

信息科学与技术丛书

程序设计系列

J2ME

手机程序

Eclipse开发基础

郎 锐 孙 方 编著

- ◎ 使用 J2ME 无线开发工具包
- ◎ 深入了解 Eclipse 工作台
- ◎ 图形界面编程实例
- ◎ 手机游戏实例开发
- ◎ MIDP 2.0 编程实例



附赠光盘



机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书

程序设计系列

J2ME 手机程序 Eclipse 开发基础

郎 锐 孙 方 编著



机械工业出版社

本书以 Eclipse 开发环境为背景,以 J2ME 手机程序开发为线索,对 J2ME 的基本知识、开发环境的配置与准备、常用的开发工具尤其是 Eclipse 的使用方法、J2ME 图形界面编程、MIDP 1.0 和 MIDP 2.0 开发手机游戏程序等主要内容做了介绍。同时,为了方便部分初学者对 Java 语言的掌握,还专门安排了一章内容来讲述 Java 编程的基础知识。

本书的重点有两方面内容:对 Eclipse 开发环境的掌握和对手机游戏的开发。其中,帮助读者掌握手机游戏的开发技巧是本书的最终目的,为了方便读者理解这部分内容,作者选择了有代表性的赛车、贪吃蛇、三子棋、华容道、俄罗斯方块和使用 MIDP 2.0 开发的 Fighter 等手机游戏供读者研习。

本书可供各大专院校电子类、计算机类专业及其相关专业师生、从事手机游戏开发的工程技术人员及所有热心编程的编程爱好者参考使用。

图书在版编目(CIP)数据

J2ME 手机程序 Eclipse 开发基础 / 郎锐, 孙方编著. —北京: 机械工业出版社, 2006.1

(信息科学与技术丛书)

ISBN 7-111-17991-9

I. J... II. ①郎...②孙... III. ①JAVA 语言—程序设计 ②移动通信—通信设备—应用程序—程序设计 IV. TN929.5

中国版本图书馆 CIP 数据核字(2004)第 141857 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 丁 诚

责任印制: 杨 曦

北京机工印刷厂印刷

2006 年 1 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·17.25 印张·421 千字

0 000—5 000 册

定价: 32.00 元(含 1CD)

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010) 68326294

封面无防伪标均为盗版

出版说明

随着信息科学技术的迅速发展,人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问,在节奏越来越快的工作和生活中,人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书,来获取新知识和新技能,从而不断提高自身素质,紧跟信息化时代发展的步伐。

众所周知,在计算机硬件方面,高性价比的解决方案和新型技术的应用一直备受青睐;在软件技术方面,随着计算机软件的规模和复杂性与日俱增,软件技术受到不断挑战,人们一直在为寻求更先进的软件技术而奋斗不止。目前,计算机在社会生活中日益普及,随着因特网延伸到人类世界的层层面面,掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用,所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待,以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状,机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容,注重理论与实践相结合,内容实用,层次分明,语言流畅,是信息科学与技术领域专业人员不可或缺的图书。

现今,信息科学技术的发展可谓一日千里,机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作,为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

继网络程序开发后的新一轮开发热潮——手机程序开发时代已经来临。目前，由于手机程序开发从业人员少、开发起点高、相关资料缺乏等因素，使得手机程序开发长期处于可望而不可及的境地。尤其是涉及到手机游戏开发的时候，会令人马上联想到是一种需要极强的程序开发能力才能够胜任的工作。其实不然，手机游戏的开发并没有我们想象的那么困难，这种误区实际是由于广大读者对手机开发过程的不了解而造成的。希望通过本书的阅读能够使读者走进 J2ME 的开发世界，从而揭开手机程序开发的神秘面纱。

本书在内容上主要分为三个部分：基础内容的介绍、Eclipse 开发环境的使用和对手机游戏实例的开发。其中，在介绍基础内容时，笔者很清楚 Java 开发环境的正确配置对于一个初学者而言是一件多么困难的事情，所以在这一部分不吝笔墨地对开发环境的配置作了详细的介绍。Eclipse 作为本书所有 J2ME 程序的开发环境，只有在熟练掌握其使用方法之后才能随心所欲地开发后续的游戏实例。所以请读者一定要注意对这一部分内容的阅读并尽可能地熟练掌握。最后进行的游戏实例开发是本书的重点，因为手机是一款娱乐性很强的便携式电子设备，这就决定了在其上运行的程序很大一部分都是游戏或与之类似的娱乐性程序。本书安排的这些游戏实例都是有一定代表性的，读者在理解了范例程序的实现方法后很容易通过适当的引申发挥来开发出其他类似的手机游戏。阅读本书的读者最好能有 C++ 和 Java 的编程经验。在本书配套光盘中提供有全部实例的源程序代码。

最后，在此成书之际，笔者要特别感谢我的父母郎益青先生和张连霞女士在我写作过程所给予的全力支持与鼓励；感谢我的妻子孙方在本书编写期间对我生活的照顾和关心和对本书稿的校核与修改；另外还要对赵振维研究员、罗发根高工等所有为本书写作提供过帮助的人士表示感谢！成书仓促，不妥之处在所难免，诚请读者提出宝贵意见。

