

Chap1(p2): JAVA

Présentation et setting up



Plan

- Origines du langage
 - Caractéristiques du langage
 - Domaines d'applications
 - Les différentes facettes de Java
 - JAVA Actuellement
 - JAVA Vs C/C++
 - Le compilateur et la JVM
-

Origines du langage

- **1994** "First Person Team" -> Oak -> première applet pour le web
 - **1995** Java est lancé sur le Web supporté par HotJava
 - **1997** Des millions d'utilisateurs du Web utilisent Java
 - Java entre dans la conception d'applications d'entreprises
 - **1998-1999** optimisation des compilateurs Java pour arriver à concurrencer C++ sur la performance
-

Caractéristiques du langage

- Indépendant de l'architecture matériel
 - Interprété (MV)
 - Tout est objet, tout est typé
 - Multithreads
 - Dynamique (classe loader)
 - Robuste et sécurisé
-

Domaines d'applications

- Créé pour les systèmes embarqués disposants de ressources CPU et mémoire limitée et hétérogène
 - Java a été très rapidement adopté pour rendre plus "sexy" les pages web : petites animations, son, vidéo, fonctionnalités
 - Aujourd'hui Java commence sa percée dans l'informatique d'entreprise.
-

Les différentes facettes de Java

- Environnement graphique
 - Langage de programmation
 - Environnement de programmation
-

JAVA Actuellement



Application desktop

- JDBC
- AWT
- SWING
- FX



Technologies J2EE

- JSP /SERVLET / JSTL
- STRUTS / SPRING
- JSF
- Primefaces
- Hibernate / JPA



Développement mobile

- Android



Java Vs C/C++

- Pas de pointeur
 - Pas de structure ou d'union struct, defstruct
 - Pas de macros #define
 - Pas de gestion de la mémoire laissée à l'utilisateur malloc(), free(), new(), delete()
 - Pas d'héritage multiple(direct: héritage de structure)
 - Pas de redéfinition d'opérateur
 - Vérification à l'exécution des dépassements de bornes de tableaux
-

Le compilateur

- Un programme Java (extension .java) est compilé en pseudo-code (byte code) et stocké dans un fichier .class
 - Le **pseudo-code n'est pas exécutable directement**, il est interprété par la machine virtuelle Java. C'est ce niveau d'indirection qui fait la flexibilité et la portabilité du code Java.
 - Ligne de commande de compilation
 - `javac -option nomfic.java`
-

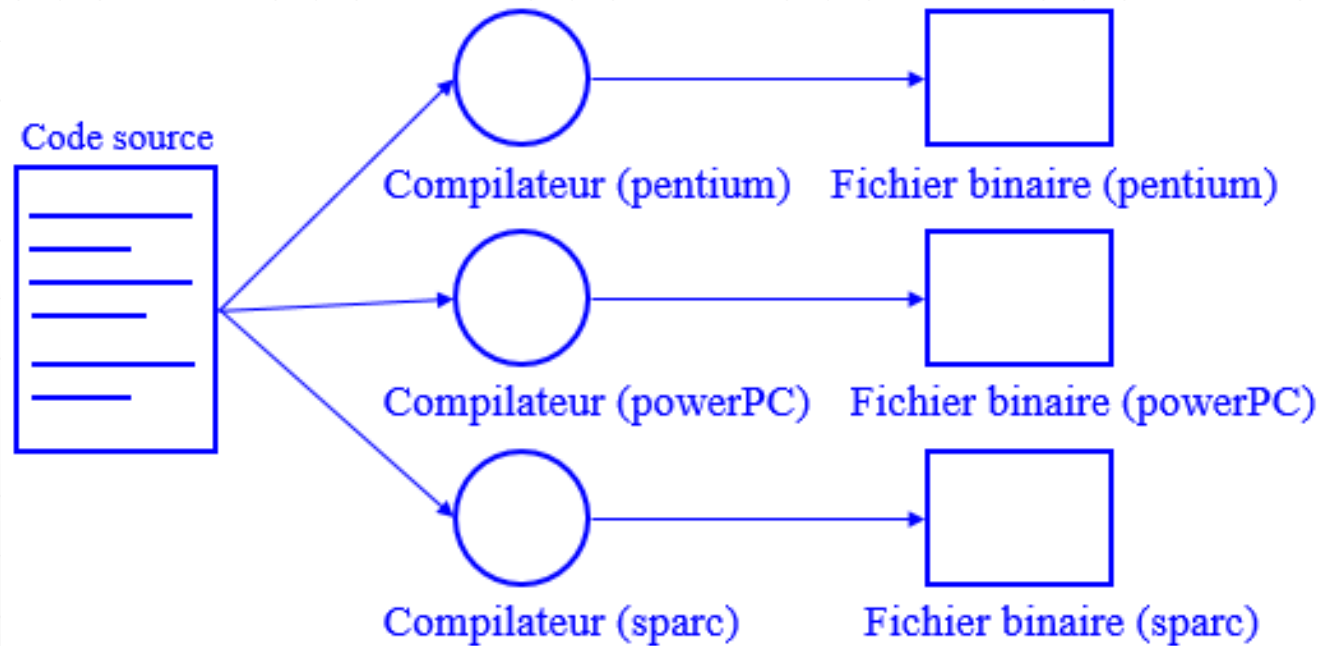
Le compilateur

- Les options de **Javac**
 - classpath** chemin spécifie le chemin pour accéder aux classes
 - d** répertoire Spécifie le répertoire racine de la hiérarchie des fichiers de classe
 - g** inhibe la génération de table de débogage
 - nowarn** inhibe les warnings
 - O** Optimise le code
 - depend** recompile toutes les classes interdépendantes
-

