

21世纪高等教育计算机规划教材

COMPUTER

Java 面向对象 程序设计

Java Object-Oriented Programming

梁燕来 程裕强 主编

- 采用最新的 Java 7 版本，含最新的 Java 特性
- 所有程序均在 Windows 和 Linux 平台下调试通过
- 讲解 JDBC 访问 MySQL 和 Oracle 数据库知识
- 增加各大公司面试题，拓宽学生视野，面向就业
- 提供实时交流平台，第一时间解答读者的问题



013023659

TP312JA-43
274

21世纪高等教育计算机规划教材

COMPUTER

Java 面向对象 程序设计

Java Object-Oriented Programming

梁燕来 程裕强 主编



TP312JA-43
274



北航

C1630505

人民邮电出版社

北京

042850810

图书在版编目 (CIP) 数据

Java面向对象程序设计 / 梁燕来, 程裕强主编. —
北京: 人民邮电出版社, 2013.4
21世纪高等教育计算机规划教材
ISBN 978-7-115-30939-6

I. ①J… II. ①梁… ②程… III. ①
JAVA语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字 (2013) 第033702号

内 容 提 要

本书根据最新的 Java 7 版本, 由浅入深地讲解了面向对象程序设计语言的主要内容, 并介绍最新的 Java 技术。所有样例程序均在 Windows 平台和 Linux 平台的 JDK 7 环境下调试通过。每章习题中添加了近年各大 IT 公司的 Java 面试题, 引导学生面向就业市场。本书主要内容包括 Windows 平台和 Linux 平台下的 Java 开发环境搭建、面向过程编程、类与对象、继承、多态、接口、面向对象思想、异常处理、Java API、泛型与集合框架、Java IO、多线程、JDBC 操作 MySQL 和 Oracle 数据库、反射机制、Annotation 以及 Eclipse 集成开发工具介绍等内容, 并对已经过时的 Applet 和图形界面编程以及网络编程做了简单的介绍。

本书可以作为高等学校计算机专业以及相关专业的“面向对象程序设计”课程教材, 也可作为 Java 软件开发的参考书。

21 世纪高等教育计算机规划教材

Java 面向对象程序设计

-
- ◆ 主 编 梁燕来 程裕强
责任编辑 李海涛
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 20 2013 年 4 月第 1 版
字数: 549 千字 2013 年 4 月北京第 1 次印刷

ISBN 978-7-115-30939-6

定价: 39.80 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

前 言

近几年，国内高校的“面向对象程序设计”课程越来越多地选择 Java 语言。纵观国内 Java 教材，大多数内容陈旧，脱离就业市场，比如一些过时或者没有实用价值的技术（Applet、AWT 和 Swing 图形界面编程、JDBC-ODBC）在教材中占了很大篇章，反而忽略了关键的面向对象的思想，本末倒置。本教材特点：采用最新的 Java 7 版本；紧扣就业市场，含近年 Java 企业面试题；有真正的跨平台性，演示了 Windows 平台和 Linux 平台下的 Java 开发过程；跨平台安装 MySQL 和 Oracle 数据库，通过 JDBC 方法访问数据库；删减 Java 过时技术，增加 Java 新的实用技术。

本书共分为 16 章。第 0 章对计算机和 Java 语言做了简单介绍。第 1 章详细讲解 Windows 平台和 Linux 平台下 Java 开发环境的配置。第 2 章讲解 Java 的基本数据类型、控制语句、数组、枚举、字符串和方法等。第 3 章、第 4 章和第 5 章是面向对象程序设计的核心内容，重点讲解类、对象、继承、多态、接口、面向对象设计的基本原则，以及简单的设计模式。第 6 章是面向对象程序设计非常实用的异常处理技术。第 7 章和第 8 章，讲解如何查看 Java 文档以及常用的 Java 核心类库，重点介绍字符串的常见处理操作以及加密算法。第 9 章引入了泛型机制，并对集合框架做了讲解。第 10 章通过讲解文件上传原理来解释复杂的 IO 部分。第 11 章介绍多线程基本概念，并对线程同步做了详细介绍。第 12 章是 Java 语言的落脚点，分别演示了在 Windows 平台和 Linux 平台下通过 JDBC 对 MySQL 数据库的访问，通过简单示例演示了 JDBC 技术；介绍了在 Windows 平台和 Linux 平台下 Oracle 数据库的安装配置以及通过 JDBC 访问 Oracle 数据库。第 13 章讲解 Java 高级特性，反射与 Annotation 是 Java EE 开发的重要技术。为了本书的完整性，第 14 章用较少的篇幅，通过简单实例介绍 Applet、图形界面编程以及网络编程。第 15 章对 Eclipse 集成开发工具做了介绍。本书所有样例程序均在 Windows 平台和 Linux 平台下的 JDK 7 环境，通过命令方式调试通过。本书的实例代码、相关软件以及电子教案均可到人民邮电出版社教学服务与资源网下载（<http://www.ptpedu.com.cn>），或者与编者联系索取。

本书由梁燕来、程裕强主编，主要参编人员有周锦芳、周国军、李昌志、肖宝、刘永健、庞伟等。编者均来自教学第一线，具有丰富的教学实践经验和软件开发经验。具体编写分工如下：第 0 章、第 2 章和第 6 章由梁燕来编写，第 1 章、第 8 章、第 12 章和第 13 章由程裕强编写，第 3 章和第 4 章由周锦芳编写，第 5 章和第 11 章由周国军编写，第 14 章和第 15 章由李昌志编写，第 7 章由肖宝编写，第 9 章由刘永健编写，第 10 章由庞伟编写。全书由梁燕来和程裕强统稿和审定。

