

CHAPITRE 2 Langage Java

1. Introduction

Java un langage facile, relativement récent...pas de pointeurs

2. Installer Les Outils De Développement

2.1. Machine virtuelle et JRE

L'un des principes phares de Java réside dans sa machine virtuelle : celle-ci assure à tous les développeurs Java qu'un programme sera utilisable avec tous les systèmes d'exploitation sur lesquels est installée une machine virtuelle Java.

Lors de la phase de compilation du code source, celui-ci prend une forme intermédiaire appelée **byte code** : interprétable par la machine virtuelle Java. Cette dernière porte un nom : on parle plus de **JRE** (Java Runtime Environment).

2.2. Editeur de texte vs. IDE

Nous pouvons saisir le code source dans un éditeur de texte puis le compiler sous Ms-DOS à l'aide d'une instruction java Mais il est préférable d'utiliser un outil de développement, ou **IDE** (Integrated Development Environment) pour la saisie et l'exécution, Exemple : **Eclipse** ou **Netbeans**.

2.3. JRE ou JDK

Commencer par télécharger l'environnement Java sur le site **d'Oracle**. Choisir la dernière version stable. Il ya 2 environnements :

1. **JRE** (Java Runtime Environment) contient tout le nécessaire pour que vos programmes Java puissent être exécutés sur votre ordinateur ;
2. **JDK** (Java Development Kit), en plus de contenir le JRE, contient tout le nécessaire pour développer, compiler...

Il y a plusieurs catégories de JDK :

- a) **J2SE** (*Java 2 Standard Edition*) : pour développer des applications.
- b) **J2EE** (*Java 2 Enterprise Edition*) : pour des applications **web**.
- c) **J2ME** (*Java 2 Micro Edition*) : **pour appareils portables**, comme des téléphones portables.

2.4. Fichiers Java

- Les fichiers contenant le code source de programmes Java ont l'extension **.java**.
- Les fichiers précompilés correspondant à codes source Java ont l'extension **.class**.
- Un programme Java, codé sous Windows, peut être précompilé sous Mac et enfin exécuté sous Linux.
- La machine NE PEUT PAS comprendre le byte code, elle a besoin de la machine virtuelle (JRE).

2.5. PATH et CLASSPATH

A l'étape suivante de l'installation, on doit définir 2 variables système. Dans Windows, dans le *Panneau de configuration* ou dans le menu *Paramètres*, sur l'icône *Système|Avancé|Variables d'environnement*, on modifie la variable PATH pour ajouter « ; »+<le chemin de JDK>.


```
boolean question=true;
```

3.2.4. **Type String** L'utilisation n'est pas comme en c.

```
//1 méthode
```

```
String phrase;
```

```
phrase = "bonjour maman";
```

```
//2ème méthode
```

```
String str = new String();
```

```
str = "Une autre chaîne de caractères";
```

```
//3ème méthode
```

```
String string = "Une autre chaîne";
```

```
//4ème méthode
```

Il y a une convention de nommage. En fait, c'est une façon d'appeler nos classes, nos variables, etc. Il faut que vous essayiez de la respecter au maximum. Cette convention, la voici :

- tous vos noms de **classes** doivent commencer par une **majuscule** ;
- tous vos noms de **variables** doivent commencer par une **minuscule** ;
- si le nom d'une variable est composé de plusieurs mots, le premier commence par une minuscule, le ou les autres par une majuscule, et ce, sans séparation ;

```
String chaine = new String("Et une de plus !");
```

3.3. Opérateurs arithmétiques

+, -, *, /, %, ++, +=, --, -= comme en c

« + » : permet d'additionner deux variables numériques mais aussi de concaténer des chaînes de caractères).

Quelques exemples de calcul :

```
int nbre1, nbre2, nbre3; //Déclaration des variables
```

```
nbre1 = 1 + 3; //nbre1 vaut 4
```

```
nbre2 = 2 * 6; //nbre2 vaut 12
```

```
nbre3 = nbre2 / nbre1; //nbre3 vaut 3
```

```
nbre1 = 5 % 2; //nbre1 vaut 1, car 5 = 2 * 2 + 1
```

```
nbre1 = nbre1 + 1;
```

```
nbre1 += 1;
```

3.4. Conversions, ou « cast »

Java est fortement typique , on ne peut utiliser un opérateur sur deux types différents.

a) De double à int

```
double nbre1 = 10, nbre2 = 3;
```

```
int resultat = nbre1 / nbre2; //impossible
```

```
int resultat = (int)(nbre1 / nbre2); //affiche 3
```

```
System.out.println("Le résultat est = " + resultat); //pour afficher
```

