

普通高等教育“十三五”规划教材
新工科建设之路·计算机类专业规划教材



Java程序设计

简明教程

张晓龙 吴志祥 刘俊 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十三五”规划教材
新工科建设之路·计算机类专业规划教材

Java 程序设计简明教程

张晓龙 吴志祥 刘俊 编著



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书系统地介绍了 Java 桌面编程的基础知识,共分 10 章。其中,前 9 章的内容包括 Java 概述及开发环境,Java 语言基础,面向对象的程序设计,多线程,集合框架与泛型,Java UI 设计,Java I/O 操作与文件读写,网络编程和 JDBC 编程;第 10 章是综合项目实训。

本书充分考虑后续课程的需要,精心安排内容结构,力求从简单到复杂,循序渐进,突出逻辑性和实用性。重要的知识点都配有示例,配套的课程网站包括了相关软件下载、上机实验指导(含项目案例)、课件下载和课程档案文件下载等。

本书可以作为高等院校计算机专业和相关专业学生学习 Java 程序设计等课程的教材,也可以作为 Web 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 程序设计简明教程/张晓龙,吴志祥,刘俊编著. —北京:电子工业出版社,2018.7

ISBN 978-7-121-34114-4

I. ①J… II. ①张… ②吴… ③刘… III. ①JAVA 语言—程序设计—高等学校—教材
IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 083115 号

策划编辑:张小乐

责任编辑:张小乐

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

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:16 字数:435 千字

版 次:2018 年 7 月第 1 版

印 次:2018 年 7 月第 1 次印刷

定 价:45.00 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, 88258888。

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。

本书咨询联系方式:(010) 88254462, zhxl@phei.com.cn。

前 言

随着网络技术应用的日益发展,Java 已成为当今最流行的面向对象程序设计语言,广泛应用于桌面开发、Web 开发和移动开发等领域。目前,关于 Java 的相关书籍较多,但真正从零基础开始、内容简明而又系统的教材并不多见。为此,笔者编写了本书,希望能够满足高校教学的需求。

在充分考虑了后续课程(Java EE 和 Android 移动平台开发等)需要的基础上,本书系统地介绍了 Java 桌面编程的基础知识,共分 10 章。其中,前 9 章的内容包括 Java 开发环境的搭建,Java 编程基础,面向对象的程序设计,多线程,集合框架与泛型,Java UI 设计,Java I/O 操作与文件读写,网络编程和 JDBC 编程;第 10 章是综合项目实训。

本书结构合理、逻辑性强。作者对内容设置和结构安排进行了深思熟虑的推敲,力求做到内容从简单到复杂,循序渐进,根据相关知识点之间的联系进行组织。例如,第 2 章介绍的异常处理,将在第 4 章 Java 多线程、第 7 章 Java I/O、第 8 章 Java 网络编程和第 9 章 JDBC 编程中用到;第 3 章介绍的匿名实现类(内部类),将在第 6 章的监听器设计中用到;第 5 章介绍的泛型与集合将在第 6 章 Java UI、第 7 章 Java I/O 和第 8 章 Java 网络编程中用到;第 7 章介绍的流的相关 API 将在第 8 章 Java 网络编程中用到。

每章对知识点的介绍简明扼要且体系严密。通过图解的方式,清晰地反映了软件包中类(或接口)的成员属性(方法)。同时,配合文字简要说明其重点。此外,每章精心设计的典型例子紧扣本章相关理论。

课后练习与实验是教学的重要环节。每章末均配有习题及实验。此外,第 10 章综合项目实训能帮助学生综合使用 Java 的各个知识点。

本书有配套的上机实验网站,提供包括实验目的、实验内容、在线测试(含答案和评分)和素材等,还包括教学大纲、实验大纲、各种软件的下载链接、课件和案例源代码下载、在线测试等,极大地方便了教与学。

本书由张晓龙、吴志祥和刘俊整体构思并编写完成,张智老师制作了精美的 PPT 课件,廖光忠、柯鹏老师和研究生李岩等都参与了许多问题的讨论和代码测试。在此,一并致谢。

