

# Introduction à Java

**Philippe Genoud**

*Philippe.Genoud@imag.fr*

<http://www.initiaipes.fr/helix/people/genoud/ENSJAVA>

<http://membres-liglab.imag.fr/genoud/ENSJAVA/M2CCI/>

© Ph. Genoud – Université Joseph Fourier

1

## JAVA c'est quoi ?

---

- Une technologie développée par SUN Microsystems™ lancée en 1995 - rachetée par Oracle en 2009
  - Un langage de programmation
  - Une plateforme, environnement logiciel dans lequel les programmes java s'exécutent.
- Présente dans de très nombreux domaines d'application : des serveurs d'applications aux téléphone portables et cartes à puces (JME)
  - Quelques chiffres (Java One 2005):
    - 4,5 millions de développeurs Java
    - 700 millions de PC équipés d'une JVM
    - 708 millions de téléphones Java
    - 600 millions de périphériques mobiles
    - 1 milliard de Java Cards

# Le Langage Java

- Dans un des premiers papiers\* sur le langage JAVA, SUN le décrit comme suit :

« *Java : a simple, object-oriented, distributed, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language* »

\* *White Paper : The Java Language Environment* - James Gosling, Henry McGilton - May 1996  
<http://java.sun.com/docs/white/langenv/>

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

3

## Le Langage Java

### Mon premier programme Java (pas très objet...)

Le code de la classe doit être enregistré dans un fichier de même nom (casse comprise) que la classe

HelloWorld.java



1 Tout code java doit être défini à l'intérieur d'une classe

```
public class HelloWorld {  
    public static void main(String[] args) {  
        int i = 0;  
        while (i < 5) {  
            System.out.println("Hello World !");  
            i++;  
        }  
    }  
}
```

2 La description de la classe est effectuée à l'intérieur d'un bloc

4 Le point d'entrée pour l'exécution est la méthode main()

Compilation :  
javac HelloWorld.java



javac

HelloWorld.java

Exécution :  
java HelloWorld



java

HelloWorld.class

```
Hello World !  
Hello World !  
Hello World !  
Hello World !  
Hello World !
```

Septembre 2013

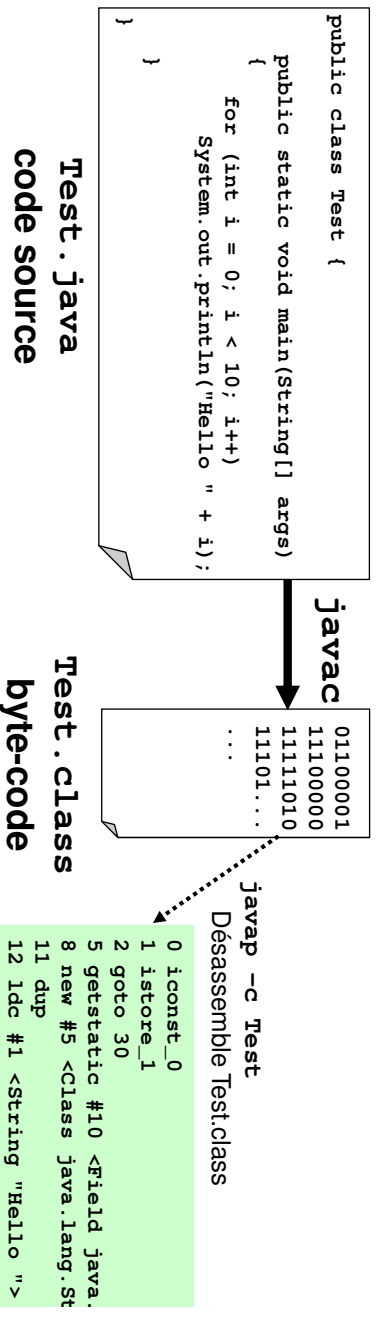
© Ph. Genoud – Université Joseph Fourier

4

# Le langage Java

## Un langage compilé / interprété

- Compilation d'un programme JAVA : génération de byte-code



- Le byte-code est :
  - proche d'un langage machine
  - indépendant de la plateforme d'exécution (matériel + OS)

Septembre 2013

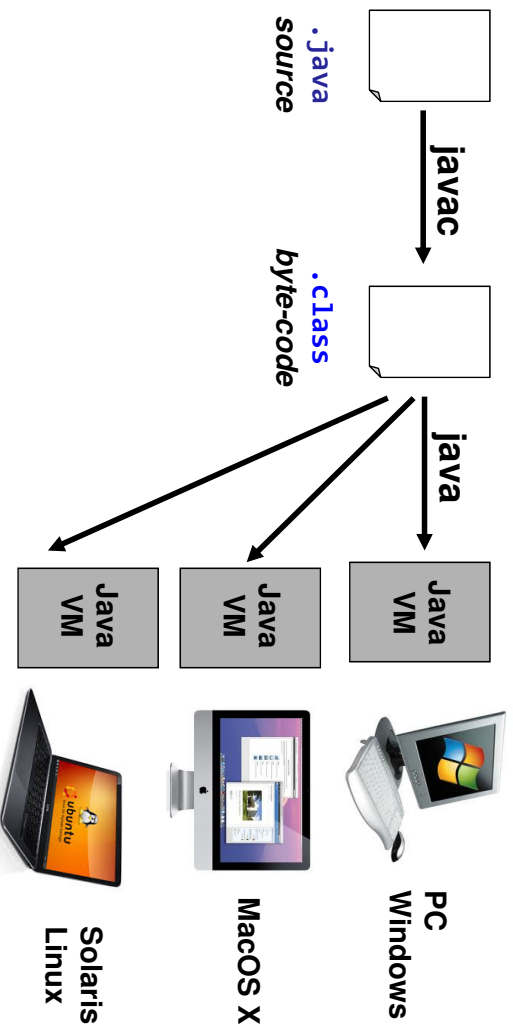
© Ph. Genoud – Université Joseph Fourier

