

# Introduction à la programmation en JAVA



# Table des matières

- **I. Introduction à Java et historique du langage**
- **II. Notre outil de développement : *Eclipse Mars***
- **III. Le langage Java et sa syntaxe**
- **IV. La POO avec Java**
- **V. API Java**
- **VI. La gestion des exceptions**
- **VII. Les collections**
- **VIII. La sérialisation**

# Introduction à Java et historique du langage



# Aperçu du chapitre

**I. Historique du langage Java**

**II. Qu'est ce que Java ?**

**III. La plateforme d'exécution (JRE) et le système de compilation (JDK)**

**IV. Exercices**

# I. Historique du langage Java

- Années 1990 avec **James Gosling**

Objectif: développer un langage de programmation indépendant de la plateforme hardware

=> langage **Oak** => ce fut un échec

- **Bill Joy** (co-fondateur de la firme **Sun Microsystems**) : nouvelle version d'Oak appelée **Java** (mot d'argot américain signifiant café)

Objectif: pallier une déficience des langages de programmation

Comment: en produisant un langage conçu pour des machines et des logiciels hétérogènes qu'englobe Internet.

# I. Historique du langage Java

- Après de très **nombreuses modifications** visant à améliorer le système, **Java** est devenu plus qu'une simple solution Internet, c'est dorénavant un langage utilisé pour toutes sortes de développements.
- Java naquit en même temps que le **Web** et toutes les qualités que requiert ce dernier étaient implémentées dans Java :
  - un langage permettant un fonctionnement sur des machines structurées différemment
  - une nécessité de bande passante la plus minime possible, pour pallier l'étroitesse de celle offerte par le Web

**Netscape** (navigateur web) fut l'un des éléments essentiels dans la création et l'implantation de Java dans le parc informatique, en intégrant Java dans son logiciel

# Aperçu du chapitre

**I. Historique du langage Java**

**II. Qu'est ce que Java ?**

**III. La plateforme d'exécution (JRE) et le système de compilation (JDK)**

**IV. Exercices**

## II . Qu'est ce que Java ?

- **Java** est une **technologie** développée par **Sun Microsystems**
- Cette technologie reprend plusieurs produits et spécifications de logiciels qui constituent un **système pour développer et déployer** des applications
- Java est utilisée dans une grande variété de plates-formes : systèmes embarqués, téléphones mobiles, serveurs, applications d'entreprise, superordinateurs, interfaces graphiques comme les applets Java du Web, ...
- Les programmeurs utilisent le mot **Java** pour désigner le langage de programmation, tandis que la plate-forme d'exécution est appelée **JRE** (*Java Runtime Environment*) et le système de compilation est appelé **JDK** (*Java Development Kit*)



## II . Qu'est ce que Java ? – *Le langage de programmation*

<b>Apparu le</b>	23 mai 1995
<b>Auteur</b>	Sun Microsystems
<b>Paradigme*</b>	Programmation orientée objet, structurée et impérative
<b>Typage**</b>	Statique, fort, sûr, nominatif
<b>Influencé par</b>	Objective-C, C++, Smalltalk, Eiffel
<b>A influencé</b>	C#, D, J#, Ada 2005, Gambas
<b>Système d'exploitation</b>	Multiplate-forme
<b>Licence</b>	GNU GPL

