



Java

项目驱动开发教程

迟殿委 刘丽贞 黄甜甜 杨嘉耀 著

清华大学出版社
北京

内 容 简 介

本书是一本以项目驱动教学模式理念设计的Java入门书。全书内容以软件项目案例驱动，代码实例丰富，项目实训任务饱满，语言通俗易懂，不仅能够使读者全面掌握Java面向对象编程技术，还能够提升读者使用Java语言开发软件项目的能力。本书配套示例源代码、PPT课件、教学大纲、课程设计指导书、实训任务指导书、课后练习答案及Java核心编程参考文档等资源。

本书分为3个专题，共14章，3个专题分别以面向过程版的商超购物管理系统、面向对象版的超市购物系统、端对端聊天系统为贯穿项目，内容包括Java开发入门、Java的基本语法、Java程序流程控制、数组、Java类和对象、Java继承和多态、Java抽象类和接口、Java异常处理、Java图形界面编程、Java集合、Java多线程、Java网络编程、Java I/O流、Java反射机制。

本书适合Java编程初学者系统地学习Java核心编程技术，同时也非常适合高等院校相关课程的师生作为教学参考书或教材使用。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

Java 项目驱动开发教程/迟殿委等著. —北京：清华大学出版社，2023.5

ISBN 978-7-302-63528-4

I. ①J… II. ①迟… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国国家版本馆 CIP 数据核字(2023)第 087468 号

责任编辑：夏毓彦

封面设计：王 翔

责任校对：闫秀华

责任印制：沈 露

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座

邮 编：100084

社 总 机：010-83470000

邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市君旺印务有限公司

经 销：全国新华书店

开 本：190mm×260mm

印 张：16.75

字 数：452 千字

版 次：2023 年 6 月第 1 版

印 次：2023 年 6 月第 1 次印刷

定 价：89.00 元

产品编号：081830-01

前言

Java语言是当今流行的面向对象编程语言之一，Java以其健壮性、安全性、可移植性等优点成为程序员必备的技术。随着大数据分析和人工智能技术的发展，市场对掌握Java语言的人才的需求量依然很大，与Java相关的就业方向很广，但无论从事什么方向，Java核心编程技术都是首先必须掌握的。本书围绕3个专题，以典型案例贯穿项目展开各个专题的讲解，将项目拆分为实训任务植入各个章节，方便读者自学以及教师开展项目驱动式教学。本书技术点全面、案例丰富，对知识点讲解细致、通俗易懂，配套讲义、教材案例及贯穿项目的源代码等资源，能够让读者在学习过程中更加轻松。通过本书的学习，读者不仅能够全面掌握Java面向对象编程技术，还能够提升使用Java语言开发软件项目的能力。

本书特点

本书内容安排由浅入深，按编程入门、基础语法、面向对象、高级特性的顺序，逐步提高难度，符合一般读者的学习规律。每个章节开始都有关于本章的内容简介，概括描述本章的主要内容和学习目标，让读者带着目的去读书；章节最后都有本章总结，归纳本章的重要内容，帮助读者形成连贯的知识体系。

本书除了配套章节知识点相关的代码实例之外，重点引入与专题章节相关的阶段性贯穿项目案例，方便读者自学以及教师开展项目驱动式教学。

本书的案例以JDK 1.8版本编写，这个版本也是企业开发中普遍采用的稳定版本，示例代码能够运行在JDK 1.8及以上版本的Java环境中。

本书作者具有丰富的企业软件研发经验和Java EE方向的教学培训经验，了解初学者学习的典型情况和容易产生混淆或疑惑的知识点，书中技术要点均以最直观化、最易懂的方式表达出来。

本书内容

本书按照Java语言的核心编程知识和企业应用开发需求，将主要内容划分为：Java语言编程基础、Java面向对象程序设计、Java API高级编程三大专题。根据三个专题的主要内容设计每个专题对应的贯穿阶段项目案例，项目案例贯穿整个专题的各个章节。

Java语言编程基础专题以“面向过程版的商超购物管理系统”作为贯穿项目,包括Java开发入门、Java基本语法、Java程序流程控制和数组4章,用于夯实Java编程语法基础。

Java面向对象程序设计专题以“面向对象版的超市购物系统”为贯穿项目,包括Java类和对象、Java继承和多态、Java抽象类和接口、Java异常处理及Java图形界面编程5章,为面向对象程序设计思想的理解和编程能力的提升奠定基础。

Java API高级编程专题以“端对端聊天系统”为贯穿项目,包括Java集合、Java多线程、Java网络编程、IO流及Java反射机制5章,用于提升读者应用Java API进行软件开发的能力。

示例源代码、PPT 课件、教学大纲等资源下载

本书配套示例源代码、PPT课件、教学大纲、课程设计指导书、实训任务指导书、课后练习答案及Java核心编程参考文档,需要使用微信扫描右面的二维码获取。阅读过程中如果发现问题或者疑问,请发送邮件至booksaga@163.com,邮件主题写“Java项目驱动开发教程”。



本书读者

本书精心选取企业开发所需的、系统的Java编程核心技术,没有额外的内容堆叠,层次清晰,实战性强,配套资源丰富,非常适合需要全面学习Java核心编程知识的初学者,也适合高等院校相关专业师生作为教材或教学参考书使用。

作 者
2023年2月

目 录

第一专题 Java 语言编程基础

第 1 章 Java 开发入门	5	1.9.4 Eclipse 项目的导入	19
1.1 Java 简介	5	1.9.5 在 Eclipse 中给 main 方法 传递参数	20
1.2 Java 基础开发环境搭建	6	1.9.6 Eclipse 的快捷键	21
1.2.1 JDK 下载	6	1.10 实训 1: 商超购物管理系统欢迎 界面	22
1.2.2 安装 JDK	7	1.11 本章总结	22
1.2.3 配置环境变量	8	1.12 课后练习	23
1.2.4 测试是否安装成功	8		
1.3 Java 编程初体验	9	第 2 章 Java 的基本语法	24
1.3.1 创建 HelloWorld.java 源文件	9	2.1 Java 程序的基本格式	24
1.3.2 javac 命令编译	9	2.2 Java 中的关键字	26
1.3.3 java 命令运行	10	2.3 Java 中的标识符	27
1.4 Java 带包类的编译和运行	10	2.4 Java 中的常量	28
1.4.1 修改 HelloWorld.java 的 源代码	10	2.5 Java 中的变量	29
1.4.2 通过 javac 命令重新编译	10	2.5.1 变量声明的语法	29
1.4.3 通过 java 命令运行有包 声明的类	11	2.5.2 Java 中的数据类型	30
1.5 javac 命令的更多参数	11	2.5.3 数据类型与默认值	31
1.6 java 命令的更多参数	12	2.5.4 成员变量与局部变量	32
1.7 main 方法接收参数	13	2.5.5 在 main 方法中访问成员 变量	32
1.8 javadoc 命令	14	2.6 Java 运算符和表达式	33
1.9 Java 开发利器	15	2.6.1 Java 中的运算符列表	33
1.9.1 下载 Eclipse	16	2.6.2 进制之间的转换	38
1.9.2 安装 Eclipse	16	2.6.3 基本类型及其包装类型	39
1.9.3 Eclipse 中 Java 项目的创建	17	2.6.4 equals 方法	39