5

# La machine virtuelle Java

## Exécution d'un programme Java compilé

- byte-code assure la portabilité des programmes Java
  - langage d'une Machine Virtuelle
  - à l'exécution un interpréteur simule cette machine virtuelle



Septembre 2013

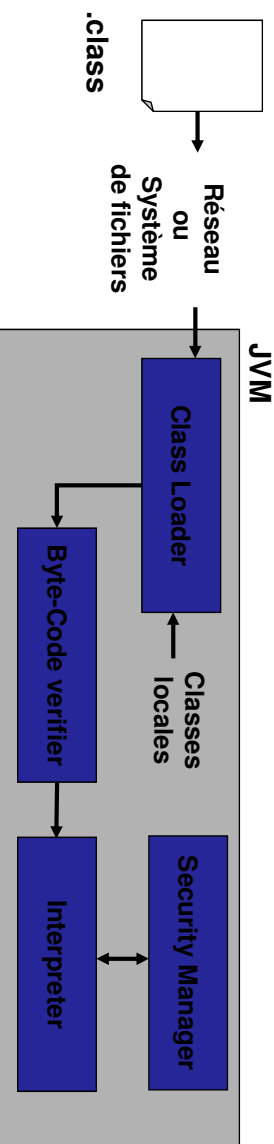
© Ph. Genoud – Université Joseph Fourier

6

# La machine virtuelle java

## Principes de fonctionnement

- **Chargement**
  - chargement sélectif et dynamique des classes
  - vérification statique du code (tentatives de modification de la machine virtuelle, ...)
- **Protection lors de l'exécution**
  - Le "security manager" possède un droit de veto (accès "sauvages" au système de fichiers, ...)
  - Cette protection peut-être modulée



Septembre 2013

© Ph. Genoud – Université Joseph Fourier

7

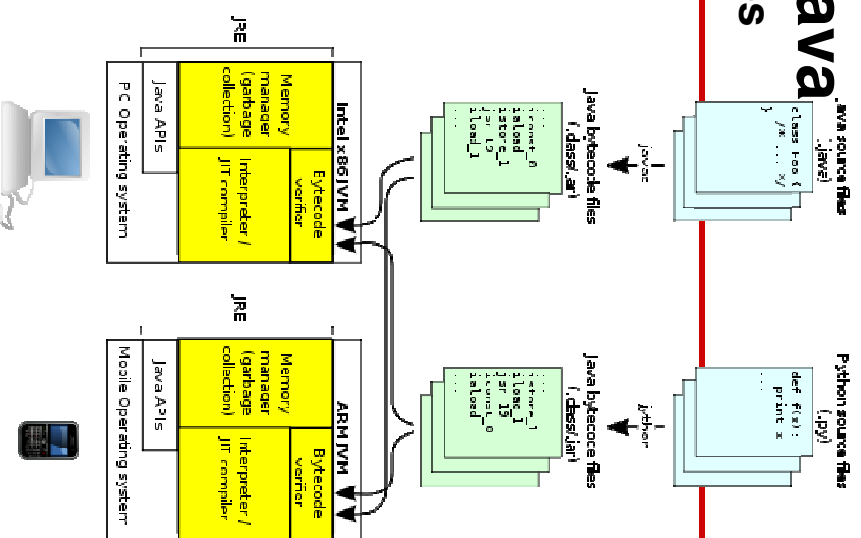
# La machine virtuelle java

## Utilisation pour d'autres langages

- JVM utilisée pour exécuter des programmes dans des langages autres que Java

Language	On JVM
Erlang	Erlang
JavaScript	Rhino
Pascal	Free Pascal
PHP	Quercus
Python	Jython
REXX	NetRexx <sup>[3]</sup>
Ruby	JRuby
Tcl	Jacl

Language
BBj
Clojure
Fantom
Groovy
Kotlin
MidJiePascal
Scala
Kawa



Septembre 2013

© Ph. Genoud – Université Joseph Fourier

8

# Performances

- **Exécution d'un programme Java**
  - le code Java est compact,
  - le chargement des classes nécessaires est sélectif et dynamique,
  - ... **mais Java est interprété**
- **Palier aux lenteurs de l'interprétation**
  - **utilisation d'un JIT (compilateur « Just-in-Time »)**
    - compilation à la volée du byte-code
    - réutilisation du code déjà compilé
  - intégration du JIT HotSpot dans JVM depuis version 1.3
  - *Performance of Java versus C++ J.P. Lewis and Ulrich Neumann (<http://www.idiom.com/%7EEzlla/Computer/javaCbenchmark.html>)*

