



重新设计
面向未来的
人工智能代码实现。

将带领读者
快速入门
中文分词
软件系统。



猎兔搜索
工程师数十年技术
积淀与升华。



算法
基本原理与具体搜
索技术
的完美结合。



中文分词
软件系统



重新设计面向未来
人工智能代码实现

Java

中文文本信息



Java

中文文本信息处理

罗刚 张子宪 崔智杰◆编著

从海量到精准

清华大学出版社

Java 中文文本信息处理——从海量到精准

罗 刚 张子宪 崔智杰 编 著

清华大学出版社
北 京

内 容 简 介

本书以让零基础的读者通过自学完成一个中文分词系统为目标,从 Java 基础语法开始讲解,然后介绍文本处理相关的数据结构和算法,最后介绍如何实现文本切分和词性标注。

本书是介绍业界热门的以 Java 开发中文分词技术的唯一书籍。本书选取相关领域的经典内容,深入理解和挖掘,也综合了实践性强的创新想法,适合对软件开发感兴趣的青少年或者大学生阅读和学习。

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 中文文本信息处理——从海量到精准/罗刚,张子宪,崔智杰编著. —北京:清华大学出版社,2017
ISBN 978-7-302-46936-0

I. ①J… II. ①罗… ②张… ③崔… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 073994 号

责任编辑:杨作梅 宋延清

装帧设计:杨玉兰

责任校对:张 瑜

责任印制:李红英

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

印装者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:25.5 字 数:619 千字

版 次:2017 年 5 月第 1 版 印 次:2017 年 5 月第 1 次印刷

印 数:1~3000

定 价:56.00 元

产品编号:073068-01

前言

“前门到了，请在后门下车。”把“前门”标注成地名就容易理解这句话了。从种地到买菜、买房、养生保健以及投资理财等，都可以用到中文分词等文本信息挖掘技术。

各行业都在构建越来越复杂的软件系统，很多系统都会用到文本处理技术。但是即使在计算机专业，也有很多人对文本信息处理相关技术不太了解。其实，学习相关技术的门槛并不高。而本书就是为了普及相关开发而做的一次新的尝试，其中也结合了作者自己的研究成果，希望为推动相关应用的发展做出贡献。

本书借助计算机语言 Java 实现中文文本信息处理，试图通过恰当的数据结构和算法来应对一些常见的文本处理任务。相关代码可以从清华大学出版社的网站下载。

本书的第 1 章到第 3 章介绍了相关的 Java 开发基础。第 4 章介绍处理文本所用到的有限状态机基本概念和具体实现。第 5 章介绍相关的基础数据结构。第 6 章到第 9 章介绍中文分词原理与实现。

书中的很多内容来源于作者的开发和教学实践。作者的实践经验还体现在相关的其他书中，如《自己动手写搜索引擎》、《自然语言处理原理与技术实现》、《自己动手写网络爬虫》、《使用 C# 开发搜索引擎》、《解密搜索引擎技术实战》等。相对于作者编写的其他书籍，本书更加注意零基础入门。

学习是个循序渐进的过程。可以在读者群中共同学习。群体往往比单个人有更多的智慧产出。为了构建出更好的技术群体，请加读者 QQ 群(453406621)交流。希望快速入门的读者也可以参加相关培训。这本书最开始是为一位从苏州专门来北京现场学习的学员入门中文分词而编写。感谢他为编写本书提供的帮助。

也希望通过本书能结识更多的同行。有您真诚的建议，我们会发展得更好。例如，通过与同行的交流，让我们的数量、日期等量化信息的提取工具更加成熟。当前，语义分析等文本处理技术仍然需要更深入的发展，来更好地支持各行业的智能软件开发。

本书由罗刚、张子宪、崔智杰编著，参与本书编写的还有石天盈、张继红、童晓军，在此一并表示感谢。感谢开源软件和我们的家人、关心我们的老师和朋友、创业伙伴，以及选择猎兔自然语言处理软件的客户多年来的支持。

编者

目 录

第 1 章	Java 软件开发	1
1.1	背景	3
1.1.1	好身体是一切成功的保证	3
1.1.2	路线图	4
1.1.3	Java	4
1.2	软件工具	7
1.2.1	搜索引擎	7
1.2.2	Windows 命令行	8
1.2.3	机器翻译	9
1.2.4	Linux	10
1.2.5	源代码比较工具	11
1.3	Java 基础	11
1.3.1	准备开发环境	11
1.3.2	Eclipse	13
1.4	本章小结	17
第 2 章	结构化程序设计	19
2.1	基本数据类型	19
2.2	变量	20
2.2.1	表达式执行顺序	22
2.2.2	简化的运算符	23
2.2.3	常量	24
2.3	控制结构	25
2.3.1	语句	25
2.3.2	判断条件	25
2.3.3	三元运算符	27
2.3.4	条件判断	27
2.3.5	循环	31
2.4	方法	36
2.4.1	main 方法	41
2.4.2	递归调用	41
2.4.3	方法调用栈	42
2.5	数组	42
2.5.1	数组求和	45
2.5.2	计算平均值举例	45
2.5.3	前趋节点数组	46
2.5.4	快速复制	47
2.5.5	循环不变式	49
2.6	字符串	50
2.6.1	字符编码	52
2.6.2	格式化	53
2.6.3	增强 switch 语句	54
2.7	数值类型	54
2.7.1	类型转换	58
2.7.2	整数运算	59
2.7.3	数值运算	60
2.7.4	位运算	61
2.8	安装 Java	69
2.8.1	服务器端安装	69
2.8.2	自动安装 Java	70
2.9	提高代码质量	72
2.9.1	代码整洁	72
2.9.2	单元测试	72
2.9.3	调试	73
2.9.4	重构	73
2.10	本章小结	74
第 3 章	面向对象编程	77
3.1	类和对象	77
3.1.1	类	78
3.1.2	类方法	78
3.1.3	类变量	79
3.1.4	实例变量	79
3.1.5	构造方法	82
3.1.6	对象	84
3.1.7	实例方法	87
3.1.8	调用方法	89
3.1.9	内部类	89
3.1.10	克隆	90



