

DVD
全书实例源代码；教学视频（包含全书知识点教学视频、全书实例教学视频）；全书完整的PPT教案；免费赠送5个全新综合案例的完整源码和实例教学视频

会当凌绝顶： Java开发修行实录

陈强 等编著

1. 循序渐进的学习主线

用引人入胜的情节，渗透作者数年学习和开发经历，由浅入深地讲解完整的编程技术知识体系，并达到学以致用的学习效果。

2. 场景式教学

生动的日记体裁，勾勒了一个程序员的成长历程。让读者在学习过程中置身其中，切换到编程思维，营造一种愉快的编程学习的氛围。

3. 精选案例，丰富的技巧提示

本书完全采用理论加实践的讲解模式，在每个知识点讲解之后，都用一个很有代表性的实例来演练知识点的用法，同时配以大量的技巧提示。

3. 用问题串联知识点，启发式讲解方式

生对学习进度，逐步提出问题，引发读者思考，环环相扣、前后关联，最终在动式地解决问题。书中提出了大量测试和问题，通过这些测试和问题将各个知识点串联起来。

5. 通俗的语言，大话的讲解模式

让读者仿佛是在看一个故事，而不是在学习深奥的编程语言知识。各处妙语，生动串起关键技术点。



会当凌绝顶： Java开发修行实录

陈强 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

Java 是目前使用最频繁的编程语言之一，它一直在开发领域中占据着重要的地位。本书循序渐进、由浅入深地介绍 Java 的核心技术，并通过具体实例的实现过程演练了各个知识点的具体使用流程。全书共 20 章，分为 4 篇。第 1~6 章是基础篇，介绍了 Java 语言开发所必需具备的基本知识，逐一讲解 Java 印象、搭建开发环境、Java 基本语法、假设语句、循环语句等领域的知识；第 7~12 章是核心技术篇，逐一讲解了数组、面向对象、类、异常处理、I/O 体系与文件处理、多线程等领域的基本知识；第 13~18 章是提高篇，逐一讲解了和网络接轨、AWT 界面开发、Swing 编程、Servlet、数据库编程等领域的知识；第 19、20 章是综合实战篇，分别通过记事本和打字游戏、综合供求信息系统项目的实现过程，讲解了 Java 语言在日常项目开发中的综合应用流程，并穿插介绍了求职和职场相处的技巧。全书采用故事性、趣味性相结合的对话讲解方式，并穿插了学习技巧和职场生存法则，引领读者全面掌握 Java。

本书不但适合 Java 初学者，也适合有一定 Java 基础的读者，甚至还可以作为有一定造诣的 Java 程序员的参考书。

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

会当凌绝顶：Java 开发修行实录 / 陈强等编著. —北京：电子工业出版社，2011.4
(编程·望岳)

ISBN 978-7-121-13001-4

I. ①会… II. ①陈… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 029187 号

策划编辑：田小康

责任编辑：李利健

印 刷：北京天宇星印刷厂

装 订：三河市皇庄路通装订厂

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：787×1092 1/16 印张：27.25 字数：697.6 千字

