

■ 高等学校计算机教材 ■

Java

实用教程 (第2版)

■ 郑阿奇 主编 ■ 姜乃松 殷红先 刘毅 编著 ■

 电子工业出版社



高等学校计算机教材

Java 实用教程

(第2版)

郑阿奇 主编

姜乃松 殷红先 刘毅 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本教程以 Java 最新的版本《Java SE Development Kit (JDK) 6 Update 10》为基础, 对第 1 版进行全新改版, 内容循序渐进、深入浅出, 精心设计每一个示例, 结构安排更为合理, 使读者准确把握 Java 的知识点。

本教程包括 Java 基础、习题、实验和习题答案四部分。本教程在讲解内容后紧跟实例, 每章的最后还配备了一个综合实例, 对已经学习的主要知识进行综合应用。实验部分通过实例引导读者进行学习, 并提出思考问题, 最后在原来基础上让读者自己进行操作和编程练习。实例程序一般均通过上机调试。全书以开源软件 Eclipse 作为 Java 的集成开发环境, 使得编写、调试、运行 Java 程序变得更为简便。

本书既可作为大学本科和专科计算机及相关专业的教材, 也可作为 Java 自学者或者应用开发者的参考书。

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

Java 实用教程 / 郑阿奇主编. —2 版. —北京: 电子工业出版社, 2009.8

高等学校计算机教材

ISBN 978-7-121-08902-2

I. J… II. 郑… III. JAVA 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 081063 号

策划编辑: 赵云峰

责任编辑: 宋兆武 李施诺

印 刷: 北京京师印务有限公司
装 订:

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: 787×1 092 1/16 印张: 33 字数: 844.8 千字

印 次: 2009 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 43.00 元

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 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

Java 是 Sun 公司开发的新一代编程语言，其简单、面向对象、多线程、跨平台等特性深受人们的欢迎。目前，学习 Java 已经成为一种时尚和潮流。

2005 年，为了满足教学和应用开发实践的要求，我们编写了《Java 实用教程》一书，把学习和应用相结合，收到了较好的效果。

《Java 实用教程（第 2 版）》以 Java 最新的版本《Java SE Development Kit (JDK) 6 Update 10》为基础，对第 1 版进行全新改版，内容循序渐进、深入浅出，精心设计每一个示例，结构安排更为合理，使读者准确把握 Java 的知识点。实用教程一般在讲解一项内容后紧跟相关的实例演示，每章的最后还配备了一个综合实例，使学习者综合应用已经学过的主要知识。实验部分也是通过实例先引导读者进行学习，再提出思考问题，最后在原来基础上让读者自己进行操作和练习编程。实例程序均通过上机调试。全书以开源软件 Eclipse 作为 Java 的集成开发环境，这使得编写、调试、运行 Java 程序变得更为简便。

实际上，本教程不仅适合教学，也非常适合 Java 的各类培训，以及使用 Eclipse 编程开发应用程序的用户学习和参考。只要阅读本书，结合上机操作进行练习，就能在较短的时间内基本掌握 Java 知识及其应用技术。

与本书配套的同步 PowerPoint 课件，可直接用于课堂教学。书中的源代码和 PowerPoint 课件，可从 <http://www.hxedu.com.cn> 或者 <http://www.huaxin.edu.cn> 网站上免费下载。

本书由南京师范大学姜乃松、殷红先和南京审计学院刘毅编写，南京师范大学郑阿奇任主编并对全书进行统稿、定稿。其他一些同志参加本书的基础工作，在此表示诚挚的谢意。

参加本套丛书（《高等学校计算机教材》）编写的还有梁敬东、顾韵华、王洪元、杨长春、王一莉、曹弋、徐文胜、丁有和、刘启芬、殷红先、张为民、丁有和、彭作民、刘毅、郑进、周怡君、赵阳、周旭琴、陈金辉、李含光、黄群等。

