

编程

指南

陆昌辉 主 编  
曾 健 熊 帅 周 魏 等编著



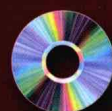
# J2ME

# 应用开发详解



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



光盘内容为  
书中范例源文件

# J2ME 应用开发详解

陆昌辉 主编

曾 健 熊 帅 · 周 魏 等编著

電子工業出版社·

**Publishing House of Electronics Industry**

北京·BEIJING

## 内 容 简 介

本书以通俗的语言，系统地介绍了 J2ME 移动应用开发的设计思想以及编程方法，为读者使用这些技术和标准进行工作提供了很好的指导。本书在撰写中坚持理论与实践并重，面向实际开发，解决实际问题。全书共分为 11 章，主要内容包括：引入 J2ME 平台、J2ME 中的开发与测试环境、MIDlet 介绍、MIDlet 中的 GUI 开发、Servlets 与 MIDlet、RMS、MIDP 2.0 的 Game 类、J2ME 游戏开发，最后作者还精心挑选了任天堂 Battle City 等 3 个手机游戏开发的典型案例，对其开发过程进行了深入细致的描述，并给出了具体代码，便于读者巩固与提高。

本书在介绍具体知识点的同时，还穿插了大量简洁、生动的小实例，最后还介绍了三个典型手机游戏的完整开发过程，难度由低到高、内容循序渐进，并注重技巧的归纳和总结。

本书适合于具备一定 Java 语言基础的专业和准专业的 Java 开发人员阅读，可以作为高等院校 J2ME 移动应用开发课程的教材，也可作为移动应用开发人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目(CIP)数据

J2ME 应用开发详解 / 陆昌辉主编. —北京：电子工业出版社，2009.4  
ISBN 978-7-121-07637-4

I. J… II.陆… 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.TP312  
TN929.5

中国版本图书馆 CIP 数据核字 (2008) 第 167009 号

责任编辑：郭鹏飞

印 刷：北京市天竺颖华印刷厂

装 订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：19.5 字数：499 千字

印 次：2009 年 4 第 1 次印刷

印 数：4000 册 定价：36.00 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：(010) 88258888。

# 前 言

在消费电子和嵌入式设备的广阔领域中，目前最受关注的是手机等移动通信设备。随着网络和无线通信技术的发展，以及无线移动设备的处理能力不断增强，使得手机已经不再是一个单一的通信设备，它融合了上网、个人信息管理、移动办公、娱乐游戏等功能。各种无线移动通信终端正朝着多功能、智能化的方向发展。并且随着 3G 网络的运行，数据通信带宽的大幅增加，而费用的急剧降低，各种针对移动技术的应用需求会越来越大。

J2ME 为运行在嵌入式消费类电子产品的设备，如移动电话、PDA、游戏终端之上的应用程序提供了一个健壮的、灵活的环境。J2ME 还提供了灵活的图形用户界面、健壮的安全模型、广泛的联网协议支持。目前 J2ME 平台已经部署到上亿个设备上，前景非常广阔。因此越来越多的人意识到了 J2ME 技术的开发与应用所带来的无限机遇。

针对这一现实情况，作者根据多年 J2ME 开发经验编写了本书，通过深入浅出的基础理论讲解和丰富的案例分析方式，引导读者在学习 J2ME 基础理论的同时，将其运用于实际的应用开发当中。

本书共分为 11 章，具体内容组织如下：

第 1 章介绍了 J2ME 平台的相关知识，具体包括 J2ME 与 WAP，J2ME 的体系结构，开发 J2ME 程序的注意事项，无线编程的新趋势，J2ME 配置、简表以及工具箱等内容。

第 2 章讲解了 J2ME 中的开发与测试环境的相关内容。

第 3 章介绍了 MIDlet 的相关知识，具体涉及 MIDlet 的生命周期，用命令行环境开发 MIDlet 实例，用 KTOOLBAR 开发 MIDlet 实例，MIDlet 与 XML 等内容。

