

<http://membres-liglab.imag.fr/donsez/cours>

# Programmation et Architecture Java 2 Micro Edition

---

**Didier Donsez**

*Université Joseph Fourier (Grenoble 1)*

*PolyTech'Grenoble LIG/ADELE*

**Didier.Donsez@imag.fr**

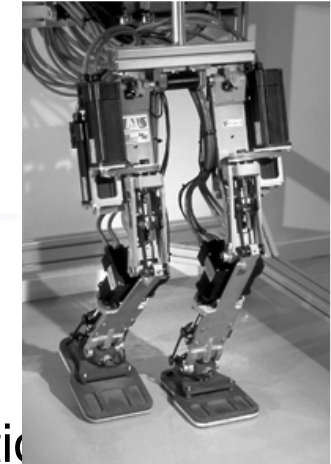
**Didier.Donsez@ieee.org**

# Exemple de systèmes mobiles et les systèmes embarqués

- La liste est longue
  - de la Carte à Puce au « Home Portal »
- SmartCard/ Carte à Puce
- Cellular Phone/ Téléphone mobile
- PalmPC, PocketPC/ Assistant Personnel
- eBook, Cartable Electronique
- Appareil photo, ...
- Set Top Box/ Decodeur TV Numérique
- Game Console/ Console de Jeu
- Electronic Consuming/ Electronique
- Terminal de paiement, Guichet, Parcmètre
- Home Portal/ Portail maison
- Voiture
- Wearable computer (Augmented Reality)
- Prothèse active, Exo squelette
- Satellite, Sonde inter-planétaire
- ...



# Constraint



## ■ Embedded

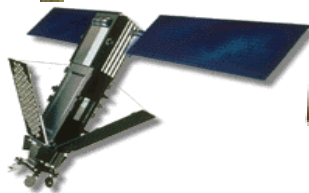
- Cost / product
- Remote Maintenance
- No UI
- Power consumption
- Physical conditions
  - Temperature, Humidity, Vibration, Ions lourds ...
- Real Time (Hard & Soft)

## ■ Mobile

- Cost / product
- Power consumption
- Compact Size
- Consumer resist.
- Limited control &
- Memory capacity
- Limited communication bandwidth
- ...



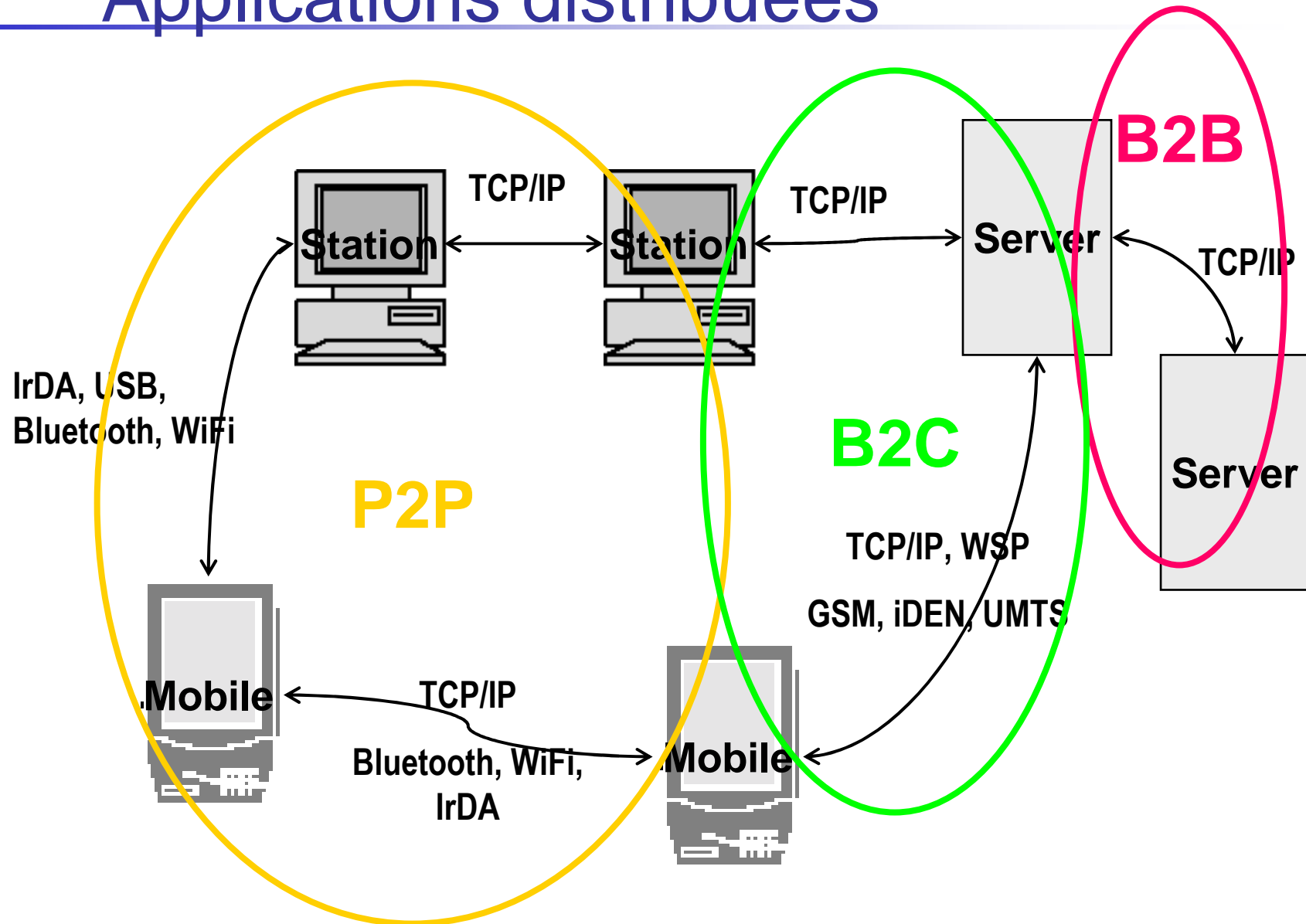
00-2010. Programmation 12ME



# Contraintes matérielles des mobiles

- Faible coût du handset
  - marché grande consommation
- Encombrement et poids faibles (pour les nomades)
- Affichage limité
  - de 100\*60 en NB à 640\*240 en 65565 col.
- Faible mémoire (car grande consommation)
  - primaire (RAM) et secondaire (FlashRAM,EEPROM)
- Autonomie (pour les nomades)
  - Processeur faible consommation d'énergie (ARM, ...)
- Clavier et pointage rudimentaire
  - tablette sensible, écran tactile, PAD, entrée de texte
- Communication
  - Série (IrDA, station d'accueil USB ou RS323), Réseau LAN, Wireless LAN, Cellulaire, ...
- Résistance aux chocs

# Applications distribuées



## WORA

### Avantage de Java pour la programmation d'applications mobiles et embarquées

---

- Partager le même langage, la même conception depuis le terminal jusqu'aux serveurs
- Permet le chargement dynamique de code
- Programmes compacts et portables
- Développement rapide et sûr
- Ateliers et outils homogènes

# Les éditions Java 2

---

- J2SE : la version Standard
  - Développement d'applications (d'applets) côté terminal
- J2EE et JavaONE
  - J2SE + Développement d'applications côté serveur (Servlet, EJB, JMS, XA, SOAP...)
- J2ME
  - 2 configurations de machine
    - CDC : JVM
    - CLDC : KVM
- EmbeddedJava
  - JVM + tout package est optionnel
- PersonalJava (pJava)
  - JVM + nombre réduit de packages (devenu un profile de CDC)
- SavaJe XE
  - J2SE + OS pour des PDA Haut de Gamme <http://www.savaje.com/>
- JavaCard
  - JCVM + packages propre à JavaCard

# Editions Java 2



		Foundation Profile	Personal Profile	RMI Profile	...	PDA Profile	MID Profile	...	GSM Pr.	OP Pr.
J2EE	J2SE	CDC				CLDC			JavaCard 2.x	
		J2ME								
HotSpot VM	HotSpot JVM	CVM				KVM			Bare Metal JCVM	
RTOS	OS				RTOS					
$\mu$ P 32 64 bits UltraSPARC Itanium i7 i5 i3		$\mu$ P 32bits Via Atom ARM11 ARM9				$\mu$ P 16 32 bits AVR32 ARM7			$\mu$ C 8 16 bits AVR8 MSP430	
RAM Usage		100MB		10MB		512KB		32KB		1KB
MW			Energy Consumption						$\mu$ W	



## Les éditions Java 2

---

- 2004Q1: Sun announced that 1.5 billion devices around the world are powered by Java technology
  - 250 million mobile phones
  - 650 million desktops
  - 500 million SIM and Smart Cards
  - 100 million other locations.

# Java 2 Micro Edition

---

- Plateforme Java pour les machines contraintes
  - Enfouies ou/et mobiles
- 2 configurations
  - Connected device config CDC (JSR36)
    - Memoire > 512 KB
    - JVM de J2SE, CVM
    - Cibles : STB, ...
  - Connected limited device config CDLC (JSR30)
    - Memoire 128KB à 512KB
    - KVM (Kauai Virtual Machine)
    - Cibles: PDA, Téléphone cellulaire, ...
- Package J2ME
  - javax.microedition.\*

# CLDC (Connected Limited Device Configuration)

---

- Cibles
  - PDA, Téléphone cellulaire, Pageur, ...
- Configuration
  - $\mu$ P 16 et 32 bits
  - >32KB memoire volatile >128KB mémoire non volatile
  - Contrainte énergétique
  - Connexion réseau possible mais lente
- KVM (Kuaui VM)
  - Langage Java avec des limites
  - VM avec des limites
- Classes
  - Sous ensemble de J2SE 1.3
  - Support limité de l'internationalisation
  - Packages additionnels `javax.microedition`
    - `io,...`

# CDC (Connected Device Configuration)

- Cibles
  - SetTopBox, WebPhone, Console de Jeux Fixes, ...
- Configuration
  - $\mu$ P 32 bits, >256KB memoire volatile >512KB mémoire non volatile
    - JVM+bibliotheque CDC = 2 MB
  - Pas de limite d'energie
- VM
  - CVM= JVM complete mais plus compact
  - JNI, JVMDI (Debugging), JVMPI (Profiling)
- Classes
  - Packages CDC
  - Packages additionnels `javax.microedition`
    - Correspondant à des packages de J2SE

# KVM (Sun)

---

- Objectif : VM légère pour CLDC
  - Pour processeur 16 et 32 bits
  - support de langage Java et presque tout le bytecode Java
- Langage
  - Pas de type float et double
    - Attention : Float et Double ne font pas partie des classes CDLC
  - Pas de méthode finalize dans java.lang.Object
- VM
  - Pas de flottant (**float, double**)
  - Pas de JNI
  - Pas de chargeur de classes défini par le développeur
  - Pas de compilation à la volée (JIT) ou HotSpot
  - Pas de reflexion (java.lang.reflect)
  - Pas de groupe de threads et de threads démon
  - Pas de finalisation
  - ...