由于 Java 技术发展迅速，加上编者水平有限，书中难免出现不当之处甚至是错误，希望各位读者指正，编者邮箱为 hadron.cheng@foxmail.com。为了方便各位教师与编者联系，请加入 Java 教学 QQ 群 194993200；读者朋友可以加入 Java 群 187449470。对大家的问题，我们将在第一时间解答。

编 者

2013 年 1 月

目 录

第 0 章 概述1

- 0.1 计算机概述1
- 0.2 计算机语言简史2
- 0.3 Java 语言简史3
- 0.4 Java 运行机制4
 - 0.4.1 Java 虚拟机4
 - 0.4.2 运行机制4
- 0.5 名词解释5
- 0.6 习题5

第 1 章 Java 开发环境配置6

- 1.1 Windows 平台 Java 环境配置6
 - 1.1.1 下载 Java 开发包 JDK6
 - 1.1.2 安装 JDK7
 - 1.1.3 环境变量配置9
 - 1.1.4 Java 环境测试10
 - 1.1.5 安装代码编辑器11
- 1.2 Linux 平台 Java 环境配置11
 - 1.2.1 Linux 简介11
 - 1.2.2 配置 Java 开发环境11
 - 1.2.3 Java 环境测试12
- 1.3 第一个 Java 程序13
 - 1.3.1 HelloWorld13
 - 1.3.2 在 Windows 下开发 Java 程序14
 - 1.3.3 在 Linux 下开发 Java 程序16
 - 1.3.4 跨平台操作17
- 1.4 集成开发环境18
- 1.5 习题18

第 2 章 面向过程编程19

- 2.1 面向过程介绍19
- 2.2 标识符20
- 2.3 注释20
- 2.4 基本数据类型20

- 2.4.1 整数型21
- 2.4.2 小数型23
- 2.4.3 字符型23
- 2.4.4 逻辑型24
- 2.4.5 基本类型的转换25
- 2.5 变量25
- 2.6 基本运算26
 - 2.6.1 算术运算26
 - 2.6.2 赋值运算27
 - 2.6.3 自运算28
 - 2.6.4 比较运算28
 - 2.6.5 逻辑运算28
 - 2.6.6 三目运算29
 - 2.6.7 位运算30
 - 2.6.8 运算优先级31
- 2.7 简单输入输出32
 - 2.7.1 输入语句32
 - 2.7.2 输出语句32
- 2.8 流程控制33
 - 2.8.1 顺序结构33
 - 2.8.2 选择结构33
 - 2.8.3 循环结构37
 - 2.8.4 控制转移41
- 2.9 循环设计44
 - 2.9.1 穷举法44
 - 2.9.2 递推法45
 - 2.9.3 迭代法47
- 2.10 数组49
 - 2.10.1 数组定义50
 - 2.10.2 数组操作51
 - 2.10.3 简单应用52
 - 2.10.4 二维数组52
 - 2.10.5 for 与数组53
- 2.11 字符串54
- 2.12 枚举55

2.12.1 枚举定义	55	3.12 内部类	83
2.12.2 枚举与 for 循环	55	3.13 包	84
2.12.3 枚举与 switch	56	3.13.1 package	84
2.13 方法	57	3.13.2 import	85
2.13.1 方法定义	57	3.14 反编译	87
2.13.2 递归机制	57	3.15 应用	87
2.13.3 两类参数	60	3.15.1 栈 Stack	88
2.13.4 可变参数	61	3.15.2 队列 Queue	89
2.14 习题	62	3.15.3 链表 LinkedList	90
第 3 章 类与对象	64	3.16 习题	94
3.1 从 struct 到 class	64	第 4 章 继承与多态	95
3.2 类	66	4.1 引例	95
3.2.1 类的定义	66	4.2 继承机制	97
3.2.2 属性	67	4.3 super	98
3.2.3 方法	67	4.4 对象构造过程	99
3.2.4 局部变量	68	4.5 继承性规则	100
3.2.5 封装	68	4.5.1 同包	100
3.3 对象	68	4.5.2 不同包	102
3.3.1 定义对象	69	4.6 instanceof	102
3.3.2 访问对象成员	69	4.7 成员覆盖	103
3.4 Java 内存结构	69	4.7.1 属性隐藏	103
3.5 构造器	70	4.7.2 方法重写	104
3.6 对象生命周期	71	4.8 final	106
3.7 this 关键字	71	4.8.1 final 属性	106
3.8 方法重载	72	4.8.2 final 方法	107
3.8.1 为什么需要重载	72	4.8.3 final 参数	107
3.8.2 方法签名	73	4.8.4 final 类	107
3.8.3 重载定义	73	4.9 多态性	108
3.8.4 两大误区	73	4.9.1 多态问题	108
3.9 权限符	74	4.9.2 多态概述	109
3.9.1 public	74	4.9.3 类型多态	109
3.9.2 private	74	4.9.4 方法多态	111
3.9.3 JavaBean	77	4.10 抽象类	112
3.10 static	77	4.11 接口	114
3.10.1 C 语言的 static	78	4.11.1 接口思想	114
3.10.2 static 成员	78	4.11.2 接口定义	114
3.10.3 静态导入	79	4.11.3 实现接口	115
3.10.4 自定义 Math 类	80	4.11.4 接口与多态	116
3.11 代码块	83	4.11.5 接口与抽象类	117

