



A	
ACID	Acronyme qui symbolise les quatre caractéristiques fondamentales de toute "transaction" : Atomicité (<i>Atomicity</i>), Cohérence (<i>Consistency</i>), Isolation et Durabilité (<i>Durability</i>)
ACL (java.security.acl)	Access Control List Structure de données à entrées multiples utilisée pour gérer les accès aux ressources (fichiers, ...). Une ACL contient des "entrées" appelées ACE (Access Control Entry). Chaque "entrée" de la liste contient un ensemble de permissions (droits d'accès).
AOP	Aspect-Oriented Programming (Programmation Orientée Aspect) Nouvelle approche méthodologique de programmation qui vise à combler certaines insuffisances de la "programmation orientée objet". Il est d'ores et déjà possible de programmer en "orienté aspect" avec des environnements comme "AspectJ" ou "AspectC++"
API	Application Programming Interface Interface de programmation fournissant un ensemble de fonctions ou de classes normalisées et documentées, permettant à des applications d'utiliser les fonctionnalités offertes par un produit, un composant ou une librairie.
applet (java.applet)	Parfois traduit en français par " appliquette ". Une "applet Java" est un composant logiciel écrit en Java (ensemble de classes), incrusté dans une page web (page HTML) et qui s'exécute dans le navigateur. Au sens J2EE, le navigateur et son "plug-in Java" sont vus comme un "applet container". Une applet est un composant mobile, téléchargé sur le poste client au moment ou celui-ci en a besoin. L'exécution est sécurisée, une applet ne peut pas accéder aux ressources locales du poste (sauf si elle est certifiée : notion d' <i>applet signée</i> , réputée fiable) (http://java.sun.com/applets/)
application server	Voir " serveur d'application "
AspectJ	Extension du langage Java qui supporte les principes de la programmation "orientée aspect" (AOP). (http://www.eclipse.org/aspectj/)
AWT (java.awt)	Abstract Window Toolkit AWT est une bibliothèque de classes graphiques standards de base (packages "java.awt" et "java.awt.*"). Les composants AWT reposent sur une encapsulation des composants graphiques natifs du système d'exploitation sur lequel s'exécute l'application Java. Ils sont donc nécessairement peu nombreux et peu évolués (pour des raisons de portabilité et de disponibilité sur différents systèmes). L'alternative à AWT est la bibliothèque « SWING » (composants graphiques totalement écrits en Java).
B	
BDK	JavaBeans™ Development Kit Kit de développement fourni par Sun pour réaliser et tester des composants "JavaBeans"
bean	Un "bean" est un composant logiciel Java réutilisable. Il existe deux grands types de composants normalisés : . les JavaBeans : destinés principalement aux applications Java "côté client" (IHM, ...) . les Enterprise Java Bean (EJB) : destinés au "côté serveur" d'une application n-tiers
BeanBox	La "BeanBox" est une application disponible dans BDK, qui permet de tester les "beans" réalisés. La BeanBox joue le rôle d'environnement de développement (IDE) de référence intégrant des Beans spécifiques.
Bean Builder	Nouvel outil de développement de JavaBeans (mis à disposition par Sun en Janvier 2002, nécessite un JDK 1.4) qui étend les possibilités de la BeanBox standard et prend en compte les nouvelles fonctionnalités offertes par les JavaBeans du JDK 1.4.
Bean-Managed Transactions (BMT)	Cas de figure où la gestion des transactions d'un EJB est gérée manuellement, par opposition au CMT (Container-Managed Transactions).
BluePrints	Littéralement "croquis" ou "plans". Les "BluePrints" (mis à disposition sur le site "java.sun.com") sont destinés à aider les développeurs Java en leur fournissant tous types d'informations utiles : des conseils, des modèles de conception, des exemples de codes, des articles de presse, des questions réponses, etc ... (http://java.sun.com/blueprints/)
BMP	Bean-Managed Persistence Un des deux modes de gestion de la persistance des EJB (BMP et CMP). En mode BMP, la persistance de l'EJB est gérée manuellement (codée par le développeur, et non prise en charge par le conteneur d'EJB).



browser	Navigateur Web, parfois aussi traduit par "butineur", "explorateur" (terminologie Microsoft) ou "fureteur". "Browser" est issu du verbe "to browse" qui signifie "feuilleter", par analogie sur Internet on consulte des pages HTML comme si on feuilletait un livre.
BTP	Business Transactions Protocol Protocole basé sur XML pour gérer des transactions complexes dans le cadre du commerce électronique (protocole normalisé par le consortium OASIS)
bytecode	Code Java après compilation. Il s'agit du code qui est chargé dans la JVM pour y être exécuté. Le bytecode est indépendant de la machine sur laquelle il sera exécuté, il ne s'agit pas de "code natif".
C	
CCI (javax.resource.cci)	Common Client Interface Interface standard d'accès aux progiciels extérieurs à la plate-forme Java, dans le cadre d'une architecture JCA (CCI est un des éléments de JCA).
CCM	Corba Component Model Modèle de composants multi-langage proposé par l'OMG dans la version 3 de CORBA (alternative aux EJB de la plate-forme J2EE)
CGI	Common Gateway Interface Interface standard qui permet à un serveur Web d'exécuter des programmes externes pour générer des pages HTML dynamiquement. En CGI, le serveur Web transmet la requête HTTP et ses paramètres à un programme qui peut être écrit en différents langages (C, C++, Perl, etc ...). Les technologies servlets et JSP de la plate-forme Java sont une alternative à CGI, elles offrent de meilleures performances ainsi que toutes les possibilités d'inter-opérabilité inhérentes à Java.
CLASSPATH	Variable d'environnement qui indique le chemin d'accès aux répertoires contenant les classes Java nécessaires à l'exécution d'une application.
Client léger	Poste de travail banalisé sur lequel les applications ne sont pas pré-installées et qui se limite à la gestion de l'IHM avec très peu de traitements réalisés "côté client" (l'essentiel des traitements étant réalisés "côté serveur"). Le principal avantage du "client léger" est de s'affranchir de la problématique du déploiement des applications. Les applications de type "I*net", dont l'IHM repose uniquement sur un navigateur web, sont l'exemple typique de l'utilisation du "client léger".
CMP	Container-Managed Persistence Un des deux modes de gestion de la persistance des EJB (BMP et CMP). En mode CMP, c'est le conteneur d'EJB qui prend en charge la gestion de la persistance de l'EJB.
