

21世纪

高等院校计算机系列教材

Java

程序设计教程

阎 菲 主 编
陈 利 向郑涛 陈宇峰 副主编



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机系列教材

Java 程序设计教程

阎 菲 主 编

陈 利 向郑涛 陈宇峰 副主编

中国水利水电出版社

内 容 提 要

本书是一本实用的 Java 教材。全书共分 9 章, 简明扼要、深入浅出地介绍了 Java 这种广泛使用的网络编程语言, 便于读者一开始就可以掌握与运用 Java 语言最基本的方法与概念。

本书注意涵盖 Java 开发工作中核心的观念, 包括 Java 的基础知识; 面向对象技术的基础理论、主要原则和思维方式; 类机制以及动态的接口模型等, 从而加强学生需要扎稳根基的种种主题。

本书指导读者编写出适合于各种情况的应用程序: 图形用户界面 (GUI) 技术, 使 Java 语言开发出的软件具有可视化、可操作化的效果; Java 的 Applet 应用程序, 可编写动态、安全、跨平台的网络应用程序; 介绍了多线程机制, 分别用不同的线程完成特定的行为, 实现网络上的实时交互行为。程序设计的例子不仅清楚, 而且易于理解, 照顾到 Java 中十分重要的细节, 而这些细节正是读者所需要的。全书给出大量调试通过的典型应用程序, 所有程序源代码均可从中国水利水电出版社网站下载, 网址为 <http://www.waterpub.com.cn>。

本书可作为高等院校本科各专业学生学习 Java 语言编程技术的教材, 也可供其他学生或高校教师以及相关专业技术人员参考使用。

本书有配套实验与实训指导, 并配有 Powerpoint 制作的电子教案。

图书在版编目 (CIP) 数据

Java 程序设计教程 / 阎菲主编. —北京: 中国水利水电出版社, 2004

(21 世纪高等院校计算机系列教材)

ISBN 7-5084-1798-4

I. J… II. 阎… III. JAVA 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 124681 号

书 名	Java 程序设计教程
作 者	阎 菲 主编 陈 利 向郑涛 陈宇峰 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机) 68331835 (营销中心) 82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092mm 16 开本 20.5 印张 465 千字
版 次	2004 年 2 月第 1 版 2004 年 2 月北京第 1 次印刷
印 数	0001—5000 册
定 价	27.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

计算机界不少人预言：“Java 语言的出现，将会引起一场软件革命”，这是因为传统的软件往往都是与具体的实现环境有关，换了一个环境就需要作一番改动，耗时费力，而 Java 语言能在执行码（二进制码）上兼容，这样以前所开发的软件就能运行在不同的机器上，只要所用的机器能提供 Java 语言解释器即可。Java 的诞生必将对整个计算机产业产生深远的影响，对传统的计算模型提出了新的挑战。有人预言：Java 将是网络上的“世界语”，今后所有的用其他语言编写的软件统统都要用 Java 语言来改写。

为使本书成为一本有较高参考价值、培养学生 Java 语言实用编程能力的教材，根据作者多年教学及工程实践中开发大型 Java 程序的经验，充分体现 Java 语言的特点：简单、面向对象、分布式、解释执行、鲁棒（抗打击性）、可移植、多线程以及动态性等特点。本书以专题的方式组织内容，本着由浅入深的原则，安排各章节的内容：