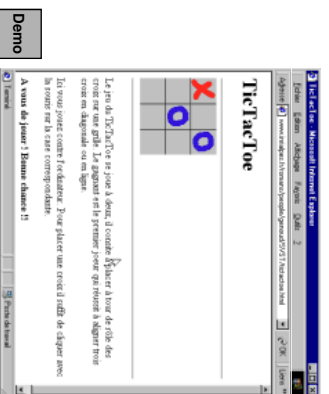
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

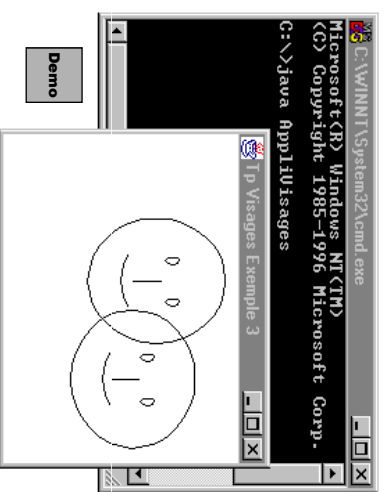
9

# Types de programmes Java

- **Les Applications indépendantes**
  - Programmes autonomes (stand-alone)



- **Les Applets**
  - Programmes exécutées dans l'environnement d'un navigateur Web et chargés au travers de pages HTML



- **Seuls diffèrent les contextes d'invocation et d'exécution**
  - Les droits des applets et des applications ne sont pas les mêmes

Septembre 2013

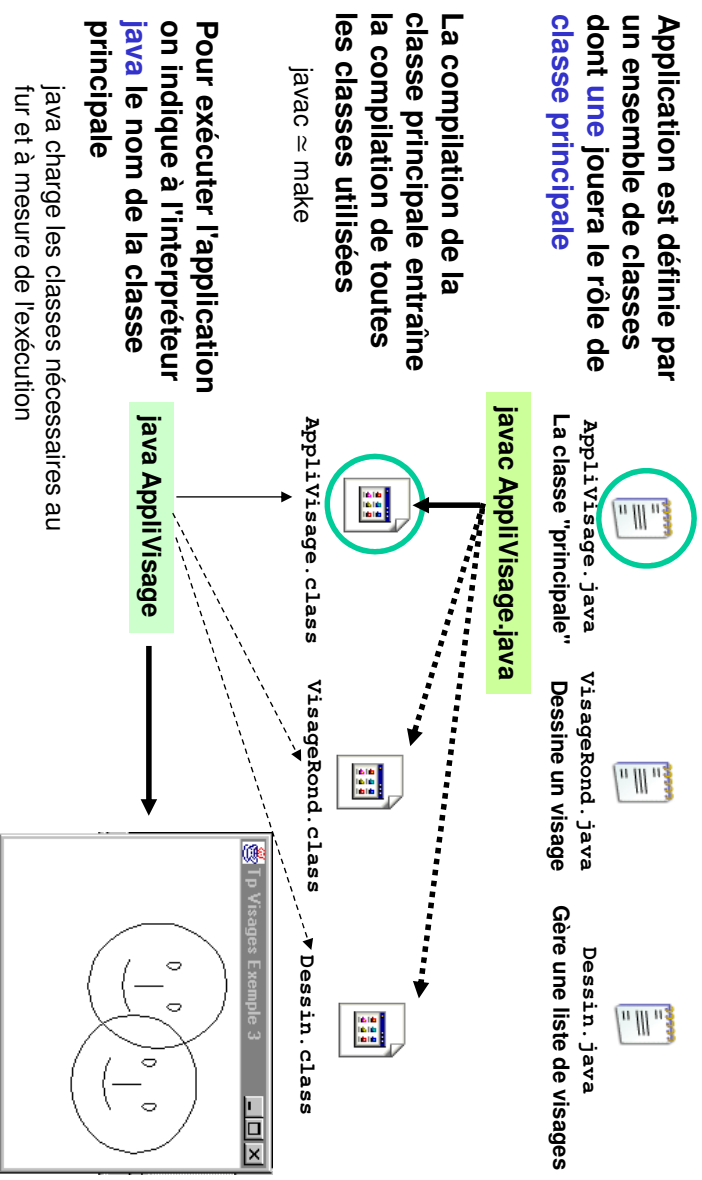
© Ph. Genoud – Université Joseph Fourier

10

# Types de programmes Java

## Application indépendante

Application est définie par un ensemble de classes dont **une** jouera le rôle de **classe principale**



Septembre 2013

© Ph. Genoud – Université Joseph Fourier

11

# Types de programmes Java

## Application indépendante

- Application doit posséder une classe principale
    - classe possédant une méthode de signature  
`public static void main(String[] args)`  
Tableau de chaînes de caractères (équivalent à `argv` du C)
  - Cette méthode sert de point d'entrée pour l'exécution
    - l'exécution de l'application démarre par l'interprétation de cette méthode
- ex : `java AppliVisage1`
- Exécute le code défini dans la méthode main contenue dans le fichier AppliVisage1.class

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

12

# Types de programmes Java

## Applet

- Classe principale ne possède pas de méthode `main()`
- Hérite de `java.awt.Applet` ou `javax.swing.JApplet`
- Son bytecode réside sur un serveur `http`
- Elle est véhiculée vers un client `http` (navigateur Web) via une page `html` qui contient son `url`
- Lorsqu'un navigateur compatible Java (avec sa propre machine virtuelle java (JVM)) reçoit cette page `HTML`, il télécharge le code de la classe et l'exécute sur le poste client
  - l'applet doit posséder un certain nombre de méthodes pour permettre cette exécution
    - `init()`, `start()`, `stop()`, `paint()`, `destroy()`

