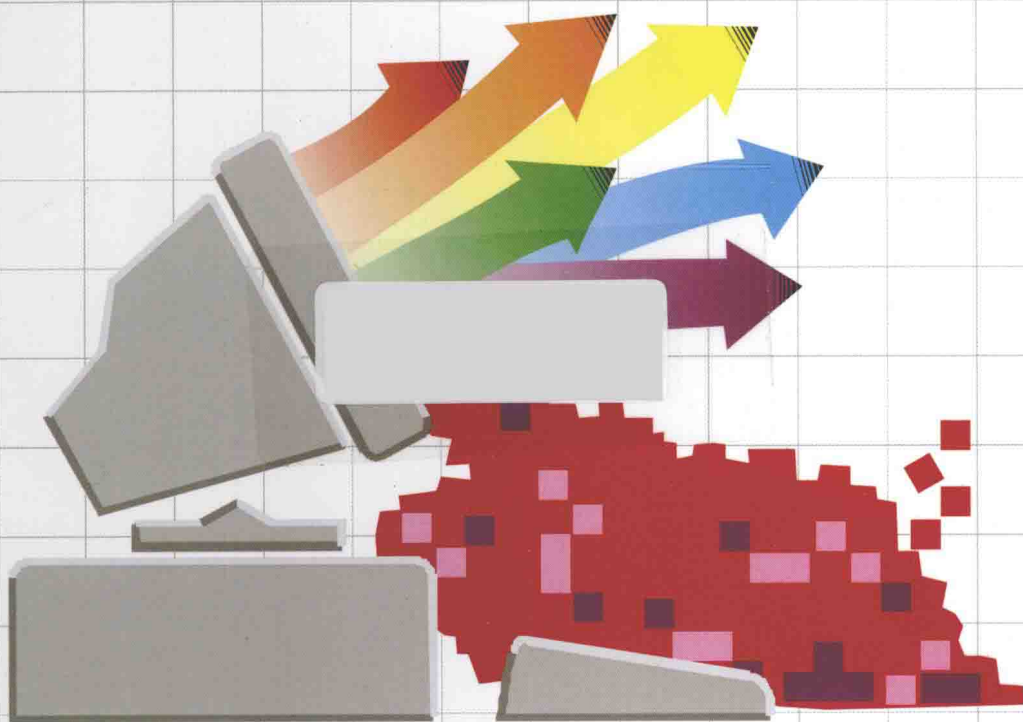


Java

程序设计详解

张 伟◎编著



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

Java 程序设计详解

张 伟 编著

东南大学出版社
· 南京 ·

内容提要

Java 语言从诞生以来一直是应用最广的开发语言,并拥有最广泛的开发人群。现在,Java 已经不再简单地是一门语言,而是一个完整的、系统的开发平台,在 web 开发、移动互联网开发等方面都占据着核心的地位。

本书深入介绍了 Java 编程的最核心内容,强调实战,对比《Java 核心技术》等大部头技术书籍,本书最大的特点是,对一些在实际开发中很少用到,影响读者入门,又比较浪费读者时间的知识点,进行了删减。全书内容覆盖了 Java 的基本语法结构、Java 的面向对象特征、异常处理、Java 的 IO 流体系、Java 多线程编程、集合框架体系、Java 泛型、Java GUI 编程、JDBC 数据库编程、Java 网络通信编程和 Java 反射机制、Java 注释。覆盖了 java.awt、java.lang、java.io、java.nio、java.sql、java.text、java.util、javax.swing 等包下绝大部分类和接口。

本书不是单纯从知识角度来讲解 Java,而是从解决问题的角度来介绍,所以书中介绍了大量实用案例,如开发仿记事本的文本编辑器、多线程下载工具、聊天程序、抓图程序、锁屏程序、网络传送文件程序。这些案例既能让读者巩固每章的知识,又可以让读者学以致用,激发编程自豪感,进而引爆内心的编程激情。章节和程序循序渐进,语言通俗易懂,注重实例,程序很好调试,注解充分,因此非常易懂,适合自学。