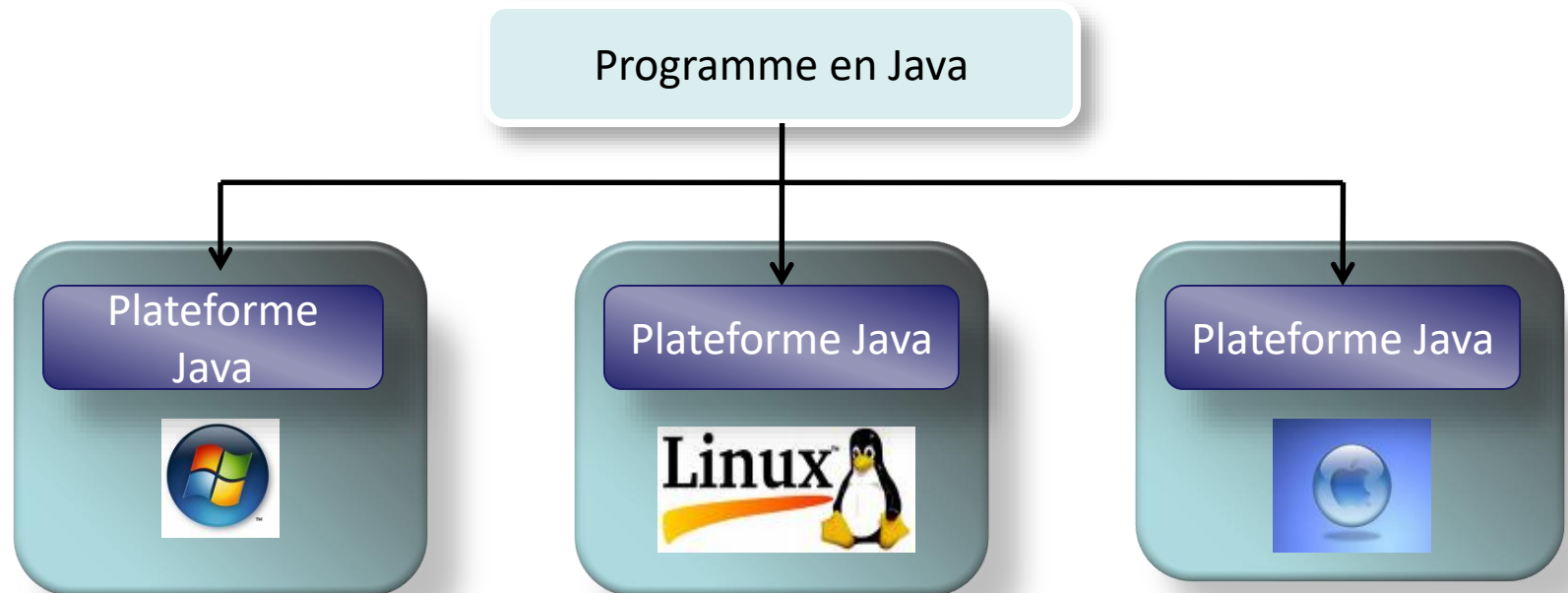
Les concepteurs ont privilégiés **l'approche orientée objet** de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)



## II . Qu'est ce que Java ? – *Le langage de programmation*

Le **Java** a la particularité principale que les logiciels écrits avec celui-ci sont facilement **portables** sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou Linux avec peu ou pas de modifications...

C'est la **plate-forme d'exécution** qui garantit la portabilité des applications développées en Java!



## II . Qu'est ce que Java ? – Exécution d'un programme Java

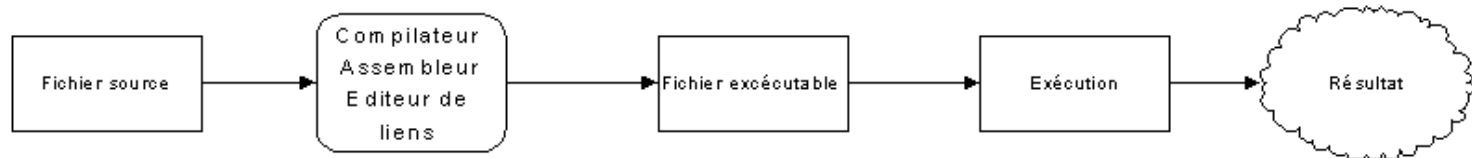
Différence entre un langage compilé et interprété :

- Un langage **compilé** est traduit en langage machine (binaire) par un compilateur pour donner un **fichier exécutable** qui pourra être utilisé indépendamment des sources.
- Un langage **interprété** est traduit par un interpréteur **au fur et à mesure** de l'exécution du programme et dépend fortement des sources.

Programme interprété



Programme compilé



## II . Qu'est ce que Java ? – Exécution d'un programme Java

Le **Java** est un langage pré-compilé !

### Compilation traditionnelle

Mon programme  
en C++



Compilateur  
spécifique à  
chaque  
environnement

Mon application

Code natif directement exécutable  
sur la machine mais spécifique

### Compilation avec Java

Mon programme  
en Java



Compilateur  
multiplateformes

Bytecode  
(qui doit être interprété)

## II . Qu'est ce que Java ? – Exécution d'un programme Java

Programme en  
Java



01001001101



### Compilateur multiplateformes:

traduit le programme en un code intermédiaire  
(**Bytecode**) indépendant de la machine  
(exécuté **une seule fois**)

### Interpréteur:

lit le Bytecode et l'**exécute** sur la  
machine  
(exécuté à **chaque fois** que le  
programme est appelé)

# Aperçu du chapitre

**I. Historique du langage Java**

**II. Qu'est ce que Java ?**

**III. La plateforme d'exécution (JRE) et le système de compilation (JDK)**

**IV. Exercices**

## III . La plateforme d'exécution (JRE) et le système de compilation (JDK)

La **plateforme Java** permet de développer et d'exécuter des programmes écrits avec le langage Java indépendante du processeur et du système d'exploitation.

