移动增值业务开发丛书

J2ME

技术开发与应用

李研　刘晶晶　俞一鸣　编著

机械工业出版社
CHINA MACHINE PRESS

移动增值业务开发丛书

J2ME 技术开发与应用

李　研　刘晶晶　俞一鸣　编著

机 械 工 业 出 版 社

本书详细讲解了J2ME开发中的用户界面、记录存储、媒体播放、联网、3D特效等技术，并且还推出三个富有代表性的实例，以不同的开发程序的方式介绍了程序开发流程，读者可以根据不同的需要选择不同的流程，从而使本书更加具有针对性和实用性。

本书内容全面、易于理解、实例众多。为读者更好地使用这项技术和标准进行工作提供了很好的指导。书中既包含了简单易懂的代码片断，也有大量实际可用的应用实例，并包括大量商业级的源程序。读者可以迅速掌握J2ME的核心API类库以及无线应用系统的开发过程，是从事无线应用系统开发人员的优秀参考书籍。

希望通过本书的学习，读者不但能够掌握J2ME的基本概念和基本操作知识，理解J2ME平台的设计理念，同时通过本书中应用实例的介绍，快速全面掌握J2ME应用开发流程，从而能够完成具体的实际工作。

图书在版编目(CIP)数据

J2ME技术开发与应用/李研，刘晶晶，俞一鸣编著.
—北京：机械工业出版社，2006.4
(移动增值业务开发丛书)
ISBN 7-111-18834-9

Ⅰ.J... Ⅱ.①李...②刘...③俞... 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.TN929.5

中国版本图书馆CIP数据核字(2006)第030462号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
策划编辑：张俊红 责任编辑：赵玲丽 版式设计：霍永明
责任校对：刘志文 封面设计：陈 沛 责任印制：李 妍

北京铭成印刷有限公司印刷

2006年5月第1版第1次印刷
184mm×260mm·24.75印张·615千字
0001—4000册
定价：39.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话(010)68326294
编辑热线电话(010)88379768
封面无防伪标均为盗版

前　言

信息产业部最新统计数据显示，截至2004年底，我国国内手机用户已超过3亿，且以高于20%的速度持续增长。随着3G通信、智能手机等新的移动技术的发展，各类增值服务层出不穷。同时，各种嵌入式移动终端也在交通运输、生产调度、电子政务、实时数据采集等企业/政务级应用领域显示出巨大的前景。因此，无论是在消费领域还是在企业级应用领域，移动软件开发技术将迎来空前的发展机遇。

从国内来看，作为全球最大的移动通信市场，手机游戏无疑拥有广阔的市场前景。目前，在中国有8000多万计算机网络用户，而手机用户数却已经达到3.4亿，手机用户数量是计算机网络用户的数倍。如果8%~10%的手机用户使用手机游戏业务的话，我国的手机游戏市场将拥有3000万用户群。业界人士算了这样一笔账：假设这3000万用户每人每月只下载一个游戏，每下载一个游戏收费5元，那么一个月的业务收入就可达1.5亿元，一年的业务收入会超过15亿元。与PC游戏不同，手机游戏摆脱了线缆的束缚，具有随时、随地、随身的特点，更适合人们在移动中休闲和娱乐。显然，手机游戏产业一旦启动，其能量将不亚于目前的计算机网络游戏。

为了帮助众多开发人员和爱好者进入移动开发领域和提高程序开发水平，笔者精心编著了本书。本书依照读者的学习规律，首先介绍基本概念和基本操作，在读者掌握了这些基本概念和基本操作的基础上，再对实际应用开发进行深入的讲解，严格遵循由浅入深、循序渐进的原则。本书期望覆盖尽量多的手机开发中出现的问题，如果读者已经具备了J2ME开发的基本技术，也可以依照需要，挑选一些章节作为重点阅读内容。

本书共包含12章。

第1章首先带领读者了解J2ME的应用领域，了解Java的三个组成部分以及各自的应用领域，讲解MIDP和CLDC的概念，以及MIDP2.0的新增功能。随即切入正题，介绍了MIDlet套件的概念和如何使用WTK来进行程序的编译、打包、混淆和运行等基本操作。

第2章主要介绍了Eclipse和JBuilder集成开发环境的配置和使用，以及如何顺利使用向导生成需要的工程。

第3章是高级用户界面部分。高级用户界面具有良好的可移植性，本章主要讲述了四个屏幕类和表单(Form)类的八个组件的使用。其中Spacer和CustonItem组件是MIDP2.0新增的。

