

魏巍文大 百年书香
www.jiaodapress.com.cn
bookinfo@sjtu.edu.cn



丛书策划 张荣昌
责任编辑 王清 孟海江
封面设计 唐韵设计

软件开发类系列教材

.NET开发综合实战

PHP开发综合实战

Web前端开发综合实战

Java开发综合实战

Java Web应用与开发

PHP应用与开发

Bootstrap应用与开发

Node.js应用与开发

Vue.js应用与开发

ASP.NET应用与开发

jQuery应用与开发

软件测试基础

项目管理

软件工程与测试

C语言程序设计

C#程序设计

R语言程序设计

Go语言程序设计

Android程序设计

Python程序设计

Java程序设计

JavaScript基础教程

PHP网络编程入门与进阶

JavaScript+jQuery入门与进阶

UI交互设计入门与进阶

HTML+CSS入门与进阶

HTML5+CSS3 Web前端设计案例教程

Java程序设计实用案例教程



扫描二维码
关注上海交通大学出版社
官方微信

ISBN 978-7-313-31872-5



9 787313 318725 >

定价: 62.00元

辽宁省“十四五”职业教育规划教材

Java开发综合实战(第2版)

主编 关星 冯颖 贾睿



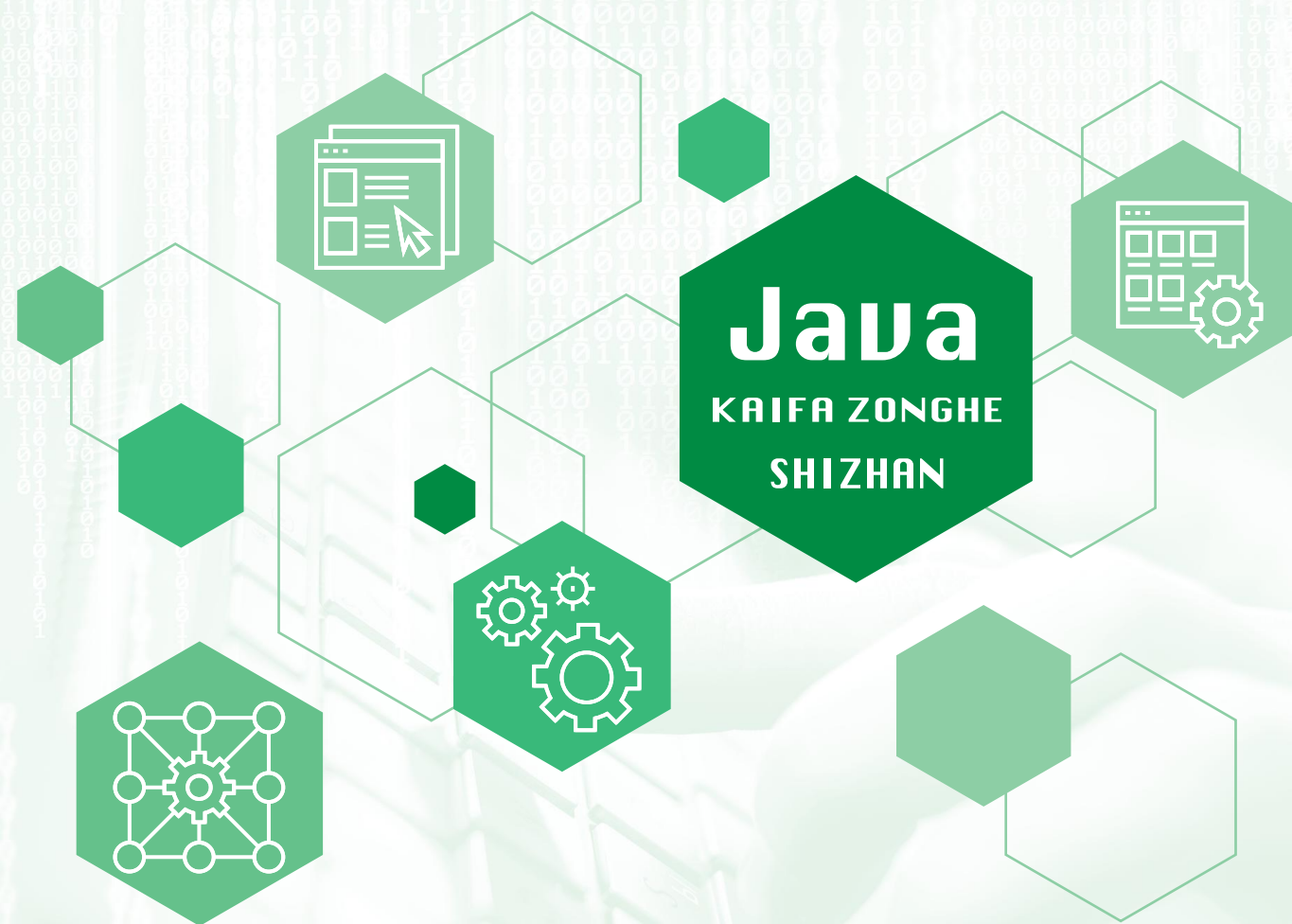
上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

辽宁省“十四五”职业教育规划教材

Java

开发综合实战 (第2版)

主编 关星 冯颖 贾睿



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书通过丰富的案例，分 17 个模块讲解了 Java 语言的基础知识和高级应用。具体内容包括 Java 概述和配置开发环境、初探 Java 程序、Java 基础语法、数组、方法、Java 面向对象程序设计基础、Java 面向对象程序设计高级应用、Java 字符串、Java 异常捕获和处理、Java 泛型、Java 常用类库、Java 集合类、Java I/O、Java 数据库编程、Java 多线程、Java 网络编程、Java 图形界面程序设计。本书既可作为高等院校计算机、云计算、信息安全、大数据、人工智能、软件工程等相关专业的教学用书，也可作为广大程序开发者的参考用书。

图书在版编目（CIP）数据

Java开发综合实战 / 关星，冯颖，贾睿主编.
2 版. -- 上海：上海交通大学出版社，2024. 12 -- ISBN
978-7-313-31872-5
I. TP312.8
中国国家版本馆 CIP 数据核字第 2024EC5443 号

Java 开发综合实战（第 2 版）
Java KAIFA ZONGHE SHIZHAN (DI 2 BAN)

主 编：关星 冯颖 贾睿	地 址：上海市番禺路 951 号
出版发行：上海交通大学出版社	电 话：021-6407 1208
邮政编码：200030	
印 制：北京荣玉印刷有限公司	经 销：全国新华书店
开 本：889 mm × 1194 mm 1/16	印 张：20.5
字 数：709 千字	
版 次：2021 年 9 月第 1 版	印 次：2024 年 12 月第 3 次印刷
2024 年 12 月第 2 版	
书 号：ISBN 978-7-313-31872-5	电子书号：ISBN 978-7-89564-016-0
定 价：62.00 元	

版权所有 侵权必究
告读者：如发现本书有印装质量问题请与印刷厂质量科联系
联系电话：010 - 6020 6144



前言

Java 是一门面向对象的编程语言，不仅吸收了 C++ 语言的各种优点，还摒弃了 C++ 语言里难以理解的多继承、指针等概念，因此 Java 语言具有功能强大和简单易用两个特征。Java 语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。Java 技术自 1995 年问世以来，在我国得到了迅速普及。无论是在移动设备、消费品中，还是在嵌入式设备中，Java 语言均得到了广泛的应用。未来，Java 语言将在嵌入式及移动设备中得到进一步的拓展，在各种电子信息产品中都将有 Java 的应用。特别是在移动设备领域，Java 语言的应用效果将更加显著，并将应用定位由系统的通信终端转变为个人信息端。Java 语言已成为目前最具吸引力且功能强大的程序设计语言之一。

本书的编写不仅旨在传授 Java 编程语言的知识，更深刻融入了课程思政的理念。教材不仅仅是技术的传授工具，更是塑造学生综合素养和价值观的平台。因此，本书在设计之初，坚持立德树人的根本任务，贯彻《高等学校课程思政建设指导纲要》和党的二十大精神，将专业知识与思政教育有机结合，实现价值引领、知识传授和能力培养紧密结合，着重于引导学生树立正确的人生观和价值观。通过学习 Java 编程，希望激发学生的专业精神、职业精神以及工匠精神，让他们在专业领域中追求卓越、善于解决问题和勇于创新。教材的编写理念体现在每一个模块和案例中，力求讲解当前最新的知识和主流技术，紧密联系实际应用场景，使学生能够学以致用，具备更强的适应能力。同时本书注重培养学生的思维方式和态度，期待学生在学习 Java 编程的过程中，能够培养解决问题的能力及勇于创新的精神，并将这种精神贯彻于未来的工作和生活中。