# Monty (Sun)

---

- Nouvelle VM pour CDLC
  - Meilleure performance qu'une KVM
    - 7 (StrongArm/WinCE) à 10 (x86/Win32) fois plus rapide sur un mix d'applications
- Améliorations
  - Compilation adaptative (pour les HotSpots)
    - une passe, simple et rapide
    - Cache LRU du code
    - Inlining et uninlining
  - Compact Object Layout
  - Explicit Type Tagging
    - (recherche des références sur la pile pour le GC)
  - Gestion unifié des ressources
    - Objets alloués, Classes chargées, structure de la VM, Code compilé
- Voir (JavaOne2002)
  - <http://servlet.java.sun.com/javaone/resources/content/sf2002/conf/sessions/pdfs/2133.pdf>

# CDC HotSpot *anciennement CVM* (Sun)

<http://java.sun.com/products/cdc-hi>

- Performance améliorée par rapport à la CDC
- Dynamic JIT
  - Econome en mémoire et rapide
  - Configurable : taille du cache de code, politiques de compilation et décompilation
- Très portable
  - ARM, PowerPC, MIPS, SPARC, x86 sur Linux, Solaris, VxWorks
  - *Retargeable* (écrit en C & Asm)
- RTOS-aware,
  - Ordonnanceur déterministe
  - Synchronisation rapide
- Threads
  - directement sur des threads natives
- Efficacité en espace mémoire
  - Classes "ROMable" (*In place execution from ROM*)
    - Classes en ROM préchargées (gain de 40% par rapport à une JVM)
  - Pile réduite
- Supporte les spécifications de Java 2 Platform
  - weak references, réflexion, sérialisation, full networking, JNI, RMI, JVMDI, JVMPI ...

# MicrochaiVM (HP)

---

- Objectif : VM légère
  - Support KVM (processeur 16 et 32 bits)
  - JNI propriétaire
  - Moins gourmande en mémoire
    - (37Ko contre 128 Ko pour la KVM)
  - Performance
- Plate-formes
  - PalmOS/Dragonball (68K), Linux/x86, PocketPC/SH3, PocketPC/StrongArm
  - Guide de Portage : VxWorks (en projet)
- Technologie Chaifreezedry
  - Exécution des applications sous une forme « concentrée »



# MicrochaiVM (HP) - API

---

- Core API:
  - Ensemble minimale de classes pour l'exécution de la VM
- Core/Lite API:
  - - multithreading et entiers 64-bit integers
- CLDC API:
  - Core API
  - + CDLC (SUN)
  - + support for byte and character streams, internationalization, class instances, collection, and utility classes

## MicrochaiVM (HP) - Chaifreezedry

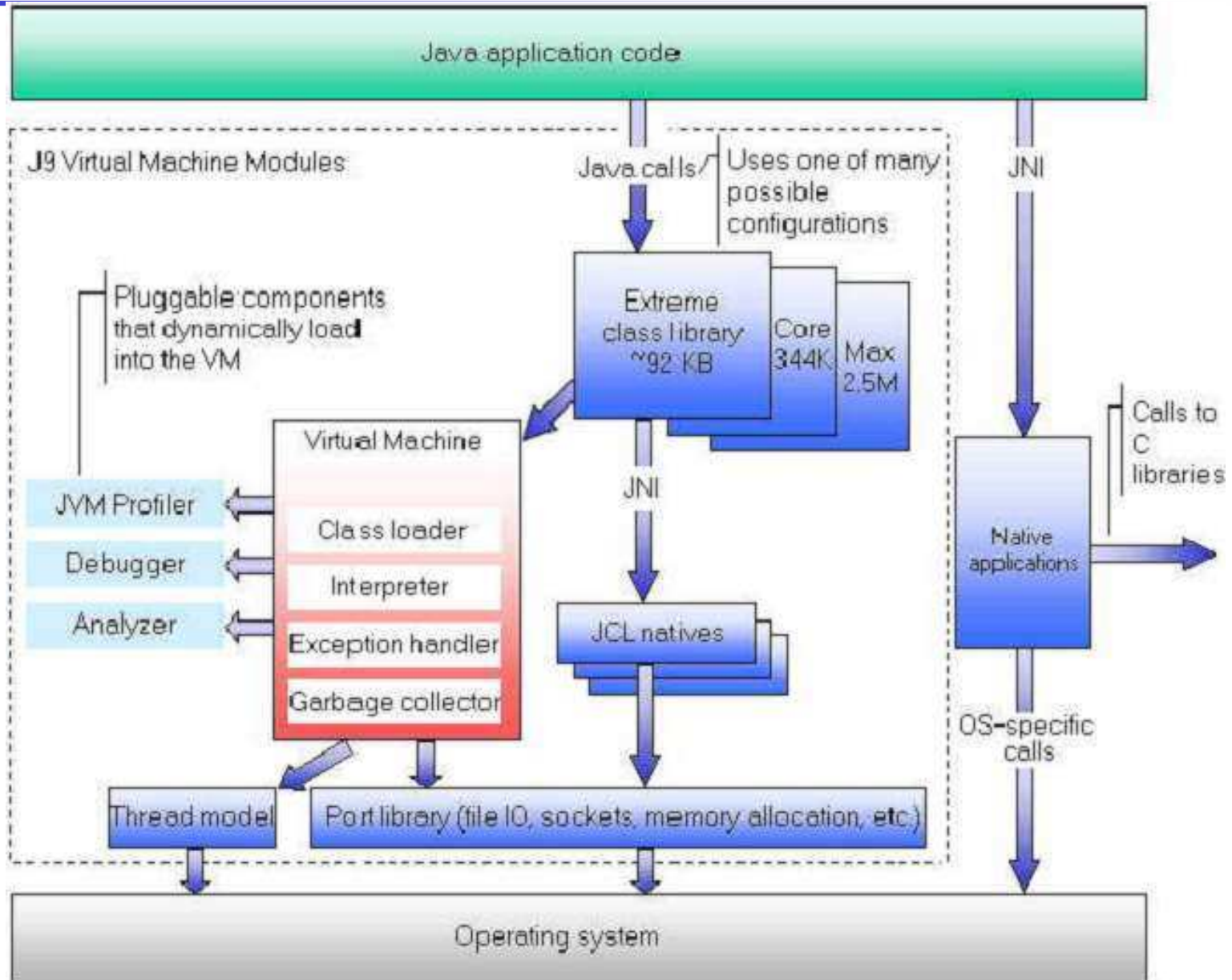
- Freezedry : Format condensé de bytecode Java
  - Spécifique à la MicroChai
  - réalisé au chargement
  - avant chargement ou ROMification (par un outil 'fdcon')
- Gain
  - Réduction de la taille des applications 50-90%
  - Moins de surcharge aux chargements des classes
    - Amélioration du délai de démarrage
  - byte-swapping rapide (pour de  $\mu$ P little-endian)
  - Application byte-swappées en ROM

# J9 d'IBM

<http://www-306.ibm.com/software/wireless/wece/>

- Machine Virtuelle pour l'embarqué
  - Contraintes mémoire
  - Contraintes Temps-Réel (RTOS)
  - Java Native Interface
    - Drivers périphérique spécifiques (MP3, Bluetooth, ...)
  - Profilage, Analyse, Debuggage
  - Basé sur les spécifications du Java 2 Platform (1.3)
- Atelier
  - WebSphere Micro Environment
    - Anciennement Visual Age Micro Edition (Windows, Linux)
- Systèmes d'exploitation
  - PalmOS, Linux, **Windows® CE**, QNX Neutrino.
- Processeurs:
  - *Xscale™, X86, PowerPC™, ARM™, MIPS, SuperH™*

# J9VM d'IBM



<http://www.insignia.com/content/products/pda.shtml>

---

- EVM (Embedded Virtual Machine)
  - Machine Virtuelle pour l'embarqué
    - Contraintes mémoire
    - Dynamic adaptive compilation
      - six times faster than interpretive JVMs in roughly the same amount of memory.
    - Profile
      - Basé sur les spécifications du JDK1.1
      - PersonalJava et EmbeddedJava
- Systèmes d'exploitation
  - Windows CE, Windows NT4, VxWorks, Linux, ITRON, Nucleus, BSDI Unix & pSOS.
- Processeurs:
  - MIPS, ARM, Intel x86, PowerPC, Hitachi SH-3 & SH-4
- Licence
  - 40 millions de licences vendus

# WABA

<http://www.wabasoft.com/> &  
<http://waba.sourceforge.net/>

---

- Waba Virtual Machine
  - Sous-ensemble strict de Java en terme de syntaxe, fichier class et bytecode
- WABASDK
  - Plate-forme de développement propriétaire
    - Format propriétaire des classes
    - Classes de base propriétaire
    - ATTENTION : Ne suit pas les spécifications de SUN (J2ME, ...)
  - Nombreux OS (BeOS, Amiga, ...)
- VMs pour
  - PalmOS, Windows CE, DOS
  - *Apple Newton, et même TI Calculator*
- License GNU GPL

# SuperWABA

<http://www.superwaba.org/>

---

- Virtual Machine
  - Waba VM
  - Support pour les entiers et flottants longs (64b)
  - Native plugin's (C et Java)
- Bibliothèques
  - GUI, Color, Font, ...
- VMs pour
  - PalmOS, Windows CE
- IDE
  - Tauschke MobileCreator, VisualWaba, JDK 1.1 x
- Licence GNU LGPL

# Kaffe (kaffe.org)

---

- JVM Open Source
  - plutôt J2SE mais nombreux portages sur les OS et processeurs nomades et embarqués
- A regarder
  - OS (contraint)
    - Windows CE, KaffeCE,  $\mu$ Clinux, Sony Playstation 2 (Japanese), iPAQ (voir Handhelds.org), Zaurus, Yopy, PocketLinux (<http://www.transvirtual.com/~jsimmons/yopy/>), Helio
  - Embedded OS / RTOS
    - eCOS, VxWorks, pSOS, ThreadX, SMX, Nucleus, RTEMS, AtheOS, Plan9, Atari, Amiga, ...
  - Processeurs
    - x86 / IA32, ARM, MIPS, PowerPC, Sparc, SuperH, m68k, CRIS, S390, IA64, Alpha, HP PA RISC, DCT lightfoot (Java core)



## Jbed (<http://www.esmertec.com>)

---

- RTOS (SE Temps Réel)
  - Ordonnancement des Threads, Allocation mémoire, GC, Exception
- Cross-Compilation Java avant chargement
  - TBCC™ (Target Byte Code Compiler)
- API et Implantations
  - Jbed CLDC, Jbed MIDP, Jbed Profile
  - Protocoles IP
    - UDP, TCP, PPP, SLIP, TFTP, HTTP, ICMP, ARP, DNS, DHCP, IGMP
  - Embedded Web server (HTTP 1.0)
  - Messaging Middleware : iBus™ Family
- Processeurs
  - ARM, PowerPC, 68xxx et Coldfire

# JTRON

<http://www.tron.org/index-e.html>

- UNDER CONSTRUCTION
- TRON
  - OS Temps Réel open-source principalement utilisé pour l'électronique grand public (ITRON) japonaise
  - Projet transféré au T-Engine Forum <http://www.t-engine.org/>
- JTRON sous projet de ITRON (I pour Industrial)
- Voir
  - [http://www.assoc.tron.org/eng/intro/TRON\\_intro.pdf](http://www.assoc.tron.org/eng/intro/TRON_intro.pdf)
  - Hachiya, S, « Java use in mobile information devices: Introducing JTRON », IEEE Micro, Volume: 21 Issue: 4 , July-Aug. 2001, Page(s): 16 -21

# CrEme

<http://www.nsicom.com/Default.aspx?tabid=138>

- Windows-CE PDA
  - Pocket PC and Windows Mobile devices
- J2ME/CDC/Personal Profile
- GUI
  - Truffle (implementation of AWT) by default
  - Swing
  - Tiny AWT
  - Eclipse SWT
  - supports non-western languages
- Standard Java Native Interface
  - For C, C++
- Third-party libraries
  - Symbol device features, like bar-code scanners, magnetic stripe readers, wireless communications, etc.