郎 锐

目 录

出版说明

前言

第 1 章 Java 2 Micro Edition 概述 1

1.1 J2ME 简介 1

1.1.1 Java 概述 1

1.1.2 J2ME 与 Java 2

1.2 J2ME 体系结构 4

1.3 J2ME 配置 6

1.4 J2ME 简表 8

1.4.1 简表综述 8

1.4.2 MIDP 9

1.4.3 其他几种常见的简表 12

1.4.4 KJava 类 12

1.5 J2ME 规范 14

1.5.1 PersonalJava 简介 15

1.5.2 K 虚拟机简介 16

1.5.3 Java 嵌入服务器简介 16

1.6 使用 J2ME 进行移动开发的

困难 17

1.6.1 有限的设备硬件 17

1.6.2 轻量级的类库 17

1.6.3 缩减应用程序占用的空间 17

1.6.4 最小化无用单元收集过程 18

1.6.5 使用移动入口 19

第 2 章 J2ME 开发环境的配置与 准备 20

2.1 开发环境的搭建 20

2.1.1 安装 JDK 20

2.1.2 安装通用 J2ME SDK 22

2.1.3 安装专用 J2ME SDK 29

2.2 J2ME IDE 开发环境的安装 35

2.2.1 常用的 J2ME IDE 开发环境 35

2.2.2 Eclipse 集成开发环境的获取与 安装 41

2.2.3 EclipseMe 插件的获取与安装 42

2.2.4 对 EclipseMe 插件的配置 48

2.3 其他软、硬件的准备 51

2.3.1 硬件的准备 51

2.3.2 安装手机连接软件 52

2.3.3 常用的图像处理软件 57

第 3 章 使用 J2ME 无线开发 工具包 58

3.1 J2ME Wireless Toolkit 内容 58

3.1.1 安装目录的结构组成 58

3.1.2 J2ME WTK 功能组成 59

3.2 使用 Sun J2ME Wireless Toolkit 62

3.2.1 打开已有项目 62

3.2.2 启动 MIDP 模拟器运行项目 63

3.2.3 新建项目 64

3.2.4 直接执行 MIDlet 66

3.2.5 包装与混淆 66

3.3 使用 SonyEricsson J2ME Wireless Toolkit 67

3.3.1 默认设备的选择 67

3.3.2 新建、打开工程 68

3.3.3 使用模拟器 70

3.3.4 打包与混淆 71

第 4 章 深入了解 Eclipse 工作台 72

4.1 工作台及其元素 72

4.2 工作台的基本操作 78

4.2.1 创建简单项目 78

4.2.2 文件的导入、导出 79

4.2.3 搜索、任务与书签的使用 80

4.2.4 透视图 86

4.2.5 使用比较操作 88

4.3 进行 J2ME 开发时的工作台 90

4.3.1 J2ME 项目与 Java 构建器 90

4.3.2 在 J2ME 开发中常用的透视图 ...	91	6.2.4 模板的使用	133
4.3.3 快速修正	93	6.2.5 组织导入	134
4.3.4 调试器及其相关元素	95	6.2.6 维护历史版本	136
第 5 章 Java 程序设计简介	97	6.2.7 抽取方法	137
5.1 面向对象编程	97	6.3 创建类	140
5.1.1 抽象	97	6.3.1 添加类	140
5.1.2 封装	98	6.3.2 覆盖/实现方法	142
5.1.3 继承	99	6.3.3 生成 Getter 和 Setter	143
5.1.4 多态性	100	6.4 运行与调试	144
5.2 运算符	101	6.4.1 运行程序	144
5.2.1 赋值运算符	101	6.4.2 调试程序	149
5.2.2 算术运算符	101	6.4.3 向真机发布程序	149
5.2.3 关系运算符	103	第 7 章 图形界面编程实例	
5.2.4 逻辑运算符	104	“Paint”	155
5.2.5 位运算符	107	7.1 概述	155
5.2.6 强制类型转换运算符	108	7.1.1 设计思路概述	155
5.2.7 运算符优先级	109	7.1.2 MIDlet 图形界面编程基础	155
5.3 控制语句	109	7.2 高级界面的实现	157
5.3.1 条件语句	110	7.2.1 创建程序框架并实现信息	
5.3.2 分支语句	111	提示框	157
5.3.3 循环语句	112	7.2.2 添加选项列表并实现 Ticker	160
5.3.4 跳转语句	115	7.2.3 接受用户的文本输入	165
5.4 基本语法	117	7.2.4 Form 类的使用	167
5.4.1 基本词汇	117	7.3 低级界面的绘制	174
5.4.2 数据类型	118	7.3.1 在画布上进行绘制	174
5.4.3 变量与数组	119	7.3.2 显示图像	175
5.4.4 类与方法	121	7.3.3 使用字体	177
5.4.5 命名空间管理	122	7.3.4 简单图形的绘制	178
5.4.6 静态成员	123	第 8 章 游戏实例开发	181
第 6 章 创建第一个实例 “Hello		8.1 手机游戏设计概述	181
World”	124	8.1.1 手机游戏与传统游戏开发的	
6.1 创建 MIDlet 项目	124	区别	181
6.1.1 检验 Eclipse 设置	124	8.1.2 手机游戏开发的特点	182
6.1.2 创建 J2ME 项目	125	8.1.3 手机游戏设计的基本原则	182
6.1.3 添加 MIDlet 到项目	126	8.1.4 手机游戏的实现技术	183
6.2 编辑 J2ME 元素	129	8.1.5 小团队开发手机游戏的策略 ...	183
6.2.1 进一步了解 Java 编辑器	129	8.2 赛车游戏实例开发	187
6.2.2 添加新方法	131	8.2.1 剧本设计	187
6.2.3 问题的标定与排查	131	8.2.2 框架搭建	188