CMR	Container-Managed Relationships Gestion automatisée (à la charge du "container") des relations entre les EJB (intégrité référentielle, etc). Il s'agit d'une possibilité récente offerte par les spécifications EJB 2.0.
CMT	Container-Managed Transactions Un des modes de gestion des transactions pour les EJB. En mode "container-managed transactions", c'est le conteneur qui prend en charge la gestion des transactions (et non le Bean lui-même)
conteneur ("container")	Le conteneur est l'environnement d'exécution contenant les différents types de composants de la plate-forme Java. L'architecture de la plate-forme J2EE 1.3 définit les conteneurs standards suivants : <ul style="list-style-type: none"> . Applet Container (pour les applets) . Application Client Container (pour les applications Java autonomes) . Web Container (pour les servlet et JSP) . EJB Container (pour les EJB)
conteneur d'EJB	Un des quatre conteneurs standards définis dans l'architecture J2EE. Ce conteneur assure la gestion des EJB (cycle de vie, persistance, ...)
cookie	Un "cookie" est un objet porteur d'informations, créé à l'initiative du serveur Web mais localisé "côté client" dans le navigateur. Sa durée de vie est variable, elle peut être de plusieurs jours ou plusieurs semaines, dans ce cas le cookie est stocké sur le poste client (par le navigateur) sous forme de fichier texte.
COS Naming	CORBA Object Service Naming Service de nommage pour les objets CORBA, accessible en Java via l'API JNDI
CORBA	Common Object Request Broker Architecture Spécification définie par l'OMG qui normalise les ORB afin de garantir l'inter-opérabilité des applications conformes à CORBA, indépendamment du langage et du système d'exploitation utilisés. CORBA s'appuie sur IDL et IIOP.




CSP	Cryptographic Service Provider Implémentation d'un service de cryptographie (dans le cadre de JCE/JCA)
CTS	Compatibility Test Suite Suite de tests de compatibilité pour valider la compatibilité d'un serveur d'applications J2EE avec les spécifications de la plate-forme
D	
DAO	Data Access Object "Design pattern" largement utilisé en environnement J2EE, qui définit des objets spécialisés pour gérer l'accès aux données (persistance des objets métiers, etc...)
DHTML	Dynamic HTML Combinaison de HTML + feuilles de style CSS + JavaScript qui permet de rendre les pages web "réactives" à certains événements et leur donne la capacité de modifier leur apparence dynamiquement. Le DHTML s'exécute exclusivement "côté client", dans le navigateur.
DNS	Domain Naming System Service de nommage Internet qui associe un nom de domaine à une adresse TCP/IP. Accessible en Java via l'API JNDI
DOM	Document Object Model Représentation d'un flux XML sous la forme d'un arbre d'objets avec des interfaces standardisées pour y accéder (specification W3C). Le DOM est également l'interface utilisée en JavaScript pour accéder aux objets présents dans un navigateur Web.
DTD	Document Type Definition Description du contenu d'un fichier XML sous la forme d'une grammaire normalisée
DSML	Directory Services Markup Language "Markup language" destiné à représenter des données sur les arborescences de répertoires, service accessible en Java via l'API JNDI .
E	
EAR (fichier ".ear")	Fichier de type "JAR" contenant une application J2EE.
ECMAScript	Version standardisée de JavaScript, limitée au "noyau" du langage. C'est l'ECMA (organisme Suisse) qui a normalisé ce langage sous la référence "ECMA-262". (http://www.ecma.ch/)
ECPerf	Benchmark de référence permettant de comparer les performances de différents serveurs d'applications J2EE.
EJB (javax.ejb)	Enterprise JavaBeans™ Composant Java standard fonctionnant "côté serveur" (sans interface de présentation), partageable, réutilisable et offrant des services et des données de "niveau entreprise" (logique métier, ...). Les EJB fonctionnent dans un "conteneur d'EJB" qui régit leur cycle de vie, leur comportement et éventuellement leur persistance. Ils offrent des fonctionnalités évoluées : système de sécurité (contrôle d'accès jusqu'au niveau des méthodes), mécanisme transactionnel, persistance, etc... Il existe deux types classiques d'EJB : . "Entity Bean" (EJB entité) . "Session Bean" (EJB session) et depuis la version 1.3 de J2EE : "Message Driven Bean" (EJB 2.0) (http://java.sun.com/products/ejb/)
EJB entité ("entity bean")	EJB correspondant à un objet métier persistant (une entité du système d'information, par exemple un client, une commande, etc...) Il existe deux type d'EJB entité : . BMP (bean managed persistence) : le bean gère lui même sa persistance . CMP (container managed persistence) : la persistance est prise en charge par le conteneur d'EJB (de façon automatique et transparente)
EJB session ("session bean")	EJB représentent un client vis à vis du serveur d'EJB Un EJB session peut être de deux types : . "stateless" ("stateless session bean") : bean session sans état (il ne maintient aucune information entre deux appels, il est réutilisable d'un appel à un autre) . "stateful" ("stateful session bean") : bean session qui maintient un état conversationnel (il est lié à un seul client, toujours le même)
EJB-QL	EJB Query Language Langage de requêtes proche de SQL utilisé pour les "finder methods" (méthodes de recherche) des EJB. EJB-QL fait partie des spécifications EJB 2.0



EMB (javax.emb)	Enterprise Media Beans (JSR-86) Framework pour intégrer des données multi-média (audio, vidéo ou image) dans des applications J2EE en utilisant les "EJB Entité".
Enterprise JavaBeans™	voir " EJB "
Entity Bean	voir " EJB entité "
F	
FTP	File Transfer Protocol Protocole standard de transfert de fichier basé sur TCP/IP.
G	
Garbage collector	Littéralement " <i>ramasse-miettes</i> ", le « garbage collector » est un système automatique de nettoyage de la mémoire de la machine virtuelle Java (JVM). Le "garbage collector" récupère l'espace mémoire en supprimant les objets qui ne sont plus utilisés. Ce système permet de décharger le développeur des tâches fastidieuses de gestion de la mémoire.
H	
HotJava™ Browser	Navigateur web écrit en Java, fourni par Sun
Hot Spot	Nouvelle génération de machine virtuelle Java (JVM) optimisée, qui décuple les performances. Notamment, le byte-code Java est compilé en code machine natif utilisable directement par le processeur, le "garbage collector" est optimisé, etc...