# Cacao

<http://www.cacaojvm.org/>

---

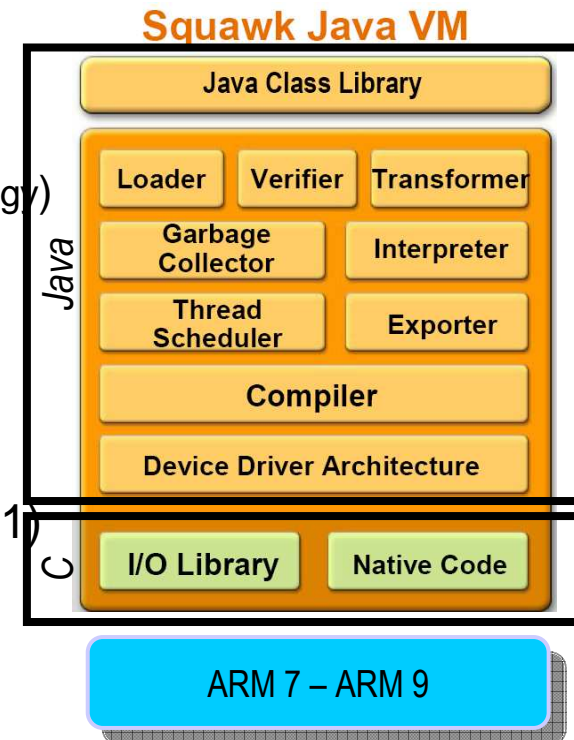
- Supported architectures
  - Alpha, MIPS (64-bit), x86, x86-64, PowerPC (32-bit), **ARM**
  - Linux, Darwin, Irix6, FreeBSD, ...
- Features
  - Native thread support
  - Java 1.4 support
  - GNU classpath 0.12
  - Preliminary AWT support
  - Boehm garbage collector
  - Reentrant JIT compiler

# TinyVM (<http://tinyvm.sourceforge.net>).

- VM utilisé par Lejos pour le RCX du Lego MindStorms
  - Limitations
    - pas de GC, pas de flottant, pas de switch, pas de constants String
  - Empreinte 10Ko
  - Optimisation de la mémoire
    - 1 octet pour 1 byte, 4 octets pour un objet...
- Runtime Environnement
  - java.io, java.lang, java.net, java.util
  - javax.servlet.http (car Lejos embarque un serveur Web)
  - josx.platform.rcx, josx.rcxcomm, josx.robotics, josx.util
- Voir le cours sur Lego MindStorms

# Squawk (Sun)

- Motivations
  - Fully capable J2ME-level Java VM with OS functionality
    - Wireless Sensor and Actuator Devices
      - SUN SPOT (Small Programmable Object Technology)
    - Memory constrained devices
- Features
  - No operating system
    - Runs on the bare metal on ARM
    - Device driver are written in Java
  - Isolate application model (sub/super set of JSR121)
    - Inter-isolate communication
      - Provides lower-level asynchronous message delivery
  - Memory footprint
    - 80K for the VM, 270K for the Runtime
- See
  - Squawk <http://research.sun.com/projects/squawk>



# Autres JVM pour $\mu$ C

---

- Cibles
  - Capteurs Zigbee & 6lowPan
  - Cartes à puce
- **JVM**
  - **NanoVM, TinyVM, VM\*, Darjeeling, ...**
- Application Specific VM
  - Maté, VMSCRIPT

# E-Bunny (U. Concordia & U. Laval)

- TODO

- See

- Mourad Debbabi et al: « E-Bunny: A Dynamic Compiler for Embedded Java Virtual Machines », in Journal of Object Technology, vol. 4, no. 1, January–February 2005, pages 81–106, [http://www.jot.fm/issues/issues2005\\_01/article2](http://www.jot.fm/issues/issues2005_01/article2)



16/11/2010

# phoneME

<https://phoneme.dev.java.net/>

---

- TODO

# Dalvik JVM

in Android, Google's Linux/Java mobile phone SDK (software development kit).

<http://code.google.com/android>



## MicroJvm (<http://www.ist-eu.com/>)

- JVM for embedded systems
  - License : commercial
  - Cores : ARM7, ARM9, AVR32, AVR, Blackfin, Octeon, PowerPC, V850, ...
  - Small footprint : 50 KB ROM
- Features
  - incremental & compacting GC, F(ast)NI (Proprietary JNI), Bare metal & RTOS support ...
- Tools
  - SDK, Romizer, Simulator, ... (Eclipse integration)

# D'autres JVM

---

- JamVm
  - <http://jamvm.sf.net>
  - Linux, ARM9
- SableVm
  - <http://sablevm.org>
- Blackdown
  - <http://www.blackdown.org>
- Myna JVM
  - [http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index\\_en.html](http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html)
  - Windows mobile, GPLv2
- GNU ClassPath: [www.gnu.org/software/classpath](http://www.gnu.org/software/classpath)
  - Runtime Java

# BREW

*Binary Runtime Environment for Wireless*



- Environnement d'exécution léger pour les téléphones mobiles (CDMA)
  - Classes pour gérer communication, UI, ...
  
- Langages
  - C, C++, Java (J9 d'IBM)
  
- Outils
  - BREW SDK
    - <http://www.qualcomm.com/brew/>

# Bada (Samsung)

<http://www.bada.com/>

---

- Annoncé début 2010
- Environnement pour touch phone
  - Alternative à Android
- Langage : C++
- SDK

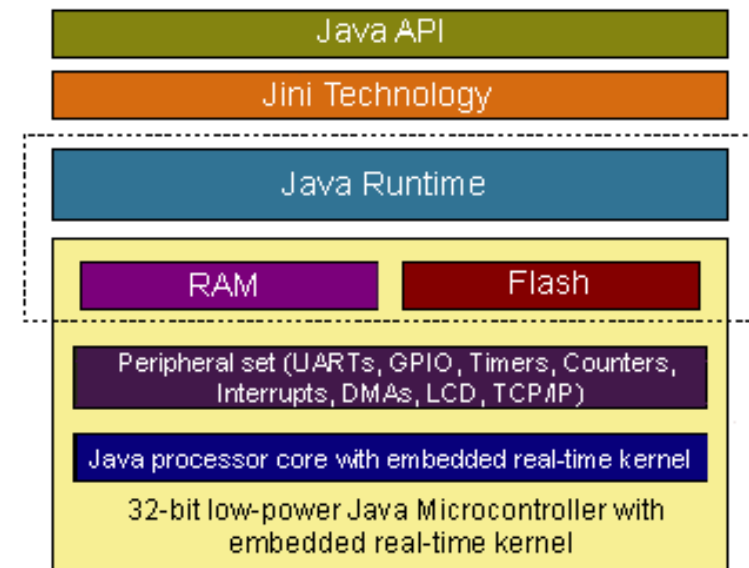
# Java pour iPhone (Apple)

---

- TODO
- Compilateur Java → Objective-C

# Java dans le silicium

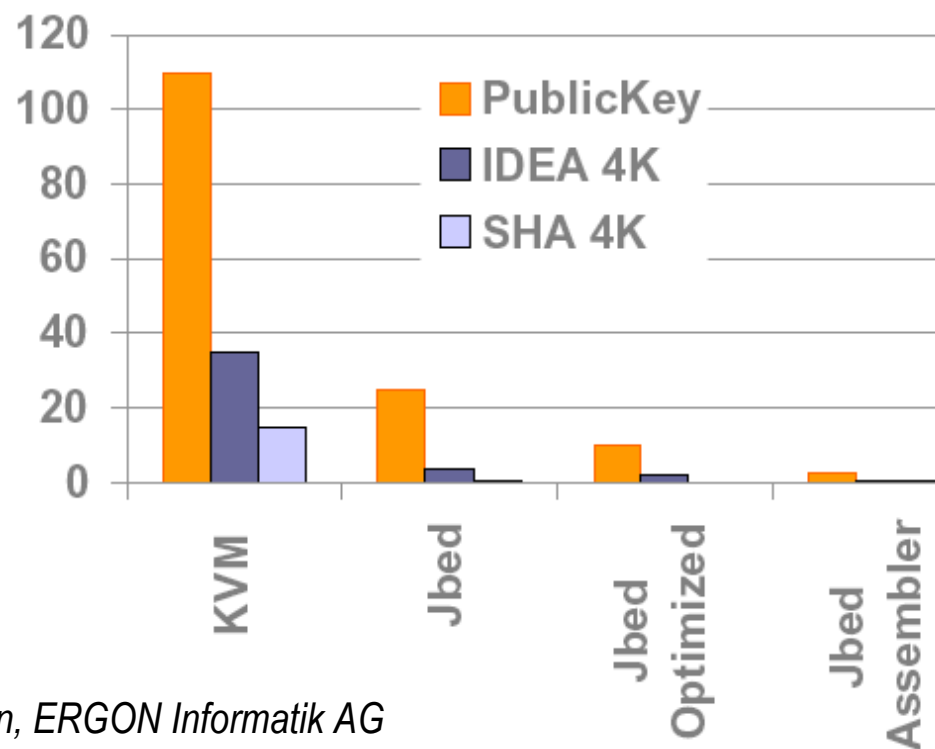
- aJile (<http://www.ajile.com>)
  - tourné vers l'embarqué / les mobiles / le temps réel
  - $\mu$ P dont le jeu d'instruction inclut ceux de la JVM
    - Micro-programmé
    - Thread switch context : 1 $\mu$ s
  - Runtime
    - J2ME/CLDC
    - J2ME/CDC (annoncé)
- ARM 926 Jazelle
  - TODO





# Performance entre VMs

- Plusieurs critères : CPU, Mémoire, ...
- Exemple
  - Algorithmes de cryptographie
    - sur PalmVx, OS3.5, Dragon Ball 20 Mhz



• D'après Jakob Magun, ERGON Informatik AG

# Les profiles J2ME

- Motivation
  - Définir des API minimums pour des « devices » similaires
- Profiles J2ME proposés (JCP) ou spécifiés
 

■ MID (Mobile Information Device ) 1.0	CDLC	JSR37
■ MID (Mobile Information Device ) 2.0	CDLC	JSR118
■ PDA	CDLC	JSR75
■ Foundation	CDC	JSR46
■ PersonalJava Basic		CDC JSR129
■ PersonalJava	CDC	JSR62
■ RMI	CDC	JSR66
■ Game	CDC	JSR134
■ ...		
- Déterminer le profile
  - Propriété `microedition.profiles`

# Le profile Foundation

---

- 436 classes/interfaces/exceptions
- Packages issus du J2SE (excepté les *deprecateds*)
  - java.io
  - java.lang
  - java.lang.ref
  - java.lang.reflect
  - java.math
  - java.net
  - java.security
  - java.security.acl
  - java.security.cert
  - java.security.interfaces
  - java.security.spec
  - java.text
  - java.text.resources
  - java.util
  - java.util.jar
  - java.util.zip
  - javax.microedition.io

# Le profile Personal Basic (JSR129)

- Cible
  - Entertainment device (voir JavaTV)
    - Game Console, iTV STB, DVB MHP terminals, ...
  - Vertical apps : Auto and telematics
    - Custom GUI, pas d'applets
- Basé sur Foundation 1.0 (J2ME/CDC)
  - AWT simple
    - Une seule frame Top Level, Transparence Alpha
  - Modèle d'applications
    - main(),
    - java.microedition.xlet
      - un classloader par Xlet
      - + IXC mécanisme de communication inter Xlet (sorte de RMI intra VM)
  - JavaBeans
  - RMI (donc Sérialisation), IXC (java.microedition.xlet.ixc)

## Le profile Personal (JSR62)

- Profil J2ME/CDC de PersonalJava (antérieur à J2ME)
- Cible
  - Entertainment device
    - STB avec des fonctions de Mailer et Web Browser ...
- APIs du profile Personal Basic
  - + Applet
  - + Sérialisation avec J2SE
  - + Extension AWT
    - Multiple frames, windows, dialogs
    - Widgets (JDK1.1) lourds (canvas, Panel, Scrollbar, Button ...)

