

内 容 提 要

本书以 Java 技术为背景,主要介绍了 Java 应用开发的基础语法和相关技术。本书共分为 9 个模块,介绍了 Java 基础语法、类和接口、Java 常用 API 和异常、集合和泛型、读写文件、数据库的连接和网络编程技术等,由基础到进阶,层层递进。学习者可以通过学习本书掌握 Java 语言的开发技术。

本书内容全面、结构清晰,以实现任务为目的,任务的设置既符合学习者的认知规律,又体现了 Java 语言知识和技能的连贯性。改变了以往介绍知识点时枯燥、晦涩、抽象和难以理解的状况,在知识点的阐述中融入了上百个例子,直接、形象、生动,让学习者能够融会贯通地掌握知识和训练技能,同时也大大提高了学习者的学习兴趣。

本书既可作为 Java 程序设计课程的教材,又可作为 Java 语言的职业培训教材或 Java 语言爱好者的参考用书。

图书在版编目(CIP)数据

Java 程序设计/邓会敏,廖福保,黄镔主编. —北
京:航空工业出版社,2023.4
ISBN 978-7-5165-3323-9
I. ①J… II. ①邓… ②廖… ③黄… III. ①JAVA 语
言-程序设计 IV. ①TP312.8
中国国家版本馆 CIP 数据核字(2023)第 055287 号

Java 程序设计 Java Chengxu Sheji

航空工业出版社出版发行

(北京市朝阳区京顺路 5 号曙光大厦 C 座四层 100028)

发行部电话:010-85672663 010-85672683

北京荣玉印刷有限公司印刷

全国各地新华书店经售

2023 年 4 月第 1 版

2023 年 4 月第 1 次印刷

开本:889 毫米×1194 毫米 1/16

字数:524 千字

印张:17.5

定价:56.00 元



模块 1 开发第一个 Java 程序/1

任务 1.1 认识 Java 语言	2	1.2.3 测试环境配置	8
1.1.1 计算机语言	2	任务 1.3 运行第一个 Java 程序	9
1.1.2 面向过程和面向对象	3	1.3.1 编写 Java 源程序	9
1.1.3 Java 语言的发展	3	1.3.2 编译和运行	10
1.1.4 Java 工作原理	4	任务 1.4 使用 Eclipse 集成开发环境	11
1.1.5 Java 语言的特点	4	1.4.1 下载和安装 Eclipse	11
任务 1.2 下载和安装 JDK	5	1.4.2 启动 Eclipse	13
1.2.1 下载和安装 JDK	5	1.4.3 开发 Java 项目	14
1.2.2 配置环境变量	6	任务 1.5 综合实训	17



模块 2 Java 基础语法/18

任务 2.1 初识简单数据类型	19	2.3.2 while 语句	51
2.1.1 标识符和关键字	19	2.3.3 do-while 语句	53
2.1.2 注释	20	2.3.4 break 语句	55
2.1.3 变量	20	2.3.5 continue 语句	56
2.1.4 常量	22	2.3.6 循环嵌套	56
2.1.5 数据类型	23	实战演练 2-3 百钱买百鸡问题	61
2.1.6 运算符	27	任务 2.4 使用数组	62
实战演练 2-1 输出学生信息表	35	2.4.1 一维数组	62
任务 2.2 实现选择结构	37	2.4.2 二维数组	70
2.2.1 数据输入	37	实战演练 2-4 幸运抽奖	75
2.2.2 if 语句	38	任务 2.5 设计方法	76
2.2.3 switch 语句	44	2.5.1 方法的定义	76
实战演练 2-2 购买机票	47	2.5.2 方法的调用	76
任务 2.3 实现循环结构	48	任务 2.6 综合实训	80
2.3.1 for 语句	48		



模块3 使用类开发程序/82

任务 3.1 使用类	83	3.3.3 覆盖	101
3.1.1 辨识类和对象	83	3.3.4 多态性	103
3.1.2 创建类	83	实战演练 3-2 封装并输出博士生信息表	105
3.1.3 创建对象	85	任务 3.4 设计构造方法	107
3.1.4 方法重载	86	3.4.1 定义构造方法	107
3.1.5 静态变量	88	3.4.2 重载构造方法	109
3.1.6 静态方法	89	3.4.3 继承关系中的构造方法	111
实战演练 3-1 封装并输出学生信息表	90	实战演练 3-3 优化设计输出学生信息表	113
任务 3.2 使用包	92	任务 3.5 使用抽象类	115
3.2.1 创建包	92	任务 3.6 使用接口	117
3.2.2 导入包	94	3.6.1 定义接口	117
任务 3.3 实现继承	95	3.6.2 使用接口	118
3.3.1 继承的概念	95	任务 3.7 综合实训	121
3.3.2 访问控制	98		



模块4 常用 API 和异常/124

任务 4.1 使用字符串类	125	4.2.4 包装类	136
4.1.1 String 类	125	任务 4.3 处理异常	138
4.1.2 StringBuffer 类	128	4.3.1 异常的概念	138
实战演练 4-1 统计子串在母串中出现的次数	130	4.3.2 捕捉异常	139
任务 4.2 使用工具包	131	4.3.3 抛出异常	144
4.2.1 数学类	131	4.3.4 异常类的层次结构	146
4.2.2 日历类	133	任务 4.4 综合实训	146
4.2.3 格式化工具	135		



模块5 集合和泛型/148

任务 5.1 熟识 Java 集合	149	任务 5.4 使用 Map 集合	161
5.1.1 Java 集合	149	5.4.1 Map 接口	161
任务 5.2 使用 List 集合	150	5.4.2 HashMap 类	162
5.2.1 List 接口	151	实战演练 5-1 管理学生信息	165
5.2.2 ArrayList 类	151	任务 5.5 使用泛型	166
5.2.3 Iterator 接口	153	5.4.1 使用泛型方法	166
5.2.4 foreach 循环	156	5.4.2 使用泛型类	169
任务 5.3 使用 Set 集合	157	5.4.3 使用泛型集合	171
5.3.1 Set 接口	157	任务 5.6 综合实训	174
5.3.2 HashSet 类	158		



模块 6 读写文件/178

任务 6.1 认识 File 类	179	6.3.3 FileWriter	189
6.1.1 创建 File 对象	179	6.3.4 BufferedReader 和 BufferedWriter	191
6.1.2 File 类的常用方法	180	任务 6.4 字节流的方式读写文件	196
任务 6.2 熟识 IO 流	184	6.4.1 InputStream 和 OutputStream	196
6.2.1 认识 IO 流	184	6.4.2 FileInputStream	197
6.2.2 字节流和字符流	184	6.4.3 FileOutputStream	199
任务 6.3 字符流的方式读写文件	185	6.4.4 BufferedInputStream 和	
6.3.1 Reader 和 Writer	185	BufferedOutputStream	201
6.3.2 FileReader	186	任务 6.5 综合实训	203



模块 7 连接数据库/210

任务 7.1 JDBC 连接数据库	211	任务 7.3 实现更新操作	220
7.1.1 认识 JDBC	211	7.3.1 使用 Statement 更新数据	220
7.1.2 连接数据库	212	7.3.2 使用 PreparedStatement 更新数据	222
任务 7.2 实现查询操作	218	任务 7.4 综合实训	225



