

Tutorial J2ME – Parte 3

Introdução

Como pudemos aprender nos tutoriais anteriores, os dispositivos celulares podem implementar em seu sistema o KVM, a máquina virtual que roda aplicações J2ME. A API de programação para aparelhos celulares está baseada em CLDC e MIDP. Para maiores detalhes veja o tutorial – parte 1

(http://sites.uol.com.br/helyrosa/rodrigo_garcia_j2me/rodrigo_garcia_j2me.html)

Também pudemos ver um exemplo de aplicação em J2ME que escreve na tela do celular a frase "Hello World!", conforme a figura abaixo:



O exemplo da aplicação "Hello World!" e as estruturas de programação utilizadas podem ser encontrados no tutorial – parte 2

(http://sites.uol.com.br/helyrosa/r_martins_j2me/r_martins_j2me.html)

Neste capítulo 3 do tutorial vamos abordar como compilar e testar a aplicação "Hello World!" utilizando o Wireless Toolkit que a Sun Microsystems disponibiliza em seu site (para maiores detalhes de como obter as ferramentas, veja o tutorial – parte 1).

Suponhamos que o Toolkit foi instalado no diretório "C:\J2MEwtk". Dentro desse diretório teremos a seguinte estrutura de arquivos e diretórios referente ao próprio Toolkit:

- Arquivo: **BinaryLicense.html** contem o contrato da licença de uso do Toolkit.
- Arquivo: **BinaryReleaseNotes.html** contem notas sobre a versão instalada do Toolkit.
- Arquivo: **index.html** página HTML que aponta para a documentação do Toolkit.
- Diretório **appdb** contem arquivos de dados tais como informações RMS de cada Midlet.
- Diretório **apps** contem as aplicações demonstração e novas aplicações/projeto que são criados adicionalmente no KToolbar. É aqui que será criada a estrutura para nossa aplicação "Hello World!".
- Diretório **bin** contem arquivos Batch e outros executáveis usados pelo Toolkit.
- Diretório **docs** contem documentação e guias de uso do Toolkit e das APIs do J2ME.
- Arquivo **lib\midpapi.zip** contem as classes com as APIs CLDC e MIDP. Esses arquivos são usados durante a compilação dos arquivos fonte (*.java) e na pré-verificação do byte-code das classes geradas por esse fontes (*.class).
- Diretório **sessions** contem dados de monitoração da execução das aplicações no simulador. Com esses dados é possível monitorar o uso da memória, rede e outros durante a execução de um Midlet.

Para uso neste tutorial é necessário que o código a seguir seja copiado para um arquivo texto sem formatação (do tipo gerado com NotePad) e salvo com o nome "HelloJ2ME.java". Note que a linguagem JAVA faz distinção entre minúsculas e maiúsculas e portanto deve-se tomar especial atenção ao nome do arquivo que tem que ser obrigatoriamente igual ao nome da classe que nele é definida.

O código do programa que será utilizado em este tutorial segue abaixo:

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

public class HelloJ2ME extends MIDlet implements CommandListener
{
    private Display display;
    private TextField t;
    private Command comandoDeSaida;
    private Form formularioPrincipal;

    public HelloJ2ME() {
        display = Display.getDisplay(this);
        formularioPrincipal = new Form("HelloJ2ME");
        comandoDeSaida = new Command("Exit", Command.SCREEN,1);
        t = new TextField("Text","Hello World!",15,TextField.ANY);
        formularioPrincipal.addCommand(comandoDeSaida);
        formularioPrincipal.append(t);
        formularioPrincipal.setCommandListener(this);
    }

    public void startApp() {
        display.setCurrent(formularioPrincipal);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean incondicional) {
    }

