

Java 2 Micro Edition (J2ME)

por

Filipe Gonçalves Barreto de Oliveira Castilho

Departamento de Engenharia Informática

Universidade de Coimbra

3030 Coimbra, Portugal

fgonc@student.dei.uc.pt

Resumo: *Apresenta-se de uma forma um pouco técnica o funcionamento da plataforma Java 2 Micro Edition. Para esse efeito, descrevem-se as características gerais da plataforma, seguindo-se uma análise pormenorizando do seu funcionamento na área dos dispositivos móveis. Finaliza-se com um exemplo simples do código de um programa para esta plataforma. Pretende-se com a abordagem seguida, dar a conhecer o J2ME e motivar o leitor a aprofundar os seus conhecimentos.*

Palavras-Chave: *Java 2 Micro Edition, J2ME, KVM, CLDC, MIDP, MIDLet.*

1. Introdução

Hoje em dia existem vários dispositivos móveis e engenhocas que se apresentam não apenas com design atractivo mas que estão, também, equipados com mini cérebros, tornando-os mais inteligentes e mais poderosos que nunca. Entre estes encontram-se os telefones móveis com as potencialidades de PDAs (computadores de mão) e PDAs com as potencialidades de telefones móveis que começaram a invadir o mercado. Num futuro próximo seremos capazes de fazer uma apresentação directamente dos nossos telefones móveis ou usá-los como um rato sem fios para mais facilmente navegarmos na Internet a partir do nosso portátil.

Os dispositivos estão a ficar cada vez mais inteligentes graças aos minicomputadores que têm embutidos. As estações de serviço, os tabliers de automóveis e as câmaras digitais são também exemplos dos dispositivos inteligentes actuais. Frigoríficos com acesso à Internet, microondas capazes de ir buscar receitas à Internet e aparelhagens que desligam o som quando o telefone toca fazem parte dos muitos dispositivos em vista para um futuro próximo. Existem excelentes páginas na Internet para aprofundar este tema (<http://java.sun.com/j2me/> - página oficial do J2ME na Sun Microsystems [1]).

O objectivo deste estudo centra-se na descrição das principais características do J2ME (tipos de configurações que existem e seus perfis), destacando-se a área das comunicações móveis e computadores de mão que usam a configuração *Connected Limited Device Configuration* (CLDC). Descrevem-se também as características do J2ME incluídos na configuração.

2. Java 2 Micro Edition (J2ME)

Durante os últimos anos tem-se assistido a grandes avanços nas tecnologias móveis e não só. Este crescimento iniciou o começo de uma nova era no mundo da computação. Tal avanço veio a incentivar a necessidade de estabelecer requisitos mínimos necessários para mobilizar esta tecnologia. O requisito mais importante foi encontrar um ambiente único e flexível que pudesse ser usado em todos estes dispositivos tendo em conta as limitações de cada um.

Apercebendo-se desta necessidade e, reconhecendo que uma só plataforma não era compatível com todos os tipos de computadores, a *Sun Microsystems* introduziu em 1999 a plataforma *Java 2 Micro Edition*, que era constituída por um conjunto de especificações da plataforma Java para os dispositivos mais pequenos, que vão dos pagers e telefones móveis até aos receptores de televisão e sistemas de navegação dos carros bem como os dispositivos com tecnologia digital embutida (microondas, frigoríficos, televisões, entre outros). Desde a sua introdução, o J2ME tem vindo a evoluir de acordo com a *Java Community Process*, que é uma organização internacional de programadores cujo objectivo é rever e patentear novas tecnologias Java, especificações e compatibilidade entre vários *kits* de desenvolvimento [1].

O J2ME teve origem numa reorganização da plataforma do Java. Daqui resultaram três edições diferentes: o *Java 2 Enterprise Edition (J2EE)*, *Java 2 Standard Edition (J2SE)* e o *Java 2 Micro Edition (J2ME)*, figura 1. Cada uma destas edições é composta por diferentes tipo de utilitários dirigidos para produtos específicos [1].

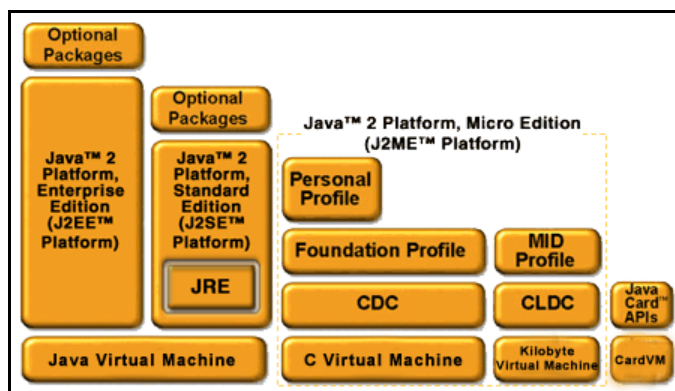


Figura 1 – Um Java, três edições

O J2ME é composto por três camadas (figura 2): a máquina virtual do Java (*Java Virtual Machine*), Configurações (*Configurations*) e Perfis (*Profiles*) [2, 3].