第4章是低级用户界面部分。低级用户界面常常被用来开发手机游戏，本章主要讲述了画布Canvas类、图形绘制Graphics类、字体Font类和图像Image类。

第5章是数据持久性存储部分。主要讲述了MIDP的记录存储系统，以及在实际的应用开发中的使用方式。

第6章是网络编程的基础部分。主要讲述了应用开发中常用的三种协议以及使用方式，并且对通用连接框架做出了详细的说明。

第7章是声音处理部分。主要说明了J2ME中声音的处理方式，以及各种播放方式的说

明，并给出了一种实际开发中使用的声音播放方式。

第 8 章讲解了如何使用 MIDP2.0 新增的 GameAPI。主要讲述了如何使用 GameAPI 创建游戏精灵、铺设游戏地图，以及如何进行图层管理。

第 9 章是 3D 游戏开发的入门部分。本章内容涵盖了如何用点和三角面创建 3D 物体，如何设置物体的外观属性，如何使用摄影机和灯光，如何使用物体的动画，包括骨骼动画、变形动画和轨迹动画，以及如何进行场景的创建和绘制。

第 10 章讲解了一个完整的单屏游戏。本章讲解了一个用 MIDP2.0 实现的炸弹王游戏，该游戏实现了包括闪屏、菜单、声音、游戏动画和高分纪录等游戏所需要具备的基本功能。

第 11 章主要说明了一个大场景动作类游戏的实现。详细讲述了游戏开发过程中使用的基本技术，包括低级界面实现的菜单、状态机结构的游戏，以及面向对象的程序运行方式。并对游戏中经常使用的技术，包括历史记录保存、碰撞判断等都做出了详细的介绍。

第 12 章以全新的开发方式一步一步实现了一个网络聊天程序，并且使用 J2ME 技术实现了服务器。本章以全新的视角，将极限编程、设计模式、测试驱动、迭代开发融合到程序开发过程中，读者可以看到创新性的程序开发方式。

本书在内容的编排和目录组织上都十分讲究，争取让读者能够快速掌握 J2ME 开发的方方面面。本书章节命名明确清晰，读者可以立即知道每一节所要学习的知识。讲解具体知识时，在进行基本的使用方法讲解后，都会尽可能使用实例来进一步强化学习效果，这样保证读者的起步层次比较高，在阅读代码和自身实践的过程中不断提高水平。

在本书的写作过程中得到了李振鹏、龚剑、蒋亮等人的大力协助，再此表示诚挚的感谢。另外在本书的写作过程中，贺可香、姜海亭、赵海波、葛树涛、蒋建新、姜雪峰、刘磊、李晓凯、徐建卿、张学静、姜海燕、赵云霞、潘天保、苏雪华、刘小强、马文勃、彭红波、丘子隽、陈智强、江峰、赵纪凯、朱启东、张爱华等人给予了一定帮助，在此一并表示感谢。

限于作者水平，加上时间仓促，书中难免存在错误和不足之处，恳请广大读者和同行批评指正。

作　者

2006 年 3 月

目　录

第 1 章　J2ME 概论

何为 J2ME？如何开发 J2ME 应用？本章将详细讲述 J2ME 的体系结构，包括虚拟机、配置和简表。此外还将详细讲解 MIDP 应用程序的开发流程，以及 MIDP2.0 所推出的新安全结构。

1.1　J2ME 的基本概念

J2ME 是一个专门面向小型设备应用的“简易”平台，这里的小型设备包括移动电话、个人数字辅助设备(PDA)、网络电话、数字电视上的机顶盒、自动娱乐系统、导航系统、网络交换以及家用自动电器等。J2ME 的体系结构包括虚拟机、配置和简表三个部分，本书的内容都是针对 MIDP 简表应用开发的，主要为普通的移动电话设备。

1.1.1　Java 平台技术和应用现状

在过去几年中，Java 已经成长为一个全面而成熟的面向对象应用程序的开发平台，它适用于广泛的、异构的编程环境，而这些应用的涉及面很广，从企业级的服务器应用到传统的桌面应用，以及各种各样面向小型设备的嵌入式应用。

