

高等学校计算机应用规划教材

Java 开发实例教程

石 磊 张 艳 主 编

吕雅丽 陶永才 副主编

清华大学出版社

北 京

内 容 简 介

本书全面讲述 Java 程序设计的相关知识。全书共分为 11 章, 深入介绍 Java 的起源和发展, Java 语言的特点, Java 软件开发包的安装方法和环境变量的配置方法, 流程控制语句, 面向对象编程的相关知识, 如类、对象、继承等, Java 常用类库及异常处理, Swing 图形用户界面, 多线程机制, 数据库编程和网络编程等内容。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实, 可作为高等院校软件开发相关专业、计算机科学与技术专业的教材, 还可作为 Java 应用开发人员的参考资料。

本书的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn> 网站下载。

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。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Java 开发实例教程 / 石磊, 张艳 主编. —北京: 清华大学出版社, 2017

(高等学校计算机应用规划教材)

ISBN 978-7-302-47983-3

I. ①J… II. ①石… ②张… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 201710 号

责任编辑: 胡辰浩 李维杰

装帧设计: 孔祥峰

责任校对: 成凤进

责任印制:

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: 010-62770175 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

课 件 下 载: <http://www.tup.com.cn>, 010-62781730

印 刷 者:

装 订 者:

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 20.5 字 数: 473 千字

版 次: 2017 年 9 月第 1 版 印 次: 2017 年 9 月第 1 次印刷

印 数: 1~4000

定 价: 48.00 元

产品编号:

前 言

信息技术的飞速发展大大推动了社会的进步，已经逐渐改变了人类的生活、工作、学习等方式。Java 语言是当今流行的网络编程语言，它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的概念，使万维网从最初的单纯提供静态信息发展到现在的能够提供各种各样的动态服务。

Java 的出现迅速引起了 IT 业和工业界的高度重视。Java 提供的强大的图形、图像、动画、音频、视频、多线程和网络交互能力，使它在设计交互式程序、多媒体网页和网络应用方面大显身手。目前，Java 已经成为最卓越的程序设计语言之一。

Java 不仅是一种程序设计语言，更是现代软件编程技术的基础。Java 还是未来新型 OS 的核心，未来将会出现 Java 芯片。Java 将构成各种应用软件的开发平台与实现环境，成为人们必不可少的开发工具，有着广泛的应用前景。

本书从 Java 的起源出发，由浅入深地详细讲述了 Java 的起源和发展，Java 语言的特点，Java 软件开发包的安装方法和环境变量的配置方法，流程控制语句，面向对象编程的相关知识，如类、对象、继承等，Java 常用类库及异常处理，Swing 图形应用界面，多线程机制，数据库编程和网络编程等，并且运用大量实例对各种关键技术进行深入浅出的分析，注重培养读者解决实际问题的能力并快速掌握利用 Java 语言进行实际开发的基本操作技术。

每一章的引言部分概述了该章的作用和内容。在每一章的正文中，结合所讲述的关键技术和难点，穿插了大量极富实用价值的示例。每一章末尾都安排了有针对性的思考和练习，思考题有助于读者巩固所学的基本概念，练习题有助于培养读者的实际动手能力，增强对基本概念的理解和实际应用能力。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实，可作为高等院校软件开发相关专业、计算机科学与技术专业的教材，还可作为 Java 应用开发人员的参考资料。

本书共 11 章，其中第 1 章由石磊编写，第 2 章和第 4 章由张艳和巴阳编写，第 3 和第 5 章由吕雅丽编写，第 6 章由陶永才编写，第 7 章由李世科编写，第 8 章和第 11 章由董俊磊和巴阳编写，第 9 章由庞海波编写，第 10 章由张亚利编写。石磊和张艳负责全书的统稿和校对工作。

本书是集体智慧的结晶，参加本书编写的人员还有石育澄、邵玉梅、火昊、赵国桦、丁鑫、贾圣杰、任鹏程、郭华杰、程秋香、赵香玉、郑丽丽、张晓菊、卢华林、吴歌等。由于作者水平有限，本书难免有不足之处，欢迎广大读者批评指正。我们的 email 是 huchenhao@263.net，电话是 010-62796045。

在编写过程中，我们也参考和采纳了国内外大量专家学者的著作和其他形式的研究成果，在此一并向他们表示深深的谢意！

本书对应的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn> 网站下载。

作 者
2017 年 6 月

目 录

第 1 章	Java 语言入门	1
1.1	概述	1
1.1.1	Java 的起源与发展	1
1.1.2	Java 与 C 和 C++ 的关系	2
1.1.3	Java 语言的特点	3
1.2	搭建 Java 开发与运行环境	5
1.2.1	常用软件包功能	5
1.2.2	安装 JDK 环境	6
1.2.3	配置 Java 开发与运行环境	8
1.2.4	安装开发平台 Eclipse	11
1.3	编写第一个 Java 程序	12
1.3.1	利用记事本编写 Java 程序 并运行	12
1.3.2	利用开发环境 Eclipse 建立 和运行 Java Application 源程序	13
1.4	本章小结	18
1.5	思考和练习	18
第 2 章	Java 语言基础	20
2.1	认识 Java 程序	20
2.1.1	类	20
2.1.2	分号和定位	21
2.1.3	缩进原则	21
2.1.4	Java 关键字	21
2.1.5	Java 标识符	22
2.1.6	常量、变量及其赋值	22
2.1.7	分隔符	25
2.2	Java 的数据类型	26
2.2.1	整型	26
2.2.2	浮点型	27
2.2.3	字符型	28

2.2.4	布尔型	29
2.3	变量	30
2.3.1	变量的声明	30
2.3.2	动态初始化	30
2.4	数据类型转换	31
2.4.1	自动类型转换	31
2.4.2	强制类型转换	32
2.5	运算符	33
2.5.1	赋值运算符	34
2.5.2	算术运算符	34
2.5.3	自增与自减运算符	35
2.5.4	位运算符	37
2.5.5	关系运算符和逻辑运算符	38
2.5.6	运算符的优先级	40
2.6	表达式	40
2.6.1	表达式	41
2.6.2	表达式的类型转换变量	42
2.7	数组	42
2.7.1	一维数组	42
2.7.2	多维数组	44
2.7.3	不规则数组	46
2.8	本章小结	47
2.9	思考和练习	48
第 3 章	流程控制语句	49
3.1	选择语句	49
3.1.1	if 语句	49
3.1.2	if-else 语句	50
3.1.3	嵌套 if 语句	51
3.1.4	switch 语句	52
3.2	循环语句	54
3.2.1	for 循环	54
3.2.2	while 循环	56

3.2.3 do-while 循环	58	4.10 接口	92
3.3.4 嵌套循环	59	4.10.1 接口的定义	92
3.3 跳转语句	60	4.10.2 接口的实现	93
3.3.1 break 语句	60	4.11 本章小结	94
3.3.2 continue 语句	60	4.12 思考和练习	95
3.3.3 return 语句	61	第 5 章 常用类库	96
3.4 本章小结	62	5.1 文件的结构	96
3.5 思考和练习	62	5.1.1 文件的分割	96
第 4 章 面向对象编程	64	5.1.2 使用包	98
4.1 类和对象	64	5.2 Java 常用类库	100
4.1.1 类的概念	64	5.2.1 Java 常用类库	100
4.1.2 类的定义格式	65	5.2.2 字符串	101
4.2 创建对象	71	5.2.3 字符串特殊操作	104
4.2.1 对象创建格式	71	5.2.4 提取字符	107
4.2.2 对象的使用	72	5.2.5 比较字符串	108
4.2.3 对象的生命周期	76	5.2.6 查找字符串	113
4.3 构造函数	77	5.2.7 修改字符串	114
4.3.1 简单构造函数	77	5.2.8 使用 valueOf()转换数据	116
4.3.2 带形参的构造函数	78	5.2.9 改变字符串中字符的 大小写	117
4.4 析构函数	80	5.2.10 连接字符串	118
4.5 this 关键字	80	5.2.11 其他 String 方法	118
4.5.1 局部变量和成员变量 同名的情况	81	5.3 StringBuffer 类	119
4.5.2 在构造函数中调用其他 构造函数	83	5.3.1 创建 StringBuffer 对象	120
4.6 垃圾回收	84	5.3.2 StringBuffer 类的常用方法	120
4.7 finalize()方法	85	5.4 Math 类	127
4.8 数据的封装	85	5.5 Object 类	132
4.8.1 包的概念	85	5.6 本章小结	132
4.8.2 包的定义	86	5.7 思考和练习	132
4.8.3 包的引入	87	第 6 章 异常处理	134
4.8.4 访问权限修饰符	87	6.1 异常处理的基础知识	134
4.9 类的继承和多态	89	6.1.1 异常的产生	134
4.9.1 继承的基本概念	89	6.1.2 异常类型	135
4.9.2 继承的声明格式	89	6.1.3 异常处理机制	136
4.9.3 变量的继承和隐藏	89	6.1.4 异常的捕获和抛出	138
		6.1.5 多重 catch 语句	139

