Le langage JAVA

Introduction

SOMMAIRE

- Introduction
- Caractéristiques du langage JAVA

CHAPITRE 1

INTRODUCTION

HISTORIQUE DE JAVA (1/2)

- Java a été développé à partir de décembre 1990 par une équipe de Sun Microsystems dirigée par James Gosling
- Au départ, il s'agissait de développer un langage de programmation pour permettre le dialogue entre de futurs ustensiles domestiques
- Or, les langages existants tels que C++ ne sont pas à la hauteur : recompilation dès qu'une nouvelle puce arrive, complexité de programmation pour l'écriture de logiciels fiables…
- **1990** : Écriture d'un nouveau langage plus adapté à la réalisation de logiciels embarqués, appelé **OAK**
  - Petit, fiable et indépendant de l'architecture
  - Destiné à la télévision interactive
  - Non rentable sous sa forme initiale
- **1993** : le WEB « décolle », Sun redirige ce langage vers Internet : les qualités de portabilité et de compacité du langage OAK en ont fait un candidat parfait à une utilisation sur le réseau. Cette réadaptation prit près de 2 ans.
- **1995** : Sun rebaptisa OAK en **Java** (nom de la machine à café autour de laquelle se réunissait James Gosling et ses collaborateurs)

HISTORIQUE DE JAVA (2/2)

Les développeurs Java ont réalisé un **langage indépendant de toute architecture** de telle sorte que Java devienne idéal pour programmer des applications utilisables dans des réseaux hétérogènes, notamment Internet.

Le développement de Java devint alors un enjeu stratégique pour Sun et l'équipe écrivit un navigateur appelé HotJava capable d'exécuter des programmes Java.

La version 2.0 du navigateur de Netscape a été développée pour supporter Java, suivi de près par Microsoft (Internet Explorer 3)

L'intérêt pour la technologie Java s'est accru rapidement: IBM, Oracle et d'autres ont pris des licences Java.

LES DIFFÉRENTES VERSIONS DE JAVA

De nombreuses versions de Java depuis 1995

- Java 1.0 en 1995
- Java 1.1 en 1996
- Java 1.2 en 1999 (Java 2, version 1.2)
- Java 1.3 en 2001 (Java 2, version 1.3)
- Java 1.4 en 2002 (Java 2, version 1.4)
- Java 5 en 2004
- Java 6 en 2006 *celle que nous utiliserons dans ce cours*
- Java 7 en 2011

Évolution très rapide et succès du langage

Une certaine maturité atteinte avec Java 2

Mais des problèmes de compatibilité existaient

- entre les versions 1.1 et 1.2/1.3/1.4
- avec certains navigateurs

QUE PENSER DE JAVA?

Les plus :

- Il est indépendant du système d'exploitation, d'où une plus grande compatibilité des applications et facilité de développement
- Il a su bénéficier de l'essor d'Internet et s'imposer dans de nombreux domaines
- Un environnement gratuit et de nombreux outils disponibles dont beaucoup gratuits aussi
- Une large communauté très active
- Base du développement pour les applications Androïd

Les moins :

- Trop 'médiatisé'?
- Problèmes de compatibilité
  - ✓ Avec les premières versions
  - ✓ Avec certains navigateurs (les navigateurs ne sont pas écrits par Sun)
- Problèmes de vitesse, mais existence de solutions pour y pallier (compilateur natif, compilation du bytecode à la volée)

RÉFÉRENCES (1)

Bibliographie

- Au cœur de Java 2 : Volume I - Notions fondamentales.
  C. Hortsmann et G. Cornell. The Sun Microsystems Press.
  Java Series. CampusPress.
- Au cœur de Java 2 : Volume II - Fonctions avancées.
  C. Hortsmann et G. Cornell. The Sun Microsystems Press.
  Java Series. CampusPress.
- Passeport pour l'algorithmique objet. Jean-Pierre Fournier. Thomsom Publishing International.
- Java la maîtrise : Jérôme Bougeault - Eyrolles
- Développons en JAVA (avec Eclipse) : Jean-Michel Doudoux, disponible gratuitement en format PDF : http://www.jmdoudoux.fr/accueil.html

