

JAVA SE6

全方位学习



朱仲杰 编著

- Java之父James Gosling隆重推荐!
- 朱仲杰Java畅销力作SE 6新版!
- 繁体版累计销售超过20000册!
- 雄踞台湾地区Java著作销售榜首!



· 碁峯 ·
www.gotop.com.tw



机械工业出版社
China Machine Press

公司核心技术丛书

TP312/2930D

2008

JAVA SE6

全方位学习

朱仲杰 编著



机械工业出版社
China Machine Press

本书全面介绍了Java SE6的相关知识。全书共24章,包括Java语言的介绍、Java的表达式和语句、Java面向对象程序设计、Object类常用方法、内存、Java Application与Applet、异常处理、容器和版面配置、AWT组件、绘图、Inner Class、Wrapper Class、字符串处理、文件输入与输出、线程、Collection以及网络程序设计。

本书叙述简洁清晰、范例丰富、内容详尽,适合不同层次的软件开发人员参考。

本书中文简体字版由中国台湾碁峰资讯有限公司授权机械工业出版社出版,未经本书原版出版者和本书出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书原版版权属碁峰资讯有限公司

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2008-1785

图书在版编目(CIP)数据

Java SE6全方位学习 / 朱仲杰编著. —北京:机械工业出版社, 2008.6
(Sun公司核心技术丛书)

ISBN 978-7-111-23877-5

I. J... II. 朱... III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2008)第047389号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:李南丰

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2008年6月第1版第1次印刷

186mm×240mm · 24印张

标准书号:ISBN 978-7-111-23877-5

ISBN 978-7-89482-628-2(光盘)

定价:59.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线(010) 68326294

前言

老实说，最近一年来，因为新工作的关系，几乎没写过一行Java代码。但这不代表我放弃Java或Java对我没有任何的影响。相反，要是没有Java训练，我今天在写别的程序语言时，不会这么顺手。学Java对我的影响，已经根深蒂固。今天在用其他程序语言设计程序时，那种结构化、模块化、对象化、层次化的观念，也是源自于这些年来编写Java的训练成果。

看了这么多程序语言，我发现Java真的是最好的。我写过VB、ASP、C#、JavaScript等，常常写起来束手束脚的，探究其原因，原来是Java提供了太完整且简单的对象环境，而当其他程序语言不那么简单、不那么完整时，我得费很大的功夫，才能把程序写得满意，或写了一大堆的代码，才能完成Java几行代码就做得到的事。

我很高兴我会Java，但Java跟其他程序语言比起来，学习曲线比较陡峭，不是很容易就可以学得好的程序语言。当年写这本书的目的，是要让初学者能够通过最简单的描述和说明来跨越学习Java的第一道门坎，很高兴这本书做到了。前一阵子在面试新的工程师时，有不少人学Java时都看过我的书呢！这是当作者最大的收获。

其实我还有很多的写作计划，但有了小孩之后，全部的心思和精力都放在小孩的身上，这也是这本书迟迟才改版的原因。不过我不会放弃Java，它是我工作上的最爱，希望它也能变成你的最爱。

朱仲杰

2007年12月20日

目 录

前言

第一部分 认 识 篇

第1章 Java的过去、现在和未来	1
1.1 Java的历史	1
1.2 Java的架构	3
1.2.1 Java的语法	3
1.2.2 Java的运行环境	4
1.2.3 Java API简介	4
1.3 Java的优点	6
1.4 Java未来的发展	7
1.5 邂逅Java SE 6	7
1.6 本章总结	8
1.7 课后习题	9
1.8 参考文献	9
第2章 程序开发工具下载、安装与使用	10
2.1 JDK下载、安装与设置	10
2.1.1 JDK	10
2.1.2 Java说明文档	18
2.2 程序编辑工具下载、安装、设置与使用	20
2.3 本章总结	20
2.4 课后习题	22
2.5 参考文献	22
第3章 程序初体验	23
3.1 开始写程序	23
3.2 使用记事本	25
3.2.1 程序编辑	25
3.2.2 程序编译	26
3.2.3 程序运行	28
3.3 程序说明	29
3.4 范例欣赏	30

3.5 本章总结	33
3.6 课后习题	33
3.7 参考文献	34

第二部分 基 础 篇

第4章 程序基本单元	35
4.1 程序中的标记	35
4.1.1 关键字	35
4.1.2 标识符	36
4.1.3 字面值	38
4.1.4 符号	38
4.2 数据类型	38
4.2.1 基本数据类型	39
4.2.2 变量的声明	41
4.2.3 类型转换	43
4.3 字符串	44
4.4 本章总结	45
4.5 课后习题	45
4.6 参考资料	46
第5章 Java的表达式	47
5.1 算术运算	47
5.2 比较运算	49
5.3 逻辑运算	50
5.4 递加、递减运算	53
5.5 赋值运算	54
5.6 位运算	56
5.7 位移运算	57
5.8 优先级和结合性	59
5.9 本章总结	60
5.10 课后习题	60
5.11 参考文献	61
第6章 Java的语句	62
6.1 认识语句	62