# Le profile Game (JSR-134)

- Profil J2ME/CDC
- Cible
  - Console de jeu fixe
  - (Console de jeu mobile + téléphone)
- APIs
  - J2SE 1.4
  - Java 3D
- Voir [JavaGaming.org](http://JavaGaming.org)
- **Prévisions pour les jeux sur téléphone mobile:**
  - 200 M€ en 2002, 3000 M€ en 2008
- **Rappel**
  - revenu mondial 2001 des jeux vidéo : 27G€ (2 X + le cinéma)

# Jeux « mobile » en Java

- Gratuit ou Payant
  - Arcade, Réflexion, Stratégie, Rôle ...
  - Tamagochi
  - 2D ou 2,5D
  - Standalone ou en réseau
- Sociétés
  - <http://www.gameloft.com/> (société devenue béni
  - <http://www.betomorrow.com>
  - <http://www.macrospace.com/games.shtml>
  - <http://midlet.org/index.jsp>
  - ...
- Remarque
  - *Nokia s'est attaqué au marché de consoles m*
  - *LG suit avec le succès de la NGage*



# Le profile J2ME CLDC/MIDP (Mobile Information Device Profile)



- **Cibles** : Mobile Information Device (Téléphone, Pager, PDA)
  - Ecran monochrome ou couleur > 96x54 pixels
  - Ecran tactile, Clavier limité (KeyPad)
  - Réseau sans fil bas débit
  - 32KB memoire volatile
  - 128 KB memoire non volatile pour les classes MIDP
  - 8 KB memoire non volatile pour la persistance des données
- **Classes**
  - MIDlet : défini et contrôle le cycle de vie d'une application
    - Affichage Texte et Graphique (orienté formulaire)
    - Persistance dans des fichiers structurés en enregistrement (Record)
    - Timer, User events
    - Réseaux : HttpConnection, UDP, SMS/MMS, BT, ...



# Le profile J2ME CLDC/MIDP (Mobile Information Device Profile)



## ■ Développement et Déploiement

- Développement (javac)
- Preverify (KVM)
  - Remarque : l'option `-target cldc10` du javac 5.0 réalise la prévérification
- Packaging (.jar)
- Description (.jad)
- Déploiement (OTA ou en direct)

## ■ Tools

- Sun Wireless TK, Inprise' Jbuilder J2ME, Forte, Plugin Nokia pour Eclipse ...

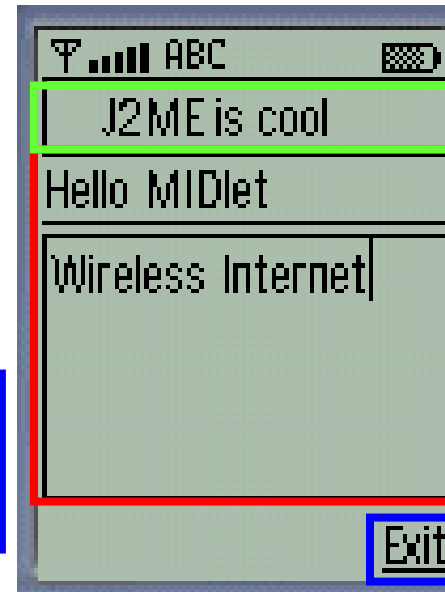
# J2ME CDLC/MIDP

## Exemple de MIDlet

```

import javax.microedition.midlet.*; import javax.microedition.lcdui.*;
public class HelloMIDlet extends MIDlet implements CommandListener {
    private Command exitCommand;
    private Display display;
    private Ticker hi = new Ticker("J2ME IS COOL");
    public HelloMIDlet() {
        display = Display.getDisplay(this);
        exitCommand = new Command("EXIT", Command.SCREEN, 2);
    }
    public void startApp() {
        TextBox t = new TextBox("Hello MIDlet",
            "Wireless Internet", 256, 0);
        t.addCommand(exitCommand);
        t.setCommandListener(this);
        t.setTicker(hi); // set the ticker
        display.setCurrent(t);
    }
    public void pauseApp() {}
    public void destroyApp(boolean unconditional) {}
    public void commandAction(Command c, Displayable s) {
        if (c == exitCommand) {
            destroyApp(false);
            notifyDestroyed(); }
    }
}

```



# Déploiement des applications

- 2 fichiers
  - Fichier JAD ( `text/vnd.sun.j2me.app-descriptor` )
    - informations partagées avec le MANIFEST du JAR
  - Archive JAR

- Exemple

**Created-By: 1.3.1 (Didier DONSEZ)**

**MIDlet-1: Hello World, /icons/hello.png, examples.hello.HelloMIDlet**

**MIDlet-2: Http Browser,, example.http.DownloadMIDlet**

**MIDlet-Description: suite de midlets de demo.**

**MIDlet-Jar-Size: 18387**

**MIDlet-Jar-URL: <http://www-adele.imag.fr/~donsez/midlets/demos.jar>**

**MIDlet-Name: Exemple Cours J2ME**

**MIDlet-Vendor: Didier DONSEZ**

**MIDlet-Version: 1.0.3**

**Manifest-Version: 1.0**

**URLTOLOAD : <http://www-adele.imag.fr/~donsez/midlets/note.txt>**

# Déploiement des Applications

- OTA : Over the Air
  - Installation
    - MIDlet-Install-Notify The URL to which a POST request is used to confirm successful installation of this MIDlet suite
  - Mise à jour
  - Suppression
    - Confirmation de l'utilisateur
      - MIDlet-Delete-Confirm contient le texte à afficher

# Déploiement

---

- TODO
- Schéma d'architecture vu en TD

# MIDP 2.0 (i)

---

- Quoi de neuf ? (disponible des été 2003)
  - UI
    - Multimédia : Mobile Media API (MMAPI)
    - Amélioration des formulaires
    - Images RGB
  - Game API
    - `javax.microedition.lcdui.game`
      - GameCanvas, LayerManager, Layer, TiledLayer, Sprite

# MIDP 2.0 (ii)

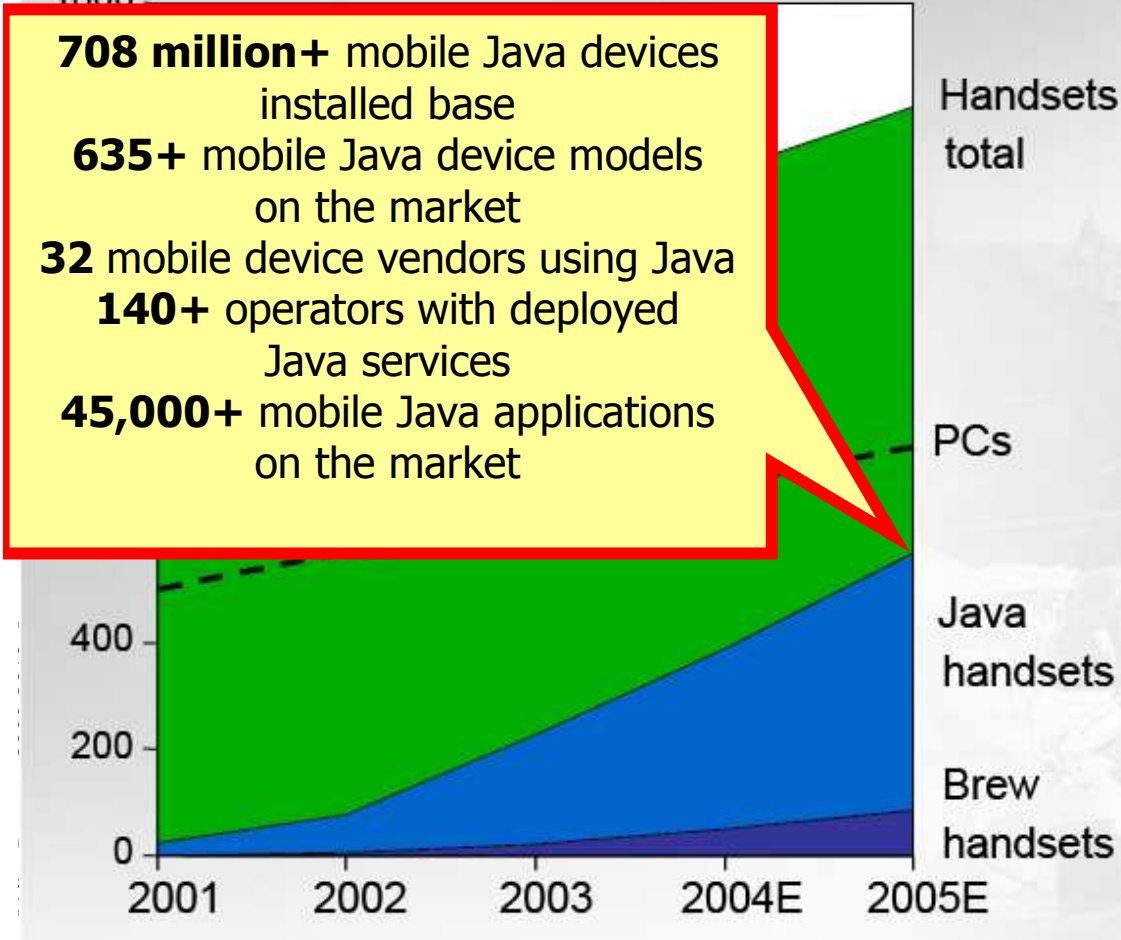
---

- Quoi de neuf ?
  - Sécurité
    - HTTPS (au dessus de TLS, SSL ou WAP WTLS)
    - PKI (Certificats)
    - JAR signé
      - La signature et la clé publique sont ajoutés au JAD
    - Permissions
      - Nouvelles entrées du JAD : MIDlet-Permissions, MIDlet-Permissions-Opt
  - Lancement Push de MIDLet sur des demandes entrantes de connexions réseaux
    - L'entrée `MIDlet-Push-<n>` du JAD précise
      - l'URL local de connection URL
      - la classe de la MIDLet
      - l'adresse IP de la machine autorisée à lancer un push
    - Exemple : `MIDlet-Push-1: socket://:76, exemple.PushLet, 192.161.60`

# Java is the leading mobile Application Development Environment

Global installed base millions

1600



Key message in 2002  
JavaOne:  
***“We will put Java in every pocket”***  
...done.

Key message in 2005  
JavaOne:  
***“We will put Java server in every pocket”***  
...working on it...