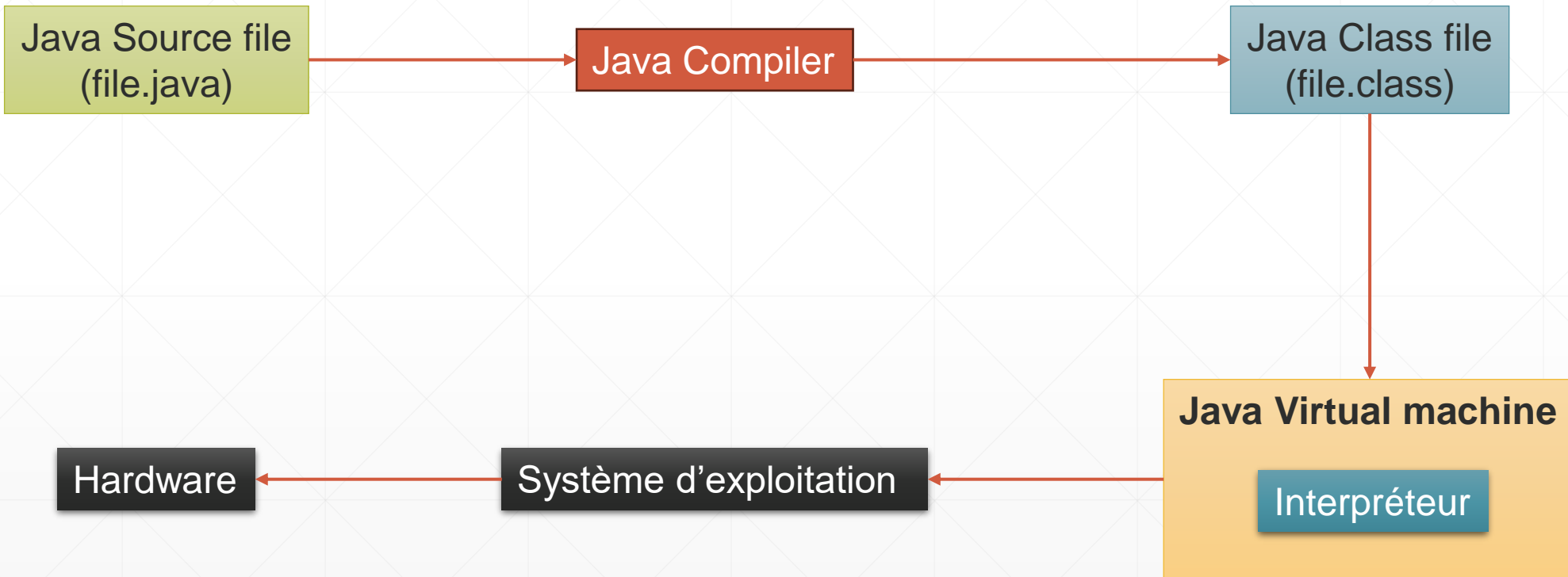
La machine virtuelle

- Permet de générer du code portable
 - Le "byte-code" Java est interprété par la MV
 - Plus lent que le code compilé de C++
 - La MV est indépendante de l'architecture matériel
 - Elle est développée pour chaque environnement en fonctions des spécifications SUN
-

Compilation usuelle



Compilation JAVA



JVM
(Java Virtual Machine)

Ramasse miette

- Gestion de la mémoire "automatique"
 - Le programmeur n'a plus besoin lui-même de libérer des objets. Plus de FREE ou autre DELETE
 - Les objets sont détruits lorsqu'ils ne sont plus utilisés
 - Pour des applications critiques une destruction manuelle des objets reste possible par appel explicite du ramasse miette. Classe Runtime
 - **Méthodes** :getRuntime(), gc(), runFinalization()
-

Le chargeur de classe

- Les classes sont chargées dans un "**NameSpace**" indépendant et identifié en fonction de l'origine du code chargé
 - Le chargeur de classe fait partie du système de sécurité de java en séparant les classes locales et les classes venant du réseau
 - Lorsque l'on a besoin d'une classe donnée, la recherche est effectuée en priorité parmi les classes locales. Il est très difficile à une classe réseau de se faire passer pour une classe locale même en usurpant le nom
 - Le chemin de recherche des classes est fourni par la variable d'environnement **CLASSPATH**
-

Le JDK (Java Development Kit)

- Le JDK est un ensemble d'outils dont un développeur a besoin pour réaliser des programmes en java
- Les principaux outils qui le composent sont :
 - **javac** : le compilateur java
 - **java**: la machine virtuelle
 - **javadoc** : le générateur de documents
 - **appletviewer** : outils de test des applets
 - **Jdb** : le debugger java
 - **Javap** : le désassembleur java. Permet de convertir un fichier .class en un fichier lisible



Premier programme

- Exemple de d'une première classe qui affiche le message HelloWorld

```
/** le code de creation de la classe Etudiant */  
public class Etudiant {  
    /** le code de declaration de la methode principale */  
    public static void main(String[] args) {  
        System.out.println("Hello world ");  
    }  
}
```

Premier programme

Compilation des classes java en utilisant l'invite de commande

- **Compilation de la classe**

- javac Etudiant.java

- **Exécution de la classe**

- java Etudiant

- **Génération de la documentation**

- javadoc Etudiant.java

```
F:\UIR\Semestre 1\Info 2 - Programmation orientée objet 1\exemple java
λ java Etudiant
Hello world
```