

J2ME MIDP 1.0/2.0

无线设备 编程指南



周怡洋 编著

北京大学出版社
<http://cbs.pku.edu.cn>

J2ME MIDP 1.0/2.0 无线设备 编程指南

闻怡洋 编著

北京大学出版社

内 容 简 介

本书主要讲解如何利用 MIDP v2.0 在无线设备上进行编程开发,内容覆盖了开发环境的建立、界面开发、多媒体播放、数据库存储、网络开发,此外在本书的附录部分对 Nokia 智能手机的 MIDP 开发包的安装和使用等知识做了简单介绍。本书以 MIDP v2.0 为基础主讲,因为 MIDP v2.0 包含了 MIDP v1.0 的所有功能,为了减少读者在实践中的误解,在书中也对 MIDP v1.0 和 2.0 的功能进行了区分和说明。

阅读本书的读者需要有一定的 Java 基础,希望通过本书读者可以掌握 MIDP 程序的开发方法。

图书在版编目(CIP)数据

J2ME MIDP 1.0/2.0 无线设备编程指南/闻怡洋编著. —北京:北京大学出版社, 2004.7

ISBN 7-301-07501-4

I. J… II. 闻… III. JAVA 语言—程序设计—应用—移动通信—通信设备 IV. TN929.5

中国版本图书馆 CIP 数据核字(2004)第 054221 号

书 名: J2ME MIDP 1.0/2.0 无线设备编程指南

著作责任者: 闻怡洋

责任编辑: 温丹丹

标准书号: ISBN 7-301-07501-4 /TP · 0765

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750581

电子信箱: xxjs@pup.pku.edu.cn

排 版 者: 北京东方人华北大彩印中心 电话: 62754190

印 刷 者: 河北涿县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 18.75 印张 450 千字

2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

定 价: 35.00 元

前 言

Java 的出现最早大概可追溯至 1991 年 4 月份, Sun 的绿色计划(Green Project)开始着手发展消费性电子产品, 当时所使用的语言是 C / C++ 及 Oak (为 Java 语言的前身), 后因语言本身和市场的问题, 使得消费性电子产品的发展无法达到当初预期的目标, 再加上网络的兴起, 绿色计划也因此而改变发展的方向。但是 Java 技术真正的兴起却是借助于 Internet 的力量, 1995 年的夏天 Java 开始在计算机业界受到了高度关注, 接下来出现的运行于 IE 与 Netscape 之上的 Java Applet, 让原本平淡而乏味的 HTML 网页显得活泼而有生气而且让网页具有了交互性, 这令所有的人都备受鼓舞, 铺天盖地的 Applet 在互联网上出现并免费供人下载。

不过事情的发展并不总像人们想像的那样, 现在我们很难看见运行在浏览器内的 Applet 程序。但是 Java 却以异常的高速度在发展, 从浏览器中运行的 Java Applet 到 Java 桌面程序, 再到运行于服务器的 Java Servlet。现在, 我们看到了一种运行于小型设备中的 Java MIDlet, 随着智能手机的兴起, 越来越多的手机设备开始支持 Java。J2ME 规范是继 Sun 公司 J2SE、J2EE 之后在小型设备方面所推出的规范, 随着 J2ME 的出现, Java 又重新回到了消费性电子产品这一个领域。Internet 给了 Java 一张入场券, Java 凭借着自己巨大的想像力将其触角伸展到电子产品的每一个角落。

根据 Sun 的 Java 语言白皮书中的定义, Java 是一种分布式的(Distributed)、解释的(Interpreted)、健壮的(Robust)、安全的(Secure)、结构中立的(Architecture Neutral)、可移植的(Portable)、高效能的(High Performance)、多线程的(Multithreaded)、动态的(Dynamic)程序语言。Java 给开发人员带来的最大好处就是“一次编译, 处处运行(Write Once, Run Anywhere)”。用 C / C++语言进行开发的程序员都知道, 一般来讲, 这两种语言都是通过各种繁杂的宏定义来保证源代码的可移植性, 但是具体实施时让人感觉到非常繁琐而且容易出错。而 Java 程序可以在任意的平台上被移植, 无需对程序进行任何改动。如果仅仅从开发的角度来讲, 在开发桌面程序时, 也许这并非特别重要, 因为桌面操作系统基本已经被微软统治。在开发服务器程序时, 也许这同样不是特别重要, 因为并不是每个人都需要为跨平台做好准备, 而且使用 C / C++之类的开发语言也是有特定的途径解决。如果您曾经为多种小型设备编写过程序, 那么一定会感谢 J2ME 的出现。因为小型设备实在是千差万别, 而且没有统一的标准, 每针对一种新的设备进行开发时, 就需要重新进行学习。这种学习不光是对设备功能调用的学习, 而且开发方式都会发生改变。