3.1.11	结束.....	91	3.11.6	关于对象不可改变.....	139
3.2	继承.....	92	3.12	日期.....	140
3.2.1	重写.....	92	3.13	大数对象.....	141
3.2.2	继承构造方法.....	94	3.14	给方法传参数.....	142
3.2.3	接口.....	95	3.14.1	基本类型和对象.....	143
3.2.4	匿名类.....	98	3.14.2	重载.....	145
3.2.5	类的兼容性.....	98	3.15	文件操作.....	146
3.3	封装.....	98	3.15.1	文本文件.....	146
3.4	重载.....	99	3.15.2	二进制文件.....	149
3.5	静态.....	100	3.15.3	文件位置.....	152
3.5.1	静态变量.....	100	3.15.4	读写 Unicode 编码的文件 ...	153
3.5.2	静态类.....	100	3.15.5	文件描述符.....	155
3.5.3	修饰类的关键词.....	101	3.15.6	对象序列化.....	156
3.6	枚举类型.....	101	3.15.7	使用 IOUtils.....	160
3.7	集合类.....	105	3.16	Java 类库.....	161
3.7.1	动态数组.....	105	3.16.1	使用 Java 类库.....	162
3.7.2	散列表.....	106	3.16.2	构建 JAR 包.....	163
3.7.3	泛型.....	109	3.16.3	使用 Ant.....	167
3.7.4	Google Guava 集合.....	112	3.16.4	生成 JavaDoc.....	167
3.7.5	类型擦除.....	112	3.16.5	ClassLoader.....	168
3.7.6	遍历.....	114	3.16.6	反射.....	172
3.7.7	排序.....	117	3.17	编程风格.....	173
3.7.8	lambda 表达式.....	119	3.17.1	命名规范.....	173
3.8	比较.....	119	3.17.2	流畅接口.....	174
3.8.1	Comparable 接口.....	119	3.17.3	日志.....	175
3.8.2	比较器.....	120	3.18	IDEA.....	181
3.9	SOLID 原则.....	122	3.19	实例.....	181
3.10	异常.....	123	3.20	本章小结.....	183
3.10.1	断言.....	123	第 4 章	处理文本.....	185
3.10.2	Java 中的异常.....	124	4.1	字符串操作.....	185
3.10.3	从方法中抛出异常.....	126	4.2	有限状态机.....	188
3.10.4	处理异常.....	128	4.2.1	从 NFA 到 DFA.....	190
3.10.5	正确使用异常.....	130	4.2.2	DFA.....	194
3.11	字符串对象.....	132	4.2.3	DFA 交集.....	197
3.11.1	字符对象.....	135	4.2.4	DFA 并集.....	203
3.11.2	查找字符串.....	135	4.2.5	有限状态转换.....	204
3.11.3	修改字符串.....	136	4.3	本章小结.....	207
3.11.4	格式化.....	136			
3.11.5	常量池.....	137			

第 5 章	数据结构	209	第 7 章	最长匹配分词	311
5.1	链表	209	7.1	正向最大长度匹配法	312
5.2	树算法	210	7.2	逆向最大长度匹配法	316
5.2.1	标准 Trie 树	211	7.3	处理未登录串	320
5.2.2	链表 Trie 树	221	7.4	开发分词	324
5.2.3	二叉搜索树	223	7.5	本章小结	326
5.2.4	数组形式的二叉树	227	第 8 章	概率语言模型的分词方法	327
5.2.5	三叉 Trie 树	233	8.1	一元模型	328
5.2.6	三叉 Trie 树交集	244	8.2	整合基于规则的方法	334
5.2.7	Trie 树词典	245	8.3	表示切分词图	336
5.2.8	平衡 Trie 树	249	8.4	形成切分词图	342
5.2.9	B 树	250	8.5	数据基础	344
5.3	双数组 Trie	251	8.5.1	文本形式的词表	344
5.4	队列	257	8.5.2	数据库词表	348
5.4.1	链表实现的队列	257	8.6	改进一元模型	349
5.4.2	优先队列	258	8.7	二元词典	352
5.4.3	找出前 k 个最大的元素	261	8.8	完全二叉数组	357
5.5	堆栈	262	8.9	三元词典	360
5.6	双端队列	264	8.10	N 元模型	361
5.7	散列表	268	8.11	N 元分词	362
5.7.1	快速查找的散列表	269	8.12	生成语言模型	368
5.7.2	HashMap	272	8.13	评估语言模型	369
5.7.3	应用散列表	276	8.14	概率分词的流程与结构	370
5.7.4	开放式寻址	279	8.15	本章小结	371
5.7.5	布隆过滤器	282	第 9 章	词性标注	373
5.7.6	SimHash	284	9.1	数据基础	376
5.8	图	286	9.2	隐马尔科夫模型	377
5.8.1	表示图	287	9.3	存储数据	385
5.8.2	遍历图	295	9.4	统计数据	390
5.9	大数据	297	9.5	整合切分与词性标注	392
5.10	本章小结	297	9.6	知识型词性序列标注	396
第 6 章	算法	299	9.7	本章小结	396
6.1	贪婪法	299	参考资源		397
6.2	分治法	301	后记		398
6.3	动态规划	302			
6.4	在中文分词中使用动态规划算法	303			
6.5	本章小结	310			

第1章 Java 软件开发

很多系统都可以看成是智能系统。例如，聊天软件，可以帮助人远程协作。人本身是有智能的，所以整体上也可以看成是一个远程协作的智能系统。

高铁系统出问题了，修改代码就可以。但是，在某些事件中，执法出了问题，需要细化相关法律条款吗？可以把法律看成是用人来执行的代码。

小李以前做过销售，想转行做软件开发。后来，他遇到了本书的作者。

把重复性的劳动变成创建一个可以在计算机上执行并完成指定任务的程序，这样一个创造性的过程，叫作程序设计。例如，早期的信息收集工作是由人工完成的，现在可以写一个爬虫程序，从互联网自动获取信息。

现代社会中，软件已经相当普及。例如，现在的电网都是由计算机控制的，通过相关软件发出指令，来调节电压、相位等关键参数。在美国，每三名工程师中就有一名软件工程师。即使不专职从事软件开发工作，会写代码也是一种有竞争优势的技能。管理人员会越来越多地了解技术，而技术人员也会越来越多地参与管理决策。

转行就是让生活从一个平衡状态过渡到另外一个平衡状态。也许人们已经习惯了不会软件开发的生活。完成从不会到会的过程，往往需要几个月的时间。小李在学习开发之前，已经辞掉了他原来的工作，为了尽快学会，他把周末的时间也用上了。

