

软件技术系列丛书

Java程序



设计

主 编 / 张 会 兰全祥
副主编 / 吴建兵 何全庆

项目资源库 微课 课件 教学大纲

教案 授课计划 实训报告 题库



西南交通大学出版社

软件技术系列丛书

Java 程序设计

主 编 张 会 兰全祥
副主编 吴建兵 何全庆

西南交通大学出版社

· 成 都 ·

图书在版编目 (C I P) 数据

Java 程序设计 / 张会, 兰全祥主编. —成都: 西南交通大学出版社, 2021.10
ISBN 978-7-5643-8312-1

I. ①J… II. ①张… ②兰… 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 (2021) 第 206914 号

Java Chengxu Sheji

Java 程序设计

主编 / 张会 兰全祥

责任编辑 / 穆 丰
封面设计 / 墨创文化

西南交通大学出版社出版发行
(四川省成都市金牛区二环路北一段 111 号西南交通大学创新大厦 21 楼 610031)
发行部电话: 028-87600564 028-87600533
网址: <http://www.xnjdcbs.com>
印刷: 四川森林印务有限责任公司

成品尺寸 185 mm × 260 mm
印张 26.25 字数 674 千
版次 2021 年 10 月第 1 版 印次 2021 年 10 月第 1 次

书号 ISBN 978-7-5643-8312-1
定价 58.00 元

课件咨询电话: 028-81435775
图书如有印装质量问题 本社负责退换
版权所有 盗版必究 举报电话: 028-87600562

前 言

Java 被称为世界第一的编程语言，近年来在计算机语言排位中一直处于前三名。本书是为了满足学习 Java 编程的人员深入研究 Java 而编写的，吸收了最新的 Java 技术和语法，将编程思想、理论、实践、应用融为一体，并且知识点前呼后应，始终站在 Java 完整程序实现的高度上来分章节讲解，使读者掌握的知识点成为有机的整体。教材中精心设计的案例都来源于企业项目，与理论知识点紧密结合、交互循环贯通，能为读者留下一定的思考空间，培养读者思考问题的能力。

本书共分 13 章，分别讲解了 Java 环境配置，语法基础，控制语句，类与对象，继承、多态和接口，内部类和异常处理，常用实用类，Swing 程序设计，泛型与集合框架，I/O，JDBC 与数据库，多线程，网络编程等内容。

绪论介绍了 Java 语言的发展、优势、运行机制，并介绍了 Java 平台。第 1 章介绍了 Java 开发环境如何进行配置，以及 Java 程序的集成开发环境 Eclipse。第 2~3 章介绍了 Java 语言的基本数据类型、运算符、表达式和数组、Java 语言的控制语句。第 4~6 章是本书的重点之一，介绍了类与对象、继承、接口、多态、内部类和异常。第 7 章讲述了常用实用类，包括字符串类、Math 类、大数字类、日期类、日期格式化类、日历类及正则表达式。第 8 章介绍了 Java 组件及事件处理机制，实现图形化界面编程。第 9 章介绍了泛型与集合框架，讲述了常用的集合框架及应用。第 10 章讲述了如何用输入输出流实现内外存数据间的交互。第 11 章介绍了 Java 与数据库的连接，使用 JDBC 操作数据库，进行预处理及事务处理等内容。第 12 章介绍了多线程的概念及实现方式、线程同步与线程通信等重要内容。第 13 章介绍了关于网络通信的知识，讲述了网络通信基础、UDP 编程、TCP 编程。

本书可作为理工科高等院校“Java 程序设计”课程教材，也可作为软件培训或者自学用书。本书是作者长期从事 Java 研究、开发、教学、实训指导所得的实践经验与心得体会的总结，希望能为广大读者在 Java 编程领域提供一定的帮助。

由于时间仓促及作者水平有限，书中疏漏与不足之处在所难免，恳请读者批评指正。

编 者

2021 年 8 月

目 录

绪 论	1
0.1 计算机系统概述	1
0.2 人机交互方式	2
0.3 Dos 常用命令	2
0.4 计算机语言发展	3
0.5 Java 语言优势	4
0.6 Java 语言发展	5
0.7 Java 语言运行机制	6
0.8 Java 三大平台	8
习 题	8
第 1 章 Java 开发环境配置	9
1.1 Windows10 平台 Java 开发环境配置	9
1.2 第一个 Java 程序	18
1.3 集成开发环境	26
习 题	33
第 2 章 Java 语言基础	35
2.1 标识符与关键字	35
2.2 基本数据类型	36
2.3 运算符与表达式	46
2.4 数 组	53
2.5 小 结	64
习 题	64
第 3 章 Java 控制语句	67
3.1 Java 语句概述	67
3.2 顺序结构	68
3.3 分支结构	69
3.4 循环结构	75
3.5 其他辅助语句	77
3.6 应用举例	78
3.7 小 结	80
习 题	81