由于作者水平有限，不当之处在所难免，恳请读者批评指正。

意见及建议邮箱：easybooks@163.com。

编 者

2009 年 6 月

目 录

第 1 部分 Java 实用教程

第 1 章	JAVA 和 ECLIPSE 集成开发环境	1
1.1	Java 语言简介	1
1.1.1	Java 语言特点	1
1.1.2	Java 运行机制	2
1.2	第一个 Java 程序	3
1.3	Java 开发环境的搭建	4
1.4	Eclipse 集成开发环境	7
第 2 章	JAVA 语法基础	9
2.1	常量和变量	9
2.1.1	数据类型	9
2.1.2	标识符和关键字	9
2.1.3	常量	10
2.1.4	变量	12
2.1.5	类型转换	13
2.2	运算符和表达式	15
2.2.1	运算符	15
2.2.2	表达式	21
2.3	流程控制	21
2.3.1	分支语句	21
2.3.2	循环语句	25
2.3.3	流程跳转语句	29
2.4	方法与递归	30
2.4.1	方法	30
2.4.2	参数传递	30
2.4.3	递归	31
2.5	命名规范	32
2.6	注释语句	33
2.7	综合实例	33
第 3 章	JAVA 面向对象编程（上）	36
3.1	类的定义及成员变量初始化	37
3.1.1	类的定义	37
3.1.2	变量的初始化	38

3.2	创建对象	40
3.2.1	构造方法	40
3.2.2	默认构造方法	43
3.2.3	构造方法重载	44
3.2.4	普通方法重载	45
3.3	this 引用	47
3.4	静态成员	49
3.4.1	静态变量	50
3.4.2	静态方法	51
3.4.3	main()方法	52
3.4.4	类的初始化	55
3.5	package 与 import 语句	56
3.6	内部类	60
3.6.1	实例内部类	60
3.6.2	匿名类	62
3.6.3	static 内部类	63
3.6.4	局部内部类	64
第 4 章	JAVA 面向对象编程 (下)	66
4.1	继承	66
4.1.1	继承的定义	66
4.1.2	初始化基类	68
4.1.3	方法的重写	68
4.1.4	super 关键字	71
4.2	对象的转型	72
4.3	多态	76
4.4	抽象类	78
4.5	接口	80
4.6	终止继承	82
4.6.1	final 类	82
4.6.2	final 方法	82
4.6.3	final 变量	84
4.7	权限修饰符	86
4.7.1	类内部	86
4.7.2	同一个包的类	87
4.7.3	不同包的子类	89
4.7.4	通用性	89
4.7.5	访问权限与继承	90
4.8	综合实例: 航班管理	91
第 5 章	常用类与异常处理	101
5.1	Object 类	101

177	5.1.1	equals()方法	101
178	5.1.2	hashCode()方法	103
181	5.1.3	toString()方法	104
181	5.2	字符串类	105
181	5.2.1	String 类	105
181	5.2.2	StringBuffer 类	108
188	5.3	包装类	110
189	5.4	Math 类	113
189	5.5	Random 类	114
191	5.6	日期类	115
191	5.6.1	Date 类	115
191	5.6.2	DataFormat 类	116
191	5.6.3	Calendar 类	118
198	5.7	正则表达式	119
198	5.7.1	替换操作	124
198	5.7.2	Scanner 定界符	125
198	5.8	异常处理	126
198	5.8.1	异常的捕获与处理	127
198	5.8.2	声明抛出异常子句	129
198	5.8.3	抛出异常语句	130
198	5.8.4	自定义异常类	131
199	5.9	综合实例	133
199	第 6 章	数组、枚举及注解	136
199	6.1	一维数组	136
199	6.2	多维数组	139
199	6.3	访问数组	143
199	6.4	数组实用类: Arrays	145
199	6.4.1	复制数组	145
199	6.4.2	数组排序	146
199	6.4.3	数组元素的查找	147
199	6.5	枚举	148
199	6.6	enum 的构造方法	150
199	6.7	使用 EnumMap	151
199	6.8	注解	152
199	6.9	内置注解	152
199	6.9.1	Override	153
199	6.9.2	Deprecated	154
199	6.9.3	SuppressWarnings	155
199	6.10	注解	156

