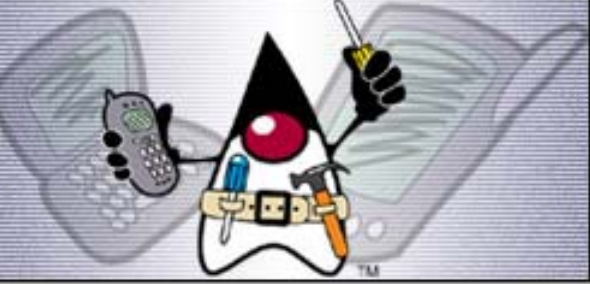


J2ME - Il micro mondo Java

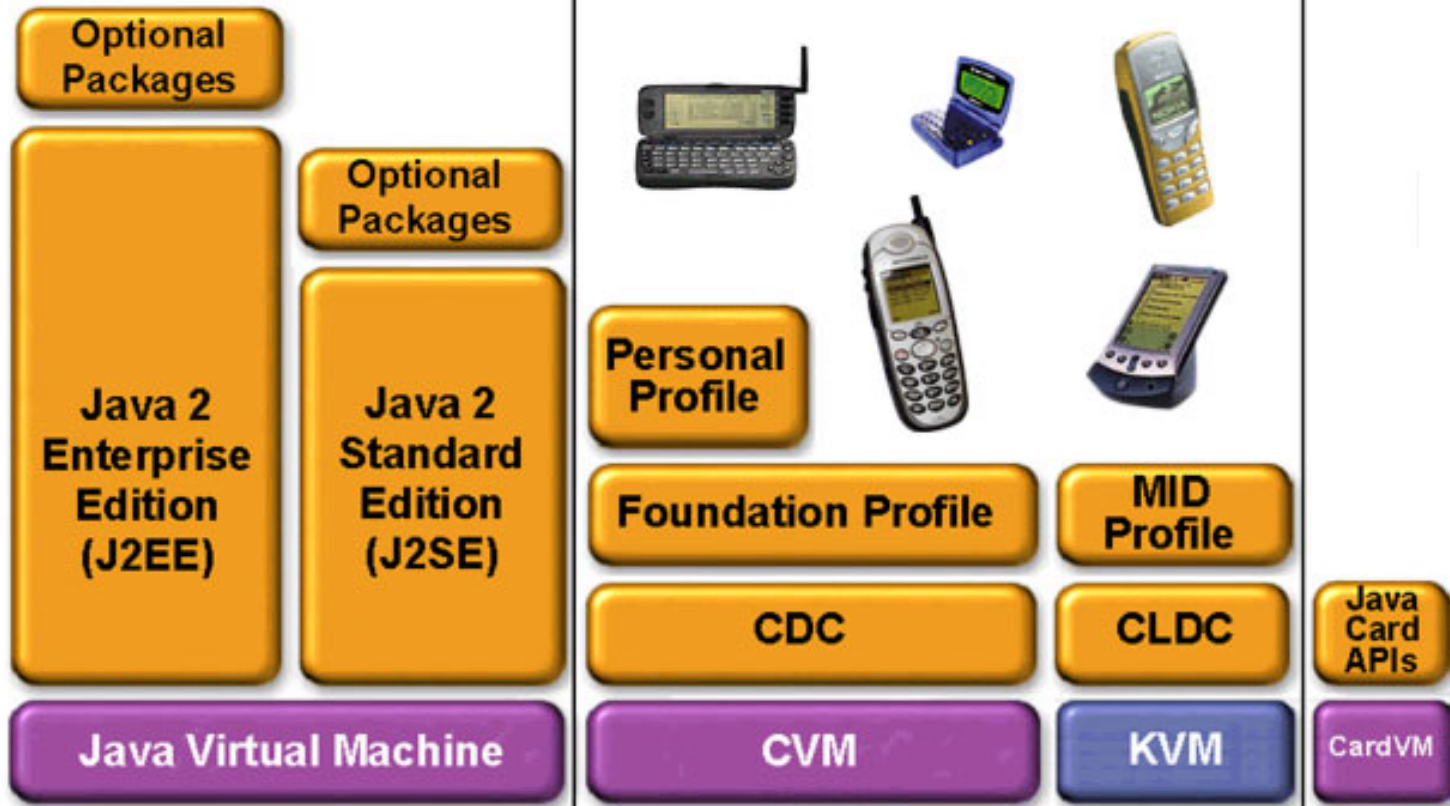


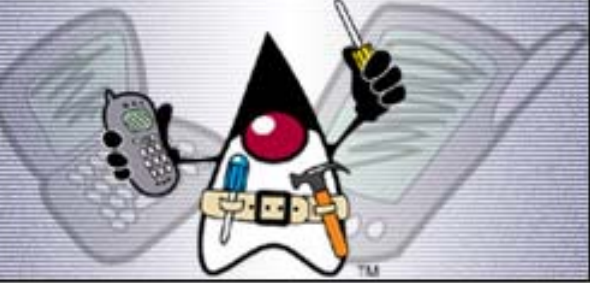
Java miniaturizzato per i dispositivi wireless



Le versioni di Java

Java™ 2 Platform, Micro Edition





Le configurazioni

Sun distingue ciò che si può tenere in mano

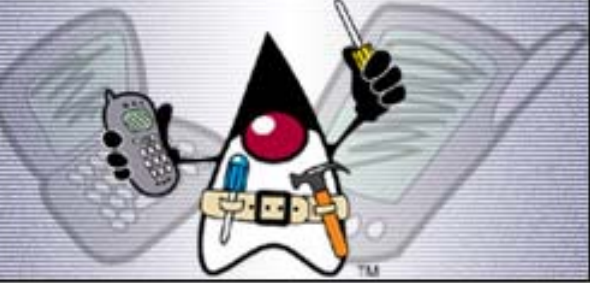
– CLDC

- 128 Kb di memoria per l'esecuzione di Java
- 32 Kb per l'allocazione a run time
- Interfaccia utente limitata
- Alimentazione elettrica ridotta (batterie)
- Connettività wireless, banda stretta, discontinua