从对开发人员的影响上来说, J2ME 的出现并不仅仅解决了代码移植的问题, 更重大的意义在于把 Java 这种语言, 这种开发方式带到了小型设备程序开发这一个领域中。虽然 J2ME 不能够定义小型设备上所有的功能, 但是设备供应商可以按照标准的方式提供 Java 包, 开发人员依然可以按照统一的方式进行程序开发和测试。

那么 J2ME 现在能够做什么? 目前国内智能手机的兴起给 Java 开辟了广阔的应用空

间,越来越多的手机开始支持 Java,从两年前对 KJava 的支持到如今对 MIDP 的广泛支持。现在很难找到一款不支持 Java 的手机,因为到现在为止手机已不仅仅是用来通话的工具。人们希望手机也能够成为一款娱乐的工具,甚至是商务办公的工具,在提供各种功能的同时也能够访问 Internet 上的各种资源,这都给 J2ME 在智能手机方面的应用留下了广阔的空间。同时无线网络和无线设备的发展也给无线设备上的软件开辟了广阔的前景,也许不久的将来人们在出差或是旅游时能够利用手机进行联网游戏,或者通过手机访问和享用各种 Internet 服务。

本书的内容集中在 J2ME 在无线设备方面的开发上。随着智能手机的普及,目前国内 J2ME 的开发正在兴起,市面上介绍 J2ME MIDP 开发知识的书籍比较少,所以希望本书能够帮助那些希望学习 MIDP 开发的人。

希望读者看完本书之后能够掌握利用 MIDP 进行程序开发的方法,从而带领读者进入手机开发的殿堂。在本书中,作者尽可能对所涉及到的内容进行详细的讲解而不是浅尝辄止,所以也希望本书能成为读者在开发中的参考书。

本书内所有例程的源代码可以通过网址 <http://www.vchelp.net/wyy/j2me/j2mecode.zip> 获得。

此外为了减少本书的代码量(而且另一个综合示例程序带有比较强的个人色彩),所以将另一个综合实例程序放在作者的个人网站上,读者可以通过网址 http://www.vchelp.net/wyy/j2me/tour_sample.asp 获得。

由于作者的知识水平有限,因此书中难免会出现纰漏和错误,希望读者能够不吝指出,作者在此表示万分感激。

在本书的编写过程中很多朋友对我提供了帮助。温丹丹编辑的耐心和细心,让我非常佩服,更是因为她的业务精神才能够让本书与读者见面。此外在本书的编写过程中我的朋友王志强、曾非、刘清源、吴而平、苏水容给我的鼓励和意见,能够让我从多个方面改进本书的内容。最后还要向我的母亲和我的弟弟表示感谢,感谢他们多年来给予我的家庭温暖,是他们给了我前进的动力。在此真诚向前面提到的所有给予我帮助的人说声感谢。

2004 年 3 月

目 录

第 1 章 J2ME 介绍	1
1.1 J2ME 的基本概念	1
1.1.1 J2ME 介绍	1
1.1.2 从 JVM 到 CVM 、 KVM 的演变	1
1.1.3 CDC 与 CLDC 介绍	2
1.1.4 MIDP 介绍	2
1.1.5 CLDC / MIDP 架构介绍	4
1.1.6 J2ME、J2SE 与 J2EE 之间的比较	5
1.1.7 J2ME 的前景	6
1.2 MIDP 的适用范围	7
1.3 本书的内容与组织	8
1.4 本书的读者	9
1.5 J2ME 与 MIDP 相关网站	10
第 2 章 开发前的准备工作	11
2.1 准备好开发环境	11
2.2 系统要求	11
2.3 需要的开发包与模拟器	11
2.4 建立开发环境	12
2.4.1 J2SE SDK 的安装	12
2.4.2 J2ME Wireless Toolkit 的安装	13
2.4.3 编译并利用模拟器运行示例	14
2.5 在手机中运行示例程序	15
2.6 通过 Web 下载程序	16
2.7 了解手机键盘	17
2.8 WTK 开发包的中文本地化	17
2.9 关于 java.io、java.lang、java.util 包	18
第 3 章 J2ME 开发尝试	20
3.1 使用 J2ME Wireless Toolkit 开发 MIDlet 程序	20
3.2 第一个 MIDlet 程序	20
3.2.1 理解 J2ME Wireless Toolkit 的目录结构	20
3.2.2 利用 J2ME Wireless Toolkit 创建工程	21