当 Java 平台发展到 Java 2 的版本以后，为了适应不同级别计算机硬件的开发需要，Java 平台形成了三个主要分支，如图 1-1 所示。

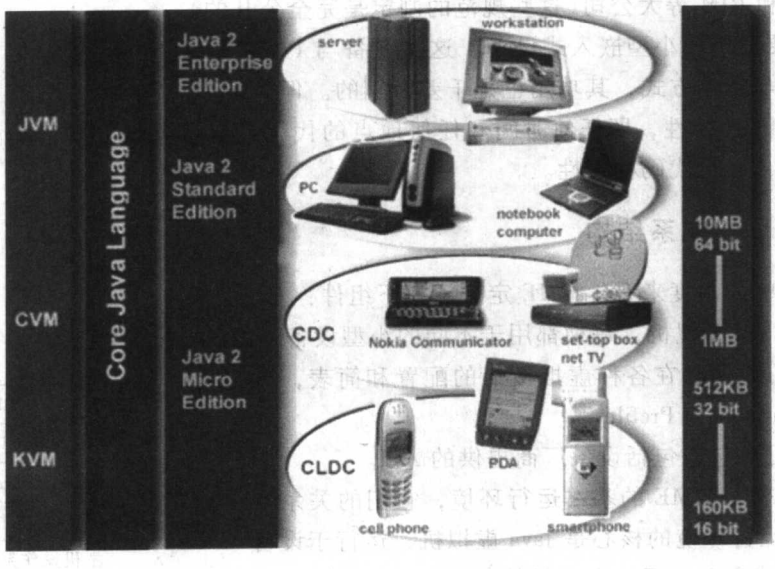

图 1-1　Java 的三个平台

1）Java 2Standard Edition(J2SE)：它是为台式计算机而设计的，主要运行在 Linux、Solaris 或 Microsoft Windows 等操作系统上。

2）Java 2 Enterprise Edition (J2EE)：它是一个适合分布式的、多用户、企业级应用系统运转的平台。它以 J2SE 为基础，增加了处理服务器端计算的功能。

3）Java 2 Micro Edition(J2ME)：和 J2SE 不同，它既不算是一个软件，也不能算是一则规范。准确地说，J2ME 是为了支持像 PDA、手机等小型的嵌入式或移动设备而推出的一系列的技术和规范的总称。它借用了 J2SE 类库的一部分，使用了更少的实际的数据包时间间隔(API)，而且其 J2ME 采用的 Java 虚拟机(JVM)比 J2SE 的 JVM 也要小得多。

1.1.2 J2ME 简介

Sun 公司将 J2ME(Java 2 Micro Edition,Java 2 微型版)定义为“一种以广泛的消费性产品为目标的、高度优化的 Java 运行环境”。自从 1999 年 6 月在 Java One Developer Conference 上声明之后，J2ME 进入了小型设备开发的行列。当时，由于分布式编程深受 Java 开发者团体的欢迎，所以大多数与会者都对 J2EE 的功能更感兴趣。然而，在随后的两年内，开发者意识到运行 Java 的小型组件同样具有很高的价值。

J2ME 通过 Java 的特性，遵循 J2ME 规范开发的 Java 程序可以运行在各种不同的小型设备上。Sun 公司希望借助 J2ME 这把利剑对嵌入式设备这个混乱的领域进行统一，让 Java 的范围扩展到所有的电子设备开发上，按照现在的发展速度，J2ME 很快将被广泛应用于消费和嵌入式设备中。

与以前 Sun 公司推出的 J2EE(Java 2 Enterprise Edition,Java 2 企业版)、J2SE(Java 2 Standard Edition,Java2 标准版)规范相比，J2ME 不是一个单独的技术规范，而是一系列技术规范总称。这些规范定义了 Java 技术在资源限制的设备中的表现形式，而且新规范在不断制定中，自从 J2ME 发布以来，已经有 600 多家公司加入了这方面的开发，包括 Palm、Nokia、Motorola 和 RIM 等大公司(这些规范的制定是完全公开的)。

J2ME 主要适用于小型嵌入式设备，这些设备与 PC 或是服务器设备相比没有统一的硬件标准、外观与操作方式，其功用也是千差万别的。但是 J2ME 在适用于这些设备的同时也保留了 Java 的传统特性，即任何时间和任何地点的代码具有可移植性、部署灵活性、安全的网络传输性，以及代码稳定性。