图书在版编目(CIP)数据

Java 程序设计详解 / 张伟编著. —南京:东南大学
出版社,2014.3

ISBN 978-7-5641-4795-2

I. ①J… II. ①张… III. ①Java 语言—程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 053593 号

Java 程序设计详解

出 版 人 江建中
责任编辑 丁志星
出版发行 东南大学出版社
(江苏省南京市四牌楼2号东南大学校内 邮政编码210096)
网 址 <http://www.seupress.com>
印 刷 南京玉河印刷厂
开 本 787 mm×1092 mm 1/16
印 张 28.25
字 数 680 千字
版次印数 2014 年 3 月第 1 版 2014 年 3 月第 1 次印刷
印 数 1~2000 册
书 号 ISBN 978-7-5641-4795-2
定 价 78.00 元

(东大版图书若有印装质量问题,请直接与营销中心调换。电话—83791830)

致 Java 初学者

在此给新手说几点学习建议

学习要循序渐进,对知识点的理解要精、要细。万丈高楼平地起,基础压倒一切,所以新手学习 Java 的时候,一定要好好学习基础知识,不要去追求速度而不求甚解。感觉自己什么都会了,其实可能还有很多不理解。

学习的过程中理解最重要,在理解的基础上多写代码!

学习过程中不要贪多贪快,一个初学者开口闭口 J2EE,谈框架,没有多大的意义,不要一入手就贪多贪全,从基础开始踏实的一步一步来。J2EE 的体系不是那么的简单,不是单纯的 JSP,现在很多人认为精通了 JSP 似乎就离精通 J2EE 不远了,这是不对的。

稳扎稳打,一步步地将 Java 基础学精学透,Java 基本语法结构、Java 的面向对象特征、异常处理、Java 的 IO 流体系、Java 多线程编程、集合框架、Java 泛型、JDBC 数据库编程、Java 网络通信编程和 Java 反射机制,这些是 Java 语言的基础、核心。要将这些章节的每个知识点都搞懂,然后写出程序,测试输出,只有经历过懂了——写出程序——再测试的过程才能深入深刻的理解并加强记忆。

学习完 Java 基础之后,还要学很多,除了 Java 语言本身之外,HTML、CSS、JavaScript、JSP 都要学习,学完 JSP 后,还要学习很多的框架,如:Struts, Spring, Hibernate, iBatis 等,学习 Java 之前,要做好学无止境的准备。但我们不要过早去追框架部分,而要重点掌握 Java 基础知识,如继承、IO、反射等,掌握了基础,去理解应用那些框架就会很简单的。

关于调试

我们要切实的重视程序的编写和调试,一个只会看代码的初学者,是成不了一个程序员的。

一个程序员 60%的工作量是在调试程序。能否快速调试好程序,也是一个程序员是否成熟的重要标志。

初学者所面临最大的困难恐怕不是不能开始写程序,而是写出的程序错误百出,不能运行。不能运行也就影响理解。这对于初学者是非常大的打击。能否想办法减少编程中的错误,对于初学者而言,能否真正进入到精彩的编程世界中,就意味着能否坚持下去。

下面就初学者如何减少犯错,如何在出错后改正阐述一下心得体会:

● 注意大小写及全角半角区别

Java 是大小写敏感的语言,int 数据类型,如果你写成 Int 那就一定是错的,通不过编译。输入的时候,不要将英文符号输入成中文符号,字符的全角还是半角状态是需要注意的,写程序时,要求的是半角符号。

● 熟悉 Java 的命名规则,减少语法错误的出现

程序中的类名、方法名、属性名、变量名、对象名、关键字都有其约定俗成的命名规则,从

开始学习 Java 就注意到名字的规定,自然就能够减少错误的发生。

Java 混合使用大小写字母来构成变量和函数的名字,如类名 `ArrayIndexOutOfBoundsException` 中的每一个逻辑断点都有一个大写字母来标记,非常容易理解,但是不同名称又有区别,具体如下:

类名——首字母大写,其后每一个单词首字母大写,如 `String`,`InputStream`。

方法名、属性名、变量名、对象名——首字母小写,其后每一个单词首字母大写。

关键字、包名——全部小写。

常量——全部大写。

● 学会看 Java 的错误提示,快速改正运行时的错误

Java 的错误提示比较清晰准确。Java 通常只提示一处错误信息,其格式如下:

程序:

```
int []a=new int[10];
```

```
a[10]=33;
```

运行报错:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:10
```

异常出现在"main"线程,数组下标越界异常:错误原因:下标 10 越界

```
at com.myt.test.Test.main(Test.java:11)
```

出错的包.类.方法(文件名:错误所在行数)。

看懂错误提示的格式,每次出错后多看,多积累,不仅能够提高排错速度,还能够少犯错误。

● 注意一些初学者常犯的错误,如下:

1. 不要使用未定义的标识符,如未声明的变量,未定义的方法等等。
2. "if();"及"for();"后的";"造成空语句,从而 if for 空执行。
3. 漏掉了语句结束符";"。
4. 数组的边界超界。
5. 局部变量未赋初值。
6. ()及{}不配对。

● 做程序讲究写代码,而不是抄代码

很多初学者打开两个编辑器,一左一右,看着左边的示例在右边敲,敲完一运行,跳出 n 多错误,千头万绪,焦头烂额,信心全失。

正确的方法是,看懂案例后独立键入,键入时将整个程序按照思路及逻辑分成若干片段,每个片段实现一个功能,敲完每段即测试运行,如果有错误,调试排错,然后再敲下一段。键入过程中,如果有不会的或忘记的,打开案例看,思考,然后再继续本程序的键入。这样的做法,将代码分成多段累积键入,多次运行,即有利于对程序的理解,也有利于减少错误的出现。因为如果你看不懂程序就不可能将代码分离成若干可以独立测试的代码段。熟能生巧,多敲代码,多看代码,多思考,多和高手探讨问题,敲写代码不出错,很容易实现。

目 录

第 1 章 Java 程序设计概述	(1)
1.1 Java 简介	(1)
1.2 名字起源	(1)
1.3 历史版本	(2)
1.4 三大领域	(2)
1.5 语言特点	(3)
1.6 运行机制	(3)
1.7 JVM (Java Virtual Machine)	(4)
1.8 Java 运行环境(JRE)	(5)
1.9 JDK	(5)
1.10 安装 Java 开发环境	(5)
1.11 配置环境变量	(8)
1.12 第一个 Java 程序	(12)
1.13 打包	(13)
1.14 生成 API 文档	(14)
1.15 Java 帮助和 API 文件的使用	(15)
1.16 掌握的关键点	(16)
1.17 课后作业	(17)
第 2 章 Eclipse 的开发工具	(18)
2.1 Java 开发工具	(18)
2.2 Eclipse 的使用	(20)
2.3 Java 程序开发	(27)
2.4 小结	(34)
2.5 课后作业	(34)
第 3 章 基本程序设计	(35)
3.1 注释	(35)
3.2 数据类型	(36)
3.3 溢出	(43)
3.4 变量定义	(44)
3.5 运算符	(45)
3.6 语句块	(49)
3.7 分支语句	(50)

3.8	循环语句	(52)
3.9	小结	(55)
3.10	作业	(56)
3.11	作业解答	(56)
第 4 章	面向对象编程	(60)
4.1	面向对象	(60)
4.2	重载(Overload)	(64)
4.3	this	(65)
4.4	构造函数	(66)
4.5	static 关键字	(67)
4.6	默认值	(69)
4.7	初始化块	(70)
4.8	对象克隆	(71)
4.9	数组	(77)
4.10	常用类	(84)
4.11	小结	(90)
4.12	作业	(91)
4.13	作业解答	(91)
第 5 章	继承	(96)
5.1	包(package)	(96)
5.2	继承	(97)
5.3	访问控制	(98)
5.4	重写(override)	(99)
5.5	多态	(99)
5.6	造型	(100)
