

J2ME 程序 移动软件 设计



卢军 岳希 周辉 编著
荣智涛 主审

取材广泛，内容丰富

涵盖J2ME基础知识、程序设计方法、高级编程技巧和应用实例分析四大方面。

典型示例，边学边用

每篇均配有典型实用的程序示例，略加修改即可在实际生活和工作中使用。

循序渐进，逐步推进

本书结构完整、重点突出，每章均配有大量习题，帮助读者迅速掌握J2ME的精髓。



中国水利水电出版社
www.waterpub.com.cn

21 世纪软件工程师规范化教材

J2ME 移动软件程序设计

卢 军 岳 希 周 辉 编 著

荣智涛 主 审



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是一本完整的介绍 J2ME 编程技术的教程,内容包括 J2ME 基础知识、程序设计方法、高级编程技巧和应用实例分析。全书详细地介绍了 J2ME 程序设计原理和方法,并通过大量有代表性的编程实例,向读者演示 J2ME 在开发移动应用软件方面的强大功能。

本书包含了近百个 J2ME 的编程实例。这些例子从最简单的 J2ME 程序开始,到完整的网络应用程序、游戏程序,覆盖了 J2ME 程序设计的方方面面。本书特别针对 J2ME 在移动信息处理、短消息处理、移动多媒体应用等方面进行了详细的介绍,适应了当前 J2ME 程序开发的发展趋势。这些例子中大多都具有很强的实用价值,略加修改就可以在实际生活和工作中使用。通过对实例的学习和分析,可以帮助读者更深刻地理解 J2ME 程序设计的方法和技巧。如果读者是一名 Java 程序员,或者读者对智能手机、PDA、3G 应用程序设计或者无线互联网应用有兴趣,那么本书对读者而言将是一本极好的参考书籍。

阅读本书并不需要已经熟练地掌握 Java 语言。从本书的阅读和学习中,读者可以同时学习到 Java 语言的程序设计知识。如果读者对 Java 语言已经比较熟悉,那么本书是读者进入 J2ME 程序设计的最佳伴侣。

本书电子教案及程序源代码可到中国水利水电出版社网站和万水书苑下载,网址为:
<http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

J2ME移动软件程序设计 / 卢军, 岳希, 周辉编著

— 北京: 中国水利水电出版社, 2009. 11

21世纪软件工程师规划教材

ISBN 978-7-5084-6931-7

I. ①J… II. ①卢… ②岳… ③周… III. ①
JAVA语言—程序设计—教材②移动通信—通信设备—应用程序—程序设计—教材 IV. ①TP312②TN929.5

中国版本图书馆CIP数据核字(2009)第195591号

策划编辑: 杨庆川 责任编辑: 张玉玲 封面设计: 李 佳

书 名	21 世纪软件工程师规划教材 J2ME 移动软件程序设计
作 者	卢 军 岳 希 周 辉 编 著 荣智涛 主 审
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 22.25 印张 572 千字
版 次	2010 年 1 月第 1 版 2010 年 1 月第 1 次印刷
印 数	0001—4000 册
定 价	35.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

随着 J2ME 的不断发展, J2ME 已经成为移动软件开发的主要平台, 广泛地应用在智能卡、机顶盒、智能手机、PDA、移动终端、嵌入式设备上。J2ME 目前已经得到了全球各大计算机和消费电子产品生产厂商的积极响应, 例如诺基亚、摩托罗拉、索尼爱立信、三星、TCL 等公司都推出了自己的支持 J2ME 技术的手机和其他电子产品。

随着 3G 时代的到来, 越来越多的电话将成为智能终端设备, 在其上可以运行各种各样的应用程序。J2ME 因为其优秀的跨平台特性、强大的功能, 使第三方软件开发商可以很容易地开发移动应用程序, 并应用在不同的设备上, 缩短了开发周期, 提高了产品的竞争力。

J2ME 发展至今, 已经全面提供了包括基本应用、游戏开发、网络应用、多媒体应用、金融应用等各个方面的能力, 可以实现几乎所有移动软件开发功能, 成为一个广泛使用的强大开发平台。相信随着 J2ME 技术的不断发展, 一定会为移动应用软件的开发带来全新的革命。

主要内容

第 1 章 J2ME 简介。详细介绍什么是 J2ME、J2ME 的体系结构、在 J2ME 中 CLDC 和 MIDP 的概念以及 J2ME 的发展前景和未来。

第 2 章 J2ME 开发环境。着重介绍 JDK 的详细安装过程、J2ME Wireless Toolkit 的安装、配置和使用方法, 以及编写、编译、校验、发布一个 J2ME 程序的完整过程, 重点介绍了目前使用广泛的 Eclipse+EclipseME 开发环境。为了帮助读者加强对 J2ME 程序结构的理解, 还对最简单的 J2ME 程序的结构和各部分功能进行详细剖析。

第 3 章 CLDC 和 MIDP API 简介。CLDC 和 MIDP 是 J2ME 的核心和基础, 本章简明扼要地介绍 CLDC 和 MIDP API, 为读者提供 J2ME 程序开发平台的一个完整概貌。

第 4 章 J2ME 的用户界面编程。J2ME 可以实现丰富多彩、功能强大的用户界面。本章内容涉及 Display 类和 Displayable 类的基本用法, 文本框 (TextBox)、报警 (Alert)、列表框 (List) 和窗口 (Form) 等屏幕部件的使用方法和示例等。

第 5 章 J2ME 数据库编程。J2ME 内置了一个微型数据库系统, 本章详细介绍如何使用 J2ME 的 RMS 系统来进行数据库编程, 内容包括 RMS 的创建、打开、关闭和删除方法; RecordListener 接口的使用方法; 多列表格的使用技术; RecordEnumeration 记录遍历方法; RecordComparator 记录排序方法; RecordFilter 记录查找方法, 最后介绍了编写高效的数据库程序的基本原则。