1.1.3 J2ME 的体系结构

从更高一些的角度来看，J2ME 定义了如下组件：

1）一组 Java 虚拟机，每种都用于不同的小型设备，满足不同的需求。

2）一组可以运行在各种虚拟机上的配置和简表，分别称为 Configuraion 和 Profile。

3）一些可选包，包括设备厂商提供的 API。

前两者组成了 J2ME 的基本运行环境，它们的关系如图 1-2 所示。运行环境的核心是 Java 虚拟机，运行于设备的主机操作系统之上，再往上是具体的 J2ME 配置，包括根据设备的资源需要提供基本功能的编程库。配置

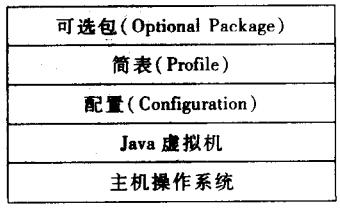

图 1-2 J2ME 运行环境的体系结构

(Configuration)的上面是一个或者多个J2ME Profile，这些附加的编程库利用相似设备的类似功能。在J2ME中还有一个重要的概念是可选包(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J2ME中制定了JSR82(Bluetooth API)提供了对蓝牙的支持。下面将按照体系结构来一一介绍每层的作用。

1.1.4 J2ME配置

应该注意到，J2ME要支持的硬件平台也有很大差异，其中有比较高端的设备，例如，电视机的机顶盒、网络电视等；也有比较低端的，像手机、寻呼机等。因此为了满足不同硬件的开发要求，J2ME规定了配置(Configuration)的概念，Configuration对不同级别的硬件在所使用的JVM和基础API集合方面做了规定。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)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连接设备配置(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。

2) CDC用于要求内存超过2MB的32位体系结构。互联网电视机机顶盒便是这类设备的一个示例。

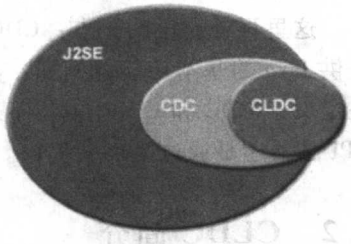

图1-3 J2SE、CDC和CLDC之间的关系

图1-3描述了CDC和CLDC之间的关系，同时该图也揭示了它们与整个J2SE API的关系。正如前面所说，CDC是加上一些额外类的J2SE的子集。同样，也可以看到CLDC是CDC的子集。

1.1.5 J2ME虚拟机

Java程序是以边解释边执行的方式运行的，所有的Java程序都会被编译成为二进制代码，并运行在Java虚拟机上。J2ME针对的设备主要是嵌入式和消费类的设备，因为这些设备内存和处理器的限制，所以J2ME所包含的类库也比较小一些，相对于J2SE的类库来说，作了一些剪裁，虚拟机的功能也相对简单。

前面提到过，CLDC和CDC配置各自定义了一组需要Java虚拟机支持的功能，因此每种配置都需要自己的Java虚拟机(J2SE中的Java虚拟机简称为JVM)。由于支持的功能比较少，CLDC虚拟机比CDC需要的虚拟机小得多。CLDC使用的虚拟机称为KVM(K Virtual Machine,K虚拟机)，CDC使用的虚拟机称为CVM(C Virtual Machine,C虚拟机)。KVM和CVM均可被看作是一种Java虚拟机，是JVM的子集，在功能上都是JVM的缩减版。这两类虚拟机的适用范围并不相同，简单地说，CVM的功能比KVM功能更为强大。

KVM是“Kilo Virtual Machine”的缩写。它是符合Java虚拟机规范的真正虚拟机，KVM专门为资源受到限制的小型设备设计，它们的内存只有几百千字节。

最初CVM是“Compact Virtual Machine”的缩写，然而Sun公司的工程师意识到商家可能混淆“Compact”的发音和KVM中的“K”，因此现在C不代表任何意思。该虚拟机用于

较多的消费者和嵌入式设备，例如 CDC 设备。在安全性、弱引用、JNI(Java 本地接口)和 RMI(远程方法调用)方面，CVM 支持 J2SE 的几乎全部功能。

1.1.6 J2ME 简表

简表为相同消费电子设备的不同生产厂商提供了标准化的 Java 类库。简表的实现是 Java 应用程序接口的一个集合，用于适应被定义配置的应用程序接口提供的服务，简表是一个完整的运行环境，一个在简表上执行的应用程序不需要额外的支持类。