6.1.6 try 语句的嵌套	140	7.7.2 容器	182
6.1.7 finally 语句	141	7.7.3 顶级容器窗格	183
6.2 Java 的内置异常	143	7.8 本章小结	183
6.3 自定义异常类	145	7.9 思考和练习	183
6.4 本章小结	149	第 8 章 多线程程序设计	187
6.5 思考和练习	149	8.1 多线程基础	187
第 7 章 图形用户界面	151	8.2 Thread 类和 Runnable 接口	188
7.1 图形界面开发工具	151	8.3 创建一个线程	188
7.2 AWT 概述	152	8.4 创建多个线程	191
7.3 AWT 容器	153	8.5 线程的生命周期	193
7.3.1 Frame(框架)	153	8.6 线程的优先级	196
7.3.2 Panel(面板)	154	8.7 同步	199
7.4 AWT 基本组件	155	8.7.1 同步方法	199
7.4.1 标签(Label)	155	8.7.2 同步语句	202
7.4.2 按钮(Button)	156	8.8 线程间通信	203
7.4.3 文本框(TextField)	157	8.8.1 notify()、wait()和 notifyAll()的线程通信	204
7.4.4 文本输入区域(TextArea)	157	8.8.2 wait()和 notify()的示例	204
7.4.5 下拉列表(Choice)	158	8.9 线程状态	209
7.4.6 列表(List)	159	8.10 本章小结	212
7.4.7 复选框(Checkbox)	160	8.11 思考和练习	212
7.4.8 单选按钮组 (CheckboxGroup)	160	第 9 章 JAVA 的 I/O	215
7.4.9 滚动条	161	9.1 Java 的 I/O 系统	215
7.4.10 AWT 组件综合案例	161	9.2 字节流和字符流	215
7.5 布局管理器	163	9.3 字节流类	216
7.5.1 FlowLayout	163	9.4 字符流类	217
7.5.2 BorderLayout	164	9.5 预定义流	217
7.5.3 GridLayout	166	9.6 字节流读写控制台	218
7.5.4 CardLayout	167	9.6.1 读控制台	218
7.5.5 CardBagLayout	169	9.6.2 写控制台	219
7.6 AWT 事件处理	172	9.7 字节流读写文件	220
7.6.1 事件处理机制	172	9.7.1 读文件	221
7.6.2 事件适配器	174	9.7.2 写文件	224
7.6.3 常用事件的分类	176	9.8 关闭文件	226
7.7 Swing	181	9.9 读写二进制数据	228
7.7.1 组件	181	9.10 随机访问文件	231

9.11	Java 字符流应用	233	11.1.2	网络编程概述	294
9.11.1	字符流的控制台输入	234	11.1.3	网络通信方式	295
9.11.2	字符流的控制台输出	237	11.1.4	InetAddress 类简介	296
9.12	字符流的文件 I/O	238	11.2	基于 URL 的网络编程	298
9.12.1	使用 FileWriter	238	11.2.1	统一资源定位器 URL	298
9.12.2	使用 FileReader	238	11.2.2	URL 的组成	298
9.13	Java 的类型封装器	239	11.2.3	创建一个 URL	298
9.14	本章小结	239	11.2.4	解析一个 URL	299
9.15	思考和练习	240	11.2.5	从 URL 读取 WWW 网络资源	300
第 10 章	数据库编程	242	11.2.6	通过 URLConnetction 连接 WWW	300
10.1	JDBC 简介	242	11.3	基于 Socket 的网络编程	301
10.2	建立数据库连接	244	11.3.1	Socket 通信	301
10.2.1	使用 JDBC-ODBC 桥接 驱动程序	244	11.3.2	Socket 通信过程	302
10.2.2	使用 JDBC 驱动程序	246	11.3.3	创建 Socket	302
10.2.3	使用属性文件	248	11.3.4	客户端 Socket	303
10.3	执行 SQL 语句	250	11.3.5	服务器端 ServerSocket	304
10.3.1	executeUpdate	251	11.3.6	打开输入/输出流	304
10.3.2	executeQuery	251	11.3.7	关闭 Socket	305
10.3.3	executeBatch	253	11.3.8	简单的 Client/Server 程序设计	305
10.4	使用 PreparedStatement	254	11.3.9	支持多客户端的 Client/Server 程序设计	308
10.5	事务处理	256	11.4	数据报通信	310
10.5.1	什么是事务	256	11.4.1	什么是数据报	310
10.5.2	一个关于事务的案例	257	11.4.2	数据报通信的表示方法: DatagramSocket 和 DatagramPacket	310
10.5.3	事务提交模式	257	11.4.3	基于 UDP 的简单的 Client/Server 程序设计	311
10.5.4	事务撤消	258	11.4.4	用数据报进行广播 通信	314
10.6	编写数据库工具类	259	11.5	本章小结	317
10.7	一个例子	265	11.6	思考和练习	317
10.7.1	数据库	265	参考文献		319
10.7.2	布局及功能简介	266			
10.7.3	源代码	267			
10.8	本章小结	291			
10.9	思考和练习	291			
第 11 章	网络编程	292			
11.1	网络编程的基本概念	292			
11.1.1	计算机网络的基本概念	292			

第1章 Java语言入门

Java 语言是当今流行的网络编程语言，它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的概念，使万维网从最初的单纯提供静态信息发展到现在的能够提供各种各样的动态服务。Java 不仅能够通过编写小应用程序实现嵌入网页的声音和动画功能，而且还能够应用于独立的大中型应用程序，其强大的网络功能可以把整个 Internet 作为一个统一的运行平台。

本章学习目标：

- 了解 Java 语言的起源和发展
- 了解 Java 语言的特点
- 了解 Java 语言的主要用途
- 熟悉 Java 常用软件包的功能
- 掌握 Java 软件的下载、安装与配置

1.1 概 述

1.1.1 Java 的起源与发展

驱使计算机语言革新的因素有两个：程序设计技术的改进和计算环境的改变。Java 也不例外。在大量继承 C 和 C++ 的基础之上，Java 还增加了反映当前程序设计技术状态的功能与精华。针对在线环境的蓬勃发展，Java 为高度的分布式体系结构提供了流水线程序设计的功能。

任职于 Sun 公司的 James Gosling 等人于 1990 年初开发了 Java 语言的雏形，最初被命名为 Oak，目标设定在家用电器等小型系统的程序语言，应用于电视机、电话、闹钟、烤面包机等家用电器的控制和通信。由于这些智能化家电市场的需求没有预期那么高，Sun 公司放弃了该项计划。随着 20 世纪 90 年代互联网的发展，Sun 公司看见 Oak 在互联网应用的前景，于是改造了 Oak，于 1995 年 5 月以 Java 的名称正式发布。Java 伴随着互联网的迅猛发展而发展，逐渐成为重要的网络编程语言。

Java 曾被美国的著名杂志 *PC Magazine* 评为 1995 年十大优秀科技产品，之所以称其为“革命性编程语言”，是因为用 Java 语言编写的软件能在任何安装了 Java 虚拟机的操作系统上执行。

Java 技术的快速发展得益于 Internet 的广泛应用，Internet 上有各种不同的计算机，它

们可能使用完全不同的操作系统和 CPU 芯片，但仍希望运行相同的程序，而 Java 的出现大大推动了分布式系统的快速开发和应用。

Java 的出现迅速引起了 IT 业和工业界的高度重视。Java 提供的强大的图形、图像、动画、音频、视频、多线程和网络交互能力，使它在设计交互式程序、多媒体网页和网络应用方面大显身手。目前，Java 已经成为最卓越的程序设计语言之一。

