



Conception Orientée Objet & Programmation JAVA

Chapitre 1 : Introduction

Année universitaire 2025/2026

Mme Hana Gharbi



Acquis d'apprentissage du module



A la fin du module, l'apprenant doit :

- ✓ Comprendre le paradigme OO et utiliser Java pour le mettre en œuvre
- ✓ Maîtriser la notion de classe et d'objet
- ✓ Décrire les éléments-clé de la plate-forme Java
- ✓ Compiler et exécuter une application Java
- ✓ Décrire la syntaxe du langage
- ✓ Comprendre et utiliser les API de java



PLAN



- **Introduction**
- Classe et Objet
- Encapsulation
- Héritage
- Polymorphisme
- Exceptions
- Interfaces
- Collection
- Interface Fonctionnelle
- Expression Lambda
- Stream

La programmation Orientée Objet





Programmation classique



- ✓ La programmation **classique** telle que étudiée au travers des langages **C, Pascal...** définit un programme comme étant:
 - Un ensemble de **données** sur lesquelles agissent des **procédures** et des **fonctions**.
 - Les données constituent la partie **passive** du programme. Les procédures et les fonctions constituent la partie **active**;
- ✓ Programmer dans ce cas revenait à:
 - Définir un certain **nombre de variables** (structures, tableaux...)
 - Ecrire des procédures pour les manipuler sans associer explicitement les unes aux autres.



P.O.O (Programmation Orientée Objets)



- ✓ La programmation orientée objet se base sur une approche de conception et de développement de logiciels
- ✓ Représenter les éléments du monde réel par des entités informatiques appelés "objets" en adoptant un haut niveau d'abstraction.



P.O.O (Programmation Orientée Objet)



- ✓ Un **objet** est une entité logicielle:
 - Ayant une **identité**
 - Capable de sauvegarder un **état** c'est-à-dire un ensemble d'information dans des variables internes (**attributs**).
 - Répondant à des messages précis en déclenchant des activations internes appropriés qui changent l'état de l'objet (**comportement**).
- ✓ Ces opérations sont appelées **méthodes**. Ce sont des fonctions liées à des objets et qui précisent le comportement de ces objets.
- ✓ **Attributs** : Les attributs d'un objet sont l'ensemble des informations se présentant sous forme de variable et permettant de représenter l'état de l'objet.



P.O.O (Programmation Orientée Objet)



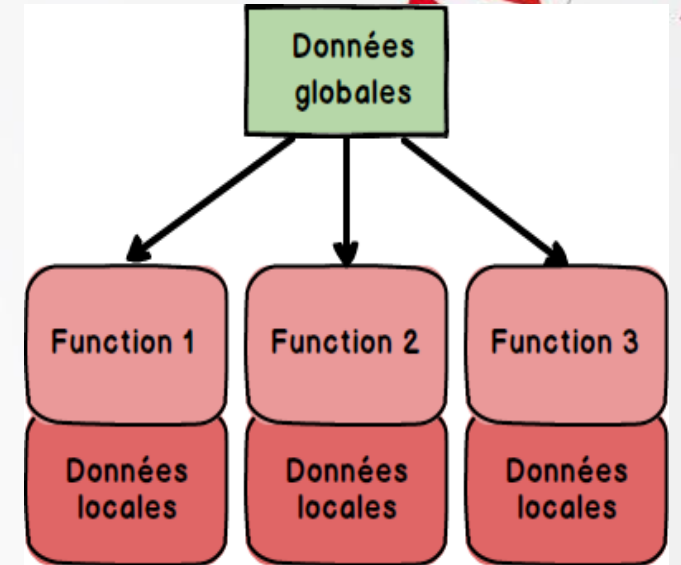
- ✓ Un programme est constitué d'un ensemble d'objets chacun disposant d'une partie procédures et d'une partie données. Les objets interagissent par envoi de messages.
- ✓ Un objet peut recevoir un msg qui déclenche:
 - Une méthode qui modifie son état ou
 - Une méthode qui envoie un msg à un autre objet



