



Java ME

嵌入式程序设计

张家耀 何雪芳 宋 梅 主编



高等学校工程创新型「十二五」规划计算机教材



Engineering Innovation



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

高等学校工程创新型“十二五”规划计算机教材

Java ME 嵌入式程序设计

张家耀 何雪芳 宋 梅 主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书共 11 章, 内容包括: Java ME 概述, Java ME 开发环境与工具, 图形用户界面体系结构, 高级用户界面设计, 低级图形用户界面, MIDP 游戏程序设计基础, MIDP 网络编程, MIDP 记录存储器, MMAPI 多媒体程序设计, 无线消息程序设计, 应用程序管理软件。本书内容涵盖 Java ME 嵌入式程序设计的主要领域, 并反映 Java ME 程序设计的新成果。书中例题全部经过精心设计, 既能帮助理解基础知识, 同时具有启发性, 程序略加修改就可以应用到实际手机上。

本书适合作为高等学校计算机及相关专业本科教材, 也可作为有一定 Java 语言基础的移动通信开发爱好者的参考书。

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

Java ME 嵌入式程序设计 / 张家耀, 何雪芳, 宋梅主编. —北京: 电子工业出版社, 2012.1

高等学校工程创新型“十二五”规划计算机教材

ISBN 978-7-121-14765-4

I. ①J… II. ①张…②何…③宋… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 203705 号

策划编辑: 章海涛

责任编辑: 冉 哲

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

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: 787×1 092 1/16 印张: 17.25 字数: 460 千字

印 次: 2012 年 1 月第 1 次印刷

印 数: 3 000 册 定价: 34.00 元

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, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

随着网络和无线通信技术的发展,手机不再是单一的通信工具,它还具有网络浏览、个人信息管理、移动办公、游戏娱乐等功能。随着 3G 技术的发展,数据通信费用大大降低,对移动通信的需求将越来越大。

当前手机操作系统并不统一,从老牌的 Symbian 和 Windows Mobile,到后起之秀 iOS 和 Android,都有不少客户群。那么,面对众多平台,如何找到统一的开发方法,开发出对各种操作系统都兼容的软件产品呢?Java ME 是最好的选择,它提供高度优化的 Java 运行环境,可实现跨平台开发,并有开发成本低、周期短等优点。

本书内容涵盖 Java ME 嵌入式程序设计的主要领域,注重知识的科学性与可接受性相结合,同时注重系统性与实用性相结合。“Java ME 嵌入式程序设计”课程的先修课程为“Java 程序设计”。

本书首先介绍 Java ME 的基础知识,包括:Java ME 概述、Java ME 开发环境与工具、图形用户界面体系结构、高级用户界面设计和低级图形用户界面。在这些知识的基础上进一步介绍对于手机程序开发非常实用的内容,包括:MIDP 游戏程序设计基础、MIDP 网络编程、MIDP 记录存储器、MMAPI 多媒体程序设计、无线消息程序设计和应用程序管理软件。本书第 9-11 章的内容反映了 Java ME 程序开发领域的新成果。

书中例题经过精心筛选,既能帮助读者理解基础知识,同时具有启发性,并且所有程序都经过测试,略加修改就可以应用到实际手机上。

本书提供教学资源包,包括例题程序源代码,可登录华信教育资源网注册后免费下载。

本书可作为高等学校计算机及相关专业本科教材,也可作为有一定 Java 语言基础的移动通信开发爱好者的入门书籍。

本书在编写过程中,参考了许多相关书籍和资料,在此对这些作者表示感谢。另外,感谢电子工业出版社在本书出版过程中给予的支持和帮助。最后,感谢这几年使用本书并提出宝贵意见的学生,特别是龙虹宏、巫梦娇和张狄,他们协助完成了本书的校对工作。