8.2.3 界面实现	191	8.6.3 新方块组的产生	233
8.2.4 角色移动与碰撞检测	193	8.6.4 方块组的移动与旋转处理	236
8.3 贪吃蛇游戏实例开发	197	8.6.5 下落与消行	240
8.3.1 剧本设计	197	第 9 章 MIDP 2.0 编程实例	
8.3.2 框架搭建	197	“Fighter”	244
8.3.3 界面与控制的实现	199	9.1 MIDP 2.0 的新特性	244
8.4 三子棋游戏实例开发	207	9.1.1 MIDP 2.0 游戏开发概述	244
8.4.1 剧本设计	207	9.1.2 新的游戏 API	244
8.4.2 框架搭建	207	9.1.3 主动查询用户按键	249
8.4.3 界面实现	214	9.1.4 使用后台屏幕缓存	249
8.4.4 实现人工智能	218	9.1.5 使用图层	250
8.5 华容道游戏实例开发	222	9.1.6 使用精灵	251
8.5.1 剧本设计	222	9.2 Fighter 实例开发	252
8.5.2 界面的框架实现	222	9.2.1 剧本与游戏框架设计	252
8.5.3 光标、图块的移动处理	225	9.2.2 移动的字符	252
8.6 俄罗斯方块游戏实例		9.2.3 添加精灵坦克	256
开发	230	9.2.4 使用背景	258
8.6.1 剧本设计	230	9.2.5 实现对战	259
8.6.2 框架搭建	230		

第 1 章 Java 2 Micro Edition 概述

本章主要介绍有关 Java 和 J2ME 编程的一些基本概念和体系结构，并阐述了 J2ME 开发周期、体系结构、配置、简表以及规范等具体内容。本章概念性和了解性的内容较多，是全书的基础部分，为后续章节的进行提供了必要的知识储备。

1.1 J2ME 简介

1.1.1 Java 概述

如果您曾经接触过与 Java 2 Micro Edition 开发相关的文档或资料，相信您一定在初次接触之后对诸如 KVM、CLDC 和 MIDP 等一大堆的技术名词不知所云。因此在开始学习本书内容之前，首先要了解有关 J2ME 的一些基本概念与知识。作为 Java 2 的缩简版本，在开始 J2ME 内容的学习之前有必要先了解一些有关 Java 的基本知识。

Java 是一种全新的计算机技术，它是建立在使同样的软件能够在不同计算机系统上运行的思想上的。这些计算机系统包括消费型器件和其他一些设备。利用 Java 技术，您能让同一种软件应用在各种各样的机器上，如：PC、Macintosh 系统或者计算机网络上。

Java 既可以看作是一种真正的程序设计语言，也可以看作是一个完整的平台。作为一种程序语言，它简洁、面向对象、安全、健壮并适用于 Internet；作为一个平台，它通过那些符合 Sun 公司发布的标准 API 开发的应用，在操作系统、数据库和中间件等方面提供服务。Java 能够做得比其他语言更灵活，它可以现在运行在桌面系统、服务器、数据库、smart 卡、PDAs、smart 电话、机顶盒等许多方面（见图 1-1）。下面是 Java 的一些典型应用：

- GUI 应用和 applets；
- 分布式组件（EJB，RMI，CORBA 等）；
- 数据库以及数据库控制；
- Web 服务器；
- 消费设备和内置设备的应用（移动电话、PDA、smart 卡、便携摄像机等）；
- 其他任何在计算机上所能想到的应用。

自 1995 年 Java 被公诸于众，就立即引起了广泛关注。下面的时间表记载了在过去几年里发生的关于 Java 的关键性事件：

1995 年：Java 技术发行，开始的重点是 applets。

1996 年：Java Development Kit (JDK) v 1.0 发布。包含核心层的功能如套接字编程、文件 I/O 和 GUI 等。

1997 年：JDK 1.1 发布并引入 Java GUI、JDBC 数据控制、RMI 分布对象等几个增强的方面。

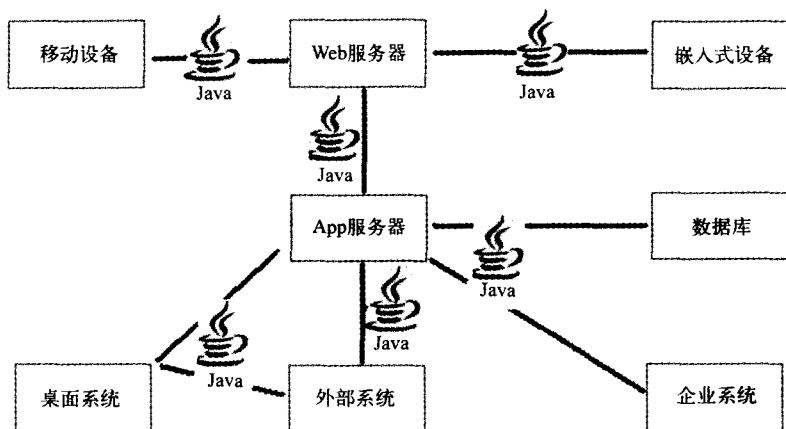


图 1-1 Java 的应用

1998 年：JDK 1.2（称 Java 2）发布。JFC/Swing 当年被下载了 500 000 多次。

1999 年：Java 技术被划分成 J2SE、J2EE 和 J2ME。

2000 年：JDK v1.3 发布。

2002 年：JDK v1.4 发布。

2004 年：JDK v1.5 发布。

Java 之所以能够如此成功，这与其奇妙的程序设计语言（它没有类似与 C/C++ 的内存问题、高度动态而不像 C/C++/VB、安全以及针对 Internet 的程序设计）和平台无关性（一次编译，到处运行）不无关系。Java 甚至还推出一个公共的途径以代替像 Windows、MQ Series、Sybase DB Library 等一样的私有技术。

