Principes du langage Java

INTRODUCTION

Plus récent, Java a su rattraper ses concurrents directs que sont C/C++ et Basic/Pascal, tout en suivant ou anticipant les nouveautés associées à ces langages : scripts pour les pages internet en JSP - Java Server Page (équivalent à l'ASP de Microsoft et au PHP du monde Linux), composants réutilisables avec les JavaBeans (comparés aux ActiveX de Microsoft) et même, aujourd'hui, langage des futurs Web Services avec J2EE (opposant de la technologie .Net de Microsoft).

L'intention claire de *Sun*, éditeur d'origine, en étendant son langage, est de se positionner à tous les niveaux d'exécution : sur le client, sur le serveur, en interaction avec les bases de données, dans un système d'architecture distribué en réseau local ou autour du phénomène internet. Pour cela, les versions de Java sont nombreuses, aux noms aussi techniques qu'humoristiques (Java signifiant café en argot américain [issu de l'arabe Kawa]), chacune assurant une fonction précise du système d'information.

Java n'est plus un simple langage mais un environnement complet.

LE FONCTIONNEMENT DES DIFFERENTS SYSTEMES JAVA

Java se décline en 6 solutions principales :

Version	Principe	S'exécute sur
Applet	Programme (une classe compilée en .class, dont le code n'est pas lisible) inclus dans une page Web par la balise <applet ...>. Peut utiliser d'autres classes.	Client téléchargé avec la page Web. Sans interaction possible avec le serveur
Bean	Composant (une classe) réalisant une fonction spécifique (par exemple accès à une base de données) et appelé par une page JSP (balise <jsp:useBean >). Peut être inscrit dans une hiérarchie de classe.	Serveur d'application Une instance de la classe est créée par le serveur d'application lorsqu'il exécute le code JSP
Servlet	Programme capable d'agir en interaction avec des bases de données, des fichiers, de produire des résultats... Il peut être incus par une balise <servlet...> dans la page JSP.	Serveur d'application Il n'est pas téléchargé sur le poste client mais est interrogé par une page JSP par exemple. Lui-même est serveur Un servlet est l'équivalent d'un Winsock, il peut donc être autonome.
JSP	Langage de script couplé au serveur Web et qui permet la construction d'une page Web. Son code n'est jamais envoyé au client (seuls les résultats de son exécution le sont)	Serveur d'application Le serveur d'application créée une instance d'un objet générique à laquelle il fait exécuter le code. Le JSP peut accéder à des bases de données.

Version	Principe	S'exécute sur
JavaScript (extension de java développée par Netscape conformément aux directives Java de Sun)	Langage de script inclus entièrement dans une page Web (dont le texte est lisible)	Client Peut manipuler l'environnement de la machine cliente (navigateur, cookies, menus...).
Application Java	programme Java (ensemble de classes compilées en .class) indépendant du monde internet (n'a pas besoin de pages web pour s'exécuter)	Sur un poste ou en réseau Au même titre que n'importe quelle application, peut posséder des éléments côté client et côté serveur, peut fonctionner en monoposte ou en réseau (avec la technologie RMI : Remote Methode Invocation ou avec des outils servlets).

Le schéma ci-dessous localise ces éléments en montrant leurs interactions possibles.

LA MACHINE VIRTUELLE JAVA

On le voit, la description précédente ne fait aucune mention du type de système d'exploitation ou de logiciel de navigation utilisé. En effet, Java étant initialement prévu pour être embarqué dans des systèmes mobiles, il devait être impératif de ne pas avoir à réécrire le code ou recompiler les programmes en fonction de l'évolution des systèmes dédiés à ces matériels. C'est pourquoi le fonctionnement de java repose sur une étape intermédiaire qui isole le code du système sur lequel il s'exécute : la machine virtuelle java (JVM) qui simule un système d'exploitation. Il suffit alors de produire une machine par système.

On créer un fichier textuel d'extension .java, on le compile avec le compilateur fourni par Sun (javac.exe dans le JDK) pour produire un fichier .class semi-compilé, et c'est ensuite la machine java (java.exe sous Windows) qui exécute le code en le transformant au format du SE.

JSP ET TOMCAT

Pour faire exécuter des pages de script par un serveur Web, il est nécessaire de rajouter à ce dernier les extensions du serveur d'application correspondant (extension ASP, PHP...). Dans le cadre des pages JSP, les extensions sont réalisées par un serveur spécifique nommé Tomcat, qui intègre le serveur Web Apache et le servlet nécessaire au traitement des JSP. Il est possible de coupler Tomcat avec d'autres serveurs Web comme IIS (Internet Information Service de Microsoft). NON TESTE.

Le téléchargement de Tomcat (produit libre) est à faire sur http://jakarta.apache.org/tomcat (en anglais). On prendra les distributions binaires (exécutables) correspondant au système d'exploitation utilisé. Il faudra aussi disposer sur la machine d'une version du JDK (Java Developper Kit) de Sun pour faire fonctionner Tomcat.

L'installation est très simple, par contre, des petits problèmes de configuration peuvent intervenir, notamment un conflit entre le port sur lequel s'exécute le serveur Web de Tomcat (port 8080 par défaut) et un autre outil déjà présent sur la machine (par exemple, Oracle ou un autre serveur Web).

Il est alors nécessaire de modifier le contenu du fichier server.xml comme suit :

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8081 -->
    <Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
port="8100"               minProcessors="5" maxProcessors="75"
              enableLookups="true" redirectPort="8443"
              acceptCount="100" debug="0" connectionTimeout="20000"
              useURIValidationHack="false" disableUploadTimeout="true" />
```

On choisira une nouvelle valeur pour le port en s'assurant qu'elle n'est pas déjà utilisée. Lors du lancement de Tomcat, une fenêtre MSDOS s'ouvrira dans laquelle on retrouvera la valeur du port sur lequel s'exécute le service. Si une erreur doit se produire pour ce type de problème, la fenêtre MSDOS apparaîtra puis disparaîtra. Elle contient en fait les messages ci-contre, et notamment

```
Address in Use : JVM_BIND:n°Port
```

EXECUTION DES PAGES JSP

Le lancement de Tomcat se réalise par le menu Start Tomcat. Ensuite, dans le navigateur, il suffira de taper l'adresse IP de la machine abritant le serveur suivi du numéro de port sur lequel s'exécute Tomcat pour accéder à la page par défaut (index.jsp).

Pour l'installation sous 2000, Tomcat est ajouté en tant que service. Il faut donc le lancer à partir de l'ensemble des services (panneau de configuration / outils d'administration /services ou poste de travail / gérer / services). Les pages écrites (JSP et HTML) seront stockées dans le répertoire root du répertoire d'installation.

Page d'accueil par défaut Page JSP écrite

Lorsque la page fait appel à un javaBean, le fichier .class devra se trouver dans le répertoire .../ROOT/Web-inf/classes, et dans un répertoire portant le nom du package correspondant au fichier .java d'origine.

```
//extrait du fichier secureDBconnection.java
package etud;

import java.sql.*;
import java.io.*;

public class secureDBconnection {
```

Les .class des servlets seront stockés directement dans le répertoire .../ROOT/Web-inf/classes.

From:
https://wiki.sio.bts/ - **WIKI SIO : DEPUIS 2017**

Permanent link:
https://wiki.sio.bts/doku.php?id=jspprincipes

Last update: **2020/07/26 16:27**