

Java2 Micro Edition

GUINKO Tonguim Ferdinand

30 novembre 2011

Sommaire

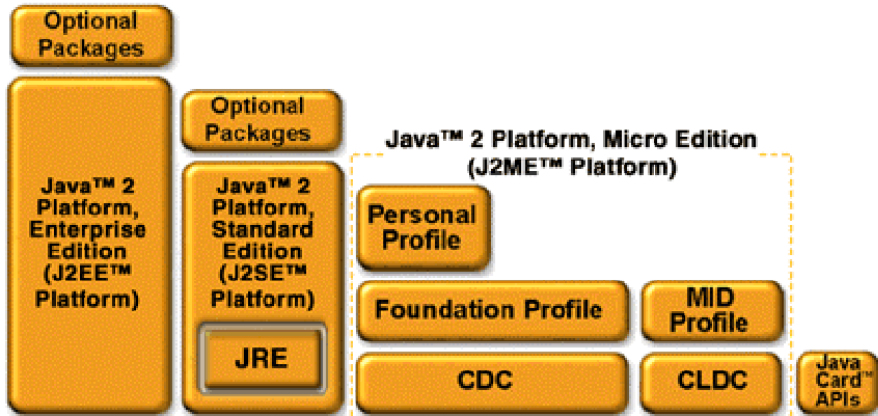
- 1 Rappels : les différentes versions de Java
- 2 J2ME
- 3 Exemple d'application
- 4 Liens utiles

Rappels

Il existe 3 plateformes de développement Java :

- 1 J2E (*Java Enterprise Edition*) : développement d'applications côté serveur ;
- 2 J2SE (*Java2 Standard Edition*) : développement d'applications (d'applets) côté terminal ;
- 3 J2ME (*Java2 Micro Edition*) : plateforme Java pour le développement d'applications pour périphériques possédant de petites ressources tel que :
 - Téléphones mobiles, PDA,
 - Systemes de navigation pour automobiles,
 - eBook, Cartable Electronique
 - Appareil photo, ...
 - Set Top Box/Decodeur TV Numérique
 - Console de Jeu
 - Terminaux de paiement, Guichet, Parcètre ...

Rappels



Contraintes matérielles des appareils mobiles

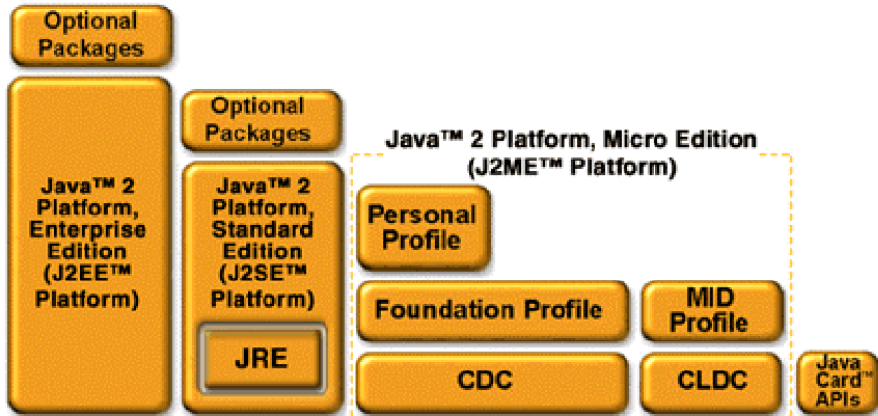
- Faible coût du handset : marché grande consommation ;
- Encombrement et poids faibles ;
- Capacité d'affichage limité : de 100*60 en NB à 640*240 en 65565 col. ;
- Faible mémoire (car grande consommation) : primaire (RAM) et secondaire (FlashRAM,EEPROM) ;
- Autonomie : Processeur faible consommation 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, ...

Généralités

J2ME :

- est la plateforme Java pour le développement d'applications à destination des périphériques mobiles et embarqués ;
- présente beaucoup d'éléments communs aux autres plateformes de développement Java : J2E et J2SE ;
- utilise des machines virtuelles différentes de celle utilisée par les plateformes J2SE et J2EE (JVM) ;
- un sous ensemble de J2SE.

