高等学校计算机语言应用教程

J2ME 应用教程

陈旭东　徐保民　张宏勋　编著

清华大学出版社 · 北京交通大学出版社

高等学校计算机语言应用教程

J2ME 应用教程

陈旭东　徐保民　张宏勋　编著

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书详细介绍了有关 J2ME 的各种概念，并以浅显易懂的例子讲解如何开发 MIDP 应用程序。主要内容包括：高级用户界面和低级用户界面编程、记录存储、游戏开发、网络编程及多媒体应用开发等。

本书内容丰富、语言通俗易懂，注重理论与实践相结合，可作为高等院校计算机或相关专业学习 J2ME 技术的教材，也可以作为 J2ME 手机应用编程人员的参考用书。

图书在版编目（CIP）数据

J2ME 应用教程 / 陈旭东，徐保民，张宏勋编著. —北京：清华大学出版社；北京交通大学出版社，2007.1

（高等学校计算机语言应用教程）

ISBN 978-7-81082-898-7

Ⅰ. J…　Ⅱ. ①陈…　②徐…　③张…　Ⅲ. JAVA 语言-程序设计-高等学校-教材　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2006）第 126905 号

责任编辑：谭文芳

出版发行：清 华 大 学 出 版 社　邮编：100084　电话：010-62776969　http://www.tup.com.cn

北京交通大学出版社　邮编：100044　电话：010-51686414　http://press.bjtu.edu.cn

印 刷 者：北京东光印刷厂

经　　销：全国新华书店

开　　本：185×260　印张：14.75　字数：378 千字

版　　次：2007 年 1 月第 1 版　2007 年 1 月第 1 次印刷

书　　号：ISBN 978-7-81082-898-7/TP・310

印　　数：0 001～5 000 册　定价：23.00 元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。

投诉电话：010-51686043，51686008；传真：010-62225406；E-mail：press@center.bjtu.edu.cn。

前　言

在计算机编程技术发展史上，Java 技术是接受最快、普及最快的一种技术。在问世后短短十余年内，Java 技术就以其独特的魅力征服了几乎所有的程序员。其中，J2ME 是 Java 技术大家族里不容被忽视的成员之一。目前，在众多的嵌入式和移动应用开发者眼里，J2ME 已经成为他们的首选工具。

本书依据 MIDP 2.0（JSR 118）和 MMAPI 1.1（JSR 135）规范，利用 JDK 5.0 和 WTK 2.2 作为开发与调试平台，结合实际应用，采用大量实例，全面介绍 J2ME 相关的应用开发技术。

全书共分 7 章。

第 1 章主要介绍 CLDC 和 MIDP 规范、MIDlet 的生命周期、J2ME 应用程序开发平台搭建和 MIDlet 应用程序模型及它的工作机理。

第 2 章主要介绍构成 MIDP 高级用户界面的常见组件 Form、Screen 等的使用方法。

第 3 章主要介绍开发 MIDP 低级用户界面所用到的类及它们所提供的常用方法。

第 4 章主要介绍 MIDP 规范所定义的记录管理系统（RMS）的基本操作、记录的增删改、记录库变化的监视及记录的查询与排序。

第 5 章主要介绍 MIDP 中网络编程的基本原理，并结合示例重点讲解了与 HTTP、Socket 及 SMS 编程相关的类和接口的使用。

第 6 章主要介绍 MIDP 规范中有关多媒体应用开发方面的知识，包括单音、音序、MIDI、音频、视频的播放，以及录音、照相、录像功能的实现。

第 7 章主要对 MIDP 中与游戏编程相关的类进行介绍，并对这些类提供的方法结合示例做了较为详细的讨论。

本书在编写时，强调应用，体现案例教学，特别适合读者自主学习。通过本书的学习，可以快速掌握基于 MIDP 2.0 的 J2ME 应用开发技术。书中提供的所有实例程序全部通过了 WTK 2.2 上的调试与运行。为了方便本书读者的学习，全部程序源代码可以从北京交通大学出版社的网站下载：http://press.bjtu.edu.cn。