本书可以作为高等院校计算机专业和相关专业学生学习 Java 程序设计等课程的教材,也可以作为 Web 开发者的参考书。

如需获取本书配套的课件、案例源代码等教学资料,可访问本书配套教学网站 <http://www.cs.wust.edu.cn/courses/java> 或 <http://www.wustwzx.com/java>。

由于编者水平有限,书中错漏之处在所难免,在此真诚欢迎读者多提宝贵意见,通过访问本书配套教学网站可与作者联系,以便再版时更正。

编 者

2018 年 4 月于武汉

目 录

第 1 章	Java 概述及开发环境	(1)
1.1	Java 语言发展简史及应用	(1)
1.1.1	Java 语言发展简史	(1)
1.1.2	Java 语言应用的三个方向	(1)
1.2	Java 语言的特点	(2)
1.2.1	面向对象特性、解释性与简单性	(2)
1.2.2	平台无关性(可移植性)	(3)
1.2.3	健壮性与安全性	(3)
1.2.4	多线程特性	(3)
1.2.5	网络支持	(3)
1.3	Java 运行环境及开发环境	(4)
1.3.1	安装 JDK 并建立环境变量	(4)
1.3.2	JDK 工具箱与 Java 虚拟机	(5)
1.3.3	Java 集成开发环境 eclipse-jee	(5)
1.3.4	统一 eclipse 项目编码	(6)
1.3.5	eclipse 若干快捷操作	(6)
1.4	Java 面向对象初步	(7)
1.4.1	类与对象、封装特性	(7)
1.4.2	Java 程序结构	(8)
1.4.3	Java 类库及 API 简介	(9)
1.4.4	导入 Java 包指令 import	(10)
1.5	在 eclipse 中创建与运行 Java 项目	(10)
1.5.1	一个简单 Java 项目的创建与运行	(10)
1.5.2	Java 项目的导入	(11)
1.5.3	Java 项目导出及打包应用程序	(12)
1.5.4	Maven 项目的创建与使用	(13)
1.6	Java 程序调试	(16)
1.6.1	单元测试 JUnit 4	(16)
1.6.2	动态调试模式 Debug	(18)
习题 1		(19)
实验 1		(20)

第 2 章	Java 语言基础	(22)
2.1	标识符、关键字与保留字	(22)
2.1.1	标识符	(22)
2.1.2	关键字与保留字	(22)
2.2	数据类型	(23)
2.2.1	8 种基本数据类型	(23)
2.2.2	引用数据类型	(24)
2.2.3	包装类	(26)
2.2.4	强制转换类型	(26)
2.3	运算符与表达式	(27)
2.4	流程控制语句	(28)
2.4.1	条件语句 if	(28)
2.4.2	开关语句 switch	(28)
2.4.3	循环语句 for/while/do...while	(29)
2.4.4	中断语句 continue /break/return	(29)
2.5	数组及其遍历	(30)
2.5.1	一维数组的创建及其遍历	(30)
2.5.2	二维数组的创建及其遍历	(30)
2.6	与日期、时间、数学相关的类	(31)
2.6.1	Date 类	(31)
2.6.2	DateFormat 类和 SimpleDateFormat 类	(31)
2.6.3	Calendar 类和 GregorianCalendar 类	(31)
2.6.4	Math 类	(32)
2.7	异常处理与异常类	(33)
2.7.1	Java 内置异常	(34)
2.7.2	Java 异常处理	(36)
2.7.3	使用 throws 抛出异常	(38)
	习题 2	(39)
	实验 2	(41)
第 3 章	Java 面向对象的程序设计	(43)
3.1	面向对象基础	(43)
3.1.1	包与类的声明	(43)
3.1.2	类的构造方法与析构方法	(44)
3.1.3	使用 this 关键字	(45)
3.1.4	访问权限控制	(46)
3.1.5	方法重载	(46)