2.7	Java 修饰符和包结构	40	3.5	本章总结	60
2.7.1	Java 包结构	40	3.6	课后练习	60
2.7.2	导入包	41			
2.7.3	访问修饰符	43	第 4 章 数组	62	
2.8	实训 2: 文件创建和数据类型转换	48	4.1	数组初探	62
2.9	本章总结	51	4.1.1	创建数组	62
2.10	课后练习	51	4.1.2	数组的维度	63
第 3 章 Java 程序流程控制	53		4.2	数组的遍历	67
3.1	Java 分支结构	53	4.3	数组的排序	68
3.1.1	单分支语句	54	4.3.1	冒泡排序	69
3.1.2	switch 语句	54	4.3.2	直接选择排序	69
3.2	Java 循环结构	55	4.3.3	插入排序	70
3.2.1	while 循环	55	4.3.4	快速排序	71
3.2.2	do-while 循环	55	4.4	数组元素的查找	72
3.2.3	for 循环	56	4.5	Arrays 工具类	73
3.3	break 和 continue 关键字	56	4.6	实训 4: 商品管理	73
3.4	实训 3: 登录及收银	58	4.7	本章总结	78
			4.8	课后练习	78

第二专题 Java 面向对象程序设计

第 5 章	Java 类和对象	83	5.7	课后练习	91
5.1	对象和类的概念	83	第 6 章	Java 的继承和多态	92
5.1.1	对象的概念	83	6.1	Java 的继承	92
5.1.2	类的概念、类与对象关系	83	6.2	重写	94
5.2	类与对象的定义和使用	84	6.2.1	重写 toString	95
5.2.1	类的设计	84	6.2.2	重写 equals	96
5.2.2	对象的创建和使用	85	6.3	类型转换	98
5.3	构造函数和重载	86	6.4	super 关键字	100
5.3.1	Java 中的构造函数	86	6.5	多态	101
5.3.2	Java 中的默认构造方法	87	6.5.1	多态的定义	101
5.3.3	构造方法及其重载	87	6.5.2	多态的实现	103
5.4	成员变量、局部变量、this 关键字	88	6.6	实训 6: 输出不同商品信息	106
5.5	实训 5: 商品价格计算	90	6.7	本章总结	107
5.6	本章总结	91	6.8	课后练习	107

第 7 章 Java 抽象类和接口	108	9.2 组件和容器	132
7.1 Java 抽象类	108	9.3 事件驱动程序设计基础	132
7.2 Java 抽象方法	109	9.3.1 事件、监视器和监视器 注册	132
7.3 实训 7: 简易超市购物系统	110	9.3.2 实现事件处理的途径	132
7.4 接口	111	9.3.3 事件类型和监视器接口	133
7.4.1 Java 的多重继承	113	9.4 界面组件	134
7.4.2 通过继承来扩展接口	114	9.4.1 窗口	134
7.4.3 接口中的常量	115	9.4.2 容器	135
7.4.4 JDK 1.8 的默认实现	115	9.4.3 标签	137
7.5 本章总结	115	9.4.4 按钮	137
7.6 课后练习	116	9.4.5 JPanel	138
第 8 章 Java 异常处理	117	9.4.6 JScrollPane	139
8.1 Java 异常概述	117	9.4.7 文本框	140
8.2 Java 异常处理方法	118	9.4.8 文本区	141
8.2.1 处理异常: try、catch 和 finally	118	9.4.9 选择框	143
8.2.2 try-catch-finally 规则	119	9.4.10 单选框	143
8.2.3 声明抛出异常	120	9.4.11 单选按钮	144
8.2.4 JDK 1.7 一次捕获多个异常	121	9.4.12 列表	144
8.3 Java 异常处理的分类	121	9.4.13 组合框	145
8.3.1 检测异常	121	9.4.14 菜单条、菜单和菜单项	146
8.3.2 非检测异常	122	9.5 布局	148
8.3.3 自定义异常	122	9.5.1 FlowLayout 布局	149
8.4 Java 异常处理的原则和忌讳	122	9.5.2 BorderLayout 布局	149
8.4.1 Java 异常处理的原则	122	9.5.3 GridLayout 布局	149
8.4.2 Java 异常处理的忌讳	122	9.5.4 CardLayout 布局	150
8.5 Java 自定义异常	123	9.5.5 null 布局与 setBounds 方法	151
8.6 常见的异常	125	9.6 实训 9: 超市管理系统图形登录 界面	152
8.7 实训 8: 商品信息查询	126	9.7 对话框	153
8.8 异常的典型举例	127	9.7.1 JDialog 类	153
8.9 本章总结	129	9.7.2 JOptionPane 类	155
8.10 课后练习	130	9.8 鼠标事件	157
第 9 章 Java 图形界面编程	131	9.8.1 MouseListener 接口	157
9.1 AWT 和 Swing	131	9.8.2 MouseMotionListener 接口	160
		9.9 键盘事件	162

9.10 本章总结	163	9.11 课后练习	163
-----------------	-----	-----------------	-----

第三专题 Java API 高级编程

第 10 章 Java 集合	169	11.6.1 资源冲突	193
10.1 Collection 接口	169	11.6.2 对象锁的实现	194
10.1.1 AbstractCollection 抽象类	170	11.6.3 线程间的同步控制	196
10.1.2 Iterator 接口	170	11.7 本章总结	199
10.2 List 接口	171	11.8 课后练习	200
10.3 Set 接口	173	第 12 章 Java 网络编程	201
10.3.1 Hash 表	173	12.1 两类传输协议: TCP 和 UDP	201
10.3.2 Comparable 接口和 Comparator 接口	173	12.1.1 两者之间的比较	201
10.3.3 SortedSet 接口	176	12.1.2 应用	202
10.3.4 HashSet 类和 TreeSet 类	176	12.2 基于 Socket 的 Java 网络编程	202
10.4 Map 接口	178	12.2.1 什么是 Socket	202
10.4.1 HashMap 类和 TreeMap 类	179	12.2.2 Socket 通信的过程	202
10.4.2 LinkedHashMap 类	181	12.2.3 创建 Socket	203
10.5 本章总结	181	12.3 实训 11: 服务器服务线程	203
10.6 课后练习	181	12.4 简单的 Client/Server 程序	205
第 11 章 Java 多线程	182	12.5 实训 12: 客户端处理线程	207
11.1 线程与线程类	182	12.6 Datagram 通信	209
11.1.1 线程的概念	182	12.6.1 什么是数据报	210
11.1.2 Thread 类和 Runnable 接口	184	12.6.2 数据报的使用	210
11.2 线程的创建	185	12.6.3 用数据报进行广播通信 (MulticastSocket)	211
11.2.1 继承 Thread 类并创建线程	185	12.7 本章总结	213
11.2.2 实现 Runnable 接口并创建 线程	186	12.8 课后练习	213
11.3 实训 10: 开启服务器主线程	187	第 13 章 Java IO 流	214
11.4 线程的状态与调度	189	13.1 输入/输出字节流	214
11.5 线程状态的改变	190	13.1.1 InputStream 类	215
11.5.1 控制线程的启动和结束	191	13.1.2 OutputStream 类	215
11.5.2 线程就绪和阻塞条件	192	13.1.3 FileInputStream 类	216
11.6 线程的同步与共享	193	13.1.4 FileOutputStream 类	216