Da ciò che si può connettere ad una presa murale

– CDC

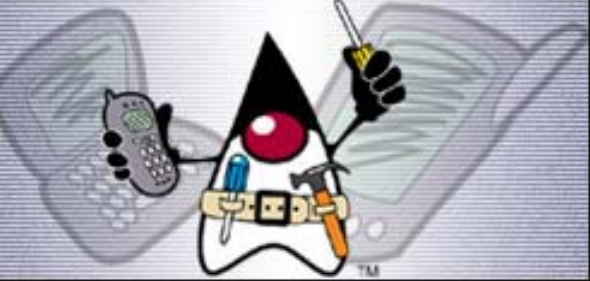
- 518 Kb minimo di memoria per l'esecuzione di Java
- 257 Kb per l'allocazione a run time
- Connettività a banda larga



KVM – Kylobyte Virtual Machine

Dietro ogni programma Java c'è una Virtual Machine, quella di J2ME ha queste caratteristiche:

- Necessita di soli 40-80 Kb di memoria
- Di soli 20 Kb di memoria dinamica
- Può essere eseguita su cpu con Clock di 25 MHz



La configurazione CLDC

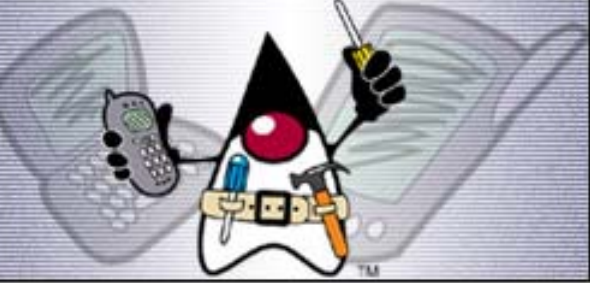
Destinata a Cellulari, PDA, Pager.

