

Le langage Java

1. Présentation du produit Java

Java est un langage moderne orienté objet développé par Sun.

L'idée de départ de Java (auparavant connu sous le nom de Oak) était de développer un langage avec lequel on puisse développer les logiciels pour l'électronique grand public, tels que décodeurs (set-top box), PDA, CD-ROM,.... Ses caractéristiques devaient donc être la robustesse, la compatibilité, la facilité de programmation et la petite taille du runtime ou des codes générés.

Java, tel qu'il est défini maintenant, est un langage simple, orienté objet, distribué, interprété, robuste, sécurisé, indépendant des architectures, portable, haute performance, multithreadé et dynamique, et qui va très au-delà des limitations des langages objets actuels qu'ils soient interprétés comme Lisp ou Smalltalk ou compilés comme C++.

1.1 Simple

La facilité d'accès à Java commence par le fait que la syntaxe est héritée de C++ et donc qu'un développeur C ou C++ pourra facilement maîtriser ce nouveau langage.

Mais C++ est un langage complexe : il intègre un certain nombre de fonctionnalités peu utilisées car trop difficiles à appréhender et maîtriser, ou des types tels que les struct ou union, qui complexifient le développement C++ sans être véritablement nécessaire. Les développeurs de Java ont décidé de simplifier considérablement C++ : certaines fonctionnalités, telles que l'héritage multiple ou l'overloading d'opérateurs, ou les types inutiles n'ont pas été retenus par l'équipe qui a conçu Java.

Certaines autres fonctionnalités ont été ajoutées comme le «ramasse-miettes» (garbage collecting) : ceci évite au développeur de se soucier de la gestion de la mémoire qui est souvent source d'erreurs de programmation.

Pour faciliter la programmation, Java intègre également une importante librairie de classes prédéfinies.

La simplicité se retrouve également au niveau de la taille de l'environnement runtime qui peut se satisfaire d'une mémoire de quelques dizaines de Ko.

1.2 Orienté-objet

L'avantage principal de la programmation orientée objet (POO) est la réutilisabilité de composants logiciels. Ceci permet de développer beaucoup plus rapidement, plus facilement et pour un coût moindre des applications. La POO est donc bien adaptée pour les développements internes des grands comptes en particulier dans le secteur tertiaire. Cela leur permet de réagir rapidement aux évolutions de leur métier. La POO permet de se focaliser sur les données (les objets) et leurs interfaces, et permet donc de définir très proprement des composants logiciels réutilisables. Java a hérité en cela des caractéristiques OO de C++ et d'Objective C (le langage de programmation en environnement NeXTstep).

1.3 Distribué

Java possède une librairie de routines permettant d'exécuter des processus à distance en s'appuyant sur les protocoles TCP/IP, tels que FTP (transfert de fichier) et HTTP (liens hypertextes du WWW).

1.4 Robuste

Pour aider le développeur à détecter les bugs le plus tôt possible, la vérification des erreurs (error checking) a lieu dès les premières étapes du cycle de développement d'un programme Java: dès la compilation ou l'édition de liens.

La principale différence entre Java et C++ est son modèle de pointeurs et l'impossibilité, avec Java, d'écrire sur des zones mémoire déjà allouées, ou d'altérer des données en mémoire. Java est donc tout à fait adapté au développement de prototypes, comme d'autres langages dynamiques tels que Lisp ou Smalltalk qui permettent de ne pas se soucier de la gestion de la mémoire.

1.5 Sécurisé

Java étant prévu pour être utilisé en réseau, la sécurité a été prise en compte dès le début de la définition de ce langage : La technique d'authentification s'appuie sur des mécanismes de chiffrement à clé publique. Il existe en fait une étroite relation entre robuste et secure : c'est grâce aux restrictions apportées au langage qui interdisent à des applications d'accéder et de modifier les structures de données en mémoire, que Java permet le développement de systèmes inaltérables totalement résistants aux virus.

1.6 Indépendant de l'architecture

Les réseaux étant hétérogènes, il fallait qu'une application Java puisse s'exécuter sur les systèmes les plus répandus en particulier Windows 95, Windows NT, MacOS et Solaris. Le code objet généré par le compilateur Java (bytecodes) est indépendant des architectures et peut donc être exécuté sur n'importe quel système sur lequel le runtime Java a été porté.