因作者水平有限,书中难免存在错漏和不妥之处,望读者指正,以利改进和提高。

作 者

目 录

第 1 章 Java ME 概述	1
1.1 Java ME 平台	1
1.2 Java ME 体系结构	1
1.3 移动信息设备简表	3
1.3.1 MIDP 目标设备的特性	4
1.3.2 MIDP 的功能范围	4
1.3.3 MIDP 类库	4
1.4 MIDP 应用程序——MIDlet	5
1.5 MIDlet 套件	7
1.5.1 清单文件	8
1.5.2 JAD 文件	9
1.5.3 Java 应用程序管理器——JAM 或 AMS	10
1.6 Java ME 的标准规范	11
习题 1	12
第 2 章 Java ME 开发环境与工具	13
2.1 安装无线工具包 WTK	13
2.2 使用 WTK 开发 MIDP 应用程序	15
2.2.1 编写程序源代码	15
2.2.2 简单开发周期	16
2.2.3 设置 WTK 开发环境	18
2.2.4 完整开发周期	21
2.3 使用 Eclipse 开发环境开发 MIDP 应用程序	22
2.3.1 Eclipse 概述	22
2.3.2 安装 Eclipse 和 EclipseME	22
2.3.3 配置 Eclipse	26
2.3.4 使用 Eclipse 创建 MIDlet 套件	28
2.3.5 创建 MIDlet	30
2.3.6 运行 MIDlet	31
2.3.7 打包与混淆	33
习题 2	34
第 3 章 图形用户界面体系结构	35
3.1 LCDUI 体系结构	35
3.2 Display 类	36

3.3	Displayable 类	39
3.4	Command 类和 CommandListener 接口	41
3.4.1	Command 类	41
3.4.2	CommandListener 接口	42
	习题 3	44
第 4 章	高级用户界面设计	45
4.1	高级用户界面——Screen 类	45
4.1.1	Screen 类概述	45
4.1.2	文本框——TextBox 类	46
4.1.3	信息窗口——Alert 类	49
4.2	选择实现——Choice 接口	51
4.2.1	构造方法	51
4.2.2	编辑 Choice 对象	52
4.2.3	Choice 对象的选项类型	52
4.3	选择列表——List 类	53
4.3.1	构造方法	53
4.3.2	列表选项的编辑	53
4.3.3	列表选项的类型	55
4.4	容器控件——Form 类	57
4.4.1	Form 类概述	58
4.4.2	组件——Item 类	59
4.4.3	StringItem 类	61
4.4.4	文本区域——TextField 类	63
4.4.5	图像操作——ImageItem 类	65
4.4.6	空间填充控件——Spacer 类	68
4.4.7	选择组组件——ChoiceGroup 类	68
4.4.8	处理日期和时间组件——DateField 类	71
4.4.9	进度条——Gauge 类	73
4.4.10	自定义组件——CustomItem 类	75
	习题 4	75
第 5 章	低级图形用户界面	78
5.1	画布 Canvas 类	78
5.1.1	Canvas 类概述	78
5.1.2	画布规格与布局	78
5.1.3	绘制和重绘制	79
5.1.4	画布可视化	79
5.2	图形绘制 Graphics 类	81

5.2.1	绘制简单图形	82
5.2.2	绘制文本	89
5.2.3	图像绘制	93
5.2.4	计时器	98
5.3	低级事件处理	102
5.3.1	标准键盘事件	102
5.3.2	游戏动作按键事件	104
5.3.3	指针事件	106
	习题 5	111
第 6 章	MIDP 游戏程序设计基础	112
6.1	游戏程序设计概述	112
6.2	游戏画布	113
6.3	游戏画布上的按键处理	117
6.4	图层 Layer 类	121
6.4.1	分块图层 TiledLayer 类	122
6.4.2	精灵 Sprite 类	129
6.4.3	图层管理 LayerManager 类	142
6.4.4	简单游戏举例	144
	习题 6	144
第 7 章	MIDP 网络编程	145
7.1	移动网络编程概述	145
7.1.1	CLDC 的通用连接框架	145
7.1.2	通用连接框架中的类	146
7.2	HTTP 编程	148
7.2.1	MIDlet 连接 HTTP 服务器	148
7.2.2	获取 HTTP 连接的基本信息	149
7.2.3	手机客户端与 HTTP 服务器通信	151
7.3	套接字编程	161
7.3.1	客户端与服务器的套接字连接	162
7.3.2	套接字连接可以得到的基本信息	162
7.3.3	套接字连接通信	165
7.4	UDP 数据报编程	172
7.4.1	客户端与服务器端数据报连接	172
7.4.2	数据报的传递	173
	习题 7	180
第 8 章	MIDP 记录存储器	181
8.1	RMS 概述	181

