HZ BOOKS

希赛®IT技术讲堂之Java篇

Java

面向对象编程

葛志春 刘志成 聂艳明 冯向科 等编著

邓子云 主审

- 根据读者的学习和认知规律来编写。采用提出问题、分析问题、解决问题、归纳概念、实践练习的思路。
- 把学习曲线变缓，把难点分散。用通俗易懂的语言来阐述抽象的术语，用生动的例子来解释复杂的概念。
- 本书所有实例的源代码，均可从华章和希赛网站下载。
- 作者在希赛网社区（http://www.csai.cn）“书评在线”版块中为读者提供全方位学习指导。

机械工业出版社
China Machine Press

希赛IT技术讲堂之Java篇

Java 面向对象编程

葛志春 刘志成 聂艳明 冯向科 等编著
邓子云 主审

机械工业出版社
China Machine Press

Java是当前最流行的程序设计语言之一。本书以Java最新版本Java SE5为基础，涵盖了Java SE5最新特性，由浅入深地介绍了Java SE5的主要内容。通过本书的学习，读者不仅可以全面掌握Java SE5，而且能够掌握与程序设计相关的知识，如面向对象思想理论与分析设计方法、UML、程序算法设计以及数据结构等。本书通俗易懂，并辅以大量的实例，使没有程序设计语言基础的读者，也可以轻松地掌握Java面向对象编程，为程序设计打好基础。

读者只要掌握一定的计算机基础知识，即可通过自学本书，轻松掌握Java程序设计语言及程序设计相关的知识，为深入掌握J2EE或J2ME等编程技术奠定扎实的基础。本书可以作为高职、高专、本科院校或计算机培训机构的教材，也可以作为计算机爱好者和程序员的自学教材或参考用书。

图书在版编目（CIP）数据

Java面向对象编程/葛志春等编著. -北京：机械工业出版社，2007.7
（希赛IT技术讲堂之Java篇）
ISBN 978-7-111-21725-1

Ⅰ.J…　Ⅱ.葛…　Ⅲ.JAVA语言-程序设计　Ⅳ.TP312

中国版本图书馆CIP数据核字（2007）第099539号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：李南丰
北京京北制版厂印刷 · 新华书店北京发行所发行
2007年8月第1版第1次印刷
184mm × 260mm · 27.25印张
定价：49.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

编写委员会

组编：希赛顾问团

顾问：张友生

主审：邓子云

编委：

边　伟	陈亿春	崔海波	方海光	冯向科
葛志春	郭　莹	赫　斌	黄　婧	黄少年
李　刚	刘　毅	陆秉炜	刘志成	娄嘉鹏
聂艳明	阮国明	孙鸿飞	施　游	唐　俊
唐天广	王　军	吴吉义	吴伟敏	薛大龙
王红安	王　冀	王　勇	谢　顺	杨　森
张晓燕	张立东	朱小平		