本书由长期从事 Java 教学和软件开发的一线教师编写而成，通过丰富的示例讲解了 Java 语言的基础知识和高级应用。全书共 17 个模块，包括 Java 概述和配置开发环境、初探 Java 程序、Java 基础语法、数组、方法、Java 面向对象程序设计基础、Java 面向对象程序设计高级应用、Java 字符串、Java 异常捕获和处理、Java 泛型、Java 常用类库、Java 集合类、Java I/O、Java 数据库编程、Java 多线程、Java 网络编程、Java 图形界面程序设计。

本书在编写上具有以下特点。

(1) 通过“目标导航”“知识导图”“模块导读”明确本模块要学习的内容，使学生做好学习的准备；通过文中的案例，讲授知识点的应用。

(2) 在语言简练的基础上，充分利用具体案例讲解知识点，结合“拓展训练”，帮助学生更好地理解所学内容，引导学生“做中学，学中做”，培养学生分析问题和解决问题的能力，为参加竞赛及后续工作打下一定的基础。

(3) 着重于当前最新知识和主流技术的讲解,保证学生所学知识和技术都与行业联系密切,使学生能够学以致用。

此外,编者还为广大一线教师提供了服务于本书的教学资源库,有需要者可发邮件至 2393867076@qq.com 领取。

本书既可作为高等院校计算机、云计算、信息安全、大数据、人工智能、软件工程等相关专业的教学用书,也可作为广大程序开发者的参考用书。

由于编写时间仓促,加之网络技术发展迅猛,书中存在的不足和疏漏之处,敬请广大读者批评指正,在此表示衷心的感谢。

编 者

2024 年 9 月



目录



模块 1 Java 概述和配置开发环境 / 1

1.1 Java 概述	2	1.2.1 开发环境介绍	5
1.1.1 什么是 Java	2	1.2.2 JDK 安装与配置	5
1.1.2 Java 特点	3	1.2.3 IDE 工具介绍	13
1.1.3 Java 运行机制	4	1.3 编写第一个 Java 程序	13
1.2 配置开发环境	5		



模块 2 初探 Java 程序 / 16

2.1 一个简单的 Java 程序	17	2.5 Java 变量	22
2.2 Java 程序注释	17	2.6 Java 常量	22
2.2.1 单行注释	17	2.6.1 常量值	22
2.2.2 多行注释	18	2.6.2 整型常量值	23
2.2.3 文档注释	18	2.6.3 实型常量值	23
2.3 Java 标识符	20	2.6.4 布尔型常量值	23
2.4 Java 关键字	20	2.6.5 字符型和字符串常量值	23



模块 3 Java 基础语法 / 25

3.1 基本数据类型	26	3.2.4 逻辑运算符	32
3.1.1 整型	26	3.2.5 三元运算符	32
3.1.2 浮点类型	27	3.2.6 位运算符	33
3.1.3 字符类型	28	3.2.7 运算符优先级	34
3.1.4 布尔类型	28	3.3 程序结构	34
3.1.5 数据类型转换	28	3.3.1 顺序结构	35
3.2 运算符	29	3.3.2 条件结构	35
3.2.1 赋值运算符	30	3.3.3 循环结构	38
3.2.2 算术运算符	30	3.3.4 跳转控制	40
3.2.3 比较运算符	31		



模块4 数 组 / 43

4.1 数组概述	44	4.4 多维数组	49
4.2 一维数组	44	4.5 数组的基本操作	50
4.2.1 创建一维数组	44	4.5.1 遍历数组	50
4.2.2 初始化一维数组	45	4.5.2 替换数组中的元素	51
4.3 二维数组	46	4.5.3 对数组中的元素进行排序	52
4.3.1 创建二维数组	46	4.5.4 经典示例	54
4.3.2 初始化二维数组	46		



模块5 方 法 / 56

5.1 方法的定义	57	5.3 方法的重载与重写	60
5.1.1 有返回值的方法	57	5.3.1 方法的重载	60
5.1.2 无返回值的方法	58	5.3.2 方法的重写	61
5.1.3 方法的修饰符	58	5.4 方法的递归调用	63
5.2 方法的参数和返回	58	5.4.1 数字的累加	63
5.2.1 参数的传递	58	5.4.2 斐波那契数列	64
5.2.2 方法的返回	59		



模块6 Java 面向对象程序设计基础 / 65

6.1 面向对象程序设计概述	66	6.2.8 对象的比较	77
6.1.1 什么是类	66	6.2.9 对象的销毁	78
6.1.2 什么是对象	66	6.3 this 关键字	79
6.1.3 类与对象的关系	67	6.3.1 使用 this 调用本类中的属性	79
6.1.4 面向对象的特点	67	6.3.2 使用 this 调用构造方法	79
6.2 类和对象	67	6.3.3 使用 this 表示当前对象	80
6.2.1 定义类	68	6.4 访问控制符	81
6.2.2 对象的创建和使用	68	6.5 封装	83
6.2.3 成员变量和局部变量	72	6.6 代码块	84
6.2.4 类的构造方法	74	6.6.1 普通代码块	84
6.2.5 类的主方法	75	6.6.2 构造块	84
6.2.6 成员方法	76	6.6.3 静态代码块	85
6.2.7 静态常量、变量和方法	76		



模块 7 Java 面向对象程序设计高级应用 / 87

7.1 继承	88	7.5 内部类	100
7.1.1 继承中的重写	90	7.5.1 实例内部类	100
7.1.2 继承中使用 super 关键字	91	7.5.2 静态内部类	102
7.2 抽象类	93	7.5.3 局部内部类	104
7.3 接口	94	7.6 Object 类	105
7.4 多态	96	7.6.1 toString() 方法	105
7.4.1 多态概述	97	7.6.2 equals() 方法	106
7.4.2 方法重写实现多态	97	7.6.3 getClass() 方法	107
7.4.3 方法重载实现多态	98	7.7 包装类	108
7.4.4 instanceof 关键字	99		



模块 8 Java 字符串 / 109

8.1 字符串概述	110	8.2.9 字符串截取	116
8.2 字符串基本操作	110	8.2.10 字符串拼接	116
8.2.1 创建 String 对象	111	8.2.11 字符串内容相等判断	117
8.2.2 获取字符串的长度	112	8.2.12 字符串替换	118
8.2.3 获取指定索引位置的字符	112	8.2.13 字符串与 char 数组	118
8.2.4 查找一个指定的字符串是否存在	113	8.2.14 字符串与 byte 数组	119
8.2.5 字符串大小写转换	113	8.3 空串与 null 串	120
8.2.6 字符串开头或结尾判断	114	8.4 StringBuffer 类	120
8.2.7 消除左右空格	115	8.5 StringBuilder 类	121
8.2.8 字符串切分	115		



模块 9 Java 异常捕获和处理 / 122

9.1 异常概述	123	9.3.3 使用 try...finally 处理异常	130
9.2 异常分类	124	9.3.4 finally 和 return 执行顺序	130
9.2.1 可控式异常	124	9.4 抛出异常	132
9.2.2 运行时异常	125	9.4.1 throws 抛出异常	132
9.3 处理异常	126	9.4.2 throw 关键字	134
9.3.1 使用 try...catch 处理异常	126	9.5 自定义异常	135
9.3.2 使用 try...catch...finally 处理异常	128		



模块 10 Java 泛型 / 136

10.1 泛型概述	137	10.3 泛型方法	140
10.2 泛型类	137	10.4 泛型变量的限定	141



模块 11 Java 常用类库 / 142

11.1 Math 类	143	11.5 正则表达式	152
11.2 随机数类	144	11.5.1 正则表达式概述	153
11.3 日期操作类	145	11.5.2 正则表达式使用	156
11.3.1 Date 类	145	11.6 System 类	157
11.3.2 Calendar 类	147	11.6.1 System 类概述	157
11.3.3 DateFormat 类	149	11.6.2 System 类常用方法	158
11.3.4 SimpleDateFormat 类	151	11.7 定时调度类	159
11.4 Arrays 类	152		



模块 12 Java 集合类 / 161

12.1 Java 集合概述	162	12.4.4 SortedSet 接口	170
12.2 Collection 接口	164	12.5 Map 接口和 SortedMap 接口	172
12.3 List 接口	165	12.5.1 Map 接口概述	172
12.3.1 List 接口概述	165	12.5.2 Map 接口基本操作	173
12.3.2 List 接口基本操作	165	12.5.3 SortedMap 接口	174
12.4 Set 接口和 SortedSet 接口	168	12.6 输出集合	175
12.4.1 Set 接口概述	168	12.6.1 Iterator 输出集合	175
12.4.2 Set 接口定义	168	12.6.2 ListIterator 输出集合	177
12.4.3 Set 接口基本操作	169		