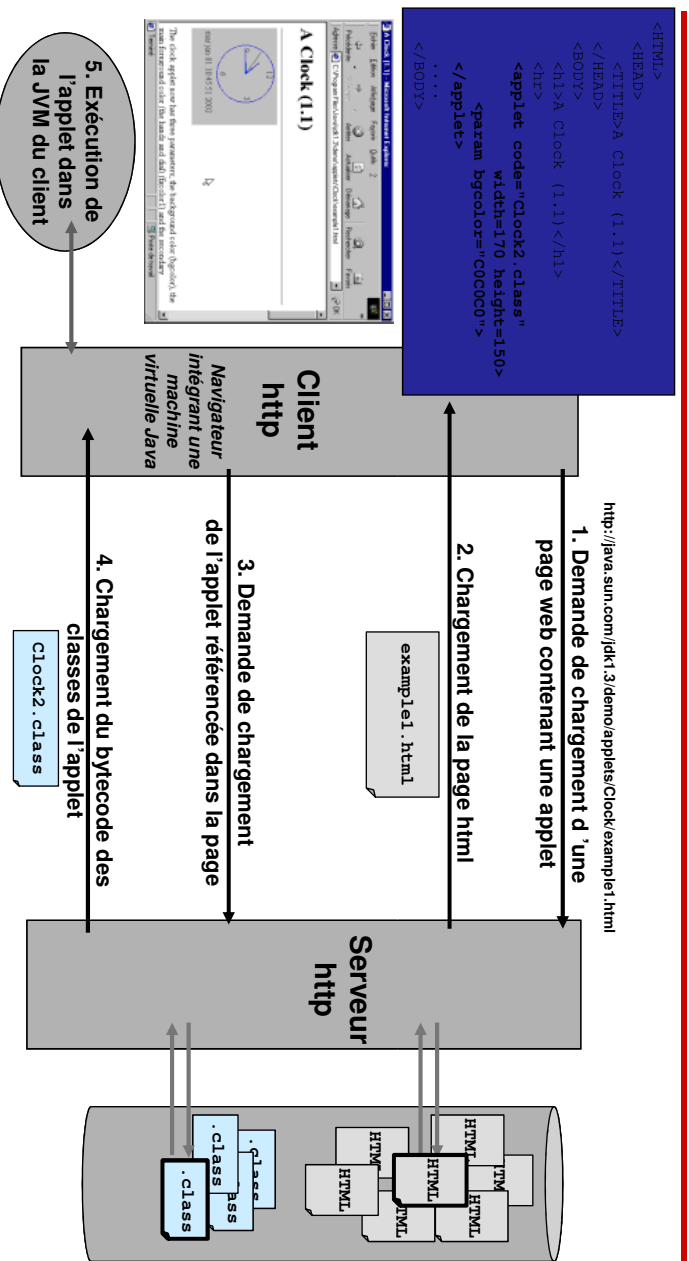
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

13

# Types de programmes Java

## Applet : Principe de fonctionnement



Septembre 2013

© Ph. Genoud – Université Joseph Fourier

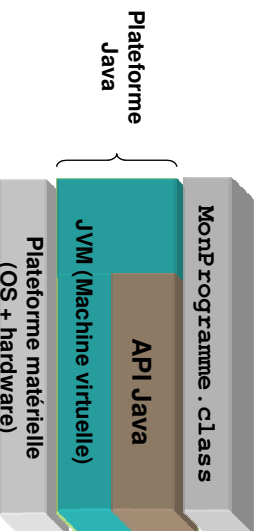
14



# La plateforme Java

- **Plateforme**
  - Environnement matériel et/ou logiciel dans lequel un programme s'exécute.
    - La plus part des plateformes sont la combinaison d'un OS et du matériel sous-jacent (*MS Windows + Intel, Linux + Intel, Solaris + Sparc, Mac Os X + Power PC*)

- **La plateforme Java est entièrement logicielle et s'exécute au dessus des plateformes matérielles**



**API (Application Programming Interface) Java :**  
Bibliothèques Java standards sur lesquelles le programmeur peut s'appuyer pour écrire son code

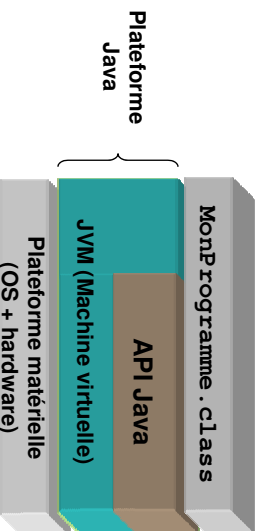
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

15

# La plateforme Java API Java

- **API Java**
  - (très) vaste collection de composants logiciels (classes et interfaces)
  - organisée en bibliothèques (***packages***)
  - offre de nombreux services de manière standard (indépendamment de la plateforme matérielle)



User Interface Looknks	AWT	Swing	Java 2D™
Accessibility	Drag n Drop	Input Methods	Image I/O
Integration Libraries Other Base Libraries	IDL	JDBC™	JNDI™
lang & util Base Libraries	Beans	Int'l Support	New I/O
	Networking	Std Override Mechanism	Security
	Lang & Util	Collections	Congifirency Utilities
	Preferences	Ref Objects	Reflection
			Serialization
			JAR
			Logging
			Versioning
			Zip



Programmer en Java nécessite une bonne connaissance de l'API.  
Attention à la courbe d'apprentissage (learning curve) qui est peut être longue

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

16



# La plateforme Java

## Les différentes éditions de Java

- 3 éditions de Java



**Standard Edition**  
JSE

Fourni les compilateurs, outils, runtimes, et APIs pour écrire, déployer, et exécuter des applets et applications dans la langage de programmation Java



**Entreprise Edition**  
JEE

Destinée au développement d'applications « d'entreprise » (« *business applications* ») robustes et interopérables. Simplifier le développement et le déploiement d'applications distribuées et articulées autour du web.