Sachez aussi qu'on peut tout à fait mettre des opérations dans un affichage :

```
System.out.print("Résultat = " + nbre1/nbre2); //affiche 3
```

b) D'un type int en type float :

```
int i = 123;
```

```
float j = (float)i;
```

c) D'un type int en double :

```
int i = 123;
```

```
double j = (double)i;
```

d) Et inversement :

```
double i = 1.23;
```

```
double j = 2.9999999;
```

```
int k = (int)i; //k vaut 1
```

```
k = (int)j; //k vaut 2
```

e) Priorité d'opération

si nous déclarons deux entiers et que nous mettons le résultat dans un double ?:

```
int nbre1 = 3, nbre2 = 2;
```

```
double resultat1 = nbre1 / nbre2; // résultat 1 au lieu de 1,5
```

```
double resultat2 = (double)(nbre1 / nbre2); // résultat 1
```

```
double resultat3 = (double)(nbre1) / (double)(nbre2); // résultat 1,5
```

```
System.out.println("Le résultat est = " + resultat);
```

Pour avoir un résultat correct, il faudrait caster chaque nombre avant de faire l'opération

f) Du numérique vers String

```
int i = 12;
```

```
String j = new String(i);
```

USTO-MB
MI,L2,POO
Dekhici L. 2015
j = j.valueOf(i);

g) Du String vers numérique *intValue()* , *floatValue()*, *doubleValue()*...

```
int i = 12;  
String j = new String();  
j = j.valueOf(i);  
int k = Integer.valueOf(j).intValue();
```

3.5. Lire Entrées Clavier

3.5.1. Classe Scanner

```
Import java.util.Scanner ; //importer la classe scanner
```

```
Scanner sc = new Scanner(System.in); // créer une instance du lecteur in  
System.out.println("Veuillez saisir un mot :");  
String str = sc.nextLine();  
System.out.println("Vous avez saisi : " + str);
```

3.5.2. Lire les types numériques

```
Scanner sc = new Scanner(System.in);  
int i = sc.nextInt();  
double d = sc.nextDouble();  
long l = sc.nextLong();  
byte b = sc.nextByte();  
(sauf les char)
```

Remarque : Il faut utiliser deux *nextLine()* après une lecture des nombres.

3.6. Conditions et boucles

Comme en c.

3.7. Tableaux

3.7.1. Tableau à une dimension

```
<type du tableau> <nom du tableau> [] = { <contenu du tableau>};
```

Exemple :

```
int tableauEntier[] = {0,1,2,3,4,5,6,7,8,9};  
double tableauDouble[] = {0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0};  
char tableauCaractere[] = {'a','b','c','d','e','f','g'};  
String tableauChaine[] = {"chaine1", "chaine2", "chaine3", "chaine4"};
```

un tableau vide de six entiers :

```
int tableauEntier[] = new int[6];
```

Ou encore

```
int[] tableauEntier2 = new int[6];
```

3.7.2. tableaux multidimensionnels

```
int premiersNombres[][] = { {0,2,4,6,8},{1,3,5,7,9} };
```

3.7.3. Utiliser et rechercher dans un tableau

Indice est mis entre crochet :

```
System.out.println(tableauCaractere[0]);
```

Le parcours jusqu'à sa longueur **length** :

```
for(int i = 0; i < tableauCaractere.length; i++)  
{  
    //traitement  
}
```

3.7.4. nouvelle syntaxe de boucle sur les tableaux

```
String auxiliaire[] = {"am", "is", "are"};
```

```
for(String mot : auxiliaire)  
    System.out.println(mot);
```

3.8. MÉTHODES DE CLASSE

3.8.1. Pour String

```
String chaine = new String("coucou ! ");
```

```
int longueur = chaine.length(); //Renvoie 9
```

La méthode equals() permet de vérifier (donc de tester) si deux chaînes de caractères sont identiques. C'est avec cette fonction que vous effectuerez vos tests de condition sur les String. Exemple concret :

```
String str1 = new String("coucou"), str2 = new String("toutou");
```

```
if (str1.equals(str2))  
    System.out.println("Les deux chaînes sont identiques !");
```

```
else  
    System.out.println("Les deux chaînes sont différentes !");  
String nbre = new String("1234567");
```

USTO-MB
MI,L2,POO
Dekhici L. 2015

```
char carac = nbre.charAt(4); //Renverra ici le caractère 5
```

3.8.2. Pour Math

```
double X = Math.random();// un nombre aléatoire entre 0 et 1
```

```
double abs = Math.abs(-120.25); //La fonction valeur absolue
```

```
double d = 2;
```

```
double exp = Math.pow(d, 2); //La fonction puissance
```

```
}
```

Exercice :

Ecrire les programmes qui :

affiche la concaténation de deux chaînes

affiche le produit de deux entiers.

Calcul le reste de la division

Affiche admis ou ajourné après calcul de moyenne.

affiche pour x allant de 1 à 10 la valeur de $1/(x-5)$