

软件开发类人才培养系列教材
“互联网+”新形态一体化教材

Java

语言程序设计实践教程

主编 李永刚 叶利华 龚迅炜

Java
YUYAN CHENGKU SHEJI
SHIJIAN
JIAOCHENG

本书
配套资源

- 微课视频
- 任务源码
- 教学课件



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

软件开发类人才培养系列教材
“互联网+”新形态一体化教材

Java

语言程序设计实践教程

主编 李永刚 叶利华 龚迅炜



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

内容提要

本书围绕 Java 语言知识体系，将内容分为 17 个单元，覆盖了 Java 语法基础、面向对象编程以及 Java 高级编程等知识点。本书的每个单元包括知识要点、设计性实验或验证性实验、拓展训练等模块，提供了丰富的典型案例，可操作性强，便于读者学习。本书可作为计算机相关专业的教学用书，也可作为相关技术人员培训或工作的参考用书。

图书在版编目 (CIP) 数据

Java 语言程序设计实践教程 / 李永刚, 叶利华, 龚
迅炜主编. — 上海: 上海交通大学出版社, 2024.2
ISBN 978-7-313-28070-1

I. ① J… II. ①李… ②叶… ③龚… III. ① JAVA 语
言—程序设计—高等学校—教材 IV. ① TP312
中国国家版本馆 CIP 数据核字 (2024) 第 030548 号

Java 语言程序设计实践教程
Java YUYAN CHENGXU SHEJI SHIJIAN JIAOCHENG

主 编: 李永刚 叶利华 龚迅炜	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: 15.5
字 数: 425 千字	
版 次: 2024 年 2 月第 1 版	印 次: 2024 年 2 月第 1 次印刷
书 号: ISBN 978-7-313-28070-1	
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



前言

Java 语言是面向对象编程语言的代表，Java 技术的应用领域十分广泛，长期占据着编程语言排行榜靠前的位置，学好 Java 这门编程语言十分重要。Java 程序设计是一门实践性很强的课程，学习者必须通过大量的编程练习培养实践编程能力。

2020 年 5 月，教育部印发了《高等学校课程思政建设指导纲要》，指出“把思想政治教育贯穿人才培养体系，全面推进高校课程思政建设，寓价值观引导于知识传授和能力培养之中”。2022 年 10 月，党的二十大报告指出“培养什么人、怎样培养人、为谁培养人是教育的根本问题”。作为教育工作者，我们应时刻把握这一根本问题，积极探寻新时代人才培养模式的创新之路，培养德智体美劳全面发展的社会主义建设者和接班人。本书围绕 Java 语言体系，将内容分为 17 个单元，将课程思政元素和党的二十大精神融于 17 个实践单元中，帮助读者在 Java 语言知识学习中树立正确的价值观。

本书作为浙江省课程思政示范基层组织的阶段成果之一，主要特色如下。

1. 德育元素与实践案例紧密结合

本书按照立德树人理念组织每个单元，从党的二十大精神中凝练出一批德育元素，并将这些德育元素融入实践内容，让学习者在学习实践案例的同时培养工匠精神、奋斗精神、实干精神和创新精神，在实践中树立职业道德，弘扬民族责任感。

2. 实践环节经过精心设计，体现知识点的层次和递进关系

本书的每个单元均包含知识要点、验证性实验或设计性实验和拓展训练等模块。单元的各个环节体现了知识点的层次和递进关系。知识要点总结了各个单元的重要知识点，便于读者快速掌握核心知识。验证性实验只需补充少量关键语句即可完成编程任务，可以锻炼读者的思维能力。设计性实验则需要完整地设计程序，可以锻炼读者解决问题的能力。题目后面的点拨、提示、注意等互动模块既能给读者解题提供提示，也拓展了与案例相关的知识。拓展训练模块则引导读者思考更深层次的问题。

3. 突出面向对象思想，深挖面向对象编程内涵

Java 语言是典型的面向对象编程语言，面向对象思想是 Java 语言的精髓。本书精心设计了大量的面向对象编程案例，突出面向对象编程的核心地位。在讲解一些高级应用时，如数据库编程、网络编程、多线程编程，本书也穿插了面向对象程序设计基本理念，有利于读者深挖面向对象编程的内涵。

4. 全面覆盖 Java 语言的知识点

本书知识覆盖面广，既包括了控制结构、数组、方法等 Java 编程基础知识内容，又涵盖了数据库编程、网络编程、集合与泛型、多线程编程、图形界面编程等高阶内容。

5. 配套资源丰富

本书的配套资源十分丰富，有在线视频、课程源码、教学课件等多种形式的配套资源。有需要者可致电 13810412048 或发邮件至 2393867076@qq.com 获取。

本书由李永刚、叶利华和龚迅炜主编并统稿。第 1 单元至第 5 单元、第 9 单元、第 10 单元和第 15 单元由李永刚执笔，第 6 单元至第 8 单元、第 11 单元由龚迅炜执笔，第 12 单元至 14 单元、第 16 单元和第 17 单元由叶利华执笔。魏远旺、王超超、桑高丽、俞侃参与了部分编写和整理工作，来自企业的张缪春、吴晓辉参与了部分编写工作并提出了修改意见。在此，一并向为本书编写做出贡献的老师表示衷心的感谢。

由于编者水平有限，书中存在的谬误之处，敬请读者批评指正。

编 者
2023 年 8 月



目录



第 1 单元 Java 语言概述和开发环境 / 1

知识要点.....1	验证性实验——在 Eclipse 环境下编译运行程序.....7
实验 使用 JDK 和 Eclipse 开发 Java 程序.....4	验证性实验——在 Eclipse 环境下调试运行程序.....8
验证性实验——JDK 的安装和设置.....4	拓展训练.....9
验证性实验——在命令提示符下编译运行程序.....7	



第 2 单元 基本程序设计 / 10

知识要点.....10	设计性实验——华氏摄氏温度转换.....14
实验 数据类型与运算符.....13	设计性实验——输出 24 个希腊字母.....15
验证性实验——求三位整数的各位数字之和.....13	设计性实验——计算矩形的面积.....15
验证性实验——古代重量单位转换.....14	设计性实验——N 体模拟粒子数.....15
	拓展训练.....16



第 3 单元 控制语句 / 17

知识要点.....17	实验 2 循环语句的使用.....26
实验 1 选择语句的使用.....21	验证性实验——猜数字大小游戏.....26
验证性实验——成绩等级判断.....21	验证性实验——求 n1~n2 范围内的素数.....28
验证性实验——回文数判断.....22	设计性实验——求两数之间的奇数和.....29
设计性实验——求解鸡兔同笼问题.....23	设计性实验——打印数字金字塔.....29
设计性实验——用 switch 语句实现成绩等级判断.....24	设计性实验——输出最大的数及其出现的次数.....30
设计性实验——猜数游戏.....25	设计性实验——求 π 的值.....30
设计性实验——求解一元二次方程.....25	拓展训练.....31
拓展训练.....26	



第 4 单元 方法 / 32

知识要点.....	32	设计性实验——打印乘法表	36
实验 方法的使用.....	33	设计性实验——求反素数	36
验证性实验——整数的倒置	33	设计性实验——查看移动通信技术的特点 和代表公司	36
验证性实验——判断闰年	34	拓展训练	37
设计性实验——求不同类型数的立方值	35		



第 5 单元 数组 / 38

知识要点.....	38	设计性实验——用二维数组模拟批改成绩	44
实验 数组的使用.....	40	设计性实验——消除重复值	45
验证性实验——数组为参数的方法重载	40	设计性实验——奇偶 0、1 矩阵	46
验证性实验——冒泡排序	42	设计性实验——查找连续 N 个相等的数	46
设计性实验——竞赛评分	43	拓展训练	47
设计性实验——增强 for 循环求和.....	44		



第 6 单元 对象和类 / 48

知识要点.....	48	实验 2 变量的作用域和 this 关键字	68
实验 1 对象的创建和使用.....	54	验证性实验——三角形类 Triangle.....	69
验证性实验——打印机类 Printer	54	验证性实验——学生类 Student.....	72
验证性实验——平面坐标系中的点类 Point.....	57	验证性实验——单例工具类 Scanner	74
设计性实验——盒子模型类 CSSBox	59	设计性实验——加法器类 Calculator.....	75
设计性实验——秒表类 StopWatch.....	66	设计性实验——产品类 Product.....	77
设计性实验——利用 Math 类中的成员方法 计算一元二次方程的根	67	设计性实验——茶叶类 Tea.....	77
拓展训练	68	拓展训练	80



第 7 单元 继承与多态 / 81

知识要点.....	81	设计性实验——LinkedList 类.....	94
实验 1 子类的派生与方法覆盖.....	82	拓展训练	95
验证性实验——冷兵器类 ColdWeapon	82	实验 2 多态与动态绑定.....	95
验证性实验——在 Printer 类的基础上定义 扫描打印一体机类	86	验证性实验——物流运输方式类	96
验证性实验——在 Timer 类的基础上定义时钟 Clock 类和手表 Watch 类.....	90	验证性实验——特殊物流运输类	98
设计性实验——教务管理系统中多种人员 对应不同功能	93	验证性实验——歌词中的多态	99
设计性实验——交通卡类	93	设计性实验——驾驶交通工具类	100
		设计性实验——键盘类	101
		设计性实验——交通灯类 TrafficLight.....	102
		拓展训练	102



第 8 单元 抽象类和接口 / 103

知识要点.....	103	验证性实验——交通工具接口实现	115
实验 1 抽象类的定义和使用.....	107	验证性实验——动物接口的多继承设计	115
验证性实验——银行信用卡贷款还款	107	设计性实验——多种职业为人民服务	118
验证性实验——几何图形抽象类	109	设计性实验——多部门在不同的场景下开展 销售工作	118
设计性实验——实现中华民族伟大复兴	112	设计性实验——美食盛宴	119
设计性实验——学校教育评价	112	拓展训练	119
拓展训练	113		
实验 2 接口的定义和使用.....	113		
验证性实验——燃放鞭炮驱赶野兽	113		



第 9 单元 异常处理 / 120

知识要点.....	120	设计性实验——字符串转换为数值	124
实验 异常处理的实现.....	122	设计性实验——多 catch 语句应用.....	125
验证性实验——处理输入异常	122	设计性实验——统计导弹中的 bug	126
验证性实验——自定义异常类	123	拓展训练	127
设计性实验——判断能否构成三角形	124		



第 10 单元 字符串应用 / 128

知识要点.....	128	验证性实验——StringBuffer 字符串应用.....	132
实验 字符串的应用.....	129	设计性实验——单词元音和长度统计.....	133
验证性实验——采用多种方式创建		设计性实验——单词出现次数统计.....	134
String 对象.....	130	设计性实验——字符数量统计.....	134
验证性实验——单词替换.....	131	拓展训练.....	135



第 11 单元 Java 输入与输出 / 136

知识要点.....	136	实验 2 Java 二进制 I/O 应用.....	145
实验 1 Java 文本 I/O 应用.....	137	验证性实验——文件的加密和解密.....	145
验证性实验——诗歌的行文转换.....	137	验证性实验——购物车对象的序列化	
验证性实验——上三角矩阵筛查.....	139	和反序列化.....	147
验证性实验——字符类型统计.....	141	验证性实验——文件的拷贝.....	149
设计性实验——获取 Web 页面中图片标签		设计性实验——保存指定 URL 页面上的图片	
的 src 属性值.....	143	到本地文件夹.....	150
设计性实验——读取指定文件夹中的文件		设计性实验——编写一个生成指定文件的	
列表.....	144	十六进制编码的文件生成器.....	151
设计性实验——日历文件生成工具		设计性实验——为文件增加校验字段.....	152
MyCalendarTool.....	144	拓展训练.....	152
拓展训练.....	145		



第 12 单元 泛型与集合 / 153

知识要点.....	153	设计性实验——泛型类的设计.....	158
实验 使用泛型与集合开发 Java 程序.....	156	设计性实验——Comparable 接口应用.....	159
验证性实验——泛型方法和泛型类应用.....	156	设计性实验——HashMap 在字符串统计中	
验证性实验——通配符在泛型中的应用.....	157	的应用.....	159
设计性实验——用 foreach 和迭代器		拓展训练.....	160
遍历元素.....	158		



第 13 单元 图形界面基础 / 161

知识要点.....	161	设计性实验——布局管理器应用	167
实验 使用 Java 图形用户容器和控件		设计性实验——菜单设计	168
开发程序.....	163	设计性实验——登录界面设计	168
验证性实验——多边形组件绘图	163	设计性实验——简易计算器界面	169
验证性实验——绘制椭圆和圆弧	164	设计性实验——文本编辑器	169
验证性实验——文本绘制	165	拓展训练	170
设计性实验——五子棋棋盘绘制	166		