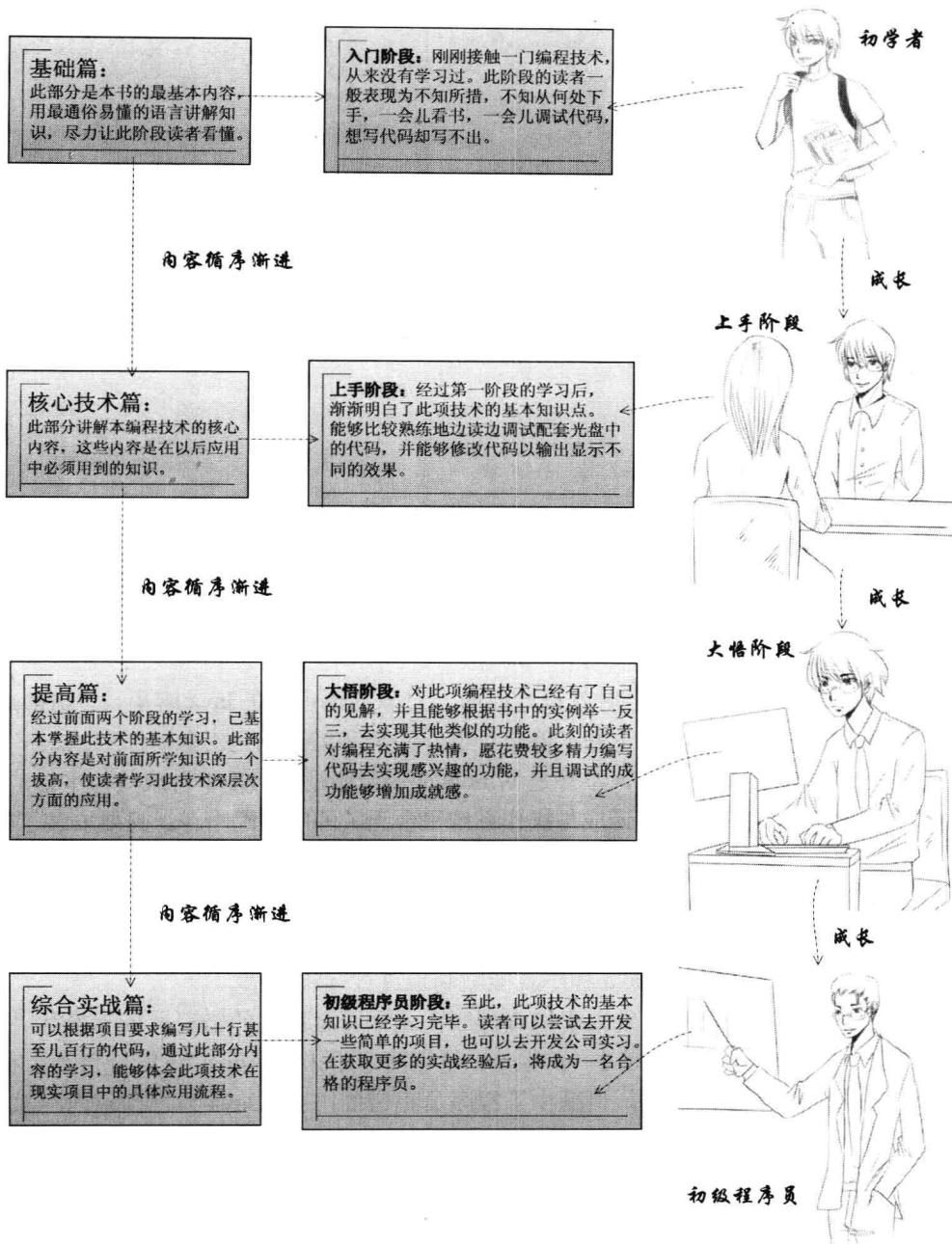
印 次：2011 年 4 月第 1 次印刷

印 数：4000 册 定价：59.80 元 (含 DVD 光盘 1 张)

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。



本书特色

本书的最大特色是用一问一答的模式穿插知识点，用故事性和趣味性的语言来讲解枯燥的编程知识。本书讲述了一个编程初学者“我”在老师 KNOWALL 的帮助下顺利掌握开发技术的成长历程。作者用日记的形式全程记录了学习历程，日记和生活密切相关，也和知识点

Contents

目 录

第 1 篇 基础篇

第 1 章 Java 之印象..... 1	2.4.3 玩转 NetBeans 21
1.1 引出问题..... 1	2.5 解决问题——探寻学习 Java 的捷径..... 23
1.2 Java 为何物..... 2	2.6 我的总结 24
1.2.1 探寻 Java 的起源..... 2	第 3 章 三种武器 25
1.2.2 看 Java 的特点..... 2	3.1 引出问题 25
1.3 我们需要准备什么 3	3.2 什么是量 26
1.4 开始体会 Java 程序之旅..... 7	3.2.1 永远都不变的常数 26
1.4.1 一个简单的 Java 程序..... 8	3.2.2 为变而生的量 27
1.4.2 编译和运行 Java 文件..... 9	3.3 数据类型 31
1.5 解决问题——谈 Java 的 江湖地位..... 10	3.3.1 数据类型的范围 32
1.6 我的总结 11	3.3.2 字符型..... 32
第 2 章 Java 宝剑出鞘 12	3.3.3 整型..... 34
2.1 引出问题..... 12	3.3.4 浮点型..... 35
2.2 Java 工具简介 13	3.3.5 布尔型..... 35
2.2.1 先看最流行的 Eclipse 13	3.4 标识符和关键字是一种身份 36
2.2.2 再看 NetBeans 13	3.5 解决问题 37
2.2.3 接着看 JBuilder 14	3.6 我的总结 37
2.3 开始步入 Eclipse 世界 15	第 4 章 Java 的三驾马车..... 38
2.3.1 获得并安装 Eclipse 15	4.1 引出问题 38
2.3.2 玩转 Eclipse 16	4.2 笑看 Java 运算符 39
2.4 使用 NetBeans..... 18	4.2.1 算术运算符..... 39
2.4.1 获取 NetBeans 19	4.2.2 关系运算符和逻辑运算符 40
2.4.2 装备 NetBeans 20	