J2ME comparé à J2SE et J2E



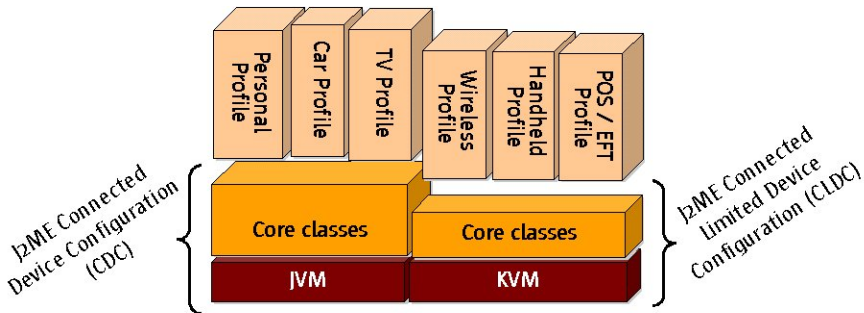
Les 2 versions de J2ME

La plateforme J2ME se décline en 2 versions :

- CLDC (Connected Limited Device Configuration) : concerne des appareils possédant des ressources faibles (moins de 512 Kb de RAM, faible vitesse du processeur, connexion réseau limitée et intermittente) et une interface utilisateur réduite. Elle s'utilise sur une machine virtuelle KVM (Kuai/Kilobyte Virtual Machine de SUN) (=V avec des limites). On la trouve par exemple dans les téléphones mobiles.
- CDC (Connected Device Configuration) : concerne des appareils possédant des ressources plus importantes (au moins 2Mb de RAM, un processeur 32 bits, une meilleure connexion au réseau). Elle s'utilise sur une machine virtuelle CVM. On la trouve dans les décodeurs de télévision numérique.

J2ME : illustration graphique

Remarque : La CVM est une machine virtuelle équivalente à la JVM ; elle est toutefois plus compacte que la JVM.



Les profils

Se composent d'un ensemble d'API particulières à un type de machines ou à une fonctionnalité spécifique. Ils permettent l'utilisation de fonctionnalités précises et doivent être associés à une configuration. Ils permettent donc d'assurer une certaine modularité à la plate-forme J2ME. (profile = ensemble d'API minimum pour des "devices" similaires)

Les profils

| Profil | Configuration | |
|----------------------------------|---------------|--|
| MIDP 1.0 | CLDC | <i>Package javax.microedition.*</i> |
| MIDP 2.0 | CLDC | |
| PDA | CLDC | <i>Symbian, PalmOS</i> |
| Mobile Media API (MMAPI) 1.1 | CLDC | <i>Permet la lecture de clips audio et vidéo</i> |
| Wireless Messaging API (WMA) 1.1 | CLDC | <i>Permet l'envoi et la réception de SMS</i> |
| Java APIs for Bluetooth | CLDC | |
| Foundation Profile | CDC | |
| Personal Profile | CDC | |
| Personal Basis Profile | CDC | |
| RMI optional Profile | CDC | <i>Remote Method Invocation</i> |
| JDBC optional Profile | CDC | |
| J2ME Web Services Specification | CDC | <i>Web Services</i> |
| Mobile 3D Graphics API for J2ME | CDC | <i>3D</i> |
| Game | CDC | <i>Pour les jeux</i> |

Exemple de profile

Avant de pouvoir développer sur un mobile il faut connaître sa configuration et ses profils qu'il peut supporter :

| Manufacturer | Model | Wireless Technology | Frequency (MHz) | Software | Screen |
|---------------|----------------|---------------------|----------------------|---|-----------------|
| Alcatel | One Touch 735i | E-GSM, GSM | 900, 1800 | MIDP 2.0, WMA 1.0, CLDC 1.0 | 128x128/12 bits |
| Motorola | A388c | GSM/GPRS | 900, 1800, 1900 | MIDP 1.0, CLDC 1.0 | 240x320/16 bits |
| Motorola | A760 | | | MIDP 2.0, CLDC 1.0 | 320x240/16 bits |
| Motorola | T720i | GSM/GPRS | 1900 | MIDP 1.0, CLDC 1.0 | 120x160/12 bits |
| Nokia | 3100 | GSM/GPRS | 900, 1800, 1900 | MIDP 1.0, WMA 1.0, CLDC 1.0 | 128x128/12 bits |
| Nokia | 3200 | GSM/GPRS/EDGE | 900, 1800, 1900 | MIDP 1.0, WMA 1.0, CLDC 1.0 | 128x128/12 bits |
| Nokia | 5140 | GSM | 900, 1800, 1900 | CLDC 1.1, MIDP 2.0 | 128x128/12 bits |
| Nokia | 9500 | GSM, GSM/GPRS | 850, 900, 1800, 1900 | JTWI 1.0, M3DAPI 1.0, CLDC 1.1, MIDP 2.0, MMAPI 1.1, LAPI 1.0, WMA 1.1 | 640x200/16 bits |
| Nokia | 7610 | GSM | 850, 900, 1800, 1900 | MIDP 1.0, MIDP 2.0, CLDC 1.0 | 176x208/16 bits |
| Samsung | SGH-S100 | GSM/GPRS | 900, 1800, 1900 | MIDP 1.0, CLDC 1.0 | 128x160/16 bits |
| Siemens | C55 | GSM/GPRS | 900, 1800 | MIDP 1.0, CLDC 1.0 | 101x64/1 bit |
| Siemens | S6V | GSM/GPRS | 900, 1800, 1900 | JTWI 1.0, M3DAPI 1.0, CLDC 1.1, MIDP 2.0, MMAPI 1.1, LAPI 1.0, JABWT 1.0, WMA 1.1 | 132x176/16 bits |
| Sony Ericsson | P908 | GSM/GPRS | 900, 1800, 1900 | MIDP 2.0, PersonalJava 1.1.1, JABWT 1.0, CLDC | 208x320/16 bits |