6.10.1	自定义注解	157
6.10.2	元注解	157
6.11	综合实例	160
第7章	容器和泛型	164
7.1	Collection 与 Iterator	164
7.2	实用类 Collections	166
7.3	Set (集)	168
7.3.1	HashSet	168
7.3.2	TreeSet	170
7.4	List (列表)	174
7.4.1	ArrayList	174
7.4.2	LinkedList	177
7.4.3	栈的实现	179
7.5	Queue (队列)	180
7.5.1	PriorityQueue	181
7.5.2	实现队列	182
7.5.3	双向队列	183
7.6	Map (映射)	185
7.6.1	HashMap	187
7.6.2	TreeMap	188
7.7	泛型	190
7.8	泛型方法	193
7.9	受限泛型	195
7.10	通配符与受限通配符	196
7.11	子类泛型	198
7.12	泛型数组	200
7.13	综合实例	202
第8章	JAVA 输入/输出系统	206
8.1	字节流	206
8.1.1	字节数组输入流	207
8.1.2	字节数组输出流	208
8.1.3	文件输入流	209
8.1.4	文件输出流	210
8.1.5	管道流	211
8.2	过滤流	213
8.2.1	缓冲流类	213
8.2.2	数据流类	216
8.2.3	PrintStream 类	217
8.3	字符流	219

682	8.3.1	转换流类	219
685	8.3.2	FileReader 和 FileWriter	221
692	8.3.3	BufferedReader 和 BufferedWriter	222
695	8.3.4	PrintWriter 类	224
695	8.4	标准 I/O	225
695	8.4.1	重新包装标准 I/O	226
700	8.4.2	标准 I/O 重定向	226
702	8.5	随机访问文件类	227
709	8.6	对象序列化	230
709	8.7	File 类	232
709	8.8	综合实例	234
709	第 9 章	AWT 组件及应用	237
709	9.1	AWT 容器	237
710	9.1.1	Window 和 Frame	238
710	9.1.2	Panel	239
717	9.2	布局管理器	240
719	9.3	事件处理机制	241
815	9.3.1	AWT 事件与监听器	242
818	9.3.2	窗口事件	243
847	9.3.3	鼠标事件	245
848	9.3.4	键盘事件	246
853	9.3.5	内部类实现监听接口	248
861	9.3.6	类自身实现监听接口	250
872	9.3.7	外部类实现监听接口	252
881	9.3.8	采用事件适配器	254
886	9.4	Graphics 类	256
893	9.5	AWT 绘图	258
895	9.6	综合实例	261
895	第 10 章	SWING 组件及应用	266
902	10.1	窗口 JFrame	266
914	10.2	Swing 常用组件 (1)	268
920	10.2.1	按钮	268
926	10.2.2	JTextField 与 JTextArea	270
929	10.2.3	JCheckBox 和 JRadioButton	273
935	10.2.4	列表框 (JList)	277
941	10.3	Swing 常用组件 (2)	278
943	10.3.1	菜单条 (JMenuBar)	278
950	10.3.2	弹出式菜单 (JpopupMenu)	282
960	10.3.3	页签面板 (JtabbedPanel)	284

