

面向21世纪高校教材

江苏省普通高校计算机等级考试系列教材

Java 程序设计教程

主编 陈华生 储军杰

江苏省教育厅组织编写



苏州大学出版社

TP 312JA

面向 21 世纪高校教材
江苏省教育厅组织编写

江苏省普通高校计算机等级考试系列教材

Java 程序设计教程

陈华生 储军杰 主编

TP312
78

1 : 8567
3 : 68
16 : 68

苏州大学出版社

内 容 提 要

Java 是一种面向对象的程序设计语言。本书以没有学过程序设计语言,直接学习 Java 语言的读者为对象,分两个层次介绍 Java 语言的基础知识以及面向对象的特性,为读者使用 Java 打下一个良好的基础。

本书通俗易懂,由浅入深,突出重点。为了便于读者的学习和理解,本书提供了大量的例题,大部分章节后备有练习题。本书可作为本、专科院校学生的 Java 语言程序设计课程的教材,也可作为计算机爱好者学习面向对象程序设计的自学教材。

图书在版编目(CIP)数据

Java 程序设计教程/陈华生,储军杰主编. —苏州:
苏州大学出版社,2000.9
面向 21 世纪高校教材·江苏省普通高校计算机等级考
试系列教材
ISBN 7-81037-715-9

I .J… II .①陈… ②储… III .JAVA 语言-程序设
计-高等学校-水平考试-教材 IV .TP312

中国版本图书馆 CIP 数据核字(2000)第 42682 号

Java 程序设计教程
陈华生 储军杰 主编
责任编辑 管兆宁

苏州大学出版社出版发行
(地址:苏州市十梓街 1 号 邮编:215006)
丹阳市兴华印刷厂印装
(地址:丹阳市胡桥镇 邮编:212313)

开本 787×1092 1/16 印张 12.875 字数 321 千
2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷
印数 1-5000 册

ISBN 7-81037-715-9/TP·33(课) 定价:15.00 元

苏州大学版图书若有印装错误,本社负责调换
苏州大学出版社发行科 电话:0512-5236943

江苏省普通高校计算机等级考试
系列教材编委会

顾问	葛锁网	张福炎	邢汉承	
主任委员	邱坤荣			
副主任委员	陈华生			
委员	(以姓氏笔画为序)			
	牛又奇	江正战	江邦人	朱 敏
	陈凤兰	陈良宽	高岳兴	奚抗生
	梅镇武	殷新春	蔡绍稷	
工作人员	王晓天	于荣良		

前 言

Java 语言自 1995 年问世以来,因其面向对象、跨平台性、多线程、稳定、安全、小巧等众多优点,特别是与 Internet 的成功结合而迅速风靡计算机界,并逐步向各个应用领域发展,而互联网及网上应用的蓬勃发展进一步推动了 Java 的发展和使用。

当今世界经济的全球化使得我们与互联网紧密地联系在一起,Java 更是有用武之地。有识之士指出“谁先掌握 Java,谁就号准了信息世界的脉搏,就能在信息时代找到自己的立足之地”。国内外许多计算机公司将掌握 Java 技术作为提升和聘用人员的优先考虑因素,因此许多人都在学习和使用 Java。

考虑到市面上多数 Java 书籍都着眼于 Java 与 C++ 的比较,要求读者有一定的程序设计语言的基础知识,不适合初学者使用,为此我们编写了此书,用作高校学生程序设计语言课程的入门教材。本书以培养学生面向对象程序设计能力为目标,通过对生活中事物的类比和大量的程序实例介绍了 Java 环境下的程序设计概念和知识。本书及与之配套的实验指导书还提供了系统丰富的实验内容和素材,用于训练程序设计的技能。

本书共分十章,第一章介绍了程序及程序设计语言基础知识;第二章介绍了 Java 的发展简史,说明 Java 平台及程序基本结构,描述了 Java 程序的编辑、编译以及执行过程;第三、四、五、六章介绍了 Java 语言的基本程序设计概念和技术;第七、八章介绍了类的定义及 Java 的资源 and 结构;第九章介绍了图形用户界面设计;第十章介绍了多线程、例外处理机制、网络应用、数据库接口 JDBC、多媒体等 Java 的高级概念和编程技术。