3.2	继承	(47)
3.2.1	子类继承父类	(47)
3.2.2	方法重写	(48)
3.2.3	关键字 <code>super</code>	(48)
3.3	抽象、接口与多态	(49)
3.3.1	使用关键字 <code>abstract</code> 定义抽象方法 (类)	(49)
3.3.2	使用关键字 <code>interface</code> 定义接口	(50)
3.3.3	使用类继承实现多态、动态绑定	(51)
3.3.4	使用接口实现多态	(53)
3.4	类的高级特征	(54)
3.4.1	使用关键字 <code>static</code> 定义静态成员和静态代码块	(54)
3.4.2	使用关键字 <code>final</code> 定义终结类 (方法或变量)	(55)
3.4.3	内部类 (接口)	(56)
3.4.4	创建接口 (抽象类) 的匿名实现类对象	(57)
3.4.5	在类方法里使用可变参数	(58)
3.4.6	<code>Class</code> 类与 <code>Java</code> 反射机制	(58)
3.4.7	动态代理类 <code>Proxy</code> 与远程过程调用 (PRC)	(61)
	习题 3	(64)
	实验 3	(66)
第 4 章	<code>Java</code> 进程与多线程	(68)
4.1	<code>Java</code> 进程	(68)
4.2	<code>Java</code> 多线程	(69)
4.2.1	线程与多线程概念	(69)
4.2.2	在某个进程中创建多个线程	(69)
4.3	线程同步	(73)
4.4	线程间的协作	(75)
4.5	死锁	(78)
4.6	异步任务处理	(80)
	习题 4	(82)
	实验 4	(83)
第 5 章	<code>Java</code> 集合框架与泛型	(85)
5.1	<code>Java</code> 集合框架概述与泛型	(85)
5.1.1	<code>Java</code> 集合框架的主要接口	(85)
5.1.2	迭代接口 <code>Iterable</code> 与迭代器 <code>Iterator</code>	(86)
5.1.3	<code>Java</code> 泛型	(87)
5.2	<code>Collection</code> 集合及其遍历	(89)

5.2.1	List 接口及其常用实现类	(90)
5.2.2	Set 集合接口及实现类	(93)
5.2.3	队列接口 Queue 及实现类	(95)
5.3	Map 集合及其遍历	(97)
5.3.1	Map 接口	(97)
5.3.2	Map 实现类 HashMap 及其他实现类	(98)
5.3.3	使用 Map.Entry 遍历 Map 集合	(102)
习题 5		(104)
实验 5		(105)
第 6 章	Java UI 设计及图形功能	(107)
6.1	抽象窗口工具集 AWT	(107)
6.1.1	Java AWT 概述	(107)
6.1.2	AWT 常用组件	(109)
6.1.3	布局管理器及常用布局	(110)
6.1.4	Java 图形功能	(112)
6.1.5	Java 事件处理模型	(115)
6.1.6	Java 小程序 Applet	(120)
6.2	Swing UI	(122)
6.2.1	Swing 概述	(122)
6.2.2	JFrame 框架	(123)
6.2.3	Swing 容器组件 JPanel	(123)
6.2.4	Swing 常用组件	(124)
6.3	安装 WindowBuilder 实现 UI 可视化	(135)
习题 6		(138)
实验 6		(139)
第 7 章	Java I/O 操作与文件读写	(142)
7.1	Java I/O 流及分类	(142)
7.2	控制台 I/O	(143)
7.2.1	PrintStream 对象 System.out 和 System.err	(143)
7.2.2	InputStream 对象 System.in	(143)
7.3	文件类 File 及 Java 文件读写	(144)
7.3.1	类 File 的常用方法	(144)
7.3.2	Java 文件读写	(146)
7.4	字节流	(148)
7.4.1	字节流抽象类 InputStream 与 OutputStream	(148)
7.4.2	文件字节流 FileInputStream 和 FileOutputStream	(150)