1.7 Portable

Au delà de l'indépendance matérielle, Java n'a aucune dépendance vis à vis des implémentations : les «float», par exemple, ne définissent que un nombre virgule flottante 32 bits IEEE 754, les int sont 32 bits,...

Toutes les librairies appartenant au système définissent des interfaces portables : il existe par exemple une classe fenêtre qui est implémentée pour Windows, Mac OS et Unix/Motif.

Le système Java lui-même est entièrement portable : le runtime est écrit en C ANSI conforme POSIX, et le compilateur est écrit en Java.

1.8 Interprété

l'interpréteur Java exécute directement les codes objets Java (bytecodes) sur tout système sur lequel est disponible le runtime Java (aujourd'hui Windows 95, Windows NT, MacOS, Solaris).

1.9 Haute performance

Bien que la performance de code interprété soit la plupart du temps suffisante, il arrive qu'il y ait besoin de performances supérieures : les «bytecodes» Java sont donc traduits à la volée (à l'exécution) en code machine. Pour cela le générateur de code machine fait partie de l'environnement runtime.

1.10 Multithreadé

Le multithreading est une technique qui permet, entre autre, aux applications d'être plus interactives. Java supporte le multithreading, et pour la simplicité de programmation définit un ensemble de primitives de synchronisation.

1.11 Dynamique

Java a été conçu pour s'adapter à un environnement qui évolue, et pousse le concept orienté objet à son optimum en permettant l'édition des liens entre modules objets dynamiquement au moment de l'exécution, en particulier dans le cas où une application s'appuie sur une librairie de classes (une librairie de composants logiciels). Même si cette librairie de classes évolue, il n'est pas nécessaire de modifier ou de recompiler l'application qui y fait appel. Ces nouvelles versions de modules seront prises en compte sans problèmes dès le moment où elles auront été installées.

2. Disponibilité

Comme de nombreuses fois dans le passé (avec NFS, CORBA 2 UNO, SPARC,...), Sun a publié les spécifications du langage Java, et des implémentations Solaris/SPARC et Windows NT/x86 sont disponibles sur Internet. Pour en savoir plus sur Java, il suffit donc de se connecter sur le serveur Web à l'adresse suivante <http://www.javasoft.com>.

Le premier usage qui a été fait de Java l'a été dans le cadre de la joint-venture Sun/Thomson Multimedia qui développe OpenTV une solution de télévision interactive. Les décodeurs (set-top box), permettant d'interagir avec le fournisseur de services, s'appuient sur des applications développées en Java.

On retrouve également Java comme support des «applets» - des petits programmes qui, téléchargés en local à l'aide du butineur (browser) dynamique HotJava, permettent d'animer le Web, et cela à la différence d'un butineur du type de Mosaïc qui ne permet que d'afficher des données statiques. Netscape a d'ailleurs acheté la licence de Java pour pouvoir l'intégrer à son produit vedette, le butineur WWW Netscape Navigator.

Plusieurs sociétés sont en train de développer des générateurs d'applications Java, d'autres développent des classes d'objets Java : tout un marché est en train de se mettre en place autour de ce nouveau langage.

SunSoft est également en train de réaliser un mapping IDL/Java permettant de développer des clients CORBA à l'aide du langage Java. On peut s'attendre à ce que mapping soit proposé à l'OMG pour standardisation.

Java n'a pas de concurrent aujourd'hui, seul Kaleida Labs qui développe ScriptX essaye de répondre aux mêmes besoins.

Le langage Java vient donc s'ajouter à la panoplie des langages mis à la disposition des développeurs, mais c'est sans doute l'un des plus intéressante récemment développé. Java rend véritablement la programmation plus facile, entre autre, grâce à ses fonctionnalités orienté-objet ou au «ramasse-miettes». De plus les codes Java étant indépendants de l'architecture cible, les applications Java sont particulièrement bien adaptées à des environnement réseaux hétérogènes tels qu'Internet ou CORBA.