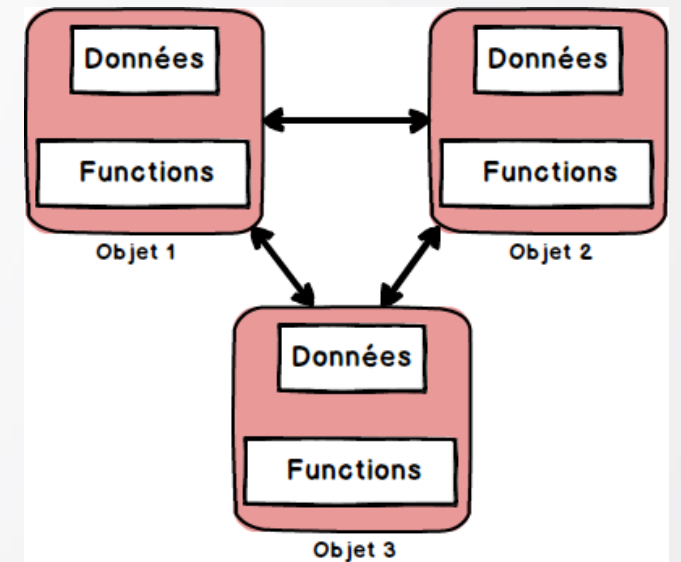


Prog.classique Vs. POO

- ✓ Approche procédurale :
"Que doit faire mon programme ? "



- ✓ Approche orientée-objet :
"De quoi doit être composé mon programme ?"





Langages Procédurales & Orientée Objet



Langages procédurales	Langages Orientée Objet
C	C++
COBOL	PHP >= 5
PHP < 5	Ruby
Pascal	Scala
Visual Basic	JAVA



C'est quoi JAVA





Compilé vs interprété



Compilé

-le code source est soumis à un compilateur, pour en faire un fichier binaire compréhensible par un microprocesseur (une sorte de pré-fichier .exe)

-le code source est, directement, interprété sans phase de compilation, et c'est l'interprète qui exécute ce code source, qu'il interprète à la volée.(JVM)

Interprété

► C'est quoi java?

- ✓ Un langage de programmation orienté objet.
- ✓ Java est portable : il est indépendant de toute plateforme

Quels types d'application pour « java » ?