6.2 一般语句	63	7.2.7 类的多态	100
6.2.1 注释	63	7.2.8 类成员与实例成员	103
6.2.2 赋值语句	65	7.2.9 再看构造函数	106
6.2.3 对象的使用(方法的调用)	65	7.2.10 遮蔽、改写与重载	108
6.2.4 其他	66	7.3 本章总结	111
6.3 声明语句	66	7.4 课后习题	112
6.3.1 变量的声明	66	7.5 参考资料	115
6.3.2 对象的声明	66	第8章 深入Java面向对象程序设计	116
6.3.3 类的声明	67	8.1 包的使用	116
6.3.4 方法的声明	67	8.1.1 什么是包	116
6.4 条件控制语句	68	8.1.2 package语句	116
6.4.1 if	68	8.1.3 import语句	120
6.4.2 else	69	8.1.4 classpath的设置	121
6.4.3 层次式的if	70	8.1.5 访问权限修饰符的使用	122
6.4.4 ?:操作符	72	8.2 final修饰符	125
6.4.5 switch	73	8.3 抽象类	126
6.5 循环控制语句	76	8.4 接口	126
6.5.1 for	77	8.5 本章总结	130
6.5.2 while	79	8.6 课后习题	130
6.5.3 do...while	80	8.7 参考文献	131
6.5.4 高级循环控制	81	第9章 Object类常用方法介绍	132
6.6 本章总结	84	9.1 类的类型转换	132
6.7 课后习题	84	9.2 对象之间的比较运算	133
6.8 参考文献	86	9.3 散列码	136
第三部分 进 阶 篇		9.4 对象的复制	137
第7章 Java面向对象程序设计	87	9.5 将对象转为字符串	139
7.1 从日常生活中看对象	87	9.6 本章总结	140
7.1.1 类与对象	87	9.7 课后习题	140
7.1.2 成员	88	9.8 参考文献	140
7.1.3 继承	88	第10章 深入内存	141
7.1.4 多态	90	10.1 声明	141
7.2 用Java实现面向对象程序设计	90	10.1.1 变量	141
7.2.1 定义类	90	10.1.2 对象	142
7.2.2 定义成员	91	10.2 数组	143
7.2.3 产生与使用对象	92	10.2.1 数组的产生	143
7.2.4 初看构造函数	93	10.2.2 数组的使用	144
7.2.5 封装	96	10.2.3 length属性	145
7.2.6 类的继承	99	10.2.4 数组的复制	146
		10.2.5 多维数组	147

10.2.6	对象数组	148
10.2.7	java.util.Arrays API使用	148
10.3	变量的访问范围	155
10.4	参数传递	159
10.4.1	基本数据类型参数值传递	159
10.4.2	类对象类型参数值传递	160
10.5	垃圾回收	161
10.5.1	Garbage Collection简介	162
10.5.2	调用GC	162
10.5.3	finalize方法	164
10.5.4	回收顺序	165
10.5.5	GC何时运行	166
10.6	本章总结	167
10.7	课后习题	167
10.8	参考文献	167
第11章	Java Application与Applet	168
11.1	Application	168
11.1.1	输入与输出	168
11.1.2	系统参数	171
11.1.3	System类使用	174
11.1.4	Runtime类使用	175
11.2	Applet	176
11.2.1	输入与输出	177
11.2.2	基本方法使用	178
11.2.3	安全性	180
11.3	Application与Applet合并	181
11.4	本章总结	181
11.5	课后习题	182
11.6	参考文献	182
第12章	异常处理	183
12.1	认识异常	183
12.1.1	Runtime Exception	184
12.1.2	Checked Exception	184
12.1.3	Error	185
12.2	捕捉和处理异常	185
12.2.1	捕捉异常try...catch	185
12.2.2	处理异常	186
12.2.3	finally	189
12.2.4	异常捕捉的顺序	190

12.2.5	throws关键字的使用	192
12.3	自己设计异常	193
12.3.1	throw关键字的使用	193
12.3.2	定义自己的Exception类	194
12.4	异常与方法的改写	195
12.5	本章总结	197
12.6	课后习题	198
12.7	参考文献	199

第四部分 GUI 篇

第13章	容器和版面配置	201
13.1	认识AWT	201
13.2	Container	202
13.2.1	Frame	202
13.2.2	Panel	205
13.3	版面配置	206
13.3.1	什么是Layout	206
13.3.2	BorderLayout	206
13.3.3	FlowLayout	209
13.3.4	CardLayout	210
13.3.5	GridLayout	212
13.3.6	GridBagLayout	213
13.3.7	不使用Layout	218
13.4	本章总结	219
13.5	课后习题	219
13.6	参考文献	220
第14章	事件处理	221
14.1	认识事件处理机制	221
14.1.1	什么是事件	221
14.1.2	Java事件处理架构	221
14.1.3	委托模式	222
14.1.4	AWT Event类架构	222
14.2	使用事件处理机制	224
14.2.1	WindowEvent	224
14.2.2	MouseEvent	229
14.2.3	KeyEvent	232
14.2.4	其他Low-level Event	234
14.3	本章总结	235
14.4	课后习题	235