# Java CDLC pour PalmOS

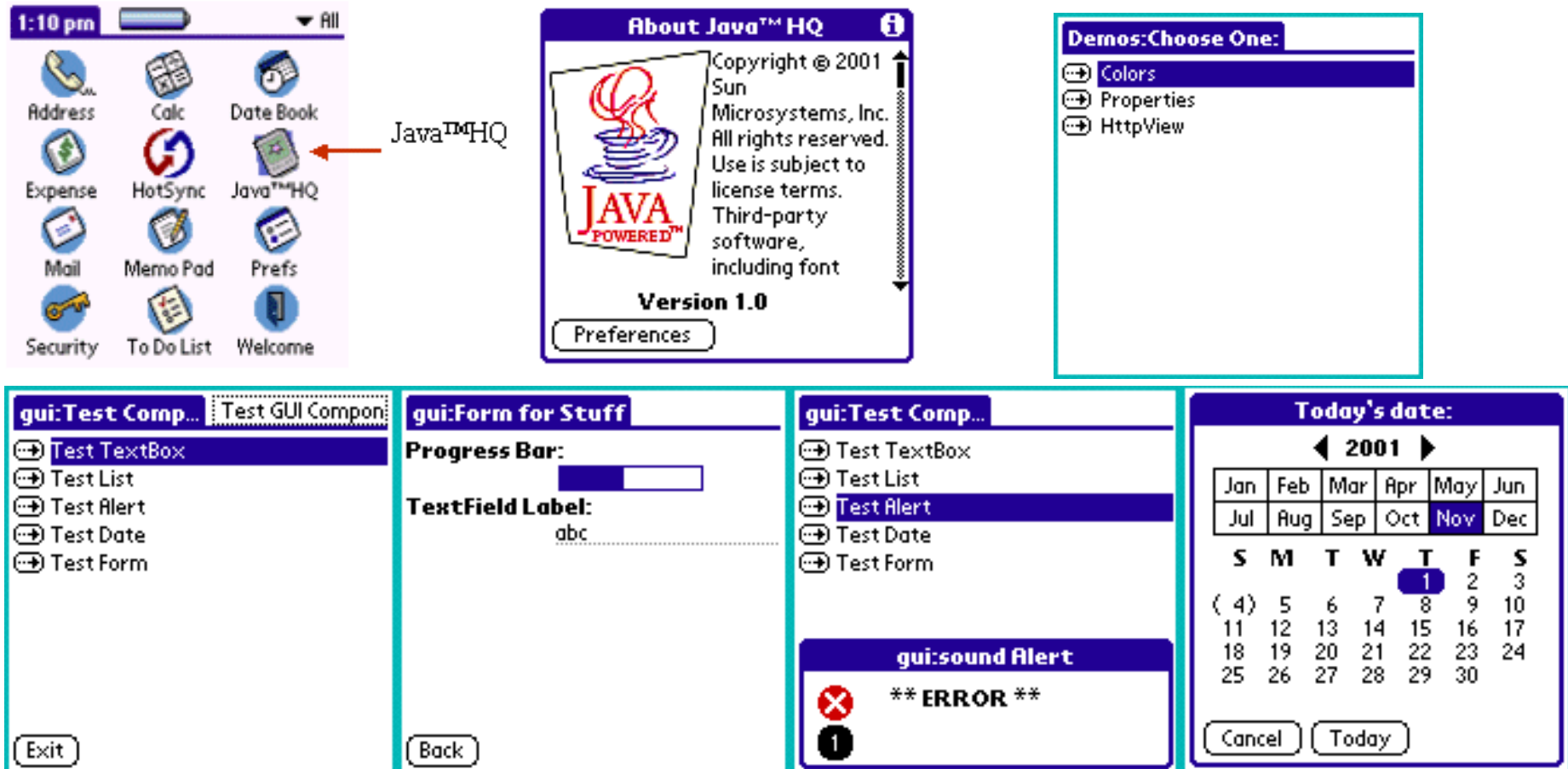
---

- Installation KVM
  - KVM.prc et KVM-util.prc (configuration de la VM)
- Compilation application
  - Compilation (javac) + preverify+ palm.database.MakePalmApp
- API
  - Application : `com.sun.kjava.Spotlet`
  - CDLC
  - Classes additionnelles
    - Événements utilisateur
    - Graphique et Contrôles
    - Database (PDB), Réseaux
- Autres
  - kAWT et ColorAWT
  - Jbed (KVM avec compilation à la volée pour Palm)

# Java MIDP pour PalmOS

<http://wireless.java.sun.com/midp/articles/palm/>

- MIDP.prc : KVM + classes MIDP (600Ko)
- Convertisseur .jad/.jar vers .prc



# Les API CDLC et MIDP

- CLDC
- java.lang
  - Version allégée du J2SE
    - - de classes, - de méthodes
- java.util
  - Version allégée du J2SE
    - - de classes, - de méthodes
- java.io
  - ByteArrayInputStream  
ByteArrayOutputStream  
DataInputStream DataOutputStream  
InputStream InputStreamReader  
OutputStream OutputStreamWriter  
PrintStream Reader Writer
- javax.microedition.io
  - *Connection ContentConnection  
Datagram DatagramConnection  
**HttpConnection** InputConnection  
OutputConnection StreamConnection  
StreamConnectionNotifier*
- MIDPv1.0
- javax.microedition.midlet
  - Cycle de vie d'une application
- javax.microedition.lcdui
  - Interface orienté formulaire
    - Alert AlertType Canvas ChoiceGroup  
Command DateField Display Displayable  
Font Form Gauge Graphics Image  
ImageItem Item List Screen StringItem  
TextBox TextField Ticker
- javax.microedition.rms
  - orienté Persistance
    - **RecordStore** *RecordEnumeration  
RecordComparator RecordFilter  
RecordListener*
- MIPD v2.0
- javax.microedition.media
- javax.microedition.lcdui.game

# Messaging

- Wireless Messaging API (JSR120) pour J2ME
  - Messageries pour pager et téléphone mobile
    - SMS, ...
  - Basé sur le Generic Connection Framework (GCF) de CDLC
  - `javax.wireless.messaging`
    - `Message`, `TextMessage` et `BinaryMessage`.
    - `MessageConnection`, `MessageListener`

- Exemple d'envoi de messages

```
String addr = "sms://+33675123456";  
MessageConnection conn = (MessageConnection) Connector.open(addr);  
TextMessage msg =(TextMessage)conn.newMessage(  
    MessageConnection.TEXT_MESSAGE);  
msg.setPayloadText("Hello World!");  
conn.send(msg);
```

# Multimedia

---

- Mobile Media API (JSR135) pour CDC/CDLC
  - Temporel (son, MIDI, vidéo, streaming ...)
  - en remplacement de JMF (trop gros 650K, AWT, Calcul flottant, ...)
  - Compact : 150K (MIDI, MPEG1, MP3, ...)

# Accès aux périphériques

---

- JSR-000246 Device Management API
  - Enabling J2ME applications to access device management implementations

# Géo-localisation

---

- JSR179 Location API for J2ME
  - Principalement coordonnées GPS et « LandMark »

## Autres profiles J2ME

---

- J2ME/CDLC for JStamp ([www.systronic.com](http://www.systronic.com), [www.ajile.com](http://www.ajile.com))
  - Droid et Robot (LEGO) MindStorm
  - CDC prévu
- STIP (Small Terminal Interoperability Platform)
  - Cible : Terminaux de paiement, ...
  - Ajoute des API liés aux lecteurs (cartes, ...) et à la sécurité
- iMODE/DoJa (NTT Docomo) <http://www.kvmworld.com/technologies/iappli>
  - midlet (iAPPLI) pour les téléphones de NTT Docomo
  - ajoute des API supplémentaires à CDLC/MIDP
    - Téléphonie, Multimedia, ...
- TINI (Tiny InterNet Interface) [www.ibutton.com/tini](http://www.ibutton.com/tini)
  - basé sur le JDK1.1.8 avec quelques limites ...



# Le profil PDA (JSR75)

- Profil sur J2ME/CDLC
  - 900 K d'empreinte
- Cible
  - Assistant Personel
    - Symbian, PalmOS, ...
- APIs
  - Application : PDAlet
    - Deploiement dans un JAD
      - Nouveaux attributs
      - Midlet et PDAlet peuvent etre dans la meme suite
  - UI : AWT (look and feel natif)
  - PIM, Fichier, Ports série

# Interfaces natives

---

- CDC
  - JNI
  
- CDLC
  - KNI

# Autres API pour J2ME

---

## ■ Persistence

- RMS, JDBC light, BD OO
- DB2e (EveryPlace), Oracle
- JDO (Java Data Object)
  
- Des SGBD Java embarqués
  - HSQLDB, TinySQL, ...

## ■ MOM

- kJORAM (client léger pour KVM)
  - Connexion TCP et SOAP/HTTP

# Autres API pour J2ME

---

## ■ Persistence

- RMS, JDBC light, BD OO
- DB2e (EveryPlace), Oracle
- JDO (Java Data Object)
  
- Des SGBD Java embarqués
  - HSQLDB, TinySQL, ...

## ■ MOM

- kJORAM (client léger pour KVM)
  - Connexion TCP et SOAP/HTTP

# Autres API pour J2ME

---

## ■ UI

- kAWT et kKVM (ex ColorKVM)
  - Widgets pour KVM (<http://www.trantor.de/kawt/>)
- HAVi (Home Audio Video initiative)
  - Widgets pour la iTV (Personal Java)
- LWUIT
  - <https://lwuit.dev.java.net/>
- JavaFX ???
  - <http://java.sun.com/javafx/>

## ■ XML

- Parseurs : kXML ([kxml.enhydra.org](http://kxml.enhydra.org)), NanoXML, TinyXML, MinML, ...
  - Voir Chap 10 du [Feng]
  - <http://servlet.java.sun.com/javaone/resources/content/sf2002/conf/sessions/pdfs/2716.pdf>
  - <http://www.wilson.co.uk/xml/minml.htm>

# LWUIT <https://lwuit.dev.java.net/>

- UI library
  - supports visual components and other UI goodies such as theming, transitions, animation and more.
- Key Features
  - [Swing](#) Like MVC (Model View Controller)
  - Layouts
  - PLAF (Pluggable Look And Feel) & Themes
  - Fonts
  - Touch Screen
  - Animations & Transitions
  - Rich widgets
  - 3D Integration (Optional)
  - Painters
  - Modal Dialogs
  - External Tools
  - I18N/L10N
  - SVG Integration (Optional)
- See
  - <http://lwuit.blogspot.com/>
  - **LWUIT 1.1 for Java ME Developers**, Biswajit Sarkar , **ISBN-10: 184719740X**

# Autres API pour KVM

---

- P2P
  - Project JXTA for J2ME
    - <http://jxme.jxta.org/servlets/ProjectHome>
    - <http://servlet.java.sun.com/javaone/resources/content/sf2002/conf/sessions/pdfs/3072.pdf>
- Midlet sur pJava : ME4SE
  - voir <http://kobjects.org>

# J2ME Web Service

---

- J2ME Web Services (JSR172 )
- Implémentation de SOAP pour KVM
  - kSOAP [ksoap.enhydra.org](http://ksoap.enhydra.org) ...



# REST pour J2ME

---

- TODO

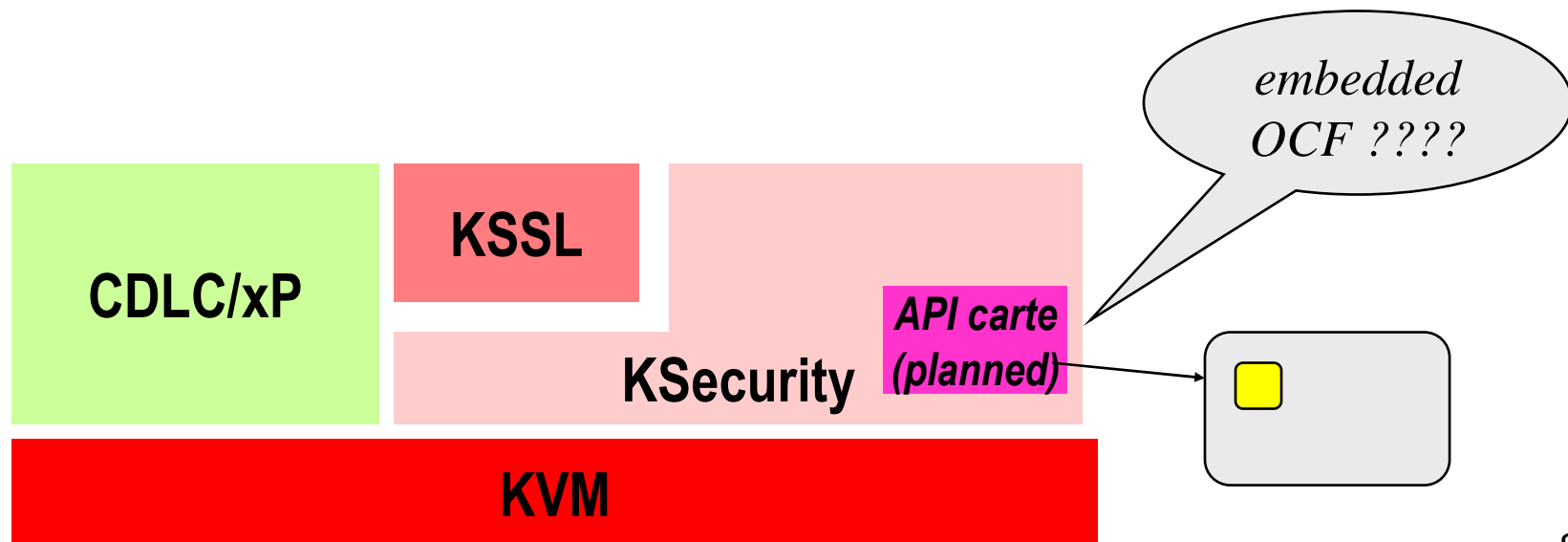
# Java et Temps Réel

---

- J-Consortium <http://www.j-consortium.org>
  - Voir cours « Java et Temps Réel »