5.7	super	(101)
5.8	final	(103)
5.9	构造顺序	(103)
5.10	垃圾回收	(104)
5.11	传值与传引用	(108)
5.12	==与 equals	(110)
5.13	toString()方法	(113)
5.14	小结	(114)
5.15	作业	(115)
5.16	作业解答	(115)

第 6 章 抽象类、接口、内部类	(119)
6.1 抽象类(abstract class)	(119)
6.2 接口	(121)
6.3 常用接口和类	(126)
6.4 综合应用	(138)
6.5 关键字 final	(140)
6.6 内部类	(141)
6.7 小结	(149)
6.8 作业	(149)
6.9 作业解答	(150)
第 7 章 异常	(154)
7.1 异常是什么	(154)
7.2 异常处理机制	(159)
7.3 throw 与 throws	(164)
7.4 自定义异常类	(165)
7.5 重写和异常	(166)
7.6 链式异常	(167)
7.7 异常堆栈填充	(168)
7.8 异常分类	(169)
7.9 综合运用	(171)
7.10 异常处理注意事项	(172)
7.11 小结	(175)
7.12 作业	(176)
7.13 作业解答	(176)
第 8 章 线程	(179)
8.1 进程	(179)
8.2 线程	(180)
8.3 时间片轮换	(181)
8.4 创建线程	(181)
8.5 线程的调度和控制	(184)
8.6 线程组	(192)
8.7 线程同步	(194)
8.8 线程通信	(198)
8.9 Timer 和 TimerTask	(204)
8.10 死锁	(205)
8.11 小结	(207)
8.12 作业	(208)

8.13 作业解答	(208)
第 9 章 IO	(213)
9.1 File 类	(213)
9.2 流	(216)
9.3 字节输入流	(218)
9.4 字节输出流	(220)
9.5 字符流	(228)
9.6 随机访问	(242)
9.7 小结	(244)
9.8 作业	(244)
9.9 作业解答	(245)
第 10 章 集合框架	(254)
10.1 Collection	(254)
10.2 List 接口	(256)
10.3 Set 接口	(262)
10.4 Collection 迭代	(266)
10.5 MAP 接口	(268)
10.6 Collections 工具类	(273)
10.7 小结	(274)
10.8 作业	(275)
10.9 作业解答	(275)
第 11 章 网络编程	(279)
11.1 InetAddress	(279)
11.2 URL	(281)
11.3 URLConnection	(283)
11.4 Socket 编程	(288)
11.5 UDP 的 Socket 编程	(305)
11.6 小结	(308)
11.7 作业	(309)
11.8 作业解答	(309)
第 12 章 图形界面程序设计	(313)
12.1 抽象窗口工具集(AWT)	(313)
12.2 Swing	(313)
12.3 组件(Component)	(314)
12.4 容器(Container)	(314)

12.5	JFrame 类	(314)
12.6	JPanel 类	(315)
12.7	布局管理器	(317)
12.8	AWT 绘图	(324)
12.9	事件	(327)
12.10	事件类型	(330)
12.11	多重监听器	(336)
12.12	事件适配器(Event Adapter)	(337)
12.13	普通内部类	(338)
12.14	匿名内部类	(339)
12.15	常用组件	(340)
12.16	练习	(349)
12.17	小结	(354)
12.18	作业	(355)
