

Prática de Programação J2ME (19)

Especialização em Desenvolvimento Web com Interfaces Ricas

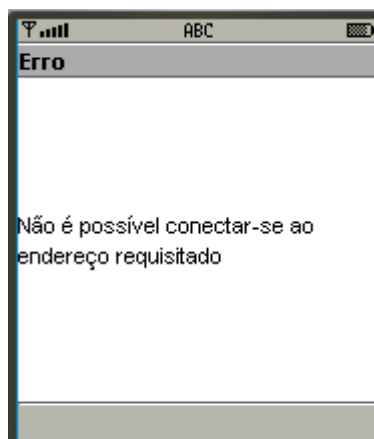
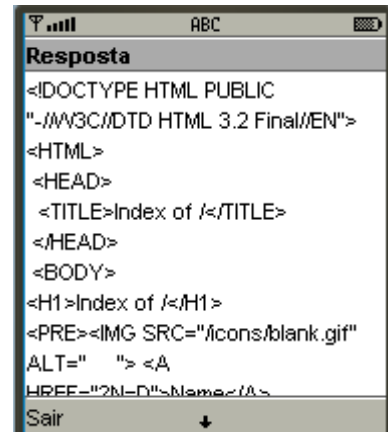
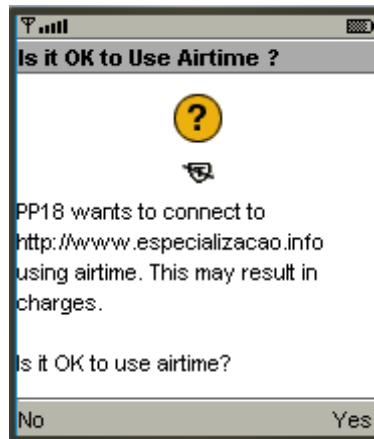
www.especializacao.info

Crie a aplicação correspondente aos itens seguintes. Execute o programa e observe os resultados.

1. Crie um MIDlet simples, chamado `WebMIDlet.java`, que ao ser executado, mostra um formulário de entrada de endereço web;
2. O formulário de entrada de endereço web `GetForm.java` deverá conter um `TextField` para entrada do site e dois comandos: “Sair” (`Command.BACK`) e “OK” (`Command.OK`);
3. Ao clicar em “Sair”, a aplicação deverá ser encerrada;
4. Ao clicar em “OK”, a aplicação se conecta ao site, recebendo os dados necessários;
5. Caso a conexão com o site não tenha sucesso, um alerta de erro (`AlertType.ERROR`) deve ser mostrado, indicando o fracasso da operação;
6. Caso a conexão tenha sucesso, é mostrado um Formulário simples e com título “Resultado”, que mostra em seu conteúdo o texto(ou código HTML) relativo ao site. Não se esqueça de incluir o comando “Sair” em tal formulário;
7. A conexão com o endereço requisitado deve ser feita através da classe `Connector` (`javax.microedition.io`) e abrindo uma `InputStream` (`java.io`), que deve ser obtida através de conexão feita pela classe `HttpConnection` (`javax.microedition.io`);
8. A conexão pela classe `HttpConnection` deverá ser feita através da passagem dos seguintes parâmetros:
 - 8.1. `RequestMethod = HttpConnection.GET`;
 - 8.2. `User-Agent = "Profile/MIDP-2.0 Configuration/CLDC-1.1"`;
 - 8.3. `Content-Language = pt-BR`;
 - 8.4. `Content-Type = text/html`;

Resposta da Prática de Programação J2ME (19)

TELA(S)



CÓDIGO EM JAVA

RemoteUtilities.java

```
package PP19.util;

import java.io.*;
import javax.microedition.io.*;

public class RemoteUtilities {

    public static byte[] getUrlAsString(InputStream is) {
        byte[] buffer = null;
        int size = 50000;
        try {
            buffer = new byte[size];
            int length = 0;

```

Prática de Programação J2ME

```
        int start = 0;
        while ((length = is.read(buffer, start, size)) > 0) {
            start += length;
            size -= length;
        }
        is.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return buffer;
}

public static InputStream obterIsDeUrl(String url) {
    HttpURLConnection conexao;
    InputStream is;
    try {
        conexao = (HttpURLConnection) Connector.open(url);
        conexao.setRequestMethod(HttpURLConnection.GET);
        conexao.setRequestProperty("User-Agent",
            "Profile/MIDP-2.0 Configuration/CLDC-1.1");
        conexao.setRequestProperty("Content-Language", "pt-BR");
        conexao.setRequestProperty("Content-Type",
            "text/html");
        is = conexao.openInputStream();
    } catch (Exception e) {
        return null;
    }
    return is;
}
}
```

GetForm.java

```
package PP19;

import java.io.InputStream;
import javax.microedition.lcdui.*;
import PP19.util.RemoteUtilities;

public class GetForm extends Form implements CommandListener, Runnable{

    TextField site;
    Command ok, sair;
    WebMIDlet controlador;

    public GetForm(WebMIDlet controlador){
        super("Pedido de conteúdo");

        this.controlador = controlador;

        site = new TextField("Endereço", "", 40, TextField.URL);
        sair = new Command("Sair", Command.EXIT, 0);
    }
}
```

Prática de Programação J2ME

```
ok = new Command("Ok", Command.OK, 1);

append(site);
setCommandListener(this);
addCommand(ok);
addCommand(sair);
}

public void commandAction(Command arg0, Displayable arg1) {
    if(arg0 == sair){
        controlador.destroyApp(true);
    }else{
        new Thread(this).start();
    }
}

public void run() {
    removeComandos();
    String endereco = site.getString();

    InputStream is = RemoteUtilities.obtemIsDeUrl(endereco);
    if(is == null){
        addComandos();
        mostrarErro("Não é possível conectar-se ao endereço " +
                    "requisitado");
    }else{
        String resposta = new String(
            RemoteUtilities.getUrlAsString(is)).trim();
        Form resultado = new Form("Resposta");
        resultado.append(resposta);
        resultado.addCommand(sair);
        resultado.setCommandListener(this);
        controlador.mudarTelaPara(resultado);
        addComandos();
    }
}

private void mostrarErro(String descricao) {
    Alert alerta = new Alert("Erro", descricao, null,
                            AlertType.ERROR);
    controlador.mudarTelaPara(alerta);
}

public void removeComandos(){
    removeCommand(sair);
    removeCommand(ok);
}

public void addComandos(){
    addCommand(ok);
    addCommand(sair);
}
}
```

WebMIDlet.java

```
package PP19;

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

public class WebMIDlet extends MIDlet{

    private Form formulario;
    private Display display;

    public void startApp() {
        display = Display.getDisplay(this);
        formulario = new GetForm(this);
        display.setCurrent(formulario);
    }

    public void pauseApp() { }

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

    void mudarTelaPara(Displayable tela) {
        display.setCurrent(tela);
    }
}
```