Si occupa di:

- Definire le caratteristiche del linguaggio Java
- Fornire un set minimo di librerie di base
- Gestire gli stream di I/O
- Sicurezza
- Networking
- Internazionalizzazione

Lasciando ai profili:

- La gestione del ciclo di vita delle applicazioni
- L'implementazione di interfacce utenti
- La cattura e la gestione degli eventi
- L'interazione tra l'utente e l'applicazione

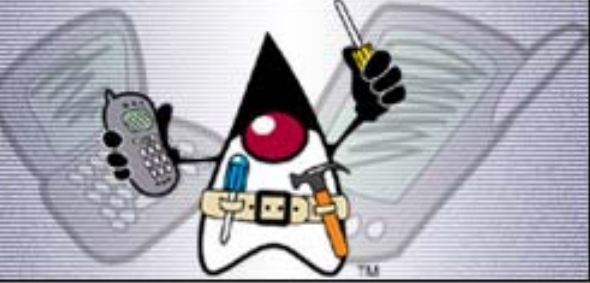


Il package java.lang

Differenze con l'omonimo package di J2SE

- Niente float
- Niente double
- Classe math: tenuti solo max, min, abs
- Niente Daemon Thread e Gruppi di thread
- Niente Java Native Interface

Per adattarsi meglio alle caratteristiche dei dispositivi

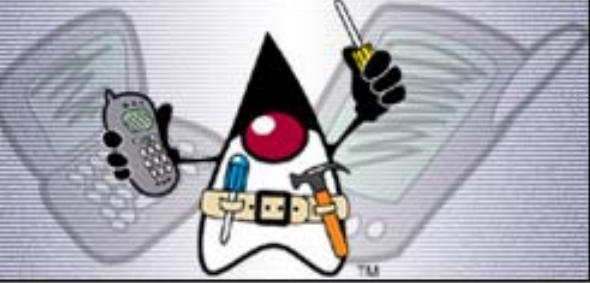


Classi di utilità: java.util

Uniche classi presenti:

- Enumeration
- HashTable
- Vector
- Stack
- Calendar
- Date
- TimeZone
- Random

Nella configurazione CDCL c'è solo lo
"stretto necessario"!



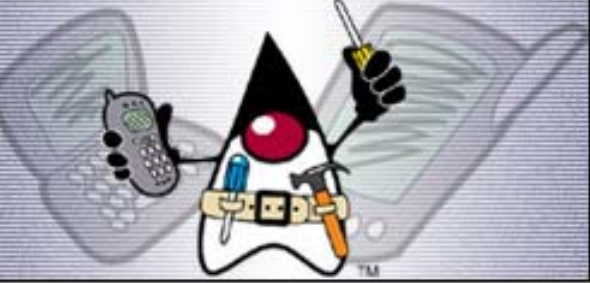
La gestione degli stream: java.io

La maggior parte dei dispositivi CDCL non possiedono un file system, sono state eliminate tutte le operazioni su file.

L'unico supporto all'internazionalizzazione è garantito da queste due classi:

- `InputStreamReader(InputStream is, String encoding);`
- `OutputStreamWriter(OutputStream os, String encoding);`

Che permettono la conversione da sequenze di byte a caratteri unicode e viceversa



Il Generic Connection Framework

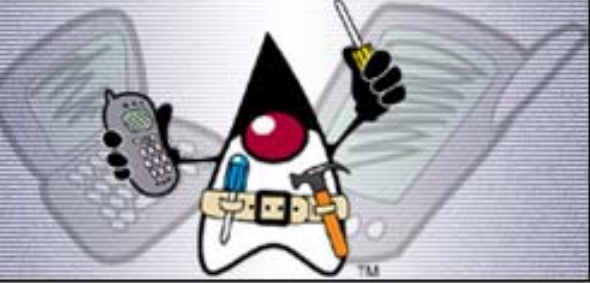
Qualunque tipo connessione viene aperta utilizzando il metodo `open()` della classe `Connector`:

```
Connector.open(string_connection);
```

dove `string_connection` ha il seguente formato:

```
<protocol> : <address> ; <parameters>
```

