



巍巍交大 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

软件开发类人才培养系列教材

.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（第2版）

Java Web开发技术任务教程

Java Web应用与开发（第2版）

PHP应用与开发

Bootstrap应用与开发

Vue.js应用与开发

jQuery应用与开发

PHP网络编程入门与进阶

项目管理

软件测试基础

软件工程与测试

计算机程序设计语言（VC++）

C语言程序设计（第2版）

C#程序设计

R语言程序设计

Android程序设计

Python程序设计项目化教程

Java语言程序设计实践教程

Java程序设计实用案例教程

Java程序设计基础案例教程

面向对象程序设计项目教程

JavaScript基础教程

网页设计基础教程（HTML5+CSS3+JavaScript）

HTML5+CSS3 Web前端设计案例教程

HTML5网页设计与实战



浙江省普通本科高校“十四五”重点立项建设教材
国家级一流本科课程配套教材

面向对象程序设计项目教程

主编 李晓蕾



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



浙江省普通本科高校“十四五”重点立项建设教材
国家级一流本科课程配套教材

面向对象程序设计 项目教程

主编 李晓蕾

MIANXIANG
DUIXIANG
CHENGXU SHEJI
XIANGMU
JIAOCHENG



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

本书提供教学资源包

网址: <https://www.sjhtbook.com>



扫描二维码
关注上海交通大学出版社
官方微信

ISBN 978-7-313-33001-7



9 787313 330017 >
定价: 49.80元



浙江省普通本科高校“十四五”重点立项建设教材
国家级一流本科课程配套教材

面向对象程序设计 项目教程

主编 李晓蕾



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

内容提要

本教材旨在通过典型项目案例,帮助读者深入理解和掌握面向对象程序设计的思维、方法和技巧。本教材精心设计了15个项目,内容涵盖Java基本语法,封装、继承、多态、抽象与接口等面向对象的基本概念,GUI与事件处理,IO和多线程,以及设计模式的应用等知识。项目内容编排由浅入深,从简单的类设计到复杂的系统开发,涵盖信息管理系统、电子商务应用、游戏开发等多个领域,使读者在实践中掌握理论知识,提升解决实际问题的能力。本教材不仅适合作为高等院校计算机相关专业的教材,也适合有一定编程基础的编程爱好者自学使用,尤其适合希望快速掌握面向对象编程技能并应用于项目开发的读者群体。

图书在版编目(CIP)数据

面向对象程序设计项目教程 / 李晓蕾主编. -- 上海 :
上海交通大学出版社, 2025. 7. -- ISBN 978-7-313-
-33001-7
I. TP312.8
中国国家版本馆 CIP 数据核字第 2025JV5604 号

面向对象程序设计项目教程

MIANXIANG DUIXIANG CHENGXU SHEJI XIANGMU JIAOCHENG

主 编: 李晓蕾	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: 17
字 数: 530 千字	
版 次: 2025 年 7 月第 1 版	印 次: 2025 年 7 月第 1 次印刷
书 号: ISBN 978-7-313-33001-7	电子书号: ISBN 978-7-89564-366-6
定 价: 49.80 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



编写委员会

主 编 李晓蕾

副主编 李 倩 林大成

编 委 胡奇跃 朱泽丹



前言

党的二十大明确了我国在科技领域的发展目标，提出要“加快实施创新驱动发展战略”“加快实现高水平科技自立自强”。在此背景下，培养具备扎实编程基础、创新思维 and 良好职业道德的计算机专业人才尤为重要。本教材聚焦软件开发主流范式面向对象程序设计，将编者多年教学经验和行业实践成果融入教材内容，旨在通过项目驱动的教学方式，培养读者面向对象程序设计的实践能力、积极探索的创新思维。同时，教材融入社会主义核心价值观，落实立德树人根本任务，为培养高素质创新型软件人才贡献力量。

本教材作为首批国家级线上线下混合式一流课程配套教材，共分为语法基础、面向对象设计思想、可视化交互设计和设计模式 4 个篇章，分别介绍了 Java 程序的基本语法及编程规范，封装、继承、多态、抽象和接口等面向对象程序设计核心思想，系统界面布局设计及交互功能的实现，以及面向对象程序设计的原则和典型设计模式。

教材内容围绕软件开发综合能力展开，主要有以下 4 个特点。

1. 项目化串联教学知识点

教材借助部分软件行业从业人员多年研发经验和校企合作单位支持，挖掘 60 个真实案例，包括 15 个驱动项目和 45 个拓展任务。15 个驱动项目串联课程所有知识点，45 个拓展任务供学生练习，并将公共交通工具租赁系统等社会热点、茶饮点单系统等日常应用、城堡游戏等趣味游戏纳入教材，吸引读者兴趣。

2. 过程化组织教学内容

教材内容按照驱动项目的工程实践开发过程编排，设置“项目任务”“项目思路”“技术准备”“项目实施”“拓展任务”5 个模块，引导读者养成规范工程项目开发流程的习惯。

3. 层次化提供拓展辅助

教材根据不同程度读者的学习需求，在每个项目设置了初级、进阶和挑战 3 级难度的拓展任务及讲解视频，满足读者个性化、渐进式提升技能的学习需求。

4. 系统化融入思政元素

设计“宁小才”和“宁小静”两个虚拟学生形象，在其发现、分析、处理实际软件项目问题的过程中，以行业视角融入行业规范、职业道德、安全意识等职业素养，以及理性严谨、精益求精和开拓创新的专业精神。

本书适合作为高等院校计算机科学与技术、软件工程等相关专业的教材，也适合对面向对象程序设计感兴趣的自学者，以及希望对现有知识进行更新升级的 IT 从业人员参考使用。

此外，本书作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 获取。

在教材的编写过程中，编者得到了众多专家、学者、校企合作单位和优秀校友的支持与帮助，在此一并表示衷心的感谢。同时，也希望广大读者能够对本教材提出宝贵的意见和建议，以便教材不断完善和改进，为培养更多具有创新精神和社会责任感的高素质信息技术人才贡献力量。

编 者

2025 年 1 月



目录



课程导入 /1



语法基础篇

项目 1 货币换算 4

- 1.1 项目任务 4
- 1.2 项目思路 4
- 1.3 技术准备 4
 - 1.3.1 开发环境 4
 - 1.3.2 运行机制 12
 - 1.3.3 Java 程序 12
 - 1.3.4 输出语句 15
 - 1.3.5 输入语句 16
- 1.4 项目实施 16
 - 1.4.1 新建项目 16
 - 1.4.2 输入港币金额 16
 - 1.4.3 计算人民币金额 17
 - 1.4.4 输出计算结果 17
 - 1.4.5 添加提示信息 17
- 1.5 拓展任务 17
 - 1.5.1 初级任务——温度转换 17
 - 1.5.2 进阶任务——计算圆的面积 18
 - 1.5.3 挑战任务——计算三角形面积 18

项目 2 在线购票系统 19

- 2.1 项目任务 19
- 2.2 项目思路 19

- 2.3 技术准备 20
 - 2.3.1 数据类型 20
 - 2.3.2 变量声明及初始化 20
 - 2.3.3 运算符 21
 - 2.3.4 分支结构 30
- 2.4 项目实施 34
 - 2.4.1 定义变量 34
 - 2.4.2 输入购票信息 34
 - 2.4.3 计算总金额 34
 - 2.4.4 输出确认信息 35
- 2.5 拓展任务 35
 - 2.5.1 初级任务——阶梯水费 35
 - 2.5.2 进阶任务——体质（量）指数 36
 - 2.5.3 挑战任务——成绩转换 37

项目 3 晚会抽奖 38

- 3.1 项目任务 38
- 3.2 项目思路 38
- 3.3 技术准备 38
 - 3.3.1 循环结构 38
 - 3.3.2 数组 45
 - 3.3.3 方法 47

3.4 项目实施	49	3.4.5 调用抽奖方法	52
3.4.1 抽奖准备	49	3.5 拓展任务	52
3.4.2 启动大屏滚动	50	3.5.1 初级任务——图形面积	52
3.4.3 定义抽奖方法	50	3.5.2 进阶任务——日期计算	52
3.4.4 定义中奖号码判重方法	51	3.5.3 挑战任务——银行账户操作	53



面向对象设计思想篇

项目 4 迎新系统

4.1 项目任务	56
4.2 项目思路	56
4.3 技术准备	56
4.3.1 面向对象的思想	56
4.3.2 类与对象的概念	57
4.3.3 类的概述	58
4.3.4 对象的创建及使用	59
4.4 项目实施	62
4.4.1 定义学生类	62
4.4.2 创建迎新系统类	62
4.4.3 定义录入新生方法	62
4.4.4 定义系统登录方法	63
4.4.5 定义系统运行方法	63
4.4.6 启动系统	64
4.5 拓展任务	64
4.5.1 初级任务——饮料信息管理	64
4.5.2 进阶任务——账户管理	65
4.5.3 挑战任务——复数	65

项目 5 迎新系统 Pro 版

5.1 项目任务	66
5.2 项目思路	66
5.3 技术准备	67
5.3.1 访问权限	67
5.3.2 封装	67

5.3.3 构造方法及其重载	69
5.3.4 this 关键字	70
5.3.5 static 关键字	72
5.4 项目实施	74
5.4.1 封装学生类	74
5.4.2 学号自增	74
5.4.3 封装 StudentSystem 类	75
5.4.4 修改 addStudent() 方法	75
5.4.5 报到人数统计	76
5.5 拓展任务	76
5.5.1 初级任务——饮料类封装	76
5.5.2 进阶任务——账户类封装	77
5.5.3 挑战任务——复数类封装	77

项目 6 会员积分系统

6.1 项目任务	78
6.2 项目思路	78
6.3 技术准备	79
6.3.1 继承	79
6.3.2 Object 类	86
6.3.3 final 关键字	88
6.3.4 多态性	90
6.4 项目实施	92
6.4.1 定义会员类	92
6.4.2 定义黄金会员类	93
6.4.3 定义钻石会员类	94

6.4.4 创建 Vip 对象	95	7.5.3 挑战任务——商品打折	106
6.5 拓展任务	95	项目 8 茶饮点单系统	107
6.5.1 初级任务——图形继承	95	8.1 项目任务	107
6.5.2 进阶任务——成绩评定	96	8.2 项目思路	107
6.5.3 挑战任务——工资管理系统	96	8.3 技术准备	108
项目 7 公共交通工具租赁系统	97	8.3.1 集合框架	108
7.1 项目任务	97	8.3.2 Collection	108
7.2 项目思路	97	8.3.3 迭代器	115
7.3 技术准备	98	8.3.4 Map 接口	116
7.3.1 抽象	98	8.3.5 泛型	121
7.3.2 接口	99	8.4 项目实施	121
7.4 项目实施	102	8.4.1 创建茶饮产品类	121
7.4.1 定义车辆租赁接口	102	8.4.2 创建用户类	122
7.4.2 定义共享车辆类	103	8.4.3 创建订单类	123
7.4.3 定义共享单车类	103	8.4.4 实现订单管理功能	124
7.4.4 定义共享汽车类	104	8.4.5 启动系统	127
7.4.5 定义共享车辆对象	105	8.5 拓展任务	127
7.5 拓展任务	105	8.5.1 初级任务——停车场收费系统	127
7.5.1 初级任务——收费系统	105	8.5.2 进阶任务——学生管理系统	128
7.5.2 进阶任务——收支管理	106	8.5.3 挑战任务——KTV 点歌系统	128



