

分形程序设计

林晓宇 赖贤伟 宁正元 编著

东南大学出版社

内 容 简 介

本书深入浅出地介绍了 ~~Python~~ 语言的语法及其面向对象的特点,包括 ~~Python~~ 语言的新特性。为了便于初学者的学习,本书力图从初学者的角度出发逐步深入理解和分析问题。书中的实例代码力求精要,既能说明问题又不冗长。

全书共分为 ~~10~~ 章,每章都附有 ~~10~~ 道习题,便于读者自测。第 ~~1~~ 章到第 ~~4~~ 章介绍了 ~~Python~~ 的基础核心内容,包括 ~~Python~~ 语言的相关概念、面向对象的基本概念、简单数据类型与复合数据类型、流程控制语句、方法的使用、面向对象的高级特性等。第 ~~5~~ 章至第 ~~7~~ 章介绍了 ~~Python~~ 语言的高级特性,包括字符串与正则表达式、异常处理与断言、输入输出处理、泛型与集合类、多线程等。第 ~~8~~ 章到第 ~~10~~ 章介绍了 ~~Python~~ 的重要应用技术,包括基于 ~~Socket~~ 的 ~~网络编程~~、网络编程、~~网络编程~~ 与 ~~网络编程~~ 等。附录一介绍了各种开发工具的使用,附录二介绍如何查阅帮助文档,附录三提供了实验指导书。

本书适合作为大学本专科计算机及相关专业的 ~~Python~~ 程序设计或面向对象程序设计等课程的教材,也适合作为读者自学 ~~Python~~ 语言的入门参考书,同时亦可供计算机技术人员参阅。

摇摇摇图书在版编目(CIP)数据

摇摇摇程序设计 林晓宇,赖贤伟,宁正元编著. —南京:
东南大学出版社, 2010.11
ISBN 978-7-305-08910-0

摇摇摇 I 援 林晓宇 II 援 ①林 ②赖 ③宁
III 援 中文编程语言—程序设计—高等学校—教材
IV 援 725.4

摇摇中国版本图书馆 CIP 数据核字(2010)第 111111 号

摇摇摇程序设计

摇摇责任编辑摇摇王全祥
摇摇封面设计摇摇王摇玥
摇摇出版发行摇摇东南大学出版社
摇摇出版人摇摇江摇汉
摇摇社摇摇址摇摇南京市四牌楼 2 号
摇摇邮摇摇编摇摇210018
摇摇电摇摇话摇摇(025) 83792333
摇摇经摇摇销摇摇江苏省新华书店

摇摇印摇摇刷

摇摇开摇摇本摇摇16 开
摇摇印摇摇张摇摇16
摇摇字摇摇数摇摇300 千字
摇摇版摇摇次摇摇2010 年 11 月第 1 版
摇摇印摇摇次摇摇2010 年 11 月第 1 次印刷
摇摇书摇摇号摇摇ISBN 978-7-305-08910-0
摇摇印摇摇数摇摇1 册
摇摇定摇摇价摇摇25.00 元

摇摇(凡因印装质量问题,请与本社读者服务部联系。电话: (025) 83792333)

前摇摇言

在多年教授 编程语言设计与面向对象程序设计课程的过程中,我们发现教与学主要存在两大矛盾。第一,教师与学生对课程认知程度不同的矛盾。如果把学习课程比作登山的过程,由于教师已经掌握课程的全部内容,所以教师在授课过程中通常的做法是站在山顶指挥学生。这样即使处于高屋建瓴的角度,也难免把每个新知识点想成是理所当然的。而学生在学习过程中,每一个新知识对于他们来说都是一个新高度。由于还未认识到课程的全部,所以难免以管窥豹,无法将知识点联系起来。例如,教师介绍到数组的时候,理所当然地认为一个所谓对象数组就是一个引用的数组,因此需要再次初始化,而学生则不这样认为,他们觉得既然是对象数组,那数组中自然应该存储的是“对象”。因此,最好的教学方法是让教师“下山”,带领学生从山脚下开始一步步往上爬。第二,教材结构与学习认知过程之间的矛盾。大多数教材侧重于知识点的归纳,总是把一类的知识点归纳成章,但这些知识点对于学生来说可能是跳跃性的,也就是说按照传统教材上的章节顺序学习,学生常常不得不跳过一些知识盲区,跳不过的学生就会停滞不前甚至出现厌学情绪。例如,很多教材一开始就介绍数据类型,但是由于还未介绍到类的相关概念,学生自然就无法理解类和接口为什么也是数据类型,为什么它们是组合数据类型,这就给后续的学习带来了困难。