4.12	Object 类	119	7.5.1	BigInteger	159
4.13	改进 Stack、Queue 和 LinkList	120	7.5.2	BigDecimal	160
4.14	匿名类	124	7.6	DecimalFormat 类	161
4.15	习题	126	7.6.1	格式化整数和小数位	161
第 5 章	面向对象思想	127	7.6.2	整数位分组	161
5.1	UML 简介	127	7.6.3	格式化字符串转换为数值	162
5.1.1	结构图	128	7.6.4	综合实例	162
5.1.2	关系图	128	7.7	Arrays 类	163
5.2	开闭原则	129	7.7.1	复制数组	163
5.3	多少原则	132	7.7.2	排序	163
5.4	高低原则	134	7.7.3	二分查找	164
5.5	设计模式简介	134	7.8	时间日期型	165
5.5.1	单例模式	134	7.8.1	Date、SimpleDateFormat	165
5.5.2	工厂模式	135	7.8.2	Calendar	165
5.5.3	适配器模式	142	7.9	System 类	166
5.6	习题	144	7.9.1	System API	166
第 6 章	异常处理	145	7.9.2	记录程序运行时间	167
6.1	引例	145	7.9.3	程序退出	167
6.2	异常类	146	7.9.4	程序运行时环境信息	168
6.3	异常处理机制	146	7.10	习题	168
6.3.1	try-catch	146	第 8 章	字符串	169
6.3.2	try-catch-finally	147	8.1	String 的 API	169
6.4	抛出异常	148	8.2	String 对象	170
6.4.1	throws	148	8.2.1	字符串常量	170
6.4.2	throw	149	8.2.2	构造字符串对象	171
6.5	异常处理的缺点	149	8.2.3	字符串引用	171
6.6	断言	149	8.2.4	简单实例	172
6.7	习题	150	8.3	字符串处理	172
第 7 章	Java API	151	8.3.1	遍历字符串	172
7.1	官方文档	151	8.3.2	字符串相等	173
7.2	包装类	153	8.3.3	字符串比较	174
7.3	Math 类	154	8.3.4	字符串连接	175
7.4	Random 类	156	8.3.5	字符串查询	175
7.4.1	Random API	156	8.3.6	字符串替换	177
7.4.2	区间随机数	156	8.3.7	去除空格	177
7.4.3	概率问题	158	8.3.8	取子串	178
7.5	大数类	159	8.3.9	分割字符串	178
			8.3.10	大小写转换	179
			8.4	字符串与数值的相互转化	179

8.5 正则表达式	180
8.5.1 初识正则表达式	180
8.5.2 规则定义	180
8.5.3 matches 方法	181
8.5.4 Pattern 和 Matcher 类	182
8.5.5 再看 split()方法	183
8.6 字符串与数组	184
8.6.1 字符串与字符数组	184
8.6.2 字符串与字节数组	185
8.6.3 最熟悉的陌生人	185
8.6.4 统计字符	186
8.7 加密算法	186
8.7.1 对称加密算法	186
8.7.2 非对称加密算法	188
8.7.3 单向加密算法	190
8.8 阅读 String 类源代码	191
8.9 StringBuffer 类	195
8.10 StringTokenizer 类	196
8.11 Scanner 类	196
8.12 习题	197
第 9 章 泛型与集合框架	198
9.1 泛型	198
9.1.1 为什么需要泛型	198
9.1.2 引入泛型	199
9.1.3 类型通配符	200
9.1.4 泛型上限	201
9.2 改进数据结构类	202
9.2.1 Stack	202
9.2.2 Queue	203
9.2.3 LinkedList	205
9.3 集合框架	207
9.3.1 Collection	207
9.3.2 Iterator	208
9.3.3 Set	209
9.3.4 List	212
9.3.5 Map	216
9.4 习题	220

第 10 章 Java IO	221
10.1 文件	221
10.1.1 文件系统	221
10.1.2 File 类	222
10.1.3 Windows 平台下的文件	222
10.1.4 Linux 平台下的文件	223
10.1.5 文件跨平台性	223
10.1.6 Scanner 解析文件	224
10.2 Stream	225
10.3 InputStream	225
10.4 OutputStream	226
10.5 文件上传原理	227
10.6 字符流	228
10.7 Serializable	229
10.8 习题	229
第 11 章 多线程	230
11.1 基本概念	230
11.2 创建多线程	230
11.2.1 Thread	231
11.2.2 Runnable	231
11.2.3 简单应用	232
11.3 线程详解	234
11.3.1 线程方法	234
11.3.2 线程生命周期	234
11.3.3 示例	235
11.4 线程安全	238
11.4.1 临界资源	238
11.4.2 互斥	238
11.4.3 同步	239
11.5 守护线程	242
11.6 习题	242
第 12 章 JDBC	243
12.1 数据库	244
12.1.1 数据库概论	244
12.1.2 在 Windows 平台下 配置 MySQL	244
12.1.3 在 Linux 平台下配置 MySQL	251