全书由南京大学陈华生教授、江南学院储军杰副教授主编,东南大学孙志挥教授主审。陈华生和陈丽娟负责第一、二章的编写,陈平负责第三、四章的编写,张景莉负责第五、六章的编写,朱嘉钢负责第七、八章的编写,储军杰负责第九、十章的编写,全书由储军杰副教授统稿。

孙玫、陈爱国、程建敏、张徽、姚晓峰、刘恺和何江涛参与了本书素材的收集和整理,示例程序的编写与调试,以及部分章节内容初稿的写作。孙玫对全书进行了初排版。

江苏省普通高校计算机基础教学 2000 年 Java 研讨会的与会老师们对本书的讨论稿提出了许多宝贵的意见,在此表示感谢。

由于编写时间仓促,不当之处在所难免,恳请读者批评指正。

储军杰

2000 年 6 月于无锡

目 录

第 1 章 概述

1.1 程序	1
1.2 计算机语言	1
1.3 程序设计与面向对象的程序设计	2
1.4 面向对象程序设计的基本概念	3

第 2 章 Java 入门

2.1 Java 简介	7
2.2 Java 平台	9
2.3 Java 程序的运行*	10
2.4 一个简单的 Java 程序	11
2.5 Java 程序的开发、执行流程	12
练习题	14

第 3 章 基本 Java 类型

3.1 变量	15
3.2 构造数据类型	20
3.3 标准 I/O	22
练习题	23

第 4 章 运算、表达式和语句

4.1 运算和表达式	24
4.2 语句和块	32
练习题	33

第 5 章 流程控制语句

5.1 循环语句	35
5.2 分支语句	40
5.3 break 和 continue 语句	46
练习题	51

第 6 章 数组和字符串

6.1 数组	52
6.2 字符串	59
6.3 数组和字符串的综合实例	63
练习题	67

第 7 章 类和对象

7.1 类的描述	68
----------------	----

7.2 对象的生命周期	75
7.3 类的继承和多态性	80
7.4 包与类成员的访问权限	86
7.5 类的修饰符	91
练习题	93
第 8 章 Java 的资源组成和使用	
8.1 Java 的资源结构	94
8.2 IO 包	95
8.3 其他常用类	113
练习题	126
第 9 章 图形用户界面(GUI)和 Applet	
9.1 GUI 的结构	129
9.2 Java GUI 的组件包:AWT 和 SWING	131
9.3 Applet 和 WWW	171
练习题	177
第 10 章 Java 高级编程入门*	
10.1 异常处理	178
10.2 线程	181
10.3 数据库接口	185
10.4 Java 网络编程	188
10.5 Java 多媒体编程	193
练习题	199

Java 是什么?

Java 是 Sun Microsystems 公司研制开发的一种程序设计语言。

什么是程序? 程序设计? 语言? 不用着急, 慢慢来!

1.1 程 序

通俗地讲, 程序就是为解决某一问题(或实现某个目标)而实施的一系列步骤。就像你想吃饭就得买菜、拣菜、洗菜、淘米烧饭、炒菜, 然后才能吃到一顿香喷喷的饭菜。

再看一个大一点的问题。如果能拥有自己的小屋, 你希望它是什么样? 尖顶的小木屋, 还是平顶的小砖房? 要多大? 把它画下来, 这个蓝图就是我们要实现的目标。动手建造之前要仔细考虑实施步骤, 如先挖地基、再砌墙、架梁、封顶、布线(电线、电话线)、排水管、排煤气管……包括每步采用什么建筑材料、用什么方法等, 这就是造房子的程序。你在脑子里或在纸上规划建造方案的过程, 就是程序设计。

光想还不行, 还得请建筑工程队来实现。工程队按你的预定方案建造, 直到你理想蓝图中的房子矗立在你的面前, 工程队实际建造的过程, 就是程序运行。而最终造出来的小屋, 就是程序的运行结果。

瞧, 就是这么简单。

1.2 计算机语言

语言是思维的载体, 人们通过语言交流思想和情感。生活中人与人之间用于交流的语言称为自然语言, 请工程队造房子, 讲自然语言就行。但如果想让计算机造一间虚拟小屋, 就必须“讲”它能够明白的“话”, 也就是计算机语言, 又称为程序设计语言。

与自然语言相同, 使用计算机语言的目的也是表达我们的思想, 并让“听者”——计算机明白你所要表达的内容。我们要学的 Java 就是一种计算机语言, 让我们看一下完成同样一件事, 用自然语言和 Java 语言分别是如何描述的。譬如, 我有一杯牛奶, 你有一杯咖啡, 想换着喝又不想交换杯子, 怎么办?