3.2.3 编写代码	23
3.2.4 编译运行	24
3.2.5 MIDlet 程序的生存周期	24
3.2.6 代码讲解	25
3.3 关于 MIDlet Suite	27
3.4 把第二个 MIDlet 程序添加到 MIDlet Suite	27
3.4.1 编写代码	27
3.4.2 在 MIDlet Suite 中添加新的 MIDlet	29
3.4.3 生成的 JAD 文件	30
3.5 自定义参数	31
3.6 使用资源	33
3.7 检测设备功能	35
第 4 章 J2ME 开发基础	37
4.1 MIDlet 程序开发的基本方法	37
4.2 MIDP 程序类 MIDlet	37
4.2.1 MIDlet 类介绍	37
4.2.2 MIDlet 类功能介绍	38
4.2.3 MIDlet 类中需要重载的方法	38
4.3 Display 类	39
4.3.1 可以在屏幕上显示的窗口	39
4.3.2 Display 类功能介绍	40
4.3.3 Display 类 MIDP v2.0 新增功能介绍	41
4.3.4 在屏幕上显示 Canvas 对象	42
4.3.5 在屏幕上显示 Screen 对象	42
4.4 菜单命令类 Command	43
4.4.1 Command 类	43
4.4.2 命令类型与优先级	44
4.5 CommandListener 接口与菜单命令的处理	45
4.5.1 CommandListener 接口	45
4.5.2 命令事件的处理	46
4.5.3 处理来自不同窗口的命令	47
4.6 编写 MIDlet 程序的基本思路	51
第 5 章 利用 Canvas 进行界面设计	52
5.1 低级界面开发	52
5.2 Displayable 类	52
5.2.1 Displayable 类介绍	52
5.2.2 MIDP v2.0 中 Displayable 类新增功能介绍	53

5.3 Canvas 类	54
5.3.1 Canvas 类介绍	54
5.3.2 Canvas 类与作图有关的功能介绍	54
5.3.3 在 Canvas 中作图需要重载的 paint 方法	55
5.4 作图	56
5.4.1 Graphics 类	56
5.4.2 坐标	56
5.4.3 输出时的对齐方式	57
5.4.4 颜色	58
5.4.5 文字输出	59
5.4.6 字体	60
5.4.7 线条, 图形与填充	63
5.4.8 图像	66
5.4.9 裁剪区	70
5.4.10 Alpha 混合功能	71
5.4.11 作图示范程序讲解	72
5.5 键盘设备输入与指针设备输入	74
5.5.1 设备的按键	74
5.5.2 Canvas 类与按键有关的功能介绍	75
5.5.3 Canvas 类键盘事件介绍	75
5.5.4 设备的指针	76
5.5.5 Canvas 类与指针设备有关的功能介绍	76
5.5.6 Canvas 类指针事件介绍	76
5.5.7 处理按键事件和指针事件	77
5.6 MIDP v2.0 新增的全屏幕模式	79
5.7 线程, 定时器与动画	80
5.7.1 J2ME 中的线程	80
5.7.2 J2ME 中的定时器	82
5.7.3 动画	84
5.8 绘图开发实例: 日历	88
5.8.1 日历功能	88
5.8.2 日历类静态类图	89
5.8.3 接口定义代码	90
5.8.4 日历类 CalendarCanvas 代码	91
5.8.5 演示程序代码	97
第 6 章 利用 Screen 进行界面设计	100
6.1 高级界面开发	100
6.2 Screen 类及其派生类	100