可视化交互设计篇

项目 9 计票器	130	9.5 拓展任务	139
9.1 项目任务	130	9.5.1 初级任务——教室平面图	139
9.2 项目思路	130	9.5.2 进阶任务——乘法计算器	140
9.3 技术准备	130	9.5.3 挑战任务——计算器	141
9.3.1 布局管理器	130	项目 10 校园外卖系统	142
9.3.2 事件处理机制	136	10.1 项目任务	142
9.4 项目实施	137	10.2 项目思路	142
9.4.1 创建计票器窗体	137	10.3 技术准备	143
9.4.2 添加投票组件	138	10.3.1 标签组件	143
9.4.3 按钮事件处理	138		

10.3.2	文本组件	144
10.3.3	按钮组件	147
10.3.4	下拉框组件	150
10.3.5	面板组件	152
10.4	项目实施	154
10.4.1	创建外卖系统窗体	154
10.4.2	设计外卖系统界面	154
10.4.3	提交事件处理	157
10.5	拓展任务	158
10.5.1	初级任务——实习平台注册	158
10.5.2	进阶任务——图书馆座位预约	158
10.5.3	挑战任务——电影院在线订座系统	159

项目 11 电子绘画板161

11.1	项目任务	161
------	------------	-----

11.2	项目思路	161
11.3	技术准备	162
11.3.1	菜单组件	162
11.3.2	鼠标事件	165
11.3.3	键盘事件	167
11.3.4	图形绘制	169
11.4	项目实施	171
11.4.1	设计电子绘画板界面	171
11.4.2	处理菜单事件	173
11.4.3	设置菜单快捷键	174
11.4.4	实现绘图功能	175
11.5	拓展任务	176
11.5.1	初级任务——记事本界面	176
11.5.2	进阶任务——绘制小乌龟	176
11.5.3	挑战任务——小乌龟动起来	177



设计模式篇

项目 12 简单记事本 179

12.1	项目任务	179
12.2	项目思路	179
12.3	技术准备	180
12.3.1	文件	180
12.3.2	流	181
12.3.3	异常	185
12.3.4	文件选择器	189
12.4	项目实施	191
12.4.1	编写打开 / 写入文件方法	191
12.4.2	实现打开 / 保存文件功能	192
12.4.3	实现文本编辑功能	193
12.5	拓展任务	194
12.5.1	初级任务——书店每日交易记录	194

12.5.2	进阶任务——获取网络图片	195
12.5.3	挑战任务——保存目录	195

项目 13 火车售票系统 196

13.1	项目任务	196
13.2	项目思路	196
13.3	技术准备	197
13.3.1	进程和线程	197
13.3.2	Java 中的线程实现	197
13.3.3	线程的同步	206
13.3.4	线程协作	209
13.4	项目实施	213
13.4.1	用单线程实现售票系统	213
13.4.2	用 Thread 类实现售票系统	215
13.4.3	对售票系统实施同步措施	216

13.5 拓展任务	217	14.5.2 进阶任务——工厂工人管理 问题	243
13.5.1 初级任务——电子时钟	217	14.5.3 挑战任务——图书出租管理 系统	244
13.5.2 进阶任务——生产与搬运	217		
13.5.3 挑战任务——哲学家就餐问题 ...	218		
项目 14 城堡游戏	219	项目 15 学生管理系统	245
14.1 项目任务	219	15.1 项目任务	245
14.2 项目思路	220	15.2 项目思路	245
14.3 技术准备	220	15.3 技术准备	246
14.3.1 重构	220	15.3.1 领域模型分析	246
14.3.2 面向对象设计的重要概念	221	15.3.2 MVC 模型	248
14.3.3 单一职责原则	221	15.3.3 用例实化	249
14.3.4 开闭原则	222	15.4 项目实施	251
14.3.5 里氏替换原则	224	15.4.1 系统用例规约撰写	251
14.3.6 接口隔离原则	226	15.4.2 系统用例实化	252
14.3.7 依赖倒置原则	228	15.4.3 系统 MVC 架构初步实现	253
14.3.8 迪米特法则	231	15.5 拓展任务	258
14.4 项目实施	233	15.5.1 初级任务——银行存取款系统 ...	258
14.4.1 城堡游戏的初步实现	233	15.5.2 进阶任务——图书管理系统	258
14.4.2 用单一职责原则优化项目	237	15.5.3 挑战任务——升级学生信息管理 系统	259
14.4.3 用封装降低类之间耦合	238		
14.4.4 框架和数据分离	241		
14.5 拓展任务	243	参考文献	260
14.5.1 初级任务——万能遥控器	243		

课程 导人

宁小才和宁小静是计算机专业的学生，本学期要开始跟李老师学习面向对象程序设计。小才想成为一名优秀的软件工程师，他程序设计基础学得不错，想通过本学期的学习开发一个酷炫的原创游戏；小静的记忆力很好，能熟练记住编程语法，但面对具体编程问题时常常没有思路，她想通过这门课程提升一下自己的计算思维和编程技术，为今后的专业课程学习打好基础。为了能更好地达成学习目标，小才和小静一起找到李老师，希望老师对课程学习给予一些个性化的建议。

小才：李老师，学完这门课程，我可以编写怎样的系统？

老师：这门课程是以 Java 为语言背景介绍面向对象程序设计的思想和技巧。通过课程学习，大家可以编写像 Windows 绘图板、植物大战僵尸这样的桌面应用程序，还可以开发学校教务系统这样的 Web 应用程序，更可以打造像淘宝这样的分布式系统。

小才：哇，太酷炫了，那这门课程是不是不好学？

老师：Java 的语法并不难，但是要从面向过程编程转向面向对象开发，还是要经历一段思维转变的过程，所以在没有领悟到面向对象程序设计的精髓之前，还是会费点儿精神，一旦领悟了，将会得心应手。

小静：老师，我之前程序设计基础学得不够好，不知道会不会影响这门课程的学习呢？

老师：完全不用有这方面的顾虑，有没有编程基础都可以学好这门课程，前提是用对方法、多多练习。

小静：那我们应该用什么方法学习呢？

老师：学习程序设计没有捷径可走，正确的打开方式只有实践、实践、再实践。语法是死的，但是用语法解决问题的方法是多种多样的。所以，在学习时一定要以项目为载体多做练习，除了课程中学习的项目外，也非常建议同学自己去发现生活中可以用系统解决的问题，独立开展实践探索。

小才、小静：好的，老师。我们知道了，一定加倍努力，实践、实践、再实践。

老师：那还等什么，赶快开启学习之旅吧。



语法基础篇

随着数字经济的发展，我国软件行业正面临着前所未有的发展机遇和挑战。作为一种广泛使用的编程语言，Java语言在企业级应用、移动开发、大数据处理等领域都有重要应用，因此，学好Java语言不仅有助于IT人的职业发展，还能为国家的振兴贡献力量。本篇章将以项目为载体，围绕基础语法重点介绍Java程序的框架、数据类型、分支和循环结构及方法的定义与重载等内容，帮助读者掌握Java编程的基础语法，为后续的学习奠定基础，也为开发更复杂的Java应用程序提供必要的技能支持。

知识目标

1. 了解Java程序的运行原理。
2. 掌握Java程序基本框架和基本语法。
3. 掌握主流的Java SE集成开发环境搭建和配置。
4. 掌握分支、循环结构及方法的定义与重载。

能力目标

1. 能正确使用变量及表达式。
2. 能正确使用if、switch分支结构对程序流程进行控制。
3. 能正确使用for、while、do-while、foreach循环结构对程序流程进行控制。
4. 能使用方法对程序结构进行分解和抽象。
5. 能够编写简单的程序来解决实际问题。

素质目标

1. 培养良好的编程规范。
2. 培养严谨的逻辑思维。



知识检测

项目 1

货币换算



项目导入

宁小才同学在国际大学生软件创意大赛中表现优异，受组委会邀请，他将于 10 月赴香港某大学，参加为期一周的作品展示和设计经验交流活动。小才了解到随着数字人民币的试点推广和人民币国际地位的不断提升，现在在香港消费可以直接采用人民币结算，小才深感自豪，同时也意识到了一个问题：如何在香港交流期间快速将商品的港币标价换算成人民币价格呢？为了解决这个问题，他决定利用自己的专业优势，开发一个小程序，实现港币与人民币之间的自动货币换算。

1.1 项目任务



效果演示——
货币换算

货币兑换是指将一种货币按一定的汇率转换为另一种货币。汇率通常受多种经济因素的影响。尤其是在香港旅行或进行商业交易时，将人民币（CNY）兑换成港币（HKD）是常见的需求。开发一个货币转换小程序，用户输入商品的港币价格，程序自动按照汇率进行换算，假设港币对人民币的汇率是 1 : 0.8394，输出对应的人民币价格。

1.2 项目思路

根据项目任务的描述，本项目需要先搭建 Java 运行环境，并创建一个 Java 应用程序，根据要求实现货币转换功能，具体实现过程如下。

- (1) 搭建开发环境，包括安装 JDK（Java 语言软件开发工具包）、配置环境变量及安装开发工具。
- (2) 创建 Java 项目。
- (3) 编写代码实现货币转换功能。输入港币价格数据，通过公式“人民币价格 = 汇率 × 港币价格”换算人民币价格，输出计算结果。

1.3 技术准备

1.3.1 开发环境

1. 安装 JDK

1) 下载 JDK 安装包

在浏览器网址栏输入“<https://www.oracle.com/>”，打开 Oracle 官方网站，Oracle 官网如图 1-1 所示。选

择“Products”选项，在“Hardware and Software”栏中选择“Java”选项，进入Java界面，Java界面如图1-2所示。单击“Download Java”按钮，进入下载页面，根据Windows/Linux/macOS等不同操作系统下载不同版本的JDK，Oracle官方JDK下载页面如图1-3所示。