12.19	作业解答	(355)
第 13 章	反射机制	(358)
13.1	获得 Class 对象	(358)
13.2	得到类名和类的修饰符	(359)
13.3	找出一个类的所有的父类	(361)
13.4	判断类实现的接口	(362)
13.5	得到类的字段	(363)
13.6	得到类的构造函数	(365)
13.7	取得方法信息	(367)
13.8	创建对象	(369)
13.9	取得字段的值	(373)
13.10	调用方法	(374)
13.11	综合练习一	(375)
13.12	综合练习二	(378)
13.13	小结	(380)
13.14	作业	(381)
13.15	作业解答	(381)
第 14 章	Java 数据库编程	(383)
14.1	MySQL 数据库简介	(383)
14.2	基础 sql 语句	(394)
14.3	JDBC 简介	(396)
14.4	加载驱动	(396)
14.5	Java 数据库操作基本流程	(397)

14.6	PreparedStatement	(398)
14.7	CallableStatement	(399)
14.8	控制事务	(401)
14.9	批量更新	(402)
14.10	元数据	(403)
14.11	滚动结果集	(405)
14.12	可更新结果集	(406)
14.13	Dao 设计模式	(407)
14.14	大字段处理	(414)
14.15	三层结构	(422)
14.16	小结	(423)
14.17	作业	(424)
14.18	作业解答	(424)
第 15 章	注解	(431)
15.1	Java 内置注解	(431)
15.2	自定义注解	(432)
15.3	枚举	(433)
15.4	泛型	(434)
15.5	组合模式	(435)
15.6	单例设计模式	(437)
参考文献		(440)

第1章 Java 程序设计概述

Java 语言从 1996 年正式发布以后就引起了很多的关注。经过十几年的发展,现在 Java 语言已经成为最主流的开发语言,特别是在互联网应用和智能手机软件开发中占有处理主流的地位,无数的程序员都已经加入了 Java 开发的行列。

本章主要介绍一下 Java 语言,讲解一下 Java 语言的特点和应用,并完成一个最简单的 Java 程序的编写。



1.1 Java 简介

Java 是一种可以编写跨平台应用程序的面向对象的程序设计语言,是 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 JavaSE, JavaEE, JavaME)的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于个人电脑、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。

Java 编程语言是个简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言。Java 平台是基于 Java 语言的平台。



1.2 名字起源

Java 的名字的来源:Java 是印度尼西亚爪哇岛的英文名称,因盛产咖啡而闻名。据 James Gosling(詹姆斯·高斯林)回忆,最初这个为 TV 机顶盒所设计的语言在 Sun 内部一直被称为 Green 项目。我们的新语言需要一个名字。Gosling(高斯林)注意到自己办公室外一棵茂密的橡树 Oak,这是一种在硅谷很常见的树。所以他将这个新语言命名为 Oak,但 Oak 是另外一个注册公司的名字。这个名字不可能再用了。在命名征集会上,大家提出了很多名字。最后按大家的评选次序,将十几个名字排列成表,上报给商标律师。排在第一位的是 Silk(丝绸)。尽管大家都喜欢这个名字,但遭到 James Gosling 的坚决反对。排在第二和第三的都没有通过律师这一关。只有排在第四位的名字得到了所有人的认可和律师的通过,这个名字就是 Java。10 多年来,Java 就像爪哇咖啡一样誉满全球,成为实至名归的企业级应用平台的霸主。