6.3 Ticker 类.....	102
6.4 TextBox 类	102
6.5 Alert 类.....	106
6.6 List 类与 Choice 接口.....	111
6.7 利用 Form 组织界面.....	119
6.7.1 Form 类与 Item 类.....	119
6.7.2 ItemStateListener 接口与状态监听.....	123
6.8 Item 类与 Form 类在 MIDP v2.0 中新增功能.....	125
6.8.1 对象大小控制	125
6.8.2 对象布局	126
6.8.3 ItemCommandListener 接口与命令处理.....	128
6.9 StringItem 类.....	131
6.10 TextField 类.....	136
6.11 ChoiceGroup 类.....	137
6.12 DateField 类	141
6.13 Gauge 类.....	142
6.14 包含 Gauge 对象的 Alert 对象.....	146
6.15 ImageItem 类.....	149
6.16 Spacer 类	154
6.17 CustomItem 类	155
6.17.1 CustomItem 类基本功能介绍	155
6.17.2 在 CustomItem 类中实现大小计算和显示	156
6.17.3 交互模式	160
6.17.4 处理方向键事件.....	161
6.17.5 处理按键与指针事件.....	163
6.17.6 命令处理	165
6.18 界面开发实例：日程管理	167
6.18.1 日程管理功能.....	167
6.18.2 数据的表示与存储.....	168
6.18.3 程序静态类示意图.....	169
6.18.4 程序的操作方式.....	170
6.18.5 程序的运行流程.....	171
6.18.6 日程记录类 ScheduleItem 代码.....	172
6.18.7 CalendarListForm 类的代码	173
6.18.8 CalendarEditForm 类的代码	175
6.18.9 程序代码	177
第 7 章 MIDP v2.0 游戏开发	182
7.1 MIDP v2.0 游戏开发简介	182

7.2	GameCanvas 类.....	182
7.3	Layer 类.....	187
7.4	TiledLayer 类.....	188
7.5	LayerManager 类.....	193
7.6	Sprite 类.....	201
第 8 章	持续数据存储.....	209
8.1	MIDP 对持续数据存储的支持.....	209
8.2	RMS 介绍.....	209
8.3	记录文件的创建与删除.....	210
8.4	空间检查.....	215
8.5	记录的添加与删除.....	215
8.6	记录的读取, 修改和遍历.....	216
8.7	记录的比较与筛选.....	222
8.8	监听记录的变动.....	225
8.9	数据存储的其他辅助功能.....	226
8.10	在多个 MIDlet Suite 间共享数据.....	227
8.11	数据库开发实例: 日程记录的保存.....	228
8.11.1	数据存储功能.....	228
8.11.2	程序静态类示意图.....	229
8.11.3	CalendarStore 类的代码.....	230
8.11.4	ScheduleItem 类的代码.....	231
8.11.5	ItemTable 类的代码.....	232
8.11.6	程序代码.....	234
第 9 章	网络通信功能.....	236
9.1	MIDP 对网络功能的支持.....	236
9.2	MIDP 中网络功能的基本结构.....	236
9.2.1	GCF 介绍.....	236
9.2.2	协议字符串.....	237
9.2.3	Connector 类.....	238
9.2.4	Connection 接口.....	239
9.3	HTTP 的使用.....	240
9.3.1	HTTP 简介.....	240
9.3.2	安装 IIS 作为 HTTP 服务器.....	242
9.3.3	HttpConnection 接口.....	243
9.3.4	利用 StreamConnection 接口访问页面.....	247
9.3.5	利用 ContentConnection 接口访问页面.....	248
9.3.6	利用 HttpConnection 接口访问页面.....	250

9.3.7 使用 POST 方法访问页面	252
9.3.8 HTTP 访问程序的完整代码	254
9.3.9 通过 HTTP 下载 PNG 文件	257
9.3.10 处理中文的显示	257
9.3.11 让程序支持 Session	259
9.4 使用 TCP 套接字创建客户方程序	260
9.4.1 套接字介绍	260
9.4.2 编写 TCP 套接字服务器程序	261
9.4.3 SocketConnection 接口	263
9.4.4 利用 TCP 套接字建立连接与交换数据	264
9.5 使用 UDP 套接字创建发送端程序	267
9.5.1 UDP 介绍	267
9.5.2 UDPDatagramConnection 接口	269
9.5.3 利用 UDP 套接字发送与接收数据	270
9.6 WTK 提供的网络监视工具	271
9.7 网络开发实例：网上下载日程记录	272
9.7.1 Web Service 简介	272
9.7.2 日程记录下载功能	273
附录 A SunONE Studio 4 ME 介绍	278
附录 B WTK 2.0 与 SunONE Studio 4 ME 的集成	282
附录 C Nokia Developer's Suite for J2ME 开发套件介绍	286

第 1 章 J2ME 介绍

1.1 J2ME 的基本概念

1.1.1 J2ME 介绍

在今天的生活中，个性化的智能信息用品已经是必需品。这些用品包括移动电话、机顶盒、双向寻呼机、智能卡和掌上电脑等。在几年前开发人员很难想像能够用一种统一的开发方式来开发运行在这些设备上的程序，也很难想像在这些设备上开发兼容的程序，而现在借助 Java 的能力这些都成为可能。

