

Plan

Introduction

Classes et objets

Héritage et polymorphisme

Exceptions

Collections

Objectifs



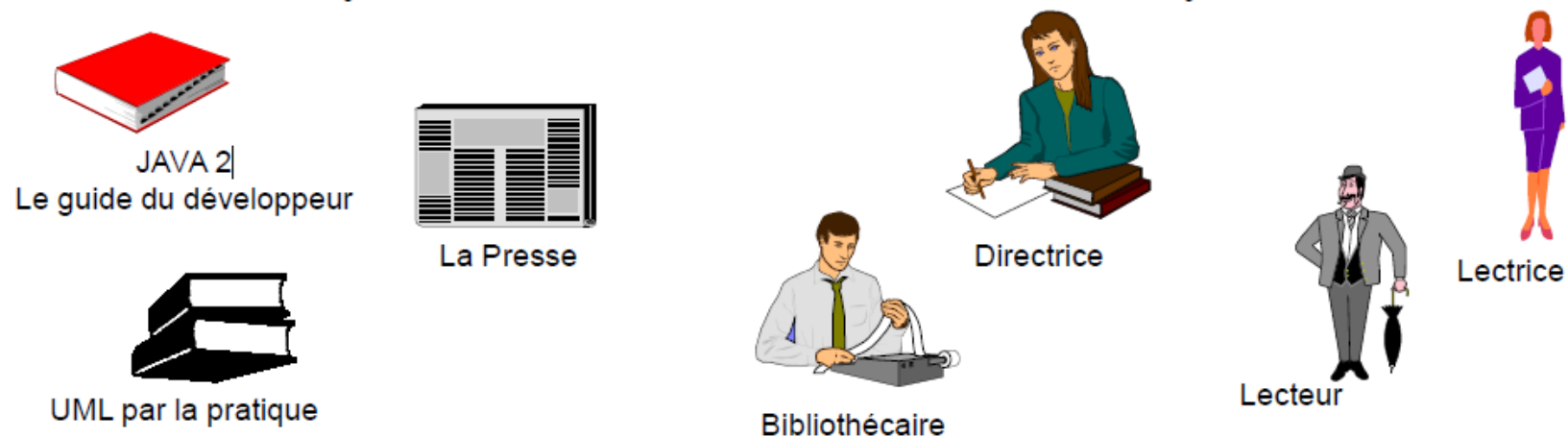
- ✓ Comprendre le paradigme OO et utiliser Java pour le mettre en œuvre
- ✓ Notion de classe et d'objet
- ✓ Décrire les éléments-clé de la plate-forme Java
- ✓ Compiler et exécuter une application Java
- ✓ Décrire la syntaxe du langage
- ✓ Comprendre et utiliser les API de java

Aperçu : Orienté Objet

- ✓ 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.