第 1 章：Java 和面向对象概述，强调 Java 网络应用程序有什么特点，阐述面向对象的软件开发过程及其优点。

第 2 章：从安装并配置 Java 平台开始，介绍了使用集成环境来编写 Java 程序以及 Java 语言的特点，使读者对 Java 语言有一个基本的了解。

第 3 章：通过本章的学习，读者应该能够对 Java 程序的构成有一个基本的理解，能够灵活运用 Java 语言提供的各种数据类型和运算符。

第 4 章：介绍 Java 编程中类和对象等非常重要的概念。

第 5 章：介绍继承和多态、接口、包及异常处理等深入进行面向对象分析的概念。

第 6 章：介绍掌握字符、字符串、数组的各种基本操作。

第 7 章：学习一些基本的图形编程知识，介绍 Java 的事件处理机制，系统地介绍图形用户界面中常用组件的用法，如布局管理器、文本框、复选框、菜单、对话框等。

第 8 章：Applet 具有“激活因特网”的强大功能。讲述 Applet 的工作原理以及如何编写 Applet 程序。

第 9 章：Java 语言中的一些高级特性：异常处理、多线程机制、流式 IO、网络通信以及远程方法调用，其中的每个部分都能够扩展成为一个专题。

本书以清晰自然的呈现方式，努力做到：目录安排得当，呈现方式清晰，让学生很快找出需要的章节。搭配恰到好处的示例，前后文相互呼应，技术上诚实，文笔流畅，用语准确，加强学生对重点、难点的理解。

本书的内容源自作者教学和科研活动的丰富经历。同学们对新技术的敏感和渴求在深深感动我们之余，也激励和敦促我们努力改进教学内容、方式和效果。本书就是这种努力的成果之一，它总结了作者两年以来的大量教学经验和科研心得，书中的个别例题甚至直接来源于科研活动的总结，为此要在这里感谢所有关心和支持这门课程的领导、老师和同学。

计算机学科是注重实践的学科，成功的软件开发人员无不经过大量上机实践的锻炼；只有在学习书本内容的同时辅以相应的实际练习和实验环节，才能真正掌握书中介绍的知识技能。为此本书中引入了大量的例题，并另配有一本专门的习题集和实训教材，只要读者能够按照书中的要求边学边练，一定能很快学有所成，享受 Java 语言和面向对象技术所带来的无限乐趣。

本书中所有的 Java 程序均在计算机上调试通过，其源代码可以从中国水利水电出版社网站下载，网址为：<http://www.waterpub.com.cn>。

本书第 1 章、第 4 章、第 5 章由阎菲编写，第 2 章、第 3 章、第 6 章、第 8 章由陈利编写，第 7 章由向郑涛编写，第 9 章由陈宇峰编写，阎菲、陈利和薛礼统稿并校对。参加本书大纲讨论及部分内容编写的还有罗小平、黄兴奎、刘玉南、贾瑜、王红芳、陈清文、李珍香、董福贵等。

由于时间仓促，作者水平有限，书中难免有错误和不妥之处，恳请读者与专家批评指正。如有意见和建议，请与作者联系。作者的 E-mail 为 yanfei131@163.com 或 laolang@dfminfo.com.cn。

作者

2003 年 12 月

目 录

前言

第 1 章 Java 和面向对象概述	1
1.1 Java 语言概述	1
1.2 面向对象软件开发方法概述	4
1.2.1 传统软件开发方法存在的问题	4
1.2.2 存在问题的原因	5
1.2.3 软件工程的新途径	5
1.3 面向对象的基本概念	7
本章小结	9
思考与练习	9
第 2 章 Java 概述	10
2.1 安装并配置 Java 平台	10
2.1.1 Java 平台概述	10
2.1.2 Java 平台的安装	10
2.2 第一个 Java Application 程序	12
2.2.1 编辑源程序	12
2.2.2 字节码的编译生成	14
2.2.3 字节码的解释与运行	14
2.3 使用集成开发环境	15
2.3.1 JCreator 的安装配置	15
2.3.2 JCreator 的使用	17
本章小结	18
思考与练习	18
第 3 章 Java 语言基础	19
3.1 Java 程序的构成	19
3.2 关键字、标识符、数据类型、常量与变量	19
3.2.1 关键字	19
3.2.2 标识符	20
3.2.3 数据类型	21
3.2.4 常量与变量	22
3.3 运算符	26
3.3.1 算术运算符	26

