

TEHNOLOGII JAVA PENTRU DEZVOLTAREA APLICAȚIILOR LUCRARE DE LABORATOR 23

J2ME – Aplicații Java Mobile

I. SCOPUL LUCRĂRII

Lucrarea de față are rolul de a prezenta și familiariza studentul cu modul de implementare și rulare a aplicațiilor mobile, folosind tehnologia Java: J2ME.

La sfârșitul acestei lucrări, studentul va avea posibilitatea să scrie aplicații Java în care să utilizeze noțiunile învățate.

II. NOȚIUNI TEORETICE

Vezi curs.

Prezentăm mai jos câteva exemple de aplicații mobile Java care să vină în completarea și aprofundarea cunoștințelor dobândite în cursurile și laboratoarele precedente.

Exemplul 1

Exemplul următor prezintă o aplicație mobilă Java ce afișează o imagine în mijlocul ecranului. Din moment ce clasa *Image*(J2ME) nu suportă vreo metodă prin care să se redimensioneze automat am ales o imagine pe care o desenez cu *low level Api*.

Am suprascris clasa *Canvas*, iar în metoda *paint* a acesteia se fac toate operațiile de desenare. Pentru a rula pe un tel mobil va trebui un cablu de date, iar proiectul (fișierele necesare - *.class și /*exemplul1.png*) trebuie arhivat cu utilitarul *jar.exe* care se găsește în versiunile J2SE compatibile.

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class TestMid extends MIDlet
{
    ImgCanvas cnv = null;

    protected void startApp() throws MIDletStateChangeException
    {
        cnv = new ImgCanvas();
        Display d = Display.getDisplay(this);
        d.setCurrent(cnv);
    }

    protected void pauseApp()
    {
        // Do stuff to interrupt midlet execution
    }
}
```

```

    }

    protected void destroyApp(boolean b) throws MIDletStateChangeException
    {
        //Cleanup stuff :)) none here
    }
}

class ImgCanvas extends Canvas    //Custom canvas
{
    private Image img = null;
    private Ticker tick = null;

    public ImgCanvas()
    {
        this.tick = new Ticker("Acesta este un exemplu!");
        try
        {
            this.img = Image.createImage("/exemplul1.png");
            this.setTicker(tick);
        }
        catch (Exception ex)
        {
            ex.printStackTrace();
        }
    }

    protected void paint(Graphics graphics)
    {
        int width = this.getWidth();
        int height = this.getHeight();

        graphics.drawImage(img, width/2, height/2, Graphics.VCENTER |
Graphics.HCENTER);
    }
}

```

Exemplul 2

In exemplul de mai jos prezentam o aplicatie J2ME(MIDlet) care afiseaza 2 intrebari referitoare la 2 imagini cu posibilitatea de a alege raspunsul corect din 3 variante date. La sfirsit va afisa punctaj obtinut in functie de raspunsurile date.

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Test extends MIDlet implements CommandListener
{
    private Display display;    // The display
    private Command ExitCmd;
    private Command PlayCmd;

```

```

private Form rez=new Form(" ");

private Command BackCmd;
private Command ValaCmd;
Main main=new Main(be("stop.png"));
Joc joc=new Joc();

public Test()
{
    display = Display.getDisplay(this);
    ExitCmd = new Command("EXIT",Command.EXIT,0);
    PlayCmd = new Command("RUN",Command.SCREEN,1);
    ValaCmd = new Command("OK",Command.EXIT,1);
    BackCmd = new Command("BACK",Command.SCREEN,0);
}

protected void startApp()
{
    main.addCommand(ExitCmd);
    main.addCommand(PlayCmd);
    main.setCommandListener((CommandListener) this);
    joc.addCommand(ExitCmd);
    joc.addCommand(ValaCmd);
    joc.setCommandListener((CommandListener) this);
    rez.addCommand(BackCmd);
    rez.setCommandListener((CommandListener) this);
    display.setCurrent(main);
}

protected void pauseApp()
{ }

protected void destroyApp( boolean unconditional )
{ }

public void exitMIDlet()
{
    destroyApp(true);
    notifyDestroyed();
}

public void commandAction(Command cmd,Displayable screen){
    if(cmd==PlayCmd){
        display.setCurrent(joc);
    }
    else if(cmd==ExitCmd){
        exitMIDlet();
    }
    else if(cmd==ValaCmd){
        if(rez.size()>0)
            for(int i=0;i<rez.size();i++)
                rez.delete(i);

        rez.append("Punctajul obtinut:"+joc.ellenoriz());
    }
}