模块 13 Java I/O / 179

13.1 流概述	180	13.4.3 文件输入流	189
13.2 输入 / 输出流	181	13.4.4 文件输出流	190
13.2.1 输入流	181	13.5 字符流	192
13.2.2 输出流	182	13.5.1 字符文件输入流	193
13.3 File 类	182	13.5.2 字符文件输出流	195
13.3.1 File 类概述	182	13.5.3 字符缓冲区输入流	196
13.3.2 File 类基本操作	182	13.5.4 字符缓冲区输出流	197
13.4 字节流	187	13.6 数据流	198
13.4.1 InputStream 类	187	13.7 压缩流	199
13.4.2 OutputStream 类	189	13.7.1 压缩文件	200

13.7.2 解压文件	201	13.9.2 Buffer	205
13.8 编码	203	13.9.3 Selector	205
13.9 Java 新 I/O	204		
13.9.1 Channel	204		



模块 14 Java 数据库编程 / 207

14.1 JDBC 概述	208	14.3.6 ResultSet 接口	219
14.1.1 JDBC 简介	208	14.4 JDBC 操作数据库	220
14.1.2 JDBC 驱动程序类型	208	14.4.1 建立和关闭连接	220
14.1.3 JDBC 典型用法	209	14.4.2 插入数据	223
14.2 结构化查询语言	209	14.4.3 查询数据	224
14.2.1 MySQL 下载和安装	210	14.4.4 更新数据	226
14.2.2 SQL 基础语法	214	14.4.5 删除数据	227
14.3 JDBC 常用类和接口	216	14.4.6 批处理	229
14.3.1 Driver 接口	216	14.4.7 调用存储过程	231
14.3.2 DriverManager 类	216	14.5 事务处理	232
14.3.3 Connection 接口	217	14.5.1 事务概述	232
14.3.4 Statement 接口	218	14.5.2 JDBC 的事务支持	233
14.3.5 PreparedStatement 接口	218		



模块 15 Java 多线程 / 236

15.1 线程概述	237	15.4.2 判断线程是否已启动	243
15.1.1 进程与线程	237	15.4.3 线程睡眠	244
15.1.2 多线程	237	15.4.4 线程的加入	244
15.2 Java 实现线程	238	15.4.5 线程的中断	245
15.2.1 继承 Thread 类实现线程	238	15.4.6 线程的后台运行	246
15.2.2 Runnable 接口实现线程	239	15.4.7 线程的让步	247
15.2.3 Thread 类和 Runnable 接口的比较	240	15.5 线程的优先级	248
15.3 线程的生命周期	241	15.6 线程同步	249
15.3.1 创建和就绪状态	241	15.6.1 线程安全	249
15.3.2 运行和阻塞状态	241	15.6.2 同步代码块	250
15.3.3 终止状态	242	15.6.3 同步方法	251
15.4 线程的操作方法	242	15.6.4 死锁	252
15.4.1 设置和获取线程名称	242	15.7 线程通信	253



模块 16 Java 网络编程 / 256

16.1 网络协议	257	16.4 TCP 程序设计	260
16.1.1 TCP/IP 协议	257	16.5 UDP 程序设计	263
16.1.2 TCP 协议与 UDP 协议	257	16.6 访问网络资源	265
16.2 Internet 地址	258	16.6.1 URL	265
16.2.1 IP 地址概述	258	16.6.2 URLConnection	266
16.2.2 InetAddress 类	258	16.6.3 编码	267
16.3 套接字和端口	259	16.6.4 解码	267



模块 17 Java 图形界面程序设计 / 269

17.1 Java 图形界面概述	270	17.7.3 密码域	291
17.2 Swing 概述	270	17.8 常用布局管理器	292
17.3 Swing 常用窗体	271	17.8.1 绝对定位布局	292
17.3.1 JFrame 框架窗体	271	17.8.2 边界布局管理器	294
17.3.2 JDialog 窗体	273	17.8.3 网格布局管理器	295
17.4 标签组件	273	17.8.4 流布局管理器	296
17.4.1 更改 JLabel 的文字样式	275	17.8.5 卡片布局管理器	297
17.4.2 在 JLabel 中设置图片	276	17.9 常用面板	298
17.5 按钮组件	278	17.9.1 JPanel 面板	298
17.5.1 提交按钮	278	17.9.2 JScrollPane 面板	300
17.5.2 不弹起的按钮	279	17.9.3 JSplitPane 面板	301
17.5.3 单选按钮	281	17.10 事件处理	302
17.5.4 复选框	282	17.10.1 鼠标事件	303
17.6 列表组件	283	17.10.2 鼠标拖曳事件	306
17.6.1 下拉列表框	283	17.10.3 键盘事件	307
17.6.2 列表框	286	17.10.4 窗体事件	308
17.7 文本组件	289	17.10.5 动作事件	310
17.7.1 文本域	289	17.11 菜单组件	312
17.7.2 文本区	290		

参考文献	315
------------	-----

模块 1

Java 概述和配置开发环境

目标导航 >

知识目标

- ① 了解 Java 的基本概念和特点，熟悉其基础理论。
- ② 熟悉 Java 的运行机制，掌握其基本工作原理。
- ③ 熟练配置 Java 程序的开发环境，了解常见开发工具的使用方法。
- ④ 掌握自主编写第一个 Java 程序所需的基本技能和语法知识。

能力目标

- ① 能够不断深入了解和应用编程知识，具备探究学习的能力。
- ② 能清晰准确地表达编程思路和问题解决方案，具有良好的语言表达和沟通能力。
- ③ 能够在团队合作中发挥作用，协作解决问题，分享并汇总编程经验。
- ④ 能够使用相关工具开发和维护基本的 Java 程序，具备基本的信息技术应用和维护能力。

素质目标

- ① 培养对计算机科学和编程的逻辑思维，认识科技对社会发展的影响。
- ② 明确专业素质要求，提升科学文化素质，了解创新对于编程的重要性。

知识导图 >



 模块导读

Java 是一种可以编写跨平台应用程序的、面向对象的程序设计语言。Java 经过一系列的演变过程,不断成熟,成为广泛使用的程序开发语言。当探索 Java 的演变历程和特性时,我们也应该思考其更深层次的意义。Java 不仅是一门编程语言,更是一种承载着社会责任的技术。我们在学习技术的同时,需要明晰技术发展对社会的影响,担当起推动技术进步的使命。Java 的跨平台特性也象征着合作与共享的精神,这启发我们在团队合作中追求共同目标,相互支持与协作。在探索 Java 技术的过程中,牢记技术与社会责任的紧密联系,努力成为技术发展的推动者和引领者。

学习 Java 语言程序设计时,必须选择一个功能强大、使用简单、能够辅助程序设计的 IDE(集成开发环境)。因此,本模块首先介绍 Java 的自身发展历程和特点,让读者对 Java 有一些基本的认识,知道 Java 的主要作用;然后介绍 Java 的开发环境,并讲解开发环境的配置;最后让读者动手去编写第一个 Java 程序。

1.1 Java 概述

Java 语言是由 Sun 公司在 1991 年开发的,是当时成立的研究小组 Green 致力于为未来的智能设备开发的一种新的编程语言。Java 语言得以开发的原因是 Green 小组中的詹姆斯·高斯林(James Gosling)对 C++ 语言在执行过程中的表现非常不满,于是他就把自己封闭在办公室里编写出了一种更高效的编程语言,并将其命名为“Oak”(即 Java 的前身)。后来在注册 Oak 商标时, Sun 公司发现它已经被另一家公司注册,所以改名为 Java。此时的 Java 已经具备安全性、网络通信、面向对象、多线程等特性。

现如今,随着各种新型信息技术的发展,Java 开始与这些新技术积极融合,融入 Internet 应用开发的诸多方面,成为当今互联网(Internet)应用开发的主流语言,是支撑 Internet 计算的庞大技术体系。

1.1.1 什么是 Java

Java 语言发展到今天经历了一系列的过程。Java 在 1995 年 5 月 23 日推出了 JDK1.0 版本,该版本标志着 Java 正式进军 Internet。1998 年,Java 对之前的 JDK 进行了升级并推出了 JDK1.2 版本,该版本加入了大量的轻量级组件包,从此之后 Java 被正式命名为“Java 2”。Java 是一个庞大的技术体系,主要由 Java SE、Java EE、Java ME 这 3 个分支组成。

1. Java SE

Java 平台标准版(Java platform standard edition, Java SE)不仅为 Java 桌面和工作组级应用的开发与运行提供了环境,还提供了部署与运行 Java Applet(一种网页小程序)和 Application(应用程序)的编译器、开发工具、运行环境与 Java API(应用程序接口)。它是 Java EE 和 Java Web Services 技术的基础,主要由 Java SE Development Kit(Java 开发工具包, JDK)和 Java SE Runtime Environment(Java 运行时环境, JRE)共同实现。