第 14 单元 图形界面高级应用 / 171

知识要点.....	171	设计性实验——时钟设计	179
实验 图形界面高级应用.....	172	设计性实验——ScaleTransition 应用	179
验证性实验——拼图游戏	172	设计性实验——贪吃蛇游戏	180
验证性实验——PathTransition 应用	176	设计性实验——MediaPlayer 媒体播放	181
验证性实验——RotateTransition 应用	178	拓展训练	181



第 15 单元 数据库编程 / 182

知识要点.....	182	实验 2 图形界面数据库操作.....	194
实验 1 数据库基本操作.....	187	验证性实验——简单图形界面数据库应用	194
验证性实验——简单的数据库查询应用	187	验证性实验——简单的登录和注册界面	196
验证性实验——独立方法的数据库连接	188	验证性实验——汽车管理系统界面	200
设计性实验——使用预编译语句连接		设计性实验——数据连接界面	206
数据库	190	设计性实验——学校信息管理系统的查询	
设计性实验——分数查询	191	和删除	206
设计性实验——修改数据表记录	191	设计性实验——学生档案管理系统	207
设计性实验——简单的货物销售系统	192	拓展训练	210
拓展训练	194		



第 16 单元 多线程编程 / 211

知识要点	211	设计性实验——Thread 类应用	219
实验 使用多线程开发 Java 程序	213	设计性实验——Runnable 接口应用	220
验证性实验——线程优先级	213	设计性实验——wait 和 notify 应用	221
验证性实验——多线程同步	214	拓展训练	221
验证性实验——线程池应用	217		



第 17 单元 网络编程 / 222

知识要点	222	设计性实验——多线程服务端程序	233
实验 Java 网络编程	226	设计性实验——基于 TCP 服务器的	
验证性实验——URL 和 URLConnection		多客户端聊天群	233
类的应用	226	设计性实验——基于 UDP 服务器的	
验证性实验——Netty 框架的应用	227	多客户端聊天群	234
设计性实验——简单客户端和服务端		拓展训练	234
通信程序	232		



参考文献 / 235

第 1 单元

Java 语言概述 和开发环境

单元导读 >

我国软件行业在数字经济时代加速转型升级，企业数字化转型也给软件行业发展带来了巨大机遇。Java 语言一直占据编程语言排名靠前的位置，是世界上最受欢迎的编程语言之一，因此，学好 Java 语言将十分受益。本单元介绍 Java 语言开发环境的搭建以及 JDK 路径的设置，通过本单元的学习，读者能够在命令提示符下编译和运行程序，能够在 Eclipse 集成开发环境下编译、运行和调试程序。

知识要点

1. JDK 介绍

JDK (Java development kit) 是 Java 语言的软件开发工具包 (SDK)。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK。JDK 是整个 Java 的核心，包括 Java 运行环境 (Java runtime environment, JRE)、Java 工具和 Java 基础的类库。JDK 提供了非常实用的功能，其版本也不断更新，运行效率得到了非常大的提高。本书使用 JDK 14.0.1 版本进行演示，建议读者使用 JDK 8 以上的版本。

2. Eclipse 开发工具介绍

Eclipse 是一个基于 Java 的开放源码的可扩展的应用开发平台，它为编程人员提供了一流的 Java 集成开发环境 (integrated development environment, IDE)。在 Eclipse 的官方网站中提供了 Eclipse IDE 的下载链接 (<https://www.eclipse.org/downloads/>)。

Eclipse 分为安装版和解压版。解压版只需要把 Eclipse 压缩包解压到本地即可运行。本书采用 eclipse-java-win32-x86_64.zip 版本的压缩文件。解压后单击  eclipse.exe 图标即可运行。在弹出的设置工作空间的对话框中，指定工作空间位置，如 “D:\JavaCourse\eclipse-workspace”。在每次启动 Eclipse 时，都会弹出设置工作空间的对话框，如果想在以后启动时不再进行工作空间的设置，可以选中 “Use this as the default and do not ask again” 多选框，如图 1-1 所示。Eclipse 的启动界面如图 1-2 所示。其他版本的 Eclipse 操作相同。

若是初次进入 Eclipse 软件，Eclipse 会弹出欢迎页面 (见图 1-3)，否则直接进入 Eclipse 的工作台 (见图 1-4)。如果出现欢迎界面，则关闭该欢迎界面即可进入 Eclipse 的工作台。Eclipse 的工作台主要由菜单栏、工具栏、项目资源管理器视图、大纲视图、编辑器、其他视图等组成。

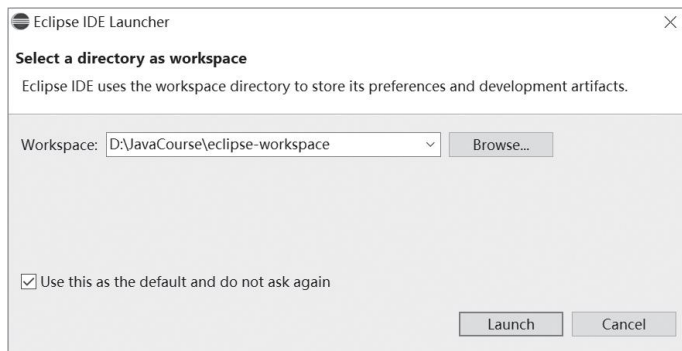


图 1-1 Eclipse 的工作空间设置



图 1-2 Eclipse 的启动界面

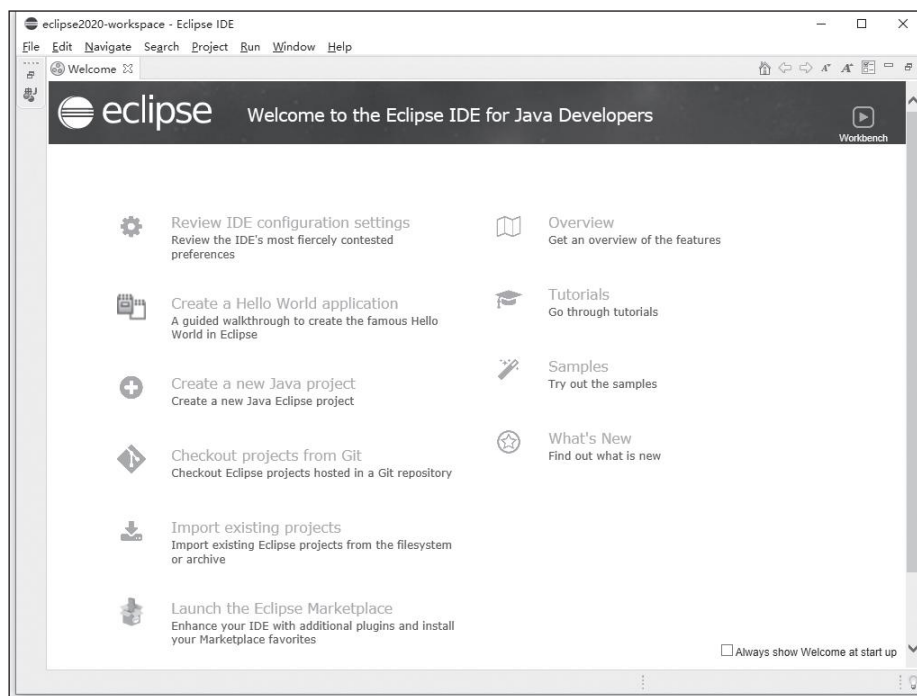


图 1-3 Eclipse 的欢迎页面



图 1-4 Eclipse 的工作台

在 Eclipse 工作台的上方提供了菜单栏，该菜单栏包含了实现 Eclipse 各项功能的命令，并且菜单栏中的菜单项与当前编辑器内打开的文件是相关联的。Eclipse 的菜单栏中共包括 10 个菜单，这些菜单中又包含了相应的子菜单。Eclipse 中常用的菜单如图 1-5 所示。

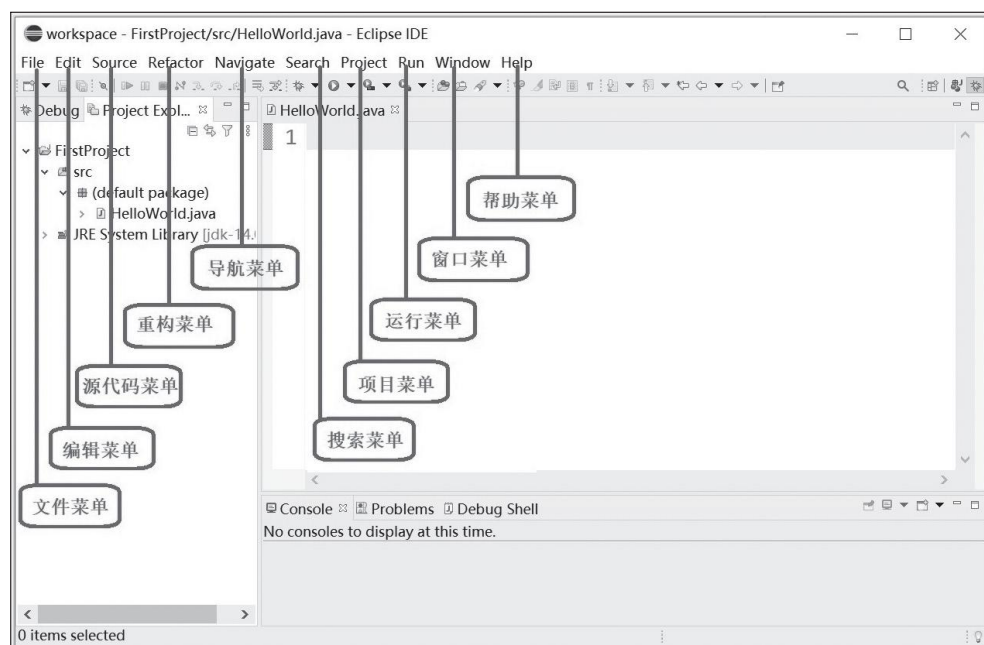


图 1-5 Eclipse 常用的菜单

Eclipse 的功能十分强大，掌握 Eclipse 相关的快捷键能够大大提高开发效率。Eclipse 提供的常用快捷键如表 1-1 所示。

表 1-1 Eclipse 的快捷键

快捷键	说明
Ctrl + /	注释或取消注释
Ctrl + Shift + /	代码块注释
Ctrl + Shift + \	取消代码块注释
Ctrl + D	删除光标所在行的代码
Ctrl + O	打开视图的小窗口
Ctrl + W	关闭单个窗口
Ctrl + 鼠标单击	跟踪方法和类的源码
Ctrl + 鼠标停留	显示方法和类的源码
Ctrl + M	将当前视图最大化
Ctrl + I	将光标停留在某个变量上，按“Ctrl+I”组合键可提供快速实现的重构方法；选中若干行，按“Ctrl+I”组合键可将此段代码放入 for、while、if、do 或 try 等代码块中
Ctrl + Q	回到最后编辑的位置
Ctrl + F6	切换窗口
Ctrl + K	将光标停留在变量上，按“Ctrl+K”组合键可查找下一个同样的变量
Ctrl + Shift + K	和“Ctrl+K”组合键查找的方向相反

续表

快捷 键	说 明
Ctrl + Shift + F	代码格式化。如果选择部分代码，仅对所选代码进行格式化
Ctrl + Shift + O	快速导入类的路径
Ctrl + Shift + X	将选中字符转为大写
Ctrl + Shift + Y	将选中字符转为小写
Ctrl + Shift + M	导入未引用的包
Ctrl + Shift + D	在 debug 模式下显示变量值
Ctrl + Shift + T	查找工程中的类
Ctrl + Alt + Down	复制光标所在行至其下一行
双击左括号（小括号、中括号、大括号）	选择括号内的所有内容

实验 使用 JDK 和 Eclipse 开发 Java 程序

知识目标

了解 Java 开发的运行环境，掌握简单 Java 程序的编写和调试方法。

能力目标

学会设置 JDK 路径，能够在命令提示符和 Eclipse 下编译和运行程序。

素质目标

养成调试、运行程序解决问题的习惯，树立科技报国的远大理想。



验证性实验——JDK 的安装和设置

- （1）从 Oracle 官方网站（<https://www.oracle.com/>）下载 Windows 平台下的 JDK 安装包，本书选用 JDK 14.0.1，读者也可以选择其他版本。
- （2）双击 JDK 安装包进入 JDK 的安装界面，如图 1-6 所示。
- （3）单击“下一步”按钮后，可以自定义安装路径。单击“更改”按钮，将安装路径设置为“C:\Program Files\Java\jdk-14.0.1\”，如图 1-7 所示。



