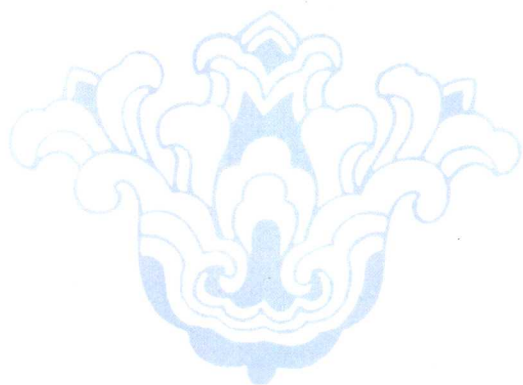


— 21 世纪 —
电 子 信 息 类 规 划 教 材

J2ME
Shiyong Jiaocheng

J2ME
实用教程

© 李新力 编著



 人民邮电出版社
POSTS & TELECOM PRESS

TP312/2540

2007

21 世纪电子信息类规划教材

J2ME 实用教程

李新力 编著

人 民 邮 电 出 版 社

北 京

图书在版编目 (CIP) 数据

J2ME 实用教程 / 李新力编著. —北京: 人民邮电出版社, 2007.9

21 世纪电子信息类规划教材

ISBN 987-7-115-16314-1

I. J… II. 李… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 079144 号

内 容 提 要

随着手机的普及和移动无线增值业务的飞速发展, 对手机游戏和无线应用开发的需求不断地增加, 使更多的程序开发人员开始关注和使用 J2ME 开发平台, 越来越多的学校将 J2ME 作为学生的专业培养方向之一。

全书内容比较浅显, 将实例贯穿于各个章节, 有利于学生学习和理解。本书内容包括 J2ME 简介, J2ME 开发环境的安装及配置, MIDP 程序开发, 高级用户界面及事件处理, 低级用户界面及事件处理, 手机游戏项目开发介绍, 手机游戏开发及案例分析, 数据存储, 无线网络开发和移动银行实例等。

本书没有仅仅局限于 J2ME 单机游戏的开发, 而是将 J2ME 的学习与开发与整个移动网络相结合, 如介绍了网络游戏、移动电视、手机银行等移动网络应用的开发。本书适用于普通本科院校通信工程、电子信息工程、网络工程、计算机软件等专业 J2ME 相关课程的教学, 同时也可作为部分高职高专院校和培训学校的教学用书。

21 世纪电子信息类规划教材

J2ME 实用教程

-
- ◆ 编 著 李新力
责任编辑 蒋 亮
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 16
字数: 381 千字
印数: 1—3 000 册
 - 2007 年 9 月第 1 版
2007 年 9 月河北第 1 次印刷

ISBN 987-7-115-16314-1/TN

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

前 言

随着中国社会的发展，人民的生活水平也日益提高，移动设备特别是手机得到了非常广泛的应用，手机游戏也被更多的游戏爱好者所接受。手机游戏和无线移动应用成为无线增值业务迅猛发展的最大亮点，因此，作为开发手机游戏和无线移动应用的主流平台，J2ME 具有非常良好的发展前景。

手机作为游戏和无线移动应用的载体，具有潜在用户群庞大和携带便利等优点。由于手机设备硬件本身的限制使得手机游戏与网络游戏又有很大的差别，在游戏程序的策划、设计和游戏美术等方面都有自己的特点。同时手机游戏和无线移动应用又是个新兴的行业。它的发展前景是十分广阔的。

本书是一本综合介绍手机游戏中各个层面知识和无线移动应用的概论性教学用书。它的内容设置从手机游戏的概述、手机游戏项目开发介绍开始，由浅入深地介绍手机游戏开发中的相关内容。它的主体模块可以分成以下几个部分：J2ME 简介，手机游戏项目开发介绍，J2ME 手机游戏编程、案例分析、数据存储和移动应用。每个部分中的内容都经过提炼和升华。本书的目的是全面讲述手机游戏编程和移动应用的知识，通过学习本书，读者可以掌握 J2ME 手机游戏和移动应用编程的基本知识和技巧，具备开发手机游戏和移动应用的能力。对于开发手机游戏还要涉及的策划和美工方面的知识，读者可以在本书所属的系列丛书中找到更加详尽的阐述内容。