基于解决以上两大矛盾的考虑,我们在编写本书的过程中,对该课程的知识体系进行了大力度的改革。首先,对传统的章节结构按知识点拆分,然后重组。章节的划分不再追求同类知识点的归类,而更侧重于初学者的认知过程。知识点重组以后,同类知识可能分属于不同的章节,但是对于初学者来说这是平稳的过渡,没有出现上述的知识盲区和学习跳跃。例如,对于“类”这个概念,既有功能上的作用又有数据类型的特性,我们将其分开阐述。先是在第 1 章中以生活中的例子引入了类的概念,而学生在学习了第 2 章介绍的数据类型的概念之后,再学习第 3 章的“类”这个组合数据类型,就很容易理解“类”也是一种数据类型。其次,在编写过程中,我们力图让自己处于初学者的状态,初学者在学习时遇到的每个新知识点将要面对的新问题可能是什么,我们都作了仔细的思考与分析。例如,在第 4 章介绍过 变量的相关知识之后,第 5 章我们就引入了类与对象的概念。我们从学生熟悉的事物出发来描述生活中的“类”,引导学生如何归纳和描述具有相同属性的一“类”事物。再把这一“类”事物转化为 变量的“类”,这个过程非常自然,学生不知不觉就会用符合 语法的类来描述问题领域。第三,设计精要的实例。很多参考书不厌其烦地列举大段大段的代码,试图让初学者接收更多的信息。事实上,一看到冗长的代码,初学者已经就不知所措,就是有经验的教师也要花大量的时间钻研。我们认为,作为入门教材,实例代码应该既要说明问题又不能冗长,这样初学者容易接受。读者会发现,本书的例子都相当简洁,只说明当前需要关注的问题,没有多余的代码。

除了传统的 语法,本书全面引入了 的新特性。在本书编写之际, 的

播



摇
前
言



月新版也正在陆续发布,但是员及员对员语言本身没有做变动,而只是增加了一些类库。所以读者可以认为阅读本书就将学习到员语言的最新特性。下面这张表列出了员新特性与新类及其在本书中的位置。

员版本	新员特性	在本书中出现的位置
员	静态导入	员静态成员与静态导入
员	枚举类型	员枚举类型
员	自动封装与解封	员自动封装与解封
员	增强 循环(循环)	员 语句
员	可变长参数	员可变长参数
员	控制台输入类 类	员 类
员	格式化输出 类	员 类
员	泛型	员泛型
员	正则表达式	员正则表达式
员	断言	员断言 员

员全书共分员章。第员章概述员技术,介绍员运行环境;第员章引入类与对象的概念;第员章介绍基本数据类型及其运算;第员章介绍组合数据类型;第员章介绍流程控制语句;第员章介绍了方法(函数)的使用及方法调用时的参数传递过程;第员章介绍了面向对象中的一些高级特性;第员章介绍了字符串及正则表达式的使用;第员章介绍了异常处理机制,同时还介绍了员版本中的断言机制;第员章介绍输入输出的处理;第员章介绍了员的新特性泛型与集合类;第员章介绍多线程编程;第员章和第员章介绍了基于员的图形用户界面的设计;第员章介绍了网络通信编程;第员章讲解了应用员进行编程的员技术并介绍员架构。附录一介绍了各种开发工具的使用;附录二介绍如何查阅帮助文档;附录三提供了实验指导书。

