

Uma Introdução ao J2ME

Computação Móvel (MAC5743/MAC330)

DCC-IME-USP

Prof. Alfredo Goldman

Monitores: Rodrigo Barbosa

Daniel Cordeiro

Visão Geral do Java 2 (1)

- A plataforma Java 2 engloba três elementos:
 - A linguagem Java.
 - A máquina virtual Java.
 - Um conjunto extenso de APIs

Visão Geral do Java 2 (2)

- Java propõe-se a servir uma grande variedade de hardware, desde smart cards até servidores corporativos. Sendo assim, a plataforma Java possui três versões:
 - Java 2 Standard Edition (J2SE): Uma versão desenvolvida para desktops.
 - Java 2 Enterprise Edition (J2EE): Ampla plataforma para aplicações do tipo multi-usuário corporativas. Baseada em J2SE, com adição de APIs para computação do lado do servidor.

Visão Geral do Java 2 (3)

- Java 2 Micro Edition (J2ME): Conjunto de tecnologias desenvolvidas para dispositivos pequenos, como pagers, celulares, etc. J2ME usa subconjuntos de componentes J2SE, tais como máquinas virtuais menores, e APIs mais enxutas.

Visão Geral do J2ME (1)

- Diferentemente do J2SE, J2ME não é um software, nem uma especificação única.
- J2ME é uma plataforma, uma coleção de tecnologias e especificações que são designadas para os diferentes setores do mercado de dispositivos pequenos.
- Como J2ME abrange tantos tipos de dispositivos, não faria sentido tentar-se criar uma solução única.

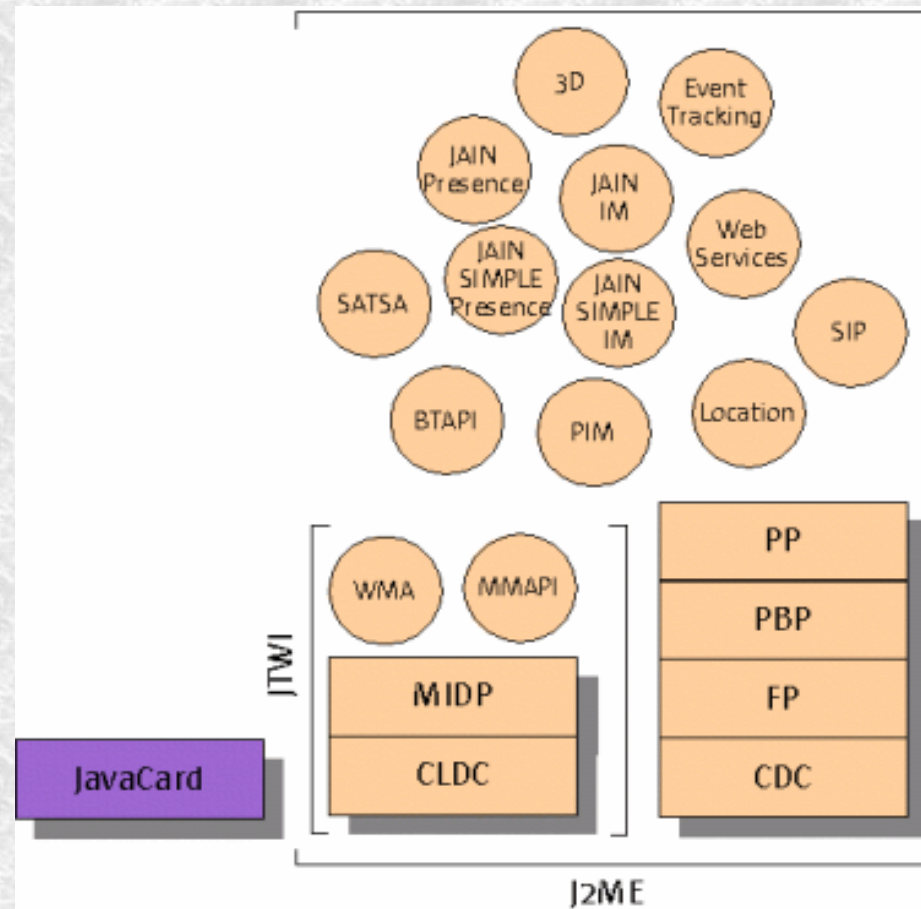
Visão Geral do J2ME (2)