A primeira camada (*Java Virtual Machine*) implementa uma máquina virtual Java que é específica para cada sistema operativo de um determinado dispositivo e que suporta uma determinada configuração.

A camada das configurações define um conjunto mínimo de opções na máquina virtual e no núcleo das bibliotecas Java disponíveis para uma categoria particular de um dispositivo. Estes representam um mercado específico e podem ser vistos como o conjunto mínimo de funções que estão disponíveis em todos os dispositivos da mesma categoria.

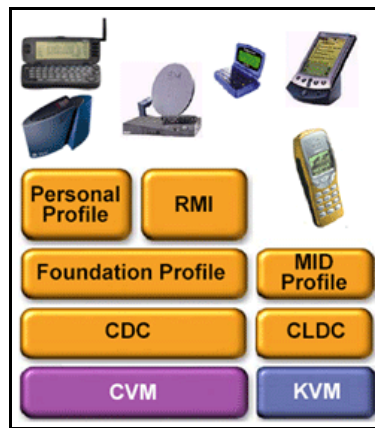


Figura 2 – Java 2 Micro Edition

A camada dos perfis define o conjunto mínimo de aplicações programáveis, disponíveis num grupo particular de dispositivos. Os perfis têm duas funções: especialização do dispositivo e a sua portabilidade. Um programa deve funcionar noutra dispositivo diferente, que use o mesmo perfil.

A configuração para dispositivos ditos fixos, com recursos limitados designa-se por, *Connected Device Configuration* (CDC). Exemplos destes dispositivos são os receptores de televisão (como os de satélite e os de cabo). A configuração CLDC é um subconjunto da anterior, e este facto garante a compatibilidade de ambos. A figura 3 mostra a relação que existe entre as duas configurações [2].

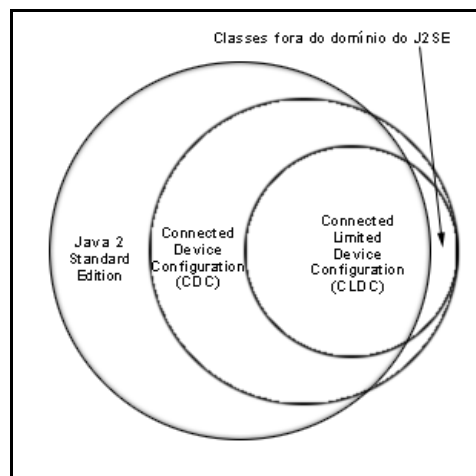


Figura 3 – Relação entre o J2SE e as configurações CDC e CLDC

A configuração *Connected Limited Device Configuration* (CLDC) aplica-se a dispositivos portáteis. Estes dispositivos (telemóveis e os pagers) têm uma memória de 160 a 512 Kbytes e como máquina virtual usam a *Kilobyte Virtual Machine* (KVM) e um processador de 16 ou 32 bit, funcionando numa rede sem fios com baixa largura de banda. O único perfil disponível para estes dispositivos é o *Mobile Information Device Profile* (MIDP) [4].

2.1. Kilobyte Virtual Machine (KVM)

O KVM é uma nova máquina virtual desenvolvida para dispositivos com recursos muito limitados. O seu tamanho vai desde os 40 aos 80 Kbytes, daí o K de Kilobyte em KVM. A ideia de

desenvolver o KVM teve origem num projecto chamado *Spotless System*, que tinha como função criar um motor para um computador de mão (PDA). A equipa responsável por esse projecto, rapidamente se apercebeu que o tamanho do sistema dependia directamente do tamanho das suas bibliotecas, havendo então, necessidade de reduzir o tamanho do sistema retirando tudo o que era considerado não essencial ou crítico para o funcionamento do sistema [2, 3].

Apenas uma pequena parte do código do KVM é dependente da plataforma onde corre, mas está isolado numa zona de fácil alteração. Para implementar o KVM num dispositivo, é necessário alterar apenas estas dependências. Ao contrário dos computadores normais, um dispositivo móvel não possui uma linha de comandos ou uma interface especial onde possa ser executado o programa. Para o conseguir usa-se o *Java Application Manager* (JAM) que serve de interface entre o KVM e o sistema operativo do dispositivo. O JAM executa a aplicação J2ME por cima do KVM, que é previamente descarregada de um servidor http onde esta se encontra. É também incluído um ficheiro *metadata* que contém informações que o JAM usa para poder descomprimir e instalar a aplicação. A esta aplicação dá-se o nome de *MIDlet* [3].

2.2. Connected, Limited Device Configuration (CLDC)

A configuração CLDC define um padrão na plataforma Java para os dispositivos com recursos limitados e, activa de forma dinâmica a distribuição das aplicações e conteúdos para estes dispositivos. A configuração é limitada a um espaço de cerca de 128 Kbytes e é desenhada para correr em vários dispositivos de pequeno porte sem que o programador se tenha de preocupar com o sistema operativo em que a aplicação vai executar. O CLDC é implementado como um conjunto adicional de classes contido numa *package* separada (a *java.io*, *java.lang*, *java.util* e a *javax.microedition.io* [5]). Tal facto facilita a portabilidade do CLDC para as diferentes plataformas.