本书由林晓宇、赖贤伟、宁正元共同编写。其中林晓宇编写了第员章至第员章以及附录一至附录三,宁正元编写了第员章,赖贤伟编写了第员章至第员章。林晓宇负责全书统稿。在本书编写和出版过程中,得到了作者所在单位的领导和同事的关心与帮助以及东南大学出版社的支持,在此深表感谢。

由于作者水平有限,加之时间仓促,书中疏漏与错误之处在所难免,希望广大读者不吝赐教,我们将不胜感谢。

编 者
赖贤伟

目摇摇录

第 员章 摇摇概念	员
摇摇的发展历程	员
摇摇的历史	员
摇摇技术现状	员
摇摇虚拟机	圆
摇摇的特点	猿
摇摇面向对象	猿
摇摇健壮并且安全	猿
摇摇可移植	源
摇摇高性能	源
摇摇其他特性	源
摇摇程序开发环境	源
摇摇安装 分运	缘
摇摇配置类路径	愿
摇摇程序的结构	愿
摇摇第一个 摇摇类	怨
摇摇第一个 摇摇类	怨
摇摇集成开发工具	员园
摇摇习题	员员
第 圆章 摇摇类与对象	员猿
摇摇对象和面向对象	员猿
摇摇创建类	员猿
摇摇类的成员	员源
摇摇数据成员	员源
摇摇方法成员	员缘
摇摇面向问题领域的抽象	员远
摇摇对象	员苑
摇摇成员的访问	员愿
摇摇包	员怨
摇摇类语句	员怨

摇



目
录



包编译	101
类声明语句	101
类声明	102
第 8 章 类的封装	103
类限定词	103
类无限定词	103
类声明限定词	104
第 8 章 习题	104
第 9 章 数据类型与运算	105
标识符与保留字	105
词法单位	105
标识符	105
关键字	106
变量	106
赋值运算	106
常量	106
数值类型	106
整数类型	106
浮点类型	106
运算符与算术运算	107
移位运算	107
扩展赋值运算	107
数学函数	107
字符型	107
布尔型	107
关系运算	107
逻辑运算	107
条件运算	107
表达式	107
运算符的优先级和结合性	107
自动类型转化	107
强制类型转化	107
习题	107
第 9 章 组合数据类型	108
类类型	108

源随员摇对象中的数据字段	源
源随员摇静态成员与静态导入	缘
源随员摇构造方法与初始化	缘
摇源圆摇数组类型	缘
源随员摇声明数组	缘
源随员摇初始化数组	缘
源随员摇使用数组	远
摇源猿摇枚举类型	远
源随员摇枚举声明与使用	远
源随员摇枚举类型的优势	远
源随员摇枚举中的方法	远
摇源源摇基本数据类型的对象封装	远
源随员摇包装类的构造函数	远
源随员摇包装类的应用	远
源随员摇自动封装与解封	远
摇习题源	远
第缘章摇流程控制语句	远
摇缘员摇语句	远
摇缘圆摇分支语句	远
缘随员摇条件语句	远
缘随员摇多分支语句	苑
摇缘猿摇循环语句	苑
缘随员摇憎循环语句	苑
缘随员摇凿循环语句	苑
缘随员摇本语句	苑
摇缘源摇跳转语句	愿
缘随员摇透语句	愿
缘随员摇精语句	愿
缘随员摇见语句	愿
摇缘缘摇注释语句	愿
摇习题缘	愿
第远章摇方法与参数传递	愿
摇远员摇方法的定义	愿
远随员摇方法的返回值	愿
远随员摇方法的参数	愿