事实上，虽然配置的开发由 Sun 公司领导，但是许多简表规范仍将继续由特殊设备的供应商领导。比如说，Motorola 公司领导了行动电话和呼叫器简表规范的开发，又如 Palm 公司领导 PDA 简表的开发。

目前 J2ME 领域里使用最广泛的是移动信息设备简表(MIDP)，它主要是针对手机和其他双向移动通信设备而设计的，本书在后面要重点讲解的也是这一简表。

这里再补充说明一点：CDC 规范和 CLDC 规范中也都定义了基本的 API 集合，这些 API 提供 Java 的基本功能，例如：java. io、java. lang、java. util、javax. microedition. io，这些包就是定义于 CDC 和 CLDC 中。不过这些包都是 Java 最基本的功能，更多的面向设备的功能性 API 必须还是通过简表来提供。

1.2 CLDC 简介

CLDC(连接有限设备配置)为资源有限的小型设备定义了一种标准的、内存占用最小的 Java 平台。前面提到过，CLDC 是 Java 的最小版本，适用于大量设备。然而，特定于某种垂直市场(例如手机和蓝牙等)的功能不包括在 CLDC 中，而是由它上层的简表定义。由于上述原因导致 CLDC 具有一个非常重要的特点：没有可选功能。CLDC 提供的任何功能都可以在支持设备上使用。毕竟 CLDC 的主要目标是确保在各种资源有限的设备上运行的应用程序具有可移植性和互操作性，而它们正是 Java 编程的主要目标。

1.2.1 特征设备

CLDC 起源可以追溯到 1999 年 JavaOne 大会上介绍的 Sun 公司的第一个袖珍版 Java 和第一个 KVM 以及相关的类库，虽然 CLDC 和所有的配置都满足成为虚拟机的条件，可它本身还不是虚拟机，CLDC 的引用实现只是包含在当前的分布中的 KVM。

根据 CLDC1. 0 规范中所说，运行 CLDC 的设备应该有以下特征：

1) 至少有 192KB 的内存空间。

2) 16/32 位处理器。

3) 一个有限的电源供给(通常是使用电池)。

4) 有限的或断断续续的网络连接性(9600bit/s 或更少)。

5) 多样化的用户界面甚至没有用户界面。

满足以上特征的一些有限连接设备如图 1-4 所示。

根据以上列出的限制，CLDC 目前为相应设备提供了如下功能：

1) Java 语言和虚拟机功能的子集；

图 1-4 有限连接设备的基本特征

2）Java 核心库的子集(java. lang 和 java. util)；

3）基本的输入/输出(java. io)；

4）基本的网络支持(java. microedition. io)；

5）安全性。

> 注意：CLDC 不包括应用程序的生命周期管理、用户界面、事件处理或用户与程序之间的交互，这些功能由简表来实现。

1.2.2 CLDC 的预审核机制

在移动信息设备中，Java 虚拟机的一个关键要求就是底层虚拟机的安全性。一个运行在虚拟机之上的应用，决不能对所在设备或者虚拟机本身造成伤害。在一个标准的 Java 虚拟机实现里，这种限制是通过类文件检查器来保证的。这个检查确保字节代码和存储在类文件里的其他项不包含非法的指令、不以非法的顺序执行，而且不包含对 Java 对象内存(对象栈)之外的无效内存地址或者内存区域的引用。一般来讲，类文件检查器的角色是保证加载进虚拟机中的类文件不能以虚拟机规范允许范围之外的方式执行。

类文件检查器的操作分为两个阶段，如图 1-5 引用于此所示：

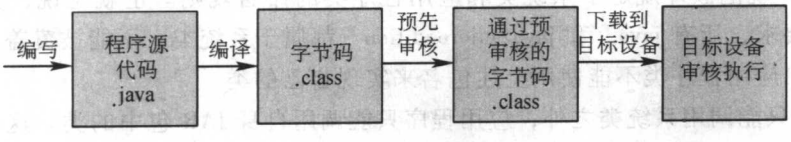

图 1-5 CLDC/KVM 中的类文件验证

1）类文件必须通过一个预检查工具，类文件必须通过一次预检查，该阶段会增加一些额外属性用以加速运行时的检查。预检查阶段一般在开发工作站上被执行，开发者用这种工具来编写和编译应用。