本书面向的读者是广大的手机游戏与移动应用开发爱好者，以及专业的手机游戏和移动应用开发人员。学习这本书的前提是读者具有一定的 Java 基础。

希望能够通过这本书给广大的手机游戏和移动应用开发爱好者提供一个全面了解手机游戏和移动应用开发的机会，帮助他们在自己喜爱的事业上找到努力的方向。更希望为中国手机游戏和移动应用事业的发展提供更多的人才。

编者 于北京

2007 年 7 月

目 录

第 1 章 J2ME 简介	1
1.1 J2ME 基本概念	1
1.1.1 J2ME 介绍	1
1.1.2 J2ME 与 Java 体系结构	2
1.1.3 JVM、CVM 和 KVM 介绍	3
1.1.4 CDC 与 CLDC 介绍	4
1.1.5 MIDP 介绍	4
1.1.6 CLDC/MIDP 架构介绍	5
1.2 J2ME 应用介绍	6
1.2.1 J2ME 手机游戏	6
1.2.2 J2ME 无线应用程序	6
1.3 J2ME 的优势和安全	7
1.3.1 J2ME 的优势	7
1.3.2 J2ME 的安全性	8
1.4 J2ME 的前景	9
1.5 习题	9
第 2 章 J2ME 开发环境的安装及配置	10
2.1 安装 JDK	10
2.2 设置环境变量	11
2.3 安装 Sun WTK	13
2.4 Eclipse 下开发环境的搭建	16
2.4.1 安装 Eclipse	16
2.4.2 安装 EclipseMe	17
2.4.3 配置 Eclipse J2ME 的环境	21
2.4.4 Eclipse 调试环境的配置	23
2.4.5 Eclipse 的第一个 J2ME 项目	24
2.5 手机游戏开发平台介绍	27
2.5.1 Microsoft Smartphone	27
2.5.2 Symbian	29
2.5.3 BREW	31
2.5.4 Mophun	32
2.6 习题	33

第3章 开发 MIDP 程序	34
3.1 MIDP 应用程序环境	34
3.2 MIDP 程序的基本框架	34
3.3 MIDlet 程序的生命周期	35
3.4 OTA 发布程序	36
3.4.1 OTA 的介绍	36
3.4.2 MIDlet 程序属性设置	36
3.4.3 在服务器上发布程序	37
3.4.4 通过 OTA 安装 MIDlet 程序	37
3.5 小结	38
3.6 习题	39
第4章 高级用户界面及事件处理	40
4.1 MIDP 用户界面 API 的结构	40
4.2 Command 和高级事件处理	41
4.3 高层用户界面	43
4.4 基本控件编程实例	44
4.4.1 列表 List	44
4.4.2 TextBox	48
4.4.3 Alert	51
4.4.4 Form 概述	53
4.4.5 StringItem 及 ImageItem	54
4.4.6 CustomItem	57
4.4.7 TextField 和 DateField	60
4.4.8 Gauge、Spacer 和 ChoiceGroup	61
4.4.9 Ticker 类	65
4.5 小结	66
4.6 习题	66
第5章 低级用户界面及事件处理	67
5.1 Canvas 简介	67
5.2 低级事件的处理	68
5.2.1 按键事件	68
5.2.2 触控事件	70
5.2.3 Canvas 与屏幕事件处理	71
5.3 Graphics 开发简介	72
5.3.1 坐标概念	73
5.3.2 绘制基本图形	73

5.3.3 颜色操作	75
5.3.4 锚点 (anchor points)	76
5.3.5 Image 类	77
5.3.6 字体类	80
5.3.7 调整原点坐标	80
5.4 MIDP Timer API	82
5.5 综合实例: 编写欢迎界面	85
5.6 小结	87
5.7 习题	88
第 6 章 手机游戏项目开发介绍	89
6.1 手机游戏策划的概念	89
6.2 手机策划的一般步骤	91
6.3 手机游戏策划书模板	92
6.4 手机游戏策划的特点及与普通游戏策划的区别	94
6.5 手机游戏美工	95
6.6 手机游戏项目开发流程介绍	95
6.6.1 第一阶段: 提案	95
6.6.2 第二阶段: 设计	96
6.6.3 第三阶段: 实现	97
6.6.4 第四阶段: 测试与评审	98
6.6.5 第五阶段: 完成	99
6.7 开发团队的建设和管理	99
6.7.1 团队的组建	99
6.7.2 团队的管理	99
6.7.3 进度管理	100
6.7.4 代码管理	101
6.8 小结	101
6.9 习题	101
第 7 章 手机游戏开发	102
7.1 手机游戏开发简介	102
7.1.1 手机游戏的类型	102
7.1.2 手机游戏开发特点	103
7.1.3 扬长避短的手机游戏开发	104
7.2 PNG 图像格式介绍	105
7.2.1 PNG 的由来	105
7.2.2 PNG 的图档格式特性	105
7.2.3 PNG 格式图像的压缩及处理	105