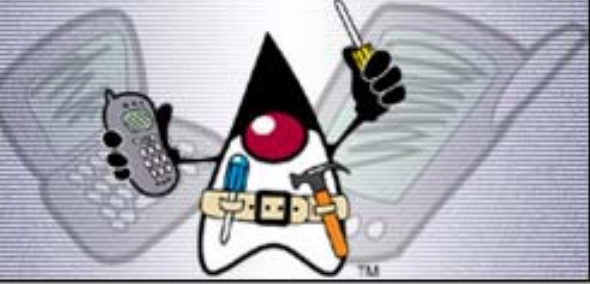
- `protocol` indica il tipo connessione che si intende utilizzare (es. `http`, `datagram`, scrittura su file, ecc.)
- `address` permette di individuare la destinazione (può essere un indirizzo IP ma anche il nome di un file)
- `parameters` contiene una serie di informazioni aggiuntive che possono essere funzionali per un dato tipo di connessione (ad es. baudrate per comunicazioni su porta seriale)



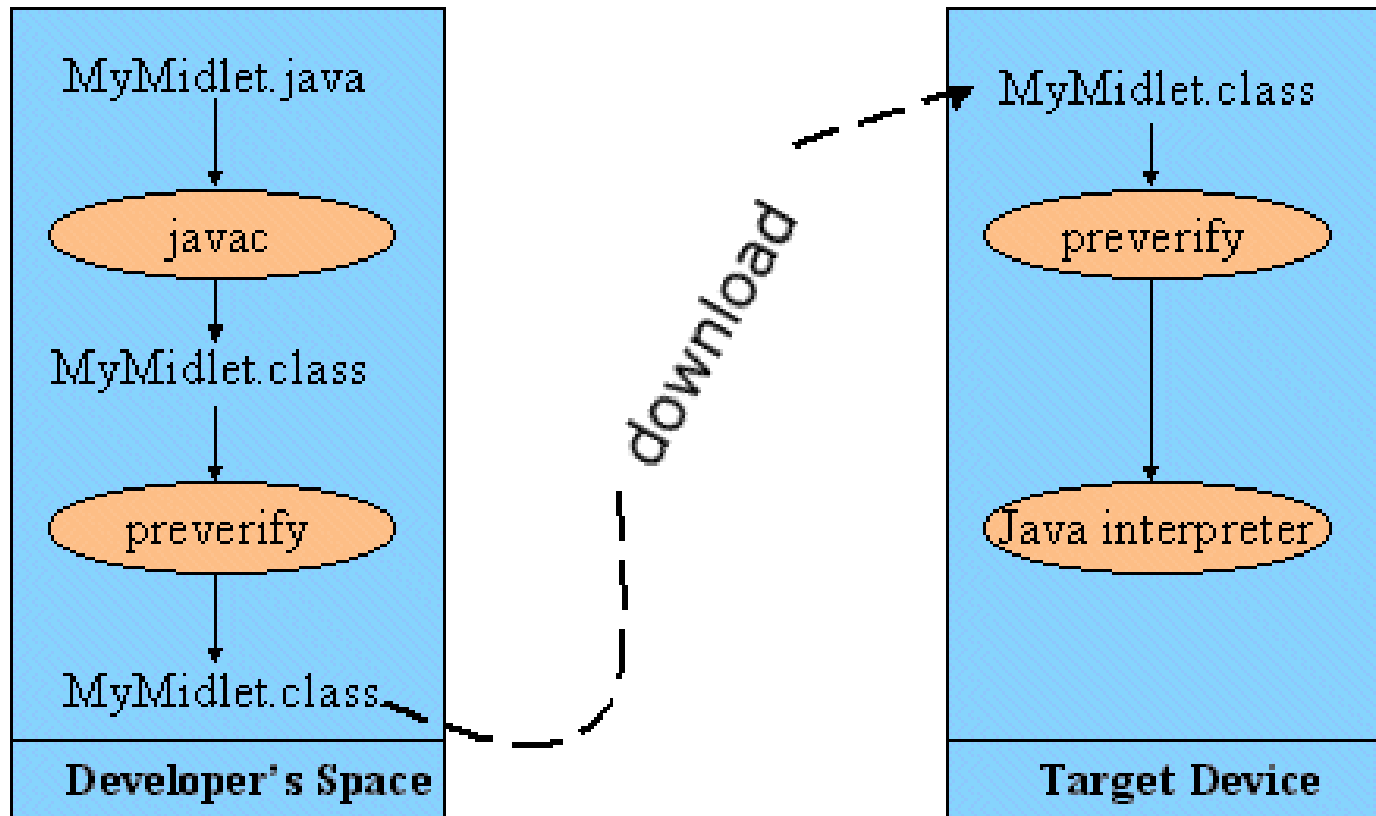
La sicurezza

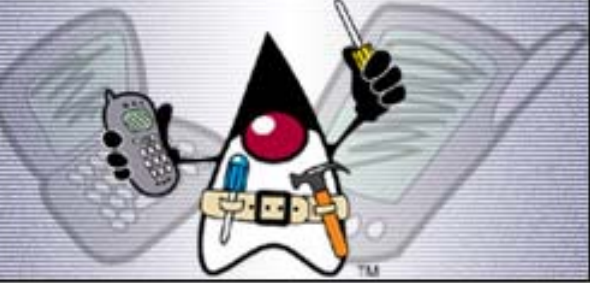
I meccanismi di J2SE richiedono troppa memoria, in J2ME è stato utilizzato un approccio a più livelli:

- Virtual Machine
- Applicazione