参与本书编写的作者，多年来从事 Java 应用的开发，积累了丰富的开发经验，书中的很多内容都是实践经验的总结。本书第 1、2、5 章由徐保民编写，第 3、7 章由张宏勋编写，第 4、6 章由陈旭东编写。全书由陈旭东负责统稿和定稿。

本书的出版得到了清华大学出版社、北京交通大学出版社及谭文芳老师的大力支持，北京交通大学计算机与信息技术学院、软件学院相关课程的老师为本书的编写提出了不少的建议，在此一并表示深深的谢意。

由于编者水平所限，难免会存在很多不足和错误，恳请各位读者不吝赐教，编者的电子邮箱为：xudong_chen@tom.com。

编　者

2007 年 1 月

目　录

第 1 章　J2ME 概述

在计算机编程技术发展史上，Java 技术是接受最快、并且普及最快的一种技术。在问世后短短十几年时间内，Java 技术就以其独特的魅力征服了几乎所有的程序员，尤其在年轻人眼里，掌握 Java 技术就等于可以拿到更高的“薪水”。

在 Java 技术的大家族里，J2ME（Java 2 Micro Edition，Java 2 微型版）是不容被忽视的成员之一。特别值得一提的是，目前，在众多的嵌入式及移动应用开发者眼里，J2ME 已经成为他们的首选工具（Linux 和 WinCE 分别名列第二、第三位）。

本章首先介绍 Java 和 J2ME 相关的背景知识、设计目标及规范等相关知识，然后较详细地介绍 MIDlet 应用程序模型，最后介绍几种常见的 J2ME 开发环境的使用。

1.1　Java 平台和 J2ME 技术概况

平台是指程序在其中运行的硬件或软件环境。Java 平台与现存的大多数平台的不同之处在于，它是一种能运行在其他硬件平台上的纯软件平台。

众所周知，Java 平台是由 Java 虚拟机（Java Virtual Machine，JVM）和 Java 应用程序编程接口（Java Application Programming Interface，Java API）两部分构成。其中 Java 虚拟机是 Java 平台的基础。

至今为止，Sun Microsystems 公司已成功地将 Java 虚拟机移植到台式计算机、服务器及移动设备等各种硬件平台上，并逐步形成一个性能不断提高、应用领域不断得到拓展的、以 Java 语言为核心的，包括 J2SE/J2EE/J2ME/JavaCard 和一系列的标准、技术及协议的 Java 技术大家族。

1.1.1　Java 语言的发展和现状

Java 语言的开发始于 1991 年，它来自于美国 Sun 公司的由 Patrick Naughton，Mike Sheridan 和 James Gosling 领导的一个名为“绿色”（Green）的项目，其前身是“Green”项目组的技术人员为家用消费类电子产品，例如交互式电视和电冰箱等，开发的一种智能化家电语言 Oak（橡树）。Oak 是一种面向对象的“可移植”的新的语言。在执行前，生成一个“中间代码”，在任何一种家电设备上只要安装了特定的解释器，就可以运行这个“中间代码”。Sun 公司曾经用这种语言的技术参与一个交互式电视项目的招标，但结果却被 SGI 公司打败。随后一段时间内，Oak 基本无任何发展。

随着 1993 年 WWW 的出现，以及 Mosaic（Spyglass 公司）和 Netscape（Netscape 公司）浏览器的先后问世，“Green”项目组成员对 Oak 进行了大规模的改造，推出了新一代的面向对象的程序设计语言 Java。同时该项目组成员利用 Java 语言及时地开发出了一个浏览器 HotJava，并获得了极大的成功。从此以后，Java 语言的开发逐步流行起来，并从一种程序设计语言逐步上升为一种技术。

1996 年，Sun 公司成立 Javasoft 分公司，专门负责 Java 事宜。同时，正式发表 Java 1.0 版。其实 Java 1.0 并不适合应用程序的开发，它甚至不支持打印功能。

