

Java面向对象 程序设计

```
modifiers.new("
mirror_mod.use_y = False
mirror_mod.use_z = True

selection at the end -add back the de
mirror_ob.select= 1
mirror_ob.select=1
context.scene.objects.active = modifier
selected" + str(modifier_ob)) # mod
mirror_ob.select = 0
key.context.selected_objects[0]
objects[one.name].select = 1

print("please select exactly two objects,

OPERATOR CLASSES

Operator):
error to the selected object""
mirror_mirror_x"
```

主 编 陆建波

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内容简介

本书从原理入手，结合实例讲解 Java 语言的基本语法和面向对象编程的思想，重点介绍 Java 语言中的类、接口、多线程、集合、网络编程、数据库编程、XML 编程、JSP 编程、Struts 框架、Spring 框架、Hibernate 框架、EJB 框架、JMS 消息队列、WebService 服务等。本书可作为高等院校计算机专业及相关专业的教材，也可供从事 Java 开发的工程技术人员参考。

Java 面向对象程序设计

主 编 陆建波

副主编 蒋雪玲 苏杨茜

参 编 李松钊 覃正优

刘春霞 戴智鹏



北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书通过丰富、实用的精选实例将面向对象的程序设计方法与Java语言相结合,注重培养读者使用面向对象的思维方法分析问题和解决问题的能力。全书共8章。第1章介绍了编程语言的发展、Java语言的特点、运行机制和Eclipse集成开发环境;第2章介绍了Java语言基础;第3章介绍了面向对象的基本理论、原理、技术方法、设计原则以及相关的哲学思考;第4章介绍了Java异常类、数据库连接类等高级特性;第5章介绍了网络编程的常用类;第6章介绍了Java常用图像处理类、图像增强技术、图像分割算法及图像处理在人脸识别、二维码生成中的应用;第7章介绍了Java数据处理的流程;第8章介绍了Android开发流程。

本书通俗易懂、实例丰富,可作为高等院校计算机类、人工智能类、数据科学与大数据技术类、信息管理与信息系统类专业相关课程的教学教材,也可作为自学、函授或培训的教材或参考书。

版权专有 侵权必究

图书在版编目(CIP)数据

Java面向对象程序设计/陆建波主编.--北京:
北京理工大学出版社,2022.1

ISBN 978-7-5763-0872-3

I. ①J… II. ①陆… 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(2022)第013224号

出版发行/北京理工大学出版社有限责任公司

社 址/北京市海淀区中关村南大街5号

邮 编/100081

电 话/(010) 68914775(总编室)

(010) 82562903(教材售后服务热线)

(010) 68944723(其他图书服务热线)

网 址/http://www.bitpress.com.cn

经 销/全国各地新华书店

印 刷/涿州市新华印刷有限公司

开 本/787毫米×1092毫米 1/16

印 张/15.25

字 数/355字

版 次/2022年1月第1版 2022年1月第1次印刷

定 价/78.00元

责任编辑/江立

文案编辑/李硕

责任校对/刘亚男

责任印制/李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前 言

近年来人工智能、大数据、互联网等科学技术迅猛发展，社会的信息化对大学生的信息处理与应用能力提出了更高的要求。面向对象思维是信息学科专业的学生必备的核心专业素养。面向对象中的最典型的语言是 Java。Java 应用范围极为广泛，在网络、图像、人工智能、数据科学、移动开发、互联网应用方面极具优势。

本书的编写顺应了当今计算机新技术的发展趋势，注重培养学生应用面向对象的思维方法分析问题和解决问题的思维方式与意识。

本书主要特点如下。

(1) 在内容组织上，力求突出知识的基础性、应用性。选择基础的面向对象的理论知识、基础的 Java 特性，结合具体应用实例的面向对象设计原则、数据库、网络编程、图像增强技术与分割算法、数据处理技术、大数据分析可视化、Android 开发为本书的主要内容。

(2) 在表达形式上，侧重以实例图形、图像以及过程步骤的截图对知识点及操作进行展示，利于读者直观、具体地理解。

(3) 在教学方法上，主要采用了实例教学法。围绕各知识点，我们精选了具有代表性、实用价值的学习、生活、工作中的实例，如人脸识别、网络聊天、聚类分析、Android 应用 App 等，力求通过对实例的分析和处理，将理论与应用结合，加深理解，从而达到举一反三、学以致用目的。

(4) 在写作方法上，本书语言精练、通俗易懂、结构清晰，各章节内容连接自然流畅，知识点分布合理。

(5) 本书有丰富的配套教学资源。为了让学生巩固所学知识，提高动手实践能力，我们编写了丰富的线上、线下练习题库，制作了丰富的微课资源、技术资料，如需获取本书的相关资源可与作者联系，联系信息为 55082388@qq.com。