12.2 SQL 语句.....	256	13.4 内置的 Annotation	283
12.3 JDBC	259	13.4.1 @Override	284
12.3.1 JDBC 简介	259	13.4.2 @Deprecated	284
12.3.2 驱动程序	259	13.4.3 @SupressWarnings	284
12.4 JDBC API	259	13.5 自定义 Annotation	285
12.4.1 DriverManager	260	13.5.1 自定义简单的 Annotation	285
12.4.2 Connection	260	13.5.2 @Target	285
12.4.3 Statement	260	13.5.3 @Retention	286
12.4.4 ResultSet	261	13.5.4 自定义 Annotation	286
12.4.5 小结	261	13.5.5 通过反射访问 Annotation	287
12.5 数据库连接	261	13.6 习题	287
12.5.1 存放驱动程序	261		
12.5.2 建立数据库连接	262	第 14 章 保留部分	288
12.6 数据操作	262	14.1 Applet	288
12.6.1 简单查询	262	14.2 图形界面	289
12.6.2 批量查询	263	14.3 网络编程	291
12.6.3 插入数据	264	14.4 习题	292
12.6.4 删除数据	264		
12.6.5 更新数据	265	第 15 章 集成开发工具	293
12.7 数据库操作类	266	15.1 集成开发工具介绍	293
12.8 连接 Oracle	267	15.2 Windows 平台下的 Eclipse 配置	293
12.8.1 在 Windows 平台下配置 Oracle 数据库	268	15.2.1 Eclipse 下载	293
12.8.2 在 Linux 平台下配置 Oracle 数据库	270	15.2.2 Eclipse 配置	294
12.8.3 建立连接	272	15.3 Linux 平台下的 Eclipse 配置	295
12.8.4 单表查询	272	15.4 创建 Java 项目	296
12.8.5 多表查询	273	15.5 在 Java 项目中新建类	298
12.8.6 改进数据库操作类	274	15.6 生成 getter 和 setter	300
12.9 习题	275	15.7 为 Java 项目创建包	302
		15.8 开发 JDBC 应用程序	304
		15.8.1 为 Java 项目添加驱动程序	304
		15.8.2 创建 JDBC 应用程序	305
第 13 章 反射与 Annotation	276	15.9 程序调试	306
13.1 引例	276	15.9.1 显示行号	307
13.2 反射机制	277	15.9.2 设置断点	307
13.2.1 Class 类	277	15.9.3 进入 Debug 模式	308
13.2.2 实例化 Class 对象	280	15.9.4 单步执行	309
13.2.3 访问构造器	280	15.9.5 查看变量值	309
13.2.4 访问属性	281	15.9.6 结束 Debug	309
13.2.5 访问方法	282		
13.3 Annotation 介绍	283	参考文献	310

第 0 章

概述

本章重要性：★★

本章学习目标：

- ◇ 了解计算机组成与基本工作原理
- ◇ 了解计算机语言发展史
- ◇ 了解 Java 语言发展史
- ◇ 掌握 Java 运行机制
- ◇ 掌握 Java 相关名称

为什么是第 0 章，而不是第 1 章呢？因为 Java 语言与 C 语言一样，数组等下标都是从 0 开始的。从数学角度看，第一个自然数是 0，数组下标从 0 开始也是科学的。

0.1 计算机概述

计算机 (Computer)，全名是电子计算机，俗称电脑。从 1946 年第一台电子计算机 ENIAC 诞生，其发展大致经历了电子管计算机、晶体管计算机、集成电路计算机、大规模集成电路计算机四个阶段，冯·诺伊曼（见图 0.1）因提出计算机基本工作原理被称为计算机之父。

图 0.2 所示为计算机的组成与基本工作原理。

(1) 计算机的组成。计算机由控制器、运算器、存储器、输入设备、输出设备五大部件组成。

(2) 二进制。程序和数据以二进制形式存储在存储器中。

(3) 存储程序原理。程序由指令组成，并和数据一起存放在存储器中，计算机启动后，能自动地按照程序指令的逻辑顺序逐条把指令从存储器中读出来，自动完成由程序所描述的处理工作。

没有安装软件的计算机硬件一般称为裸机，裸机就像一堆废铁，任何事都做不了（植物人）。如图 0.3 所示，在裸机上安装第一层软件，也就是操作系统 OS (Operating System)，这样计算机功能就第一次得到扩展。操作系统隐藏了计算机硬件的复杂细节，向用户提供了一台功能扩展的机器，是硬件与软件相互作用的产物，称为虚拟计算机 (Virtual Machine)。常见的操作系统有 Windows (PC 机系统为主)、Linux (服务器系统为主)、UNIX (服务器系统为主)、Android (智能终端系统为主)、IOS (智能终端系统为主)。



图 0.1 冯·诺伊曼

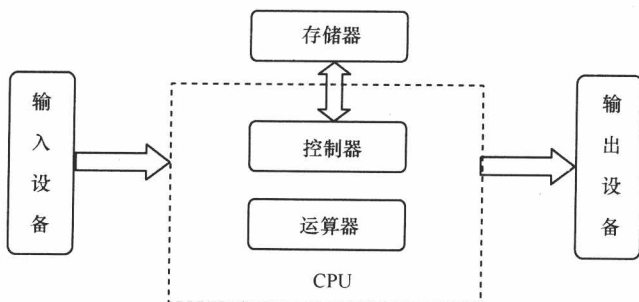


图 0.2 计算机工作原理

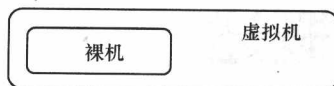


图 0.3 裸机与虚拟机

0.2 计算机语言简史

自然语言是人类交流的工具，计算机语言（Computer Language）是人与计算机交流的工具。广义上，计算机语言包含程序设计语言（Programming Language）；在计算机语言中，程序设计语言的发展最具有代表性，反映了计算机语言发展阶段，所以狭义上讲计算机语言就是程序设计语言。计算机语言大致经历了机器语言、汇编语言、面向过程语言和面向对象语言四个阶段，面向过程语言与面向对象语言又合称为高级语言。