RÉFÉRENCES (2)

Webographie

- Pour récupérer le kit de développement de Sun
  - http://java.sun.com/ (racine du site)
  - http://www.eclipse.org/babel/ (traductions)
- Outils de développement
  - Eclipse : http://www.eclipse.org
  - NetBeans : https://fr.netbeans.org/ ou https://netbeans.org/
- Une source d'exemples de programmes commentés
  - http://www.oracle.com/technetwork/java/index.html (en anglais)

DU PROBLÈME AU PROGRAMME

Nécessité d'analyser le problème pour pouvoir le traduire en une solution informatique
(cela semble évident, mais pourtant!)

- avant de construire un bâtiment, on fait un plan. Ce n'est pas différent en informatique : conception de l'algorithme, i.e. une réponse (rationnelle) au problème posé
- puis mise en œuvre technique - le *codage* - dans un langage de programmation, dans notre cas Java.

ANALYSE DU PROBLÈME (1)

Se poser les bonnes questions

- Quelles sont les objets qui interviennent dans le problème? Quelles sont les données, les objets, que le programme va manipuler?
- Quelles vont être les relations entre ces objets? Quelles sont les opérations que je vais pouvoir effectuer sur ces objets?

ANALYSE DU PROBLÈME (2)

Savoir être :

- **efficace** : quelle méthode me permettra d'obtenir le plus vite, le plus clairement, le plus simplement possible les résultats attendus ?
- **paresseux** : dans ce que j'ai développé avant, que puis-je réutiliser ?
- **prévoyant** : comment s'assurer que le programme sera facilement réutilisable et extensible ?

CHAPITRE 2

CARACTÉRISTIQUES DU LANGAGE JAVA

CARACTÉRISTIQUES DU LANGAGE JAVA (1)

Simple

- Apprentissage facile
  - faible nombre de mots-clés
  - simplifications des fonctionnalités essentielles
- Développeurs opérationnels rapidement

Familier

- Syntaxe proche de celle de C/C++

CARACTÉRISTIQUES DU LANGAGE JAVA (2)

Orienté objet

- Java ne permet d'utiliser que des objets *(hors les types de base)*
- Java est un *langage objet* de la famille des langages de *classe* comme C++ ou SmallTalk
- Les grandes idées reprises sont : encapsulation, dualité classe /instance, attribut, méthode / message, visibilité, dualité interface/implémentation, héritage simple, redéfinition de méthodes, polymorphisme

Sûr

- Seul le bytecode est transmis, et «vérifié» par l'interpréteur
- Impossibilité d'accéder à des fonctions globales ou des ressources arbitraires du système

CARACTÉRISTIQUES DU LANGAGE JAVA (3)

Fiable

- Gestion automatique de la mémoire
  (ramasse-miette ou "garbage collector")
- Gestion des exceptions
- Sources d'erreurs limitées
  - ✓typage fort,
  - ✓pas d'héritage multiple,
  - ✓pas de manipulations de pointeurs, etc.
- Vérifications faites par le compilateur facilitant une plus grande rigueur du code

CARACTÉRISTIQUES DU LANGAGE JAVA (4)

Java est **indépendant de l'architecture**

- Le bytecode généré par le compilateur est indépendant de toute architecture. Toute application peut donc tourner sur une plate-forme implémentant une machine virtuelle Java

« Ecrire une fois, exécuter partout »

Java est **multi-tâches**

- Exécution de plusieurs processus effectuant chacun une tâche différente
- Mécanismes de synchronisation
- Fonctionnement sur des machines multiprocesseurs

JAVA, UN LANGAGE DE PROGRAMMATION

- Applications Java : programmes autonomes, "stand-alone"
- Applets (mini-programmes) : Programmes exécutables uniquement par l'intermédiaire d'une autre application
  - navigateur web : Netscape, Internet explorer, Hotjava
  - application spécifique : Appletviewer
- Java est souvent considéré comme étant uniquement un langage pour écrire des applets alors que c'est aussi un vrai langage de programmation
- Java est souvent confondu avec le langage de script Javascript auquel il n'est en aucune manière apparenté