```

```

        display.setCurrent(rez);
    }
    else if(cmd==BackCmd){ display.setCurrent(joc);}
}
public Image be(String img){
    Image image = null;

    try {
        image = Image.createImage("/"+img);
    } catch (Exception e) {}
    return image;
}

}

//-----

//-----
class Joc extends Form{

    String kerdesek []= {"Care e semnificatia indicatorului?", "Ce
semnificatie are indicatorul alaturat?"};
    ChoiceGroup[] cg;
    int i,j;
    String opt [][] ={ {"Drum cu prioritate.", "Cedeaza
trecerea.", "Trecere la nivel."}, {"Viteza admisa", "Distanța de un
obstacol.", "Limitare de viteza."}};
    int corect [] = {1,2};
    String poz[]= {"cedeaza.png", "limitare.png"};

    public Joc(){
        super("");

        cg = new ChoiceGroup[kerdesek.length];

        for(i = 0;i<cg.length;++i){
            append(new StringItem(null, kerdesek[i]));
            append(be1(poz[i]));
            cg[i] = new ChoiceGroup("Optiuni:", Choice.EXCLUSIVE);
            for(j = 0;j<opt[i].length;++j){
                cg[i].append(opt[i][j], null);
            }
            append(cg[i]);
        }
    }

}

//-----
public Image be1(String img){
    Image image = null;

```

```

        try {
            image = Image.createImage("/"+img);
        } catch (Exception e) {}
        return image;
    }
    public int ellenoriz(){
        int nota = 0;
        for(i = 0;i<cg.length;i++)
            if(cg[i].getSelectedIndex() ==corect[i] )
                nota++;
        return nota;
    }
}

//-----
class Main extends Canvas{
    Image image;
    public Main(Image image){
        this.image= image;
    }
    public void paint(Graphics g){
        if(image != null)

g.drawImage(image,getWidth()/2,getHeight()/2,Graphics.VCENTER |
Graphics.HCENTER);
    }
}
//-----

```

Exemplul 3

In exemplul urmator se va implementa un meniu particularizat.

Exemplul consta in doua clase:

- prima este clasa MIDlet: *SimpleCustomMenu*
 1. a doua este de tipul *Canvas*: *MenuScreen* (clasa care se ocupa cu toate operatiile de desenare).

Codul include cateva metode care incarca sa deseneaza o imagine ca background.

Pentru a insera imaginea puteti sa adaugatzi in root-ul proiectului, in folderul "res" imaginea "exemplu3.png".

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.util.*;
import java.io.IOException;

public class SimpleCustomMenu extends MIDlet
{
    Display display;

```

```

Display pauseDisplay;
boolean isSplash = true;
MenuScreen menuScreen;

public SimpleCustomMenu()
{
    MenuScreen menuScreen = new MenuScreen();
    display = Display.getDisplay(this);
    display.setCurrent(menuScreen);
}

protected void startApp() throws MIDletStateChangeException
{
}

protected void pauseApp()
{
}

protected void destroyApp(boolean flag) throws MIDletStateChangeException
{
}
}

class MenuScreen extends Canvas implements Runnable
{
    // Set Fonts
    static final Font lowFont = Font.getFont(Font.FACE_MONOSPACE,
        Font.STYLE_PLAIN, Font.SIZE_SMALL);
    static final Font highFont = Font.getFont(Font.FACE_MONOSPACE,
        Font.STYLE_BOLD, Font.SIZE_MEDIUM);
    // Set Color
    Image img = null; //background

    static final int lowColor = 0x000000FF; // Not Highlighted
    static final int highColor = 0x00FF0000; // Highlighted

    static int width; // screen width
    static int height; // screen height

    static int startHeight; // height where the menu starts

    static final int spacing = highFont.getHeight() / 2; // spacing between
    menu items
    // Menu Item Labels
    static final String[] mainMenu = {"Add Record", "Get
Record", "Find", "Settings", "Help", "About"};
    // To hold the current highlighted menu option
    static int menuIdx;
    // Menu Thread
    Thread menuThread;

    // Constructor

```

```

public MenuScreen()
{
    try
    {
        img = Image.createImage("/exemplul3.png");
    } catch (IOException e)
    {
        System.out.println("Eroare la incarcarea imaginii !");
        System.exit(0);
    }
// Get Width and Height of Canvas
    width = getWidth();
    height = getHeight();