- Con due livelli di verifica:
 - Fuori dal dispositivo (off-device pre-verification)
 - Nel dispositivo (in-device verification)



Il processo di verifica

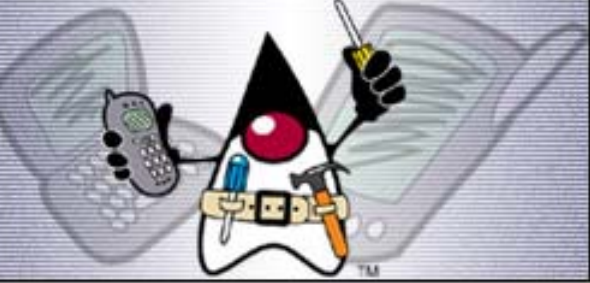




SandBox Model

Modello di sicurezza:

- Ogni applicazione deve aver superato il processo di verifica
- Ha accesso ad un ben determinato set di API (quelle previste dalla CLDC, dal profilo utilizzato e eventuali altri classi)
- Non è prevista la possibilità di ridefinire caricatori di classe a livello applicazione
- Il codice nativo non è accessibile alle applicazioni



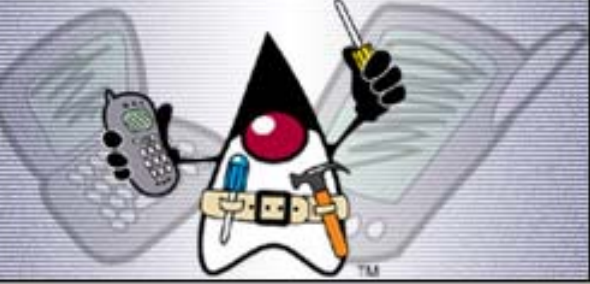
Mobile Information Device Profile – MIDP

Si occupa:

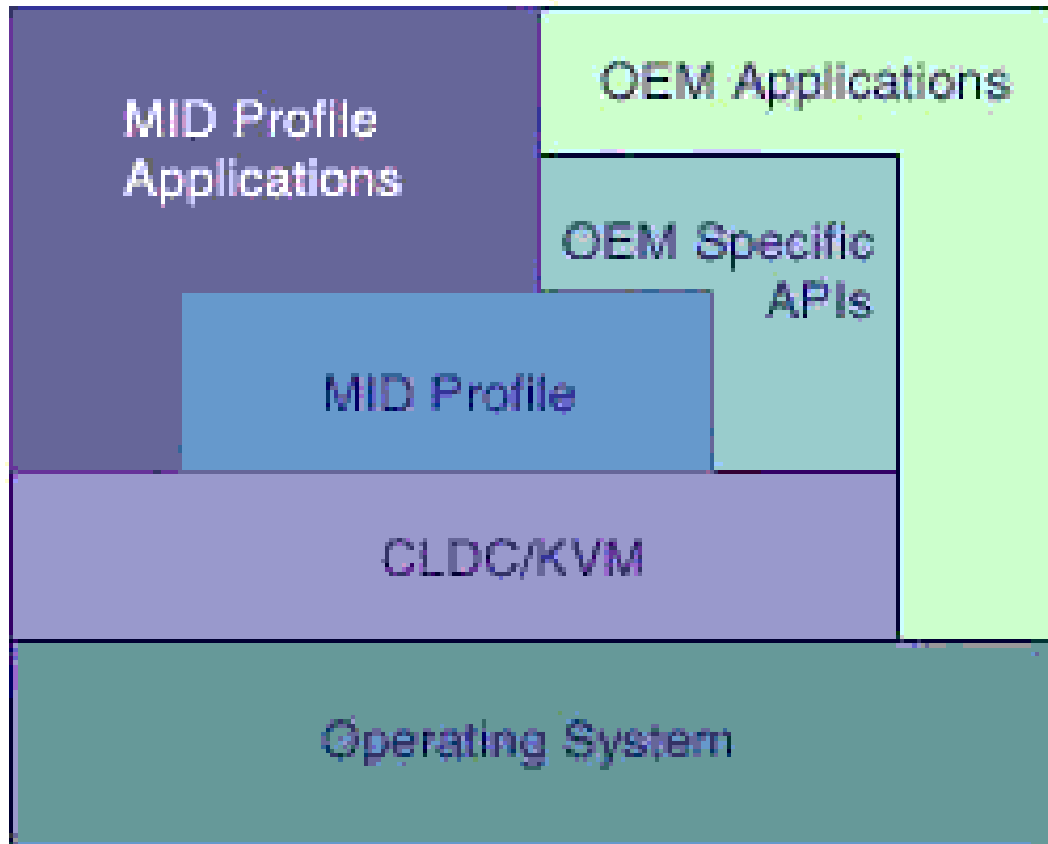
- Di gestire il ciclo di vita delle applicazioni (caricamento esecuzione - distruzione delle applicazioni)
- Di gestire l'interfaccia utente (dispositivi di input/output)
- Del salvataggio persistente dei dati
- Del networking (implementazione dei protocolli)

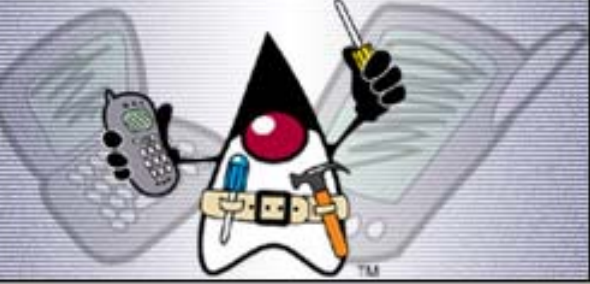
Questi quattro punti sono implementati in altrettanti packages, rispettivamente:

- `javax.microedition.midlet`
- `javax.microedition.lcdui`
- `javax.microedition.rms`
- `javax.microedition.io`