4.2.3	位运算符	43	5.3.3	case 没有执行语句的情况	62
4.2.4	条件运算符	43	5.3.4	default 可以不在末尾	64
4.2.5	赋值运算符	44	5.4	进一步分析条件语句	64
4.3	表达式想对你说	45	5.4.1	正确使用 switch 语句	65
4.3.1	表达式为何物	45	5.4.2	正确使用 if 语句	66
4.3.2	表达式的优先级	45	5.4.3	有秩序的 switch	67
4.3.3	表达式的应用	46	5.5	解决问题	68
4.4	字符串	47	5.6	我的总结	69
4.4.1	字符串的初始化	48	第 6 章	如人生的循环语句	70
4.4.2	String 类	48	6.1	引出问题	70
4.4.3	StringBuffer 类	53	6.2	有多少语句可以循环	71
4.5	解决问题	55	6.2.1	简易 for 循环	71
4.6	我的总结	56	6.2.2	唠唠叨叨的 while 循环	75
第 5 章	Java 里的假设	57	6.2.3	do-while 循环语句	76
5.1	引出问题	57	6.3	跳转功能的实现	79
5.2	让 if 来选择你	58	6.3.1	break (终止)	79
5.2.1	if 控制语句	58	6.3.2	return (返回)	84
5.2.2	if 语句的延伸	59	6.3.3	continue (继续)	84
5.2.3	多个条件判断的 if 语句	60	6.3.4	随心所欲地使用跳转 语句之总结	86
5.3	有多个选择的 switch	60	6.4	解决问题	87
5.3.1	switch 语句的形式	60	6.5	我的总结	88
5.3.2	无 break 的 switch 语句	61			

第 2 篇 核心技术篇

第 7 章	数组是心灵的港湾	89	7.4.2	创建三维数组	98
7.1	引出问题	89	7.4.3	初始化三维数组	98
7.2	简单的一维数组	90	7.5	对数组的操作	99
7.2.1	声明一维数组	90	7.5.1	复制数组	100
7.2.2	创建一维数组	90	7.5.2	比较数组	100
7.2.3	轻松初始化一维数组	93	7.5.3	搜索数组中的元素	101
7.3	复杂点的二维数组	94	7.5.4	排序数组	102
7.3.1	声明二维数组	94	7.5.5	填充数组	103
7.3.2	二维数组的创建	95	7.6	解决问题	105
7.3.3	二维数组的初始化	96	7.7	我的总结	106
7.4	更复杂的多维数组	98	第 8 章	看 Java 如何面向对象	107
7.4.1	声明三维数组	98	8.1	引出问题	107