HDML	Handheld Device Markup Language Version allégée de HTML destinée aux équipements mobiles ayant des écrans de taille réduite (téléphones cellulaires, PDA, pagers, etc...)
HTML	HyperText Markup Language Langage de présentation dérivé de SGML basé sur un jeu de balises (ou "tags") qui définissent la présentation du document. Les pages affichées dans les navigateurs Web sont des fichiers HTML, mis en forme par le navigateur (pages HTML).
HTTP	HyperText Transfer Protocol Protocole standard de transfert de fichiers hypertexte (pages HTML) basé sur TCP/IP. Les navigateurs Web utilisent HTTP pour récupérer les pages HTML des serveurs Web.
HTTPS	HyperText Transfer Protocol Secure HTTP sécurisé basé sur SSL
I	
I*net	Application ou architecture reposant sur les protocoles standards d'Internet. I*net désigne de façon générique les environnements Internet , Intranet ou Extranet .
IDL	Interface Definition Language Langage de définition des objets CORBA . IDL est utilisé pour générer les "skeletons" et les "stubs" dans le langage de programmation choisi (Java, C++, Smaltak, ...) à l'aide d'un compilateur IDL.
IIOP	Internet Inter-ORB Protocol Protocole standard (basé sur TCP/IP), normalisé par l'OMG dans le cadre de la norme CORBA , qui assure la communication entre les différents ORB du marché. IIOP est un des protocoles utilisable pour les échanges de type RMI. Sun préconise maintenant l'utilisation d'IIOP plutôt que JRMP.
IMAP (ou IMAP4)	Internet Message Access Protocol Protocole standard de récupération des messages électroniques reçus (plus élaboré que POP) (http://www.imap.org/)
InfoBus (javax.infobus)	InfoBus est un package qui permet des échanges dynamiques de données entre des composants JavaBeans en définissant un nombre réduit d'interfaces et en spécifiant le protocole à utiliser. Tous les composants qui implémentent cette interface peuvent se "brancher" sur le bus et échanger des données avec d'autres composants.
Invocation API	API qui permet à des applications natives (généralement écrites en C ou C++) de charger une JVM et d'exécuter du code Java. Cette API évite une édition de liens avec la JVM. Invocation API est la réciproque de l'utilisation habituelle de JNI qui consiste à appeler du code natif à partir de Java (elle fait d'ailleurs partie des spécifications JNI).
introspection	Capacité de Java permettant de découvrir dynamiquement les attributs et les méthodes d'une classe pendant l'exécution.



<p>ISOC</p> 	<p>Internet SOCIety Organisme international non-commercial qui gère les standards liés à Internet. L'ISOC fédère les groupes de standardisation suivants : Internet Engineering Task Force (IETF), Internet Architecture Board (IAB), Internet Engineering Steering Group (IESG), Internet Research Task Force (IRTF). (http://www.isoc.org/)</p>
<p>J</p>	
<p>J2EE™</p>	<p>Java™ 2, Enterprise Edition Version "Enterprise" de la plate-forme "Java 2" qui se compose de l'environnement "J2SE" plus de nombreuses API et composants destinés à une utilisation "côté serveur" au sein du système d'information de l'entreprise. J2EE dispose notamment des servlets, JSP, EJB, JCA, JMS, JTA, etc... (http://java.sun.com/j2ee/)</p>
<p>J2ME™</p>	<p>Java™ 2, Micro Edition Version hautement optimisée de "Java 2" destinée à une large gamme de produits du type "pagers", téléphones cellulaires, systèmes de navigation, etc... (http://java.sun.com/j2me/)</p>
<p>J2SE™</p>	<p>Java™ 2, Standard Edition Version "Standard" de la plate-forme "Java 2" qui dispose de nombreuses fonctionnalités de base : JDK, JFC-Swing, applets, AWT, JDBC, RMI, JNDI, ... (http://java.sun.com/j2se/)</p>
<p>JAAS (javax.security.auth)</p>	<p>Java™ Authentication and Authorization Service Package qui fournit des services pour authentifier et renforcer les contrôles d'accès sur les utilisateurs. JAAS est intégré dans le JDK 1.4 (http://java.sun.com/products/jaas/)</p>
<p>JAE</p>	<p>Java™ Application Environment Version source du JRE et du JDK</p>
<p>JAF (javax.activation)</p>	<p>JavaBeans™ Activation Framework (extension standard) Framework ("100% pure Java") qui offre des services permettant de travailler avec des types de données arbitraires : . détermination automatique du type des données, . encapsulation de l'accès à ces données, . découverte des opérations possibles pour ce type de données, . instanciation des beans appropriés (correspondants aux opérations à réaliser) Exemple d'utilisation de JAF : un afficheur de fichier, différent selon le type du fichier Sun fournit une "implémentation de référence" (RI) de JAF (http://java.sun.com/products/javabeans/glasgow/jaf.html)</p>
<p>JAI</p>	<p>Java™ Advanced Imaging API API dédiée au traitement d'images numériques, utilisable en complément de "Java 2D". JAI dispose des principaux algorithmes standards de traitement d'image, de capacités à traiter de nombreux formats d'images, des encodeurs/décodeurs (codec), etc... (http://java.sun.com/products/java-media/jai/)</p>
<p>JAR (fichier "jar")</p>	<p>Java™ ARchive Format de fichier standard (généralement compressé) qui contient les composants logiciels (classes Java) et les ressources nécessaires pour exécuter une application. Les fichiers JAR permettent de simplifier le déploiement des applications Java.</p>
<p>Java 2D™</p>	<p>Ensemble de classes pour la création et la gestion d'images en deux dimensions. Cette API offre un nombre important de possibilités graphiques : transparence, transformation, "text layout", etc... (http://java.sun.com/products/java-media/2D/)</p>
<p>Java 3D™</p>	<p>Ensemble de classes pour la gestion d'images en trois dimensions (http://java.sun.com/products/java-media/3D/)</p>
<p>JavaBeans™ (java.beans)</p>	<p>Composant logiciel Java standard et réutilisable. Les JavaBeans s'auto-décrivent en publiant leurs propriétés et les événements auxquels ils réagissent, ce qui permet de les intégrer facilement dans différents outils de développement. Cette faculté repose sur les capacités d'introspection des objets Java. (http://java.sun.com/products/javabeans/)</p>
<p>Javadoc</p>	<p>Outil livré avec le JDK qui permet de générer automatiquement la documentation des classes Java d'un projet à partir des commentaires contenus dans le code source. La documentation est générée au format HTML et respecte la présentation standard utilisée par Sun pour les classes de base du langage</p>



JavaHelp™ (javax.help)	Système d'aide en ligne, basé sur des documents XML et HTML, qui permet aux développeurs d'inclure de l'aide en ligne dans des applets, des composants, des applications Java, etc... JavaHelp est intégralement écrit en Java, il offre donc un système d'aide indépendant de l'environnement d'exécution. (http://java.sun.com/products/javahelp/)
Java™ IDL (org.omg.*)	Technologie pour les objets distribués, similaire à RMI, mais qui permet de faire interagir des objets indépendamment du langage dans lequel ils ont été codés. Java IDL est basée sur CORBA (dont IDL est le langage de définition des interfaces) et supporte le mapping "IDL-to-Java". Java IDL fournit un compilateur "IDL-to-java" ("idlj") et un ORB (sous la forme d'une bibliothèque de classes) qui prend en charge la communication de bas niveau entre les applications "Java IDL" et les autres applications compatibles CORBA.