**Mobile Edition**  
JME

Environnement d'exécution optimisé pour les dispositifs « légers » :

- Carte à puce (smart cards)
- Téléphones mobiles
- Assistants personnels (PDA)

Versions MS Windows, Linux, Solaris téléchargeables **gratuitement** sur [java.sun.com](http://java.sun.com)

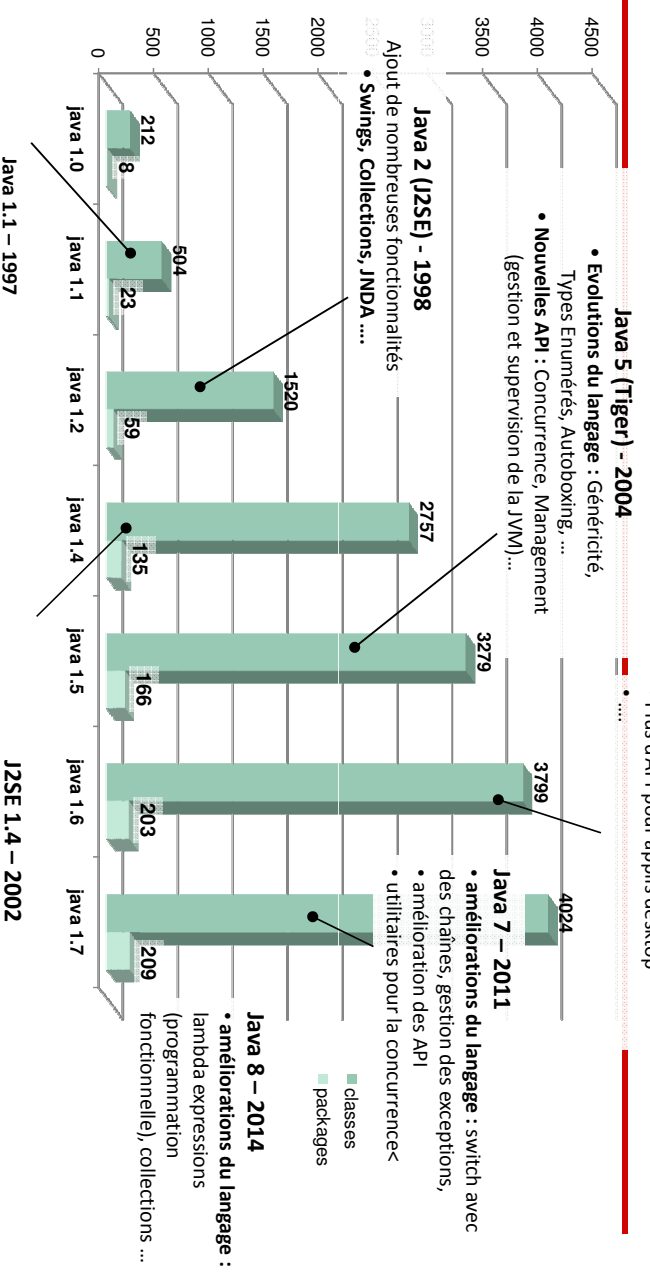
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