Aperçu : Orienté Objet

Exemple: Gestion d'une bibliothèque

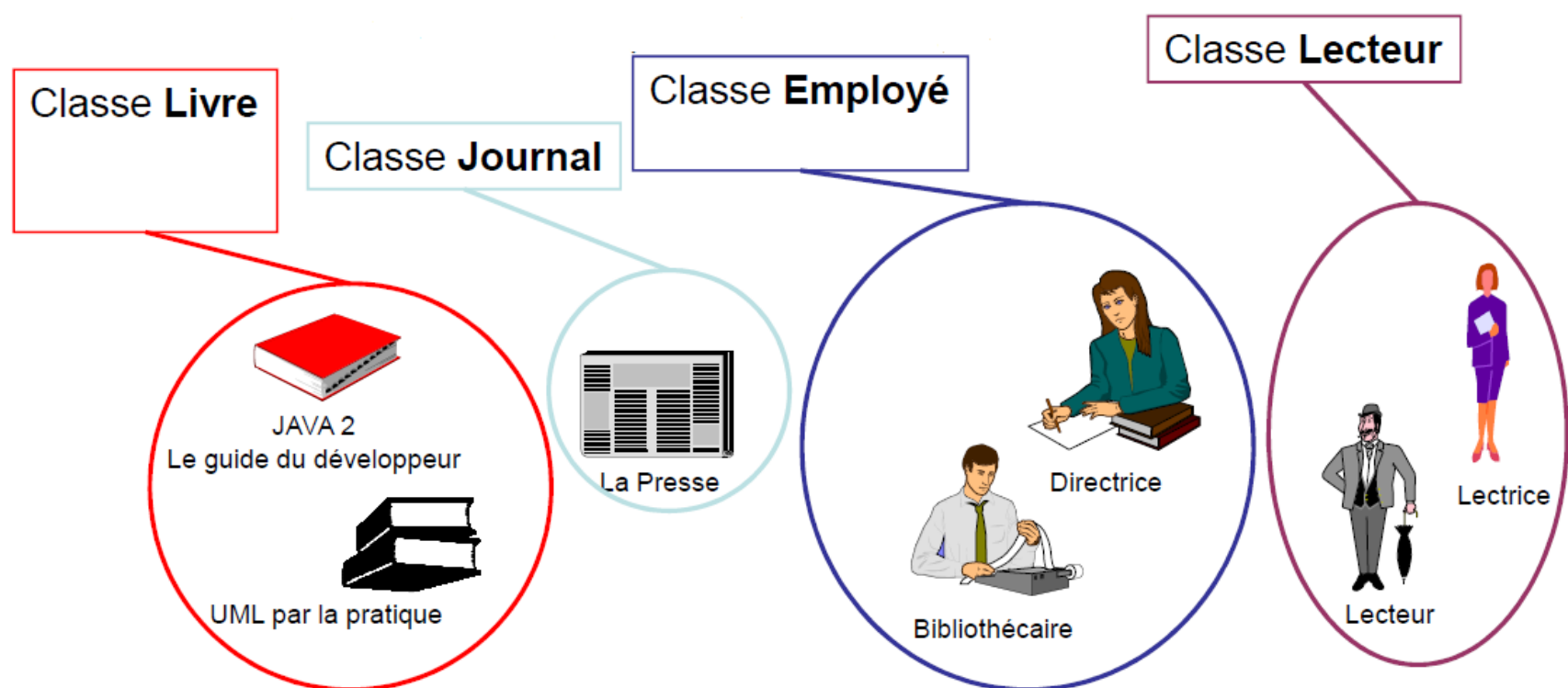


- **Approche procédurale :**
"Que doit faire mon programme ? "
- **Approche orientée-objet :**
"De quoi doit être composé mon programme ?"

Aperçu : Orienté Objet

Des objets similaires peuvent être informatiquement décrits par une même abstraction : une **classe**

- même structure de données et méthodes de traitement
- valeurs différentes pour chaque objet



JAVA: La plateforme



Présentation

Java est un langage de programmation moderne développé par **Sun Microsystems**(aujourd'hui racheté par **Oracle**)

Une des plus grandes forces est son excellente portabilité : un programme java peut s'exécuter sous Windows, Mac, Linux, etc....

Avec le langage Java on peut avoir:

- ✓ des applications sous forme de fenêtre ou de console ;
- ✓ Des applications web
- ✓ Des applications mobiles

Classement IEEE 2022 : <https://spectrum.ieee.org/top-programming-languages-2022>

Programme JAVA (1)