8.2	面向对象是什么	108	9.4	类的同胞——接口	144
8.2.1	理解面向对象的观念	108	9.4.1	定义接口	145
8.2.2	面向对象的特点	109	9.4.2	接口的量和方法	146
8.3	类是面向对象的第一特征	109	9.4.3	实现接口	149
8.3.1	如何编写一个类	110	9.4.4	引用接口	151
8.3.2	特殊的方法——构造方法	110	9.5	解决问题	153
8.3.3	一般的方法	111	9.6	我的总结	156
8.4	属性和方法的修饰符	112	第 10 章 异常也疯狂		157
8.4.1	公有的 public	113	10.1	引出问题	157
8.4.2	私有的 private	113	10.2	什么是异常	158
8.4.3	用 protected 来保护你	114	10.2.1	首次结识异常	158
8.4.4	其他修饰符	115	10.2.2	一些异常处理类	159
8.5	用 this 来指定你	116	10.3	知错就改的异常处理方式	160
8.6	演练类和对象	118	10.3.1	使用 try...catch 处理异常	160
8.6.1	创建和使用对象	118	10.3.2	处理多个异常	161
8.6.2	使用静态变量和静态方法	118	10.3.3	将 finally 关键字 使用在异常中	162
8.7	特殊的类——抽象类	119	10.4	异常处理的含蓄 方式——抛出	163
8.7.1	创建抽象类	120	10.4.1	使用 throws 将异常抛出	163
8.7.2	抽象类的规则	121	10.4.2	使用 throw 将异常抛出	165
8.8	Java 包不是钱包	123	10.5	自己来处理异常	166
8.8.1	定义包	123	10.6	异常处理的陋习	168
8.8.2	在 Eclipse 里定义包	124	10.6.1	异常丢弃	169
8.8.3	在程序里插入包	125	10.6.2	不指定具体的异常	170
8.9	解决问题	127	10.6.3	占着坑不用	171
8.10	我的总结	129	10.6.4	不说明异常的详细信息	171
第 9 章 类，我轻轻地来了		130	10.6.5	庞大的 try 块	172
9.1	引出问题	130	10.6.6	输出数据的不完整	172
9.2	类的继承	131	10.7	解决问题——勇敢地 面对错误	173
9.2.1	父与子	131	10.8	我的总结	174
9.2.2	调用父类的构造方法	132	第 11 章 I/O 体系与文件处理		175
9.2.3	可以随时访问父类的 属性和方法	134		给你带来额外的收获	
9.2.4	复杂的继承关系	137	11.1	引出问题	175
9.3	重载与重写	139	11.2	Java I/O 简介	176
9.3.1	再写一遍	139			
9.3.2	再处理一遍	140			
9.3.3	当重写与重载相逢	143			

11.3	黄河流	176	12.3.1	创建主线程	192
11.3.1	字节流	177	12.3.2	通过 Runnable 接口 创建线程	193
11.3.2	字符流	179	12.3.3	通过 Thread 类创建线程	195
11.4	加快 I/O 操作效率	181	12.4	创建多线程	196
11.4.1	缓冲字节流	182	12.5	线程的优先级	198
11.4.2	缓冲字符流	183	12.6	随心所欲地控制线程	201
11.5	文件处理	185	12.6.1	让当前的线程等待	201
11.5.1	文件类	185	12.6.2	让当前线程进入睡眠状态	205
11.5.2	使用文件类处理文件	186	12.6.3	让当前线程做出让步	208
11.6	解决问题	188	12.7	多线程同步	209
11.7	我的总结	189	12.7.1	线程同步的重要性	210
第 12 章	帮你多干活的多线程	190	12.7.2	轻松实现同步	211
12.1	引出问题	190	12.7.3	什么是死锁	213
12.2	线程基础	191	12.8	线程之间的交互	214
12.2.1	线程与进程的理解	191	12.9	解决问题	215
12.2.2	理解多线程	191	12.10	我的总结	217
12.3	创建线程	192			

第 3 篇 提高篇

第 13 章	和网络接轨	219	14.2	什么是 AWT	234
13.1	引出问题	219	14.3	创建窗口	235
13.2	一个服务器端程序	220	14.4	创建窗口组件	235
13.3	什么是网络通信	224	14.5	布局利器	237
13.3.1	TCP 和 IP 概述	224	14.5.1	布局武器 FlowLayout	237
13.3.2	使用 URL 进行网络链接	225	14.5.2	布局武器 BorderLayout	239
13.3.3	编写 URL 程序常用的 方法	226	14.5.3	布局利器 GridLayout	240
13.4	和网络编程一起笑看风云	227	14.5.4	布局利器 CardLayout	242
13.4.1	创建 socket	228	14.5.5	布局利器 Null	242
13.4.2	有多个客户端连接的情况	230	14.6	编写监听接口	244
13.5	解决问题——互联网给 我们的学习带来了什么	231	14.6.1	窗口监听的接口	245
13.6	我的总结	232	14.6.2	按钮监听的接口	246
			14.6.3	文本框监听的接口	247
第 14 章	AWT 的奇幻世界	233	14.7	解决问题	249
14.1	引出问题	233	14.8	我的总结	252
			第 15 章	Swing 更加绚丽	253
			15.1	引出问题	253

