中国科学出版集团
新世纪书局

实用经验全面分享 手机开发者的必备手册

手机开发大潮已至，Java程序员，你能赶上这一轮热点吗？

专家指导

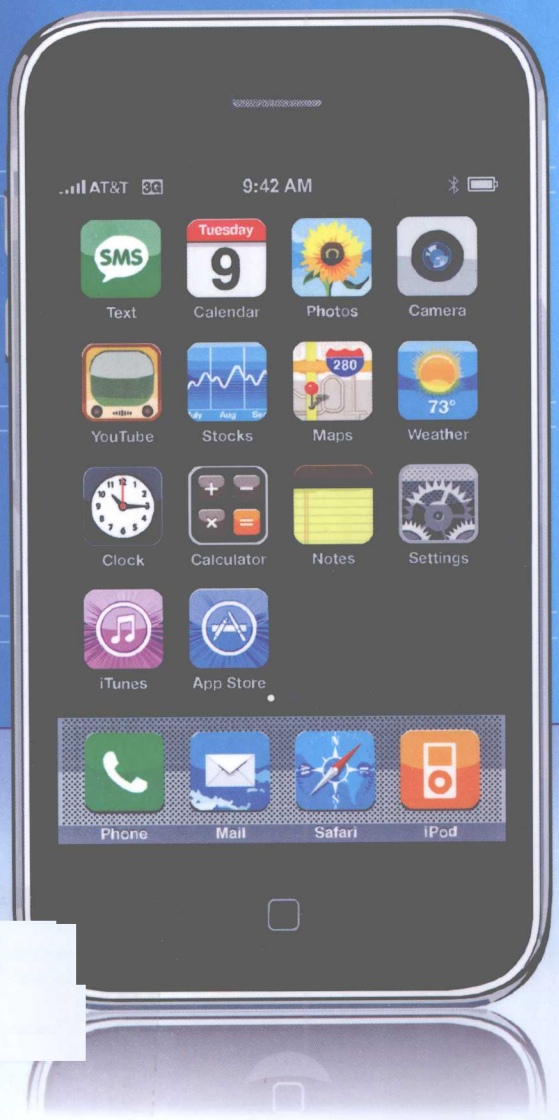
东软集团资深嵌入式开发工程师总结多年开发经验，揭密手机应用开发技术，毫无保留，倾力巨献

