choisi d'installer la version 8 du JDK étant données nos besoins minimes et à cause de la taille du fichier à télécharger (seulement 170Mo). Il suffit donc de double cliquer sur le fichier « jdk-8u301-windows-x64.exe » et suivre l'assistant d'installation.

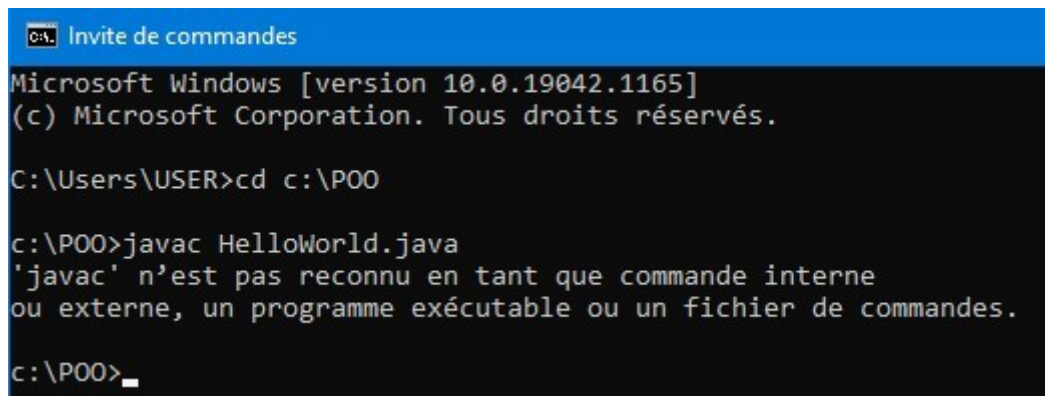


A la fin, il est possible de vérifier les différents outils disponibles dans le JDK dans le dossier « C:\Program Files\Java\jdk1.8.0_261\bin ». Celles dont on aura le plus besoin sont « javac » pour la compilation et « java » pour l'exécution.



Etape 3 : Configuration des variables d'environnement

Lorsqu'on écrit un programme en java et qu'on désire le compiler en mode commande, l'éditeur de commande ne reconnaît pas les commandes javac et java. Il faut donc configurer la variable d'environnement « Path ».

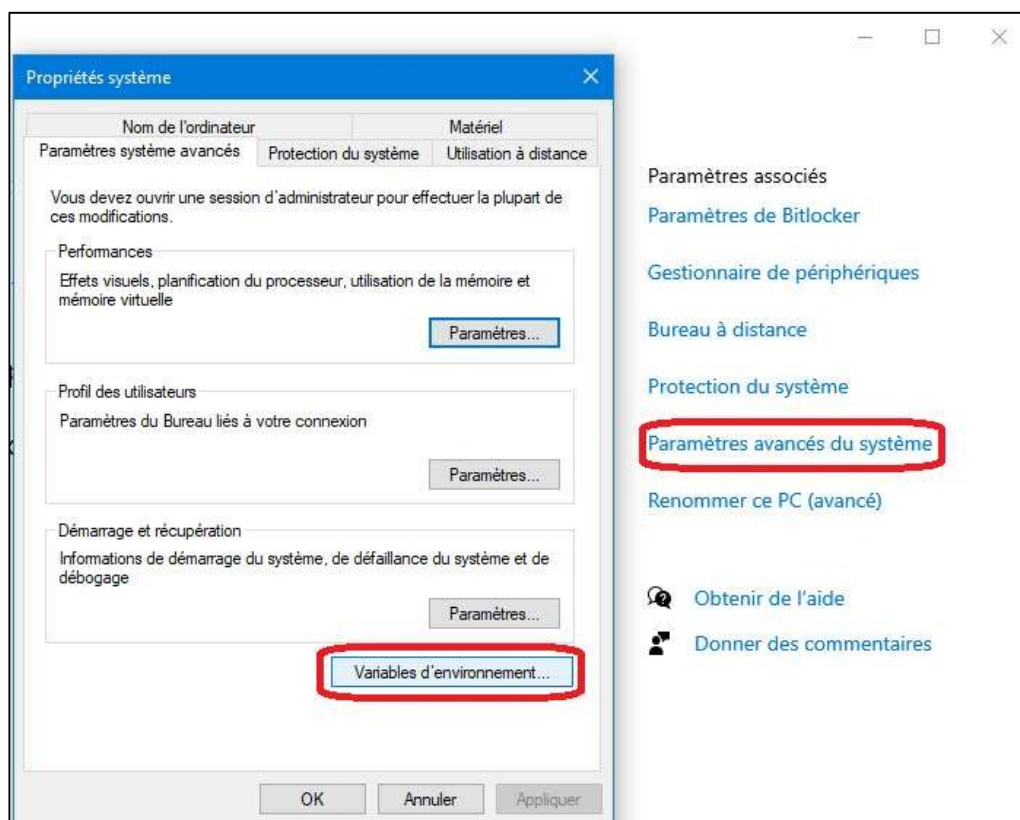


```
C:\Users\USER>cd c:\P00

c:\P00>javac HelloWorld.java
'javac' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

c:\P00>_
```