很多时候，我们以往的经历对于学习技术并无太大的帮助。例如，这本书就是试图通过一般生活常识来帮助学习新的知识。经常地，学习中用到的例子对我们的实际开发并没有帮助，而这本书则试图让例子更实用。

写程序俗称“敲代码”。虽然有的程序员自嘲为打字员，但并不需要特别学会盲打，代码是通过键盘敲进去的。只要坚持每天用电脑 1 小时以上，自然就能熟悉键盘了。

小李的高考成绩并不好，所以他没有条件上好的大学。有些声称“提供教育而非培训”的大学，其实是对学生的一种伤害，他们往往没有看到当今这个领域的真实情况。学院派通常认为“对于学生来说，他们有很多的机会去学习那些陈旧的编程方法，他们可以在日后的工作实践中按需学习那些知识”。与此同时，招聘经理们却痛恨这种现状，因为要想把一个从不错的大学里走出来的应届毕业生培养成一个完全能工作的软件开发人员，平均所需的时间是一年半。所以，许多公司不喜欢要刚毕业的学生。

如果想成为一名职业开发人员，就需要拿到 Offer 并安全度过试用期。而很多时候，通过面试不是问题，但顺利通过试用期往往更重要。不要让自己有这样的遗憾：曾经有一份好的 Offer 摆在我面前，但是我没有珍惜，等到失去的时候才后悔莫及。小李找罗老师做他的后盾，罗老师能帮他解决学习中遇到的一些技术难题。



有两个和尚住在隔壁(所谓隔壁,就是隔壁那座山)——他们分别住在相邻的两座山上的庙里。这两座山之间有一条小溪,于是这两个和尚每天都会在同一时间下山,去溪边挑水,久而久之,他们变成了好朋友。

就这样,时间在每天挑水中不知不觉已经过了五年。突然有一天,左边这座山的和尚没有下山挑水,右边那座山的和尚心想“他大概睡过头了。”便不以为然。

哪知道第二天,左边这座山的和尚还是没有下山挑水,第三天也一样。过了一个星期还是一样。直到过了一个月,右边那座山的和尚终于受不了,他心想“我的朋友可能生病了,我要过去拜访他,看看能帮上什么忙。”

于是,他便爬上了左边这座山,去探望他的老朋友。

等他到了左边这座山的庙,看到他的老友之后,大吃一惊,因为他的老友正在庙前打太极拳,一点也不像一个月没喝水的人。他很好奇地问:“你已经一个月没有下山挑水了,难道你可以不用喝水吗?”左边这座山的和尚说:“来来来,我带你去看。”于是他带着右边那座山的和尚走到庙的后院,指着一口井说:“这五年来,我每天做完功课后都会抽空挖这口井,即使有时很忙,能挖多少就算多少。如今终于让我挖出井水了,我就不用再下山挑水了,我可以有更多时间练我喜欢的太极拳。”

我们在公司领的薪水再多,那都是挑水。而把握下班后的时间挖一口属于自己的井,对我们来说更重要。岁月不饶人,当我们年龄大了,体力拼不过年轻人了,还是能喝到自己挖出来的井水,而且喝得很悠闲。

所以,一定要有自己的特长或者技术核心,挖一口属于自己的井,培养自己独特而坚实的實力。所谓“白天求生存,晚上求发展”,“昨天的努力就是今天的收获,今天的努力就是未来的希望”。多年前不分伯仲的同窗好友,如今的境遇可能大不相同。挖井的过程则告诉我们,知识需要日积月累,最终才能达到一个很高的高度。

西蒙教授曾提出了这样一个见解:“对于一个有一定基础的人来说,他只要真正肯下功夫,在6个月内就可以掌握任何一门学问。”

Java只需要花3个月就可以学会,到6个月后,就可以掌握它。

人生苦短,朝如青丝暮成雪。不能人没了,知识却还没用上。所以这里选择更先进的语言,有垃圾回收后,就不用为释放内存费心思,节省时间。

有很多人从开发Java后受益。例如美国的Cutting,以前是一个C++程序员,他在1999年写下第一个Java项目,即名字叫作Lucene的全文检索项目。在后来的10多年里, Lucene越来越流行,成为开源组织Apache基金会的项目,并在维基百科网站等项目中得到了广泛使用。Cutting后来开发MapReduce的Java版本Hadoop也同样成功。Cutting因此进入了Apache基金董事会,并在2010年成为董事会主席。

古代才女的标准是精通琴、棋、书、画,但现在,精通这些往往只能作为业余爱好。会Java的美女对IT男有很大的吸引力,而且能独立谋生。

Andy Rubin在2002年把Danger公司以5亿美元卖给微软后, Sidekick手机后来退出市场。而使用Java标准版本开发应用的Android却取得了巨大的成功。Android底层使用Linux操作系统内核,但是,手机应用运行在一个叫作Dalvik的Java虚拟机上。

Java是一门投入小、见效快、收益大的编程语言。学习基础Java只是5天的课程。众多的Java开源工具包能让你避免重复发明轮子。“开源软件哪里找? code.google.com 和

apache.org 以及 sourceforge.net。”其中, apache.org 像是一个品牌店, 包含一些知名的项目, 例如, 实现搜索功能的 Lucene 或者实现下载网页的 HTTPClient。而 code.google.com 和 sourceforge.net 则包含一些开发人员自己建立的小项目。就好像一个集市, 容纳了各种琳琅满目的小玩意。

相对于很多数据库编程语言的昙花一现、专业 Web 开发语言的不冷不热, Java 的持续流行与 Oracle 和 IBM 等公司的支持分不开。Java 也是云计算的首选开发语言。

如果觉得 Java 程序运行速度慢, 还可以把 Java 代码当成伪代码。

1.1 背 景

开始一段长途旅行前, 首先做好身体上的准备, 然后看一下路线图。

1.1.1 好身体是一切成功的保证

一个人有多种可能的死法, 其中一种是饿死, 还有一种是累死。软件开发人员更有可能累死, 而不是饿死。就好像买房子至少要准备足够的首付款, 开发软件也要做好身体上的准备。