A configuração CLDC necessita que o JVM possa identificar e rejeitar as classes que sejam inválidas. As formas padrão para fazer este tipo de verificações são muito dispendiosas em termos de memória. Para reduzir as verificações no lado do cliente, o CLDC usa um processo de pré-verificação e verificação, onde parte é feita do lado do servidor antes de se proceder ao descarregamento da aplicação CLDC *MIDlet* para o dispositivo móvel. Este procedimento de pré-verificação gera um ficheiro chamado *stackmap* que é descarregado juntamente com a *MIDlet* para o dispositivo. Este ficheiro contém informações acerca da pré-verificação, nomeadamente sobre o estado válido ou inválido da aplicação. Qualquer aplicação para correr nos dispositivos com a configuração CLDC precisa de estar na forma de JAR (ficheiro comprimido com todas as componentes da aplicação) [2, 5].

2.3. Mobile Information Device Profile (MIDP)

A MIDP é um conjunto de bibliotecas Java que juntamente com o CLDC fornece os requisitos mínimos para a construção de uma aplicação J2ME dirigida para telemóveis, PDAs e pagers com duas vias de comunicação. Este é o único perfil completamente especificado para a configuração CLDC. Existe outro exclusivo para PDAs, mas ainda se encontra em desenvolvimento. A MIDP contém bibliotecas com classes relacionadas com interfaces, armazenamento de dados, comunicações e

aplicações modelo. Este perfil também inclui um conjunto de requisitos, dirigido aos fabricantes caso eles queiram que o seu dispositivo seja certificado, para correr com as configurações CLDC e com os perfis MIDP. Por exemplo, o requisito do tamanho mínimo do visor de um dispositivo que use o perfil MIDP terá de ser de 96x54 pixels, a profundidade do ecrã de 1 bit e o tamanho de cada pixel deverá ser 1x1. Outro requisito deste perfil é ter no mínimo 128 Kbytes de memória RAM [4].

Para a distribuição de aplicações (*MIDlets*) por conta de outrem é necessário que os seus programadores gerem um ficheiro *metadata*, compactem o ficheiro e os seus componentes adicionais num JAR e lhe associem um *Java Application Descriptor* (JAD - que funciona como uma espécie de identificador). O JAD contém informações que o JAM usa para configurar, verificar e instalar as *MIDlets* e por sua vez esta usa a informação durante a sua execução. Cada fabricante é livre de incluir bibliotecas específicas para aumentar a funcionalidade do seu equipamento e incluir ou criar também *MIDlets* que usem estas funcionalidades.

2.4. Exemplo de uma MIDLet

```
// OlaMundo.java

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

// Uma simples MIDLet a dizer Olá Mundo

public class OlaMundo extends MIDlet
{
    private Form mainscreen;
    private Display myDisplay;

    OlaMundo()
    {
        myDisplay = Display.getDisplay(this);
        mainscreen = new Form("Olá Mundo");
        StringItem strItem = new StringItem("Olá,", "e bem vindo!");
        mainscreen.append(strItem);
    }
    public void startApp() throws MIDletStateChangeException
    {
        myDisplay.setCurrent(mainscreen);
    }

    protected void pauseApp(){}
    protected void destroyApp(boolean unconditional) throws MIDletStateChangeException{}
}
[6]
```



Figura 4 – Nokia 6310i a executar uma MIDlet

3. Conclusão

Pretendeu-se que este trabalho proporcionasse uma familiarização com as principais características do *Java 2 Micro Edition* com pormenor para o funcionamento da configuração CLDC usada nos dispositivos móveis e que o sensibilizasse para o novo mundo de tecnologias que estão a entrar em cena. Para satisfazer este objectivo, optou-se por uma descrição geral da plataforma, seguindo-se uma análise das várias componentes da configuração CLDC e concluindo com um exemplo de um programa. Pensa-se que este estudo será uma boa referência para quem queira aprender os básicos da plataforma J2ME. Assim sendo, as informações nele contidas devem ser entendidas apenas como uma primeira abordagem para um utilizador que se queira iniciar nesta área.

Agradecimentos

O autor agradece aos pais que muito ajudaram e a todos os colegas que o incentivaram e ajudaram na pesquisa e elaboração do artigo.

Referências

- [1] Java™ 2 Platform, Micro Edition (J2ME™): <http://java.sun.com/j2me/>
- [2] Helal, S. (2002) – Pervasive Java, IEEE Pervasive Computing, 1(1) 2002, 82-85
- [3] JavaOne Session Topics: <http://servlet.java.sun.com/javaone/javaone2000/tracks.jsp#4>
- [4] Micro Java Network – The J2ME Resource: <http://www.microjava.com/>
- [5] CLDC: <http://java.sun.com/products/cldc/>
- [6] J2ME and the Mobile Information Device Profile: <http://www.developer.com/ws/article.php/630611>