JavaMail™ (javax.mail)	API offrant un framework pour l'accès aux services de messagerie électronique (lecture, composition et envoi de messages), indépendamment de la plate-forme et du protocole. JavaMail utilise les protocoles standards de messagerie électronique : SMTP, POP, IMAP (http://java.sun.com/products/javamail/)
Java™ Media API	Ensemble des API Java relatives au multi-média (Image, Son, ...) : Java2D, Java3D, JAI, JMF, Java Sound API, Java Speech API, etc... (http://java.sun.com/products/java-media/)
JavaOne™	<i>Sun's Worldwide Java Developer's Conference</i> Conférence mondiale des développeurs Java organisée par Sun une ou deux fois par an
Java Pet Store Demo	Application J2EE de démonstration (fournie avec les sources sur le site de SUN) pour apprendre Java et J2EE par l'exemple.
Java™ Plug-in	Environnement d'exécution Java (JRE) fourni sous forme de "plug-in" afin de s'interfacer avec un outil existant ayant besoin d'exécuter du code Java. Le "Java Plug-in" est notamment utilisé par différents navigateurs Web pour exécuter les applets Java de façon identique (indépendamment de la JVM propre à chaque navigateur). (http://java.sun.com/products/plugin/)
Java™ Portlet API	Future API qui sera définie prochainement par le JCP (JSR-168) (voir aussi WSRP)
JavaScript	Langage de script interprété, développé à l'origine par Netscape, puis adopté par Microsoft sous le nom de "JScript". "ECMAScript" est la version standardisée de JavaScript (norme limitée à la syntaxe et aux objets de base du langage).
Java Web Start™	Technologie qui permet de télécharger et d'exécuter une application Java autonome. Java Web Start permet un déploiement automatique des applications Java (à la demande et en fonction des besoins de chaque poste). Les mises à jour des applications sont également automatisées, Java Web Start vérifie les versions et, le cas échéant, télécharge et installe les composants qui ont été modifiés. (http://java.sun.com/products/javawebstart/)
Java™ XML Pack	Pack logiciel qui regroupe toutes les API Java concernant XML : <ul style="list-style-type: none"> . Java API for XML Messaging (JAXM) . Java API for XML Processing (JAXP) . Java API for XML Registries (JAXR) . Java API for XML-based RPC (JAX-RPC) Les prochaines versions contiendront JAXB
JAX-RPC (javax.xml.rpc)	Java™ API for XML based RPC API pour les appels de services distants (RPC) : services Web via SOAP et WSDL (http://java.sun.com/xml/jaxrpc/)
JAXB	Java™ Architecture for XML Binding (JSR-31) Architecture fournissant un ensemble d'API et d'outils pour automatiser le mapping entre les documents XML et les objets Java. (http://java.sun.com/xml/jaxb/)
JAXM (javax.xml.messaging, javax.xml.soap)	Java™ API for XML Messaging (JSR-67) API pour l'échange de messages basés sur XML (messages de type SOAP) (http://java.sun.com/xml/jaxm/)
JAXP (javax.xml.parsers javax.xml.transform)	Java™ API for XML Processing JAXP est une API qui normalise l'utilisation des services de traitement des flux XML : "parsers XML" (SAX ou DOM) et transformation XML (XSLT). JAXP permet d'analyser ("parser") et de transformer des documents XML en toute indépendance de l'implémentation des outils utilisés (parsers XML ou transformateurs XSL). JAXP supporte les "XML schemas". Des implémentations de références sont disponibles (JAXP-RI : JAXP Reference Implementation) avec les JDK 1.3 et 1.4. JAXP-RI repose sur des parsers et transformateurs standards de la communauté "Apache" (Crimson, Xalan et prochainement Xerces 2) (http://java.sun.com/xml/jaxp/)



JAXR (javax.xml.registry)	Java™ API for XML Registries (JSR-93) API pour l'accès aux annuaires UDDI (http://java.sun.com/xml/jaxr/)
JAXTX	XML Transactioning API for Java (JSR-156 & JSR-95) Future API, en cours de spécification (JSR) pour packager et transporter des transactions de type "ACID" (comme JTA) et des transactions étendues ("non ACID") en utilisant XML et des protocoles de type SOAP ou BTP.
JCA (javax.security)	Java Cryptography Architecture Architecture normalisée conçue pour développer et utiliser des fonctionnalités de cryptographie. Cette architecture assure une indépendance vis à vis des implémentations et des algorithmes utilisés en utilisant des "providers" (CSP)
J2EE CA (ou JCA) (javax.resource.*)	J2EE Connector Architecture Les spécifications J2EE CA standardisent l'accès aux différentes applications hétérogènes du Système d'Information (ERP, CRM, SCM, applications spécifiques, TP mainframe, SGBD, ...) (http://java.sun.com/j2ee/connector/)
JCE (javax.crypto)	Java™ Cryptography Extension Ensemble de packages fournissant des fonctionnalités de cryptage, de génération de clés, d'agrément de clés et des algorithmes MAC (Message Authentication Code). JCE fait partie intégrante du JDK 1.4. (http://java.sun.com/products/jce/)
JCP	Java™ Community Process Processus standard d'évolution des spécifications Java. Ensemble de procédures de gestion des évolutions qui décrit des JSR (Java Specification Request) qui aboutissent éventuellement à de nouvelles spécifications si elles sont approuvées (cf. site de référence : www.jcp.org)
JDBC™ (java.sql, javax.sql)	Java™ Database Connectivity API normalisée pour les accès aux bases de données relationnelles (http://java.sun.com/products/jdbc/)
JDC	Java™ Development Connection Site de Sun destiné aux développeurs Java et offrant de nombreuses ressources et informations : newsletter, forum, tutoriaux, trucs et astuces, etc... (cf. site de référence : http://developer.java.sun.com/)
JDK	Java™ Development Kit Le kit de développement Java contenant les outils de développement (compilateur, machine virtuelle Java, débogueur, ...) et l'ensemble des classes de base Java. Remarque : ce kit de développement est libre d'utilisation, il suffit de le télécharger sur le site de Sun.