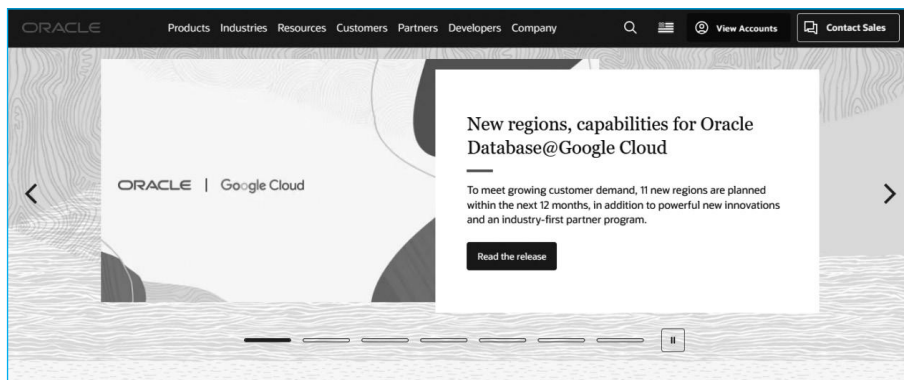


图 1-1 Oracle 官网

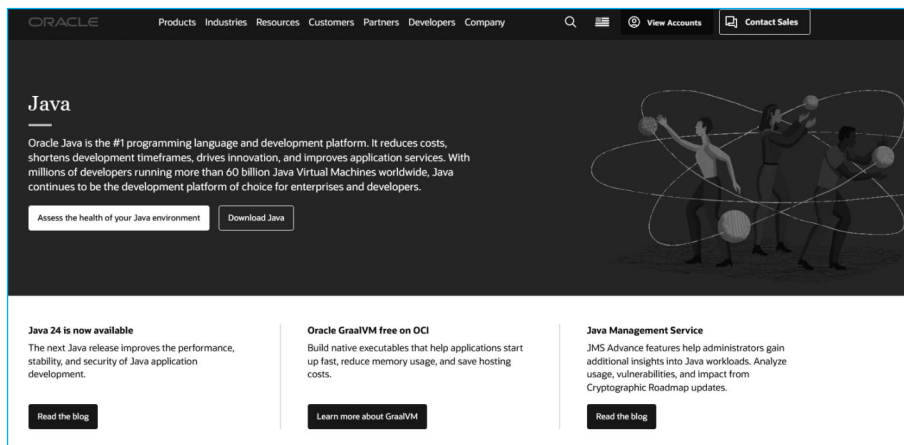


图 1-2 Java 界面

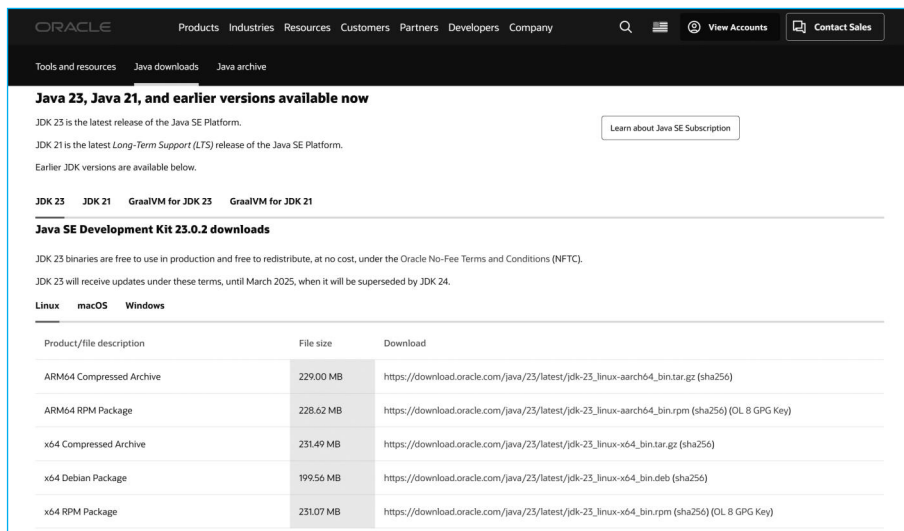


图 1-3 Oracle 官方 JDK 下载页面

截至本书编写时，官方提供的最新版本为 JDK 23，但鉴于 JDK 17 为官方发布的长期支持（Long-Term Support, LTS）版本，有很好的稳定性和安全性，是目前行业内企业的主流选择，因此本书以 JDK 17 版本为例讲解安装及配置过程。

单击图 1-3 中的“Java archive”选项卡，进入 Java 历史版本下载界面，Java 历史版本下载界面如图 1-4 所示。选择“Java SE downloads”栏中的“Java SE 17(17.0.12 and earlier)”选项，进入下载界面，选择合适的版本下载即可。

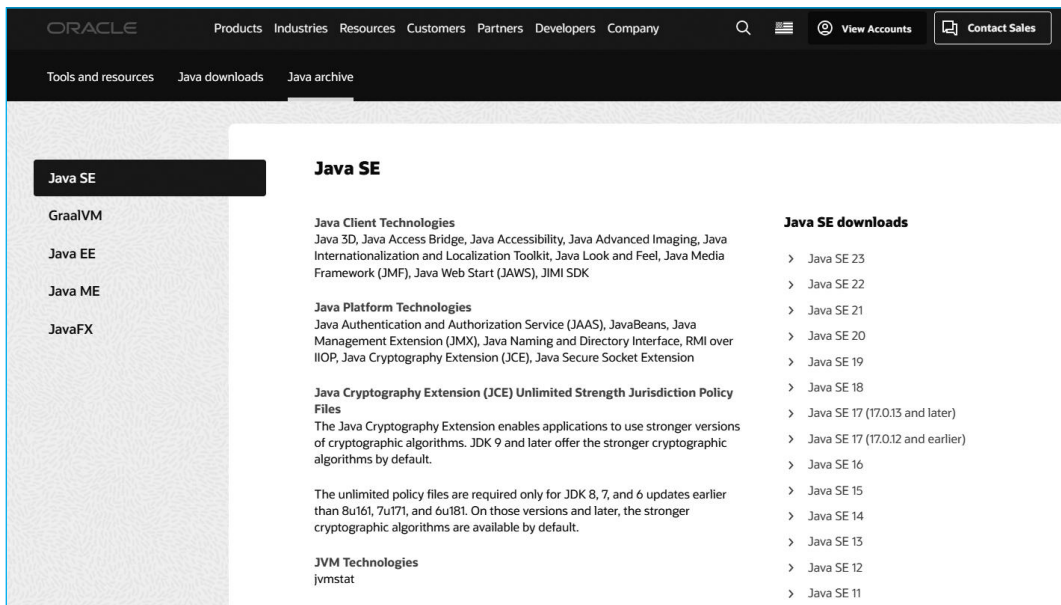


图 1-4 Java 历史版本下载界面

2) 安装 JDK 过程

双击下载的 JDK 17.0.8 安装包，进入安装向导页面，JDK 安装向导页面如图 1-5 所示。

单击“下一步”按钮，进入 JDK 目标文件夹界面，JDK 目标文件夹界面如图 1-6 所示。用户可以通过单击“更改”按钮选择 JDK 的安装位置。



图 1-5 JDK 安装向导页面



图 1-6 JDK 目标文件夹界面

单击“下一步”按钮开始安装，安装结束后会跳转到完成界面，JDK 安装完成界面如图 1-7 所示。单击“关闭”按钮，完成 JDK 的安装工作。



图 1-7 JDK 安装完成界面

3) JDK 目录介绍

JDK 安装完毕后会 在磁 盘上生成一个目录，该目录被称为 JDK 目录，JDK 目录如图 1-8 所示。

bin	2024/8/12 10:44	文件夹
conf	2024/8/12 10:44	文件夹
include	2024/8/12 10:44	文件夹
jmods	2024/8/12 10:45	文件夹
legal	2024/8/12 10:45	文件夹
lib	2024/8/12 10:45	文件夹
LICENSE	2024/8/12 10:44	文件
README	2024/8/12 10:44	文件
release	2024/8/12 10:45	文件

图 1-8 JDK 目录

JDK 目录下的子目录及其作用各不相同，各个子目录的介绍如下。

- （1）bin 目录。该目录主要存放 JDK 的各种工具命令，如 Java 编译器（javac.exe）、Java 运行工具（java.exe）、打包工具（jar.exe）、文档生成工具（javadoc.exe）等。其中 javac.exe 负责将 Java 源文件（扩展名为 .java）编译成 Java 可执行文件（扩展名为 .class），java.exe 会启动 Java 虚拟机进程，运行 Java 可执行文件。
- （2）conf 目录。该目录主要存放 JDK 的相关配置文件，如权限、安全、网络、日志等。
- （3）include 目录。由于 JDK 是使用 C 和 C++ 语言开发的，因此在启动时需要引入一些头文件。该目录就用于存放这些头文件。
- （4）jmods 目录。该目录主要用于存放 JDK 的各种模块。从 Java 9 开始，原有的 Java 标准库由一个单一巨大的 rt.jar 拆分成了几十个模块，这些模块以 .jmod 扩展名标识。
- （5）legal 目录。该目录存放 JDK 各模块的授权文档。
- （6）lib 目录。该目录主要存放 JDK 工具的一些补充 jar 包和源代码 src.zip。

2. 环境变量配置

环境变量是指在操作系统中用来指定操作系统运行环境的一些参数，如临时文件夹位置和系统文件夹位置等。Java 11 以后，开发环境只需配置 path 系统环境变量（Java 8 以前还需要配置 classpath 环境变量，指定 .class 文件位置），用来指定可执行文件 java.exe 和 javac.exe 的路径，这样无论当前目录在哪里，操作系

统都可以通过环境变量设置的路径，找到需要执行的文件。

1) 配置 path 环境变量

右击桌面上的“此电脑”图标按钮，选择菜单中的“属性”选项，打开“设置”界面，“设置”界面如图 1-9 所示。



图 1-9 “设置”界面

选择“高级系统设置”选项，打开“系统属性”界面，“系统属性”界面如图 1-10 所示。单击“环境变量”按钮，进入配置界面，环境变量配置界面如图 1-11 所示。选择“系统变量”栏中的“Path”选项，单击“编辑”按钮，即可进行环境变量配置。



图 1-10 “系统属性”界面

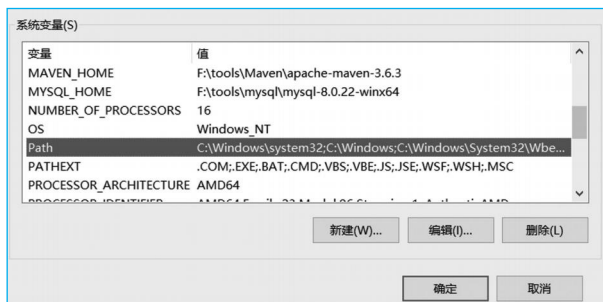


图 1-11 环境变量配置界面

查看 JDK 的安装路径，如果 JDK 的安装路径为“C:\Program Files\Java\jdk-17”，那么 Java 开发环境必要组件和工具则在“C:\Program Files\Java\jdk-17\bin”目录下。单击“编辑”按钮，在对话框中将其添加到变量值中即可，系统变量编辑界面如图 1-12 所示。注意，变量值之间用“;”隔开。

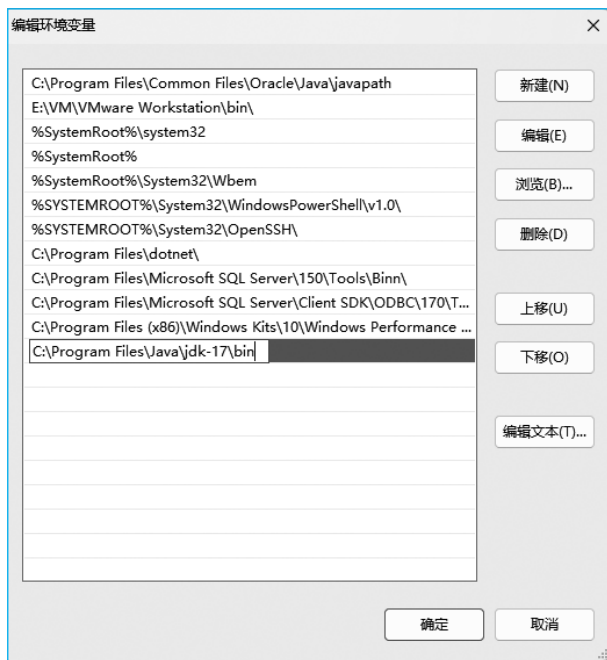


图 1-12 系统变量编辑界面

2) 配置验证

按下“Win+R”组合键，输入“cmd”回车，运行命令行工具。键入命令“java -version”，如果配置正确，则显示当前安装的 Java 版本信息，配置正确显示效果如图 1-13 所示。需要注意的是，显示结果会根据安装的 JDK 版本不同略有差异。



图 1-13 配置正确显示效果

3. 集成开发环境

Java 的集成开发环境有很多，如 NetBeans、Eclipse、IntelliJ IDEA 等。其中，IntelliJ IDEA 是 JetBrains 公司的产品，在智能代码助手、代码自动提示、重构、Java EE 支持、各类版本工具（如 git、svn 等）、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面功能强大。并且 JetBrains 公司为教育者提供了免费学习账号支持，师生可以通过公司提供的免费许可证计划申请免费使用权限。因此，本书将以 IntelliJ IDEA 作为集成开发环境。