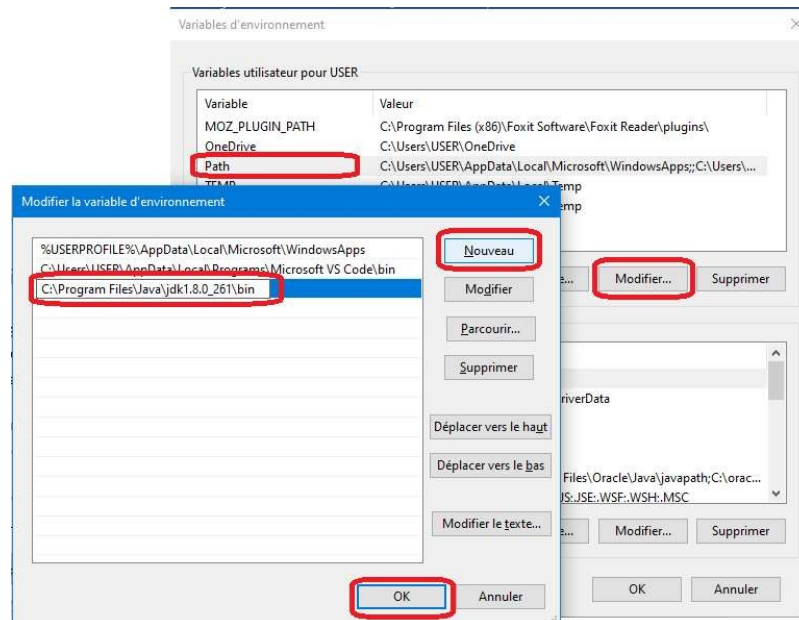
Pour accéder aux variables d'environnement sur un ordinateur, il suffit de faire un clic droit sur l'icône « My PC » puis choisir « Propriétés ». Dans la fenêtre qui s'ouvre, choisissez « Paramètre avancés du système ». Dans la fenêtre suivante, cliquez sur le bouton « Variables d'environnement ».



Dans la fenêtre qui suit, il suffit de sélectionner la variable « Path » et cliquer sur le bouton « Modifier ». Puis dans la nouvelle fenêtre, on clique sur le bouton « Nouveau »

pour ajouter un nouveau chemin de commandes, celui des outils fournis par le JDK :

« C:\Program Files\Java\jdk1.8.0_261\bin ». Validez toutes les fenêtres (Bouton OK).



NB : Cette opération est réalisée une seule fois et pour prendre en considération ces modifications, il faut fermer l'éditeur de commande et le rouvrir de nouveau.