Java SE



Java ME



Java EE



Java
Embedded



Java
Card



JAVA: Langage de programmation

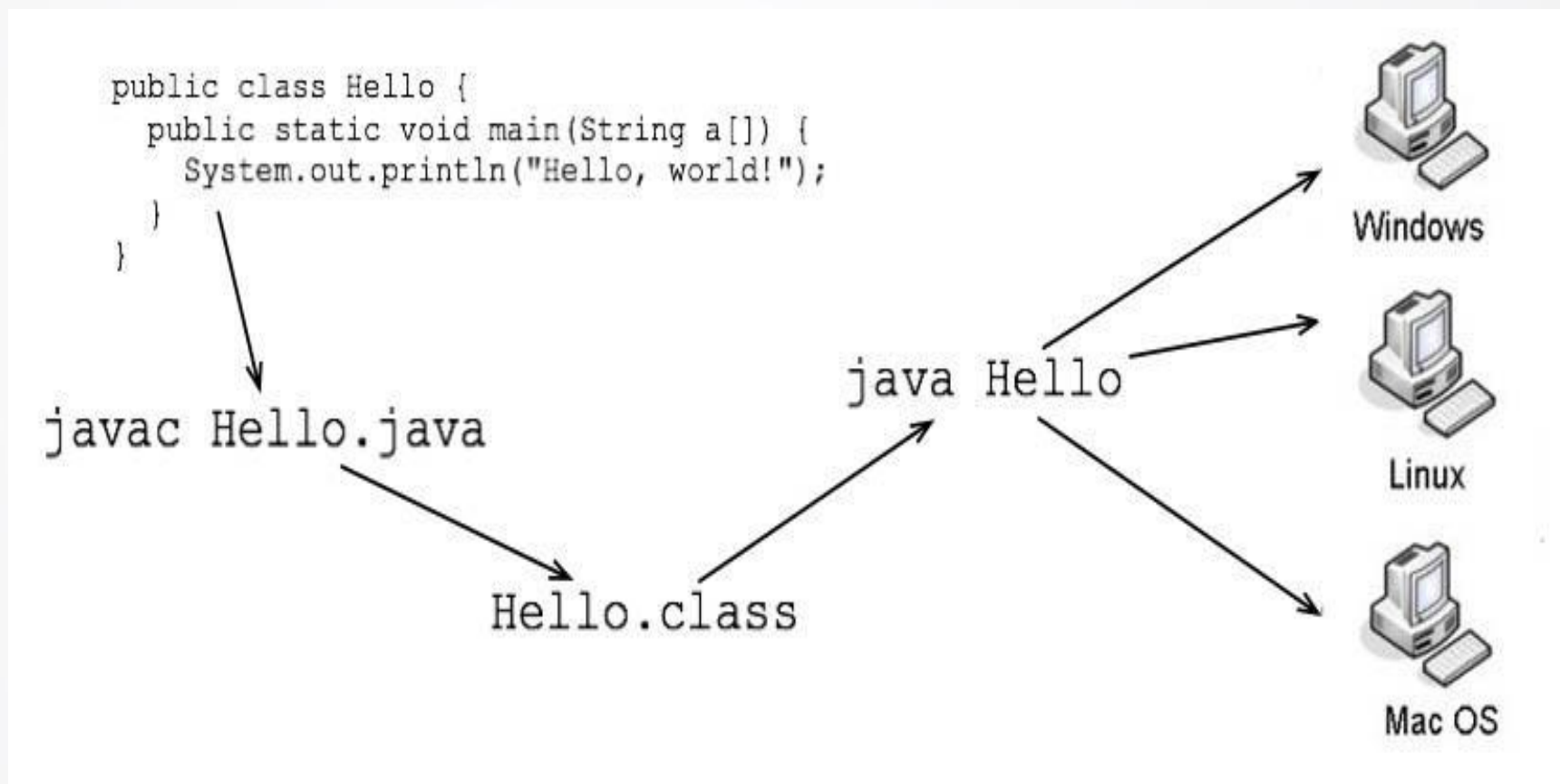


✓ Java est un langage de programmation

- Un programme Java est compilé et interprété
- Simplicité et productivité:
 - Intégration complète de l'OO
 - Gestion mémoire (« Garbage collector »)
- Robustesse, fiabilité et sécurité
- Indépendance par rapport aux plateformes
- Ouverture:
 - Support intégré d'Internet
 - Connexion intégrée aux bases de données (JDBC)
 - Support des caractères internationaux