14.5 参考文献	236	17.3.2 字体设置	278
第15章 AWT组件使用	237	17.3.3 语句块处理	280
15.1 Button	237	17.4 本章总结	280
15.2 Checkbox	239	17.5 课后习题	281
15.2.1 多重选择	239	17.6 参考文献	281
15.2.2 单一选择	241		
15.3 Choice	242	第五部分 内功心法篇	
15.4 List	244	第18章 内部类	283
15.5 Label	246	18.1 认识内部类	283
15.6 Scrollbar	248	18.2 成员式内部类	284
15.7 ScrollPane	250	18.3 区域式内部类	289
15.8 TextField	251	18.4 匿名式内部类	290
15.9 TextArea	253	18.5 本章总结	291
15.10 本章总结	255	18.6 课后习题	292
15.11 课后习题	255	18.7 参考文献	292
15.12 参考文献	255	第19章 包装类	293
第16章 高级AWT组件使用	256	19.1 认识包装类	293
16.1 Font	256	19.2 Integer类	293
16.2 Color	258	19.3 Boolean类	295
16.3 Menu	262	19.4 Byte类	296
16.4 Dialog	266	19.5 Character类	296
16.5 FileDialog	268	19.6 Double类	297
16.6 本章总结	271	19.7 Float类	298
16.7 课后练习	271	19.8 Long和Short类	298
16.8 参考文献	271	19.9 本章总结	298
第17章 绘图	272	19.10 课后习题	299
17.1 认识Graphics类	272	19.11 参考文献	299
17.1.1 产生Graphics对象	272	第20章 深入字符串处理	300
17.1.2 paint方法	272	20.1 深入了解String对象	300
17.1.3 Canvas类	273	20.1.1 产生String对象	300
17.1.4 坐标	273	20.1.2 String类的类方法使用	302
17.2 基本绘图方法	274	20.2 StringBuffer类使用	303
17.2.1 画线	274	20.2.1 产生StringBuffer对象	303
17.2.2 画字符串	275	20.2.2 StringBuffer类的方法使用	304
17.2.3 画矩形	276	20.3 StringTokenizer类使用	305
17.2.4 画圆	277	20.3.1 创建StringTokenizer对象	306
17.2.5 画多边形	277	20.3.2 StringTokenizer类的方法使用	306
17.3 高级绘图方法	278	20.4 本章总结	307
17.3.1 颜色设置	278	20.5 课后习题	308

20.6 参考文献	308	23.5 Map接口使用	362
第21章 文件与输入/输出处理	309	23.6 JDK 1.1中的Collection库	363
21.1 java.io包简介	309	23.7 本章总结	363
21.1.1 Stream	309	23.8 课后习题	364
21.1.2 Node	309	23.9 参考文献	364
21.1.3 方法使用	310	第24章 网络程序设计	365
21.2 文件处理	312	24.1 网络基本概念	365
21.2.1 文件操作	312	24.1.1 Client-Server架构	365
21.2.2 文件访问	319	24.1.2 互联网地址	365
21.3 数据流连接	325	24.1.3 TCP/IP传输协议	366
21.4 对象串行化	329	24.1.4 面向连接协议	366
21.5 本章总结	332	24.1.5 非面向连接协议	367
21.6 课后习题	332	24.1.6 连接端口	367
21.7 参考文献	333	24.1.7 Socket	367
第22章 线程	334	24.2 Java网络类	367
22.1 认识线程	334	24.2.1 InetAddress类	367
22.2 创建线程	335	24.2.2 Socket类	368
22.3 线程操作	336	24.2.3 ServerSocket类	371
22.3.1 启动线程	337	24.2.4 DatagramSocket类和 DatagramPacket类	372
22.3.2 停止线程运行	338		
22.3.3 暂停线程运行	339		
22.3.4 取得线程信息	344		
22.4 数据同步处理	345		
22.5 线程高级操作	350		
22.6 本章总结	355		
22.7 课后习题	355		
22.8 参考文献	356		
第23章 Collection	357		
23.1 认识Collection库	357		
23.2 Set接口使用	358		
23.3 List接口使用	360		
23.4 Iterator接口使用	361		

第一部分 认识篇

第1章 Java的过去、现在和未来

Java这个名词从几年前的默默无闻，到现在的大红大紫，信息产业界不管是硬件、软件的厂家，都纷纷推出Java相关的产品。到底Java是什么？它为什么有这么大的魅力？它有什么神奇的功能？本章就要带你来掀开这层神秘的面纱，带你进入Java的世界。

本章的学习目标：

- 了解Java的历史
- 了解Java的架构
- Java有哪些优点
- Java的运行环境

本章以文字性叙述为主，学习时间预计为一个小时。

1.1 Java的历史

早在1990年12月，SUN就由Patrick Naughton、Mike Sheridan和James Gosling成立一个叫做Green Team的小组。这个小组的主要目标是要开发一种分布式系统架构，使其能在消费性电子产品操作平台上运行，例如PDA、手机、信息家电（IA，Internet/Information Appliance）等。隔年，也就是1992年的9月3号，Green Team发表了一款名叫Star Seven（*7）的机器，如图1-1所示。它有点像现在我们熟悉的PDA，不过它却有着比现在PDA还强大的功能，像无线通信（wireless network）、5吋彩色的LCD、PCMCIA接口等，现在市面上的PDA几乎都不是它的对手，更不要说是早在10年前那个计算机还不普及的时代了。