图 1-6 进入安装界面



图 1-7 设置安装路径

(4) 单击“下一步”按钮开始安装 JDK，如图 1-8 所示。

(5) 等到安装完成，软件会弹出显示安装完成的对话框，此时完成了 JDK 14.0.1 的安装，如图 1-9 所示。



图 1-8 开始安装 JDK



图 1-9 JDK 14.0.1 安装完成

(6) 安装好 JDK 14.0.1 后，可以打开安装的目录，JDK 安装好后生成的文件如图 1-10 所示。

如果想在系统的任意目录下编译和运行编写好的 Java 程序，需要先设置环境变量，请按以下步骤配置（JDK 的默认安装路径为 C:\Program Files\Java\jdk-14.0.1）。



图 1-10 JDK 安装好后生成的文件

(7) 鼠标右键单击“此电脑”图标，选择“属性”选项，弹出的界面如图 1-11 所示，单击“高级系统设置”，弹出的“系统属性”对话框如图 1-12 所示。



图 1-11 系统信息界面



图 1-12 “系统属性”对话框

(8) 单击“环境变量”按钮弹出“环境变量”对话框，在“系统变量”中选择 Path，单击“编辑”按钮（见图 1-13），在弹出的变量值文本框中添加“C:\Program Files\Java\jdk-14.0.1\bin”，单击“确定”按钮完成环境变量的设置，如图 1-14 所示。

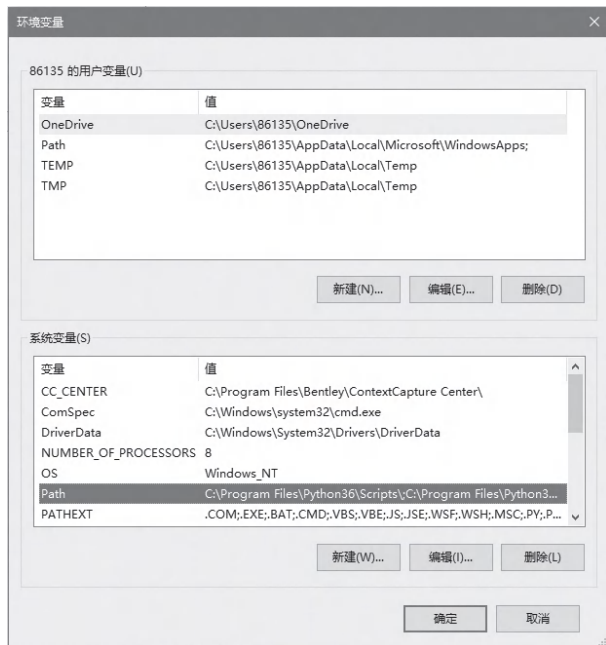


图 1-13 Windows 系统“环境变量”对话框

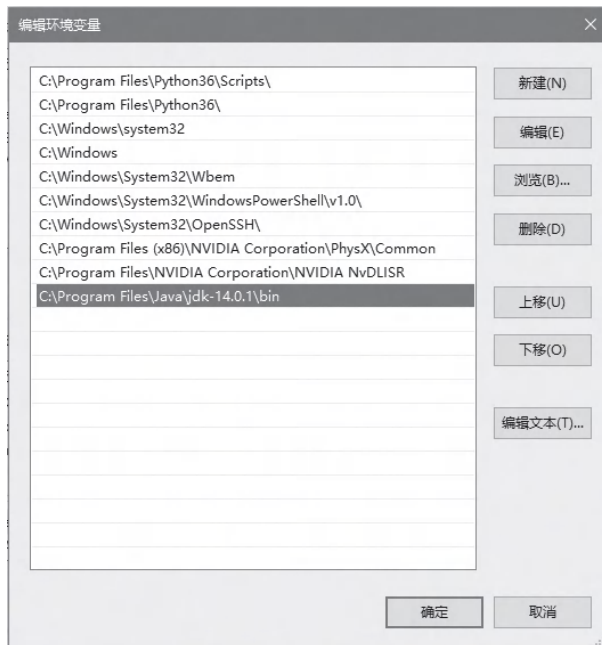


图 1-14 编辑环境变量 Path

(9) 环境变量设置完毕后，打开“开始”菜单中的 Windows 系统，找到“命令提示符”图标，如图 1-15 所示。单击  图标，打开命令提示符窗口，在窗口中输入“javac”，按“Enter”键检查环境变量设置是否成功，如果设置成功则可以看到如图 1-16 所示的结果。



图 1-15 Windows 系统的“命令提示符”

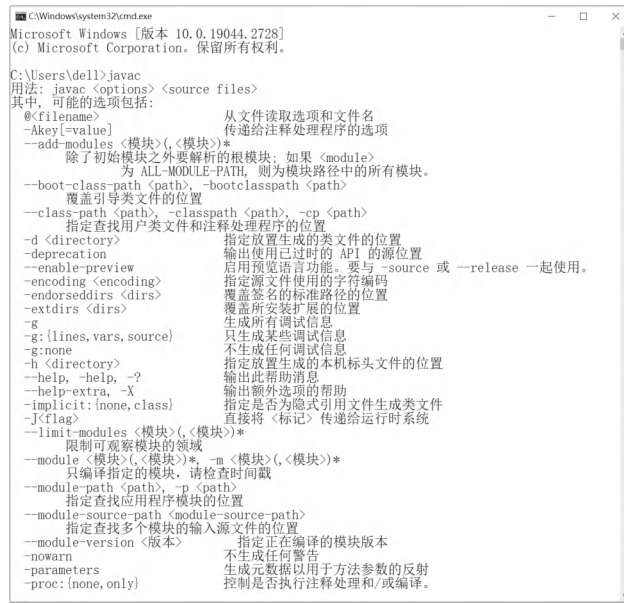


图 1-16 验证环境变量是否设置成功

在之后需要进入命令提示符直接编译和运行 Java 程序时，只需进入 Java 程序所在的目录，通过“javac”和“java”命令即可编译和执行相应的程序。

验证性实验——在命令提示符下编译运行程序

(1) 在安装完 JDK 并设置好 JDK 路径后, 使用任意文本编辑器编写程序 HelloJava.java (代码见参考程序 1.1), 编写完成后将文件保存, 并命名为“HelloJava.java”, 如图 1-17 所示。

(2) 在命令提示符环境下使用“cd 文件名”命令进入下一级目录(可以通过“cd ..”返回上一级目录), 然后输入“javac HelloJava.java”并按“Enter”键开始编译程序, 如图 1-18 所示。

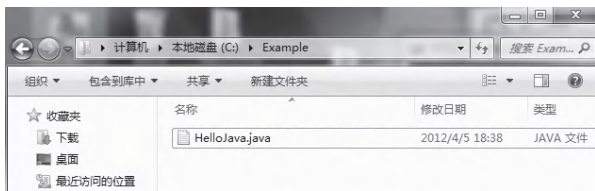


图 1-17 HelloJava.java 文件

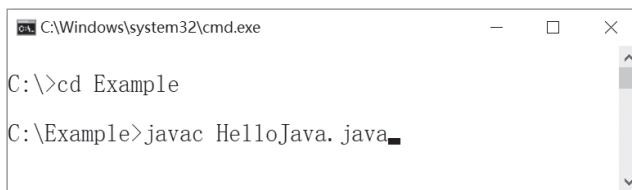


图 1-18 编译程序 HelloJava.java

(3) 编译完成后, 打开 Example 文件夹(此为本书放置程序的目录, 读者可自行选择), 发现系统自动生成了 HelloJava.class 文件, 如图 1-19 所示。

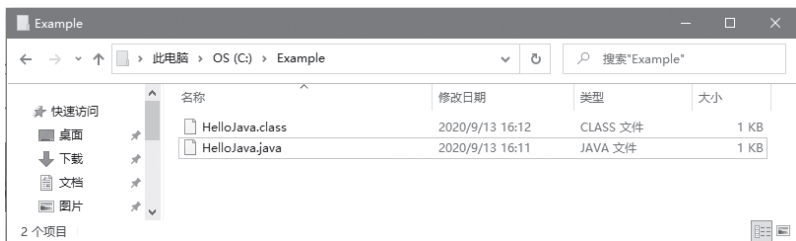


图 1-19 生成的 HelloJava.class 文件

(4) 在命令提示符中继续输入“java HelloJava”命令, HelloJava 的输出结果如图 1-20 所示。

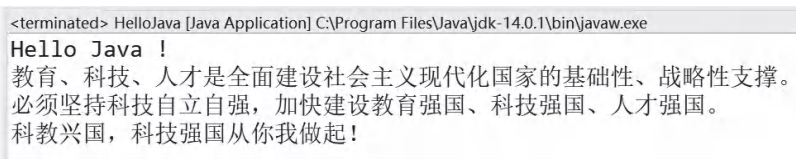


图 1-20 Hello Java 的输出结果

【参考程序 1.1】

```
public class HelloJava {
    public static void main(String[] args) {
        System.out.println("Hello Java !");
        System.out.println("教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑。");
        System.out.println("必须坚持科技自立自强，加快建设教育强国、科技强国、人才强国。");
        System.out.println("科教兴国，科技强国从你我做起！");
    }
}
```

验证性实验——在 Eclipse 环境下编译运行程序

(1) 新建一个“Project”文件夹, 然后选择“Java Project”, 如图 1-21 所示。

(2) 指定工程名“Project name”和存储位置“location”, 并使用默认的 JRE, 单击“Finish”按钮完成创建, 如图 1-22 所示。

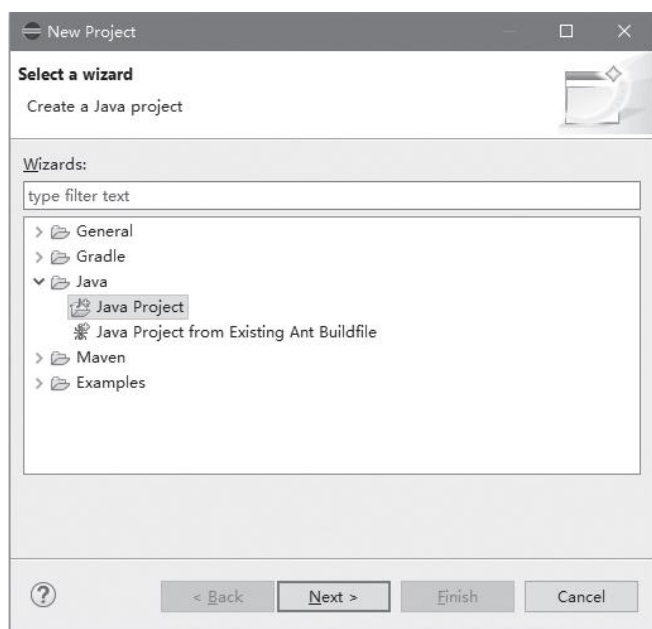


图 1-21 新建一个 Java Project

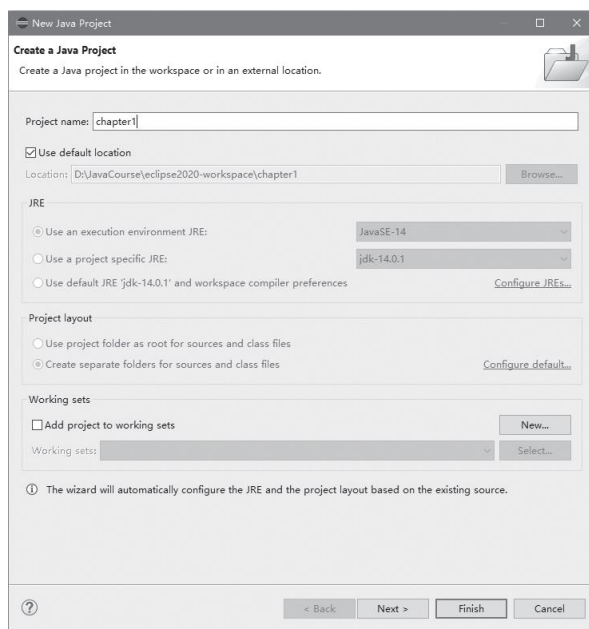


图 1-22 指定 Project 名称和存储位置

(3) 展开新建的工程名，选中其中的源程序目录“src”，右击，选择“New”选项新建一个 Class 文件  **Class**，输入类名“FirstJava”，单击“Finish”按钮，如图 1-23 所示。

(4) 编写程序并运行，在 FirstJava.java 文件中输入如图 1-24 所示的代码，单击按钮工具栏上的按钮  或者按“Ctrl+F11”组合键开始运行，运行结果如图 1-24 所示。