Java 不仅是一种程序设计语言，更是现代软件编程技术的基础。Java 还是未来新型 OS 的核心，未来将会出现 Java 芯片。Java 将构成各种应用软件的开发平台与实现环境，成为人们必不可少的开发工具。Java 语言有着广泛的应用前景，例如：

- 所有面向对象应用的开发，包括面向对象的事件描述、处理和综合等。
- 计算过程的可视化，可操作化软件的开发。
- 动态画面的设计，包括图形图像的调用。
- 交互操作的设计(选择交互、定向交互和控制流程等)。
- Internet 的系统管理功能模块，包括 Web 页面的动态设计、管理和交互操作设计等。
- Intranet(企业内部网)上的软件开发(直接面向企业内部用户的软件开发)。
- 与各类数据库连接查询的 SQL 语句实现。
- 其他应用类型的程序。

1.1.2 Java 与 C 和 C++的关系

Java 与 C 和 C++直接相关。Java 继承了 C 的语法，Java 的对象模型是从 C++改编而来的。Java 与 C 和 C++的关系之所以重要，是出于以下几个原因：

首先，许多程序员都熟悉 C/C++语法。这样对于他们而言，学习 Java 就简单多了。同样，Java 程序员学习 C/C++也是很简单的。

其次，Java 设计者并没有做重复工作。相反，他们进一步对已经成功的程序设计范式进行了提炼。现代程序设计始于 C，而后过渡到 C++，现在则是 Java。通过大量的继承和进一步的构建，Java 提供了强大的、可以更好利用已有成果的、逻辑一致的程序设计环境，并且增加了在线环境需求的新功能。然而，最重要的一点或许在于，由于它们的相似性，C、C++和 Java 为专业程序员定义了统一的概念架构。程序员从其中一种语言转为另一种语言时，不会遇到太大的困难。

C 和 C++的核心设计原理之一就是程序员拥有控制权。Java 也继承了这一原理。除了 Internet 环境施加的约束以外，Java 为程序员提供了完全的控制权。如果程序编写得出色，就会体现出来；而如果较糟糕，也会体现出来。换句话说，Java 并不是一种教学式语言，而是为专业程序员准备的语言。

Java 还有与 C 和 C++共有的特性：都由真正的程序员设计、测试和修改，与设计者的需求和经验紧密结合。因此，再没有比这更好的方法来创建如此一流的专业程序设计语言了。

因为 Java 与 C++的相似性，特别是它们对面向对象程序设计的支持，所以有些程序员可能会将 Java 简单地看成“C++的 Internet 版”。然而，这种观点是错误的。因为 Java 在

实际应用以及基本原理上与 C++ 有着显著的区别。尽管 Java 受到 C++ 的影响，但它绝不是 C++ 的增强版。例如，Java 不提供对 C++ 的向上或向下兼容。当然，Java 与 C++ 相似是十分明显的，如果你是一名 C++ 程序员，那么在使用 Java 时会有驾轻就熟的感觉。另外，Java 不是为替代 C++ 而设计的，而是为了解决一系列特定问题而设计的。C++ 则用来解决另一个不同系列的问题。两者将在未来共存。

1.1.3 Java 语言的特点

Java 语言适合用来开发网络上的应用程序。作为较晚问世的程序设计语言，Java 集中体现并充分利用了现代软件技术的新成果，如面向对象、多线程等。Java 语言的特点如下：

1. 简单性

Java 是一种面向对象的语言，它通过提供最基本的方法来完成指定任务，只需要理解一些基本概念就可以使之适合于开发各种情况的应用程序。Java 语言与 C++ 语言的风格相似，出于安全性考虑，Java 略去了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外，Java 也适合在小型机上运行，它的基本解释器及类的支持只有 40KB 左右，加上标准类库和线程的支持也只有 215KB 左右。

2. 面向对象

面向对象技术较好地解决了当今软件开发过程中出现的各种面向过程语言不能处理的问题，包括开发规模的扩大、升级加快、维护难度加大等。面向对象技术是按人们的思维方式建立问题空间模型，利用类和对象的机制将数据及操作封装在一起，通过统一的接口与外界交互。在面向对象技术中，对象和消息表示事物之间的相互联系，类和继承是适应人们一般思维方式的描述方式，方法表示作用于该类对象上的各种操作。

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

3. 分布性

Java 是面向网络的语言。它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

4. 鲁棒性

Java 在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时产生错误。通过集成的面向对象的例外处理机制，在编译时，Java 提示可能出现但未被处理的例外，帮助程序员正确进行选择以防止系统崩溃。另外，Java 在编译时还可捕获类型声明中许多常见的错误，防止动态运行时不匹配问题的出现。

5. 安全性

Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁和破坏。用于网络、分布式环境的 Java 必须要防止病毒的侵入，Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

6. 体系结构中立

对于 Java 设计人员来说，核心问题是程序代码的持久性和可移植性。在创建 Java 时，程序员面临的一个主要问题是，即使是在同一台机器上也不能保证今天编写的程序到了明天仍然能够运行。操作系统升级、处理器升级以及核心系统资源的变化，都可能导致程序出现故障。Java 设计人员对 Java 语言做出了一些艰难的决策，Java 虚拟机就是试图用于解决这个问题的。它们的目标是“编写一次，无论何时、何地都能永远运行”。在很大程度上，这个目标已经实现了。

7. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器上。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语言实现的，Java 运行时系统由标准 C 语言实现，这使得 Java 系统本身也具有可移植性。

8. 解释执行

Java 是解释型语言，它编译后并不生成特定的 CPU 机器代码，而是生成一种被称为字节码的目标代码。Java 解释器可以直接运行字节码指令，对其进行解释执行。Java 虚拟机使 Java 字节码指令可以在某一特定软硬件平台环境中直接运行，而工程人员不必考虑运行环境。

9. 高性能

和其他解释执行的语言(如 BASIC、TCL)不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

10. 多线程

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不必采用全局的事件循环机制，这样就会很容易实现网络上的实时交互行为。

11. 动态性

Java 利用了面向对象技术的优点，新的或升级的库函数不需要更改源程序就能正确运行。

Java 的设计使它适合那种不断发展的环境。在 Java 类中可以自由地加入新的方法和实

例变量而不会影响用户程序的执行，并且 Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

1.2 搭建 Java 开发与运行环境

1.2.1 常用软件包功能

1. SDK

为了开发 Java 程序，计算机上必须安装 Java SDK(Software Development Kit, 软件开发工具包)，它是用来辅助开发 Java 程序的相关文档、范例和工具的集合，是专门用于帮助开发人员提高工作效率的 Java 开发环境。它包括开发及运行 Java 所需要的主要软件。

2. JDK

Java SDK 最早被称为 Java Software Development Kit，后来改名为 JDK，它是 Java Development Kit 的缩写，中文称为 Java 开发工具包。它是整个 Java 的核心，是用来编写 Java Applet 和 Java 应用程序的开发环境。它由一个处于操作系统层之上的运行环境，以及翻译、调试和运行 Java 程序所需的工具组成。不论什么 Java 应用服务器，实质都是内置了某个版本的 JDK，JDK 是一切 Java 应用程序的基础，所有的 Java 应用程序都离不开 JDK。它包含所有对 Java 程序最有用的 Java 编译器、Applet 查看器和 Java 解释器。

在下载 Sun 公司提供的 Java EE SDK 软件包时，可以同时下载捆绑的 JDK 软件包，否则需要专门下载 JDK。如果仅仅是开发一些 Java 程序，使用 JDK 即可。

3. JRE

JRE(Java Runtime Environment)称为 Java 运行环境或 Java 平台。所有的 Java 程序都要在 JRE 下才能运行，JDK 开发工具是由 Java 程序组成的，也需要 JRE 才能运行。为了保持 JDK 的独立性和完整性，在 JDK 的安装中，JRE 也是安装的一部分。所以，在 JDK 的安装目录下有一个名为 jre 的目录，用于存放 JRE 文件。也就是说，JDK 中包含 JRE。

4. JVM

JVM(Java Virtual Machine, Java 虚拟机)是 JRE 的一部分。它是一个虚构出来的计算机，是通过在实际的计算机上模拟各种计算机的功能来实现的。JVM 有自己完善的硬件架构，如处理器、堆栈、寄存器等，还具有相应的指令系统。Java 语言最重要的特点就是跨平台运行。JVM 就是实现跨平台运行的主要工具。JRE 中包含 JVM。