8.2	记录存储器	182
8.2.1	管理记录存储器	182
8.2.2	RecordStore 类中存取记录存储器信息的常用方法	183
8.3	记录的基本操作	187
8.3.1	简单记录的读/写	187
8.3.2	复杂记录的处理	192
8.4	记录的遍历、查询和排序	199
8.4.1	记录的遍历	199
8.4.2	建立和使用过滤器	200
8.4.3	建立和使用排序器	200
8.5	记录存储器的事件处理	207
8.6	综合实例	207
	习题 8	207
第 9 章	MMAPI 多媒体程序设计	208
9.1	MMAPI 概述	208
9.1.1	MMAPI 的体系结构	208
9.1.2	管理器 Manager 类	209
9.1.3	播放器 Player 接口	210
9.1.4	数据源 DataSource 类	211
9.1.5	控制器 Control 接口	211
9.2	音频播放	212
9.3	视频播放	216
9.4	手机拍照的实现	221
	习题 9	226
第 10 章	无线消息程序设计	227
10.1	无线消息概述	227
10.1.1	GSM 短消息服务	227
10.1.2	GSM 小区广播	228
10.2	WMA 概述	228
10.3	使用 WTK 中的 WMA 控制台	229
10.3.1	配置和启动 WTK 中的 WMA 控制台	229
10.3.2	使用 WMA 控制台发送文本消息	231
10.3.3	使用 WMA 控制台发送小区广播	232
10.3.4	使用 WMA 控制台发送多媒体消息	233
10.4	编写利用 WMA 控制台收发短消息的程序	234
10.4.1	发送和接收 SMS 消息	234
10.4.2	发送和接收二进制消息	239

10.4.3 发送和接收多媒体消息	245
习题 10	251
第 11 章 应用程序管理软件	253
11.1 MIDP 应用程序经 OTA 下载安装	253
11.1.1 OTA 下载安装过程	253
11.1.2 在 Eclipse 中形成 JAR 文件和 JAD 文件	253
11.1.3 在服务器中搭建 OTA 下载环境	254
11.1.4 从服务器中 OTA 下载应用程序	254
11.2 Push 技术	256
11.2.1 概述	257
11.2.2 PushRegistry 类的主要方法	257
11.2.3 静态注册方式	258
11.2.4 动态注册方式	263
11.2.5 使用 Push 技术应注意的问题	265
习题 11	265
参考文献	266

第 1 章 Java ME 概述

本章简介：本章重点讲解 Java ME 的体系结构，明确 Java ME 具有自下而上的分层结构：操作系统、虚拟机、配置、简表和可选包。首先分析每一个软件层次的组织结构，随后介绍 MIDP 应用程序——MIDlet 的基本结构和 MIDlet 套件的基本组成，最后介绍 Java ME 的相关规范。

移动互联网的时代到来了！这是越来越多的人从不同的角度喊出了同一口号。

现在，几乎每一个人手中都有一部手机，但他们中的很多人没有 PC 机，手机普及率远高于 PC 机。近日发布的调查显示：我国通过手机上网的用户已经超过 1.176 亿人。随着我国具有独立知识产权 3G 技术的普及和发展，必然开拓出属于手机应用的广阔空间。想象将来的某一天可以用手机打开车门，在回家的路上通过手机打开热水器准备好洗澡的热水，用手机把刚刚收集到的数据发给老板，汇报工作。

然而，手机操作系统市场并不像 PC 机那样有统一：从老牌的 Symbian 和 Windows Mobile 到后起之秀 iPhone 和 Android 都有着不少的客户群，而且可以预见，不久的将来还会有更多的厂商进入这个炙手可热的领域。面对如此众多的平台，如何找到一种统一的开发方法，开发出对各种操作系统都兼容的软件产品呢？Java Me 为我们提供了一个很好的解决方案。