10.3	10.3.4 工具栏 (JToolBar)	286
10.4	Swing 常用组件 (3)	291
10.4.1	表格 (JTable)	291
10.4.2	树型视图 (JTree)	293
10.5	Swing 常用组件 (4)	296
10.5.1	对话框 (JDialog)	296
10.5.2	消息框 (JOptionPane)	297
10.5.3	文件对话框 (JFileChooser)	300
10.6	综合实例	303
第 11 章	并发	307
11.1	线程的创建与启动	307
11.1.1	继承 java.lang.Thread 类	307
11.1.2	实现 Runnable 接口	309
11.2	线程的状态转换	311
11.3	线程调度	313
11.3.1	调整各个线程的优先级	313
11.3.2	线程让步	314
11.3.3	合并线程	316
11.4	后台线程	317
11.5	线程互斥	318
11.5.1	临界区	318
11.5.2	对象锁机制	321
11.5.3	显式的 Lock 对象	324
11.6	线程同步	327
11.7	综合实例	331
第 12 章	综合实例	337
12.1	设计思路	337
12.2	汉诺塔上盘子模拟	337
12.3	汉诺塔上的对象的定位及盘子的存放	338
12.4	创建汉诺塔及实现手动搬运盘子	339
12.5	自动搬运盘子	347
第 13 章	JAVA 网络编程	355
13.1	网络程序设计基础	355
13.1.1	TCP 和 UDP	355
13.1.2	端口和套接字	355
13.2	TCP 网络编程	356
13.2.1	InetAddress 类	358
13.2.2	TCP 通信程序	360
13.3	UDP 网络编程	364

13.3.1	UDP 通信程序	366
13.3.2	组播	369
13.4	URL	373
13.4.1	URL 类	373
13.4.2	URLConnection 类	375
13.5	综合实例	377
第 14 章	JDBC 编程	383
14.1	SQL 语言	383
14.2	JDBC	385
14.3	MySQL 数据库	387
14.3.1	MySQL 服务器的安装	388
14.3.2	MySQL 服务器的配置	388
14.3.3	MySQL 的环境	390
14.4	访问数据库	391
14.4.1	加载并注册数据库驱动	392
14.4.2	建立到数据库的连接	393
14.4.3	执行数据库的访问	394
14.5	JDBC 编程	396
14.6	批处理	402
14.7	事务处理	405
14.8	综合实例	407

第 2 部分 习题集

第 1 章	Java 和 Eclipse 集成开发环境	410
第 2 章	Java 语法基础	410
第 3 章	Java 面向对象编程 (上)	411
第 4 章	Java 面向对象编程 (下)	411
第 5 章	常用类与异常处理	411
第 6 章	数组、枚举及注解	412
第 7 章	容器和泛型	412
第 8 章	Java 输入/输出系统	412
第 9 章	AWT 组件及应用	413
第 10 章	Swing 常用组件	413
第 11 章	并发	414
第 13 章	Java 网络编程	414
第 14 章	JDBC 编程	414

第 3 部分 实 验

实验 1	在 ECLIPSE 集成开发环境中调试程序	416
------	-----------------------	-----

实验目的	416
实验准备	416
实验内容	416
思考与练习题	418
实验 2 JAVA 语法基础	419
实验目的	419
实验准备	419
实验内容	419
思考与练习题	422
实验 3 JAVA 面向对象编程 (上)	423
实验目的	423
实验准备	423
实验内容	423
思考与练习题	425
实验 4 JAVA 面向对象编程 (下)	426
实验目的	426
实验准备	426
实验内容	426
思考与练习题	429
实验 5 常用类与异常处理	430
实验目的	430
实验准备	430
实验内容	430
思考与练习题	433
实验 6 数组、枚举及注解	434
实验目的	434
实验准备	434
实验内容	434
思考与练习题	437
实验 7 容器和泛型	438
实验目的	438
实验准备	438
实验内容	438
思考与练习题	441
实验 8 JAVA 输入/输出系统	442
实验目的	442
实验准备	442
实验内容	442
思考与练习题	445

实验 9 AWT 组件及应用	447
实验目的	447
实验准备	447
实验内容	447
思考与练习题	453
实验 10 SWING 常用组件及应用	454
实验目的	454
实验准备	454
实验内容	454
思考与练习题	460
实验 11 并发	461
实验目的	461
实验准备	461
实验内容	461
思考与练习题	464
实验 13 Java 网络编程	465
实验目的	465
实验准备	465
实验内容	465
思考与练习题	472
实验 14 JDBC 编程	473
实验目的	473
实验准备	473
实验内容	473
思考与练习题	480

