Listando Informações do PIM a Partir da JSR-75

Introdução

A API do Gerenciador de Informações Pessoais (Personal Information Manager - PIM, JSR-75) permite que aplicações java leiam e escrevam do/a partir das informações pessoais salvas no dispositivo, tais como contatos telefônicos, eventos do calendário, etc. Existem 3 categorias de informações salvas no PIM: Contatos, Calendário e To-Do (lista de afazeres). O código abaixo mostra a lista dos banco de dados do PIM (também conhecido como lista do PIM) que existe no dispositivo.

Observação: É normal que o dispositivo realiza diversas perguntas para permissão de leitura dos dados do usuário. Isto acontece devido ao método `javax.microedition.pim.PIM.listPIMLists()` e o comportamento não confiável das MIDlets.

Código-fonte: PIMMIDlet.java

```
import javax.microedition.lcdui.Display;
import javax.microedition.midlet.MIDlet;
```

```
public class PIMMIDlet extends MIDlet {
    private PIMDbForm pimDatabases;

    /**
     * Constructor. Constrói os objetos necessários
     */
    public PIMMIDlet() {
        pimDatabases = new PIMDbForm(this);
    }

    /**
     * From MIDlet.
     * Invocado quando a MIDlet é inicializada
     */
    public void startApp() {
        Display.getDisplay(this).setCurrent(pimDatabases);
    }

    /**
     * From MIDlet.
     * Invocado para colocar a MIDlet no estado Pause
     */
    public void pauseApp() {
        // Nenhuma implementação é necessária neste exemplo
    }

    /**
     * From MIDlet.
     * Invocado para finalizar a MIDlet.
     * @param unconditional
     * terminated
     */
    public void destroyApp(boolean unconditional) {
        // Nenhuma implementação é necessária neste exemplo
    }
}
```

Código-fonte: PIMDbForm.java

```
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.pim.PIM;
```

```
/**
 * Lista de banco de dados do PIM
 */
public class PIMDbForm extends Form implements CommandListener {
    private Command exitCommand;
    private PIMMIDlet parent;
    private int itemCounter = 0;  // Número de itens listados

    public PIMDbForm(PIMMIDlet parent) {
        super("PIM databases");

        this.parent = parent;

        exitCommand = new Command("Sair", Command.EXIT, 0);
        addCommand(exitCommand);
        setCommandListener(this);

        // Cria uma instância de PIM
        PIM pim = PIM.getInstance();
        // Adiciona cada tipo de dado PIM (PIM databases) no form
        addList(PIM.CONTACT_LIST, pim);
        addList(PIM.EVENT_LIST, pim);
        addList(PIM.TODO_LIST, pim);
    }

    public void commandAction(Command command, Displayable displayable) {
        if (command == exitCommand) {
```


(C) Copyright Nokia 2013. All rights reserved.

```
            // Finaliza a MIDlet
            parent.notifyDestroyed();
        }
    }

    /**
     * Adiciona os nomes de cada tipo específico da lista de PIM no form
     * @param listType o tipo da lista
     */
    private void addList(int listType, PIM pim) {
        String[] list = pim.listPIMLists(listType);
        for (int i = 0; i < list.length; i++) {
            itemCounter++;
            StringItem item = new StringItem("Database " + itemCounter, list[i]);
            append(item);
        }
    }
}
```

Pós-condições

A MIDlet mostra a partir de um objeto Form todos os banco de dados que existem no dispositivo

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.