Présentation de Java

Qu'est-ce que Java
Ses caractéristiques
Les outils
Les APIs
Réferences

XH

Ressource

- http://docs.oracle.com/javase/specs/index.html

XH

Qu'est-ce que Java ?

- Un langage de programmation orienté objets
- Une architecture de *Virtual Machine*
- Un ensemble d'API variées
- Un ensemble de commandes (outils) (le JDK)

XH

Bref historique

- **1993** : projet Oak (langage pour l'électronique grand public)
- **1995** : Java / HotJava à WWW3
- **Mai 1995** : Netscape prend la licence
- **Sept. 1995** : JDK 1.0 b1
- **Déc. 1995** : Microsoft se dit intéressé
- **Janv. 1996** : JDK 1.0.1
- **Eté 1996** : Java Study Group ISO/IEC JTC 1/SC22
- **Fin 1996** : RMI, JDBC, JavaBeans, ...
- **Fév. 1997** : JDK 1.1
- **1999** : JDK 1.2 (JAVA 2)
- **2006** : JDK 1.5
- **2008** : JDK 6
- ...
- **2015** : JDK 8
- **2017** : JDK 9
- **2018** : JDK 10

XH

Les caractéristiques du langage Java

- Syntaxe proche du langage C
- Orienté objets
- Compilation partiel avant le runtime
- Portable
- Simple
- Robuste
- Sécurisé
- Multi-threads
- Distribué

XH

Java est un langage orienté objets

- Tout est classe (fabrique à objet) sauf les types primitifs (`int`, `float`, `double`, `...`)
- Toutes les classes dérivent de **`java.lang.Object`**
- Héritage simple pour les classes
- Héritage multiple pour les interfaces
- Les objets se manipulent via des références
- Une API objet standard est fournie
- La syntaxe est proche de celle de C

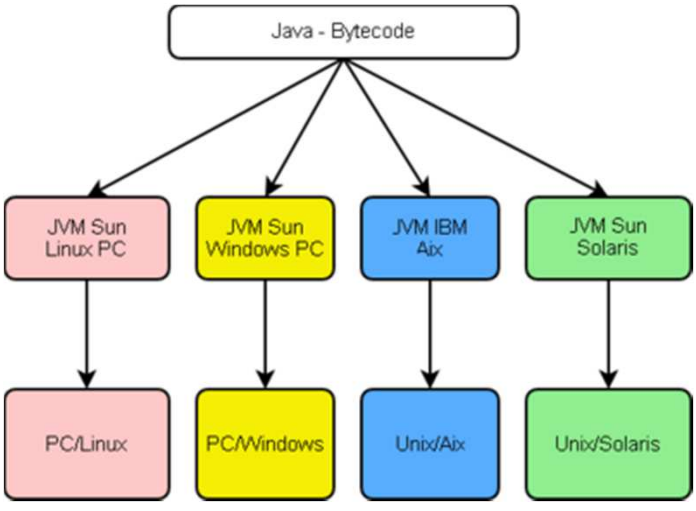
XH

Java est portable

- Le compilateur Java génère du *byte code*
- La Java Virtual Machine (JVM) est présente sur Unix, Win32, Mac, chrome, Firefox, IE, ...
- Le langage a une sémantique très précise
- La taille des types primitifs est indépendante de la plate-forme
- Java supporte un code source écrit en Unicode
- Java est accompagné d'une librairie standard

XH

Portabilité d'un logiciel écrit en Java



XH

Quelques application écrites en Java et portable

LibreOffice
The Document Foundation

- Jenkins

- Rappel: une jre est nécessaire

XH

Java est robuste

- A l'origine, c'est un langage pour les applications embarquées
- Gestion de la mémoire par un *garbage collector*
- Pas d'accès direct à la mémoire.
- Mécanisme d'exception
- Accès à une référence `null` → exception
- compilateur contraignant (erreur si exception non gérée, si utilisation d'une variable non affectée, ...)
- Tableaux = objets (taille connue, débordement → exception)
- Seules les conversions sûres sont automatiques
- Contrôle des *cast* à l'execution

XH

Java est sécurisé

- Indispensable avec le code mobile
- Pris en charge dans l'interpréteur
- Trois couches de sécurité :
  - *Verifier* : vérifie le *byte code*
  - *Class Loader* : responsable du chargement des classes
  - *Security Manager* : accès aux ressources.
- Code certifié par une clé

XH

Java est multi-thread

- Intégrés au langage et aux API :
  - `synchronized`
  - *`garbage collector`* `dans un thread de basse priorité`
  - `java.lang.Thread, java.lang.Runnable`
- Accès concurrents à objet gérés par un *monitor*
- Implémentation propre à chaque JVM
- Difficultés pour la mise au point et le portage

XH

Java est distribué

- API réseau (java.net.Socket, java.net.URL, ...)
- Chargement / génération de code dynamique
- Applet
- Servlet
- *Protocole | Content handler*
  - *Remote Method Invocation (RMI)*
  - *IIOP (CORBA)*