    public void commandAction(Command comando, Displayable s) {
        if (comando == comandoDeSaida) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}
```

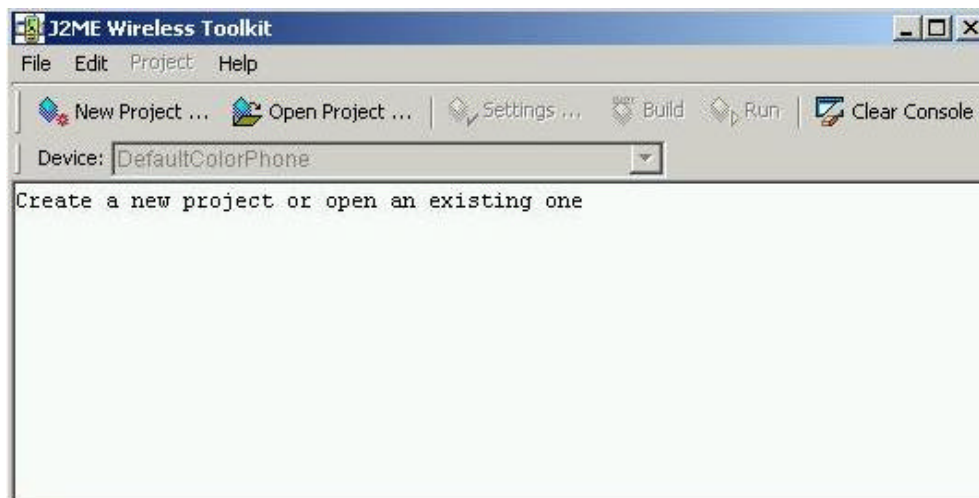
Como usar o Wireless Toolkit?

Uma vez instalado conforme as instruções descritas no site da SUN, podemos iniciar o desenvolvimento de aplicações baseadas em J2ME. No nosso tutorial vamos demonstrar o uso do Wireless Toolkit através do programa "Hello World!".

O primeiro passo é iniciar o KToolbar que é onde criaremos um projeto. A figura abaixo ilustra onde pode ser localizada a aplicação em ambientes Windows.



Uma vez iniciado o aplicativo KToolbar, o mesmo se apresentará como a seguir.



Após iniciar o KToolbar devemos abrir um novo projeto. Na tela da aplicação KToolbar existe o botão "New Project ...". Aperte esse botão para que seja aberto um formulário que solicitará que seja dado um nome de projeto (qualquer) e o nome da classe MIDlet que rodará o programa J2ME. No nosso exemplo, o nome do projeto é "Hello World!" e conforme descrito no código fonte do programa, a classe é chamada "HelloJ2ME".



Ao apertar o botão "Create Project" o sistema automaticamente criará uma estrutura de diretórios referentes ao projeto dentro do diretório de instalação do J2ME Wireless Toolkit. Essa estrutura estará abaixo do diretório "Apps\<Nome do Projeto>" que fica dentro da estrutura de diretórios do Toolkit (por exemplo "C:\J2MEwtk\Apps\Hello World!\"). Nela teremos os seguintes diretórios:

- **\Src** – nele colocaremos o código fonte de nossa aplicação, ou seja, o arquivo "HelloJ2ME.java".
- **\Bin** – nele o Toolkit criará os arquivos .JAR e .JAD que são necessários para poder executar a aplicação em um celular ou mesmo no simulador.
- **\Res** – nele são colocados todos os arquivos adicionais que são necessários para que a aplicação seja executada. Um exemplo típico é uma imagem no formato PNG com o ícone da aplicação. Todos os arquivos que forem colocados nesse diretório serão empacotados dentro do .JAR e serão acessíveis pela aplicação J2ME. No nosso exemplo não colocaremos nada lá.

Ao pressionar o botão "Create Project", o KToolbar criará os diretórios acima descritos e também o próprio diretório do projeto "Hello World!". Porém antes o Toolkit pedirá para que sejam confirmadas as informações do descritivo do Projeto/aplicação. Essas informações ou atributos serão o conteúdo do arquivo .JAD (Java Descriptor) a ser criado. No nosso caso não nos preocuparemos com esses dados e simplesmente apertaremos o botão "OK". O resultado aparece a seguir.



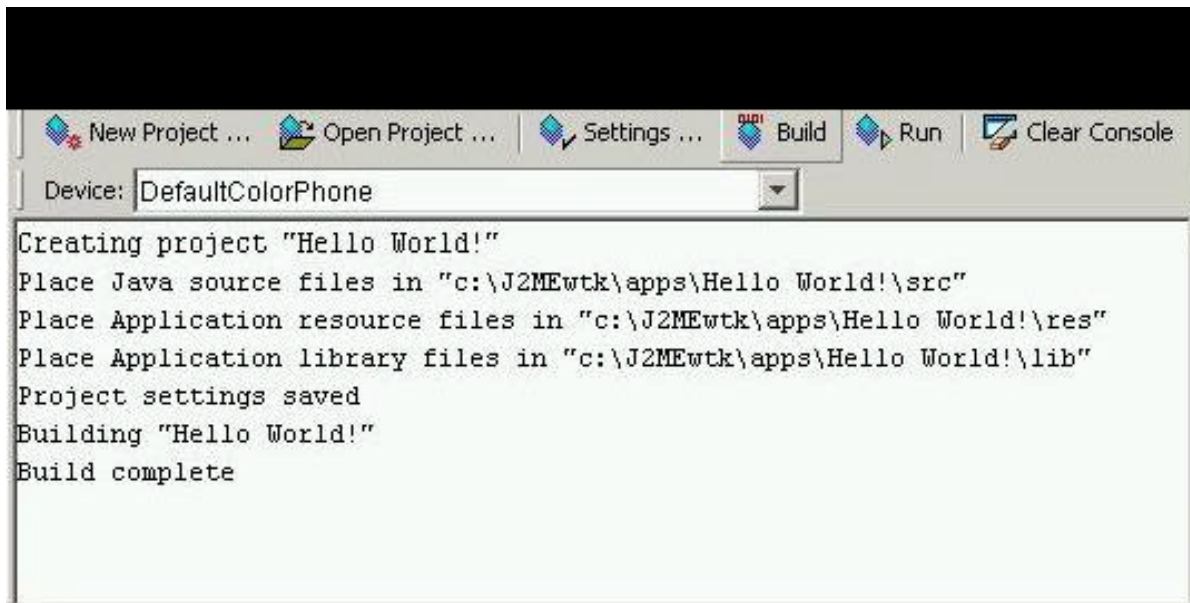
Após apertar o "OK", o Toolkit indicará no seu console que os diretórios do projeto foram criados conforme a figura abaixo.



Nesse momento o Toolkit nos sugere que os arquivos fonte sejam colocados no diretório "...\apps\Hello World!\src\". Nesse momento devemos copiar o arquivo "HelloJ2ME.java" para esse diretório.