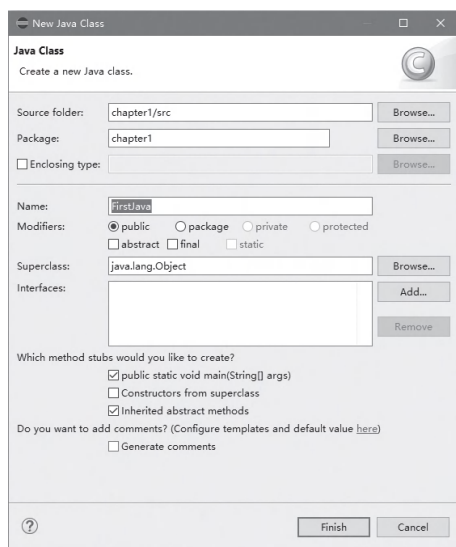


图 1-23 新建一个 Java 类

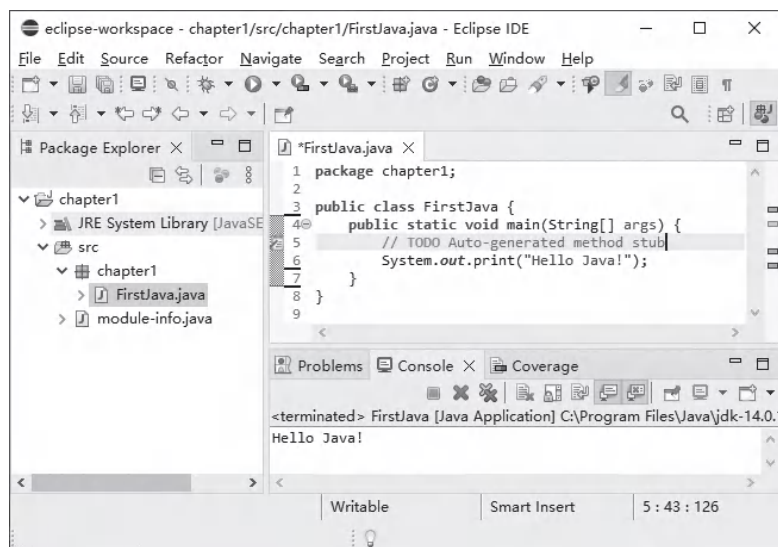


图 1-24 FirstJava.java 的代码和运行结果

验证性实验——在 Eclipse 环境下调试运行程序

在程序开发过程中读者会不断体会到程序调试的重要性。使用 Java 调试器可以设置程序的断点，让程序单步执行，在调试过程中还可以查看变量和表达式的值。通过调试可以避免在程序中编写大量的 System.out.println 方法输出调试信息。

(1) 设置断点。设置断点是程序调试中必不可少的手段。Java 调试器每次遇到程序断点时都会将当前线程挂起，即暂停当前程序的运行。可以在 Java 编辑器中显示代码行号的位置双击来添加或删除

当前行的断点，或者在当前行的位置右击，在弹出的快捷菜单中选择“Toggle Breakpoint”选项实现断点的添加与删除，如图 1-25 所示。

(2) 以调试方式运行 Java 程序。要在 Eclipse 中调试程序，可以在 Eclipse 中程序文件的空白位置右击，在弹出的快捷菜单中执行“Debug As”→“Java Application”命令。调试器将在断点处挂起当前线程，使程序暂停，如图 1-26 所示。

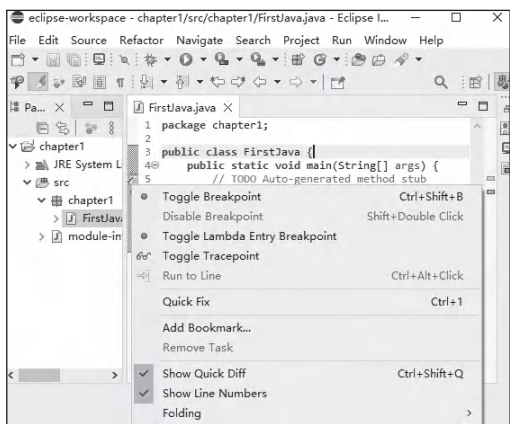


图 1-25 向 Java 编辑器中添加断点

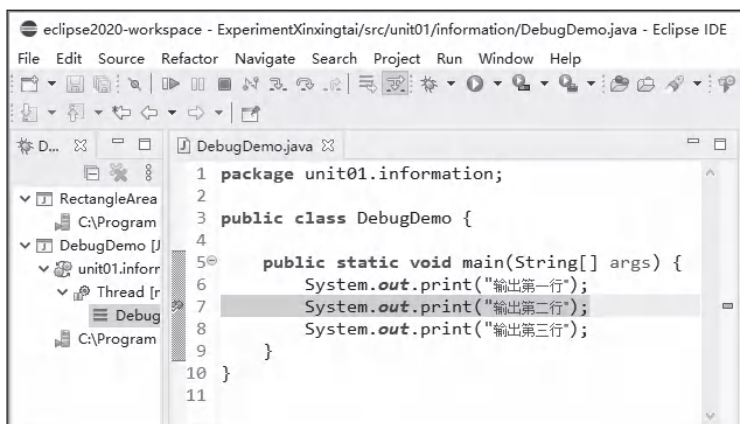


图 1-26 程序执行到断点后暂停

(3) 程序调试。程序执行到断点被暂停后，可以通过“Debug”（调试）视图工具栏上的按钮执行相应的调试操作，如“运行”、“停止”等。“Debug”（调试）视图如图 1-26 所示。在“Debug”（调试）视图的工具栏中单击按钮或按“F6”键，将执行单步跳过操作，即运行单独的一行程序代码，但是不进入调用方法的内部，然后跳到下一个可执行点并暂时挂起线程。在“Debug”（调试）视图的工具栏中单击按钮或按“F5”键，将跳入调用方法或对象的内部单步执行程序并暂时挂起线程。

拓展训练

(1) 分析在未设置 Path 环境变量前 Eclipse 软件不能正常运行 Java 程序的原因，并思考如何设置 Path 环境变量以及设置环境变量后有什么效果，进而思考 Path 环境变量的用途。

(2) 理解编译命令 javac 和执行命令 java 的用法。“javac HelloJava.java”和“javac helloJava.java”有什么区别？“java helloJava”和“java HelloJava”有什么区别？

(3) 命令提示符是 Windows 兼容 DOS (disk operating system, 磁盘操作系统) 的命令符输入窗口，可以通过它调用系统服务，实现管理文件、执行程序、获取系统信息等操作，从而更好地了解和使用计算机。请尝试在命令提示符下实现上述操作。

第 2 单元

基本程序设计

单元导读

对于一个程序员来说，编程语言的基础知识就是内功，只有内功深厚的人，行走江湖才能立于不败之地。将基础知识重新组合，就能变化出众多的奇妙程序。本单元练习利用基本数据类型、运算符优先级、变量、常量、表达式以及标准输入 / 输出流等知识实现基本的程序设计。

知识要点

1. Java 的数据类型

Java 语言的数据类型包括基本数据类型和引用数据类型两大类。基本数据类型包括 byte、short、int、long、float、double、char 和 boolean 8 种。引用类型（reference type）包括类、接口和数组。Java 的基本数据类型如表 2-1 所示。

表 2-1 Java 的基本数据类型

类型名	所占内存 / 字节	取值范围	说 明
byte	1	-128~127	8 位带符号数
short	2	-32 768~32 767	16 位带符号数
int	4	-2 147 483 648~2 147 483 647	32 位带符号数
long	8	$-2^{63} \sim 2^{63} - 1$	64 位带符号数
float	4	$[-3.402\ 823\ 5\text{E}3\ 8, -1.4\text{E}-45] \cup [1.4\text{E}-45, 3.402\ 823\ 5\text{E}3\ 8]$	32 位带符号数
double	8	$[-1.797\ 693\ 134\ 862\ 315\ 7\text{E}3\ 08, -4.9\text{E}-324] \cup [4.9\text{E}-324, 1.797\ 693\ 134\ 862\ 315\ 7\text{E}3\ 08]$	64 位带符号数
char	2	0~65 535	Unicode 编码可以保存世界上文字的常用符号，注意 char 类型和数字类型间是可以转换的。汉字范围：19 968~40 869
boolean	1	true / false	逻辑判断

2. Java 的运算符优先级

Java 的运算符包括算术运算符、一元运算符、关系运算符、逻辑运算符、赋值运算符等，每种运算符都有自己的优先级和结合性。Java 的运算符优先级及结合性如表 2-2 所示，其中序号越小，优先级越高。

例如：

```
int x = 1;
int y = 2;
boolean z = x * y == x + y;
```

在上述表达式中，* 的优先级最高，其次是 +，然后是关系运算符 ==，而赋值运算符 = 的优先级是最低的，因此上面的表达式也可以修改为：

```
boolean z = ((x * y) == (x + y));
```

表达式修改后运行结果是不变的。

表 2-2 Java 的运算符优先级及结合性

序号	类别	运算符	结合性
1	后置运算符	a++、a--	右结合
2	一元运算符、前置运算符	+, -, ++a, --a	右结合
3	转换运算符	(type)Casting	右结合
4	算术运算符	*, /, %	左结合
5		+, -	左结合
6	关系运算符	<, <=, >, >=	左结合
7		==, !=	左结合
8	逻辑运算符	!	右结合
9		&	左结合
10		^	左结合
11			左结合
12		&&	左结合
13			左结合
14	赋值运算符	=, +=, -=, *=, /=, %=	右结合

在结合方向上，除赋值运算符外，所有的双目运算符都是左结合的，如“a - b - c - d”等价于“((a - b) - c) - d”。

赋值运算符是右结合的，“a = b += c = 7”等价于“a = (b += (c = 7))”。

表达式的计算遵循如下规则。

规则 1：可能的情况下，从左向右依次计算所有的表达式。

规则 2：根据运算符的优先级进行运算。

规则 3：对优先级相同的相邻运算符，根据结合方向进行运算。

3. Java 的标准输入 / 输出流

java.lang.System 类提供了 3 种标准流：标准输入流（System.in）、标准输出流（System.out）和标准错误流（System.err）。通过它们可以实现数据的输入和输出操作。

JDK 5.0 之后的版本增加了一个新的类——java.util.Scanner，结合 System.in 可以实现对指定数据

的输入。通过“new Scanner(System.in)”创建一个 Scanner 对象，控制台就会一直等待输入。例如：

```
Scanner scan = new Scanner(System.in); // 创建一个 Scanner 对象 scan，System.in 表示标准化输入，也就是键盘
                                         输入
int i = scan.nextInt(); // 利用 Scanner 对象 scan 获取从键盘输入的整数
double y = scan.nextDouble(); // 利用 Scanner 对象 scan 获取从键盘输入的双精度浮点数
String str = scan.nextLine(); // 利用 Scanner 对象 scan 获取从键盘输入的字符串
```

此外，Scanner 类还提供了 nextByte()、nextShort()、nextLong()、nextFloat()、next() 等方法来实现不同类型数据的输入。

4. 格式化输出

System.out 的 print() 和 println() 方法可以向控制台输出不同类型的数据，可以满足程序调试信息的输出要求。其中 print() 表示输出后不换行，println() 表示输出后换行。在实际应用中，输出数据时，要求数据必须按照一定的格式输出，如小数点后保留 2 位有效数字、按照规定的格式输出日期、按照表格方式输出数据等。

System.out 的 printf() 方法可以对数据进行格式化输出，其常用的格式说明符如表 2-3 所示。

表 2-3 printf 方法常用格式说明符

格式化符号	含义
%c	单个字符
%d	十进制整数
%f	十进制浮点数
%o	八进制数
%s	字符串
%u	无符号十进制数
%x	十六进制数
%%	百分号 %
%b	逻辑值 (true 或 false)

printf 的常用格式控制：%m.n^① 格式字符。

% 表示格式说明的起始符号，不可缺少。m 代表域宽，即对应的输出项在输出设备上所占的字符数。n 代表精度，用于说明输出的实型数的小数位数。未指定 n 时，隐含的精度为 6 位。在 %.2f 中，m 位是默认位数，小数保留 2 位；%9.2f 的意思是位数为 9 位，小数保留 2 位（四舍五入）；%09.2f 的意思是位数为 9 位，小数保留 2 位，位数不足的用 0 补齐。