Java 开发工具(development)包括 Java 语言的编译器、调试器及文档工具等。

Java 部署技术(deployment)提供了部署与运行 Java 应用的支持,包括以 Web 方式加载与运行 Java Application 的 Java Web Start、支持 Applet 在网景领航员(netscape navigator)和 Microsoft Internet Explorer 浏览器中运行的 Java Plug-in。

用户界面工具集(user interface toolkits)包括图形化用户界面工具 AWT 和 Swing;二维图形和图像的显示与操作工具 Java 2D;声音捕获、处理和播放的 Java Sound API;支持多种语言输入的输入法框架(input method framework);开发面向残疾人的 Java 应用的相关技术 Accessibility,如屏幕阅读器、语言识别系统和盲文显示系统等。

集成 API 包括分布式对象操作系统的相关内容,如 Java 远程方法调用(remote method invocation, RMI)技

术与 CORBA（公共对象请求代理体系结构）技术、数据库连接 API——JDBC、向 Java 应用提供命名和目录服务的 API——JNDI 等。

Java 应用程序接口（Java API）是一个很大的 Java 类库，提供了丰富的功能，既能使应用系统访问底层平台服务，又能保证 Java 应用系统不依赖具体的底层平台，在支持和简化应用系统开发的同时，使应用程序具有可移植性。

Java 虚拟机（Java virtual machine, JVM）是 Java 平台的基础，它提供了 Java 程序运行的必要环境。因为 Java 的目标代码是字节码，不是位码，不直接针对某个具体的平台，所以在执行之前，需要将字节码转换为本机代码。为了实现语言的动态性和安全性，Java 编译器将符合引用的信息保留在字节码中，这样类的装载和符号引用的消除都能够在运行时进行。因此，在 Java 平台中专门引入 JVM，以支持字节码的运行。

2. Java EE

Java 平台企业版（Java platform enterprise edition, Java EE）基于 Java SE，包括 Enterprise JavaBeans（EJB）、Java Servlet API 及 Java Server Pages（JSP）等技术，定义了基于组件的多层企业级应用的开发标准，用于面向企业级和高端服务器的 Internet 应用开发，并为企业级应用的开发提供了各种服务和工具。Java EE 的应用程序模型是多层模型，包括客户端表示层、服务器端表示层、应用逻辑层、企业信息系统层。

3. Java ME

Java 平台微缩版（Java platform micro edition, Java ME）是针对消费类电子设备（如移动电话、电视机顶盒、汽车导航等）的嵌入式计算的一组技术和规范，目前已被广泛应用。它在 Java SE 的基础上，结合这类设备计算资源的限制，对 Java SE 进行语言精简，并对运行环境进行了高度优化。

虽然 Java 语言的发展方向有 3 个，但是它们最核心的部分是 Java SE，Java EE 和 Java ME 是在 Java SE 基础上发展起来的。

1.1.2 Java 特点

在詹姆斯·高斯林（James Gosling）和亨利·麦吉尔顿（Henry McGilton）等撰写的 Sun 公司 Java 白皮书中指出，Java 是一种“简单（simple）、面向对象（object oriented）、分布式（distributed）、解释型（interpreted）、强壮（robust）、安全（secure）、体系结构中立（architecture neutral）、可移植（portable）、高性能（high performance）、多线程（multithreaded）和动态（dynamic）”的编程语言，对这些特征的理解，是领会 Java 语言精髓的关键。这里对前 3 个特征进行简要说明，其他特征将在后续相关内容中进行介绍。

1. 简单

Java 语言句法和语义都比较简单，容易学习和使用。另外，Java 对 C++ 中容易引起错误的成分进行了相当成功的改造，例如去掉指针、取消多重继承和运算符重载，内存管理由程序员移向 Java 内嵌的自动内存回收机制，等等。Java 还提供大量功能丰富的可重用类库，简化语义，减少出错机会，从而简化了编程工作量。

2. 面向对象

作为一种面向对象的编程语言，Java 不仅最为“纯洁”，而且它对面向对象的支持也最为全面。与 C++ 一样，Java 的对象有模块化性质和信息隐藏能力，满足面向对象的封装性要求。Java 支持面向对象的继承性，并且 Java 通过抽象类和接口支持面向对象的多态性要求，即一个对外接口，多种内部实现。

3. 分布式

Java 具有支持分布式计算的特征。分布式计算中的“分布”具有两层含义：一是数据分布，即应用系统所操作的数据可以分散存储在不同的网络节点上；二是操作分布，即应用系统的计算可由不同的网络节点完成。

Java 实现了以下两种层次上的分布。

(1) 数据分布支持：通过 Java 的 URL 类，Java 程序可以访问网络上的各类信息资源，访问方式完全类似于本地文件系统。

(2) 操作分布支持：Java 通过嵌在万维网（WWW）页面中的 Applet（小应用程序）将计算从服务器分布至客户机。Applet 由 WWW 浏览器在客户端执行，从而避免了网络拥挤，提高了系统效率。

1.1.3 Java 运行机制

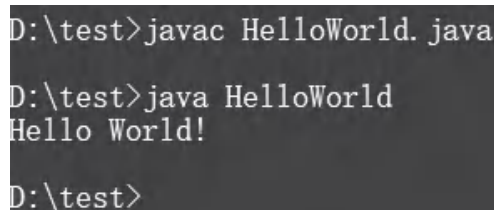
Java 程序的运行不同于大多数程序设计语言。Java 程序从源码到可运行，需要经过编译、解释两个过程。此处用一个案例做具体解释。

【例 1.1】 输出 “Hello World!” 示例。代码如下：

```
public class HelloWorld{  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

本程序的功能是输出 “Hello World!”，这里的代码仅作为说明 Java 运行机制的参考，暂时不需要懂代码的具体含义。

将上述代码录入到文本文档中并将文件命名为 “HelloWorld.java”，存放在 “D:\test” 目录下，在 Windows 的命令行窗口进行编译运行，操作如图 1-1 所示。



```
D:\test>javac HelloWorld.java  
  
D:\test>java HelloWorld  
Hello World!  
  