    // Calculate Start Height of Menu
    startHeight = (highFont.getHeight() * mainMenu.length) +
((mainMenu.length - 1) *
    spacing);
    startHeight = (height - startHeight) / 2;
// Set Selected Menu Item to the first menu item
    menuIdx = 0;
// Create Thread and Start
    menuThread = new Thread(this);
    menuThread.start();
}
// Simple Run -- Should be modified for better performance/efficiency
public void run()
{
    while (true)
    {
        repaint();
    }
}
// Paint Main Menu
public void paint(Graphics g)
{
    g.setColor(0x00FFFFFF);
    g.fillRect(0, 0, width, height);
    g.drawImage(img,0,0,0);
    for (int i = 0; i < mainMenu.length; i++)
    {
        if (i == menuIdx)
        {
            g.setFont(highFont);
            g.setColor(highColor);
            g.drawString(mainMenu[i],
                (width - highFont.stringWidth(mainMenu[i])) / 2,
                startHeight + (i * highFont.getHeight()) + spacing,
                20);
        } else
        {
            g.setFont(lowFont);
            g.setColor(lowColor);

```

```

        g.drawString(mainMenu[i],
            (width - lowFont.stringWidth(mainMenu[i])) / 2,
            startHeight + (i * highFont.getHeight()) + spacing,
            20);
    }
}
}
// Capture Keypresses for Menu Selection
protected void keyPressed(int code)
{
    if (getGameAction(code) == Canvas.UP && menuIdx - 1 >= 0)
    {
        menuIdx--;
    }
    else if (getGameAction(code) == Canvas.DOWN && menuIdx + 1 <
        mainMenu.length)
    {
        menuIdx++;
    }

    else if (getGameAction(code) == Canvas.FIRE)
    {
        System.out.println(height);
        System.out.println(width);
    }
}
}
}

```

Resurse

<http://www.java.sun.com>
www.wireless.java.sun.com
www.nokia.com

III. MODUL DE LUCRU

Pentru a putea să dezvoltați propriile aplicații în Java pentru telefoanele voastre mobile, folosind tehnologia J2ME, trebuie să aveți instalate următoarele

1. **Java 2 Standard Edition** (v. 1.4.2 sau mai evoluată), adică necesarul pentru a programa în Java; include compilatorul Java, mașina virtuală Java, și alte unelte de dezvoltare și documentare.
2. **J2ME Wireless Toolkit**, cel care se ocupă de partea mobilă a Java, și care vă ajută să realizați și să testați aplicații scrise pentru device-uri mobile. Există și varianta de a downloada și instala și alte unelte de dezvoltare, dar aceasta este configurația minimă pentru a putea să lucrați. De remarcat că aceste două pachete sunt disponibile și atât pentru Linux, cât și pentru Windows.

Presupunând că lucrați pe platformă Windows (2000, XP, etc) pentru a realiza un nou proiect, trebuie să faceți următorii pași:

1. **Start -> Programs -> J2ME Wireless Toolkit 2.0 -> Ktoolbar** pentru a porni aplicația **KToolbar**, aplicație care vă ajută la realizarea de proiecte **J2ME**.
2. Apăsați pe butonul '**New Project**' pentru a începe un nou proiect, și introduceți numele proiectului, precum și al clasei **MIDLet**-ului vostru.
3. Apăsând butonul '**Create Project**' va apare pe ecran fereastra cu opțiuni și setări ale proiectului. Opțiunea foarte importantă este aceea numită "**MicroEdition-Profile**" pe care o veți seta la **MIDP-1.0** pentru a asigura compatibilitatea cu device-urile care folosesc ca specificație **MIDP-1.0** (majoritatea device-urilor din ziua de azi, deși cele mai noi și moderne au trecut la specificații mai avansate).
4. Apăsând butonul '**OK**' setările proiectului vor fi salvate, și puteți trece la scrierea codului sursa pentru **MIDLet**.

Dupa ce ati terminat crearea proiectului, atunci pe ecran vor apare setările proiectului nostru, incluzând directoarele în care se vor copia sursele programului, bibliotecile folosite, și alte resurse.

În cazul nostru, dacă am denumit proiectul nostru ca fiind Proiect1, atunci directorul unde trebuie să copiem o clasă pentru a o compila și pentru a rula programul este **c:\WTK20\apps\Proiect1\src**.

Scrierea codului Java se face ca pentru orice aplicație, în orice editor de text. De remarcat că numele clasei principale trebuie să fie cel declarat la crearea proiectului, iar denumirea fișierului cu extensie **.java** trebuie să fie aceeași cu numele clasei. Odată realizate aceste deziderate, apăsați butonul '**Build**' și dacă totul este ok, și sursa este corect scrisă, atunci sunteți gata să rulați prima voastră aplicație Java2ME. Apăsați '**Run**' și emulatorul de telefon prezent în **J2WTK** pornește, deci veți putea să vedeți rezultatul codului scris de voi.

IV. TEMĂ

1. Se vor parcurge exemplele prezentate în platforma de laborator testându-se practic.