内容全面

汇集多种Java ME开发技术，包括UI组件开发、记录管理系统、移动网络应用、个人信息管理、无线消息服务、游戏和多媒体开发、单元测试等

范例教学

精心挑选最具代表性的61个典型范例，囊括大量开发技巧，其中的关键技术是开发者梦寐以求的解决方案



超值光盘

1DVD 大型多媒体教学课程

长达300分钟的多媒体语音教学视频，重点难点全面解析

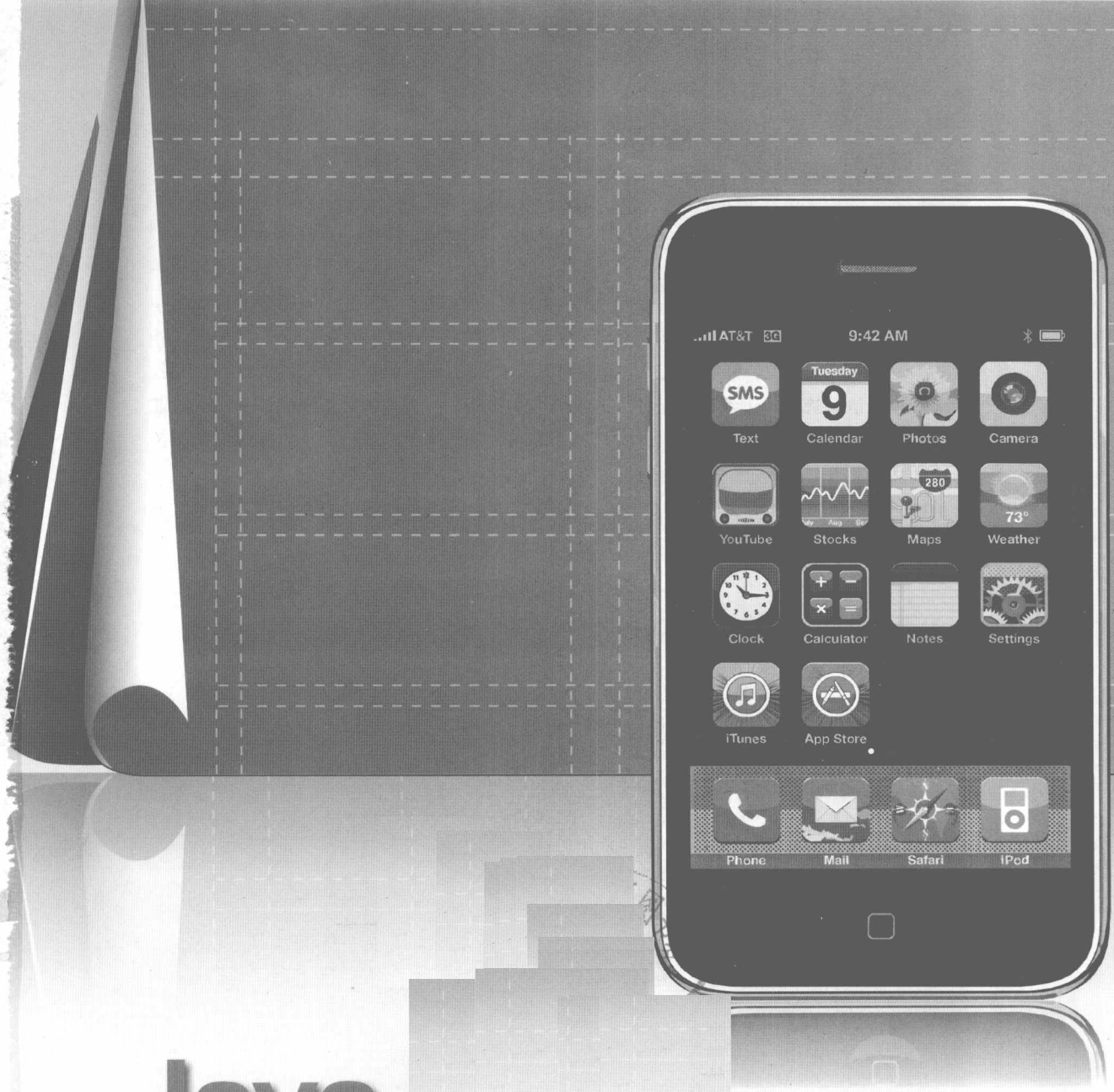
61个范例源代码，完全可以套用到实际开发中

Java ME 手机应用开发大全

（零起点范例教学）

黄正环 编著

科学出版社



Java 手机应用开发大全

（零起点范例教学）

黄正环 编著

科学出版社

内 容 简 介

随着移动互联网技术的飞速发展，以手机为核心的新兴内容承载形式正越来越多地融入人们日常生活的方方面面，这使得手机应用开发获得了广阔的发展空间。而在众多手机应用开发技术中，Java ME 由于获得了几乎所有手机厂商的支持，已成为目前最流行的手机应用编程语言之一。

基于这种形势，本书由浅入深地讲解了基于 Java ME 的手机应用相关开发技术。全书共 19 章，从 Java ME 的基础知识、开发环境配置和用户界面设计，到记录管理系统、移动网络应用、文件管理、个人信息管理、无线消息服务、游戏和多媒体开发技术，内容几乎涵盖了 Java ME 手机应用开发的各个方面。同时，本书针对性地引入了大量范例代码以帮助读者深入理解相关概念和技术。全书最后提供了一个完整的商务应用案例，供读者学习研究。