13.1.5 其他输入输出字节流	217	第 14 章 Java 反射机制	242
13.2 实训 13: 用户注册功能	221	14.1 获取类的方法	242
13.3 实训 14: 用户登录功能	228	14.2 获取构造函数信息	243
13.4 输入/输出字符流	233	14.3 获取类的字段	244
13.4.1 字符输入流 Reader	234	14.4 根据方法的名称来执行方法	245
13.4.2 字符输出流 Writer	235	14.5 改变字段的值	246
13.4.3 转换输入/输出流	236	14.6 类加载与反射创建对象	247
13.5 File 类	237	14.6.1 类加载机制	247
13.5.1 File 类的对象代表文件 路径	237	14.6.2 通过反射创建对象及获取 对象信息	248
13.5.2 File 类的常用方法	237	14.7 实训 15: 添加好友和好友列表	251
13.6 本章总结	239	14.8 实训 16: 好友聊天功能	255
13.7 课后练习	240	14.9 本章总结	258
		14.10 课后练习	258

第一专题

Java语言编程基础

本专题主要讲解Java开发入门、Java基本语法、Java流程控制语句和数组。本专题对应的贯穿项目案例为：商超购物管理系统，具体项目需求和最终效果描述如下。

商超购物管理系统包括商品维护、前台收银两大功能。基本需求和效果如专题一图1所示。

商超购物管理系统

1.商品维护
2.前台收银

请选择，输入数字或按0退出：

专题一图1

1. 商品维护

(1) 商品维护菜单的显示：输入数字进入相应操作界面，输入0返回主菜单，如专题一图2所示。

商超购物管理系统 商品维护

1.添加商品
2.更改商品
3.删除商品
4.显示所有商品
5.查询商品

请选择，输入数字或按0返回上一级菜单：

专题一图2

(2) 商品添加：输入商品名称、商品价格和商品数量。输入的商品价格应为大于0的实数，商品数量应为大于0的整数，否则显示输入错误。完成一件商品录入后，可选择输入“y”继续进行添加操作，或者输入“n”返回商品维护菜单，如专题一图3所示。

请选择，输入数字或按0返回上一级菜单：

1

执行添加商品操作：

添加商品名称：

香蕉

输入添加商品价格：

3

输入添加商品数量：

200

是否继续 (y/n)

专题一图3

(3) 商品的更改：输入要更改的商品名，显示现有的商品名称、商品价格和商品数量，选择要更改的项（商品名称、价格或者数量），输入变更内容，完成本次变更。可输入“y”继续进行更改操作，或者输入“n”返回商品维护菜单，如专题一图4所示。

请选择，输入数字或按0返回上一级菜单：

2

执行更改商品操作

输入更改商品名称：

香蕉

商品名称	商品价格	商品数量	备注
香蕉	3.0	200	

选择您要更改的内容：

1、更改商品名称：

2、更改商品价格：

3、更改商品数量：

3

请输入已更改商品数量

600

是否继续 (y/n)

专题一图4

(4) 商品的删除：输入要删除的商品名，显示现有的商品名称、商品价格和商品数量，再次确认即可删除该商品的全部数据。输入“y”继续进行删除操作，或者输入“n”返回商品维护菜单，如专题一图5所示。

请选择，输入数字或按0返回上一级菜单：

3

执行商品删除操作

请输入要删除的商品名称：

香蕉

是否继续 (y/n)

专题一图5

（5）商品列表显示：可显示已有所有商品的名称、价格和数量，另有一列备注项，可提示不足库存，如专题一图6所示。

请选择，输入数字或按 0 返回上一级菜单：
4
显示所有商品

商品名称	商品价格	商品数量	备注
杯子	15.0	150	
碗	10.0	100	
盘	20.0	193	
勺子	3.0	2	*该商品已不足 10 件！
筷子	3.0	1999	
苹果	5.5	100	

专题一图6

（6）商品查询：可以选择按商品数量升序查询，或者按商品价格升序查询，以及输入关键字查询商品，如专题一图7所示。

请选择，输入数字或按0返回上一级菜单：
5
执行查询商品操作

1、按商品数量升序查询
2、按商品价格升序查询
3、输入关键字查询商品

请选择，输入数字或按0返回上一级菜单：

专题一图7

2. 前台收银

（1）售货员登录：选择登录系统，随机根据提示输入用户名和密码，若用户名和密码校验正确则进入系统，若校验失败则需重新输入用户名和密码。共有3次登录机会，若连续3次登录均校验失败，则退出程序，如专题一图8所示。

请选择，输入数字： 1
请输入用户名： xiaoming

请输入密码： 123

用户名和密码不匹配！
您还有2次登录机会,请重新输入：
请输入用户名：

专题一图8

(2) 购物结算：输入商品关键字可显示商品相关信息，输入商品名称，填写购买数量，可自动显示商品单价和总价。输入“y”继续添加商品，输入“n”不再添加新商品，显示总计金额。输入实际交费金额，显示找零金额，确认后商品收银成功，商品库存数量减少，如专题一图9所示。

1.购物结算

输入商品关键字：
勺

商品名称	商品价格	商品数量	备注
勺子	3.0	2	*该商品已不足10件！

请选择商品： 勺子

请输入购买数量： 1
勺子 ¥3.0 购买数量1 勺子总价3.0

是否继续（y/n） n

总计： 3.0

请输入实际交费金额：
10

找钱： 7.0

谢谢光临！

专题一图9

环境要求：

- 要求使用Java集成开发环境Eclipse控制台开发程序。
- 要求使用Java数据类型转换、分支结构和循环结构，以及数组来实现所有功能。

项目要求：

该综合实训任务将作为本专题最后的测验项目。

第 1 章

Java开发入门

本章内容分为三部分。第一部分主要介绍Java编程语言的由来、发展和版本信息等。第二部分详细说明Java开发环境的安装和配置。开发Java程序前，必须安装Java开发环境，就像写Doc文档前要安装WPS或MS Office软件一样。开发Java程序需要安装JDK（Java Development Kit，Java开发工具包）。在安装JDK的同时，自带安装一个JRE（Java Runtime Environment，Java运行环境）。JRE也可以理解成我们经常说的JVM（Java Virtual Machine，Java虚拟机）。JRE/JVM就是Java程序运行的地方。第三部分带领读者体验Java编码、编译和运行的过程。该部分带领读者开发第一个Java源程序，并通过javac命令将Java源程序编译成可执行的字节码文件，了解Java程序的开发。在初学阶段，读者编写Java源程序时，可以使用记事本或者EditPlus、UltraEdit等高级记事本工具。

1.1 Java简介

Java最早是由SUN公司（已被Oracle收购）的詹姆斯·高斯林（Java之父）在20世纪90年代初开发的一种编程语言，最初被命名为Oak，目标是针对小型家电设备的嵌入式应用，结果市场没什么反响。互联网的崛起让Oak重新焕发了生机，于是SUN公司改造了Oak，在1995年以Java（Oak已经被人注册了，因此SUN注册了Java这个商标）的名称正式发布。随着互联网的高速发展，Java逐渐成为重要的网络编程语言。

Java介于编译型语言 and 解释型语言之间。编译型语言（如C、C++）直接编译成机器码执行，但是不同平台（x86、ARM等）的CPU指令集不同，因此需要编译出每一种平台的对应机器码。解释型语言（如Python、Ruby）没有这个问题，可以由解释器直接加载源码然后运行，代价是运行效率太低。Java将代码编译成一种“字节码”，类似于抽象的CPU指令，然后针对不同平台编写虚拟机，不同平台的虚拟机负责加载字节码并执行，这样就实现了“一次编写，到处运行”的效果。当然，这是针对Java开发者而言的。对于虚拟机，需要为每个平台分别开发。为了保证不同平台、不同公司开发的虚拟机都能正确执行Java字节码，SUN公司制定了一