D:\test>
```

图 1-1 Java 命令行窗口编译运行

如图 1-2 所示，HelloWorld.java 文件由编译器进行编译，生成 HelloWorld.class 文件，该文件称为 Java 字节码（Java bytecodes）文件，经解释器解释后，运行成功，输出 “Hello World!”。

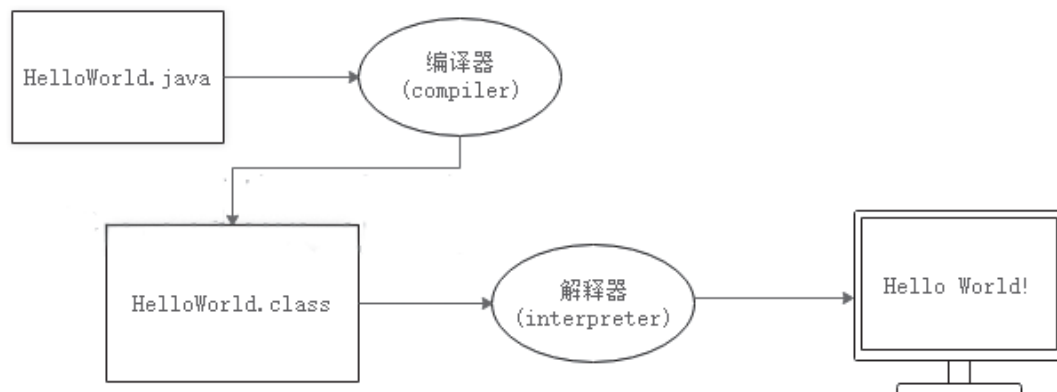


图 1-2 Java 的运行过程

在 Java 中有两种应用程序，分别对应两种不同类型的运行系统。Java 应用的运行系统是 Java 解释器，而

Java Applet 的运行系统是兼容 Java 的 Web 浏览器，该浏览器中包括了支持 Applet 运行的环境。

运行系统一般包括类装配器、字节码验证器、解释器、代码生成器和运行支持库，里面运行的是字节码文件。执行字节码的过程分为以下 3 步。

1. 代码的装入

由类装配器装入程序运行时需要的所有代码，其中包括程序代码中调用到的所有类。当装入了运行程序需要的所有类后，运行系统便可以确定整个可执行程序内存的布局。

2. 代码的验证

由字节码检验器进行安全检查，以确保代码不违背 Java 的安全性规则，同时字节码检验器还可以发现操作数栈溢出、非法数据类型转化等多种错误。

3. 代码的运行

Java 字节码的运行有以下 2 种方式。

(1) 即时编译：由代码生成器先将字节码编译为本机代码，然后再执行全部本机代码，这种运行方式效率高。

(2) 解释执行方式：解释器每次把一小段代码转换成本机代码并执行，如此往复完成所有操作。

1.2 配置开发环境

开发 Java 程序必须安装 JDK 开发环境，这个是 Java 程序运行的基础环境。一般为了方便开发都会安装一个集成开发环境。

1.2.1 开发环境介绍

学习 Java 语言，最好使用 Sun 公司（后被 Oracle 收购）的 JDK。JDK 采用命令行方式进行程序的编译和运行，在使用时，需要有编译软件与 JDK 配合使用，主要有下面两种方式。

(1) 用普通文本编辑器作为 Java 程序的编译器。

(2) 用能够与 JDK 配合使用的、具有简单开发与调试环境的编译器。

1.2.2 JDK 安装与配置

在 Oracle 官网 <https://www.oracle.com> 下载 JDK。访问 Oracle 官网，单击“Products”，选择“Java”选项，进入 JDK 下载界面。本书选择下载 JDK 8（8u251）的 Windows 版本。在 Oracle 官网下载 JDK 的流程如图 1-3 至图 1-7 所示。