1. 机器语言

机器语言（Machine Language）是直接使用二进制代码表示的机器指令语言。使用机器语言编写的程序可以直接被计算机理解执行，但是对人而言不直观，难写难记。在计算机发明的初期，只有计算机专家可以二进制的机器语言与计算机交流，最早的程序是通过打孔机在纸带上编写的，如图 0.4 所示。不同的机器，机器指令可能是不同的，也就是说机器语言针对特定的机器，为一台机器编写的机器语言程序很难在另一台计算机上执行。

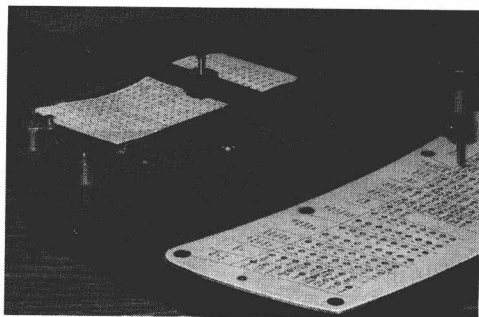


图 0.4 打孔机

2. 汇编语言

针对机器语言不适合人使用的问题，通过一些简洁的英文字母或符号串来替代一个特定的二进制串指令，比如用 ADD 代表加法操作、MOV 代表数据传递操作，这就产生了汇编语言（Assembly Language）。计算机不能认识这些符号，需要将这些符号翻译成二进制数的机器指令，这种翻译程序称为汇编程序；汇编语言同样依赖于具体的机器硬件，移植性不好，但是执行效率较高。机器语言和汇编语言，都是直接操作硬件的，解决问题时不仅要考虑问题求解思路，更要熟悉机器内部结构。

3. 面向过程的语言

为了解决机器语言和汇编语言遇到的问题，把人们的精力集中于解题思路，关注数据加工过程，完成特定的功能，而不是过多地关注机器低层细节，面向过程的计算机语言（Procedure Oriented Language）诞生了。经过努力，1970 年第一个面向过程的程序设计语言——Pascal 语言出现，标志着结构化程序设计时期的开始；1972 年贝尔实验室在 B 语言的基础上，开发出著名的 C 语言。

4. 面向对象的语言

自 20 世纪 80 年代初开始,在软件工程设计思想上,又产生了一次革命,产生了面向对象的程序设计语言(Object-Oriented Programming Language)。面向对象思想更接近人的思维方式,把现实世界抽象为类,类产生对象,对象具有属性和行为。最早出现了面向对象的程序设计 Smalltalk 语言;1983 年在 C 语言的基础上产生了 C++;在 C++的基础上,1995 年 SUN 公司推出了 Java;在 Java 诞生之后,2000 年微软推出了 C#。

0.3 Java 语言简史

著名的 SUN (Stanford University Network) 微系统公司 1982 年创建。1991 年 SUN 公司的 James Gosling (Java 之父,如图 0.5 所示)带领团队做关于家电的 Green 项目。Green 项目组一开始使用 C++语言来开发,由于家电产品硬件的多样性和差异性以及 C++的不足,最后项目组放弃了 C++,他们设计出了一种基于 C++且平台独立的语言 Oak,就是 Java 的前身。很不幸,Green 项目失败了,没有给 SUN 带来任何收益。直到 1994 年,随着互联网的飞速发展,Green 项目组使用 Oak 技术开发 HotJava 浏览器,第一次证明了该语言强大的能力。但是 Oak 已被注册,于是改名为 Java。之后,世界上的浏览器市场最后被微软的 IE 吞并,HotJava 浏览器从人们的视野中消失。

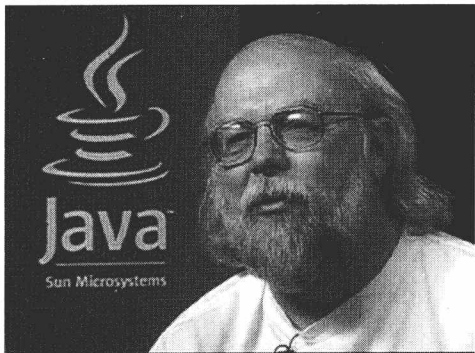


图 0.5 James Gosling



图 0.6 SUN 墓碑

在哲学上,新事物的诞生与发展是前进性与曲折性的统一,前途是光明的,道路是曲折的,Java 语言的发展史正好说明了这一规律。

1995 年,SUN 公司正式发布 Java 语言,次年发布了 JDK 1.0 (Java Developer's Kit, Java 开发工具包)。1998 年 12 月,发布了 JDK1.2,这个版本标志着 Java 已经进入 Java 2 时代,这个时期也是 Java 飞速发展的时期。JDK1.2 发展为 J2EE、J2SE 和 J2ME 三大分支,得到了 IT 市场的追捧。但微软对 Java 眼红,由“J#事件”到“微软抛弃 Java 案”,虽然 SUN 最终胜利,但是 Java 错过了可视化编程的黄金时期。这一时期可视化编程如日中天,VB、Delphi 成为可视化编程的佼佼者。也许是天意,Java 语言因祸得福,躲过了可视化编程衰亡一劫。