7.2.4 使用 PNG 图片	106
7.2.5 游戏菜单制作	109
7.3 游戏设计中的动画	112
7.3.1 动画基础	112
7.3.2 双缓冲技术	114
7.3.3 图片的裁剪	115
7.3.4 播放动画框架	116
7.4 手机游戏 2D 动画开发	120
7.4.1 MIDP 2.0 2D 游戏开发	121
7.4.2 GameCanvas 的使用	121
7.4.3 Sprite 的使用	123
7.4.4 碰撞检测	128
7.4.5 Layer 的使用	129
7.5 手机音效开发	134
7.5.1 MMAPAPI 体系结构	134
7.5.2 Player 接口	135
7.5.3 播放简单音调	137
7.5.4 播放歌曲	138
7.5.5 控制音量	145
7.6 综合示例：飞机碰撞	149
7.7 小结	155
7.8 习题	155
第 8 章 游戏案例分析	156
8.1 扫雷游戏的设计与实现	156
8.1.1 游戏的设计	156
8.1.2 游戏核心算法	157
8.2 Hero 游戏的设计与实现	162
8.2.1 游戏的设计	162
8.2.2 游戏核心算法	163
8.3 小结	166
8.4 习题	166
第 9 章 数据存储	167
9.1 J2ME 数据库系统的概况	167
9.2 RMS (Record Management System)	168
9.3 RecordStore 的管理	169
9.3.1 RecordStore 的打开	169
9.3.2 RecordStore 的关闭	169

9.3.3	RecordStore 的删除	170
9.3.4	其他相关操作	170
9.4	RecordStore 的基本操作	171
9.4.1	增加记录	171
9.4.2	修改与删除记录	171
9.4.3	自定义数据类型与字节数组的转换技巧	171
9.4.4	利用 RMS 实现对象序列化	172
9.4.5	RecordStore 的进阶操作	173
9.5	游戏中数据存取范例	178
9.6	小结	181
9.7	习题	182
第 10 章	无线网络开发	183
10.1	J2ME 联网技术简介	183
10.2	通用连接框架	184
10.2.1	GCF 的层次结构	184
10.2.2	GCF 的使用	185
10.3	HTTP 连接	185
10.3.1	HTTP 简介	185
10.3.2	HTTP 连接状态	186
10.3.3	建立 HTTP 连接	187
10.3.4	设置 HTTP 请求头	187
10.3.5	使用 HTTP 连接	189
10.3.6	关闭 HTTP 连接	190
10.3.7	HTTP 示例	190
10.4	Socket 连接	199
10.4.1	Socket 连接简介	199
10.4.2	Socket 示例	200
10.5	Datagram 连接	208
10.5.1	Datagram 连接简介	208
10.5.2	Datagram 示例	209
10.6	小结	217
10.7	习题	217
第 11 章	移动银行实例	218
11.1	缴费系统的整体设计	218
11.2	缴费系统的数据序列化	219
11.3	缴费系统的功能模块设计	222
11.4	进度屏幕的设计	223

11.5	网络处理模块的设计	225
11.6	客户端屏幕类的设计	231
11.6.1	登录屏幕	231
11.6.2	查询屏幕的设计	233
11.6.3	设置账户信息屏幕的设计	234
11.6.4	主屏幕的设计	237
11.7	服务器端的设计	239
11.7.1	表的设计	239
11.7.2	编写 Servlet	240
11.7.3	部署和运行程序	243
11.8	小结	243
11.9	习题	243