A partir de agora estamos prontos para construir a aplicação!

Basta apenas apertar o botão "Build" no KToolbar que o Toolkit compilará os arquivos fonte e criará os arquivos .class que serão usados para rodar o simulador. Abaixo segue a ilustração da construção do aplicativo "Hello World!".



Caso não ocorra nenhum problema de compilação teremos a mensagem "Build Complete", ao final das mensagens, indicando que está tudo pronto para executar a aplicação "Hello World!" no simulador. Caso ocorram erros de compilação os mesmos serão apresentados no console do Toolkit.

No nosso caso, o programa está depurado e o código fonte está correto. Portanto não há nenhum problema de compilação. Neste caso estamos prontos para executar a aplicação. Para isso basta apertar o botão "Run" no KToolbar. Ao fazer isso será aberta uma nova janela com o simulador escolhido. No nosso exemplo é o simulador genérico colorido, conforme a figura abaixo.



A figura acima mostra o estágio inicial do simulador que permite que seja escolhido qual Midlet executar. Nosso caso só há um Midlet especificado no arquivo .JAD, portanto temos apenas que pressionar o SoftKey para lançar a aplicação "Hello World!" (botão em destaque logo abaixo da palavra "Launch"). O resultado da execução do "Hello World!" segue abaixo.



Uma vez que a aplicação tenha executado a contento no simulador, é possível criar o arquivo .JAR (Java Archive) que é, na verdade, um arquivo compactado no formato PkZip com todas as classes e arquivos do diretório "...\\res" (resources).

Para criar-se o .JAR basta selecionar "Project → Package" no menu do KToolbar. Esse arquivo somente é necessário para executar o programa no próprio aparelho celular.

Com este tutorial, o estudante será capaz de compilar e simular seus programas desenvolvidos com a linguagem J2ME para aparelhos celulares. No próximo e último capítulo do tutorial, apresentarei uma versão mais sofisticada do programa "Hello World!", onde mostrarei como fazer animações simples usando caracteres de texto, o que pode ser melhorado, futuramente pelo próprio estudante, para fazer animações com imagens.

Até lá!

Sobre o Autor:

Rodrigo P. Garcia Corbera tem 10 anos de experiência em arquitetura e projetos de tecnologia em telecomunicações. Mais recentemente, trabalhou na Lucent Brasil, onde atuou como gerente comercial de soluções de Messaging no mercado de operadores móveis, gerente da área de consultoria de redes de dados e óticas liderando um time com 25 consultores espalhados pela América do Sul e também, foi por dois anos, responsável por gerenciar a área técnico-comercial de integração de tecnologia para ofertas de redes de dados da Lucent. Antes disso, trabalhou no projeto Pay-Per-View da Globo Cabo como consultor contratado através da AL&CA e como Executivo de Vendas da Hughes Network Systems. Rodrigo tem formação em engenharia da computação pela UNICAMP e é fluente em português, espanhol e inglês.

Contato via e-mail: rgarcia@cy.com.br