在 2004 年 10 月,SUN 发布了 JDK1.5。J2SE 更名为 Java SE, J2EE 更名为 Java EE, J2ME 更名为 Java ME。Java 进入了 Java 5 时代,实现了第二次飞跃。

如图 0.6 所示,2009 年 4 月 IT 巨头 Oracle 收购 SUN,取得 Java 的版权,从此 Java 属于 Oracle 公司。2010 年 Java 之父 James Gosling 离开了 Oracle,Java 的命运堪忧。2011 年 7 月,Oracle 发布 Java 7 的正式版,Java 8 即将问世。Oracle 不肯放弃 Java,凭借 Oracle 巨大财力和技术,Java 的明天更美好!

0.4 Java 运行机制

Java 语言的特点很多，最大的特点就是跨平台性，就是 Java 的程序不需要经过修改可以运行在多个操作系统平台上。我们知道不同的计算机硬件和不同的操作系统，差异性很大，Java 跨平台性是怎么实现的呢？这就需要对虚拟计算机进一步扩展。

0.4.1 Java 虚拟机

JDK 是 Java 开发工具包，JDK 包含了 JRE（Java Runtime Environment，Java 运行环境），JRE 中又包含 JVM（Java Virtual Machine，Java 虚拟机）软件。

在裸机上安装操作系统，这是计算机的第一次扩展，得到虚拟计算机；在虚拟计算机上安装 JDK 或 JRE，计算机第二次扩展得到 Java 虚拟机 JVM（运行 JVM 软件的虚拟计算机系统）。原来虚拟机的功能得到扩展，可以执行 Java 命令、运行 Java 应用软件。裸机、虚拟机和 JVM 之间的关系如图 0.7 所示。Java 虚拟机屏蔽了虚拟计算机（比如 Windows 系统计算机和 Linux 系统计算机）的差异性，提供统一的 Java 编译技术。在 JVM 层次，针对 Java 语言而言，所有的计算机都是相同的，具有一般机器的特性。但是 JVM 本身不是跨平台的，一般由 C/C++ 语言编写，Java 提供了不同操作系统下的 JVM 版本，常见的有 Linux 版、Mac OS 版、Solaris 版和 Windows 版。

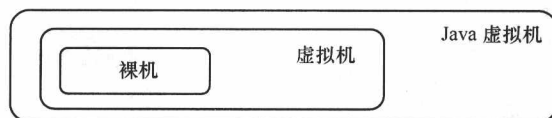


图 0.7 裸机、虚拟机和 JVM 之间的关系

0.4.2 运行机制

如图 0.8 所示，假设我们有一台 Windows 系统计算机和一台安装了 Linux 系统的计算机，并且均安装了对应的 JDK（包含 JVM 软件），也就是说这两台计算机就是两台 Java 虚拟机。首先在 Windows 计算机上编写 Java 源程序 A.java（Java 源文件的扩展名是.java），经编译器编译 A.java 源文件后产生字节码（Bytecode）文件 A.class（字节码文件的扩展名.class）。字节码文件是经过 Java 编译器处理后的一种文件，只有 JVM 才能识别执行。然后将字节码文件 A.class 迁移到 Linux 计算机上，该计算机上的 JVM 使用 JIT（Just In Time，即时编译）技术将字节码文件编译成本地机器码，然后让 Linux 系统运行该机器码，不是解释一句执行一句。

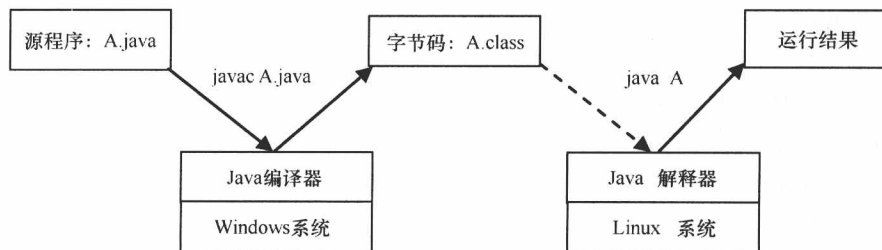


图 0.8 Java 程序运行机制

Java 语言的跨平台性，是指字节码文件不经过修改直接移植到其他平台上运行，也就是 Java 的口号“一次编译，处处运行”（Write once, run anywhere）。字节码文件跨平台的关键是 JVM 提供了跨平台的环境，但是 JVM 不是跨平台的，JVM 完成字节码到本地机器码的转换。

0.5 名词解释

Java 语言中专业名称较多，容易产生概念混淆，下面集中解释一下。

（1）Java SE: Java Standard Edition，即 Java 标准版，提供基本的低层支持，最新版是 Java SE 7，Java SE 是一个 Java 平台的名称。

（2）Java EE: Java Enterprise Edition，即 Java 的企业版，前身是 J2EE，主要构建企业应用系统。

（3）Java ME: Java Micro Edition，即 Java 微型版本，主要用于嵌入式系统开发，现在主要用于开发手机游戏。

（4）JDK: 最新版本是 JDK 1.7，是基于 Java SE 平台的开发程序发行版本，JDK 是相对开发者而言，JDK 包含 JRE，JRE 包含 JVM，JVM 包含 JIT。