而Java语言的前身Oak就是在那时诞生的，主要的目的当然是用来编写在Star 7上的应用程序。为什么要叫Oak呢？原因是James Gosling办公室的窗外，正好有一棵橡树（Oak），顺手就取了这个名字。当我们在学Java时所提供的一些特性，其实在Oak中早已具备了，如安全性、网络通信、面向对象、垃圾回收（Garbage Collected）、多线程等，是一款相当优秀的程序语言。至于为什么Oak会改名为Java呢？这是因为当时Oak去注册商标时，发现有另外一家公司已经先用了Oak这个名字。Oak这



图1-1 Star 7的展示图

个名字不能用,那要取个什么新名字呢?工程师们边喝咖啡边讨论着,看看手上的咖啡,突然灵机一动,就叫Java好了。就这样它就变成了我们所熟知的Java了。

图1-2是本文作者与Java之父James Gosling合影。

可惜的是,这么优秀的产品,却不被当时的消费市场所接受。正当这个小组快要被SUN取消时,全世界第一个万维网浏览器——Mosaic诞生了。Java就以它优异的功能,在万维网的平台上编写高交互性的网页程序,我们称为Applet。因为那时没有其他的程序语言能够做到,所以原本坐以待毙的Java,又在万维网上创造了另一片天地。1995年5月23日,JDK(Java Development Kits)1.0a2版本正式对外发布,如图1-3至图1-6所示。

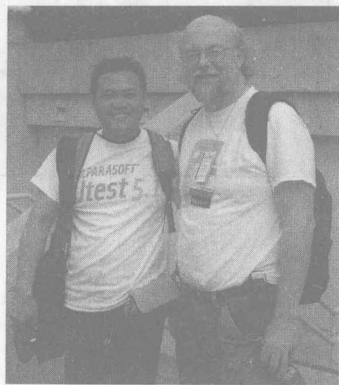
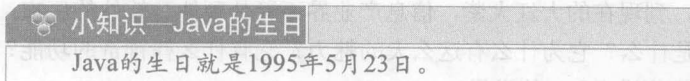


图1-2 笔者与Java之父-James Gosling
在2004年JavaOne大会上合影



WebRunner

WebRunner is a World Wide Web browser that brings true interactivity to the Internet. WebRunner makes the Internet "come alive". It builds on the network browsing techniques established by Mosaic and expands them by adding dynamic behavior that transforms static documents into dynamic applications capable of real-time interactive response. Using WebRunner you can create applications that range from interactive games to dynamic forms to customized newspapers to interactive shopping - the possibilities are endless.

WebRunner also provides a new way for users to access these applications. Software transparently migrates across the network. There is no such thing as "installing" software. It just comes when you need it. Content developers can embed new software, media types and protocols in WWW pages, and the extensions reach the user's system automatically.

What makes this dynamic behavior possible is Java (formerly "Oak"), the underlying environment in which WebRunner is built. Java is a simple, dynamic, multithreaded, safe, compact and portable object-oriented programming language and runtime system. Java has an architecture-neutral distribution format, so that Java content runs on anyone's WebRunner home page, regardless of the underlying CPU architecture.

WebRunner Alpha2 Features:

- Full-Function WWW Browser
 - HTML-compatible with Mosaic and Netscape
 - supports all standard Internet protocols
 - fast performance
- Enables Interactive Content
- Dynamic Content Loading
 - occurs transparently across the network
 - add new protocols and applications on the fly
 - platform-independent
- WWW Newsreader
- Security and Authentication Support
 - file system protection and code checking
 - Includes full Java language and runtime system
- Available on Solaris

WebRunner Beta Features:

- WYSIWYG HTML Editor
 - no HTML knowledge necessary
 - implements all common HTML extensions
 - drag and drop links, images, audio, applications into browser
- Security and Authentication Support
 - S-HTTP, RSA, public key encryption and authentication
 - HTTP server with support for real-time data feeds
- Available on Solaris, Windows95, and MacOS

For instructions on installing WebRunner, visit <http://tashyon.eng.soloa/>

Sun

图1-3 为Java量身订做的浏览器——Web
Runner, 后来改名叫Hot Java

San Jose Mercury News

THURSDAY MARCH 18, 1995

Why Sun thinks Hot Java will give you a lift

Now software designed to make World Wide Web's "home pages" more useful — and spur computer sales

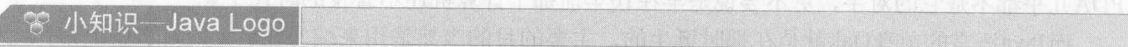
Blizzard

Yes, it really is springtime: Snow closes I-80; wind KO's power; rain spurs flooding