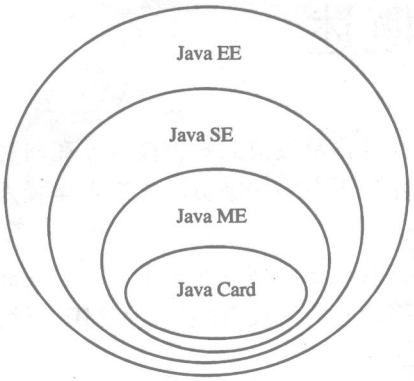

图 1-1 Java 平台的划分

直到 1998 年 Java 1.2 版本的出现，Java 才真正成为现代开发工具中的利器。

时至今日，Java 的最新版本是 Java 6.0（或 JDK 6）。按照对函数库的支持程度来分，Java 6.0 家族已经是拥有 Java EE(Java Platform Enterprise Edition，Java 企业版)，Java SE（Java Platform Standard Edition，Java 标准版），Java ME 和 Java Card（Java 卡）四个各具特色成员的大家庭了，如图 1-1 所示。如果从纵向的角度来看所有的 Java 平台的划分，则如图 1-2 所示。

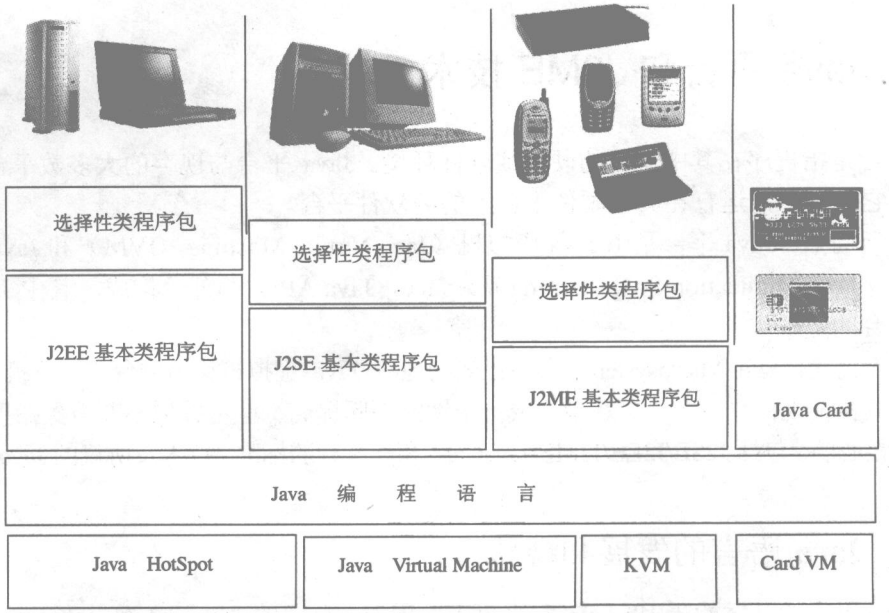

图 1-2 从纵向的角度来看 Java 平台的划分

Java EE 主要用于开发企业服务端应用程序。其类库除了 Java 语言的所有特性外还包含针对企业计算的各种编程接口和规范，如 JSP （JavaServer Pages）、RMI（Remote Method Invoke）、EJB（Enterprise JavaBeans）、JNDI（Java Naming and Directory Interface）、JMS（Java Message Service），JIDL（Java Interface Definition Language）、JTA（Java Transaction API）、JavaBean 等。Java EE 针对的设备主要是后端的服务器等。

Java SE 主要用于开发常见的桌面应用程序。其类库包含了 Java 语言的所有特性即基本 Java 的核心应用编程接口，它是标准的开发工具包。Java SE 针对的设备主要是台式机等。

Java ME 主要用于开发信息家电应用程序。由于受设备内存和处理器的限制，其类库比较小，在对 Java SE 的类库作了一些剪裁的基础上增加了若干新的特性。Java ME 针对的设备主要是嵌入式和消费类的设备，如移动电话。