第1章 J2ME 简介

随着移动通信技术的突飞猛进,移动开发这个新鲜的字眼慢慢成为开发者关注的热点。在网上进行的一次调查显示,有 24.34%的受访者涉足嵌入式/移动设备应用开发,这个数字可能略高于实际的比例,但也足可说明嵌入式/移动设备应用开发是一块诱人的蛋糕。J2ME (Java 2 Micro Edition) 是嵌入式/移动应用平台的王者, Linux 和 Windows CE 分列二、三位。目前, Nokia 等厂商力推的 Symbian 平台的开发者占有率尚未达到满意水平,考虑到调查项合并了嵌入式设备(例如 PDA)和移动设备(例如智能手机), Symbian、Windows CE 系列在移动平台上会是竞争的主要两方。如果开发厂商能在标准实现上做得更加规范,则 J2ME 的跨平台特性会发挥得更加淋漓尽致,继续保有王者地位。本章将从 J2ME 的体系结构和 MIDlet 应用程序模型讲起。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象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,也很难想象在这些设备上开发兼容的程序,而现在借助 Java 语言的跨平台能力这些都成为可能。

1.1 J2ME 基本概念

J2ME 是 Sun 公司针对嵌入式、消费类电子产品推出的开发平台,与 J2SE 和 J2EE 共同组成 Java 技术的三个重要的分支。J2ME 实际上是一系列规范的集合,由 JCP 组织制定相关的 JSR (Java Specification Request) 并发布,各个厂商会按照规范在自己的产品上进行实现,但是必须要通过 TCK 测试,以确保兼容性。如 MIDP2.0 规范就是在 JSR118 中制定的。可能接触过 J2ME 的开发者会觉得说 J2ME 是一系列的规范不准确,因为在开发中经常会用到 CLDC (Connected Limited Devices Configuration)、MIDP (Mobile Information Devices Profile) 等内容。其实这并不矛盾,因为这些就是在相关规范中制定的。

1.1.1 J2ME 介绍

Sun 公司将 J2ME 定义为“一种以广泛的消费性产品为目标的、高度优化的 Java 运行时环境”。自从 1999 年 6 月在 JavaOne Developer Conference 上发布之后, J2ME 进入了小型设备开发的行列。由于 Java 的跨平台特性,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 这把利剑将嵌入式设备这个混乱的领域统一,让 Java 的使用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)、J2SE (Java 2 Standard

Edition) 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의 总称。这些规范定义了 Java 技术在资源有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 的力量使编写的程序能够运行在其他型号或厂商的设备上。

1.1.2 J2ME 与 Java 体系结构

熟悉 Java 发展历史的人或多或少都听说过, Java 技术一开始并非就叫做 Java, 而是叫做 OAK, 而且最早的时候就是为了嵌入式系统而设计的一项产品。OAK 的诸多特性刚好又适合用在网络上 (例如可移植性、编译后程序码很小), 后来 OAK 被改名成 Java, 随着网络的发展成为 Internet 上的闪亮巨星, 成为了 Internet 上的“世界语”。虽然 Java 在企业级的 Web 应用上得到了广泛的应用, 可是其实骨子里面还是非常适合用在嵌入式系统之中。Java 1.0 发表之后, Java 就被广泛地使用在桌上型应用程序以及 Applet 的开发上, 但是, 从 Java 1.1 开始, Java 又回到了嵌入式系统应用开发领域, 在当时 Sun 公司发表了 Embedded Java 与 Personal Java (也有人称为 PJava) 这两项规格。其中 Embedded Java 是为了资源十分有限, 而且没有显示设备的嵌入式装置而设计; Personal Java 则是为了在能够与网络连接、并拥有显示系统 (例如彩色 LCD) 的消费性电子装置而设计。接着 Java 的版本演进到 Java 2, 这时为了再明显区分各种 Java 的应用, 所以分割出了 J2EE、J2SE 以及 J2ME 三种版本。图 1-1 所示的是 Java 的体系结构和这三个版本的应用领域。