Mad rush for U.S. citizenship

What these guys are doing is undeniably, absolutely new. It's great stuff. . . . These guys are really pushing the envelope. ♡

图1-4 1995.5.23 San Jose Meccury News
报道Java问世的新闻



接触过Java的人,一定会对Java两个专用的Logo印象深刻,一个是Java Cup,另一个就是叫做Duke的吉祥物了。这位可爱的Duke是由Joe Palrang在1992年时创作出来的,Duke当时在Star 7上所扮演的角色是类似Office 2000中Office助手的功能。



图1-5 最早的java.sun.com网页

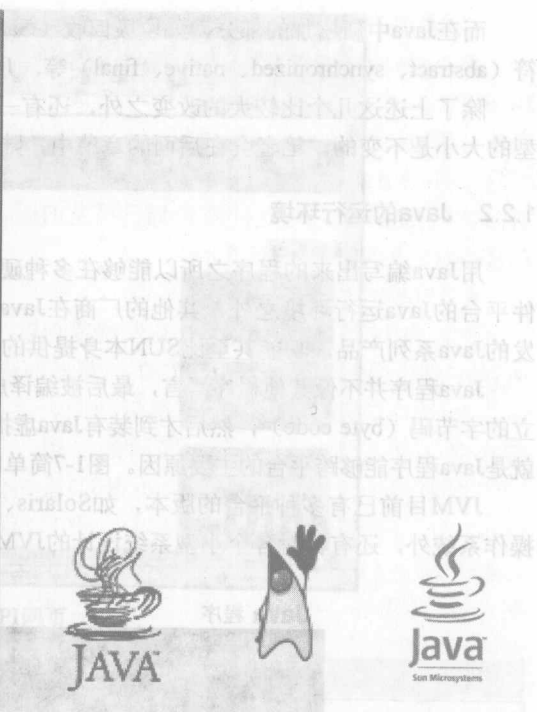


图1-6 Java Cup (旧)、Duke和Java Cup (新)

1.2 Java的架构

在了解了Java的历史背景后，我们来看看整个Java的架构。下面分成三个部分来探讨，一是Java本身的语法，二是Java运行的环境，三是Java丰富的API。

1.2.1 Java的语法

Java的语法和C语言大致上是一样的，不论是语句（statement）、条件流程控制（if）、循环（for）等。^①也许你会有疑问，C语言加上面向对象的机制那不就是C++了吗？那Java和C++有什么差别呢？简单的说，Java改进了C++中的一些缺点，并且增加了一些新的优点，让Java变得更简单、更容易学习，并且设计出来的程序威力更强大且更健壮（robust）。哪些是C++中有而Java中没有，又有哪些是Java特殊的设计，而C++中所没有的呢？我们就来简单比较一下。

在Java中没有了C语言中大家又爱又恨的指针（pointer），指针使用得当的话，对于程序的功能有很大的帮助，但是使用不当的话，死机是家常便饭。因此为了系统安全、程序的健壮性，在Java中没有指针。另外Java中也没有了C语言中程序常用的预处理器（preprocessor），如#define、#ifdef、常量声明等，当然也少了#include命令，因此也没有了头文件（.h）。对于C++的另一个差异，Java也不支持多重继承的观念，原因是不让对象和对象之间的关系变得复杂。

^① 笔者将在基础篇中详细说明这些语法的使用。

而在Java中新增加的部分,如垃圾回收(Garbage Collection)、异常处理(Exception)、新的修饰符(abstract、synchronized、native、final)等,几乎全都是出于对整个系统和程序本身安全性的考虑。

除了上述这几个比较大的改变之外,还有一些小的修正,如Java不论在何种平台上,基本数据类型的大小是不变的。笔者会在后面的章节中,陆续介绍Java这些新增功能的使用方法。

1.2.2 Java的运行环境

用Java编写出来的程序之所以能够在多种硬件平台上运行,最大的功臣除了SUN本身提供这些硬件平台的Java运行环境之外,其他的厂商在Java运行环境的开发上也投入相当多的心力,如IBM所开发的Java系列产品,功能甚至比SUN本身提供的还要强大。

Java程序并不像其他程序语言,最后被编译成所在平台的机器语言后再运行,而是先编译成一个中立的字节码(byte code),然后才到装有Java虚拟机(Java Virtual Machine, JVM)的硬件上运行,这也就是Java程序能够跨平台的主要原因。图1-7简单的表示Java源程序、字节码、JVM和硬件之间的关系。

JVM目前已有多种平台的版本,如Solaris、Linux、Windows、Macintosh等,除了这些较大型的操作系统外,还有针对各个小型系统设计的JVM,如PDA、手机、Java Card等。