1.1 Java ME 平台

1991 年，Sun 公司中由 James Gosling、Bill Joe 等人组成 Green 小组开发了名为 Oak 的软件，其目的是用于电视机等家用电器的程序开发。Oak 语言是 Sun 公司为一些消费产品设计的一个通用环境，最初的目的只是开发一个独立于平台的软件技术，后来发展成为 Java。Java 语言发布之后风靡 WWW 世界，广泛应用于网络计算。Java 语言的设计特点是简单、安全、易于维护、可移植性强。它采用了虚拟机技术，把源程序编译成二进制的中间代码，然后在设备虚拟机上运行，这就是“一次编程、到处运行”的思想。在 Java 网络应用大获成功的同时，也面临着更多设备都要接入互联网这样的挑战。尤其是众多厂家和型号的手机要接入互联网，它们的接入设备操作系统不同，输入、输出方式各异，内存和处理机的能力有限，因此对其可移植性提出了更强的要求。为了解决这个问题，Sun 公司推出了 Java 的微型版，即 Java ME (Java Platform Micro Edition)。

Sun 公司将 Java Me 定义为“Java Platform, Micro Edition provides a robust, flexible environment for application running on mobile and other embedded devices”。好了，从现在开始我们希望能用 Java Me 这把嵌入式开发利剑统一手机应用程序开发这一混乱的领域。

1.2 Java ME 体系结构

Java ME 是专门面向小型手持设备，用于嵌入式应用软件开发的平台，可以应用于移动电话、个人数字助理 (PDA)、网络 IP 电话、机顶盒、家庭娱乐多媒体系统、信息家用电器及车载导航等系统中。

Java ME 面对的是大量不同的设备，这种不同不仅表现为硬件结构的不同，还表现为功能应用的不同。为了适应众多不同设备的需求，Java ME 在组织这些设备的软件时采用了分层的概念，它将运行在硬件上的软件分成若干个层次，如图 1-1 所示。

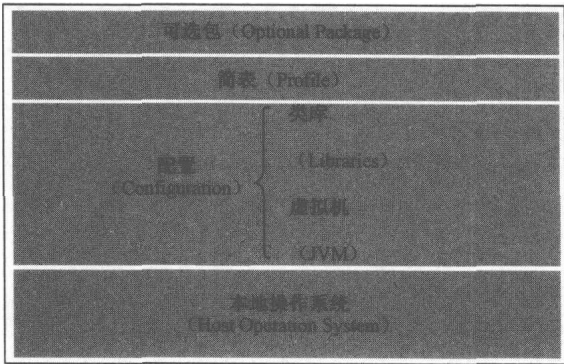


图 1-1 J2ME 软件的层次结构

1. 本地操作系统

在软件的底层是本地操作系统（Host Operating System），它负责管理和配置该手持设备的所有硬件，也就是说，其他所有的软件必须通过本地操作系统才能使用硬件资源。当前流行的操作系统主要有以下几种：

- 用于开发移动电话的 Symbian、Linux、Windows Pocket PC、Palm OS 以及最近迅速发展的 iPhone 和 Android 等；
- 用于开发其他嵌入式设备的 VxWorks、Linux、Windows CE 等。

2. 虚拟机

在操作系统之上是 Java 虚拟机（Java Virtual Machine, JVM）。所有的 Java 程序只能运行在 JVM 上，为此，要实现 Java 开发平台必须首先实现 JVM。对于 Java ME 的标准参考实现的虚拟机有 CVM 和 KVM 两种。

CVM 虚拟机：允许设备将 Java 线程映射为本地线程，完成垃圾收集、Java 同步等功能；在可移植性方面，采用 C 语言实现，可实现快速、安全的移植。CVM 虚拟机适用于瘦客户端，如数字电视机顶盒、车载电子系统等。

KVM 虚拟机：KVM 的最大特点是小而高效，只需要几万字节的存储空间就可以运行。KVM 虚拟机和类库只需占有 K 量级的存储空间，即 50~80KB，具有较高的可移植性和可扩展性。KVM 专门执行下载到低端 CLDC 设备上的 MIDlet 程序。KVM 虚拟机是根据资源受限设备完全重新编写的，它不是现有标准 Java 虚拟机的改进产品。KVM 虚拟机常应用于那些电池供电的手持移动设备，如移动电话、PDA 等。

CVM 和 KVM 适用的硬件资源，前者高后者低，根据不同的硬件可以选择不同的虚拟机。KVM 在功能上是 CVM 的子集。

虽然 Sun 公司推荐 CVM 和 KVM 作为 Java ME 的虚拟机，但这不是必须的，实际上，只要满足配置中的规范定义，通过兼容性测试就可以作为虚拟机应用在小型设备上。