1.1.2 J2ME 与 Java

Java 2 Micro Edition（Java 2 袖珍版）通常简称为 J2ME，属于整个 Java 技术范畴中的一个子集，在认识它之前必须首先对其在整个 Java 技术之中的定位做个了解。

如果您熟悉 Java 的沿袭历史，相信您一定听说过 Java 技术一开始并非就叫做 Java，而是叫做“OAK”，而且它原本是专为机顶盒这种嵌入式系统设计的。但令人惊奇的是它并没有顺利地进入嵌入式领域，而是随着 Internet 的发展占领了 PC 端和 Server 端，并在被抢注了商标后将其更名为 Java。随着技术的不断发展和成熟，Java 技术逐渐派生出一些并未出现在原本构思蓝图中的其他应用，并应用到许多企业级系统中去，但是 Java 技术从骨子里来说还是非常适合应用在嵌入式系统当中的。当初自 Java 1.0 发布之后，Java 被广泛地使用在桌面应用程序和 Applet 的开发上，但是从 Java 1.1 开始，Java 又开始重复其过去的职责——嵌入式系统方面的应用。例如当时 Sun Microsystems 陆续推出的 Embedded Java 和 Personal Java（PJava）就是分别为资源十分有限且无显示设备的嵌入式设备和能够与网络连接且拥有显示系统的消费性电子设备而设计的。在进入 Java 2 版本后，为了明显区分各种 Java 的应用，又相继推出了 J2EE、J2SE 以及 J2ME 这三种不同应用的 Java 版本（分别是企业版、标准版以及袖珍版）。按照 Java 2 的策略，这三种 Java 版本的应用方向分别为基于 Internet 技术的企业网、PC 和嵌入式设备或消费性电器（参见图 1-2）。正是由于 Java 2 将 Java 应用分

成三大块，才使得 Java 程序语言的发展不会再像 Java 1.1 时那样如树枝般扩散，也不会出现今后不可收拾的混乱局面。

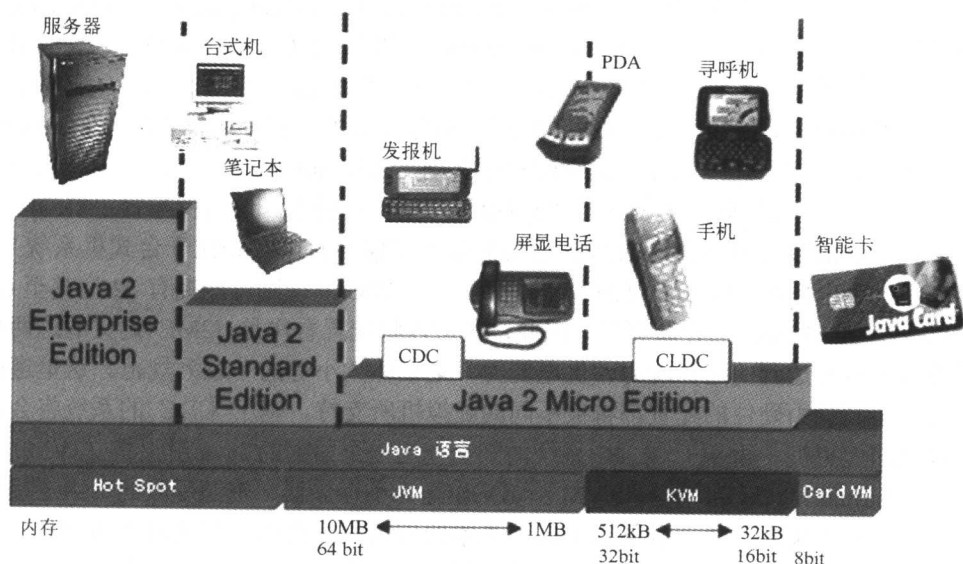


图 1-2 不同 Java 版本的应用

不同的 Java 版本，在其支持的核心类别函数库的完整性以及所支持的 Java 基本类别这两方面均有所差异，但对程序设计人员而言，这些版本的关系如图 1-3 所示。也就是说，不管开发的是企业用的 Java 程序，还是在嵌入式装置上执行的 Java 程序，或是在浏览器上执行的 Applet，都必须首先安装 J2SE，然后再安装各种版本的核心类别函数库及额外的扩充类别函数库，如此这般才能开发出各种用于不同目的的 Java 程序。