# Cryptographie sous J2ME

- Allégement du JCE J2SE
  - <http://www.bouncycastle.org> a « allégé » l'API crypto de JCE pour la KVM
- Combiné J2SE et JavaCard
  - KSSL <http://playground.sun.com/~vgupta/KSSL>



# Crypto J2ME

---

## ■ Interfaces et Classes

### ■ com.sun.ksecurity

- Key\*, KeyStore ‡, PrivateKey\*, PublicKey\*, RSAPrivateKey\*, RSAPublicKey\*, Certificate ‡, KeyBuilder\*, MessageDigest\*, RandomData\*, SecretKey\*, CryptoException\*

### ■ com.sun.kssl:

- SSLStreamConnection, HandshakeListener, Cipher\*\*, Signature\*, X509Certificate ‡

## ■ Modèle d'origine

- ‡J2SE java.security
- \*\* javacardx.crypto
- \* javacard.security

# Crypto J2ME

---

- Keys
  - Symmetric keys (different lengths),
  - RSA Public/Private Keys with modulus lengths up to and including 1024. No support for DSA keys.
- Ciphers
  - ALG\_RSA\_PKCS1, ALG\_ARCFOUR (no DES, 3DES, AES etc).
- Message Digests
  - MD5 and SHA (no MD2).
- Signatures
  - ALG\_RSA\_MD5\_PKCS1 and ALG\_RSA\_SHA\_PKCS1. No DSA support.
- Certificates:
  - Only supports X.509 certificates containing RSA keys and signed using RSA\_MD5 or RSA\_SHA. X.509v3 extensions are handled correctly
- KeyStore
  - can only store certificates (no private keys or symmetric keys).

# Crypto J2ME Supported Features (SSL)

---

- Client-side implementation of SSLv3.0.
  - Supports session reuse. No support for v2.0 or v3.1 (aka TLS1.0).
  - Only supports two cipher suites - RSA\_RC4\_128\_MD5 and RSA\_RC4\_40\_MD5 (most common and fast).
  - No client-side authentication (rarely used).
  - Server authenticated through RSA signatures. No restrictions on cert chain lengths (other than those from basicConstraints).
- Maintains an extensible set of trusted Certification Authorities covering > 95% of SSL server certificates [Netcraft].

# Exemple

---

```
public class MyApp {
    public static void main(String[] args) {
        MyApp myapp;
        ...
        StreamConnection sc = (StreamConnection)
            Connector.open("sslsocket://xyz.com:443" +
                ";CertificateErrorHandling=warn" +
                ";HandshakeCommentary=on");
        InputStream in = sc.openInputStream();
        OutputStream out=sc.openOutputStream();
        // send data by writing to out, receive by
        // reading from in
        ...
        in.close(); out.close(); sc.close();
    }
}
```

# Exemple

---

```
public class MyApp implements HandshakeListener {
    public void SSLHandshakeUpdate(String s) {...}
        public boolean SSLCertificateOk(X509Certificate c, byte error) {...}
    public static void main(String[] args) {
        MyApp myapp;
        ...
        StreamConnection sc = new
            SSLStreamConnection("xyz.com", 443, myapp);
        InputStream in = sc.openInputStream();
        OutputStream out=sc.openOutputStream();
        // send data by writing to out, receive by
        // reading from in
        ...
        in.close(); out.close(); sc.close();
    }
}
```



# Exemple

---

```
byte[] mod = ...;
byte[] exp = ...;
byte[] data = ...;
RSAPrivateKey priv = (RSAPrivateKey)
KeyBuilder.buildKey(KeyBuilder.TYPE_RSA_PUBLIC, (short) 1024, false);
priv.setModulus(mod, 0, mod.length);
priv.setExponent(exp, 0, exp.length);
Signature sig = Signature.getInstance(Signature.ALG_RSA_SHA_PKCS1, false);
byte[] res = new byte[sig.getLength()];
sig.init(priv, Signature.MODE_SIGN);
sig.sign(data, (short) 0, (short) data.length, res, (short) 0);
// send data along with signature in res
...
```

# Performance (Palm)

## ■ Mémoire

- `com.sun.ksecurity + com.sun.kssl = 87KB`
  - en partie substituable par une carte

## ■ Temps

- RC4 (100 Kbits/s), MD5 (63 Kbits/s), SHA (57 Kbits/s).
- SSL 37Kbits/s.
  - 20 sec for full handshake when client parses & verifies server certificate
  - (b) cert caching reduces this to ~15 sec
  - (c) abbreviated handshake takes 5 sec
  - (d) 0 with persistent HTTP
    - Offloading RSA to smartcard can reduce (a) (b) by 30-50% (1024-bit RSA operations on new Cyberflex take 0.5s independent of exponent size v/s 4-7s for small exponents)

# Performances

---

- Benchmarking
  - [http://www.javaworld.com/javaworld/jw-10-2002/jw-1025-j2mebenchmark\\_p.html](http://www.javaworld.com/javaworld/jw-10-2002/jw-1025-j2mebenchmark_p.html)
- GrinderBench <http://www.grinderbench.com>
  - Mix of Chess, Crypto, kXML, Parallel, PNG

# SyncML (<http://www.syncml.org>)

- Message XML comportant des éléments décrivant une synchronisation
  - Add, Alert, Atomic, Copy, Delete, Exec, Get, Map, Replace, Search, Sequence, Sync
- 2 représentations
  - texte      WBXML                      `text/vnd.syncml+xml`
  - bytecode                                      `application/vnd.syncml-wbxml`
- Transport
  - HTTP, WSP, ..., JMS
- API pour CDLC
  - [http://www.kvmworld.com/articles/perspective/syncml?content\\_id=1348](http://www.kvmworld.com/articles/perspective/syncml?content_id=1348)
  - Chap12 du [Feng]
  - <http://ksync.enhydra.org>

# Outils J2ME

---

- Javac
  - Celui du JAE : produit du bytecode JVM
- Preverify
  - Prévérification du bytecode (spécification compatibilité KVM)
- JavaCheck
  - Vérification de la compatibilité des paquetages utilisés par une application
- ROMisation

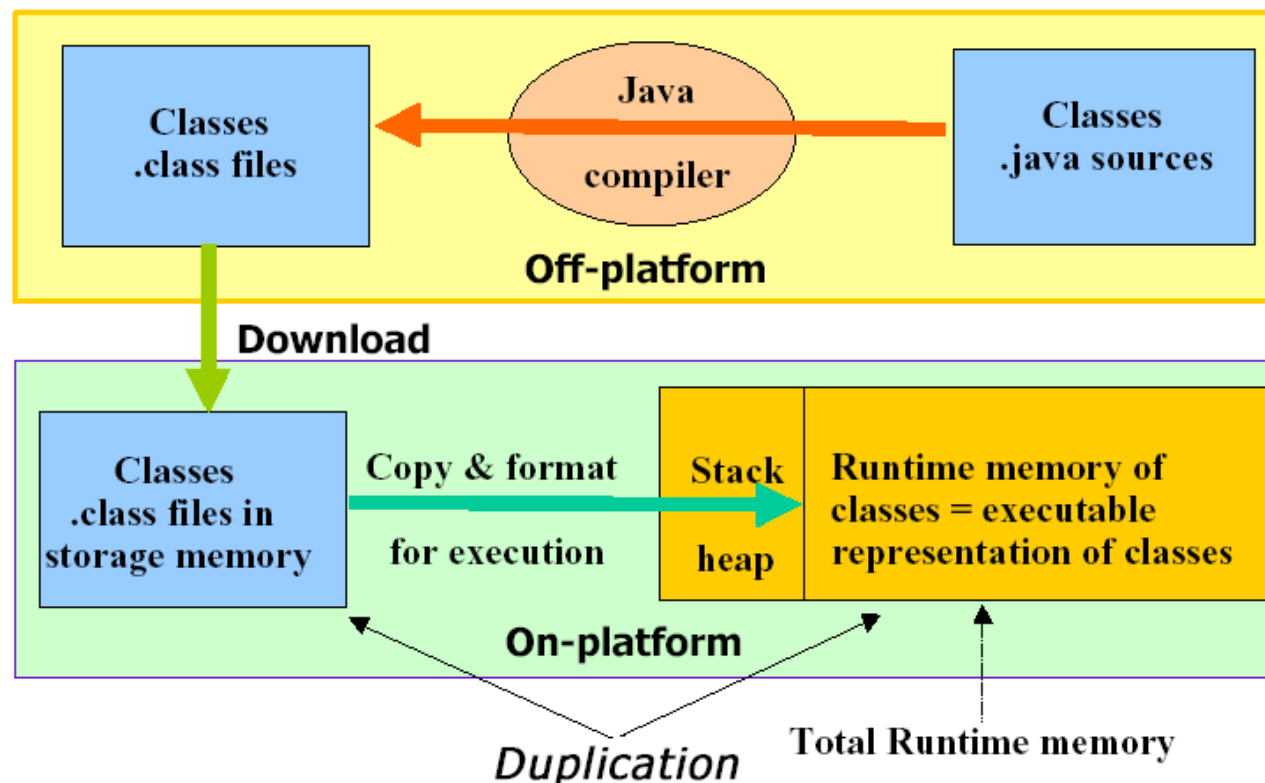
# Outils J2ME : JavaCheck

---

- Analyse de package utilisés par une application
  - Détermine s'ils sont présents ou optionnels dans le profile
  
- En entrée
  - Un fichier de description de la plate-forme
    - platform specification (.spc)
  - Une application, applet, ...
  