本书的参编人员均为从事本课程教学的一线教师，具有多年教学经验。本书由南宁师范大学陆建波担任主编，南宁师范大学蒋雪玲、苏杨茜担任副主编。各章节的编写分工为，陆建波、刘春霞负责编写第 1 章，蒋雪玲负责编写第 2 章和第 4 章，陆建波负责编写第 3 章，李松钊负责编写第 5 章，覃正优负责编写第 6 章，苏杨茜负责编写第 7 章，李松钊、戴智鹏负责编写第 8 章。

在编写过程中，南宁师范大学计算机与信息工程学院研究生孟一帆、刘晓彬、宋庆兰、丘馥祯，本科生刘龙锦等参与了本书的部分代码编写、插图绘制工作。在此，向这些同学表示感谢。

计算机技术发展日新月异，由于编者能力有限，书中难免存在不足之处，恳请各位读者和专家给予批评指正。

编者

2021 年 10 月

目 录

第 1 章 Java 概述及开发环境搭建	001
1.1 Java 概述	001
1.1.1 Java 的诞生	001
1.1.2 Java 的发展历史	002
1.1.3 Java 的特点	003
1.1.4 Java 的运行机制	004
1.1.5 Java 的现状和应用前景	005
1.2 Java 开发环境	006
1.2.1 JDK 的安装与配置	007
1.2.2 Eclipse 的使用	011
1.2.3 Eclipse 的常用功能	016
1.3 课程学习建议	018
1.4 本章小结	019
习题	019
第 2 章 Java 基础程序设计	020
2.1 标识符与关键字	022
2.1.1 标识符	022
2.1.2 关键字	022
2.2 基本数据类型	022
2.2.1 变量与常量	023
2.2.2 布尔型	024
2.2.3 字符型	024
2.2.4 整型	024
2.2.5 浮点型	025
2.3 类型转换	025
2.4 输入、输出数据	026
2.4.1 输入基本型数据	026
2.4.2 输出基本型数据	027
2.5 数组	027
2.5.1 声明数组	028

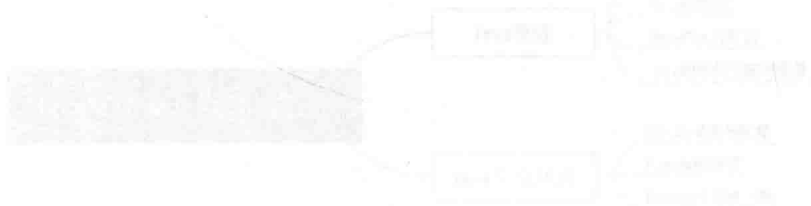
2.5.2	为数组分配元素	028
2.5.3	数组元素的使用	029
2.5.4	length 的使用	029
2.5.5	数组的初始化	030
2.5.6	数组的引用	030
2.6	运算符与表达式	031
2.6.1	算术运算符与算术表达式	031
2.6.2	关系运算符与关系表达式	032
2.6.3	逻辑运算符与逻辑表达式	033
2.6.4	赋值运算符与赋值表达式	033
2.6.5	运算符综述	034
2.7	语句概述	034
2.7.1	条件语句	035
2.7.2	switch 开关语句	037
2.7.3	循环语句	039
2.8	本章小结	043
	习题	044
第3章	面向对象	045
3.1	面向过程与面向对象	046
3.2	类与对象	048
3.2.1	类与对象的定义	048
3.2.2	访问权限	050
3.2.3	Java 类的命名规范	051
3.3	面向对象的主要特征	051
3.3.1	封装	051
3.3.2	继承	053
3.3.3	多态	055
3.4	类的使用	057
3.4.1	类的创建与引用	057
3.4.2	匿名对象	062
3.4.3	内部类	062
3.4.4	final 关键字	065
3.4.5	instanceof 关键字	066
3.4.6	this 关键字	066
3.5	抽象类	067
3.6	接口	068
3.7	统一建模语言	070
3.8	面向对象设计原则	074
3.8.1	开闭原则	075
3.8.2	依赖倒转原则	076