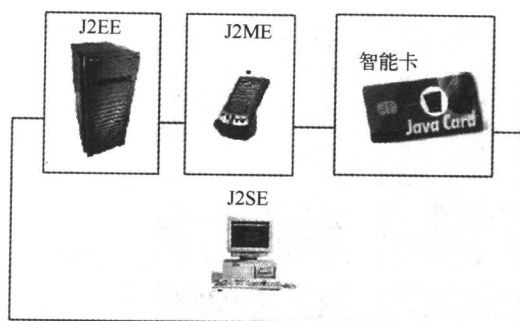


图 1-3 不同 Java 版本之间的关系

作为本书核心的 J2ME 是 Sun 公司在 J2SE 和 J2EE 之后于 1999 年 6 月推出的，用以满足消费性电子产品和嵌入式设备的需要。它是为了那些使用有限的资源和网络连接（通常是无线连接）以及有限图形用户界面能力的设备开发的，其最初的目标是 16 位或 32 位处理器、16 MHz 时钟频率、512KB 甚至更少内存的设备。J2ME 将所有的嵌入式装置大体上区分为以下两种：一种是运算功能有限、电力供应也有限的嵌入式装置，如 PDA 和手机等；

另一种是运算能力相对较佳、电力供应也相对充足的嵌入式装置，如空调、冰箱和机顶盒等。针对这两种型态的区分，Java 引入了一个概念：“配置（configuration）”并将前述两种区分分别定义在“有限连接设备配置”（Connected Limited Device Configuration，简称 CLDC）和“连接设备配置”（Connected Device Configuration，简称 CDC）中。也就是说，J2ME 先把所有的嵌入式装置利用“配置”的概念区隔成两种抽象的型态。在 CLDC 这个配置中，定义了 Java 应用程序接口以及支持手持设备的技术，就像 Sun 的文档中所描述的“devices that you hold in your hand（手持设备）”。CDC 这个配置定义则支持像 Sun 文档中所说的“devices that you plug into the wall（从墙取电的设备）”的应用程序接口和技术。这两种配置主要的区别就在于它们所应用于的装置的能力，CLDC 设备的处理器能力有限（相比于台式机系统），并且存储器大小一般也只在 128 KB 到 512 KB 之间。而 CDC 系统则可能拥有 32 位甚至 64 位处理器，以及超过 512KB 的存储容量。因此，也可以将配置看作是 J2ME 定义的这两种类型嵌入式设备至少要达到的运算能力、供电能力、记忆体大小等的一系列规范。这里遵循一个原则：每个不同能力硬件的配置都将被不同的虚拟机所支持。基于 CDC 的系统将会使用一个功能强劲的虚拟机，而基于 CLDC 系统的则会使用一个紧缩的 Java 虚拟机（JVM）——K 虚拟机（K Virtual Machine，简称 KVM）。KVM 是为小存储器、资源受限的网络连接设备而专门设计的。每个配置代表一种低水平的、基本的应用程序接口，在这两个相似基础之上是简表（profile），用于特殊设备的额外的应用程序接口。之所以有简表的概念，是为了要更明确地区分出各种嵌入式设备上 Java 程序该如何开发以及它们应该具有哪些功能。因此在简表中定义了与特定嵌入式设备非常相关的扩充类别函数库。简表中所定义的扩充类别函数库是根据底层配置内所定义的核心类别函数库所建立的。下面通过图 1-4 所示的这张由 SUN 提供的图表来进一步解释其中的关系。图中 J2ME 的体系结构被横向地分成三层，纵向分成两部分。配置包括一个控制配置核心类的虚拟机，具体的简表位于每个配置之上。简表的实现是 Java 应用程序接口的一个集合，用于适应被定义配置的应用程序接口提供的服务，简表是一个完整的运行环境，一个在简表上执行的应用程序并不需要额外的支持类。

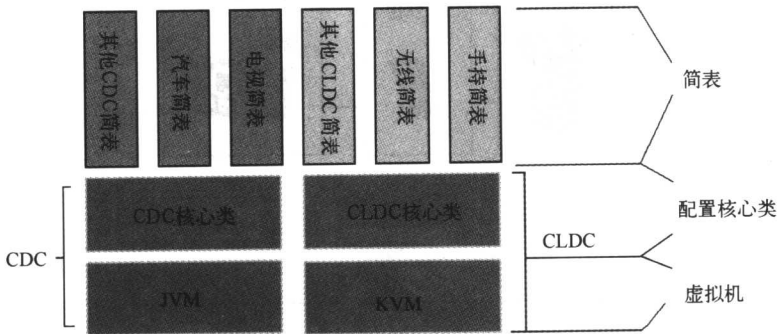


图 1-4 配置与简表的体系结构

1.2 J2ME 体系结构

学习 J2ME 最重要的部分是掌握 J2ME 的体系结构，J2ME 是 Sun 公司针对嵌入式消费

类电子产品推出的开发平台，事实上它是一系列技术和规范的总称，在众多的技术和规范帮助下，J2ME 才能满足众多消费产品的不同需要。

现在个人计算机系统的数量和种类已经发展得非常繁多，各种信息家电、个人数字助手（PDA）、机顶盒、POS 终端以及其他消费电子设备等层出不穷。如此繁多的设备拥有各自不同的特性和界面。为了使 Java 能够更好的应用于这些信息设备，就要求把 Java 的精髓压缩进一个非常小的程序包中，这就是 J2ME。和使用所有 Java 技术一样，J2ME 的核心也在一种虚拟机中。最初，用于 J2ME 应用程序虚拟机的被称作“Kilobyte Virtual Machine”（千字节虚拟机）或简称 KVM。正如它名称的含义，KVM 比较小，通常只有 128KB 或更少。这比起 J2SE 的 Java 虚拟机（JVM）的 32 MB 来说就小得多了。

