

第 1 部分

基 础 篇

- 第 1 章 J2ME 概述
- 第 2 章 MIDP 应用程序

J2ME 概述

学习目标

- 了解 J2ME 的发展历史
- 了解 J2ME 与 J2EE 及 J2SE 的联系和区别
- 理解 J2ME 的体系结构
- 掌握 KVM
- 掌握 CLDC 和 CDC, 以及它们之间的联系和区别
- 理解 MIDP
- 掌握 J2ME 的开发环境

本章主要讲述 J2ME 的基本知识, 通过讲述 J2ME 的历史, 并将 J2ME 与 J2EE 和 J2SE 进行比较, 说明 J2ME 的适用范围。本章的重点是 J2ME 的体系结构, 它包含 KVM (Kilo Virtual Machine, 千字节虚拟机)、CLDC/CDC 和 MIDP。J2ME 构建于 MIDP (Mobile Information Device Profile, 移动信息设备简档) 和 CLDC (Connected Limited Device Configuration, 有限连接设备配置) 或 CDC (Connected Device Configuration, 连接设备配置) 基础之上。MIDP 是设备生产厂商根据需要构造的一系列 Java APIs, 不同的移动设备厂商可以根据不同的需要构造不同的 APIs 来满足设备开发需要; 而 CLDC 或 CDC 是由 Sun 公司提供的所有移动设备必须支持的 Java APIs。本章最后介绍一种较常见的 J2ME 开发环境, 并通过一个小例子来讲述 J2ME 的编译和运行机制。

1.1 J2ME 简介

1.1.1 J2ME 的历史

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。

J2ME(Java 2 Micro Edition)是 Java 家族中的一个成员。Java 的前身是 Oak,它最初的设计目的是用于洗衣机、电视机顶盒等电子产品,但一直没有得到大的发展,阴差阳错,由于 Internet 技术的发展,使得 Java 成为了最受欢迎的网络编程语言,或者反过来说,Java 技术的应用推进了 Internet 的发展。正是由于 Java Applet 的应用,使得原来死板的网页变得灵动鲜活起来,并且得到因特网用户的喜爱。由此,Java 的发展也慢慢偏离了设计者的初衷。当然,这个结果是 Java 的发明者始料未及而又乐意看到的。1998 年 1 月,Sun 公司成立了一个名为 Spotless 的项目组,开始研究将 Java 技术应用到资源有限的电子设备上,在 Motorola 等厂商的推动下,这个研究项目组发布了商业化的产品——KVM,并于 1999 年 6 月推出了 J2ME,而且重新划分了 Java 的体系结构,这就是针对企业级应用的 J2EE(Java 2 Enterprise Edition)、针对普通 PC 应用的 J2SE(Java 2 Standard Edition)以及 J2ME(Java 2 Micro Edition)。而 J2ME 所面向的目标设备就是智能卡、手机、PDA 传呼机、电子元器件、电视机顶盒乃至洗衣机、微波炉等电子设备。但是,由于 J2ME 面向的设备在电源、处理器、内存等方面存在很大的差别,即使对于同一种产品,不同的型号和不同的生产厂商的实现方法也有很大的差距。因此,Sun 并没有给 J2ME 制订一个规范,而是根据目标设备的处理能力、存储大小等,将它们划分为两个范畴:有限连接设备和连接设备,并在此基础上发展出了两个规范,即 CLDC 和 CDC。目前,针对有限连接设备的技术已经得到长足的发展,并在此基础上推出了 MIDP(移动信息设备简档)。

2001 年 6 月 4 日至 8 日,来自全球各地的 25000 多名 Java 开发商和客户代表云集旧金山,第六届 JavaOne 开发商大会在这里举行。在大会上,Java 技术的发明者、全球著名的硬件、软件与服务的领先供应商——Sun 公司在大会上推出了新倡议、新网站、新产品和新项目,大力促进了 Java 技术的升华。业界广泛支持 Java 技术和 J2SE、J2ME、J2EE 平台,研发和热卖 Java 产品,让 Java 热浪再次席卷美国西部大都市旧金山。其中与手机相关的项目归纳如下。

- Sun 公司与业界领先厂商 Motorola、Nokia、Nextel、Openwave、Siemens 和 Symbian 等结成联盟,推出一个新的 Java 无线开发商倡议。
- 全球许多电信运营商,如 Cingular Interactive、Far EasTone、J-Phone、KDDI、Omnitel、One2One、SmarTone、Sprint PCS、Telefonica 和 Vodafone 等,都将配置或试用基于 Java 技术的服务与器件。