3.3.2	关系运算符	28
3.3.3	逻辑运算符	29
3.3.4	位运算符	30
3.3.5	类型转换	31
3.3.6	其他运算符	33
3.3.7	运算符的优先级与结合性	35
3.4	流程控制语句	35
3.4.1	结构化程序设计的三种基本流程	36
3.4.2	选择结构	36
3.4.3	循环结构	41
3.4.4	跳转语句	44
	本章小结	46
	思考与练习	47
第 4 章	类和对象	50
4.1	面向对象的基本概念	50
4.1.1	抽象	50
4.1.2	封装	52
4.1.3	对象	52
4.2	类	54
4.2.1	系统定义的类	54
4.2.2	用户程序自定义类	56
4.2.3	创建类及类的实例对象	56
4.2.4	构造函数	59
4.3	方法	61
4.3.1	方法概述	61
4.3.2	定义方法	63
4.3.3	方法的调用	64
4.3.4	JavaAPI 中的方法	64
4.4	访问控制符	67
4.4.1	公共访问控制符 public	67
4.4.2	缺省访问控制符	68
4.4.3	私有访问控制符 private	68
4.4.4	保护访问控制符 protected	69
4.4.5	私有保护访问控制符 private protected	70
4.5	静态修饰符、静态字段和方法	71
4.5.1	静态修饰符	71
4.5.2	静态字段	72

4.5.3	静态常量	73
4.5.4	静态方法	73
4.5.5	main 方法	74
4.6	抽象类与抽象方法	75
4.6.1	抽象类	75
4.6.2	抽象方法	77
4.7	最终类、最终属性与最终方法	80
4.7.1	最终类	80
4.7.2	最终属性	80
4.7.3	最终方法	81
	本章小结	81
	思考与练习	84
第 5 章	继承	86
5.1	继承的概念	86
5.1.1	继承的定义	86
5.1.2	父类和子类	87
5.2	扩展类	88
5.2.1	继承关系的定义	88
5.2.2	属性继承与隐藏	88
5.2.3	方法继承、覆盖与重载	89
5.2.4	在子类中使用构造函数	91
5.2.5	父类对象与子类对象的关系	92
5.2.6	扩展类继承的应用示范	95
5.3	多态与动态绑定	99
5.3.1	多态	100
5.3.2	动态绑定	100
5.3.3	父类对象与子类对象的使用与转化	101
5.3.4	多态性在工资系统中的应用	103
5.4	构造函数的继承与重载	110
5.4.1	默认字段初始化	110
5.4.2	默认构造函数	110
5.4.3	显式字段初始化	111
5.4.4	参数名	111
5.4.5	调用另一个构造函数	112
5.4.6	初始化块	112
5.4.7	构造函数的继承与重载示例	114
5.5	包	115

5.5.1	包的用途	115
5.5.2	创建包	116
5.5.3	引用包	116
5.5.4	Java API 包	118
5.6	接口	119
5.6.1	接口的概念	119
5.6.2	接口声明	120
5.6.3	接口实现	121
	本章小结	126
	思考与练习	128
第 6 章	字符、字符串和数组	129
6.1	字符	129
6.1.1	Character 类的构造函数	129
6.1.2	Character 类提供的常用方法	129
6.2	字符串	131
6.2.1	字符串常量与 String 类	132
6.2.2	字符串变量与 StringBuffer 类	140
6.3	数组	143
6.3.1	数组的创建和初始化	143
6.3.2	数组的数组	146
6.3.3	复制数组	148
6.3.4	数组类 Arrays	150
6.3.5	命令行参数	151
	本章小结	152
	思考与练习	153
第 7 章	图形用户界面的设计与实现	154
7.1	图形用户界面概述	154
7.2	简单图形用户界面	155
7.2.1	第一个图形用户界面	155
7.2.2	框架外观	157
7.2.3	在框架中显示信息	159
7.2.4	文本和字体	161
7.2.5	绘制简单图形	163
7.2.6	颜色	166
7.2.7	图像	169
7.3	事件处理	172
7.3.1	事件处理原理	172