J2ME 使用配置和简表定制 Java 运行时环境（JRE）。作为一个完整的 JRE，J2ME 由配置和简表组成，它们提供了用于特定 J2ME 环境的类应用程序接口（见图 1-5）。每个配置和简表用来处理一般或具体的消费产品，其规范由各个设备生产商和用户共同开发并建立。配置是用于一组通用设备的最小的 Java 平台，它决定了所使用的 JVM，常常归为一种横向的设备分组和一套纵向的服务于广泛的设备的 API，相对来说，横向分组设备是那些共享相同的内存安排、通信带宽、能量需求以及用户能力的设备，一般认为配置能够提供这些众多的设备的所有需求。而简表则为一类具有某些重要特性的设备家族或特别的应用程序提供更具体的能力。这些特性都很重要，但是它们又不是所有的设备上都具有的。从图 1-5 可以看出，作为 J2ME 技术核心的 Java 虚拟机也是在配置这一层规定的。

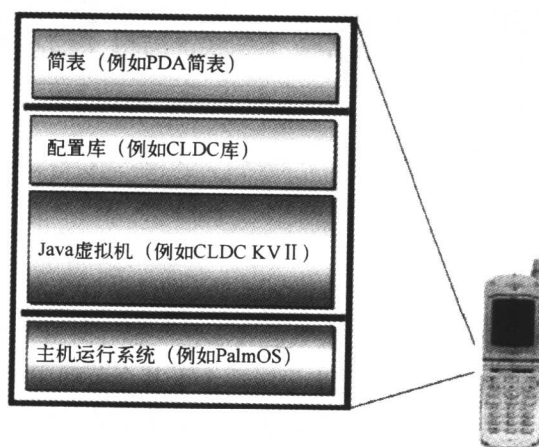


图 1-5 J2ME 层次

在刚开始接触 J2ME 时会经常对上述事实感到困惑。事实上，Sun 的第一个配置（现在只是一种配置的引用实现）带有称为 KVM 的虚拟机引用实现，KVM 满足配置的虚拟机的必要条件。另一方面，简表完善了配置，为某个具体的设备家族或工业片段应用程序提供更高层次的性能。也就是说，简表为某一种具体的设备比如手机提供了更优良的性能与服务。这里的关键在于简表必须完善了配置，如果没有配置和虚拟机提供的核心类应用程序接口与运行时环境的话，显然简表并不会做任何工作。

一般来说，简表是一种给定的垂直分组设备提供界面、输入法以及持久性等的机制。这种简表将被认为是开发此类设备应用程序的完整工具包。例如手机简表或个人数字助手

(PDA) 简表等。还有其他一些简表可以为更大范围的设备提供一些特殊的功能，如远程方法调用 (RMI) 等。

为了能够更清楚地理顺虚拟机、配置与简表的关系，这里对 J2ME 的体系结构做一下简化：如果是为小型信息设备开发 Java 应用程序，那么就要有一个配置和至少一个的简表。现在一般是配置捆绑了虚拟机和一套针对开发平台所能够使用的横向分组设备的 Java 类库。除此之外，至少还需要一个简表为平台提供附加的 Java 类，这个简表通常会为所要开发的设备提供用户界面、输入和数据库类等。这就构成了开发所需的基本 J2ME 环境。

图 1-6 给出了不同虚拟机、配置和简表之间的关系。这里也一并将 J2SE API 和它的 Java 虚拟机进行了比较。虽然 J2SE 虚拟机通常也被称为一种 JVM，但实际上 J2ME 虚拟机、KVM 以及 CVM 都是 JVM 的子集。KVM 和 CVM 均可被看作是一种 Java 虚拟机，只不过它们是 J2SE JVM 的压缩版，并特定于 J2ME 而已。

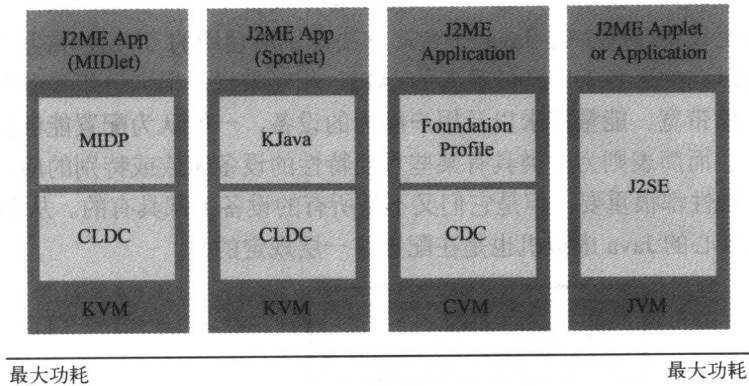


图 1-6 不同虚拟机、配置和简表之间的关系

1.3 J2ME 配置

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的设备可用的规范，这就是“配置”。读者可以将其理解为对硬件的描述，所以通过定义配置的方法就能够清楚地描述硬件功能。J2ME 可以在好几个不同的配置中进行配置。正如前面所述，每个配置为一组通用设备提供最小的 Java 平台。通过配置将基本运行时环境定义为一组核心类和一个运行在特定类型设备上的特定 JVM。虽然将来可能会定义有其他新的配置，但是到目前为止 J2ME 仅有有限的两种配置：CLDC 和 CDC。