3. 配置

在虚拟机之上的软件层次是配置（Configuration）。Java ME 支持的硬件有很大的差异，不可

能构建一个适用于所有设备的开发平台。为了满足不同设备的开发需求,Java ME 引入了配置的概念,屏蔽了不同硬件设备的物理特性。它包含一些核心的类库,定义了 Java 虚拟机类型和一些基础 API。当前 Java ME 存在两种配置 CDC 和 CLDC。

将具有固定连接的、不间断网络连接的共享连接信息设备,如数字电视机顶盒、网络电视(Web TV)、支持 Internet 的可视电话和汽车娱乐/导航系统等归为一类,称为连接设备配置(Connected Device Configuration, CDC)。

另一类是具有间断网络通信能力的个人移动信息设备,如手机、双向寻呼机、个人数字助理 PDA、销售点终端 POS 等,称为连接受限设备配置(Connected Limited Device Configuration, CLDC)。

在编写 Java ME 的程序之前,要根据运行程序的硬件情况选择合适的配置。

4. 简表

在配置之上的软件层次是简表(Profile),它是某个行业或者某个领域内的特性概括,每套简表专门针对某一类设备。例如,移动电话具有一套简表,PDA 也具有一套简表,数字电视机顶盒具有另外一套简表。

简表与它的上一层配置必须组合使用。简表以配置为基础定义了一些附加的类和包,简表是专门针对某个特定行业或某类设备上使用 API 的最小集合。

每个配置上都定义了若干个简表,如前所述,Java ME 中包含两种配置 CDC 和 CLDC,相应地,存在着两套支持不同配置的简表。支持 CDC 的简表有:FP (Foundation Profile, 基础简表)、PP (Personal Profile, 个人简表)和 PBP (Personal Basic Profile, 个人基础简表)。

支持 CLDC 的简表目前只有 MIDP (Mobile Information Device Profile, 移动信息设备简表),它是目前移动电话上使用的主要简表。

5. 可选包

Java ME 通过 CLDC、CDC 及其对应的简表规范了小型设备大部分的通用功能,为了保持良好的扩展性,J2ME 引入了可选包(Optional Package),达到进一步扩展功能的目的。

下面介绍一些常用的、功能强大的可选包。

蓝牙功能:它提供用于开发蓝牙通信程序的接口,当然小型设备中必须具有蓝牙设备才能运行该接口编写的程序。

无线消息功能:它支持无线消息以与平台无关的访问方式访问无线资源,它既支持 CDC 也支持 CLDC。

移动多媒体功能:它提供了在小型移动设备上处理音频和视频等多媒体的能力,包括播放音频视频、录制语音等。它允许在具备摄像功能的设备上录制视频,允许在具备麦克风的设备中录制音频。

移动 3D 图形功能:主要用于 CLDC 1.1 平台,支持开发 3D 图形程序,特别是 Java 3D 游戏。

可选包一般针对新兴的技术,进行一些试验性的探索,如蓝牙、Web 服务、无线消息等,一旦技术成熟,这些可选包就很可能合并到简表甚至配置中。

1.3 移动信息设备简表

移动信息设备简表 MIDP 是建立在 CLDC 连接受限设备配置基础上的简表,它定义了移动信息设备特定设备家族的特殊需求,是本书重点介绍的信息设备。

1.3.1 MIDP 目标设备的特性

MIDP 定义的移动信息设备的特性如下：

- 小屏幕尺寸，单色或彩色的显示设备；
- 单手数字键盘、双手标准键盘或触摸屏的输入设备；
- 较少的易失性内存，非易失性内存一般需要 128KB 用于 MIDP 组件，8KB 用于持久存储数据，32KB 用于 Java 的虚拟运行时环境（注意，这里并不包括 CLDC 对内存的需求）；
- 有限带宽条件下的无线通信；
- 音频和视频采集设备。

1.3.2 MIDP 的功能范围

目前，MIDP 有两个版本 v1.0 和 v2.0。1.0 版本的功能包括应用程序的下载、网络连接和传输、数据库存储、计时器和用户界面。2.0 版本扩展了的功能有：应用下载的计费、网络安全传输、数字签名、域的安全模式、注册和音频处理。MIDP 功能结构图如图 1-2 所示。