摇远圆摇方法的调用	愿怨
摇远陵摇参数传递	愿怨
摇远陵摇基本数据类型传递	愿圆
摇远圆摇引用传递	愿猿
摇远源摇变量的作用域	愿源
摇习题 远	愿苑
第 苑章摇面向对象高级特性	员员员
摇苑员摇继承	员员员
摇苑员员摇类选择	员员员
摇苑员圆摇构造类	员员猿
摇苑员猿摇类成员限定词	员员猿
摇苑员源摇类	员员缘
摇苑员缘摇继承与初始化	员员缘
摇苑圆摇多态性	员员怨
摇苑圆员摇重载	员员圆
摇苑圆圆摇构造方法重载	员员圆
摇苑圆猿摇可变长参数	员员猿
摇苑圆源摇重写	员员源
摇苑圆缘摇隐藏	员员怨
摇苑圆陆摇覆盖方法	员员员
摇苑圆柒摇类型转换	员员员
摇苑猿摇抽象类和接口	员员猿
摇苑猿员摇抽象方法与抽象类	员员猿
摇苑猿圆摇接口	员员源
摇苑源摇内部类	员员正
摇苑源员摇类成员内部类	员员苑
摇苑源圆摇本地内部类	员员愿
摇苑源猿摇静态内部类	员员愿
摇苑源肆摇匿名内部类	员员怨
摇习题 苑	员员起
第 愿章摇字符串与正则表达式	员员猿
摇愿员摇类	员员猿
摇愿员员摇生成 类对象	员员猿
摇愿员圆摇字符串和字符	员员源
摇愿员猿摇获取新串	员员缘

源源源 搜索字符串	员苑
源源源 字符串转换	员苑
摇源源 正则表达式类	员怨
源源源 构造方法	员怨
源源源 长度和容量	员怨
源源源 修改字符串	员园
源源源 正则表达式类	员员
摇源源 字符串的比较	员员
摇源源 正则表达式	员源
源源源 模式匹配	员源
源源源 预编译模式类	员缘
摇习题 愿	员远
第 怨章 异常处理和断言	员愿
摇怨源 什么是异常	员愿
摇怨园 异常处理机制	员愿
怨源源 捕获异常	员怨
怨源源 内置异常类	员员
怨源源 本地语句块	员猿
怨源源 声明抛弃异常	员源
怨源源 出现异常对象	员缘
怨源源 自定义异常	员远
摇怨特 断言 粤粤粤粤粤	员远
怨源源 粤粤粤粤粤 的语法与语义	员苑
怨源源 粤粤粤粤粤 的编译	员苑
怨源源 粤粤粤粤粤 的运行	员苑
怨源源 粤粤粤粤粤 与继承	员怨
摇习题 怨	员园
第 员园章 输入输出处理	员源
摇员源源 字节流	员源
员源源 限制流	员源
员源源 缓冲流	员缘
摇员源园 文件处理	员缘
员源源 流类	员缘
员源源 文件操作	员远
员源源 目录操作	员愿





顺序流读写文件	页码
随机读写文件	页码
摇 过滤流	页码
阅读器流和 阅读器流	页码
阅读器流和 阅读器流	页码
生产者流	页码
摇 字符流	页码
阅读器流和 写流	页码
阅读器流和 写流	页码
云流和 云流	页码
阅读器流和 写流	页码
摇 控制台交互程序	页码
方法参数	页码
类	页码
类	页码
类	页码
摇 对象串行化	页码
摇 序列流	页码
摇 管道流	页码
摇 习题	页码
第 章 泛型与集合类	页码
早期的集合类	页码
泛型	页码
泛型类	页码
泛型接口	页码
通配泛型参数	页码
泛型方法	页码
泛型使用时的限制	页码
集合类	页码
通用接口	页码
类	页码
类	页码
类	页码
类	页码
习题	页码

第 4 章 多线程	4-1
多线程的概念	4-1
多线程的状态	4-2
多线程编程	4-2
创建线程	4-3
线程的优先级和调度	4-4
线程同步	4-5
习题 4	4-6
第 5 章 图形用户界面编程	5-1
组件和容器	5-1
容器	5-1
容器面板	5-2
布局管理器	5-2
容器布局管理器	5-3
容器布局管理器	5-4
容器布局管理器	5-5
容器布局管理器	5-6
容器布局管理器	5-7
容器其他布局管理方式	5-8
容器布局管理器的综合使用	5-9
事件处理机制	5-9
事件监听与处理	5-10
事件适配器	5-11
习题 5	5-12
第 6 章 杂项组件	6-1
基本组件	6-1
标签	6-1
文本组件	6-2
按钮	6-3
单选框	6-4
多选框	6-5
下拉框	6-6
列表框	6-7
对话框	6-8