7.4.3	数据流 DataInputStream 和 DataOutputStream	(150)
7.4.4	字节缓冲流 BufferedInputStream 和 BufferedOutputStream	(152)
7.5	字符流	(154)
7.5.1	字符流抽象类 Reader 和 Writer	(154)
7.5.2	字符流类 InputStreamReader 和 OutputStreamWriter	(155)
7.5.3	文件字符流类 FileReader 和 FileWriter	(156)
7.5.4	字符缓冲流 BufferedReader 和 BufferedWriter	(157)
7.6	对象序列化与反序列化	(159)
习题 7		(161)
实验 7		(162)
第 8 章	Java 网络编程	(164)
8.1	TCP/IP、HTTP 协议与 Socket	(164)
8.1.1	TCP/IP 连接	(164)
8.1.2	HTTP 连接	(165)
8.1.3	Socket 及其工作原理	(165)
8.2	Socket 网络编程	(167)
8.2.1	基于 TCP 和 UDP 的 Socket 编程	(167)
8.2.2	使用多线程支持多客户端	(170)
8.3	基于 URL 的网络编程	(176)
8.3.1	从 C/S 到 B/S	(176)
8.3.2	基于 URL 和 URLConnection 的网络编程	(177)
8.4	Java WebService	(182)
习题 8		(187)
实验 8		(188)
第 9 章	JDBC 编程	(190)
9.1	MySQL 数据库	(190)
9.1.1	关系型数据库与 SQL 语言	(190)
9.1.2	MySQL 服务器软件安装与基本使用	(192)
9.1.3	MySQL 前端工具 SQLyog	(193)
9.2	使用 JDBC 访问 MySQL 数据库	(195)
9.2.1	JDBC 概述	(195)
9.2.2	连接 MySQL 数据库	(196)
9.2.3	创建执行数据库操作的命令对象	(196)
9.2.4	增、删、改操作	(197)
9.2.5	查询操作	(197)
9.2.6	使用预处理封装 MySQL 通用类	(199)

9.3 JDBC 高级编程	(204)
9.3.1 可滚动结果集和可更新结果集	(204)
9.3.2 使用数据库事务机制	(206)
9.3.3 数据库应用程序代码分层架构	(208)
习题 9	(213)
实验 9	(214)
第 10 章 综合项目实训	(216)
10.1 打坦克游戏	(216)
10.1.1 游戏规则及界面设计	(216)
10.1.2 项目文件系统	(216)
10.1.3 项目实施主要代码	(217)
10.2 简易人事管理信息系统	(226)
10.2.1 系统目标	(226)
10.2.2 系统功能	(226)
10.2.3 数据库设计	(226)
10.2.4 界面设计	(227)
10.2.5 主要模块设计	(229)
习题 10	(238)
实验 10	(240)
习题答案	(242)
参考文献	(246)

第 1 章

Java 概述及开发环境

Java 是具有开源特性、跨平台特性及面向对象特性的计算机程序设计语言，广泛应用于移动设备、嵌入式设备上的 Java 应用程序和 Web 应用程序。JDK（Java Development Kit）是整个 Java 开发的核心，它提供了 Java 的运行环境、Java 开发工具包和 Java 基础类库。本章学习要点如下：

- 了解 Java 语言的发展简史、特点及应用三个方向；
- 掌握类与对象的关系，类的封装特性；
- 掌握 Java 集成开发工具 eclipse-jee 的使用；
- 了解 Java 虚拟机的作用；
- 掌握 Java 程序结构；
- 掌握 Java 类库的语言包中常用类的使用；
- 掌握 Java 程序的动态调试和单元测试方法；
- 掌握使用 Maven 管理项目依赖包。

1.1 Java 语言发展简史及应用

1.1.1 Java 语言发展简史

1991 年，Sun 公司的 James Gosling、Bill Joe 等人，为在电视机、烤箱等家用消费类电子产品上进行交互式操作而开发了一个名为 Oak（即一种橡树的名字）的软件，即 Java 语言的前身。当时，这款软件并没有引起人们的注意。直到 1994 年下半年，因特网的迅猛发展，推动了 Java 语言的发展，使得它逐渐成为因特网上最受欢迎的编程语言。2009 年，甲骨文公司（Oracle）收购了 Sun 公司。此后，由甲骨文公司继续推进 Java 语言的发展。

Java 语言发展迅速并被广泛应用，成为目前主流的网络编程语言，这与 Java 语言本身的特点是密切相关的。Java 语言具有简单、动态、面向对象、分布式、解释执行、健壮、安全、结构中立、可移植、高效能、多线程等多种特点。

1.1.2 Java 语言应用的三个方向

根据用途的不同，Java 语言可分为以下三种版本：

- Java SE (Java Standard Edition): Java 标准版, 主要用于桌面级的应用和数据库的开发。
- Java EE (Java Enterprise Edition): Java 企业版, 主要用于企业级开发, 提供企业级 Web 应用开发的各种技术。
- Java ME (Java Micro Edition): Java 移动版, 主要用于嵌入式的、移动式的应用开发, 如手机应用软件开发。