2）在运行时，虚拟机在运行时检查器组件使用由检查器生成的附加属性进行真正的、

高效的类文件检查。

在这种安全检查机制下，无效的类文件将在开发阶段就被摒弃，通过检查的类不能违反 Java 虚拟机的类型系统。经过预处理的带有额外属性的类文件大约比原始类文件大 5%~10%。

> 注意：类检查机制与代码签名的实现不同，这种保证并不基于认证或者属性检查。后者属于应用级安全检查。

> 说明：运行时的 CLDC 审核机制把它交给了设备自己去实现。设备可以根据自身的需要在加载类或是安装应用程序的过程中执行。在运行时，虚拟机迅速地对字节码进行线性扫描，将每个有效的指令与合适的堆栈映射项相匹配。运行时的审核过程是建立在预审核机制之上的，所以比预审核还要快，占用的动态内存更少。

1.2.3 CLDC 的安全机制

CLDC 的安全机制要比 J2SE 更加严格，安全机制主要涉及两个方面：

1）虚拟机级别的安全：一定要防止 KVM 执行的应用程序损害运行它的设备。这一点由类验证程序来保证，该程序确保类的字节码不包含对无效内存地址的引用，并且确保被加载的类以符合 Java 虚拟机规范的方式执行。CLDC/KVM 使用的类验证包含两个步骤：设备外的预验证和最小化的设备内验证。另外，还应确保不能在运行时调用本地方法。

2）应用程序级别的安全：与 J2SE 不同，CLDC/KVM 不允许定制安全管理器，支持 CLDC 的 JVM 提供了一种简单的沙箱安全模型，该模型通过确保应用程序存在封闭的环境内运行，并且只调用设备支持的类来实施安全保证。

沙箱的需求主要有：

1）类文件必须经过审核且是可用的 Java 程序。

2）应用程序的下载、安装和管理等操作都不能修改、覆盖或者绕过类虚拟机实现的标准的加载机制。

3）只有预先定义好的、封闭的 Java API 和类可以被应用程序调用。这包括配置、简表、可选包以及该应用自定义的类。

4）任何没有被 CLDC 定义的 native code 都不允许调用。这意味着应用程序不能下载一个新的含有本地代码的类库并使用。

5）CLDC 规范额外规定了系统类和应用程序类的命名规则。也就是说，为了满足上面说的沙箱的要求，所有 java.* 和 java. microedition.* 都属于系统类，不能被覆盖、修改，也不能任意增减。应用程序类不能使用上述包名来实现自己的类。

6）除了只能调用系统类之外，应用程序只能调用自身 JAR 包中的类。这样保证了应用程序不能从其他的应用程序中“盗取”数据(或类的实现)来达到自身的目的。

> 说明：另外，CLDC 上层的简表也会提供某些安全机制，例如端对端的安全机制，主要指在数据传输时的安全，如数字签名、加密等机制。考虑到网络不是 CLDC 设备必须支持的功能，这方面的定义是由上层相应的简表来完成的，如 MIDP；CLDC 规范中并没有详细的规定。

1.3 MIDP应用程序开发

移动信息设备简表(MIDP,Mobile Information Device Profile)针对的是移动信息设备，或者体积较小的能力有限的手持设备(MID)，但这些设备的共性是具有用户接口，即屏幕和输入机制。MIDP目标设备的典型例子就是手机。

1.3.1 MIDP设备需求

移动信息设备的硬件要求很简单，即单色或彩色显示屏幕，最小尺寸为96像素×54像素；用户的输入机制为：小型键盘或者触摸屏；双向无线网络连接能力；最小128KB永久存储空间用于MIDP系统软件(MIDP2.0需要256KB)，8KB用于应用程序定义的存储空间，32KB(MIDP2.0需要128KB)内存用于Java运行时堆的使用。其中，用户输入机制的定义是灵活的，MIDP引用了电话上的小键盘作为单手键盘，随着技术的创新，还会包括更多的输入方法，例如触摸屏、游戏键盘(例如Nokia的游戏手机)等。总之，应用程序一定要在有键盘的设备上运行，而该设备必须提供键盘到字母的映射。