5. Java 软件

为了在浏览器中运行 Java 小程序，需要在计算机中安装 Java 软件。这是一个特殊的

Java 软件包，它可以让浏览器支持 Java 程序的功能，它是 Java 程序与 Web 进行交互的工具。它包含 Java 虚拟机和许多其他内容。它可以通过浏览器尽情享受 Internet 提供的带有 Java 功能的最佳内容，包括游戏、体育、聊天、电子邮件、艺术、财务工具等。

1.2.2 安装 JDK 环境

为了开发 Java 程序，计算机上必须安装和配置 Java 开发环境，然而在编译并运行这些程序之前，必须在计算机上安装一个 Java 开发包(Java Development Kit, JDK)。JDK 是 Java Development Kit 的缩写，中文称为 Java 开发工具包，是一切 Java 应用程序的基础。JDK 支持两个主要程序：第一个是 Java 的编译器 `javac`；第二个是标准 Java 解释器 `java`，也称为 application launcher。

JDK 可以从 www.oracle.com/technetwork/java/javase/downloads/index.html 免费下载。本节以 Windows 版的 JDK 8 为例介绍 JDK 的安装过程。

(1) 双击下载的安装文件，如图 1-1 所示。

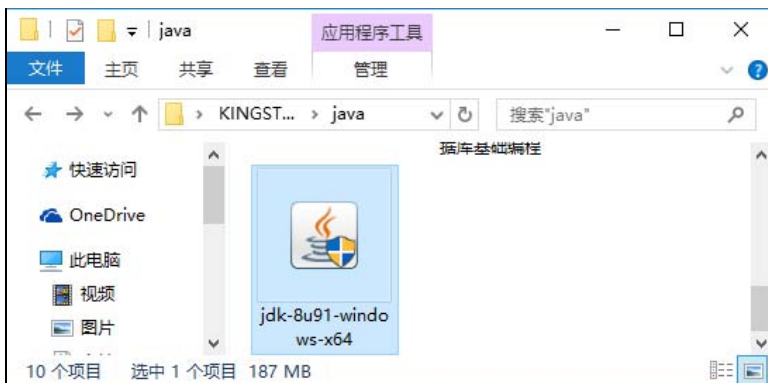


图 1-1 打开 JDK 安装包

(2) 进入安装向导，如图 1-2 所示。



图 1-2 安装向导界面

(3) 单击【下一步】按钮进入自定义安装界面，默认安装到“C:\Program Files\Java\jdk1.8.0_91\”目录下，如图 1-3 所示。可以通过【更改】按钮对安装路径进行自定义，这里我们使用默认的安装路径。



图 1-3 自定义安装界面

(4) 选择安装 JDK 所有组件后，单击【下一步】按钮进入安装进度界面，如图 1-4 所示。

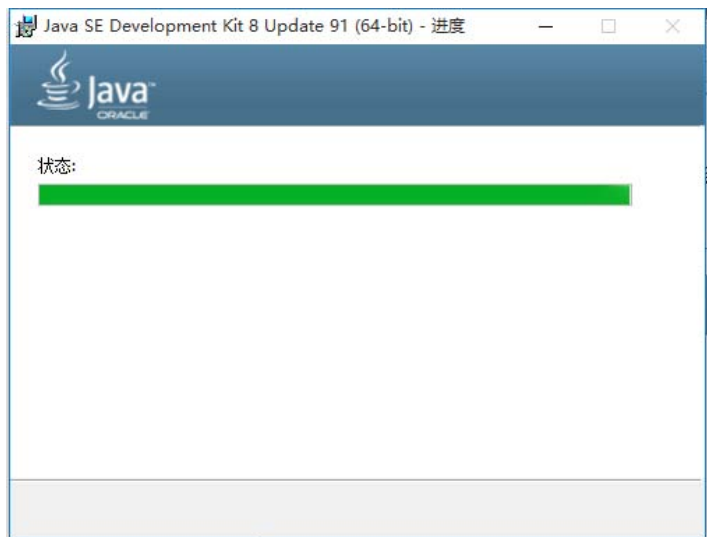


图 1-4 安装进度界面

(5) JDK 安装完成后，如图 1-5 所示，接下来可以配置 Java 开发环境。

安装成功后，可以看到基本工具将安装在 bin 文件中，如图 1-6 所示：包含编译器 (javac.exe)、解释器(java.exe)、Applet 查看器(appletviewer.exe)等可执行文件；基础应用类库将安装在 lib 文件夹中，包含了所有的类库以便于开发 Java 程序；sample 文件夹包含开源代码程序实例；src 压缩文件夹包含类库开源代码。



图 1-5 安装完成

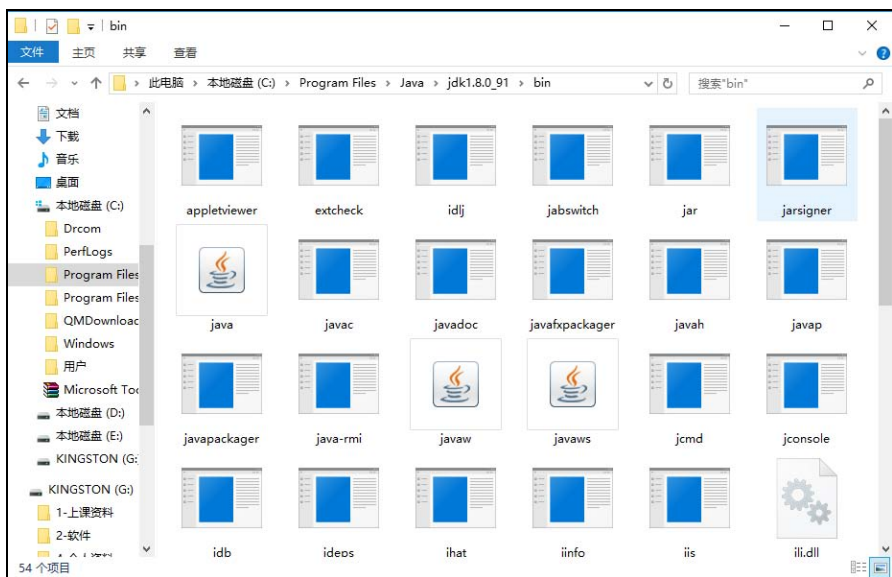


图 1-6 基本工具

1.2.3 配置 Java 开发与运行环境

为了方便使用 JDK 中的 Java 工具, 需要进行环境变量的配置。设置环境变量是为了能够正常使用所安装的开发包, 主要包括两个环境变量: Path 和 classpath。Path 称为路径环境变量, 用来指定 Java 开发包中的一些可执行程序所在的位置; classpath 环境变量, 用于告诉 Java 执行环境, 在哪些目录下可以找到所要执行的 Java 程序。

我们以 Windows 操作系统为例进行环境变量的设置。

(1) 右击【计算机】, 选择【属性】菜单, 弹出【系统属性】对话框, 选择【高级】选项卡, 如图 1-7 所示。



图 1-7 【系统属性】对话框的【高级】选项卡

(2) 在【高级】选项卡中单击【环境变量】按钮，将出现【环境变量】对话框，如图 1-8 所示。

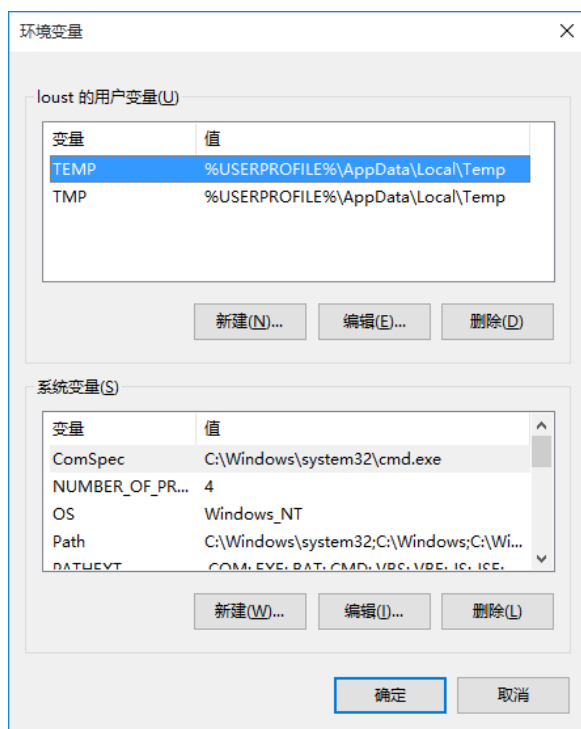


图 1-8 【环境变量】对话框

(3) 在【系统变量】列表框中选择【Path】，然后单击【编辑】按钮，在出现的【编辑系统变量】对话框中，在【变量值】栏添加“C:\Program Files\Java\jdk1.8.0_91\bin”（以 JDK 安装在 C 盘为例），如图 1-9 所示。