MIDP : prérequis

MIDP : profile standard défini pour un ensemble de machines embarquées possédant des ressources et une interface graphique limitée :

- 128KB of non-volatile memory for the MIDP implementation
- 32KB of volatile memory for the runtime heap
- 8KB of non-volatile memory for persistent data
- A screen of at least 96×54 pixels
- Some capacity for input, either by keypad, keyboard, or touch screen
- Two-way network connection, possibly intermittent

MIDP

- 1 Le MIDP (Mobile Information Device Profile) est une API JAVA dont l'objectif est de fournir des Classes JAVA permettant d'interagir avec un téléphone mobile ;
- 2 Les constructeur de téléphones mobile disposent donc de classes de gestion du mobile au niveau interface graphique (`javax.microedition.lcdui`), au niveau réseau (`javax.microedition.io`), au niveau base de donnée (`javax.microedition.rms`) ;
- 3 La classe de gestion du cycle de vie d'une application MIDP est la classe `javax.microedition.midlet.MIDlet`. Toute application MIDP doit implémenter cette classe afin de pouvoir être lancée ou arrêtée par le mobile compatible MIDP (il n'y a pas de méthode public static void `main(String arg[])` en MIDP).

MIDP : bibliothèques usuelles

- 1 java.io
- 2 java.lang
- 3 java.lang.ref
- 4 java.util

MIDP : bibliothèques non usuelles

- 1 javax.microedition.io
- 2 javax.microedition.lcdui
- 3 javax.microedition.lcdui.game
- 4 javax.microedition.media
- 5 javax.microedition.media.control
- 6 javax.microedition.midlet
- 7 javax.microedition.pki
- 8 javax.microedition.rms

MIDP : Qu'est ce qu'une application MIDP

Une application MIDP est composée de 2 parties :

- 1 fichiers `JAR` : ils contiennent toutes les classes et les ressources utilisées par l'application ;
- 2 fichiers `JAD` : ils constituent le descripteur de l'application ; ils décrivent comment exécuter l'application.
 - fichier de configuration ;
 - nom de l'application ;
 - version MIDI ;
 - Copyright et l'information sur la version ;
 - l'emplacement des fichiers `JAR`.

Outils de développement

Quelques outils :

- J2ME Wireless Toolkit 2.0.1
- Sun ONE Studio Mobile Edition 5
- CodeWarrior Wireless Studio 7.0
- Nokia Developer's Suite Beta J2ME
- RIM BlackBerry JDE 5.0
- Motorola LWT1.0

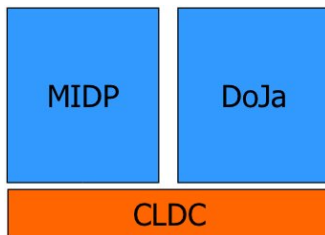
Qu'est ce qu'une application DoJa

DoJa :

- Environnement de spécification des applications créées pour les téléphones mobiles compatibles I-mode, téléphones développés par DoCoMo (Japon) ;
- Pour «téléphone portable puissant» : écran couleur, réseau GPRS, son polyphonique, HTTPS, etc., et autres périphériques (consoles de jeux, etc.) Au dessus de CLDC (donc // à MIDP)

Qu'est ce qu'une application DoJa

- DoJa demande plus de fonctionnalités que MIDP : plus de composants graphiques, HTTPS et SSL (business oblige)
- Finalement :



Définition

Système d'exploitation open source, basé sur le langage Java, pour smartphones, PDA et terminaux mobiles conçu par Android, une société rachetée depuis par Google.