Java Card 主要是针对智能芯片和 IC 卡等设备而定制的最小 Java 子集。它甚至比 Java ME 还要小，只支持 java.lang.*这个核心类库及 boolean 和 byte 两种基本数据类型。它对资源需求极小，可运行在无图形用户接口和网络的设备上。

从图 1-2 可以看出 ，要贯彻所谓的“一次编写，到处运行”（Write once, run anywhere）的目标，不管是哪种开发平台都必须包括虚拟机、基本的类程序包及应用所需的类程序包。

在后续章节中，统一使用 J2ME 来描述 Java ME。

1.1.2　J2ME 概述

Java 语言本来是为消费类电子产品设计的，特别是交互电视市场，然而由于当时的技术条件所限，Java 无法在消费类电子产品的应用领域内广泛应用，却在网络应用中得到发展，后来又逐渐演变为越来越适合嵌入式设备、桌面和企业计算的要求。就目前而言，J2ME 作为对嵌入式设备和智能家电应用领域的最佳解决方案已经成为不争的事实。

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”。显然，将 J2ME 等同于手机程序开发，认为 J2ME 是一个标准，一个规范，都是不正确的说法。

从技术的角度讲，J2ME 实际上是由一系列规范，即由 JCP（Java Community Process，Java 社团）组织制定的有关 JSR（Java Specification Request，Java 规范）文档构成的。例如 MIDP（Mobile Information Devices Profile，移动信息设备简表）2.0 规范就是在 JSR118 中制定的。

由 Sun 公司给出的 J2ME 定义可以看出，J2ME 所面对的是成千上万种不同类型的移动终端设备，而且它们之间的计算能力和存储容量等性能指标之间存在着极大的差异，导致不可能像桌面系统那样仅仅提供几个版本的 JVM 即可满足 Windows，Linux 和 UNIX 等系统。

因此，为了满足千差万别的移动设备的需求，J2ME 的基本类程序包又划分为两大类，如图 1-3 所示。CDC（Connected Device Configuration，面向连接的设备配置）适用于高端信息家电，如电视机顶盒、网络可视电话等，所用的 Java 虚拟机为 CVM（C Virtual Machine，C 虚拟机）。字母“C”代表 Compact、Connected、Consumer。CLDC（Connected, Limited Device

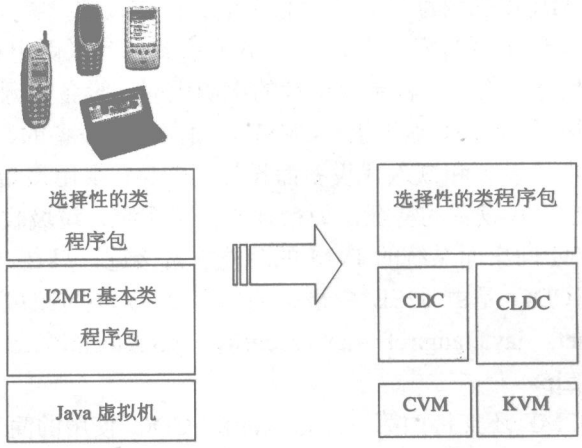

图 1-3　J2ME 的基本类程序包

Configuration，面向连接受限的设备配置）适用于低端信息家电，如手机、个人数字助理（Personal Digital Assistant，PDA）等，所用的 Java 虚拟机为 KVM（Kauai Virtual Machine，K 虚拟机）。从包含关系来讲，K 虚拟机是 CVM 的一个子集。

显然，CDC 和 CLDC 只是对各类信息家电中最具共性的内容进行抽象，形成适合于某个范畴中设备可用的规范即配置（Configuration）。所谓配置就是由一组核心类库和一个运行在特定类型设备上的虚拟机组成，它为应用程序提供了运行基础。配置是针对设备软硬件环境严格定义的，例如 CLDC1.0 定义了内存大小为 64～512 KB，任何设备如果支持 CLDC1.0，就必须严格满足定义，不能有可选的或者含糊的功能。

