

中国电子教育学会高教分会推荐
普通高等教育电子信息类“十三五”课改规划教材

Java 程序设计基础

主编 罗刚 原晋鹏



西安电子科技大学出版社
<http://www.xduph.com>



XDUP 538700

封面设计：

内容简介

Java语言具有面向对象、平台无关性、可靠稳定、分布式以及多线程等特点，是近年来最为流行的程序设计语言。

本书包括Java语言基础、Java面向过程编程、Java数组与字符串、Java类与对象、Java继承与抽象类、多态与接口、异常处理、Java输入/输出、Java常用类、图形界面设计以及事件处理等内容，语言深入浅出，通俗易懂，知识点循序渐进，重点突出，既注重理论的说明，也强调实际动手能力的培养。

ISBN 978-7-5606-5085-2



9 787560 650852 >

定价：35.00元

中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

Java 程序设计基础

主 编 罗 刚 原晋鹏

西安电子科技大学出版社

内 容 简 介

Java 语言具有面向对象、平台无关、可靠稳定、分布式以及多线程等特点,是近年来最为流行和优秀的程序设计语言。目前国内外市场对 Java 程序开发人员的需求巨大。

本书共 12 章,内容涉及 Java 概述,Java 语言基础,Java 面向过程编程,Java 数组与字符串,Java 类与对象、Java 继承与抽象类,多态与接口,异常处理,Java 输入/输出,Java 常用类介绍,图形界面设计以及事件处理等 Java 的基础知识与初步应用。

本书语言深入浅出,通俗易懂,知识点循序渐进,重点突出,既注重理论的说明,也强调实际动手能力的培养。

本书可以作为高等学校计算机等专业的 Java 语言程序设计课程教材,也可供自学者及软件开发人员参考使用。

图书在版编目(CIP)数据

Java 程序设计基础/罗刚,原晋鹏主编. —西安:西安电子科技大学出版社,2018.8

ISBN 978-7-5606-5085-2

I. ① J… II. ① 罗… ② 原… III. ① JAVA 语言—程序设计 IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2018)第 198366 号

策划编辑 毛红兵

责任编辑 买永莲

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com

电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 北京虎彩文化传播有限公司

版 次 2018 年 8 月第 1 版 2018 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 14.5

字 数 341 千字

印 数 1~1000 册

定 价 35.00 元

ISBN 978-7-5606-5085-2/TP

XDUP 5387001-1

如有印装问题可调换

前 言

本书是编者多年开发和教学经验的积累，面向 Java 初学者，没有深奥晦涩的技术原理与编程技巧，兼备易读性和易操作性，使 Java 学习者在学习过程中能够把握 Java 学习的重点，掌握 Java 的基础编程思想和方法，提高编程能力。

本书共 12 章，从初识 Java 语言、Java 环境安装到第一个程序的编译运行，在介绍 Java 基础语法过程中通过与 C 语言进行对比，逐步过渡到 Java 基础语法；从 Java 面向过程编程到 Java 面向对象编程，详细阐述了 Java 面向对象的三大特征，即从封装到继承、从继承到多态、从抽象类到接口，这些是本书的核心内容。本书还介绍了一些 Java 的后续知识和应用，如 Java 的异常处理机制、Java 的文件处理、Java 输入/输出、Java 常用类的使用、Java 图形界面开发和事件处理等内容，舍弃了一些对于初学者而言较为困难的或可在后续课程中深入学习的应用知识，如 Java 的网络编程、多线程、数据库编程、数据结构、泛型与集合框架以及绘图等，让初学者能够专注于掌握 Java 的基础知识，更好地把握 Java 的编程思想。

对编程的学习应该是循序渐进的，因此编者认为 Java 基础的学习主线是 Java 基础语法→Java 面向过程编程→Java 面向对象编程→Java 应用开发。前导课 C 语言是学习 Java 的重要基础，Java 的基础语法和面向过程来源于 C 语言，C 语言的数据类型、指针地址、函数定义与调用、结构体以及结构化编程等知识都在 Java 中得到了继承和扩展，故而本书采用对比法来介绍这部分内容，让学习者更好地从 C 语言过渡到 Java 语言的学习。如果读者对于 C 语言掌握得较好，则对 Java 面向过程编程的学习来说就事半功倍；如果 Java 面向对象编程掌握得好，后续的 Java 应用开发学习就比较容易，甚至可以自主学习。Java 面向对象的编程思想和方法是进行 Java 应用开发的重要基石，这是本书强调的重点。

本书具有以下几个特点：

(1) 根据 Java 初学者的学习情况以及编者多年教学经验进行内容组织、章节安排、程序示例选择与分析讲解，尽量消除 Java 初学者的学习障碍。

(2) 采用对比教学方式，通过与 C 语言的对比，让初学者更好地过渡到 Java 基础。

(3) 重点介绍 Java 的面向对象三大特征，让学习者从程序代码的角度理解和掌握 Java 面向对象编程。

(4) 在 Java 的应用开发上选取了输入/输出、Java 常用类和窗口编程进行介绍，让学习者应用前面学习的 Java 基础知识解决实际问题，积累和提高编程能力。