JDO (javax.jdo)	Java™ Data Objects (JSR-12) JDO offre des fonctionnalités de persistance automatisée et transparente pour les objets Java et assure le "mapping objet/relationnel". Avec JDO, tout objet est potentiellement persistant (les capacités de persistance sont ajoutées par enrichissement du "byte-code"). JDO a été adopté par le JCP et est devenu une spécification Java (API normalisée) en Mars 2002. (http://java.sun.com/products/jdo/)
JDOQL	JDO Query Language Langage de requête de JDO permettant de récupérer des "collections d'objets" correspondant à un "filtre" (critère).
JFC	Java™ Foundation Classes Bibliothèque de classes graphiques pour réaliser des IHM standards indépendantes du système d'exploitations. Les JFC sont souvent assimilées uniquement aux " Swings ", mais en réalité, elles englobent de nombreux modules : les composants Swings, Java 2D, AWT, l'internationalisation, le "Drag and Drop", le "Pluggable Look and Feel", ... (http://java.sun.com/products/jfc/)
JIT	Just - In - Time (Compiler) "Compilateur juste à temps" Compilateur qui convertit le "byte code" en code natif juste au moment où il va être exécuté, afin d'améliorer les temps de traitement.
JMF	Java™ Media Framework API (package optionnel) Framework qui permet aux applications et applets Java de gérer des flux multimédia (audio, vidéo, ...). Avec JMF il est possible de capturer, lire et transformer différents formats multimédia (MPEG, ...) (http://java.sun.com/products/java-media/jmf/)






JMI	Java™ Metadata Interface (JSR-40) Spécification qui permet la création, le stockage, la découverte et l'échange de "méta données". JMI est basé sur la spécification "MOF" (Meta Object Facility) de l'OMG et échange les "méta-modèles" et les "méta-données" selon le standard XMI (XML Metadata Interchange) (http://java.sun.com/products/jmi/)
JMS (javax.jms)	Java™ Message Service API API normalisée qui permet d'accéder aux services d'un Middleware Orienté Message (MOM). JMS permet de travailler dans deux modes : . "Point-to-Point (PTP)" (mode point à point) : chaque message est adressé à une file spécifique de laquelle il sera extrait par le destinataire . "Publish/Subscribe (pub/sub)" (mode publication/abonnement) : chaque message "publié" peut être obtenu par plusieurs clients. (http://java.sun.com/products/jms/)
JMX	Java™ Management eXtensions Les spécifications JMX décrivent un standard destiné à l'administration et la supervision d'applications Java à travers le réseau. JMX définit une couche d'isolation entre les ressources à gérer (objets Java) et le système d'administration. Cette couche intermédiaire est basée sur un standard permettant de réaliser des « objets JMX » ou « MBeans » (managed beans) qui vivent dans un « conteneur JMX » et qui offrent une capacité d'administration centralisée à partir de différents outils. Un « client JMX » peut accéder aux attributs et invoquer les méthodes d'un « MBean » ; il peut également recevoir des notifications émises par un « MBean ». (http://java.sun.com/products/JavaManagement/)
JNI	Java™ Native Interface Interface permettant à du code Java d'utiliser du code natif (par exemple des DLL sous Windows). JNI est la réciproque de "Invocation API"
JNDI (javax.naming)	Java™ Naming and Directory Interface™ API définissant les accès aux services de nommage (ou services de désignation) et aux services d'annuaires tels que LDAP, NDS, DNS, NIS(YP), COS Naming (CORBA), RMI Registry, etc... (une seule et même API pour différents types d'annuaires). Ces services de nommage et d'annuaires offrent aux applications la possibilité d'accéder à une ressource à partir du nom qui lui est associé. Par ailleurs, dans le cadre d'une informatique distribuée, JNDI permet de définir la localisation des objets Java disponibles sur le réseau. (http://java.sun.com/products/jndi/)
JNLP (javax.jnlp)	Java™ Network Launch Protocol API définissant l'utilisation du protocole utilisé par Java Web Start™ pour lancer, déployer et mettre à jour les applications Java.
JPDA	Java™ Platform Debugger Architecture Architecture standard qui vise à faciliter la mise en œuvre d'outils de mise au point. JPDA repose sur deux interfaces (JVMDI et JDI), un protocole (JDWP) et deux composants logiciels qui lient la partie visuelle du "debugger" à la partie "arrière-plan" du programme. (http://java.sun.com/products/jpda/)
JRE	Java™ Runtime Environment L'environnement d'exécution Java contenant notamment la machine virtuelle (JVM) et l'ensemble des classes de base. Remarque : le JRE est libre d'utilisation, il suffit de le télécharger sur le site de Sun.
JRMP	Java™ Remote Method Protocol Protocole spécifique à la technologie Java qui permet d'accéder aux objets répartis de type RMI (on parle alors de RMI/JRMP par opposition à RMI/IIOP)
JSF	JavaServer Faces (JSR-127) Framework qui simplifie le développement des interfaces utilisateur pour les applications web (servlet+JSP). JSF repose sur un ensemble d'API pour la gestion des composants IHM et sur une librairie de "custom tags JSP" (taglib) (http://java.sun.com/j2ee/javaxserverfaces/)
JSP (javax.servlet.jsp)	JavaServer Pages™ Technologie Java permettant de générer dynamiquement des pages HTML (ou XML, WML, texte, etc...) à partir d'un serveur HTTP et d'un Web container (moteur de servlet/JSP). Une "page JSP" est un fichier avec une extension ".jsp" qui contient les parties statiques du document et du code Java qui sera exécuté sur le serveur pour générer les parties dynamiques. En fait, les pages JSP sont automatiquement transformées en servlets pour une exécution plus efficace. Dernières spécifications : JavaServer Pages 1.2 (JSR-53) (http://java.sun.com/products/jsp/)



JSR	Java Specification Request Description des demandes de nouvelles spécifications pour la plateforme Java™ Les JSR sont gérés dans le cadre du Java Community Process (JCP) Program
JSSE	Java™ Secure Socket Extension APIs ajoutant des capacités de sécurité réseau et permettant au développeur d'utiliser les fonctionnalités de SSL (Secure Sockets Layer) et TLS (Transport Layer Security).