1.3 历史版本

1995年5月23日,Java语言诞生。1996年1月,第一个JDK-JDK1.0诞生。1996年4月,10个最主要的操作系统供应商声明将在其产品中嵌入Java技术。1996年9月,约8.3万个网页应用了Java技术来制作。1997年2月18日,JDK1.1发布。1997年4月2日,JavaOne会议召开,参与者逾一万人,创当时全球同类会议规模之纪。1997年9月,Java Developer Connection社区成员超过十万。1998年2月,JDK1.1被下载超过2,000,000次。1998年12月8日,Java2企业平台J2EE发布。1999年6月,Sun公司发布Java的三个版本:标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。2000年5月8日,JDK1.3发布。2000年5月29日,JDK1.4发布。2001年9月24日,J2EE1.3发布。2002年2月26日,J2SE1.4发布,自此Java的计算能力有了大幅提升。2004年9月30日下午6时,J2SE1.5发布,成为Java语言发展史上的又一里程碑。为了表示该版本的重要性,J2SE1.5更名为Java SE 5.0。2005年6月,JavaOne大会召开,Sun公司公开Java SE 6。此时,Java的各种版本已经更名,以取消其中的数字“2”:J2EE更名为Java EE,J2SE更名为Java SE,J2ME更名为Java ME。2006年12月,Sun公司发布JRE6.0。2009年4月20日,甲骨文以74亿美元收购Sun,取得Java的版权。2011年7月,甲骨文公司发布Java7的正式版。



1.4 三大领域

- J2SE

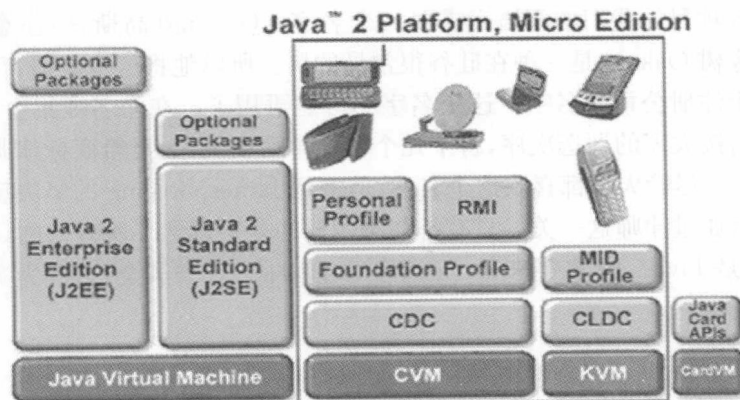
Java 2 Platform, Standard Edition 标准版

- J2EE

Java2 Platform Enterprise Edition 企业级版本

- J2ME

Java 2 Platform, Micro Edition 微小版



如上图, J2ME 主要是做嵌入式设备, 如手机上的软件开发。J2SE 做标准的图形界面程序开发, 即电脑上运行的应用程序。J2EE 主要用来做企业级应用的开发, 即 B/S 结构程序的开发。



1.5 语言特点

- 跨平台
- 面向对象
- 编译加解释型
- 拥有较好的性能
- 健壮性
- 多线程
- 安全性

跨平台: Java 开发的应用程序, 既可以在 Windows 上运行, 又可以在 Linux 等其他操作系统下运行, 这就是跨平台特性, Java 程序经过虚拟机的翻译, 编写一次的程序, 可运行于各种平台之间, 而其他语言, 如 C 语言无法实现跨平台的应用。

面向对象, Java 语言是面向对象的语言, 所有类的顶层父类都是 Object 类, 是个单根结构的程序设计语言, 在 Java 早期的版本中, Java 的基本数据类型还不是对象, 在 1.5 之后, Java 有了装箱和拆箱机制后, 基本数据类型和类对象之间可以进行自动转换, Java 基本上就算是完全面向对象的程序设计语言了。

编译加解释型, Java 程序的源代码, 经过编译后, 生成 class 文件, Java 虚拟机解释 class 文件来执行 Java 程序, 因此 Java 程序要经过解释才能执行, 不像 C 语言等编译语言编译后, 直接在操作系统上执行。