La plateforme se compose principalement d'une **machine virtuelle** et d'un **compilateur**

Il existe plusieurs versions de cette plate-forme :

- ***Java Standard Edition*** ou Java SE destinés aux **ordinateurs de bureau**
- ***Java Enterprise Edition*** ou Java EE destinés aux **serveurs Web**
- ***Java Micro Edition*** Java ME destinées aux portables comme les **smartphones**

## III . La plateforme d'exécution (JRE) et le système de compilation (JDK)

La **JRE** est constitué de :

- une **Java Virtual Machine (JVM)**, le programme qui exécute des applications compilés sous forme de bytecode
- une **bibliothèque standard** à partir de laquelle doivent être développés tous les programmes en Java

Les trois tâches principales de la **JRE** sont :

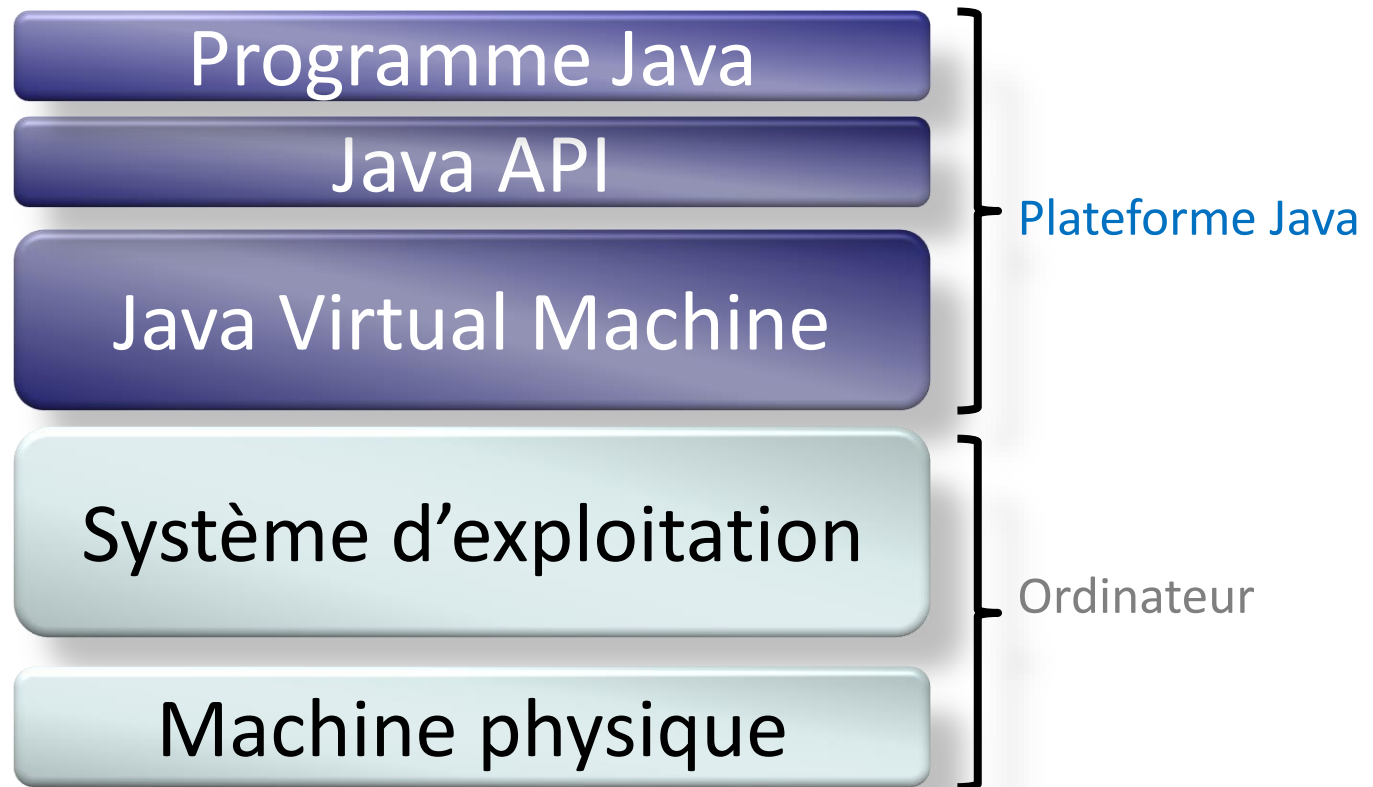
- charger le code (class loader)
- vérifier le code (bytecode verifier)
- exécuter le code (runtime interpreter)

Le **JDK** est constitué de :

- la **jre**
- **d'outils de développement** permettant de compiler du code Java pour produire le bytecode



### III . La plateforme d'exécution (JRE) et le système de compilation (JDK)



## III . La plateforme d'exécution (JRE) et le système de compilation (JDK)

La JVM est spécifique à chaque plateforme (ou couple machine/système d'exploitation) et permet aux applications Java compilées en bytecode de produire les mêmes résultats quelle que soit la plateforme

En résumé, elle

- Définit les spécifications hardware de la plateforme
- Lit le bytecode compilé
- Est implémentée dans des environnements de développement ou dans les navigateurs Web

## III . La plateforme d'exécution (JRE) et le système de compilation (JDK) – *Java API*

**L'API Java** (Application Programming Interface) est constitué de librairies (packages) qui reprennent des ensembles fonctionnels de composants:

- data type, objets, string, I/O, date, ....
- Applet
- Abstract Window Toolkit (AWT)
- Sécurité
- Networking
- Internationalisation
- ...