我们用自然语言是这么解决的:

- 第一步 另拿一只新杯子;
- 第二步 把我杯中的牛奶倒入新杯子;
- 第三步 把你杯中的咖啡倒入我的杯子;
- 第四步 把新杯子里的牛奶倒入你的杯子。

用 Java 语言则是这样描述的:

1. newcup = new(cup); // 另拿一只新杯子
2. newcup = mycup; // 我的牛奶倒入新杯子
3. mycup = yourcup; // 你的咖啡倒入我的杯子
4. yourcup = newcup; // 新杯子里的牛奶倒入你的杯子

以上四句就是与自然语言相对应的计算机语言。

下面再看一下,有两个数 a,b,如何求两数中的较大者?

我们只要将两个数比较一下,如 a 大于 b,则 a 是较大者,否则,b 是较大者。用 Java 表达:

```
if (a > b)
    c = a;
else
    c = b;
```

这段话很容易看明白,它把 a 与 b 之中较大的那一个值赋给了 c,如 a 为 3,b 为 4,则 c 为 4。

其实,用 Java 语言描述问题与用自然语言非常相似,只是所使用的符号不同而已。计算机语言也是由词构成句,由句构成文,所以也有其相应的词法、语法和文法规则。但人具有思考能力,生活中我们说话可以比较随意,同一件事可采用不同的说法,如“吃饭没?”、“饭吃了没?”,听者都能理解,而计算机则没有这种能力,所以计算机语言的使用不可能与自然语言一样灵活。计算机是采用形式化语言来进行问题描述的,这就使计算机语言的表达比自然语言刻板得多。用计算机语言描述问题,必须做到准确、无二义,一是一、二是二,先做什么、后做什么,都必须很明确地说明和限定。

1.3 程序设计与面向对象的程序设计

如上所述,程序就是为解决一个问题而实施的一系列步骤。用计算机语言描述的、完成一个任务的语句序列就称为**计算机程序**(以下简称程序),计算机程序设计就是考虑如何用计算机程序解决问题。

程序设计方式可以分为面向过程的程序设计和面向对象的程序设计。面向过程的程序设计是传统的程序设计方法,而面向对象的程序设计是一种崭新的程序设计方法。

举例来说,在自然经济时代,要吃饭也不是件容易的事,菜、米得自己种出来;柴火得自己去捡;然后割菜、洗菜、切菜、配菜、点火、炒、煮、炖。在市场经济时代,你到超市买点熟菜,再买点净菜回家一炒,很快就能吃上可口的饭菜;更进一步,你只要到喜欢的餐馆,告诉侍者想吃什么就行。同样是吃饭,在自然经济时代就好比是用面向过程的方法完成,而在市场经济时代则是用面向对象的方法实现。

再看一看建筑问题,如要搭一个窝棚,你当然能行,和泥、垒砖、搭棚,或许只要半天时间,而且可以做得很好。但假定要建一座漂亮、坚固的小屋,你肯定会请建筑工程队来做,这是再自然不过的事了。你会用到泥瓦工、管道工、油漆工,但你只需关心房屋的形状(墙的大小、厚薄)、功能、门窗开在何处。至于墙该如何砌、窗该何时安装等等都由泥瓦工决定和完

成。同样的道理,泥瓦工只关心所用的砖是什么样的,而砖是如何烧出来的则由窑厂考虑。你还要考虑几道工序之间的衔接:土木工程结束后是由泥瓦工通知你,你再通知管道工开始布排管线呢,还是由泥瓦工直接通知管道工。

面向对象的程序设计借鉴的就是这个思路。面向对象的程序设计语言提供了大量像“砖头”、“门”、“窗”这样的半成品,当要搭简单的窝棚时,你只需按要求想办法将砖头垒起来;它甚至提供“建筑工程队”,你告诉它要何种房子,“工程队”就帮你造好。也就是说在面向对象的程序设计环境里,你不需要也不应该“事事自己动手”。

显而易见,面向对象的程序设计带来的明显优势就是高效和资源的可重用。例如,买来砖造房子显然比烧出砖来再造房子快,工程队成立后既可帮你造房子也可去帮别人造房子。

1.4 面向对象程序设计的基本概念

“面向对象程序设计”或“面向对象编程”简称为 OOP(Object Oriented Programming)。OOP(面向对象编程)是目前占据主流地位的一种程序设计技术。Java 是一个纯面向对象的语言,它所提供的面向对象技术为软件开发人员和软件产品提供了很多便利。在学习用 Java 进行程序设计前,本节将介绍面向对象程序设计的一些重要概念,包括对象(Object)、类(Class)、消息(Message)、继承(Inherit)等。