员缘园员缘通用对话框	员缘园
员缘园员缘标准对话框	员缘园
员缘园员缘文件对话框	员缘园
员缘园员缘菜单	员缘园
员缘园员缘粤亮域	员缘园
员缘园员缘粤亮域概念	员缘园
员缘园员缘粤亮域与粤亮域列表的异同	员缘园
员缘园员缘粤亮域的安全机制	员缘园
员缘园员缘粤亮域的生命周期及主要方法	员缘园
员缘园员缘习题	员缘园
第 4 章 裁子栽孕网络编程	员缘园
员缘园员缘裁子栽孕	员缘园
员缘园员缘栽孕域通信	员缘园
员缘园员缘栽孕域	员缘园
员缘园员缘栽孕域通信过程	员缘园
员缘园员缘创建栽孕域	员缘园
员缘园员缘客户端的栽孕域	员缘园
员缘园员缘服务器端的栽孕域	员缘园
员缘园员缘打开输入输出流	员缘园
员缘园员缘关闭栽孕域	员缘园
员缘园员缘栽孕程序	员缘园
员缘园员缘多客户的栽孕程序	员缘园
员缘园员缘数据报通信	员缘园
员缘园员缘网络层多播域和网络层单播域	员缘园
员缘园员缘基于裁子的栽孕程序	员缘园
员缘园员缘用数据报进行广播通信	员缘园
员缘园员缘习题	员缘园
第 4 章 栽孕与允孕	员缘园
员缘园员缘宰藻编程原理	员缘园
员缘园员缘宰藻体系结构	员缘园
员缘园员缘栽与允	员缘园
员缘园员缘宰藻语言	员缘园
员缘园员缘栽对栽服务器	员缘园
员缘园员缘允孕通用语法规则	员缘园
员缘园员缘允孕指令	员缘园

员源猿员源 变量声明指令	猿缘
员源猿员源 变量赋值指令	猿苑
摇员源源 脚本	猿苑
员源源员源 声明	猿愿
员源源员源 表达式	猿愿
员源源员源 小程序	猿怨
摇员源缘 注释	猿员
员源缘员源 内容注释	猿员
员源缘员源 符号注释	猿圆
员源缘员源 脚本语言注释	猿圆
摇员源远 摇动作指令	猿圆
员源远员源 约瑟夫环	猿猿
员源远员源 约瑟夫环	猿猿
员源远员源 约瑟夫环	猿猿
摇员源苑 摇内置对象	猿源
摇员源愿 摇表单初步	猿苑
摇员源怨 摇介绍数据库编程	猿愿
摇员源员园 摇介绍	猿愿
员源员园员源 摇的特点	猿怨
员源员园员源 摇的四层模型	猿圆
员源员园员源 摇应用程序组件	猿圆
员源员园员源 摇的结构	猿圆
员源员园员源 摇应用中的角色分配	猿猿
员源员园员源 摇的核心 粤孕与组件	猿源
员源员园员源 摇基础	猿远
摇习题 员远	猿苑
附录一 摇各种开发工具的使用	猿愿
附录二 摇帮助文档的使用	猿猿
附录三 摇实验指导书	猿缘
摇实验一 摇介绍开发环境	猿缘
摇实验二 摇类与对象	猿缘
摇实验三 摇数据类型与运算	猿远
摇实验四 摇数组	猿远
摇实验五 摇流程控制语句	猿远





摇实验六摇继承.....	猿苑
摇实验七摇多态性.....	猿苑
摇实验八摇多线程.....	猿苑
摇实验九摇多线程编程.....	猿怨
摇实验十摇综合实验.....	猿怨
参考文献.....	猿园