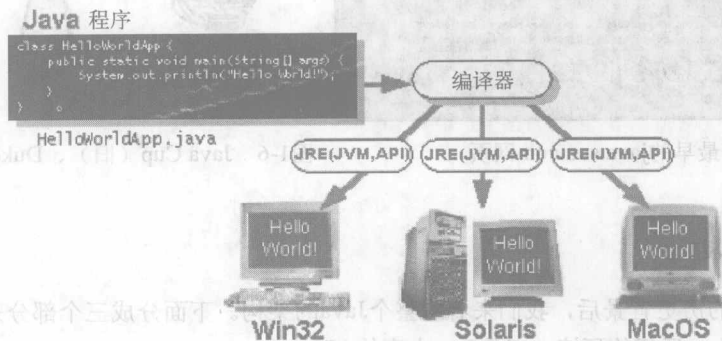


图1-7 Java运行架构示意图

整个Java的运行环境可以统称为JRE (Java Runtime Environment),它除了包括JVM之外,还包括供浏览器使用的Java插件。

1.2.3 Java API简介

Java应用程序之所以能够这么简单、快速地开发完成,而又能在各种不同的硬件平台上运行,最大的功臣莫过于一堆为它量身订做的API,笔者称它们为类库。你可以到SUN的网页上找到这些类库(<http://java.sun.com/products/>, 见图1-8),笔者把它们分成四类来介绍。

小知识

在Java 1.0或Java 1.1时,称这些类库为JDK (Java Development Kit),但在Java 1.2版之后已改名为Java 2 SDK (Software Development ToolKit) 简称为J2SDK,不过有些人习惯上还是称它们为JDK。

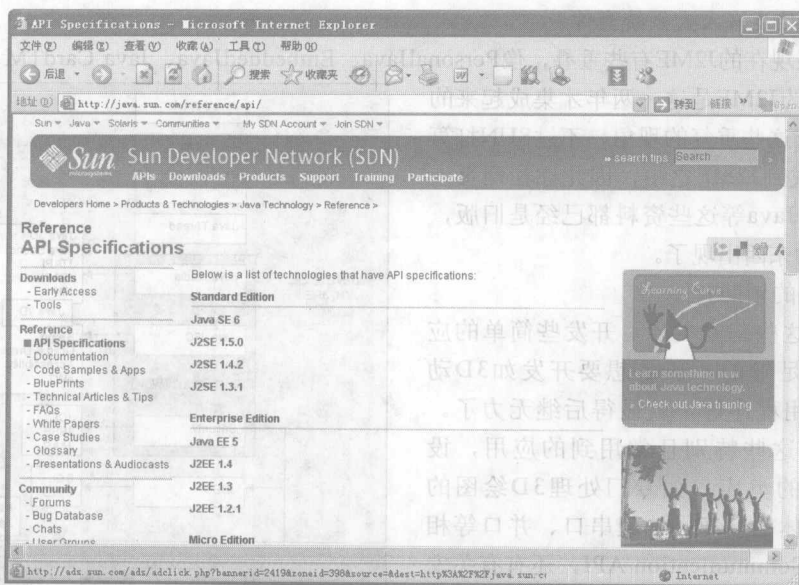


图1-8 Java API网页

名词解释-API

API是Application Programming Interface（应用程序接口）的简写。API是一组其他程序员写好的程序，你只用遵循它的使用规则，就可以在你的程序中使用这些程序了。

名词解释-SDK

SDK是Software Development Toolkit的简写。当用某种程序语言开发一个应用程序时，除了会用到那个语言所提供的API之外，还会用到如编写、编译、运行、调试等工具，而这些全部合起来就叫做SDK。

• 基本平台套件

不管开发任何Java平台的程序，一定少不了这个基本的套件，它就是Java 2 Platform, Standard Edition，简称J2SE。它除了包含开发Java程序所需的基本类库之外，还包含一些编译的程序、额外的辅助工具等，而它也是本书所会使用到的主角。除J2SE之外，SUN另外针对企业级应用程序推出了Java 2 Platform, Enterprise Edition，简称J2EE。J2EE除了J2SE中有的基本类库之外，还包含编写服务器端、分布式应用程序、事务处理等其他企业级应用程序所会用到的类库，如Servlet、Enterprise JavaBean等。

最后一个平台套件就是Java 2 Platform, Micro Edition，简称J2ME。J2ME是Java平台套件中最小的一个，它是针对消费性电子产品和嵌入式系统所设计的，例如对讲机、手机、PDA等。

❖ 注意

只有J2SE中包含了编译Java程序所需要的一些工具程序，而在J2EE和J2ME中只有类库和运行这类Java程序所需的Java虚拟机而已。所以就算你只开发J2EE或是J2ME的程序，还是会需要J2SE的这些工具程序。

● 消费性产品及嵌入式系统套件

这一部分跟现在的J2ME有些重叠，像PersonalJava、EmbeddedJava、Java CardTM、JavaPhone、Java TV等，因为J2ME是这一两年才集成起来的套件，所以会有这些重复的现象，不过SUN打算把这个部分集成到J2ME之中，因此你在网页上看到的PersonalJava等这些资料都已经是旧版，而且没有新版的东西出现了。