软件开发是个脑力活, 需要保证大脑思考所需要的氧气和营养物质。所以可以在工作间摆放陶粒防蚊的水培绿色植物, 可以常吃健脑食品, 例如花生、菠菜、鸡蛋等。注意力集中时间太久了会感觉累。代码或者同类文字写多了, 需要转移注意力。要锻炼好身体, 例如常跑步和跳跃, 有条件的, 可以游泳或打乒乓球、羽毛球等。经常进行杠铃这样的力量练习, 有助于提升克服困难的能力。如果在企业从事软件开发工作, 更要注意及时休息和调节自己。就好像吃饭不要偏食, 一个人要有多个社交圈子。

脑子越用越活, 但是, 用脑过多会头疼, 这时候需要睡个好觉。除了注意休息, 喝茶也能缓解头疼。除了专业开发工作, 还需要业余兴趣来平衡大脑。简历模板上的“业余爱好”这项并不是多余的。

显示屏幕发出的蓝光会影响睡眠质量, 为了能够不影响睡眠, 可以佩戴防蓝光眼镜。下午 5 点或者 6 点以后关机, 保证睡眠充足, 不再用电脑。每天上午在绿树成荫的地方进行力量练习或者负重行走, 这有助于改善晚上的睡眠。晚餐可以经常吃小米、藕、虾皮、鸡蛋等, 喝决明子、玉兰花、薰衣草、绞股蓝等花草茶。

虽然抽烟也能提神, 但为此而损害健康就不值得了。不要多喝酒。李白斗酒诗百篇, 但没听说过“酒后能多写程序”。冷酒伤胃, 热酒伤肝, 所以最好别喝。

硬盘用一段时间后可以重新整理碎片, 所以人也是白天工作后, 晚上需要睡觉, 让大脑整理工作和学习的经历。不同的时间, 睡眠效果不一样。晚上 11 点和 12 点以及 1 点这三个小时都一直在睡眠状态时, 能达到最佳睡眠。这段时间叫作核心睡眠区间。睡眠时间长度因人而异, 往往在 7 到 9 个小时之间。喝绞股蓝茶有助于获得良好的睡眠。

长时间对着屏幕, 容易“眼涨”。多看绿色植物或者闭目养神就能缓解。可以使用眼霜来消除加班后的黑眼圈。如果总是盯着显示器, 就容易导致眼疲劳。可以做新式眼保健操, 也就是眼球运动。



据说长期面对电脑受到的电磁辐射容易让人掉头发，所以，可以把机箱加上防辐射罩，或者准备一个防辐射帽。连续坐在电脑面前的时间最好控制在两小时以内。

要想办法保持健康状态。虽然据说感冒能预防癌症，但与疾病做斗争的时候，就没法专心思考问题了。即使吃不到特供食品，最好也要避开地沟油或毒韭菜。要学会自己做饭，这是保证“食品安全”的一个办法。

很多软件开发是在大城市中完成的，城市的空气污染容易导致咳嗽。轻唱可以预防和治疗咳嗽。游泳也可以治疗咳嗽。

把食物做熟的过程是能量的积分。就是说，即使鸡蛋的加热温度只有 60 多度，但是，这样的温度如果能保持几个小时，鸡蛋也能熟。同理，人活得更长就能做更多的事情(而每日一次的饭后半粒二甲双胍据说有可能更长寿)。

1.1.2 路线图

我们的目的是开发出专业的中文文本信息处理程序。首先从结构化程序设计入手，然后开始面向对象程序设计。将介绍编程所需要的数据结构与算法，以及处理文本的方法等。

在电影《源代码》中，主人公一开始并不明白为什么镜子中的形象并不是他自己，但这并不妨碍他在布置好的场景中做一些简单的事情。很多时候，不可能一次性地把见到的东西完全看明白，有时候，会再次回到以前的代码，每次多明白一点点。到最后，对常用的东西，基本上就都明白了。

小李喜欢通过用他的笔记本电脑看教学视频来学习 Java。但是，笔记本的屏幕太小了，视频中的代码看不清楚。他外接了一个显示器，用这个外接的显示器看教学视频，他自己笔记本上的显示器则用来练习。

1.1.3 Java

首先看一下开发 Java 所使用的台式机。打开台式机的机箱，最引人注目的是主板，主板上 CPU 和内存条。硬盘架上有硬盘。内存就像大脑的短期记忆，而硬盘就像大脑的长期记忆。机器没有启动的时候，程序位于硬盘中。当程序开始运行的时候，程序位于内存中。而 CPU 则是进行计算的逻辑处理单元。

因为可以根据地址直接读写任意位置的数据，所以内存又叫作随机访问存储器。早期的程序往往要考虑如何在很有限的内存中运行。而今天，对内存的使用则更多地关注缓存。

CPU 执行机器语言 也就是一套指令集。例如，我们日常使用的 CPU 一般都是运行 x86 指令集。计算机上的可执行程序体现为机器代码。

在超市买牙签时，至少要买一盒，不会按根卖的。而 CPU 往往按字节来取数据。一个字节是 8 位。为什么一个字节是 8 位而不是 64 位？设计成这样并不是为了图吉利，而是因为早期计算机的数据总线宽度是 8 位。

汇编语言用助记符来表示每一条机器语言。汇编语言源代码要翻译成机器代码才能执行。但汇编语言编写的程序只能在特定的机器上运行。

另外，程序一般运行在操作系统中，例如 Windows 操作系统。操作系统中运行的一个

应用程序叫作一个进程。可以同时启动多个进程，其中，每个进程都有自己独立的内存空间。在 Windows 操作系统中，可以用任务管理器看到所有在运行的进程。

高级语言则是一种可以在各种计算机上通用的程序设计语言。例如 C++ 和 Java，因为主要是在美国开发的，所以都很像英语。但是，Java 比 C++ 更加贴近自然语言。美国大学的教授比中国大学的教授更紧跟潮流，所以往往教他们更容易学习的 Java 语言，而不是 C++。

由专门的程序把源代码翻译成机器代码。翻译成机器语言的过程叫作编译，所以，这个翻译程序叫作编译器。C++ 在源代码级别兼容各种平台。

但是，C++ 编译出的可执行代码无法跨平台。因为即使是运行在 x86 指令集上的 Linux，它与 Windows 上的可执行代码尽管都是 x86 机器码，但操作系统不一样，可执行代码的框架就不一样。具体来说，Windows 下可执行文件是 PE 格式，而 Linux 则是 ELF 格式。