第 4 章 类与对象	83
4.1 面向对象	83
4.2 使用 Java 类描述事物	85
4.3 封装	94
4.4 构造方法	100
4.5 构造代码块	102
4.6 this 关键字	104
4.7 static 关键字	107
4.8 Java 包	117
习题	118
第 5 章 继承、多态与接口	120
5.1 类和类之间的常见关系	120
5.2 继承的特点	125
5.3 子类的继承性	126
5.4 super 关键字	128
5.5 重写 (Override)	130
5.6 instanceof 关键字	132
5.7 final 关键字	133
5.8 抽象类	134
5.9 继承关系实现多态	136
5.10 接口 (Interface)	146
习题	150
第 6 章 内部类与异常	152
6.1 内部类概述	152
6.2 成员内部类	154
6.3 局部内部类	158
6.4 匿名内部类	159
6.5 异常	162
习题	177
第 7 章 常用实用类	178
7.1 字符串类	178
7.2 正则表达式	182
7.3 Math 类	186
7.4 随机数相关类和方法	186
7.5 大数字类	188
7.6 日期类 java.util.Date	191
7.7 日期格式化类 java.text.DateFormat	192

7.8	日历类 java.util.Calendar	193
	习 题	198
第 8 章	Java 组件及事件处理	200
8.1	Java Swing 概述	200
8.2	窗 口	202
8.3	常用组件与布局	208
8.4	处理事件	217
8.5	对话框	227
	习 题	234
第 9 章	泛型与集合框架	237
9.1	泛 型	237
9.2	集合概述	245
9.3	Collection	246
9.4	Map	263
9.5	Iterator	268
	习 题	272
第 10 章	输入输出流	273
10.1	流类概览	273
10.2	I/O 类基本继承结构	277
10.3	四个重要抽象父类	277
10.4	常用 I/O 类	279
10.5	RandomAccessFile 类	306
	习 题	308
第 11 章	JDBC 与数据库	310
11.1	JDBC 概述	310
11.2	JDBC API	311
11.3	JDBC 编程	317
11.4	示 例	324
	习 题	332
第 12 章	多线程	333
12.1	线程概述	333
12.2	实现线程的两种方式	335
12.3	线程的生命周期及状态转换	343
12.4	线程的操作	347
12.5	多线程同步	360

12.6 线程通信	369
习 题	376
第 13 章 网络通信	377
13.1 网络通信基础	377
13.2 网络编程 API	382
13.3 UDP 编程	396
13.4 TCP 编程	403
习 题	409
参考文献	411

绪 论

【学习要求】

- 了解计算系统的组成及基本工作原理；
- 掌握常用的 Dos 命令；
- 了解计算机语言发展史；
- 了解 Java 语言的优势及发展史；
- 掌握 Java 语言运行机制；
- 了解 Java 三大平台。

0.1 计算机系统概述

计算机 (computer)，全名是电子计算机，俗称电脑，是现代一种用于高速计算的电子计算机器，既可以进行数值计算，又可以进行逻辑计算，同时具有存储记忆功能。计算机是能够按照预先设定的程序运行，并能实现自动、高速处理海量数据的现代化智能电子设备。1946年，第一台电子计算机 ENIAC 在美国诞生。随着计算机的普及和发展，计算机大致经历了电子管计算机、晶体管计算机、集成电路计算机、大规模集成电路计算机四个阶段。个人计算机的发展是从台式机、便携式计算机再发展到便于人类携带及使用的平板计算机和智能手机。计算机由硬件系统和软件系统所组成，没有安装任何软件的计算机称为裸机。

0.1.1 硬 件

迄今为止，计算机都是采用计算机之父匈牙利科学家冯·诺伊曼提出的冯诺依曼体系结构的电子数字计算机，所有的计算机都具有存储程序的功能。计算机由运算器、控制器、存储器 and 输入输出设备组成，如图 0.1 所示。