本书配有长达 300 分钟的多媒体语音教学视频，适合对 Java ME 技术感兴趣的初学者，书中大量范例对已有手机开发经验的程序员也有很大的参考价值。

图书在版编目（CIP）数据

Java ME 手机应用开发大全：零起点范例教学/黄正环编著. —北京：科学出版社，2010.6

ISBN 978-7-03-027689-6

I. ①J… II. ①黄… III. ①JAVA 语言—程序设计 ②移动通信—携带电话机—应用程序—程序设计

IV. ①TP312 ②TN929.53

中国版本图书馆 CIP 数据核字（2010）第 094084 号

责任编辑：赵东升 何 武 / 责任校对：赵丽平

责任印刷：新世纪书局 / 封面设计：周智博

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

中国科学出版集团新世纪书局策划

北京市鑫山源印刷有限公司

中国科学出版集团新世纪书局发行 各地新华书店经销

*

2010 年 7 月 第 一 版 开本：16 开

2010 年 7 月第一次印刷 印张：27

印数：1—3 000 字数：657 000

定价：49.00 元（含 1DVD 价格）

（如有印装质量问题，我社负责调换）

前 言

第一次写书，刚刚开始动笔的时候感觉很茫然。本来，作为一名经历了湖北“黑色七月”（那个时候的高考安排在 7 月）的人，写点东西并不是太难，因为老师告诉我们，任何文章都是有模板可以套用的。举个简单例子，如果只是想得个平均分的话，英语作文的开头都可以是“With the development of science and society, more and more people find that…”。不过，就在我开始构思这本书的时候，遇到了下面这样一件事。

前些日子，我给朋友介绍了一本当下很流行的历史书。

朋友：这本历史书有什么特别么？

我：语言很活泼，读起来像小说。

朋友：那就是戏说咯？

我：不是。它是用小说的语言写的正史，所以读起来很好玩。

朋友：现在的人啊，读点有用的东西，还要选写得生动有趣的，真难伺候。

我：… …

诚然，这应该也是社会的一种进步吧。人总是追求更好品质的享受——即便是在学习的过程中。

现在市面上也有一些关于 Java ME 的书，模式差不多。我本来也打算照着他们的模板来写，不过，现在有了一些新的想法——那就是写一本比较好看的 Java ME 教程。

本书会用一种较轻松的语气向读者介绍 Java ME 的相关技术，一些重要的概念在书中会用粗体字标注出来。全书分为 3 篇——基础入门、高手进阶和实践指南。基础入门篇主要介绍 Java ME 的基本概念和语法、开发环境的配置和用户界面设计；高手进阶篇介绍了 Java ME 的记录管理系统、异常处理、多线程应用、移动网络应用、文件管理、个人信息管理、无线消息服务、Push 技术、XML 应用、多媒体和游戏开发；实践指南篇则分析了常见的设计模式、编程思想和规范，并讲解了单元测试的相关技术，最后一章是一个完整的商务应用案例。本书配有大量的代码和范例，供读者参考学习。结合本人的学习经验，建议大家在学习的过程中实际动手运行这些代码，相信一定会取得更好的效果。

最后，祝愿读者朋友们工作顺利，万事如意！

编 者

2010 年 5 月

目 录

第 1 篇 基础入门

第 1 章 初识 Java ME 1

1.1 Write one, Run anywhere 1

1.2 Java ME 平台体系结构 2

1.3 MIDlet 的生命周期 3

1.4 MIDlet 套件 5

1.5 移动开发中的常用规范 7

第 2 章 开发环境配置 9

2.1 JDK 的安装和配置 9

2.2 Eclipse 的安装和配置 14

2.3 EclipseME 的安装 16

2.4 SDK 的安装 20

2.5 创建 Java ME 工程 24

2.6 运行和打包 25

第 3 章 用户界面 29

3.1 LCDUI 的架构 29

3.2 Display 类 30

实例 3-1 测试设备的属性 32

3.3 Displayable 类 35

3.4 Command 类 36

实例 3-2 菜单命令的类型对菜单命令的位置和顺序的影响 37

实例 3-3 Command 第二个构造方法的使用及菜单命令监听 40

第 4 章 高级 UI 组件开发 42

4.1 列表类 List 42

4.1.1 List 的 3 种类型 42
实例 4-1 List 类提供的各种对选项的维护操作 44
4.1.2 List 中的选中事件 47
实例 4-2 在隐式类型列表中处理选中事件 48
实例 4-3 在多选类型列表中处理选中事件 49
4.2 文本输入类 TextBox 52
实例 4-4 TextBox 的文本编辑功能 54
4.3 表单类 Form 57
4.3.1 Form 中的 Item 布局 58
实例 4-5 Form 中的 Item 布局规则 59
4.3.2 ChoiceGroup 63
实例 4-6 3 种类型 ChoiceGroup 的用法 64
4.3.3 DateField 66
实例 4-7 DATE_TIME 类型的 DateField 67
4.3.4 Gauge 69
实例 4-8 Gauge 的使用方法 70
4.3.5 CustomItem 74
实例 4-9 CustomItem 的使用方法 75
4.3.6 其他的 Item 79
4.3.7 Item 的菜单命令 80
实例 4-10 Item 中菜单命令的使用方法 80
4.4 提示类 Alert 82
实例 4-11 常见 Alert 的使用方法 83
4.5 文字滚动类 Ticker 87
实例 4-12 Ticker 的使用方法 88
第 5 章 低级 UI 组件开发 90
5.1 Canvas 概述 90
5.2 绘图类 Graphics 91
5.2.1 绘制文本 92
实例 5-1 Canvas 中绘制文本的方法 93
5.2.2 绘制简单图形 95
实例 5-2 Canvas 中绘制各种图形的方法和绘制效果 98