本书对于每一章节的知识点均精心设计了程序示例和课后习题，通过阅读程序、分析程序和上机练习可掌握 Java 知识点，这也是学习好 Java 编程语言的有效方法。希望本书能帮助初学者披荆斩棘，迈入 Java 开发之门。

由于编者水平有限，书中疏漏与不妥之处在所难免，恳请广大读者批评指正。

罗 刚

2018 年 3 月

目 录

第一章 Java 概述	1	第三章 Java 面向过程编程	33
1.1 Java 历史简介.....	1	3.1 Java 的顺序结构.....	33
1.2 Java 的方向划分.....	2	3.2 Java 的分支结构.....	35
1.3 Java 语言的特点.....	2	3.2.1 if 语句.....	35
1.4 Java 语言的地位.....	3	3.2.2 switch 语句.....	38
1.5 Java 开发环境.....	5	3.3 循环结构.....	40
1.5.1 安装 Java JDK.....	5	3.3.1 while 循环结构.....	41
1.5.2 环境变量设置.....	6	3.3.2 for 循环结构.....	42
1.5.3 集成开发环境.....	8	3.3.3 循环控制语句.....	43
1.6 第一个 Java 程序.....	10	3.4 结构嵌套.....	43
1.7 Java 编程规范.....	12	3.5 函数.....	46
1.7.1 初识 Java 编程规范.....	12	3.5.1 函数的定义与调用.....	46
1.7.2 Java 编程规范归纳.....	13	3.5.2 Java 函数与帮助文档.....	47
本章小结.....	15	本章小结.....	51
习题一.....	16	习题三.....	51
第二章 Java 语言基础	17	第四章 Java 数组与字符串	53
2.1 Java 标识符与关键字.....	17	4.1 数组.....	53
2.1.1 Java 标识符.....	17	4.1.1 数组的基本概念.....	53
2.1.2 Java 关键字.....	17	4.1.2 数组的定义与初始化.....	54
2.2 Java 数据类型.....	18	4.1.3 数组遍历.....	56
2.2.1 整数类型.....	18	4.1.4 二维数组.....	59
2.2.2 字符类型.....	19	4.1.5 Arrays 类.....	62
2.2.3 小数类型.....	21	4.2 字符串.....	63
2.2.4 布尔类型.....	22	4.2.1 字符串基本概念.....	63
2.2.5 引用变量.....	22	4.2.2 String 类.....	65
2.3 Java 运算符.....	23	4.2.3 StringBuffer 和 StringBuilder 类.....	71
2.4 Java 数据类型转换.....	24	本章小结.....	72
2.4.1 自动类型转换.....	24	习题四.....	73
2.4.2 强制类型转换.....	25	第五章 Java 类与对象	74
2.5 Java 的标准输入/输出语句.....	26	5.1 面向对象编程基础.....	74
2.5.1 Java 标准输出语句.....	26	5.2 类与对象.....	75
2.5.2 Java 标准输入语句.....	28	5.2.1 类的基本概念.....	75
本章小结.....	30	5.2.2 类的结构与定义.....	75
习题二.....	31		

5.2.3 对象的基本概念	77	7.5.4 面向接口编程	115
5.2.4 对象的初始化	78	本章小结	116
5.3 构造函数	83	习题七	116
5.4 成员修饰符	84	第八章 异常处理	118
5.4.1 访问控制符	85	8.1 异常处理基础	118
5.4.2 static 修饰符	86	8.2 异常处理语法	120
本章小结	90	8.2.1 try-catch-finally	120
习题五	90	8.2.2 throw/throws	125
第六章 Java 继承与抽象类	92	8.3 自定义异常类	127
6.1 继承的概念	92	本章小结	129
6.2 继承的基本语法	93	习题八	130
6.3 UML 图	95	第九章 Java 输入/输出	131
6.4 final 修饰符	95	9.1 输入/输出的基本概念	131
6.4.1 最终类	96	9.1.1 输入与输出	131
6.4.2 最终方法	96	9.1.2 流对象	132
6.4.3 最终变量	96	9.2 输入/输出类层次结构	133
6.5 Object 类	97	9.3 面向字节的输入/输出	133
6.5.1 equals()方法	97	9.3.1 面向字节的文件输入流	134
6.5.2 toString()方法	98	9.3.2 面向字节的文件输出流	136
6.5.3 getClass()方法	99	9.3.3 带缓冲的字节输入/输出流	138
6.5.4 hashCode()方法	100	9.3.4 格式化输入/输出流	142
6.6 抽象类	100	9.4 面向字符输入/输出	145
本章小结	102	9.4.1 面向字符的文件输入流	145
习题六	102	9.4.2 面向字符的文件缓冲输入流	146
第七章 多态与接口	103	9.4.3 面向字符的文件输出流	148
7.1 多态	103	9.4.4 面向字符的文件缓冲输出流	149
7.2 多态的支撑技术	103	9.5 其它输入/输出流	150
7.2.1 向上转型	103	9.5.1 对象输入/输出流	150
7.2.2 动态绑定	105	9.5.2 数组/字符串输入/输出流	152
7.3 多态实现	106	9.5.3 顺序输入流	153
7.4 多态分析	109	9.6 File 类	154
7.4.1 多态发生的地方	109	9.6.1 File 对象	154
7.4.2 多态的作用	109	9.6.2 对文件进行操作	155
7.5 接口	110	9.6.3 对文件夹进行操作	156
7.5.1 接口声明	110	本章小结	158
7.5.2 实现接口	111	习题九	158
7.5.3 接口与多态	112		