从设备的硬件功能角度来说，CLDC 是为使用较小的存储容量的设备设计的（参见图 1-7）。它主要针对拥有 16 位、32 位的主频在 16MHz 以上的处理器，设备内存在 128KB 到 512KB 之间的消费电子设备。CLDC 使小型设备所拥有的资源只要比一张智能卡多一点就可以进行 Java 编程了，由于它的大小限制比 CDC 要严格得多，所以进行 CLDC 开发将要比 CDC 更有挑战性；CDC 主要针对 32 位的处理器，主频通常在 75MHz 以上，内存比 PC 机要小得多但是比 CLDC 的 512KB 内存要多得多，通常在 1~4MB 之间。CDC 使小型设备只要具有少

量的资源，至少比台式机要少的资源就能够进行 Java 编程。

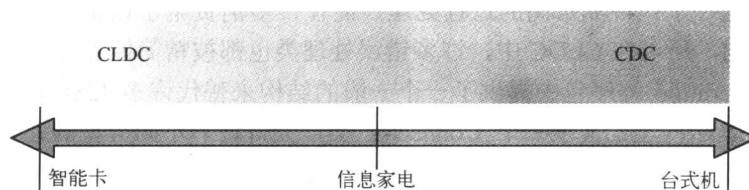


图 1-7 设备覆盖的范围

由于 CDC 与 CLDC 所针对的设备 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 的核心类库比 CLDC 的核心类库要大，它使用的是 C 虚拟机（CVM），CLDC 则使用的是 KVM。CVM 比 KVM 包含更多的功能和支持更多的特性。相应的，运行在 CDC 配置上的程序将在 CVM 上进行解释，称为 J2ME Application；运行在 CLDC 配置上的程序将在 KVM 上进行解释，称作 MIDlet。

如果从设备的用途和功能的角度来看，CDC 主要针对数字电视、机顶盒、网络电话、汽车导航及娱乐系统等。这几类设备都有如下几个共同特点：有线连接、电源稳定、设备资源比较受限。CLDC 针对的设备主要是各种移动手持设备，例如：PDA、POS 机和手机（智能）等。这一类设备的最大特点就是无线连接、没有稳定的电源供应（通常使用电池）、设备资源极少。目前 CLDC 在这一领域应用最广的还是在智能手机上的应用。

除了在容量和能力方面对虚拟机规定了必要条件外，配置还规定了类应用程序接口至少应包括 java.io、java.net、java.util 和 java.lang 等几个程序包。根据需要，配置可能还会包括其他的程序包。

CLDC 的起源可以追溯到 1999 年 JavaOne 大会上介绍的 Sun 的第一个袖珍版 Java 和第一个 KVM 及其相关类库，尽管 CLDC 和所有的配置都满足成为虚拟机的条件，但它在当时却并不是虚拟机。根据规范中所说，运行 CLDC 的设备应该有 512KB 或更少的内存空间、一个有限的电源供给（通常是电池）、有限的或断断续续的网络连接（9600 bit/s 或更少）以及多样化的用户界面甚至没有用户界面。与 J2SE 相比，CLDC 缺少下列几个特征：

- AWT（抽象窗口开发包），Swing 或其他图形库；
- 用户定义类装载器；
- 类实例的最终化；
- 弱的引用；
- RMI；
- 映射（reflection）。

CLDC 共有四个包：三个核心类别函数库 java.lang、java.util、java.io 和一个扩充类别函数库 javax.microedition。除了 microedition 包以外，其他三个包都是 J2SE 包的核心子集。虽然 CLDC 采用了这些 J2SE 类库，但由于它只有 512KB 的内存空间，而 RMI 和映射需要的内存太大，加之配置必须满足为一组通用设备提供最小的 Java 平台，因此 CLDC 不得不将其中与微型设备无关的类、属性和方法都精简掉，使得 CLDC 类库与 J2SE 类库有许多细微差别。在个人移动信息设备领域中，就有许多系统不支持 J2SE 的高级特征。例如，许多消费电子产品不能支持浮点数，因此 Float（浮点类）和 Double（双精度类）就被删除了。另

外，许多系统没有或不提供访问一个文件系统的功能或权限，所以与文件有关的类也被丢弃了。错误处理也是一个代价非常高的过程处理，而在许多消费电子设备中，故障恢复是很难的甚至是不可能的。所以在 CLDC 中，许多错误处理类也都被精简掉了。

在 java.microedition 程序包中提供了一个一般的结构来替代许多 J2SE 网络输入/输出类。CLDC 一般连接器结构还定义了一个 Connector 类，允许许多不同类型的连接能够使用静态方法，表 1-1 列出使用同一个 Connector 类创建和打开五种不同类型的连接的方法：

表 1-1 Connector 类打开连接的方法

类 型	打 开 方 法
HTTP	Connector.open("http://www.xyz.com");
套接字	Connector.open("socket://111.222.111.222:9000");
通信端口	Connector.open("comm:1;baudrate=9600");
数据报	Connector.open("datagram://111.222.111.222");
文件	Connector.open("file:/xyz.dat");