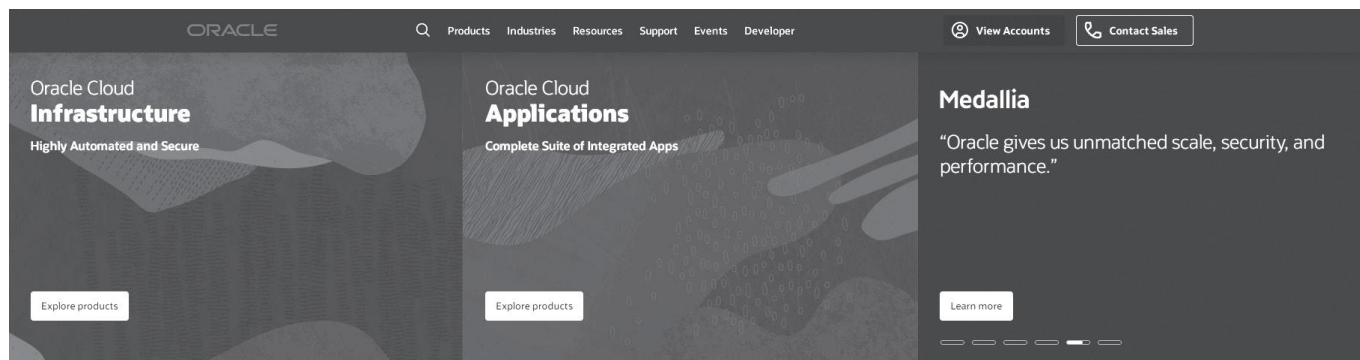


图 1-3 Oracle 官网

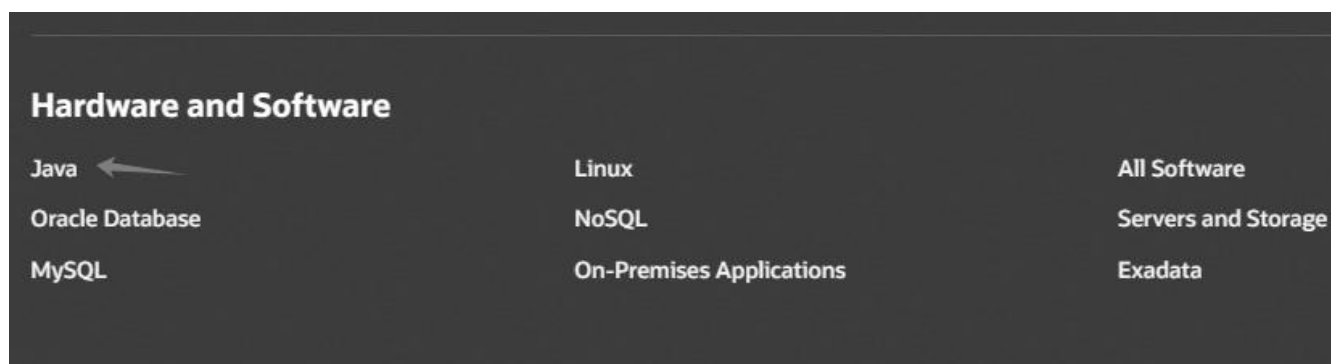


图 1-4 选择“Java”选项

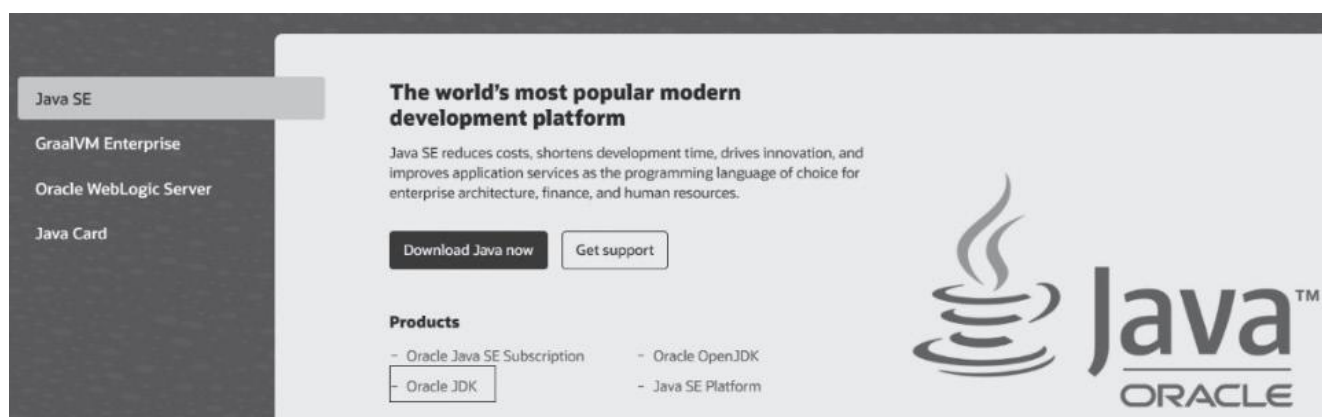


图 1-5 选择“Oracle JDK”选项

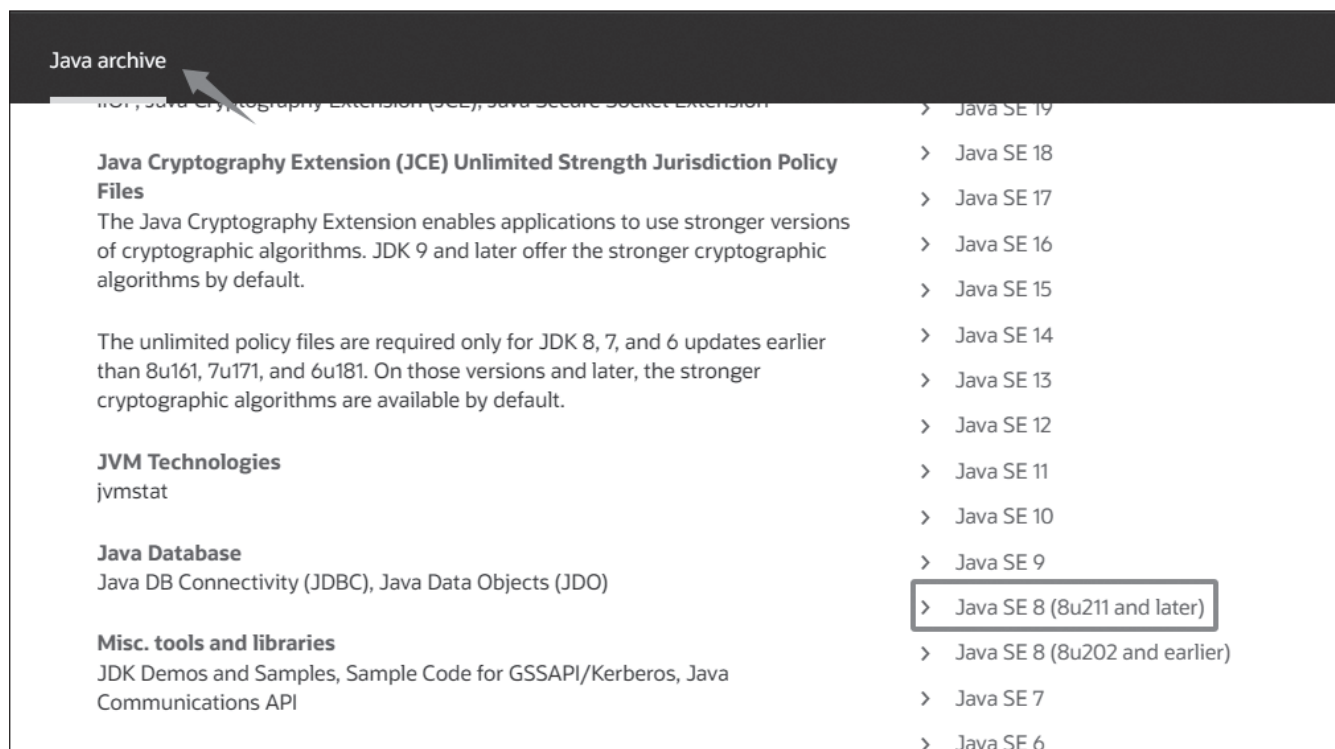


图 1-6 选择之前的版本

macOS x64	254.78 MB	 jdk-8u251-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.19 MB	 jdk-8u251-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	88.16 MB	 jdk-8u251-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.64 MB	 jdk-8u251-solaris-x64.tar.Z
Solaris x64	91.9 MB	 jdk-8u251-solaris-x64.tar.gz
Windows x86	201.17 MB	 jdk-8u251-windows-i586.exe
Windows x64	211.54 MB	 jdk-8u251-windows-x64.exe

图 1-7 下载对应版本

JDK 下载完成后，双击下载的 .exe 文件进行安装（见图 1-8）。JDK 的安装界面如图 1-9 所示。

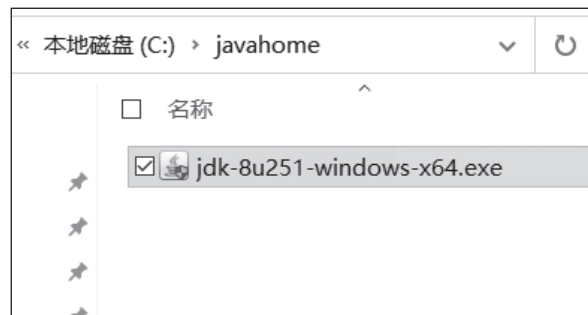


图 1-8 双击 JDK 安装文件



图 1-9 JDK 安装界面

选择 JDK 的安装路径，单击“下一步”按钮开始安装，如图 1-10、图 1-11 所示。



图 1-10 选择 JDK 的安装路径



图 1-11 JDK 正在安装

接下来安装 JRE，单击“更改”按钮，选择 JRE 的安装目录，单击“下一步”按钮开始安装，如图 1-12、图 1-13 所示。



图 1-12 选择 JRE 的安装路径

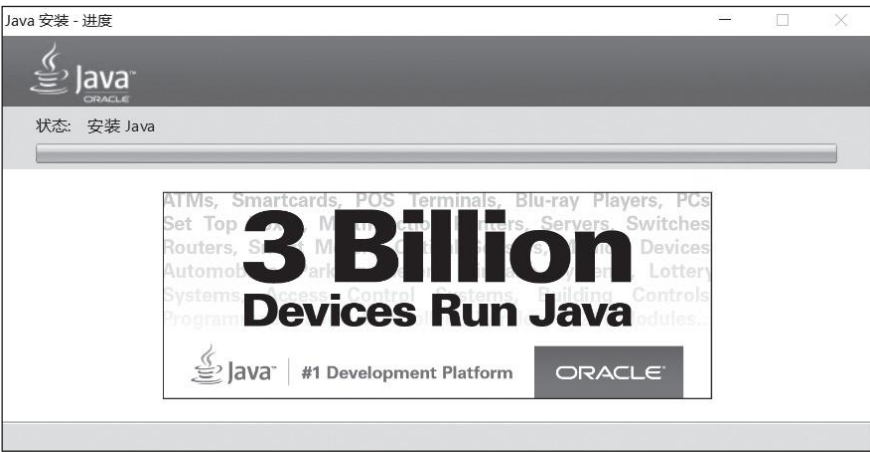


图 1-13 JRE 正在安装

当弹出如图 1-14 所示的界面时,说明 JDK 和 JRE 已经安装完成。



图 1-14 JDK、JRE 安装完成