为了实现跨平台，需要定义一套统一的程序格式。为了兼容各种 CPU，还需要定义一套统一的虚拟指令集，让程序运行在统一的内存框架下。这就是虚拟机技术。Java 程序运行在专门为它定义的虚拟机上。这个虚拟机叫作 Java Virtual Machine(JVM)。

很多其他语言也采用了虚拟机技术来实现，例如 C# 采用的虚拟机是公共语言运行库，Perl 采用的虚拟机是 Parrot，PowerBuilder 采用的虚拟机是 PBVM。在 FlashPalyer 中，也有个 AVM2 虚拟机。

Java 编译器把源代码编译成字节码。在生活中，有些词需要避讳。编译器也有留做自己专用的保留字，或者叫作关键字(Keyword)。例如，public 就是一个关键字。

程序源代码是文本文件，早期的文本文件采用 ASCII 编码。英文有 26 个字母，算上大小写、数字、特殊字符，加上回车、空格等，共有 128 个字符。用 7 个 1 或 0 的组合，就可以把这些字符全部显示出来。最后加上一位识别码。所以在电脑上，ASCII 编码的一个字符占用一个字节，也就是 8 位。ASCII 编码对每个英文字符都进行编号，例如规定 a 字符编号为 97，b 字符编号为 98 等。大写字母和小写字母有不同的编号。

术语：ASCII(American Standard Code for Information Interchange) 美国资讯交换标准码的简称。ASCII 主要用于显示现代英语和其他西欧语言。使用指定的 7 位或 8 位二进制数组合来表示 128 或 256 种可能的字符。其中 48~57 为 0 到 9 十个阿拉伯数字，65~90 为 26 个大写英文字母，97~122 为 26 个小写英文字母。其余为一些标点符号、运算符号、控制字符等。因为早期程序主要面向讲英语的用户，主要由讲英语的程序员开发，所以 ASCII 很流行。

如果要支持中文，ASCII 编码要扩展成专门的中文编码 GBK。英文字母最高位编码是 0，而汉字最高位编码是 1。Java 中的源代码还可以采用通用字符集 Unicode 编码。

术语：Unicode Translation Format(UTF) 即 Unicode 转换格式。Unicode 表示一种唯一的、统一的和通用的编码。

JVM 本身并没有定义 Java 程序设计语言。它只定义了包含 JVM 指令集的 class 文件的格式，以及一个符号表等。

根据 CPU 的寻址空间，可以分为 32 位和 64 位 CPU。JVM 为这两种 CPU 做了专门的版本，字长是 32 位和 64 位的 JVM。为了实现这样的跨平台运行想法，每种常用的 CPU 和



操作系统组合都有相应的 JVM 实现，如图 1-1 所示。

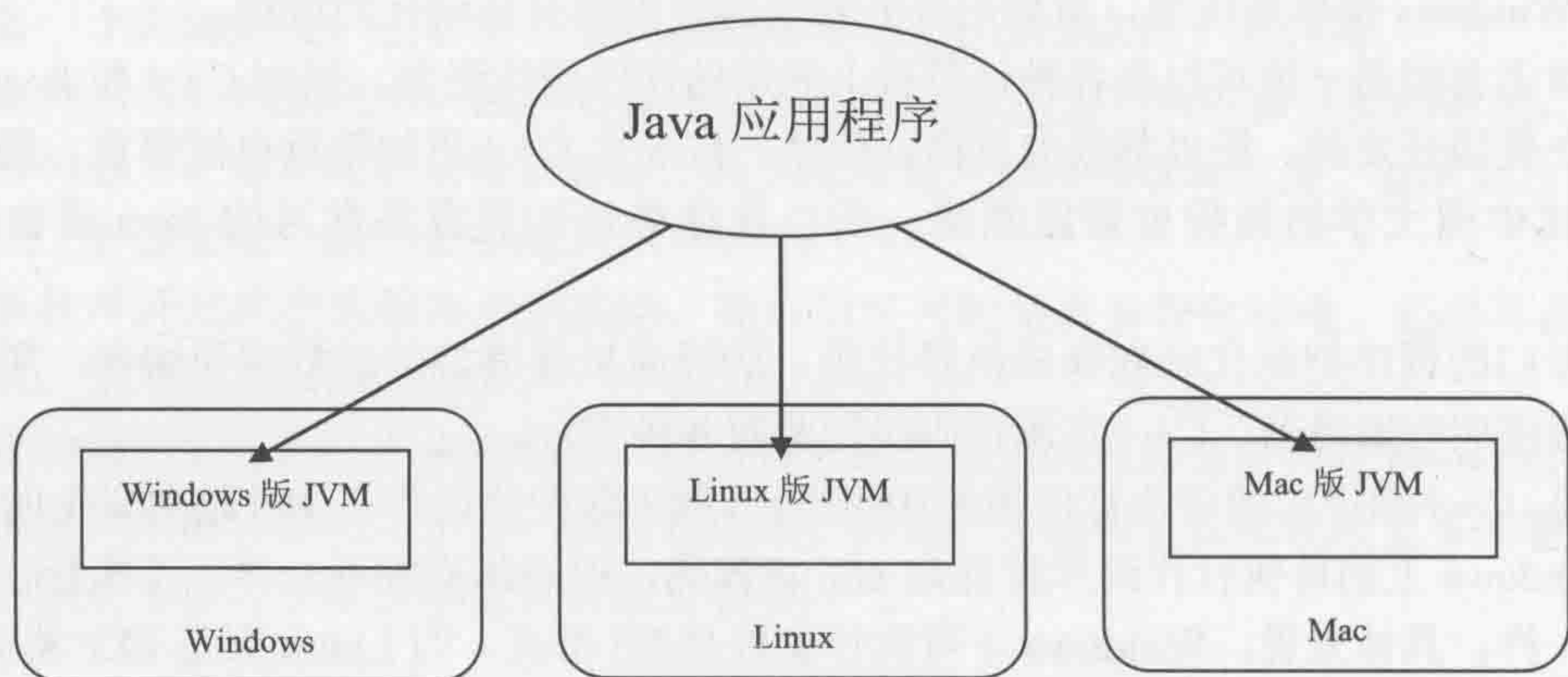


图 1-1 运行于 JVM 的 Java 应用程序

可以在 Windows 平台编译一个 Java 程序，并且在 Linux 平台运行它，因为 Windows 和 Linux 都支持 JVM 实现。Linux 平台下的指令集可能与 Java 编译器生成的字节码不一样。一个 JVM 可以一次性地解释字节码，或者把字节码编译成它所运行的本地代码，这种优化技术叫作 JIT(Just-In-Time)。Java 字节码程序可以运行在任何有字节码解释器的程序上，如图 1-2 所示。