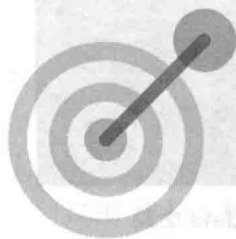
3.8.3	单一职责原则	077
3.8.4	合成复用原则	078
3.8.5	里氏替换原则	078
3.8.6	接口隔离原则	081
3.8.7	迪米特法则	081
3.9	面向对象的哲学思考	082
3.10	本章小结	084
	习题	084
第4章	包与常用类	085
4.1	包	086
4.1.1	包的作用	087
4.1.2	创建包	087
4.1.3	类的存放路径	087
4.1.4	Java 中的常用包	088
4.1.5	import 语句	088
4.2	异常类	090
4.2.1	try-catch 语句	092
4.2.2	自定义异常类	093
4.3	常用系统类	095
4.3.1	String 类	095
4.3.2	StringBuffer 类、StringBuilder 类	098
4.3.3	Date 类	099
4.3.4	Calendar 类	100
4.3.5	随机类	101
4.4	数据库类	102
4.4.1	JDBC	102
4.4.2	连接数据库	103
4.4.3	数据库操作	105
4.4.4	预处理语句	112
4.4.5	事务	114
4.5	本章小结	117
	习题	117
第5章	Java 网络编程	118
5.1	概述	119
5.2	网络编程常用类	119
5.2.1	InetAddress 类	119
5.2.2	URL 类	120
5.2.3	Socket 类与 ServerSocket 类	123
5.2.4	DatagramSocket 类与 DatagramPacket 类	126

5.3	Java 网络编程应用实例	128
5.3.1	使用 UDP 实现聊天功能	128
5.3.2	使用 TCP 实现上传文件功能	132
5.4	本章小结	136
	习题	136
第 6 章	Java 图像处理	138
6.1	图像处理基础	139
6.1.1	基础知识	139
6.1.2	基于 Java 的图像基本操作	140
6.2	图像增强	152
6.2.1	概述	152
6.2.2	图像灰度变换	153
6.2.3	直方图增强	155
6.2.4	图像平滑	158
6.2.5	图像锐化	162
6.3	图像分割	164
6.3.1	图像分割原理	164
6.3.2	图像的边缘检测	164
6.3.3	基于 K-Means 的图像分割算法	166
6.4	图像处理应用	172
6.4.1	生成二维码	172
6.4.2	人脸识别	174
6.5	本章小结	176
	习题	176
第 7 章	Java 与数据科学	178
7.1	数据获取及清洗	179
7.1.1	使用 Apache Commons IO 进行文件操作的常用方法	179
7.1.2	使用 Apache Tika 提取 PDF 文本	180
7.1.3	使用 Jsoup 从网站获取数据	183
7.1.4	使用正则表达式清洗数据	185
7.2	数据统计	186
7.2.1	使用 Apache Commons Math 计算描述性统计指标	187
7.2.2	使用 Apache Commons Math 进行频率分布统计	188
7.3	聚类分析	189
7.3.1	使用 Weka 的图形界面完成聚类	189
7.3.2	使用 Weka 的 Java K-Means 实现聚类	190
7.3.3	使用 Apache Commons Math 进行聚类分析	193
7.4	数据可视化	195
7.4.1	使用 JFreeChart 绘制图形	196

7.4.2 使用 GRAL 绘制图形	202
7.5 本章小结	203
习题	204
第 8 章 Android 与 Java	205
8.1 初识 Android	206
8.1.1 Android 的由来	206
8.1.2 Android 简介	206
8.2 Java 与 Android 的关系	207
8.3 Android 开发主要流程	207
8.4 搭建 Android 应用开发环境	211
8.4.1 安装 Android Studio	211
8.4.2 配置安装 Android SDK	215
8.4.3 配置 Android 运行与调试环境	216
8.5 开发一个简单的 Android 应用	219
8.5.1 使用 Android Studio 创建 Android 项目	219
8.5.2 一个简单的 Android 应用——Weather	220
8.6 本章小结	230
习题	230
参考文献	231

本章思维导图





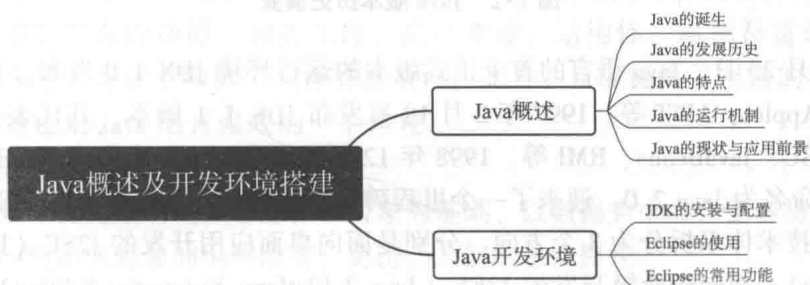
第 1 章

Java 概述及开发环境搭建

本章目标

- 了解 Java 的发展历史、特点。
- 理解 Java 的运行机制。
- 了解 Java 的现状和应用前景。
- 掌握 JDK 的安装与配置。
- 掌握 Eclipse 的安装、使用、常用功能。

本章思维导图



1.1 Java 概述

1.1.1 Java 的诞生

Java 是斯坦福大学网络 (Stanford University Network, SUN) 公司于 1995 年推出的一种可以开发跨平台应用软件的高级程序设计语言, 主要创作者为詹姆斯·高斯林 (James Gosling)。James Gosling 出生于加拿大, 是一位计算机编程天才, 于 1984 年加入 SUN 公司, 被