第 1 章 摇钱树概述

摇钱树的发展历程

摇钱树的历史

摇钱树是一种可跨平台的面向对象的程序设计语言,由美国 奈特公司 奈特等人始创。如今,基于 摇钱树的众多技术已经成为 行业界的领先技术。摇钱树的前身称为 静噪被设计作为一种小型家用电器的编程语言,试图用于解决诸如电视机、电话、闹钟、烤面包机等智能家用电器的控制和通讯问题。由于这些智能化家电的市场需求没有预期的高,奈特放弃了该项计划。就在 静噪几近夭折之际,随着 网络的发展,奈特看到了 静噪在计算机网络上的广阔应用前景,于是改造了 静噪在 1995 年 1 月以“ 摇钱树”的名称正式发布了。

最初的 摇钱树是单调的,要实现用户和服务器的交互也是比较困难的,摇钱树的出现使得整个 网络生动起来,人们在浏览器中使用 摇钱树技术和服务器交互,并且可以享受到由 摇钱树语言编写的动画和多媒体效果。支持 摇钱树的浏览器伴随着 网络的迅猛发展而发展,逐渐成为重要的 网络编程语言。

但是好景不长,在流行几年之后,摇钱树在浏览器中的地位一落千丈。原因是它在简单交互动画方面的表现力及简易性远不及现在大名鼎鼎的 网景。微软公司也因为商业竞争的原因拒绝在新版本的浏览器中附带 摇钱树平台。因为开销较大,摇钱树在桌面系统上也一直没能取得突破。柳暗花明又一村,硬件和互联网的快速发展再次给予 摇钱树一次机会,摇钱树在服务端和移动设备方面取得了相当大的成功。互联网的出现使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机是异构的,即计算机的体系结构和操作系统是不一样的。例如 服务器的硬件是 体系,操作系统是 或者是 而 工作站的硬件是 体系,软件是基于 的 操作系统,相应的编程语言基本上只能应用在某种特定的平台上。网络计算模式的另一个特点是代码可以通过网络在各种计算机上进行迁移,这就迫切需要一种跨平台的编程语言,使得用它编写的程序能够在网络中的各类计算机上正常运行。

摇钱树语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点。这些特点恰好符合互联网网络计算的要求,于是 摇钱树和 网络迅速融合并互相推动、快速发展。尽管 奈特公司的设计师们的初衷只是想设计出一种可以在不同小型硬件上通用的控制语言。但正是他们的这种跨平台的思想造就了 摇钱树今天的成功。

摇钱树技术现状

摇钱树发展到今天,已经不仅仅是一种单纯的程序开发语言,而发展成为了一个以 摇钱树

摇



第 1 章
摇钱树概述



语言为核心的应用于各种计算平台的一整套解决方案。

它不仅包括语言、标准、组件、计算模式,还包括设备、网络技术及软件开发思想、体系结构等等一系列相关技术。我们统称之为**Java技术**。

2006年,Sun公司发布**Java**的三个版本:标准版(**Java SE**)、微型版(**Java ME**)和企业版(**Java EE**),目的是让**Java**在不同的市场目标和设备上都有用武之地。

Java SE是**Java Standard Edition**的缩写,主要目的是为台式机和 workstation 提供一个开发和运行的平台。我们在学习**Java**的过程中,主要是采用**Java SE**来进行开发。

Java ME是**Java Micro Edition**的缩写,主要是面向消费类电子产品,为消费类电子产品提供一个**Java**的运行平台,使得**Java**程序能够在手机、机顶盒、PDA等产品上运行。

Java EE是**Java Enterprise Edition**的缩写,主要目的是为企业计算提供一个应用服务器的运行和开发平台。**Java EE**本身是一个开放的标准,任何软件厂商都可以推出自己的符合**Java EE**标准的产品,使用户可以有多种选择。IBM、SAP、Oracle、Microsoft等大公司已经推出了自己的产品,其中尤以 Oracle 公司的**WebLogic**产品和 IBM 公司的**WebSphere**最为著名。**Java EE**将逐步发展成为可以与微软的**.NET**战略相对抗的网络计算平台。