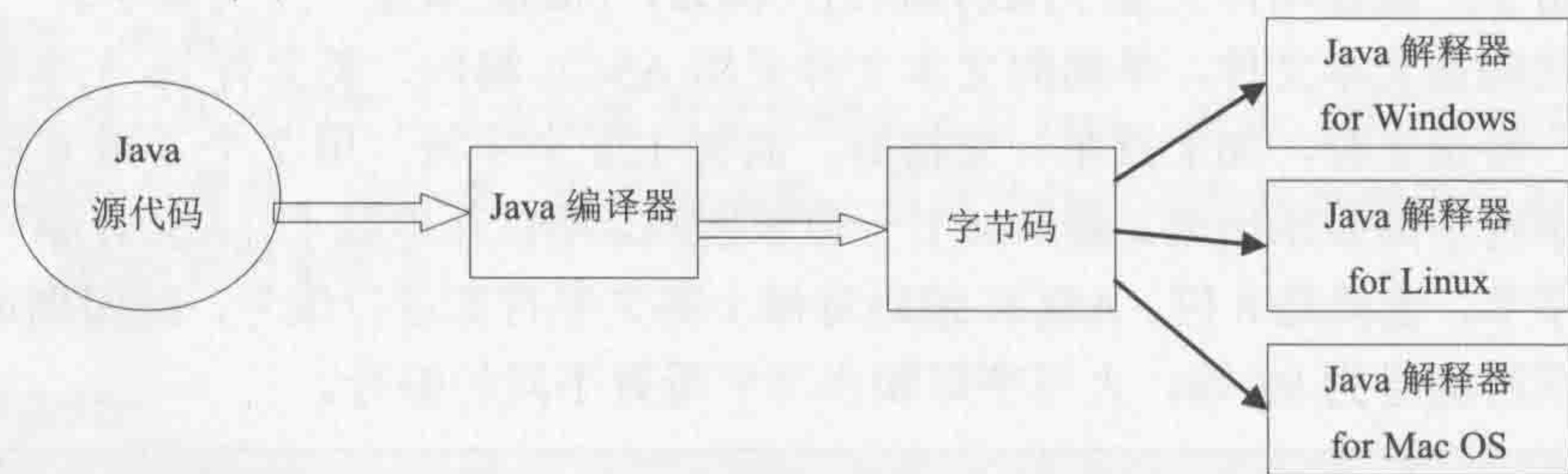


图 1-2 一次编译，到处运行的 Java 应用程序

JVM 定义了一套通用的虚拟指令集，叫作 JVM 指令集。JVM 是基于栈结构的计算机。JVM 指令集中的相关操作数在栈顶运算。例如，如果解释器要执行整数加法，就是从栈顶弹出两个整数进行加法运算，最后将结果压入栈顶。

术语：operand——操作数。

Java 源代码可以被编译成字节码的一种中间状态，然后由已提供的虚拟机来执行这些字节码。Java 程序需要一个运行环境来执行代码。Java 运行环境(JRE)是一个 JVM 的实现。图 1-3 是 Java 中的概念图。

使用 javap 查看 Java 编译器生成的字节码，通过比较字节码和源代码，可以了解编译器内部的工作机制。

Java 需要支持移动计算，所以 Java 的 class 文件设计得很紧凑，大部分指令只占一个字节。class 是动态扩展和执行的，不必等到所有 class 下载完毕才开始执行。