15.2	Swing 创建窗口	253	16.4.5	会话	296
15.2.1	JFrame 简介和方法	254	16.4.6	Servlet 上下文	299
15.2.2	创建第一个 Swing 窗口	254	16.4.7	Servlet 监听	300
15.3	Icon 接口	256	16.5	Servlet 过滤器	301
15.4	添加组件	259	16.5.1	过滤器的简介	301
15.4.1	弹出式菜单	259	16.5.2	过滤器的应用	302
15.4.2	文本框	262	16.5.3	Servlet 过滤器的配置	303
15.4.3	菜单	266	16.5.4	Servlet 过滤器的实例	303
15.4.4	单选按钮	267	16.6	将请求进行封装	305
15.4.5	复选框按钮	270	16.6.1	doGet	305
15.4.6	列表框	271	16.6.2	doPost	307
15.4.7	选项卡	273	16.7	解决问题	309
15.4.8	文本域	274	16.8	我的总结	311
15.4.9	按钮	276	第 17 章	轻松玩转数据库	313
15.4.10	进度条	277	17.1	引出问题	313
15.5	常用的布局管理器	278	17.2	数据库的类别	313
15.5.1	当不使用布局管理器时	278	17.3	安装 MySQL	314
15.5.2	体会边界布局管理器	279	17.4	MySQL 的管理工具	317
15.5.3	体会网格布局管理器	280	17.4.1	创建数据库	318
15.6	解决问题	281	17.4.2	创建表	318
15.7	我的总结	283	17.4.3	输入记录	318
第 16 章	Servlet 很神奇	284	17.5	SQL Server 很简单	319
16.1	引出问题	284	17.5.1	创建数据库	320
16.2	什么是 Servlet	284	17.5.2	创建表	321
16.2.1	Servlet 技术简述	284	17.5.3	创建记录	323
16.2.2	Servlet 技术特性	285	17.6	解决问题	323
16.2.3	Servlet 的生命周期	286	17.7	我的总结	326
16.3	如何开发一个简单的 Servlet	286	第 18 章	数据库编程是最重要的形式	327
16.3.1	开发一个简单的 Servlet	286	18.1	引出问题	327
16.3.2	编译并发布 Servlet	287	18.2	JDBC 是一座桥	327
16.4	Servlet 的常用接口和类	290	18.2.1	JDBC API	328
16.4.1	与 Servlet 配置相关的接口	290	18.2.2	JDBC 驱动类型	328
16.4.2	Servlet 编程接口	291	18.3	数据库的连接	329
16.4.3	HTTPServlet 类	293	18.3.1	轻松连接 MySQL	330
16.4.4	请求与响应	296	18.3.2	轻松连接 SQL Server 2000	333

18.4 SQL 语句才是核心	337	18.4.3 数据库操作语句	339
18.4.1 新建数据库表	337	18.5 解决问题	342
18.4.2 数据库查询语句	338	18.6 我的总结	343

第 4 篇 综合实战篇

第 19 章 简单的记事本	345	第 20 章 综合供求信息系统	374
19.1 老师的作业	345	20.1 求职路上	374
19.2 Java API 手册	345	20.2 系统需求	376
19.2.1 什么是 Java API	345	20.2.1 开发背景	376
19.2.2 如何获得 Java API 手册	346	20.2.2 需求分析	377
19.2.3 使用 Java API 手册	347	20.3 网站功能结构	377
19.3 记事本的编写	348	20.4 规划流程	378
19.3.1 插入包	348	20.5 搭建数据库	378
19.3.2 新建文件和打开文件	348	20.5.1 数据库需求分析	378
19.3.3 保存文件	349	20.5.2 数据库概念设计	379
19.3.4 文件操作	350	20.5.3 数据库逻辑结构设计	380
19.3.5 字体的处理	351	20.6 设计公共类	381
19.3.6 状态栏和帮助功能	353	20.6.1 数据库连接及操作类	381
19.3.7 将功能实例化	355	20.6.2 业务处理类	383
19.4 打字游戏的编写	358	20.6.3 分页类	387
19.4.1 需要插入的包	358	20.6.4 字符串处理类	389
19.4.2 设计版式	358	20.7 开始编码	389
19.4.3 声音的处理	359	20.7.1 设计前台页面	390
19.4.4 游戏过关的设计	360	20.7.2 前台信息显示设计	390
19.4.5 文字的下移	362	20.7.3 信息发布模块设计	402
19.5 计算器的编写	365	20.8 后期的编码工作	406
19.5.1 计算器的最终结果	365	20.8.1 登录模块	406
19.5.2 计算器组件的编写	365	20.8.2 后台页面模块	409
19.5.3 计算器的按钮事件	368	20.8.3 后台信息管理模块	410
19.6 系统调试	372	20.9 测试	420
		20.10 我的总结	421