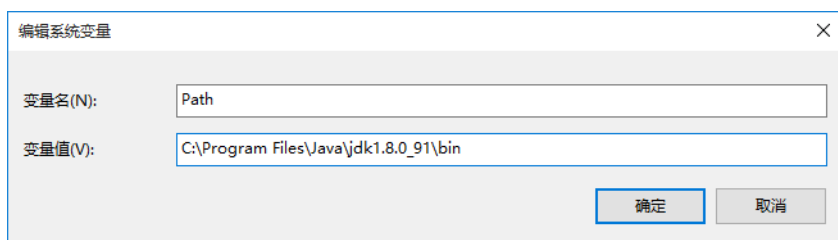


图 1-9 设置 Path 环境变量

(4) 在【系统变量】列表框中，单击【新建】按钮，在出现的【新建系统变量】对话框中，在【变量名】栏中输出“classpath”，在【变量值】栏输入“.;C:\Program Files\Java\jdk1.8.0_91\lib”，如图 1-10 所示。其中“.”表示当前目录。

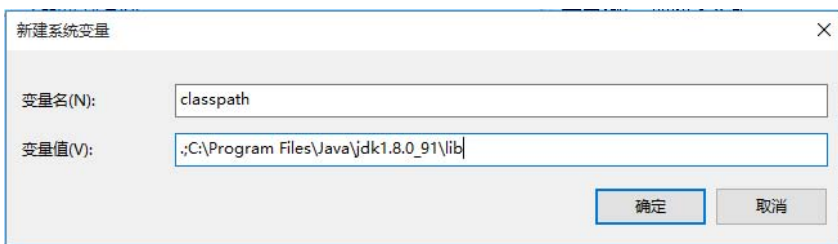


图 1-10 新建 classpath 环境变量

至此，完成环境变量的设置工作。设置完成后，单击【开始】|【所有程序】|【运行】，输入“cmd”后回车，打开 DOS 窗口。在命令行提示符下输入“javac”后回车，如果出现其用法参数提示信息，则安装正确，如图 1-11 所示。

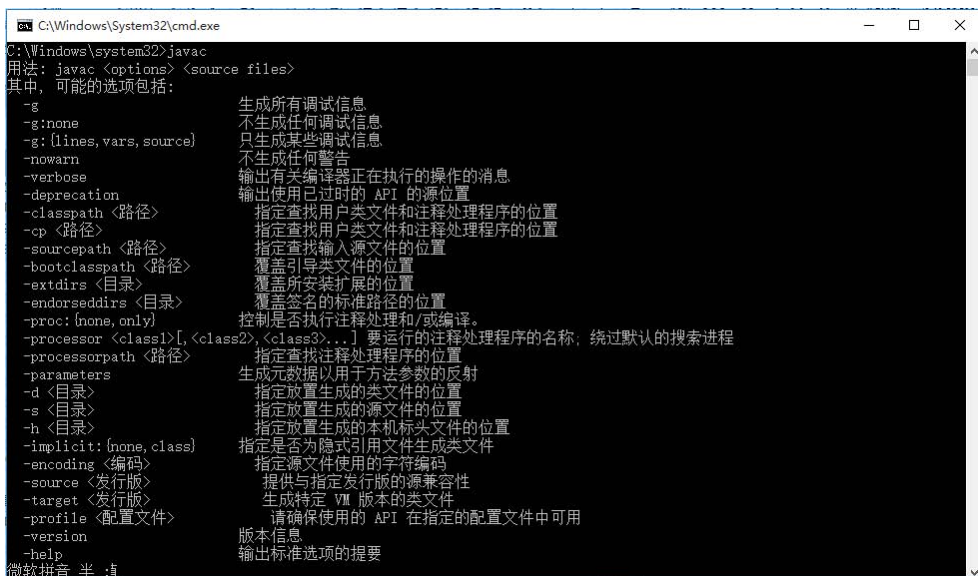


图 1-11 用法参数显示结果

1.2.4 安装开发平台 Eclipse

这里介绍 IBM 公司开发的 IDE 开发环境 Eclipse，它是一个开放源代码的、基于 Java 的可扩展开发平台。

进入 Eclipse 官方网站<http://www.eclipse.org/downloads>，下载软件，下载后解压安装包，如图 1-12 所示。双击运行 eclipse.exe 文件，启动安装过程，如图 1-13 所示，根据提示完成软件的安装。

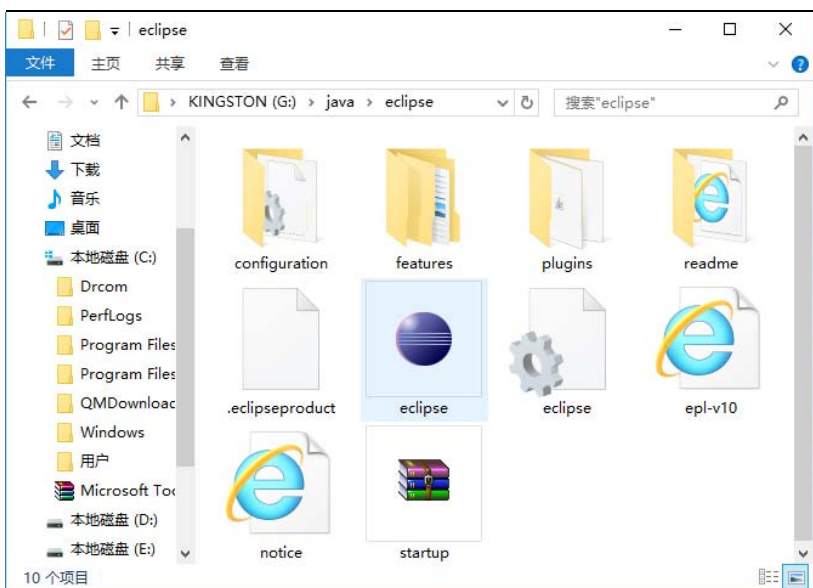


图 1-12 Eclipse 安装文件



图 1-13 Eclipse 安装界面

安装成功后，双击桌面图标，即可启动软件。

1. Eclipse 窗口的组成

Eclipse 启动后的窗口，主要分为 5 个部分：菜单栏、工具栏、包资源管理器、大纲视图及信息视图，如图 1-14 所示。

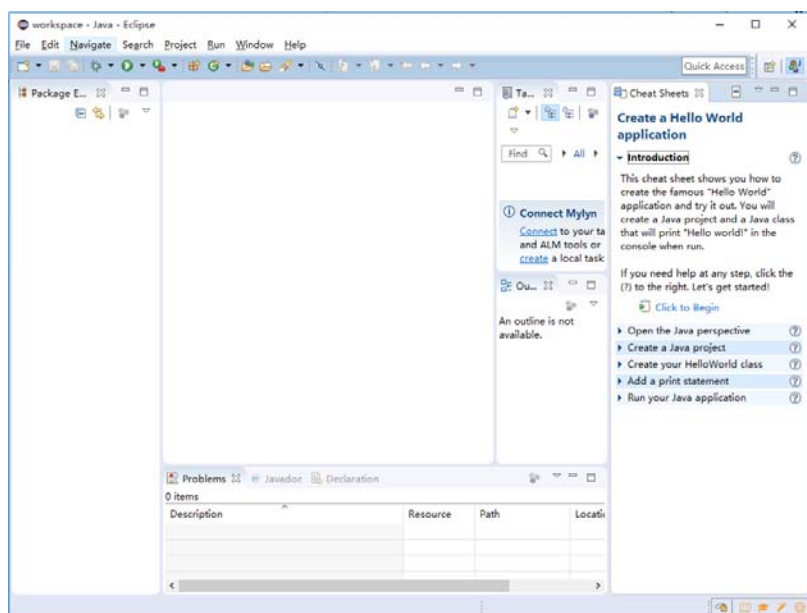


图 1-14 Eclipse 窗口的组成

1.3 编写第一个 Java 程序

1.3.1 利用记事本编写 Java 程序并运行

首先介绍以最简单的方式来编写、编译与运行 Java 程序。打开记事本，在文本编辑界面中键入程序，如图 1-15 所示，命名为“first.java”，将文件保存到 d:\java 目录中。注意：文件名必须和程序中所声明的类名一致，并且扩展名为“java”。