JAVA: Langage de programmation



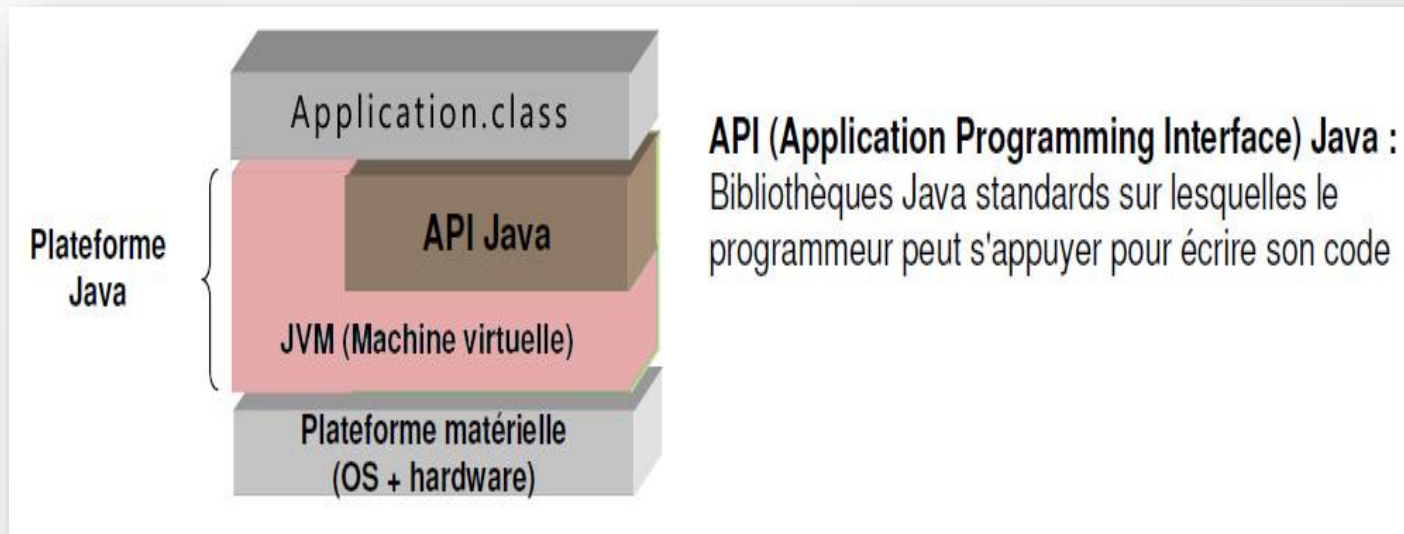


JAVA: Une plateforme



✓ Java est une plateforme

- La plateforme Java, uniquement software, est exécutée sur la plateforme du système d'exploitation
- La « Java Platform » est constituée de :
 - La « Java Virtual Machine » (JVM)
 - Des interfaces de programmation d'application (Java API)





JVM (Java Virtual Machine)



Une machine virtuelle est un ordinateur fictif s'exécutant sur un ordinateur réel :

- Possède un langage natif propre et traduit un programme écrit dans ce langage vers le langage natif de l'ordinateur.
- Définit les spécifications hardware de la plateforme
- Lit le bytecode compilé (indépendant de la plateforme)



API (Application Programming Interface)



- ✓ L'API Java est structuré en librairies (packages).
- ✓ Les packages comprennent des ensembles fonctionnels de composants (classes)..
- ✓ Le noyau (core) de l'API Java (inclus dans toute implémentation complète de la plateforme Java) comprend notamment :
 - Essentials (data types, objects, string, array, vector, I/O,date,...)
 - Applet
 - Abstract Windowing Toolkit (AWT)
 - Basic Networking (URL, Socket –TCP or UDP-,IP)
 - Evolved Networking (Remote Method Invocation)
 - Internationalization
 - Security
 -



Python Vs. Java



Thème	Python	JAVA
Syntaxe	<ul style="list-style-type: none">- Ne requiert pas un point-virgule à la fin de chaque instruction- Utilise l'indentation	<ul style="list-style-type: none">- Un point-virgule à la fin de chaque instruction est obligatoire- Des accolades traditionnelles pour débiter et terminer les blocs
Multi-plateforme	Oui	Oui (JSE, JEE, etc...)
Variables	typé dynamiquement (Pas besoin de spécifier le type)	typé statiquement
Collections	tuple, set , dictionary	list, set , map