例如：

```
double y = 15.3673;
System.out.printf("y is:%.2f", y);
```

输出结果为：y is:15.37

① 为与代码保持一致，本书中部分与代码相关的变量采用正体。

实验 数据类型与运算符

知识目标

掌握 Java 语言基本数据类型的使用方法；掌握从键盘输入数据的方法；了解 Java 的表达式求值方法及操作符的优先级；掌握 Java 的格式化输出方法。

能力目标

能够使用基本数据类型和运算符编写 Java 应用程序，并实现交互式输入 / 输出。

素质目标

树立不怕困难、勇于探索创新的品格；树立尊重规则、遵守法律的意识。



基本程序设计

验证性实验——求三位整数的各位数字之和

从键盘输入一个三位整数，输出该数的各位数字的和。要求采用标准输入 / 输出流。

【参考程序 2.1】

```
import java.util.Scanner;
public class ThreeDigSum {
    public static void main(String[] args) {
        int n,a,b,c;
        int digsum = 0;
        Scanner scanner;
        System.out.print(" 请输入一个整数: ");
        scanner = _____; // 实例化一个 Scanner 对象;
        n = _____; // 获得一个整型数值
        a = n%10; // 个位
        b = (n%100)/10; // 十位
        c = _____; // 百位
        digsum = a+b+c; // 数字和
        System.out.print(" 整数 "+n+" 的各位数字之和是: "+digsum);
    }
}
```

运行结果如图 2-1 所示。

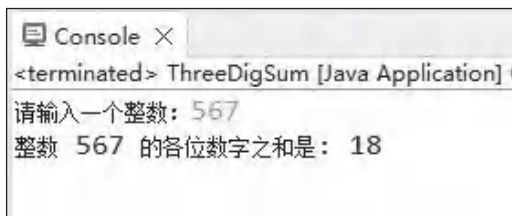


图 2-1 三位整数的各位数字之和程序的运行结果

点拨

首先利用 Scanner 类创建一个 Scanner 对象 scanner，利用 Scanner 对象的 nextInt() 方法从键盘获取输入的整数。然后利用运算符“%”和“/”的性质，分别获得三位数整数的个位、十位和百位上的数字。三个数字之和即为最后结果。

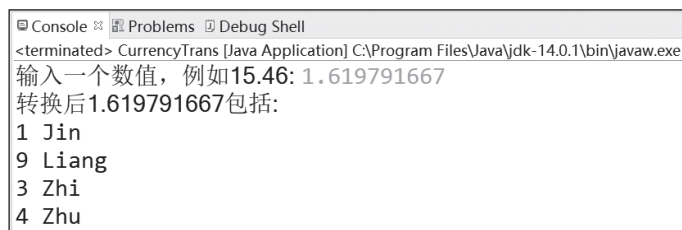
验证性实验——古代重量单位转换

中国古代的重量单位是比较复杂的，即使秦始皇统一中国后统一了度量衡制度，促进了重量单位的统一，但重量单位的换算依然比较复杂。重量单位换算规则如下：6 铢等于 1 镒，4 镒等于 1 两，16 两等于 1 斤。编写程序，输入一个较大值的重量，将其转换为较小的重量单位。

【参考程序 2.2】

```
import java.util.Scanner;
public class AncientWeightTrans {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print(" 输入一个数值，例如 15.46: ");
        double amount = input.nextDouble();
        int remainAmount = (int)(amount * 384);
        int numberOfJin = remainAmount/384;
        remainAmount = remainAmount % 384;
        int numberOfLiang = remainAmount / 24;
        remainAmount = remainAmount % 24;
        int numberOfZhi = _____;
        remainAmount = _____;
        int numberOfZhu = remainAmount;
        System.out.println(" 转换后 " + amount + " 包括 :");
        System.out.println(numberOfJin + " Jin");
        System.out.println(numberOfLiang + " Liang ");
        System.out.println(numberOfZhi + " Zhi");
        System.out.println(numberOfZhu + " Zhu");
    }
}
```

运行结果如图 2-2 所示。



```
<terminated> CurrencyTrans [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
输入一个数值，例如15.46: 1.619791667
转换后1.619791667包括:
1 Jin
9 Liang
3 Zhi
4 Zhu
```

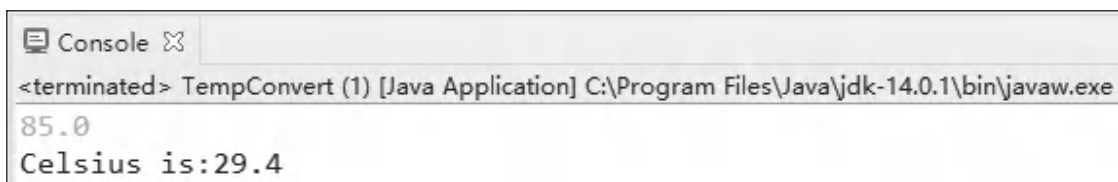
图 2-2 古代重量单位转换程序的运行结果

点拨

首先利用 Scanner 类创建一个 Scanner 对象 input，利用 Scanner 对象的 nextDouble() 方法从键盘获取输入的浮点数。然后把浮点数转化为整数。再利用较小重量单位的数值特点及运算符“%”和“/”的性质，分别获得重量单位转换后的结果。

设计性实验——华氏摄氏温度转换

从键盘输入一个数值作为华氏温度，将其转换为摄氏温度，并保留 2 位小数，转换公式为摄氏温度 = $(5/9) * (\text{华氏度} - 32)$ 。运行结果如图 2-3 所示。



```

Console
<terminated> TempConvert (1) [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
85.0
Celsius is:29.4

```

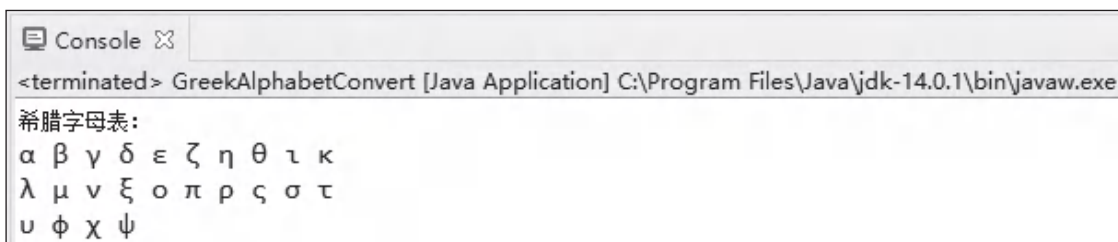
图 2-3 华氏摄氏温度转换程序的运行结果

点拨

利用 System.out 的 printf() 方法对数据进行格式化输出，保留 2 位小数。

设计性实验——输出 24 个希腊字母

每行输出 10 个字母后换行。运行结果如图 2-4 所示。



```

Console
<terminated> GreekAlphabetConvert [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
希腊字母表:
α β γ δ ε ζ η θ ι κ
λ μ ν ξ ο π ρ σ τ
υ φ χ ψ

```

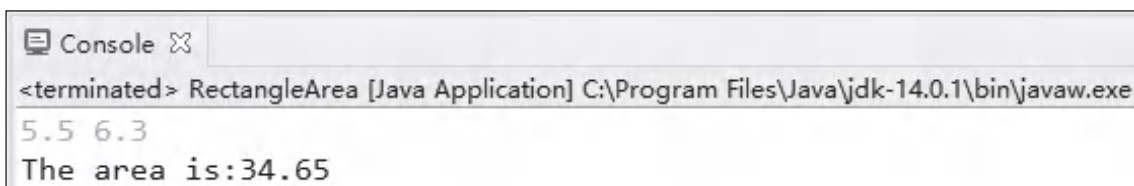
图 2-4 输出希腊字母程序的运行结果

点拨

在 Unicode 字符表中，希腊字母是从 945 开始的 24 个字符，可以使用 char 进行强制类型转换。

设计性实验——计算矩形的面积

从键盘分别输入一个矩形的长和宽，其类型为 double 类型，输出矩形的面积，并保留 2 位小数。运行结果如图 2-5 所示。



```

Console
<terminated> RectangleArea [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
5.5 6.3
The area is:34.65

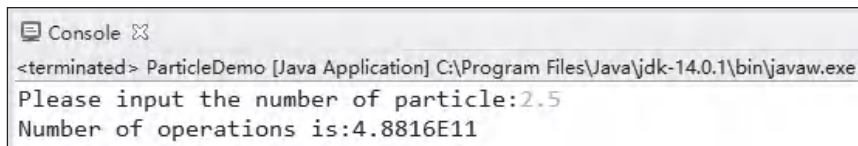
```

图 2-5 计算矩形面积程序的运行结果

设计性实验——N 体模拟粒子数

“天河二号”是由国防科技大学研制的超级计算机系统，以每秒 5.49 亿亿次的峰值计算速度和每秒 3.39 亿亿次双精度浮点运算的持续计算速度位居榜首，成为 2013 年全球最快超级计算机。 N 体模拟是现代天体物理学中探索粒子在引力相互作用下动力学问题的重要方法。2015 年 5 月，“天河二号”成功进行了 3 万亿粒子数中微子和暗物质的宇宙学 N 体数值模拟，揭示了宇宙大爆炸 1600 万年之后至今约 137 亿年的漫长演化进程，同时这也是当时天文学 N 体模拟粒子数的世界纪录。

编写 N 体模拟粒子数程序，假设在每秒 3.39 亿亿次双精度浮点运算的“天河二号”上运行，提示输入粒子数（亿粒），计算每小时的模拟次数。运行示例如图 2-6 所示。



```
<terminated> ParticleDemo [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
Please input the number of particle:2.5
Number of operations is:4.8816E11
```

图 2-6 N 体模拟粒子数程序的运行示例

点拨

超级计算机的浮点运算能力与模拟次数成正比，输入的粒子数与模拟次数成反比。

拓展训练

（1）数据类型的转换分为自动转换和强制转换。自动转换是指数字表示范围小的数据类型会自动转换成范围大的数据类型。试分析哪些自动转换不会造成数据丢失，哪些转换可能会出现数据丢失。

（2）在同一个表达式中使用自增操作或自减操作，多次修改同一个变量会产生副作用吗？比如“`int x = ++i+i;`”“`int y = j+++j;`”这两条语句执行后，分别会得到什么结果？

第 3 单元

控制语句

单元导读 >

Java 语言中的基本控制语句是指程序中用来控制程序流程的语句。这些基本控制语句包括顺序语句、选择语句和循环语句。复杂程序都可以使用上述三种基本控制语句实现，实现的过程就是逐步求精的过程。解决遇到的实际问题亦需要不断迭代优化，将党的二十大精神融入工程实践中，并在实践过程中培养精益求精的工匠精神。本单元重点练习选择语句和循环语句的使用方法和技巧，通过本单元的学习读者可以掌握用 Java 语言流程控制语句编写程序的能力，可以了解 break、continue 的使用方法。

知识要点

1. 选择语句

选择语句有单分支 if 语句、双分支 if-else 语句、多分支 if-else 语句和 switch 语句。

(1) 单分支 if 语句。单分支 if 语句的语法如下：

```
if(布尔表达式) {  
    // 布尔表达式的值为真时执行的语句；  
}
```

单分支语句只处理布尔表达式的值为真的情况。单分支 if 语句的流程图如图 3-1 所示。

(2) 双分支 if-else 语句。双分支语句根据条件的真假决定执行的路径。双分支 if-else 语句的语法如下：

```
if(布尔表达式) {  
    // 布尔表达式的值为真时执行的语句；  
} else {  
    // 布尔表达式的值为假时执行的语句；  
}
```

双分支 if-else 语句的流程图如图 3-2 所示。

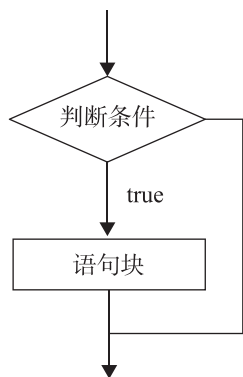


图 3-1 单分支 if 语句流程图

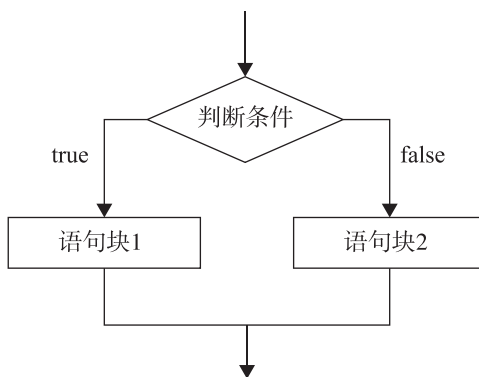


图 3-2 双分支 if-else 语句流程图

(3) 多分支 if-else 语句。多分支 if-else 语句根据条件的真假决定各分支执行的路径。多分支 if-else 语句的语法如下：

```
if(布尔表达式 1) {
    // 布尔表达式 1 的值为真时执行的语句；
} else if(布尔表达式 2) {
    // 布尔表达式 2 的值为真时执行的语句；
}
.....
else {
    // 上述布尔表达式的值均为假时执行的语句；
}
```