一般连接器结构提供给应用程序开发者一个到通用低水平硬件的简单的映射表。在成功执行 open 语句后将返回一个实现一般连接界面的对象。

与 CLDC 相比，CDC 占领了个人电脑到拥有 512KB 内存的小型设备之间的中间地带。CDC 是基于 J2SE 1.3 应用程序接口的，包含了定义在 CLDC 规范中的所有 Java 应用程序接口。与 CLDC 相比，CDC 补充完全了 CLDC 所有缺少的特性和类，如：映射、错误处理、浮点数、属性、输入/输出（File、FileInputStream 等等）和弱的引用等。一般说来，CDC 中预期的类包括一个 J2SE 的子集和一个完整的 CLDC 超集，可用图 1-8 表示。

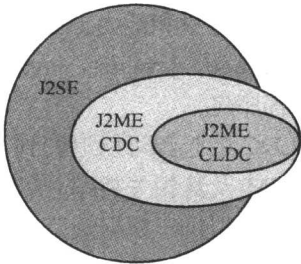


图 1-8 J2SE、CLDC 和 CDC 的关系

1.4 J2ME 简表

1.4.1 简表综述

在区分出 CDC 和 CLDC 这两种主要的配置之后，J2ME 在此基础上引申出了简表的概念。简表可以说是架构在配置之上的规范。之所以有简表的概念，是因为配置必须满足所有的设备的最低要求。而用户界面、输入机制和数据持久性等有高度地设备具体性，即每一种设备都有自己的用户界面、输入机制和数据存储方法，这些往往不在配置所满足的最小要求的范围之内。这样，虽然配置为一组通用设备提供了最小的 Java 平台，但是应用程序开发者感兴趣的是为某个特定的设备生产应用程序，如果仅依靠配置的话，他们编写的应用程序将存在不同程度的欠缺。为了能够更明确地区分出各种嵌入式设备上 Java 程序该如何开发以及它们所应具备的功能，这才引入简表的概念。简表中为相同消费电子设备的不同的生产商定义了与特定嵌入式设备密切相关的扩充类别函数库。事实上，虽然配置规范的开发由 Sun 公司领导，但是许多简表规范仍将继续由特殊设备的供应商领导。例如，Motorola 领导

了移动电话和传呼机简表规范的开发，又如 Palm 领导了 PDA 简表的开发等。

这里再补充说明一点：CDC 和 CLDC 规范中也都定义了基本的 API 集合，这些 API 提供 Java 的基本功能，例如：java.io、java.lang、java.util 和 javax.microedition.io，这些包就是定义在 CDC 和 CLDC 中的。不过这些包都是 Java 最基本的功能，更多的面向设备的功能性 API 必须还是通过简表来提供，以完善配置的不足。现在，五个已知简表已经有了规范，表 1-2 列出了这五个简表的具体情况。下面通过图 1-9 来进一步给出嵌入式系统上 Java 程序、简表、配置、虚拟机、操作系统以及硬件之间的相互关系。

表 1-2 简表及其完善的配置

简 表	完 善 配 置
MIDP (Mobile Information Devices Profile, 移动信息设备简表)	移动电话和寻呼机 CLDC
Personal digital assistant profile	Palm 和 Handspring 的 PDA 设备 CLDC
Foundation profile	用于所有不需要 GUI 的 CDC 设备的标准简表 CDC
Personal profile	替代 PersonalJava 的 Foundation 完善的简表 CDC
RMI profile	提供 RMI 的 Foundation 完善的简表 CDC

从图 1-9 中可以看出，根据某个简表规范所撰写的 Java 程序除了可以直接使用简表中定义的扩充类别函数库外，还可以直接调用配置里定义的核心类别函数子集合与扩充类别函数库。

除了上述几个简表，还有一个 Java 类库集在表现上更像一个简表。当 Sun 为 Palm 开发第一个 KVM 时，他们需要一组类来开发 Palm 的演示程序。这套类库最后被封装到 com.sun.kjava 程序包，在 CLDC 早期开发中，这些类曾被广泛地用来测试和演示 J2ME。由于 KJava 是唯一允许开发者使用 J2ME 和 KVM 开发应用程序的类，所以很快就被推广。直至今日，尽管用于 PDA 或更特殊一点的 Palm 的简表都已经推出，而且 KJava 类不被支持，仅仅用于设计测试程序或演示程序，并且将来可能被其他新的简表所替代，但许多开发人员仍热衷于使用它进行开发。

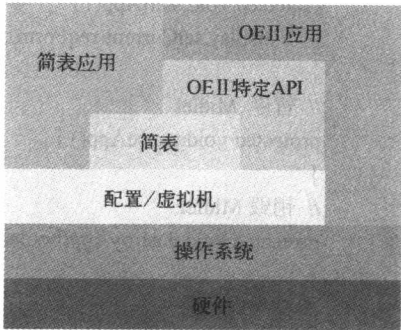


图 1-9 简表与 Java 程序、配置等的关系

1.4.2 MIDP

与配置相比，简表更多地是针对软件接口的定义，简表有必须实现的功能，也有可选的功能，因此简表更加灵活。在目前这几个简表中最为重要的显然是 MIDP。MIDP 定义了能在 Java 手机上运行的 Java 程序的规范，更加面向用户，而且比 CLDC 更高级，具体包括应用程序生命周期、各种用户接口界面组件、支持持久存储、网络 and 用于移动电话的计时器、双通道呼叫器和其他无线电设备等。由于 MIDP 和 CLDC 两者都有引用实现，所以这里首先通过一段例程来分析一下这个简表：

```
// 导入需要引用的 J2ME 类
```