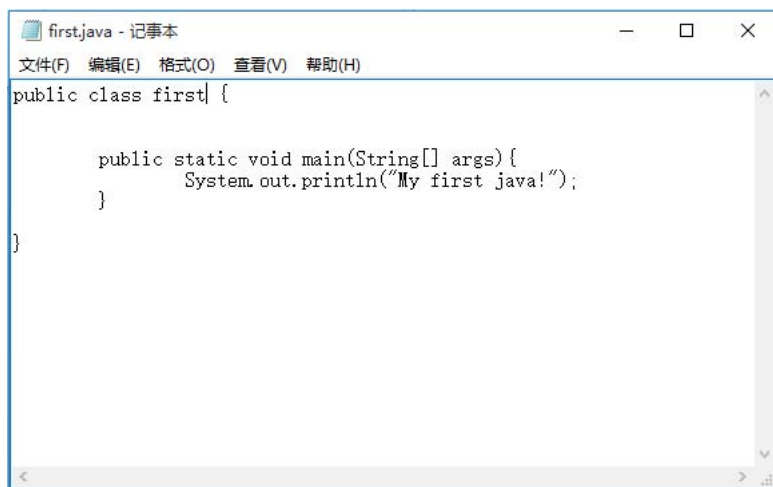


图 1-15 使用记事本编辑 Java 程序

文件保存成功后，从命令提示符窗口进入 d:\java 目录，在此目录下输入程序编译命令

javac: first.java。编译后目录下多了一个“first.class”文件，这是 javac 编译器将源代码编译成字节代码生成类文件的结果。

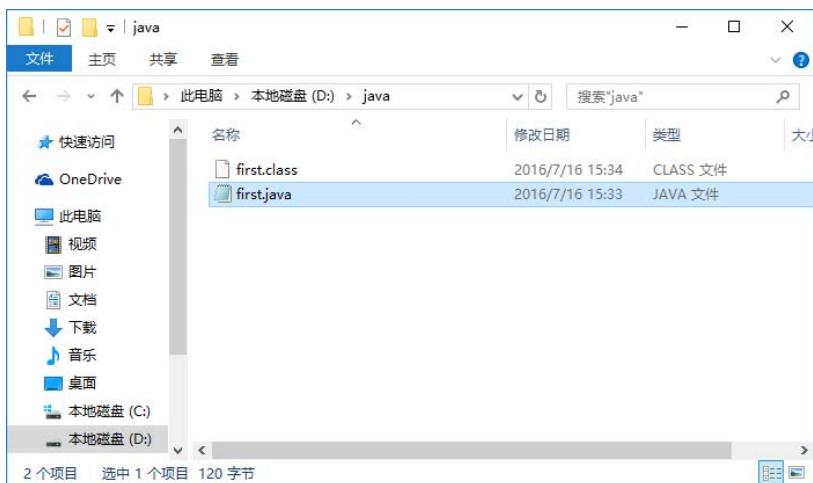


图 1-16 编译结果

再由 Java 解释，执行“first.class”类文件。输入程序运行命令：java first。程序运行后，输出“My first java! ”，如图 1-17 所示。

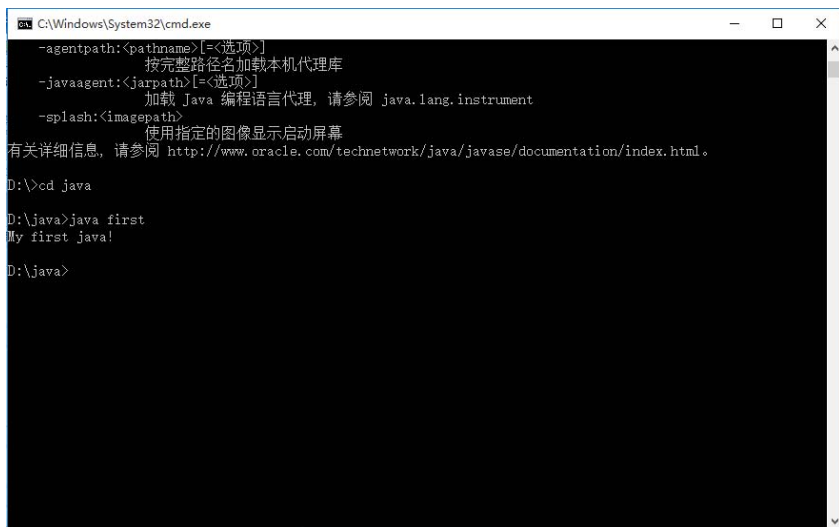


图 1-17 运行结果

1.3.2 利用开发环境 Eclipse 建立和运行 Java Application 源程序

具体步骤如下：

- (1) 建立工程，单击【File】|【New】|【Java Project】选项，如图 1-18 所示。

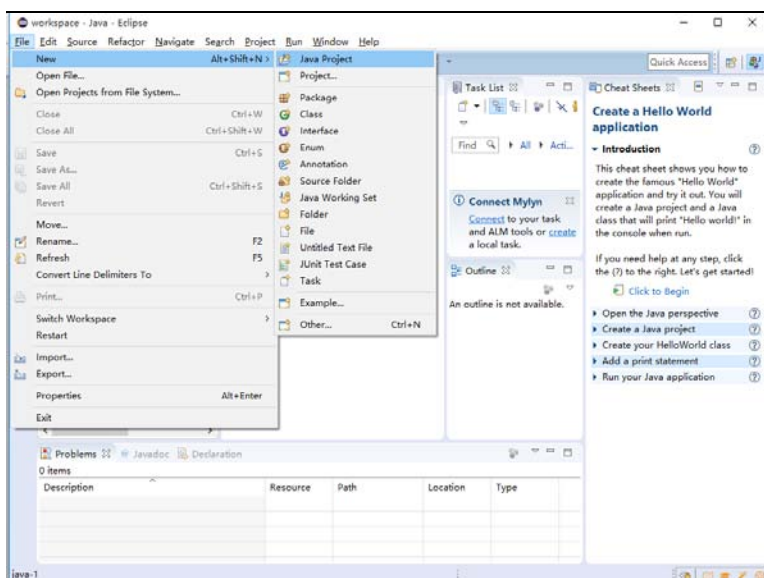


图 1-18 建立一个 Java 工程

(2) 打开图 1-19 所示窗口，在 Project name 文本框中输入工程名“java1-1”，然后单击【Finish】按钮，弹出图 1-20 所示窗口。