不同应用场合的信息家电，事实上还是有很大差异的。所以，Sun 公司的做法是在共同的 CDC、CLDC 配置上，再添加适用特定规格的配置文件，即将某一个行业或领域内设备的特性提取出来，形成相关的配置文件即简表（Profile）。显然简表不是通用的东西，是针对某一类设备所制定的规范和 API，它们只在某些设备上可用。与配置相比，简表更多的是针对软件接口的定义，简表有必须实现的，也有可选的功能，因此，简表更灵活。目前，比较实用的可选简表有 JSR135 定义的 MMAPI（Mobile Media API），实现多媒体播放功能；JSR184 定义的 M3G API（Mobile 3D Graphics API），实现 3D 功能；JSR120 定义的 WMA（Wireless Messaging API），实现短消息收发。

不同的虚拟机、配置和简表之间的关系如图 1-4 所示。

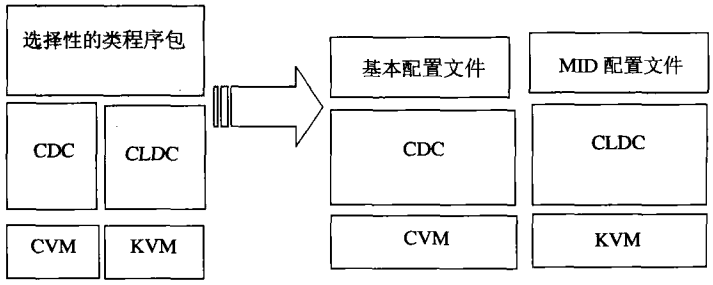

图 1-4 J2ME 的配置及配置文件间的关系

CDC 是应用在 2 MB 以上内存的高端信息家电。目前，这一类设备通常是共享的、固定的网络连接信息设备，像电视机机顶盒，网络电视系统、互联网电话与汽车导航/娱乐系统等。这些设备的特点是有线连接，稳定而持续的电源供应，设备资源比较受限。

CDC 配置下应用的是 CVM 虚拟机，CVM 是面向消费设备的、工作于连接环境中的虚拟机，专门为消费类电子设备和嵌入式设备而设计。CVM 采用最新的虚拟机技术，占用空间较小而保持了适合嵌入式设备的特征，如精确的存储系统、垃圾收集、快速 Java 同步等。

从本质上讲，CVM 同桌面系统的 JVM 的功能非常接近，只是可以使用的类包数量大大少于 J2SE 的包，即 CDC 是由 J2SE 中最小的如下 Java 类包组成的：java.io，java.lang，java.lang.math，java.net，java.lang.ref，java.security，java.security.cert，java.text，java.util，java.util.jar 和 java.util.zip。

需要指出的是：J2ME 还支持由第三方定义的供 CDC 使用的简表。

目前，在消费性产品的应用开发领域，由于硬件和网络的一些限制，CLDC 和 CDC 是市场上支持 J2ME 的消费性产品开发平台中使用最多的两个规范，本书将重点介绍 CLDC、

MIDP 及有关编程技术。

1.1.3　KVM

由图 1-4 可知，用于 CLDC 配置的虚拟机是 KVM，其创始人是 Antero Taivalsaari。关于字母“K”有两种解释，一种解释是 KVM 是按照 K 字节衡量的而不是兆字节；另一种解释是 K 来自一个项目的代号“Kauai”。

Sun 公司开发 KVM 的主要目的是为使用 16/32 位 RISC/CISC 微处理器或控制器且其可用内存为 160～512 KB 的设备而开发的一个依赖于目标平台的虚拟机。考虑到使用 KVM 的设备一般只具有有限的内存空间和处理能力的事实，KVM 采用 C 语言编写。此外，KVM 是模块化的，也就是说，它是由模块构建的，当某个模块实现了预先设定的目标后，就可以很容易地把该模块加载到系统内。

KVM 的本地界面以轻便性为原则构建，所以在 KVM 中任务切换不依赖硬件产生的记时器中断，即任务的切换不是抢先式，它通常发生在虚拟机执行了一个预设编号的字节码之后。另外，KVM 的垃圾收集是用一个标记清扫算法来实现的。因此，对象引用是直接的，就像标准 Java 一样。