第 6 章 J2ME 的网络编程。本章是 J2ME 的非常重要的内容, 详细介绍 J2ME 中支持网络通信的类和接口, 以及 file、socket、comm、datagram、http 等通信协议的使用方法, 重点介绍最重要的通信方法——HTTP 通信, 介绍了网络编程中的数据加密问题。最后给出了大量实用网络编程的例子, 如获得股票和位置信息。这些实例都是非常实用的, 许多例子略加修改就可以应用在实际生活中。

第 7 章 J2ME 应用程序的性能优化。J2ME 是面向小型化移动设备的。本章主要集中讨论如何优化一个 J2ME 程序的性能, 对 J2ME 中的字符串编程、用户界面编程、MIDlet 打包和加

密等工作进行了详细讨论，并对如何提高性能给出了切实可行的方法。

第 8 章 kAWT 编程。kAWT 是为 J2ME 设计的可以在手机等移动信息设备上运行的 AWT 的微型版本，可以在移动终端上实现类似 PC 的用户界面。本章介绍了 kAWT 的基础编程知识和方法，并且给出几个 kAWT 编程的详细例子，可以指导读者进一步学习 kAWT 编程方法。

第 9 章 J2ME 的低级用户界面设计。如果希望在 J2ME 中实现功能更为强大的屏幕输出或者编制游戏程序，那么就必须使用 J2ME 的低级用户界面设计。本章详细介绍低级用户界面设计的基础类——Canvas 类，然后通过几个例子程序向读者演示如何处理用户输入：键盘输入和指针输入；如何编写程序启动画面；如何编写屏幕保护程序；如何编写屏幕清除程序以及在程序中如何显示和滚动图片等 J2ME 低级用户界面设计技术。

第 10 章 J2ME 游戏程序设计。J2ME 是一个很好的游戏平台，使用 J2ME 编写游戏是一件非常有趣的工作。本章详细介绍如何在游戏程序中响应用户输入；如何模拟实现连续按键；如何实现精灵的加速移动和连续移动；如何实现丰富多彩的屏幕显示效果；如何在支持透明图像和不支持透明图像的 KVM 平台上实现精灵的绘制。最后，讨论 J2ME 游戏程序设计中最高吸引人的地方——网络游戏设计。本章介绍了如何实现 J2ME 的网络游戏，并且给出了一个简单的网络游戏 NumberPick 的完整例子。通过这个例子，读者可以从整体上了解 J2ME 网络游戏是如何编写的。

第 11 章 J2ME 的无线消息处理。无线消息 API（Wireless Message API）为 J2ME 提供了发送和接收短消息的功能，重点介绍 WMA1.1（收发 SMS）和 WMA2.0（发送和接收 MMS）的程序设计过程，以及如何利用 J2ME 来使用 GPRS 网络连接进行无线通信。

第 12 章 J2ME 的移动多媒体程序设计。MMAPI（Mobile Media API）是 J2ME 设备的多媒体编程 API，本章重点介绍如何利用 J2ME 来实现音频、视频的播放、采集等功能，给出两个实例：利用 J2ME 实现的媒体播放器和照相机。

第 13 章移动软件设计技术。移动软件设计不同于普通模式软件，移动软件有其自身的体系架构。本章介绍移动软件设计的基本原理和架构，分析了 C/S、B/S 和混合模式的特点及不同，最后介绍了移动软件中的一个重要技术——PUSH 技术的实现。通过 PUSH 功能，可以实现设备的信息推送。

第 14 章给出了一个详细的例子——手机日志系统。该例使用 J2ME 来实现一个基于手机的日志系统，可以实现员工使用手机来书写、查看、统计工作日志的功能。这个例子充分展示了 J2ME 在界面设计、网络通信和数据库管理方面的强大功能，是一个非常实用的例子。

第 15 章给出了一个手机移动学习系统的例子。这个基于 J2ME 的手机移动学习系统面向移动教育，可以让用户利用手机随时随地地浏览、学习知识，并对学习结果进行考核，实现了一种新颖的移动学习功能。该系统分为 3 个部分：手机客户端软件、服务器端软件和数据库系统。手机移动学习系统集合了客户端 J2ME 程序和服务器端 Java 系统的强大功能，体现了最新的面向 3G 移动计算模式和架构，具有很典型的使用价值和示范作用。

本书特点

本书是一本全方位介绍 J2ME 编程技术的教程，内容包括 J2ME 基础知识、程序设计方法、高级编程技巧和应用实例分析等。

本书包含了近百个 J2ME 的编程实例。这些例子从最简单的 J2ME 程序到完整的网络应用

程序、游戏程序，覆盖了 J2ME 程序设计的方方面面。

本书中的例子程序，如手机日志系统、手机移动学习系统，大都具有很强的实用价值，略加修改就可以在实际生活和工作中使用。

本书每一章后均提供了习题，供读者对自己的学习效果进行检查。

通过对实例的学习和分析，读者能更深刻地理解 J2ME 程序设计的方法和技巧。

适用对象

本书内容安排由浅入深、循序渐进。

如果读者是一名熟练的 Java 程序员，那么本书是进入 J2ME 编程世界的一本极好的参考书籍。

如果读者对 Java 语言还不熟悉，那么通过本书的学习也可以掌握 Java 程序设计的基本方法和 J2ME 程序设计技术。

无论是对 J2ME 感兴趣的初学者，还是对 Java 程序开发工程师，本书都是学习和使用 J2ME 的最佳伴侣。

编写分工

本书由卢军、岳希、周辉编著，荣智涛审校。参与本书编写工作的人员还有：刘伟、王恒、曾茂城、李新星、邓丽君、兰琳、高明娟、张光兰、宁忠亮、陈文学、杨伟通、沈增光、陶武文、余鹏、黄钰耀、王治国、冯强、曾德惠、许庆华、程亮、周聪、黄志平、胡松、邢永峰、邵军、边海龙、刘达因、赵婷、马鸿娟、侯桐、赵光明、李胜、李辉、侯杰、王红研、王磊、闫守红、康涌泉、蒋杼倩、王小东、张正亮、宋利梅、何群芬、程瑶等，在此一并表示感谢。