● J2SE额外的辅助套件

光靠J2SE这些基本类库，开发些简单的应用程序应该是足够的，但是想要开发如3D动画、多媒体应用程序等，就显得后继无力了。因此SUN为了这些特别且常用到的应用，设计了许多相关的类库，如专门处理3D绘图的Java3D、多媒体的JMF，处理串口、并口等相关通信的Java Communication API，还有有关安全方面的JCE等，种类非常多，让你在开发这方面的应用程序时，能更容易、更快速。

● 其他套件

除了上面那些套件之外，还有一些其他的类库或是相关应用程序等，如JINI、JAIN、JavaTM Message Queue等。有兴趣的读者，可以自行到Java的网站上去查看详细的资料。

最后用图1-9来表示Java 2三部分中主要的API，相信你会更清楚它们之间的分类。

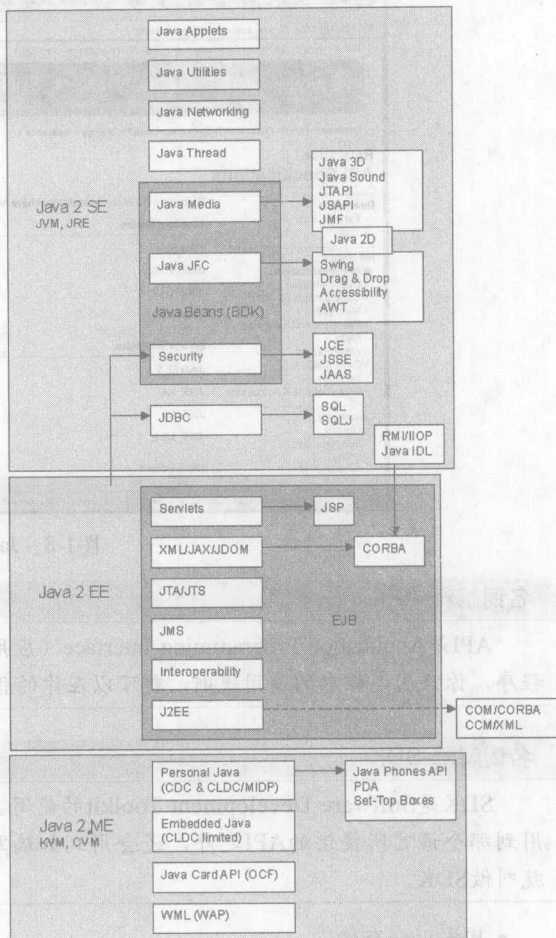


图1-9 Java API分类图

1.3 Java的优点

看完了上面那些介绍后，笔者简单地整理一下Java的优点，按照SUN官方的说法，网页上所写的优点洋洋洒洒的十几项，不过笔者只选择了几个比较明显的来介绍。

● 一次编写，处处运行 (Write once, run anywhere)

这句话一直是Java程序员的精神指南，也是Java能够受到程序设计员们喜爱的原因之一，在1-2节中已经提到过，最大的功臣就是JVM的技术。

● 简单

纯面向对象的特性，再加上众多类库的支持，使用Java开发各式各样的应用程序，可以说是易如反掌。除此之外，不论在调试或是程序修改、增加新功能方面，因为对象的特性，使得这些维护也变得非常容易。

● 安全性

前面的小节中，提到了不少Java新增的功能，绝大部分都是针对安全性的问题而考虑的，Java甚

至可以对于不同的安全需求，而设置Java应用程序的安全等级。原则上一个未经您允许的Java程序，是不可能伤及你的系统一根汗毛的。

• Unicode

Java使用Unicode作为它的标准字符，这项特性使得Java程序在不同语言平台上都能编写和运行。简单地说，你可以把程序中的变量、类名称使用中文来表示，当你的程序移植到其他语言平台时，仍然可以正常地运行^①。Java也是目前所有计算机语言中，惟一天生使用Unicode的语言。

• 网络

Java可以说是通过互联网而重生的，自然对于具备网络功能程序的编写，也是非常容易的。不论是一般互联网的程序如Socket、Email，还是服务器网页的程序Servlets、JSP，甚至分布式网络程序CORBA等的相关套件支持，可以说是非常的丰富，且使用起来也是非常的容易。

• 垃圾回收机制（Garbage Collection）

垃圾回收是由JVM协助程序对内存做动态的管理，程序需要多少内存、哪些对象的内存已经不使用了，需要归还给系统，这些繁琐且危险的工作全部交由JVM管理。让我们能够更专心的写程序，不需要担心内存的问题，而内存的统一管理，对于跨平台也有相当的帮助。

• 异常处理（Exception）

为了让Java程序更安全、更稳定，程序随机应变的能力就要好。Java引入了异常处理机制，能够在程序中比较可能有异常情况产生的地方，加上相对应的处理，让程序不致于因为突发的错误，造成运行中断或死机的情况。异常处理也让你更能控制整个程序运行的流程，也使得程序设计更为严谨。