当然，除了虚拟机以外还有许多可用的执行环境，在小型设备中，虚拟机必须要么被扩展，要么在附加工具协助下提供一个更加完整的运行环境，正是这个原因，KVM 需要一些附带的工具，例如 JavaCodeCompact 工具提供了预链接和预加载类，允许 Java 类被直接地链接进虚拟机中。

KVM 常用的一个可选的附件是 Java 应用程序管理器（Java Application Manager，JAM），它负责 CLDC 设备上的 J2ME 应用程序的下载、安装、更新和删除。如果更新过程失败，JAM 甚至可以重新使用旧的应用程序。JAM 是由移动设备本身所提供的，不同公司的实现可能会存在部分差异。

1.1.4　CLDC

CLDC 是为使用资源受限的、连接受限的较小设备制定的一个 Java 应用开发规范。CLDC 规范的早期版本是由 JCP 于 2000 年推出的 CLDC1.0（即 JSR30）。该规范版本规定能运行 CLDC1.0 的设备的典型特征是具有一个 16 位或 32 位的微处理器和用来支持虚拟机和类库的 160～512 KB 之间的消费的总内存。这一类别中的典型设备包含手机、PDA 等。需要指出的是：早期的 Java 手机大部分都支持 CLDC1.0，如 Nokia 3650，Siemens 6688i。

与 J2SE 相比，CLDC1.0 的类库仅保留了 Java 规范中定义的最核心的三个包：java.io、java.lang 和 java.util，并重新定义了一个用于支持通用连接框架（Generic Connection Framework，GCF）的新包 javax.microedition。

CLDC1.0 所定义的三个核心包的内容与 Java 规范中所定义的三个核心包的内容并不完全等价。CLDC1.0 对 Java 规范中所定义的三个核心的包的内容进行了一些裁减，仅保留了小型移动设备可能用到的一些类、方法及属性。例如 java.util 的类与接口由 J2SE 的 47 个缩减到 10 个。

CLDC1.0 引入 javax.microedition 包的主要目的是替代 J2SE 中 java.net 包，解决 J2ME 应用的联网问题。javax.microedition 包定义了 GCF 中的类和接口，其中最关键的是 Connector 类和 Connection 接口。Connector 类中定义了一组静态方法用来生成特定类型的 Connection，然后利用该 Connection 就可以连接网络和各种其他设备。关于 GCF 的使用将在第 5 章详细介绍。

随着半导体元器件和小型移动设备制造技术的发展，小型移动设备的内存空间和处理能力都得到了显著的改善，导致 CLDC1.0 规范的某些规定有点落伍。于是 JCP 于 2003 年推出 CLDC1.1（即 JSR139）。

与 CLDC1.0 相比，CLDC1.1 并没有发生本质的变化。CLDC1.1 仅在硬件的兼容性和可用性上作了一些改进，并增加了一些新的特性。例如 CLDC1.0 只支持整数运算，不支持浮点运算，而 CLDC1.1 则增加了浮点运算，因此，在支持 CLDC1.1 的设备上，可以使用实型数据类型（包括 float 和 double 类型）的变量。现在新出 Java 手机大部分都能支持 CLDC1.1，如 Nokia 9500，Siemens S65。

1.1.5 MIDP

如图 1-5 所示，MID 简表与 CLDC 配合使用，就能构造起一个供 J2ME 应用程序运行所需的环境。

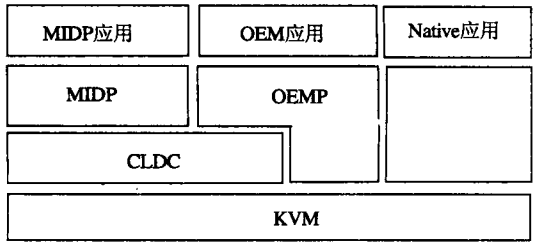

图 1-5 支持 J2ME 平台架构图

通常，J2ME 平台由一个 CLDC 和一个或者多个 MID 简表构成。目前，J2ME 支持的 MID 简表包括以下两种。

- MIDP 简表：该简表所定义的 API 是 MIDP 规范所要定义的 API。
- OEMP 简表：MIDP 规范所涉及的无线通信设备多种多样，因此它不可能满足所有设备的需求。因此该简表所定义的 API 是由 OEM 厂商提供的以便访问特定设备的特定功能。但基于这些 API 的应用可能无法在其他的设备上运行。