5.2.3 绘制图片 …… 101
5.3 绘图的相关类 …… 102
5.3.1 图片类 Image …… 102
5.3.2 字体类 Font …… 104
5.4 按键响应 …… 105
实例 5-3 Canvas 中对按键事件的处理 …… 107

第 2 篇 高手进阶

第 6 章 记录管理系统（RMS） …… 110
6.1 RMS 的基本概念 …… 110
6.2 RecordStore 的打开、关闭和删除 …… 111
6.3 记录的管理 …… 113
6.3.1 添加和删除记录 …… 114
6.3.2 获取和修改记录 …… 115
6.4 记录的存储 …… 116
6.4.1 系统数据类型的数据存储 …… 117
实例 6-1 图片的存储和读取 …… 118
6.4.2 自定义数据类型的数据存储 …… 124
实例 6-2 自定义数据类型的存储和读取 …… 126
6.5 遍历记录 …… 134
6.6 记录管理的高级应用 …… 136
6.6.1 记录过滤 …… 136
实例 6-3 过滤器的实现方法 …… 136
6.6.2 记录排序 …… 140
实例 6-4 排序策略的实现方法 …… 140
6.6.3 记录监听 …… 143
实例 6-5 记录监听的实现方法 …… 144
第 7 章 异常处理 …… 152
7.1 MIDP 定义的异常类 …… 152
7.2 Exception 类 …… 154
7.3 异常处理 …… 155

7.3.1 捕获异常 156
7.3.2 抛出异常 157
7.3.3 finally 关键字 160
7.4 异常处理的一些基本原则 162
第 8 章 多线程应用 163
8.1 多线程概述 163
8.2 多线程的实现 164
8.2.1 通过继承 Thread 类 164
8.2.2 通过实现 Runnalbe 接口 165
8.2.3 使用任务组合 166
实例 8-1 使用任务组合实现多线程 167
8.3 线程状态控制 171
8.4 同步和死锁 173
实例 8-2 使用 synchronized 关键字来实现线程同步 174
第 9 章 移动网络应用 179
9.1 通用连接框架概述 179
9.2 搭建服务器 182
9.3 HTTP 连接 186
实例 9-1 使用 GET 和 POST 方法进行 HTTP 通信 188
实例 9-2 处理客户端请求的 Servlet 193
9.4 Socket 连接 196
实例 9-3 Socket 服务器端程序 197
实例 9-4 Socket 客户端程序 202
9.5 Datagram 连接 204
实例 9-5 Datagram 客户端程序 205
实例 9-6 Datagram 服务器端程序 209
第 10 章 文件管理 211
10.1 文件/文件夹的访问 211
实例 10-1 访问手机上的文件系统 213
10.2 文件/文件夹的创建和删除 219
实例 10-2 创建和删除一个文件或者文件夹 220

10.3 文件的读写 225
实例 10-3 文件读写的实现 225
10.4 文件的属性 229
实例 10-4 获取文件的属性 230
第 11 章 个人信息管理 234
11.1 PIM 包概述 234
11.1.1 访问 PIMList 235
11.1.2 访问支持的字段 237
实例 11-1 获取联系人信息列表中支持的字段信息 238
11.2 管理联系人 242
11.2.1 访问联系人列表 242
实例 11-2 访问手机和 SIM 卡上的联系人列表 243
11.2.2 添加联系人 248
实例 11-3 向手机上的联系人列表添加一个联系人 249
11.2.3 删除联系人 259
11.2.4 修改联系人 261
11.3 管理日程安排 265
实例 11-4 管理日程安排 265
11.4 管理待办事项 268
实例 11-5 管理待办事项 268
第 12 章 无线消息服务 271
12.1 无线消息 API 272
12.2 发送和接收文本消息 273
12.2.1 发送文本消息 274
实例 12-1 发送文本消息 276
12.2.2 接收文本消息 280
实例 12-2 接收文本消息 280
12.3 发送和接收二进制消息 283
实例 12-3 发送一个二进制消息 283
实例 12-4 接收一个二进制消息 285
12.4 发送和接收多媒体消息 285