1) 版本选择

IntelliJ IDEA 在官方网站“<https://www.jetbrains.com/idea/>”上提供了 IntelliJ IDEA Ultimate 和 IntelliJ IDEA Community Edition 两个版本，IntelliJ IDEA 下载界面如图 1-14 所示。

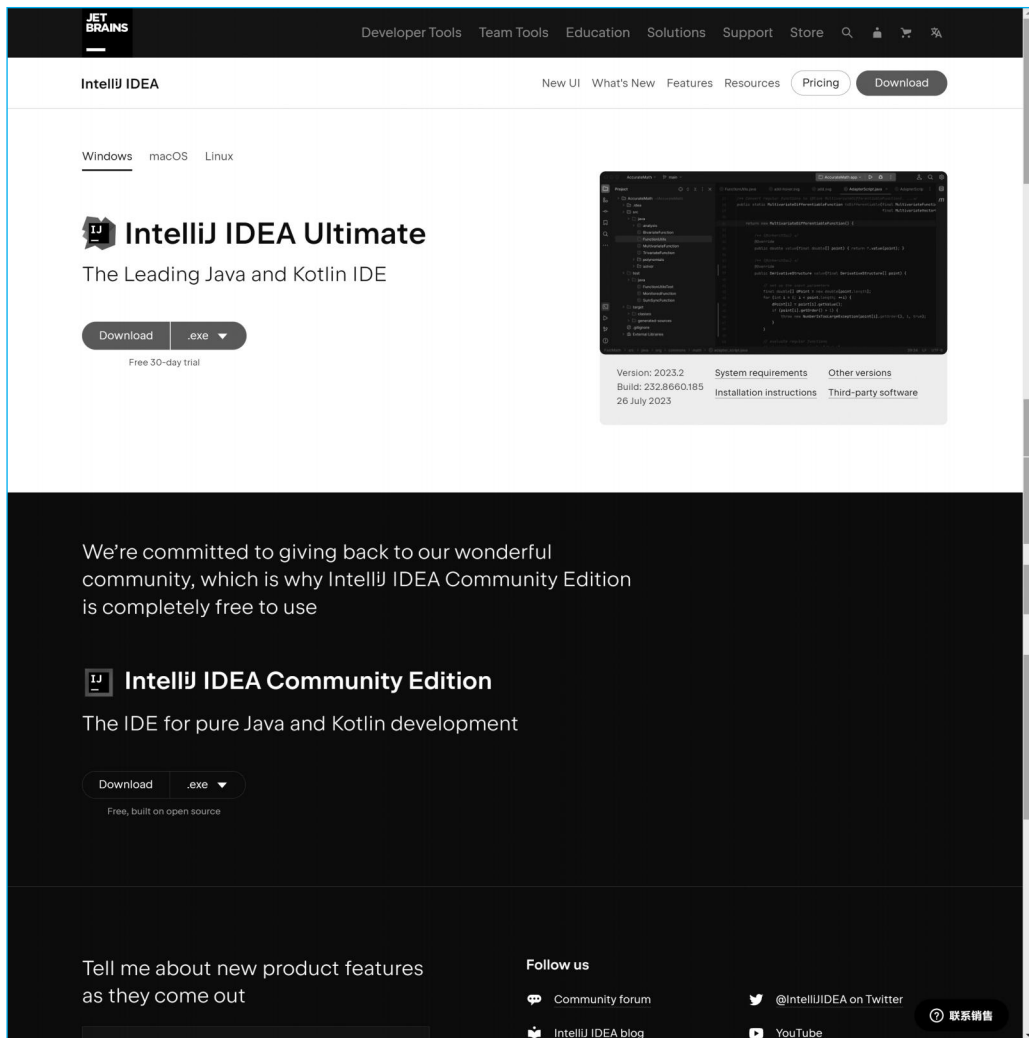


图 1-14 IntelliJ IDEA 下载界面

(1) IntelliJ IDEA Ultimate。该版本为旗舰版，提供了更多的高级功能，如代码审查、重构、UML 设计、Spring Framework 支持和更丰富的插件等。在语言支持方面，除了支持 Java 开发外，还支持 Groovy、Scala、Kotlin 等语言，适合专业的开发者或团队使用，需购买许可证书或申请教育者账号进行使用。本书使用组件和功能更加齐全的旗舰版。

(2) IntelliJ IDEA Community Edition。该版本为社区版，提供 Java 基础的开发功能，不收取许可费用。

2) IntelliJ IDEA 安装

单击 Ultimate 版的“Download”按钮，下载安装文件。下载完成后双击安装包，弹出 IDEA 安装欢迎界面，IDEA 安装欢迎界面如图 1-15 所示。

单击“Next”按钮，弹出安装路径设置界面，安装路径设置界面如图 1-16 所示。界面中显示了 IDEA 默认的安装路径，单击“Browse”按钮可以修改路径。



图 1-15 IDEA 安装欢迎界面

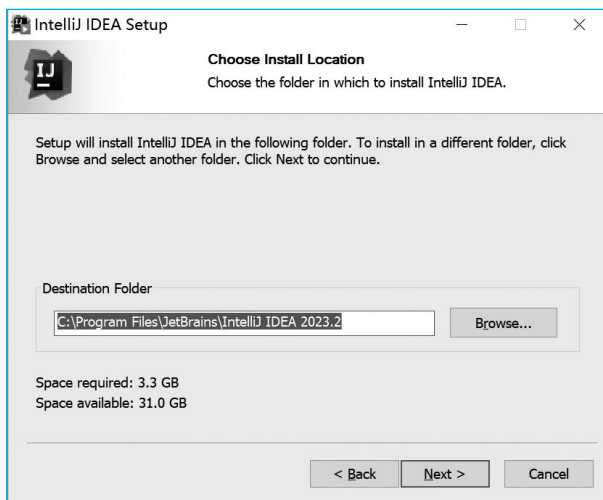


图 1-16 安装路径设置界面

设置完成后，单击“Next”按钮，弹出安装基本选项选择界面，如图 1-17 所示，用户可以进行一些个性化设置。

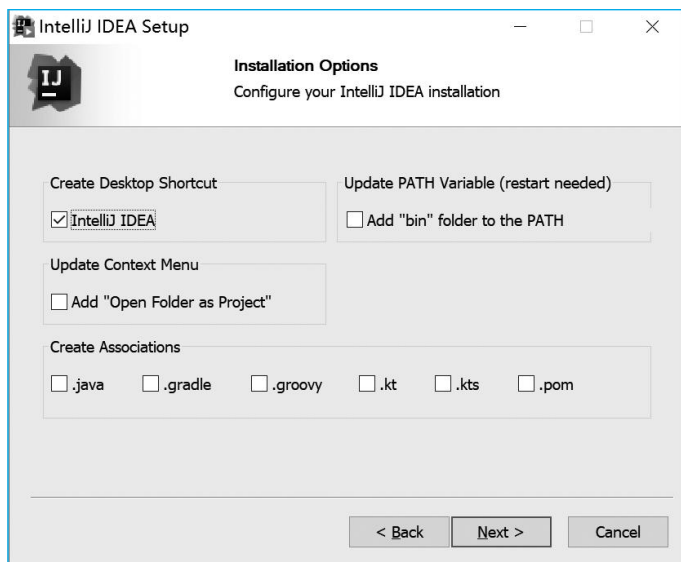


图 1-17 安装基本选项选择界面

从图 1-17 可以看出，基本安装选项分为 4 个部分，详细介绍如下。

- (1) Create Desktop Shortcut: 创建桌面快捷方式图标，本书勾选 IntelliJ IDEA。
- (2) Update PATH Variable (restart needed): 将 IDEA 启动目录添加到环境变量中，即可以从命令行中启动 IDEA，根据需要勾选。
- (3) Update Context Menu: 将“从文件夹打开项目”选项添加至鼠标右键，根据需要勾选。
- (4) Create Associations: 关联文件格式，选择后将默认使用 IDEA 打开相关文件，根据需求选择。

单击“Next”按钮，进入选择开始菜单界面，选择开始菜单界面如图 1-18 所示。单击“Install”按钮，开始安装 IDEA，IDEA 安装完成界面如图 1-19 所示。

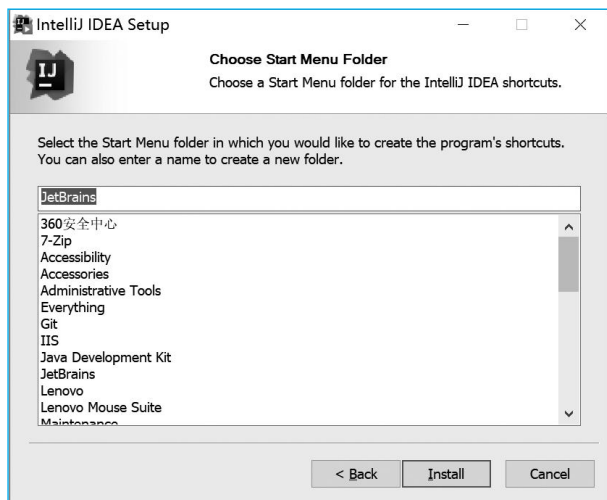


图 1-18 选择开始菜单界面

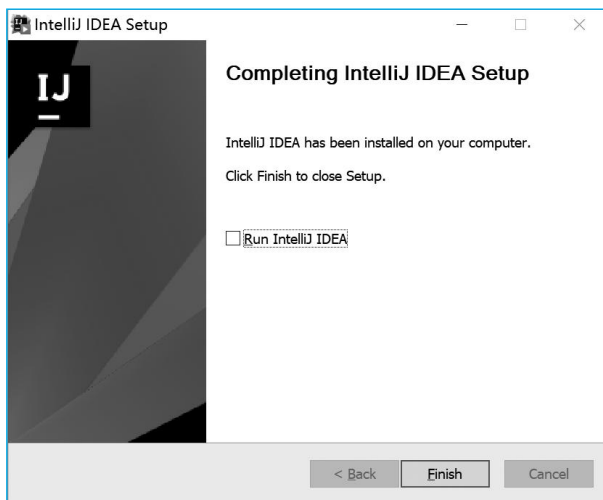


图 1-19 IDEA 安装完成界面

1.3.2 运行机制

Java 程序最显著的特点就是跨平台，那么它是如何做到不必重新编译就能在不同操作系统上运行的呢？

Java 源程序被编译成后缀为 .class 的字节码文件，这种字节码文件在 Java 虚拟机（Java virtual machine，JVM）上解释执行，Sun 公司为不同版本操作系统设计了对应的 JVM，这样 Java 程序运行便与操作系统无关了。Java 程序跨平台原理如图 1-20 所示。

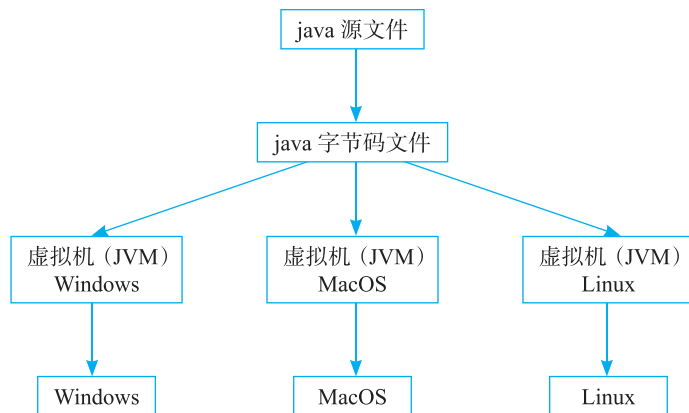


图 1-20 Java 程序跨平台原理

1.3.3 Java 程序

1. Java 程序基本框架

Java 程序的基本框架如下：

```

class 主类名称 {
    public static void main(String[] args){
        // 方法体
    }
}
  