1.4 Java未来的发展

目前Java主要应用于电子商务及企业电子化的部分，市面上介绍JSP的书籍也像雨后春笋般一本接一本地上市，对象的特性配合分布式的系统，用Java设计这类程序确实容易多了。除此之外，Java也渐渐在消费性电子平台上发挥跨平台的特性，各手机厂商、PDA厂商也纷纷发表内置JVM的系统，Java在这方面跟其他程序语言比起来，可以说是如鱼得水。回想1.1节中介绍的Java历史，消费性电子产品，原本就是当初Java发展的目的。至于我们常用的PC平台呢？新的Mac操作系统也内置了JVM，SUN也有为Linux推出Linux版的JDK，把Java视为头号敌人的微软，虽然不能再使用Java的标识，但其他家厂商如IBM、SUN一样会为Java推出Windows平台的解决方案，而且微软也推出了所谓JUMP的技术，可以把Java程序转换成.NET平台的C#程序，所以PC平台上Java的发展也是不成问题的。

未来，除了要更加强J2EE、J2SE和J2ME中的功能外，Java还会更进一步地结合XML，把现在外挂的XML套件也包含进Java标准套件中，那么对于跨平台来说将更是如虎添翼。Java的未来可以说是一片光明。

1.5 邂逅Java SE 6

你可能会对新的Java版本名称有些纳闷，以往熟悉的J2SE跑哪儿去了？是的，从去年开始，SUN对Java的版本命名做了很大的调整。以前我们一直习惯用Java 2（J2）做为Java三大平台的前缀，

^① 使用其他的文字也可以，但是不能使用一些特殊符号，我们将在第4章讨论。至于显示文字的方面，还是需要运行平台有能力显示其他语言的文字才行。

J2SE、J2ME和J2EE，这样的命名对初次接触Java的人来说很难理解，明明Java已经出到1.5版了，怎么叫Java 2呢？而J2SE这样的缩写也看不出跟Java的关系。新的命名方式直接解决了上述两个问题，三大平台的缩写叫做JavaSE、JavaME和JavaEE，而版本直接加在名称后面，这样一来不论是谁都可以很清楚知道这些缩写的意义了。那Java SE 6到底带来了哪些新的特点呢？官方网站或其他网络资源里杂七杂八有许多Java SE 6的特性，笔者简单归类成四大类。

兼容性、稳定性与性能的提升

这些特性几乎是每个Java SE版本的重点项目，因为面对强大的竞争对手（微软的.Net平台），Java惟有更稳定、运行性能更好，才不会被对手击败。在Java SE 6里有新的类文件验证机制（Class file verification）和新的类文件格式（Class file format），有助于启动性能的提升，并且降低恶意程序代码攻击的可能性。现今已经不再是Windows平台独大的天下了，越来越多商业化易用的Linux平台推出，Java SE 6也特地测试了各式的硬件与Linux操作系统，以便在不同的软硬件平台上，仍然保持Java Write Once, Run Anywhere的优良传统。更多兼容性的细节资料，可以参考Java官方网页上的说明：

<http://java.sun.com/javase/6/webnotes/compatibility.html>

功能上的加强

Java SE 6不像J2SE 5那样在Java的语法上有很大的更新，但针对API、Framework和工具上也有不少改进。大部分的改进是延伸自J2SE 5，如JMX、Annotation等。对于Windows平台的API支持也更完整，如系统托盘图标（System Tray Icon）、开机画面等。至于API也有不小的增加，我们常用的Collection、File等都有新的方法与类。API新增项目的详细说明，可以参考Java官方网页：

<http://java.sun.com/javase/6/webnotes/features.html>

开发程序更容易

这里特别是针对Annotation而言，在上一版J2SE 5中，Annotation是个全新且陌生的语法，J2SE 5让我们认识了它，却又感受不出它的强大威力，不过在Java SE 6里可要对它另眼看待了。除了有官方标准内置的APT（Annotation Processing Tool），以及最新的JDBC 4.0、JAXB 2.0、JAX-WS 2.0、JMX 1.3等API，而且完全利用了Annotation简易的特性，让原本一些繁杂的工作（程序代码），用Annotation的特殊tag来描述。大大简化了程序代码，并且增加了程序代码的可读性。不过这些内容超出本书讲解范围，大家还是先把基本的Annotation语法学好，等需进一步使用时，即可派上用场。

脚本语言的支持

脚本语言的支持可以说是Java SE 6最受瞩目的一项特性。拜Web 2.0所赐，脚本语言成为开发动态RIA（Rich Internet Application）最受欢迎的语言，主要是因为脚本语言有易学、易用、易开发、易运行的特点。像我们所熟知的Java Script、PHP、Python和目前最红的RoR（Ruby on Rails）等，都是脚本语言。Java SE 6可以让你运行这些脚本语言所写出来的程序，让你在做原型开发（Prototyping）、单元测试（Unit Testing）和动态网页上，更容易地开发与测试。

在语法上，Java SE 6没有新的语法；在兼容性上，Java SE 6兼容于之前的Java版本，且在不同的软硬件平台都能运行得很好；在性能上比起J2SE 5也有很大的提升。既然如此，我们为什么不换用Java SE 6呢？不要犹豫了，快去下载Java SE 6来感受它的威力吧！

1.6 本章总结

看完了对Java简单的介绍后，是不是对Java整个使用环境有更深入的了解了呢？不论你是程序设