12.4.1 发送多媒体消息 286
实例 12-5 发送多媒体消息 288
12.4.2 接收多媒体消息 292
实例 12-6 接收多媒体消息 293
12.5 小区广播服务 296
实例 12-7 接收小区广播消息 297
第 13 章 Push 技术 301
13.1 Push 机制 301
13.2 静态注册方式 303
实例 13-1 监听并接收来自服务器的 Push 信息 304
13.3 动态注册方式 307
实例 13-2 基于 Alarm 时间的动态注册 307
13.4 Push 开发中应注意的一些问题 309
第 14 章 XML 应用 311
14.1 XML 的概念 311
14.2 常用的 XML 解析方法 314
14.3 KXML2 解析器 315
实例 14-1 使用 KXML2 解析器来解析一个 XML 文档 316
14.4 JSR172 提供的 SAX 解析器 321
实例 14-2 使用 SAX 解析器来解析一个 XML 文档 322
第 15 章 多媒体开发 328
15.1 MMAPI 的基本框架 328
15.1.1 管理器 329
15.1.2 播放器 331
15.1.3 数据源 332
15.1.4 控制器 333
15.2 音频播放 333
实例 15-1 播放一个音调 334
实例 15-2 一个简单的音乐播放器 336
15.3 视频播放 341

实例 15-3 播放一段视频 ····· 341
15.4 相机控制 ····· 344
第 16 章 游戏开发 ····· 347
16.1 游戏设计基本概念 ····· 347
16.2 GameCanvas ····· 350
16.2.1 离屏图像缓冲 ····· 350
16.2.2 按键状态查询 ····· 351
实例 16-1 GameCanvas 中离屏图像缓冲的用法以及对按键状态的监听方法 ····· 352
16.3 Layer ····· 354
16.3.1 Sprite ····· 355
实例 16-2 Sprite 中动画效果的实现 ····· 358
16.3.2 TiledLayer ····· 361
实例 16-3 实现一个动态的背景效果 ····· 363
16.3.3 碰撞检测 ····· 366
16.4 LayerManager ····· 367
实例 16-4 LayerManager 的用法 ····· 368

第 3 篇 实践指南

第 17 章 设计模式 ····· 373
17.1 MVC 概述 ····· 373
17.2 MVC 的实现策略 ····· 374
17.3 一个基于 MVC 模式的笔记管理程序实现 ····· 376
实例 笔记管理程序 ····· 376
17.4 其他一些常见的设计方法 ····· 382
17.4.1 合理使用静态类 ····· 382
17.4.2 单态方法 ····· 383
17.4.3 视图栈 ····· 384
第 18 章 单元测试 ····· 386
18.1 单元测试概述 ····· 386
18.2 集成 J2MEUnit ····· 387
18.3 J2MEUnit 测试代码的框架 ····· 389

实例 Java ME 的单元测试 389
18.3.1 TestCase 389
18.3.2 TestSuite 392
18.3.3 TestRunner 393
第 19 章 一个商务应用实例 395
实例 制作“在线听歌”商务应用 395
19.1 需求分析 395
19.2 程序整体设计 396
19.3 功能的实现策略 397
19.3.1 音乐列表的实现 397
19.3.2 音乐播放的实现 402
19.3.3 网络通信的实现 404
19.3.4 XML 解析的实现 406
19.3.5 文件管理的实现 408
19.3.6 各个功能模块的控制 410
19.4 功能扩展 414
附录 A HTTP 请求的响应码 416
附录 B HTTP 的头部信息 418
B.1 HTTP 请求头参数 418
B.2 HTTP 响应头参数 418
附录 C PIM 中的标准字段 419
C.1 Contact 标准字段列表 419
C.2 Event 标准字段列表 420
C.3 ToDo 标准字段列表 420