```

```
}  
}
```

(1) `class` 是一个关键字，用于定义一个类。在 Java 中，类相当于一个文件，代码要写在类中。每个类都要有名称，关键字 `class` 与类名之间需要用空白字符进行分隔，类名后要有一对花括号，表示类的管辖范围。

(2) “`public static void main(String[] args)`” 语句定义 Java 程序的主方法，它是 Java 程序的入口，程序将从 `main` 方法花括号内的代码开始依次执行。

需要注意的是，一个 Java 程序可以由多个类组成，但 `main` 方法只能有一个。

2. 创建 Java 程序项目

打开 IntelliJ IDEA 集成开发环境的主界面，IDEA 主界面如图 1-21 所示。单击“New Project”按钮，弹出新建项目界面，新建项目界面如图 1-22 所示。输入项目名称，将第一个项目命名为“HelloWorld1_1”，设置项目路径为“D:\workspace”，JDK 版本为 17.0.8。如果用户安装了多个 JDK 版本，可以通过下拉框按需选择。单击“Create”按钮，完成项目创建。

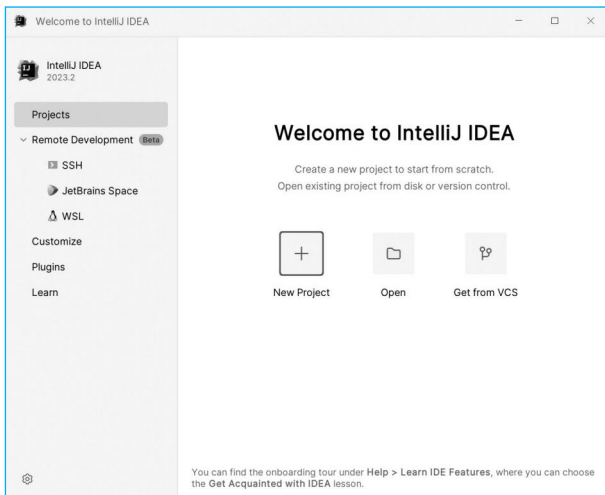


图 1-21 IDEA 主界面

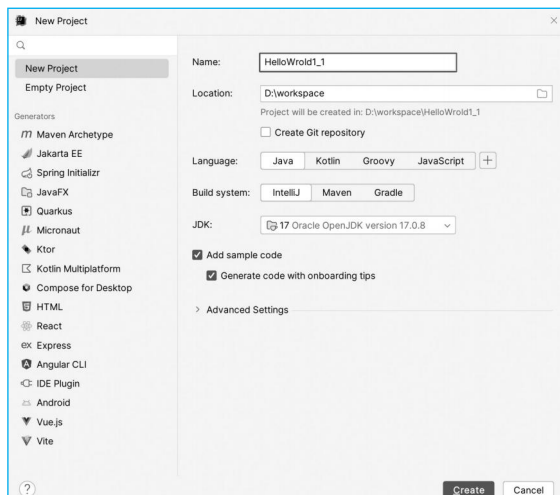


图 1-22 新建项目界面

3. Java 项目目录结构

在“Project”视图中，空项目目录中包含 `.idea` 文件夹、`src` 文件夹及外部库 `External Libraries` 等内容，空项目目录结构如图 1-23 所示。

- (1) `.idea` 文件夹：存放项目的配置信息，包括历史记录、版本控制信息等。
- (2) `src` 文件夹：存放项目源代码目录。
- (3) `External Libraries`：项目运行所需的外部库。

4. 创建程序源文件

右击 `src` 文件夹，执行“New”→“Java Class”命令，在类创建向导中输入类名“HelloWorld”，新建源文件界面如图 1-24 所示。按回车键，完成类的创建。

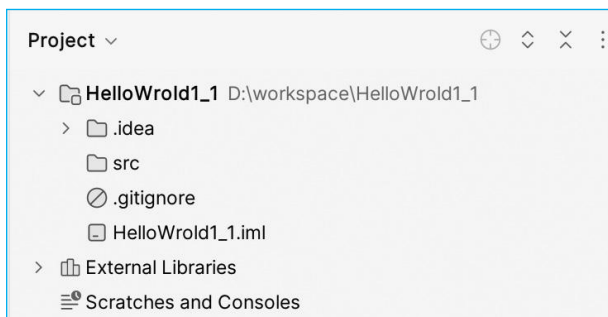


图 1-23 空项目目录结构

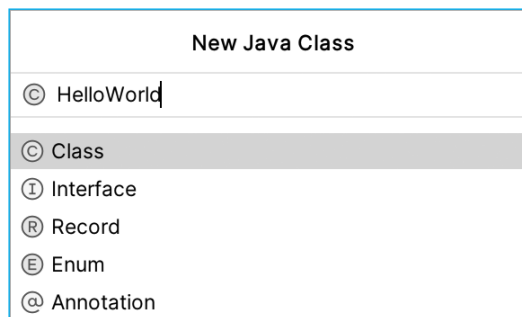


图 1-24 新建源文件界面

可以看到，在 src 文件夹下，出现了一个名为“HelloWorld”的源文件，并且该文件自动在代码编辑区域打开，HelloWorld.java 源文件如图 1-25 所示。

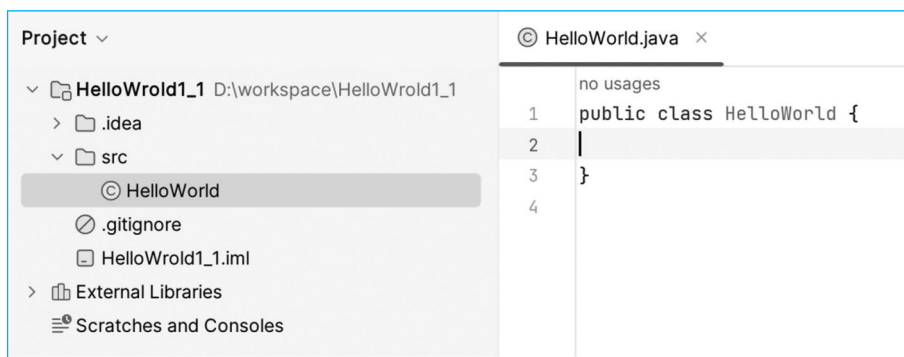


图 1-25 HelloWorld.java 源文件

为了便于项目源文件的管理，在程序开发中会将编写的类分目录存放。为此，Java 中引入了包机制，通常将相同功能的类存放在一个包中，通过 package 关键字声明。例如，右击 src 文件夹，执行“New”→“Package”命令，输入包名“com.nbufo.edu”，即可创建一个包。可以将源文件“HelloWorld”移动到新建的包下，打开源文件，可以看到源文件的头部自动出现了 package 声明，分包存放源文件如图 1-26 所示。

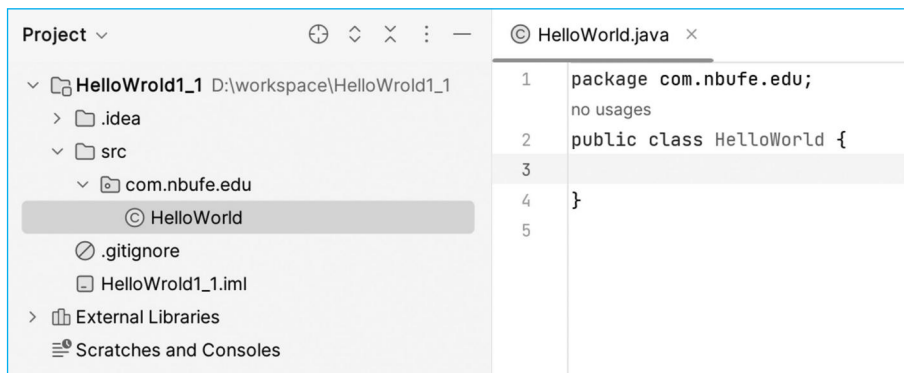


图 1-26 分包存放源文件

5. 编辑代码及运行程序

在文本编辑器中，可以完成程序代码的编写。例如，下面程序中，只在 main 方法中编辑了一条输出语句“System.out.println("Hello World");”。

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

程序编辑完成后，可以右击项目主类文件，执行“Run 'HelloWorld.main()’”命令，运行程序，其中 HelloWorld 为主类的名称，也可以通过单击工具栏中的快捷按钮“”运行程序，运行结果将显示在控制台中，运行结果如图 1-27 所示。

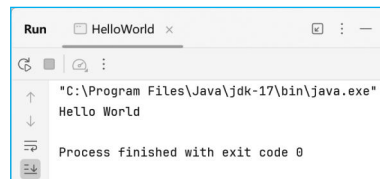


图 1-27 运行结果

1.3.4 输出语句

Java 的输出语句有多种实现方法，其定义都在 System.out 包下，具体包括 printf、print 和 println 3 种。

1. printf

printf 是格式化输出的形式，其语法与 C 语言基本一致，格式如下：

```
System.out.printf(format,args);
```

其中，参数 format 规定了输出的格式，包括字符串 (%s)、整型 (%d)、浮点型 (%f) 等，args 代表格式说明符引用的参数。例如，字符串变量 name^① 的值为“宁小才”，语句“System.out.printf(“我是 %s”,name);”的输出结果如下：

```
我是宁小才
```

2. print

print 是带有一个参数的输出方法，其作用是将参数显示在命令窗口，并将输出光标定位在所显示的最后一个字符之后，格式如下：

```
System.out.print(args);
```

其中，参数 args 支持输出字符串、整型、浮点型、布尔型、数组、对象等类型。例如，main 函数中的代码如下：

```
double rate = 3.5;  
System.out.print(" 我是宁小静! ");  
System.out.print(rate);
```

两条 print 语句的输出结果如下：

```
我是宁小静! 3.5
```

3. println

println 与 print 的用法基本一致，区别是 println 在输出完成后，在语句最后增加换行符。

^① 为与编写代码时计算机显示效果一致，本书变量统一采用正体表示。

1.3.5 输入语句

java.util.Scanner 是 Java5 的新特征，它是一个文本扫描器，可以通过 Scanner 类来扫描并获取用户的输入。

1. 导入 Scanner 类

要创建 Scanner 对象，首先需要导入 Scanner 类，导入语句如下：

```
import java.util.Scanner;
```

2. 创建 Scanner 对象

要扫描用户输入，首先需要创建一个 Scanner 对象用以调用各个扫描方法。例如，创建一个名为 “in” 的 Scanner 对象，代码如下：

```
Scanner in = new Scanner(System.in);
```

3. 接收输入数据

Scanner 类提供了一系列的扫描接收输入方法，支持多种数据类型。代码示例如下：

```
int age = in.nextInt();  
double weight = in.nextDouble();  
String name = in.next();
```

其中，nextInt() 接收整型数据、nextDouble() 接收浮点型数据、next() 接收字符串数据。

1.4 项目实施

1.4.1 新建项目

打开 IDEA，单击 “New Project” 按钮，进入新建项目界面。输入项目名称为 “Exchange1_2”，完成项目创建。右击源代码目录文件夹 src，执行 “New” → “Java Class” 命令，输入 “Exchange” 完成类的创建，代码如下：

```
public class Exchange{  
    public static void main(String[] args) {  
    }  
}
```

