

The Development of Mobile Applications with Java

Lect.dr. Paul POCATILU, asist. Cristian TOMA
Catedra de Informatică Economică, A.S.E. București

The development of mobile applications is growing every day. There are many domains where the mobile applications could be used. This paper presents the main characteristics of Java applications development for mobile devices. Also, is presented a mobile application that uses low user interface APIs.

Keywords: mobile devices, mobile applications, J2ME, CLDC, MIDP.

Aplicații și dispozitive mobile

La ora actuală există o multitudine de dispozitive mobile: telefoane mobile, PDA (Personal Digital Assistants), POS, comunicatori, telefoane inteligente (smartphones) și pagere. Caracteristicile comune ale acestor dispozitive mobile sunt:

- capacitate redusă de calcul;
- memorie puțină (RAM și ROM);
- dispozitive de afișare de dimensiuni reduse;
- interfață utilizator limitată;
- dimensiuni reduse;
- bandă de transfer limitată.

Aceste caracteristici influențează modul de proiectare și realizare a aplicațiilor mobile. Aplicațiile mobile sunt aplicații concepute să fie executate pe dispozitive mobile care suportă o tehnologie adecvată. Avantajele aplicațiilor mobile sunt:

- implementarea directă și rapidă a aplicațiilor având în vedere compatibilitatea între protocoalele cunoscute – HTTP, HTTPS și WAP (Wireless Application Protocol);
- costurile reduse de obținere a informației și costurile scăzute în raport cu alte dispozitive cu aceeași funcționalitate
- portabilitate din punct de vedere al utilizatorului foarte ridicată pentru utilizatorii și ce care dezvoltă m-aplicații;
- absența conexiunilor fizice care limitează mobilitatea în timp și spațiu a utilizatorului. Aplicațiile mobile prezintă și o serie de dezavantaje precum:
 - limitarea introdusă de memoriile folosite de telefoanele mobile;
 - limitarea introdusă de suprafața ecranului; pot fi afișate în general circa 5 linii de text;

▪ viteză mai scăzută a ratei de transfer decât în alte conexiuni, de exemplu decât într-o linie închiriată la un ISP (Internet Service Provider),

ceea ce conduce la un proces mai dificil de dezvoltare a acestora.

În momentul de față dispozitivele mobile, tehnologiile și aplicațiile pentru aceste dispozitive cunosc o dezvoltare continuă. Domeniile în care se dezvoltă astfel de aplicații sunt variate, precum: comerț electronic (*m-commerce*); managementul informațiilor; divertisment (jocuri, aplicații multimedia); afaceri.

Există mai multe criterii de clasificare aplicațiilor mobile, cum ar fi criteriul portabilității (native, portabile), al distribuirii în spațiu (distribuite, de tip desktop) ș.a. La ora actuală, cele mai cunoscute sisteme de operare pentru care se dezvoltă aplicații mobile native sunt Windows, Symbian și Palm.

Platforma J2ME

Platforma J2ME (Java 2 Micro Edition) dezvoltată de firma Sun funcționează pe diverse dispozitive mobile pe care este instalată o mașină virtuală Java (JVM). Cel mai mare avantaj al utilizării platformei Java pentru dezvoltarea aplicațiilor mobile îl constituie posibilitatea de a realiza cod portabil care poate rula pe platforme diferite. Totuși, este aproape imposibil de a porta întreaga funcționalitate a unei aplicații pe toate dispozitivele mobile, deoarece acestea au caracteristici care diferă de la un dispozitiv la altul, cum ar fi:

- memoria disponibilă (RAM și ROM);
- capacitatea de procesare;

- dimensiunea ecranului;
- lățimea de bandă;
- durata de viață a bateriei.

J2ME este împărțită în configurații și profiluri. Configurațiile conțin bibliotecile de bază ale limbajului Java pentru o categorie de dispozitive. Deasupra fiecărei configurații este un profil. Profilurile definesc biblioteci specifice dispozitivelor mobile pentru interfața utilizator, rețea și stocarea datelor. Fiecare profil are propriul mediu de execuție și este realizat pentru un număr de dispozitive mobile similare.