Un programme Java est compilé et interprété

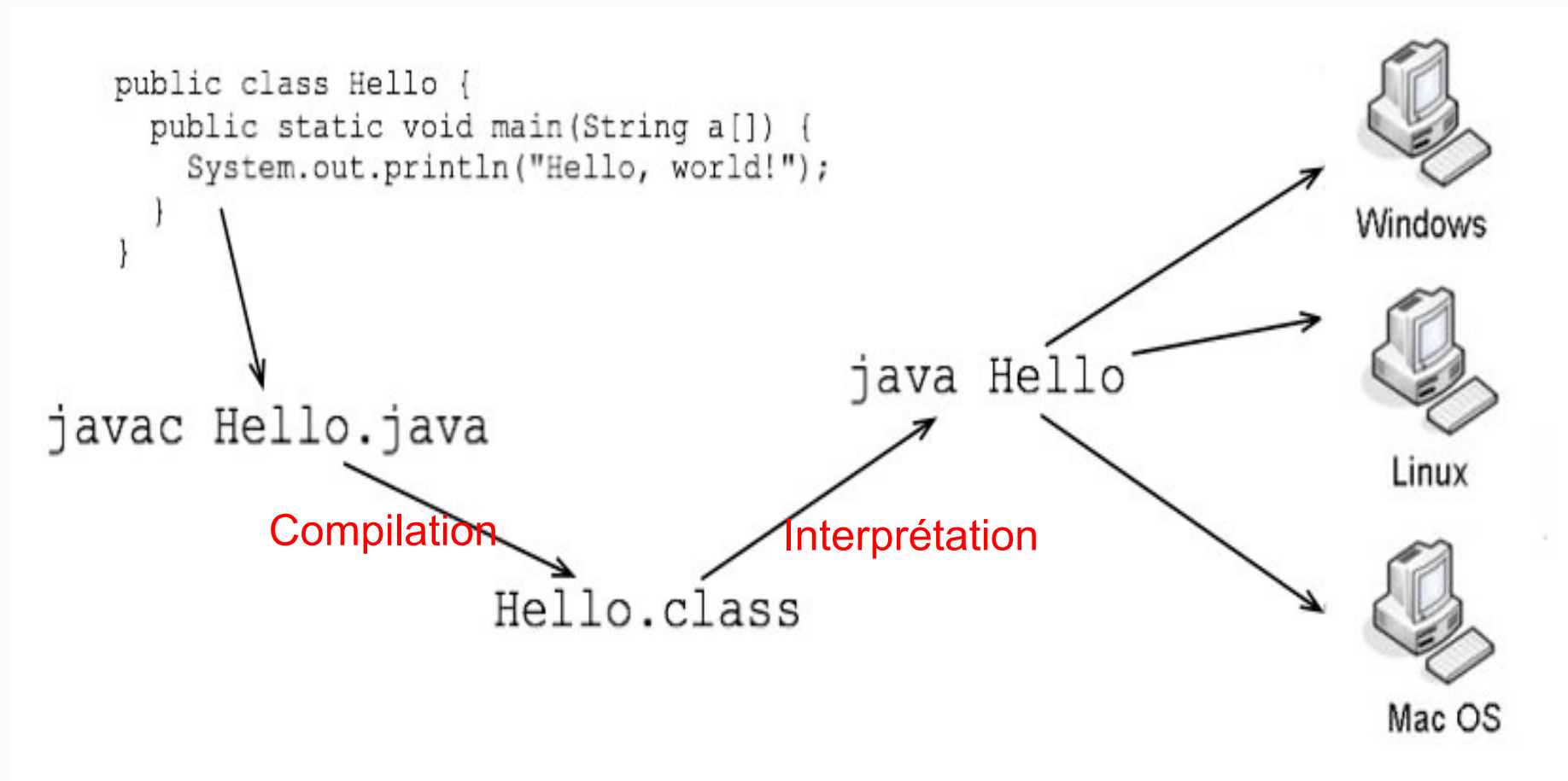
Compilé

-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.

Interprété

Programme JAVA (2)