上述三个**Java**平台的关系如图 1-1 所示。

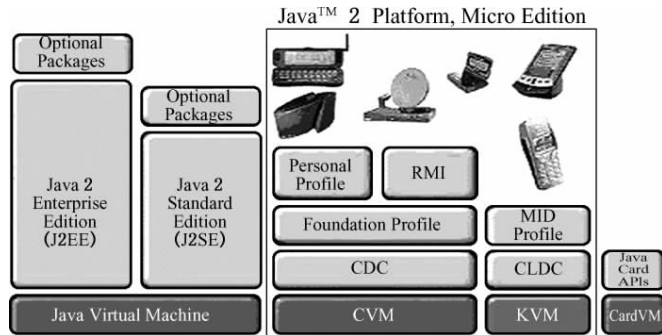


图 1-1 **Java**平台的关系

在**Java**语言的演进中,**Java 1.0**版本是较为成熟和成功的版本,这以后的**Java**统称为**Java 1.x**。而2006年,Sun公司发布**Java 2.0**版本,这个版本又对**Java**语言作了一次较大的增强,引入了泛型、枚举、不定长参数和自动解封装等多种特性,是**Java**语言发展史上一个重要的里程碑,此时**Java 2.0**有了一个别名:**Java SE 5.0**。

2006年,Sun公司又发布**Java 6.0**版本用以取代以前版本中的数字“2.0”。**Java 6.0**更名为**Java SE 6.0**,**Java ME 6.0**更名为**Java ME 6.0**,**Java EE 6.0**更名为**Java EE 6.0**,所以某些特定名词提法不同,含义相同,读者应该要理解。在编写本书之际,**Java**的第 6 个版本的**Java 6.0**版也正在陆续发布,但是第 6 版及第 7 版对**Java**语言本身没有作变动,而只是增加一些**API**供开发人员调用。本书将在各个章节中根据需要介绍**Java 6.0**版本的新特性,所以读者可以认为阅读本书就将学习到**Java**语言的最新特性。

Java 虚拟机

Java语言区别于其他语言的最大特点之一是**Java**是运行在虚拟机之上的。虚拟机并

不是 独有的,在一个操作系统上模拟的一个软件平台都可以称为虚拟机。虚拟机是支撑 程序运行的平台,可以在多种处理器上(无论是在计算机中还是在其他电子设备中)安全并且兼容的执行 应用程序。虚拟机在计算机系统中的层次位置如图 所示。



图 虚拟机在计算机系统中的层次位置

一般的程序设计语言分为编译型和解释型,而 兼具这二种特性。首先, 编译器把 程序源代码编译成 文件,也可以称之为中间字节码文件或者字节码文件。字节码文件并不能直接在操作系统上运行,而必须运行在 虚拟机上。 程序的跨平台主要是指字节码文件可以在任何具有 虚拟机的计算机或者电子设备上运行。接着, 虚拟机中的 解释器负责将字节码文件解释成为特定的机器码进行运行。 程序的编译和执行过程如图 所示。

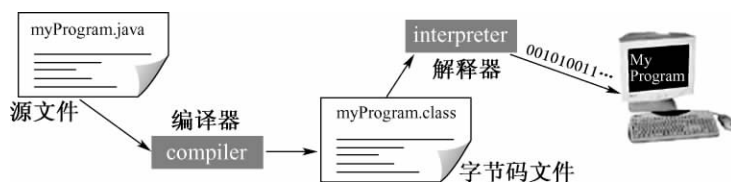


图 程序编译执行过程

虚拟机的特点

虚拟机面向对象

面向对象 是程序设计方法的一种,其核心之一就是开发者在设计软件的时候可以使用自定义的类型和关联操作。代码和数据的实际集合体叫做“对象”。一个对象就是一个绑定了许多“行为(代码)”和“状态(数据)”的实体。面向对象设计让大型软件工程的计划和设计变得更容易管理,能增强工程的健康度,减少失败工程的数量。面向对象设计的另外一个目标就是能产生很多的有关联的类,可以让软件的再开发变得简单。

采用的是相对简单的面向对象技术,去掉了运算符重载、多继承的复杂概念,而采用了单一继承、类强制转换、多线程、引用(非指针)等方式。

虚拟机健壮并且安全

语言在编译及运行程序时,都要进行严格的检查。作为一种强制类型语言,在

播



第 1 章
C
>
A
概述