La ora actuală există două configurații:

- *Connected Limited Device Configuration (CLDC)* – proiectată pentru dispozitive cu restricții de resurse: telefoane mobile, unele PDA-uri; aceste dispozitive pun la dispoziție între 128 KB și 512 KB memorie pentru mașina virtuală Java (KVM);

- *Connected Device Configuration (CDC)* – proiectată pentru dispozitive mobile mai puternice: PDA evaluate, dispozitive de rețea; Mașina virtuală Java pentru aceste dispozitive (JVM) are un set mai mare de clase, de aceea memoria pusă de la dispoziție de aceste dispozitive ajunge până la 2 MB;

Configurația CDC are mai multe funcții comparativ cu configurația CLDC (funcții matematice, de intrare/ieșire, securitate), de fapt configurația CLDC fiind inclusă în configurația CDC. Pentru configurația CLDC cel mai important profil este *Mobile Information Device Profile (MIDP)*. Pentru configurația CDC există două profiluri:

- *Foundation Profile (FP)* – este profilul pe primul nivel al configurație CDC și furnizează suportul de rețea pentru dispozitive mobile

- *Personal Profile (PP)* – destinat dispozitivelor mobile cu interfață grafică și/sau acces la Internet și include biblioteca Java AWT și suport pentru applet-uri Web; un subset al PP este *Personal Base Profile (PBP)* destinat dispozitivelor cu acces la rețea și anumite elemente de interfață grafică.

În figura 1 este prezentată platforma J2ME cu cele două configurații și profilurile existente. Peste profilurile existente pot fi adăugate profiluri opționale, specifice fiecărui dispozitiv mobil.

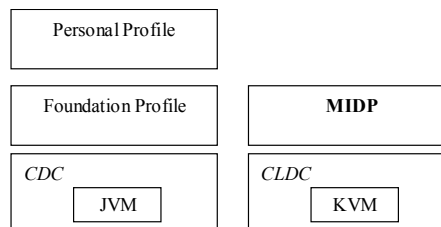


Fig.1. Platforma Java 2 Micro Edition

Bibliotecile de clase puse la dispoziția aplicațiilor mobile Java sunt limitate la caracteristicile dispozitivelor mobile, spre deosebire de aplicațiile Java care folosesc J2SE sau J2EE. Astfel, pachetele existente în CLDC și MIDP sunt:

- java.lang
- java.io
- java.util
- javax.microedition.io
- javax.microedition.lcdui
- javax.microedition.midlet
- javax.microedition.rms

Clasele comune din pachetele J2SE și J2ME sunt puține și la unele dintre acestea lipsesc o funcții și date membre față de configurațiile evaluate. De exemplu, CLDC nu oferă suport pentru virgulă mobilă, de aceea clasele *Double* și *Float* au fost eliminate, precum și metodele din alte clase care operează asupra acestor tipuri de date.

În momentul de față există specificațiile MIDP 2.0, care sunt o extindere a specificațiilor MIDP 1.0. Specificațiile MIDP 2.0 aduc o serie de îmbunătățesc și extensii în ceea ce privește accesul la rețea, securitatea, interfața grafică și managementul aplicațiilor.

Noile pachete introduse în MIDP 2.0 sunt:

- javax.microedition.lcdui.game
- javax.microedition.media
- javax.microedition.pki

În ceea ce privește rețeaua, MIDP 2.0 pune la dispoziție noi clase precum suport pentru *HTTPS* și noi clase: *CommConnection*, *UDPDatagramConnection*, *SocketConnection* și *ServerSocketConnection*.

La capitoulul securitate este pus la dispoziție un manager de securitate bazat pe domenii, semnarea codului aplicațiilor și verificarea certificatelor digitale.

Sunt aduse elemente de interfață grafică noi: *CustomItem*, *POPUP ChoiceGroup*, *Spacer*

și *GameCanvas* și sunt actualizate o serie de elemente grafice existente:

- obiectele de tip *Form* și componentele acestora: *ChoiceGroup*, *Gauge*, *ImageItem*, *Item*, *StringItem* și *TextField*

- *Alert*, *Canvas*, *Choice*, *Command*, *Display*, *Displayable*, *Image*, *List*, *Screen*, *TextBox* și *Ticker*.

Este oferit suport pentru recepționarea datelor prin metoda PUSH, modalitatea de descărcare și instalare a aplicațiilor fiind standardizată (OTA – Over The Air). Sunt aduse noi capacități în ceea ce privește multimedia, grafica și jocurile. Multe dispozitive mobile de pe piața internă care au mașina virtuală Java sunt oferă suport doar pentru specificațiile MIDP 1.0.

Realizarea MIDlet-urilor

Pentru realizarea aplicațiilor Java destinate dispozitivelor mobile sunt urmate o serie de etape, printre care:

- proiectare;
- codificare;
- compilare;
- pre-verificare (verificarea modului în care programul utilizează clasele puse la dispoziție de bibliotecile J2ME);
- împachetare (crearea unei arhive JAR și a unui fișier de descriere a aplicației, JAD);
- execuție;
- testare;
- depanare.

Un MIDlet este o aplicație Java care se execută în cadrul mașinii virtuale Java. Fiecare MIDlet conține o clasă principală derivată din *javax.microedition.midlet.MIDlet*. Sistemul de management a aplicațiilor instalat pe dispozitivul mobil încarcă MIDlet-ul pentru execuție. Este executat constructorul clasei, după care aplicația trece în starea *Înterupt*. Activarea MIDlet-ului se realizează prin apelul metodei *startApp*. În figura 2 sunt prezentate stările pe care le poate avea un MIDlet, precum și modalitățile prin care acesta poate să treacă dintr-o stare în alta.