第十章 Java 常用类介绍	160	11.5.7 菜单.....	193
10.1 基本数据包装类.....	160	11.6 布局管理.....	194
10.2 System 类.....	162	11.6.1 绝对布局.....	195
10.3 Random 类.....	163	11.6.2 流式布局管理器.....	197
10.4 日期时间类.....	165	11.6.3 边界布局管理器.....	199
10.4.1 Date 类.....	166	11.6.4 网格布局管理器.....	202
10.4.2 Calendar 类.....	166	11.6.5 网格包布局管理器.....	205
10.4.3 SimpleDateFormat 类.....	168	本章小结.....	210
本章小结.....	171	习题十一.....	210
习题十.....	171		
 		第十二章 事件处理	212
第十一章 图形界面设计	173	12.1 事件基本概念.....	212
11.1 Java 图形界面设计简介.....	173	12.1.1 事件.....	212
11.2 AWT 概述.....	174	12.1.2 事件源.....	213
11.3 Swing 概述.....	175	12.1.3 监听器接口与监听器对象.....	214
11.4 JFrame 窗口.....	177	12.1.4 监听器适配器.....	215
11.5 常用窗口组件.....	180	12.2 委托事件模型.....	215
11.5.1 标签.....	180	12.3 事件处理程序.....	216
11.5.2 字体、颜色与图像.....	182	12.3.1 标准事件处理.....	216
11.5.3 面板.....	183	12.3.2 标准事件处理的另外两种形式.....	219
11.5.4 按钮.....	185	12.3.3 具体事件处理.....	221
11.5.5 文本组件.....	189	本章小结.....	223
11.5.6 下拉列表.....	191	习题十二.....	224



第一章 Java 概述

本章学习内容:

- ◇ Java 的发展历史
- ◇ Java 语言的特点
- ◇ Java 的课程地位与市场地位
- ◇ Java 开发环境的搭建
- ◇ Java 的编程规范/风格
- ◇ 第一个 Java 程序的开发和运行

1.1 Java 历史简介

Java 语言源于 Sun 公司在 1990 年 12 月开始研究的一个内部项目, Sun 公司的工程师们在项目开发中发现 C 语言和可用的 API(Application Programming Interface, 应用程序编程接口)难以完成项目开发, 1991 年 4 月, Sun 公司的 James Gosling 博士和几位工程师在改进 C 语言的基础上创造出一门新的语言——Oak(橡树)。Oak 语言有望于控制嵌入在有线电视交换盒、PDA(Personal Digital Assistant, 掌上电脑)等中的微处理器, 后来更名为“Java”(爪哇咖啡)。Java 既安全、可移植, 又可跨平台, 而且人们发现它能够解决 Internet 上的大型应用问题。Java 的发展历程如下:

1995 年 5 月 23 日, Oak 语言改名为 Java, 并且在 SunWorld 大会上正式发布 Java 1.0 版本。Java 语言第一次提出了“Write Once, Run Anywhere”的口号。

1996 年 1 月 23 日, JDK 1.0 发布, Java 语言有了第一个正式版本的运行环境。JDK 1.0 提供了一个纯解释执行的 Java 虚拟机实现(Sun Classic VM)。JDK 1.0 版本的代表技术包括 Java 虚拟机、Applet、AWT 等。

1998 年 12 月 4 日, JDK 迎来了一个里程碑式的版本 JDK 1.2, 工程代号为 Playground(竞技场), Sun 公司在这个版本中把 Java 技术体系拆分为 3 个方向, 分别是面向桌面应用开发的 J2SE(Java 2 Platform, Standard Edition)、面向企业级开发的 J2EE(Java 2 Platform, Enterprise Edition)和面向手机等移动终端开发的 J2ME(Java 2 Platform, Micro Edition)。

2000 年 5 月 8 日, 工程代号为 Kestrel(美洲红隼)的 JDK 1.3 发布。

2002 年 2 月 13 日, JDK 1.4 发布, 工程代号为 Merlin(灰背隼)。JDK 1.4 是 Java 真正走向成熟的一个版本, Compaq、Fujitsu、SAS、Symbian、IBM 等著名公司都有参与甚至实现自己独立的 JDK 1.4。

2004 年 9 月 30 日, JDK 1.5 发布, 工程代号为 Tiger(老虎)。自 JDK 1.2 以来, Java 在



语法层面的变化一直很小，而 JDK 1.5 在 Java 语法易用性上做出了非常大的改进。

2006 年 12 月 11 日，JDK 1.6 发布，工程代号为 Mustang(野马)。在这个版本中，Sun 公司终结了从 JDK 1.2 开始的已经有 8 年历史的 J2EE、J2SE、J2ME 的命名方式，启用 Java SE 6、Java EE 6、Java ME 6 的命名方式。

