Lista com Imagens em Java ME

Este artigo explica como desenvolver uma lista em J2ME com imagens e ao clique do usuário em cada item exibir uma tela com detalhamentos. Bom para ser utilizado como template para apps voltadas para Series 40.

Introdução

A você desenvolvedor que deseja implementar apps que se baseiam em lista com imagens e detalhamento destas imagens (em tamanho maior) e com descrição o código abaixo ajuda bastante. O código se encontra documentado. E neste post você pode baixar o template para produzir sua app.

Código

```
import java.io.IOException;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
/*
 * Template para criação de apps com listagem de elementos e uma tela para
 * detalhamento da imagem e informações.
 */
public class ListaMidlet extends MIDlet implements CommandListener{
 private Display display;
 private List list;
 private Command exit, select, back;
 private Image img1,img2,img3,img4,img5, img6;
 String[] stringElements = {"Batman", "Carro", "Guitarra", "Maça","Sapo", "Simpsons"};
        Image[] images;

        //Método para criar, carregar para a memória as imagens que serão
        //utilizadas durante a app.
 public ListaMidlet(){
  try{
   img1 = Image.createImage("/batman.gif");
   img2 = Image.createImage("/carro.png");
   img3 = Image.createImage("/guitarra.jpg");
   img4 = Image.createImage("/maca.jpg");
   img5 = Image.createImage("/sapo.jpg");
                        img6 = Image.createImage("/simpsons.gif");
  }catch(Exception e){
   System.err.println(e.getMessage());
  }
 }

 public void startApp() {
  display = Display.getDisplay(this);
  Image[] imageElements = {img1,img2,img3,img4,img5,img6};
                images = imageElements;

                //A lista é criada passando o vetor com as images e o vetor com
                //o detalhamento das descrições de cada imagem
                //Cada posição tanto do vetor de imagens quanto do de string
                //batem como informação dos elementos.
                //Exempo:
                // imageElements[0] devolve a imagem do batman
                // stringElements[0] devolve a descrição da imagem: Batman
  list = new List("List + Image", List.IMPLICIT, stringElements, imageElements);

                //Adicionando dois comandos: o primeiro para selecionar um elemento
                //E o segundo para sair da App
  select = new Command("Select", Command.SCREEN, 0);
  exit = new Command("Exit", Command.EXIT, 0);
  list.addCommand(select);
  list.addCommand(exit);

                //Configurando o listener para que os eventos de clique possam
                //ser escutados
  list.setCommandListener(this);

                //Com essa linha faço com que meu display, minha tela exiba
                //a lista.
  display.setCurrent(list);
 }

 public void pauseApp() {}

 public void destroyApp(boolean unconditional){
  notifyDestroyed();
 }

        //Neste método tratamos os eventos com elementos da tela
 public void commandAction(Command c, Displayable s){
                // Em index recupero o index do item selecionado na lista
  int index = list.getSelectedIndex();

                //Se for clicado o botão select ou o usuário tiver clicado
                //diretamente em um elemento na lista
  if (c == select || c == List.SELECT_COMMAND) {
                        //Irei exibir a tela de detalhamento das informações
                        //Veja o método showDetailsImage aqui neste arquivo.
                        Display.getDisplay(this).setCurrent(showDetailsImage(list.getString(index),images[index] ));
  } else if(c == exit){
   destroyApp(true);
  } else if(c == back){
                        //Se tiver selecionado botão voltar
                        //configuramos o display para exibir novamente a losta
                        Display.getDisplay(this).setCurrent(list);
                }
 }
        /*
         * Método que retorna uma Screen, uma tela
         * Recebe dois parâmetros
         * @param String text Texto descritivo da Imagem ou elemento da lista
```


(C) Copyright Nokia 2013. All rights reserved.

```
         * @param Image img Imagem do elemento selecionado na lista em tamanho maior
         */
         public Screen showDetailsImage(String text, Image img){
             Form detailImage = new Form("Detalhes");

             ImageItem imgItem = new ImageItem("Imagem", img,ImageItem.LAYOUT_CENTER | ImageItem.LAYOUT_TOP, text);
             detailImage.append(imgItem);

             StringItem description = new StringItem("Descrição:", text);
             detailImage.append(description);

             back = new Command("Voltar", Command.SCREEN, 1);
             detailImage.addCommand(back);

             detailImage.setCommandListener(this);

             return detailImage;
         }
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.