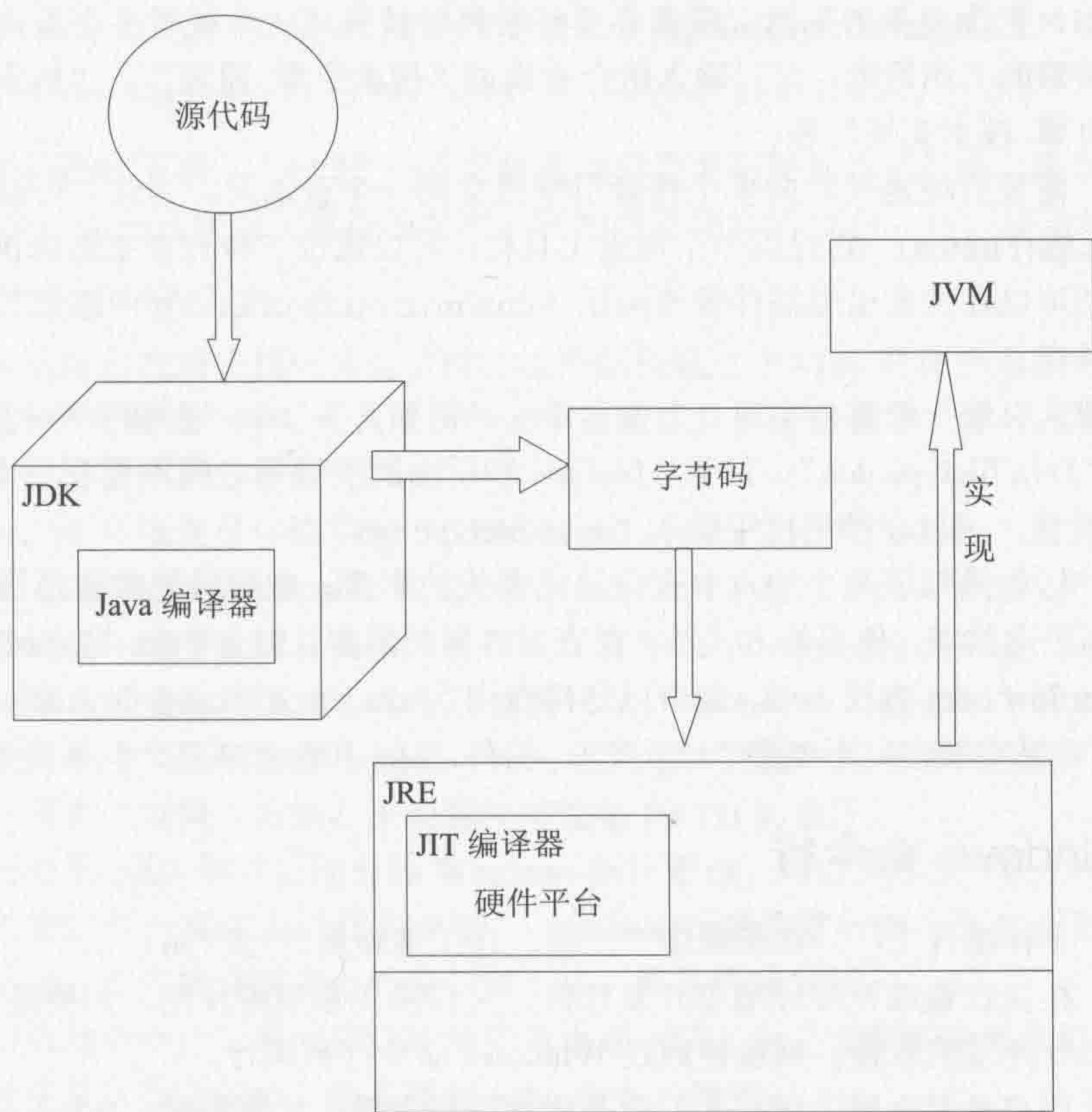


图 1-3 Java 中的概念图

因为计算机中往往有很多程序同时在运行，例如，可以同时启动多个 JRE，所以由操作系统来调度程序的执行。

1.2 软件工具

据说，尼安德特人灭亡的部分原因，是他们不会使用针来缝衣服。我们为了提高学习和工作效率，需要学会使用一些有用的软件工具。

1.2.1 搜索引擎

有问题的话，首先可以到 Google 上搜索，搜索引擎已经成为人类大脑的外部存储设备。习惯于使用搜索引擎的人们，会忘记自己能在网上找到的信息，而记住自己认为无法在网上找到的信息。

如果说开发软件就像种菜，不是每个人都会，则使用搜索引擎就像做菜，很简单，每个人都能轻松学会！最常规的做法是打开浏览器，输入网址。然后在搜索框中输入你要找的东西的描述，或者你要问的问题，然后按回车键或者单击提交按钮，即可等待搜索引擎返回结果了。如果第一页结果找不到想要的，可以翻到下一页，或者换个查询词再找。



关键是如何把你要找的东西，或者你要问的问题转换成一个或者多个查询词。例如，想知道搜索引擎的工作原理，可以输入组合查询词“搜索引擎 原理”、“搜索引擎 开发”或者“搜索引擎 技术实战”等。

有时候，需要方便地同时向多个搜索引擎提交同一个查询词。为此，可以使用英文版本的火狐浏览器(Firefox)，它提供一个搜索工具栏，可以通过下拉列表来选择执行搜索的引擎。另外，还可以在搜索引擎插件管理网址 <http://mycroft.mozdev.org/>中添加其他的搜索引擎到搜索工具栏。

专业的搜索引擎一般都会实现一个搜索语法，例如，找 Java 相关的 Word 文档，就可以这样查询“Java filetype:doc”。注意，filetype:和后面的关键词之间不要有空格。要找 Java 相关的 PPT 文件，可以在搜索框中输入“Java filetype:ppt”。

有时候，您如果知道某个站点中有自己需要找的东西，就可以把搜索范围限定在这个站点中，提高查询效率。使用的方式是，在查询内容的后面，加上“site:站点域名”。例如，要从 stackoverflow.com 查找 Java，就可以这样查询“Java site:stackoverflow.com”。应注意，“site:”后面跟站点域名，不要带“http://”，另外，site:和站点域名之间不要带空格。

1.2.2 Windows 命令行

假设有一个标准件工厂。在车间生产产品，在工地使用这些产品。

类似地，往往在集成开发环境中开发软件，又在操作系统中使用。如果是在 Windows 操作系统中运行开发的软件，则往往通过 Windows 命令行来运行。

在图形化用户界面出现之前，人们就是用命令行来操作计算机的。Windows 命令行是通过 Windows 系统目录下的 cmd.exe 执行的。执行这个程序最直接的方式是找到这个程序，然后双击。但 cmd.exe 并没有一个桌面快捷方式，所以这样还是太麻烦。

可以在“开始”菜单的“运行”窗口中直接输入程序名，按 Enter 键后运行这个程序。选择“开始”→“运行”，这样就会打开资源管理器中的运行程序窗口。或者使用快捷键——窗口键+R，来打开运行程序窗口。总之，输入程序名 cmd 后，单击“确定”按钮，就会出现一个黑色界面的命令提示窗口。因为能够通过这个黑洞洞的窗口直接输入命令来控制计算机，所以也称为“控制台窗口”(Console)。

术语：Console(控制台) 遥控器上有控制面板。更复杂的设备往往有控制台，例如一台机床或者数控设备的控制箱(通常会被称为控制台)。顾名思义，控制台就是一个直接控制设备的“台面”，往往是一个上面有很多控制按钮的面板。在计算机中，可以把直接连接在电脑上的键盘和显示器叫作控制台。

通常用扩展名来表示文件的类别。例如，exe 表示可执行文件。Windows 系统中的文件名称由文件名和扩展名组成。文件名和扩展名之间由小数点分隔，例如 java.exe。

当我们建立或修改一个文件时，必须向 Windows 指明这个文件的位置。文件的位置由三部分组成：驱动器、文件所在路径和文件名。路径是由一系列路径名组成的，这些路径名之间用“\”分开。例如 C:\Program Files\Java\jdk1.7.0_03\bin\java.exe。

公园的地图上往往会标出游客的当前位置。Windows 命令行也有个当前目录的概念。例如使用 C:\Users\Administrator 作为当前路径。

可以用 `cd` 命令改变当前路径，例如改变到 `C:\Program Files\Java\jdk1.7.0_03` 路径：

```
C:\Users\Administrator>cd C:\Program Files\Java\jdk1.7.0_03
```

如果在当前路径下输入“`cd d`”，效果就是改变当前路径到 `d` 子目录。所以切换盘符不能使用 `cd` 命令，而是直接输入盘符的名称。例如想要切换到 `D` 盘，可以使用如下命令：

```
C:\Users\Administrator>D:
```