17

## La plateforme Java

### Evolution de l'API Standard



Septembre 2013

© Ph. Genoud – Université Joseph Fourier

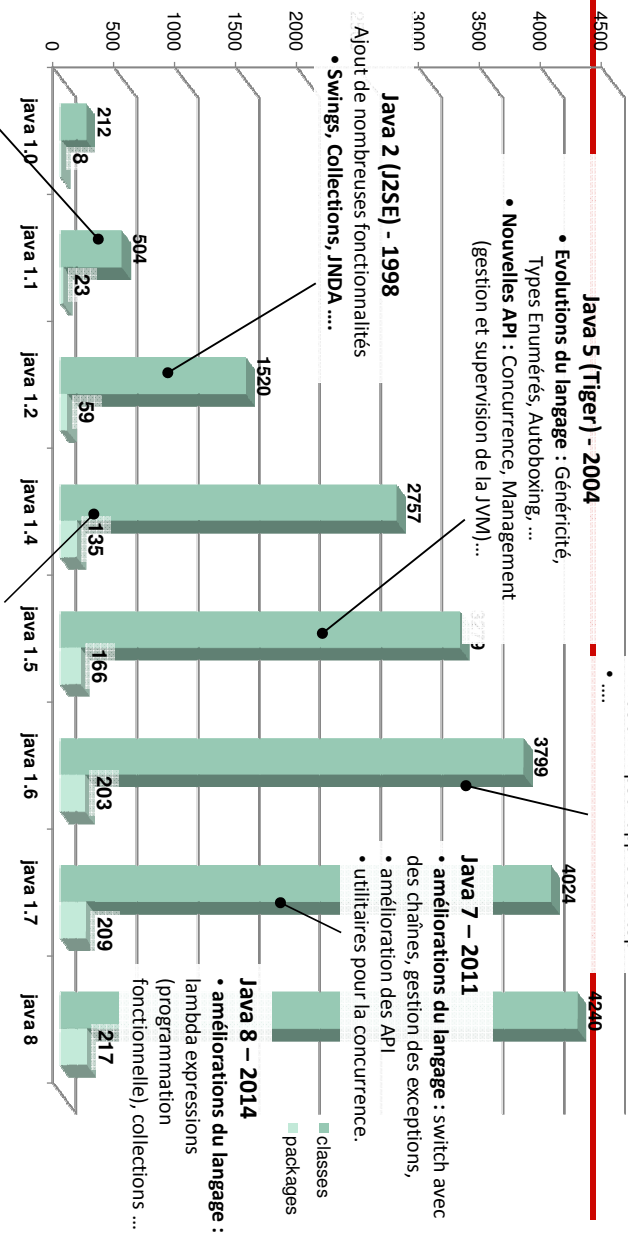
18

# La plateforme Java

## Evolution de l'API Standard

Java 6 (Mustang) - 2006

- Services Web
- Langages de scripts
- Accès au compilateur
- Plus d'API pour applis desktop



- Modification du langage : classes internes
- Modification du modèle événementiel pour awt Java Beans
- Amélioration des performances de la VM

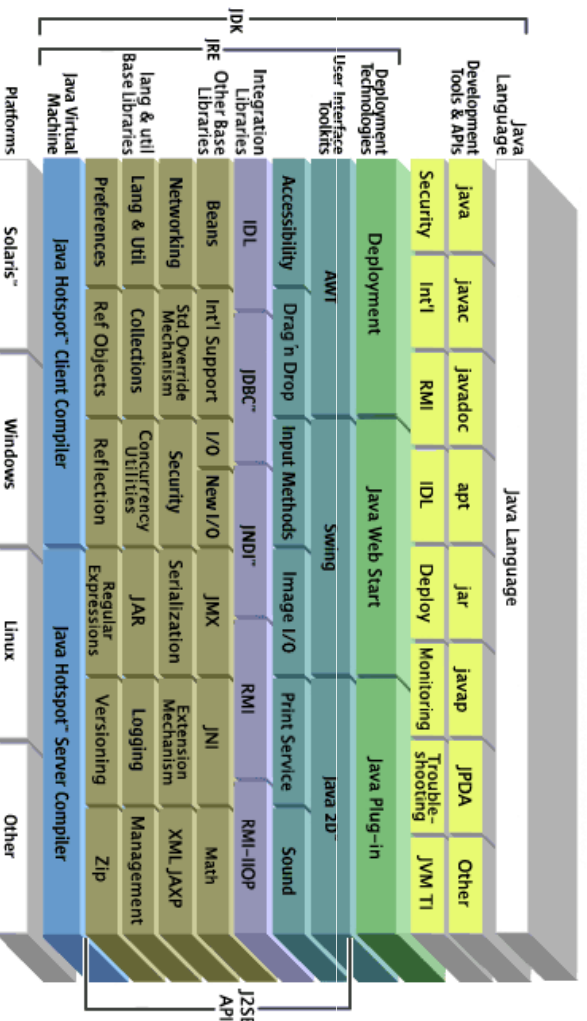
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

19

# JSE : Java Standard Edition

## Organisation générale



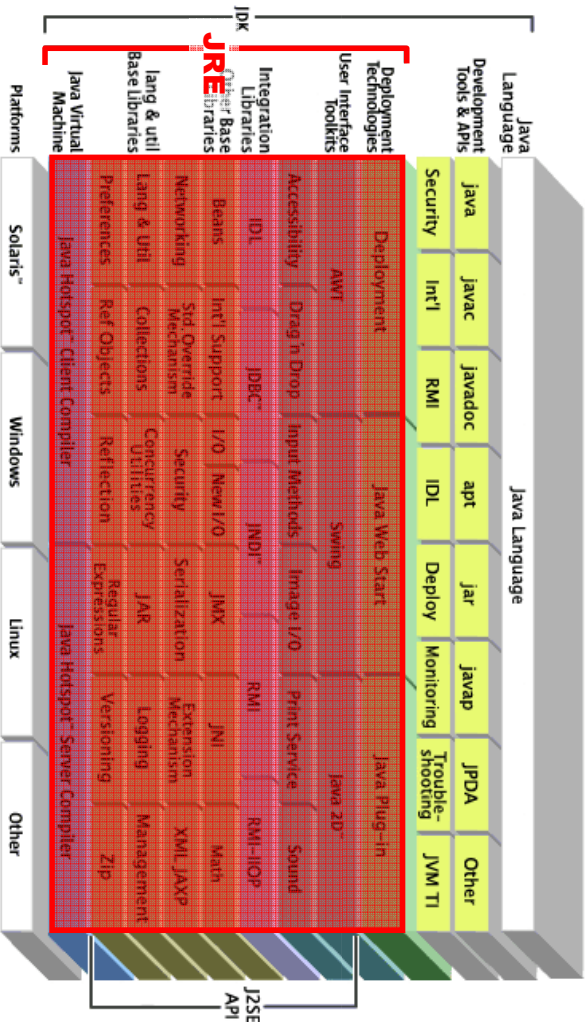
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