称为“Java之父”，如图1-1所示。

1991年4月，James Gosling为发明一种能够在各类消费性电子产品（如机顶盒、冰箱、收音机等）上运行的通用程序架构启动了“Green”计划，这个计划的产品即Java的前身是Oak（橡树）语言。由于Oak的名字已被一家显卡制造商注册，因此1995年5月23日，其正式更名为Java。随后Java很快被工业界认可，许多大公司如IBM、Microsoft、DEC等均购买了Java的使用权。此外，Java语言被美国杂志评为1995年十大优秀科技产品之一。值得注意的是，Java在2009年之前由SUN公司负责维护，2009年4月，Oracle公司花费74亿美元成功收购SUN公司，因此目前Java官网网址为Oracle官网。

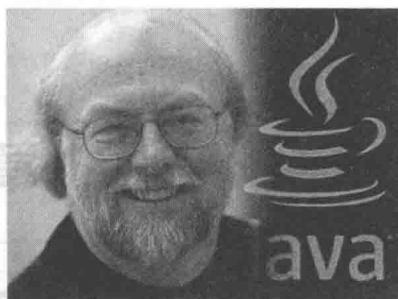


图1-1 Java之父

1.1.2 Java的发展历史

Java自1995年起至今已经发展了20多年，这20多年间，Java经历了多个阶段，版本的不断更新使Java的功能不断完善。Java版本历史演变如图1-2所示。

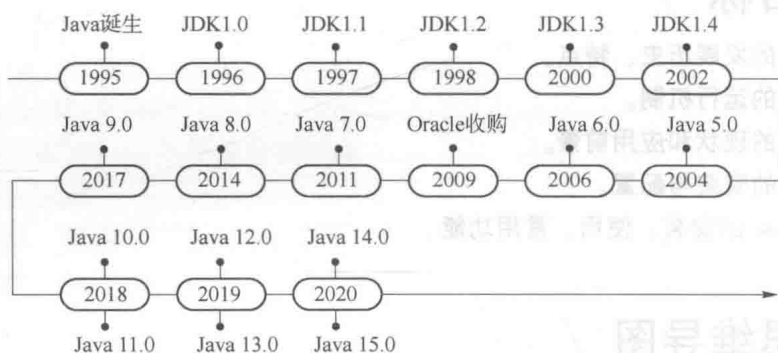


图1-2 Java版本历史演变

1996年1月23日，Java语言的首个正式版本的运行环境JDK 1.0发布，其代表技术有Java虚拟机、Applet、AWT等。1997年2月18日发布JDK 1.1版本，其代表技术涵盖JAR文件格式、JDBC、JavaBeans、RMI等。1998年12月4日，Java彻底脱离Windows图形界面的控制，重新命名为Java 2.0，迎来了一个里程碑式的运行环境JDK 1.2。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）（在下文中将详细介绍这3种技术方向）。2000年5月8日JDK 1.3发布，JDK 1.3相对于JDK 1.2改进了一些类库。从JDK 1.3开始，SUN公司维持了每隔两年发布一个JDK的主版本的习惯。在2002年2月13日发布了JDK 1.4版本，其涵盖了很多新的技术特性，如NIO、XML解析器和XSLT转换器等，自此Java的计算能力有了大幅提升。2004年9月30日J2SE 1.5版本发布，成为Java发展史上的又一个里程碑，随后J2SE 1.5更名为Java 5.0。2006年12月11日Java 6.0版本发布。2009年4月20日，Oracle公司宣布正式以74亿美元的价格收购SUN公司，Java商标从此正式归Oracle公司所有。随后2011年7月28日Java 7.0版权发布，2014年Oracle公司推出了Java 8.0版本，增加了函数式编程（Lambda表达式）和数据流（MapReduce）的处理。2017年3

月18日, Java 9.0版本发布。自Java 9.0之后, Java采用了基于时间发布的策略,每6个月发布一个版本。2018年3月20日Java 10.0版本发布,同年9月26日,Java 11.0版本正式发布;2019年3月20日Java 12.0版本发布。

目前,Java最新版本为Java 16.0版本。考虑了现阶段企业的需求及程序开发中的稳定性,本书选择学习使用的版本仍为Java 8.0版本。

在上述描述中曾提到SUN公司按照Java技术关注的重点业务领域,将Java技术分成3大版本,即J2SE、J2EE、J2ME。

J2SE是Java标准版本,它是整个Java技术的基础与核心,包含数据库连接、接口、网络编程等内容。学习J2SE之后可以开发基于C/S架构的桌面应用,如QQ(QQ有桌面应用版与网页版,网页版QQ属于J2EE)、微信等。

J2EE是Java的企业级版本,包含J2SE中的所有类,并添加了用于企业级应用的类,如JSP、XML和事务控制等,是Java应用的主要方向。学习J2EE后可以开发基于B/S架构的应用,如京东、天猫、企业办公系统等。