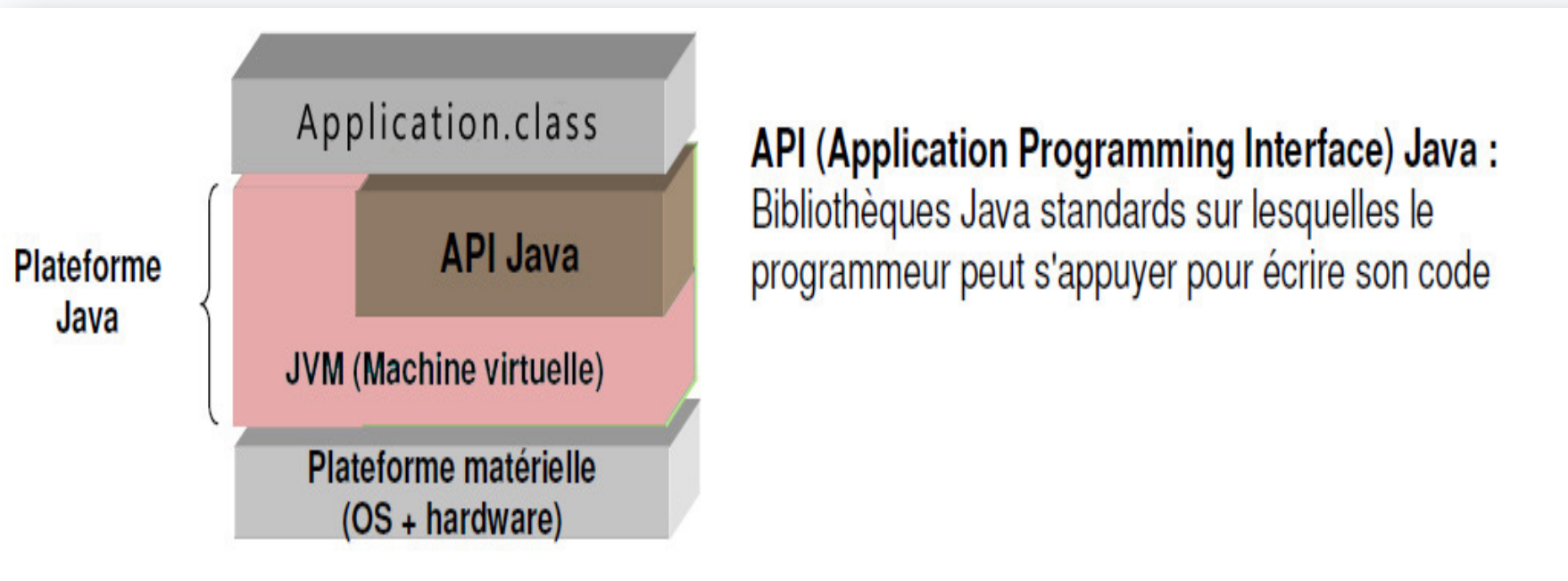


JAVA: la Plateforme

Plateforme = environnement hardware ou software sur lequel le programme est exécuté.

La Java « Platform » se compose de:

- ✓ Java Virtual Machine (Java VM)
- ✓ Java Application Programming Interface (Java API)



JAVA: Les versions



JAVA: Les versions

Année	Evénements
1995	JDK 1.0
1997	JDK 1.1
2004	J2SE 5.0 (1.5)
2006	Java SE 6.0 (1.6)
2010	Rachat de Sun par Oracle
2011	Java SE 7
2014	Java SE 8
2017	Java SE 9
2018	Java SE 10, 11
2019	Java SE 12, 13
2020	Java SE 14, 15
2021	Java SE 16, 17
2022	Java SE 18, 19

JAVA: Notions, mots clé...