建立在 MID 简表之上的是 MID 应用，目前，J2ME 支持的 MIDP 应用包括以下三种。

- MIDP：一个 MIDP 应用程序称为 MIDlet，它只能使用 MIDP 和 CLDC 规范中所定义的 API。MIDlet 程序有点类似于常见的 Applet 或 Servlet 程序。
- OEMP：一个 OEMP 应用会使用一些不在 MIDP 中定义的 API（即 OEMP 简表中所定义的 API）。
- Native：一个本地应用是指使用非 Java 语言编写的直接运行于设备操作系统之上的应用。

支持 J2ME 设备平台上的所有应用的层次图，如图 1-5 所示。

由于 MIDP 规范中不涉及 OEM 和 Native 的应用。所以，后面所提到的 MID 应用和 MID 简表均是指 MIDP 简表和 MIDP 应用。

在 J2ME 中，MIDP 简表的内容包含在 MIDP 专家组（MIDP Expert Group）共同制定的 MIDP 规范中。具体来讲，MIDP 规范定义了能在 Java 手机上运行的 Java 程序的规范，

包括应用程序生命周期，各种用户界面组件，支持存储和网络连接等。换言之，MIDP 规范定义了移动信息设备的类型和提供相关的 API 集合，MIDP 所定义的功能更加面向用户，而且比 CLDC 更高级。

早期的 MIDP 规范是由 JCP 于 2000 年推出的 MIDP 1.0（即 JSR 37）。MIDP 1.0 规范中成功实施的功能包括：

- 应用程序下载；
- 应用程序的生命周期；
- 端到端的传输；
- 网络连接；
- 数据库存储；
- 计时器；
- 用户界面。

显然，利用 MIDP 1.0，就能在手机上运行带有用户界面的 Java 程序。

通过用户对 MIDP 1.0 的使用经验和反馈意见，JCP 于 2002 年推出了 MIDP 2.0（即 JSR 118），MIDP 2.0 规范是在 MIDP 1.0 规范的基础上设计的，它保证了同 MIDP 1.0 的兼容性，即在 MIDP 1.0 上编写的 J2ME 应用程序能在 MIDP 2.0 上执行。

MIDP 2.0 的主要新功能包括以下几个方面。

- 用户界面的强化：不仅能够以更强的交互方式轻松地使用，而且全面地改进了易用性。提高了画面尺寸不同的各种终端之间的移植性，可根据需要灵活调整显示尺寸。
- 支持各种媒体：由于能够完全使用终端所具有的音频和视频功能，因此在 MIDP 中可以使用来信提示音和播放音频和视频。
- 支持游戏：通过新增游戏 API，可以充分利用终端产品所具有的图形功能。简化了游戏开发，图形和性能的控制也变得更加容易。
- 连接性的强化：除了标准的 HTTP 以外，还支持 HTTPS、Datagram、Socket、Server Socket 以及串行端口连接。
- 压入架构：支持来自服务器的信息压入发送（Push）。注册 J2ME 应用程序后，终端在接受来自服务器的信息时就可以实现自动化。
- OTA（Over-The-Air）应用程序下载：它使得用户能够动态地部署和更新移动设备上的应用程序；同时，提供应用程序下载的服务提供商还能够判断该 MIDlet 程序是否能够运行在申请下载的设备上，并且从设备上获取安装、更新和删除的信息。
- 端对端（End-to-End）安全性：除了 HTTPS 以外，还支持 SSL 和 WTLS，并且可以传输加密数据。