J2ME是Java的微型版本,包含J2SE中的部分类,主要用于开发移动端(手机相关应用),以及机顶盒等设备中的程序。随着时代的发展,其已慢慢趋于淘汰。

Java各平台具有不同的应用领域,学习J2SE是学习J2EE与J2ME的前提与基础。值得注意的是,在2005年,3个版本均进行了更名,舍弃了原命名中的数字“2”,分别更名为JavaSE、JavaEE、JavaME。

1.1.3 Java 的特点

Java的主要特点如下。

(1) 简单、高效。

Java是开源的,其底层由C++实现。换句话说,Java是由C++衍生而来的,与C++十分相似,但Java相较于C++语言要简单得多。因为Java语言在计算机软件开发的过程中,舍弃了C++中难以掌握的功能,如头文件、指针变量、结构体、运算符重载、多重继承等复杂特性,在很大程度上降低了编程的复杂性。此外,Java提供了丰富的工具类库,方便开发程序。这也是Java语言高效的一个体现。

(2) 面向对象。

面向对象是一种先进的编程思想,以对象为基础、以数据为中心,更容易解决复杂的问题。Java是一门纯面向对象的编程语言,更符合人们的思维模式。

(3) 跨平台性。

跨平台性是指Java语言编写的应用程序在不同的系统平台上都可以运行,即“Write Once, Run Anywhere”(一次编译,到处运行),极大地降低了开发难度。跨平台性是Java语言最大的优势,也是Java语言风靡全球的主要原因。

(4) 多线程。

多线程的应用程序能够同时运行多项任务,程序响应更快。Java的特点之一就是内置对多线程的支持,并提供了多线程之间的同步机制,这样使单位时间内,处理的性能得到提升,资源利用率更高。

(5) 安全性。

Java 通常被用在网络环境中，为此，Java 提供了一个安全机制以防止恶意代码的攻击。此外，Java 还为通过网络下载的类添加一个安全防范机制（类 ClassLoader），分配不同的名字空间以防替代本地的同名类，并提供了安全管理机制（类 SecurityManager）。

(6) 健壮性。

Java 语言的强类型机制、异常处理机制、在运行过程中的垃圾自动回收机制等安全检查机制是 Java 程序健壮性的重要保证。

(7) 动态性。

Java 语言的设计目标之一是适用于动态变化的环境。在程序编写过程中可以将需要的类动态地载入运行环境中，有利于软件的升级。



1.1.4 Java 的运行机制

学习使用 Java 语言，除了需要了解 Java 语言的特点，还需要了解程序的运行机制。计算机高级编程语言按其程序的执行方式可分为编译型语言和解释型语言。

编译型语言是指使用专门的编译器，针对特定操作系统将源代码一次性翻译成计算机能识别的机器指令，如 C、C++、FORTRAN、Pascal 等；而解释型语言是指使用专门的解释器，将源代码逐条地解释成特定平台的机器指令，解释一句执行一句，类似于“同声翻译”，如 Python、Ruby、JavaScript 等。Java 语言并不属于上述两种语言类型之一，而是两种语言类型的结合体。

Java 程序运行需要经过编写、编译、运行 3 个步骤。编写是指在 Java 开发环境中进行代码的编写，最终形成扩展名为“.java”的源文件；编译是指使用 Java 编译器（JDK 中自带的编译器 javac.exe）对源文件进行错误排查，检查其是否符合 Java 的语法规则，若源文件符合 Java 的语法规则，则编译后将生成扩展名为“.class”的字节码文件，字节码文件不是纯粹的二进制串，无法在操作系统中直接执行；因此，若想运行程序，则需要使用解释器将字节码文件解释成机器代码，执行并显示结果。Java 的运行机制如图 1-3 所示。

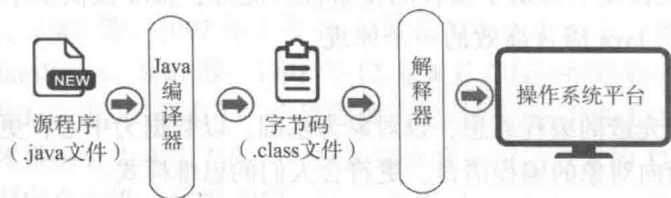


图 1-3 Java 的运行机制

在 Java 语言中最突出的特点就是其对跨平台的支持，在 Java 中如果要想实现跨平台的控制，则主要依靠 Java 虚拟机。图 1-3 中的解释器即 Java 虚拟机（Java Virtual Machine, JVM）。JVM 是由软件和硬件模拟出来的计算机，所有的 Java 程序只要有 JVM 的支持，就可以执行。可以简单地认为：JVM 是 Java 程序实现跨平台的关键，是 Java 程序与系统沟通的“桥梁”。需要注意的是，不同的操作系统需要不同版本的 JVM，其基本原理如图 1-4 所示。