第 1 章

初识 Java ME

1.1 Write one, Run anywhere

移动互联网的时代到来了！

越来越多的人怀着各种各样的目的喊出了同一个口号。据中国互联网络信息中心（CNNIC）近日发布的调查报告显示，我国通过手机上网的用户数已经超过 1.176 亿，其中一半都是在过去一年里突然出现的，而每天多次使用手机上网的用户占到 34%，手机互联网在产品内容和用户规模上正直追传统互联网。

正如 Internet 的普及给 PC 机的应用带来了“大爆炸”式的发展，互联网技术与移动通信的融合，也必然开拓出一个属于手机应用的广阔空间。可以想象将来的某一天，你用手机刷开了车门，从仓库旁边经过便得到了所有的仓储数据，然后来到一家咖啡厅，一边品尝甜点一边通过手机上网预订一份给孩子的小礼物。哦，老板来邮件检查工作进度了！不用担心，手机会把刚刚收集到的数据生成报告，然后提交过去。生活就是这么 easy！

不过手机操作系统的市场现状可不像 PC 操作系统那样统一，具体见图 1-1。

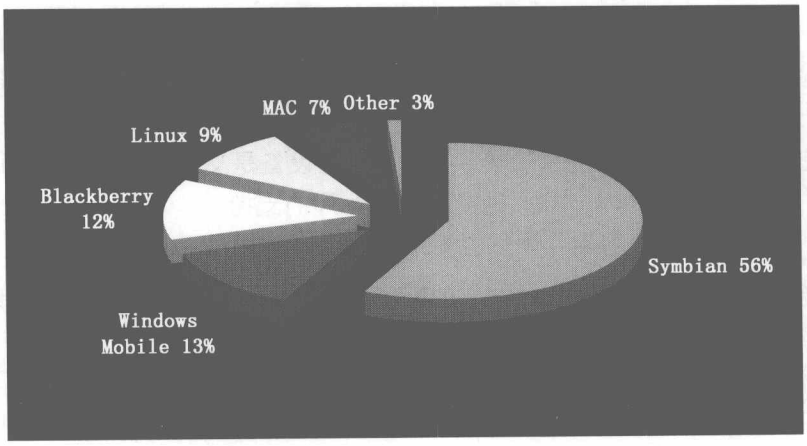


图 1-1　2008 年各种手机操作系统的市场份额

无论是老牌的 Symbian 和 Windows Mobile，还是后起之秀的 iPhone 和 Android，都有着不俗的用户数量。以后肯定还会有更多的厂商进入这个炙手可热的领域，比如中国移动定

制的移动操作平台 OMS 目前已开始了市场化运作。

面对如此众多的平台，如何能够用一种统一的开发方式来开发一种兼容各种操作系统的程序呢？Java 为我们提供了一个很好的解决方案。

这里还有一个比较戏剧性的小故事。Java 语言其实最早诞生于 1991 年，起初被称为 OAK 语言，是 SUN 公司为一些消费性电子产品而设计的一个通用环境。他们最初的目的只是为了开发一种独立于平台的软件技术，而且在网络出现之前，OAK 可以说是默默无闻，甚至差点夭折。互联网技术兴起以后，从 1994 年起，SUN 公司的工程师开始将 OAK 技术应用于 Web 上，并且开发出了 HotJava 的第一个版本。当 SUN 公司 1995 年正式将它以 Java 这个名字推出的时候，几乎所有的 Web 开发人员都心生感叹：噢，这正是我想要的！之后的事情完全可以用峰回路转来形容，Java 成了一颗耀眼的明星，丑小鸭一下变成了白天鹅。

Java 按照功能划分为 3 个版本，每个版本都有独立软件开发包（SDK）。Java 的原始版本称为 Java 2 标准版（Java SE），主要用于桌面应用软件开发；Java 2 企业版（Java EE）主要用于企业级应用的开发；而本书的主角 Java ME 是 Java 2 的微型版，被使用在各种各样的消费电子产品上，如手机、PDA 等。

SUN 公司设计 Java ME 时，将其定义为“Java Platform, Micro Edition (Java ME) provides a robust, flexible environment for applications running on mobile and other embedded devices”，并希望借助这把利剑对嵌入式设备这个混乱的领域进行统一。而现在，在这一思想下诞生的 Java ME 继续了他的祖先 OAK 的征程，并且已经在嵌入式领域得到了广泛的关注和应用。

好吧，从现在开始，我们的口号就是——“Write one, Run anywhere!”