第 4 章介绍了 MIDlet 中的 GUI 开发方法，具体包括 MIDP 图形坐标系统、部署不同的 MIDP GUI API、MIDP GUI 编程中的 GUI 组件、编写应用的代码等内容。

第 5 章讲解了 Servlets 与 MIDlet 的相关知识，内容涉及用于 MIDlet-Servlets 内部通信的类和接口、实现 MIDLETS 和 SERVLETS 之间的通信方法等方面。

第 6 章主要介绍了 RMS 包的相关知识，以及如何利用 RMS API 编写 MIDLET 应用等内容。

第 7 章对 MIDP 2.0 的 Game 类进行了详细介绍，具体涉及 GameCanvas 类、GameCanvas 基础实例、Sprite 类、LayerManager 类、TiledLayer 类等内容。

第 8 章重点介绍了 J2ME 游戏开发的相关知识，具体包含与传统游戏的差别，市场推广的方法，移动游戏的限制，编码之前的准备工作，以及开发过程控制等内容。

在第 9、10、11 章这 3 章中，作者精心挑选了 3 个手机游戏开发的典型案例，并对其设计与开发过程进行了深入细致的描述，便于读者的学习与提高。

本书以通俗的语言，系统地介绍了 J2ME 移动应用开发的设计思想以及编程方法，为读者更好地使用这些技术和标准进行工作而提供了很好的指导。本书既有简单易懂的代码片断，也有大量实际可用的应用系统实例，为读者迅速掌握 J2ME 的核心 API 类库及无线应用系统的开发过程提供了很好的参考。

本书内容全面，逻辑与结构合理，讲解由浅入深，示例丰富，注释详尽。本书适合于具备一定 Java 语言基础的专业和准专业的 Java 开发人员阅读，可以作为高等院校 J2ME 移动应用开发课程的教材，也可作为移动应用开发人员和希望利用 J2ME 技术寻求移动解决方案的其他程序员或管理人员的参考书。

本书由陆昌辉主编，曾健、熊帅、周魏等编著。因作者水平有限，尚有许多不周和不准确之处，恳请广大读者提出批评和建议。

编著者  
2009年2月

# 目录

## C O N T E N T S

第 1 章 引入 J2ME 平台 .....	1
1.1 J2ME 和 WAP .....	1
1.2 J2ME 的体系结构 .....	2
1.2.1 配置概述 .....	2
1.2.2 简表概述 .....	3
1.2.3 J2ME 的目标设备 .....	3
1.3 开发 J2ME 程序注意事项介绍 .....	3
1.3.1 设计开发小型设备应用程序要注意的事项 .....	3
1.3.2 设计开发移动设备应用程序的注意事项 .....	4
1.3.3 性能注意事项 .....	4
1.3.4 编译注意事项 .....	4
1.3.5 打包和部署注意事项 .....	5
1.4 无线编程的新趋势 .....	5
1.4.1 J2ME 使能设备 .....	5
1.4.2 J2ME 使能设备的优点 .....	6
1.4.3 服务器应用和应用门户 .....	6
1.4.4 J2ME 与其他 Java 技术的汇集 .....	7
1.4.5 Java 卡和 J2ME 灵巧卡的演化 .....	8
1.4.6 J2ME 在 GADGET 编程中的作用 .....	8
1.4.7 用 J2ME 的娱乐软件编程 .....	9
1.4.8 J2ME 与多媒体的汇集 .....	10
1.5 J2ME 配置 .....	11
1.5.1 连接限制设备配置 (CLDC) .....	12
1.5.2 CLDC 要求 .....	13
1.5.3 CLDC 与 CDC 的比较 .....	13
1.6 J2ME 简表 .....	14

1.6.1	KJava 简表 .....	14
1.6.2	MIDP 简表 .....	14
1.6.3	MIDP 取代 KJava .....	15
1.7	J2ME 无线工具箱 .....	16
1.7.1	J2ME 无线工具箱的结构 .....	16