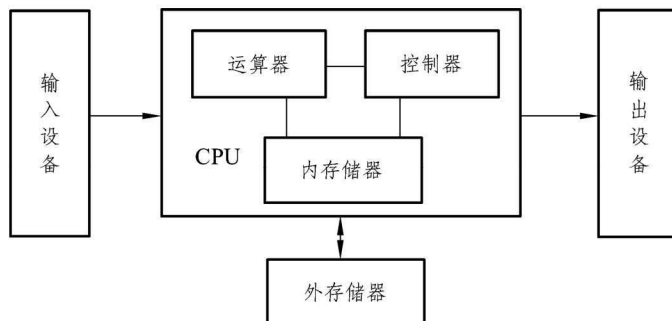


图 0.1 计算机工作原理

计算机中程序的工作原理是程序首先以二进制代码的形式事先存放到计算机内存存储器中，再按程序在存储器中存放程序的首地址执行其第一条指令，之后就按照该程序的执行顺序自动执行其他指令，直至程序执行结束。

0.1.2 软件

软件是按照一定顺序编写的计算机数据与指令的集合。通常把没有安装软件（software）的计算机硬件（hardware）称为裸机。裸机只有在安装了软件后才能被计算机用户有效使用。计算机软件分为系统软件和应用软件。

在计算机硬件系统之上安装操作系统是为了实现向计算机用户隐藏计算机硬件操作细节，通过系统软件来管理计算机系统中各种独立的硬件，使各硬件可以协调工作。系统软件包括操作系统和一系列基本的工具（比如编译器，数据库管理，文件系统管理，驱动管理，网络连接等工具）。常见的操作系统有用于 PC 机的 Windows 系统、苹果计算机专用操作系统 Mac OS X、多用于服务器的 Linux 系统、UNIX 操作系统，以及智能终端的 Android 和 IOS 系统。

应用软件是为了某种特定的用途而被开发的软件。应用软件通常是定制程序，是为某个应用领域专门开发的软件，比如绘图工具、图像处理软件、Office 办公软件等。

0.2 人机交互方式

随着操作系统的发展，用户与计算机的交互方式从最初的命令行（Dos）交互方式发展到图形用户界面方式。图形用户界面在实现人机交互方面较大程度地提高了计算机的使用效率，但是一些更复杂的操作只能用命令行交互方式来操作，因此有必要掌握常用的 Dos 命令。

0.3 Dos 常用命令

Dos 是 disk operating system 的缩写，即磁盘操作系统，是一个基于磁盘管理的操作系统。随着 Windows 操作系统的盛行，Dos 系统虽然逐渐退出舞台，但使用 Dos 命令进行远程调试或处理一些复杂问题时会达到事半功倍的效果，同时熟悉常用 Dos 命令有助于对 Java 程序进行调试或测试。

0.3.1 目录操作命令

盘符（如 C:）：命令提示符进入指定的盘符。

dir：查看当前目录下所有的文件以及文件夹。

md：创建目录。

rd：删除空目录但不能删除非空目录。

cd: 命令提示符切换到指定目录。
cd.: 退回到当前目录的上一级目录。
cd \: 退回到当前目录的根目录, 即回到当前目录的盘符下。

0.3.2 文件操作命令

echo file content>文件名: 写文本到指定文件。
如: echo "hello world">hello.txt, 实现将 hello world 写入到 hello.txt 文件中。
type 文件名: 显示文件内容命令。如: type hello.txt。
del 文件名或目录名: 删除文件或目录及其下的所有文件。

0.3.3 其他

exit: 退出 Dos 命令行。
cls: 清屏幕命令, 清除当前 Dos 屏幕上的所有显示内容, 同时光标置于屏幕左上角。
: 通配符, 表示任意多个任意字符, 如.txt 表示当前路径下的所有的 txt 文件, *.*表示当前目录下的所有文件。
tab: 对 Dos 命令后的文件夹名称或文件名称自动补全。
方向键↑↓: 找回写过的命令。

0.4 计算机语言发展

计算机语言 (computer language) 是人与计算机间通信的语言, 是人与计算机间传递信息的媒介。人类通过一些字符和一系列的语法规则组成计算机指令集合的程序来使计算机完成各项工作, 而程序是通过计算机语言来进行编写的, 因此程序设计语言也可以说是计算机语言。计算机语言的发展大致经历三个阶段: 机器语言、汇编语言、高级语言, 其中面向过程语言和面向对象语言合称为计算机高级语言。