第 4 部分 习题答案

第 1 章 习题答案	481
第 2 章 习题答案	481
第 3 章 习题答案	484
第 4 章 习题答案	486
第 5 章 习题答案	490
第 6 章 习题答案	491
第 7 章 习题答案	495
第 8 章 习题答案	497
第 9 章 习题答案	499
第 10 章 习题答案	502
第 11 章 习题答案	506
第 13 章 习题答案	506
第 14 章 习题答案	509

第 1 部分 Java 实用教程

第 1 章 Java 和 Eclipse 集成开发环境

1.1 Java 语言简介

Java 是由 Sun 公司于 1991 年开发的新一代编程语言，初衷是为家用消费类电子产品开发一个分布式代码系统。为了使整个系统与平台无关，采用了虚拟机器码（Virtual Machine Code）方式，虚拟机运行在一个解释器上，每一个操作系统均有一个解释器。这样，Java 就成了平台无关的语言。

Java 的发展速度是惊人的。1995 年 3 月 Sun 公司发布了 Java 的 Alpha1.0a2 版本，1996 年 1 月发布了 Java 的第一个开发包 JDK v1.0，1997 年 2 月发布了 Java 语言的开发包 JDK v1.1，从此奠定了 Java 在计算机语言中的地位。1998 年 12 月 Sun 公司发布 Java 2 平台 JDK v1.2，这是 Java 发展史上的里程碑。1999 年 6 月，Sun 公司重新组织 Java 平台的集成方法，并将企业级应用平台作为 Java 发展方向，主要有三个成员：

- J2ME—Java 2 Micro Edition，用于嵌入式应用的 Java2 平台。

- J2SE—Java 2 Standard Edition，用于工作站、PC 的 Java2 标准平台。

- J2EE—Java 2 Enterprise Edition，可扩展的企业级应用的 Java2 平台

2004 年，J2SE1.5 发布，为了表示这个版本的重要性，J2SE1.5 更名为 J2SE5.0。2005 年，JavaOne 大会召开，Sun 公司公开 JavaSE6，此时，Java 的各种版本被更名，取消其中的数字“2”，J2SE 更名为 Java SE，J2EE 更名为 Java EE，J2ME 更名为 Java ME。

1.1.1 Java 语言特点

Java 是一个广泛使用的网络编程语言，它简单、面向对象，不依赖于机器的结构，不受 CPU 和环境的限制，具有可移植性、安全性，并且提供了并发的机制，具有很高的性能。此外，Java 还提供了丰富的类库，使程序设计人员可以方便地建立自己的系统。

(1) 简单性。Java 语言是一种面向对象的语言，语法规则和 C 类似，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 略去了指针、运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集机制大大简化了程序设计者的内存管理工作。另外，Java 对环境要求非常低，它的基本解释器及类的支持只有几十千字节，加上标准类库和线程的支持也大约 200KB。

(2) 面向对象。Java 语言的设计集中于对象及其接口，它提供了简单的类机制及动态的接口模型。对象中封装了它的状态变量及相应的方法，实现了模块化和信息隐藏；而类则提供了一

类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法实现代码的复用。

(3) 分布性。Java 是面向网络的语言，通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

(4) 安全性。用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

(5) 可移植性。与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。

(6) 高性能。和其他解释执行的语言不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

(7) 多线程。多线程机制使应用程序能够并发执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易地实现网络上的实时交互行为。

1.1.2 Java 运行机制

Java 有两种核心机制：一种是 Java 虚拟机 (Java Virtual Machine)；另一种是垃圾收集机制 (Garbage Collection)。