系列的Java虚拟机规范。从实践的角度看，JVM的兼容性做得非常好，低版本的Java字节码完全可以正常运行在高版本的JVM上。

随着 Java 的发展，SUN 给 Java 分出了 3 个不同版本：

- Java SE: Standard Edition。
- Java EE: Enterprise Edition。
- Java ME: Micro Edition。

简单来说，Java SE 就是标准版，包含标准的 JVM 和标准库；Java EE 是企业版，在 Java SE 的基础上添加了大量的 API 和库，以便开发 Web 应用、数据库、消息服务等；Java EE 使用的虚拟机和 Java SE 完全相同。

Java ME和Java SE不同，它是一个针对嵌入式设备的“瘦身版”，Java SE的标准库无法在Java ME上使用，Java ME的虚拟机也是“瘦身版”。

毫无疑问，Java SE是整个Java平台的核心，而Java EE是进一步学习Web应用所必需的。我们熟悉的Spring等框架都是Java EE开源生态系统的一部分。不幸的是，Java ME从来没有真正流行起来，反而是Android开发发展成为移动平台的标准之一。因此，没有特殊需求，不建议学习Java ME。

我们推荐的 Java 学习路线如下：

- 首先要学习Java SE，掌握Java语言本身、Java核心开发技术以及Java标准库的使用。
- 如果继续学习Java EE，那么Spring框架、数据库开发、分布式架构就是需要学习的。
- 如果要学习大数据开发，那么Hadoop、Spark、Flink这些大数据平台就是需要学习的，它们都基于Java或Scala开发的。
- 如果想要学习移动开发，就要深入学习Android平台，掌握Android App开发。

无论怎么选择，Java SE 的核心技术是基础。

1.2 Java基础开发环境搭建

要用Java进行开发，就需要准备开发、编译、运行各个阶段需要的软件或工具。Java开发所需要的工具集合包含在JDK中，所以要先到网上下载JDK的安装程序。不同的操作系统对应不同的版本，具体下载、安装、配置的过程会在下面具体介绍。

1.2.1 JDK下载

JDK可到官网下载，下载页面如图1-1所示。

选择同意选项，并根据自己的操作系统选择不同的版本，如图1-2所示。



图1-1

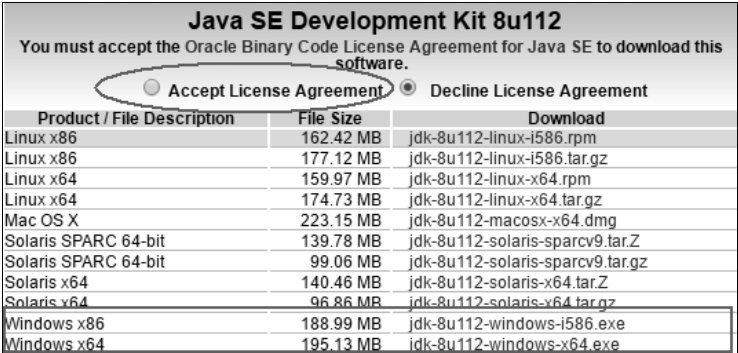


图1-2

1.2.2 安装JDK

下面演示如何在Windows操作系统上安装JDK。双击如图1-3所示的安装程序，选择安装目录（见图1-4），设置开发工具为JDK。选择安装“源代码”，可以方便在开发时查看源代码。公共JRE即独立的JVM运行环境。其实，在开发工具内部也包含一个公共JRE。

安装成功的界面如图1-5所示，直接关闭即可。



图1-3



图1-4



图1-5

1.2.3 配置环境变量

配置环境变量主要是为了让命令程序可以自动识别javac.exe可执行程序。javac.exe是编译Java源文件的命令,通常叫javac编译命令。它位于JDK安装目录下的/bin目录下,如C:\Program Files\Java\JDK1.8_xx\bin。为了让javac编译命令可运行,需要配置以下两个环境变量:

- JAVA_HOME=C:\Program Files\Java\JDK1.8_xx;, 即配置JAVA_HOME为JDK的安装目录。
- PATH=%JAVA_HOME%\bin;, 即配置PATH为JDK安装目录下的bin目录。

配置环境变量: 右击“我的电脑”, 选择“属性”菜单, 在弹出的窗口中选择“高级系统设置”项, 再在弹出的对话框中单击“环境变量”按钮, 弹出如图 1-6 所示的对话框, 添加 JAVA_HOME 环境变量。



图1-6

接着添加PATH环境变量, 如图1-7所示。

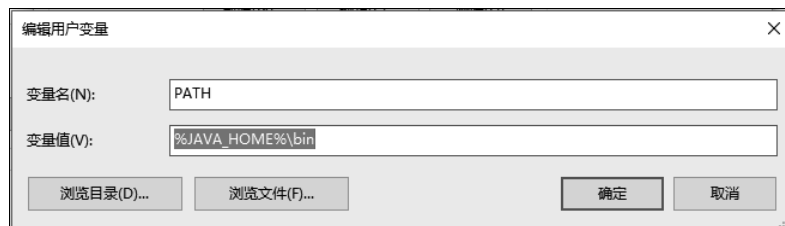


图1-7

注意:

- (1) 建议配置用户环境变量即可。如果配置系统环境变量, 那么所有用户登录都可以。
- (2) 如果已经存在PATH环境变量, 就应该在原有变量的基础上, 通过英文分号 (;) 分开并追加到后面。
- (3) 建议环境变量的名称使用大写字母。

1.2.4 测试是否安装成功

打开命令行工具界面 (可以通过按 Win+R 快捷键, 并在打开的对话框中输入“cmd”的方式快速打开), 然后输入:

```
C:\>javac -version  
javac 1.8.0_77
```

如果通过 `javac -version` 命令输出了 `javac` 编译的版本，并且输出正确，则说明安装成功。

1.3 Java编程初体验

Java源文件就是一个以`*.java`结束的文本文件。Java语言是编译执行的语言。运行Java程序，必须先将`*.java`文件通过`javac`编译成`*.class`文件，然后通过`java`命令运行`*.class`文件，整个编译的运行过程如图1-8所示。



图1-8

在开发之前，建议创建一个目录，用于保存所有的Java源文件。本章中的所有源代码都将保存到D:\java目录下。

1.3.1 创建HelloWorld.java源文件

建议选择一个比较干净的目录，然后创建一个名称为`HelloWorld.java`的文本文件。

创建`HelloWorld.java`源文件，如图1-9所示。



图1-9

【文件 1.1】 HelloWorld.java

```
1. public class HelloWorld{
2.     public static void main(String[] args){
3.         System.out.println("HelloWorld");
4.     }
5. }
```

在上面的代码中，`public` 为权限修饰关键字，意为公开的。`class` 用于声明一个类。在 Java 中，所有的函数（方法）都必须放到一个类中，这也是面向对象的基本特性之一。`HelloWorld` 为类的名称。Java 规定，以 `public` 修饰的类必须与文件名相同，并区分大小写。`main` 为程序的入口方法。一个 Java 类，甚至是一个 Java 程序（可能包含 N 个类）一般都只有一个 `main` 入口方法。在目前学习阶段，我们可以在每一个类中都声明 `main` 方法。`String[] args` 为形式参数。

