



## POO : Le langage JAVA



### JAVA : Présentation, caractéristiques

- Le langage Java est un langage de programmation informatique orienté objet créé par **James Gosling** et **Patrick Naughton** employés de **Sun Microsystems**. Il a été racheté par la société **Oracle** en 2009 ;
- Java est un langage de programmation :
 

<ul style="list-style-type: none"> <li>- Simple</li> <li>- Orienté Objet pur</li> <li>- Interprété (code intermédiaire)</li> <li>- Portable (indep de la plate-forme)</li> </ul>	<ul style="list-style-type: none"> <li>- Robuste (difficile d'avoir des échecs)</li> <li>- Sécurisé</li> <li>- Multi-threads (Supporte le multitâches)</li> <li>- Distribué (Support des protocoles réseau, support de la distribution)</li> </ul>
--	--

### Fichier source, Compilation et Bytecode

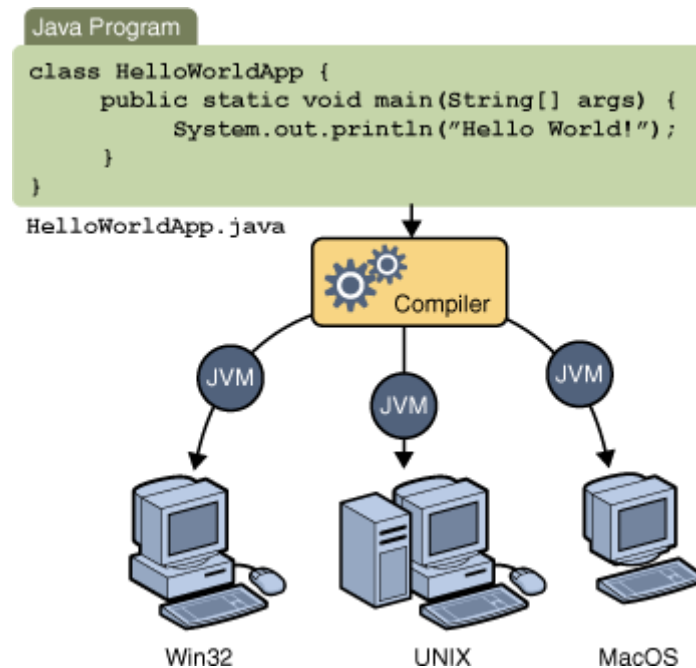
- Le fichier source d'un programme écrit en Java est un simple fichier texte dont l'extension est par convention **(.java)**
- Lorsque le programme est prêt à être "essayé", il s'agit de le compiler (**le traduire en langage machine**) à l'aide d'un **compilateur**. Toutefois, contrairement aux langages compilés traditionnels, pour lesquels le compilateur crée un fichier binaire directement exécutable par un processeur donné (c'est-à-dire un fichier binaire contenant des instructions spécifiques à un processeur), le code source Java est compilé en un langage intermédiaire (appelé **pseudo-code** ou **bytecode**) dans un fichier portant le même nom que le fichier source à l'exception de son extension **(.class)**.



### Portabilité et machine virtuelle(JVM)

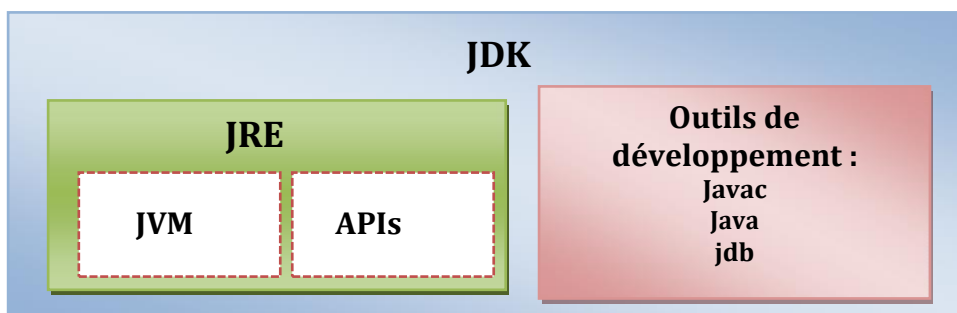
- Cette caractéristique est majeure, car c'est elle qui fait qu'un programme écrit en Java est **portable**, c'est-à-dire qu'il ne dépend pas d'une plate-forme donnée. En réalité le code intermédiaire n'est exécutable sur aucune plate-forme sans la présence d'une **machine virtuelle, un interpréteur** (la machine virtuelle est d'ailleurs parfois appelée interpréteur Java) tournant sur une plate-forme donnée, et capable d'interpréter le code intermédiaire.
- Ainsi, pour peu qu'une plate-forme (windows 95, Unix, Linux, Amiga, ...) possède une

machine virtuelle fonctionnant sous son système, celle-ci est capable d'exécuter n'importe quelle application Java!



### ***Les outils pour programmer Java(JDK, JRE, JVM, API)***

- Pour coder en Java, il faut disposer d'un **JDK** (Java Development Kit) .
- Celui-ci est composé d'un **JRE** (JAVA Runtime Environment) et d'un ensemble d'outils (developers tools) permettant le développement d'application JAVA.
- La JRE quant à elle repose sur la JVM (JAVA Virtual Machine) et propose un ensemble de bibliothèques (les APIs) pour simplifier l'implémentation des programmes Java.



### **JVM (Java Virtual Machine - Machine Virtuelle Java)**

Le programme qui interprète le code Java et le convertit en code natif.