0.4.1 机器语言

机器语言是针对特定型号的计算机全部的二进制代码指令集合。用机器语言编写的程序可以直接被计算机识别并执行, 但由于每台计算机的指令系统各不相同, 因此, 在一台计算机上执行的程序, 无法直接在另一台计算机上执行, 可移植性差, 由于机器语言能直接被计算机识别执行, 因此其执行效率是所有计算机语言中最高的。由于机器语言采用二进制代码指令编写程序, 程序员在编写程序控制计算机操作时极不方便, 因此促使了汇编语言的产生。

0.4.2 汇编语言

为解决使用机器语言编程的不便, 早期程序用一些方便记忆的英文字符串代替一个个特

定的二进制指令(如用 ADD 代表加法指令, MOV 代表数据传递指令),因此形成了汇编语言。汇编语言相较于机器语言易于程序员理解、纠错及维护,但计算机不能直接识别汇编助记符(如 ADD, MOVE),因此需要一种程序将这些符号翻译成二进制的机器语言,这种翻译程序称为汇编程序。汇编语言同机器语言一样依赖机器硬件,移植性差,但相对于高级语言其在计算机上的执行效率更高。汇编语言程序是针对特定计算机硬件进行编写的,能准确发挥计算机硬件的功能和特长,程序精炼而质量高,所以至今仍是一种常用而强有力的软件开发语言。

0.4.3 面向过程语言

为了解决机器语言和汇编语言编写的程序可移植性差、依赖计算机硬件等问题,人们又创造了面向过程的计算机高级语言。面向过程计算机语言接近人类自然语言,具有更强的表达能力,可方便地表示数据的运算和程序的控制结构,能更好地描述各种算法,而且容易学习掌握。1971年,第一个面向过程的程序设计语言——Pascal 语言的出现,标志着结构化程序设计时期的开始;1972年,贝尔实验室在 B 语言和 BCPL 语言的基础上,开发出著名的 C 语言。

0.4.4 面向对象语言

20世纪80年代初,在软件工程设计思想上产生了面向对象的程序设计语言。面向对象思想更接近人的思维方式,把现实世界抽象为类,类产生对象,对象具有属性和行为。最早出现的面向对象程序设计语言是 Smalltalk 语言;1983年,在 C 语言的基础上产生了面向对象的 C++语言;在 C++的基础上,1995年 SUN 公司推出了 Java 语言;在 Java 的基础上,2000年微软推出了 C#语言。此外,面向对象语言还有 Objective-C(苹果公司专用语言)、PHP、Python 等语言。

0.5 Java 语言优势

Java 语言与 C++语言相比较,虽然其产生晚于 C++语言,但 Java 是一种纯面向对象程序设计语言,而 C++是从 C 的基础上发展起来,所以既面向对象又面向过程。Java 语言的特点是摒弃了 C++中的指针与内存管理及多重继承,其采用单一继承,内存管理由 Java 虚拟机完成。而 C++语法复杂,由程序员管理内存,容易产生内存泄漏。

Java 语言是第一个跨平台的编程语言。Java 语言能做到“一次编写,处处运行”(Write once, Run everywhere)。而 C++语言,编译生成的可执行文件(.exe)是针对某一系统生成的可执行文件,一旦程序运行平台发生变化,C++程序必须重新编译,才能运行,无法实现跨平台。

Java 语言多用于大型网站和 App 后台服务器,因此 Java 语言是 Java Web 和 Android 应用程序开发的基础语言,应用范围广阔,使用 Java 语言的程序员会越来越多。