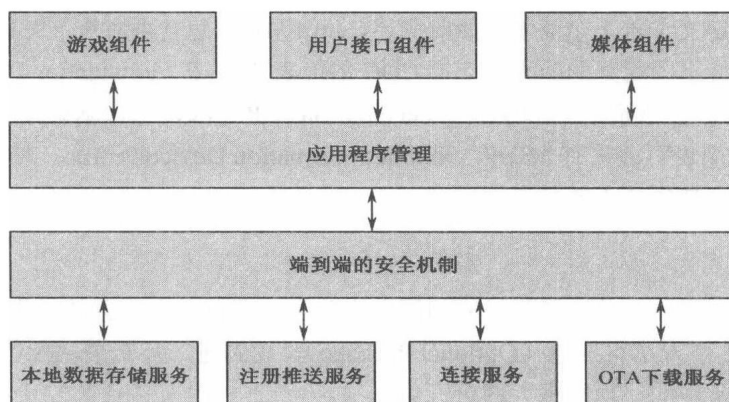


图 1-2 MIDP 功能结构图

目前，大多数移动手机都支持 MIDP v2.0 版的简表，本书也将以 2.0 版本为例，介绍 Java ME 平台的编程方法。

1.3.3 MIDP 类库

1. CLDC 的软件包

MIDP 类库是建立在 CLDC 类库基础上的，它可以使用 CLDC 的 4 个软件包中的类，其中有 3 个是从 Java 标准版继承的，另外一个 CLDC 所特有的。

（1）从 Java 标准版继承的软件包

① 核心输入、输出包

java.io：通过数据流提供系统的输入、输出。

② 核心语言包

java.lang：定义 MIDP 的语言类，在 CLDC 中 java.lang 包的基础上增加了类 java.lang.Illegal_

StateException。它是一个 RuntimeException，指出在不合法或不合适的时间调用了方法。例如，在一个 TimerTask 安排中调用或者在用户界面组件容器中请求时抛出这个异常。

③ 核心实用工具包

java.util: 定义 MIDP 的工具类，在 CLDC 中 java.util 包的基础上增加了类 java.util.Timer 和 java.util.TimerTask。java.util.Timer 用于为后台线程中将要执行的任务确定时间；java.util.TimerTask 被 java.util.Timer 类使用，用于为后台线程中执行定义任务。

(2) CLDC 特有的包

javax.microedition.io (网络包): MIDP 提供了基于 CLDC 通用连接框架的支持，在 CLDC 的基础之上，新增加了一个接口 javax.microedition.io.HttpConnection，为建立 HTTP 连接提供必要的方法和常量。

2. MIDP 自己特有的包

(1) 用户界面包

javax.microedition.lcdui: 为 MIDP 应用程序提供用户界面。

(2) 游戏包

javax.microedition.lcdui.game: 为 MIDP 应用程序提供游戏 API (MIDP 2.0 规范支持)。

(3) 数据持久存储包

javax.microedition.rms: 用来为 MIDP 提供数据持久存储机制。应用程序可以存储数据记录，以供在以后需要时获取。

(4) 应用程序生命周期包

javax.microedition.midlet: 定义 MIDP 应用程序，以及应用程序和它所运行环境之间的交互。

(5) 声音媒体包

javax.microedition.media: 提供对移动多媒体的支持。

javax.microedition.media.control: 用来提供对声音媒体的管理、播放和控制功能 (MIDP 2.0 规范支持)。

MIDP 和 CLDC 的特有包是本书讲述的核心内容，在以后的各章中都会指出其所在的包。

1.4 MIDP 应用程序——Midlet

Midlet 是 MIDP 应用程序的基本执行单元，是 Sun 公司对 MIDP 上的应用程序的一个独特叫法，意思是“MIDP 小应用程序”。它是为满足小型资源受限设备的特殊要求，由 MIDP 规范所定义的一种全新的应用程序模型。Midlet 与标准 Java 程序中的 Applet 小应用程序一样，必须运行在某特定的环境中，或者说，运行在作为容器的大应用程序中。这个大容器可以适应不同厂家、不同型号和多种功能的移动设备。然而这个容器的内部为 Midlet 提供了一个统一的接口。程序的开发者只需要编写规范的 Midlet 就可以了。