- Pelas razões citadas, J2ME é dividido em:
 - Configurações: Uma configuração fornece um ambiente Java completo. O conjunto de classes é pequeno e deve ser estendido por perfis. Em particular, uma configuração não implementa classes de interface com o usuário. Uma configuração é dividida em:
 - Uma máquina virtual Java (VM) para executar bytecode.
 - Código proprietário que faz interface com o sistema provido pelo dispositivo.
 - Um conjunto de classes núcleo.

Visão Geral do J2ME (3)

- Perfil: Adiciona classes para domínios específicos, para fornecer funcionalidades e prover usos específicos de um dispositivo. Por exemplo, a maioria dos perfis define classes de interface com o usuário para a construção de aplicações interativas.
- Pacotes opcionais: Provêem funcionalidades que podem não estar associadas com uma configuração específica ou com um perfil, visto que um dispositivo que atende a um perfil deve atender a **todo** ele. Um exemplo de pacote opcional é a API Bluetooth. Um pacote opcional pode ser implementando em inúmeras combinações de configurações e perfis.

O Universo do J2ME (1)



O Universo do J2ME (2)

- Connected, Limited Device Configuration (CLDC): Para dispositivos sem fio com conexões a rede intermitentes, como pagers, celulares e PDAs.
- Connected Device Configuration (CDC): Para dispositivos maiores em termos de memória e processamento com conexões de rede robustas.
- Os pacotes opcionais podem ser implementados tanto para o ramo do CLDC como para o ramo do CDC.

CLDC (1)

- A especificação do CLDC define três coisas:
 - As capacidades de uma pequena máquina virtual.
 - Um pequeno subconjunto das classes do J2SE.
 - Um conjunto de APIs para entrada e saída, denominado Generic Connection Framework.
- CLDC não define como as aplicações são carregadas e, depois, ativadas ou desativadas. Isso é definido pelo perfil.

CLDC (2)

- A máquina virtual do CLDC têm várias restrições que fazem frente às limitações de memória e de suprimento de energia:
 - Ausência do tipo ponto flutuante.
 - Ausência de finalização de objetos ou referências fracas.
 - Ausência de JNI e reflexão.
 - Ausência de daemon threads.
 - Ausência de carregadores de classes definidos pelo usuário.
- CLDC (1.1) relaxa algumas dessas restrições.

CLDC (3)

- O subconjunto do J2SE fornecido pelo CLDC é composto por algumas classes (não todas) dos seguintes pacotes:
 - java.lang
 - java.io
 - java.util
- Algumas classes são subconjuntos das classes equivalentes no J2SE.

MIDP (1)

- MIDP: Mobile Information Device Profile.
- MIDP foi criado para dispositivos com as seguintes características:
 - Dispositivos com memória suficiente para rodar aplicações MIDP.
 - Uma tela com, ao menos, 95 pixels de largura e 56 pixels de altura, colorida ou monocromática.
 - Teclado ou touch-screen.
 - Capacidades de envio e recebimento via rede.

MIDP (2)

- A configuração usada pelo MIDP é a CLDC. No entanto, como a CLDC é um subconjunto do CDC, MIDP também pode rodar sobre CDC.
- APIs adicionadas pelo MIDP:
 - Suporte a ciclo de vida de aplicações, de maneira similar a como os applets são definidos no J2SE.
 - Armazenamento persistente de dados.
 - Conectividade de rede baseada em HTTP, construída sobre o Generic Connection Framework do CLDC.
 - Suporte simples a interface com o usuário.

MIDP (3)

- O ciclo de vida de uma aplicação MIDP é semelhante ao dos applets:
 - O ponto de entrada da execução não é via método main.
 - Seu ponto de entrada é uma classe que estende a classe `javax.microedition.midlet.MIDlet`
- Um aplicação MIDP é composta de MIDlets, classes do usuário, imagens e recursos, juntos com um arquivo de descrição da aplicação e um arquivo manifest do MIDlets, tudo isso empacotado em um JAR.