1. Java 虚拟机

Java 程序是如何做到“一次编译，到处运行”呢？这正是通过 Java 虚拟机来实现的。JVM 可以理解成一个以字节码为机器指令的 CPU。首先，Java 编译程序将后缀名为.java 的 Java 源程序编译为 JVM 可执行的代码，即后缀名为.class 的 Java 字节码文件，如图 1.1 所示。运行 JVM 字节码的工作是由解释器来完成的。解释执行过程分代码的装入、代码的校验和代码的执行三步进行。装入代码的工作由“类装载机”完成，类装载机负责装入一个程序运行需要的所有代码。字节码校验器负责代码的校验。每种类型的操作系统都有一种对应的 Java 虚拟机，Java 虚拟机屏蔽了底层操作系统的差异。所以 Java 程序能够做到“一次编译，到处运行”。

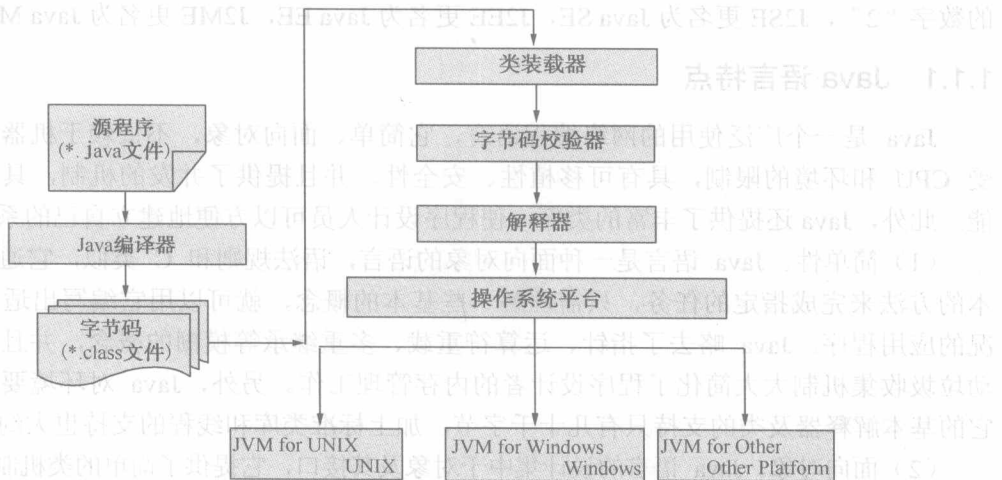


图 1.1 Java 程序执行流程

2. 垃圾收集机制

垃圾回收器能够自动回收垃圾，即无用的对象所占据的内存空间被回收。在 C/C++ 语言中，这些工作由程序员负责，无疑增加了程序员的负担。而 Java 语言消除了程序员回收垃圾的责任：它提供一种系统级线程来跟踪存储空间的分配情况，并在 JVM 的空闲时，检查并释放那些可被释放的存储空间。在 Java 中，对象被创建后，就会在堆区中分配一块内存。当对象不再被程序引用时，它就变成一个垃圾，所占用的堆空间可以被回收，以便空间被后续的新对象所使用。Java 的垃圾回收器能断定哪些对象不再被引用，并且能够把它们所占用的堆空间释放出来。

1.2 第一个 Java 程序

下面先看一个 Java 小程序，对 Java 编程有一个初步的认识。

【例 1.1】 求圆面积。

用文本编辑器（如 Windows 的记事本）编辑下列程序，文件名为 Area.java。

Area.java

```
/** 求圆的面积 */
public class Area {
    double pi = 3.1415;           // 定义变量 pi
    public static void main(String[] args) {
        double r,area;
        r = 3;
        area = pi * r * r;       // 求圆的面积
        System.out.println("圆的面积="+area); // 在屏幕上显示结果
    }
}
```