1.2 Java 语言的特点

1.2.1 面向对象特性、解释性与简单性

Java 语言是一种面向对象的语言, 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大大简化了程序设计者的内存管理工作。另外, Java 也适合在小型机上运行, 它的基本解释器及类库代码体量都很小。

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, 实现了代码的复用。

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息, 使得连接过程更加简单。

Java 在编译和运行程序时, 都要对可能出现的问题进行检查, 以避免错误的产生。它提供自动垃圾收集机制来进行内存管理, 防止程序员在管理内存时产生人为错误。通过集成的面向对象的异常处理机制, 在编译时, Java 提示可能出现但未被处理的异常, 帮助程序员正确地进行选择以防止系统崩溃。

注意:

- (1) 面向对象是 Java 的重要特性, 能实现良好的代码重用。
- (2) Java 语言的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承, 使之比严格的类继承具有更灵活的方式和扩展性。
- (3) 与其他解释执行的语言 (如 BASIC) 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, 从而获得较高的性能。
- (4) Java 在编译时还可捕获类型声明中的许多常见错误, 防止出现动态运行时不匹配的问题。

1.2.2 平台无关性（可移植性）

为了使程序能在任何平台上运行而不需要程序员针对不同的平台重写或重编译，Java 虚拟机（Java Virtual Machine, JVM）使这个愿望变为可能。因为它能知道每条指令的长度和平台的其他特性。JVM 是通过在实际的计算机上仿真模拟各种计算机功能来实现的。Java 虚拟机有自己完善的硬件架构，如处理器、堆栈、寄存器等，还具有相应的指令系统。JVM 屏蔽了与具体操作系统平台相关的信息，这使得 Java 程序只需要生成在 Java 虚拟机上运行的目标代码（字节码），就可以在多种平台上不加修改地运行。Java 虚拟机在执行字节码时，最终还是把字节码解释成具体平台上的机器指令后执行。

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功能，使这些类库可以移植。另外，Java 编译器是由 Java 语言实现的，Java 运行时环境由标准 C 语言实现，这使得 Java 系统本身也具有可移植性。

Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行时环境，Java 程序就可在任意的平台上运行。

作为一种独立于硬件平台的编程语言，Java 代码比本地代码慢一些，但随着技术的不断发展，Java 语言的表现不牺牲可移植性的前提下也正在不断地接近本地代码。

通过编译器，可以把 Java 程序源代码编译成一种中间代码，称为字节码。字节码可以被视为运行在 Java 虚拟机上的机器代码指令，可以运行在任何 JVM 上。Java 字节码使得“Java 程序编译一次，到处运行”成为可能。

1.2.3 健壮性与安全性

Java 不需要进行指针运算和存储器管理，从而简化设计，减少出错的可能性。

用于网络、分布式环境下的 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。

1.2.4 多线程特性

多线程机制使应用程序能够并行执行，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，可以分别用不同的线程完成特定的行为，而无须采用全局的事件循环机制，这样可以很容易地实现网络上的实时交互行为。

1.2.5 网络支持

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。事实上，Java 已经作为开发服务器端和移动设备应用的程序设计语言（将在后续课程 Java EE 和 Android 中详细介绍）。

1.3 Java 运行环境及开发环境

1.3.1 安装 JDK 并建立环境变量

SDK (Software Development Kit) 是一个使用广泛的专业名词, 是指软件开发工具包, 是辅助开发某一类软件的相关文档、范例和工具 (包括方法库、编译程序等) 的集合。

JDK 是 Sun Microsystems 公司提供给 Java 开发员的 SDK, 它提供了 Java 的开发环境和运行时环境 (Java Runtime Enviroment, JRE), 用于构建在 Java 平台上运行的应用程序、Applet (小程序) 和组件等。

如同 Win 32 应用程序需要操作系统来运行一样, Java 程序需要 JRE 才能运行。

注意: JRE 面向 Java 程序的使用者, 而不是开发者。

访问<http://download.oracle.com>, 进入官网下载页面, 选择 Java SE 及对应版本进行下载和安装。本书使用 JDK 8, 安装后的文件系统如图 1.3.1 所示。

