Les concepts

Présentation

Java : langage orienté objet inspiré de Smalltalk (polymorphisme dynamique, bibliothèques de classes, garbage collector, ...), d'Ada (paquetage, exception...), dans sa sémantique et de C++ dans sa syntaxe.
Avec environnement (classes de base + debugger)

permet d'écrire des programmes orientés objets puissants (graphique, sons, multimédia, ...).

permet de construire du code chargeable par Internet et interprétable sur de multiples architectures => une partie d'Internet est banque de programmes

La documentation est en ligne à `http://java.sun.com`.

Un programme Java est ou bien :
- un programme à interpréter sans le Web : application autonome (standalone)
- un programme chargeable par Internet à l'aide d'un client Web et exécuté en local à l'aide de l'interpréteur intégré dans le client Web : les applets.

Il peut être les 2.

Le fonctionnement d'une application Java

on interprète un fichier byte-code:
`% java Classe`
où `Classe` est le nom de la classe contenant la méthode `main()`.

exemple

```
// Ma première application Java

// A compiler par :
// javac HelloWorld

// L'exécuter par :
// java HelloWorld

class HelloWorld {
      public static void main (String args[]) {
            System.out.println("Bonjour a tous");
      }
}
```

Exécuter

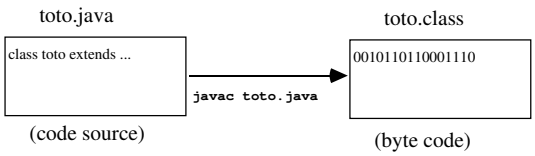
```
% java HelloWorld
Bonjour a tous
%
```

exemples d'application Java :
`jdb` (le débogueur Java), `javac`, `hotjava`, •••

Java : la compilation

C'est le programme `javac`.

construction



Il existe des compilateurs `javac` sur les architectures :
Windows 95, NT, Solaris 2.4, Linux, MacOS 7.5.x, AIX 4.1.3, OSF1.

Le fonctionnement d'une applet

Sur le site serveur de programmes

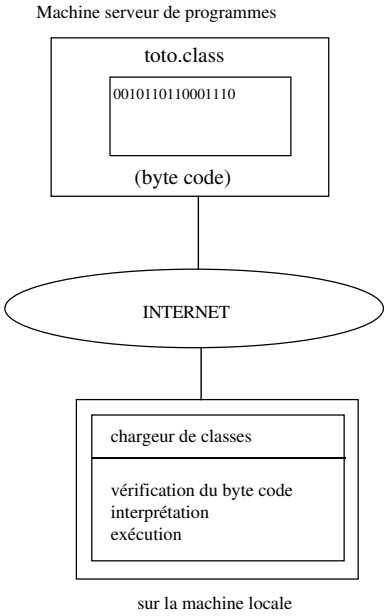
programme Java déjà compilé = Byte code.

En local

une connexion Internet sur le serveur de byte code, un client Web (browser) qui charge ce byte code, un interpréteur de byte code (dans le browser). Le code à charger est une applet : une applet est un programme qui s'exécute dans une page Web.

Le fonctionnement d'une applet (suite)

utilisation



Fonctionnement en local

L'ensemble chargeur de classes + interpréteur est intégré dans le client Web.

Certains clients Web sur une architecture donnée, ne peuvent pas interpréter du byte code Java (exemple Netscape 1.1. N)

Des clients Web (ex `hotjava`) peuvent interpréter du byte code Java version alpha mais pas du byte code version bêta.

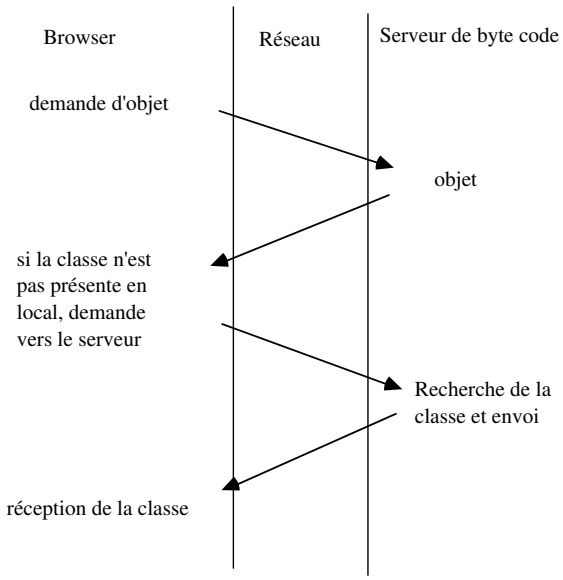
D'autres clients Web (Netscape version 2.0 et suivantes) interprètent tout byte code

En résumé
Java version alpha ≠ Java version bêta = 1.0

`appletviewer` interprète les applets versions béta.

Fonctionnement du chargeur de classes

Le moteur Java charge des objets Java. S'il dispose de la classe en local, il peut interpréter cet objet. Sinon il demande et reçoit sa classe au site serveur de l'objet.