Sun 公司将 J2ME (Java 2 Micro Edition, Java 2 微型版)定义为“一种以广泛的消费性产品为目的、高度优化的 Java 运行时环境”。自从 1999 年 6 月在 JavaOne Developer Conference 上声明之后，J2ME 进入了小型设备开发的行列。通过 Java 的特性，遵循 J2ME 规范开发的 Java 程序可以运行在各种不同的小型设备上。

Sun 公司希望借助 J2ME 这把利剑对嵌入式设备这个混乱的领域进行统一，让 Java 的范围扩展到所有的电子设备开发上，按照现在的发展速度，J2ME 很快将被广泛应用于消费和嵌入式设备中。

与以前 Sun 公司推出的 J2EE (Java 2 Enterprise Edition, Java 2 企业版)、J2SE (Java 2 Standard Edition, Java 2 标准版)规范相比，J2ME 不是一个单独的技术规范，而是一系列技术规范总称。这些规范定义了 Java 技术在资源限制的设备中的表现形式。J2ME 主要适用于小型嵌入式设备，这些设备与 PC 或是服务器设备相比没有统一的硬件标准、外观与操作方式，其功用也是千差万别。但是 J2ME 在适用于这些设备的同时也保留了 Java 的传统特性，即任何时间和任何地点的代码具有可移植性、部署灵活性、安全的网络传输性，以及代码稳定性。正是因为如此，开发人员才可以借助 J2ME 力量让编写的程序能够运行在其他型号或厂商的设备上。

1.1.2 从 JVM 到 CVM、KVM 的演变

Java 程序是以边解释边执行的方式运行的，所有的 Java 程序都会被编译成为二进制代码并运行在 Java 虚拟机上(J2EE、J2SE 中的 Java 虚拟机简称为 JVM)。而 J2ME 针对的设备主要是嵌入式和消费类的设备，因为这些设备内存和处理器的限制，所以 J2ME 所包含的类库也比较小一些，相对于 J2SE 的类库来说作了一些剪裁，虚拟机的功能也相对简单。

在 J2ME 中有两类虚拟机：CVM (C Virtual Machine, C 虚拟机)与 KVM (K Virtual Machine, K 虚拟机)。KVM 和 CVM 均可被看作是一种 Java 虚拟机，是 JVM 的子集，在功能上都是 JVM 的缩减版。这两类虚拟机的适用范围并不相同，简单地说，CVM 的功能比 KVM 功能更为强大。在读者阅读完 1.1.3、1.1.4 节后，将会更清楚地了解这两种虚拟机的具体适用范围。

运行在 J2ME 上的程序大致分为两种：

- J2ME Application：运行在 CVM 之上。
- MIDlet (Mobile Information Devices let, 移动信息设备小程序)：程序运行在 KVM 之上，MIDlet 程序的开发也就是本书要讲述的内容。

1.1.3 CDC 与 CLDC 介绍

J2ME 面对的是大量各种不同的小型嵌入式设备，它们在外观和功能上均各不相同。J2ME 将这些设备进行分类，将一些共性提取出来形成适合于某个范畴中设备可用的规范，称为“配置”。读者也可以将配置理解成为对硬件的描述，所以通过定义配置的方法就能够清楚地描述硬件功能。

在 J2ME 中配置可以分为两类：

- CDC (Connected Device Configuration, 连接设备配置)，运行在这一配置上的程序称为 J2ME Application，在 CVM 上进行解释。
- CLDC (Connected Limited Device Configuration, 连接限制设备配置)，运行在这一配置上的程序称为 MIDlet，在 KVM 上进行解释。

从设备的硬件功能角度来看，CDC 主要针对 32 位的处理器主频通常在 75MHz 以上，内存可能在 1~4MB；而 CLDC 主要针对 16 位、32 位主频在 16MHz 以上的处理器，设备的内存比较少，可能只有 512KB，甚至更少。因为 CDC 与 CLDC 针对的设备 CPU 处理能力和内存是不一样的，所以它们所使用的虚拟机和核心类库也不相同，CDC 的核心类库也比 CLDC 的核心类库要大。CDC 用的虚拟机是 CVM，而 CLDC 用的是 KVM，CVM 比 KVM 包含更多的功能和支持更多的特性。