为更通俗地理解 JVM 的原理及其存在意义，可参照 JVM 理解示例，如图 1-5 所示。

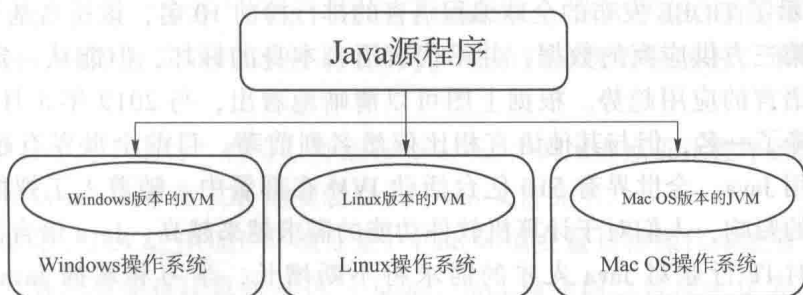


图 1-4 JVM 的基本原理

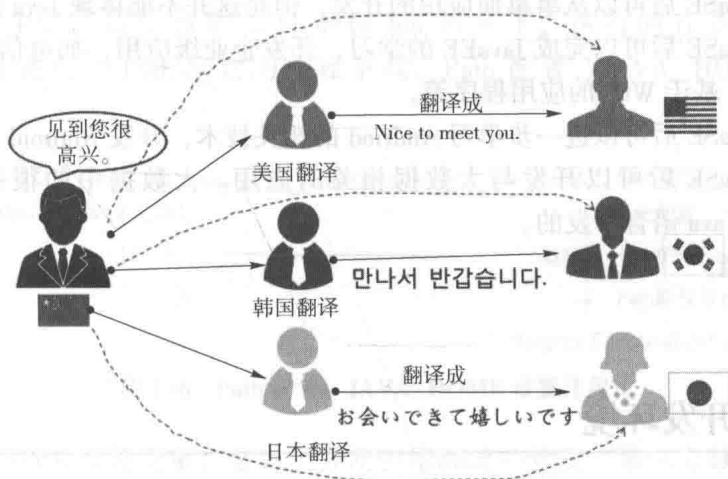


图 1-5 JVM 理解示例

图 1-5 中，当中国人与其他国家的人进行交流时，需要第三方翻译的人，且不同的国家需要不同的翻译，JVM 就相当于上图中的翻译人员，起到了中间桥梁的作用。

1.1.5 Java 的现状和应用前景

Java 不仅可以用来开发大型的桌面应用程序，而且特别适合于网站的开发。目前，Java 已经成为软件设计开发者应当掌握的一门基础语言。TIOBE 发布的 2020 年 8 月全球编程语言热门排行榜如图 1-6 所示。

Aug 2020	Aug 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.98%	+1.83%
2	1	▼	Java	14.43%	-1.60%
3	3		Python	9.69%	-0.33%
4	4		C++	6.84%	+0.78%
5	5		C#	4.68%	+0.83%
6	6		Visual Basic	4.66%	+0.97%
7	7		JavaScript	2.87%	+0.62%
8	20	▲	R	2.79%	+1.97%
9	8	▼	PHP	2.24%	+0.17%
10	10		SQL	1.46%	-0.17%

图 1-6 TIOBE 发布的 2020 年 8 月全球编程语言热门排行榜

图 1-6 展示了 TIOBE 发布的全球编程语言的排行榜前 10 名, 该排名基于全球技术工程师、课程和第三方供应商的数据, 并不代表语言本身的好坏, 但能从一定程度上让读者了解到编程语言的应用趋势。根据上图可以清晰地看出, 与 2019 年 8 月的数据相比, Java 排名虽下降了一名, 但与其他语言相比依然名列前茅。目前全世界有超过 69% 的专职开发人员使用 Java, 全世界有 510 亿台活动 JVM 在部署中。随着人工智能的发展, 以及大数据时代的影响, 人们对于计算机软件功能的需求越来越高, Java 语言涉及的领域将越来越广, 同时 IT 行业对 Java 人才的需求将不断增长, 学习和掌握 Java 已逐步成为共识。

学习 Java 可以有以下 6 个发展方向。

- (1) 学习 JavaSE 后可以从事桌面应用的开发, 但是这并不能体现 Java 的优势。
- (2) 学习 JavaSE 后可以完成 JavaEE 的学习, 开发企业级应用, 如电信、移动、证券等企业信息化平台, 基于 Web 的应用程序等。
- (3) 学习 JavaSE 后可以进一步学习 Android 的相关技术, 开发 Android 平台的软件。
- (4) 学习 JavaSE 后可以开发与大数据相关的应用。大数据中的很多主流框架, 如 Hadoop 等是基于 Java 语言开发的。
- (5) 人工智能。
- (6) 物联网。