由此可见,J2ME 是无线通信行业极为热门的方向,未来也有着良好的发展趋势。现在国内外使用 J2ME 比较多的应用包括车载信息和计算、数字电视、电视机顶盒、Internet 访问设备、家庭 Audio/Video 和无线手持设备等。

J2ME 具有很多 Java 技术特性,主要有如下特性:

- 可以在各种支持 Java 的设备上运行;

- 代码短小;
- 充分利用 Java 语言的优势;
- 安全性好;
- 用 J2ME 实现的应用可以方便地升级到 J2SE、J2EE。

1.1.2 J2ME 与 J2EE 及 J2SE 的比较

J2SE™(The Java™ 2 Platform, Standard Edition)是 Java 的标准版本,它主要用于桌面应用程序的编程。

J2EE™(The Java™ 2 Platform, Enterprise Edition)是目前比较流行的多层企业应用技术。它的优点之一是能够适应不同类型的客户端,如 Web 浏览器、Java Applets 和 Java 应用程序,这些客户端能够很容易地部署在笔记本电脑、台式计算机和工作站上,实现跨平台应用。

J2ME™(The Java™ 2 Platform, Micro Edition)是当前使用比较多的基于移动设备的开发技术。它提供了一种新型的企业客户端类型,如手机、PDA 等手持设备。J2ME 构建于 MIDP 和 CLDC 或 CDC 基础之上。MIDP 是设备生产厂商根据需要构造的一系列 Java APIs,不同的移动设备厂商可以根据不同的需要构造不同的 APIs 来满足设备开发需要;而 CLDC 或 CDC 是由 Sun 公司提供的所有移动设备必须支持的 Java APIs。因此,移动设备的开发难点之一在于如何开发出具有通用性的程序。

结合 J2ME 和 J2EE 技术,用户可以建立一个完整的移动电子商务平台。移动电子商务(Mobile Commerce,简称 M-Commerce)是传统电子商务在移动网络中的应用,但是 M-Commerce 提供了比 E-Commerce 更灵活、更方便、更及时的信息服务。随着无线接入网络 GPRS(2.5G)和 3G 的发展,在无线网络中数据的传送速度会越来越快,为移动电子商务提供了良好的网络通信环境。

J2ME、J2EE 和 J2SE 的比较如图 1-1 所示。

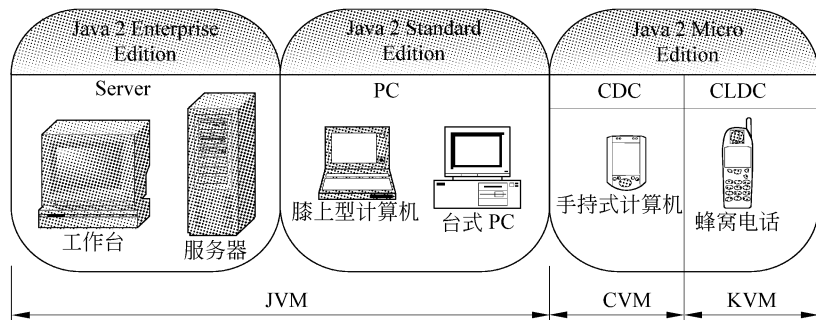


图 1-1 J2ME 与 J2EE 和 J2SE 的比较

在图 1-1 中,JVM 表示 Java Virtual Machine,它是 J2SE 和 J2EE 的虚拟机;CVM 表示 Coherent Virtual Machine,是 CDC 设备所用的虚拟机;KVM 表示 The K Virtual Machine,是 CLDC 设备所用的虚拟机,因为设备中的内存以 K 为单位,所以叫做 KVM。

1.2 J2ME 的体系结构

J2ME 应用程序的体系结构主要分为四个部分:操作系统、Java 虚拟机、CLDC/CDC 和 MIDP 应用程序,如图 1-2 所示。

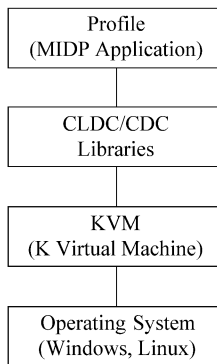


图 1-2 J2ME 应用程序的体系结构

1.2.1 KVM 虚拟机

KVM(Kilo Virtual Machine,千字节虚拟机)的名字反映了这些设备的存储量是以千字节的倍数来度量的。KVM 虚拟机适用于拥有 16/32 位的 RISC/CISC 微处理器或控制器,总的存储器只有 160KB 左右,其中 128KB 用来存放虚拟机和各种库的设备。与传统的 Java 虚拟机相比,KVM 具有以下特性:

- 虚拟机本身仅仅需要 40~80KB 内存;
- 只需要 20~40KB 动态内存(堆);
- 能够运行在 16 位 25MHz 处理器上。

因为 KVM 和 JVM 针对的设备不同,因此它们的功能也有所不同。KVM 主要针对的是移动设备,由于移动设备与服务器和台式设备相比具有很大的局限性,因此,KVM 相对 JVM 来说也有很大程度上的简化。KVM 不支持以下功能:

- 不支持浮点数。因此在 CLDC 设备上的编程都不能使用 double 和 float 类型的变量。去掉对浮点数的支持主要是因为大多数 CLDC 设备不支持浮点数,同时,在虚拟机上的浮点运算 CPU 消耗太大。

- 不支持 JNI(Java Native Interface)。主要是考虑到安全性和系统的开销。
- 不支持用户自己定义类类装载器(Class Loader)。在 KVM 中有内置的类装载器,它是不可被覆盖(Override)和替换的,这主要是考虑到设备的安全性。
- 不支持线程组(Thread Groups)和守护线程(Daemon Threads)。
- 不支持映射(Reflection)。没有映射的特征,因此不能使用 RMI(远程方法调用)和 Object Serialization(对象串行化)。
- 不支持类实例的销毁(Finalization)。因此 KVM 不可以在对象被回收前销毁实例。
- 不支持弱引用(Weak Reference)。
- 有限的错误处理机制。在 CLDC 中,仅定义了 3 个错误处理类: java. lang. Error、java. lang. OutOfMemory 和 java. lang. VirtualMachineError。非运行错误是设备独立的,可以通过终止应用程序或者重启设备来实现。

1. 2. 2 CLDC/CDC

1999 年, Sun 公司针对低端消费电子产品发布了两个规范: CLDC 和 MIDP。CLDC——有限连接设备配置为低端消费电子产品的运行提供了最小的库集和 APIs,它是所有低端消费电子产品必须支持的,因此这些 APIs 在所有的设备上都是可以运行的。CDC——连接设备配置是 CLDC 的超集,它包含 CLDC 中的所有 APIs,同时它还有自己的 APIs,它为高端的消费电子产品提供运行环境。本文主要讲述 CLDC,对 CDC 并不做主要介绍。CLDC 1.0 具有表 1-1 所示的属性和限制。

表 1-1 CLDC 1.0 的属性和限制

设备属性	最低要求
RISC/CISC 处理器	16bit/RISC, 32bit/CISC
内存	160KB
不变内存	128KB
可变内存(RAM)	32KB
供用户使用的不变内存	8KB

CLDC 并不是 J2SE 严格上的子集,图 1-3 表示 CLDC、CDC 和 J2SE 之间的关系。

CLDC 包含 java. io. *, java. lang. *, java. util. *和 javax. microedition. io. *四个包。这四个包所包含的类和接口如表 1-2 所示。

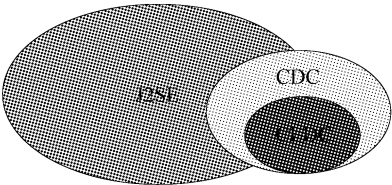


图 1-3 CLDC、CDC 和 J2SE 之间的关系

表 1-2 CLDC 包

包	说 明	类 和 接 口
java.io.*	输入输出包 (主要处理数据的输入和输出, 这个包与 J2SE 中的 java.io 包相同)	ByteArrayInputStream ByteArrayOutputStream DataInput DataInputStream DataOutput DataOutputStream EOFException InputStream InputStreamReader InterruptedIOException IOException OutputStream OutputStreamWriter PrintStream Reader UnsupportedEncodingException UTFDataFormatException Writer
java.lang.*	基本类 (J2ME 中使用的基本类, 这个包是 J2SE 中 java.lang 包的子集)	ArithmeticException ArrayIndexOutOfBoundsException ArrayStoreException Boolean Byte Character Class ClassCastException ClassNotFoundException Error Exception IllegalAccessException IllegalArgumentException IllegalMonitorStateException IllegalStateException IllegalThreadStateException IndexOutOfBoundsException InstantiationException Integer InterruptedException Long