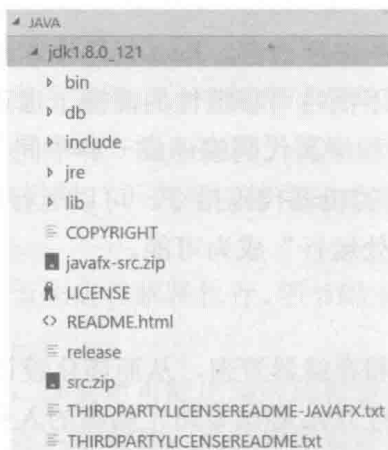


图 1.3.1 安装 JDK 后的文件系统

系统文件夹里的主要文件夹及文件的含义如下:

- 文件夹 bin 提供 JDK 工具程序;
- 文件夹 jre 存放 Java 运行时环境文件;
- 文件夹 lib 存放 Java 的类库文件, 即工具程序使用的 Java 类库;
- 文件夹 include 存放用于本地方法的文件;
- 文件 src.zip 是 Java API 的源代码压缩文件。

为了在命令行方式下方便使用 JDK 工具程序, 需要建立 Windows 环境的系统变量 Path, 其值为 %Java_Home%\bin。其中, Java_Home 也是系统变量, 其值为 JDK 的安装路径, 如图 1.3.2 所示。

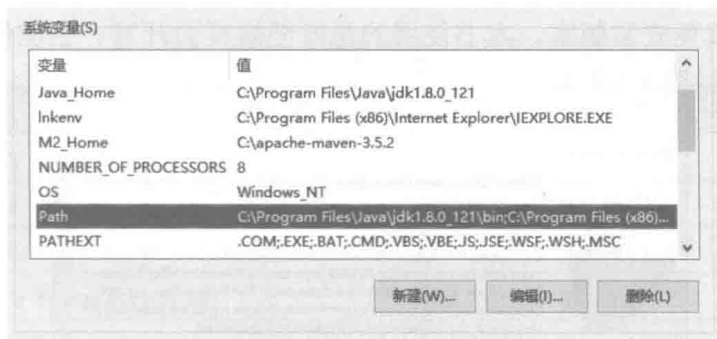


图 1.3.2 Windows 环境下的系统变量设置

注意：

- (1) 系统变量 `Java_Home` 在后续课程 Java Web 中使用 Tomcat 时是必须配置的。
- (2) 为方便在命令行方式下使用 JDK 工具，通常需要建立 Windows 环境变量 `Path`。

1.3.2 JDK 工具箱与 Java 虚拟机

JDK 工具箱提供了众多的 Java 开发工具，下面介绍其中常用的几个。

- (1) `javac.exe`: Java 编译工具，用于编译 Java 源代码文件。
- (2) `java.exe`: Java 运行工具，用于运行 `.class` 字节码文件或 `.jar` 文件。
- (3) `jar`: `jar` 文件管理工具，主要用于打包压缩、解压 `.jar` 文件。
- (4) `appletviewer.exe`: 用于运行并浏览小程序，参见 6.1.6 节。
- (5) `wsimport`: XML Web Service 2.0 的 Java API，主要用于根据服务端发布的 `.wsdl` 文件生成客户端存根及框架，参见 8.4 节。

Java 平台由两大部分组成：Java 虚拟机 (JVM) 和 Java 应用程序编程接口 (Java Application Programming Interface, Java API)。

JVM 是在不同硬件、不同操作系统之上定义的完全相同的支持 Java 程序运行的虚拟计算机，它隐藏了复杂的计算机硬件和操作系统，使程序员只需要面对单一的支持 Java 的计算机。

Java 编译器将 Java 源代码转换成 JVM 的指令序列 (字节码)，保存在 `.class` 文件中。执行 Java 程序时，JVM 负责解释字节码，将 JVM 的指令转换成真实的机器指令后执行。

1.3.3 Java 集成开发环境 eclipse-jee

早期的 Java 开发，使用记事本之类的文本编辑工具编辑源文件，使用 JDK 工具编译 Java 源程序而得到 `.class` 文件。再借助于 JRE，使用 Java 运行工具来运行 Java 应用程序。这种手工开发方式，其过程较为烦琐、效率低下，不推荐使用。