1.4.1 对象

对象经常被用来模拟现实世界中的实体,你正在读的书是一个对象、你每天乘的汽车或骑的自行车是对象、你脚边的小卷毛狗也是一个对象……

实体总是通过其特性来进行描述的,通过提取实体的特性,从中抽象出状态和动作,由此可以构成对象的两大特征:属性和行为。例如,对汽车这一对象来说,其属性有门、窗、车轮数、时速、方向盘、刹车装置等,其行为有开门、关窗、刹车、转向等。而小狗这一对象的属性状态有名字、毛色、年龄等,其行为有跑、跳、吠等。

软件对象用以表示现实世界中的物体,其对象模型是根据现实世界中物体的状态和动作来确定的,程序设计中的对象用变量表示其属性、用方法描述其行为。

在程序中,不仅现实世界中的物体可以表示为一个对象,一个抽象的概念也可以是一个对象,如在图形用户界面交互式窗口系统中,“事件”就是一个普通的对象,用来表达用户对鼠标或键盘的操作。

对象中的变量(属性)构成了该对象的核心,而方法则是包围在外,将其变量和方法实现细节隐藏起来。要改变这些变量的值(也就是改变其属性),就必须调用在该对象中定义的方法,这就是封装。见图 1-1。

比如,当汽车需要急停时(时速变为 0),你总是通过刹车装置来进行刹车,而不会直接去设法

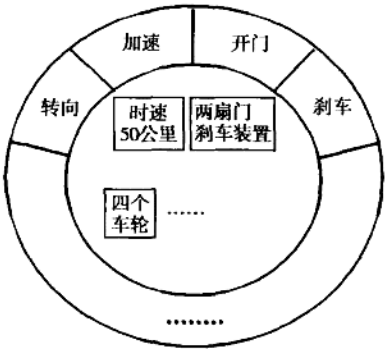


图 1-1 汽车对象及其封装

让车轮停止转动。至于刹车装置是如何工作的,完全由汽车的设计者、生产者考虑和实现,驾驶员无需关心。

封装有如下两大优点:

- ① 模块化 可以独立于其他对象进行开发、维护和修改该对象的源代码;
- ② 信息隐藏 一个对象有一个公共的接口为其他对象所使用,在保证公共接口不变的前提下,每一个对象可以修改私有的信息和方法,而不影响其他相关对象。

例如,不管刹车装置是如何工作的,刹车总是让车辆减速,所以当汽车厂商改变刹车装置的工作原理时并不影响驾驶者对刹车的使用。

1.4.2 消息

通常一个大的应用程序中有多个对象,这些对象通过相互作用和合作完成更加高级的功能和复杂的行为。软件对象之间的交互和通信是通过彼此发送消息来完成的。

例如,在大扫除中,劳动委员向擦窗者、扫地者、拖地板者发布“开始”消息,擦窗者、扫地者在完成任务后向拖地板者发送已“完成”消息,拖地板者完成后向劳动委员发送“完成”消息,劳动委员在检查完毕后向擦窗者、扫地者、拖地板者发送“结束”消息,大扫除结束。大扫除中消息传递过程见图 1-2。

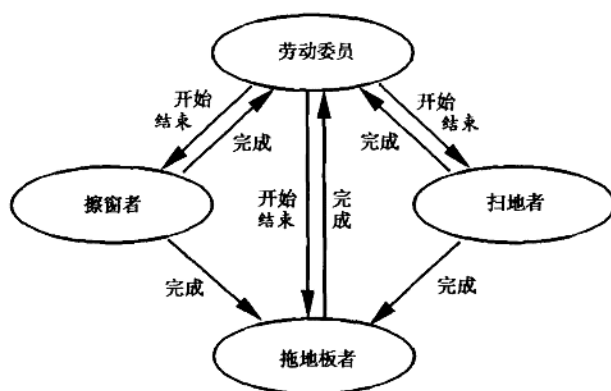


图 1-2 大扫除中消息的传递

从上图可以看出,对象间的消息传递往往是双向的,而且,在每个方向上可以定义多个不同的消息。

消息的三个组成部分如下:

- ① 指定接收方对象;
- ② 需调用方法的名字;
- ③ 执行方法所需要的参数。

例如,劳动委员发送的扫地信息是给扫地者的,扫地者执行的方法是扫地,参数是打扫的范围。

消息的优点如下:

- ① 消息提供了对象交互的统一手段;
- ② 不同进程中或不同机器上的对象也可以通过消息相互作用。

1.4.3 类和对象(实例)

类是某一类对象的原型,在现实世界中,通常有许多对象属于同一类。例如,你的自行车(对象)只是许多同类自行车中的一辆,因而称你的自行车为该自行车类的一个“实例”(instance)。

类和对象非常相似,它们之间不明显的差异常引起理解上的困惑。

在现实世界中类很明显地区分于它描述的对象,类是一个概念,而实际对象是一个实体。如房屋建筑设计图是一个类,依此建造的房屋就是该类描述的一个对象。

但是在软件中区分类和对象有一些困难,部分原因是软件中的对象只是现实世界中物体或抽象概念的电子模型,此外,还与许多人在类和实例中不一致地使用“对象”这个词有关。

类是对象的原型(抽象模型),对象是具有类指定特性和方法的实体,由类得到对象的过程称为类的实例化。正如按同一建筑设计图可以造很多相似的房子一样,由同一类也可生成许多对象。

1.4.4 继承

继承是面向对象技术的一个重要特性,它允许编程人员在已有类的基础上作进一步的引申。类的层次结构可用图 1-3 所示的例子加以说明。

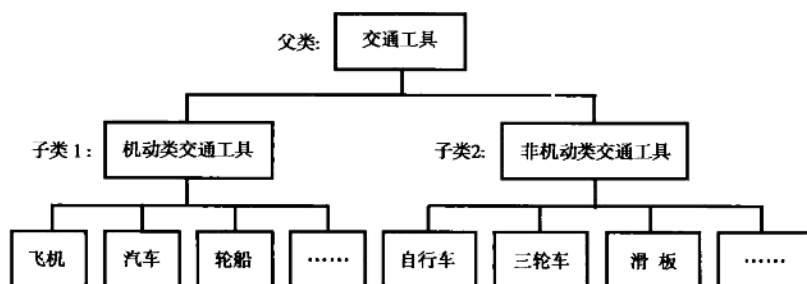


图 1-3 类继承的层次结构

交通工具是一个类,把它作为父类可以引申出机动类和非机动类两个子类,从这两个子类作进一步的引申,就分别得到飞机、汽车或自行车、三轮车等子类。这反映了一个从一般到具体的不断衍生的过程。父类和子类在逻辑上的关系是:子类是父类的一种。

每个子类从父类处继承属性和行为,但是子类并不局限于其父类所拥有的变量和方法的定义,它们可以在继承的基础上扩展自己的状态和行为。继承可以是多层次的。

继承有如下两大优点。

(1) 可重复使用。

交通工具共有的属性:大小、重量、时速等,在“交通工具”中定义后,不需要在子类中再重复定义。在机动类交通工具中我们可以增加发动机功率等新属性,这些属性不需要在飞机、汽车等类中重复定义,只要声明它们是继承关系,这些属性和方法都会加入到子类中,程序员不必重写这些代码。

(2) 父类的最主要的部分被定义以后,其他的细节可以留待以后(子类中)再实现。这样为类的使用者提供一个统一的接口。

在父类交通工具中定义“制动”这一行为,而具体的实现方法可在它的各个子类中实现,如汽车靠刹车装置,而滑板通过垂直转弯实现等。不管使用何种交通工具,当需要减速时你只要“制动”即可。

本书目标

初学者通过本课程的学习,应该达到下列要求:

- (1) 掌握基于 Java 技术的面向对象程序设计思想;
- (2) 掌握 Java 的基本程序设计概念和技术;
- (3) 了解 Java 的资源组成和结构,重点掌握其中最基本、常用的一部分,并具备进一步利用 Java 资源的能力;
- (4) 掌握简单 GUI 的设计和实现(利用 Java 的 Swing 包中基本构件)。

此外,对 Java 的高级概念和编程技术,如多线程、例外处理机制、网络应用、数据库接口 JDBC、多媒体等,本书将有选择地进行入门介绍,辅以可运行的例子让大家有一个概念(Java 具有很强的能力),一旦有应用需要,就知道该从何处入手。