从设备的用途和功能的角度来看，CDC 针对的设备主要是数字电视、机顶盒、网络电话、车载计算设备等。这些设备的特点是有线连接、电源稳定、设备资源比较受限。而 CLDC 针对的设备主要是移动手持设备，例如 PDA、POS 机和智能手机。这些设备的特点是无线连接、没有稳定的电源供应(通常使用电池)、设备资源极少。目前在国内外用得比较广泛的 CLDC 是在智能手机上的应用。

1.1.4 MIDP 介绍

J2ME 使用配置(CDC 或 CLDC)和简表(Profile)定制 JRE(Java Runtime Environment, Java 运行时环境)。一个完整的 JRE 由配置和简表组成，配置决定了所使用的 JVM(可能是 CVM 或 KVM)，而简表通过定义特定的类来为应用程序提供功能上的支持，一个简表定义了设备所提供的 API 集合。

对比前面简表的定义可以知道，MIDP (Mobile Information Devices Profile, 移动信息

设备简表)是简表的一种, 定义了移动信息设备的类型和提供相关的 API (Application Programming Interface, 应用程序编程接口)集合, MIDP 所定义的功能更加面向用户, 而且比 CLDC 更高级。

具体地讲, 在进行 MIDlet 程序开发时所使用的 API 来自于:

- CLDC v1.0 规范中定义的 4 个包 `java.io`、`java.lang`、`java.util` 与 `javax.microedition.io`, 提供 Java 语言和 CLDC 设备所支持的基本功能。
- MIDP 所定义的其他包有 `javax.microedition.midlet`、`javax.microedition.lcdui`、`javax.microedition.lcdui.game`、`javax.microedition.rms` 等, 提供诸如程序生命周期控制、用户界面、游戏、持久存储这样的功能。

为了理解虚拟机、配置和简表之间的关系, 图 1.1 将这 3 个概念进行对比。在对比 J2ME 内的不同虚拟机、配置和简表的同时, 也将 J2ME 结构和 J2SE 结构进行了对比。

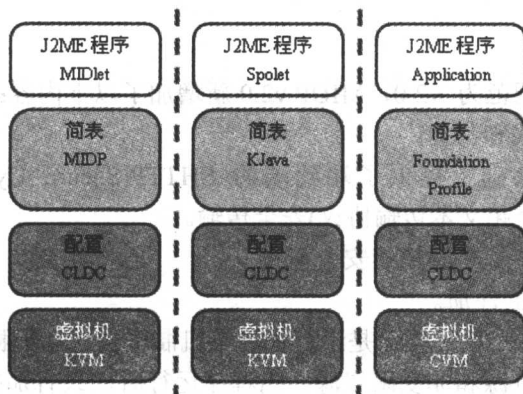


图 1.1 不同虚拟机，配置与简表之间的关系

在图 1.1 中出现了 KJava 这个字眼, 这对于了解过手机程序开发的朋友一定不陌生, 曾经有一段时间 KJava 程序开发几乎等同于手机上的 Java 程序开发。在这里, 为了纠正一下这个看法有必要解释清楚 KJava 所代表的含义。KJava 是 Sun 公司开发的一种简表, 它包含了 KJava API。KJava 并不作为一种完整的、功能齐全的简表, Sun 公司最初推出 KJava 时只是作为一种示范简表, 即用于演示简表如何与 CLDC 一起工作。由于 KJava 出现较早, 因此早期的部分手机设备都选择支持 KJava 所定义的 API。因此, 如果把手机上的 Java 编程都称为 KJava 编程说法那是不正确的, 把 KJava 程序开发和 MIDP 程序开发等同也是不正确的。由于历史原因, 到目前为止 KJava 已经被广泛使用, 但是作者相信 MIDP 将会逐渐取代 KJava 的地位, 毕竟 MIDP 是一个更为开放的标准。

MIDP 和 KJava 一样, 也是建立在 CLDC 之上的, 并且提供一个标准的运行时环境, 允许在终端用户设备上动态地部署新的应用程序和服务。与 KJava 不同点在于, MIDP 是一个公共的、为移动设备设计的工业标准简表, 它不依赖某个特定的商家。对于移动应用程序开发来说, 它是一个完整的、受支持的基础。因此现在新的智能手机设备, 如果支持 J2ME 都会选择支持 MIDP, 而不是 KJava。

除 MIDP 简表外, 目前还存在其他类型的简表, 例如:

- Foundation Profile(基础简表): 提供除了用户界面以外 J2SE 所能够提供的标准

类库。

- **Personal Profile(个人简表)**: 针对那些资源相对有限, 但是对网络访问要求很高的设备, 例如 Web-TV、汽车导航系统等。

相信在以后随着 J2ME 更广泛地推广和应用, 会逐渐出现其他类型的简表。

在功能方面, MIDP v1.0 规范中包括:

- 应用的下载。
- 应用的生命周期。
- 端到端的传输协议, 例如 HTTP (Hyper Text Transmission Protocol , 超文本传输协议)。
- 网络连接。
- 持续性存储。
- 定时器。
- 用户界面。

现在 MIDP 最新的规范为 v2.0, MIDP v2.0 新增加了以下内容:

- 应用程序的下载和计费。
- 在 HTTP 的基础上提供了端到端的 SHTTP(Secure Hyper Text Transmission Protocol , 安全超文本传输协议)安全传输。
- 应用程序的数字签名和域的安全模式。
- MIDlet 的 push 注册。

在安全方面, J2SE 中 Java 程序是运行在沙箱机制下的, 为了限制程序对资源的访问, 资源被分为两类: 受限资源和非受限资源。当程序运行时, 只有那些被信任的程序才能够访问受限资源与调用相关的 API。在 J2ME 中为了达到安全的目的也继承了沙箱机制, 此外, MIDP v2.0 引入了信任的 MIDlet 的新概念。当无线信息设备检测到一个程序是被信任的程序时, 则此程序允许访问安全机制所规定的 API。例如, 这些 API 可以访问手机内地址本, 或者可能是允许访问无线网络。

1.1.5 CLDC / MIDP 架构介绍

J2ME / CLDC / MIDP 架构是专为移动通讯设备而设计的。有了前面的介绍, 下面再用图 1.2 说明 MIDP 和 CLDC 之间的关系。

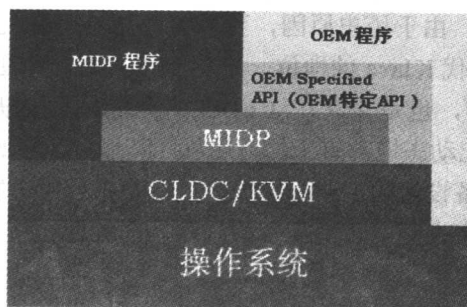


图 1.2 CLDC 与 MIDP 之间的关系

在图 1.2 中，读者可以看出：

- MIDP 建立在 CLDC 功能的基础之上。
- J2ME 规范允许设备制造商提供与自己设备相对应的 API 供程序调用，一般来说，设备厂商都会提供一些自己的 API 供开发人员使用，这些功能都是有针对性的，因此在不同的设备之间可能无法移植。一般来说，这部分 API 能够提供额外的功能或者在运行性能上有所提高。

此外，在图 1.2 中可以看出 CLDC 之上有两类 API：

- MIDP API：这些 API 正是 MIDP 规范所要定义的 API。
- OEM Specific API：MIDP 规范所涉及的无线通讯设备多种多样，因此它不可能涉及所有设备的需求。因此这一类的 API 是由 OEM 厂商提供的，以便访问特定设备的特定功能。但基于这些 API 的应用可能不在其他的 MIDP 设备上运行。关于这一部分，各个手机厂商都提供了自己的开发包，读者可以到手机厂商网站下载。

这里需要澄清一个概念，无线通信设备并不等同于移动信息设备。尽管就目前的实际情况来讲，MIDP 规范主要是在无线通信设备中得到了广泛的应用。

用 Java 编写程序的最大好处莫过于一次编写，随处运行。但是由于 J2ME/CLDC/MIDP 架构的限制，同一个 Java 程序只能在相同的 MIDP 支持下运行。要完整地定义所有的移动设备是非常困难的，但是随着 MIDP 的标准化和各个设备厂家对 MIDP 的支持，在标准的 MIDP 下开发的 Java 应用程序能够在支持标准 MIDP 的设备上运行。目前市面上大多数的智能手机都能够支持 MIDP v1.0，而且已经开始出现支持 MIDP v2.0 的手机，例如 Nokia 的 S90 系列，SonyEricsson 的 P900 系列。