第3行为系统输出语句，用于输出`HelloWorld`到命令行。

开发时，请注意大小写，执行语句结束使用英文分号（`;`），大括号（指上面代码中的“`{`”和“`}`”）要有开始和结束。

1.3.2 javac命令编译

在命令行中输入以下代码：

```
D:\java> javac HelloWorld.java
```

将会发现在同一目录下已经将 HelloWorld.java 编译成 HelloWorld.class，如图 1-10 所示。



图1-10

1.3.3 java命令运行

在使用 java 命令运行 HelloWorld.class 文件时，不需要输入.class 扩展名：

```
D:\java>java HelloWorld
HelloWorld
```

如果输出 HelloWorld，则表示 HelloWorld 程序运行成功。

1.4 Java带包类的编译和运行

包声明的关键字为package。在Java中，可以将相同的类放到不同的包中加以区分。同时，package包声明语句还可以进行基本的权限控制。

1.4.1 修改HelloWorld.java的源代码

修改HelloWorld.java的源代码，在第一句添加package关键字声明的包。

【文件 1.2】 HelloWorld1.java

```
1. package cn.oracle;
2. public class HelloWorld1{
3.     public static void main(String[] args){
4.         System.out.println("HelloWorld");
5.     }
6. }
```

第1行为新添加的包声明语句，后面通过点（.）声明带有层次的包，如cn.oracle（在cn包下的oracle子包）。

1.4.2 通过javac命令重新编译

javac命令拥有一个参数-d <目录>，可以直接将包声明语句编译成目录。

```
D:\java>javac -d . HelloWorld.java
```

-d参数后面的点（.）为当前目录，即将HelloWorld.java源文件带包名直接编译到当前目录下，编译以后的文件名如图1-11所示。

在cn目录下有一个oracle目录，oracle目录下有HelloWorld1.class源文件。

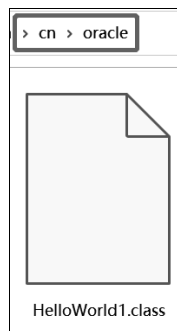


图1-11

使用package声明的包最终将编译成文件夹。其实也可以直接将包声明语句理解为目录或文件夹，只要便于记忆即可（记住，拥有自己独特的学习和记忆方法是成功的关键）。

1.4.3 通过java命令运行有包声明的类

在使用javac -d <目录>编译成功以后，编译的目录（源代码所在的目录）叫源代码目录。编译后的目录叫classpath目录（存放所有*.class的目录）。我们不能直接进入cn/oracle目录中去运行一个Java程序。注意：只能在classpath的根目录（D:/java）下执行Java运行命令。

运行Java程序：

```
D:\a>java cn.oracle.HelloWorld1
HelloWorld
```

注意：在创建Java源文件时，通过“我的电脑→查看→文件→选项”操作，将“隐藏已知文件类型的扩展名”选项取消，如图1-12所示。

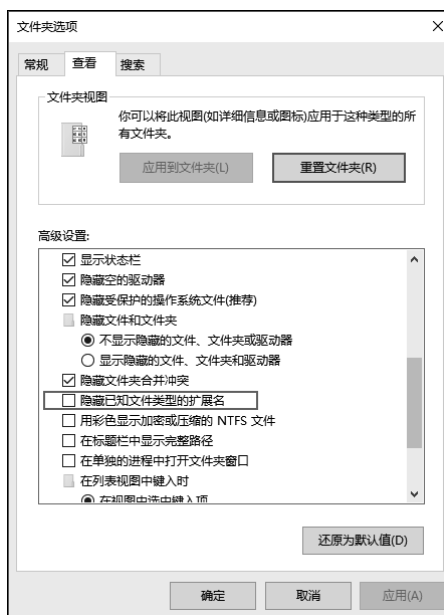


图1-12

1.5 javac命令的更多参数

javac命令的更多参数可以通过运行javac -help命令来查看：

```
D:\java>javac -help
```

用法：

```
javac <options> <source files>
```

其中，可能的选项包括：

-g	生成所有调试信息
-g:none	不生成任何调试信息
-g:{lines,vars,source}	只生成某些调试信息
-nowarn	不生成任何警告
-verbose	输出有关编译器正在执行的操作的消息
-deprecation	输出使用已过时的 API 的源位置
-classpath <路径>	指定查找用户类文件和注释处理程序的位置
-cp <路径>	指定查找用户类文件和注释处理程序的位置
-sourcepath <路径>	指定查找输入源文件的位置
-bootclasspath <路径>	覆盖引导类文件的位置
-extdirs <目录>	覆盖所安装扩展的位置
-endorseddirs <目录>	覆盖签名的标准路径的位置
-proc:{none,only}	控制是否执行注释处理和/或编译
-processor <class1>[,<class2>,<class3>...]	要运行的注释处理程序的名称，绕过默认的搜索进程
-processorpath <路径>	指定查找注释处理程序的位置
-parameters	生成元数据以用于方法参数的反射
-d <目录>	指定放置生成的类文件的位置
-s <目录>	指定放置生成的源文件的位置
-h <目录>	指定放置生成的本机标题文件的位置
-implicit:{none,class}	指定是否为隐式引用文件生成类文件
-encoding <编码>	指定源文件使用的字符编码
-source <发行版>	提供与指定发行版的源兼容性
-target <发行版>	生成特定 VM 版本的类文件
-profile <配置文件>	请确保使用的 API 在指定的配置文件中可用
-version	版本信息
-help	输出标准选项的提要
-A 关键字 [=值]	传递给注释处理程序的选项
-X	输出非标准选项的提要
-J<标记>	直接将<标记>传递给运行时系统
-Werror	出现警告时终止编译
@<文件名>	从文件读取选项和文件名

1.6 java命令的更多参数

java命令的更多参数可以通过运行java -help命令查看：

```
D:\java>java -help
```

用法：

```
java [-options] class [args...]  
      (执行类)
```

或

```
java [-options] -jar jarfile [args...]  
      (执行 jar 文件)
```

其中选项包括：

```
-d32          使用 32 位数据模型 (如果可用)
```

```

-d64          使用 64 位数据模型 (如果可用)
-client       选择 "client" VM
-server       选择 "server" VM
              默认 VM 是 client

-cp <目录和 zip/jar 文件的类搜索路径>
-classpath <目录和 zip/jar 文件的类搜索路径>
           用 ; 分隔的目录, JAR 档案
           和 ZIP 档案列表, 用于搜索类文件
-D<名称>=<值>
           设置系统属性
-verbose:[class|gc|jni]
           启用详细输出
-version      输出产品版本并退出
-version:<值>
           警告: 此功能已过时, 将在
           未来发行版中删除。
           需要指定的版本才能运行
-showversion  输出产品版本并继续
-jre-restrict-search | -no-jre-restrict-search
           警告: 此功能已过时, 将在
           未来发行版中删除。
           在版本搜索中包括/排除用户专用 JRE
-? -help     输出此帮助消息
-X           输出非标准选项的帮助
-ea[:<packagename>...|:<classname>]
-enableassertions[:<packagename>...|:<classname>]
           按指定的粒度启用断言
-da[:<packagename>...|:<classname>]
-disableassertions[:<packagename>...|:<classname>]
           禁用具有指定粒度的断言
-esa | -enablesystemassertions
           启用系统断言
-dsa | -disablesystemassertions
           禁用系统断言
-agentlib:<libname>[=<选项>]
           加载本机代理库 <libname>, 例如-agentlib:hprof
           另请参阅 -agentlib:jdwp=help 和-agentlib:hprof=help
-agentpath:<pathname>[=<选项>]
           按完整路径名加载本机代理库
-javaagent:<jarpath>[=<选项>]
           加载 Java 编程语言代理, 请参阅 java.lang.instrument
-splash:<imagepath>
           使用指定的图像显示启动屏幕