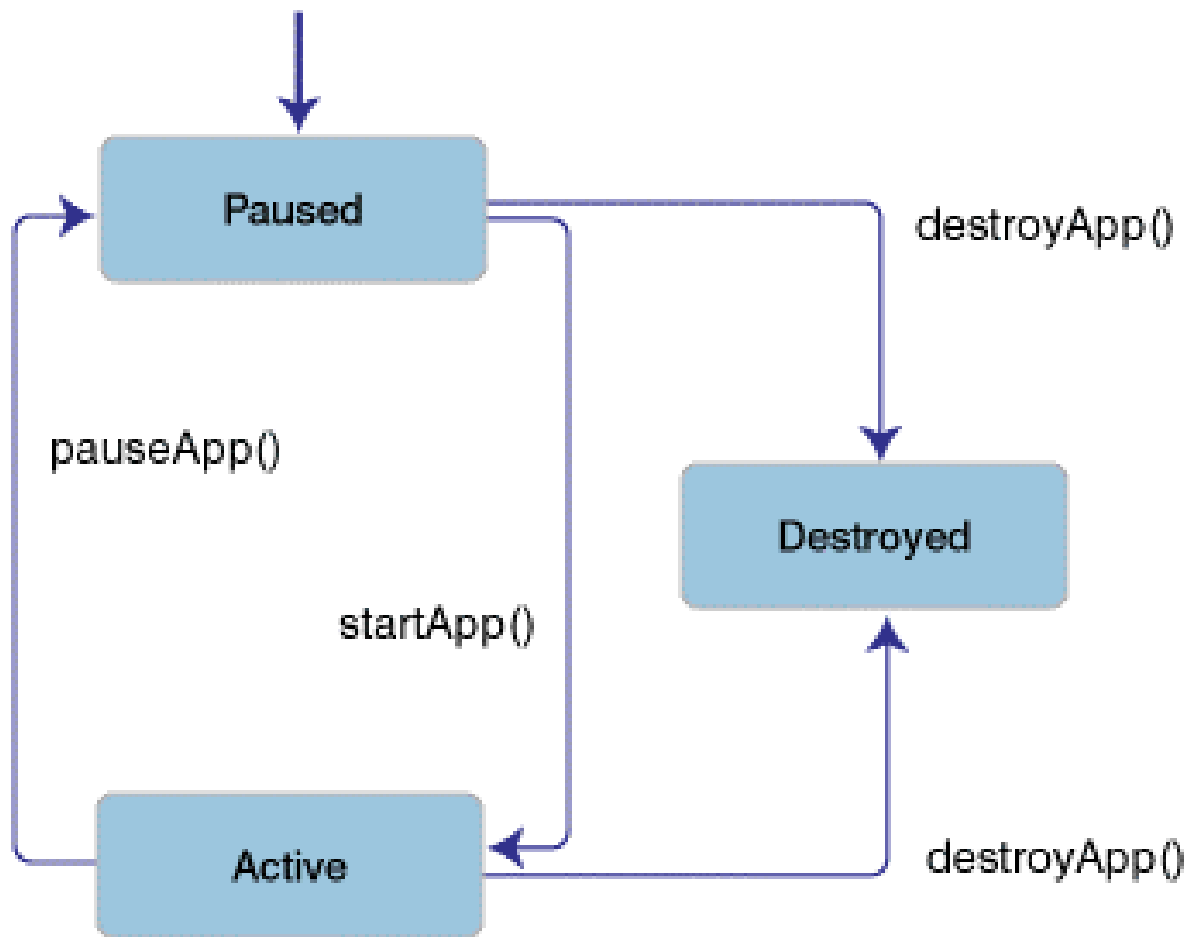


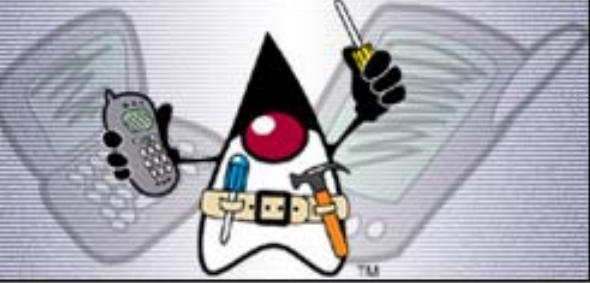
Architettura MIDP





Ciclo di vita di una MIDlet





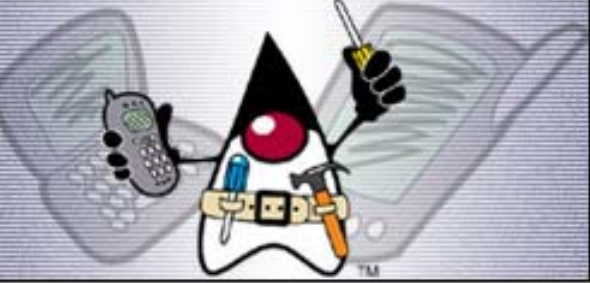
I file Manifest e Jad

File MANIFEST:

```
MIDlet-1: HelloWorld, HelloWorld.png
MIDlet-Name: HelloWorld
MIDlet-Vendor: java2me.org
MIDlet-Version: 1.1
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
```

File JAD: (Java Application Descriptor)

```
MIDlet-1: HelloWorld, HelloWorld.png
MIDlet-Jar-Size: 127
MIDlet-Jar-URL: HelloWorld.jar
MIDlet-Name: HelloWorld
MIDlet-Vendor: java2me.org
MIDlet-Version: 1.1
```

Metodi di approvvigionamento

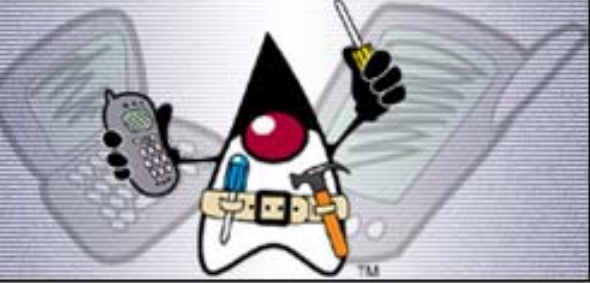
OTA

- Over The Air Initiated Provisioning Recommended Practice è un addendum del MIDP 1.0 per il download delle midlet.

In pratica nel file JAD viene indicato l'url del file JAR

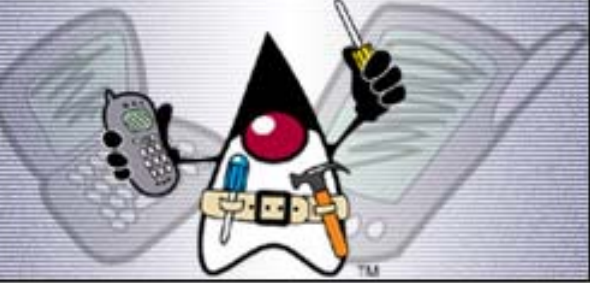