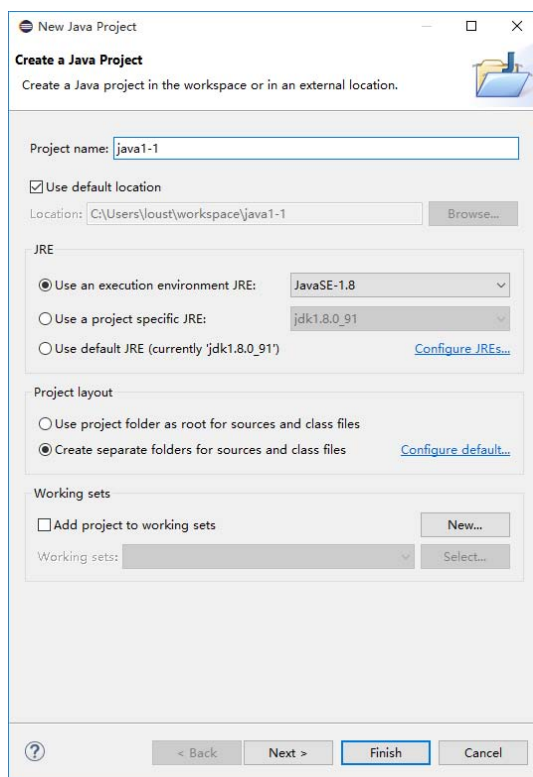


图 1-19 New Java Project 窗口

(3) 图 1-20 中显示了建好的工程，可在此操作界面中编写 Java 程序。

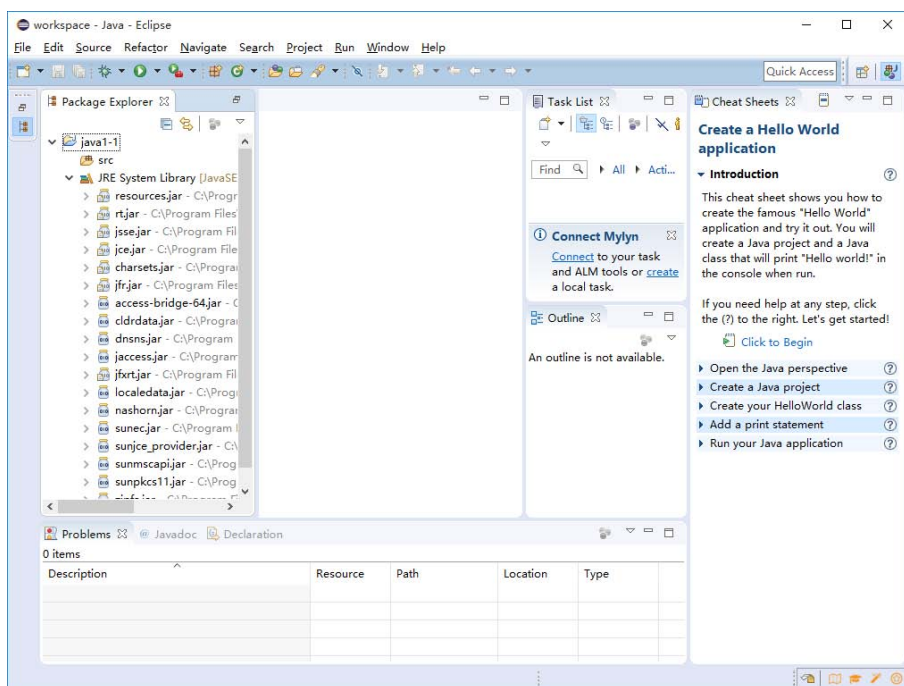


图 1-20 建好的工程窗口

(4) 单击【File】|【New】|【Class】选项，如图 1-21 所示。

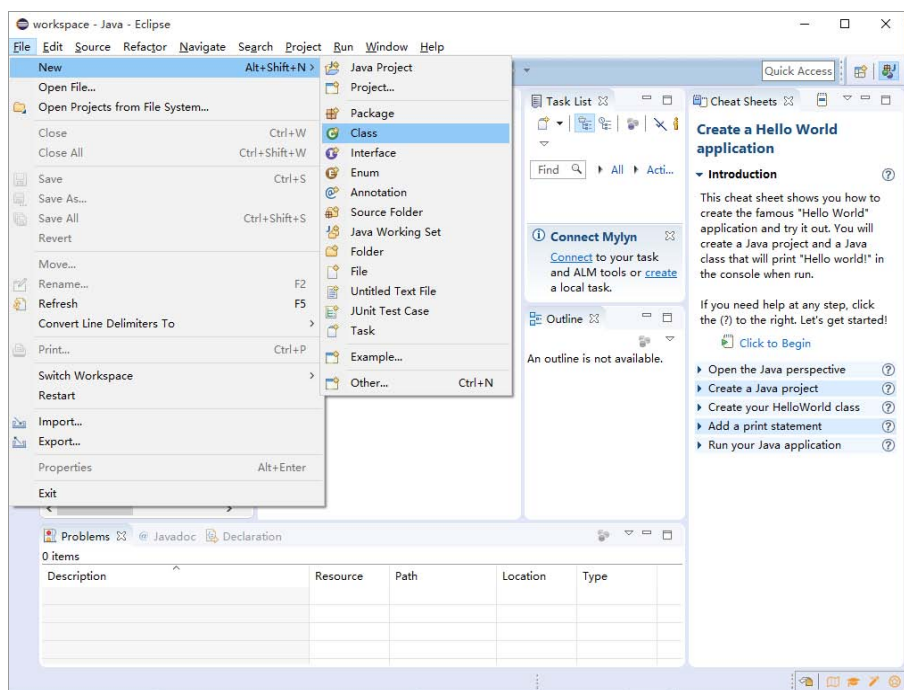


图 1-21 在建好的工程中建立 Java 源程序

(5) 打开图 1-22 所示窗口，在 Name 文本框中输入 Java 源程序名“helloworld”，扩展名不用写，默认为“.java”。然后单击【Finish】按钮，弹出图 1-23 所示窗口。

