



L'histoire de Java

Programmation objet
et technologie de pointe

Programmation objet

- Premières idées (notion de classe) dans SIMULA 67 (*Norwegian Computing Centre* d'Oslo)
- Vrai décollage avec Smalltalk 80 au *Palo Alto Research Center* de Xerox en Californie
- Quelques choix ont freiné le développement de Smalltalk



Prémices de Java

- James Gosling, M. Sheridan et P. Naughton
- Un projet : gérer les contenus interactifs diffusés sur la télévision numérique (câble)
- Un langage : Oak
- Trop avancés pour l'époque : début 1990
- Mais c'était aussi les débuts d'Internet grand public avec le Web...



L'origine de Java

- Un langage objet pour apporter du contenu dynamique aux sites Web
 - HTML = contenu statique
 - GIF = animation (limitée) d'images
 - Vidéo = animations préenregistrées
- Applet Java : (petit) programme téléchargé à partir du serveur et s'exécutant chez le client

Les contraintes nouvelles

- Il faut permettre l'exécution sur des plates-formes multiples
⇒ WORA (*Write Once, Run Anywhere*)
- Il faut restreindre les capacités d'intrusion chez le client
- Il faut assurer la protection du code exécutable face aux modifications

Les contraintes évidentes (et pas nouvelles)

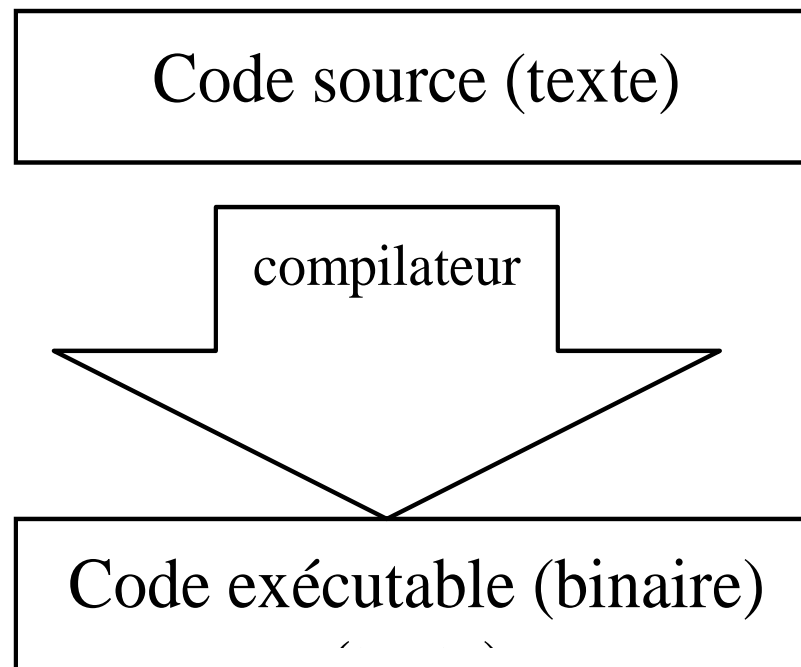
- Code compact (car devant être téléchargé)
- Code facilement exécutable (pour l'intégration aux navigateurs Web)
- Programmes graphiques, interactifs
- Et, si possible, langage utilisable pour des « vraies » applications !

Les réponses de Java

- D'abord et avant tout :
la *programmation objet* !
(Trop tôt pour détailler)
et la conformité avec la syntaxe du langage C
- La technique des *machines virtuelles*
et du *code intermédiaire*

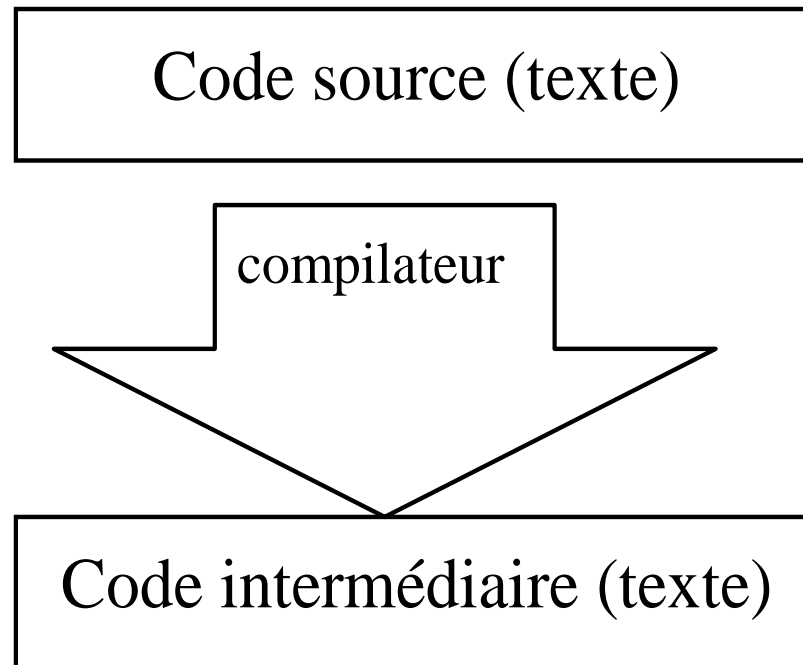
Code intermédiaire (bytecode) (1)