Installation de la plateforme J2ME

- 1 Allez dans la rubrique `Help`; du menu sélectionnez l'option `Install new software`
- 2 Cliquez sur le bouton `Add`, puis collez l'URL suivante :
`http://download.eclipse.org/mtj/updates/1.1.2/stable` dans la zone de saisie `Add URL`
- 3 Poursuivez et terminez l'installation.

Installation du plugin Eclipse pour le développement d'application J2ME pour Blackberry

Ce plugin contient une machine virtuelle XXX de base et les bibliothèques nécessaires au développement, sous Eclipse d'applications pour Blackberry.

- 1 Si vous n'en possédez pas déjà, créez un compte à cette adresse : <https://www.blackberry.com/bdsc/?lang=na#register>
- 2 Allez dans la rubrique `Help`; du menu sélectionnez l'option `Install new software`
- 3 Cliquez sur le bouton `Add`, puis collez l'URL suivante : <http://www.blackberry.com/go/eclipseUpdate/3.6/java> dans la zone de saisie `Add URL`, puis poursuivez l'installation ;
- 4 A la fin de l'installation il vous sera demandé d'entrer, dans une fenêtre qui s'ouvrira à cet effet, votre nom d'utilisateur de développeur Blackberry et votre mot de passe, créés plus haut.

Création du projet

- 1 Créez un nouveau projet Blackberry : File → New → Project → Blackberry Project
- 2 Cliquez sur le bouton Next
- 3 Nommez le projet PremiereAppBerry
- 4 Vérifiez bien que le JRE sélectionné est Blackberry JRE 7.1.0
- 5 Cliquez sur Finish

Configuration du projet

- 1 Cliquez droit sur le nom du projet puis cliquez sur `Properties`
- 2 Cliquez sur `Blackberry Project` puis sur `Application Descriptor`
- 3 Renseignez alors les informations sur le modèle à l'intention duquel vous développez le projet en cours
- 4 Entrez comme informations :
 - Version : 1.0.0
 - Vendor : Version de Test

Création de l'application

- 1 Créez un nouveau package dans le projet : clique droit sur le nom du projet puis sur `package`
- 2 Nommez ce package `ca.uqar`
- 3 Vérifiez que le package nouvellement créé se trouve dans le repertoire `src` du projet
- 4 Créez une classe Java : nommée `BonjourLesAmis` vérifiez que cette classe se trouve bien dans le repertoire `PremiereAppBerry/src/ca/uqar`

Création de l'application

Voici le code de la classe BonjourLesAmis ;

```
package ca.uqar;  
  
import net.rim.device.api.ui.UiApplication;  
  
public class BonjourLesAmis extends UiApplication  
{  
    public static void main(String[] args)  
    {  
  
    }  
}
```

Création de l'application

Complétez le code de la classe BonjourLesAmis ainsi qu'il suit :

```
package ca.uqar;

import net.rim.device.api.ui.UiApplication;

public class BonjourLesAmis extends UiApplication
{
    public static void main(String[] args)
    {
        BonjourLesAmis theApp = new BonjourLesAmis();
        theApp.enterEventDispatcher();
    }
    public BonjourLesAmis()
    {
        //Affichage d'un nouvel écran
        pushScreen(new BonjourLesAmisScreen());
    }
}
```

Création de l'application

Complétez le code de la classe `BonjourLesAmis` ainsi qu'il suit :

```
package ca.uqar;

import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.MainScreen;

public class BonjourLesAmis extends UiApplication
{
    public static void main(String[] args)
    {
        BonjourLesAmis theApp = new BonjourLesAmis();
        theApp.enterEventDispatcher();
    }
    public BonjourLesAmis()
    {
        pushScreen(new BonjourLesAmisScreen());
    }
}
```

Création de l'application (... suite)

```
final class BonjourLesAmisScreen extends MainScreen
{
    public BonjourLesAmisScreen()
    {
        super();
        //Ici l'on fixe le titre de l'application à travers l'objet LabelField title
        LabelField title = new LabelField("PremiereAppBerry Essai",
                                         LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);
        setTitle(title);
        add(new RichTextField("Bonjour les amis!"));
    }
    //Ici l'on instancie une boîte de dialogue qui apparaîtra lorsque l'utilisateur
    //voudra quitter l'application
    //Il faut pour cela réécrire la méthode onClose de la classe BonjourLesAmis
    public boolean onClose()
    {
        Dialog.alert("Aurevoir!");
        System.exit(0);
        return true;
    }
}
```

Sites web ayant servis à la rédaction de ce cours

- <http://us.blackberry.com/developers/javaappdev/javaupdate.jsp>
- Java Micro Edition : une présentation, Jean-Marc Farinone
- J2ME, Tébourbi Riadh