除了硬件要求，MIDP规范还定义了其实现的软件需求，即支持MIDP的设备应该能够做到以下几点：

1) 管理软件，并能够提供运行KVM虚拟机的线程。

2) 提供读写永久存储空间，可以访问无线网络、显示输出等。

3) 能接受按钮或触摸屏事件的触发功能。

4) 实现MIDP应用程序的生存周期安装、选择、启动、关闭和删除的管理。

需要指出，Java虚拟机、KVM仅仅单个执行线程，如果当前设备的操作系统不支持多线程，虚拟机可以模拟运行。此外，MIDP规范没有定义应用管理软件(Application Management Software,AMS)如何工作和编写，仅定义了其应该实现的功能，实现时，该应用程序管理软件可以用非Java语言编写，在不同的设备上该软件与操作系统的交互其差别可能会比较大，因此支持J2ME的产品中的Java应用程序管理器(Java Application Manager)都是厂商自行研发的。

1.3.2 MIDlet Suites详解

在MIDP规范中最重要的一件事情就是定义一个MIDP应用程序的组成。换句话说，它应该回答任何开发人员都可能问的问题，如何打包类文件并将其正确下载？程序的入口在哪里？设备如何识别该MIDP应用程序是自己想要的？这些问题的答案在MIDP规范中无法找到，这是因为随着设备的不同答案各有差异。

同Applet类似，一个MIDP程序被称为“MIDlet”，MIDlet应用的主类需要从一个特殊的类(javax. microedition. midlet. MIDlet)中继承出来，并提供共用的默认的构造函数。一个或多个MIDlet打包在一起叫做MIDlet Suite，其表现形式是以一个.jar为后缀名的文件，这个JAR文件非常重要，所包含的主要信息如下：

1) 包含MIDlet运行所需要的所有类，与Applet不同，MIDlet不运行在运行时下载的其他类，而只运行在同一个MIDlet中的类。

2）JAR 的 minafest 文件是个文本文件，用于描述 JAR 文件的内容，定义了在 MIDlet 中的重要信息，如名称、主类、图标等等。

3）其他资源文件，如使用的图像、声音文件。

另外，JAR 文件是作为一个整体提供给设备的，其中个别的类或者文件提取出来都是没有意义的。从概念上讲，一个 MIDP 应用的边界是 JAR 文件，在同一个 MIDlet Suite 中的 MIDlet 可以共享数据和类，但这并不是绝对的。

一个 MIDlet Suite 实际上有两个文件：一个是标准的 JAR 文件，其中有相关的类、MIDlet 信息描述和相关资源文件；第二个是文本文件，称为 Java 应用程序描述器(Java Application Descriptor,JAD)，它重复了一些在 JAR 文件(Manifest File)中的信息以及 Suite 的名字、大小和版本等。这样可以让设备中的 J2ME 程序管理器快速决定是否需要下载和安装等问题。

除了为下载提供单一入口外，MIDlet Suite 的另一个重要用途是提供 MIDlet 的活动范围，换句话说，在不同的 MIDlet Suite 中的 MIDlet 是不能相互通信的，所有 MIDlet Suite 的类必须放在同一个 JAR 文件中。

1.3.3 清单文件和描述文件

在每个 JAR 文件中有一个文本文件叫做 Manifest。Manifest 文件描述了 JAR 文件并在 JAR 文件中名为/META-INF/MANIFEST. MF。该文件是一系列名字和值的列表。

```
MIDlet-1:HelloJ2ME,HelloJ2ME. png,HelloMIDlet
MIDlet-2:MyMIDlet,MyMIDlet. png,MyMIDlet
MIDlet-Name:HelloJ2ME
MIDlet-Vendor:Unknown
MIDlet-Version:1.0
MicroEdition-Configuration:CLDC-1.0
MicroEdition-Profile:MIDP-2.0
```

文件中重要属性就是“MIDlet-1”、“MIDlet-2”到“MIDlet-n”，它们定义了在 Suite 中所有 MIDlet。如果 Manifest 文件中出现任何安装程序不能解析的内容，则会导致程序安装失败。

应用程序描述文件和 Manifest 有相同的文件格式：

```
MIDlet-1:HelloJ2ME,HelloJ2ME. png,HelloMIDlet
MIDlet-2:MyMIDlet,MyMIDlet. png,MyMIDlet
MIDlet-Jar-Size:1152
MIDlet-Jar-URL:HelloJ2ME. jar
MIDlet-Name:HelloJ2ME
MIDlet-Vendor:Unknown
MIDlet-Version:1.0
MicroEdition-Configuration:CLDC-1.0
MicroEdition-Profile:MIDP-2.0
```

需要注意的是，MIDlet 的属性名是区分大小写的。还要注意所有以“MIDlet-”开始的