有一点必须要提醒读者注意，CLDC 和 MIDP 是 J2ME 中两个不同的规范，在 CLDC v1.0 和 MIDP 1.0 规范下，这两个部分是没有交叉的，但是当 MIDP 2.0 发布时，除了在中 MIDP 所提供的 API 中进行了改动，还在 CLDC 所提供的 API 中增加了部分功能。例如 javax.microedition.io 是由 CLDC 规范所定义的，但在 MIDP 2.0 发布时 javax.microedition.io 包中增加了对套接字，串口等的支持。读者在学习时需要注意这一点，当查看某个 API 时，可以从帮助文档中发现此 API 是属于哪个规范定义的，避免引起理解上的混乱。

1.1.6 J2ME、J2SE 与 J2EE 之间的比较

J2SE 为桌面开发和低端商务应用提供了可行的解决方案，J2EE 为了各种企业应用的开发提供了解决方案，而 J2ME 则希望让自己成为小型设备和嵌入式设备的开发人员的最佳选择。

图 1.3 描述了 Java 体系结构中 J2EE、J2SE 与 J2ME 对于设备的应用范围。

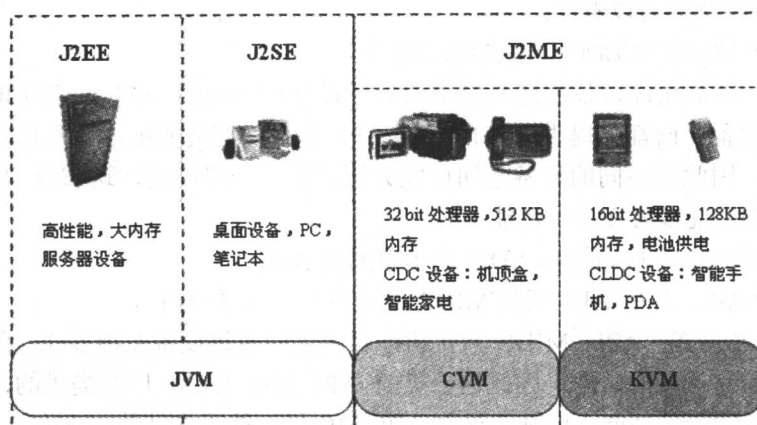


图 1.3 J2EE、J2SE 和 J2ME 的应用范围

- J2SE: 是为桌面 PC 而设计的, 主要运行在 Unix、Linux、Microsoft Windows 等操作系统上。
- J2EE: 是一个适合分布式的、多用户、企业级应用系统运转的平台。它以 J2SE 为基础, 增加了处理服务器端计算的功能。
- J2ME: 和 J2SE 不同, 它并不是一个单一的规范。准确地说, J2ME 是为了支持像小型嵌入式设备或移动设备而制定的一系列的技术和规范的总称。在提供类似的类库的同时也对 J2SE 中的类库进行了裁剪。

1.1.7 J2ME 的前景

目前 J2ME 在 PDA 方面的竞争主要来源于 Palm OS 和 Pocket PC 两大阵营。Palm OS 来自 Palm 公司, 是一个开放的系统。Palm OS 曾经在 PDA 市场上占有绝对主导地位, 已经有非常多的第三方厂商和一大批用户。而 Pocket PC 是微软及其合作伙伴 Casio、Compaq、Hewlett Packard 和 Symbol 等联合推出的, 基于 WinCE 3.0, 也是一个开放的标准系统。因为 WinCE 是一个易于扩充的系统, 所以在这个平台上厂商可以自行扩充设备和开发应用软件。

而 J2ME 在智能手机方面的竞争主要来源于 Symbian 的 EPOC。EPOC 最早由 Psion 开发, 主要面向智能手机, 也具备 PDA 的特征。此外微软推出的 PocketPC 2002 电话版和 SmartPhone 2002 也提供了对智能手机的支持。而嵌入式 Linux 方面近年来也取得了长足的发展, 到目前为止已经可以看见以 Linux 作为内核的手机推向市场, 而且是国产手机, 这实在值得大家高兴。EPOC 和嵌入式 Linux 目前也都在其本身操作系统的基础之上提供了对 J2ME 的支持。

移动信息设备的主要操作系统平台都是开放的, J2ME 的标准也是开放的, 因此基本上所有的主要操作系统平台都支持或将支持 J2ME。此外 J2ME 虽然和 J2SE 有着许多不同, 但同属于 Java 技术, 也使得 Java 程序员在学习移动信息设备开发时可以轻松过渡。

随着硬件的发展和标准的完善, J2ME 必将能够提供更强大的功能。截止到 2003 年 2 月, 已经有 21 个生产厂家推出了 100 多种支持 Java 的手持设备。读者通过 Sun 公司的