## 第 2 章 J2ME 中的开发与测试环境 ..... 19

2.1	安装和配置准备 .....	19
2.1.1	Eclipse 及 EclipseMe 插件的安装 .....	19
2.1.2	配置 EclipseME 插件 .....	20
2.2	开发 J2ME 应用程序 .....	21
2.2.1	创建 J2ME 项目 .....	21
2.2.2	开发 J2ME 项目 .....	24
2.3	J2ME 的测试环境 .....	26
2.3.1	KToolbar 介绍 .....	26
2.3.2	运行 J2ME 应用程序 .....	27

## 第 3 章 MIDlet 介绍 ..... 29

3.1	MIDlet 的生命周期 .....	30
3.2	用命令行环境开发 MIDlet 实例 .....	32
3.2.1	问题描述 .....	32
3.2.2	问题分析 .....	32
3.2.3	编码 MIDlet .....	33
3.2.4	编译 MIDlet .....	35
3.2.5	预检验 MIDlet .....	36
3.2.6	打包 MIDlet .....	36
3.2.7	测试 MIDlet .....	37
3.3	用 KTOOLBAR 开发 MIDlet 实例 .....	37
3.3.1	问题描述 .....	38
3.3.2	问题分析 .....	38
3.3.3	编码 MIDlet .....	40
3.3.4	执行 MIDlet .....	43
3.3.5	验证输出 .....	45
3.4	MIDlet 与 XML .....	45
3.4.1	J2ME 中 XML .....	45
3.4.2	基于树的 XML 分析程序连接 MIDlet 到 XML 文件的例子 .....	46

第 4 章 MIDlet 中的 GUI 开发..... 49

- 4.1 MIDP 图形坐标系统 .....49
  - 4.1.1 MIDlet GUI 编程中图形类 .....50
  - 4.1.2 使用 Graphics 类.....50
- 4.2 部署不同的 MIDP GUI API (1) .....54
  - 4.2.1 问题描述 .....54
  - 4.2.2 识别应用的要求 .....55
- 4.3 MIDP GUI 编程中的 GUI 组件 (2) .....57
  - 4.3.1 Screen 类的组件 .....57
  - 4.3.2 Item 类的组件.....60
- 4.4 编写创建应用的代码 (3) .....62
  - 4.4.1 代码的编写 .....62
  - 4.4.2 执行程序 .....65
  - 4.4.3 验证输出 .....66

第 5 章 Servlets 与 MIDlet..... 67

- 5.1 用于 MIDlet-Servlets 内部通信的类和接口 .....67
  - 5.1.1 用于 MIDlet-Servlets 内部通信的类 .....68
  - 5.1.2 HttpURLConnection 接口.....68
- 5.2 实现 MIDlets 和 Servlets 之间的通信 .....70
  - 5.2.1 问题描述 .....70
  - 5.2.2 问题分析 .....70
  - 5.2.3 识别编码 MIDlet 要用到的包、类 和接口.....71
  - 5.2.4 编写处理数据库的 Servlet 的代码.....72
  - 5.2.5 编译部署 servlet .....74
  - 5.2.6 编写创建类将使 MIDlet 建立与 HTTP 的连接 .....74
  - 5.2.7 编写创建将连接到 servlet 的 MIDlet 的代码.....76

第 6 章 RMS..... 83

- 6.1 javax.microedition.rms 包.....83
  - 6.1.1 RMS 包中 RecordStore 类.....83
  - 6.1.2 RMS 包中接口.....84
  - 6.1.3 RMS 包中异常类.....85
- 6.2 部署 RMS API 编写 MIDLET 应用代码 .....85



6.2.1	问题描述 .....	85
6.2.2	创建输出的格式 .....	86
6.2.3	识别应用中使用的的方法 .....	86
6.2.4	编码实现 RMS 应用的程序 .....	90
6.2.5	执行程序 .....	96
6.2.6	验证输出 .....	96