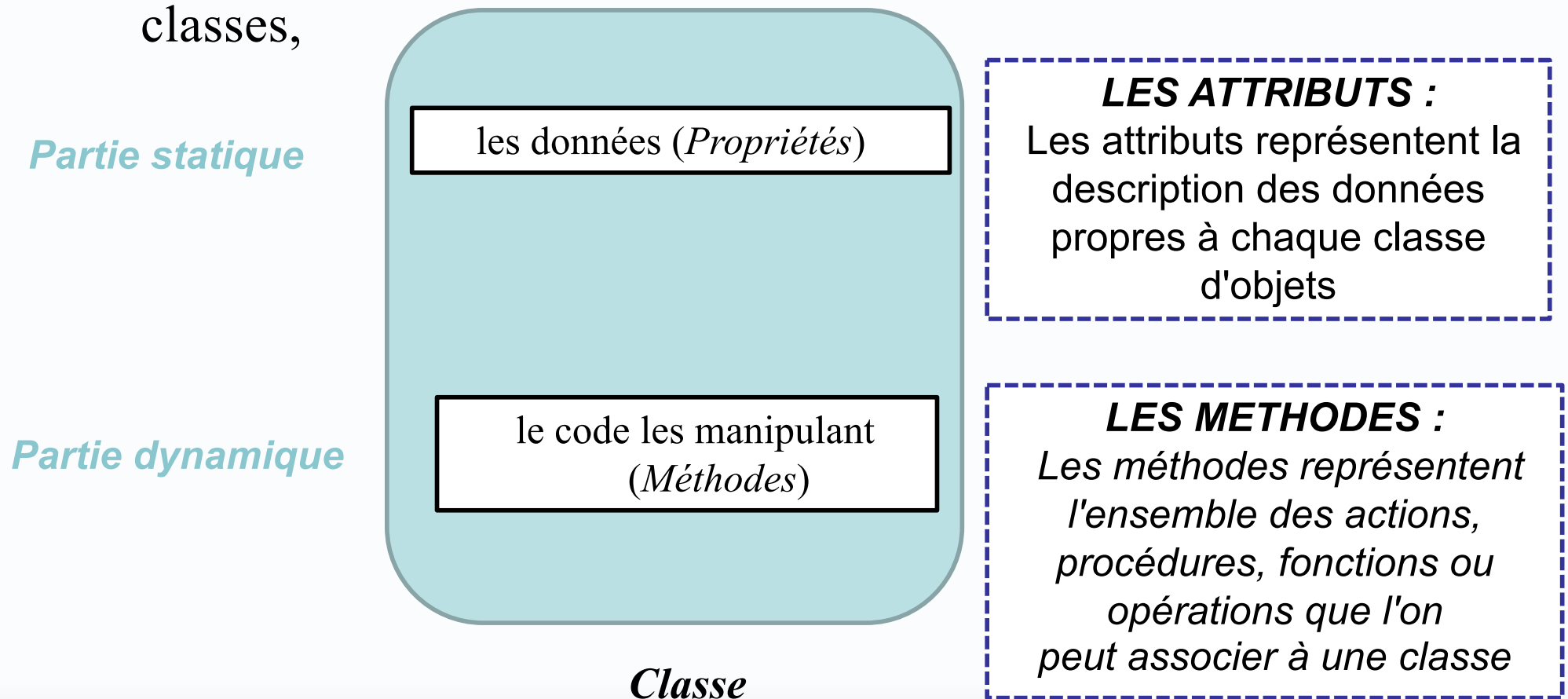


Notions fondamentales

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

JAVA: Classe

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



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 Java

Abstract	boolean	break	byte
case	catch	char	class
continue	default	do	double
else	extends	final,	finally
float	for	if	implements
import	instanceof,	int	interface
long	native	new	null
package	private	protected	public
return	short	static	super
switch	synchronized	this	throw
throws	transient	try	
void	volatile	while	

Les 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

- On distingue entre 2 grands groupes de types de données :
 - types primitifs
 - objets (instances de classe)

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 et valeurs

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

Les enveloppeurs (Wrappers)

- Les primitives peuvent être "enveloppées" dans un objet provenant d'une classe prévue
- Les enveloppeurs sont des objets pouvant contenir une primitive et auxquels sont associés des méthodes permettant de les manipuler.
- Ces enveloppeurs héritent tous de la classe Object et bénéficient de plusieurs méthodes utilitaires (comparaison, valeur maximale et minimale etc.).

Les enveloppeurs (Wrappers)

Primitive	Wrapper Class	Constructor Arguments
<code>boolean</code>	<code>Boolean</code>	<code>boolean</code> or <code>String</code>
<code>byte</code>	<code>Byte</code>	<code>byte</code> or <code>String</code>
<code>char</code>	<code>Character</code>	<code>char</code>
<code>double</code>	<code>Double</code>	<code>double</code> or <code>String</code>
<code>float</code>	<code>Float</code>	<code>float</code> , <code>double</code> , or <code>String</code>
<code>int</code>	<code>Integer</code>	<code>int</code> or <code>String</code>
<code>long</code>	<code>Long</code>	<code>long</code> or <code>String</code>
<code>short</code>	<code>Short</code>	<code>short</code> or <code>String</code>

Structure conditionnelle/ittératif

- **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 (1)

- 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 (2)

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 (javac), débiter (jdb) et exécuter un programme java.</p> <p>Il y a aussi des outils d'archivage (jar), de génération de documentation (javadoc).</p>

Outils de développement (3)

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)

Premier Programme (1)



```
*HelloWorld.java X

public class HelloWorld {

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

}
```

Premier Programme (2)

- Editer le code source Java suivant et le sauver dans Bienvenue.java :

```
class Bienvenue  
{  
public static void main( String args[]  
    {  
System.out.println( "Bienvenue dans le monde Java " );  
    }  
}
```

Premier Programme (3)

- Compiler avec la commande :

```
javac Bienvenue.java
```

- ➔ Remarquer que une classe « Bienvenue.class » est générée
- Exécuter en lançant la machine virtuelle java et en lui spécifiant le point d'entrée

```
Java Bienvenue
```

Scanner

- C'est une classe utilisée pour «balayage» des types primitifs et les chaînes de caractères.
- Elle a fait son apparition dans le package `java.util` dans la version 1.5.0 de java

Premier Programme

```
import java.util.Scanner;
public class Addition
{
    public static void main( String args[] )
    {
        Scanner input = new Scanner( System.in );
        int n1,n2,somme;
        System.out.print(" Donner le premier entier: " );
        n1 = input.nextInt();
        System.out.print(" Donner un deuxième entier: " );
        n2 = input.nextInt();
        somme = n1 + n2;
        System.out.printf( "Somme est %d\n", somme ); } }
```