特别说明

为充分体现本书的特点，帮助读者深刻理解本书的编写意图和内涵，进一步提高本书的教学使用效果，欢迎读者将本书在使用过程中的问题、探讨与建议反馈给我们。另外，我们提供本书的配套服务，举办“J2ME 移动软件程序设计与应用”培训，欢迎联络。我们的 E-mail: china_54@tom.com。

编者

2009 年 10 月

目 录

前言

第一篇 J2ME 基础知识

第 1 章 J2ME 简介	1	2.1.1 安装软件的过程	12
1.1 Java 版本的演进	1	2.1.2 开发环境的配置	12
1.1.1 Java 的演进历史	1	2.1.3 检查配置是否正确	14
1.1.2 Java 1 平台	2	2.2 第一个例子—— HelloWorld.java	14
1.1.3 Java 2 平台	2	2.2.1 HelloWorld.java 的源代码	14
1.2 各种 Java 版本的定位	2	2.2.2 编译、预校验和运行	15
1.2.1 Java 不同版本之间的关系	2	2.2.3 批处理——快速开发的技巧	16
1.2.2 J2SE、J2ME、J2EE 和 Smart Card	2	2.2.4 开发 MIDlet 的完整流程	17
1.2.3 不同 Java 版本的虚拟机	3	2.3 在网络上发布 MIDlet 程序	18
1.3 J2ME 与手机编程的关系	3	2.4 使用 J2ME Wireless Toolkit	20
1.3.1 手机编程的历史	4	2.4.1 J2ME Wireless Toolkit 的硬件和 软件要求	20
1.3.2 手机上的 Java 开发平台——J2ME	4	2.4.2 J2ME Wireless Toolkit 的安装步骤	20
1.4 Java 程序开发简述	4	2.4.3 在 J2ME Wireless Toolkit 中运行 例子程序	22
1.4.1 各种 Java 版本的程序开发关系	5	2.4.4 在 J2ME Wireless Toolkit 中创建 自己的程序	23
1.4.2 Java 程序开发流程	5	2.4.5 J2MEWTK 中的模拟设备简介	25
1.5 J2ME 简介	6	2.4.6 在 J2MEWTK 中进行 MIDlet 的打包	26
1.5.1 使用配置对不同设备进行抽象	6	2.5 使用 Eclipse 开发 J2ME 程序	27
1.5.2 使用框架对 API 进行定义	6	2.5.1 Eclipse 概述	27
1.5.3 J2ME 系统体系结构	7	2.5.2 Eclipse 的安装和配置	27
1.5.4 J2ME 已有的配置和框架	7	2.5.3 使用 Eclipse 开发 HelloWorld 程序	27
1.5.5 J2ME 的技术特性	7	本章小结	29
1.6 CDC、CLDC 和 MIDP 简介	8	习题二	29
1.6.1 CDC 和 CLDC 简介	8	第 3 章 CLDC 和 MIDP API 简介	31
1.6.2 典型框架简介	9	3.1 CLDC/MIDP 开发环境	31
1.7 J2ME 的一些参考网站	9	3.1.1 MIDlet 简介	31
1.7.1 主要 J2ME 的 Web 站点	9	3.1.2 应用程序可移植性	31
1.7.2 主要的 J2ME 讨论区	10	3.2 CLDC API 简介	31
本章小结	10		
习题一	11		
第 2 章 J2ME 开发环境	12		
2.1 JDK 的安装和配置步骤	12		

3.2.1	java.lang	32	3.3.4	javax.microedition.io	36
3.2.2	java.io	32	3.3.5	javax.microedition.lcdui	36
3.2.3	java.util	33	3.3.6	MIDP 中增加的功能	37
3.2.4	javax.microedition.io	34	3.4	HelloMidlet 的详细解释	38
3.3	MIDP API 简介	34	3.4.1	总体运行功能	38
3.3.1	java.util	34	3.4.2	各部分代码功能	40
3.3.2	javax.microedition.rms	35	本章小结		41
3.3.3	javax.microedition.midlet	35	习题三		42

第二篇 程序设计方法

第 4 章	J2ME 的用户界面编程	43	4.8.2	交互式标尺	65
4.1	J2ME 界面编程概述	43	4.8.3	非交互式标尺	66
4.1.1	Display 类和屏幕显示部件	43	4.9	有效地校验用户输入	68
4.1.2	典型的屏幕显示部件	44	4.9.1	监测部件状态的变化—— ItemStateListener 接口	68
4.1.3	界面设计部件的层次关系	49	4.9.2	对用户输入的字符进行校验	68
4.2	如何使用 Display 类	50	4.9.3	输入电话号码的例子	71
4.2.1	获得 MIDlet 的 Display 类	50	4.10	处理日期和时间 (DateField)	73
4.2.2	显示 Displayable 部件	50	4.10.1	使用 DateField	73
4.3	使用 Screen 类和滚动条	50	4.10.2	设置正确的日期和时间	74
4.3.1	Screen 类的基本属性	50	4.10.3	clearDate ()和 combineDateTime()	74
4.3.2	在 Screen 中使用滚动条	51	4.10.4	使用日期和时间的完整例子	75
4.4	使用命令按钮响应用户操作	52	4.11	定时执行任务 (Timer)	78
4.4.1	创建命令按钮	52	4.11.1	定时器——Timer	78
4.4.2	设置命令按钮的响应函数	53	4.11.2	调度任务的方法	79
4.4.3	使用命令按钮的例子	53	4.11.3	星空模拟程序	79
4.5	使用文本框 (TextBox) 输入信息	54	4.12	综合例子——股票价格跟踪程序	81
4.5.1	创建一个文本框	54	本章小结		84
4.5.2	文本框的演示例子	55	习题四		84
4.6	使用报警显示 (Alert) 提示信息	58	第 5 章	J2ME 数据库编程	86
4.6.1	两类报警: timed 和 modal	58	5.1	RMS 的表——Record Store	86
4.6.2	如何使用报警	58	5.1.1	操作 Record Store 的方法	86
4.6.3	演示报警使用的例子	59	5.1.2	Record Store 的维护信息——Header	88
4.7	向用户提供选择——列表 (List)	60	5.1.3	获得 Record Store 的其他信息	89
4.7.1	生成一个列表	60	5.1.4	Record Store 的限制	89
4.7.2	响应用户对列表的操作	60	5.2	在 Record Store 中操作记录	89
4.7.3	使用列表的完整例子	62	5.2.1	添加记录	90
4.8	显示或改变值的大小——标尺 (Gauge)	64	5.2.2	删除记录	91
4.8.1	标尺的两种类型——交互标尺和 非交互标尺	64	5.2.3	修改一条记录	92