Java: Les versions

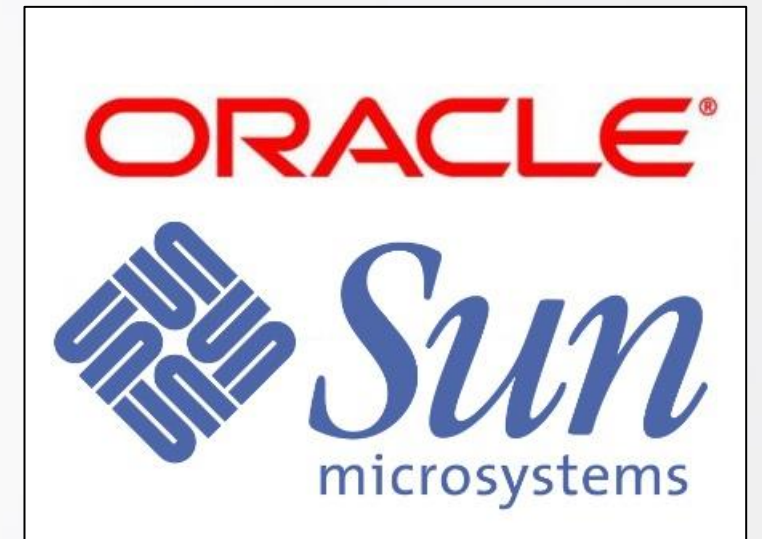




Les milestones

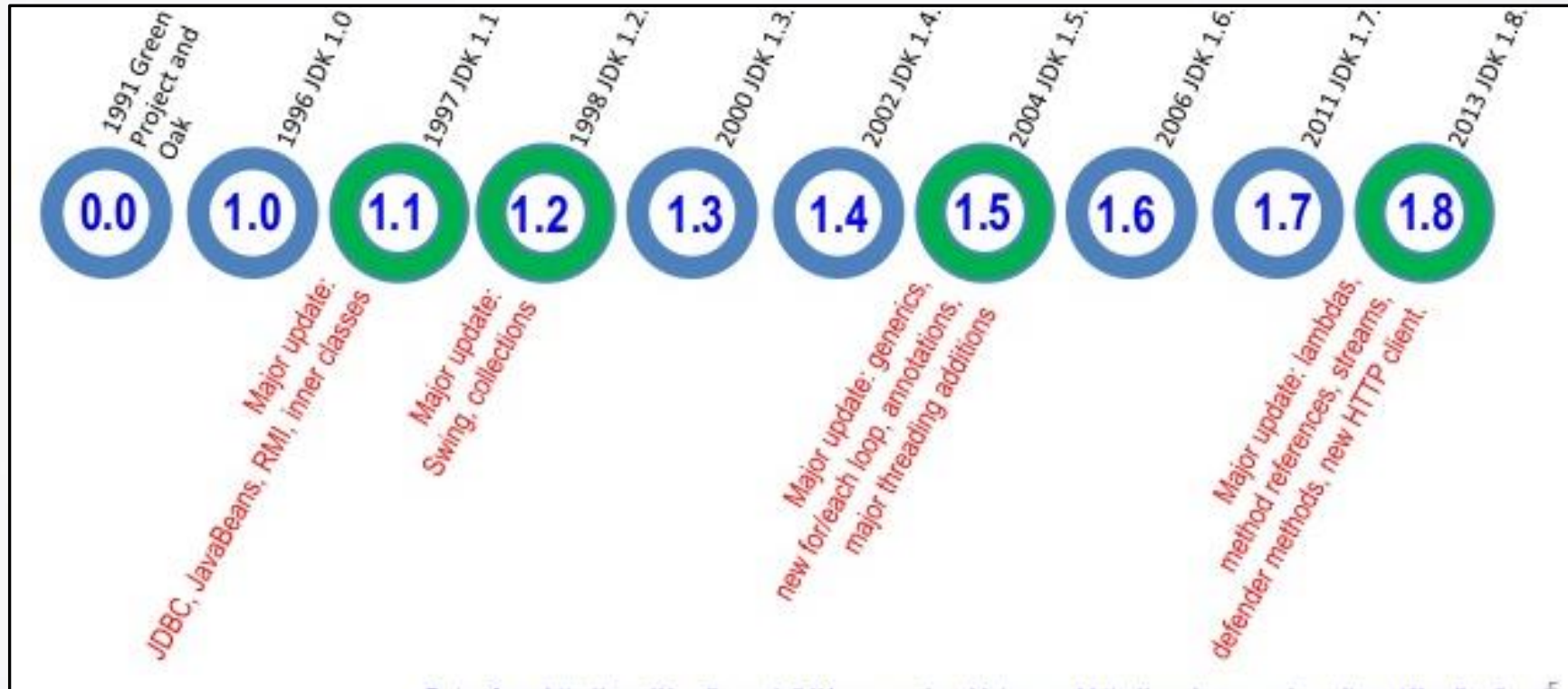


- JAVA est apparue en Mai 1995 avec la version 1.0
- En janvier 2010, Sun a été racheté par Oracle
- Dernière version: 16.0.2





Les versions JAVA



Java: Les notions fondamentales





Les notions de l'Orientée Objet



- Classe / Objet / Instance
- Attributs / Méthodes
- Encapsulation
- Héritage
- Polymorphisme

Une classe



- les briques de constructions d'un programme JAVA sont les classes:

Partie statique

les données (*Propriétés*)

LES ATTRIBUTS :

Les attributs représentent la description des données propres à chaque classe d'objets

Partie dynamique

le code les manipulant
(*Méthodes*)

LES METHODES :

Les méthodes représentent l'ensemble des actions, procédures, fonctions ou opérations que l'on peut associer à une classe

Classe



Les identificateurs



- Nommer les classes, les variables, les méthodes, ...
- Un identificateur Java
 - ✓ est de longueur quelconque
 - ✓ commence par une lettre Unicode
 - ✓ peut ensuite contenir des lettres ou des chiffres ou le caractère souligné«_»
 - ✓ ne doit pas être un mot réservé du langage (mot clé) (if, for, true, ...)
- les caractères suivants sont autorisés pour construire un identificateur Java : "\$" , "_" , "µ" et les lettres accentuées.