## 第 7 章 MIDP 2.0 的 Game 类详细介绍 ..... 99

7.1	GameCanvas 类 .....	99
7.2	GameCanvas 基础实例 .....	100
7.3	Sprite 类 .....	103
7.3.1	Sprite 的定义 .....	103
7.3.2	Sprite 类构造函数 .....	103
7.3.3	8 bit、16 bit 还是 32 bit .....	104
7.3.4	精灵碰撞 .....	104
7.3.5	精灵显示 .....	105
7.3.6	显示精灵序列 .....	105
7.3.7	精灵透视 .....	105
7.3.8	精灵变换 .....	106
7.3.9	精灵优化 .....	106
7.3.10	Sprite 类实例 .....	107
7.3.11	扩充 Sprite 类 .....	110
7.3.12	建立自己的 Sprite 类 .....	111
7.3.13	Sprite 百草园 .....	111
7.4	LayerManager 类 .....	112
7.4.1	什么是 LayerManager .....	112
7.4.2	如何使用 LayerManager .....	112
7.4.3	setViewWindow (int x, int y, int width, int height) .....	113
7.4.4	LayerManager 类实例 .....	113
7.4.5	LayerManager 类和滚动背景 .....	118
7.5	TiledLayer 类 .....	121
7.5.1	什么是 TiledLayer 类 .....	121
7.5.2	TiledLayer 类的构造函数 .....	123
7.5.3	TiledLayer 处理 .....	123
7.5.4	TiledLayer 显示 .....	123
7.5.5	检索当前 TiledLayer 类的设置 .....	123
7.5.6	动态的单元格 .....	123

7.5.7 没有动态贴砖的 TiledLayer 类实例 .....	124
7.5.8 包含动态贴砖的 TiledLayer 类实例 .....	127

## 第 8 章 J2ME 游戏开发 ..... 131

8.1 与传统游戏的差别 .....	131
8.1.1 开发工具 .....	132
8.1.2 3D .....	132
8.1.3 多人游戏 .....	133
8.1.4 运行移动游戏的设备 .....	133
8.2 开始开发 .....	134
8.2.1 游戏的种类和流派 .....	135
8.2.2 学习游戏创意 .....	135
8.2.3 竞争 .....	135
8.3 推向市场 .....	136
8.3.1 直接出售 .....	136
8.3.2 加入集团(Aggregator) .....	136
8.3.3 游戏发行商 .....	136
8.3.4 运营商 .....	137
8.3.5 制造商 .....	137
8.3.6 无线产品代理商 .....	137
8.4 移动游戏的限制 .....	137
8.4.1 内存的分类 .....	138
8.4.2 存储空间的碎片 .....	138
8.4.3 显示大小和分辨率 .....	138
8.4.4 手机界面 .....	139
8.4.5 缺少类文件 .....	139
8.4.6 开发者和使用者眼中的性能 .....	139
8.4.7 先编码，然后考虑性能的提高 .....	140
8.4.8 减少面向对象的代码 .....	140
8.4.9 减少第三方库的使用 .....	140
8.4.10 最少的通信 .....	140
8.4.11 组合图像 .....	141
8.4.12 垃圾回收 .....	141
8.4.13 短小的名称及混淆机制 .....	141
8.4.14 编码技巧 .....	141
8.4.15 好的编码习惯 .....	142
8.5 编码之前 .....	142

8.5.1	游戏策划 .....	143
8.5.2	复用性和组件开发 .....	143
8.5.3	用户调查和目标用户 .....	144
8.5.4	游戏种类 .....	144
8.5.5	长期的计划 .....	144
8.5.6	技术 .....	144
8.5.7	游戏性 .....	145
8.5.8	真实性 .....	145
8.5.9	创意 .....	145
8.5.10	模仿 .....	145
8.5.11	设计模式 .....	145
8.6	开发过程控制 .....	146
8.6.1	开发模型 .....	147
8.6.2	编译过程 .....	149
8.6.3	源代码控制 .....	150