参考资料

Java 是集语言、开发环境、运行环境于一体的综合技术,涉及的内容多、应用面广,并且它在不断地更新和发展。很难通过一本教材、一门课程就学好它,本书作为入门指导,只能就其基本知识作铺垫性介绍,需要进一步的学习才能真正熟练地掌握 Java 语言。下面推荐一些网址,在那里可以获得最新而且实用的信息。

http://www.mantiscorp.com/java/	提供了到 Java 其他站点的链接,通过它可以及时了解 Java 语言的重要发展
http://www.javasoft.com	Sun 公司在这里提供最新的 Java 工具、文档和源代码例子
http://www.gamelan.com	该站点提供综合性的 Java 小应用程序和源代码资源库
http://www.jar.com	在这里可找到最好的小应用程序
http://www.javaworld.com	是专门针对 Java 开发的基于 Web 的杂志
http://www.java.sun.com	这个站点可以找到 Java 语言规范

结语

餐厅的厨师能在几分钟内就烧一盘可口的菜,你在家就不行,为什么? 因为有许多人在为厨师作准备,他们用的都是准备好的原材料,姜已切成丝或挤成姜汁、葱已切成末……

与此类似,有了面向对象程序设计语言,你就能像厨师烧菜一样迅速地设计程序,因为面向对象程序设计语言提供了大量的“预制件”或“半成品”,你要做的事就是能看懂它们,必要时按要求进行适当的修改,然后合理地把它们组装起来。听起来不错吧? 但是要记住计算机是很“笨”的,没有任何灵活性,你必须讲它的语言(如 Java),并且严格按语言规定的语法去描述你的要求,它才会按你说的去做,只要有一丁点差错,计算机就会对你的命令不予理睬。下面就让我们开始,看看 Java 提供哪些“原料”、该如何告诉计算机你的要求、可以用哪些词、说哪些话……

本章介绍 Java 的发展简史,说明 Java 平台及程序基本结构,描述 Java 程序的编辑、编译以及执行过程。

2.1 Java 简介

2.1.1 Java 是什么

Internet 将世界各地成千上万的计算机子网连成一个庞大的计算机网,而这些子网是由各种各样不同型号、不同规模、使用不同操作系统,从而具有不同应用软件平台的计算机构成的。这样就自然提出一个问题:有没有一种语言,用这种语言编写的程序在各种不同的计算机上都能运行?有,Java 就是这种语言。由于它与 Internet 的成功结合而风靡全球,从而得到广泛的应用。通常说的 Java 有三层含义,首先是一种编程语言,同时又是一种开发环境和一种应用环境。

2.1.2 Java 的历史

Java 是 Sun 公司的产品,该公司一直以经营工作站闻名,但由于 PC 机功能越来越强,对 Sun 的工作站构成很大的压力,于是该公司希望通过开拓电子类消费产品市场来减轻这种压力。

James Gosling 受命领导 Green 小组,负责开发能对电子类消费产品实现互操作的软件。由于电子类消费产品硬件更新很快,假如用 C 等传统计算机语言开发软件就不得不为每一代产品写不同的程序,这是生产商不愿看到的。为此,他们设计了一种新的语言,其目标是用以开发可靠、紧凑、易于移植的分布式嵌入系统,并用办公室窗外的一棵橡树命名为 Oak。后来 Green 小组发现曾经有过一种叫 Oak 的语言,于是决定用 Java 作为新语言的名字,寓意为“请你喝杯热咖啡”,在 Java 的文档中,你常常可以见到一杯冒着热气的咖啡。

尽管 Java 语言在技术上非常成功,但由于商业上的种种原因,这些电子类消费产品未能推向市场,Java 也差点夭折。然而,Internet 和 WWW 的飞速发展却为 Java 提供了广阔的舞台,Java 与生俱来的跨平台特性使其在 Internet 和 WWW 环境下如鱼得水,加上 Sun 公司通过 Internet 让全球的软件设计者免费使用 Java,使得 Java 受到越来越多的重视。同时,他们用 Java 开发了一个 Web 浏览器 HotJava,它在 Internet 上充分展示了 Java 的风采,也真正有效地将 Java 推向全球,使之迅速风靡全球。