1.2 Java ME 平台体系结构

Java ME 平台由两个部分构成：配置（Configuration）和简表（Profile）。配置包括 Java 虚拟机和核心类库等，它定义了设备制造商必须在该类别设备上实现的最小 Java 平台。简表基于配置之上，是一系列设备提供的开发包的集合。简表中还包含可选包（Optional Package），它是针对特定设备提供的类库，比如某些设备是支持蓝牙的，针对此功能，Java ME 中制定了 JSR82（Bluetooth API），提供了对蓝牙的支持。

目前，Java ME 的设备根据其硬件性能可以分为以下两种。

（1）**连接设备（Connected Devices）**：它的配置层为 CDC（Connected Devices Configuration），这种设备上使用的是传统的 JVM 虚拟机。

（2）**受限连接设备（Connected Limited Devices）**：它对应的配置层为 CLDC（Connected Limited Devices Configuration）。这种设备上，由于受到内存资源和执行速度的影响，不能再使用传统的 Java 虚拟机，而是另外一种被称为 KVM（Kilo Virtual Machine）的虚拟机。受限连接设备对应的简表是 MIDP（Mobile Information Device Profile），它提供了应用模型（MIDlet）、用户界面支持（LCDUI）、记录存储系统（RMS）、定时器等功能。

这个分类的标准是在 2001 年制定的，而现在的移动终端设备的 CPU 处理速度和内存大小都有很大的发展，所以按照这个标准已经不能评判一个设备是属于连接设备还是受限连接设备了。比如，该标准中规定 CLDC 设备的内存在 512KB 以下，事实上 2007 年上市的 N95 的内存已经达到了 160MB。

Java ME 的构成及其在整个 Java 体系中的位置如图 1-2 所示。

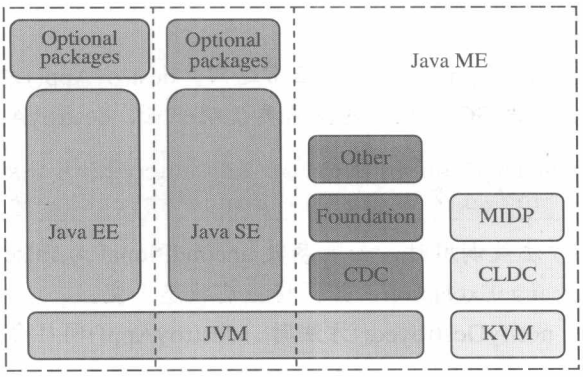


图 1-2　Java ME 的构成以及它与 Java EE，Java SE 的关系

手机属于受限连接设备，因此手机上的 Java ME 程序是基于 CLDC 和 MIDP 的。这些程序也称为 MIDlet（Mobile Information Devices let）。

1.3 MIDlet 的生命周期

MIDlet 的应用程序模型定义在 MIDP 中，它是受 AMS（Application Management Software）管理的。AMS 负责 MIDlet 的安装、下载、运行和删除等操作。在开发一个 MIDlet 程序之前必须先弄清楚其生命周期，这样才能理解 MIDlet 的各种行为和状态。

MIDlet 定义有 3 种状态：活动状态、暂停状态和死亡状态。状态之间的转换除了受到 AMS 的控制之外，MIDlet 也可以通过和 AMS 通信，通知 AMS 自己状态的变化，这通常是通过方法 notifyDestroyed()和 notifyPaused()实现的。

（1）**暂停状态（Paused）**：此状态下，MIDlet 已经被初始化，并且暂停运行。此时，它不再控制或者使用任何共享资源。以下情况会进入此状态。

- MIDlet 通过 new()方法被创建。MIDlet 的 public 类型的无参数构造方法会被调用，并且正常执行完毕，没有抛出任何异常。这个过程非常短暂，一般并不处理太多初始化的逻辑。在这一过程中，如果发生任何异常，应用程序会立刻进入死亡状态。
- 当 MIDlet 处于运行状态的时候，AMS 调用 pauseApp()方法，并且该方法被正常执行，没有抛出任何异常。
- 当 MIDlet 处于运行状态的时候，应用程序自己调用 notifyPaused()方法，并且该方法被正常执行，没有抛出任何异常。