模块 8 多线程编程/229

任务 8.1 认识多线程	230	8.1.4 线程的生命周期	233
8.1.1 线程与进程	230	实战演练 8-1 模拟多个售卖点售卖电影票 ..	234
8.1.2 继承 Thread 类实现多线程	230	任务 8.2 设计多线程的同步	235
8.1.3 使用 Runnable 接口实现多线程	232	任务 8.3 综合实训	238



模块 9 网络编程/240

任务 9.1 熟悉网络编程基础知识	241	9.2.5 基于多线程的通信聊天程序	254
9.1.1 TCP/IP 概述	241	任务 9.3 Java 中的 UDP 编程	258
9.1.2 InetAddress	244	9.3.1 UDP 协议	258
任务 9.2 Java 中的 TCP 编程	247	9.3.2 DatagramSocket 类	259
9.2.1 TCP 协议	247	9.3.3 DatagramPacket 类	260
9.2.2 ServerSocket 类创建 TCP 服务器端	247	9.3.4 基于 UDP 的网络通信程序	262
9.2.3 Socket 类实现 TCP 客户端	249	任务 9.4 综合实训	264
9.2.4 简单的 TCP 网络程序	250		



参考文献/268

模块 1

开发第一个 Java 程序

学习目标

知识目标

- (1) 了解 Java 语言的特点。
- (2) 理解 Java 的工作原理。
- (3) 熟悉 Java 程序的基本构成。
- (4) 掌握 Java 程序的开发过程。

技能目标

- (1) 会下载和安装 JDK 和 Eclipse。
- (2) 会配置环境变量。
- (3) 会使用 Eclipse 编写、编译、运行 Java 程序。

素质目标

- (1) 激发科技报国的家国情怀。
- (2) 激发对软件行业的热情，提高编程兴趣，坚定自己的理想信念。

任务 1.1 认识 Java 语言



【任务导入】

必须坚持自信自立，是党的二十大报告对习近平新时代中国特色社会主义思想世界观和方法论高度凝练、科学概括提出的“六个必须坚持”之一。坚持自信自立，是中国式现代化的题中要义。在新中国成立特别是改革开放以来长期探索和实践基础上，经过党的十八大以来在理论和实践上的创新突破，我们党成功推进和拓展了中国式现代化。

我国拥有世界上门类最齐全的工业体系，连续多年位居世界第一制造业大国。工业软件作为关键软件的重要组成部分，在建设制造强国和网络强国中发挥了关键支撑作用，是提升我国制造业水平的重要手段。根据国家有关部门发布的数据，2022 年我国软件和信息技术服务业运行态势良好，软件业务收入跨上 10 万亿元台阶。软件业利润总额 12648 亿元，同比增长 5.7%。软件业务出口 524.1 亿美元，同比增长 3.0%。我国软件行业的发展拥有远大的前景。

Java 语言是我国软件行业使用最广泛的编程语言之一。Java 是由 Sun Microsystems 公司（2010 年被 Oracle 公司收购）于 1995 年 5 月开发的面向对象的高级程序设计语言。Java 可运行于多个平台，如 Windows、Mac OS 及其他多种 Unix 版本的系统。它具有安全性、跨平台性、可移植性、分布式等显著特点，功能强大、易学易用。同时，Java 与 Internet 结合紧密，已经成为当今主要的网络应用开发语言。自问世以来，Java 语言受到众多软件开发人员的喜爱，并得到广泛的应用。本任务就是初步认识 Java 语言。

思考

- (1) Java 究竟有哪些特点，为什么会受到软件开发人员的喜爱？
- (2) Java 的工作原理是什么？

1.1.1 计算机语言

计算机语言（Computer Language）是人与计算机之间通信的语言，它主要由一些指令组成，这些指令包括数字、符号和语法等，软件开发人员可以通过这些指令来指挥计算机进行各种工作，如图 1.1 所示。

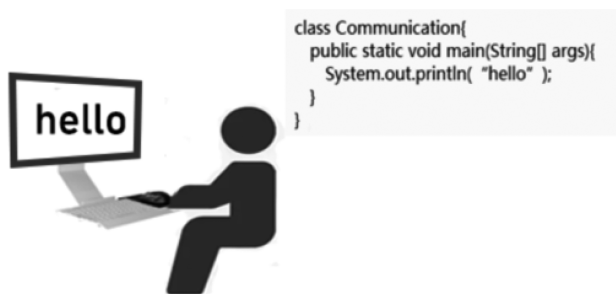


图 1.1 计算机语言——人机交流

计算机语言的种类非常多，主要分为机器语言、汇编语言和高级语言三大类。计算机所能识别的语言只有机器语言（由 0 和 1 组成），但人们在编写程序时通常不采用机器语言，这是因为机器语言都是由二进制的 0 和 1 组成的编码，不便于记忆和识别，不方便编写程序。汇编语言采用了英文缩写的标识符，较容易识别和记忆，但是编写程序仍然不够简便。高级语言是最接近人类语言的天然语言，使用高级语言进行编程进一步简化了程序编写

的过程，所以，高级语言是目前绝大多数软件开发人员的选择，常见的 Java 语言、Python 语言和 C 语言都是高级语言。

1.1.2 面向过程和面向对象

在程序设计中有两类主流的设计方法：一类是面向过程的结构化程序设计方法；另一类是面向对象的程序设计方法。

面向过程的程序设计方法的关键点是“过程”，它是一种“自上而下”的设计模式，先确定好框架，再增砖添瓦，即以“过程”为中心，先设计 `main()` 函数，再逐步实现 `main()` 函数中所要用到的其他方法。其基本设计思路就是分析出解决问题所需要的步骤，然后用函数把这些步骤一个一个地实现，使用的时候依次调用就可以了。面向过程的程序设计方法注重“自顶向下，逐步分解”，把大问题分解成若干个小问题，将功能进行抽象，降低问题的复杂度。

面向对象的程序设计方法的关键点是“对象”，它是一种“自下而上”的设计模式，其基本思想是一切皆对象，把现实中的事物都抽象成程序设计中的“对象”，先设计组件（对象），再完成拼装（对象与对象之间相互作用）。其基本设计思路就是对所需求解的问题域进行分析，识别出问题域中的对象，分析出各个对象之间的相互关系，然后通过各个对象的相互作用解决问题。面向对象的程序设计方法在编程的时候会尽可能地模拟现实世界，按照现实世界中的逻辑处理问题，分析问题中涉及哪些实体，这些实体应该有什么属性和方法，如何通过调用这些实体的属性和方法解决问题。

以火车运行为例来理解面向过程和面向对象。火车运行时，面向过程涉及三个过程：火车启动、火车行驶和火车到站。在设计时，将这三个过程分别编写成函数，程序设计时关心的是这三个过程，而不是火车本身。而面向对象需要建立一个“火车”的实体类，由实体来引发事件，程序设计时关心的是由“火车”实体类实例化成的对象，“火车”对象有自己的属性，如长度、速度和颜色等。“火车”对象还有自己的方法，如启动、行驶、加速、刹车和到站等。其中的方法是“火车”对象的行为动作，而不是火车的每个事件。