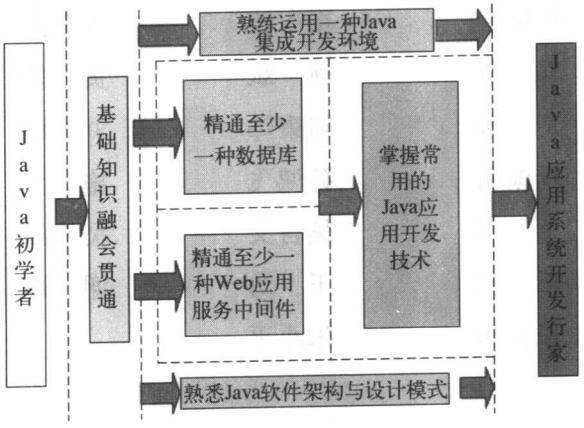
丛书介绍

Java语言的技术体系越来越庞大，J2EE的各种繁杂的技术与架构，随处可见的开源软件，使得Java程序员目不暇接，无所适从。程序员觉得每天都有学不完的知识。

一、本丛书的知识结构

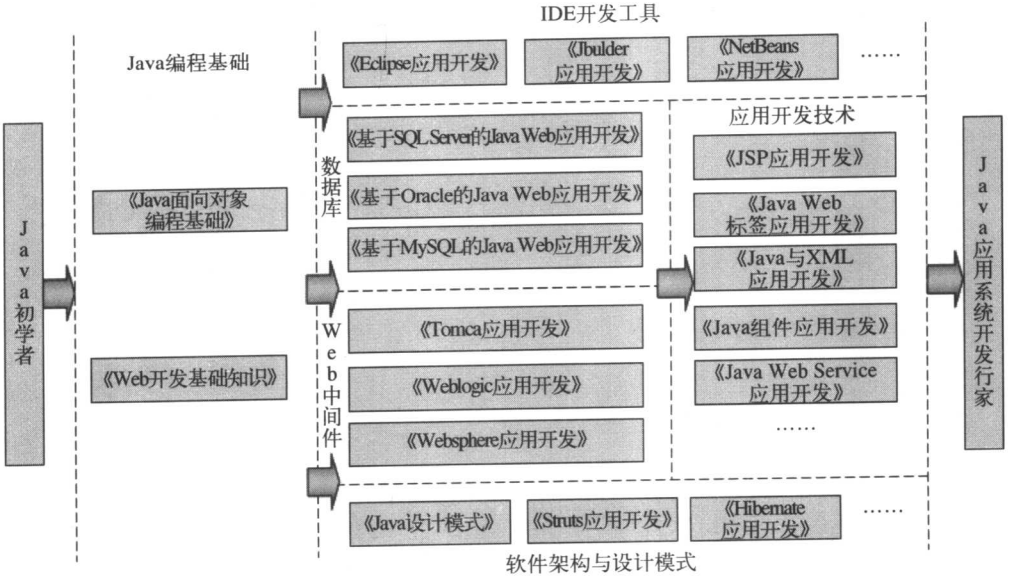
希赛IT技术讲堂（之Java篇）丛书将以清晰的思路，引导初学者在学习了一定的编程基础知识后，能掌握一种集成开发工具、一种数据库、一种Web开发中间件。当理解与运用常用的Web开发技术，熟悉一些架构与设计模式知识后，你将逐渐成长为一名经验丰富的开发人员。

对于初学者，希赛IT技术讲堂丛书给出了如下图所示的知识结构组合。



二、建议的学习路线

根据上述知识结构组合编写了系列丛书，丛书建议学习路线图如下。



Java初学者通过该学习线路中书籍的深入学习，可以逐渐成为一名Java应用系统的开发行家。有一定Java开发经验的人员也可以从中找到所需学习的知识。

丛书中将当前流行的集成开发工具、实践工程中使用的主流关系型数据库、常用的软件开发架构与设计模式推荐给读者，并做了详细地讲解。书中还附有精选的项目案例，以总结所学的知识点，并引领读者在接近真实的工程实战环境中演练，以快速积累开发经验。

丛书的学习线路中，读者可以根据自己的兴趣做一些组合。如在学习了基础知识后，对Eclipse、Tomcat、SQL Server、Struts、JSP感兴趣的读者，可以阅读如下的5本图书：《Eclipse应用开发》、《Tomcat应用开发》、《基于SQL Server的Java Web应用开发》、《Struts应用开发》和《JSP应用开发》。

这些图书中每一本的内容都保持了一定的独立性，注重知识的详细解析与运用，遵循了循序渐近的原则，以方便读者深入理解某一领域的知识并可运用到软件项目之中。

三、本丛书的作者

希赛顾问团拥有近500名顾问，其中有的是活跃在各行各业信息化工作一线的高级工程师、项目经理，有的是高校计算机专业的教学骨干，有的是IT公司的工程师、技术总监等。在这个高素质的团队中选择了一批工程实践经验丰富、有一定教学经验的人员来组织编写这套丛书。

为了更好地为读者服务，借助希赛网（http://www.csai.cn）的平台优势，作者还将在希赛网上开办技术论坛，进行答疑活动。

有关本系列丛书的意见反馈和咨询，读者可在希赛网社区（http://bbs.csai.cn）“书评在线”版块中的“机械工业出版社”栏目中与作者进行交流。

本系列丛书的配套光盘中的内容（如果有配套光盘的话），读者也可以在希赛网下载中心（http://data.csai.cn）下载。

编　者

2007年6月

前　言

Java是当前最流行的程序设计语言之一，它的出现大大地促进了软件产业和互联网的发展。Java之所以如此地流行是因为它是一种简单易学易用的、纯面向对象的、可移植的、安全的、高效的、健壮的、分布式的、多线程的、结构中立的、可解释执行的和动态的语言。Java拥有包括Sun、IBM在内的世界各大软件厂商的支持，因而发展迅速。经过10多年的发展，Java语言已从最初的Oak发展到了Java SE5。本书以Java SE5为基础，全面讲解了Java SE5的主要内容。但本书并不是纯粹地介绍Java SE5，还融合了程序设计人员必备的基础理论知识，如面向对象思想理论、UML、面向对象软件分析、设计、程序算法设计、数据结构等内容。读者学完本书后，可继续选读本系列丛书的其他图书，进而成为一名基础理论扎实、实践经验丰富的Java高级程序设计人才。

本书共分为17章，由浅入深地讲解Java SE5的主要内容和程序设计基础理论知识。

第1章　Java语言概述。主要讲解计算机语言以及Java语言的发展过程、Java语言的特点及Java平台体系结构、Java程序开发工具的安装及使用等内容。让读者对Java语言有一个全面的认识，为日后的学习打下基础。

第2章　Java程序概述。主要讲解Java语言的词法结构、Java程序的组成结构和Java程序设计、算法设计等内容。使读者对Java程序有一个总体的认识，同时掌握程序算法设计知识，为日后的程序设计工作打好一定的理论基础。