- Habituellement :



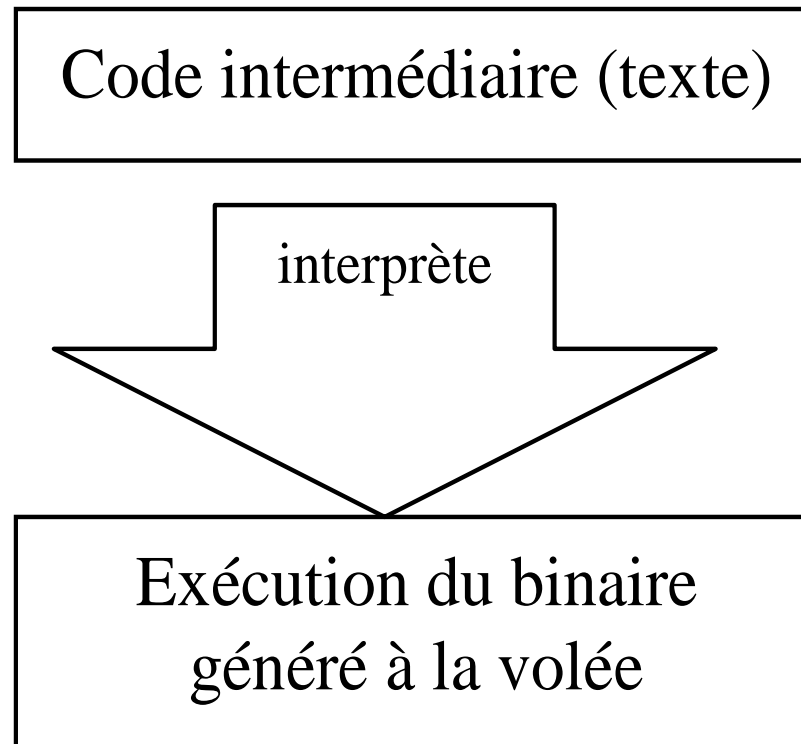
Code intermédiaire (bytecode) (2)

- Le compilateur Java produit un code intermédiaire :



Code intermédiaire (bytecode) (3)

- Pour l'exécution :



Machine virtuelle Java (JVM)

- C'est une machine virtuelle Java qui joue le rôle d'interprète du code intermédiaire
- Il suffit de définir une machine virtuelle par plate-forme et le code Java s'exécute (sans recompilation) sur la plate-forme !



Java actuellement

- Après avoir été développé par Sun
- Après le rachat de Sun par Oracle
- Java poursuit son chemin : 10 millions d'utilisateurs annoncés, utilisé partout et pour tout
(p. ex. enregistrement par Google de 500 millions d'activation de dispositifs sous Android à la mi-septembre 2012).





3 milliards d'appareils exécutent du Java :
des ordinateurs, des imprimantes des téléphones cellulaires,
le BlackBerry, le Kindle (liseuse d'Amazon), des parcs-mètres,
des passes pour les transports publics, des distributeurs bancaires,
des cartes de crédit, des systèmes de sécurité domestique,
des box pour la télévision par câble, des téléviseurs, ...

Diffusion de Java

- Outil gratuit dès le départ
- Devenu (en partie) open source
- Pour les navigateurs Web : le JRE
Java Runtime Environment
- Pour les programmeurs : le JDK
Java Development Kit
- Le JDK contient divers outils dont le compilateur officiel et une machine virtuelle
- Deux versions de Java :
SE et EE (orienté Entreprise, Web Services)

Environnements de développement

- Oracle distribue un EDI : *NetBeans*
- Eclipse Foundation en offre un autre : *Eclipse*

Remarque :

Le compilateur Eclipse n'est pas celui du JDK
L'environnement pour programmation d'interfaces graphiques non plus (SWT) mais Eclipse peut utiliser les environnements standard (AWT et Swing).

- *JCreator* : environnement construit autour des outils du JDK.