Java 语言是面向对象的程序设计语言，C 语言是面向过程的程序设计语言。

1.1.3 Java 语言的发展

1991 年，Sun 公司为了开拓电视机顶盒、微波炉和烤面包机等消费类电子产品市场，启动了一个名为“Green”的项目。该项目的目的是与这些电子产品进行信息交流，从而控制它们，而项目组的负责人正是 James Gosling——Java 之父。当时，Gosling 准备采用当时最流行的编程语言 C++ 来进行开发，但是不久后，他发现 C++ 太过复杂而且庞大，安全性不够好，运行程序所需要的内存和处理器的成本，已经超过了家电的成本。于是在 1991 年 6 月，他准备开发一种新的编程语言，给它起一个什么样的名字呢？猛然，Gosling 回首向窗外望去，看见一棵老橡树，他顿生灵感，新建了一个名叫 Oak 的目录，这就是 Java 语言的前身。后来由于 Oak 已是 Sun 公司的另一种语言的注册商标，Gosling 在 1995 年把他创建的编程语言改名为 Java（太平洋上一个盛产咖啡的岛屿的名字），并在当年的 5 月 23 日正式对外发布 Java 开发工具包 JDK（Java SE Development Kit）1.0 版本，Java 语言就这样正式诞生了。JDK 的发展历经了 JDK 1.0、JDK 1.1、JDK 1.2、JDK 1.3 和 JDK 1.4，到 JDK 1.5 时，由于这个版本的进步是巨大的，所以不再沿用以前的版本编号，于是把这个版本更名为 JDK 5.0。截至 2023 年 2 月，JDK 的版本已经发展到了 JDK 20。

Java 技术体系主要有 3 个技术平台。

（1）Java SE（Java Platform, Standard Edition）：Java 的标准版，主要用于开发桌面程序和商务应用程序。该版本包含了 Java 的最核心类库，同时也包含了支持 Java Web 服务开发的类，并为 Java EE 提供基础支持。本书的主要内容就是介绍 Java SE 的使用。

（2）Java EE（Java Platform, Enterprise Edition）：Java 的企业版，主要用于开发企业级分布式的应用，比如大型的网站系统。该版本包含了 Web 应用开发的各种类库，如 Servlet、JSP、EJB 等。

(3) Java ME (Java Platform, Micro Edition): Java 的小型版, 主要用于嵌入式系统和移动通信设备开发, 比如手机、汽车导航仪等。当前的绝大部分手机都支持 Java 技术。对手机游戏开发感兴趣的同学可以关注该版本。

1.1.4 Java 工作原理

一般的高级语言是如何工作的呢? 通常程序员在编辑器或者 IDE (集成开发环境) 里根据每种高级语言的语法规则编写源代码。源代码程序在计算机运行时, 机器是不能识别高级语言的, 因此需要将高级语言转换成机器能够识别的机器语言——二进制代码, 即生成二进制文件, 这个过程就是编译过程, 由编译器完成。简单地说, 就是程序员写好源代码, 编译器将源代码编译成可执行的机器码, 然后 CPU 读取机器码, 执行程序。

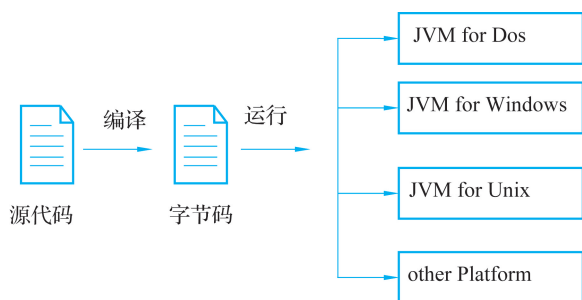


图 1.2 Java 工作原理

那么 Java 语言是如何执行的呢? 它同样遵循这样的过程: 源代码→机器码。但是, Java 源代码 (文件后缀名是 .java) 经过 Java 编译器 (javac.exe) 编译之后, 并没有直接转化为机器码, 而是转化成一种中间格式——字节码 (文件后缀名是 .class), 字节码再经过 Java 虚拟机转化成机器码。正是因为这一中间格式, Java 才有了跨平台性。比如, 在 Windows 平台上编译好的字节码, 复制到 Unix 平台后, 只要经过为 Unix 平台设计的 Java 虚拟机 JVM (Java Virtual Machine) 转化后即可执行, 如图 1.2 所示。

1.1.5 Java 语言的特点

知道了 Java 语言的发展历史后, 再来了解 Java 语言的特点。Java 语言是一门非常具有创造力的面向对象的语言, 经久不衰并不断发展, 得到了广泛的应用, Java 的流行主要是因为它具有以下特点。

1. 简单易学

Java 的简单首先体现在它具有精简的系统, 并对硬件的要求不高。其次 Java 语言的语法与 C 语言非常接近, 大多数程序员很容易学习和使用。另外 Java 语言不再使用容易出错的指针, 而且省略了 C++ 语言中难以理解、容易混淆的特性, 如运算符重载、虚拟基础类等, 并提供了对内存的自动管理机制, 使得程序员不必为内存管理而担忧, 让程序设计更加简单。

2. 面向对象

Java 语言是一门完全面向对象的程序设计语言, 提供类、接口和继承等原语, 为了简单起见, Java 语言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。实现了代码的复用, 缩短了软件开发的时间, 降低了成本。

3. 健壮性

Java 的异常处理机制、强类型机制、内存自动管理机制等是 Java 程序健壮性的重要保证。

4. 跨平台性

所谓跨平台性, 是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。Java 语言是一种解释型语言, Java 源代码在 Java 平台上被编译为与体系结构无关的字节码格式文件, 任何类型的计算机只要安装了 Java 运行环境或配备了 Java 解释器, 字节码文件就可以在该计算机上运行, 实现了“一次编写, 处处运行”。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联接阶段被载入运行环境中。

5. 支持多线程

在 Java 语言中，线程是一种特殊的对象。多线程机制使得应用程序可以并发执行，而且同步机制保证了对共享数据的正确操作，这样可以在很大程度上提高程序的运行效率。

6. 分布性

Java 是支持网络编程的语言，在网络程序设计上非常具有优势。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上方方便地访问其他对象。

任务 1.2 下载和安装 JDK



【任务导入】

开发 Java 程序，必须安装 Java 语言的开发工具包 JDK。JDK 是一切 Java 应用程序的基础，是整个 Java 开发的核心，它包含了 Java 的运行环境和 Java 工具，它是一组应用程序编程接口（Application Programming Interface, API），即 Java Class。

安装和调试 Java 程序的运行环境，看似是一个简单的过程，但是由于 JDK 版本较多以及个人计算机的软硬件配置不同，导致在实际的操作中会遇到不少的问题，这需要我们充满信心，大胆尝试。

思考

- (1) 如何下载和安装 JDK?
- (2) 如何配置环境变量?

1.2.1 下载和安装 JDK

使用 Java 语言编写程序，首先必须下载和安装 Java 语言的开发工具包 JDK。

(1) 登录 Oracle 公司的官方网站 (<https://www.oracle.com>)，下载 JDK 的最新版本，如图 1.3 所示。在 JDK 发展的各版本中，JDK 8 是比较稳定的版本。本书使用 JDK 8 讲解相关内容，并为读者提供了 Windows 平台的 JDK 8 版本的安装包。

(2) 双击“jdk-8u261-windows-x64_bin.exe”可执行文件，进入安装界面，如图 1.4 所示。在弹出的界面中单击“更改”按钮，对 JDK 安装目录进行设置和修改，在安装路径的选择上，建议安装在 C 盘。设置好安装路径后（也可以选择默认路径），单击“下一步”按钮进行安装。



1.1 下载和安装 JDK 和 Eclipse 微课视频

Linux macOS Windows		
Product/file description	File size	Download
x64 Compressed Archive	172.79 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip (sha256 )
x64 Installer	153.37 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe (sha256 )