5.2.4 监视记录的改变.....	95	6.4 网络通信中的数据加密	141
5.2.5 Record Store 中记录号的重用问题	97	6.4.1 MIDlet 的安全性	141
5.2.6 使用 RecordEnumeration 遍历记录	98	6.4.2 使用加密类库	141
5.3 创建多列表格	101	6.4.3 一个加密类——Encryptor	142
5.3.1 编码和解码的方法	101	6.4.4 Encryptor 的使用	144
5.3.2 多列表格的保存	102	6.4.5 数据加密的完整例子	144
5.3.3 检索多列表格	104	6.5 网络通信中的常用实例	147
5.4 使用 RecordComparator 对记录 进行排序	105	6.5.1 从 Web 服务器上取回一个页面	147
5.4.1 实现 RecordComparator 接口	105	6.5.2 从移动电话上访问 Servlet 并 获得结果	149
5.4.2 数据库排序的例子	105	6.5.3 与 Web 服务器建立 socket 连接	150
5.5 使用 RecordFilter 查找记录	108	6.5.4 从 Yahoo 获得股票信息	151
5.5.1 实现 RecordFilter 接口	108	6.5.5 Google Map API 的使用	155
5.5.2 查找记录的例子	109	本章小结	158
5.6 高效使用 RMS 的方法	112	习题六	159
5.6.1 减少写记录次数	112	第 7 章 J2ME 应用程序的性能优化	160
5.6.2 减少读记录次数	113	7.1 代码优化的一些基准	160
5.6.3 防止出现“抖动”	114	7.1.1 内存测试基准	160
5.6.4 重用数组	114	7.1.2 时间测试基准	161
5.6.5 高效率使用 RMS 的例子	115	7.2 优化内存使用	161
本章小结	118	7.2.1 创建和删除对象	161
习题五	119	7.2.2 字符串和字符串缓存	162
第 6 章 J2ME 的网络编程	121	7.2.3 正确处理内存不足	162
6.1 J2ME 的网络连接框架	121	7.3 优化代码运行速度	163
6.1.1 J2ME 的网络连接层次	121	7.3.1 优化循环	163
6.1.2 J2ME 支持的网络协议	122	7.3.2 使用数组代替对象	163
6.1.3 例子: StreamConnection 通信	123	7.3.3 使用缓存 I/O	163
6.2 使用 HTTP 协议通信概述	124	7.3.4 主动清理内存	164
6.2.1 HttpURLConnection 类	125	7.4 用户界面的优化	164
6.2.2 例子: 访问 Web Service	126	7.4.1 显示程序运行的进度	165
6.3 使用 HTTP 协议发送和接收数据	129	7.4.2 开发易于使用的程序	165
6.3.1 HTTP 协议通信模型	129	7.5 合理地打包应用程序	165
6.3.2 发送数据的方法	130	7.5.1 仅仅打包使用的代码	165
6.3.3 接收数据的方法	132	7.5.2 使用代码加密保护程序	165
6.3.4 服务器端的程序	133	7.6 J2ME 的程序兼容性	166
6.3.5 例子 1: 使用 HTTP 协议发送和 接收文本数据	135	7.6.1 J2ME 的程序兼容性	166
6.3.6 例子 2: 使用 HTTP 协议发送 二进制数据	138	7.6.2 JSR 介绍	166
		7.6.3 JSR 兼容性的测试工具	169
		本章小结	171

习题七	171
第 8 章 kAWT 编程	172
8.1 在 J2ME 中使用 kAWT	172
8.1.1 在 J2ME Wireless Toolkit 中 使用 kAWT	172
8.1.2 命令行方式使用 kAWT	173

8.1.3 KawtHello 的分析	174
8.1.4 定制编译 kAWT 类库	175
8.2 kAWT API 简介	175
8.3 kAWT 编程例子	176
本章小结	181
习题八	181

第三篇 高级编程技巧

第 9 章 J2ME 的低级用户界面设计	182
9.1 低级用户界面设计概述	182
9.1.1 屏幕显示的基础——Canvas 类	182
9.1.2 使用 Canvas 类的“Hello,World!” 例子	183
9.1.3 程序退出的方法	183
9.2 HappyPush——一个详细的例子	185
9.2.1 绘画出屏幕背景	185
9.2.2 响应用户操作	185
9.2.3 使用 Image 类画出图像	186
9.2.4 响应重复按键或指针移动事件	187
9.3 编写 MIDlet 的启动画面	191
9.3.1 使用报警的方法	191
9.3.2 使用 Canvas 类绘制启动画面	193
9.4 编写一个屏幕保护程序	195
9.4.1 如何实现 ScreenSaver	195
9.4.2 ScreenSaver 的代码	196
9.5 清除屏幕显示内容	199
9.5.1 清除屏幕的实现原理	199
9.5.2 清屏程序的例子	199
9.6 图片处理的例子	201
9.6.1 显示网络上下载的 Png 图片	201
9.6.2 如何实现滚动显示图像	204
本章小结	210
习题九	211
第 10 章 J2ME 游戏程序设计	213
10.1 在移动电话上开发游戏需要讨论 的问题	213
10.1.1 在移动电话上开发游戏的限制	213
10.1.2 开发游戏有利的方面	214
10.2 游戏编程基本技术	214