JAVA, UN LANGAGE INDÉPENDANT? (1)

Java est un langage interprété

- La compilation d'un programme Java crée du pseudo-code portable : le "byte-code"
- Sur n'importe quelle plate-forme, une machine virtuelle Java peut interpréter le pseudo-code afin qu'il soit exécuté

Les machines virtuelles Java peuvent être

- des interpréteurs de byte-code indépendants (pour exécuter les programmes Java)
- contenues au sein d'un navigateur
  (pour exécuter des applets Java)

JAVA, UN LANGAGE INDÉPENDANT? (2)

Avantages :

- Portabilité
  - ✓Des machines virtuelles Java existent pour de nombreuses plates-formes dont : Linux, Windows, MacOS
- Développement plus rapide
  - ✓courte étape de compilation pour obtenir le byte-code,
  - ✓pas d'édition de liens,
  - ✓déboguagge plus aisé,
- Le byte-code est plus compact que les exécutables
  - ✓pour voyager sur les réseaux.

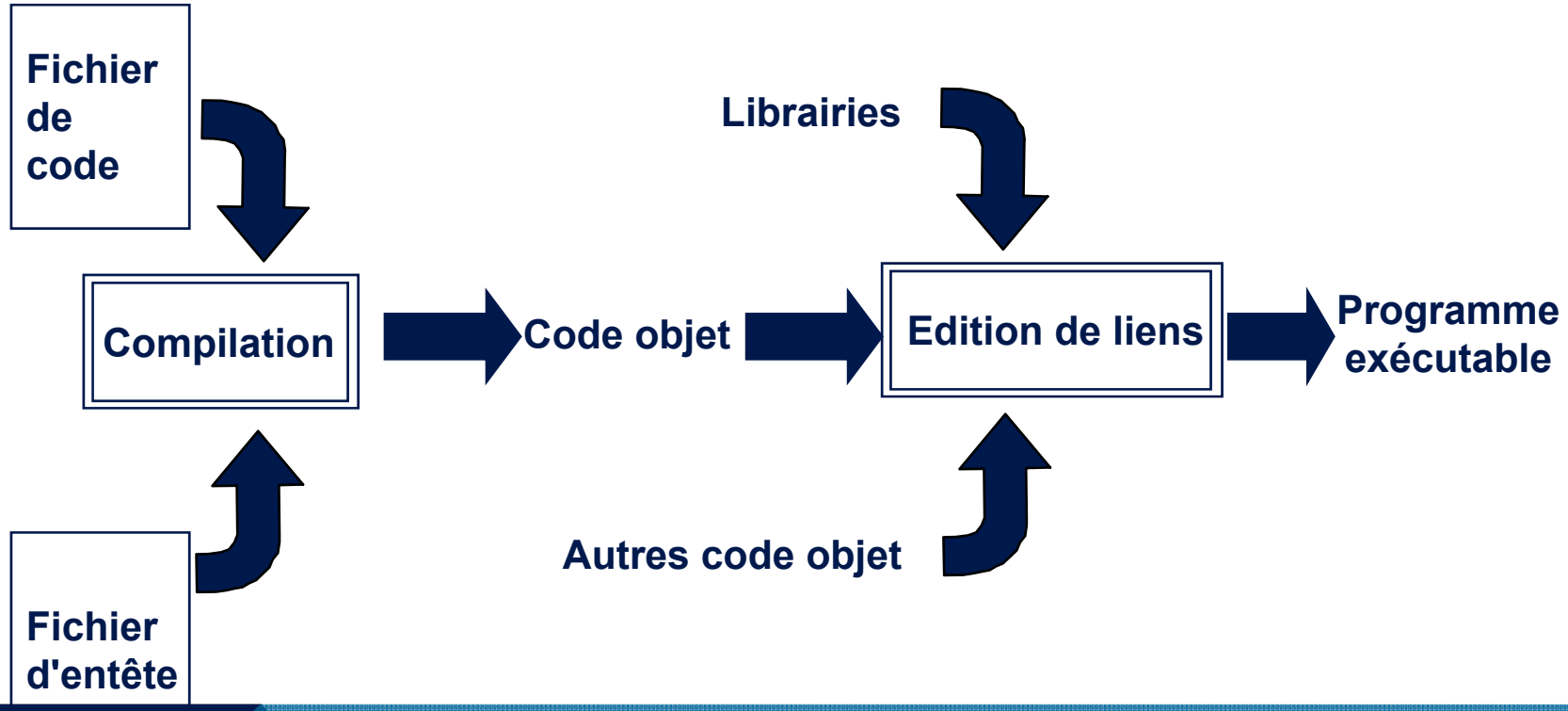
JAVA, UN LANGAGE INDÉPENDANT? (3)