`[a..z, A..Z, $, _, µ]{a..z, A..Z, $, _, µ, 0..9,
Unicode}`



Mots clés JAVA²⁷



abstractdouble		int	strictfp **
booleanelse		interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	package	throw
char	for	private	throws
class	goto *	protected	transient
const *	if	public	try
continue	implements	return	void
default	import	short	volatile
do	instanceof	static	while

* Indique un mot clé qui est peu utilisé

** A partir de la plate-forme Java2



Règles de nommage



Classe :

- ✓ 1ère lettre en majuscule
- ✓ Mélange de minuscule, majuscule avec la première lettre de chaque mot en majuscule
- ✓ Donner des noms simples et descriptifs

Packages

- ✓ Tout en minuscule.
- ✓ Utiliser seulement [a-z], [0-9] et le point '.': Ne pas utiliser de tiret '-', d'underscore '_', d'espace, ou d'autres caractères (\$, *, accents, ...).

Constante:

- Les constantes sont en majuscules et les mots sont séparés par le caractère souligné« _ »: UNE_CONSTANTE



Types de données en JAVA



- Java est un langage fortement typé
- Deux types de données:
 - **Donnée primitive**: contient physiquement la valeur (caractère, nombre, booléen)
 - **Référence**: contient l'adresse mémoire où l'information relative à l'objet, l'interface, etc. est réellement stockée



Types primitifs



- **Valeur logique**
 - ✓ boolean (true/false)
- **Nombres entiers**
 - ✓ byte (1 octet)
 - ✓ short (2 octets)
 - ✓ int (4 octets)
 - ✓ long (8 octets)
- **Nombres non entiers (à virgule flottante)**
 - ✓ float (4 octets)
 - ✓ double (8 octets).
- **Caractère (un seul)**
 - ✓ char (2 octets)



Types primitifs



Type	Bits	Bytes	Minimum Range	Maximum Range
byte	8	1	-2^7	$2^7 - 1$
short	16	2	-2^{15}	$2^{15} - 1$
int	32	4	-2^{31}	$2^{31} - 1$
long	64	8	-2^{63}	$2^{63} - 1$
float	32	4	n/a	n/a
double	64	8	n/a	n/a



Types primitifs



Déclaration, Initialisation et Assignment des types primitifs

<code>int t;</code>	Déclaration d'un entier t (t est l'identificateur)
<code>int u = 3;</code>	Déclaration et initialisation d'un entier
<code>t=7;</code>	Initialisation de t à la valeur 7
<code>u=t;</code>	Assignment (affectation) de la valeur de t à u
<code>m=9;</code>	Erreur déclaration préalable nécessaire
<code>char c;</code>	Déclaration
<code>c= 'a' ;</code>	Initialisation



Types primitifs



Exemple:

```
int a = 5;  
int b = 8;
```

Déclaration et initialisation de 2 entiers: a et b

```
a=b;
```

Affectation de la valeur de b à a



Désormais, il existe deux variables en mémoire qui ont la même valeur



La classe String



- String n'est pas un type primitif, c'est une classe
- Déclaration de deux String:

```
String s1, s2;
```