XH

Les performances

- Actuellement le *byte code* est compilé avant exécution
- Plusieurs types de génération de code machine :
  - Conversion statique en C (j2c, Tabo, ...)
  - Conversion statique en code natif
  - Compilation en code machine à la volée (JIT)

XH

Les différences avec C++

- Pas de structures ni d'unions
- Pas de *typedef*
- Pas de préprocesseur
- Pas de variables ni de fonctions en dehors des classes
- Pas d'héritage multiple de classes
- Pas de surcharge d'opérateurs
- Pas de passage par copie pour les objets
- Pas de pointeurs, seulement des références d'objet

XH

Les nouveautés JDK 5 (2006)

- JMX (JSR3)
- Arithmétique (JSR13)
- Generics (JSR14)
- SASL (JSR28)
- JDBC Rowset (JSR114)
- Threading (JSR133)
- JMX Remote (JSR160)
- Profiling (JSR163)
- Concurrency Utilities (JSR166)
- JVM Monitoring (JSR174)
- Metadata (JSR175)
- Transfert d'archives (JSR200)
- Enum, autoboxing, foreach(JSR201)
- Unicode (JSR204)
- JAXP 1.3 (JSR206)

XH

Les nouveautés JDK 8 (2016)

- Interface fonctionnelle
- Lambda expression
- …, etc.

XH

Les outils (1)

- **Environnements de développement** :
  - Sun JDK 8U152 (compilateur, interpréteur, *appletviewer*,...)
  - IDE : Eclipse, IntelliJ de Jetbrains (ideal'j), NetBeans, JDeveloper ...
- **Browsers** :
  - Firefox
  - Internet Explorer -Edge
  - Chrome

XH

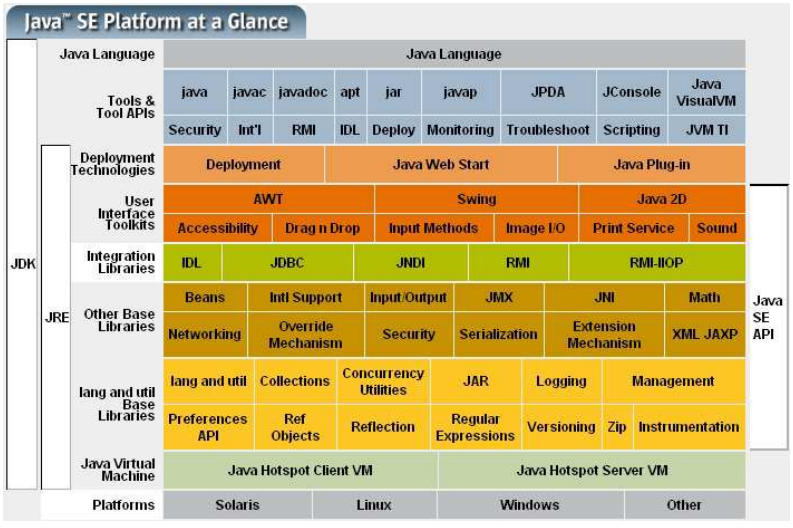
Les outils (2)
du Java Development Kit (Java SE)

- **javac** : compilateur de sources java
- **java** : interpréteur de *byte code* (la Machine Virtuelle)
- **javadoc** : générateur de documentation (HTML, MIF)
- **jar**: créateur de JAR
- **appletviewer** : interpréteur d'applet
- **javah** : générateur de *header* pour l'appel de méthodes natives
- **javap** : désassembleur de *byte code*
- **jdb** : debugger
- **javakey** : générateur de clés pour la signature de code
- **rmic** : compilateur de *stubs* RMI
- **rmiregistry** : "*Object Request Broker*" RMI

XH

Vue très générale des APIs (JDK8)

- A chacune des cases correspond un ou des packages (ex: java.lang, java.util)



XH

Les packages du *core* API 1.0.2 et 1.1

- **java.lang** : Types de bases, Threads, ClassLoader, Exception, Math, ...
- **java.util** : Hashtable, Vector, Stack, Date, ...
- **java.applet**
- **java.awt** : Interface graphique portable
- **java.io** : accès aux i/o par flux, wrapping, filtrage
- **java.net** : Socket (UDP, TCP, multicast), URL, ...

XH

Les packages du *core* API 1.1

- **java.lang.reflect** : introspection sur les classes et les objets
- **java.beans** : composants logiciels
- **java.sql** (JDBC) : accès homogène aux bases de données
- **java.security** : signature, cryptographie, authentification
- **java.serialisation** : sérialisation d'objets
- **java.rmi** : *Remote Method Invocation*
- **java.idl** : interopérabilité avec CORBA

XH

Les autres API

- **Java Server :** jeeves / servlets
- **Java Commerce*** : JavaWallet
- **Java Management** (JMAPI) : gestion réseau
- **Java Média** : 2D*, 3D, Média Framework, Share, Animation*, Telephony

(*) Feront partie, à terme, des *core* API

XH

Références

- La page wikipédia de Java
- Thinking in Java
  - B. Eckel. http://www.EckelObject.com/Eckel

XH