- *Platform specifications* (courantes)
  - Pour l'instant pour Personal Java

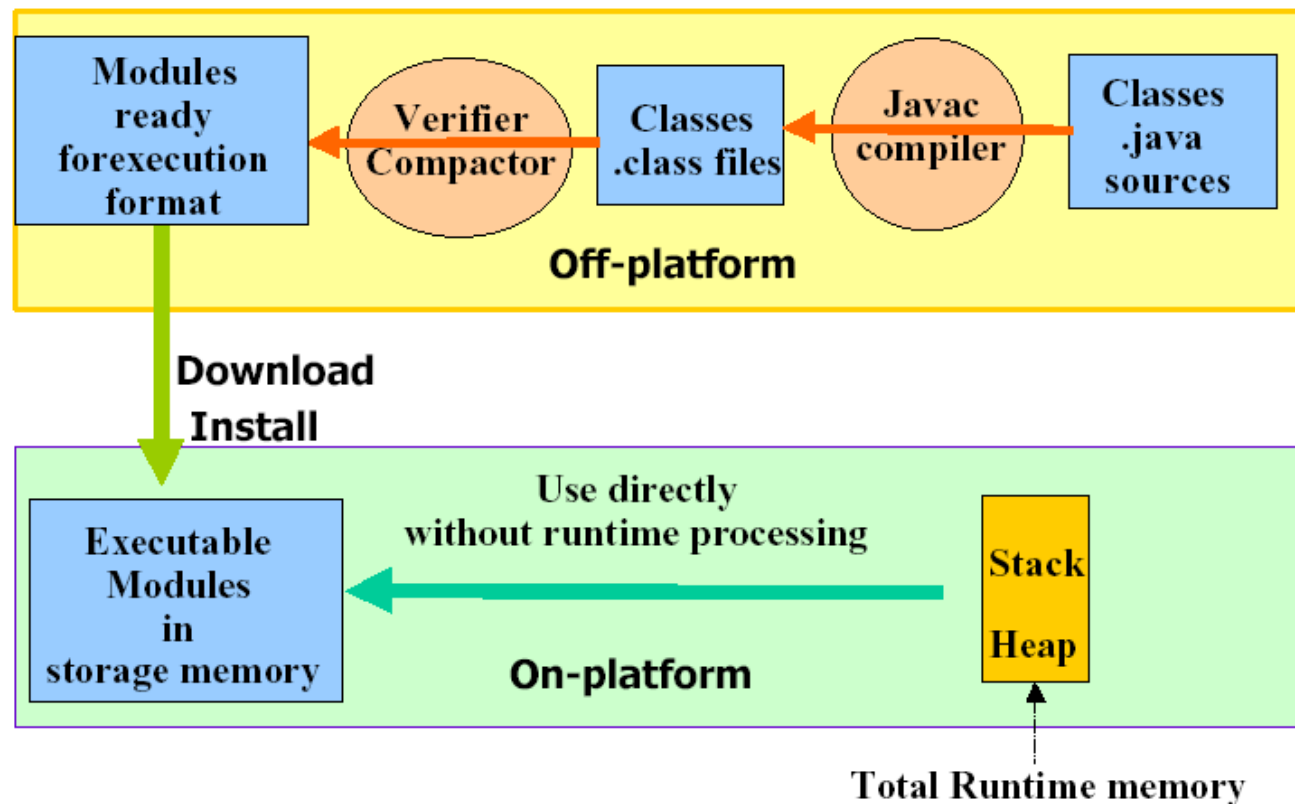
# Outils J2ME : ROMisation (i)

- Problème du format standard .jar de .class
  - Duplication en RAM et en FlashRAM/ROM
  - CPU pour effectuer la conversion, Démarrage ralenti, ...



## Outils J2ME : ROMisation (ii)

- One level format (JEFF, JavaCard CAP, TINI, ...)
  - économise RAM et FlashRAM/ROM
  - démarrage « immédiat »





## Outils J2ME : ROMisation (iii)

- JavaCard CAP File Format
  - Les symboles (constant pool) sont pré-résolés (édition de lien) avant installation
    - Plus d'édition de lien dynamique possible
- JEFF (ISO/IEC 20970) (03/2002)
  - <http://www.j-consortium.org/jeffwg/index.shtml>
  - optimiser le stockage des classes (et ressources) en FlashRAM/ROM pour minimiser l'usage de la RAM
    - contrairement au .jar de .class qui replit
  - Principe
    - Constant pool commun
    - Pré-résolution locale dans le fichier JEFF
    - Tables d'indirection vers les symboles des autres JEFF pour édition de lien dynamique et les « patches »
- TINI File Format
  - Format propriétaire pré-résolu

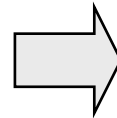
# Outils J2ME : ROMisation (iv)

- 2 outils SUN
  - JavaCodeCompact
    - Pour la ROMisation des classes Java (Embedded Java)
    - Produit des .c qui compilés (.o) sont liés à la VM s'exécutant sur un RTOS
      - Fichier .c indépendant de la plate-forme
  - JavaDataCompact (Embedded Java)
    - Pour la ROMisation des ressources (HTML, Images, ...)
    - Produit des .c qui compile (.o) sont lié à la VM s'exécutant sur un RTOS
      - Fichier .c indépendant de la plateforme
- ROMisation finale (indépendant des outils Java)
  - Utilise le compilateur C, l'assembleur et l'éditeur de lien du RTOS
  - Edition de lien finale :
    - bytecode Java, fichier de données,
    - Objet .o (.c compilé) et implantation native,
    - Objet (.o) de la VM et objet (.o) du RTOS

# Retro-compileur

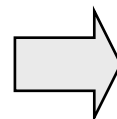
- Motivation
  - La KVM et la plupart des VM embarquées ne supportent pas les nouvelles fonctionnalités de la plateforme 5.0+
    - **generics, extended for loops, static imports, autoboxing/unboxing, varargs, enumerations, annotations**
- Outils de conversion 5.0 vers 1.4, 1.3 ...
  - Retroweaver <http://retroweaver.sourceforge.net/>
- Exemples

```
public void foo( String... ) {  
}
```



```
public void foo( String[] ) {  
}
```

```
public class Foo<T extends Comparable> {  
    public void foo( T t ) { ... }  
}
```



```
public class Foo {  
    public void foo( Comparable t ) { ... }  
}
```

# SavaJe XE



<http://www.savaje.com>

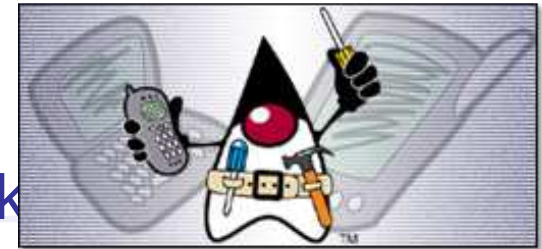
- Système d'exploitation Java pour PDA Haut de Gamme
  - 32 bits et multitaches
  - En remplacement de WinCE et PalmOS
- JVM optimisée
  - Classes du J2SE
  - Applications
    - MP3 media player, Espial's web browser, Ebox email client
    - Assistant PIM
- Cibles
  - Compaq Ipaq, Psion netBook
  - Intel StrongARM et Xscale >200Mhz
  - = 12MB ROM (sur une carte FlashRAM pour le Psion)
  - >32MB DRAM

# Environnements de développement

- Intègrent les outils et des émulateurs
- J2ME Wireless Toolkit
  - Plusieurs cible (dont Palm jusqu'à la version 1.0.4)
  - Contient le prévérifieur KVM, un profileur d'application, un moniteur de trafic réseau, un simulateur de performance et l'ofuscateur RetroGuard, signature de JAR et gestionnaire de certificats X509
  - Intégrable à FORTE
  - <http://java.sun.com/products/j2mewtoolkit/>
- VisualCafe, JBuilder, CodeWarrior...
- IBM WebSphere Device Dev. (J2ME & J9) ex VisualAge
- WabaSDK
- Nokia, Motorola (MERI)
- Docomo NTT : Emulateur DoJa pour les i-αpplis
- ...
- Voir <http://wireless.java.sun.com/midp/articles/emulators/>

# J2ME Wireless Toolkit

<http://java.sun.com/products/j2mewtoolkit>



The screenshot shows the J2ME Wireless Toolkit interface. At the top is a menu bar with 'File', 'Edit', 'Project', and 'Help'. Below it are buttons for 'New Project...', 'Open Project...', 'Settings...', 'Build', and 'Run'. A 'Device:' dropdown menu is set to 'DefaultColorPhone'. The main area is divided into three panes. The left pane is a code editor showing GC (Garbage Collection) logs. The middle pane shows a blue mobile phone emulator with a screen displaying 'Apple 2 tix \$37.00', 'Auction Menu', and 'Ticket Info'. The right pane shows a grey mobile phone emulator with a screen displaying '\$115.00 Fiona App', 'Auction Menu', 'Ticket Info', 'Make a Bid', 'Set an Alert', and 'Settings'.

# Autres Environnements de développement

---

- Pré-résolution des « tables de symboles »
  - Diminue la durée du startup
  - Vérifie la fermeture des dépendances de classes à partir d'une racine
  - Cas de TINI, de Lejos, JITS ...
- Compilation du source/bytecode Java en code natif (*ou C*)
  - Avant installation dans la ROM
  - Ahead-Of-Time Compiler
  - éventuellement sélective
    - Certaines classes (les moins cruciales en perf) restent interprétées
  - Le RE doit être aussi compilé
  - Produits: TurboJ, GNU GCJ ...
- Compilation Flash
  - Serveur de compilation avant chargement
  - Produits: Silicomp Flash Compiler, ...

# Environnements de développement

## *Divers*

- Construction
  - Ant tasks
    - Antenna, Antic, J2ME Polish, ...
      - <http://ant.apache.org/external.html>
  - Maven J2ME plugin and archetype
    - <http://snapshot.pyx4me.com/pyx4me-maven-plugins/j2me-maven-plugin/>
    - <http://pyx4me.com/pyx4me-archetypes/j2me-simple>
    - <http://mojo.codehaus.org/j2me-maven-plugin/howto.html>
- Réduction du Jar
  - MiniJar
    - <http://mojo.codehaus.org/minijar-maven-plugin/index.html>
- Ofuscateur
  - ProGuard
    - <http://snapshot.pyx4me.com/pyx4me-maven-plugins/proguard-maven-plugin/>
- Test unitaire
  - J2MEUnit
- Emulateur
  - MicroEmulator <http://microemu.org/>
    - Exécutable depuis JWS (.jnlp)
- Autres
  - Opera mini
    - <http://www.operamini.com/demo/>







# La concurrence (MicroSoft)

- .NET Compact Framework
  - Version allégé de .NET FrameWork
  - Cible les profiles CDC et CDLC
  - fonctionnement offline
    - Exemple: cache de ligne SQL (ADO)
  - RTE
    - Core CLI, réseau, XML, Web Services, ADO .NET
    - Garbage collector : simple Mark and Sweep
    - JIT MSIL→Natif
      - au premier appel
      - Cache de code JIT
  - Environnement matériel
    - StrongARM, MIPS, x86, SH4, Xscale, ...
    - Windows CE
    - RAM : 1.5 Mo minimum

# Quelques Recommandations pour un code Java performant et compact

---

# Quelques Recommandations pour un code Java performant et compact

---

- 1) produisez un code de qualité
  - Critère qualité : maintenabilité, testabilité, ...
  