图 1.3 Oracle 官网

(3) 接下来根据安装提示继续进行安装，当出现如图 1.5 所示的界面时，表明 JDK 安装成功。



图 1.4 JDK 安装界面 (1)



图 1.5 JDK 安装界面 (2)

1.2.2 配置环境变量

安装好 JDK 之后，在不使用 IDE 集成开发工具的情况下，还需要配置环境变量，环境变量的作用是保存一系列路径。当通过 Java 指令运行一个程序时，计算机除了会在当前路径中寻找相关指令之外，还可以到环境变量所保存的路径中寻找，这样在任何位置输入 Java 的编译指令或调试指令都可以执行，否则必须回到 JDK 的安装目录下才能执行相关指令。需要设置的环境变量为系统路径变量 Path，Path 是路径环境变量，Path 中所有路径的内容在任何位置都可以使用。

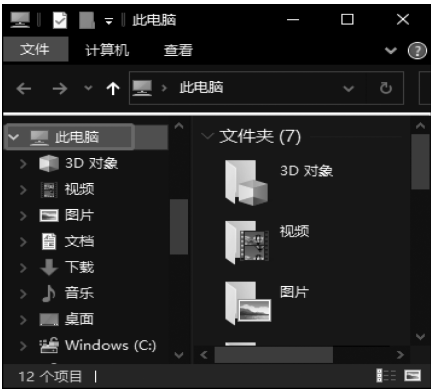


图 1.6 右击“此电脑”

设置系统的环境变量（JAVA_HOME、Path 和 classpath）的步骤如下（以 Windows 10 操作系统为例）。

(1) 右击“此电脑”图标，在弹出的快捷菜单中单击“属性”，如图 1.6 所示。

(2) 依次单击“设置”→“关于”→“高级系统设置”，如图 1.7 所示。

(3) 在弹出的“系统属性”对话框中，单击“环境变量(N)...”按钮，如图 1.8 所示。



图 1.7 单击“高级系统设置”



图 1.8 “系统属性”对话框

(4) 新建环境变量 JAVA_HOME。在“系统变量(S)”处单击“新建”按钮，在弹出的“新建系统变量”的“变量名(N)”处输入“JAVA_HOME”，在“变量值(V)”处输入 JDK 的安装路径“C:\Program Files\Java\jdk1.8.0_261;”，然后单击“确定”按钮即可，如图 1.9 所示。

(5) 编辑环境变量 Path。在“环境变量”对话框的“系统变量(S)”中，选择变量参数“Path”，单击“编辑(I)...”，如图 1.10 所示。在弹出的“编辑环境变量”对话框中，单击“新建(N)”按钮新建路径，分别输入“%JAVA_HOME%\bin”和“%JAVA_HOME%\jre\bin”，单击“确定”即可，如图 1.11 所示。

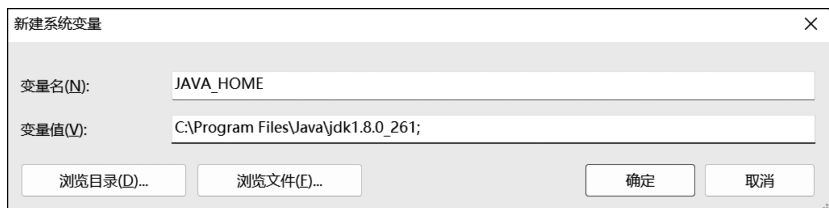


图 1.9 “环境变量”对话框

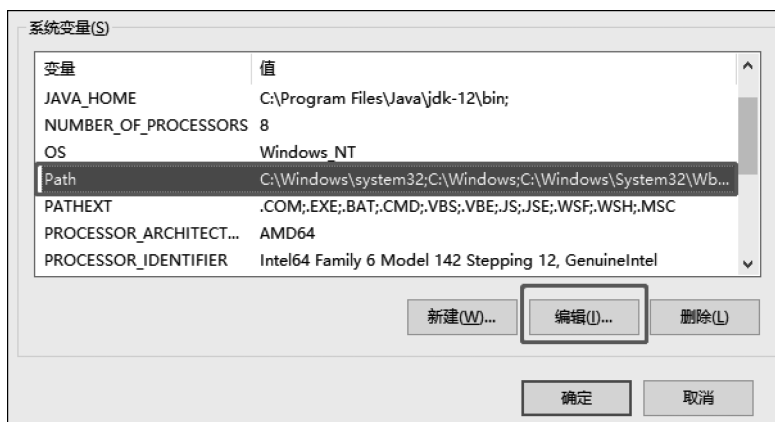


图 1.10 设置环境变量 Path

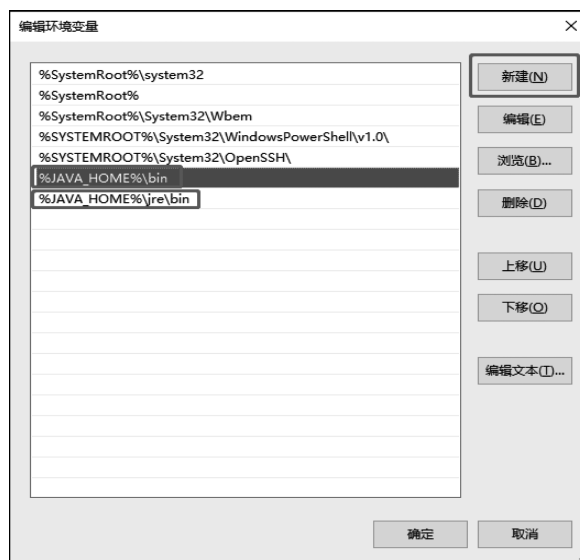


图 1.11 新建两条路径

(6) 新建环境变量 classpath。在“环境变量”对话框的“系统变量(S)”中单击“新建(W)...”按钮，如图 1.12 所示。在弹出的“新建系统变量”对话框的“变量名(N)”处输入“classpath”，“变量值(V)”处输入“.;%JAVA_HOME%\lib”，然后单击“确定”即可，如图 1.13 所示。这样环境参数就设置好了。

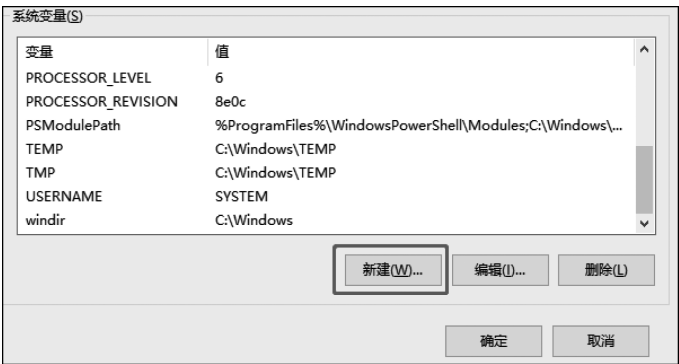


图 1.12 新建环境变量“classpath”