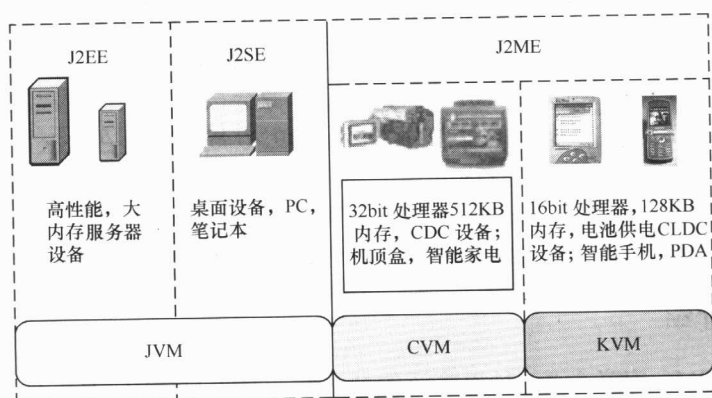


图 1-1 J2EE、J2SE 和 J2ME 的应用领域

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의 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。

1.1.3 JVM、CVM 和 KVM 介绍

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，因为这些设备受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裁剪，虚拟机的功能也相对简单。

在 J2ME 中有两类虚拟机：C 虚拟机（C Virtual Machine, CVM）与 K 虚拟机（K Virtual Machine, KVM）。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，下面将会更清楚地介绍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。

（1）J2ME 应用程序（J2ME Application）：运行在 CVM 之上。

（2）移动信息设备小程序（Mobile Information Devices let, MIDlet）：运行在 KVM 之上，MIDlet 程序的开发正是本书要讲述的内容。

JVM 运行结构如图 1-2 所示。

在传统的 Java 环境中，为了防止程序在传送途中被篡改以及其他一些安全上的考虑，当程序被类别载入器载入后，紧接着要进行 Byte Code 审核，审核通过以后才允许 Java 虚拟机执行它。这一操作 PC

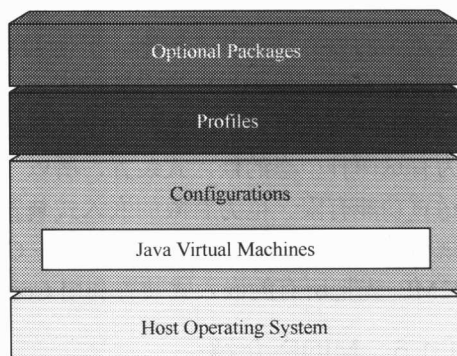


图 1-2 JVM 运行结构图

上执行速度很快，但是在 CLDC 所描述的这类系统中，要想作完全相同的处理，从处理能力和速度上来看就显得有些力不从心了。为了解决这个问题，程序设计人员需要在程序设计结束后再多做一件事：预先审核。通过预先审核，会在最终的类文件中加入一些特殊的符号，当该程序下载到目标平台上去执行时，可以以较快的速度完成审核操作。

KVM 是用于 J2ME 平台最小的虚拟机，并且是用于 CLDC 配置的虚拟机。但是 J2ME 应用程序并不一定非要使用 KVM，J2ME 技术可以使用任何虚拟机，不过至少应当有 KVM 这样的功能。

由于 KVM 的设备一般只有狭小的内存空间和有限的处理能力，KVM 使用 C 语言编写（它不是现有的 VM 改进以后的产品）。此外，KVM 是模块化的，也就是说，它是由模块构建的，当模块实现了预先设定的目标后，就可以很容易地把这一模块卸载。可选的模块包括：大的数据类型（long、float 和 double）、多维数组、类文件验证等。

构建 KVM 的本地界面时以轻便性为原则，所以在 KVM 中任务切换不依赖硬件产生的计时器中断，因此从这种意义上来说不是抢先式。任务切换发生在虚拟机执行了一个预设编号的字节码之后。并且，KVM 的无用单元收集利用一个标记清扫（mark and sweep）算法来实现无用单元的释放。因此，对象引用是直接的，就像标准的 Java 程序一样。

当然，除了虚拟机以外还有许多可用的执行环境，在小型设备中，虚拟机必须要么被扩展，要么在附加工具的协助下提供一个更加完整的运行期环境，正是这个原因，KVM 需要

附带的工具，比如说，JavaCodeCompact 工具提供了预链接和预加载类，允许 Java 类被直接地链接进虚拟机中。

KVM 一个可选的附件就是 Java Application Manager (Java 应用程序管理器，简称 JAM)。JAM 的工作就是处理下载、安装、执行和卸载 CLDC 设备上的应用程序的细节问题，因为资源有限，在 CLDC 设备上有可能不存在这些功能。JAM 也处理更新安装应用程序的操作。如果更新过程失败，它甚至可以重新使用旧的应用程序。