下一步,还需要把 Java 的环境变量配置到 Windows 操作系统中。这里需要配置 JAVA_HOME 环境变量,前期 JAVA_HOME 是用不到的,主要是在后期编写 Java Web 应用程序时,需要用到 Tomcat 服务器。Tomcat 依赖 JDK,会找 JAVA_HOME 的环境变量,从而应用 JAVA_HOME 所指向的 JDK。环境变量配置过程如下。

在计算机的控制面板,单击“系统和安全”(见图 1-15),选择“系统”(见图 1-16),配置高级系统环境变量,操作界面如图 1-17、图 1-18 所示。



图 1-15 控制面板—系统和安全



图 1-16 系统和安全—系统



图 1-17 系统—高级系统设置



图 1-18 高级系统设置—环境变量

环境变量有两种，一种是管理员环境变量，另一种是系统环境变量。这里为了让登录此计算机的用户都能使用，选择配置系统变量。单击“系统变量”下的“新建”按钮（见图 1-19），进行 JAVA_HOME 的配置。

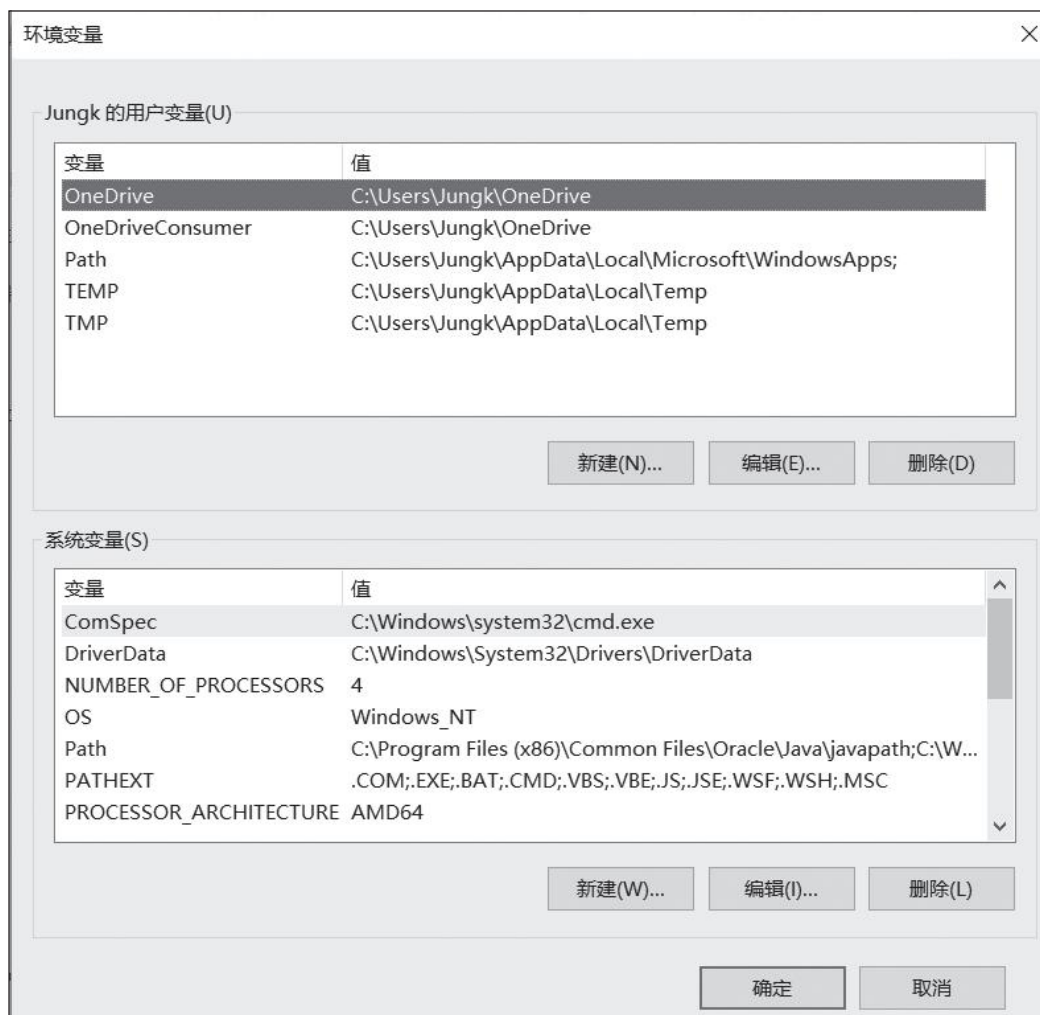


图 1-19 环境变量

进入配置界面，填入变量名，变量值就是本机 JDK 的安装目录，单击“确定”按钮即可，如图 1-20 所示。

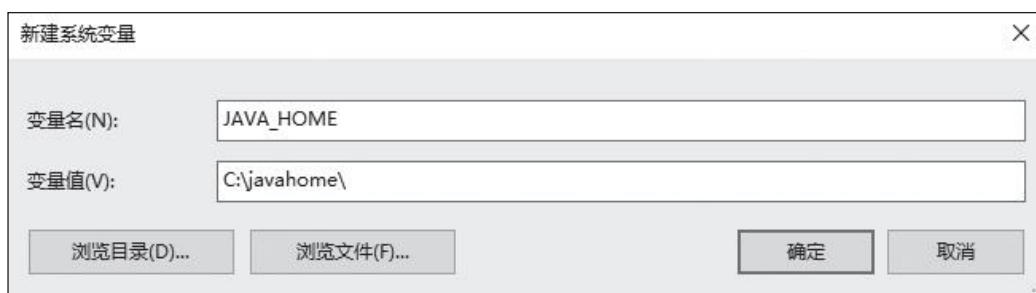


图 1-20 配置 JAVA_HOME

下一步需要配置系统环境变量 Path 的值。Path 环境变量的配置是为了在 Windows 命令行窗口下，可以直接使用某些环境。这里具体演示将 Java 配置到 Path 环境变量中去的方法。

如图 1-21 所示，在“系统变量”中选择 Path，单击“编辑”按钮。

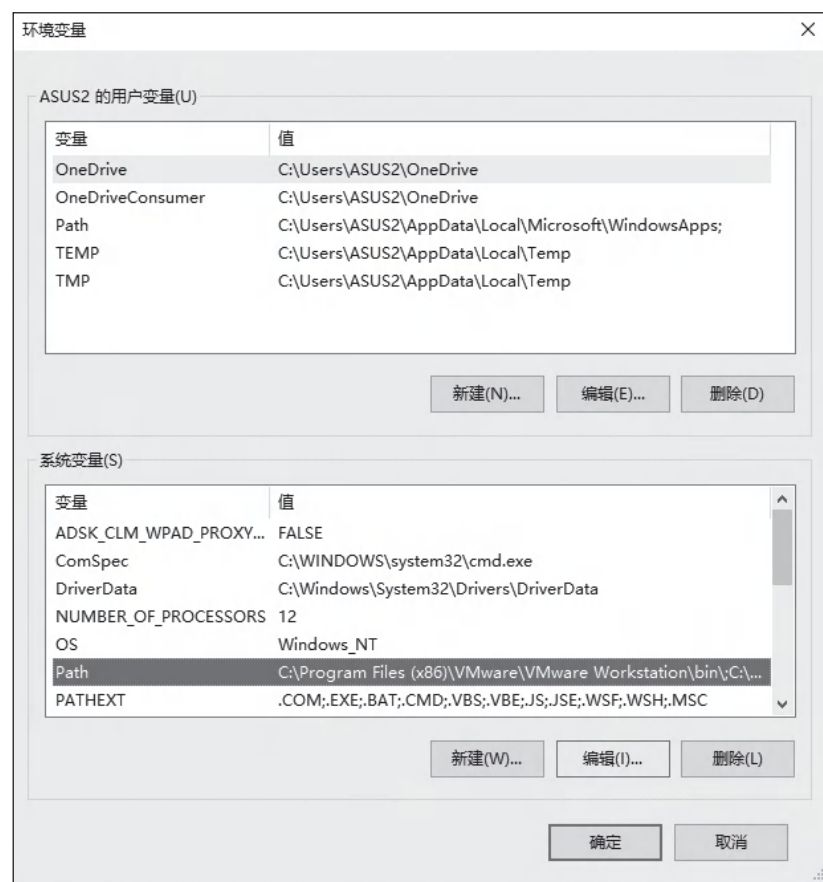


图 1-21 环境变量

在配置界面单击“新建”按钮，“%”之间的值，就是刚刚配置 JAVA_HOME 的变量名，在后面加上“\bin”，指向 Java 的 bin 目录（见图 1-22），因为 bin 目录下很多都是 Java 的命令。单击“确定”按钮，Path 环境变量就配置好了。

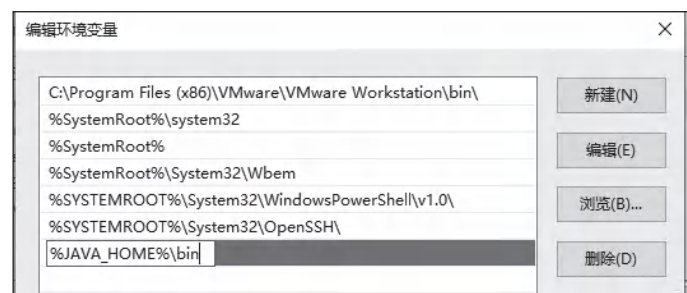


图 1-22 Path 环境变量

打开 Windows 的命令行窗口（用组合键“Win+R”，输入“cmd”，回车），在窗口输入“javac -version”和“java -version”，如果出现如图 1-23 所示的效果，则说明 Java 配置成功。