第 1 篇 基础篇

第 1 章 Java 之印象

前些年，张艺谋拍了一部宣传片《印象西湖》，这部宣传片曾风靡大江南北，可谓是给没去过杭州西湖或者初次去西湖的人留下了深刻的印象。Java 是什么呢？在打开本书之前，它又留给你什么样的印象？为什么让你在众多的语言中选择并学习这么优秀的技术呢？它能带来什么呢？这些疑惑在本章将会为读者朋友讲解。学完本章后，初学者对 Java 应有一个深层次的印象。

1.1 引出问题

9月9日，7:00，晴

秋高气爽，天空依然很蓝！今天是我升入大学后的第八天，此时我的心里还是充满了各种新奇：学校的图书馆很大，师兄师姐们的毕业产品很了不起……我带着新奇的心情领来了自己的教材。

我：“我领回教材了，竟然发现有 Java 语言！上中学时我就听说了 C 语言和 Visual Basic，后来又听说了 C++、Java，我一直以为它们和 JavaScript 脚本语言差不多，现在我才知道原来 Java 是一门独立的语言。现在学 Java 还有用吗？”

KNOWALL：“我只能很负责地对你说，Java 是当今使用最频繁的编程语言之一，并且一直位列前三甲！”

我：“Java 真有这么厉害吗？”

KNOWALL：“Java 吸取了 C 语言和 C++ 的精华，遵循了面向对象思想，是一种高级语言。我们来看看主流编程语言之间的关系，如图 1-1 所示。”

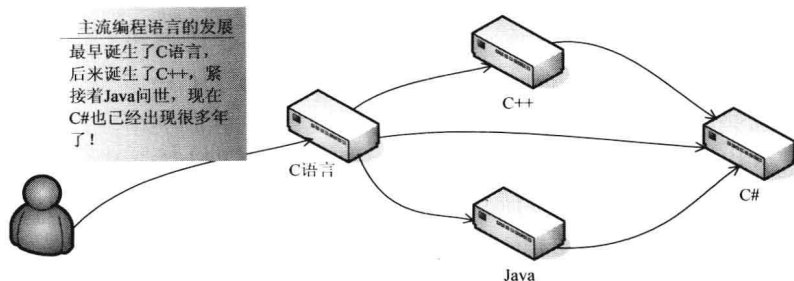


图 1-1 主流编程语言的发展

1.2 Java 为何物

我：“Java 真的有那么神奇吗？”

KNOWALL：“Java 语言当然十分优秀，它颠覆了整个程序世界。让风靡一时的 C++、Basic 都感到了它的光辉，Java 在 Web 应用、桌面应用和服务器等领域都有很深的造诣。”

1.2.1 探寻 Java 的起源

9月9日，7:10，晴

我：“既然 Java 源于 C，那么 Java 肯定是在 C 语言之后诞生的。”

KNOWALL：“Java 从 1995 年 5 月推出，到 2009 年 4 月 20 日被 Oracle 收购，直到现在，有 15 年的历史了！具体发展历程如图 1-2 所示。”