Il s'agit d'**outils** mis à la disposition du programmeur pour lui permettre de développer ses applications à base de Java

# Aperçu du chapitre

**I. Historique du langage Java**

**II. Qu'est ce que Java ?**

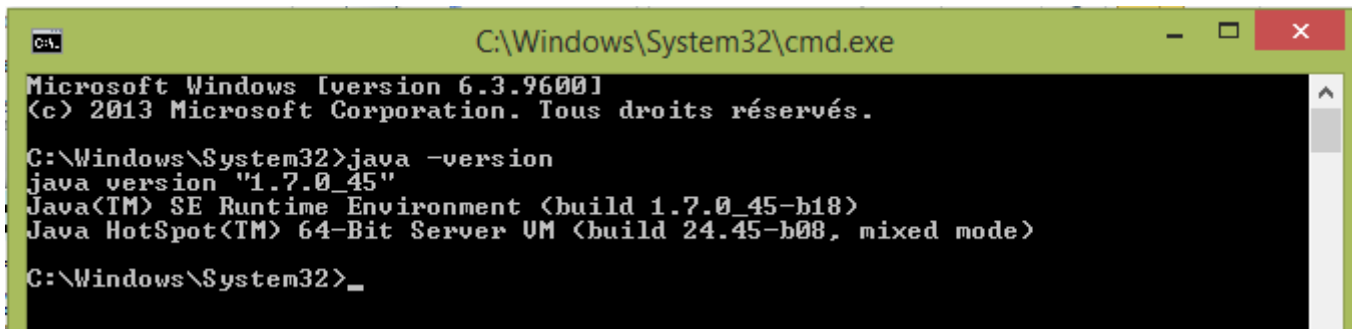
**III. La plateforme d'exécution (JRE) et le système de compilation (JDK)**

**IV. Exercices**

## IV . Exercices - Création d'un premier programme en Java *Hello World!*

### Déroulement de l'exercice:

- Vérifier l'installation de la plateforme **Java** sur vos machines (JDK + JRE)

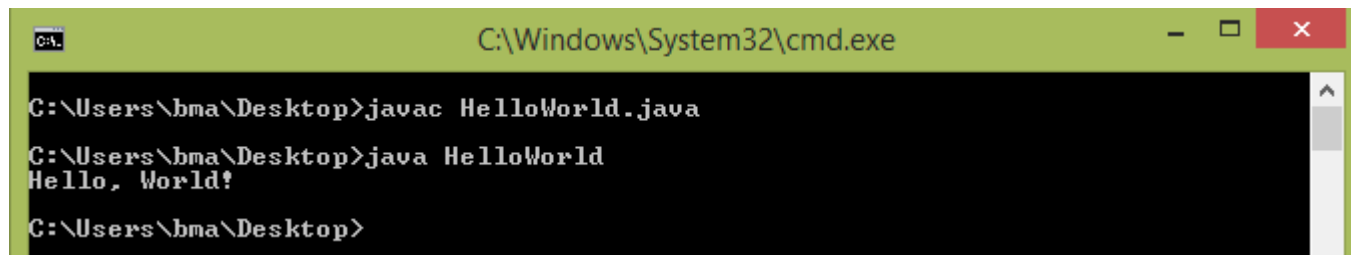


```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 6.3.9600]
(c) 2013 Microsoft Corporation. Tous droits réservés.

C:\Windows\System32>java -version
java version "1.7.0_45"
Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
Java HotSpot(TM) 64-Bit Server VM (build 24.45-b08, mixed mode)

C:\Windows\System32>_
```

- Ecriture du programme **Java** Hello World
- Compilation et exécution du programme en ligne de commandes



```
C:\Users\bma\Desktop>javac HelloWorld.java
C:\Users\bma\Desktop>java HelloWorld
Hello, World!
C:\Users\bma\Desktop>
```

## IV . Exercices - Création d'un premier programme en Java *Hello World!*

Analysons maintenant la structure de notre premier programme:

*Définition de la classe HelloWorld*

```
public class HelloWorld  
{
```

*Tout programme Java doit comporter une méthode Main avec la signature suivante*

```
    public static void main(String args[])  
    {
```

*Affichage d'un message à l'écran*

```
        System.out.println("Hello World!");
```

```
    }
```

```
}
```