- 当 MIDlet 处于运行状态的时候，执行 startApp()方法时抛出了 MIDletStateChange-Exception。

（2）**活动状态（Active）**：此状态下，MIDlet 正在正常运行。当 AMS 调用 startApp()方法时会进入此状态。

（3）**死亡状态（Destroyed）**：此状态下，MIDlet 已经释放了全部的资源，并停止运行。以下情况会进入此状态。

- 当 AMS 调用 destroyApp()方法，并正常返回。destroyApp()方法应该释放所有的资源，以便这些资源被 GC 回收。这里需要说明一点，destroyApp()方法的定义如下：

```
protected abstract void destroyApp(boolean unconditional)
                  throws MIDletStateChangeException
```

- 当 destroyApp()方法被调用时，如果参数 unconditional 为 false，MIDlet 可以抛出一个 MIDletStateChangeException，并保持当前状态不变。
- 当应用程序调用 notifyDestroyed()方法时，destroyApp()同样会被首先执行。

注 意

MIDlet 的死亡状态只能进入一次。

图 1–3 展示了 MIDlet 各个状态之间的转换情况。

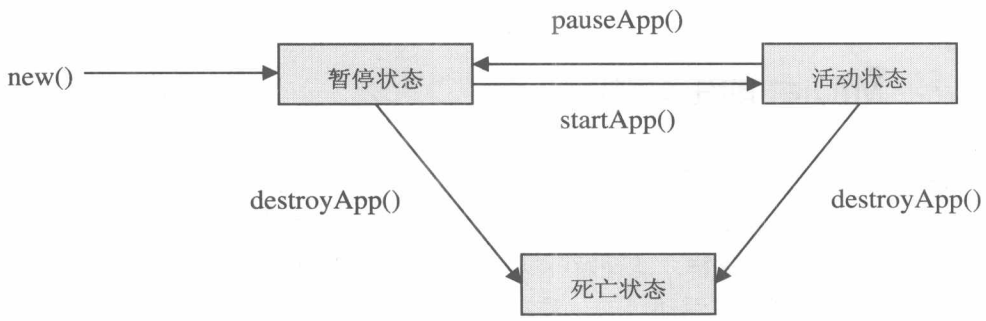


图 1-3　MIDlet 各个状态之间的转换

在开发 MIDlet 程序的时候，必须从 javax.microedition.midlet.MIDlet 类继承并实现该类的 3 个方法。下面的代码是一个 MIDlet 程序的结构。

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;


public class MyFirstMIDlet extends MIDlet {

    public MyFirstMIDlet() {
        // TODO Auto-generated constructor stub
    }
```

```
    protected void destroyApp(boolean arg0) throws
MIDletStateChangeException {
        // TODO Auto-generated method stub

    }

    protected void pauseApp() {
        // TODO Auto-generated method stub

    }

    protected void startApp() throws MIDletStateChangeException {
        // TODO Auto-generated method stub

    }

}
```

1.4 MIDlet 套件

MIDlet 套件（MIDlet Suite）是一个 MIDlet 应用程序所有的文件以及作为 MIDlet 一部分的必需资源的集合，它是发布一个 Java ME 应用程序的常见形式。一个 MIDlet 套件包括以下几个部分：

- 封装成 JAR 包的 Java 类文件
- 封装在 JAR 包内的资源文件（如图片等）
- 封装在 JAR 包中描述其文件内容的清单文件 MANIFEST.MF
- Java 应用程序描述符文件 JAD

下面将分别介绍这些组成部分。

一个 Java ME 应用程序通常由多个文件组成，这些文件包括 Java 类文件以及其他一些资源文件。我们把这些文件封装成为一个整体，也就是 JAR 包。

除了类和资源文件，一个 JAR 包文件还包括描述 JAR 内容的清单文件。清单文件名为 MANIFEST.MF，存储在 JAR 包的 META-INF 目录下。下面是一个 MANIFEST.MF 文件的样例：

```
Manifest-Version: 1.0
MIDlet-Vendor: MIDlet Suite Vendor
MIDlet-Version: 1.0.0
MIDlet-1: startMIDlet,,com.nokia.test.startMIDlet
MicroEdition-Configuration: CLDC-1.1
MIDlet-Name: StoreImageExample MIDlet Suite
MicroEdition-Profile: MIDP-2.1
```