0.6 习 题

1. Java 之父是谁？
2. J2EE 是什么？（某企业面试题）
3. 简要说明 Java 程序的运行机制。
4. 说明 JDK、JRE、JVM、JIT 之间的关系。

第 1 章

Java 开发环境配置

本章重要性：★★★

本章学习目标：

- ◇ 掌握在 Windows 平台下 JDK 的安装与配置
- ◇ 了解在 Linux 平台下 JDK 的安装与配置
- ◇ 掌握 Java 环境测试
- ◇ 掌握第一个 Java 程序
- ◇ 掌握编译和执行命令
- ◇ 了解跨平台操作问题
- ◇ 了解集成开发环境

工欲善其事，必先利其器。Java 开发环境配置是学习 Java 的基本功！下面将分别以 Windows 平台和 Linux 平台介绍 Java 开发环境的配置。然后演示第一个 Java 程序的编辑、编译和执行过程。

1.1 Windows 平台 Java 环境配置

Windows 操作系统是当今最为流行的 PC 机操作系统，国内以 Windows XP 系统为主。本教材将以 Windows XP 系统为例来介绍 Windows 平台下 Java 环境配置，Windows 7 版本的操作相似，稍微有些差别。

1.1.1 下载 Java 开发包 JDK

JDK (Java Developer's Kit) 是 Java 开发工具包，是针对 Java 开发员的产品，用于开发 Java 程序。JDK 包含了 Java 运行环境 JRE、Java 工具和 Java 基础类库，学会安装 JDK 是学好 Java 的第一步。JDK 是免费的。

(1) 如图 1.1 所示，登录 Oracle 官方网站 (<http://www.oracle.com>)，将鼠标放到“DOWNLOADS”菜单上（无须单击），单击“Java for Developers”，进入 Java 下载页面。

(2) 在下一页面中单击第一个 Java 图标，如图 1.2 所示。在最近的一年期间，JDK7 发布了 6 个更新版本，截稿时 JDK 最新版是 JDK 7u7，也就是 JDK 7 的第 7 个更新版本。读者朋友在此页面中看到的应该是再次更新的版本。

(3) 接受协议“Accept License Agreement”，针对 Windows XP 32 位系统下载 Windows x86 版本，如图 1.3 所示；对于 64 位的操作系统，请下载 Windows x64 版本的 JDK。此外，读者朋友可以看到其他平台下的 JDK 版本，比如 Linux 平台、Mac OS 平台等。

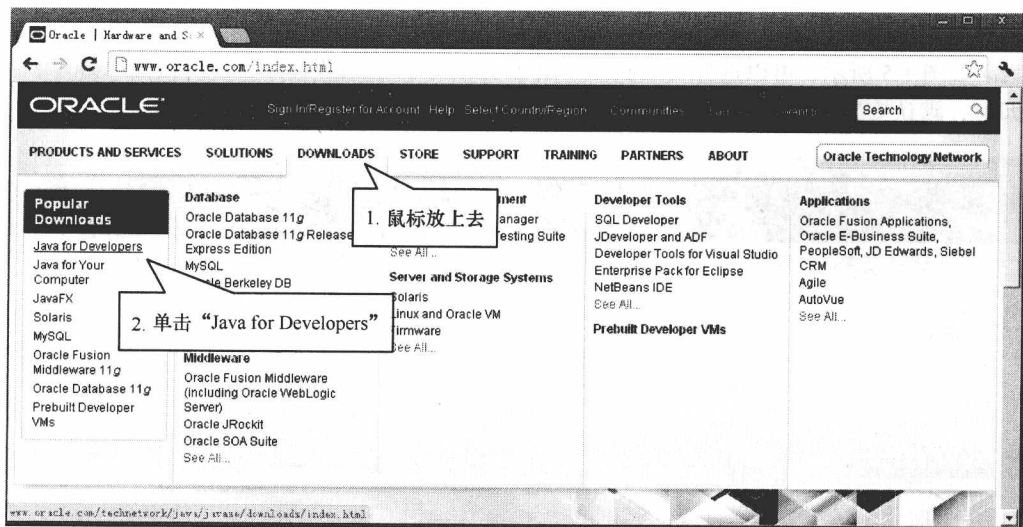


图 1.1 Oracle 官方网站



图 1.2 单击“Java”图标

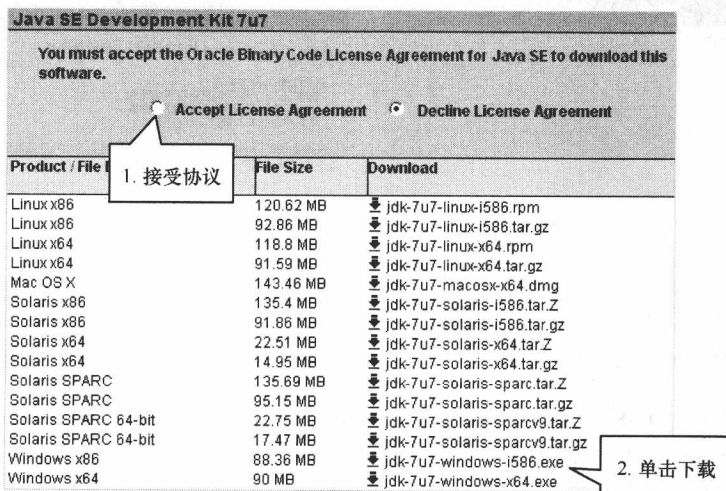


图 1.3 选择 Windows x86 平台的 JDK

1.1.2 安装 JDK

(1) 下载后得到“jdk-7u7-windows-i586.exe”文件。双击该可执行文件的文件名，将出现安

装向导界面，如图 1.4 所示，单击“下一步”按钮。

(2) 如图 1.5 所示，可以选择安装路径。由于 Java 平台是系统程序，建议直接单击“下一步”按钮即可，默认安装到 C:\Program Files\Java 目录下面。请记住 JDK 的安装目录。