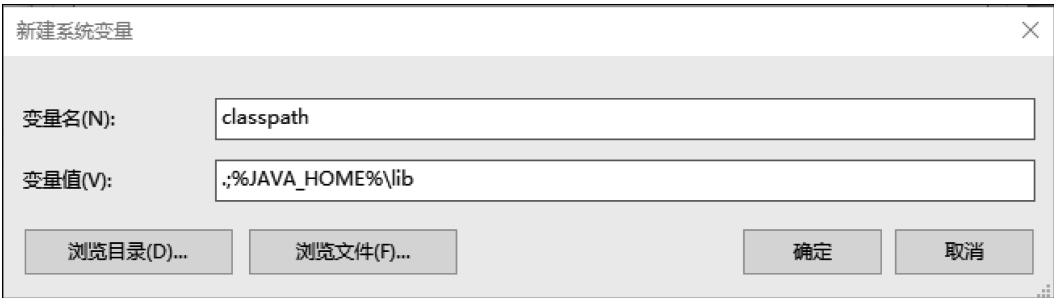


图 1.13 设置环境变量“classpath”的值

说明

- (1) 在设置环境变量时，必须切换到英文输入模式后再输入相应的路径。
- (2) %JAVA_HOME%是 JDK 的安装路径，相当于“C:\Program Files\Java\jdk1.8.0_261”。
- (3) 设置环境变量 Path 的目的是告诉操作系统 javac、java 等命令程序存放的位置，设置环境变量 classpath 的目的是告诉 JVM 可执行文件的存放位置。
- (4) 借助 JDK 开发 Java 程序，学会环境变量的配置是非常重要的。

1.2.3 测试环境配置

前面已经配置了环境变量，配置是否成功还需要手动测试一下。测试方法如下。

- (1) 右击“开始”菜单，执行“运行”命令，输入“cmd”启动控制台，如图 1.14 所示。



图 1.14 控制台

(2) 在控制台中输入“javac”，控制台会输出 JDK 编译命令的帮助信息，如图 1.15 所示。成功输出帮助信息表示配置成功。

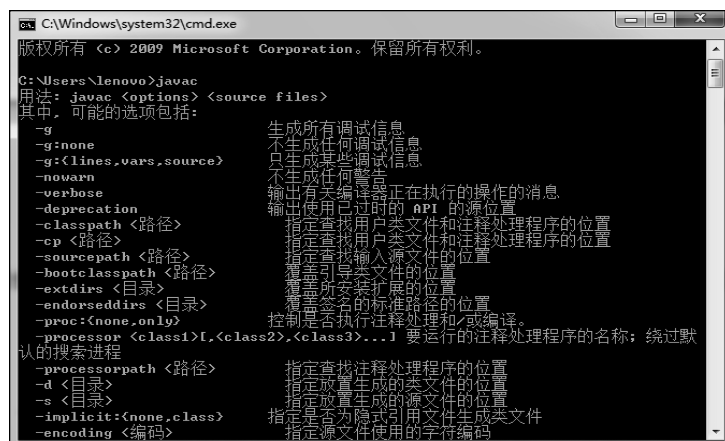


图 1.15 编译命令的帮助信息

任务 1.3 运行第一个 Java 程序



【任务导入】

安装好 JDK 并配置了相关的环境参数之后，就可以编写并运行第一个 Java 应用程序了。

思考

- (1) Java 程序的基本结构是什么？
- (2) 如何编写并运行 Java 程序？

1.3.1 编写 Java 源程序

可以使用任何一种文本编辑器编写 Java 源代码，如记事本、Word、EditPlus 等，必须要注意命名文件时，文件的后缀名必须是 .java（即 Java 源程序文件的后缀名是 .java）。然后用 JDK 提供的命令来编译和运行。编写 Java 源程序的步骤如下。

- (1) 在 D 盘下新建一个目录 studyjava。
- (2) 在该目录下新建一个文件 Test.java，使用记事本打开该文件并输入源代码，如图 1.16 所示。

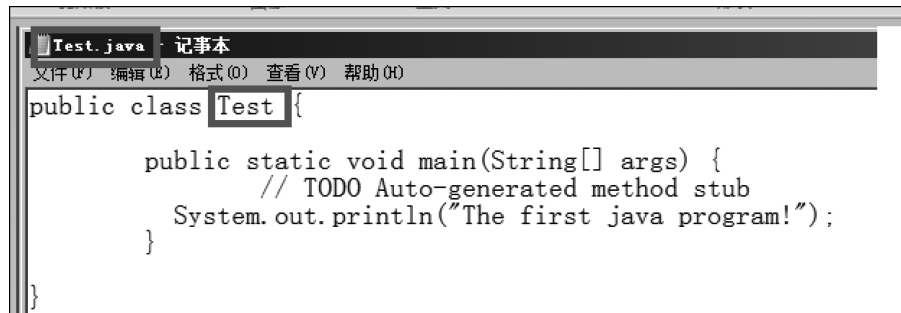


图 1.16 java 源程序代码

(3) 将文件保存。注意文件名必须与源代码中的类名一致。

1.3.2 编译和运行

编译和运行步骤如下。

- (1) 右击“开始”菜单，执行“运行”命令，输入“cmd”启动控制台。
- (2) 依次输入“d:”（进入 d 盘）和“cd studyjava”命令进入源程序文件所在目录，如图 1.17 所示。
- (3) 对源程序进行编译，输入“javac Test.java”命令，生成字节码文件（Test.class）。
- (4) 输入“java Test”命令，运行 Java 程序，得到运行结果，如图 1.17 所示。

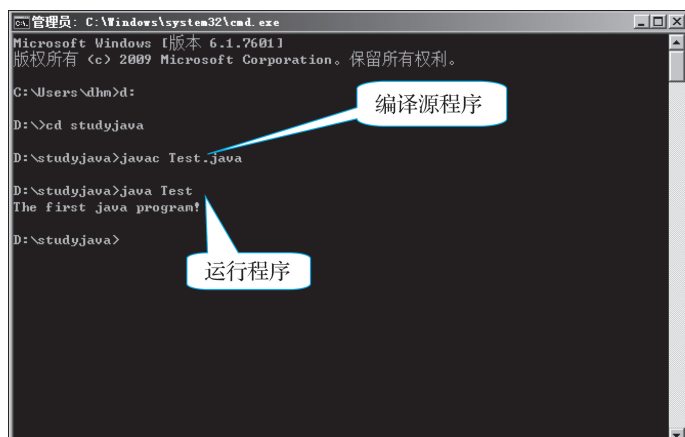


图 1.17 编译和运行

说明

(1) Java 应用程序由若干个类组成，有且仅有一个类含有主方法（main），这个类通常被 **public** 修饰，称为主类。Java 应用程序是从 **main** 方法开始执行的。

(2) **System.out.println()** 的作用是在显示器上输出双引号中的内容，其作用相当于 C 语言中的 **printf** 函数。

(3) **public** 和 **class** 是 Java 的关键字，**Test** 是用户自定义的类名，需要注意的是，源代码的文件名必须与主类名一致。

(4) **public static void main(String[] args)** 是 **main** 方法的固定定义格式。在 C 语言中 **main** 称为主函数，而在 Java 语言中 **main** 称为主方法。

(5) 为了提高程序的可读性，可以对 Java 程序进行注释。Java 的注释有三种类型：以“**//**”开始的单行注释；以“**/***”开始，以“***/**”结束的多行注释；以“**/****”开始，以“***/**”结束的文档注释。

练一练

编写一个 Java 程序，输出以下信息。

```
#####  
Hello Java  
#####
```

任务 1.4 使用 Eclipse 集成开发环境

【任务导入】

任务 1.3 中开发了第一个 Java 应用程序。在记事本中输入源代码，在控制台下输入编译命令“javac”和运行命令“java”对源程序进行编译和运行，这样操作非常麻烦，而且记事本的功能也非常有限。本任务介绍使用集成开发环境 Eclipse 开发 Java 程序。