JSTL	JSP Standard Tag Library Librairie standard de Tag JSP ("TagLib"). JSTL est une spécification (JSR-52) et une implémentation fournie par le projet Jakarta de la communauté Apache (ver 1.0 Beta 1 disponible depuis mars 2002)
JSWDK	JavaServer™ Web Development Kit Kit de développement pour réaliser et tester des servlets et des JSP NB : le JSWDK n'est plus à jour, il ne supporte que les servlets 2.1 et les JSP 1.0 (on utilise généralement Tomcat comme conteneur Web de référence pour tester les servlets et les JSP)
JTA (javax.transaction, javax.transaction.xa)	Java™ Transaction API API permettant d'utiliser les services d'un moniteur transactionnel. JTA décrit les différentes interfaces entre le moniteur transactionnel et les différentes parties impliquées dans la gestion des transactions distribuées : (http://java.sun.com/products/jta/)
JTS (javax.jts, org.omg.CORBA, org.omg.CosTransactions, org.omg.CosTSPortability)	Java™ Transaction Service JTS est une spécification qui décrit l'implémentation d'un moniteur transactionnel supportant l'API JTA et le mapping Java pour OTS (service de transactions objet de CORBA). Dans JTS les transactions sont propagées via IIOp. (http://java.sun.com/products/jts/)
JUG	Java™ User Group Groupe d'utilisateurs Java. Il existe de nombreux JUG dans le monde, leur vocation est de fournir aux membres du groupe les moyens d'échanger des informations et des retours d'expérience. Les principaux JUG sont référencés sur le site http://servlet.java.sun.com/jugs/ .
JVM	Java™ Virtual Machine Machine virtuelle Java qui permet à un ordinateur d'exécuter le "byte code" Java. Une "machine virtuelle" est un programme natif qui simule un environnement d'exécution réel. La JVM répond à des spécifications très précises définies par Sun, c'est elle qui garantit la portabilité de Java en prenant en charge l'exécution du "byte code". Le même "byte code" fonctionne de façon identique sous différents environnements, c'est la JVM qui change en fonction de l'environnement et non les applications Java. Une JVM peut être interfacée avec des programmes "natifs" grâce à l'API "JNI".
L	
LDAP	Lightweight Directory Access Protocol Protocole de gestion d'annuaires de réseau (version allégée de la norme X500), conçu à l'Université du Michigan , et reconnu comme un standard par tous les grands éditeurs de logiciels. Les annuaires LDAP sont accessibles en Java via l'API JNDI
LDIF	LDAP Data Interchange Format Format standard d'échange de données LDAP (format texte)
M	
MDB	Message Driven Bean "Bean piloté par message" Il s'agit d'un nouveau type d'EJB (apparu avec la norme EJB 2.0 de la version 1.3 de J2EE) qui prend en charge les messages asynchrones reçus (messages JMS) et réagit en conséquence. Un "message driven bean" peut être "abonné" à certains types de messages ("topic") ou consommateur d'une file de message ("queue").
Message Driven Bean	voir MDB
MIME	Multipurpose Internet Mail Extensions Le "type MIME" définit la nature des données transportées sur le réseau (par mail, par HTTP, etc...). Il ne s'agit pas d'un protocole mais d'une définition de contenu (format du message, attachements, etc...) Exemples de type MIME : "text/plain", "text/html"



MOM	Message-Oriented Middleware (acronyme identique en français : "Middleware Orienté Message") Middleware permettant à des applications hétérogènes d'échanger des informations via des files de messages ("message queuing"). Les MOM permettent des échanges asynchrones sécurisés et réduisent le nombre de relations à gérer entre les différentes applications. En environnement Java l'accès à ce type de middleware se fait via une API normalisée : " JMS ".
MVC	Model – View – Controller (Modèle – Vue – Contrôleur) "Design pattern" standard, largement utilisé dans les architectures logicielles basées sur des langages orientés objet, qui partage les différents composants en trois parties : . ce qui est présenté à l'utilisateur (IHM graphique, page HTML, etc ...) : la vue . la représentation interne du/des objet(s) : le modèle . la gestion des actions sur les différents objets : le contrôleur
N	
NDS	Novell Directory Services Service d'annuaire de Novell, accessible en Java via l'API JNDI
NIS / NIS+	Network Information Services Service d'annuaire fournissant des informations concernant le réseau, connu aussi sous le nom de "Yellow Pages" (YP), accessible en Java via l'API JNDI
NNTP	Network News Transfer Protocol Protocole standard pour la lecture des "news"
O	
OASIS 	Consortium non commercial qui dirige le développement, la convergence et l'adoption de standards dans le domaine du "e-business" en s'appuyant sur la technologie XML (ebXML, SAML, BTP, ...) (site de référence : http://www.oasis-open.org/)
ODMG 	Object Data Management Group Groupe de travail (issu de l'OMG) dont la vocation est de définir des standards pour le stockage des objets dans des bases de données. Par exemple, l'ODMG a soumis les principes de base des spécifications JDO au JCP. (site de référence : http://www.odmg.org/)
OMG 	Object Management Group Organisme indépendant ayant pour vocation la normalisation des technologies informatiques liées à l'objet. L'OMG gère notamment les normes suivantes : UML, CORBA, MDA (site de référence : http://www.omg.org/)
OOP	Object-Oriented Programming Programmation Orientée Objet
ORB	Object Request Broker Logiciel offrant des services de communications normalisées (CORBA, IIOP) permettant à des objets répartis d'inter-agir.
OTM	Object Transaction Monitor Moniteur transactionnel permettant de gérer des transactions dans un environnement d'objets distribués.