1.4.2 输入港币金额

在 main 方法中，定义 Scanner 对象，获取用户输入的港币金额并赋值给变量 “hkd”，代码如下：

```
Scanner input = new Scanner(System.in);
```

```
double hkd = input.nextDouble();
```

1.4.3 计算人民币金额

定义人民币价格变量“cny”，将港币价格变量“hkd”乘以汇率的计算结果赋值给变量“cny”，代码如下：

```
double cnv = hkd * 0.8394;
```

1.4.4 输出计算结果

用输出语句输出人民币价格变量“cny”的值，代码如下：

```
System.out.println(cny);
```

1.4.5 添加提示信息

为了使用户体验更加友好，可以在输入港币金额前加入提示信息“请输入港币金额(元):”。在输出时，也可以增加输出信息提示，代码如下：

```
System.out.println("请输入港币金额(元):"); // 输入提示  
System.out.println(hkd + "元" + "换算成人民币为" + cnv + "元"); // 输出信息提示
```

1.5 拓展任务

1.5.1 初级任务——温度转换

写一个将华氏温度转换成摄氏温度的程序，转换的公式是：

$$C = (F - 32) \times 5/9$$

其中，“C”表示摄氏温度，“F”表示华氏温度。



任务讲解——
温度转换

思路提示

编写华氏温度与摄氏温度的转换程序，需要设计程序结构。程序的输入是一个整数，表示华氏温度。输出也是一个整数，表示华氏温度对应的摄氏温度。

(1) 输入。用户输入华氏温度。

(2) 处理。使用温度转换公式 $C = (F - 32) \times 5/9$ 计算摄氏温度。需要注意的是，C 是整型，要防止右侧进行相除操作时取整数，直接导致结果为 0。

(3) 输出。将计算得到的摄氏温度输出到控制台。



任务讲解——
计算圆的面积

1.5.2 进阶任务——计算圆的面积

编写程序，要求输入圆的半径 r ，计算并输出圆的面积。其中，圆的半径需要精确到小数点后 2 位。

思路提示

编写程序时，需先设计程序结构。

- (1) 输入。程序需要从用户那里获取圆的半径。
- (2) 处理。使用圆的面积公式 $area = \pi * r * r$ 计算面积。其中 $area$ 是面积， r 是半径， π 是圆周率。 π 可以使用 Math 库中的常量 PI， r 的平方可以使用 $r * r$ ，也可以使用 Math 库中的 pow() 方法。
- (3) 输出。将计算得到的面积输出到控制台。



任务讲解——
计算三角形
面积

1.5.3 挑战任务——计算三角形面积

编写程序，输入三角形的三条边长 a 、 b 、 c ，计算并输出三角形的面积。

思路提示

编写程序时，需先根据任务需求设计程序结构。

- (1) 输入。程序需要从用户那里获取三角形的三条边长。
- (2) 验证。可以通过检查任意两边之和是否大于第三边，来判断输入的边长是否可以构成三角形。
- (3) 处理。如果输入有效，可以使用海伦公式 $area = \sqrt{l \times (l-a) \times (l-b) \times (l-c)}$ 来计算三角形的面积。其中， l 为三角形的半周长，计算公式为 $l = (a+b+c)/2$ 。
- (4) 输出。将计算得到的面积输出到控制台。

项目 2

在线购票系统



项目导入

宁小才是一个天文爱好者，从“嫦娥”奔月、“天问”探火，到“天宫”遨游、“北斗”织网，他始终密切关注着我国天文事业的蓬勃发展。他得知香港太空馆不仅有先进的数码天象投影系统，让人能够沉浸式观看天体运动，还特别展出了我国航空事业从无到有、蓬勃发展的辉煌历程珍贵照片。宁小才深感振奋，计划在赴香港某大学交流期间去博物馆参观学习，感受我国航空事业的巨大成就。但香港太空馆只能线下购票，小才觉得如果能开发一款购票系统，支持在线预订门票，购票会更加便捷。

2.1 项目任务

为香港太空馆开发一款在线购票系统，票价可以随参观时段、门票种类的改变进行调整。该程序可以输入时段、种类及购票数量，计算并输出应付金额。假设具体票价标准如表 2-1 所示。

表 2-1 香港太空馆门票价目表

时段	种类	价格	购票条件
平日	成人票	30 HKD	1. 儿童票：身高 1.2 米至 1.4 米孩童 2. 学生票：中小学及全日制在校大学生，出示有效学生证 3. 老人票：60 周岁（含）以上老人，出示有效身份证件
	学生票	半价	
	老人票	6 折	
	儿童票	10 HKD	
节假日 (周末及法定 节假日)	成人票	45 HKD	
	学生票	半价	
	老人票	6 折	
	儿童票	15 HKD	



效果演示——
在线购票系统

2.2 项目思路

根据项目任务描述，具体实现思路如下。

- (1) 输入相应的信息，如时段、门票种类、购票数量等。
- (2) 根据输入的时段、种类确定单张票价，用票价乘以购票数量，计算订单的总金额。
- (3) 输出应付总金额。

2.3 技术准备

2.3.1 数据类型

Java 中有两大类数据类型，分别是基本数据类型和引用数据类型。其中，基本数据类型包括 8 种，分别为字节、短整型、整型、长整型、单精度浮点型、双精度浮点型、字符型及布尔型，Java 基本数据类型如表 2-2 所示；引用数据类型涵盖 class（类）、interface（接口）和 Array（数组）3 种，同时可以通过类库或自定义扩展功能，如 Integer 就是类库中定义的整型数据类型，用户也可以根据需求自定义类作为数据类型，Java 引用数据类型如表 2-3 所示。

表 2-2 Java 基本数据类型

类别	名称	数值
整数型	byte	$-128 \sim 127$
	short	$-2^{15} \sim 2^{15}-1$
	int	$-2^{31} \sim 2^{31}-1$
	long	$-2^{63} \sim 2^{63}-1$
浮点型	float	$-3.403\text{E}38 \sim 3.403\text{E}38$
	double	$-1.798\text{E}308 \sim 1.798\text{E}308$
字符型	char	采用 unicode 编码
布尔型	boolean	值只能为 true 或 false

表 2-3 Java 引用数据类型

类别	备注
类	String、Double、Scanner 等
接口	List、Set、Map 等
数组	int[]、char[] 等

2.3.2 变量声明及初始化

变量主要用于保存输入、输出及程序运行过程中的中间数据。在 Java 语言中，所有的变量在使用前必须声明。声明变量的基本格式如下：

```
type identifier [= value][, identifier [= value] ...];
```

其中，“type”表示 Java 合法数据类型，“identifier”表示变量名称，“value”代表变量的初始值，多个变量之间用“,”分隔。

需要注意的是，变量命名是有规范的。具体规范有以下几点。

- (1) 一般变量名由数字、字母和下划线组成，其中首字符必须是字母或下划线。
- (2) 开发过程中尽量使用与变量实际含义相符的英文单词进行命名，做到见名知义。
- (3) 对于由多个单词组成的变量名，推荐使用小驼峰命名法，即第一个单词首字母小写，其余单词首字

母大写。例如，变量“学生姓名”可以命名为 `studentName`。

1. 整型变量的定义及赋值

整型中最常用的是 `int` 类型，如果需要使用 `long` 型，则需要在数值中添加后缀 `L`。理论上 `L` 不区分大小写，但是小写的英文字母“`l`”容易与数字的“`1`”混淆，所以一般建议用大写的 `L`。示例代码如下：

```
byte bit = 1;
short age = 20;
int amount = 20;
long stuNo = 20110301L; // 或 long amount = 1000l;
```

2. 浮点型变量的定义及赋值

在浮点型中，`double` 类型是应用最多的。如果需要使用 `float` 类型，就需要在数值中添加后缀 `F` 或 `f`。示例代码如下：

```
float rate = 0.28F; // 或 float rate = 0.28f;
double pi = 3.1415926;
```

如果写成“`float f = 3.5;`”，则系统默认 `f` 为 `double` 类型，而左边实际定义的是 `float`，程序编译就会报错。

3. 字符变量定义及赋值

字符变量的值可以是字符常量，需要使用单引号括起来；也可以是字符常量的 ASCII 码（美国信息交换标准代码），如“`98`”代表字符“`b`”；还可以使用转义字符，也需要使用单引号括起来。示例代码如下：

```
char c1 = 'a';
char c2 = 98;
char c3 = '\\';
```

4. 布尔型变量定义及赋值

由于布尔类型的常量值只能是 `true` 或 `false`，因此 `boolean` 型变量只能赋值为 `true` 或者 `false`。示例代码如下：

```
boolean failure = false;
boolean success = true;
```

2.3.3 运算符

Java 有 9 大类运算符，包括算术运算符、赋值运算符、自增和自减运算符、比较运算符、逻辑运算符、位运算符、移位运算符、条件运算符和字符串连接运算符。这些运算符可以组合成表达式，进而执行更复杂的计算和操作。

1. 算数运算符

1) 加法运算符

加法运算符（`+`）用于进行基本的加法运算。示例代码如下：

```
int a = 1, b = 2;  
int sum = a + b;  
System.out.println(" 两数相加的值为: " + sum);
```

运行结果如下:

两数相加的值为: 3

2) 减法运算符

减法运算符(-)用于进行基本的减法运算。示例代码如下:

```
int a = 2, b = 1;  
int sub = a - b;  
System.out.println(" 两数相减的值为: " + sub);
```

运行结果如下:

两数相减的值为: 1

3) 乘法运算符

乘法运算符(*)用于进行基本的乘法运算。示例代码如下:

```
int a = 5, b = 3;  
int mul = a * b;  
System.out.println(" 两数相乘的值为: " + mul);
```

运行结果如下:

两数相乘的值为: 15

4) 除法运算符

除法运算符(/)用于进行基本的除法运算。示例代码如下:

```
int a = 5, b = 1;  
int div = a / b;  
System.out.println(" 两数相除的值为: " + div);
```

运行结果如下:

两数相除的值为: 5

5) 求余运算符

求余运算符(%)用于进行基本的取余运算。示例代码如下:

```
int a = 12, b = 9;  
int rem = a % b;  
System.out.println(" 两数取余的值为: " + rem);
```

运行结果如下:

两数取余的值为：3

2. 赋值运算符

1) 简单赋值运算符

简单赋值运算符(=)用于将右侧的值赋给左侧的变量，进行基本的赋值运算。示例代码如下：

```
int a = 10;  
System.out.println(a);
```

运行结果如下：

10

2) 复合赋值运算符

复合赋值运算符用于在变量赋值的同时执行某些操作，包括加法赋值、减法赋值、乘法赋值、除法赋值等运算符。

(1) 加法赋值运算符(+=): 用于将右侧的值与左侧变量的值相加，并将结果赋给左侧变量，如“a+=3;”等价于“a=a+3;”。示例代码如下：

```
int n = 10;  
n += 4;  
System.out.println(n);
```

运行结果如下：

14

(2) 减法赋值运算符(-=): 用于将右侧的值与左侧变量的值相减，并将结果赋给左侧变量，如“a-=3;”等价于“a=a-3;”。示例代码如下：

```
int n = 10;  
n -= 4;  
System.out.println(n);
```

运行结果如下：

6

(3) 乘法赋值运算符(*=): 用于将右侧的值与左侧变量的值相乘，并将结果赋给左侧变量，如“a*=3;”等价于“a=a*3;”。示例代码如下：

```
int n = 3;  
n *= 2;  
System.out.println(n);
```

运行结果如下：

6

(4) 除法赋值运算符 (/=)：用于将右侧的值与左侧变量的值相除，并将结果赋给左侧变量，如“a/=3;”等价于“a=a/3;”。示例代码如下：

```
int n = 3;
n /= 3;
System.out.println(n);
```