- Si vous avez des problèmes de performance
  
- 2) Analysez (*profilez*) votre code pour identifier les goulots d'étranglement (bottlenecks)
  
- 3) Optimisez les goulots en conservant la qualité

# Quelques Recommandations pour un code Java performant et compact

- Construire de petites applications
  - Ofusquer le source
    - Méthodes privées
    - Compactage des identifiants de méthodes, ...
      - <http://www.alphaworks.ibm.com/tech/JAX>
      - <http://www.retrologic.com/retroguard-main.html>
    - « Dégraisser » les jarfiles des classes non chargées
      - Outils calculant la fermeture transitive du graphe de chargement
  - JAR non compressé pour accélérer le démarrage
    - Décompression
    - Mapping mémoire virtuelle ou Flash ou ROM
  - Simplifier votre application
  - Internationalisation ou Profile à l'installation
    - Une version de l'application (JAR) par langue, pays, ...
    - Précompilation des `#ifdef` avec `cpp` (ou ses versions Java)

# Économie de mémoire à l'exécution

- Utilisez les types scalaires plutôt que des objets  
void setSize(int width, int height); **plutôt que** void setSize(Dimension size);
- Aidez le Ramasse-Miette
  - Mettre les références qui ne seront plus utilisées à null
- Réutilisez les objets instanciés
  - Pool d'objets, ...
- Prévoyez la capacité initiale des collections
  - constructeur Vector(int initialCapacity, int capacityIncrement)
- Instanciez à la demande (lazy ou on-demand)

```
public class LazyClass {  
    private Vector v;  
    public Vector getVector() { if(v==null) v=new Vector(); return v; }  
}
```

# Economie de mémoire à l'exécution

- Evitez la récursion
  - Élégant mais lent et gourmand en pile
  - Dérécursivez quand c'est possible
- Evitez les exceptions
  - Car les exceptions sont des objets et leur déclenchement augmente la taille du code

# Code performant

- Utilisez des variables locales

```
for(i=0;i<buf.length;i++) { if( buf[i]>='0' && buf[i]<='9' && ...) }
for(i=0;i<buf.length;i++) {Char c=buf[i]; if( ch>='0' && ch<='9' && ...) }
```

- Evitez la concaténation de chaînes et utilisez plutôt StringBuffer

- Utilisez les threads mais limitez la synchronisation

```
public synchronized Vector getVector() { // peu performant
    if(v==null) v=new Vector();
    return v; }
public Vector getVector() { // synchronisation circonscrite
    if(v==null) { synchronized(this) { if(v==null) v=new Vector();} }
    return v; }
```

# Optimisation sur les boucles

## static int charCounter(String s, char ch)

- La boucle avec charAt 8564ms sur TINI

```
for (int i = 0; i < s.length(); i++) {  
    if (s.charAt(i) == ch) { ++count; }  
}
```

- Cacher la longueur 5997ms

```
int len = s.length();  
for (int i = 0; i < len; i++) {  
    if (s.charAt(i) == ch) { ++count; }  
}
```



# Optimisation sur les boucles

## static int charCounter(String s, char ch)

- Accès Direct 3248ms

```
int len = s.length();
char[] ca = new char[len];
s.getChars(0, len, ca, 0);
for (int i = 0; i < len; i++) {
    if (ca[i] == ch) { ++count; }
}
```

- Copie locale 2877ms

```
int len = s.length();
char[] ca = new char[len]; // Get a local copy of char[]
s.getChars(0, len, ca, 0);
while (len > 0) {
    if (ca[--len] == ch) { ++count; }
}
```

# Optimisation sur les boucles

```
static int charCounter(String s, char ch)
```

- IndexOf 630ms

```
int count = 0; int index = 0;
while (true) {
    index = s.indexOf(ch, index);
    if (index != -1) { ++count; ++index;
    } else { break; }
}
```

# Graphisme

---

- Ne chargez que les images dont vous avez besoin
  - Utilisez Image createImage(String ressource) pour MIDP
  - Utilisez MediaTraker pour Personal
- Adaptez la taille/nb couleurs des images
- Utilisez le clipping et les Image Buffer pour les animations
  - Evite le scintillement (flicking)

## Utilisez des outils d'analyse

---

- Deboggage
- Profilage
- Analyse de la pile d'appel et du tas
  - Fuite mémoire, ...
  
- Sur des émulateurs
- Puis sur de vrais terminaux et de vrais réseaux
  - KDWP (K-Java Debug Wire Protocol)
    - Sur USB, WiFi, Bluetooth, RS232

# Exemple d'analyseurs

Methods' Profiler - Wireless Toolkit

File View

Save Open

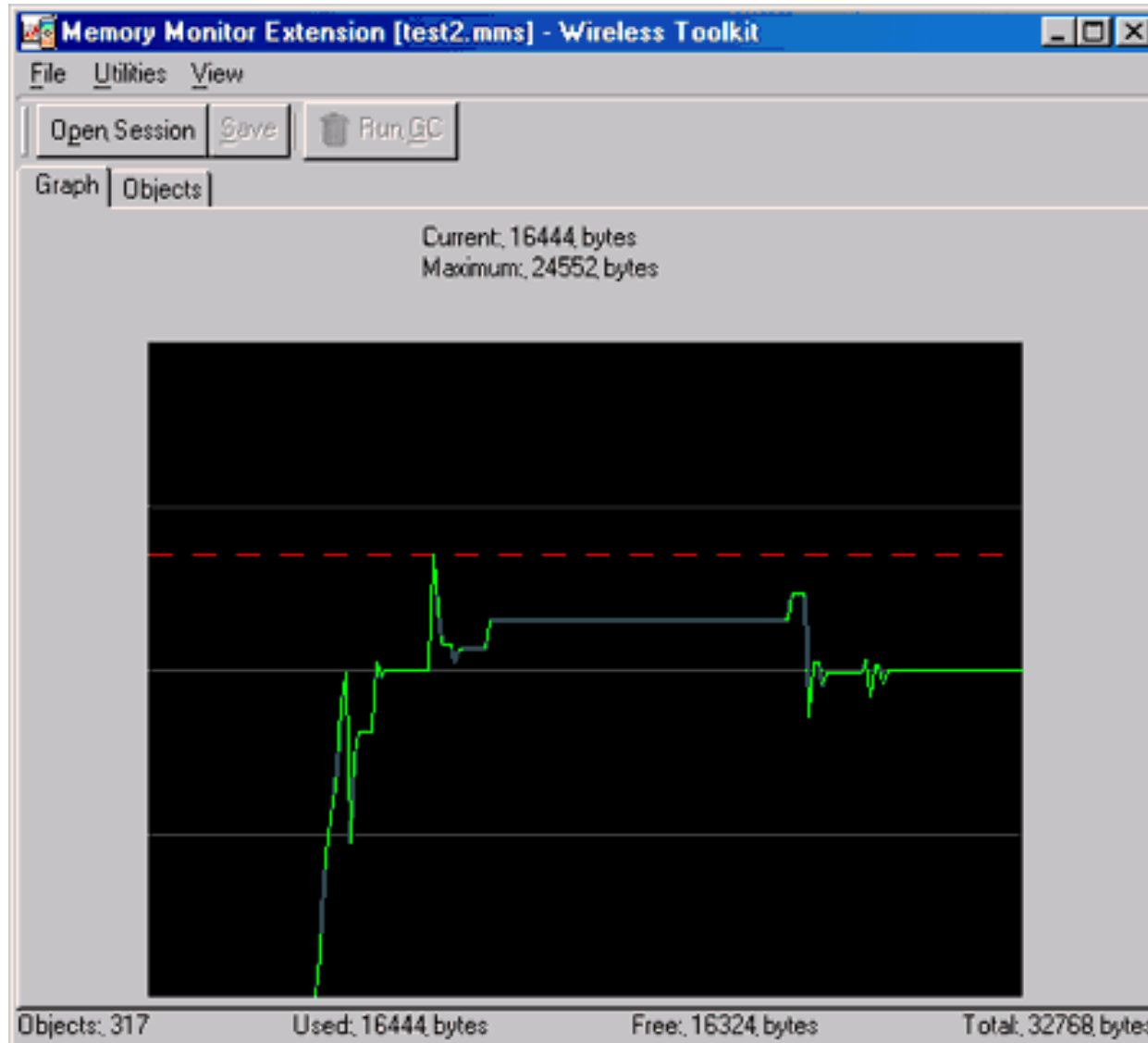
Call Graph

ALL calls, under, <root>

Name	Count	Cycles	%Cyc...	Cycles With Children	%Cycles With Children
com.sun.midp.io.SystemOutputStream.putchar	1259	5347272	6.8	5347272	6.8
java.io.DataInputStream.readUTF	376	5275874	6.7	12852893	16.3
java.lang.Class.forName	10	3802703	4.8	3802703	4.8
com.sun.midp.io.j2me.storage.RandomAccessSt...	25	3733595	4.7	3733595	4.7
java.lang.StringBuffer.append	1276	3583082	4.5	3583082	4.5
java.lang.StringBuffer.<init>	347	2935018	3.7	3179020	4
java.io.Reader.read	168	2933933	3.7	6673682	8.5
java.io.OutputStream.write	172	2383143	3	8825407	11.2
javax.microedition.lcdui.Font.charWidth	343	1888244	2.4	1888244	2.4
javax.microedition.lcdui.ImmutableImage.decod...	5	1866114	2.3	1866114	2.3
com.sun.kvm.cldc.i18n.j2me.GenericWriter.ch...	78	1740068	2.2	1740068	2.2
java.lang.String.<init>	222	1657291	2.1	2295716	2.9
EnumDemo\$EnumList.compare	39	1384064	1.7	30364857	38.7

Find

# Exemple d'analyseurs



## Bibliographie - Livre

---

- Michael Juntao Yuan , Enterprise J2ME: Developing Mobile Java Applications, Ed Prentice Hall PTR ISBN: 0131405306; Oct 2003,
  - <http://www.enterprisej2me.com/pages/enterprisej2me/code.php>
  - De très nombreux exemples
- Harvey M. Deitel, Paul J. Deitel, Tem R. Nieto, Kate Steinbuhler, "Wireless Internet and Mobile Business How to Program, 1/e", 2002, 1000 pp;, Prentice Hall, ISBN 0-13-009288-6,  
<http://vig.prenhall.com/catalog/academic/product/1,4096,0130092886,00.html>
- Voir la bibliographie de « Programmation PalmOS »

## Bibliographie -Livre

---

- Eric Giguère, Java 2 Micro Edition, Ed Wiley, 2000, ISBN 0-471-39065-8
- Bruno Delb, J2ME, Applications Java pour terminaux mobiles, Ed Eyrolles, 06/2002, ISBN : 2-212-11084-7
- Yu Feng, Dr. Jun Zhu, Wireless Java Programming with Java 2 Micro, 1st edition (May 24, 2001), Ed Sams; ISBN: 0672321351
- Wong, Java 2 Micro Edition, Ed Addison Wesley. ISBN 0-201-70244-4 (06/2001)
- Qusay Mahmoud , Learning Wireless Java, Ed Oreilly, December 2001 , ISBN 0-59600-243-2
- Jonathan Allin, Wireless Java for Symbian Devices, [www.symbian.org/books](http://www.symbian.org/books)
- Barbara Ballard, User Interface Guidelines for J2ME MIDP 2.0, Publisher: Little Springs Design, Inc. (December 2003), ISBN: 1-4116-2429-7



# Bibliographie

---

- J2ME Tips

- <http://developer.java.sun.com/developer/J2METechTips/index.html>
- <http://wireless.java.sun.com/midp/articles/>

- Présentations JavaOne

- <http://servlet.java.sun.com/javaone/sf2002/conf/sessions/18-all-regular.en-36836.jsp>

# Sites

---

- Un FAQ
  - <http://www.kvmworld.com/developer/faq/>
- Des sources
  - <http://www.kvmworld.com/downloads>
- Des articles
  - <http://www.kvmworld.com/devices/europe>
- Des Devices MIDP
  - <http://www.kvmworld.com/devices/europe>
- jGuru J2ME FAQ:
  - <http://www.jguru.com/faq/J2ME>
- J2ME Archive:
  - <http://www.billday.com/j2me>
- JAD Central
  - <http://www.jadcentral.com>
- Des utilitaires
  - <http://kobjects.org/>
- Un base de *devices* CLDC/MIDP
  - <http://devicedb.kobjects.org/index.html>

# Sites

---

- JAD Central
  - <http://www.jadcentral.com>
- Des utilitaires
  - <http://kobjects.org/>
- Midlets à tester
  - <http://midlet.org>
  - <http://www.spruce.jp/freemidlets/>
  - <http://www.j2mepolish.org/>

# Sites

---

- Nokia J2ME toolkits
  - <http://www.nokia.com/java/>
  - <http://www.forum.nokia.com>
- IBM WebSphere Device Dev
  - <http://www.embedded.oti.com>
- News J2ME
  - <http://www.blueboard.com/j2me/>
- Applications
  - <http://www.billday.com/j2me/index.html#Applications>

# Exercice

---

- Installez le J2ME Wireless Toolkit
  - <http://wireless.java.sun.com/midp/articles/wtoolkit/>
- Testez les exemples fournis
- Testez les exemples de Spruce
  - <http://www.spruce.jp/freemidlets/>
- Testez les exemples de J2ME Tips
  - <http://developer.java.sun.com/developer/J2METechTips/index.html>
  - <http://wireless.java.sun.com/midp/articles/>