系统约定从指定的路径找可执行文件。这个路径通过 `PATH` 环境变量指定。环境变量是一个“变量名=变量值”的对应关系，每一个变量都有一个或者多个值与之对应。如果是多个值，则这些值之间用分号分开。

例如，`PATH` 环境变量可能对应“`C:\Windows\system32;C:\Windows`”这样的值，表示 `Windows` 会从 `C:\Windows\system32` 和 `C:\Windows` 这两个路径寻找可执行文件。

设置或者修改环境变量的具体操作步骤是：首先在 `Windows` 桌面右击“我的电脑”，从弹出的快捷菜单中选择“属性”命令，在弹出的“系统属性”对话框中单击“高级”选项卡标签，然后单击“环境变量”按钮，即可设置用户变量，或者系统变量。这时就可以通过“新建”或者“编辑”按钮，来设置环境变量 `PATH` 的值了。

如果使用的是 2008 年以后推出的 `Windows` 操作系统，可能找不到“我的电脑”这样的快捷图标。其实，打开桌面上“我的电脑”，就是运行资源管理器。而打开资源管理器的另外一种方法是：首先按住键盘上的窗口键不放，然后再按 `E` 键。

需要重新启动命令行工具才能让环境变量设置生效。为了检查环境变量是否设置正确，可以在命令行中显示指定的值。需要用到 `echo` 命令。例如：

```
C:\Users\Administrator>echo Hello
```

执行上面的命令，将在命令行输出 `Hello`。

如果要引用环境变量的值，可以用前后两个百分号把变量名包围起来：“%变量名%”。`echo` 命令可以用来显示一个环境变量中的值：

```
C:\Users\Administrator>echo %PATH%
```

1.2.3 机器翻译

很多软件方面的技术文档最开始是使用英文写的。虽然很多经典文档已经翻译成了中文，但是，如果译者水平不够，就会导致信息丢失。很多开源软件也是只有英文说明。能够阅读英文技术文档对学习软件开发会很有帮助。但学习外语又是一件很费时的事情。

机器翻译技术的发展，部分地解决了阅读英文的问题。英语基础不太好的读者可以在开始时使用在线机器翻译，来阅读英文技术文档，然后逐渐过渡到阅读英文原文。可以使用 `Google` 机器翻译(<http://translate.google.cn>)来查看英文网页。例如用 `Google` 机器翻译来查看最新的 `Java` 开发文档。

假如只要能把一个操作过程用自然语言描述出来，就能写出对应的程序，那该多好啊。`Java` 程序中的处理逻辑都是由英文字母和一些简单的符号来描述的。可以借助机器翻译中的概念来学习程序设计。写代码可以看成是把自然语言翻译成机器语言的过程。机器翻译中有个对齐的概念，类似于双向映射。例如，把“如果”和“`if`”对齐。



对齐是一个很多学科都使用的概念。例如 DNA 测序也会用到对齐。可以把不同人种的基因对齐，找出同样功能的基因。

1.2.4 Linux

虽然用 Linux 操作系统办公的不多见，但有很多 Java 程序运行在 Linux 操作系统中。

关于 Linux，有这样的比喻：从前，有一个叫 Linus 的人发明了自助车，把它叫作 Linux。后来有个叫 Sun 的家伙觉得这东西不错，但是开起来有点颠簸，一般人控制不了，于是在 Linux 牌自助车上装上了一个 Java 牌减震器，于是就变成了企业商用车，用了都说好。

再后来，又来了个叫 Andy 的哥们儿，觉得自助车上装个柴油机还是开不快。他干了两件事：一是把柴油机换成了 Dalvik 型汽油发动机，二是把自助车轮子加粗了，又装上了漂亮的车壳。他宣布，Android 牌摩托诞生了。Andy 给 Google 公司的老板 Page 发了一封邮件推销，Page 觉得这摩托不错，就花 5000 万美元买下来了。Google 后来在这辆摩托车上加了很多广告，每年靠移动广告赚 10 亿美元。Page 觉得当初买得太划算了。

没想到这摩托一下子就特别火，很快地，大街上除了开农用车的果农，就都是开摩托的码农了。Linus 跟 Sun 可不干了，说你这不是剽窃我的么。不过 Google 理直气壮：Sun 你看，我发动机都换了，不是你家的了。而 Linus 呢，我把你底下的轮子都改了，不是原来的自助车轮了。

因为 Windows 图形界面简单，所以大部分时候都是在使用 Windows，尝试使用新的操作系统往往很费劲儿。不能像邯郸学步，没学会 Linux，却牺牲了 Windows 的方便易用。

如果有现成的 Linux 服务器可用，可以使用支持 SSH 协议的终端仿真程序 SecureCRT 连接到远程 Linux 服务器。因为可以保存登录秘密，所以比较方便。除了 SecureCRT，还可以使用开源软件 PuTTY(<http://www.chiark.greenend.org.uk/~sgtatham/putty>)，以及可以保存登录密码的 PuTTY Connection Manager。

小袋鼠在袋鼠妈妈的袋子里长大。使用 VMware，Linux 可以运行在 Windows 系统下。VMware 让 Linux 运行在虚拟机中，而且不会破坏原来的 Windows 操作系统。

首先要准备好 VMware，当然，仍然需要 Linux 光盘文件。

就好像华山派有剑宗和气宗，Linux 也有很多种版本，例如 RedHat 或者 Ubuntu 以及 SUSE。这里选择 CentOS(<http://www.centos.org/>)。

如果需要安装软件，可以下载 RPM 安装包，然后使用 RPM 安装。但操作系统对应的 RPM 安装包找起来往往比较麻烦。一个软件包可能依赖其他的软件包。为了安装一个软件，可能需要下载其他好几个它所依赖的软件包。

为了简化安装操作，可以使用黄狗升级管理器(Yellowdog Updater, Modified)，一般简称 yum。yum 会自动计算出程序之间的相互关联性，并且计算出完成软件包的安装需要哪些步骤。这样，在安装软件时，就不会再被那些关联性问题所困扰了。

yum 软件包管理器自动从网络下载并安装软件。yum 有点类似于 360 软件管家，但是，不会有商业倾向的推销软件。例如安装支持 wget 和 rzsz 命令的软件：

```
#yum install wget
#yum install lrzsz
```

可以同时安装几个软件包，例如同时安装 MySQL 服务器和客户端：