```

1.7 main方法接收参数

在main方法中, `String[] args`为命令行参数。在执行时, 可以利用空格通过“java 参数1 参数2...”的方式将所有参数传递给入口方法main。

【文件 1.3】 HelloWorld2.java

```
1. package cn.oracle;
2. public class HelloWorld2{
3.     public static void main(String[] args){
4.         System.out.println("参数的个数为: "+args.length);
5.         for(int i=0;i<args.length;i++){
6.             System.out.println(args[i]);
7.         }
8.     }
9. }
```

在上面的代码中，第4行输出命令行参数的个数。for是循环控制语句（后面将会讲到），用于从第一个参数输出到最后一个参数。

使用javac编译上面的代码，然后使用以下命令运行编译以后的程序：

```
D:\a>javac -d . HelloWorld2.java
D:\a>java cn.oracle.HelloWorld2 Jack Mary Alex Mrchi
参数的个数为: 4
Jack
Mary
Alex
Mrchi
```

1.8 javadoc命令

javadoc 命令用于将标准的 javadoc 注释生成文档。javadoc 标准注释一般是：注释到类上，对类起说明作用；注释到方法或成员变量上，对方法或者功能成员变量的含义做出说明。例如，存在以下 javadoc 注释。

【文件 1.4】 ExampleJavaDoc.java

```
1. package cn.oracle;
2. /**
3.  用 javadoc 对类做出功能说明<br>
4.  本类演示如何使用 javadoc 注释<br>
5.  并演示如何通过 javadoc 命令生成文档
6.  @author oracle
7.  @version 1.0
8.  */
9. public class ExampleJavaDoc{
10.     /**
11.      用 javadoc 对成员变量添加说明
12.      */
13.     private String name;
14.
15.     /**
16.      用 javadoc 对方法添加说明<br>
17.      以下方法用于输出一个 HelloWorld 串
18.      */
```

```
19.     public void print(){
20.         System.out.println("HellOworld");
21.     }
22. }
```

使用以下命令生成标准文档：

```
D:\java>javadoc -author ExampleJavaDoc.java
正在加载源文件 ExampleJavaDoc.java...
正在构造 Javadoc 信息...
标准 Doclet 版本 1.8.0_77
正在构建所有程序包和类的树...
正在生成.\cn\oracle\ExampleJavaDoc.html...
正在生成.\cn\oracle\package-frame.html...
正在生成.\cn\oracle\package-summary.html...
正在生成.\cn\oracle\package-tree.html...
正在生成.\constant-values.html...
正在构建所有程序包和类的索引...
正在生成.\overview-tree.html...
正在生成.\index-all.html...
正在生成.\deprecated-list.html...
正在构建所有类的索引...
正在生成.\allclasses-frame.html...
正在生成.\allclasses-noframe.html...
正在生成.\index.html...
正在生成.\help-doc.html...
```

第 1 行是生成的命令，后面是自动生成文档时输出的信息。生成以后的文档如图 1-13 所示。打开index.html，将会看到标准的javadoc文档，如图1-14所示。

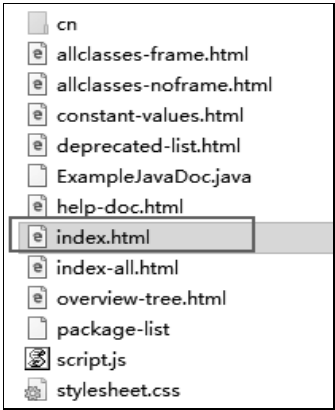


图1-13



图1-14

1.9 Java开发利器

工欲善其事，必先利其器。由于记事本编写代码速度慢且不易排查错误，为了提高程序

的开发效率，可以使用集成开发工具（Integrated Development Environment, IDE）进行 Java 开发，目前市场上的 Java IDE 很多，接下来为读者推荐几款 Java 开发工具：

- Eclipse（推荐）：免费开源的Java IDE，企业Java开发经典的IDE工具，有巨大稳定的用户群体、强大的插件支持和完善的技术资料。
- JetBrains的IntelliJ IDEA：目前有不少企业使用该开发工具，代码提示较为智能，功能强大。
- Notepad++：Notepad++是在微软Windows环境之下的一个免费的代码编辑器。
- NetBeans：开源免费的Java IDE，是Oracle公司收购的一个Java集成开发环境。

本书我们将使用 Eclipse 作为开发环境，采用的版本为 4.6.1。

1.9.1 下载Eclipse

Eclipse是一个开源且免费的开发环境，在www.eclipse.org官网上即可下载到最新版本的Eclipse。

Eclipse的下载页面如图1-15所示，找到跟JDK匹配的版本，本书采用的Eclipse的版本为4.6.1，读者需要根据操作系统位数下载相应的版本。

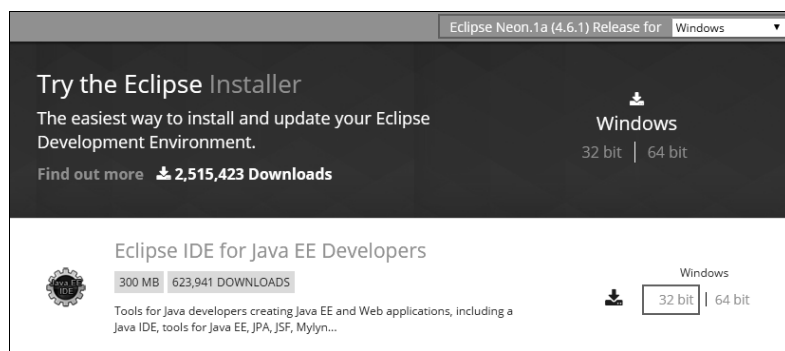


图1-15

1.9.2 安装Eclipse

下载的Eclipse软件是一个ZIP类型的压缩文件，解压即可使用。请保证你已经安装了JDK，并正确地配置了JAVA_HOME和PATH两个环境变量。

在解压以后，得到如图1-16所示的目录。

其中，eclipse.exe为运行Eclipse的可执行文件，双击后，将启动Eclipse，然后选择一个工作区（workspaces，今后所有Java项目所保存的目录）。

启动时要选择工作区，其中，workspace默认的目录为C:/administrator/workspaces，但是不建议将所有的项目都放到C盘，所以这里可以输入一个你喜欢的其他任意目录。建议工作区也不要放在C盘上。比如我们把它放在D盘，如图1-17所示。