Inconvénients :

- Nécessite l'installation d'un interpréteur pour pouvoir exécuter un programme Java
- L'interprétation du code ralentit l'exécution
- Les applications ne bénéficient que du dénominateur commun des différentes plate-formes
  - limitation, par exemple, des interfaces graphiques
- Gestion gourmande de la mémoire
- Impossibilité d'opérations de « bas niveau » liées au matériel

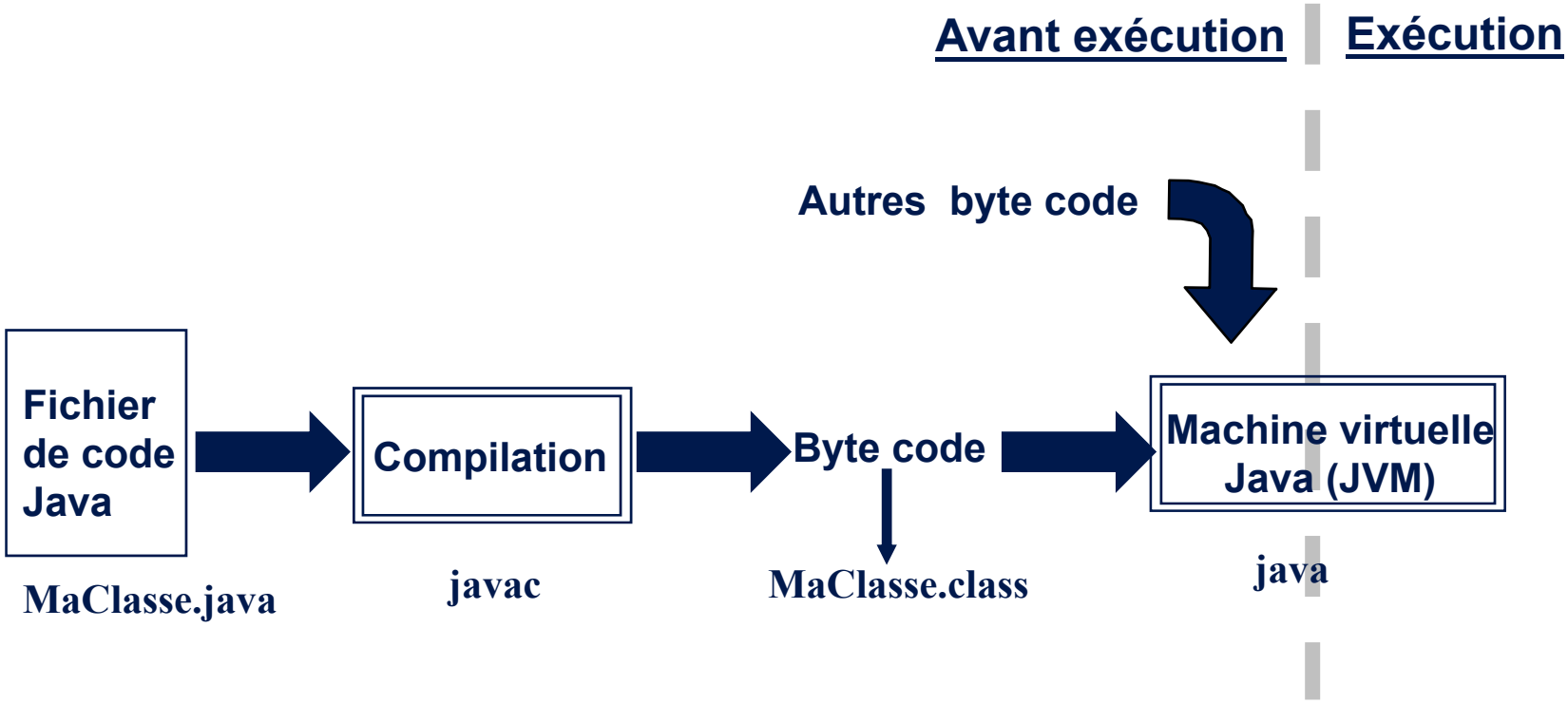
LANGAGE COMPILÉ

Etapes qui ont lieu avant l'exécution pour un langage compilé comme C++



LANGAGE INTERPRÉTÉ

Cas de Java



L'API DE JAVA

Java fournit de nombreuses librairies de classes remplissant des fonctionnalités très diverses : c'est l'API Java

- API (Application and Programming Interface /Interface pour la programmation d'applications) : Ensemble de bibliothèques permettant une programmation plus aisée car les fonctions deviennent indépendantes du matériel.

Ces classes sont regroupées, par catégories, en paquetages (ou "packages").

L'API DE JAVA (2)

Les principaux paquetages

- java.util : structures de données classiques
- java.io : entrées / sorties
- java.lang : chaînes de caractères, interaction avec l'OS, threads
- java.applet : les applets sur le web
- java.awt : interfaces graphiques, images et dessins
- javax.swing : package récent proposant des composants « légers » pour la création d'interfaces graphiques
- java.net : sockets, URL
- java.rmi : Remote Method Invocation
- java.sql : fournit le package JDBC

L'API DE JAVA (3)

La documentation de Java est standard, que ce soit pour les classes de l'API ou pour les classes utilisateur

- possibilité de génération automatique avec l'outil Javadoc.

Elle est au format HTML.

- intérêt de l'hypertexte pour naviguer dans la documentation

Pour chaque classe, il y a une page HTML contenant :