2009 年 2 月 19 日，工程代号为 Dolphin(海豚)的 JDK 1.7 完成了其第一个里程碑式的版本。之后，JDK 1.7 发布了 9 个 Update 版本，最新的 Java SE 7 Update 9 于 2012 年 10 月 16 日发布。

2009 年 4 月 20 日，Oracle 公司以 74 亿美元的价格收购 Sun 公司，Java 商标从此正式归 Oracle 所有(Java 语言本身并不属于某个公司所有，它由 JCP(Java Community Process)进行管理，JCP 主要是由 Sun 公司或者说 Oracle 公司所领导的)。

2011 年 7 月 28 日，Oracle 公司发布 Java SE 1.7。

2014 年 3 月 18 日，Oracle 公司发布 Java SE 1.8。

1.2 Java 的方向划分

Java 的版本演进到了 Java 2，根据其应用的不同层面进行划分，Java 2 平台被分为三个版本，各版本及其说明如表 1-1 所示。

表 1-1 Java 2 版本划分

版本	描述	英文
J2EE	Java 平台企业版，适用于服务器，目前已成为企业运算、电子商务等领域的热门技术	Enterprise Edition
J2SE	Java 平台标准版，适用于一般的计算机，开发 PC 上的应用软件，是 Java 平台的基础	Standard Edition
J2ME	Java 平台微型版，适用于手持设备、消费产品、嵌入式设备等的应用开发，如手机移动商务应用开发等	Micro Edition

1.3 Java 语言的特点

Java 语言具有以下几个特点：

1. 简单

Java 由 C、C++ 发展而来，其语言风格与 C++ 十分相似；另一方面，Java 比 C++ 简单，它删除了 C++ 中难理解、易引起安全隐患的内容，如最典型的指针操作、多继承等，降低了学习的难度；同时，Java 还有一个特点，即它的基本语法部分与 C 语言的几乎一模一样，容易为人们接受。

2. 面向对象

Java 是一种面向对象的程序设计语言，在面向对象上相较于 C++ 更加合理和易于理解，更接近于现实世界的概念。同时，Java 语言支持静态和动态风格代码的继承和重用，所采



用的自动装箱和拆箱技术让 Java 的各个基本类型也可以作为对象进行处理。

3. 分布式

Java 具有支持分布式计算的特征。分布式计算中的“分布”指的是数据分布和操作分布。数据分布即应用系统所操作的数据可以分散存储在不同的网络节点上；操作分布即应用系统的计算可由不同的网络节点完成。Java 能实现这两种分布。

4. 安全

由于 Java 是应用于网络的开发语言，因而其安全至关重要。Java 在语言机制和运行环境中都引入了多级安全措施，主要如下：

(1) 内存分配及布局由 Java 运行系统规定，取消了指针的操作，不允许直接对内存进行操作，实现了内存管理自动化，内存布局由 Java 虚拟机(Java Virtual Machine, JVM)决定。

(2) 运行系统提供字节码验证、代码认证与代码访问权限控制的安全控制模型。

① 提供字节码检验器，以保证程序代码在编译和运行过程中接受一层层的安全检查，这样可以防止非法程序或病毒的入侵。

② 提供了文件访问控制机制，严格控制程序代码的访问权限。

③ 提供了多种网络软件协议的用户接口，用户可以在网络传输中使用多种加密技术来保证网络传输的安全性和完整性。

5. 编译和解释的结合

Java 应用程序由编译器编译成字节码，这个字节码不是最终可执行的程序，不能在具体的平台上运行，还需要由运行系统上的字节码解释器将其解释成机器语言，从而达到边解释边执行的目的。

6. 可移植

Java 的最大特点是“一次编程，多次使用”。任何机器只要配备 Java 虚拟机，便可以运行 Java 程序，因为 Java 语言不是针对某个具体平台结构设计的。Java 源程序经编译后产生的字节码是一种与具体指令无关的指令集合，通过 Java 虚拟机就可以在任何平台上运行，即 Java 通过虚拟机屏蔽了平台的差异性，使开发人员大大降低了开发、维护和管理开销。

7. 多线程

多线程技术允许应用程序并行执行，即同时做多件事，满足了一些复杂软件的要求。Java 不但内置多线程功能，而且提供语言级的多线程支持，即定义了一些用于建立、管理多线程的类和方法，使得开发具有多线程功能的程序变得简单和有效。

1.4 Java 语言的地位

下面从课程地位和市场地位两个方面来说明 Java 语言的地位。

1. 课程地位

Java 语言在计算机相关专业中的课程地位如图 1-1 所示，可以看出，Java 最主要的前



导课程是 C 语言，Java 的面向过程以及结构化编程主要来自于 C 语言；Java 的后续课程很多，主要是因为 Java 技术的应用范围非常广泛，而作为这些课程的基础，如果不能很好地掌握 Java 语言，对于后续课程的学习将有很大影响，所以有关 Java 语言的课程在软件开发中占有非常重要的地位。