OTS	Object Transaction Service Interfaces (normalisées par l'OMG) permettant à des objets distribués (CORBA) de participer à des transactions. En environnement Java, les packages <i>org.omg.CosTransactions</i> and <i>org.omg.CosTSPortability</i> de JTS définissent le mapping Java pour les interfaces OTS 1.1 .
P	
package	Un package est un ensemble de classes. Les packages permettent d'organiser les composants logiciels en ensembles cohérents. Les classes Java standards sont regroupées en packages (exemples : java.io, java.net, java.sql, ...)
persistance	Capacité des objets à perdurer indépendamment de leur environnement d'exécution (la JVM dans le cas de Java). Il s'agit en fait de la possibilité pour un objet de se sauvegarder sur disque (sous forme de fichier ou dans une base de données) afin d'être reconstitué à l'identique ultérieurement.
PLAF (javax.swing.plaf)	Pluggable Look-And-Feel "Look and feel" modifiable (caractéristique des "Swings")
POP (ou POP3)	Post Office Protocol Protocole standard utilisé pour récupérer les messages électroniques reçus




proxy	Objet technique dont le rôle est de simuler localement un objet distant. Le proxy intercepte et gère les requêtes destinées à l'objet distant dont il est le "représentant local". Un "stub" CORBA ou RMI est un "proxy".
R	
RAR (fichier ".rar")	Resource Adapter Archive Fichier de type "JAR" (archive Java) qui contient les composants d'un " Resource Adapter " nécessaire pour se connecter à une ressource avec JCA .
Reflection	"Réflexion" ou "introspection" : principe qui permet de connaître dynamiquement les attributs et les méthodes de n'importe quel objet Java, au moment de l'exécution
Resource adapter	Sorte de "driver" permettant de se connecter à une ressource du système d'information (application, ERP, moniteur transactionnel,...) via l'interface CCI de l'architecture JCA .
RI	Reference Implementation (Implémentation de référence) Pour chaque spécification d'une technologie ou d'un service, Sun fournit une "implémentation de référence" qui peut être utilisée librement. Parmi les différentes implémentations de référence, on trouve notamment : un conteneur Web (pour les servlets et les JSP), un conteneur d'EJB, etc...
RMI (java.rmi, javax.rmi)	Remote Method Invocation (Littéralement : "appel de méthode à distance") Technologie standard offerte par Java qui permet d'utiliser un objet distant comme s'il était présent dans la JVM locale. Un serveur instancie un objet qui va être utilisable par des clients distants à travers le réseau. RMI peut être considéré comme un ORB à part entière, mais contrairement à CORBA il ne fonctionne qu'avec des objets Java. RMI repose sur un service de désignation des objets ("rmiregistry"), des "stub" et "skeleton", un protocole d'échange de données qui est soit JRMP, soit IIOp (les spécifications Java 2 version 1.3 imposent l'utilisation d'IIOp). (http://java.sun.com/products/jdk/rmi/)
RMI-IIOP	Remote Method Invocation over Internet Inter-ORB Protocol Version de RMI qui utilise le protocole standard IIOp de CORBA pour échanger des données entre objets distants. Ce qui permet à des objets définis en utilisant les interfaces de type RMI d'être accessibles de façon standard via le protocole IIOp. La technologie "RMI-IIOP" intègre "Java RMI" et "Java IDL".
RMI-JRMP	Version de RMI qui utilise le protocole JRMP pour la communication entre objets distants (protocole initial de RMI, spécifique à la technologie Java RMI)
RPC	Remote Procedure Call (Appel de procédure distante) Technologie qui permet à un composant logiciel de déclencher l'exécution d'une procédure localisée sur une machine distante.
S	
SAAJ (javax.xml.soap)	SOAP with Attachments API for Java API qui permet à une application Java d'envoyer ("produire") et de recevoir ("consommer") des messages conformes aux spécifications "SOAP" et "SOAP Attachment Feature" du W3C (http://java.sun.com/xml/saaj/)
sandbox	Littéralement "bac à sable", il s'agit de l'espace de confinement (espace sécurisé) dans lequel les applets sont exécutées sur le poste client. Le "sandbox" garantit la sécurité du poste.
SAX	Simple API for XML Principe d'analyse événementielle d'un flux XML (il existe des parsers SAX et des parsers DOM)
sérialisation (ou "serialization")	Mécanisme standard offert par Java permettant d'enregistrer le contenu d'un objet dans un flux (généralement un fichier) puis de le reconstituer ultérieurement. La sérialisation offre donc un moyen (primaire mais parfois suffisant) de gérer la persistance des objets Java.
servlet (javax.servlet)	Technologie Java destinée à réaliser des traitements "côté serveur", sans interface utilisateur, en réponse aux requêtes de postes généralement distants. Il existe différents types de servlets, les plus utilisées sont des servlets de type HTTP qui permettent de générer dynamiquement des pages HTML (ou XML, texte, WML, ...) en réponse aux requêtes HTTP reçues. Une servlet est un objet Java, dont le cycle de vie est géré par un "Web Container J2EE" (moteur de servlet). Dernières spécifications : Java Servlet 2.3 (JSR-53) (http://java.sun.com/products/servlet/)
serveur d'application ("application server")	Serveur logique qui fournit un ensemble de services applicatifs et qui assure les interactions entre les postes des utilisateurs et les différentes ressources du système d'information (SGBD, annuaires, progiciels, middlewares, ...). En environnement Java, les serveurs d'applications sont normalisés, ils doivent être conformes aux spécifications J2EE.



service web (ou "web service")	Service distant invocable par le Web en utilisant le protocole SOAP , les annuaires UDDI et le langage de description WSDL . Les services web (et les technologies associées) sont indépendants des normes Java. Java permet de créer et d'utiliser des "services web"
Session Bean	voir " EJB Session "
skeleton	Littéralement : "squelette" (ou "châssis") Composant logiciel ayant en charge la communication entre objets distants. Chaque "skeleton" (situé côté serveur) communique avec un "stub" (situé côté client). Les échanges RMI reposent sur des "stubs" et des "skeletons".
SMTP	Simple Mail Transfer Protocol Protocole utilisé pour envoyer des messages électroniques
SOAP	Simple Object Access Protocol Protocole standard d'invocation de services web basé sur XML et HTTP
SPI	Service Provider Interface (Interface implémentée par un "fournisseur de service") Java définit différentes architectures (JNDI, JCA, ...) qui reposent sur deux parties : API et SPI (exemple : JNDI API et JNDI SPI)
SQL	Structured Query Language Langage standardisé pour lire et mettre à jour les informations contenues dans une base de données relationnelle
SQL/J	Ensemble de spécifications permettant de standardiser l'utilisation du SQL "embarqué" dans du code Java.