La sécurité

voir
http://java.sun.com/1.0alpha3/doc/security/security.html
4 niveaux de sécurité :

niveau 1 : à la compilation

plus de pointeurs (ni leur arithmétique) donc :
- plus d'accès détourné à des parties privées d'objets
- les tableaux sont des objets
- le mécanisme d'allocation mémoire est sécurisé (et renvoyé à l'exécution)

niveau 2 : par l'interpréteur

interdit l'accès à des parties mémoires interdites, vérifie que les objets sont utilisés à bon escient (un `InputStream` n'est pas un `OutputStream`), interdit les dépassements de pile, interdit les conversions illégales (entier en adresse)

niveau 3 : le chargeur de classes

Chaque classe est rangée dans un espace de noms propre à son lieu d'origine. Quand le système Java demande à accéder à d'autres classes, ce sont les classes locales (sécurisées) qui sont d'abord consultées.

La sécurité par le browser

niveau 4 : vis à vis du réseau

L'utilisateur de `hotjava` peut configurer 4 niveaux de sécurité réseau. Ces niveaux indiquent les droits d'un byte code à charger des éléments (autre applets, données, ...) au travers du réseau :

- No Access : le byte code doit pouvoir s'exécuter sans devoir chercher des éléments complémentaires, quelqu'en soit la source.

- Applet host : le byte code ne peut charger des éléments complémentaires qu'à partir du serveur d'où provient cette applet.

- Firewall : les byte code externes au firewall ne peuvent que charger des éléments provenant de serveurs externes au firewall

- Unrestricted : le byte code peut accéder à des donnees complémentaires quelque soit le serveur sur lequel elles sont présentes.

Java : langage ...

orienté objet i.e.
- classes
- encapsulation donc masquage de données
- objets
- héritage
- polymorphisme

d'abord compilé (vérifié avant sa mise sur le réseau) puis interprété.

indépendant des architectures : portable

supporte les notions "avancées" comme :
- les exceptions
- multithread
- programmation réseau
- connexion sur des URL

garbage collector (=> ce n'est plus au programmeur de gérer la désallocation mémoire)

bibliothèque de classes graphiques : classes d'objets conteneur, classe d'objets "contrôle" (boutons, labels, ...)

bibliothèque de classes audio et image.

Java : notions de base

Les structures de données fondamentales (tableaux, chaîne de caractères, ...) sont des objets avec leurs méthodes associées voire leur exception (indice non valide, etc.)

```
String chaine1 = "bonjour";
String chaine2;

chaine2 = chaine1 + " a tous" // surdéfinition de +



int a[ ] = new int [ 10 ];

a [5] = 1;
a [1] = a[0] + a [2];

a[-1] = 4; // lève une exception
           // ArrayIndexOutOfBoundsException
System.out.println (a.lenght);  // donnée associée aux tableaux
```

un peu de code

trame d'une applet

paquetage des classes à utiliser

classes du programme avec données et méthodes

exemple

```
import java.awt.*;
import java.applet.*;

public class MaClasse extends Applet {
      Image imCD;
      AudioClip son;

public void init() {
        ....
      imCD = getImage(....);
      son = getAudioClip(....);
}

public void paint(Graphics g) {
      imCD = getImage(....);
      g.drawImage(...);
        ....
}

public boolean mouseDown(Event evt, int x, int y) {
        ....
      return true; }
}
```

Ce programme est lancé dans une page html entre deux tags

```
<applet code="MaClasse.class">
</applet>
```


Première applet

Dans une page html :

```
<html>
<head><title>Ma page JMF</title></head>

<body>
<h1>Bienvenue sur cette nouvelle page</h1>
<P>

Execution du programme hello :

<applet code="hello.class" width=300
height=200>
</applet>

</body>
</html>
```

correspondant à l'applet :

```
// A compiler par :
// javac <ce_fichier>

import java.awt.Graphics;
import java.applet.Applet;

public class hello extends Applet {
      public void paint (Graphics g) {
            g.drawString("Bonjour a tous", 25,25);
      }
}
```


Le présent et le futur

le compilateur Java

Java 1.0 disponible sur Windows 95 et NT sur Intel et Solaris sur SPARC depuis le 23 Janvier 1996.

disponible sur MacOS 7.5 depuis mi Février 1996.

IBM a annoncé qu'il va porter Java sur Windows 3.1 et OS/2 et OSF sur d'autres plates-formes Unix.

les browsers

aujourd'hui `hotjava` et Netscape Navigator 3.0. Voir :
`http://home.netscape.com/eng/mozilla/2.0/relnotes/`

Des centaines d'applets déjà écrites.

demain PowerBrowser d'Oracle et Mosaic

historique et conclusion

au début était le chêne puis fut le café

`oak` (1990 par James Gosling) : langage pour puces informatiques (four micro-onde, T.V., CD-ROM, etc.)
`oak` nom trop répandu changé en `Java` (le café le meilleur vient de Java !!??)

avec `Java` les données comme les programmes n'ont pas à être résidentes en local: notion de machine virtuelle.
La machine et le logiciel qui la gère peut être quelconque (aie Bill G.)
Cette machine est très simple : un "bon" `browser`.

1ere semaine décembre 1995

Acte 1 : Sun annonce qu'IBM, Apple, DEC, Adobe, Silicon Graphics, HP, Oracle et Toshiba vont supporter Java.
Acte 2 : On s'attend à une riposte Microsoft avec son réseau et script visual Basic
Acte 3 : Microsoft arrête son réseau, choisit Internet et ... `Java`

L'avenir

Java =? le langage de programmation de l'Internet

La machine de demain =? un processeur + une mémoire + une zone de swap (et pas de disque =>? exit le PC)

indépendant machine mais il faut des serveurs de stockage d'applets.

The Planet is the Computer