```
C:\Users\Jungk>javac -version
javac 1.8.0_251

C:\Users\Jungk>java -version
java version "1.8.0_251"
Java(TM) SE Runtime Environment (build 1.8.0_251-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)
```

图 1-23 测试 Java 的配置是否成功

1.2.3 IDE 工具介绍

随着 Java 学习的深入，我们写的程序越来越复杂，这时候使用普通的文本编辑器会出现很多问题，比如没有代码和错误的提示、没有智能的代码补齐，以及开发效率低等。因此拥有一个非常好用的工具去书写代码，对于以后的学习和开发都是很有好处的。这种“智能的工具”称为集成开发环境（integrated development environment，IDE）。比较常用的 Java IDE 有 IntelliJ IDEA、Eclipse 等。其中 IntelliJ IDEA 是由一家捷克的软件开发公司——JetBrains 开发的。本书使用的是 IDEA 社区版，在 JetBrains 官网 <https://www.jetbrains.com/> 下载，可以免费安装使用。

1.3 编写第一个 Java 程序

下面具体演示用 IDEA 创建 Java 项目。

在菜单栏中单击“File”，选择“New”→“Project”选项，如图 1-24 所示。

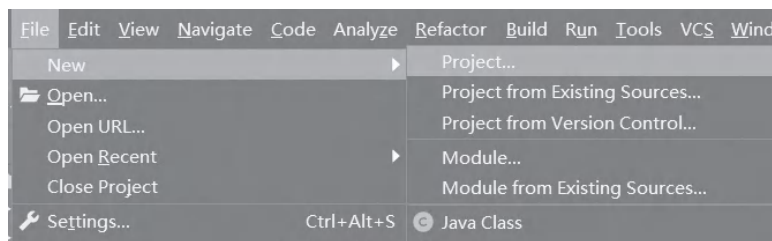


图 1-24 新建 Project

选择左栏中的“Java”选项，然后在右栏的“Project SDK”中选择本地安装的 JDK，单击“Next”按钮继续，如图 1-25 所示。

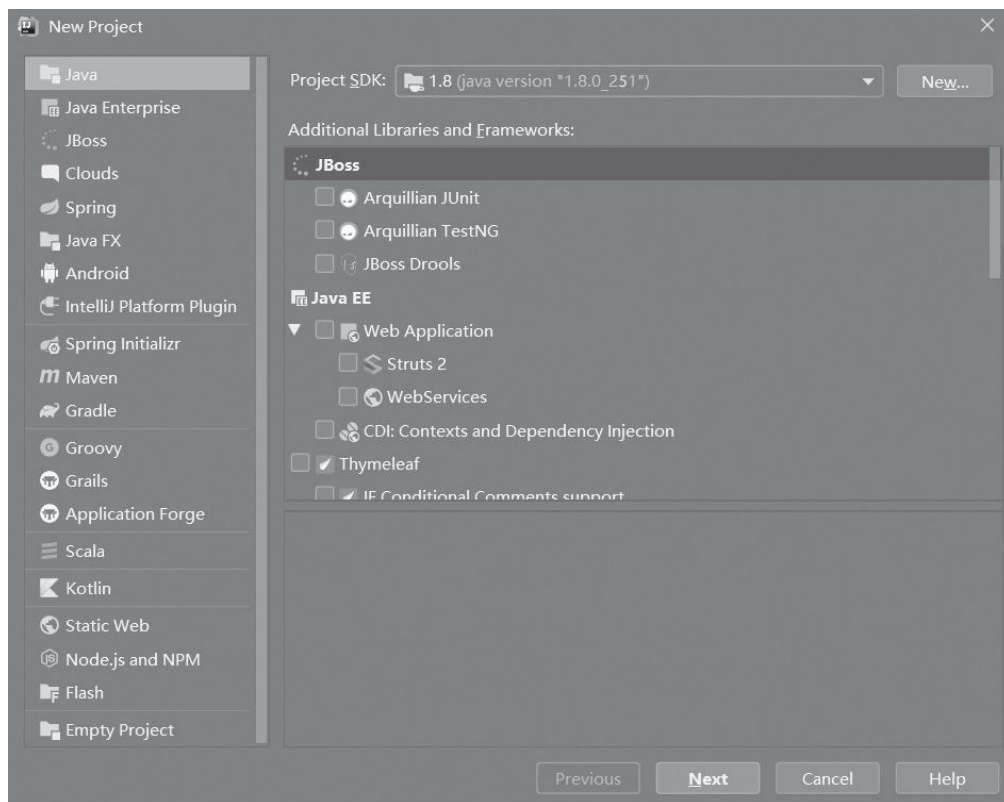


图 1-25 新建 Project SDK

选择是否采用模板来创建 Java 项目，若选择，则表示用模板来创建。本书未选择模板，直接单击“Next”按钮继续下一步，如图 1-26 所示。

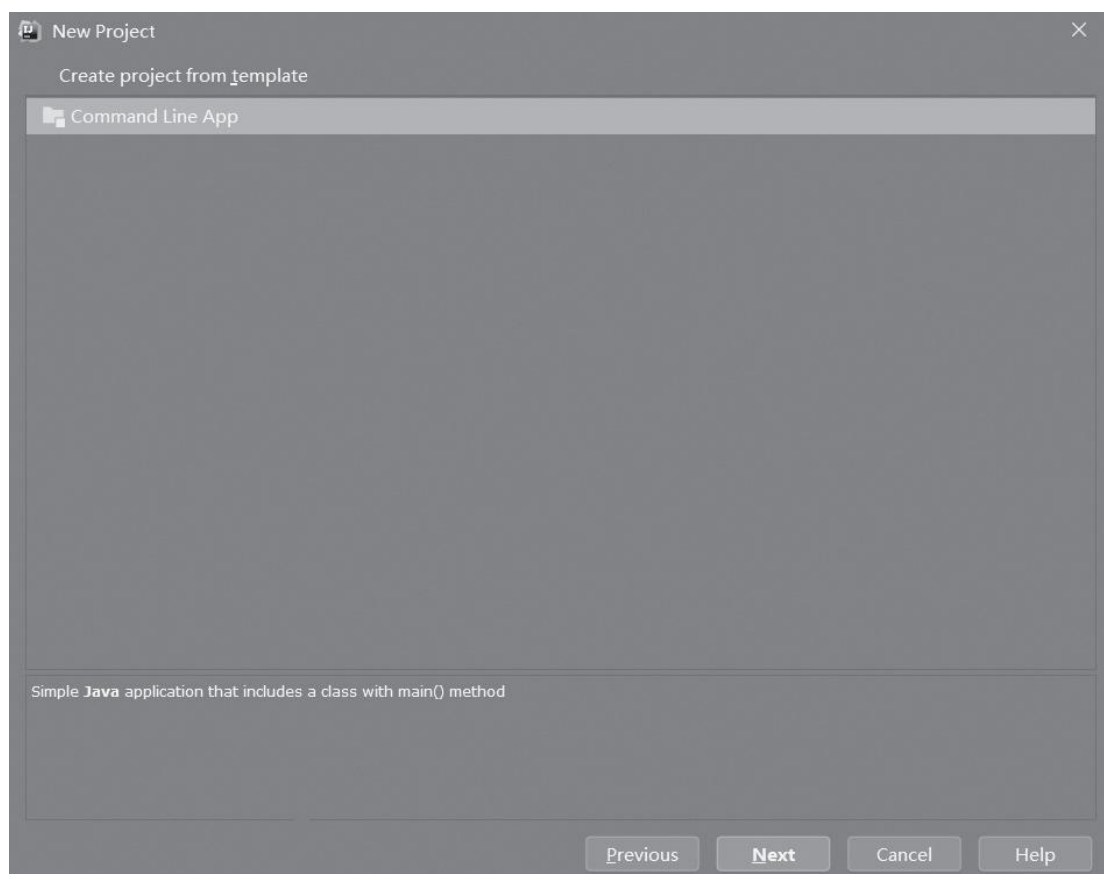


图 1-26 选择是否采用模板

输入项目名，并填写项目路径（见图 1-27），然后单击下面的“Finish”按钮（图 1-27 中未展示），出现如图 1-28 所示界面。若单击“This Window”按钮，则在本窗口打开；若单击“New Window”按钮，则在一个新窗口打开。

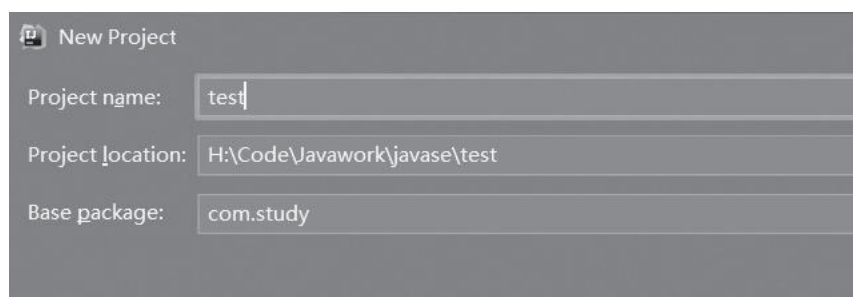


图 1-27 命名与选择路径

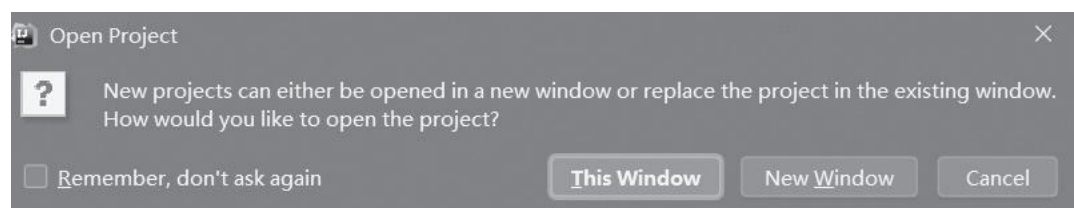


图 1-28 是否在本窗口打开

创建完成的界面如图 1-29 所示。

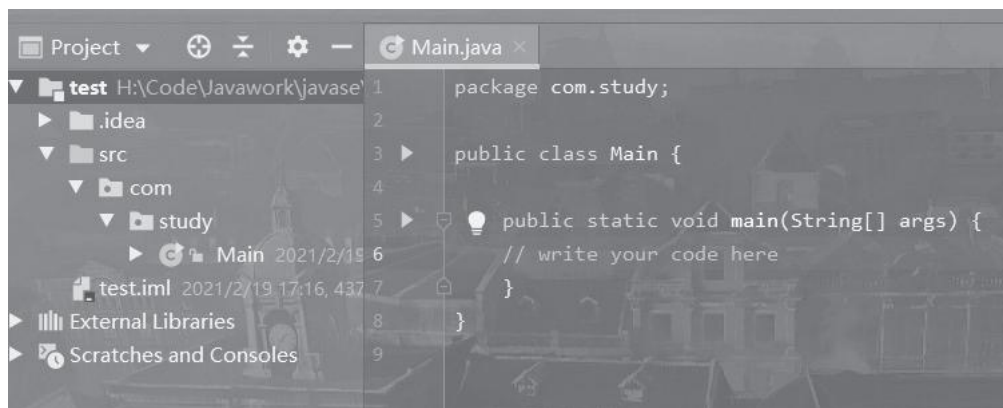


图 1-29 创建完成的界面

【例 1.2】第一个 Java 程序示例。代码如下：

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello Java!");
    }
}
```

将代码写在 main 方法中，本程序实现的功能是输出“Hello Java! ”。单击左侧的三角按钮（▶），选择“Run 'Main.main()'”即可运行程序，如图 1-30 所示。

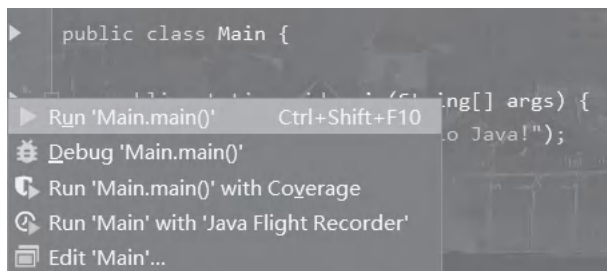


图 1-30 运行程序

本程序的运行结果如图 1-31 所示。

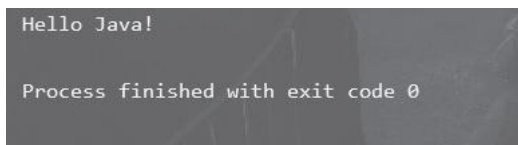


图 1-31 运行结果

拓展训练



拓展训练 (1)