目前，MIDP 规范的图形界面基本上都是独立于 J2SE 的 AWT 和 Swing 组件，因为目前手机的计算能力还比较有限，但是，随着手机的 CPU 越来越快，使得 AWT 和 Swing 移植到手机上也将成为可能，因此，基于 CDC 规范的最新的 PBP 1.0（Personal Basic Profile）和 PP 1.0（Personal Profile）提供了部分 AWT 和 Swing 的支持，目前，部分高端 PDA 已经可以运行 PBP 和 PP 的 Java 程序了。可以预见，未来大部分的 AWT 和 Swing 组件都能移植到手机开发平台上。

1.2 第一个 MIDlet 程序

学习一门新的编程语言，传统的做法都是用著名的“Hello World”程序作为第一个例子，通过它来讲解用该语言编制的程序的基本构成。这里也用“你好!，J2ME”程序来说明 MIDlet 程序的基本构成，如例 1-1 所示。

例 1-1：一个基本的 MIDlet 程序
文件名：HelloWorld.java

```
//HelloWorld.java,一个最简单的 MIDlet 程序。

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloWorld extends MIDlet implements CommandListener {

private Command exitCommand;
private TextBox tb;

/**
*默认构造方法
*/
public HelloWorld(){
      exitCommand =new Command("退出",Command.EXIT,1);
      tb =new TextBox("你好!，J2ME ","你好!，J2ME ",15,0);
      tb.addCommand(exitCommand);
      tb.setCommandListener(this);
}

/**
*启动方法
*/
protected void startApp(){
      Display.getDisplay(this).setCurrent(tb);
}

/**
*暂停方法
*/
protected void pauseApp(){
   //语句
}

/**
```

```
*销毁方法
*/
protected void destroyApp(boolean u){
  //语句
}

/**
*事件响应方法
*/
public void commandAction(Command c,Displayable d){
    if (c ==exitCommand){
        destroyApp(false);
        notifyDestroyed();
    }
}
}
```

以上程序将会在 MIDP 设备的显示屏上显示“你好!，J2ME”和【退出】按钮，按【退出】按钮会终止应用程序。下面逐行分析该程序的源代码。

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
```

上述两行代码告诉编译器将使用 midlet 和 lcdui 两个类库包内定义的一些类。LCDUI（Limited Configuration Device UI，有限配置设备用户界面）包是一个针对移动设备而最佳化的用户界面包。如同在开发 Java 程序时一样，只要用到用户界面，程序就会引入 AWT 或 Swing 包一样。LCDUI 既包含高级 UI 类（例如 Form、Command、DateField 和 TextField 等），又包含低级 UI 类（允许用低级方式控制 UI）。MIDP 规范中规定所有 MIDlet 程序都必须继承 javax.microedition.midlet.MIDlet。因此所有的 MIDlet 程序都必须引入该类库包。其中 MIDlet 类定义有允许应用管理软件创建、运行、停止和销毁应用程序的一些方法。这有点类似 J2SE 中的 Applet 程序，即用户编写的 Applet 程序都必须继承 java.applet.Applet。

```
public class HelloWorld extends MIDlet implements CommandListener
```

上述一行代码定义了一个新类 HelloWorld，并告诉编译器它是 MIDlet 类的子类，同时该类也将实现 CommandListener 接口内定义的抽象方法，以便能处理反映用户动作的事件。

MIDlet 类是一个抽象类，该类声明了 startApp()、destroyApp()、pauseApp()等抽象方法，这些方法的作用分别类似于 Applet 类中的 start()、stop()和 destroy()方法。

```
private Command exitCommand;
private TextBox tb;
```

上述两行代码定义两个私有对象，tb 是容器类型 TextBox 的对象，这个程序在 TextBox 容器中显示文本“你好!，J2ME”。exitCommand 是命令类型 Command 的对象。Command 类定义了所有按键动作和组件动作，它必须与 CommandListener 接口配合使用才能截获并且