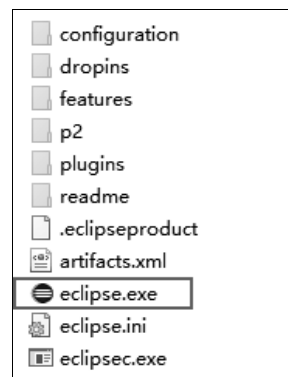


图1-16

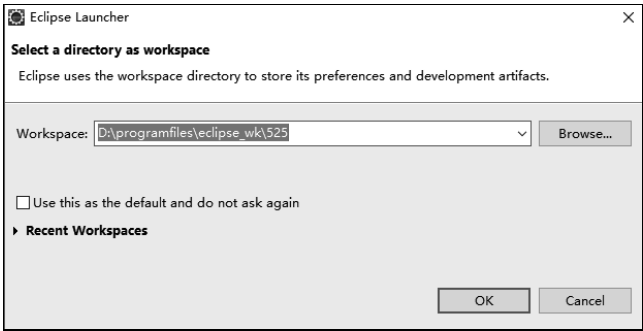


图1-17

1.9.3 Eclipse中Java项目的创建

在 Eclipse 中创建的 Java 项目为 Java 源代码项目，一般包含两个目录:src 为源代码目录，bin 为 classpath 目录。以下是 Java 项目的目录结构：

- project: 项目名，包含源代码目录src。
- bin: 字节码目录，所有编译后的.class文件都自动保存到这个目录下。
- .project: Eclipse项目的配置文件。

1. 创建Java项目

依次选择菜单项File→New→Java Project，如图1-18所示。



图1-18

2. 输入项目名称

在New Java Project窗口中输入项目名称以后，直接单击Finish按钮，如图1-19所示。

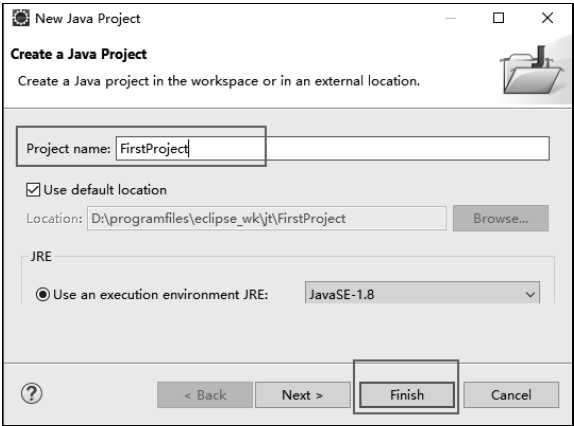


图1-19

3. 开发Java类

建议使用Package Explorer来查看项目的结构，它将会隐藏bin目录。虽然看不见bin目录，但是它依然存在。如果想要查看bin目录，则可以通过Navigation Explorer来查看，不过建议使用Package Explorer。创建以后的项目结果如图1-20所示。

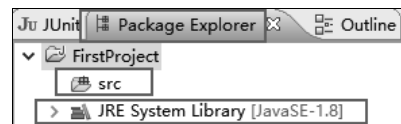


图1-20

在图1-20所示的窗口中，第一个框为显示的视图，第二个框src为源代码目录，第三个框JRE System Library为引用的JDK版本。

在src处右击，在弹出菜单中选择菜单项New→Class即可创建一个Java类，如图1-21所示。

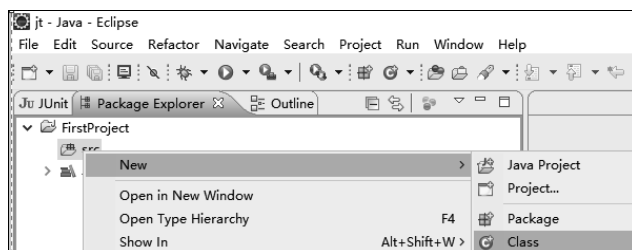


图1-21

输入类名及包名，如图1-22所示。



图1-22

创建的Java类已经有了类的结构，如包名和类名都已经自动填充完毕。

【文件 1.5】 HelloWorld3.java

```
1. package cn.oracle;
2. public class HelloWorld3 {
3. }
```

4. 填充main方法

此时，只要在HelloWorld类里面实现main方法即可。

【文件 1.6】 HelloWorld4.java

```
1. package cn.oracle;
2. public class HelloWorld4 {
3.     public static void main(String[] args) {
4.         System.err.println("HelloWorld");
5.     }
6. }
```

5. 运行程序

在Eclipse中运行一个main方法，只要在拥有main方法中的类中右击，在弹出菜单中选择Run As → Java Application即可，如图1-23所示。

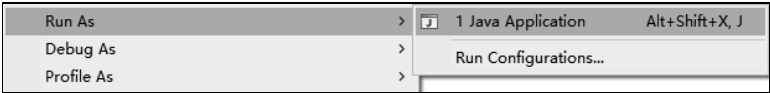


图1-23

运行结果可以通过控制台查看。

至此，就可以使用Eclipse开发Java项目了。

1.9.4 Eclipse项目的导入

如果已经存在一个Java项目，则可以使用Eclipse的导入功能直接导入，具体步骤如下。

- (1) 依次选择File→Import命令，如图1-24所示。
- (2) 选择已经存在的Eclipse项目，导入当前项目中，如图1-25所示。

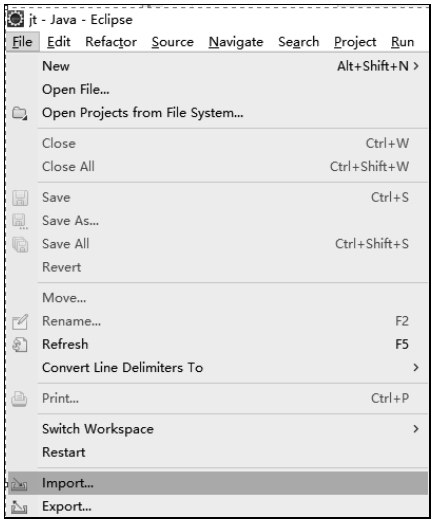


图1-24

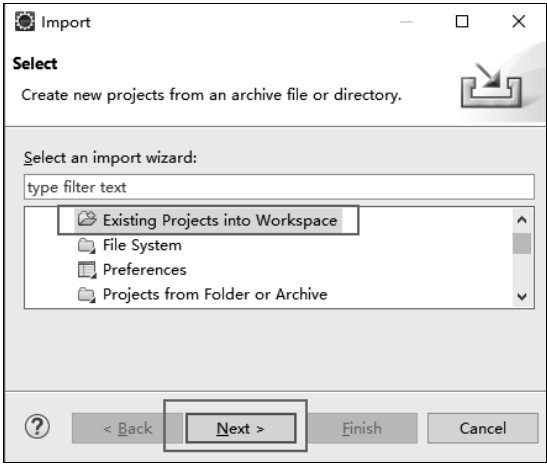


图1-25

(3) 选择需要导入的项目，并选中Copy projects into workspace，如图1-26所示。

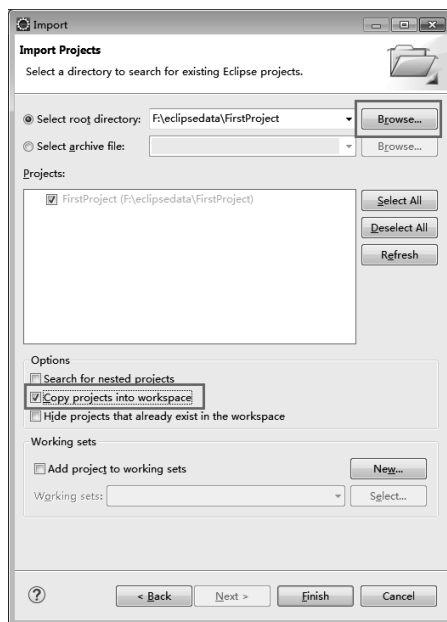


图1-26

(4) 单击Finish按钮，导入项目成功。

注意：在导入项目之前，要保证在Eclipse中不存在重名的项目。

1.9.5 在Eclipse中给main方法传递参数

在命令行使用java命令，可以将多个参数通过空格分开后传递给main方法。在Eclipse中也有同样的传递参数的位置。选择菜单项Run As→Run Configurations→Arguments打开配置参数的窗口，在Program arguments下添加参数，如图1-27所示。