JVM、CVM 和 KVM 的关系如图 1-3 所示。

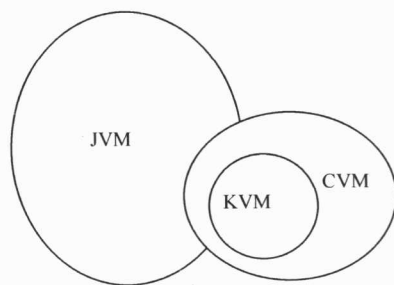


图 1-3 JVM、CVM、KVM 关系图

1.1.4 CDC 与 CLDC 介绍

J2ME 在设计其规格的时候，遵循“对于各种不同的装置而造出一个单一的开发系统是没有意义的事”这个基本原则。于是 J2ME 先将所有的嵌入式装置大体上区分为两种：一种是运算功能有限、电力供应也有限的嵌入式装置（例如 PDA、手机）；另外一种则是运算能力相对较佳、并请在电力供应上相对比较充足的嵌入式装置（例如冷气机、电冰箱、电视机顶盒）。因为有两种型态的嵌入式装置，所以 Java 引入了一个叫做 Configuration 的概念，然后把上述运算功能有限、电力有限的嵌入式装置定义在 Connected Limited Device Configuration (CLDC) 规格之中；而另外一种装置则定义在 Connected Device Configuration (CDC) 规格中。也就是说，J2ME 先把所有的嵌入式装置利用 Configuration 的概念区隔成两种抽象的型态。

1.1.5 MIDP 介绍

J2ME 使用配置 (CDC 或 CLDC) 和简表 (Profile) 定制运行时环境 (Java Runtime Environment, Java, JRE)。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，移动信息设备简表 (Mobile Information Devices Profile, MIDP) 是简表的一种，定义了移动信息设备的类型和提供相关的应用程序编程接口 (Application Programming Interface, API) 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 来自于以下两个方面。

(1) 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。

(2) 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 1-4 将这 3 个概念进行对比。

在图 1-4 中出现了 KJava 这个字眼，这对于了解

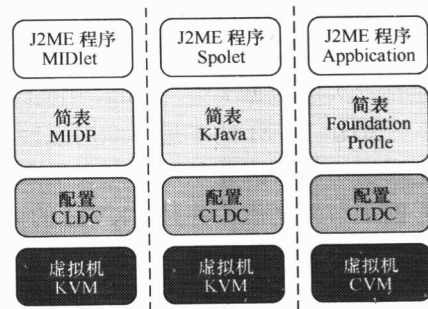


图 1-4 不同虚拟机，配置与简表之间的关系

手机程序开发的读者一定不陌生,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,因此早期的很多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 编程那是不正确的,把 KJava 程序开发和 MIDP 程序开发等同也是错误的。由于历史原因,到目前为止 KJava 已经被广泛使用,但相信 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 的不同点在于, 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;还有 Web-TV、汽车导航系统等。相信在以后随着 J2ME 技术的发展,会逐渐出现更多类型的简表。

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,或者可能允许访问无线网络。

1.1.6 CLDC/MIDP 架构介绍

J2ME、CLDC 和 MIDP 架构是专为移动通信设备而设计的。有了前面的介绍,下面再用图 1-5 说明 MIDP 和 CLDC 之间的关系。

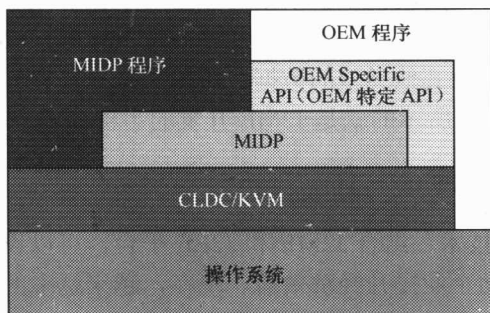


图 1-5 CLDC 与 MIDP 之间的关系

从图 1-5 中可以看出: 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 1-5 中可以看出 CLDC 之上有两类 API。