多分支 if-else 语句的流程图如图 3-3 所示。

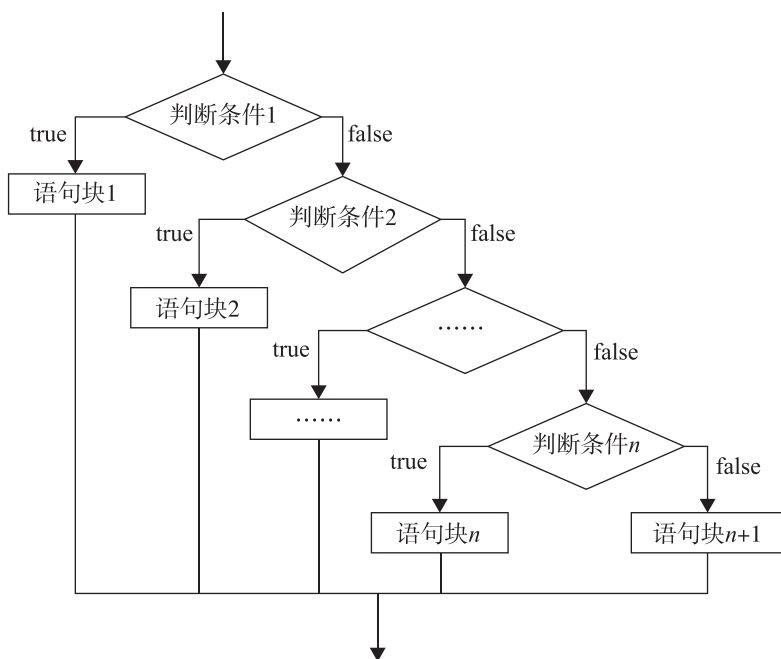


图 3-3 多分支 if-else 语句流程图

(4) switch 语句。switch 语句基于变量或者表达式的值执行分支语句。switch 语句的语法如下：

```

switch( switch 表达式 ) {
    case 值 1: // 值为 1 时执行的语句;
        break;
    case 值 2: // 值为 2 时执行的语句;
        break;
    .....
    case 值 n: // 值为 n 时执行的语句;
        break;
    default: // 默认执行的语句;
}

```

switch 语句的流程图如图 3-4 所示。

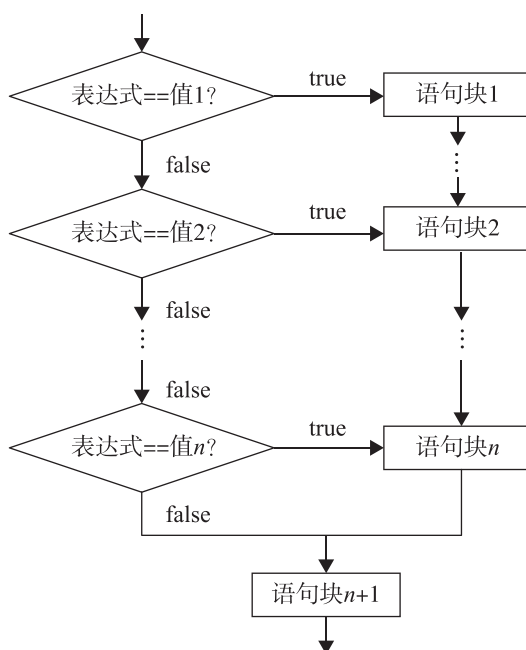


图 3-4 switch 语句流程图

switch 语句执行遵循如下规则：

- ① switch 表达式必须能计算出一个 char、byte、short、int 等可枚举类型或者 String 类型的值，且必须用小括号括起来；
- ② case 后面的值与 switch 表达式的值类型一致，且都是常量或常量表达式，不能有变量；
- ③ switch 语句执行时，从与 case 语句相匹配的值开始，直到遇到第一个 break 或者到达 switch 的结束位置；
- ④ default 是可选的，当没有一个表达式的值与 case 后面的值相匹配时执行该语句。

2. 循环语句

Java 提供三种类型的循环语句：while 循环、do-while 循环和 for 循环。

(1) while 循环。while 循环的语法如下：

```

while( 循环条件 ) {
    // 循环体;
    // 语句(s);
}

```

(2) do-while 循环。do-while 循环能够确保循环体至少执行一次，其语法如下：

```
do {
    // 循环体；
    语句 (s);
} while( 循环条件 );
```

(3) for 循环。for 循环的语法如下：

```
for( 初始化操作 ; 循环条件 ; 每次循环后的操作 ) {
    // 循环体；
    语句 (s);
}
```

for 循环的初始操作可以是由 0 个或多个逗号隔开的语句，循环后的操作可以是由 0 个或多个逗号隔开的语句，例如：

```
for (int k = 1; k < 100; System.out.println(k++));
```

或

```
for (int i = 0, j = 0; (i + j < 10); i++, j++) {
    // 循环体；
}
```

三种循环语句的流程如图 3-5 所示。

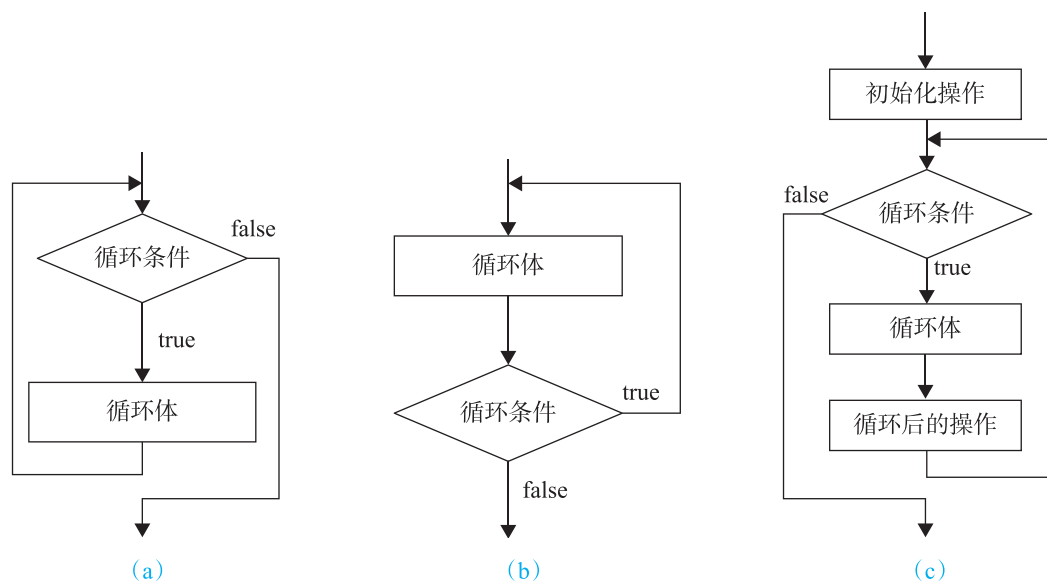


图 3-5 三种循环语句流程图

(a) while 循环；(b) do-while 循环；(c) for 循环

选用循环结构时，建议遵循以下原则：

- ①如果重复次数已知，用 for 循环；
- ②如果不知道重复次数，用 while 循环；
- ③如果在检验循环条件前需要执行循环体，就用 do-while 循环代替 while 循环。

在 Java 中，有三种可以实现跳转功能的语句：break、continue、return。

break 含义：终止本次条件判断，从 switch 语句中退出或终止当前循环，从当前循环中退出。

continue 含义：无条件地使控制语句转移到循环语句的条件判断部分，终止本次循环，直接进入下一次循环。

return 含义：既可以返回方法的结果，也可以直接跳出循环。

实验 1 选择语句的使用

知识目标

了解 Java 语言中的 if 语句、if-else 语句、嵌套 if-else 多分支语句和 switch 多分支语句的特性和使用方法。

能力目标

具有使用 Java 语言流程控制语句编写程序的能力。

素质目标

培养创新精神，养成突破陈规、敢于创新的素质。



选择语句的
使用

验证性实验——成绩等级判断

设计一个成绩等级判断程序，要求从键盘输入学生的 Java 成绩，判断该学生的成绩等级，成绩级别划分如下：成绩大于等于 90 为 A 级；成绩在 80 到 89 之间为 B 级；成绩在 70 到 79 之间为 C 级；成绩在 60 到 69 之间为 D 级；成绩小于 60 为 E 级。

【参考程序 3.1】

```
import java.util.Scanner;
public class JavaMultiGrade {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        char grade;
        double score;
        score = scanner.nextDouble();
        if(score>100 || score<0) {           // 判断输入的成绩是否合法
            System.out.println("Illegal input, please reenter");
            System.exit(0);
        }
        // 利用多分支 if-else 语句输出学生的成绩等级
        if(score>=90) {                       // 根据成绩，判断等级为 A
            grade='A';
        } else if(score>=80 && score<90) {    // 根据成绩，判断等级为 B
            grade='B';
        } else if(score>=70 && score<80) {
            grade='C';
        } else if(score>=60 && score<70) {
            grade='D';
        } else {
            grade='E';
        }
        System.out.println("成绩等级为：" + grade);
    }
}
```

```

        } _____ (score>=70&&score<80) {    // 根据成绩，判断等级为 C
            grade='C';
        } _____ (score>=60&&score<70) {    // 根据成绩，判断等级为 D
            grade='D';
        } else {                                // 其他成绩，判断等级为 E
            grade='E';
        }
        System.out.println("Grade is:"+grade);
    }
}

```

成绩等级判断程序的运行结果如图 3-6 所示。

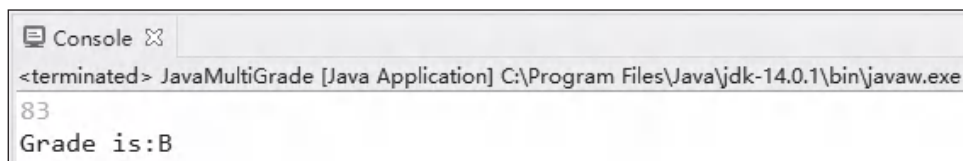


图 3-6 成绩等级判断程序的运行结果

点拨

首先利用 Scanner 类创建一个 Scanner 对象，利用 Scanner 对象的 nextDouble() 方法从键盘获取一个输入的浮点数 score。如果 score 大于 100 或者小 0，提示重新输入。利用多分支 if-else 语句将百分制的 score 转为等级制的 grade，最后输出 grade 的结果。

验证性实验——回文数判断

回文是指正读反读都一样的句子，如“天连碧水碧连天”“寒宫对月对宫寒”等。在数学中也有一类数有这样的特征，这种数被为回文数。假设 n 是一个任意自然数，若 n 的各位数字反向排列所得自然数 m 与 n 相等，则称 n 为回文数。例如，若 $n=123454321$ ，则 n 为回文数；但若 $n=12345$ ，则 n 不是回文数。从键盘输入一个 1~65 535 之间的数，判断这个数是几位数，并判断这个数是否为回文数。

【参考程序 3.2】

```

public class PalindromeDemo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println(" 输入 1~65535 之间的整数 ");
        int number = input.nextInt();
        int n1,n2,n3,n4,n5;
        if(number<=65535 && number>=1) {
            n1 = number % 10;
            n2 = number % 100 / 10;
            n3 = _____;
            n4 = _____;
            n5 = number / 10000;
            if(n5>0) { //number 是 5 位数
                System.out.println(number + " 是 5 位数 ");
            }
        }
    }
}

```



```

if( ____ ) // 判断是否为回文数
    System.out.println(number + " 是回文数 ");
else
    System.out.println(number + " 不是回文数 ");
} else if(n4>0) { //number 是 4 位数
    System.out.println(number + " 是 4 位数 ");
    if( ____ ) // 判断是否为回文数
        System.out.println(number + " 是回文数 ");
    else
        System.out.println(number + " 不是回文数 ");
} else if(n3>0) { //number 是 3 位数
    System.out.println(number + " 是 3 位数 ");
    if(n3==n1)
        System.out.println(number + " 是回文数 ");
    else
        System.out.println(number + " 不是回文数 ");
} else if(n2>0) { //number 是 2 位数
    System.out.println(number + " 是 2 位数 ");
    if(n2==n1)
        System.out.println(number + " 是回文数 ");
    else
        System.out.println(number + " 不是回文数 ");
} else { //number 是 1 位数
    System.out.println(number + " 是 1 位数 ");
    System.out.println(number + " 是回文数 ");
}
} else {
    System.out.println(number + " 不在 1~65535 之间 ");
}
}
}