20

# JSE : Java Standard Edition

## Organisation générale



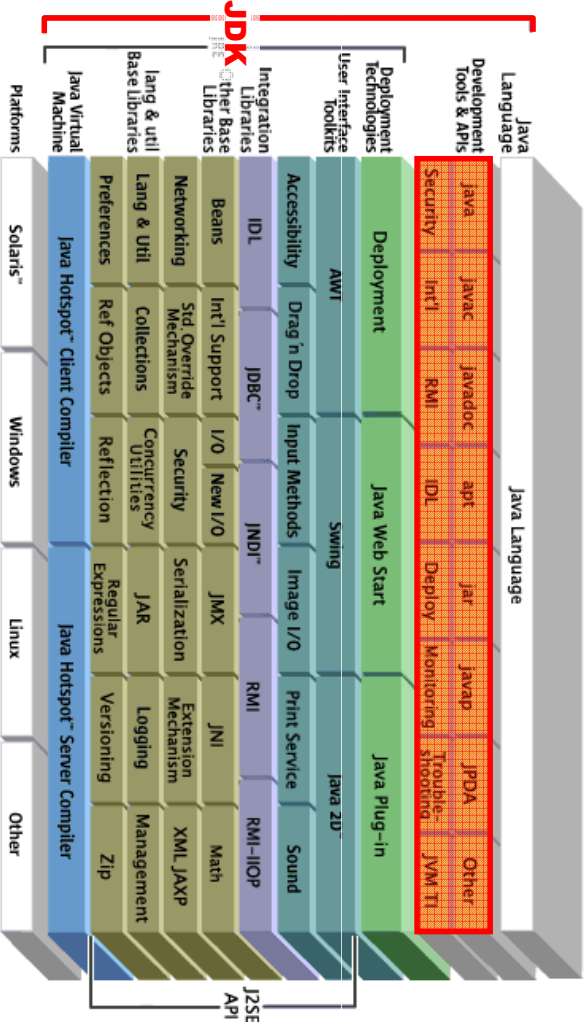
JRE (Java Runtime Environment) pour l'exécution de code java compilé

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

# JSE : Java Standard Edition

## Organisation générale



JDK (Java Developer's Kit) outils de base pour le développement d'applications Java

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

# JEE Java Enterprise Edition

- **Plateforme définie par SUN pour le développement d'applications réparties**
  - basée sur un ensemble de technologies Java
    - Servlets, JSP(Java Server Pages), JSF (Java Server Faces) : pages web dynamiques
    - EJB (Entreprise Java Beans) : objets métiers
    - JDBC : API d'accès à des SGBD
    - JNDI (Java Naming and Directory Interface) : API pour nommage des objets
    - JTA (Java Transaction API) ; API pour gestion des transactions
    - JAAS (Java Authentication and Authorization Service)
    - Et de nombreuses autres...
  - S'appuie sur un modèle d'architecture multi-tiers (multi-couches)

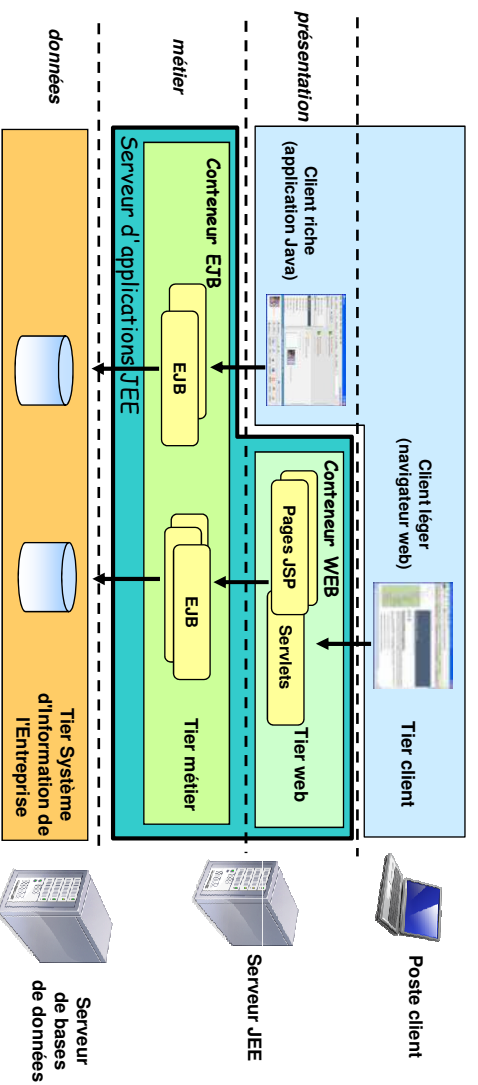
Septembre 2013

© Ph. Genoud – Université Joseph Fourier

23

## Architecture des applications JEE

- **Architecture multi-tiers**



- **Logique de l'application :**
  - Composants web (Servlet, JSP, JFS)
  - Composants métiers (EJB)
- **Services standards (cycle de vie des composants, multithreading, transactions, persistance...) pris en charge par les conteneurs Web et EJB du serveur d'application JEE**