运行结果：

1

需要注意的是，使用复合赋值运算符时会发生强制类型转换，当赋值运算符右边的操作数与左边的操作数的数据类型不一致时，会将右边的操作数强制转换成与左边一致的数据类型再进行运算。

3. 自增和自减运算符

1) 自增运算符

自增运算符(++)放在变量前则表示先自增后取值，放在变量后则表示先取值后自增。示例代码如下：

```
int a = 2, b;
b = ++a;
System.out.println("a = " + a + " b = " + b);
int c = 2, d;
d = c++;
System.out.println("c = " + c + " d = " + d);
```

运行结果如下：

```
a = 3 b = 3
c = 3 d = 2
```

2) 自减运算符

自减运算符(--)放在变量前则表示先自减后取值，放在变量后则表示先取值后自减。示例代码如下：

```
int a = 2, b;
b = --a;
System.out.println("a = " + a + " b = " + b);
int c = 2, d;
d = c--;
System.out.println("c = " + c + " d = " + d);
```

运行结果如下：

```
a = 1 b = 1
c = 1 d = 2
```

4. 比较运算符

1) 小于运算符

小于运算符 (<) 在左侧表达式的值小于右侧时返回 true，否则返回 false。示例代码如下：

```
int a = 2, b = 3;  
System.out.println(a < b);
```

运行结果如下：

```
true
```

2) 大于运算符

大于运算符 (>) 在左侧表达式的值大于右侧时返回 true，否则返回 false。示例代码如下：

```
int a = 2, b = 3;  
System.out.println(a > b);
```

运行结果如下：

```
false
```

3) 相等运算符

相等运算符 (==) 用于比较运算符左侧和右侧表达式值是否相等，若相等则返回 true，否则返回 false。需要注意的是，如果左侧和右侧均为引用类型，则判断的并非对象的值，而是对象的地址。示例代码如下：

```
System.out.println(4 == 4);  
System.out.println(97 == 'a');  
System.out.println(5.0 == 5);  
System.out.println(true == false);
```

运行结果如下：

```
true  
true  
true  
false
```

4) 小于等于运算符

小于等于运算符 (<=) 在左侧表达式的值小于等于右侧时返回 true，否则返回 false。示例代码如下：

```
System.out.println(4 <= 2);
```

运行结果如下：

```
false
```

5) 大于等于运算符

大于等于运算符 (>=) 在左侧表达式的值大于等于右侧时返回 true，否则返回 false。示例代码如下：

```
System.out.println(4 >= 2);
```

运行结果如下：

```
true
```

6) 不等于运算符

不等于运算符 (!=) 在左侧表达式的值不等于右侧时返回 true，否则返回 false。示例代码如下：

```
System.out.println(4 != 2);
```

运行结果如下：

```
true
```

5. 逻辑运算符

1) 逻辑与运算符

逻辑与运算符 (&&) 在运算符左侧和右侧的表达式均为 true 时返回 true，否则返回 false。示例代码如下：

```
System.out.println(2 > 1 && 3 < 4);
```

运行结果如下：

```
true
```

2) 逻辑或运算符

逻辑或运算符 (||) 在运算符左侧、右侧的表达式有一个为 true 时返回 true，否则返回 false。示例代码如下：

```
System.out.println(2 < 1 || 3 > 4);
```

运行结果如下：

```
false
```

3) 逻辑非运算符

逻辑非运算符 (!) 在运算符右侧的表达式为 true 时返回 false，否则返回 true。示例代码如下：

```
System.out.println(!(2 > 4));
```

运行结果如下：

```
true
```

6. 位运算符

1) 按位与运算符

按位与运算符 (&) 表示按位进行与运算。需要注意的是，任何数与 0 进行按位与运算，其结果都为 0。示例代码如下：

```
int x = 5, y = 12; // 创建整型变量保存两个数
int z = x & y; // 对这两个数进行按位与运算，将结果保存到 z
System.out.println("z = " + z);
```

按位与运算过程如图 2-1 所示。

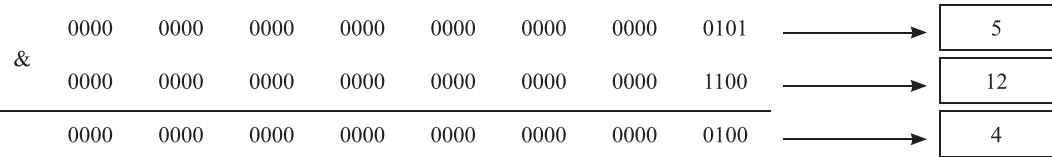


图 2-1 按位与运算过程

运行结果如下：

```
z = 4
```

2) 按位或运算符

按位或运算符 (|) 表示按位进行或运算。对应的二进制位只要有一个为 1，那么结果就为 1；如果对应的二进制位都为 0，结果才为 0。示例代码如下：

```
int x = 11, y = 7; // 创建整型变量保存两个数
int z = x | y; // 对这两个数进行按位或运算，将结果保存到 z
System.out.printf("z = " + z);
```

按位或运算过程如图 2-2 所示。

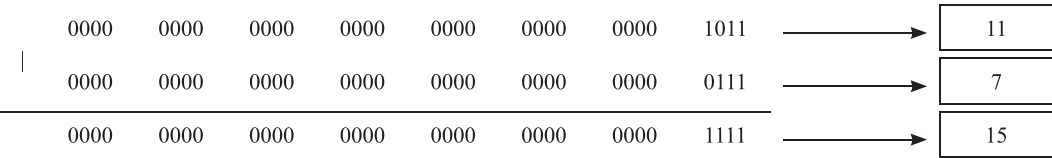


图 2-2 按位或运算过程

运行结果如下：

```
z = 15
```

3) 按位取反运算符

按位取反运算符 (~) 表示按位进行取反运算，只对一个操作数进行运算，将操作数二进制中的 1 改为 0，0 改为 1。示例代码如下：

```
int i = 10;
System.out.printf("%x \n", ~ i);
```

按位取反运算过程如图 2-3 所示。

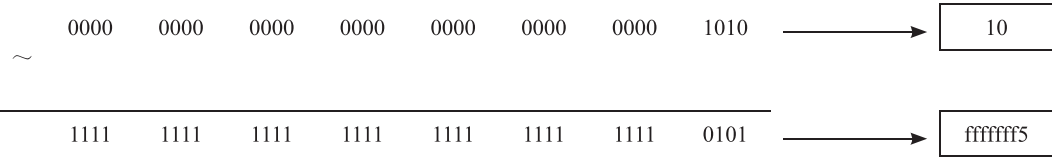


图 2-3 按位取反运算过程

运行结果如下：

```
ffffff5
```

4) 按位异或运算符

按位异或运算符 (^) 表示按位进行异或运算，如果对应的二进制位相同，结果为 0；反之结果为 1。示例代码如下：

```
int x = 11, y = 7; // 创建整型变量保存两个数
int z = x ^ y; // 对这两个数进行按位异或运算，将结果保存到 z
System.out.println("z = " + z);
```

按位异或运算过程如图 2-4 所示。

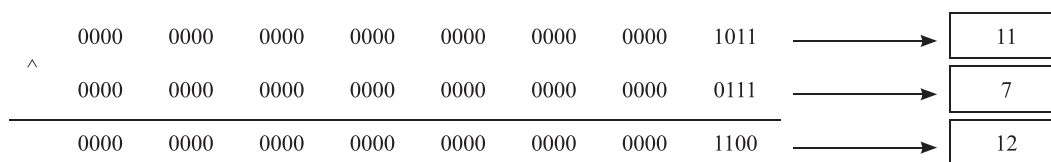


图 2-4 按位异或运算过程

运行结果如下：

```
z = 12
```

7. 移位运算符

1) 右位移运算符

右位移运算符 (>>) 的语法结构为 `result = exp1 >> exp2`，其作用是把 `exp1` 的所有位向右移动 `exp2` 位数。`exp1` 的符号位被用来填充右移后左边空出来的位，向右移出的位将被丢弃。示例代码如下：

```
int temp;
temp = -14 >> 2;
System.out.println("temp = " + temp);
```

运行结果如下：

```
temp = -4
```

2) 左位移运算符

左位移运算符 (<<) 的语法结构为 `result = exp1 << exp2`，其作用是把 `exp1` 的所有位向左移动 `exp2` 位数。示例代码如下：

```
int temp;
temp = 14 << 2;
System.out.println("temp = " + temp);
```

运行结果如下：

```
temp = 56
```

3) 无符号右移运算符

无符号右移运算符 (>>>) 的语法结构为 `result=exp1 >>> exp2`, 其作用是把 `exp1` 的各个位向右移动 `exp2` 位数, 右移后左边空出的位用零来填充, 向右移出的位将被丢弃。示例代码如下:

```
int temp;  
temp = -14 >>> 2;  
System.out.println("temp = " + temp);
```

运行结果为:

```
temp = 1073741820
```

8. 条件运算符

条件运算符 (?) 的语法结构为 `exp1?exp2:exp3`, 其作用为判断 `exp1` 是否成立, 如果成立, 则计算结果为 `exp2`, 否则为 `exp3`。示例代码如下:

```
int x, y, z;  
x = 6;  
y = 2;  
z = x > y ? x - y : x + y;  
System.out.println("z = " + z);
```

运行结果如下:

```
z = 4
```

9. 字符串连接运算符

前几类运算符和 C 语言中的含义与用法基本一致, 而字符串连接运算符和 C 语言中的概念不太相同, 因此本书将着重介绍字符串连接运算符。其基本格式为 `op1 + op2`, 其中 `op1` 和 `op2` 中至少有一个是字符串, 另一个可以是任意合法的数据类型。示例代码如下:

```
String s = "宁小才的年龄: ";  
int age = 20;  
System.out.println(s + age);
```

以上程序将输出“宁小才的年龄: 20”, 这里的“age”可以是任何类型的数据。

```
int stuNo = 11020301;  
System.out.println("" + stuNo);
```

以上程序输出为“11020301”, 看似没有变化, 但实际上输出内容的类型已经改变, 不再是整型, 而是字符串类型。

2.3.4 分支结构

1. if-else 语句

if-else 是分支控制中的重要语句，它可以让程序有选择地执行。

1) 单分支语句

当条件表达式为 true 时，就会执行“{}”中的代码，反之不执行。其基本语法如下：

```
if( 条件表达式 ){  
    代码块;  
}
```

需要注意的是，在 C 语言中，“if(1)”可以代表条件恒成立，但在 Java 中这种写法会产生编译错误，可以写成“if(true)”，恒不成立可以写成“if(false)”。

🔧【案例 2-1】编写程序，实现判断用户是否可以租车的功能。如果大于 12 岁，可以租借公共自行车，具体代码如下：

```
import java.util.Scanner;  
public class Bike {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        System.out.println(" 请输入年龄 ");  
        int age = in.nextInt();  
        if (age > 12) {  
            System.out.println(" 请扫码租车！ ");  
        }  
    }  
}
```

2) 双分支语句

当条件表达式成立时，执行代码块 1，否则执行代码块 2。其基本格式如下：

```
if( 条件表达式 ){  
    代码块 1;  
}else{  
    代码块 2;  
}
```

🔧【案例 2-2】编写程序，根据学生性别判断选修的体育课科目。若为男生，则选择篮球；若为女生，则选择瑜伽，具体代码如下：

```
import java.util.Scanner;  
public class SportCourse {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);
```

```

        System.out.println(" 请输入性别: ture 代表男, false 代表女 ");
        boolean sex = in.nextBoolean();
        if (sex) {
            System.out.println(" 选修篮球 ");
        } else {
            System.out.println(" 选修瑜伽 ");
        }
    }
}