Wap Push

- Viene inviato al cellulare un messaggio con un link al file JAD



Il futuro: MIDP 2.0

- Multimedia
- Immagini in RGB
- Game API
- Modello di Sicurezza Migliorato
- HTTPS
- Gestione della rete migliorata
- Distribuzione delle applicazioni migliorata



Conclusioni

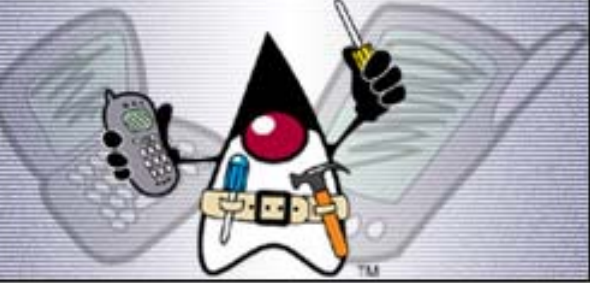
J2ME è una tecnologia molto potente e può essere integrata con le numerose API liberamente scaricabili da internet

- Jxta.org per il P2P
- Bouncycastle.org per la crittografia
- Ksoap, Kxml

Limitata dall'Hw degli attuali dispositivi

Con i cellulari 3G sarà possibile sviluppare applicazioni molto potenti

- Sistemi GIS
- Applicazioni Multimediali



Un esempio pratico...

Il progetto “Fluid~Time” utilizza

- Grafica di alto e basso livello
- Conessione HTTP
- Pianificazione dei compiti (Timer)
- HTTPS
- ... prossimamente
- Record Management System