0.6 Java 语言发展

1990年年末，Sun 公司成立了一个由 James Gosling（Java 之父）领导的“Green 计划”项目组，其目的是为智能家电（电视机、微波炉）编写一个通用控制系统。Green 项目组最初使用 C++ 语言来开发，但发现 C++ 和可用的 API（应用程序接口）在某些方面存在很大的问题，并且 C++ 语法太复杂，缺少垃圾回收系统、分布式和多线程等功能，可移植性差，最终项目组放弃了 C++。于是项目组研发了一种全新的 Oak 语言（用于家用电器等小型系统的编程语言），来解决如电视机、微波炉、烤面包机等家用电器的控制和通信问题，由于当时智能化家电的市场需求没有预期的高，Sun 公司放弃了 Green 计划。就在 Oak 几近失败之时，在 1994 年 James Gosling 等人决定将 Oak 技术应用于互联网，将该语言改造为网络编程语言，Oak 语言当时在互联网上得到了很大的应用和发展，Oak 语言就是 Java 语言的前身。由于 Oak 已被注册，James 于 1995 年将 Oak 更名为 Java。Java 语言的产生过程如图 0.2 所示。

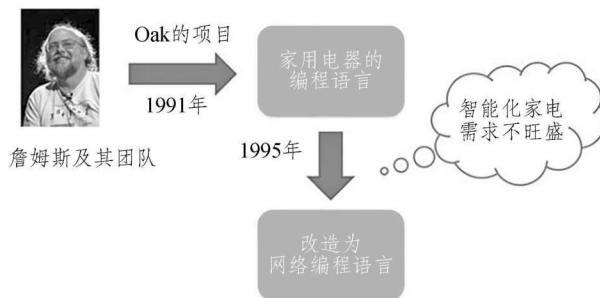


图 0.2 Java 语言的产生

Java 语言随着互联网技术的应用得到了快速发展，逐渐成为重要的网络编程语言。图 0.3 展示了 Java 的发展轨迹和历史变迁。

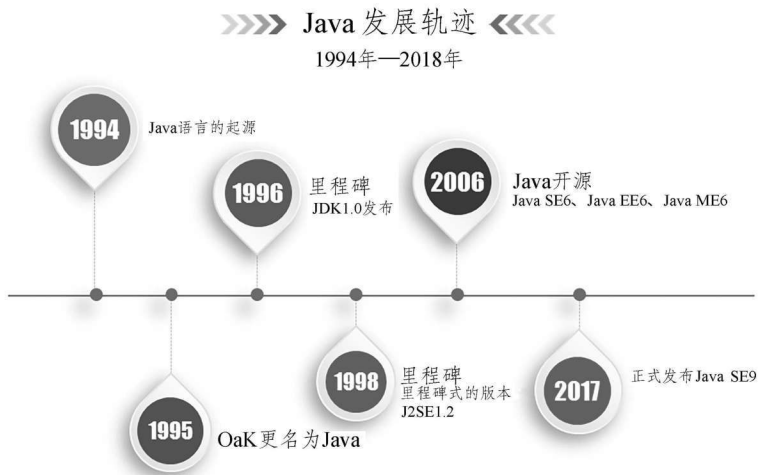


图 0.3 Java 发展时间线

1996 年是 Java 语言里程碑的一年，在这一年，Java 开发人员熟悉的 JDK 1.0 发布了，Java

语言有了第一个正式版本的运行环境。JDK 1.0 (Java Development Kit 1.0,Java 开发工具包 1.0) 版本包括 Java 虚拟机、网页应用程序 (可以嵌套在网页中运行)、用户界面组件 (通过用户界面组件可以开发窗口应用程序)。SUN 直接把 JDK 连同源代码免费发布到互联网上。在短短几个月的时间互联网上出现了大量的 Java 小程序。

1998 年,对 Java 语言来说,又是一个里程碑。Java 正式发布了 J2SE 1.2,在这个版本中,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、面向企业级开发的 J2EE 和面向手机等移动终端开发的 J2ME。J2EE 在企业中得到了广泛应用,涌现了大量基于 Java 语言的开源框架,如 Struts、Hibernate、Spring 等。

在 2004 年 10 月,SUN 发布了 JDK 1.5。J2SE 更名为 Java SE,J2EE 更名为 Java EE,J2ME 更名为 Java ME。Java 进入了 Java 5 时代,实现了第二次飞跃。

2009 年 4 月,IT 巨头 Oracle 收购 SUN,取得 Java 的版权,从此 Java 属于 Oracle 公司。2010 年,Java 之父的 James Gosling 离开了 Oracle。