10.2.1 MIDlet 的基本结构	214
10.2.2 使用 Canvas 对象绘制图形	214
10.3 响应游戏中用户的按键	215
10.3.1 按键事件	215
10.3.2 游戏按键	215
10.3.3 响应键盘事件	215
10.3.4 使用通用按键	217
10.4 使用重复按键简化用户输入	218
10.4.1 判断系统是否支持重复按键	219
10.4.2 模拟重复按键	219
10.4.3 演示重复按键的例子	220
10.4.4 实现连续移动	227
10.4.5 实现加速移动	227
10.4.6 演示连续移动和加速移动的例子	229
10.5 在程序中实现屏幕滚动	237
10.5.1 定时器	237
10.5.2 用定时器实现重复按键	237
10.5.3 使用定时器实现屏幕滚动	238
10.6 实现镜头的移动	241
10.6.1 屏幕滚动和镜头移动	242
10.6.2 镜头移动的实现	242
10.7 在游戏程序中绘制精灵	245
10.7.1 在支持透明图形的系统上实现 绘制精灵	245
10.7.2 在不支持透明图形的系统中 实现绘制精灵	246
10.8 井字棋的例子	249
10.8.1 井字棋介绍	249
10.8.2 源代码	249
10.9 网络游戏编程	254
10.9.1 网络游戏介绍	254

10.9.2 数据通信的方法.....	254	本章小结.....	285
10.9.3 简单的网络游戏——NumberPick	255	习题十二.....	286
10.9.4 服务器端的代码.....	258	第 13 章 移动软件设计技术.....	287
本章小结.....	260	13.1 软件体系和计算模型.....	287
习题十.....	261	13.1.1 B/S 移动计算模型.....	287
第 11 章 J2ME 的无线消息处理.....	263	13.1.2 B/S 和 C/S 混合移动计算模式.....	288
11.1 无线消息与 WMA.....	263	13.2 基于 B/S 代理的移动计算模型.....	289
11.1.1 无线消息技术常用术语.....	263	13.3 基于移动 Agent 的移动计算模型.....	291
11.1.2 无线消息传递方式.....	263	13.3.1 移动 Agent 的概念.....	291
11.1.3 WMA 技术.....	264	13.3.2 移动 Agent 的优点.....	291
11.2 WMA 应用示例.....	264	13.3.3 移动 Agent 技术的关键问题.....	291
11.2.1 收发 SMS 文本短信示例.....	264	13.3.4 移动 Agent 计算模型.....	292
11.2.2 接收 CBS 文本消息示例.....	269	13.3.5 移动 Agent 计算模型的特点.....	293
11.3 建立 WAP 和 GPRS 连接.....	272	13.3.6 移动 Agent 系统的体系结构.....	293
11.3.1 WAP 和 GPRS 简介.....	272	13.3.7 移动 Agent 技术的应用和 发展趋势.....	294
11.3.2 建立 WAP 和 GPRS 连接.....	272	13.4 基于广域网的移动计算模型.....	295
本章小结.....	272	13.4.1 概述.....	295
习题十一.....	273	13.4.2 基于移动 Agent 的广域网普适 计算模型.....	296
第 12 章 J2ME 的移动多媒体程序设计.....	274	13.4.3 移动 Agent 在广域网中的服务 迁移机制.....	297
12.1 MMAPI 简介.....	274	13.4.4 移动 Agent 在广域网中的资源 迁移机制.....	299
12.1.1 MMAPI 中的重要成员.....	274	13.5 信息推送获取方式和 Push.....	300
12.1.2 MMAPI 与 MIDP 2.0 多媒体 API 的区别.....	275	13.5.1 MIDlet 启动过程.....	301
12.1.3 Player 的生命周期.....	275	13.5.2 PushRegistry 类.....	301
12.1.4 设备系统多媒体属性.....	276	13.5.3 静态注册和动态注册.....	301
12.1.5 MMAPI 编程基础.....	276	13.5.4 监听与启动.....	303
12.2 MMAPI 编程示例.....	277	13.5.5 处理数据.....	303
12.2.1 使用单音 (Tone).....	277	13.5.6 动态注册与基于计时器的 Push.....	304
12.2.2 单音序列音乐.....	279	本章小结.....	305
12.2.3 使用音频资源.....	281	习题十三.....	305
12.2.4 播放视频资源.....	282		
12.2.5 音频采集.....	284		
12.2.6 视频采集——照相.....	285		

第四篇 应用实例分析

第 14 章 手机日志系统的设计和实现.....	306	14.2.2 系统设计思路.....	307
14.1 手机日志系统的设计目的与工作现状.....	306	14.2.3 系统总体架构.....	307
14.2 系统总体设计.....	306	14.2.4 系统流程图.....	309
14.2.1 系统设计目标.....	307	14.3 服务器端的设计.....	310

14.3.1 数据库存储层设计	311	15.2.1 创建试卷	326
14.3.2 Web 服务器子系统设计	312	15.2.2 根据用户手机号码查询试卷	327
14.3.3 服务器端的代码实现和分析	312	15.2.3 手机端从服务器端下载题目	328
14.4 客户端设计	314	15.2.4 用户查看以往考试成绩	329
14.4.1 手机客户端程序设计	315	15.3 系统的实现流程	329
14.4.2 界面设计	316	15.3.1 客户端流程	330
14.4.3 程序的运行流程	319	15.3.2 数据库端流程	331
14.4.4 手机客户端的代码实现和分析	320	15.3.3 数据库结构	331
本章小结	321	15.4 系统测试和运行效果	333
习题十四	322	15.4.1 数据库端运行效果	333
第 15 章 手机考试系统的设计和实现	323	15.4.2 客户端运行效果	335
15.1 系统总体设计架构	323	本章小结	336
15.1.1 客户端模块管理	324	习题十五	336
15.1.2 数据库管理模块	325	附录 习题参考答案	337
15.1.3 题库系统管理模块	325	参考文献	343
15.2 试卷设计	325	参考资料	344