### **API Java ((Application Programming Interface)**

Une API a pour objet de faciliter le travail d'un programmeur en lui fournissant les outils de base nécessaires à tout travail à l'aide d'un langage donné. Elle constitue une interface servant de fondement à un travail de programmation plus poussé.

- Une (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)

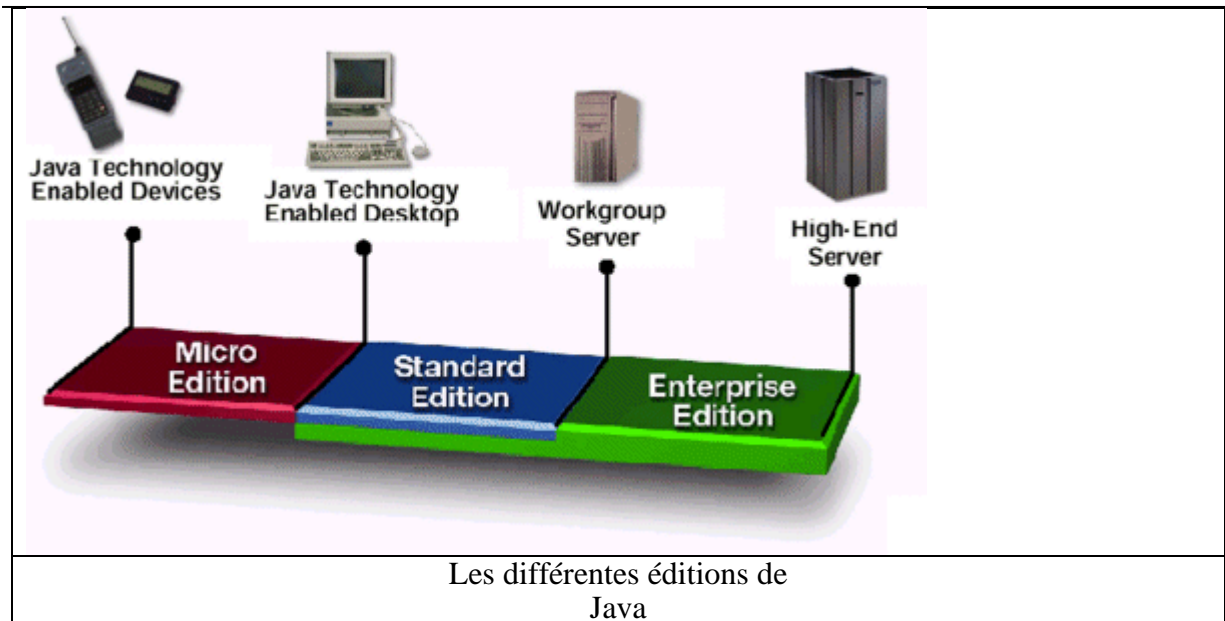
### Java Development Kit (JDK) :

- Le JDK est l'environnement dans lequel le code Java est compilé pour être transformé en bytecode afin que la JVM (machine virtuelle de Java) puisse l'interpréter.
- Les composants primaires du JDK sont une sélection d'outils de programmation, incluant :
  - **javac** – le compilateur, qui convertit le code source en fichier .class (contenant le bytecode Java)
  - **java** – l'interpréteur de bytecode
  - **jar** – l'archiveur, qui met sous forme d'un paquetage unique l'ensemble des fichiers class en un fichier JAR,
  - **javadoc** – le générateur de documentation, qui génère automatiquement de la documentation à partir des commentaires du code source,
  - **jdb** – le débogueur,

### *Les différentes éditions de java*

Il existe 3 éditions de java

- **J2SE (java 2 Standard Edition )**: environnement d'exécution et ensemble complet d'API pour des applications de type **desktop**. Cette plate-forme sert de base en tout ou partie aux autres plate-formes.
- **J2EE (java 2 Enterprise Edition)** : destinés aux serveurs **Web**,
- **J2ME java 2 Mobile Edition)** : environnement d'exécution et API pour le développement d'applications sur **appareils mobiles** et **embarqués** dont les capacités ne permettent pas la mise en œuvre de Java SE



### Environnement de développement intégré (IDE)

- Un IDE n'est pas obligatoire pour programmer Java, mais il amène un confort d'édition, d'accès rapide à la compilation, et d'exécution et de débogage.
- Commencer par utiliser un IDE simple et gratuit tel que : **Jcreator** ([www.jcreator.com](http://www.jcreator.com)), **Eclipse SDK** ([www.eclipse.org](http://www.eclipse.org)), ou la version d'évaluation de **JPadPro**([www.modelworks.com](http://www.modelworks.com)).



### (installer le JDK et l'API)

### L'installation d'abord !

- Se procurer le Kit de développement Java (**JDK**). Vous pouvez le télécharger à partir de ([oracle.com](http://oracle.com))
- Pour nos TP vous pouvez utiliser le **JDK8** qui contient tout ce qu'il faut pour développer et exécuter les programmes Java.
- Exécuter pour cela le fichier **jdk-8u101-windows-x64.exe**
- Vérifier la réussite de l'installation en tapant « **java -version** » dans une fenêtre de

commandes.

- Vous pouvez également télécharger depuis le site oracle la documentation relative au jdk que vous venez d'installer. Pour pouvoir consulter la javadoc il suffit de la décompresser dans un dossier.
- Veillez à ce que la doc soit dans le sous-répertoire **docs** de votre répertoire d'installation du JDK.
- Régler **la variable d'environnement PATH** en lui affectant le chemin qui mène au dossier **bin** du JDK.