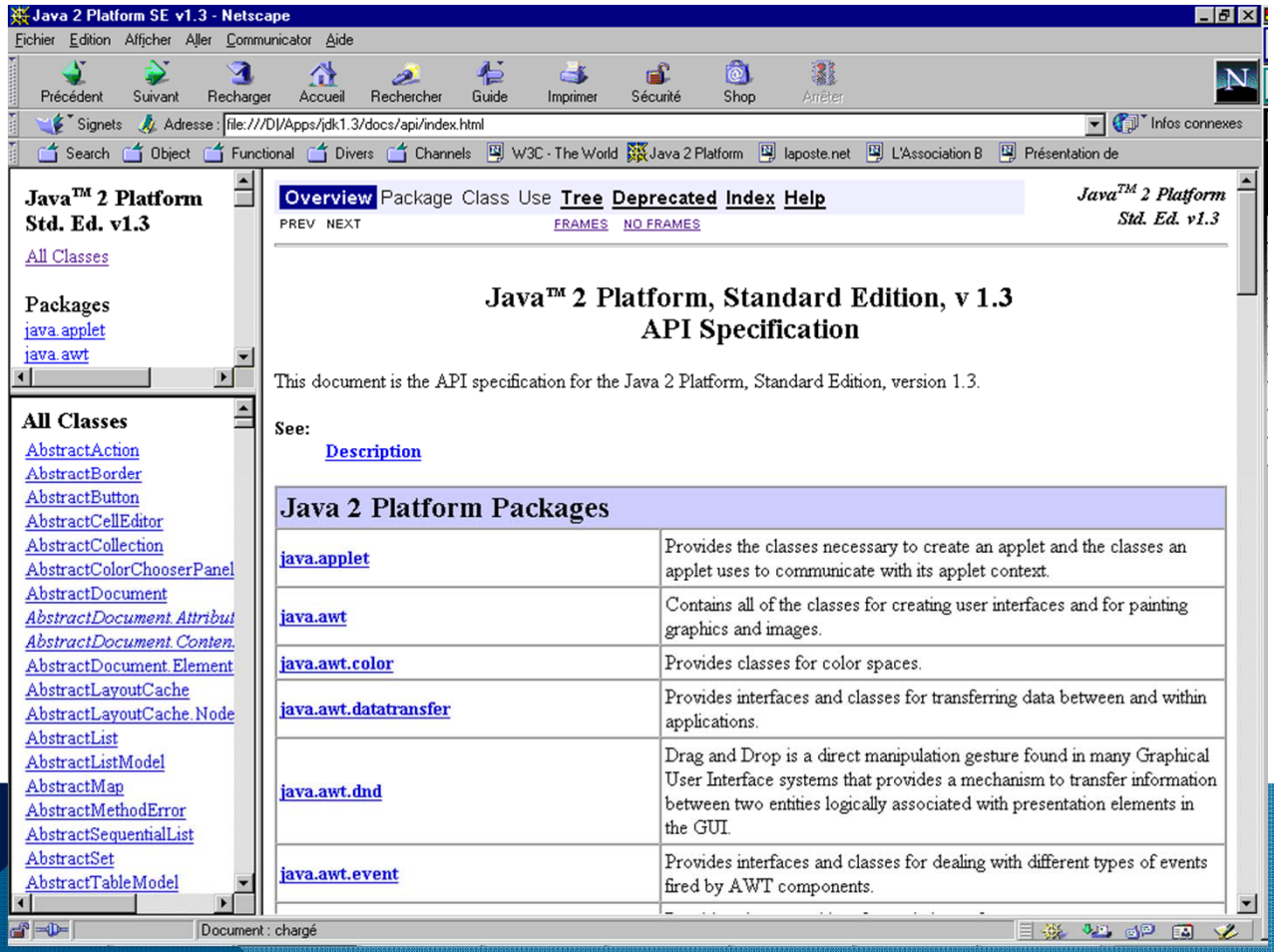
- la hiérarchie d'héritage de la classe,
- une description de la classe et son but général,
- la liste des attributs de la classe (locaux et hérités),
- la liste des constructeurs de la classe (locaux et hérités),
- la liste des méthodes de la classe (locaux et hérités),
- puis, chacune de ces trois dernières listes, avec la description détaillée de chaque élément.

L'API DE JAVA (4)

Où trouver les informations sur les classes de l'API

- sous le répertoire jdk1.x/docs/api dans le JDK
  - les documentations de l'API se téléchargent et s'installent (en général) dans le répertoire dans lequel on installe java.
    Par exemple si vous avez installer Java dans le répertoire D:/Apps/jdk1.4/, vous décompresser le fichier zip contenant les documentations dans ce répertoire.
    Les docs de l'API se trouveront alors sous :
    D:/Apps/jdk1.4/docs/api/index.html
- Sur le site de Sun, on peut la retrouver à http://java.sun.com/docs/index.html

L'API DE JAVA (5)



OUTIL DE DÉVELOPPEMENT : LE JDK

Environnement de développement fourni par Sun

JDK signifie Java Development Kit (Kit de développement Java).

Il contient :

- les classes de base de l'API java (plusieurs centaines),
- la documentation au format HTML
- le compilateur : javac
- la JVM (machine virtuelle) : java
- le visualiseur d'applets : appletviewer
- le générateur de documentation : javadoc
- etc.

JAVA, UN LANGAGE NOVATEUR?

Java n'est pas un langage novateur : il a puisé ses concepts dans d'autres langages existants et sa syntaxe s'inspire de celle du C++.

Cette philosophie permet à Java

- De ne pas dérouter ses utilisateurs en faisant "presque comme ... mais pas tout à fait"
- D'utiliser des idées, concepts et techniques qui ont fait leurs preuves et que les programmeurs savent utiliser

En fait, Java a su faire une synthèse efficace de bonnes idées issues de sources d'inspiration variées

- Smalltalk, C++, Ada, etc.