## 第 9 章 任天堂 Battle City 游戏的设计与实现..... 151

9.1	任天堂 (Nintendo) 的 8 位 FC 机器和 Battle City 背景介绍 .....	151
9.2	开发环境及相关技术的介绍 .....	152
9.2.1	开发环境 .....	152
9.2.2	关于 JBuilder 9.....	152
9.2.3	关于 Wireless Tool Kit.....	152
9.2.4	Java Application Manager .....	153
9.3	程序结构、思想和相关技术 .....	153
9.3.1	本程序需解决的有关技术问题 .....	153
9.3.2	程序流程 .....	154
9.4	绘图与 MIDP2.0 新增的 GameCanvas 包 .....	156
9.4.1	提供低级绘制的 Canvas 类 .....	156
9.4.2	Graphics 类.....	157
9.4.3	PNG 格式 .....	157
9.4.4	Game 包中的新功能.....	157
9.4.5	有关绘图的一些技术 .....	158
9.4.6	坦克的控制和敌方的智能运行 .....	158
9.4.7	子弹的运行和控制 .....	159
9.4.8	RMS 数据库系统.....	160
9.4.9	内存使用的最佳化 .....	161
9.4.10	混淆器 (Obfuscator) 的使用 .....	161

9.4.11	模拟器的相关调试 .....	162
9.5	程序分析和具体实现 .....	162
9.5.1	游戏进入前的选择 .....	162
9.5.2	主游戏逻辑及其涉及到的若干类 .....	166
9.5.3	坦克的共同行为 .....	176
9.5.4	玩家坦克的功能属性 .....	180
9.5.5	敌人坦克的功能属性 .....	183
9.5.6	子弹的运行和控制 .....	190
9.5.7	记分系统 .....	199
9.6	总结 .....	206

## 第 10 章 J2ME 项目应用手机游戏开发 ..... 207

10.1	手机游戏的特征 .....	207
10.2	游戏背景介绍及开发环境 .....	208
10.2.1	游戏背景介绍 .....	208
10.2.2	开发环境 .....	208
10.3	程序结构、思想和相关技术 .....	208
10.3.1	程序流程 .....	209
10.3.2	Canvas 类 .....	210
10.3.3	Graphics 类 .....	210
10.3.4	MIDP1.0 技术下的绘制背景技术 .....	210
10.3.5	MIDP2.0 新增的 GameCanvas 包 .....	211
10.3.6	PNG 图片格式 .....	212
10.3.7	玩家飞机的控制方式和敌人方的智能运行 .....	212
10.3.8	子弹的运行和控制 .....	213
10.3.9	内存的优化 .....	213
10.3.10	内存检测器 .....	213
10.3.11	混淆器 .....	213
10.4	程序分析和具体实现 .....	214
10.4.1	mybullets 类 .....	215
10.4.2	游戏逻辑及 gameScreen 类 .....	215
10.4.3	游戏中的奖励及相关飞机的行为 .....	220
10.4.4	普通敌人相关属性 .....	221
10.4.5	白云的实现原理 .....	222
10.4.6	BOSS 及相关属性 .....	223
10.5	测试 .....	223
10.5.1	发现的 BUG 及解决情况 .....	226

10.5.2 未完善的功能 .....	226
10.6 小结 .....	226
10.7 源代码 .....	226
10.8 操作说明 .....	246

## 第 11 章 综合实例开发: Eliminator ..... 247

11.1 游戏简述 .....	247
11.2 游戏启动画面 .....	247
11.3 游戏主菜单 .....	249
11.3.1 主菜单外观 .....	249
11.3.2 基本主菜单 .....	250
11.3.3 包含子菜单的主菜单 .....	254
11.3.4 菜单设计的建议和提高 .....	262
11.4 异常处理 .....	262
11.5 设置和高分记录 .....	263
11.5.1 RMS 数据结构 .....	263
11.5.2 设置 Settings 和高分记录 High Score 设计 .....	264
11.5.3 源代码 .....	264
11.6 地形 (滚动背景) .....	276
11.6.1 游戏场景 .....	276
11.6.2 一个简单 Terrain (地形) .....	278
11.6.3 多重地形的地图类 .....	284
11.7 玩家和子弹 .....	289
11.7.1 玩家 .....	289
11.7.2 子弹 .....	290
11.7.3 源代码 .....	290
11.7.4 游戏屏幕 .....	293
11.8 场景切换 .....	296
11.9 游戏内容 .....	298
11.9.1 敌人 .....	298
11.9.2 声音和振动 .....	300