Un MIDlet ajunge în starea *Înterupt* în momentul în care sunt întreprinse acțiuni externe aplicației (sună telefonul, primirea unui mesaj) sau prin metodei *notifyPaused*.

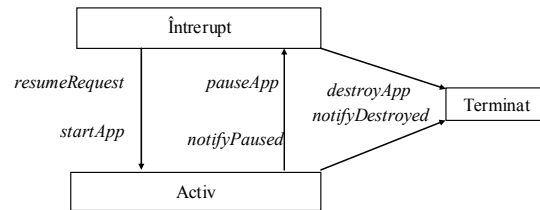


Fig.2. Stările unui MIDlet

Metodele *pauseApp* și *destroyApp* conțin cod pentru eliberarea resurselor în cazul în care aplicația este întreruptă sau terminată.

Funcționalitatea MIDlet-ului este asigurată prin intermediul claselor interfeței grafice. Interfața grafică a unui MIDlet este asigurată prin API de nivel scăzut sau de nivel ridicat.

API-ul de nivel scăzut presupune utilizarea obiectelor de tip *Canvas*, care permit accesul la ecranul dispozitivului mobil pixel cu pixel. Tratarea evenimentelor se realizează de către programator la nivelul obiectului de tip *Canvas*. API-ul pentru interfața grafică de nivel ridicat presupune lucrul cu obiecte grafice predefinite, cum ar fi formularele (*Form*) și elemente de interfață care pot fi asociate unui formular (*ChoiceGroup*, *Gauge*, *ImageItem*, *DateField*, *StringItem* și *TextField*) sau de tip *Screen* (un singur obiect pe ecran), precum *List*, *TextBox* și *Alert*. Tratarea evenimentelor se realizează standard, la nivelul platformei.

Dintre instrumente de dezvoltare a aplicațiilor mobile Java sunt amintite Borland JBuilder X, care include extensia Mobile Set și Sun One Studio, Mobile Edition. Java Wireless Toolkit conține bibliotecile necesare dezvoltării de aplicații mobile Java și un emulator pentru testarea aplicațiilor.

Aplicație Java pentru dispozitive mobile

Pentru exemplificare a fost realizată o aplicație care folosește funcțiile API de nivel scăzut pentru interfața utilizator. Aplicația este implementarea jocului "X și 0" pentru 2 jucători, pe o tablă de 3 x 3 celule. În cadrul clasei sunt declarate trei obiecte de tip *Command*, corespunzătoare celor trei comenzi implementate în aplicație:

```

private Command exitCommand;
private Command jocNouCommand;
private Command despreCommand;
  
```

De asemenea, sunt declarate două obiecte utilizate în realizarea interfeței cu utilizatorul, unde clasa *Cnvs* este derivată din *Canvas*:

```
private Display display;
private Cnvs canvas;
```

Constructorul clasei inițializează obiectele de tip *Display* și *Command*. Fiecare MIDlet are un singur obiect de tip *Display* care este obținut prin apelul metodei statice *getDisplay*. Inițializarea obiectelor de tip *Command* se realizează prin precizarea în cadrul constructorului a etichetei asociate comenzii, tipul comenzii și prioritatea asociată.

```
public Xsi0MIDlet()
{display = Display.getDisplay(this);
 exitCommand = new Command("Iesire",
 Command.EXIT, 1);
 jocNouCommand = new Command("Nou",
 Command.SCREEN, 2);
 despreCommand = new Command("Despre",
 Command.SCREEN,
 3);
}
```

În metoda *startApp* este creat un obiect derivat din clasa *Canvas*. Acestui obiect îi sunt asociate comenzile care vor fi acționate în timpul execuției aplicației. Prin apelarea metodei *setCurrent*, obiectul de tip *Canvas* va fi activat.

```
public void startApp()
{canvas=new Cnvs(display);
 canvas.addCommand(exitCommand);
 canvas.addCommand(jocNouCommand);
 canvas.addCommand(despreCommand);
 canvas.setCommandListener(this);
 display.setCurrent(canvas);
}
```

Desenarea suprafeței ecranului se realizează în cadrul clasei *Cnvs*, derivată din *Canvas*, prin metoda *paint*, care are prototipul:

```
protected void paint(Graphics g)
unde g este obiectul de tip Graphics prin intermediul căruia sunt desenate pe ecran diverse elemente grafice.
```

Tratarea evenimentelor de la tastatură se realizează în cadrul funcției *keyPressed* din clasa *Canvas*. În cadrul acestei aplicații sunt utilizate tastele numerice de la 1 la 9, celelalte fiind ignorate.

```
protected void keyPressed(int keyCode)
{if (keyCode<KEY_NUM1||keyCode>KEY_NUM9)
 return;
 int p=(keyCode-KEY_NUM0)-1;
```

```
int i,j;
i=p/3;
j=p-i*3;

//...

repaint();
}
```

În figura 3 este prezentată interfața grafică a aplicației pe două simulatoare: unul pentru Siemens C60 și un simulator generic. Datorită dimensiunilor diferite ale ecranului se observă că există mici diferențe între ceea ce se afișează.