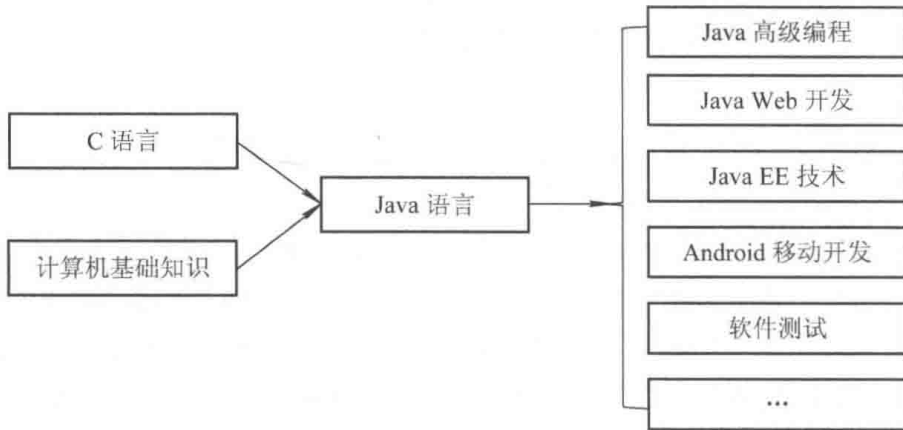


图 1-1 Java 课程地位

2. 市场地位

TIOBE 编程语言社区排行榜是编程语言流行趋势的一个指标，每月更新。该排行榜的排名基于互联网上有经验的程序员、课程和第三方厂商的数量，排名使用著名的搜索引擎(如 Google、MSN、Yahoo!、Wikipedia、YouTube 以及 Baidu 等)进行计算。

自 2000 年初以来，Java 在 TIOBE 上不管是历史排名，还是未来走势，一直都在前三。Java 从出道至今，经过多年的积累和沉淀，出现了很多优秀的开源社区，如 Apache 和 Spring。这些优秀的社区提供了很多非常好的框架，借助于这些框架，我们不用关注 Java 底层的开发，而只需关注业务的实现。

Java 程序员职业发展路线所表现出的 Java 的市场地位如图 1-2 所示。

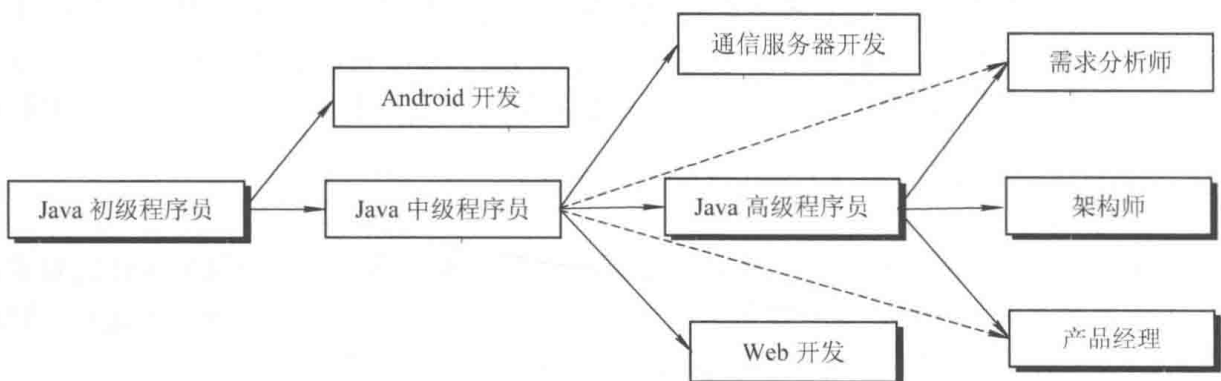


图 1-2 Java 市场地位

Java 程序员主要有三个开发方向：

(1) Android 移动端开发：主要是移动端应用的开发。移动端包括搭载 Android 系统的手机、平板、电视盒子等设备。

(2) Web 开发：主要有 HTML5 开发、B/S 应用开发、微信开发等，包括企业的信息系统开发，是 Java 应用的一个非常大的市场。



(3) 通信服务器开发: 主要是通信服务器的开发, 如游戏服务器、腾讯的 QQ 服务器等。

最新的一些报告证明, Java 程序员是业内薪资最高的程序员之一。根据全球数字化业务媒体机构 Quartz 的分析, 掌握 Java 语言有利于提高薪资。职业规划公司 Gooroo 在 2016 年薪资和需求报告中指出, Java 仍然是中国、美国、英国和澳大利亚等国最受欢迎和薪资最高的编程语言之一, 且雇主对 Java 编程技能有着很高的需求。

Java 是广泛使用的编程语言, 拥有庞大的客户群。据估计, 全球范围内有超过 30 亿台设备在运行 Java, 超过其他任何一种语言。使用 Java 编写的程序几乎可用于任何设备, 包括智能手机、服务器、自动取款机、销售点终端机(POS)、蓝光播放器、电视、机顶盒、物联网网关、医疗设备、Kindle 电子阅读器、汽车等。