拥有较好的性能, Java 语言虽然经过解释才执行, 但是解释的是编译后的 class 文件, 因此其性能比直接解释源代码执行的语言性能要高很多。经过开发团队不懈的努力, Java 语言在性能方面, 已经是比较优秀的了。

健壮性, Java 语言中内存垃圾自动回收, 由虚拟机来负责内存垃圾的回收, 不像 C 语言, 由程序员来负责回收而经常有忘记回收的情况, 因此 Java 语言是健壮的。

多线程, Java 语言有自己的多线程实现机制, 而 C 语言等, 是通过调用操作系统的 API 来实现, 在这一点上, Java 较其他语言, 有很大的优势。

安全性, 安全性也是 Java 语言的优势之一, Java 有沙箱机制来保证代码不能进行非法的、不安全的访问和操作。



1.6 运行机制

1.6.1 编译型和解释型

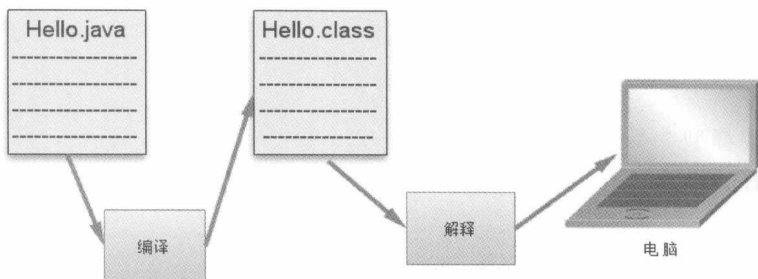
计算机高级编程语言有两类:

● 编译型:特定的编译器和操作系统,将高级语言代码一次性“翻译成”该平台执行的机器码,并包装成该平台可执行格式。

● 解释型:特定的解释器将源码翻译成特定平台的机器码并立即执行。解释性语言一般不会进行整体性的编译和链接处理。

编译型的代表语言是 C 语言,解释型的代表语言是 Ruby 语言。

1.6.2 Java 执行过程



Java 语言,将两种类型的语言进行了融合,运行前首先需要进行编译,编译成与平台无关的字节码(class 文件)。然后运行时,再对字节码文件进行解释执行。JVM 就是可以运行字节码的虚拟计算机。

如上图所示,Java 语言的源文件为.java,经过 Java 编译器的编译,生成.class 文件,.class 文件经过解释器的解释运行,最后在电脑上执行。

编译 Java 源文件,使用命令 javac,而执行 Java 源程序,使用命令 java。

.java 文件——javac 进行编译为 class 文件——class——文件解释执行——特定平台的机器码。

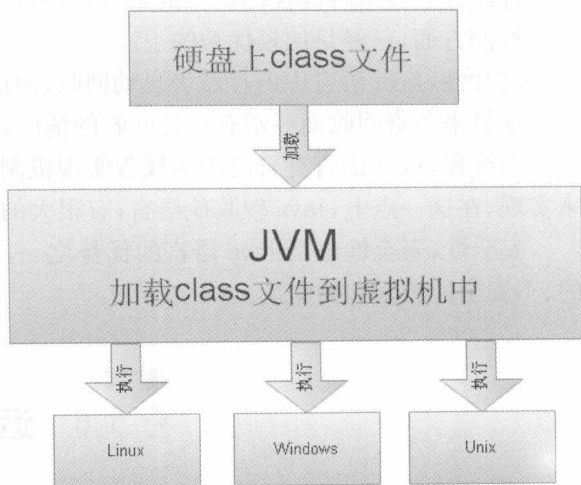


1.7 JVM (Java Virtual Machine)

JVM 就是 Java 虚拟机,它是为了支持 Java 跨平台特性。有了 Java 虚拟机,Java 就可以与操作系统无关地运行在各个操作系统之中。虚拟机就是在台计算机上由软件或硬件模拟的计算机。

Java 编译器将 Java 源文件针对 Java 虚拟机产生 class 文件,因此是独立于平台的,class 文件只和虚拟机有关,只要虚拟机能解释就可以了。Java 解释器负责将 Java 虚拟机的代码在特定的平台上运行。Java 虚拟机负责 class 文件的解释和执行,因此 Java 才能跨平台。

如右图所示,JVM(Java 虚拟机)处于



class 文件(字节码文件)和操作系统之间,联系二者,因此.class 文件得以跨平台运行。



1.8 Java 运行环境(JRE)

Java 运行环境组成:JRE = JVM + Runtime Interpreter

Java 运行环境的三项主要功能:

- 加载代码:由 Class Loader 完成,将.class 文件加载到虚拟机中。
- 校验代码:由 Bytecode Verifier 完成校验代码的功能。
- 执行代码:由 Runtime Interpreter 完成执行。

Java 程序要执行,必须由 JVM 负责加载代码和校验代码,由 Runtime Interpreter(运行时解释器)负责执行代码。Java 程序要执行必须有 JRE 才可以。



1.9 JDK

JDK 即 Java Development Kit,就是 Java 开发工具包,JDK 比 JRE 多了 javac 和 java 等众多的命令。要进行 Java 软件开发,必须要安装 JDK。



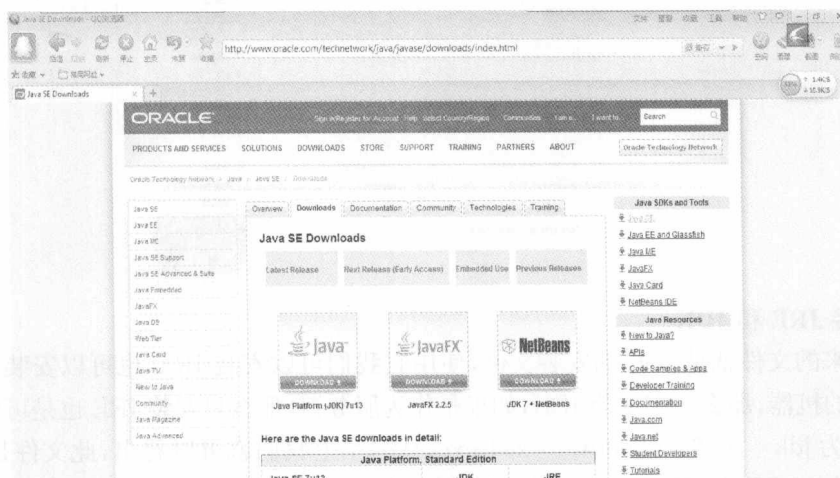