(1) MIDP API: 这些 API 正是 MIDP 规范所要定义的 API。

(2) OEM Specific API: MIDP 规范所涉及的无线通信设备多种多样, 因此它不可能满足所有设备的需求。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 编写程序的最大优点是可以一次编写, 随处运行。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 1.0, 而且已经开始出现支持 MIDP 2.0 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 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。

1.2 J2ME 应用介绍

1.2.1 J2ME 手机游戏

如同 Internet 的普及导致网络游戏的高速发展一样, 手机的普及导致手机游戏市场呈现爆炸式的发展, 随着手机性能的增强, 在未来几年里对手机游戏开发的需求将大幅上涨。现代手机的一个特性就是它们还是网络计算机, 能够高速发送和接收数字数据。除了语音数据以外, 它们还可以发送和接收其他类型的数据。所以类似《传奇》、《千年》这样的网络游戏也可以在手机上实现。图 1-6 所示的就是 J2ME 开发的手机游戏。

1.2.2 J2ME 无线应用程序

利用 Java 的无线应用技术, 结合 J2ME 和 J2EE 平台, 可以设计并开发出健壮、高效的企业级无线应用程序, 帮助企业提升经济效益。现在, 越来越多的电信厂商生产出性能优越的 Java 手机, 而且在电信运营商、电信设备制造商和应用程序开发者之间存在着一种较好的营利模式。相信 Java 无线应用会有更好的发展。应用 Java 无线应用技术设计和实现一个移动客户支持管理系统。该系统是桌面型客户关系管理系统在智能移动终端上的扩展, 主要应用于中小型企业的客户管理部门。移动客户支持管理系统有助于现场销售人员和服务工程师

及时地了解公司产品信息,更好地管理客户和联系人资料,制造和把握更多的商业机会,并且能够快速有效地处理好客户发出的请求。近年来,随着 Internet 的快速发展,它已经成为我们生活中不可缺少的一部分,而移动通信设备的不断发展使我们可以摆脱 PC 的束缚,实现移动访问 Internet。可以预见,移动通信设备和无线通信设备的发展将会大大改变 Internet 的现状。今天,人们已能够享受从手机或 PDA 通过 Internet 访问实时信息(如新闻、股票数据和天气)的便利了。而目前大部分的无线网络应用都是由 WAP 和 I-Mode 这样的技术所支持的。随着无线网络带宽和无线设备处理能力的增长,无线工业愈加成熟,人们不再满足于仅仅通过手机在 Internet 上冲浪,他们需要交互性更好和更人性化的无线应用,使其可以更显著地改进生活方式,进一步地简化商务活动。图 1-7 所示的就是 J2ME 开发的发送邮件的应用。



图 1-6 J2ME 开发的手机游戏

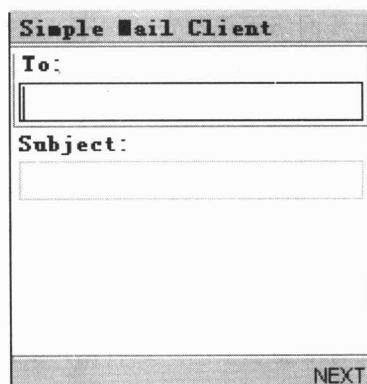


图 1-7 J2ME 开发的无线应用

1.3 J2ME 的优势和安全

1.3.1 J2ME 的优势

1. 平台独立性

J2ME 将 Java “一次编写,到处运行”的设计理念扩展到无线领域。使用 Java 开发无线应用程序,可以在不同厂商的不同设备上运行。对于应用程序开发人员而言,只要手机安装了 Java 虚拟机就可以执行 Java 应用程序,不需要为每一个目标平台各写一个版本,这极大地提高了程序的可移植性。对于用户而言,也不必担心下载的程序无法执行,目前 Java 已经成为手机应用程序的标准。

2. 简单

在 Java 平台所建立的对象,会通过垃圾回收机制来释放这个对象所占用的内存,因此程序开发人员不需要随时去检查未释放的内存,这对资源有限的手机等设备而言是很重要的,一般人常常会因为忘记释放某一块内存而导致程序无法执行甚至手机当机。

3. 丰富的类库和网络功能