2.1.3 Java 的特性

Java 具有跨平台、面向对象、多线程、半编译半解释、稳定性好、安全性好等特点,并且提供 Applet 功能以及与此相关的图形功能,因而取得成功。

1. 跨平台性

Java 语言最突出的特点是跨平台性,这是以往任何语言都不具备的特点。也就是说,用 Java 语言编写的程序可以在不同计算机上运行,这是因为 Java 语言中没有一个功能和工作平台有直接关系。

2. 面向对象

面向对象原先是自然科学中的一个通用术语,现在面向对象成了计算机技术的一个专用名词,它把程序实现的每一个具体功能作为类,然后用类来构成对象。其优点为:代码可重用、系统易扩展。以后的章节里会详细介绍。

3. 多线程

Java 采用多线程机制,系统只要花很小的开销就可以同时执行多个任务。

4. 半编译半解释

对于一个高级语言编写的源程序而言,编译是指将其翻译成可运行的目标程序,目标程序作为一个独立的文件可以无数次地运行。编译过程所需要的存储空间较大,同时,编译所需要的时间较长,但目标程序执行速度快,这种方式特别适用于重复使用的程序。而解释则是指对高级语言编写的源程序翻译一句执行一句,翻译和运行交叉进行,如果要再运行一次,那就必须重新翻译、重新执行,解释型语言适用于需要经常修改程序的情况。由于边翻译边执行,解释型语言的速度远远低于编译型语言。Java 是一个半编译半解释的语言,一个 Java 程序的执行过程是:先由 Java 编译器将其编译成字节码,运行时由 Java 运行系统上的字节码解释器将其翻译成机器语言。Java 字节码解释器的工作是边翻译边执行,但因为字节码已非常接近于机器码,所以速度仍相当快,而且在一个系统上编写并编译好的 Java 程序字节码可以不加任何变动地放到另一系统上运行。

5. 稳定性好

大多数程序设计语言在使用时,如果编写的程序有严重错误,常常会造成系统死机。但是,由于 Java 具有对运行错误的异常处理机制,所以,当用 Java 语言编写的程序出错时,系统会转入异常处理,并自动寻找相应的异常处理方法,而不使程序中断或系统死机。这一措施大大加强了 Java 系统的稳定性。

Java 对内存的垃圾收集机制其实是一种内存保护机制,这是 Java 系统稳定性的又一保证。这个机制使 Java 应用程序只能访问和修改有限的并经过许可的一部分内存,它改变了传统的应用程序可以访问内存任何区域、可以修改任何内存单元的做法,而这种做法恰恰最容易造成系统故障。另外,Java 不使用指针操作,从而避免了内存地址管理的混乱,也避免了有用数据的破坏,这也是对 Java 系统稳定性的有力支持。

6. 高度安全性

Java 的安全性是通过多个层次上的措施得到保证的。首先,从内存管理上加强安全,对程序员来说,内存分配是透明的,即程序员没有对内存的分配权,这样减少了许多内存出错的可能;另一方面,在 Java 中编译器也没有对内存布局的确定权,Java 把内存分配权交给运

行平台。此外,类装载器也为安全性提供了措施。类装载器在装载字节码时,将本地的类组成一个统一的空间,而将外来的类组成另一个空间,这样就为本地类建立了一种安全屏障。

Java 安全性最重要的保证来自字节码检验器,在字节码进行解释执行之前,字节码检验器通过一个规则验证程序,对每个代码段作安全检测。这样,保证进入解释执行系统的字节码符合以下条件:

- ① 符合访问权限规定;
- ② 不含错误指针;
- ③ 符合访问对象的规则;
- ④ 符合调用方法规则;
- ⑤ 不会使堆栈操作出错。

7. Applet

Java 的程序有两种应用形式,即 Application 和 Applet,也就是所谓的应用程序和小应用程序。应用程序指可以独立运行的程序,小应用程序是指只能在浏览器中运行的程序。Java 系统提供特殊的技术,可以使这种小应用程序的功能很方便地加入 Internet 的网页上去,从而使 Internet 网页增加各种图形效果,包括多媒体功能。正是 Applet 及其跨平台功能使 Java 和 Internet 成功地结合起来。

8. 编程简单

对初学者来说,最吸引人的是 Java 语言编程的相对简单、易学。由于 Java 是一个全新的语言,它不存在与老版本兼容的问题,语法相对要简单得多,再加上它是一个纯面向对象的语言,让程序员以自然的方式去考虑和解决问题,所以容易学习。