7.3.2	处理按钮点击事件	173
7.3.3	捕获窗口事件	178
7.3.4	AWT 事件层次结构图	180
7.3.5	焦点事件	183
7.3.6	键盘事件	184
7.3.7	鼠标事件	188
7.4	Swing 组件	194
7.4.1	Swing 组件的类继承关系	194
7.4.2	布局管理器	195
7.4.3	文本组件	199
7.4.4	标签	206
7.4.5	按钮	206
7.4.6	Swing 常用的选择类组件	208
7.4.7	边界	221
7.4.8	菜单以及弹出式菜单	224
7.4.9	对话框	233
	本章小结	235
	思考与练习	235
第 8 章	Applet	237
8.1	Applet 的基本工作原理	237
8.2	第一个 Applet 小程序	239
8.2.1	源程序的编辑与编译	239
8.2.2	代码嵌入	239
8.2.3	Applet 的运行	240
8.3	Applet 基础	241
8.3.1	Applet 类与 JApplet 类	241
8.3.2	Applet 与 Application 的区别	243
8.3.3	Applet 安全基础	244
8.4	Applet 的应用	244
8.4.1	向 Applet 传递参数	244
8.4.2	Application 转化为 Applet	246
8.4.3	在 Applet 中弹出窗口	250
8.4.4	Application 与 Applet 组合	251
8.5	Applet 与 JAR 文件	254
8.5.1	JAR 文件概述	254
8.5.2	jar 命令	254
8.5.3	JAR 缓存	256

本章小结	256
思考与练习	257
第 9 章 Java 高级编程	258
9.1 异常处理	258
9.1.1 异常分类	258
9.1.2 方法如何抛出异常	259
9.1.3 捕获及处理异常	260
9.1.4 创建自己的异常类	264
9.2 Java 多线程机制	266
9.2.1 创建线程	270
9.2.2 实现多个线程	273
9.2.3 线程的状态	275
9.2.4 Runnable 接口	276
9.3 流式输入输出与文件处理	280
9.3.1 Java 输入输出类库继承关系	281
9.3.2 基于标准输入输出的 IO 操作	282
9.3.3 文件读写及随机访问	287
9.3.4 Java 的文件管理	299
9.4 Java 网络通信	302
9.4.1 网络基础知识及 Java 网络模型	302
9.4.2 面向连接的流式套接字	304
9.4.3 无连接的数据报	313
9.4.4 Java 访问网络资源	313
本章小结	315
思考与练习	316
参考文献	318

第 1 章 Java 和面向对象概述

学习导读

本章将概要介绍 Java 和面向对象的基本概念。

Java 语言是对软件开发技术有深远影响、应用前景广泛、具有丰富的类库、继承了 C++ 传统（摒弃了某些不足）而广泛使用的网络编程语言。Java 语言的特性使它可以最大限度地利用网络。

本章介绍面向对象的基本概念：对象和类，对象的属性。

教学重点与难点

- Java 的特点
- Java 对 C/C++ 的继承和改进
- Java Applet 特性使它可以最大限度地利用网络
- 了解面向对象的软件设计（OOD）中软件分析、设计、实现、维护等基本环节
- 初步了解对象、类、属性、方法、消息、对象和对象变量等基本概念

1.1 Java 语言概述

如今，性能强大的个人计算机可以与十年前几百万美元的计算机相比。个人计算机以及工作站为用户提供了许多新的功能，信息可以很方便地通过计算机网络实现共享。作为网络文件服务器的计算机，可以给分布在网络上不同站点的客户提供共享的公用程序和数据，这就是客户/服务器计算模式。

过去，大多数计算机运行在没有与其他机器连接的单机上。现在 WWW 为您提供了一个可以轻松驾驭的图形化用户界面，以查阅 Internet 上的文档，这些文档与它们之间的链接一起构成了一个庞大的信息网，使得全世界的信息可以及时、方便地获得。无论是生产制造、电子商务，还是服务咨询领域，软件都已成为担当重任的核心力量，互联网和软件已成为推动新经济发展的重要基础。

由于 Java 语言的突出特点，一些著名的计算机公司纷纷购买了 Java 语言的使用权，微软公司认为“Java 是长时间以来最卓越的程序设计语言”，并以此软件确定微软整个软件开发的战略从 PC 单机时代向着以网络为中心的计算时代转移，购买 Java 是微软的重大战略决策的实施部署。

1. Java 语言对软件开发技术的影响

计算机界不少人预言：“Java 语言的出现，将会引起一场软件革命”，这是因为传统的

软件往往都是与具体的实现环境有关, 换了一个环境就需要作一番改动, 耗时费力, 而 Java 语言能在执行码 (二进制码) 上兼容, 这样以前所开发的软件就能运行在不同的机器上, 只要所用的机器能提供 Java 语言解释器即可。Java 语言将对未来软件的开发产生影响, 可从如下几个方面考虑:

(1) 软件的需求分析: 可将用户的需求进行动态的、可视化描述, 以满足设计者更加直观的要求。而用户的需求是各种各样的, 不受地区、行业、部门、爱好的影响, 都可以用 Java 语言描述清楚。

(2) 软件的开发方法: 由于 Java 语言的面向对象的特性, 所以完全可以用面向对象的技术与方法来开发, 这是符合最新的软件开发规范要求的。

(3) Java 语言的动画效果: Java 语言比图形用户界面 (GUI) 技术能创建更加逼真的动画效果, 尤其是利用 WWW 提供的巨大动画资源空间, 可以共享全世界的动态画面的资源。

(4) 软件最终产品: 用 Java 语言开发的软件可以具有可视化、可操作化的效果, 这要比电视、电影的效果更为理想, 因为它可以做到“即时、交互、动画与动作”, 要它停就停, 要它继续就继续, 而这是在电影与电视播放过程中难以做到的。

(5) 其他: 使用 Java 语言对开发效益、开发价值都有比较明显的影响。

2. Java 语言的应用前景及特点

Java 语言有着广泛的应用前景, 大体上可以从以下几个方面来考虑:

(1) 所有面向对象的应用开发, 包括面向对象的事件描述、处理、综合等。

(2) 计算过程的可视化、可操作化的软件的开发。

(3) 动态画面的设计, 包括图形图像的调用。

(4) 交互操作的设计 (选择交互、定向交互、控制流程等)。

(5) Internet 的系统管理功能模块的设计, 包括 Web 页面的动态设计、管理和交互操作设计等。

(6) Intranet (企业内部网) 上的软件开发 (直接面向企业内部用户的软件)。

(7) 与各类数据库连接查询的 SQL 语句实现。

(8) 其他应用类型的程序。

3. Java 语言的特点

Java 是一个广泛使用的网络编程语言, 它是一种新的计算概念。Java 语言有下面一些特点: 简单、面向对象、分布式、解释执行、鲁棒性 (抗打击性)、安全、体系结构中立、可移植、高性能、多线程以及动态性。

(1) 简单。Java 语言是一种面向对象的语言, 它通过提供最基本的方法来完成指定的任务, 只需理解一些基本的概念, 就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念, 并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外, Java 也适合于在小型机上运行, 它的基本解释器及类的支持只有 40KB 左右, 加上标准类库和线程的支持也只有 215KB 左右。

(2) 面向对象。Java 语言的设计集中于对象及其接口, 它提供了简单的类机制以及

动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

(3) 分布性。Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

(4) 鲁棒性。Java 在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时容易产生的错误。通过集成的面向对象的例外处理机制，在编译时，Java 提示出可能出现但未被处理的例外，帮助程序员正确地进行选择以防止系统的崩溃。另外，Java 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

(5) 安全性。用于网络、分布环境下的 Java 必须要防止病毒的入侵，Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止了程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

(6) 体系结构中立。Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行时系统，Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

(7) 可移植性。与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语言实现的，Java 运行时系统由标准 C 实现，这使得 Java 系统本身也具有可移植性。

(8) 解释执行。Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时信息，使得连接过程更加简单。

(9) 高性能。和其他解释执行的语言如 BASIC 不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

(10) 多线程。多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就能很容易地实现网络上的实时交互行为。

(11) 动态性。Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

4. Java Applet

Java 语言的特性使它可以最大限度地利用网络。Applet 是 Java 的小应用程序，它是动态、安全、跨平台的网络应用程序。Java Applet 嵌入 HTML（超文本标记语言）语言，通过主页发布到 Internet。网络用户访问服务器的 Applet 时，这些 Applet 从网络上进行传输，然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制，用户一旦载入 Applet，就可以放心地生成多媒体的用户界面或完成复杂的计算，而不必担心病毒的入侵。虽然 Applet 可以和图像、声音、动画等一样从网络上下载，但它不同于这些多媒体的文件格式，它