第一篇 J2ME 基础知识

第 1 章 J2ME 简介

随着 Java 技术的不断发展, Java 自身也根据市场进行了版本的细分。Java 2 分为针对企业级应用的 J2EE(Java 2 Enterprise Edition)、针对普通 PC 应用的 J2SE(Java 2 Standard Edition)和针对嵌入式设备及消费类电子产品的 J2ME (Java 2 Micro Edition) 3 个版本。

J2ME 推出之后, 全球各大计算机和消费类电子产品生产厂商积极响应, 如诺基亚、摩托罗拉、索尼爱立信、三星、TCL 等公司都推出了自己的支持 J2ME 技术的手机和其他电子产品。

本章首先对 J2ME 进行简单介绍, 让读者了解 J2ME 的发展历史和基本概貌。

1.1 Java 版本的演进

熟悉 Java 历史的读者都知道 Java 技术一开始并非就叫做 Java, 而是叫做 OAK (橡树)。OAK 最早是专门针对嵌入式系统而设计的产品。后来, 因为 Internet 的快速发展, 而 OAK 的很多特性 (如可移植性和编译后程序代码很小等) 又适合用在 Internet 上, 因此被改名为 Java。由此, Java 也才成了 Internet 应用上的一颗闪亮的明星。随着 Java 技术的发展, Java 在许多领域都应用起来, 逐渐产生了许多原来并没有预测到的功能。虽然 Java 已经被用到许多企业级软件上, 但是 Java 的设计初衷和特点还是非常适合应用在嵌入式系统和消费类电子产品中的。

1.1.1 Java 的演进历史

Java 版本的演进历史如图 1-1 所示。从图中可以看到从 Java 1.1 版本开始, Java 就细分为 Personal Java、Java 和 Embedded Java。当 Java 发展到 Java 2 后, Personal Java 和 Embedded Java 的版本划分方法就停止使用了, 取而代之的是将 Java 2 细分为 J2SE、J2EE 和 J2ME。

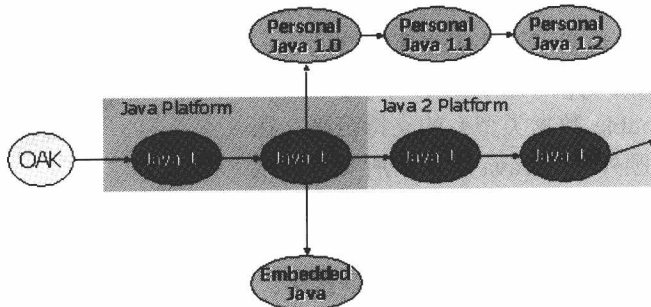


图 1-1 Java 版本的演进示意图

1.1.2 Java 1 平台

从图 1-1 中可以发现, 虽然从 Java 1.0 发表之后, Java 就被广泛地应用在桌上型应用程序以及 Applet 的开发上, 但从 Java 1.1 开始, Java 又回到了它的设计初衷——嵌入式系统和消费类电子产品上。当时, Sun 公司发表了针对嵌入式系统和消费类电子产品的 Embedded Java 与 Personal Java 两项 Java 规范。

其中, Embedded Java 是为资源十分有限且没有显示设备的装置设计的; Personal Java 则是为能够与 Internet 联网, 并拥有显示系统 (如彩色 LCD) 的消费类电子产品设计的。这两项技术从某种意义上可以看做 J2ME 的前身。

1.1.3 Java 2 平台

当 Java 的版本演进到 Java 2 时, Sun 公司就认识到 “one size doesn't fit all (一双鞋不能适用于所有的脚)”, 因此将其 Java 2 技术重新改组成 3 个版本: Micro (微型版, J2ME)、Standard (标准版, J2SE) 和 Enterprise (企业版, J2EE)。由于 Java 2 将 Java 应用区分成 3 种版本, 使 Java 的发展不像 Java 1.1 如树枝状扩散出去。这样有助于清晰地区分 Java 的各种应用, 而不会造成混乱的局面。

1.2 各种 Java 版本的定位

本节将介绍各种 Java 版本的应用领域和范围。要了解各种 Java 版本的应用领域和范围, 需要首先了解 Java 不同版本之间的关系。

1.2.1 Java 不同版本之间的关系

Java 规范中有一组类被称为核心类库 (Core Class, 即 `java.*`), 在核心类库外还有所谓的扩充类库 (Extended Class, 即 `javax.*`)。根据对这两种类库的支持程度, 可以将 Java 分为 4 种不同的版本, 如图 1-2 所示。

图 1-2 是按不同 Java 版本所支持的核心类库的不同分类的, 而不是按 Java 版本所支持的额外扩充类库分类的。此外, 图 1-2 也说明了这些版本所支持的 Java 基本数据类型 (Primitive Type, 即 `boolean`、`byte`、`short`、`int`、`long`、`float` 和 `double` 数据类型) 的支持范围。在图 1-2 中, 越在同心圆外圈的 Java 版本, 所支持的核心类库就越完整。反之, 在同心圆内圈的 Java 版本, 所支持的就仅仅是核心类库的子集, 而且越往里层, 所支持的核心类库子集就越小。同样, 在同心圆外部, 所支持的 Java 基本数据类型就越多, 而越在同心圆内部, 所支持的 Java 基本数据类型就越少。