注意：手工开发方式，需要建立 Windows 系统环境变量 `Classpath`。

eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件和组件构建开发环境。幸运的是，eclipse 附带了一个标准的插件集，包括 Java 开发工具。访问 eclipse 官网下载专区 (<https://www.eclipse.org/>)

downloads) 可下载免安装版本。本书使用的是既能做桌面开发、也能做 Java Web 开发的 Oxygen 版本, 如图 1.3.3 所示。



图 1.3.3 eclipse 的 Oxygen 版本

eclipse 软件提供了工具栏和菜单两种使用方式。

对于 Java 课程的学习, 主要使用工具  来运行程序, 使用工具  以调试模式运行程序。

菜单操作也是必须掌握的。例如, 为了实现在保存 Java 类时自动编译, 应选择菜单 Project→Build Automatically (显示为打勾状态)。

注意:

(1) 教学网站 <http://www.wustwzx.com> 上本课程的下载专区, 提供了 eclipse-jee 的下载链接。

(2) 在 eclipse 集成环境中, 不需要建立 Windows 系统环境变量 Classpath。

(3) 使用集成环境 eclipse 后, 保存源文件时就自动编译生成字节码。此时, 查看系统文件夹中 bin 文件夹里的 .class 文件可验证这一点。

(4) 在 eclipse 中, 按下运行按钮, 就会调用 JDK 解释运行字节码。

(5) 在命令行方式下编译含有引用第三方 jar 包的 Java 源程序时, 需要建立系统变量 Classpath, 其值为相关类所在的路径。

1.3.4 统一 eclipse 项目编码

进入 eclipse 安装目录, 使用 NotePad 之类的文本编辑软件, 打开 eclipse 配置文件, 在最后增加一行代码:

```
-Dfile.encoding=utf-8
```

1.3.5 eclipse 若干快捷操作

软件开发需要讲究时间效率, 使用快捷操作能提高开发效率。在 eclipse 开发环境中, 常用的快捷操作如表 1.3.1 所示。

表 1.3.1 eclipse 开发环境中的若干快捷操作

功 能	操作（或快捷键）
快速选取文本，供复制和修改用	双击文本
项目或文件的重命名	选中对象，按功能键 F2
搜索包含特定字符的文档	Ctrl+H
产生控制台输出命令 System.out.println()	输入 sysout 后按回车键
关闭所有已打开的文档	Ctrl+Shift+W
最大化（或还原）编辑或信息显示窗口	双击标题栏
自动导入所需要（或去掉不必引入）的软件包	Ctrl+Shift+O
程序或页面文档格式化	Ctrl+Shift+F
产生类的 main() 方法块	输入 main 后按回车键
快速修复：包括自动导包、创建类实例时自动补全等	输入 new 类名 () 后按快捷键 Ctrl+I，然后单击 Assign statement to new local variable (Ctrl+2, L)
产生类属性的所有 get/set 方法	空白处单击右键→Source→Generate Getters and Setters
自动生成实体类的 toString() 方法	空白处单击右键→Source→Generate toString()
自动生成要实现的接口方法块	在类名前出现  时，单击  ，在出现提示信息 must implements the inherited abstract method 时，单击 Add unimplemented methods
查看类（或接口）提供的所有方法	按 Ctrl 键，当指针在类（或接口）名上呈现超链接时单击，再单击左边  Package Explorer 窗口里的 
注释代码	选中文本后按快捷键 Ctrl+Shift+/
标签、标签属性、属性值和重写方法等的自动提示	Alt+/
标识符关联修改（可自动修改类名）	Alt+Shift+R
取消代码注释	选中文本后按快捷键 Ctrl+Shift+\
删除光标所在的一行	Ctrl+D
自动产生 try...catch 代码块	当类名前出现  时，单击  ，在出现提示信息 Unhandled exception type Exception 时，单击  Surround with try/catch
查看程序（类）结构	先按快捷键 Alt+Shift+Q，松开后再按 O 键，或使用菜单 Window→Show View→Outline

注意：在分析、修改别人的项目时，可能会重新命名某个类名。在相应的类文件里，使用快捷键 Alt + Shift + R 修改类名标识符进而修改类名，是非常快捷的方式。

1.4 Java 面向对象初步

1.4.1 类与对象、封装特性

计算机程序设计的本质就是将现实生活中遇到的问题抽象后，利用计算机语言转化