可以接受用户的输入，动态地进行改变，而不仅仅是动画的显示和声音的播放。

5. 丰富的类库

Java 提供了大量的类，以满足网络化、多线程、面向对象系统的需要。

(1) 语言包提供的支持包括字符串处理、多线程处理、例外处理、数学函数处理等，可以用它简单地实现 Java 程序的运行平台。

(2) 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

(3) 输入输出包用统一的“流”模型来实现所有格式的 I/O，包括文件系统、网络、输入及输出设备等。

(4) 低级网络包用于实现 Socket 编程。

(5) 抽象图形用户接口包实现了不同平台的计算机的图形用户接口部件，包括窗口、菜单、滚动条、对话框等，使得 Java 可以移植到不同平台的机器。

(6) 网络包支持 Internet 的 TCP/IP 协议，提供了与 Internet 的接口。它支持 URL 连接，WWW 的即时访问，并且简化了用户/服务器模型的程序设计。

6. Java 和 C/C++

对于变量声明、参数传递、操作符、流控制等，Java 使用了和 C/C++ 相同的传统，使得熟悉 C/C++ 的程序员能很方便地进行编程。同时，Java 为了实现其简单、鲁棒、安全等特性，也摒弃了 C 和 C++ 中许多不合理的内容。

Java 的诞生必将对整个计算机产业产生深远的影响，对传统的计算模型提出了新的挑战。有人预言：Java 将是网络上的“世界语”。

1.2 面向对象软件开发方法概述

1.2.1 传统软件开发方法存在的问题

1. 软件生产率低

随着计算机技术、电子技术的惊人进步，计算机与全球互联网络 Internet 相连接，使今天的社会进入了以计算机为核心的信息社会。在信息社会中，信息的获取、处理、交流和决策都需要大量高质量的计算机软件，这样就促使人们对计算机软件的品种、数量、功能、质量、成本和开发时间等提出了越来越高的要求。

然而，不幸的是，要想使软件功能越强、使用越方便，开发出来的软件就越庞大，人们的软件开发能力就显得越力不从心。有资料表明：以美国的软件为例，经过 1955 年到 1985 年，美国的软件生产的增长率是较慢的，30 年间只翻了将近两番。软件生产增长速度远远低于社会对计算机软件需求量的增长速度，以至软件开发计划一拖再拖，成本失去控制，软件质量得不到保证。

2. 软件重用程度低

利用可重用的软件成分开发软件的技术叫做软件重用，软件重用是节约人力，提高软件生产率的重要途径，这样将有助于加快开发进度。结构化分析方法没能解决软件重用问题，

几乎每开发一个新的软件系统时，都要针对这个具体的系统做大量重复而又烦琐的工作。

3. 软件维护艰辛

软件研制成本越大，软件维护成本就更大。软件维护成本是指修复那些已正式发行的以及正在运行的软件错误所需的代价总和。这部分工作量十分巨大，它往往接近、甚至超过软件的初期成本。

4. 软件非用户所需

应当指出，在描述系统需求时，有时系统的要求是模糊的，同时系统的变动是不可避免的。所以用传统方法开发出的软件系统往往不能真正满足用户的需要。据报道，在美国开发出的软件系统中，真正符合用户需要并顺利投入使用的系统仅占总数的四分之一，因此造成了用户对“已经完成”的软件系统不满意的现象经常发生。

1.2.2 存在问题的原因

存在上述这些问题的原因是：传统结构化技术的缺点。

面向过程的语言把注意力从完成某一任务或功能的机器转移到了问题本身，致力于用计算机能够理解的逻辑来描述需要解决问题的具体方法和步骤，面向过程的程序的核心是数据结构和算法，要解决的问题并不依赖于具体的计算机，使得程序的移植推广成为可能。缺点是管理维护困难，可重用性差等致命弱点。

首先，结构分析与结构设计技术的本质是功能分解，是围绕实现处理功能的过程来构造系统的。结构化方法强调过程抽象和模块化，将现实世界映射为数据流和加工，加工之间通过数据流进行通信，数据作为被动的实体被主动的操作所加工，是以过程（或操作）为中心来构造系统和设计程序的。