思考

- (1) C 语言有 VC++ 6.0 这样的集成开发环境，那么 Java 语言是否也有相应的集成开发环境呢？
- (2) 如何使用 Eclipse？

1.4.1 下载和安装 Eclipse

IDE (Integrated Development Environment, 集成开发环境) 是用于提供程序开发环境的应用程序，包括代码编辑器、编译器、调试器和图形用户界面工具。它集成了代码编写、分析、编译和调试等功能。Java 语言的集成开发环境很多，比如 IBM 公司的 Eclipse、Borland 公司的 JBuilder 和 Sun 公司的 NetBeans 等，Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。本书采用 Eclipse 作为开发环境，安装步骤如下。

(1) 登录 Eclipse 的官方网站 (<https://www.eclipse.org>) 下载安装包。

(2) Eclipse 有很多个版本，不同的操作系统有不同的安装包，比如 Windows 版、Linux 版、Mac OS 版。本书选择“Eclipse IDE 2020-06”版本进行开发，单击“Download Packages”下载安装包（读者看到的可能是其他版本页面，此处不影响），如图 1.18 所示。

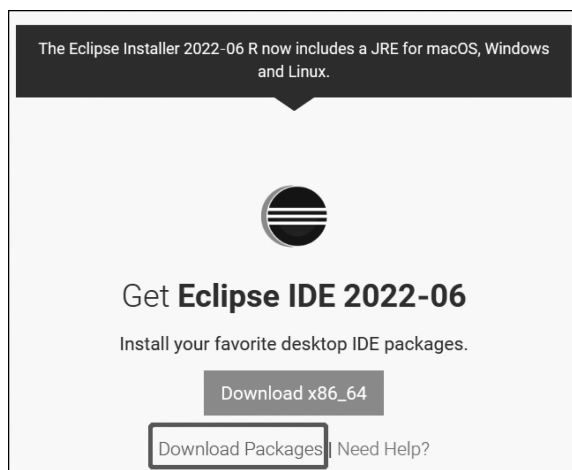


图 1.18 Eclipse 下载页面 (1)

(3) 接下来，在弹出界面中选择 2020-06 版本，如图 1.19 和图 1.20 所示。

(4) 在图 1.21 所示的界面中，单击“Windows x86_64”，下载 Eclipse。

(5) 单击链接下载“eclipse-java-2020-06-R-win32-x86_64.zip”压缩文件，下载完成后直接解压缩，无需安装，只要双击“eclipse.exe”就可以运行 Eclipse 了，如图 1.22 所示。

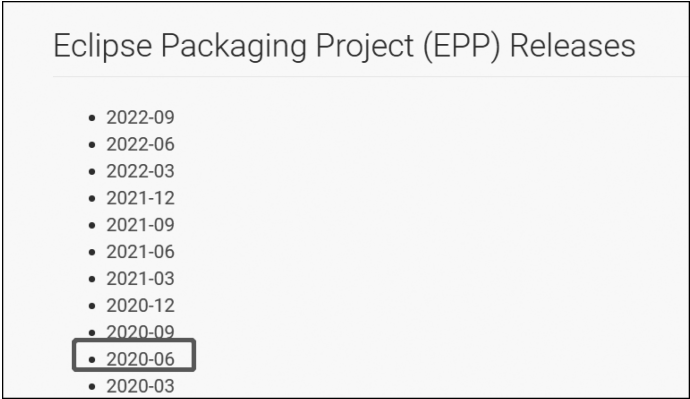


图 1.19 Eclipse 下载页面 (2)

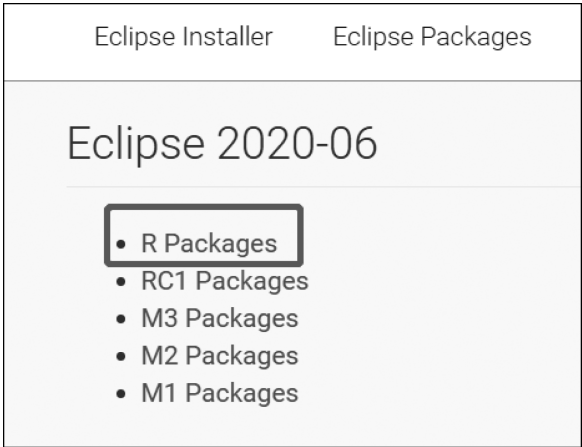


图 1.20 Eclipse 下载页面 (3)

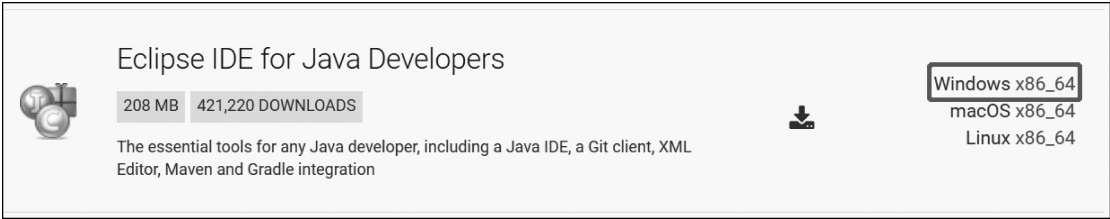


图 1.21 Eclipse 下载页面 (4)

名称	修改日期	类型
configuration	2020/9/10 11:17	文件夹
dropins	2020/9/10 11:17	文件夹
features	2020/9/10 11:17	文件夹
p2	2020/9/10 11:16	文件夹
plugins	2020/9/10 11:17	文件夹
readme	2020/9/10 11:17	文件夹
.eclipseproduct	2020/9/2 22:06	ECLIPSEPRODUCT ...
artifacts.xml	2020/9/10 11:17	XML 文档
eclipse.exe	2020/9/10 11:18	应用程序
eclipse.ini	2020/9/10 11:17	配置设置
eclipsesec.exe	2020/9/10 11:18	应用程序

图 1.22 运行 Eclipse

1.4.2 启动 Eclipse

启动 Eclipse，步骤如下。

(1) 双击“Eclipse IDE for Java Developers-2020-06”图标启动 Eclipse。

(2) Eclipse 会弹出“Eclipse IDE Launcher”对话框，如图 1.23 所示。工作空间（Workspace）的作用是保存 Eclipse 建立的程序项目和相应的设置。在“Workspace”输入框中输入路径“D:\javaproject”，也可以单击“Browse...”按钮修改路径。