- Initialisation :

```
s1 = "Hello";
```

```
s2 = "le monde";
```

- Déclaration et initialisation :

```
String s3 = "Hello";
```

- Concaténation :

```
String s4 = s1 + " " + s2;
```

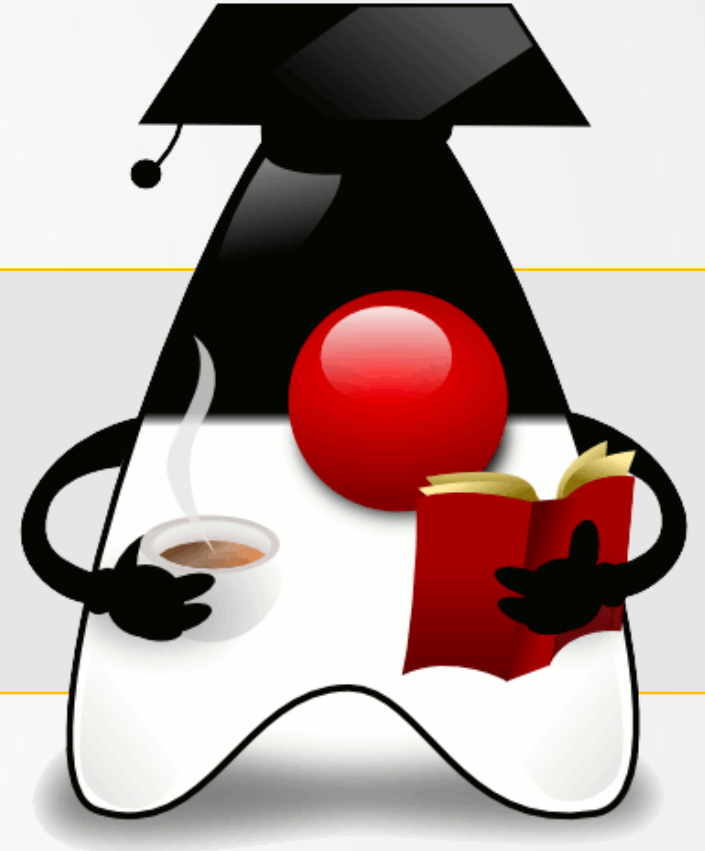


Structures conditionnelles/itératives



- **Schémas conditionnels** : *exécuter une série d'instructions dans le cas où une condition est vraie, et d'exécuter une autre série d'instructions dans le cas contraire*
 - if (même syntaxe qu'en C/C++)
 - switch-case (même syntaxe qu'en C/C++)
- **Schémas itératifs** : *Le traitement itératif est utilisé pour exécuter une ou plusieurs instructions plusieurs fois*
 - for (même syntaxe qu'en C/C++)
 - while (même syntaxe qu'en C/C++)
 - do-while (même syntaxe qu'en C/C++)

Java: Premiers pas...





Outils de développement



- Pour développer un programme JAVA on utilise un **IDE**,
- Afin de pouvoir créer une application JAVA consistante, on a besoin du **JDK**,
- Pour tester et lancer le code écrit, on doit, nécessairement, disposer du **JRE**.



Outils de développement



JRE (Java Runtime Environment)	<p>L'environnement qui permet d'exécuter les applications java,</p> <p>Il est constitué de la JVM en particulier (Java Virtual Machine).</p>
JDK (Java Development Kit)	<p>Outils permettant de développer, compiler (<i>javac</i>), débiter (<i>jdb</i>) et exécuter un programme java.</p> <p>Il y a aussi des outils d'archivage (<i>jar</i>), de génération de documentation (<i>javadoc</i>).</p>



Outils de développement



IDE

(Integrated
Development
Environment)

Un programme regroupant un ensemble d'outils pour le développement de logiciels.

En général, un IDE regroupe un éditeur de texte, un compilateur, des outils automatiques de fabrication, et souvent un débogueur. (Exp: Eclipse, Netbeans)



Pré-configuration



- Installer le JDK sous Windows
 - Télécharger et d'exécuter le programme "jdk-xux-windows-x64/32.exe" depuis le site d'Oracle
(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- Paramétrer correctement la variable PATH (Variables d'environnement)



Premier programme

A screenshot of a Java IDE window titled '*HelloWorld.java'. The code is as follows:

```
public class HelloWorld {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("Hello world!");  
    }  
}
```

The line `System.out.println("Hello world!");` is highlighted in light blue. The IDE has a standard window with a title bar, a menu bar, and a toolbar on the left.



Premier programme



✓ Compiler avec la commande :

```
Javac HelloWorld.java
```

Remarquer qu'une classe « HelloWorld.class » est générée

✓ Exécuter en lançant la machine virtuelle java et en lui spécifiant le point d'entrée

```
Java HelloWorld
```



Merci pour votre attention