# 第 1 章 引入 J2ME 平台

本章将从定义 J2ME 开始讲起，然后介绍它的总体架构并学习 J2ME 目标设备。作为架构讨论的一部分，将提供有关简表和配置的概述（后面的小节中将详细介绍简表和配置）。同时会简要介绍打包和配置 J2ME 应用程序过程中的一些注意事项。

Sun Microsystems 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，包括寻呼机、移动电话、可视电话、数字机顶盒和汽车导航系统。”自从 1999 年 6loper Conference 上声明之后，J2ME 为小型设备带来了 Java 语言的跨平台功能，允许移动无线设备共享应用程序。有了 J2ME，Sun 已经使 Java 平台能够适应集成了或基于小型计算设备并在 JavaOne Deve 的用户产品。

Sun Nucrosystems 已发布了几个 Java 版本，且它们维护着不同版本的体系结构的一致性。以系列方式发布的这些版本，由于每个版本都加入了一些新的功能，从而改进了它们的能力。但在 Java1.2 版发行之后，在 Java 的打包和许可证发放的方式上有些改变。这导致了 Java 平台被拆成三个版本。

即不同环境的三个 Java 平台是：

- (1) Java2 标准版 (J2SE)
- (2) Java2 企业版 (J2EE)
- (3) Java2 微型版 (J2ME)

J2SE 是工具和 API 的收集，主要用于创建 Java Applets 和单独的、基于 Web 的应用。J2EE 是对跨分布网络和远程访问的单个应用的企业计算 Java 的回答。J2EE 包含 J2SE 的所有功能，着重于用企业 Java Beans(EJB), Servlets 和 Java 服务器页面(JSP), 以及扩展置标语言(XML), 进行服务器端编程。

## 1.1 J2ME 和 WAP

开发无线应用的一种方法是 J2ME，它着重于无线计算环境，这是一个在不断推进的领域。无线计算的主要限制是对硬件设备资源的制约，诸如手机，个人数字助手 (PDA)，及其他的电子和嵌入设备。Java2 微型版 (J2ME) 是 Sun 公司的一种模块化、定制的平台，处理这些小到 128 KB RAM 的设备，且其处理器的能力低于台式计算机的计算要求。

J2ME 的多数特征是从 J2SE 派生出来的。J2ME 的某些特征如下：

- (1) 可移植性。
- (2) 跨不同产品的内在的一致性。
- (3) 跨网络的安全性。
- (4) 与 J2SE 平台和 J2EE 平台的兼容性。

开发无线应用的另一种方法是通过无线应用协议 (Wireless Application Protocol) (WAP)。WAP 是适应于由 Web 协议 (诸如 TCP/IP 和 UDP) 使用的几种数据处理技术的一种特例。



WAP 应用开发比较简单,因为它重用了现有的 Web 技术。WAP 不是为使用 HTML 而设计的,尽管许多 WAP 网关可把 HTML 转换为特定无线设备的数据。为生成动态的 WAP 内容,可使用 Servlets 和 Java Server Pages(JSP)或无线置标语言(Wireless Markup Language)(WML)。尽管 WAP 和 J2ME 是互补的技术,但它们以不同的方式进行操作。对于 WAP 的情形,移动设备运行一个访问 Intranet 或 Internet 应用的浏览器。移动设备不包含任何其他的应用。另一方面,J2ME 应用驻留在移动设备上,用户可得到这些应用,即使个别没有连接到网络上。

## 1.2 J2ME 的体系结构

J2ME 使用配置和简表定制 Java 运行时的环境(JRE)。作为一个完整的 JRE,J2ME 由配置和简表组成,配置决定了使用的 JVM,而简表通过添加特定于域类来定义应用程序。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。本书将在 J2ME 配置一章中详细讨论配置。