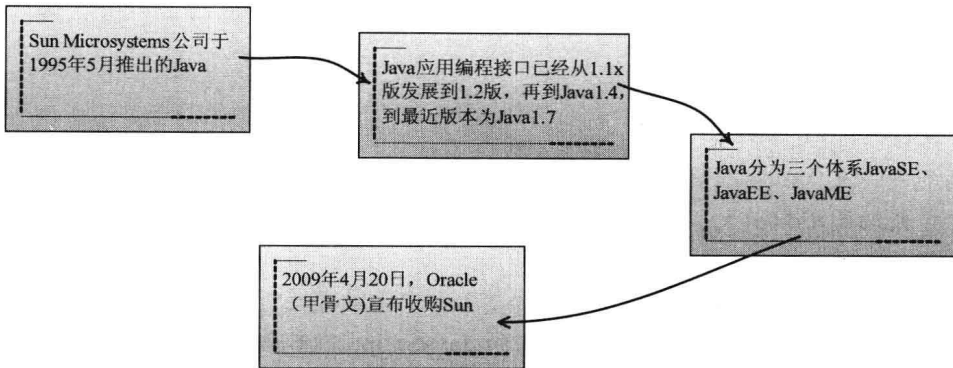


图 1-2 Java 的发展历程

1.2.2 看 Java 的特点

Java 是一种简单的、面向对象的、分布式的、解释型的、健壮安全的、结构中立的、可移植的、性能优异和多线程的动态语言。当 1995 年 Sun 公司推出 Java 语言之后，全世界的目光都被这个神奇的语言所吸引。现在的 Java 主要有如下特性。

(1) **Java 语言是简单的：**一方面，Java 语言的语法与 C 语言和 C++ 语言很接近，使得大多数程序员很容易学习和使用 Java。另一方面，Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的那些特性，如操作符重载、多继承、自动的强制类型转换。特别地，Java 语言不使用指针，并提供了自动的废料收集，使得程序员不必为内存管理而担忧。

(2) **Java 语言是一个面向对象的：**Java 语言提供类、接口和继承等原语，为了简单起见，只支持类之间的单继承，但支持接口之间的多继承，并支持类与接口之间的实现机制（关键字为 implements）。Java 语言全面支持动态绑定，而 C++ 语言只对虚函数使用动态绑定。总之，Java 语言是一个纯的面向对象的程序设计语言。

(3) **Java 语言是分布式的**: Java 语言支持 Internet 应用的开发, 在基本的 Java 应用编程接口中有一个网络应用编程接口(Java.NET), 它提供了用于网络应用编程的类库, 包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI (远程方法激活) 机制也是开发分布式应用的重要手段。

(4) **Java 语言是健壮的**: Java 的强类型机制、异常处理、废料的自动收集等是 Java 程序健壮性的重要保证。对指针的丢弃是 Java 的明智选择。Java 的安全检查机制使得 Java 更具健壮性。

(5) **Java 语言是安全的**: Java 通常被用在网络环境中, 为此, Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外, Java 对通过网络下载的类型有一个安全防范机制(类 ClassLoader), 如分配不同的名字空间以防替代本地的同名类、字节代码检查, 并提供安全管理机制(类 SecurityManager) 让 Java 应用设置安全哨兵。

(6) **Java 语言是可移植的**: 这种可移植性来源于体系结构中立性, 另外, Java 还严格规定了各个基本数据类型的长度。Java 系统本身也具有很强的可移植性, Java 编译器是用 Java 实现的, Java 的运行环境是用 ANSI C 实现的。

(7) **Java 语言是解释型的**: 如前所述, Java 程序在 Java 平台上被编译为字节码格式, 然后可以在实现这个 Java 平台的任何系统中运行。在运行时, Java 平台中的 Java 解释器对这些字节码进行解释执行, 执行过程中需要的类在链接阶段被载入到运行环境中。

(8) **Java 是高性能的**: 与那些解释型的高级脚本语言相比, Java 的确是高性能的。事实上, Java 的运行速度随着 JIT (Just-In-Time) 编译器技术的发展, 越来越接近于 C++。

(9) **Java 语言是多线程的**: 在 Java 语言中, 线程是一种特殊的对象, 它必须由 Thread 类或其子(孙)类来创建。通常有两种方法来创建线程: 其一, 使用型构为 Thread (Runnable) 的构造子将一个实现了 Runnable 接口的对象包装成一个线程; 其二, 从 Thread 类派生出子类并重写 run 方法, 使用该子类创建的对象即为线程。值得注意的是, Thread 类已经实现了 Runnable 接口, 因此, 任何一个线程均有它的 run 方法, 而 run 方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。Java 语言支持多个线程同时执行, 并提供多线程之间的同步机制(关键字为 synchronized)。

(10) **Java 语言是动态的**: Java 语言的设计目标之一是适应于动态变化的环境。Java 程序需要的类能够动态地被载入到运行环境, 也可以通过网络来载入所需要的类。这也有利于软件的升级。另外, Java 中的类有一个运行时刻的表示, 能进行运行时刻的类型检查。

1.3 我们需要准备什么

9月9日, 7:30, 晴

我: “开始编写程序之前, 我得先安装一种开发工具吧?”