下面考察一个最简单的 Midlet 应用程序 (HelloChina 程序)，通过这个例子可以了解 Midlet 程序的基本结构特征。

例 1-1 在屏幕上显示“中国 你好!”。

源程序名: HelloChina.java

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class HelloChina extends MIDlet implements CommandListener {
    private Display display;
    private Command exitCommand;
    private TextBox t;
    public HelloChina() {
        display = Display.getDisplay(this);
        exitCommand = new Command("Exit", Command.SCREEN, 1);
        t = new TextBox("Hello", "中国 你好!", 256, TextField.ANY);
        t.addCommand(exitCommand);
        t.setCommandListener(this);
    }
    public void startApp() {
        display.setCurrent(t);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
    public void commandAction(Command c, Displayable s) {
        if(c==exitCommand) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}

```

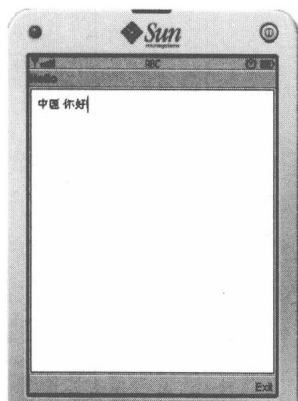


图 1-3 HelloChina 运行结果

经过编译，在仿真器上的运行结果如图 1-3 所示。

从这个简单程序可总结出如下 MIDlet 程序的基本特征。

(1) 每一个 MIDlet 程序必须是 javax.microedition.midlet.* 包中 MIDlet 类的子类。

作为 MIDP 应用程序的 MIDlet，它一定是 MIDlet 类的子类。这样，设备的应用管理软件才能对 MIDlet 进行管理和控制。在这个例子中，MIDlet 应用程序 HelloChina 类就是继承 MIDlet 类而来的。

(2) 每一个 MIDlet 可以有一个构造方法。

MIDP 应用程序模型规定，构造方法仅被系统调用一次，用来初始化一个 MIDlet 的状态。构造方法要执行的操作取决于程序的需要。通常，将所有要在启动时执行，且只执行一次的操作放在构造方法中。同时，应捕获在构造方法中可能出现的异常，并处理之。

(3) 每一个 MIDlet 必须实现用于控制程序生命周期的三个抽象方法。

当 MIDlet 被初始化后就进入它的生命周期，该生命周期包含三个状态：激活态 (Active)、暂停态 (Paused) 和销毁态 (Destroyed)，这三种状态的控制完全由应用程序管理器 JAM (Java Application Manager) 来完成。当出现状态迁移时，JAM 会自动调用 MIDlet 在本例中出现的三个抽象方法：startApp()、pauseApp() 和 destroyApp()，如图 1-4 所示。需要注意状态迁移与方法调用的因果关系，状态迁移是因，方法调用是果。

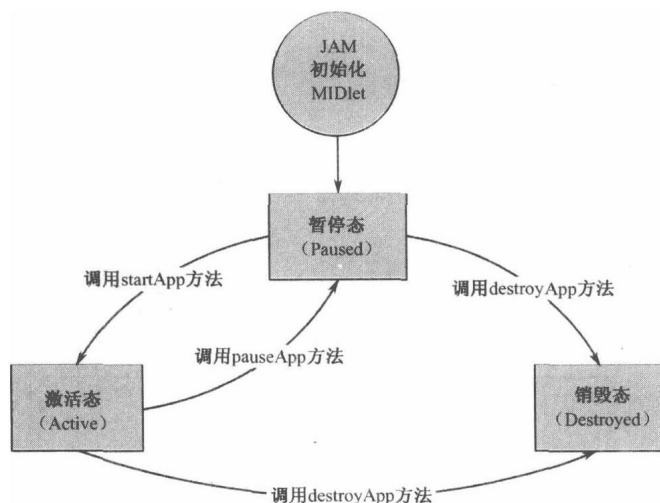


图 1-4 MIDlet 状态迁移触发的方法调用