简表定义应用程序;特定地,它向 J2ME 配置中添加特定于域的类,定义设备的某种作用。在本章 1.6 小节中将深入介绍简表。下面的图 1-1 描述了不同的虚拟机、配置和简表之间的关系。它同时把 J2SE API 和它的 Java 虚拟机进行了比较。虽然 J2SE 虚拟机通常被称为一种 JVM,但是 J2ME 虚拟机、KVM 和 CVM 都是 JVM 的子集。KVM 和 CVM 均可被看作是一种 Java 虚拟机——它们是 J2SE JVM 的压缩版,并特定于 J2ME。

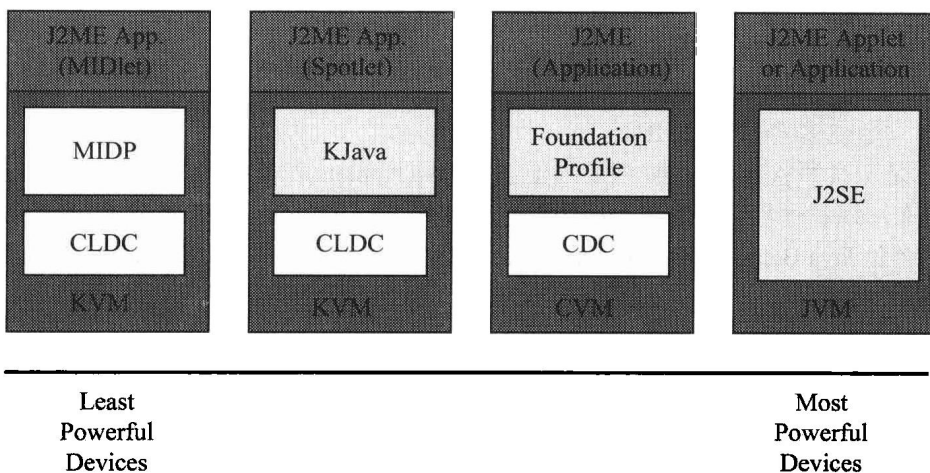


图 1-1 不同的虚拟机、配置和简表之间的关系

### 1.2.1 配置概述

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。虽然还可能在将来定义其他的配置,但当前 J2ME 存在两种配置。

#### (1) 连接限制设备配置 (CLDC)

特别与 KVM 一起用于内存有限的 16 位或 32 位设备。这是用于开发小型 J2ME 应用程序的配置(虚拟机)。(从开发的角度来看)它的大小限制让它比 CDC 更有趣、更具挑战性。CLDC 同时还是用于开发绘图工具应用程序的配置。Palm 电脑便是一个运行小应用程序的小

型无线设备的示例。本书将在本章 1.5 小节中深入介绍 CLDC。

### (2) 连接设备配置 (CDC)

与 C 虚拟机 (CVM) 一起使用, 用于要求内存超过 2 MB 的 32 位体系结构。因特网电视机顶盒便是这类设备的一个示例。稍后将在本章 1.5.3 小节中简要介绍 CDC。

## 1.2.2 简表概述

---

简表定义了用户的应用程序所支持的设备类型, 它向 J2ME 配置添加了特定于域的一类来定义设备的某种作用。简表建立在配置的顶部。已经为 J2ME 定义了两种简表: KJava 和移动信息设备简表 (MIDP), 它们也被建立在 CDLC 上。这两种简表适用于小型设备。有一种纲要简表, 用户可以在它的上面创建自己的简表, 这种纲要简表也称为基础表, 可供 CDC 使用。我们将在后面的章节中讨论上述这些简表, 还会使用 KJava 和 MIDP 建立一些示例应用程序。

## 1.2.3 J2ME 的目标设备

---

使用 CLDC 开发的 J2ME 应用程序的目标设备通常具有以下特征:

- (1) 可供 Java 平台使用的 160 KB 到 512 KB 的总内存。
- (2) 功率有限, 常常是电池供电。
- (3) 网络连通性, 常常是无线的、不一致的连接并且带宽有限。
- (4) 用户接口混乱, 程度参差不齐; 有时根本就没有接口。

一些 CLDC 支持的设备, 包括无线电话、寻呼机、主流个人数字助手 (PDA), 以及小型零售支付终端。

依照 Sun Microsystems, CDC 的目标设备通常具有以下特征:

- (1) 使用 32 位处理器。
- (2) 2 MB 或更多可供 Java 平台使用的总内存。
- (3) 设备要求的 Java 2 “蓝皮书”虚拟机的全部功能。
- (4) 网络连通性, 常常是无线的、不一致的连接并且带宽有限。
- (5) 用户接口混乱, 程度参差不齐; 有时根本就没有接口。

一些 CDC 支持的设备, 包括常驻网关、智能电话和通讯器、PDA、管理器、家用电器、销售网点终端以及汽车导航系统。

## 1.3 开发 J2ME 程序注意事项介绍

在本节中, 首先介绍在为小型设备开发应用程序时需要牢记的一些注意事项, 然后介绍在使用 J2SE 编译 J2ME 应用程序时调用编译器的方法, 最后将探究打包和部署, 以及在这个过程中提前验证所扮演的角色。

### 1.3.1 设计开发小型设备应用程序要注意的事项

---

为小型设备开发应用程序, 需要用户在设计阶段制定某种策略。最好是在开始编写代码之前, 战略性地为小型设备设计应用程序。由于无法考虑到所有的 “gotchas”, 在开发应用





程序之前更正代码是一件很痛苦的工作。下面是一些可以考虑的设计策略：

(1) 保持程序简单。除去不必要的功能，如果可能的话，将它们做成独立的、次要的应用程序。

(2) 程序越小越好。这一点对所有的开发者来说应该是显而易见的。越小的程序占用的设备内存越少，并且花费的安装时间越少。可考虑将用户的 Java 应用程序打包，作为压缩的 Java 档案 (jar) 文件。

(3) 运行时占用最少的内存。为尽可能减少运行时占用的内存，使用标量类型代替对象类型。同时，不依赖垃圾收集程序。用户应该在使用完对象时将对象引用置空，这样可有效管理内存。另外一种减少运行时所需内存的方法是使用“惰性”实例，它仅在必需时才分配对象。其他一些减少小型设备上过量和峰值内存使用的方法有快速释放资源、重新使用对象以及避免异常。

### 1.3.2 设计开发移动设备应用程序的注意事项

开发移动设备应用程序的规则与前面提及的开发小型设备的相同：先设计再编码。让用户检查一下开发移动设备应用程序时可考虑的一些设计建议：

(1) 让服务器做大部分的工作。将计算性较强的任务放到服务器上，让服务器为用户做这些工作。让移动设备处理界面和最少的计算工作，而让服务器做繁重的工作。当然，用户为其开发应用程序的移动设备对设备连接到服务器上的难易程度和频率有重要影响。

(2) 谨慎地选择编程语言。J2ME 仍然处于成长期，可能还不是最好的选择。根据用户的需要，选择其他的面向对象语言，如 C++，可能会更好。

### 1.3.3 性能注意事项

为性能而编码。下面介绍一些以获得最优性能为目标的编码方法：

(1) 使用局部变量。访问局部变量比访问类成员更快。

(2) 避免字符串串联。字符串串联不仅会降低性能，而且会增加应用程序的内存峰值占用量。

(3) 使用线程，避免同步。任何运行时间超过 1/10 s 的操作都需要一个独立的线程。避免同步同样能提高性能。

(4) 使用模型视图控制器 (MVC) 分离模型。MVC 将代码中控制显示的逻辑分离出来。

### 1.3.4 编译注意事项

同其他的 Java 应用程序一样，用户在打包和部署应用程序之前要先进行编译。尽管有了 J2ME，用户仍然使用 J2SE 编译器并且需要用适当的选项来进行调用。用户需要使用 `-bootclasspath` 选项来指示编译器使用 J2ME 类，而不是 J2SE 类。不要在编译器的 `CLASSPATH` 中设置配置类。这样将导致运行时错误，因为不管 `CLASSPATH` 中有什么，编译器将首先自动搜索 J2SE 的核心类。换句话说，编译器将无法引用特定 J2ME 配置中缺少的类或方法，结果导致在尝试运行应用程序时出现运行错误，如图 1-2 所示。