2011 年 7 月,Oracle 公司发布 Java 7 的正式版。

2014 年 3 月,Oracle 公司发布 Java 8 的正式版。Java 仍然是世界第一的开发平台,并且 Java 用户组的数量以每年 10%的速度增长。

2017 年 9 月,Java 9 正式版发布。

2018 年 3 月,Java 10 正式版发布。

2018 年 9 月,Java 11 正式版发布。

2019 年 3 月,Java 12 正式版发布。

2019 年 9 月,Java 13 正式版发布。

0.7 Java 语言运行机制

0.7.1 高级语言运行机制

机器语言编写的程序能被计算机直接识别并执行,但机器语言相对于人来说不直观,难记难写,因此产生了计算机高级语言,高级语言分为面向过程与面向对象编程语言,高级语言很直观,类似人类自然语言,编程人员使用起来方便,但高级语言编写的程序不能直接被计算机识别并执行,需要一种翻译程序,把高级语言编写的程序转换成计算机能直接识别的二进制代码。而高级语言的翻译程序的转换方式分为编译型和解释型。

编译型是指在源程序执行之前,先将程序源代码“翻译”成目标代码(机器语言),再将该目标代码通过链接组装成当前计算机硬件可以识别的可执行程序的过程。生成可执行文件可以脱离其语言环境独立执行,使用比较方便,效率较高。但应用程序一旦需要修改,必须先修改源代码,再重新编译生成新的可执行文件才能执行,如果只有目标文件而没有源代码,修改很不方便。常见的 Pascal、C 语言、C++ 等高级语言都是编译型语言。

解释型执行方式类似于我们日常生活中的“同声翻译”,应用程序源代码一边由相应语言的解释器逐行“翻译”成目标代码(机器语言),一边执行。解释方式不能生成可独立执行的可执行文件,同时应用程序不能脱离其解释器,因此效率比较低。但这种方式比较灵活,只

需要在一个特定平台上提供对应的解释器，就可以对源程序进行“翻译”运行，每个特定的平台编译器只负责将源程序翻成本地的机器码。那么解释型语言相对编译型语言，容易实现程序跨平台性，比如 JavaScript、Ruby、MatLab 等。

0.7.2 Java 虚拟机

Java 语言的一个非常重要的特点就是与平台的无关性，Java 的跨平台是基于 Java 虚拟机 (JVM) 的。高级语言如果要在不同的平台上运行，需要编译生成不同平台下的目标代码。而引入 Java 语言虚拟机后，Java 语言在不同平台上运行时不需要重新编译。Java 编译程序首先将 Java 源程序编译生成在 Java 虚拟机上运行的目标代码 (字节码)，再在 Java 虚拟机中执行字节码，java 虚拟机能够实现把字节码文件解释成不同平台上的机器指令执行。这就要求在不同的平台上安装不同的 Java 虚拟机，从而就可以在多种平台上不加修改地运行同一 Java 字节码文件，实现 Java 的跨平台。

由于需要在不同平台上安装不同的 Java 虚拟机，因此 Java 提供了不同平台下的 Java 开发工具包 (JDK) 和 Java 运行时环境 (JRE) 版本 (其中 JRE 内含 JVM)，常见的有 Linux 版、mac OS 版和 Windows 版。因此计算机平台不同，安装的 Java 虚拟机也就不同，通过 Java 虚拟机来实现 Java 跨平台。

0.7.3 运行机制

Java 语言是一类特殊的高级语言，既具有编译型又具有解释型语言的特性。Java 源程序 (.java 文件) 需要先经过编译过程生成字节码文件 (.class 文件) 后，再经过安装的不同平台上的 Java 虚拟机对字节码文件进行解释执行。在解释执行的过程中，Java 虚拟机首先将编译好的字节码文件由类加载器加载到内存，然后虚拟机针对加载到内存中的 Java 类进行解释执行。

因此，Java 程序是由虚拟机负责解释执行的，而非操作系统。这样做的好处是可以实现跨平台，对于同一段 Java 程序 (或字节码文件)，如若要在不同的操作系统上运行，只需安装不同版本的虚拟机即可，从而做到一次编译，处处运行。Java 运行机制如图 0.4 所示。