KNOWN: “是的, 任何高级语言都有与之想配合的开发工具。但是我们现在先不说开发工具, 在安装开发工具之前必须先搭建 Java 开发环境。”

我: “Java 的开发环境有很多吗?”

KNOWALL：“Java 在各个领域都发挥得不错，所以开发环境也有多个，这得根据你的需求来选择对应的环境。”

我：“选择开发环境后，才能安装开发工具吗？”

KNOWALL：“是的。简单地说，JDK 是 Java 的工作环境，JRE 是 Java 程序的运行环境，IDE 是集成开发平台，例如 Eclipse、Lomboz 等，具体如图 1-3 所示。”

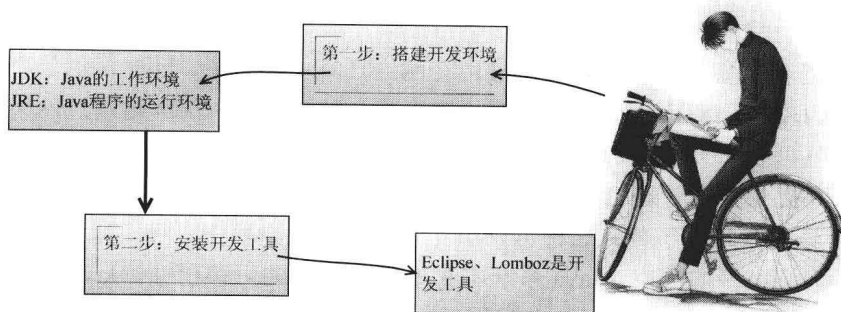


图 1-3 Java 开发环境和工具

第 1 步：在使用 Java 开发之前，需要先搭建其开发环境，即需要 JRE 的支持。在 Windows 上安装 JRE/JDK 非常简单，首先在 Sun 官方网站下载，网址为 <http://developers.sun.com/downloads/>，如图 1-4 所示。

第 2 步：在图 1-4 中可以看到有很多版本，运行 Eclipse 时，虽然只需要 JRE 就可以了，但是在开发 Android 应用程序的时候，需要完整的 JDK（JDK 已经包含了 JRE），且要求其版本在 1.5 以上，这里选择 Java SE (JDK) 6，其下载页面如图 1-5 所示。

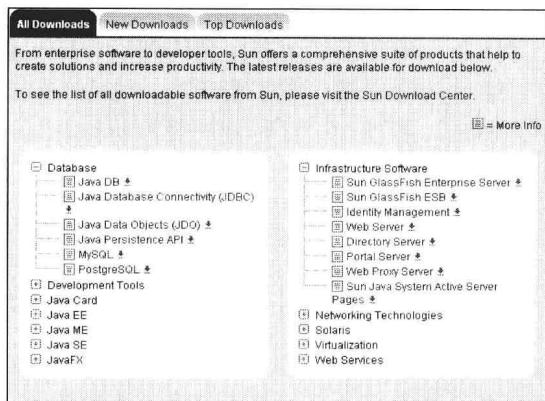


图 1-4 Sun 官方下载页面

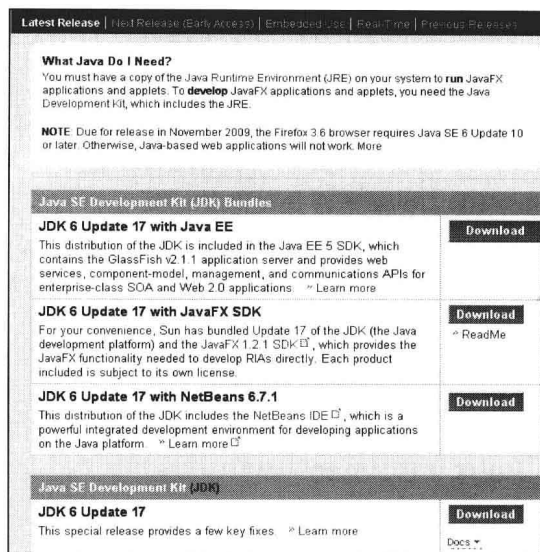


图 1-5 JDK 下载页面

第 3 步：在图 1-5 中找到“JDK 6 Update 13”，单击其右侧的“Download”按钮，出现