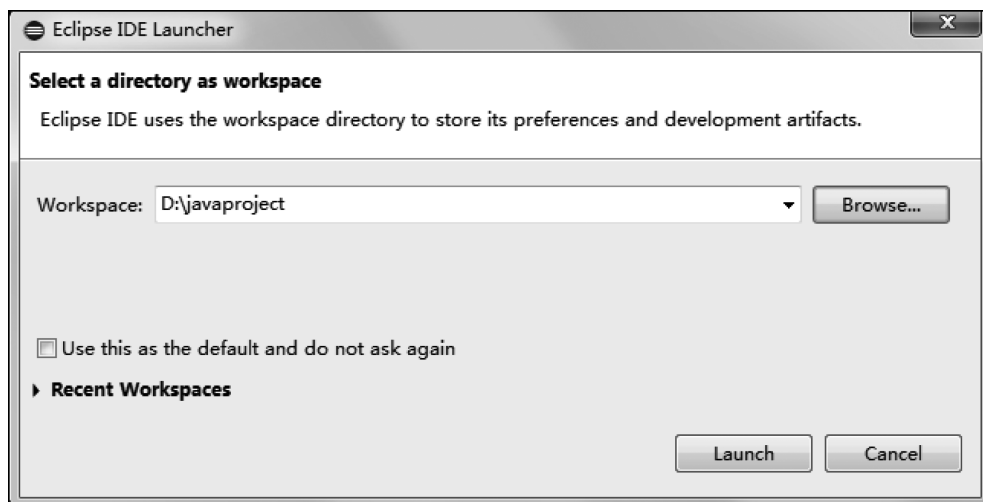


图 1.23 启动工作空间对话框

(3) 单击“Launch”按钮，进入 Eclipse 欢迎界面，如图 1.24 所示。



图 1.24 Eclipse 欢迎界面

(4) 单击“welcome”右侧的关闭按钮，进入 Eclipse 工作界面，如图 1.25 所示。

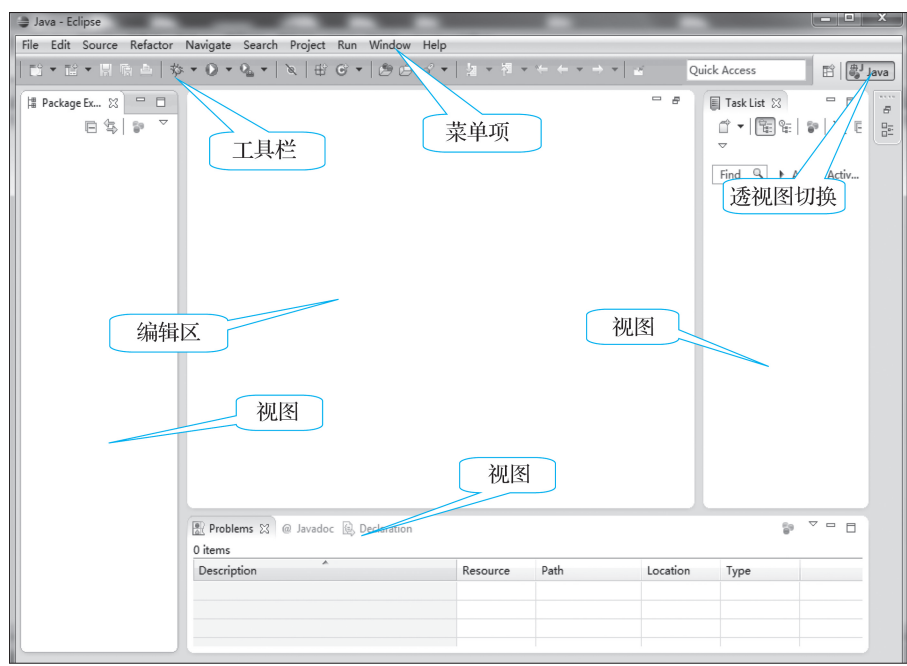


图 1.25 Eclipse 工作界面

1.4.3 开发 Java 项目

Eclipse 安装好之后，Java 就有了强大的开发工具。下面使用 Eclipse 来完成 Java 程序的编写和运行，步骤如下。

(1) 在 Eclipse 中编写程序，首先要建立项目，Eclipse 可以创建的项目有很多种，其中的 Java 项目用于编写和运行 Java 程序。单击“Create a Java project”，建立 Java 项目，如图 1.26 所示。

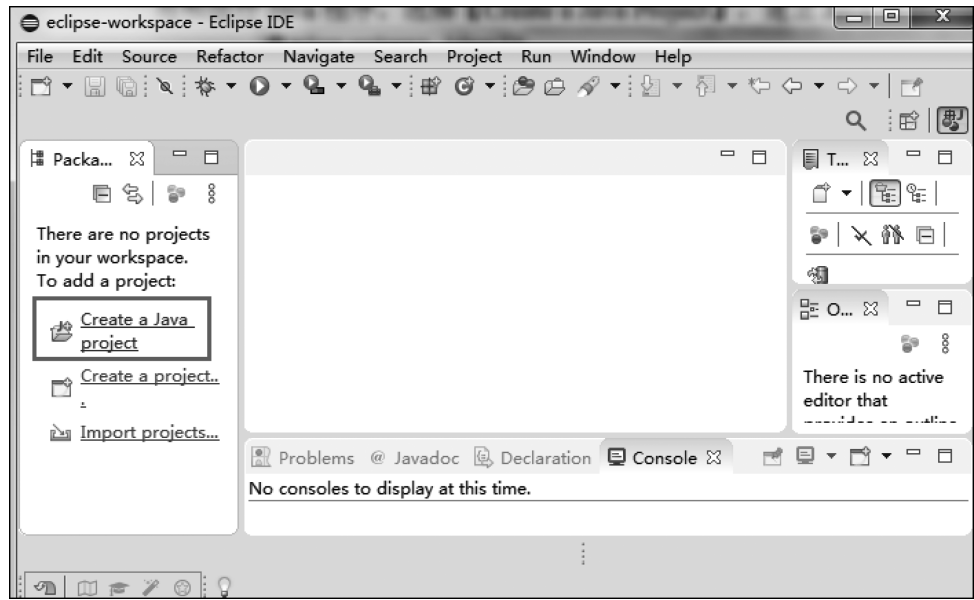


图 1.26 创建 Java 项目 (1)

(2) 在弹出的“New Java project”对话框的“Project name”栏中输入项目的名称。其他选项采用默认值，单击“Finish”按钮，即可完成创建 Java 项目，如图 1.27 所示。

(3) 依次执行“File”→“New”→“Class”，新建 Java 类。在弹出的“New Java Class”对话框的“Name”

栏输入类的名称“Demo”，勾选上“public static void main(String[] args)”自动生成 main()方法，其他选项采用默认值，如图 1.28 所示。

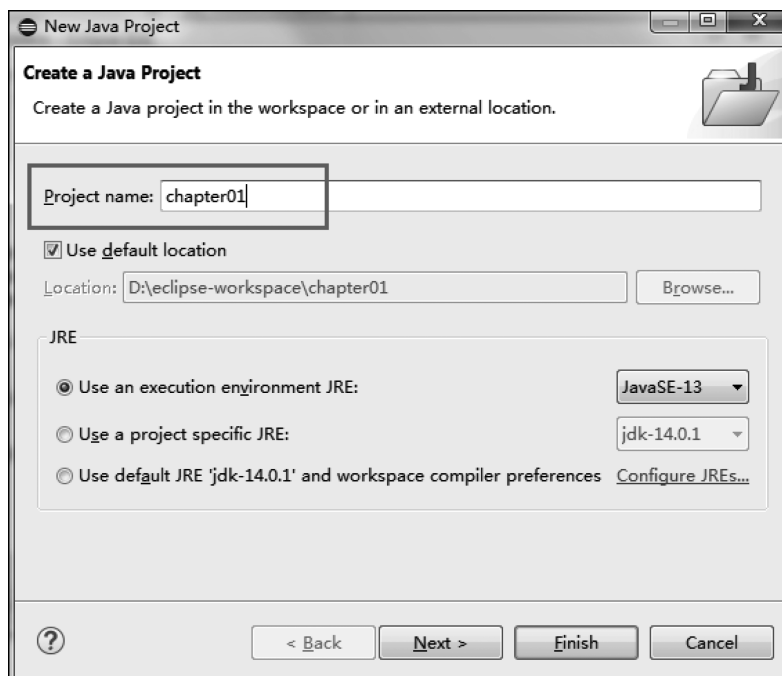


图 1.27 创建 Java 项目 (2)