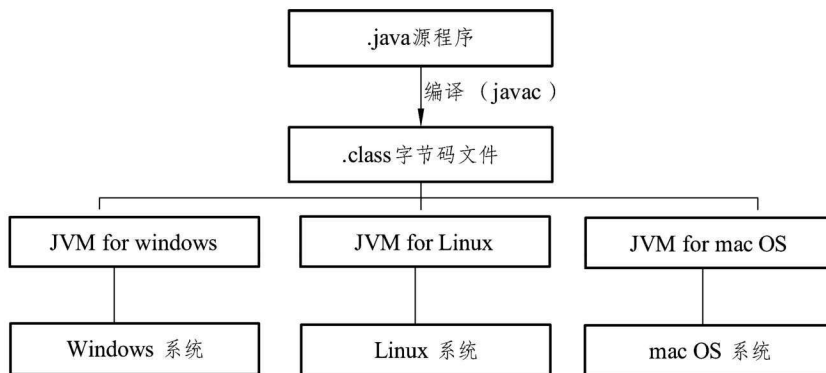


图 0.4 Java 运行机制

Oracle 公司给 Java 开发人员提供的 JDK (Java Development Kit, Java 开发工具包) 包含

Java 的开发工具及 JRE (Java Runtime Environment, Java 运行环境)。因此安装了 JDK, 就不用再单独安装 JRE 了。

JRE 包括 Java 虚拟机 (JVM Java Virtual Machine) 和 Java 程序所需的核心类库等 (如果只是运行一个开发好的 Java 程序而不开发 Java 程序, 则计算机中只需要安装 JRE 即可)。注意不同的操作系统需要安装不同的 JDK 或 JRE。

JDK = JRE + 开发工具集 (例如 javac 编译工具打包工具 jar.exe 等)

JRE = JVM + Java SE 标准类库

三者之间的关系如图 0.5 所示。

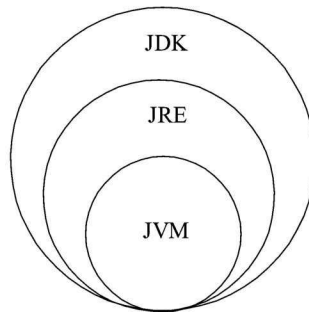


图 0.5 JDK, JRE, JVM 之间的关系

0.8 Java 三大平台

Java SE: Java Standard Edition, 即 Java 标准版, 提供基本的底层支持, 是一个 Java 平台的名称。最新版本的 JDK 14, 是基于 Java SE 平台的开发程序发行版本, JDK 是相对开发者而言的。JDK 包含 JRE, JRE 包含 JVM, JVM 包含 JIT (Just In Time, 即时编译器)。

Java EE: Java Enterprise Edition, 即 Java 的企业版, 前身是 J2EE, 主要构建企业应用系统。

Java ME: Java Micro Edition, 即 Java 微型版本, 主要用于嵌入式系统开发, 现在主要用于开发手机游戏。

习 题

1. Java 之父是谁?
2. J2EE 是什么?
3. 简要说明 Java 程序的运行机制。
4. 说明 JDK、JRE、JVM、JIT 之间的关系。

第 1 章 Java 开发环境配置

【学习要求】

掌握 Windows 平台下 JDK 安装与配置；

掌握第一个 Java 程序 HelloWorld；

掌握编译和执行命令；

了解注释及编程风格；

了解 Java 集成开发环境。

Java 开发环境配置是学习 Java 的基本内容。本章将以 Windows 10 平台为基础介绍 Java 开发环境的配置，第一个 Java 程序的编辑、编译和执行过程及集成开发环境和使用。

1.1 Windows10 平台 Java 开发环境配置

Windows 操作系统是当今最为流行的 PC（个人计算机）操作系统。2014 年 4 月，微软公司停止对 Windows XP 操作系统的服务支持，Oracle 的 Java 8 也停止了对 Windows XP 的支持。如果在 Windows XP 系统上安装 JDK 8 将出现“无法找到入口”的错误，如图 1.1 所示。如果需要在 Windows XP 系统中开发 Java 程序，只能安装 JDK 7 及以下的版本。



图 1.1 JDK 8 不支持 Windows XP

目前，大多数 PC 安装的是 Windows 7 和 Windows 10 操作系统。本章以 Windows 10 操作系统为平台来详细讲解 Java 开发环境配置。

1.1.1 下载 Java 开发包 JDK

JDK 是 Java 开发工具包，是针对 Java 开发人员的工具包，用于开发 Java 程序。JDK 包括了 Java 运行环境 JRE、Java 工具和 Java 基础类库，学会安装 JDK 是学习 Java 编程的第一步。JDK 是免费的，直接在 Oracle 官网下载安装使用，无须破解。