1.2 Java 开发环境

Java 的开发环境配置, 一般包含 JDK 的安装、开发工具 (如 Eclipse) 的安装。进行安装之前, 先学习一些相关知识。

JRE (Java Runtime Environment, Java 运行环境), 包括 JVM 和 Java 程序运行所需的核心类库。

JDK (Java Development Kit, Java 开发工具包), 包括 JRE 以及 Java 简单开发工具。Java 简单开发工具包括 `javac.exe`、`java.exe`。可以这么理解: JDK 是 Java 开发人员用来开发 Java 程序的, JRE 负责运行。JDK 包括 JRE, 当我们安装 JDK 软件时, 一般也会选择安装 JRE。

JVM、JRE、JDK 的关系如图 1-7 所示。

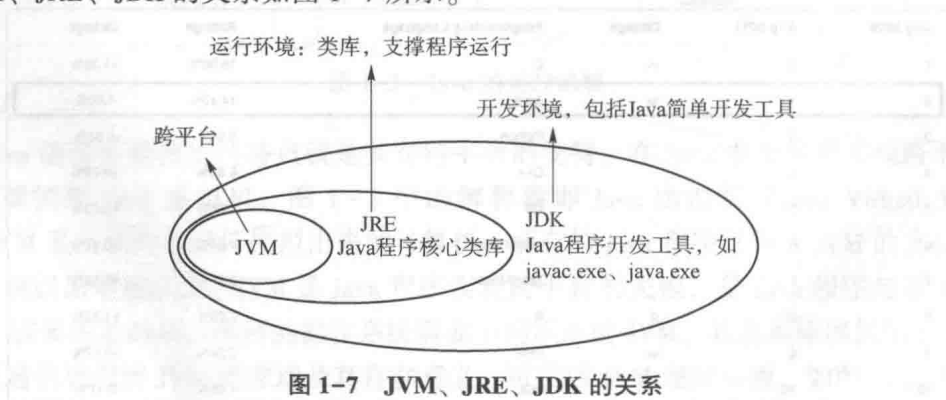


图 1-7 JVM、JRE、JDK 的关系

javac 命令: javac 是“编译”命令,即运行 javac.exe,该程序会启动编译器,将编好的 Java 源码(.java)编译为字节码文件(.class)。编译只是检查语法等,不检查程序的逻辑性。

java 命令: java 是“运行”命令,即运行 java.exe,调用虚拟机让程序运行起来。

javac.exe 和 java.exe 都在 JDK 的安装目录的 bin 文件夹下。

Path 环境变量、JAVA_HOME 环境变量: Path 环境变量是指操作系统查找或执行应用程序的路径。当我们需要在任意目录执行 javac、java 命令时,程序会优先在当前目录下寻找 javac.exe 来执行,但 javac.exe 和 java.exe 只在 JDK 的安装目录的 bin 文件夹下,如果没有配置 Path,程序找不到就直接报错,那我们只能在 bin 目录下编写代码,极不方便。如果配置了 Path,即使源码文件不在 bin 目录下,执行时在当前目录找不到 javac.exe,程序还是会去 Path 配置的路径下找。Path 配置、JAVA_HOME 设置作用如图 1-8 所示。

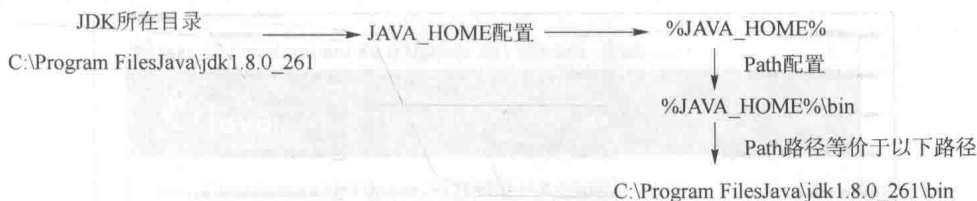


图 1-8 Path 配置、JAVA_HOME 设置作用

配置 JAVA_HOME 环境变量,是为了方便引用和统一管理。第三方软件如 Tomcat,如果要关联 JDK 或者 JRE,都需要找 JAVA_HOME 这个系统变量。

CLASSPATH 环境变量: CLASSPATH 环境变量的作用是当 Java 虚拟机在当前目录下查找或运行.class 字节码文件时,如果没有找到就在 CLASSPATH 环境变量中设置的路径中去找。JDK 1.5 版本之后可以不用再设置 CLASSPATH,但是为保证向下兼用,特别是在安装多版本 JDK 时,建议配置 CLASSPATH 变量,以便 Java 解释器准确调用配置版本的标准类库。