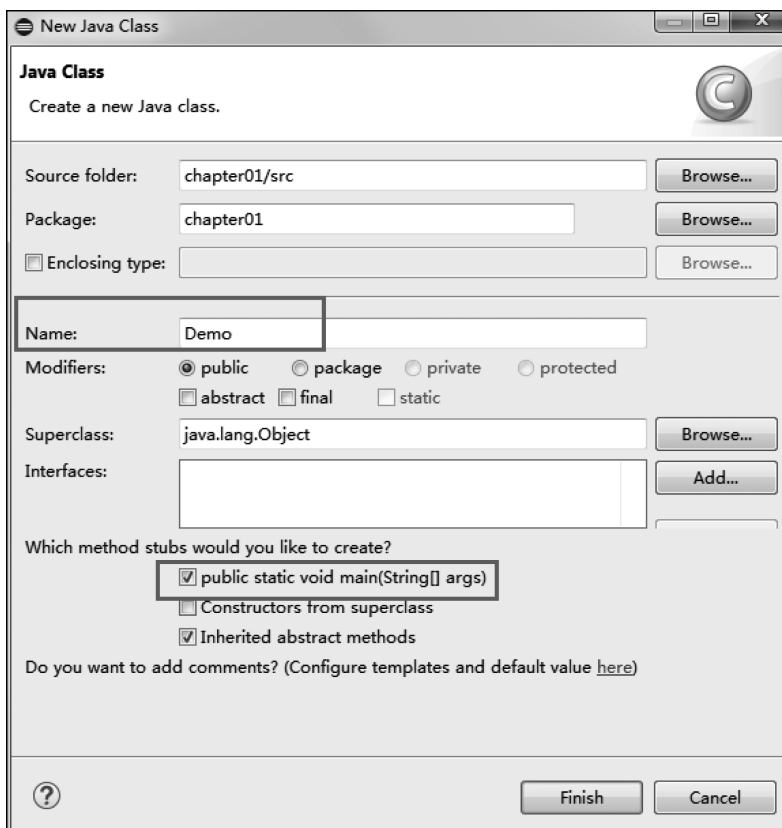


图 1.28 创建 Java 类

(4) 在编辑器中输入源程序代码，如图 1.29 所示。

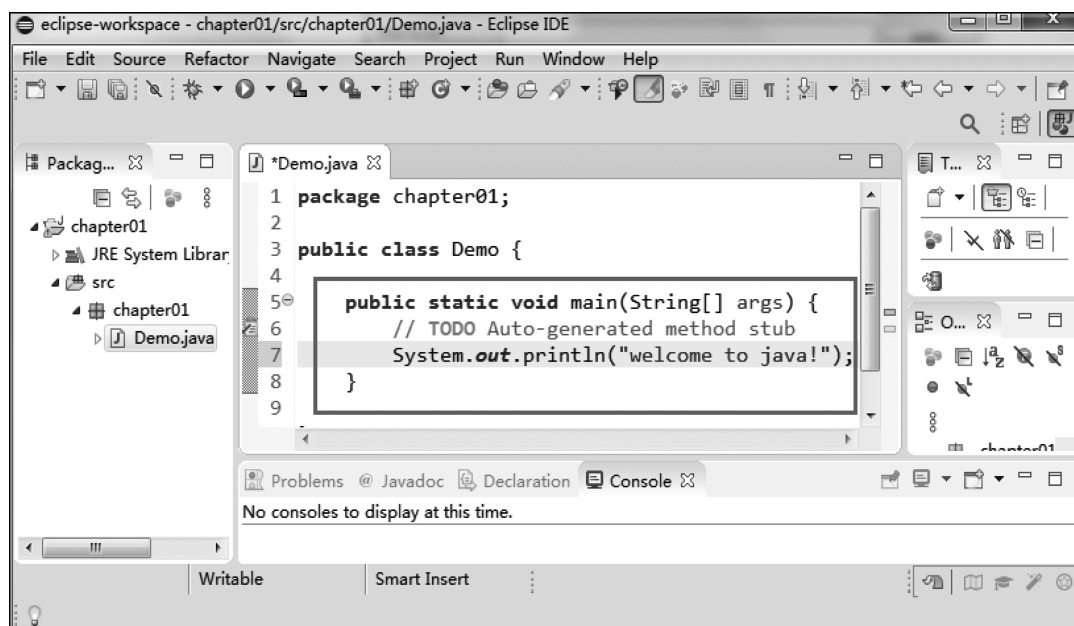


图 1.29 输入源程序代码

(5) 单击“”按钮运行 Java 程序，运行结果如图 1.30 所示。

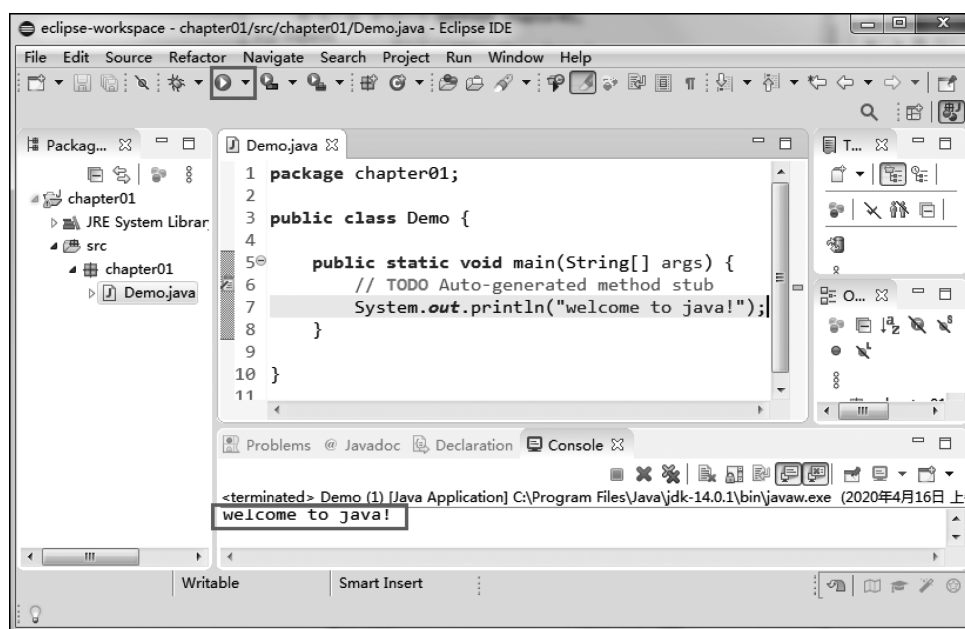


图 1.30 运行 Java 程序

练一练

在 Eclipse 中，开发一个输出以下信息的 Java 程序。

Welcome to java

任务 1.5 综合实训

【任务描述】

编写运行简单的 Java 程序。在 Eclipse 中编写源代码，运行并查看运行结果。要求运行结果如下所示。

```
*****  
*****  
*****  
*****
```

【任务目的】

- (1) 会安装 Java 运行环境。
- (2) 会使用 Eclipse 集成开发工具完成 Java 程序的编写、编译和运行。

【关键步骤】

- (1) 下载、安装和配置 JDK。
- (2) 下载、安装 Eclipse。
- (3) 编写源代码，调试并运行。

【实现代码】

```
public class Excecise {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("*****");  
        System.out.println("*****");  
        System.out.println("*****");  
        System.out.println("*****");  
    }  
}
```

课后测评

扫描二维码查看测评内容。



课后测评一