下面对程序进行简要说明：

(1) `/**到*/`之间的内容为注释。

(2) 保留字 `class` 声明一个类，其类名为 `Area`，保留字 `public` 表示它是一个公共类。类定义由花括号 `{}` 括起来。当编写一个 Java 源代码文件时，此文件通常被称为编译单元。每个编译单元都必须有一个后缀名为 `.java`，而在编译单元内则可以有一个 `public` 类，该类的名称必须与文件的名称相同（包括大小写）。每个编译单元最多只能有一个 `public` 类，也可以没有，否则编译器就不会接收。

(3) 语句 `“double pi = 3.1415;”` 表示定义实型变量 `pi` 的值为 3.1415。

(4) 在该类中定义了一个 `main()` 方法，其中 `public` 表示访问权限，指明所有的类都可以使用这一方法；`static` 指明该方法是一个类方法，它可以通过类名直接调用；`void` 则指明 `main()` 方法不返回任何值。对于一个应用程序来说，`main()` 方法是必需的，而且必须按照如上的格式来定义。Java 解释器在没有生成任何实例的情况下，以 `main()` 方法作为入口来执行程序。Java 程序中可以定义多个类，每个类中可以定义多个方法，但是最多只能有一个公共类，`main()` 方法也只能有一个。

1.3 Java 开发环境的搭建

要想编译和运行上面的程序，离不开 Java 的编译和运行环境。SUN 公司提供了自己的一套 Java 开发环境，通常称为 JDK (Java Development Kit)，又称 J2SDK。目前最新的版本是 Java SE Development Kit (JDK) 6 Update 10，可以到 SUN 公司的网站下载。输入网址 <http://java.sun.com/javase/downloads/index.jsp>，进入第二栏，单击“Download”按钮，如图 1.2 所示。

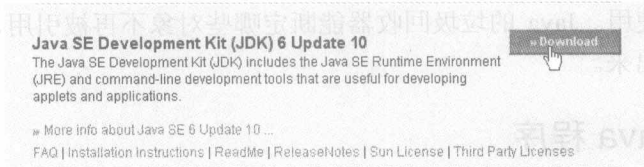


图 1.2 选择操作系统

进入图 1.3，在“Platform”栏中选择 Windows，如果自己的平台是 Linux 操作系统，请选择 Linux。选上前面的按钮，单击“Continue”按钮。

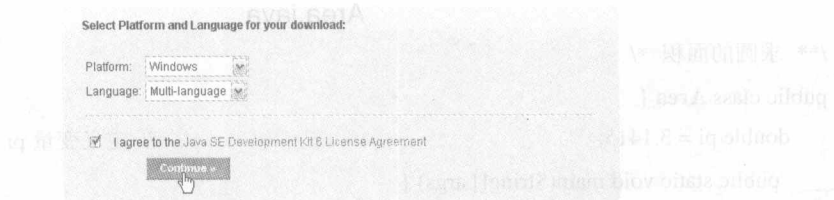


图 1.3 选择操作系统

然后进入图 1.4，在单选按钮上打勾，JDK 就可以开始下载了。



图 1.4 JDK 的下载

下载完成后，双击可执行文件 `jdk-6u10-windows-i586-p.exe`。按照提示完成安装，这里 JDK 的安装路径改为“C:\Java\jdk1.6.0_10\”，如图 1.5 所示。

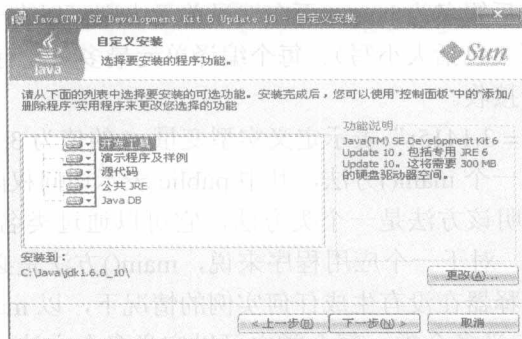


图 1.5 JDK 的安装