1.2.1 JDK 的安装与配置

Java 的安装与配置有两个步骤:下载并安装 JDK、配置环境变量。

1. 下载并安装 JDK

本书中使用的 JDK 版本为 JDK 1.8,可以从 Oracle 公司的官方网站下载。从其官方网站页面中找到 Java SE 8u261,如图 1-9 所示。选择 JDK Download 进入下载页面,根据操作系统的信息下载所需的安装包,如 Windows 64 位操作系统选择 Windows x64,Mac OS 64 位操作系统选择 Mac OS x64。本书选择 Windows x64 进行下载,如图 1-10 所示。

按网站要求注册用户,下载成功后,便可直接进行安装。首先进入的是 JDK 的安装界面,如图 1-11 所示(本次安装使用默认的安装路径)。JDK 安装完成之后进入 JRE 的安装选择界面,如图 1-12 所示。

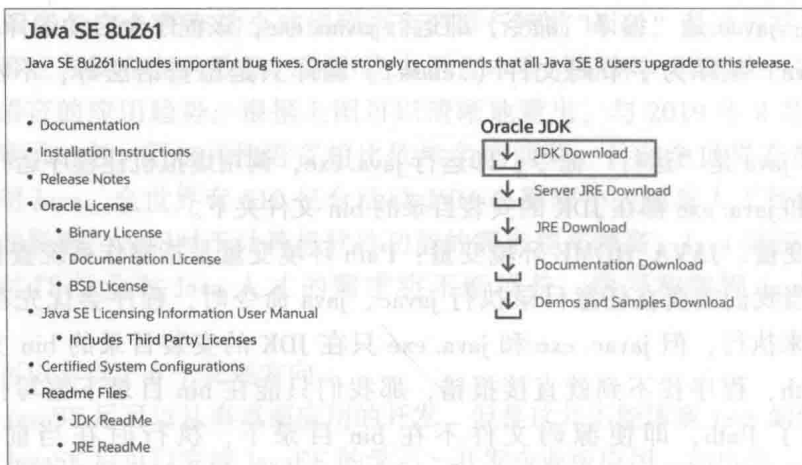


图 1-9 JDK 1.8 的下载页面

Linux x64 RPM Package	121.55 MB	jdk-8u261-linux-x64.rpm
Linux x64 Compressed Archive	156.48 MB	jdk-8u261-linux-x64.tar.gz
macOS x64	205.94 MB	jdk-8u261-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.77 MB	jdk-8u261-solaris-sparcv9tar.Z
Solaris SPARC 64-bit	88.72 MB	jdk-8u261-solaris-sparcv9tar.gz
Solaris x64 (SVR4 package)	134.25 MB	jdk-8u261-solaris-x64.tar.Z
Solaris x64	92.47 MB	jdk-8u261-solaris-x64.tar.gz
Windows x86	154.52 MB	jdk-8u261-windows-i586.exe
Windows x64	166.28 MB	jdk-8u261-windows-x64.exe

图 1-10 JDK 1.8 不同版本列表

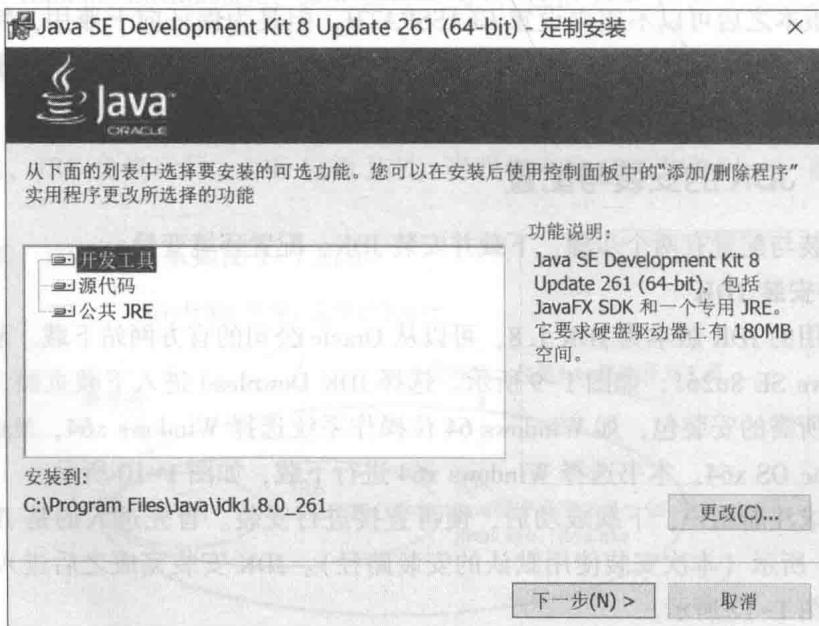


图 1-11 JDK 1.8 开始安装界面