可以看出, Java 语言的市场地位非常高, 是具有很高学习价值的编程语言和技术。

1.5 Java 开发环境

Java 开发环境的安装主要有三个步骤, 如图 1-3 所示。

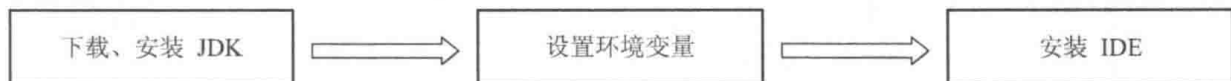


图 1-3 Java 开发环境安装

1.5.1 安装 Java JDK

JDK(Java Development Kit)是 Java 语言的软件开发工具包, 是整个 Java 的核心, 包括 Java 运行环境、Java 工具和 Java 基础类库。要进行 Java 程序的开发, 首先要在电脑上安装 JDK, 其下载地址为 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>。其下载页面如图 1-4 所示。

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	jdk-8u161-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.88 MB	jdk-8u161-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.96 MB	jdk-8u161-linux-i586.rpm
Linux x86	183.76 MB	jdk-8u161-linux-i586.tar.gz
Linux x64	166.09 MB	jdk-8u161-linux-x64.rpm
Linux x64	180.97 MB	jdk-8u161-linux-x64.tar.gz
macOS	247.12 MB	jdk-8u161-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.99 MB	jdk-8u161-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.29 MB	jdk-8u161-solaris-sparcv9.tar.gz
Solaris x64	140.57 MB	jdk-8u161-solaris-x64.tar.Z
Solaris x64	97.02 MB	jdk-8u161-solaris-x64.tar.gz
Windows x86	198.54 MB	jdk-8u161-windows-i586.exe
Windows x64	206.51 MB	jdk-8u161-windows-x64.exe

图 1-4 JDK 下载示意图



JDK 1.8 是目前的主要版本, 对于初学者来说, 使用 1.6 或者 1.7 版本并没有太多影响, Java 主要的语法是一样的, 但在后期进行开发的时候要注意版本之间的变化。下载时要注意 JDK 的使用平台, 如 Linux32 位/64 位、macOs、Solaris、Windows x86 以及 Windows x64 等, 如图 1-4 所示, 可以根据自己的电脑选择 Windows x86(32bit)或者 Windows x64(64bit) 进行安装。

JDK 的安装方法和一般的软件安装一样, 跟随软件安装的向导界面, 按照默认的安装设置进行即可。安装完后可以在开始菜单中查看, 也可以在控制面板中查看是否安装成功, 如图 1-5 所示。



图 1-5 检查 JDK 是否安装成功

双击 Java 图标, 然后在弹出的界面中点击“关于”, 可以查看 Java 的版本号, 如图 1-6 所示。

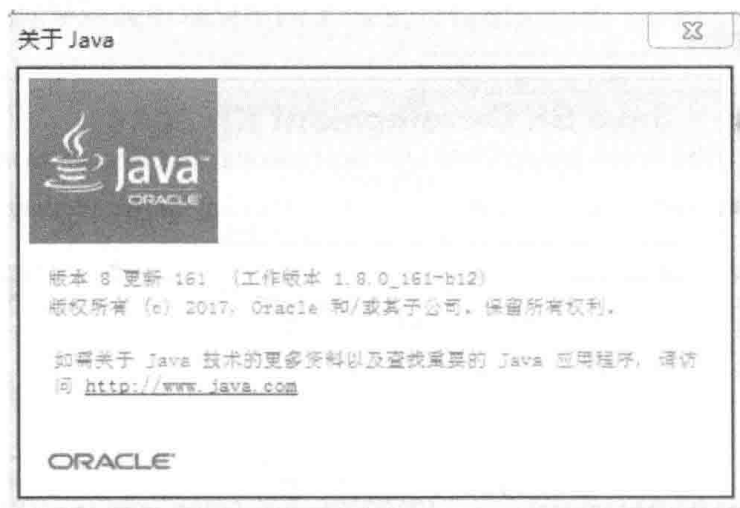


图 1-6 查看 JDK 版本

1.5.2 环境变量设置

安装好 JDK 后需要进行环境变量的设置, 具体流程如图 1-7 所示。



(1) 右键点击“计算机”→“属性”，在弹出的系统属性对话框中点击“高级系统设置”→“高级”→“环境变量(N)...”。

(2) 在“环境变量”窗口中的“系统变量(S)”下点击“新建(W)...”，新建“JAVA_HOME”变量，变量值为 JDK 的安装目录名，默认安装目录为 C:\Program Files\Java\jdk1.8.0_161。

(3) 在“系统变量(S)”栏找到 Path 变量，双击进行编辑，在变量值后输入“%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;”，如图 1-8 所示。