```

3) 多分支语句

当条件表达式 1 成立时, 执行代码块 1; 如果不成立, 则判断条件表达式 2, 若成立, 执行代码块 2, 以此类推, 如果所有条件表达式均不成立, 则执行代码块 n+1。其基本格式如下:

```

if( 条件表达式 1){
    代码块 1;
}else if( 条件表达式 2){
    代码块 2;
}
:
else{
    代码块 n+1;
}

```

多分支语句可以没有 else, 当所有条件表达式均不成立时, 则不执行任何代码块。

 **【案例 2-3】** 编写程序, 实现分段函数 y 的求解。

$$y = \begin{cases} 0 & (x < -10) \\ \sqrt{x+10} & (-10 \leq x < 100) \\ 5x+1 & (x \geq 100) \end{cases}$$

可以使用多分支结构, 具体代码如下:

```

import java.util.Scanner;
public class Function {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println(" 请输入 x 的值: ");
        double x = in.nextDouble();
        double y = 0;
        if (x < -10) {
            y = 0;
        } else if (x >= -10 && x < 100) {
            y = Math.sqrt(x + 10);
        }
    }
}

```

```

        } else {
            y = 5 * x + 1;
        }
        System.out.println(" 函数 y 的值: " + y);
    }
}

```

2. switch-case 语句

switch-case 语句也可以实现分支结构，其基本语法如下所示。当条件表达式等于常量 1 时执行代码块 1，等于常量 2 时执行代码块 2，以此类推，如果一个都没有匹配上，则执行 default 中的代码块 n+1。

```

switch( 条件表达式 ){
    case 常量 1: 代码块 1;
    case 常量 2: 代码块 2;
    :
    case 常量 n: 代码块 n;
    default : 代码块 n+1;
}

```

需要注意的是，case 子句中必须是常量，不能是变量；代码块最后可以加入 break 语句，使得程序不再继续往下执行，直接退出 switch 分支判断；default 子句是可选的，如果不写，则表示当没有匹配的 case 子句时，程序不执行任何分支代码块。

 **【案例 2-4】** 编写程序，实现输入月份，输出该月份有几天的功能。具体代码如下：

```

import java.util.Scanner;
public class Month {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println(" 请输入月份 :");
        int month = in.nextInt();
        int day = 0;
        switch (month) {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                day = 31;
                break;
            case 2:
                day = 28;
                break;

```

```
        case 4:
        case 6:
        case 9:
        case 11:
            day = 30;
            break;
        default:
            System.out.println(" 您的输入有误! ");
    }
    System.out.println(month + " 月:" + day + " 天 ");
}
}
```

3. 嵌套分支

在一个分支结构中嵌套另一个完整的分支结构，里面的分支结构被称为内层分支，外面的分支结构被称为外层分支。理论上分支结构可以无限嵌套，但实际编程时需考虑到程序的可读性，最好不要超过 3 层。

【案例 2-5】 编写程序，实现判断歌手是否晋级的功能。若歌手初赛成绩大于 8.0 分则进入决赛，否则提示淘汰。进入决赛的选手根据性别提示进入男子组或者女子组，具体代码如下：

```
import java.util.Scanner;
public class Nested {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println(" 请输入该歌手的成绩 ");
        double score = in.nextDouble();
        if (score > 8.0) {
            System.out.println(" 请输入性别 ");
            char gender = in.next().charAt(0);
            if (gender == '男') {
                System.out.println(" 进入男子组 ");
            } else if (gender == '女') {
                System.out.println(" 进入女子组 ");
            } else {
                System.out.println(" 你的性别有误，不能参加决赛～ ");
            }
        } else {
            System.out.println("sorry，你被淘汰了～");
        }
    }
}
```

2.4 项目实施

2.4.1 定义变量

在程序编写中需要定义时段、种类、数量、金额等变量。其中，时段分为淡季和旺季，因此采用布尔型；门票种类分为4种，可以使用字符A、S、O、C分别代表成人票、学生票、老人票和儿童票，采用字符串类型；此外购票的数量应为整型，而购票的总金额需要按照类型进行打折计算，因此采用浮点型，代码如下：

```
boolean time;
String ticketType = "";
int ticketNum;
double totalPrice = 0.0;
```

2.4.2 输入购票信息

创建 Scanner 对象 input，根据购票时段、种类和数量等变量的数据类型，依次调用相应的方法接收时段 time、门票类别 ticketType 及购买数量 ticketNum 的值，代码如下：

```
Scanner input = new Scanner(System.in);
System.out.println(" 请选择时段 (true 代表平日 ,false 代表节假日 ): ");
time = input.nextBoolean();
System.out.println(" 请输入字母选择购票类型 (A 代表成人票 ,S 代表学生票 ,O 代表老人票 ,C 代表儿  
童票 ): ");
String option = input.next();
System.out.println(" 请输入购买票数: ");
ticketNum = input.nextInt();
```

2.4.3 计算总金额

可以先按照节假日和平日分为两个分支，再分别嵌套不同门票种类进行票价总金额计算，代码如下：

```
if (time) {
    if (option.equals("A")) {
        ticketType = " 成人票 ";
        totalPrice = ticketNum * 30;
    } else if (option.equals("S")) {
        ticketType = " 学生票 ";
        totalPrice = ticketNum * 15;
    } else if (option.equals("O")) {
        ticketType = " 老人票 ";
        totalPrice = ticketNum * 30 * 0.6;
    }
}
```

```
        } else if (option.equals("C")) {
            ticketType = " 儿童票 ";
            totalPrice = ticketNum * 10;
        }
    } else {
        if (option.equals("A")) {
            ticketType = " 成人票 ";
            totalPrice = ticketNum * 45;
        } else if (option.equals("S")) {
            ticketType = " 学生票 ";
            totalPrice = ticketNum * 45 * 0.5;
        } else if (option.equals("O")) {
            ticketType = " 老人票 ";
            totalPrice = ticketNum * 45 * 0.6;
        } else if (option.equals("C")) {
            ticketType = " 儿童票 ";
            totalPrice = ticketNum * 15;
        }
    }
}
```

2.4.4 输出确认信息

采用字符串连接运算符 (+)，根据输出要求进行购买门票的数量、种类及应付总价的拼接并输出，代码如下：

```
System.out.println(" 您购买了 " + ticketNum + " 张 " + ticketType + ", 请支付购票金额: " + "\n" + totalPrice + " 元 ");
```

2.5 拓展任务

2.5.1 初级任务——阶梯水费

为了增强居民节水意识，减少水资源浪费，现在全国实施居民生活用水“阶梯水费”制度，居民生活用水阶梯水费如表 2-4 所示。请编写一个程序，根据月用水量 x ($\text{m}^3/\text{月}$) 计算居民本月生活用水的水费金额 y (元)，程序运行效果如图 2-5 所示。



任务讲解——
阶梯水费

表 2-4 居民生活用水阶梯水费

居民生活用水量		自来水销售价格 (元 / m^3)
第一阶梯	$30\text{m}^3/\text{月} \cdot \text{户}$ (含) 以下	1.88

续表

居民生活用水量		自来水销售价格 (元 /m ³)
第二阶梯	30m ³ ~ 38m ³ /月·户	2.82
第三阶梯	38m ³ /月·户以上	5.64

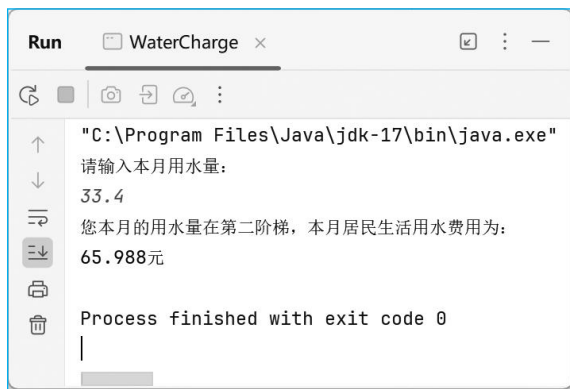


图 2-5 程序运行效果

思路提示

根据任务的功能需求, 设计程序结构。

- (1) 输入。使用键盘输入月用水量。同时, 注意输入的合法性判断, 用户输入的用水量应该为正数, 如果输入负数或非数字数据类型, 需要给出相应提示。
- (2) 编写选择结构。设计 if 语句的多分支结构, 计算三个阶梯的水费。
- (3) 输出。在控制台输出应缴水费。

2.5.2 进阶任务——体质(量)指数



任务讲解——
体质(量)指数

体重指数 (Body Mass Index, BMI), 又称体质(量)指数, 是用体重除以身高的平方得出的数字, 是目前国际上常用的衡量人体胖瘦程度及是否健康的一个标准。男性低于 20, 女性低于 19, 属于过轻; 男性 20 ~ 25, 女性 19 ~ 24, 属于适中; 男性 25 ~ 30, 女性 24 ~ 29, 属于过重; 男性 30 ~ 35, 女性 29 ~ 34, 属于肥胖; 男性高于 35, 女性高于 34, 属于非常肥胖。专家指出最理想的体重指数是 22, 并给出了健康建议, BMI 标准及相关健康建议如表 2-5 所示。按任务要求编写程序: 当输入性别、身高、体重后计算出体重指数, 并输出健康建议。

表 2-5 BMI 标准及相关健康建议

女	男	健康建议
BMI < 19	BMI < 20	XX 女士 / 先生, 您的体重过轻, 请加强膳食的补充!
19 ≤ BMI < 24	20 ≤ BMI < 25	XX 女士 / 先生, 恭喜您, 您的体重适中, 请保持!
24 ≤ BMI < 29	25 ≤ BMI < 30	XX 女士 / 先生, 您的体重过重, 请注意锻炼身体!
29 ≤ BMI < 34	30 ≤ BMI < 35	XX 女士 / 先生, 您已经属于肥胖行列, 请注意饮食习惯!

续表

女	男	健康建议
$BMI \geq 34$	$BMI \geq 35$	XX 女士 / 先生, 您已经非常肥胖, 必须下定决心减肥了!

思路提示

根据任务的功能需求, 设计程序结构。

(1) 输入。定义姓名、性别、身高、体重等变量, 并输入相应的值。体重单位为千克, 可以设计成整型; 身高单位为米, 可以设计为浮点型; 性别只有男或女两种选项, 可以设计成布尔型。注意输入的合法性判断, 若输入的身高、体重数据不是正数, 需进行必要的提示。

(2) 数据计算。根据身高和体重, 计算体质(量)指数 BMI。

(3) 定义嵌套分支结构。外层嵌套分为男和女两个分支, 内层嵌套根据 BMI 的值分为 5 个分支, 分别输出不同的健康建议。

2.5.3 挑战任务——成绩转换

编写程序, 分别使用 if-else 分支结构和 switch-case 分支结构实现成绩转换, 由百分制转换为五级制, 90 分及以上为优秀, 80 ~ 89 分为良好, 70 ~ 79 分为中等, 60 ~ 69 分为及格, 60 分以下为不及格。



任务讲解——
成绩转换

思路提示

根据任务的功能需求, 设计程序结构。

(1) 输入。使用键盘输入百分制成绩。输入的成绩为 0 到 100 的整数, 如输入不合法, 给出相应的提示。

(2) 处理。用 if-else 或 switch-case 语句编写 5 个分支结构分别对应不同成绩段, 并根据百分制成绩转换成不同五级制等级。

(3) 输出。输出转换后的五级制等级。