JAM 加载 MIDlet 后，首先进行初始化的工作，将状态设置为暂停态。如果在初始化的过程中没有出现任何异常，那么 MIDlet 的状态转为激活态，否则将 MIDlet 的状态设置为销毁态。在每次状态转换的时候都要调用相应的方法。

为此，在定义 MIDlet 类时需要实现这些状态转移时调用的方法，从而完成自定义的过程。一般，在 startApp 方法中定义程序运行时用到的资源，在 pauseApp 方法中释放暂时不使用的资源（避免因状态切换使得系统资源被占用），在 destroyApp 方法中释放所有自己定义的资源。

在任何时刻，JAM 都可能改变 MIDlet 的状态，为此这些方法有可能多次被调用，于是问题产生了。如果多次调用 startApp 方法就意味着多次定义资源，为此必须设计好一个资源定义的规划。一般来说，在 startApp 方法中定义那些临时性的资源，在 pauseApp 方法中释放，除 startApp 方法中定义的那些资源外，剩余的资源在 MIDlet 构造方法中定义。

(4) MIDlet 通常都会实现 CommandListener 接口。

CommandListener 接口的实现是为了使应用程序对用户的操作做出反应。这个接口及 TextBox、Command 和 Display 类，都是 javax.microedition.lcdui.*包的一部分。

1.5 MIDlet 套件

一个或多个 MIDlet 程序及其相关资源的集合称为 MIDlet 套件 (MIDlet Suit)，它是在目标设备上安装、更新和删除 MIDP 应用程序的基本单位。MIDlet 套件通常被封装到一个 Java 档案文件 (Java Archive File，即 JAR 文件) 中。

它包括如下一系列文件：

- 实现 MIDlet 的类文件；
- MIDlet 中所有用到的资源文件（如图标文件，声音文件等）；
- 描述 JAR 文件内容的清单文件（Manifest File），后缀为 .mf 的文本文件。

此外，每一个 JAR 文件还配有一个用来描述 MIDlet 套件的描述文件，该文件以 .jad 为后缀（即 JAD 文件）。套件结构如图 1-5 所示。

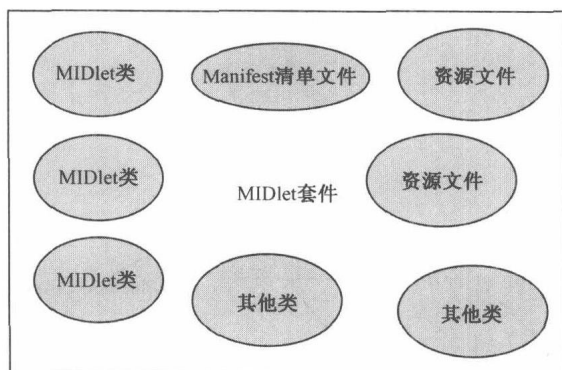


图 1-5 套件结构

1.5.1 清单文件

清单（Manifest）文件是文本格式的文件。当 MIDlet 套件安装到硬件设备上时，清单文件的扩展名会被更新为 .mf。它定义了 MIDlet 套件的属性，其中有 6 个属性是必须包含的。除此之外，用户还可以自定义一些属性。

Manifest（清单）文件包含的 6 个必要属性如下。

- MIDlet-Name MIDlet 套件的名称。
- MIDlet-Version MIDlet 套件的版本，格式为：a.b.c。
- MIDlet-Vendor MIDlet 套件的开发人员。
- MIDlet-<n> MIDlet 套件中每个 MIDlet 的描述，该描述可以分成三部分内容：名称、图标名和类名。其中，n 要用一个从 1 开始的数值替换。
- MicroEdition-Configuration MIDlet 套件所需要的配置的类型和版本。
- MicroEdition-Profile MIDlet 套件所需要的简表的类型和版本（如果出现多个版本，要用空格分离）。

以下是清单文件定义的可选属性。

- MIDlet-Icon MIDlet 套件的图标，必须是扩展名为 .png 的文件。
- MIDlet-Description MIDlet 套件的描述。
- MIDlet-Info-URL MIDlet 套件更多信息的 URL。
- MIDlet-Data-Size MIDlet 套件需要的最小持久存储空间的大小，以字节为单位。如果不存在此属性，则说明不需要持久存储。
- MIDlet-Permissions MIDlet 套件的权限许可列表。
- MIDlet-Permissions-Opt MIDlet 套件的可选权限许可列表。