注：如果原来的 Path 变量值的最末尾没有“;”号，则需要先输入“;”号后再输入上面的代码。

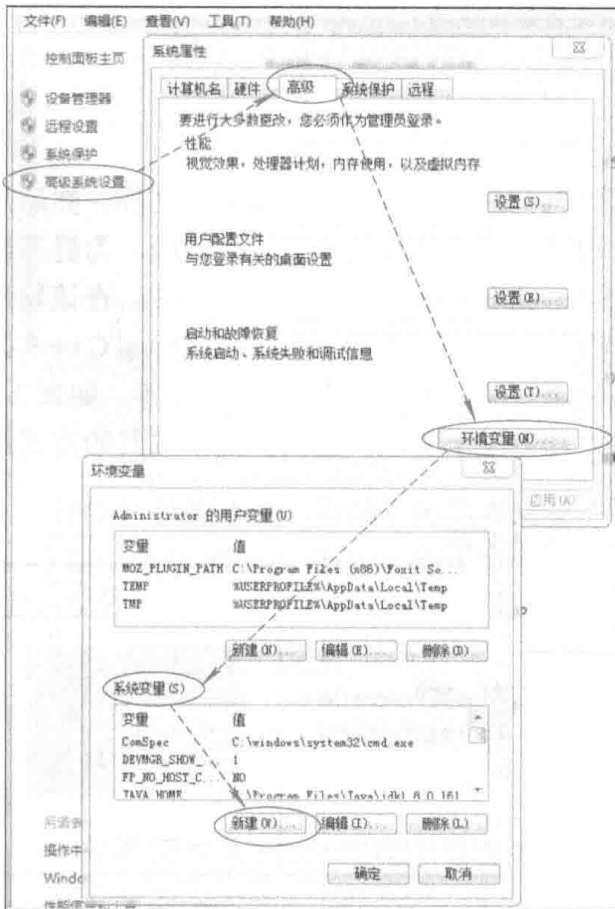


图 1-7 Java 环境变量设置流程

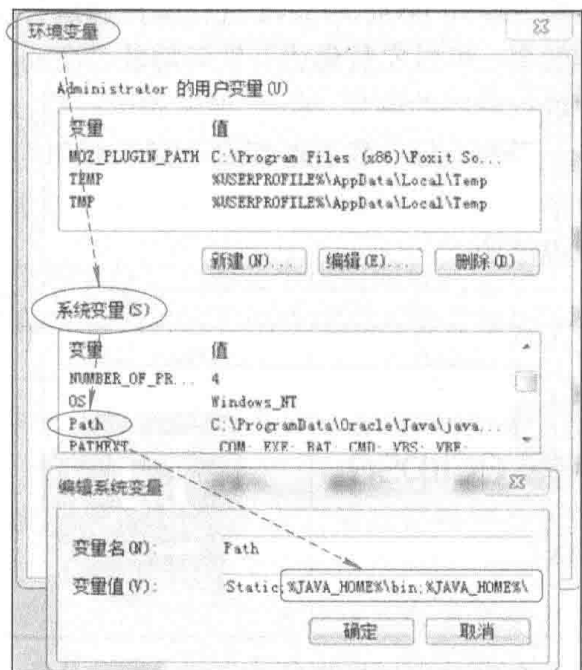


图 1-8 编辑环境变量

(4) 再次在“系统变量(s)”栏点击“新建(W)...”，输入 CLASSPATH，变量值后输入“;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar”(注意最前面有一个点)，如图 1-9 所示。

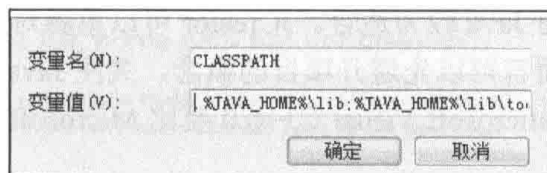


图 1-9 输入环境变量

(5) 检验是否配置成功，运行 cmd，在黑色窗口中输入“java -version”(java 和 -version 之间有空格)。若如图 1-10 所示，显示 Java 的版本信息，则说明安装配置成功。



```

管理员: C:\windows\system32\cmd.exe
C:\Users\Administrator>java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)

C:\Users\Administrator>
  
```


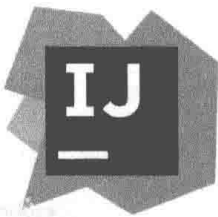


图 1-10 环境变量设置是否成功

1.5.3 集成开发环境

安装好 JDK 后就可以进行 Java 程序的开发了, 使用记事本编辑 Java 源文件, 然后采用命令语句 `javac` 和 `java` 进行编译和运行 Java 程序, 但是这样的开发效率很低。为提高编程效率, 可以安装集成开发环境软件(Integrated Development Environment, IDE), 在该软件中进行程序的编辑、编译、调试和运行等, 这类似于 C 语言的 IDE: Microsoft Visual C++ 6.0。

目前, Java 的 IDE 很多, 较为常用的有 Eclipse、NetBeans、IntelliJ IDEA 等, 如表 1-2 所示, 但是这些 IDE 对于初学者来说过于庞大, 功能过于复杂, 且大多是以项目的方式组织程序的。

