



Esercitazione su J2ME: Come si Progetta Applicazione MIDlet

- ❑ Identificare i **dispositivi** target
- ❑ Identificare i **profili** supportati da quei dispositivi
- ❑ Sviluppare applicazione
 - Ad esempio, se in forma di MIDlet perché supportato CLDC/MIDP, applicazione adatta per cellulari e piccoli pager
- ❑ Richiede che l'ambiente di sviluppo sia integrato con ambiente CLDC/MIDP: MIDP reference implementation
<http://www.oracle.com/technetwork/java/index-jsp-138820.html>
- ❑ Possibilità di esecuzione per validazione e testing su **emulatori MIDP**
 - Ad esempio si può utilizzare J2ME Wireless Toolkit e **ktoolbar** per creare un progetto, progettare, implementare, configurare e validare la MIDlet -
<http://developers.sun.com/mobility/midp/articles/wtk21/>
 - Probabilmente sarete più comodi a utilizzare **Eclipse ME**
<http://eclipseme.org/>



Ad es. Nokia J2ME SDK

The screenshot shows the Eclipse IDE interface. The main editor window displays the code for 'HelloWorldMIDlet.java'. The code includes methods for pausing, destroying, and exiting the application. The left sidebar shows the project structure with 'HelloExamples.jpx' and its associated files. The bottom status bar shows the emulator command and connection status.

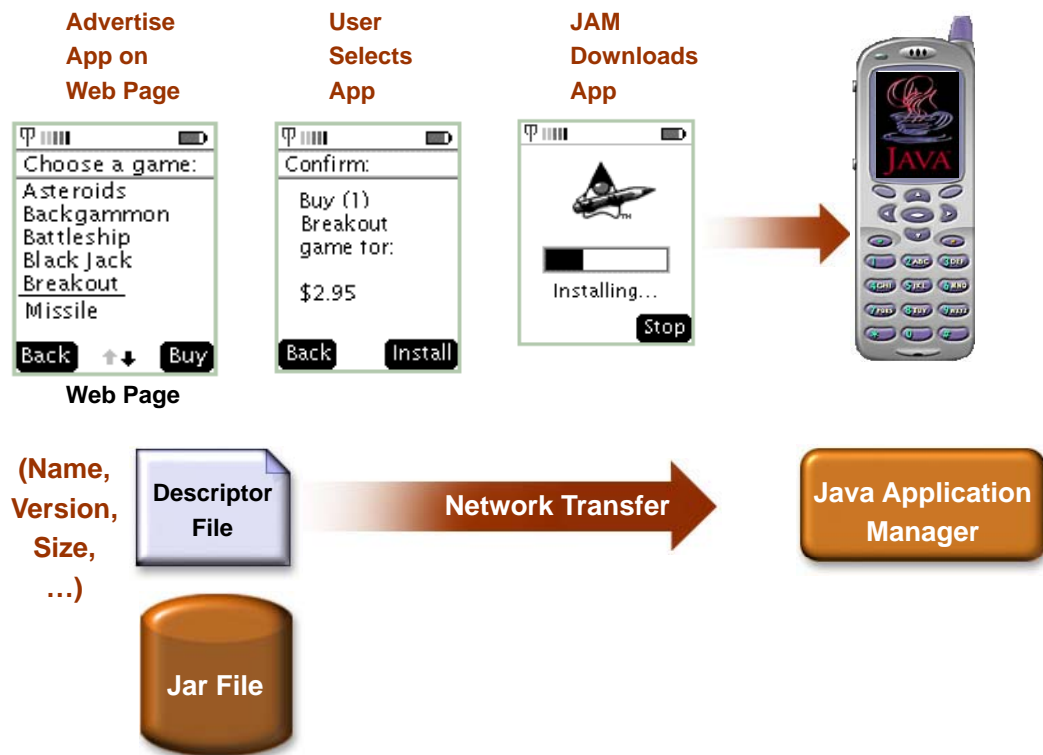
```
public void pauseApp()
{
}

public void destroyApp(boolean b)
{
}

// A convenience method for exiting
public void exitRequested()
{
    destroyApp(false);
    notifyDestroyed();
}

Emulator command: C:\Nokia_Emulators\Midp\Nokia_MIDP_Phone_Emulator_Beta_0.1\bin\nokia_beta_j2mewtk.exe -faceplate generic_phone
Debug proxy command: kdp -l 1056 -p -r localhost 2810 -classpath C:/Nokia_Emulators/Midp/Nokia_MIDP_Phone_Emulator_Beta_0.1/lib/
Connecting to localhost on port 2810
Waiting for debugger on port 1056
Connection received.
```

Ricordatevi: Processo Caricamento MIDlet



Esercitazione su J2ME: Come si Progetta Applicazione MIDlet

Realizzare una piccola **MIDlet context-dependent** (ad esempio, capace di fare playing di suoni o video differenti a seconda della locazione corrente dell'utente), sfruttando funzionalità definite in JSR 135 (nota anche come Mobile Media API, <http://java.sun.com/products/mmapi/>) e JSR 179 (nota anche come Location API, diverse implementazioni più o meno "facilmente" disponibili, ad es. <http://code.google.com/p/openlapi/>)

- ❑ Utilizzare, a propria scelta, strumenti di sviluppo come **J2ME Wireless Toolkit** oppure **Eclipse ME**

Può essere anche il seme iniziale per una possibile attività progettuale (anche se sarete probabilmente più attratti da tecnologie meno anziane ☺ come Android o iOS)...