TP 1 : INTRODUCTION À JAVA

Matière : Atelier POO

Enseignante : Sonia

Guerbouj Classes : DSI2, MDW2, SEM2

Durée : 2h30

Objectif

Le but de ce TP est la familiarisation des étudiants avec le langage de programmation Java. Ils apprendront la structure d'un programme en Java, l'utilisation des variables et les commandes nécessaires à la compilation et exécution d'un programme.

Exercice 1 :

- 1) Ouvrez un éditeur de texte (Bloc Note, Edit Plus ou Sublime Text) et créez sous "**c:\votre_classe**", un nouveau fichier intitulé **Bonjour.java**
Copiez dedans le programme suivant :

```
public class Bonjour {  
    public static void main (String args[]) {  
        //premier programme java  
        System.out.println ("Bonjour ! Voici mon 1er programme Java") ;  
    }  
}
```



- 2) Lancez une session DOS et allez au répertoire "**c:\votre_classe**" pour pouvoir compiler le fichier à l'aide de la commande suivante : **javac Bonjour.java**
- 3) Maintenant, exécutez le programme en tapant la commande : **java Bonjour**. Qu'affiche-t-il ? Identifier ses différentes parties.
- 4) Remplacer "main" par "Main", recompilez et exécutez. Que se passe-t-il ? Pourquoi ?
- 5) A présent, modifiez le code précédent pour ajouter des arguments au main en remplaçant l'instruction d'affichage par :

```
System.out.println ("Bonjour "+args[0]+"! Voici mon 1er programme Java") ;
```

Exécutez en tapant la commande : **java Bonjour votre_nom**.

Exercice 2

- 1) Copiez le programme suivant et enregistrez le sous "c:\votre_classe\Bienvenu.java"

```
import java.util.Scanner;

public classe Bienvenu {
    public static void main (String
        args[]) { Scanner sc = new
        Scanner(System.in);
        System.out.println ("Quel est votre
        prénom ? ") ; String prenom =
        sc.nextLine() ;
        System.out.println ("\nBienvenu " + prenom + " !") ;
    }
```



- 2) Compilez et exécutez le fichier, que se passe-t-il ? Déterminez ce qu'il fait.
- 3) Exécutez en entrant un nombre à la place du prénom. Que se passe-t-il ? Pourquoi ?
- 4) Dans le même fichier, ajoutez des lignes de code en utilisant la méthode « nextInt() » de la même façon que « nextLine() » pour saisir votre âge dans une variable (int age) et l'afficher comme suit :
 Bienvenue Toto
 ! Vous avez 17
 ans.

Exercice 3

Ecrire un programme Java permettant de saisir au clavier le **prix** d'un produit et le pourcentage de la **remise** puis d'afficher le **prix avec remise** sachant que :

$$\text{prix_ar} = \text{prix} - \text{prix} * \text{remise} / 100$$

Exercice 4

- 1) Recopiez le programme de l'exercice 2 et enregistrez le dans un nouveau fichier intitulé "MaClasse.java". Puis organisez le programme comme suit, en ajoutant une méthode afficher() :

```
import java.util.Scanner;

public class
    MaClasse { public
    void afficher() {
        // A compléter : saisir le nom et l'âge puis afficher le message de
        bienvenu
        ... }

    public static void main (String args[]) {
        MaClasse m = new MaClasse(); // créer un objet de la classe
        m.afficher() // faire appel à la méthode
        afficher
    }
```

- 2) Compilez et exécutez le fichier.
- 3) A présent, modifiez la méthode `afficher()` comme suit pour accepter des arguments :
 - Remplacez l'entête de la méthode ainsi : **`public void afficher(int age)`**.
 - Supprimez, dans le corps de la méthode, la partie saisie au clavier uniquement.
- 4) Maintenant ajouter juste avant la méthode `afficher`, une ligne de code permettant de déclarer un attribut **`prenom`** de type entier initialisé par votre prénom.
- 5) Compilez et exécutez le programme obtenu. Qu'affiche-t-il ?