表 1-2 常用的 Java IDE

Eclipse	IntelliJ IDEA	NetBeans	JCreator
			

我们推荐初学者使用 JCreator, 它是 Xinox Software 公司开发的一个用于 Java 程序设计的集成开发环境软件, 具有编辑、调试、运行 Java 程序的功能。JCreator 分为 Pro 版和 Le 版, Pro 版是专业版, 需要付费, 功能较为强大; Le 版是免费版本, 功能有一定限制。JCreator 具有小巧、简洁、美观、速度快、效率高、语法着色、代码自动完成、代码参数提示等特点, 对于初学者学习 Java 较为适合。JCreator 可以单独对 Java 的源文件进行编译、调试和执行, 对于初学者而言可以先避开项目的概念, 关注 Java 的基本语法, 就像对于 C 语言的学习, 初学者使用 Microsoft Visual C++6.0 要比 Microsoft Visual C++2010 容易掌握和使用。

JCreator Le 5.0 软件可以在 JCreator 官网(<http://www.jcreator.com/>)下载, 也可以在很多常用的软件下载网站上搜索下载, 下载完按照安装向导提示进行安装即可。安装完毕后打开软件, JCreator 的主界面如图 1-11 所示。

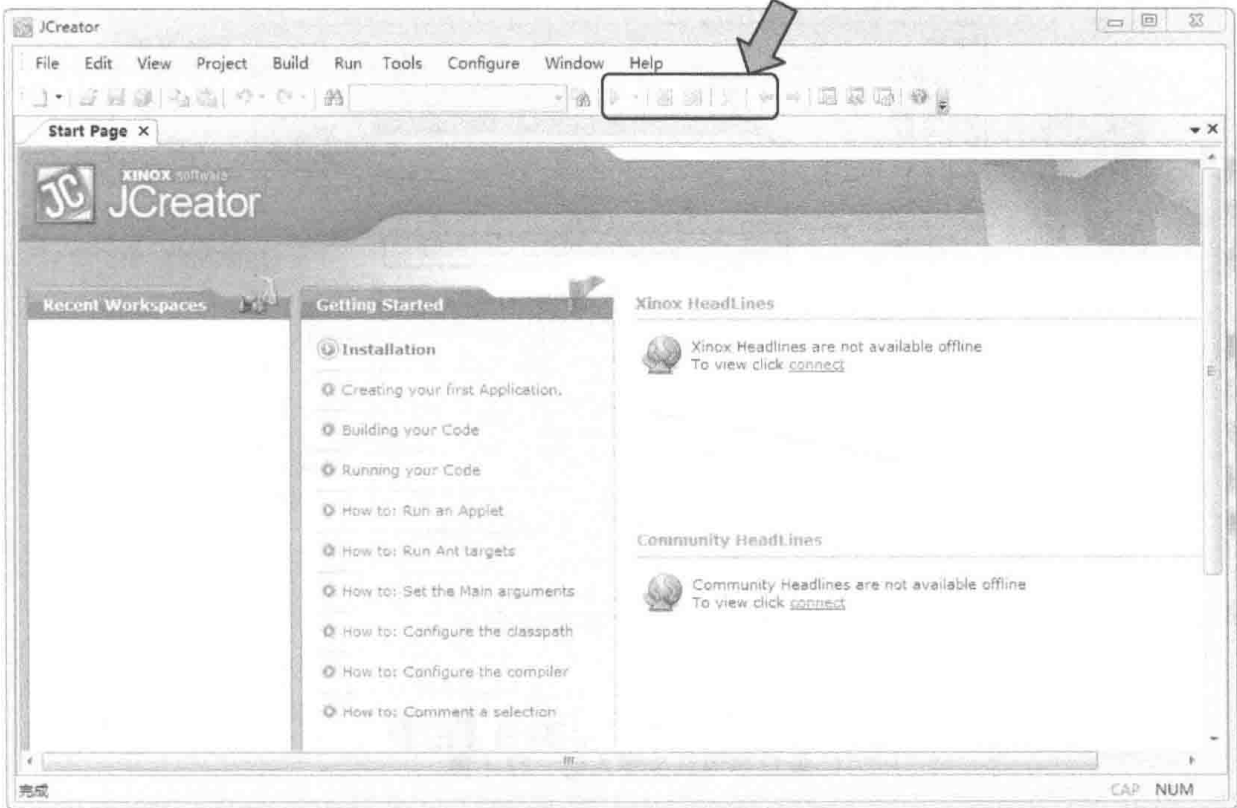




图 1-11 JCreator 主界面

该软件是英文版，但对于初学者来说，主要使用工具栏上的几个按钮即可：

: **Build** 按钮，将 Java 源程序编译为能被 Java 虚拟机运行的字节码文件，类似于在 VC++6.0 里面将 .c 文件进行编译的操作。

: **Run** 按钮，运行 Java 程序，经过上面的 Build 操作，如果没有语法错误，就可以在 IDE 中运行程序，观察结果。

: **Stop** 按钮，如果程序在运行中，则该按钮变成黄色，点击该按钮能终止程序运行。

如果在使用 JCreator 过程中出现图 1-12 所示的提示，表示后台的 Java JDK 还没有和我们之前安装的 JCreator 关联起来，解决方法如下：在 JCreator 菜单 Configure 中选择 Options → JDK Profiles，检查是否有 JDK；如果为空，就点击 New 按钮，然后找到 JDK 安装的目录，如图 1-13 所示。



图 1-12 JDK 设置问题