```

回文数判断程序的运行结果如图 3-7 所示。

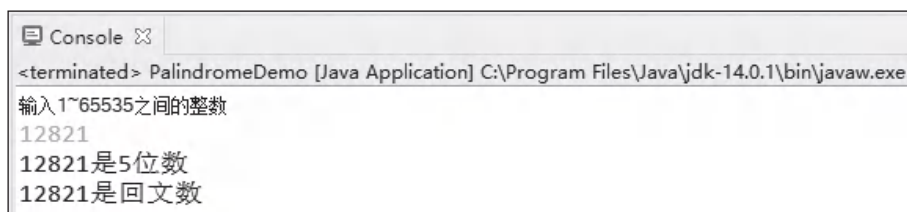


图 3-7 回文数判断程序的运行结果

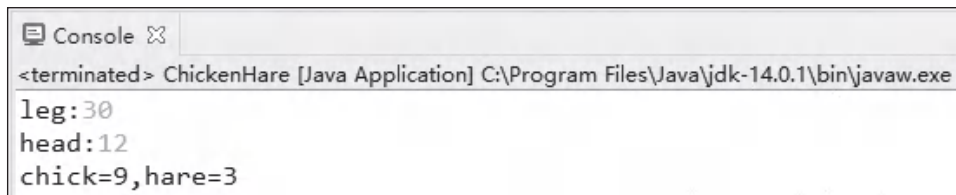
点拨

- (1) 两个整数相除，结果也为整数，如 123/100 的结果为 1，而不是 1.23。
- (2) 为了计算出 12821 百位上的数字 8，可以先计算 12821%1000 得到 821，然后再计算 821/100 得到数字 8。

设计性实验——求解鸡兔同笼问题

《孙子算经》是中国古代重要的数学著作，成书于南北朝时期，其中记载了一个有趣的问题：鸡和

兔在同一个笼子里，鸡和兔共有 n 条腿， m 个头，问鸡和兔各有多少只？请设计一个程序求解决问题，输入总腿数和总头数，可以得出鸡的只数和兔的只数。程序运行结果如图 3-8 所示。



```

Console
<terminated> ChickenHare [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
leg:30
head:12
chick=9,hare=3

```

图 3-8 鸡兔同笼问题程序的运行结果

点拨

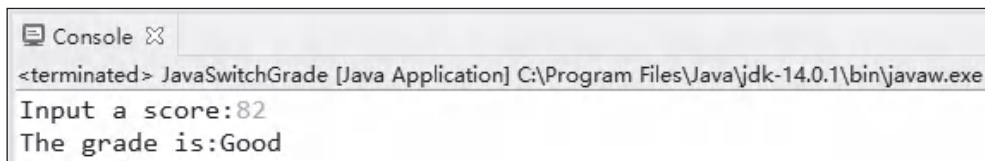
《孙子算经》给出的解题方案是上置头，下置足，半其足，以头除足，以足除头。这种求解方法的公式为：

鸡的只数 = (兔的腿数 × 总只数 - 总腿数) ÷ (兔的腿数 - 鸡的腿数)。

求完鸡的只数后，兔的只数 = 总只数 - 鸡的只数。

设计性实验——用 switch 语句实现成绩等级判断

用 switch 语句实现 Java 课程成绩等级判断，同样要求从键盘输入学生的 Java 课程成绩，判断该学生的成绩等级，大于等于 90 为 Very good，80 至 89 为 Good，70 至 79 为 Medium，60 至 69 为 Pass，小于 60 为 Not passed。程序运行结果如图 3-9 所示。



```

Console
<terminated> JavaSwitchGrade [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
Input a score:82
The grade is:Good

```

图 3-9 用 switch 语句实现课程成绩等级判断程序的运行结果

点拨

首先利用 Scanner 类创建一个 Scanner 对象，利用 Scanner 对象的 nextDouble 方法从键盘获取用户输入的一个实数 score。将 score 转换为 [0,10] 之间的整数。转换后小于 6 的数均可设为 switch 语句的 default 分支。

小技巧

在 case 分支中，不能采用 case 1~3 的方式使用连续的数值，可以采用 case 1:case 2:case 3: 的方法。例如，将百分制的分数转换为 [0,10] 的范围后，把 10 和 9 均转换为 “Very good”，可以使用如下方式：

```
int g= (int)score/10;
```

```
case 10:case 9:grade="Very good";
```

而小于 6 的数，均转化为 “Not passed”，可以使用如下方式：

```
default:grade="Not passed";
```

设计性实验——猜数游戏

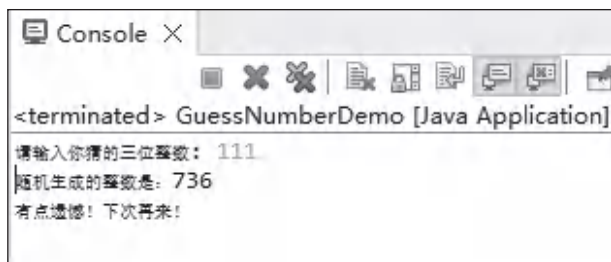
设计一个三位数的猜数游戏，随机生成一个三位数，程序提示用户输入一个三位数，依照以下的规则决定赢取多少奖金。

- (1) 如果用户输入的数字和随机数字完全一致，输出“恭喜恭喜！完全猜对了！获得三个赞！”。
- (2) 如果用户输入的数字覆盖了随机生成的所有数字（不论顺序），输出“输入的数字覆盖了随机生成的所有数字！很棒，获得两个赞！”。
- (3) 如果用户输入的数字匹配了1个或2个数字，输出“输入的数字匹配了1个或2个数字！还不错，获得一个赞！”。
- (4) 如果用户输入的数字在随机数中一个都没有，输出“有点遗憾！下次再来！”。

程序运行结果如图3-10(a)和3-10(b)所示。



(a)



(b)

图 3-10 猜数游戏程序的运行结果

▲点拨

先由程序随机生成一个三位的整数 n ，再利用 Scanner 对象的 nextInt() 方法从键盘获取输入的一个三位数 m 。声明变量 $n1$ 、 $n2$ 、 $n3$ 分别保存各数位的值，声明变量 $m1$ 、 $m2$ 、 $m3$ 分别保存输入用户猜想的数字各数位的值。比较 m 和 n 的值或比较各数位的值，输出比较结果。

▲小技巧

可以利用 java.util.Random 类通过实例化一个 Random 对象创建一个随机数生成器。例如生成一个 $[100,200)$ 区间的整数的方法如下：

```
import java.util.Random;
Random random=new Random();
int n = random.nextInt(100)+100;
```

设计性实验——求解一元二次方程

怎样求一元二次方程 $ax^2+bx+c=0(a \neq 0)$ 在实数域上的解（实根）？

根的求解公式：

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

判断式 $\Delta = b^2 - 4ac$ ，先判断 Δ ，若 $\Delta < 0$ ，则方程无实根；若 $\Delta = 0$ ，则方程有两个相同的解，为

$$x = -\frac{b}{2a} ; \text{ 若 } \Delta > 0, \text{ 则 } x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}。$$

编写程序，提示输入 a 、 b 和 c 的值，并输出结果。如果 Δ 为正，打印两个根；如果 Δ 为 0，打印 1 个根；否则，输出 “No real roots”。

提示：输入 a 、 b 和 c 的值后，先计算判别式 Δ 的值，根据判别式的值设计选择语句结构。程序运行结果如图 3-11 (a)、图 3-11 (b)、图 3-11 (c) 所示。

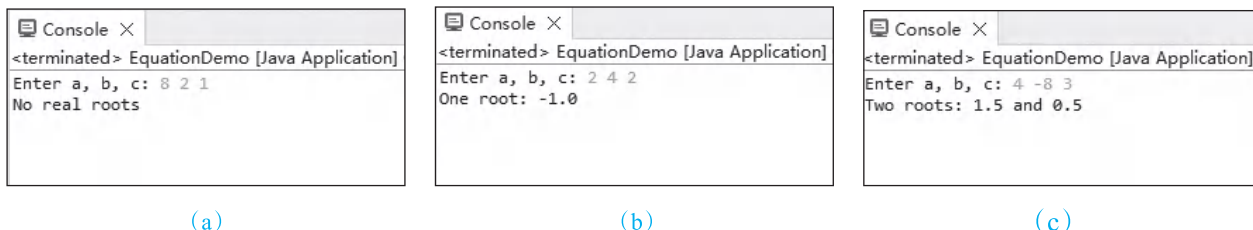


图 3-11 求解一元二次方程程序的运行结果

点拨

求解平方根可以使用 `java.lang.Math` 类的 `sqrt(double a)` 方法。该方法会返回正确舍入的 `double` 值的正平方根。

拓展训练

- (1) 探索在 `switch` 语句中，`case` 语句后不加 `break` 语句会导致什么结果。
- (2) 比较多分支 `if-else` 语句和 `switch` 语句的异同。
- (3) 下面语句的输出结果是什么？若有错误，请指出。

```
int a=2,b=1;
if(a>b) {
    System.out.print("a");
}else{
    System.out.print("b");
}
```

实验 2 循环语句的使用

知识目标

熟练掌握使用 `for`、`while` 和 `do-while` 语句来控制循环；掌握 `break` 和 `continue` 的使用方法。

能力目标

具有使用 Java 语言循环语句编写程序的能力。

素质目标

弘扬奋斗精神，传承中华优秀传统文化。



循环语句的
使用

验证性实验——猜数字大小游戏

编制一个程序，用循环控制结构实现简单的猜数字大小游戏，实现如下任务：

- (1) 由程序随机分配一个 1~100 之间的整数；
- (2) 用户输入自己猜测的整数；

(3) 程序返回提示信息, 提示信息分别是“猜大了”“猜小了”和“猜对了”;

(4) 用户可根据提示信息再次输入猜测数值, 直到提示信息是“猜对了”。

【参考程序 3.3】

```
import java.util.Scanner;
public class GuessNumber {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println(" 给你一个 1 至 100 之间的整数, 请猜测这个数 ");
        int realNumber = (int) (Math.random() * 100) + 1;
        int myGuess = 0;
        System.out.println(" 输入您的猜测: ");
        myGuess = in.nextInt();
        while ( ) { // 循环条件
            if ( ) { // 条件代码
                System.out.println(" 猜大了, 再输入你的猜测: ");
            } else {
                System.out.println(" 猜小了, 再输入你的猜测: ");
            }
            myGuess = in.nextInt();
        }
        System.out.println(" 恭喜你猜对了! ");
    }
}
```

程序运行结果如图 3-12 所示。

```

Console
<terminated> GuessNumber [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
给你一个1至100之间的整数,请猜测这个数
输入您的猜测:
8
猜小了,再输入你的猜测:
11
猜小了,再输入你的猜测:
15
猜小了,再输入你的猜测:
50
猜小了,再输入你的猜测:
88
猜大了,再输入你的猜测:
75
猜大了,再输入你的猜测:
67
猜小了,再输入你的猜测:
69
猜小了,再输入你的猜测:
71
猜大了,再输入你的猜测:
70
恭喜你猜对了!
  