续表

包	说 明	类 和 接 口
java. lang. *	基本类 (J2ME 中使用的基本类, 这个包是 J2SE 中 java. lang 包的子集)	Math NegativeArraySizeException NullPointerException NumberFormatException Object OutOfMemoryError Runnable Runtime RuntimeException SecurityException Short String StringBuffer StringIndexOutOfBoundsException System Thread Throwable VirtualMachineError
java. util. *	工具类 (处理日期、时间等)	Calendar Date EmptyStackException Enumeration Hashtable NoSuchElementException Random Stack Timer TimeZone Vector
javax. microedition. io. *	网络类 (主要处理移动设备与外界的网络连接)	Connection ConnectionNotFoundException Connector ContentConnection Datagram DatagramConnection HttpConnection InputConnection OutputConnection StreamConnection

1.2.3 简档

在 CLDC 中,没有用户界面的包不包含具体设备的信息,这给移动设备应用程序的开发带来了困难,因为这种移动设备上可以运行的程序可能在另外一种移动设备生产厂商的产品中就不能运行,这极大地降低了应用程序的可移植性。为了提高应用程序的可移植性,很多移动设备生产厂商制定了一些大家共同遵循的规范,这种规范称为“简档”(Profile)。根据这些简档开发出的应用程序可以在这些不同厂商的产品上运行。MIDP 就是一种最常见的简档。MIDP 构架于 CLDC 之上,它具有以下特征:

- 最小屏幕尺寸为 96×54 像素;
- 显示深度为 1 位;
- 128KB 不变内存用于 MIDP 组件;
- 8KB 不变内存用于持久数据存储;
- 32KB 可变内存用于运行时虚拟机;
- 双向无线连接。

在 MIDP 中定义一些 APIs,包括 `javax.microedition.lcdui.*`、`javax.microedition.midlet.*` 和 `javax.microedition.rms.*`,如表 1-3 所示。

表 1-3 MIDP 包

包	说 明	类 和 接 口
<code>javax.microedition.midlet.*</code>	MIDP 的超类 (用于创建应用程序)	MIDlet MIDletStateChangeException
<code>javax.microedition.lcdui.*</code>	界面类 (用于创建图形用户界面)	Alert AlertType Canvas Choice ChoiceGroup Command CommandListener DateField Display Displayable Font Form Gauge Graphics Image ImageItem Item

续表

包	说 明	类 和 接 口
javax. microedition. lcdui. *	界面类 (用于创建图形用户界面)	ItemStateListener List Screen StringItem TextBox TextField Ticker
javax. microedition. rms	数据记录类 (相当于移动设备中的数据库)	InvalidRecordIDException RecordComparator RecordEnumeration RecordFilter RecordListener RecordStore RecordStoreException RecordStoreFullException RecordStoreNotFoundException RecordStoreNotOpenException

目前有两种比较流行的简档，一种就是本文介绍的 MIDP，另一种是 KJava。这两种简档各有优点。MIDP 不依赖任何设备生产厂商，它是一种普遍的工业标准，基于 MIDP 简档开发的程序可以在任何设备上运行。KJava 中的 APIs 主要是 Sun 公司特别制定的，主要运行在 Palm OS 上；KJava 有点像 J2SE 中的 AWT (Abstract Windowing Toolkit, 抽象窗口工具包)，它并不是 J2ME 中的标准包，并且它的主包是 com. sun. kjava。图 1-4 描述了 J2ME 中不同简档之间的比较。

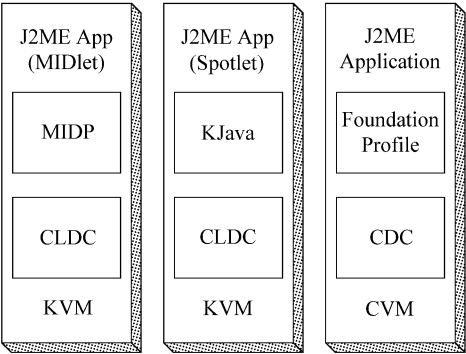


图 1-4 J2ME 中不同简档之间的比较