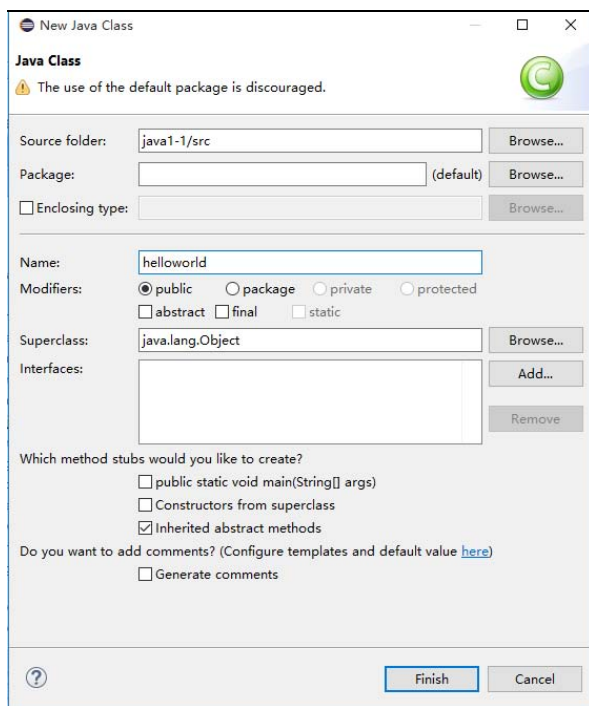


图 1-22 New Java Class 窗口

(6) 创建图 1-23 所示的空的 Java Application 应用程序框架，可在此编写程序，输入的源程序如图 1-24 所示。

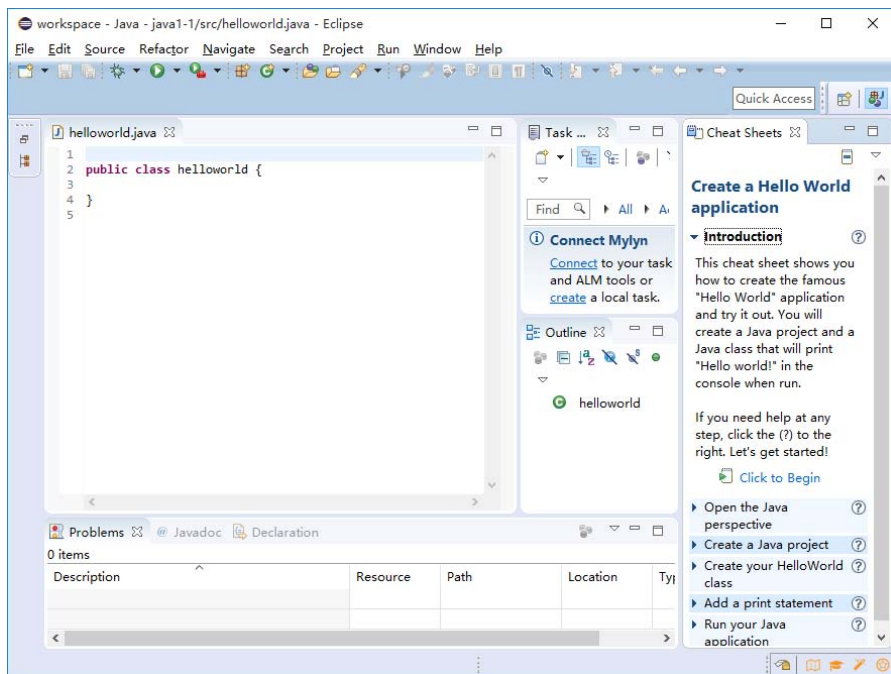


图 1-23 空的 Java Application 应用程序框架

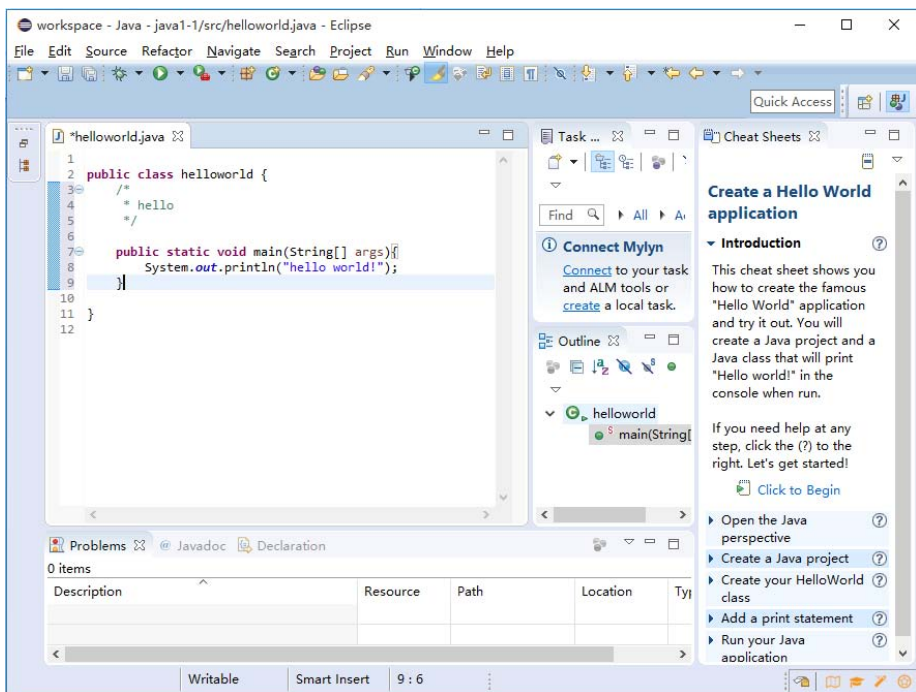


图 1-24 输入“helloworld.java”源程序

(7) 编译并运行程序。单击【Run】|【Run as】|【1 Java Application】选项，如图 1-25 所示。

(8) 在弹出的图 1-26 所示窗口中单击【OK】按钮，运行结果显示在 Console 控制台选项卡中，如图 1-27 所示。

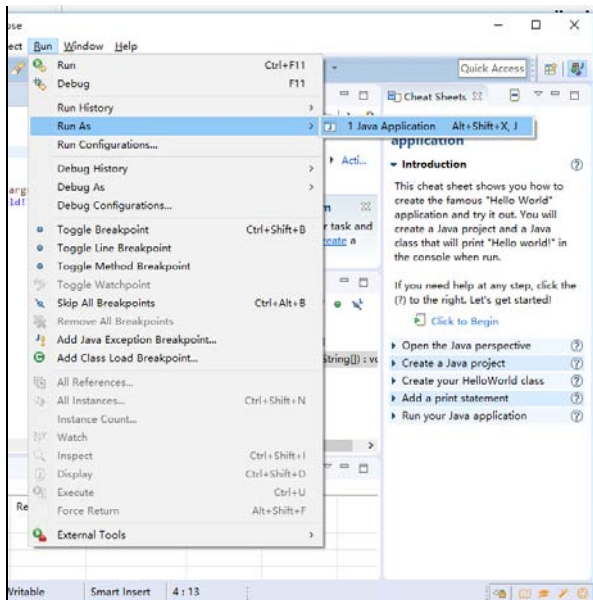


图 1-25 编译并运行程序

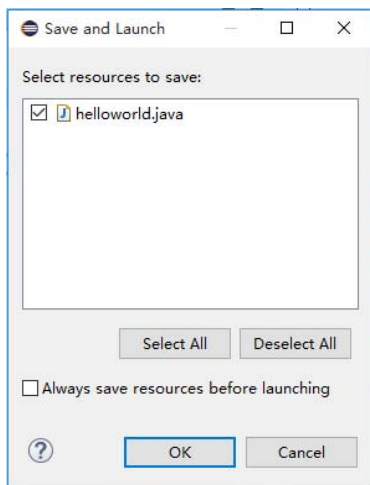


图 1-26 Save and Launch 窗口

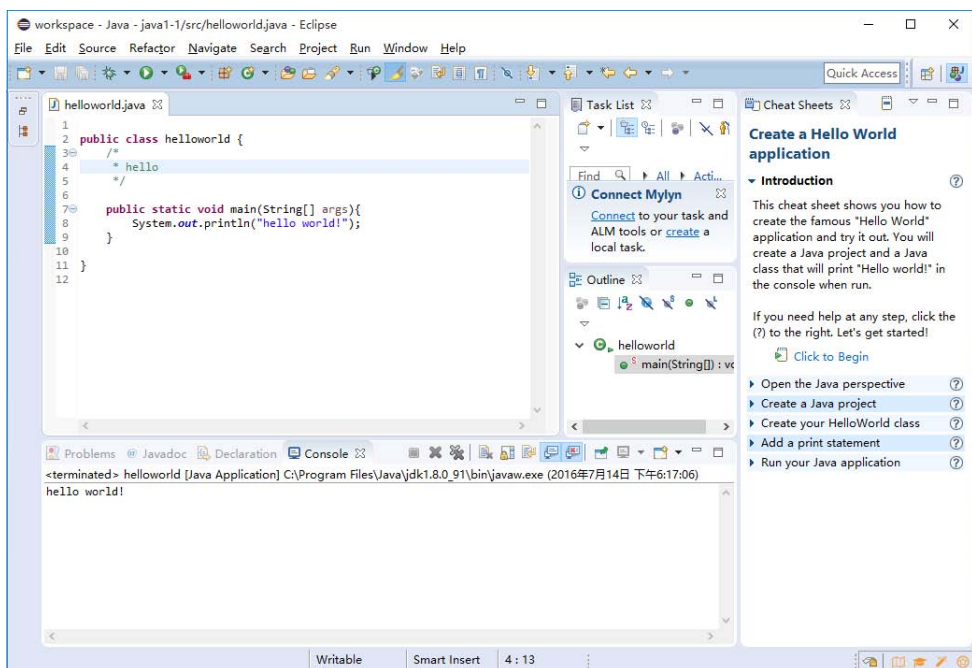


图 1-27 运行结果

1.4 本章小结

本章主要介绍了 Java 的基础知识,包括 Java 的起源和发展,Java 语言的特点,Java 软件开发包的安装方法和环境变量的配置方法等内容。通过本章内容的学习,读者应当熟练掌握 Java 开发环境的搭建以及 Java 应用程序的编写、编译和运行过程。

1.5 思考和练习

一、选择题

- 1、JDK 是()。
 - A、一种全新的程序语言
 - B、一种程序开发辅助工具
 - C、一种由 Java 写成的并支持 Java Applet 的浏览器
 - D、一种游戏软件
- 2、Java 源文件和编译后的文件扩展名分别为()。
 - A、.class 和 .java
 - B、.java 和 .class
 - C、.class 和 .class
 - D、.java 和 .java

3、Java 语言具有许多优点和特点，下列选项中，哪个反映了 Java 程序并行机制的特点()?

A、安全性 B、多线程 C、跨平台 D、可移植

4、在 Java 中，不属于整数类型变量的是()。

A、double B、long C、int D、byte

5、下面哪条语句是正确的()。

A、Object o=new Button("A"); B、Button b=new Object("B");

C、Panel p=new Frame(); D、Frame f=new Panel();

6、执行下列方法 method()，返回值为()。

```
int method() {  
    int num=10;  
    if (num>20)  
        return num;  
    num=30;  
}
```

A、10 B、20 C、30 D、编译出错

二、编程题

1、在记事本中编写一个显示“欢迎和我一起学习 Java!”的 Java 应用程序并查看运行结果。

2、在 Eclipse 开发平台中编写一个显示“欢迎使用 Eclipse!”的 Java 应用程序并查看运行结果。