SSL	Secure Socket Layer Couche de communication sécurisée (de niveau "socket") qui garantit la confidentialité des échanges pour les applications "I*net".
stub	Littéralement : "souche" ou "talon" Composant logiciel ayant en charge la communication entre objets distants. Chaque "stub" (situé côté client) communique avec un "skeleton" (situé côté serveur). Les échanges RMI reposent sur des "stubs" et des "skeletons".
SVG	Scalable Vector Graphics Langage standard de représentation d'images vectorielles au format XML. SVG est normalisé par le W3C.
"Swings" (javax.swing)	Bibliothèque de classes graphiques permettant de réaliser des IHM standards indépendantes du système d'exploitations. Les "Swings" font partie des "JFC"
T	
Taglib (JSP Taglib)	Librairie de tags spécifiques ("custom tags") utilisables dans les JSP . Une "taglib" se définit via un fichier XML avec une DTD normalisée (fichier ".tld") et un préfixe utilisé pour identifier tous les tags de cette librairie. Chaque tag est associé à deux classes Java (à développer spécifiquement) qui fournissent une description et une implémentation du tag spécifique. La possibilité de définir de nouveaux tags fait partie des spécifications JSP 1.1.
TCK	Test Compatibility Kit Kit de test, destiné aux éditeurs de logiciels, qui permet de vérifier la compatibilité et le respect des spécifications Java.
TLD (fichier ".tld")	Tag Library Descriptor Fichier, au format XML, qui décrit les tags d'une librairie de "custom tags" pour les JSP ("TagLib")
transaction	Opération fonctionnelle de mise à jour réalisée par un utilisateur ou un logiciel sur un système d'information. Une transaction doit respecter un certain nombre de qualités résumées par l'acronyme "ACID", qui signifie : Atomicité, Cohérence, Isolation et Durabilité. Une transaction est généralement composée de plus mises à jour unitaires qui doivent être réalisées en totalité ou pas du tout (notion de "commit" et "rollback").
translet	Classe Java résultant de la compilation d'une feuille de style XSL. L'exécution de ce code Java réalise une transformation XML identique à celle qu'aurait réalisée la feuille de style d'origine (voir "XSLT compiler" dans JAXP)
TrAX (javax.xml.transform)	Transformation API for XML API standard pour réaliser des transformations XSLT sur des flux XML (cette API fait partie de JAXP)
U	
UDDI	Universal Description Discovery and Integration Système d'annuaire normalisé utilisé pour le référencement et la recherche de "services web" (http://www.uddi.org/)



UML	Unified Modeling Language UML est normalisé par l'OMG depuis 1997.
Unicode	Jeux de caractères encodés sur 16 bits (soit plus de 65.000 caractères), au lieu de 8 en ASCII, afin de disposer des différents caractères internationaux. Unicode est une norme ISO (ISO-10646). Java traite tous les caractères en Unicode. (http://www.unicode.org/)
URI	Uniform Resource Identifier Une URI est un identifiant unique (représenté sous la forme d'une chaîne de caractères) qui permet de rechercher et d'utiliser une ressource sur internet (ou intranet). La syntaxe générique des URI est définie par la RFC-2396 Il existe deux type d'URI : les URL et les URN (cf : http://www.w3.org/Addressing/)
URL	Uniform Resource Locator Une URL commence par le protocole de communication utilisé ("http:", "ftp:", "gopher:", ...), suivi des caractéristiques nécessaires pour identifier la ressource. Les URL doivent respecter une syntaxe précise, définie par RFC-1738 et RFC-1808 (Relative URL) Exemples d'URL : http://www.w3.org file://ftp.toto.com/pub/files/foo.txt
URN	Uniform Resource Name Une URN est une URI qui désigne une ressource ayant un engagement de persistance et de disponibilité, indépendamment de son emplacement. Les URN doivent respecter une syntaxe précise, définie par la RFC-2141 (les URN commencent par "urn:")
W	
W3C	 World Wide Web Consortium Organisme de normalisation indépendant. Principales normes gérées par le W3C : HTML, XML, XHTML, DOM, CSS, SOAP, SVG, SMIL, etc... (cf. site de référence : http://www.w3.org/)
WAP	Wireless Application Protocol Protocole permettant à des téléphones mobiles d'accéder à des applications de type "I*net" via des réseaux sans fil.
WAR (fichier ".war")	Fichier de type "JAR" contenant une "application Web" ("webapp" : application basée sur les servlets et les JSP).
Web application (ou "webapp")	Application de type "web" destinée à fonctionner en environnement "I*net / client léger" (internet/extranet/intranet). Ce type d'application est basé sur l'utilisation des servlets et des JSP.
WML	Wireless Mark-up Language Langage de présentation à base de tags ("markup language") proche de HTML, destiné à l'affichage d'informations issues du Web sur de petits écrans, en particulier les téléphones mobiles via le WAP. WML est aux téléphones WAP ce que HTML est aux navigateurs.
"WORA"	"Write Once, Run Anywhere" ("écrire une fois, exécuter n'importe où") Slogan de Sun qui illustre la portabilité quasi-universelle du langage Java
WSDL	Web Services Description Language Langage de description des "services web", basé sur XML
WSDP	Java™ Web Services Developer Pack Kit de développement pour les services web comprenant entre autres : JAXP, JAX-RPC, JAXM, JAXR, Tomcat, ...
WSRP	Web Services for Remote Portals Standard destiné à garantir l'interopérabilité entre les "portlets" et les portails. Les WSRP sont des services web qui s'intègrent dans des portails ("portlets" publiés en tant que services web de type WSRP). Voir aussi : Java Portlet API.
X	
XML	eXtensible Markup Language XML est normalisé par le W3C La plate-forme Java offre de nombreux outils et packages pour gérer des documents XML et accéder aux services Web (JAXP, JAXM, JAXR, ...) (http://java.sun.com/xml/)
XSL	eXtensible Stylesheet Language Langage de définition des feuilles de styles applicables aux documents XML XSL est normalisé par le W3C
XSLT	eXtensible Stylesheet Language Transformation Langage de transformation d'un flux XML. XSLT fait partie de XSL et est normalisé par le W3C.