2.1.4 Java 语言的应用方向

Java 本身是一个嵌入系统,加上其虚拟机非常紧凑、小巧,任何规模的应用系统,小到家用电器、智能仪器,大到企业级的计算平台都可以使用它。

Java 的跨平台性,使它能很好地在不同系统之间进行数据交换和处理,完成协调控制、综合管理等功能。这方面的应用有基于 Intranet/ Internet 的企业计算、E-business 等等。

Java 具有良好的多媒体和网络支持功能,用 Java 语言可以设计质量很高的图形软件、可视化软件、动画软件和计算机交互软件,在商业、娱乐业和教育界有着广阔的应用空间。

由于 Java 语言具有 Applet 功能,使它能非常方便地将动画和各种信息嵌入网页,为 Internet 网络用户设计生动活泼的带动画的主页,因此 Java 对网络用户具有强大的吸引力。

2.2 Java 平台

Java 应用程序必须在 Java 平台上运行,一个 Java 平台包括实际计算机、适配器、Java 虚拟机、Java 基本软件和 Java 应用程序接口。如图 2-1 所示。

Java 虚拟机(JVM——Java Virtual Machine)是一种虚拟的计算机,从结构上看它与实际的计算机相似,它的作用是使得一台实际的机器能够运行 Java 字节码。

JVM 是 Java 平台的核心,JVM 的下面是和实际计算机的接口,被称为适配器,不同类型的计算机其适配器是不同的。Java 的基本软件也称为基本类,而 Java 应用程序接口

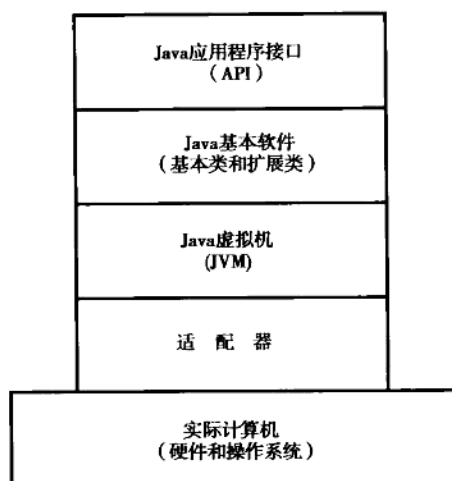


图 2-1 Java 平台

(API – Application Program Interface)是已编译好的程序代码库,就像建筑行业中的预制件和机械行业中的标准件一样,你可以直接使用它们以节约编程的时间。事实上,Java 的基本类和 API 的规模并不固定,许多 Java 平台对这两部分内容进行了补充,即除了基本类和 API 外,还有扩展类和扩展 API。

2.3 Java 程序的运行 *

Java 程序作为半编译半解释的语言,其程序的运行过程与编译型和解释型语言都不同。

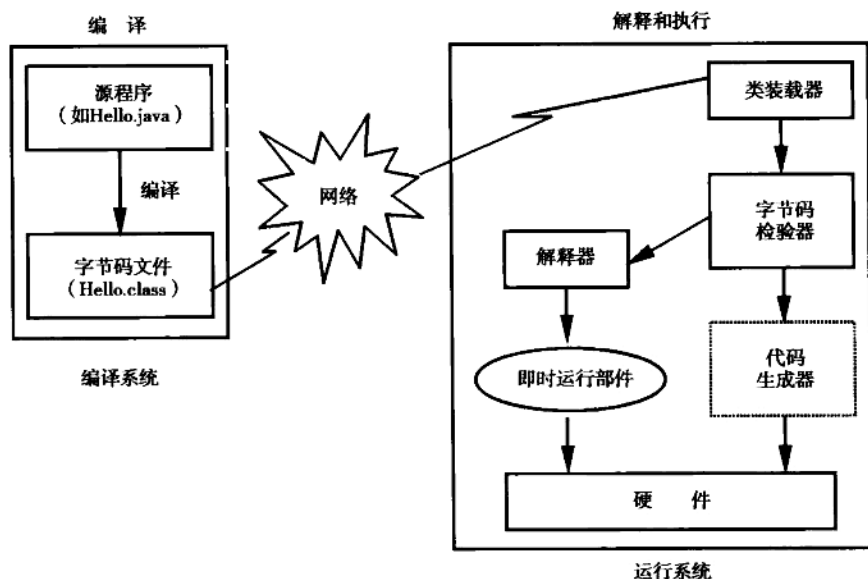


图 2-2 Java 编译和运行系统功能图