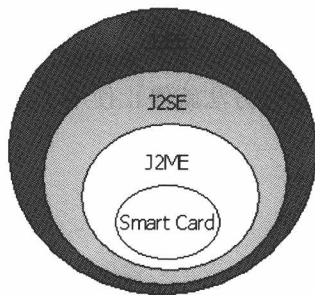


图 1-2 Java 不同版本之间的关系

1.2.2 J2SE、J2ME、J2EE 和 Smart Card

要理解图 1-2, 就必须以 J2SE 作为基准, 这个版本实现了所有 Java 标准规范中所定义的核心类库, 也支持所有的 Java 基本数据类型。J2SE 定位在普通桌面计算机应用上。

从 J2SE 往外延伸,其外部的同心圆为 J2EE。此版本支持所有的标准核心类库,并增加了许多支持企业级应用的扩充类库。例如,支持 Servlet/JSP 的 javax.servlet.*类库和 Enterprise Java Bean 的 javax.ejb.*类库。当然,J2EE 必定支持所有的 Java 基本数据类型。J2EE 定位在服务器端应用上。

从 J2SE 向内看是 J2ME,它仅支持核心类库的子集,例如在 J2ME 的 CLDC 规范中就只支持 java.lang.*、java.io.*及 java.util.*等基本类库。此版本也增加了一些支持“微小设备”的扩充类库,如 javax.microedition.io.*类库。然而,J2ME 并不支持所有的 Java 基本数据类型。以标准的 J2ME CLDC 为例,它就不支持属于浮点数(float 和 double)的 Java 基本数据类型。J2ME 定位在嵌入式系统和消费类电子产品应用上。

同心圆的最里层是 Java 的 Smart Card 版本。Smart Card 版本只支持 java.lang.*核心类库,比 J2ME 所支持的核心类库更少,但它也有属于自己的扩充类库,如 javacard.*和 javacardx.*等扩充类库。Smart Card 版本只支持 boolean 和 byte 两种 Java 基本数据类型。Java 的 Smart Card 版本定位在 Smart Card (智能卡)应用上。

1.2.3 不同 Java 版本的虚拟机

在以上每一种 Java 版本中,都有属于自己的虚拟机(VM, Virtual Machine)以实现 Java 的“Compile once, run anywhere (一次编译,到处运行)”目的。不同版本的 Java 虚拟机如图 1-3 所示。在 JavaCard 下有 JavaCard VM,负责执行下载到 Smart Card 上的 Card Applet;在 J2ME 下有 KVM (Kilobyte Virtual Machine, 千字节虚拟机),用来执行下载到嵌入式装置和消费类电子产品上的 MIDlet;在 J2SE 与 J2EE 下有 JVM (Java 虚拟机)和 HotSpot VM (HotSpot 虚拟机),负责执行 Java Applet、Java servlet 或 Java 应用程序。

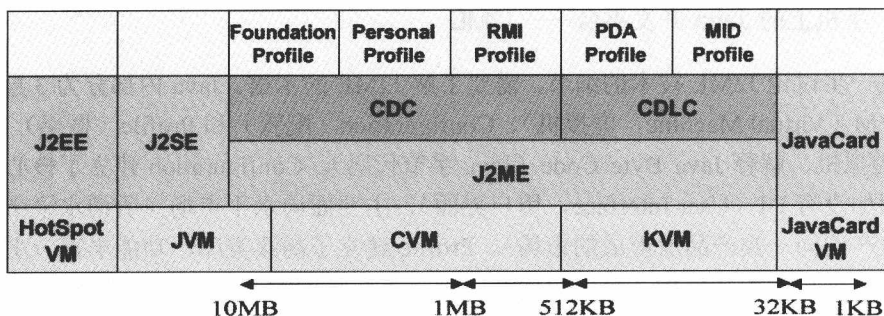


图 1-3 不同 Java 版本的下层虚拟机

HotSpot VM 是新一代的 JVM,目前广泛使用的 Classic VM 则是传统的 JVM。HotSpot VM 可以让 Java 程序的执行效率大幅提升。

以上所述的 VM 都只是 Javasoft 根据 Java 虚拟机规范实现的,实际上有许多公司都有自己开发出来的 VM,而且也符合 Java 虚拟机规范,例如 IBM 的 J9 VM。因此,JVM 不仅是由 Sun 提供,很多软件公司都提供自己的 JVM。

1.3 J2ME 与手机编程的关系

Sun 公司为了使 Java 语言进入嵌入式系统和消费类电子产品领域,在 Java 2 中推出了 J2ME。摩托罗拉是 J2ME 的主要支持者,第一个在手机上移植了 KVM,又领导全球手机厂家

制定了 MIDP 规范。在 2000 年的 JavaOne 大会上, 摩托罗拉总裁宣布: “摩托罗拉的主要手机机型都将支持 J2ME”。到目前为止, 几乎在所有厂商生产的手机上都支持运行 J2ME 程序。

1.3.1 手机编程的历史

回顾手机的发展历史, 可以看到手机上的软件经历了 3 个阶段。最早只有基本的语音服务, 而后增加了简单的附加应用(如电话簿), 最近又加入了 WAP(Wireless Application Protocol, 无线应用程序协议)方式访问 Internet 的功能。发展到当前这个阶段, 手机厂商遇到了两个问题: 一个是缺乏开放的编程平台; 另一个是访问 Internet 只能通过 WAP 方式。

传统手机类嵌入式系统普遍使用专用的实时操作系统, 基于 C 语言开发应用软件, 软件开发速度慢, 软件在不同设备之间缺乏良好的兼容性。所以, 单纯依靠手机厂商自身的软件开发能力难以满足市场的需求; 另一方面, 由于没有开放的编程平台, 广大软件开发商又无法开发适用于手机的应用程序。