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

24

# JEE situation actuelle

- Une spécification (Sun puis Oracle) :
  - JEE 5 (2006), JEE 6 (2009), JEE 7 (2013)
- Différentes implémentation de la plateforme
  - Des implémentations commerciales
    - WebLogics server (Oracle), WebSphere (IBM), ...
    - Sun Java System Application Server, Oracle Application Server...
  - Ou open-source
    - GlassFish, JBoss, Geronimo (Apache), JOnAS...
- Processus de certification
  - TCK (Test Compatibility Kit) (~ 20000 tests)
  - Payant sauf pour plateformes open-source

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

25

## Environnements de développement intégrés

- Nombreux IDE (Integrated Development Environment) pour java
  - Editeur syntaxique, débogueur, compilateur, exécution
- Commerciaux



JCreator  
Xinox



WebSphere Studio  
Site Developer for Java  
IBM



JBUILDER  
Codegear

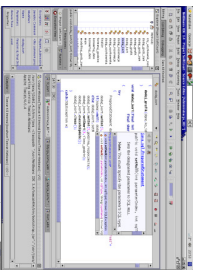


ORACLE  
JDEVELOPER  
10g  
JDeveloper  
Oracle

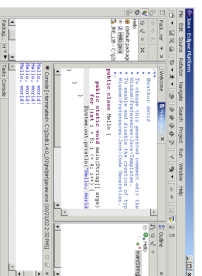


IntelliJIDEA  
JetBrains  
...

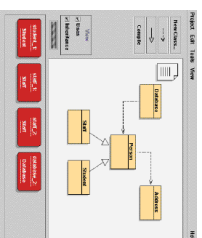
C# .net



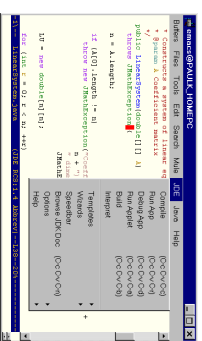
NetBeans  
[www.netbeans.org](http://www.netbeans.org)



Eclipse  
[www.eclipse.org](http://www.eclipse.org)



BlueJ  
[www.bluej.org](http://www.bluej.org)



Emacs + JDE  
<http://sunsite.unc.dk/jde>

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

26

# Java aujourd'hui

---

- **Une maturité certaine**
  - Technologie largement reconnue dans le monde industriel
- **Evolutions de JAVA contrôlées par organisation indépendante Java Community Process ([www.jcp.org](http://www.jcp.org))**
  - Évolutions du langage (généricité dans version 1.5(Tiger))
  - Nouvelles API
- **Passage de Java dans la communauté open source (2007) → OpenJDK**
  - Jusqu'alors JAVA était une marque propriétaire de SUN
- **Bataille avec C#.NET sur la plateforme Windows**
- **Rachat de SUN par Oracle (avril 2009)**

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

27

## 1ère séance de TP

---

- **utilisation depuis ligne de commande des outils de base pour le développement d'applications Java**
  - Compilation :
    - javac
  - Exécution :
    - java (applications)
    - appletviewer (applets)
  - Documentation
    - javadoc
  - "Archivage"
    - jar
- **Pourquoi ne pas utiliser tout de suite un IDE Java ?**
  - Être capable de travailler avec Java quelque soit l'environnement
  - Comprendre des mécanismes parfois masqués par des environnements intégrés (ex. CLASSPATH)

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

28



## 2ème séance de TP

- **Objectif** : expérimenter les constructions de base du langage Java (types simples, instructions de contrôle, itérations) qui sont très proches syntaxiquement de celles utilisées par le langage C

- **Ecriture de programmes simples**

- Thème 1 : expressions
- Thème 2 : instructions conditionnelles
- Thème 3 : itérations
- Thème 4 : instanciación d'objets - envois de messages

<http://1ig-membres.imag.fr/genoud/ENSJAVA/M2CCI/cours/>

TYPES ET INSTRUCTIONS ELEMENTAIRES	
<b>Types simples</b>	entiers réels caractère booléen
<b>Enoncés élémentaires</b>	déclaration de variable affectation
<b>Flot de contrôle</b>	bloc d'instructions if... else while do while for
<b>Antichage</b>	System.out.println System.out.print

<http://1ig-membres.imag.fr/genoud/ENSJAVA/M2CCI/cours/cours/Java/typessimples.html>

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

29

## Références

« *Java - tête la première* »  
Kathy Sierra, Bert Bates, 2ème édition, O'Reilly, septembre 2005

« *Java 7 : les fondamentaux du langage Java* »  
Thierry Groussard, Editions ENI, 2011

« Au coeur de Java 2 - Volume 1 : Notions fondamentales »  
Cay S. Horstmann, Gary Cornell, – 8ème édition, Pearson education, avril 2008

« Programmer en Java »,  
Claude Delannoy, 9ème édition, Editions Eyrolles, juin 2014

et beaucoup d'autres ...

Septembre 2013

© Ph. Genoud – Université Joseph Fourier

30



# Références (suite)

---

## URLs

- <https://www.oracle.com/java/index.html>  
**Site officiel Java d'ORACLE**
  - Téléchargement, Tutoriels, Documentations, spécifications, ...
  - <http://docs.oracle.com/javase/tutorial/index.html>
  - <http://www.oracle.com/technetwork/java/index.html>
- <http://java.developpez.com/>  
des tutoriels, des FAQ, des ressources....
- <http://www.javapractices.com/>  
nombreux exemples de code
- <http://www.javaworld.com>  
Magazine électronique, l'actualité de java
- ...