```

图 3-12 猜数字游戏程序的运行结果

 点拨

首先利用 Math 类的 random() 方法获得一个 [0,1) 之间的随机数, 并转换为 [1,100] 之间的整数 realNumber, 再从键盘获取输入的一个整数 myGuess。利用 while 循环判断输入的整数 myGuess 与随机生成数 realNumber 的比较结果, 如果 myGuess 不等于 realNumber, 则一直循环, 直到猜对结果为止。如果 myGuess 大于 realNumber, 提示“猜大了, 再输入你的猜测”; 如果 myGuess 小于 realNumber, 提示“猜小了, 再输入你的猜测”; 如果猜出正确结果, 提示“恭喜你猜对了!”。

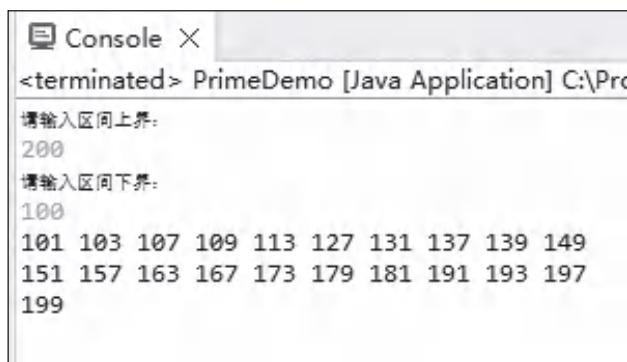
验证性实验——求 $n_1 \sim n_2$ 范围内的素数

素数指的是大于 1 的整数中, 只能被 1 和它本身整除的数。判断一个整数 m 是否为素数, 只需判断在 $[2, m-1]$ 之间是否存在能将 m 整除的整数, 如果都不能整除, 那么 m 就是一个素数。进一步思考, 判断 m 能否被 $[2, m/2]$ 区间的整数整除也可以判断出 m 是否为素数。

【参考程序 3.4】

```
import java.util.Scanner;
public class PrimeDemo {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("请输入区间上界: ");
        int n1 = in.nextInt();
        System.out.println("请输入区间下界: ");
        int n2 = in.nextInt();
        int count = 0;
        for (int m = n2; m < n1; m++) {
            // 设置逻辑值 flag, 初始化为 true
            boolean flag = true;
            for (int n = 2; n <= m / 2; n++) {
                if (m % n == 0) { // 若 m 能被 n 整除, 则当前数不是素数
                    // 当前数不是素数, 则 flag 的值更改为 false
                    flag = false;
                    break; // 跳出当前循环
                }
            }
            // 根据 flag 的结果判断 m 是否是素数
            if (flag) {
                count++;
                System.out.print(m + " ");
                if (count % 10 == 0)
                    System.out.println();
            }
        }
    }
}
```

程序运行结果如图 3-13 所示。

图 3-13 求 $n_1 \sim n_2$ 范围内素数程序的运行结果

小技巧

如果一个数不是素数且不等于1，那么它的最小质因数小于等于它的平方根。因此，只需判断 m 能否被 $[2, \sqrt{m}]$ 区间内的整数整除即可。

例如整数 77， $\sqrt{77} \approx 8.77$ ，而 77 能被 $[2, 8.77]$ 之间的整数 7 整除，因此 77 不是素数。根据此思路，上述程序可以怎么改进？

设计性实验——求两数之间的奇数和

任意输入两个整数，求两数之间（含两端）的奇数和。程序运行结果如图 3-14 所示。

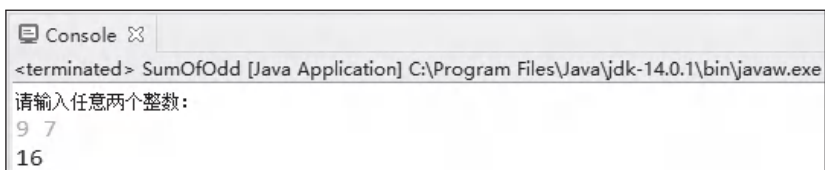


图 3-14 求两数之间奇数和程序的运行结果

点拨

从键盘输入 2 个整数 num_1 和 num_2 。如果 num_1 大于 num_2 ，交换 num_1 和 num_2 的值。求 num_1 和 num_2 两数之间（含两端）的奇数和。

设计性实验——打印数字金字塔

输入数字金字塔图案的行数，打印如图 3-15 所示的数字金字塔。

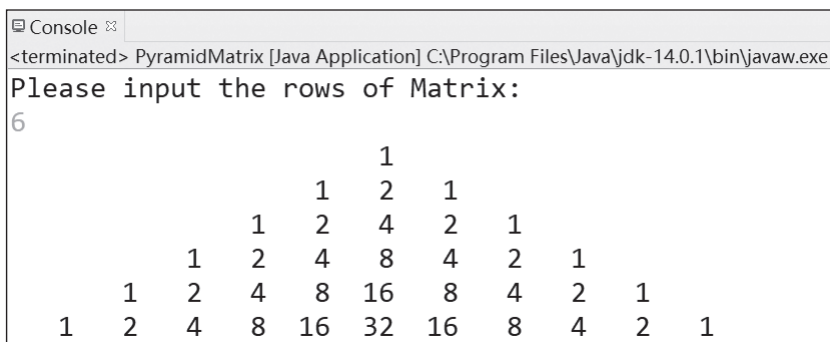


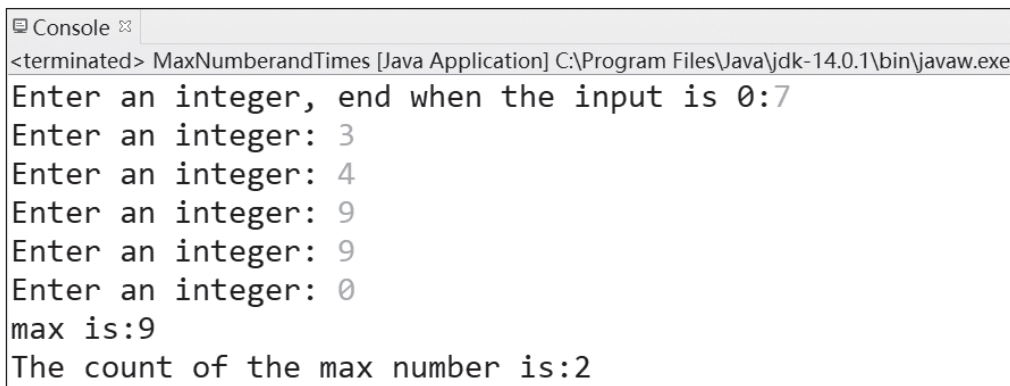
图 3-15 数字金字塔程序的运行结果

点拨

每一行先打印空格；然后打印中心线及左边的数字，右边的数字是左边的 2 倍；再打印中心线右边的数字，左边的数字是右边的 2 倍。

设计性实验——输出最大的数及其出现的次数

输入 1 个整数，如果该整数不是 0，则继续输入，直到输入 0 时为止。查找这一组数中最大的数，并输出最大的数出现的次数。如果输入的第一个数为 0，就输出“Only 0 is inputed”。程序运行结果如图 3-16 所示。



```

Console
<terminated> MaxNumberandTimes [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
Enter an integer, end when the input is 0:7
Enter an integer: 3
Enter an integer: 4
Enter an integer: 9
Enter an integer: 9
Enter an integer: 0
max is:9
The count of the max number is:2
  
```

图 3-16 输出最大的数及其出现的次数程序的运行结果

点拨

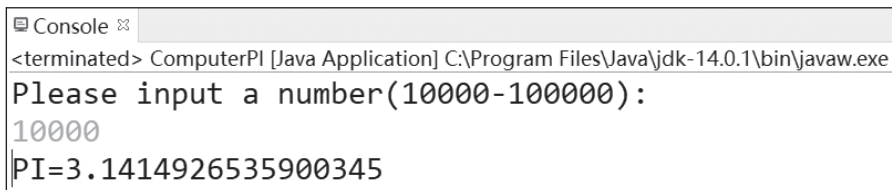
维护两个变量 max 和 total。max 存储当前最大的数，total 存储它出现的次数。输入第一个数时，如果该数不为 0，就将第一个数赋给 max，total 赋值为 1。将后面输入的每一个数与 max 进行比较，如果该数大于 max，就将其赋给 max，total 重置为 1；如果该数等于 max，则 total 加 1。

设计性实验——求 π 的值

π 的近似值可由如下公式计算：

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \cdots + \frac{1}{2n-1} \right)$$

读入一个在 [10 000, 100 000] 之内的整数 n ，求 π 的值。古代人是如何计算圆周率的？魏晋时期数学家刘徽利用割圆术求出了圆周率 π 的近似值，其思路如下：以圆内接六边形开始计算，令边数加倍，以圆内接正 3×2^n 边形的面积为圆面积的近似值，再利用公式 $\pi = \frac{S}{r^2}$ 来得到圆周率 π 的近似值。南北朝时期杰出的数学家、天文学家祖冲之在他的著作《缀术》中计算出圆周率 π 的真值在 3.141 592 6 和 3.141 592 7 之间，相当于精确到小数第 7 位，简化成 3.141 592 6，祖冲之因此入选世界纪录协会中世界第一位将圆周率值计算到小数第 7 位的科学家。古人采用最原始的人工计算，尚能将圆周率计算出这么高的精度，可见他们严谨的科学态度。请设计一个循环程序，通过调整精度对比运行结果。从循环次数可知，每增加一位精度，运算次数就要多几十倍甚至几百倍。程序运行结果如图 3-17 所示。



```

Console
<terminated> ComputerPI [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe
Please input a number(10000-100000):
10000
PI=3.1414926535900345

```

图 3-17 求圆周率 π 程序的运行结果

点拨

在求和表达式中，关于每一项的符号值，奇数项是 1，偶数项是 -1，每一项的数值乘以符号值为每一项的修正值。每一项修正值累加求和的 4 倍即为 π 的值。

拓展训练

- (1) for 循环都可以转换为 while 循环吗？尝试写出初始化条件语句、循环条件判断语句和循环变量修改语句在 for 循环、while 循环中的位置。
- (2) 用 break 和 break label 分别实现二重循环案例，比较 break 和 break label 的异同。
- (3) 嵌套循环需要运行较长的时间。考虑如下的三层循环中的输出语句需要执行多少次？如果 1 微秒执行 1 次输出，整个循环需要花费多长时间？

```

for (int i = 0; i < 10000; i++)
    for (int j = 0; j < 1000; j++)
        for (int k = 0; k < 10000; k++)
            System.out.println("Java");

```

第 4 单元

方 法

单元导读

Java 是一门面向对象的编程语言，每个对象都有自己的行为，这些行为又称为方法。Java 的方法可以使程序变得更简洁、更清晰，能够提高程序开发的效率，有利于程序维护，提升代码的可重用性。本单元主要讲解方法定义、方法参数传递、方法重载等内容，通过本单元的学习读者可以掌握使用 Java 语言编写方法的能力。

知识要点

1. 方法的定义

一般情况下，定义一个方法的语法如下：

```
修饰符 返回值类型 方法名 ( 参数类型 参数名 ){  
    .....  
    方法体  
    .....  
    return 返回值;  
}
```

修饰符决定编译器如何调用该方法。其中访问权限修饰符包括 `private`、`default`、`protected`、`public`。`private` 表示本类可以访问；`default` 表示本类和本包可以访问；`protected` 表示本包、本类和子类可以访问；`public` 在其他包、本包、本类和子类中都能访问。此外还有静态修饰符 `static`、最终修饰符 `final`、抽象修饰符 `abstract`。

返回值类型用来表述该方法返回值的类型。有些方法仅执行操作，没有返回值，在这种情况下，返回值的类型是 `void`。

方法名是方法的实际名称。方法名和参数列表共同构成方法签名。

形式参数（形参）列表是由多个形式参数组成的列表。一个方法可以有一个或多个参数，也可以不包含参数。

方法体就是方法包含的语句，用来实现该方法的功能。

一个完整的方法的定义如图 4-1 所示。

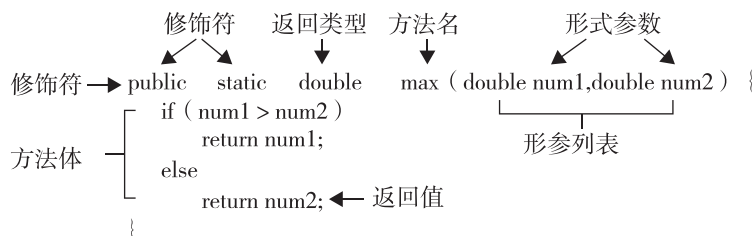


图 4-1 方法的定义

2. 方法的调用

方法的调用遵循以下几条基本规则：

- (1) 定义方法的时候，不会执行方法中的代码，只有方法被调用时才会执行；
- (2) 方法被调用时，会将实际参数（实参）赋值给形参，也称为参数形实结合；
- (3) 参数传递完毕后，会执行方法体内的代码；
- (4) 当方法执行到 `return` 语句或者执行完毕后，会回到调用位置继续往下执行。

3. 方法的重载

方法重载是指在同一个类中，有多个名称相同，但形参列表不同的方法。方法的重载遵循以下几条基本规则：

- (1) 方法名称必须相同；
- (2) 参数列表必须不同（参数个数不同，或形参类型不同，或参数排列顺序不同）；
- (3) 方法返回的类型不作为区分标志。

4. 变量的作用域

变量的作用域是程序中该变量可以被引用的部分。局部变量的作用域从该变量的声明处开始，到包含该变量的块结束为止，局部变量必须先声明后使用。

方法内定义的变量被称为局部变量。方法的参数作用范围涵盖整个方法，参数实际上是一个局部变量。在嵌套体中不能声明两次同名的变量。`for` 循环的初始化部分声明的变量，其作用范围为整个循环。但循环体内声明的变量的作用范围是从它声明的位置到循环体结束。

实验 方法的使用

知识目标

掌握方法的定义、方法的调用、方法的重载以及变量的作用域，了解如何使用 `Math` 类中的方法。

能力目标

学会创建方法、调用方法、给方法传递参数。

素质目标

培养持之以恒、坚持不懈的奋斗精神。



方法的使用

验证性实验——整数的倒置

要求使用递归编写方法 `reverse(int number)`，实现整数的倒置，并编译运行该程序。