Exemplo de Código de MIDlet (1)

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

public class HelloMIDlet
    extends MIDlet
    implements CommandListener {

    private Form mMainForm;

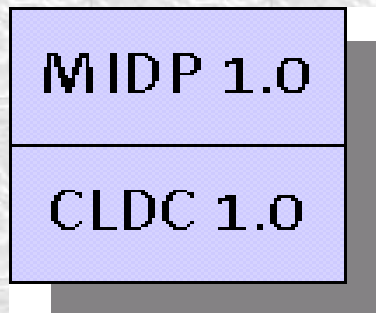
    public HelloMIDlet() {
        mMainForm = new Form("HelloMIDlet");
        mMainForm.append(new StringItem(null, "Hello, MIDP!"));
        mMainForm.addCommand(new Command("Exit", Command.EXIT,
0));
        mMainForm.setCommandListener(this);
    }
}
```

Exemplo de Código de MIDlet (2)

```
public void startApp() {  
    Display.getDisplay(this).setCurrent(mMainForm);  
}  
public void pauseApp() {}  
  
public void destroyApp(boolean unconditional) {}  
  
public void commandAction(Command c, Displayable s) {  
    notifyDestroyed();  
}  
}
```

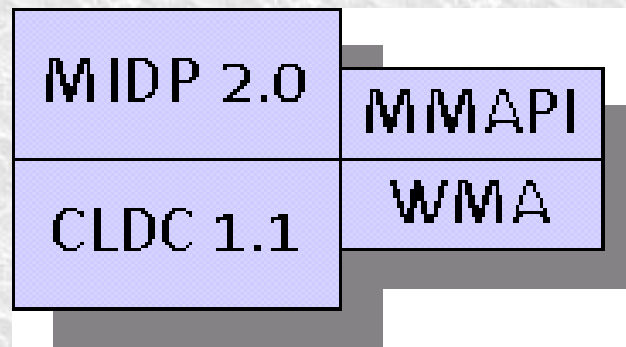
Pilhas e JSR 185 (1)

- Os dispositivos, normalmente, implementam uma pilha de software, a qual consiste de uma configuração, um perfil e APIs opcionais. As primeiras gerações de telefones celulares implementavam a seguinte pilha:



Pilhas e JSR 185 (2)

- Com a abundância de configurações, perfis e pacotes opcionais, como um desenvolvedor sabe o que esperar de um dispositivo? O JSR 185, Tecnologia Java para Indústria de Equipamentos Sem-fio (JTWI), enfrenta esse problema com a adição de outros blocos para a criação de um ambiente de aplicações completo:



O Escopo da Tecnologia Sem-fio Java (1)

- A Tecnologia Sem-fio Java e J2ME não são a mesma coisa:
 - J2ME engloba mais que dispositivos sem-fio. Algumas partes do J2ME são designadas somente para dispositivos sem-fio, outras partes não – por exemplo, dispositivos CDC, muitas vezes, possuem conexões Ethernet padrões.
 - A Tecnologia Sem-fio Java não é confinada somente ao J2ME. Você pode ter um laptop ou palmtop rodando aplicações J2SE, conectando-se via rede 802.11.

O Escopo da Tecnologia Sem-fio Java (2)

- MIDP não constitui todo o J2ME:
 - MIDP foi o primeiro perfil terminado e teve a primeira base de dispositivos lançada no mercado.
 - MIDP é apenas um ramo do J2ME.
- MIDP não constitui toda tecnologia sem-fio:
 - A plataforma Java oferece muitas escolhas para a programação sem-fio.

Por que Usar Java para Desenvolvimento Sem-fio?

- A plataforma Java é segura.
- A plataforma Java encoraja desenvolvimento robusto.
- Portabilidade.
- Assim, como existe um arcabouço para testar aplicações Java, o Junit, existe também um arcabouço para testar aplicações J2ME (J2MEUnit).

Bibliografia

- Essa apresentação é fortemente baseada nos seguintes documentos:
 - <http://developers.sun.com/techttopics/mobility/getstart/>
 - <http://www.ericgiguere.com/articles/index.html>
 - <http://j2meunit.sourceforge.net/>