1.10 安装 Java 开发环境

Java 程序要编译和运行,需要 Java 开发工具包(JDK)的支持,首先要下载 JDK 并安装它。

1.10.1 下载 JDK

JDK 包含 JRE 及常用编译等命令,这些是开发过程中必备的,它的下载地址为:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

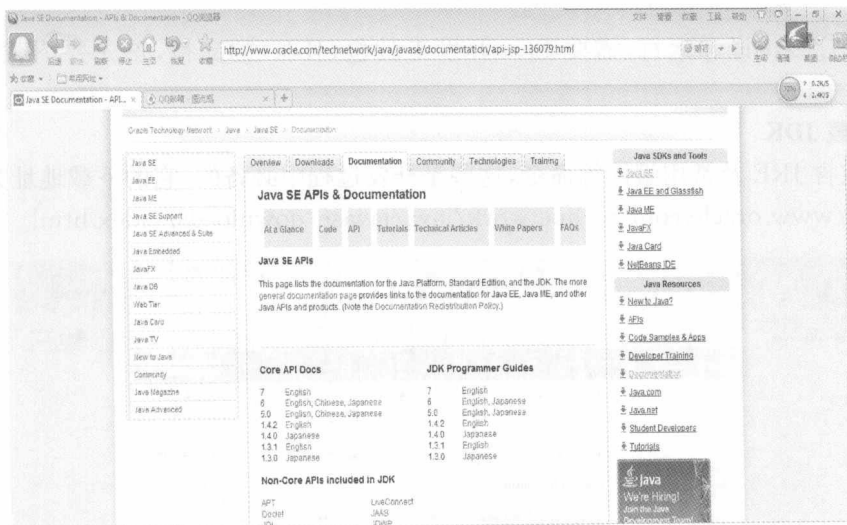


此页面包括 Java JDK7 的下载,还包括 JavaFX 的下载及 NetBeans 开发工具的下载。我们只需要下载第一项即可。点击后进入如下界面:

Java SE Development Kit 7u13		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
☐ Accept License Agreement ☒ Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	106.64 MB	jdk-7u13-linux-i586.rpm
Linux x86	92.97 MB	jdk-7u13-linux-i586.tar.gz
Linux x64	104.77 MB	jdk-7u13-linux-x64.rpm
Linux x64	91.69 MB	jdk-7u13-linux-x64.tar.gz
Mac OS X x64	143.71 MB	jdk-7u13-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.55 MB	jdk-7u13-solaris-i586.tar.Z
Solaris x86	91.95 MB	jdk-7u13-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.54 MB	jdk-7u13-solaris-x64.tar.Z
Solaris x64	14.96 MB	jdk-7u13-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	135.83 MB	jdk-7u13-solaris-sparc.tar.Z
Solaris SPARC	95.28 MB	jdk-7u13-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.89 MB	jdk-7u13-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.58 MB	jdk-7u13-solaris-sparcv9.tar.gz
Windows x86	88.74 MB	jdk-7u13-windows-i586.exe
Windows x64	90.41 MB	jdk-7u13-windows-x64.exe

在此界面中我们需要同意 Oracle 公司对 Java SE 的协议。然后选择我们需要的操作系统的版本即可。名字中 7u13,表示 JDK7 第 13 次更新版本。如果是 64 位操作系统,下载时请选择 Windows x64 版本的 JDK 进行下载。

同时下面的页面中,我们可以将 Java 的 API 文档和帮助向导下载下来。



1.10.2 安装 JRE 和 JDK

下载下来的文件就是 JDK 的安装文件,使用它我们可以安装 JRE,也可以安装 JDK,如果是用来开发的机器,那么一定要装 JDK,如果是作为服务器,那么只安装 JRE 也是可以的。

文件名为 jdk-7u13-windows-x64.exe,从名字上我们就可以看出,此文件是 JDK7 的安装文件,第 13 次更新,Windows64 位版本的安装程序。接着我们将此 JDK 安装在我们的