Fig.3. Aplicația în simulatoare de telefoane mobile

Fișierele binare care conțin clasele aplicației și eventualele resurse sunt arhivate într-un fișier de tip JAR. Pe lângă acest fișier arhivă mai există și un fișier de descriere a aplicației JAD (Java Application Descriptor). Structura fișierului JAD asociat aplicației este:

```
MIDlet-Version: 1.0
MIDlet-Vendor: PoL
MIDlet-Jar-URL: Xsi0.jar
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
MIDlet-1: Xsi0MIDlet, Xsi0MIDlet
MIDlet-Jar-Size: 5294
MIDlet-Name: Xsi0
```

Înregistrările sunt de tipul *atribut: valoare*. Semnificația atributelor reiese din numele acestora. Astfel, sunt precizate versiunea aplicației, producătorul, URL-ul arhivei, versiunea configurației și profilului pentru care a fost realizată aplicația ș.a. Codul sursă precum și arhiva Java și fișierul de descriere a aplicației sunt disponibile pe site-ul <http://pocatilu.ase.ro>, secțiunea Download.

Concluzii

În realizarea de aplicații Java pentru telefoane mobile trebuie să se țină cont de mulți factori, care țin în general de caracteristicile dis-

pozitivului. Se observă faptul că există diferențe ale interfeței grafice de la telefon la telefon. Utilizarea claselor și funcțiilor oferite de configurațiile și profilurile standard cresc portabilitatea, însă realizarea aplicațiilor este mai dificilă, având la dispoziție un set restrâns de facilități. De aceea, în dezvoltarea aplicațiilor mobile dezvoltate pentru platforma Java 2 ME, trebuie luat în considerare familia de dispozitive mobile pe producător și clase de produse și să se utilizeze bibliotecile dezvoltate special pentru acele dispozitive mobile de către producători. Un exemplu ar fi API-ul pus la dispoziție de Nokia și Siemens pentru telefoane mobile.

Bibliografie

- [GIGU99] Eric GIGUÈRE – *Designing Mobile Applications for the Enterprise* in Palm Database Programming: The Complete Developer's Guide, John Wiley & Sons, 1999
- [IVAN01] Ion IVAN, Paul POCATILU, Cristian TOMA, Alexandru LEAU – *e³-commerce: e-commerce, mobile application. Aplicația e3com*, în *Informatica Economică* vol. V, nr. 3 (19), 2001, pp. 16-23
- [IVAN03] Ion IVAN, Cristian TOMA, “Requirements for building distributed informatics applications”, *The Automatics and Computer Science Romanian Magazine*, vol. 13, No. 4, November 2003.
- [KANE96] KANER, Cem – *Quality Cost Analysis: Benefit and Risks*, Software QA, Volume 3, No. 1, 1996, pp. 23
- [PIRO02] Vartan Piroumian – *Wireless J2ME™ Platform Programming*, Prentice Hall PTR, 2002
- [POCA03] Paul POCATILU – *Mobile Applications Testing* în volumul *International Workshop IE&SI*, ediția a II-a, Timișoara, 23-24 Mai 2003, pag 41-44
- [POCA03a] Paul POCATILU – *Aplicații mobile pentru gestiunea afacerilor*, în volumul *Simpozionului Internațional ”Specializare, Dezvoltare si Integrare”*, Universitatea Babes-Bolyai, Cluj-Napoca, 14-15 Noiembrie 2003, pp. 91-95
- [POCA04] Paul POCATILU, Cristian TOMA – *Securing Mobile Commerce Applications*, în volumul *Conferinței Internaționale Central and East European Conference on Business Information Systems (CEECBIS)*, Universitatea Babes-Bolyai, Cluj-Napoca, 20-22 Mai 2004, pp. 270-279
- [SCHL01] Christian SCHLÄPFER, Michal KUBIK, Guido ZAVAGLI, - *Mobile Applications with J2ME™*, Ericsson Radio Systems AB, 2001
- [*****] Java 2 Platform, Micro Edition (J2ME), <http://java.sun.com/j2me>
- [*****] Nokia Web Site, <http://www.nokia.com>
- [*****] Siemens Developer Site <http://www.siemens-mobile.com/developer>