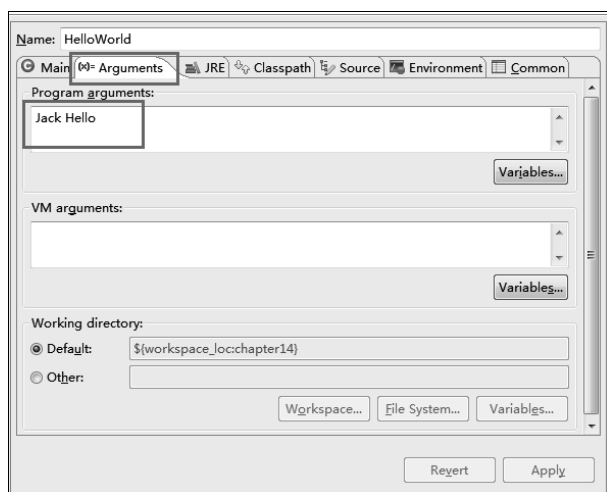


图1-27

1.9.6 Eclipse的快捷键

Eclipse中有很多快捷键，它将让你的程序开发变得快步如飞。建议读者经常使用这些快捷键，从而达到运用自如的地步。

在Eclipse中，输出System.out.println("")时只要打出sysout或者syso+Alt+/即可补全所有代码。Eclipse中常用的快捷键说明如下：

- (1) Ctrl+Space: 提供对方法、变量、参数、javadoc等信息的提示，应用在多种场合。总之，需要提示的时候可先按此快捷键。
- (2) Ctrl+Shift+Space: 变量提示。
- (3) Ctrl+/: 添加/消除//注释，在Eclipse 2.0中，消除注释为Ctrl+\。
- (4) Ctrl+Shift+/: 添加/* */注释。
- (5) Ctrl+Shift+\: 消除/* */注释。
- (6) Ctrl+Shift+F: 自动格式化代码。
- (7) Ctrl+l: 批量修改源代码中的变量名。此外，还可用在catch块上。
- (8) Ctrl+F6: 界面切换。
- (9) Ctrl+Shift+M: 查找所需要的包。
- (10) Ctrl+Shift+O: 自动引入所需要的包。
- (11) Ctrl+Alt+S: 源代码的快捷菜单。
- (12) Alt+/: 内容辅助。

更多快捷键可以参考 Eclipse 的官方网站，或者通过图 1-28 所示的界面去了解默认的快捷键。

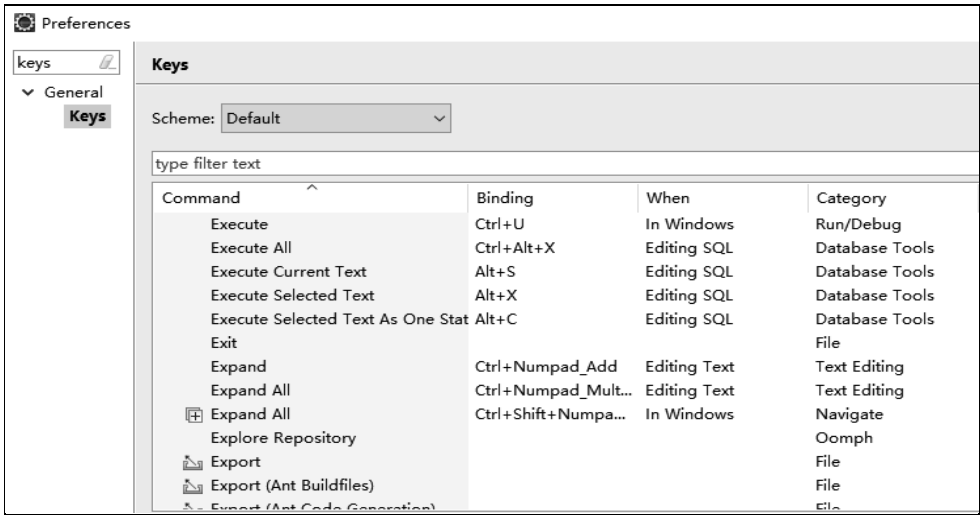


图1-28

1.10 实训1：商超购物管理系统欢迎界面

1. 需求说明

打印输出商超购物管理系统的欢迎界面和商品维护菜单的显示界面，界面参看专题一的专题一图1和专题一图2所示。

2. 训练要点

熟练使用Java开发工具，如Eclipse等，学会创建Java项目，能够正确运行程序。

3. 实现思路

- (1) 打开Eclipse，创建Java项目。
- (2) 编写代码，打印输出商超购物管理系统的欢迎界面。

4. 解决方案及关键代码

```
1. package cn.oracle;
2. public class Shop1 {
3.     public static void main(String[] args) {
4.         System.out.println("\n\n\t\t\t 商超购物管理系统\n");
5.         System.out.println("*****");
6.         System.out.println("\t\t\t\t\t1. 商品维护\n");
7.         System.out.println("\t\t\t\t\t2. 前台收银\n");
8.         System.out.println("*****");
9.         System.out.println("请选择，输入数字或按 0 退出: ");
10.    }
11. }
```

1.11 本章总结

Java语言是目前最受企业欢迎的编程语言之一，有许多相关的工作岗位。无论从事哪个岗位，Java的核心技术都是基础。JDK是Java必备的开发工具。Java是跨平台的语言，JDK并不跨平台，要根据不同的操作系统选择不同的JDK版本。Java是运行在JRE里面的。

安装JDK后，需要配置两个环境变量：JAVA_HOME和PATH。

使用java -version命令可以检查当前JDK是否安装成功，并且可以显示版本信息。

Java源程序就是扩展名为*.java的文本文件。在*.java源文件开发完成以后，通过javac命令将*.java文件编译成*.class字节码文件，然后通过java命令运行Java的字节码文件。

Eclipse是一个集成开发工具，经常使用就会熟练。需要说明的是，只有掌握了Java代码的运行和编译才是关键。这样，无论使用什么Java开发工具，都可以信手拈来。

1.12 课后练习

1. 简述在Eclipse中Java项目src、bin目录的含义。
2. 解释什么是JDK和JRE，并说明二者的区别。
3. () 可以将HelloWorld.java编译成HelloWorld.class。
A. java HelloWorld.java B. javac HelloWorld.java
C. java -d . HelloWorld.java D. javac HelloWorld
4. 若HelloWorld.java保存在package cn.oracle;包结构声明中，则 () 可以正常运行这个Java程序。
A. javac cn.oracle.HelloWorld B. java cn/oracle/HelloWorld
C. java cn.oracle.HelloWorld D. java HelloWorld
5. () 是生成标准文档的命令。
A. javac B. java C. doc D. javadoc
6. 开发Java程序必须安装 () 环境。
A. JRE B. JDK C. PATH D. JAVA_HOME
7. JAVA_HOME一般配置成 () 。
A. JDK的安装目录 B. JRE的安装目录
C. 当前目录 D. C盘的根目录
8. () 是Eclipse中Java项目中存放Java源代码的目录。
A. bin B. src C. jre D. java