然而用户需求的变化大部分是针对加工的，因此这种变化对基于过程的设计来说是灾难性的，用这种技术设计出的系统往往是不稳定的。也就是说，用户需求的变化往往造成系统结构的较大变化，从而需要花费很大代价才能实现这种变化。

其次，结构分析与结构设计技术清楚地定义了目标系统的接口。当系统对外界的接口发生变动时，例如，设备的支持程度，其他系统的要求，对其他系统的要求，以及人在使用系统时的要求，当上述情况发生变动时，结构分析与结构设计技术很难扩充新的接口，也就是说，这样的系统较难修改和扩充。

第三，结构化方法从本质上仍具有冯·诺依曼计算机的结构特点，把数据和操作作为分离的事实，以至在实现阶段，一些具有潜在可重用价值的软件部件（也称为软构件）已和具体应用环境密不可分。上述种种原因都使得用结构分析与结构设计技术开发出的软件可重用性较差。

1.2.3 软件工程的新途径

为克服传统方法的缺点，人们在实践中逐渐创造出面向对象方法等软件工程的新途径。20 世纪 80 年代后期，面向对象的语言开始出现，从而面向对象的思想正式进入了整个软件开发界。面向对象作为一种思想及编程语言，为软件开发的整个过程——从分析设

计到实现, 提供了一个完整的解决方案。面向对象堪称是软件发展过程中取得的里程碑式的伟大成就。

1. 面向对象方法简介

从 20 世纪 80 年代后期开始, 进行了面向对象分析 (OOA)、面向对象设计 (OOD) 和面向对象程序设计 (OOP) 等新的系统开发方式模型的研究, 在有些文献中统称为 OO 范型。

我们先要回答什么是面向对象这个基本问题。为了研究什么是面向对象, 先要考察一下为什么需要面向对象。

计算机的能力可以概括为 3 个方面:

- (1) 计算能力。
- (2) 推理能力。
- (3) 人机交互能力。

能同时支持这 3 种能力开发的技术是面向对象技术。面向对象是一种新的非常有效的程序设计范型, 对软件的生产率、可靠性、可重用性等都有帮助, 面向对象正在为新的信息系统的设计方法论开辟道路, 面向对象的作用还涉及更加广泛的领域。所以面向对象是正在兴起的影响面最广的新技术。

2. 面向对象方法的主要优点

面向对象方法在某种程度上可以克服传统方法学的缺陷, 缓解软件危机, 它具有许多特点, 主要表现在:

(1) 符合人们通常的思维方式。面向对象方法强调把问题域的概念直接映射到对象以及对象之间的接口, 符合人们通常的思维方式, 减少了结构化方法从问题域到分析阶段的映射误差。

传统的程序设计技术忽略了数据和操作之间的内在联系, 用这种联系方法设计出来的软件系统其解空间与问题空间并不一致, 令人感到难以理解。实际上, 用计算机解决的问题都是现实世界中的问题, 这些问题无非是由相互间存在一定联系的事物组成的。每个具体的事物都具有行为和属性两方面的特征。因此, 把描述事物静态属性的数据结构和表示事物动态行为的操作放在一起构成一个整体, 才能完整、自然地表示客观世界中的实体。

面向对象的软件技术以对象 (Object) 为核心, 用这种技术开发出的软件系统由对象组成。对象是对现实世界的正确抽象, 它是由描述内部状态表示静态属性的数据, 以及可以对这些数据施加的操作 (表示对象的动态行为), 封装在一起所构成的统一体。对象之间通过传递消息互相联系, 以模拟现实世界中不同事物彼此之间的联系。

将上述特点称为方法的惟一性, 它是对软件开发过程进行综合考虑得到的, 提高了程序设计的抽象层次, 为处理复杂软件提供了一种较好的方式。

(2) 高度连续性。面向对象方法从分析到设计再到编码采用一致的模型表示, 后一阶段可以直接复用前一阶段的工作成果, 弥合了结构化方法从数据流图到模块结构图转换的鸿沟, 减少了工作量和映射误差。

面向对象分析文档的视点不同于过程性分析的文档。传统的文档是面向功能的, 其视