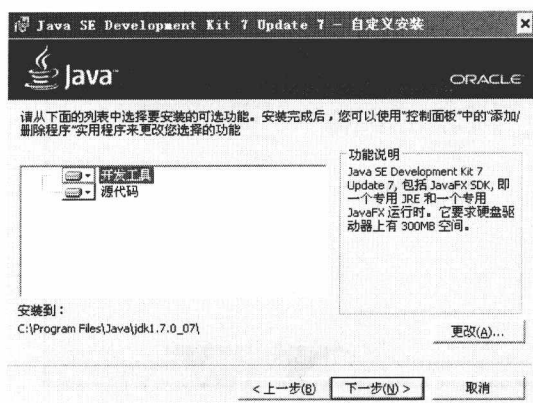


图 1.4 安装向导

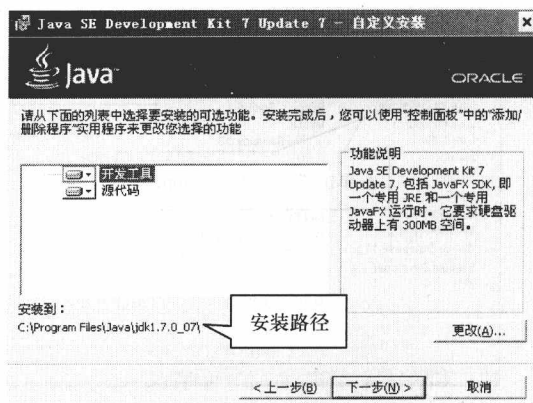


图 1.5 选择安装目录

(3) 如图 1.6 所示，单击“下一步”按钮安装 JRE。在第 0 章已经介绍，JRE (Java Runtime Environment) 是 Java 的运行环境，JRE 面向 Java 程序的使用者，而 JDK 是面向 Java 程序的开发者。JDK 中已经包含了一个 JRE，为什么还要再安装一个 JRE 呢？因为有些用 Java 开发的软件运行需要 JRE，比如集成开发工具 Eclipse 就需要单独的 JRE。JRE 是运行 Java 应用程序所必需环境的集合，包括 JVM、核心类以及支持文件，但是不包含开发工具（比如编译器）。

(4) 如图 1.7 所示，安装完成，单击右上角“X”关闭按钮即可。此后会弹出窗口提示安装“JavaFX SDK”，无须安装，取消关闭即可。

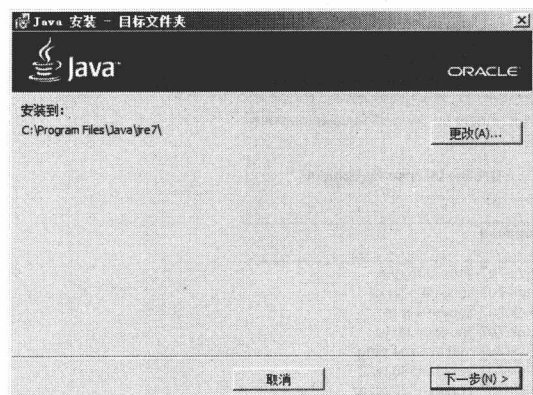


图 1.6 安装 JRE

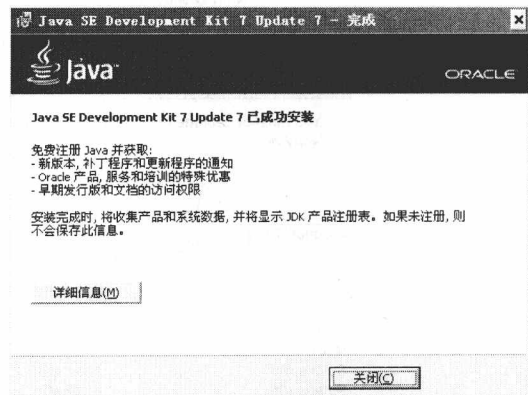


图 1.7 安装完成

(5) 安装完成后，JDK 默认安装到目录 C:\Program Files\Java\下，此目录下读者可以发现一个 jdk 目录和一个 jre 目录。打开 jdk 目录，如图 1.8 所示。

- bin 目录。bin 是二进制 binary 缩写，表示编译后的二进制可执行文件；在 JDK 的 bin 目录下可以看到“java.exe”和“javac.exe”两个文件，这就是 Java 的解释器和编译器命令。
- lib 目录：lib 是 library 的缩写，表示库文件；JDK 的 lib 目录存放 Java 开发时所需要的类库和支持文件。
- jre 目录。JDK 的 JRE，即 Java 运行环境，包含 JVM、类库等文件。