Wireless Internet(无线 Internet)是未来 Internet 发展的必然方向, 但现在的 WAP 解决方案要求手机通过 WAP Gateway(WAP 网关)才能访问 Internet, 并且只能访问 WML(WAP Markup Language, WAP 标记语言), 而不是主流的 HTML(Hypertext Markup Language, 超文本标记语言)。这种 WAP 和 HTML 的不同, 将无线网络和 Internet 隔成了两个世界。技术的发展已经显现出来了这样的趋势: 未来无论是无线网络, 还是 Internet, 都会使用统一的页面语言——HTML。在技术上, WAP 最终将被抛弃。因此, WAP 不能解决手机网络通信问题。

利用 J2ME, 正好可以解决两个问题: 首先, Java 语言是跨平台运行的, 软件开发商可以很容易地开发应用程序, 也可以方便地安装在手机上; 其次, J2ME 提供了 HTTP 等 Internet 协议支持, Java 程序可以自由地访问 Internet。

1.3.2 手机上的 Java 开发平台——J2ME

为了进一步讨论 J2ME 技术的细节, 需要了解 J2ME 的术语。Java 内部分为 3 层, 从下到上分别是 VM(Virtual Machine, 虚拟机)、Configuration(配置)和 Profile(框架)。VM 负责建立 Java 虚拟机, 解释 Java Byte Code(Java 字节代码)。Configuration 建立了核心类库, 功能较少, 例如没有 UI(User Interface, 用户界面), 主要面向水平市场(所谓水平市场, 是指多个厂商生产的同一类产品所形成的市场)。Profile 建立了高级类库, 功能丰富, 主要面向垂直市场(所谓垂直市场, 是指某个厂商生产的同一类产品中的不同型号产品所形成的市场)。

针对手机内存少、速度慢和 I/O 差的特点, J2ME 对 VM、Configuration 和 Profile 三层做了特殊的实现。在 VM 层, J2ME 在手机上移植了 KVM, 只需要几百 KB 的内存就可以运行; 在 Configuration 层, J2ME 规定了 CLDC(Connected Limited Device Configuration, 连接的有限设备配置), CLDC 适用于拥有双向网络连接, 但是硬件资源有限的设备; 在 Profile 层, J2ME 规定了 MIDP(Mobile Information Device Profile, 移动信息设备框架), 适用于手机或其他移动设备。

1.4 Java 程序开发简述

Java 的不同版本之间在程序开发时不是截然分开的。实际上, 必须通过各个版本之间的配合才能开发出 Java 应用程序。

1.4.1 各种 Java 版本的程序开发关系

如前所述, 各种不同的 Java 版本在其支持的核心类库的完整性以及所支持的 Java 基本数据类型上都有所差异。但对于程序开发人员而言, 这些版本的关系如图 1-4 所示。无论开发的是企业所使用的 Java 程序、手机上执行的 Java 程序或浏览器上执行的 Applet, 还是在 PC 机上执行的应用程序, 用户都必须在自己的计算机上先安装 J2SE, 然后再安装各种版本的额外扩充类库, 这样才能成功地开发各种不同功能的 Java 程序。

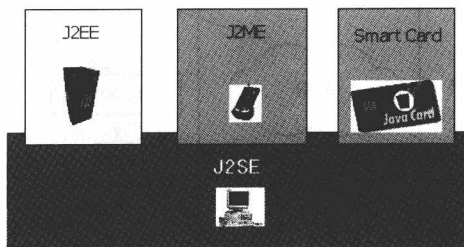


图 1-4 Java 各个版本的程序设计关系图

1.4.2 Java 程序开发流程

J2SE 所提供的 Java 编译器 (javac.exe) 可以帮助用户编译各种不同平台上的 Java 程序, 而 J2SE 所提供的 Java 虚拟机 (java.exe) 可以帮助用户在 PC 机上测试这些程序的执行结果是否正确。所以, J2SE 是各种 Java 版本程序开发的共同基础平台。

此外, Java 编译器并不会帮用户检查程序是否符合各种平台上所支持的核心类库与 Java 基本数据类型。例如, 在 Smart Card 版本中并不支持除 boolean 和 byte 以外的 Java 基本数据类型, 而且该平台也只支持 java.lang.* 核心类库。但当编写 Smart Card 平台上的程序时, 即使在程序代码中使用了除 boolean 和 byte 以外的 Java 基本数据类型, 或者使用了除 java.lang.* 之外的其他核心类库, 编译器仍然可以照常成功编译。这样编译成功的程序放到 Smart Card 上执行肯定会出现问题。因此, 针对这个问题, 在各种不同版本的开发工具中, 通常会附加检查器 (Checker) 或预先审核器 (Preverifier), 这两个工具可以帮助用户在将程序放到目标平台上运行之前先做好检查和预先审核的工作, 防止出现上述的问题。

检查器会帮用户找出程序中使用的类库是否有不符合目标平台规范的部分, 并提醒用户这些地方可能无法在目标平台上执行。因此, 只要有检查器的协助, 就可以确定应用程序是否符合目标平台的规范并可以顺利执行。

在某些平台的开发工具中附有预先审核器 (如 J2ME 平台)。预先审核器的作用可以从如图 1-5 所示的 Java 程序执行过程中找到答案。

在传统的 Java 程序中, 出于安全考虑, 任何进入执行环境的 class 文件 (不管该 class 文件是来自本机还是远程机器), 都必须先经过 Byte Code Verifier (Byte Code 审核器) 的验证, 以防止程序在传送过程中遭到恶意修改, 而使 Java 程序在执行时对系统产生不良影响。经过验证之后的 class 文件才能在 JVM 中执行。

如果审核工作在一般的 PC 机上执行, 在速度方面还能接受, 但是放到像手机这样的 CPU 运行速度较慢, 并且内存空间也较小的机器上运行就显得十分吃力了。为了节省宝贵的 CPU 运算时间 (既能省电又能加速程序执行), 在生成能够在特定平台上执行的 class 文件之前, 必