第3章　数据类型。主要讲解Java的数据类型及其用法。数据类型、运算符、流程控制语句是所有程序设计语言中最基础的内容，在Java中也不例外。读者必须熟练地掌握这些基础知识，并应用于程序设计中。

第4章　运算符。主要讲解Java的运算符及其用法。

第5章　流程控制语句。主要讲解了Java语言函数的基本控制结构及相关控制语句，包括顺序、条件选择、循环、转向、返回等内容。

第6章　面向对象基础。主要介绍面向对象思想理论知识和UML的用法，并结合面向对象思想理论、UML及实际项目案例，深入讲解了采用面向对象与UML技术进行系统分析和设计的过程。面向对象是Java程序语言的灵魂，UML是程序设计过程中常用的工具，读者必须深入领会并掌握它们。面向对象思想理论是比较抽象和难以理解的内容，本书通过将面向对象理论与现实相结合，引用了大量生活中常见的例子，使这部分内容浅显易懂。

第7章　面向对象Java实现。主要讲解Java语言中的类（class）、对象（object）、包（package）、继承（inheritance）、接口（interface）、枚举（enum）、对象类型转换、类的嵌套等内容。Java是纯面向对象的程序设计语言，通过它可实现面向对象的所有内容和特性。该章与第6章介绍的面向对象思想理论相对应，全面讲述了面向对象思想理论在Java中实现的方法。该章内容是Java SE5的重点和难点，通过学习该章内容，读者可以轻松地掌握、定义、实现并灵活地运用这些知识。

第8章　Java类库介绍。主要讲解Java API类库的结构层次，对常用的类库进行深入地介

绍。通过学习该章内容，读者可以深入地理解领会Java API的结构层次和常用类的用法，并可借助JDK帮助文档，灵活地运用JavaAPI类库中的其他类。

第9章　异常和断言。主要讲解Java的异常处理和断言机制、异常处理的各个组成部分、内置的异常类及自定义异常的方法。Java的异常处理和断言机制对于编写安全、健壮、简洁的Java程序具有重要的作用。通过学习该章内容，读者可以深入理解并掌握这些知识，并将其应用于Java应用程序中，以提高Java程序的健壮性。

第10章　多线程。主要讲解Java线程的基本概念和应用方法，包括线程的基本概念、线程实现的机制、线程的4种状态、线程的调度和优先级等内容。Java的多线程机制大大提高了Java程序的执行效率和交换性能。通过该章的学习，读者可以理解并掌握线程的概念，灵活应用多线程进行编程。

第11章　泛型。主要讲解Java的泛型机制及其用法。Java泛型是Java SE5中新增的功能特性，加强了Java程序的安全性和稳定性，并使Java程序代码更加清晰。通过该章的学习，读者可以深入理解Java的泛型机制和使用方法，并能将其应用于程序实践中。

第12章　Java虚拟机。主要讲解Java虚拟机的运行机制、指令系统和体系结构。Java虚拟机是Java平台的基石，所有的Java程序都执行于Java虚拟机之上。Java虚拟机实质是一种虚构的计算机，与普通计算机一样，拥有自己的虚拟处理器、堆、栈、寄存器等存储机制及相应的指令系统。通过该章的学习，读者可以深入理解Java平台的各种机制和特性，为编写出高质量的Java程序打好坚实的基础。

第13章　Java流与文件操作。主要讲解Java输入/输出操作的基本概念和应用，包括流的基本概念、字节码流、字符流、文件、对象序列化等。通过该章的学习，读者可以理解并熟练掌握Java流的基本概念和输入、输出操作，并将其应用于实际的程序开发中。

第14章　Java GUI编程。主要讲解JavaGUI编程的基本思想和方法，包括GUI编程概述、常用容器、常用GUI组件、布局管理、事件处理、高级GUI组件应用等内容。通过本章的学习，读者可以理解GUI编程的基本思想，GUI基本组件和容器之间的关系，借助JDK帮助文档编写所需的GUI程序。

第15章　Java小应用程序Applet。主要讲解Applet的基本结构、安全性、与Application的区别、Applet应用程序的开发等内容。Applet为静态的HTML页面提供了强大的表达和交互能力，同时又具有Java的跨平台等特性，在Internet应用中非常有优势。通过该章的学习，读者可以熟练地掌握与Applet相关的内容，并能够很好地编写Applet程序。

第16章　网络编程。主要讲解Java网络编程的基本概念和应用，包括网络编程基础知识、使用URL方式进行网络连接的方法、基于连接的流式套接字通信和基于无连接的数据报通信等内容。通过该章的学习，读者可以深入理解Socket的概念和Socket编程思想，掌握应用Socket和UDP进行编程的一般步骤，并能编写简单的网络应用程序。

第17章　数据库编程。主要讲解Java数据库编程的方法，包括数据库基本操作、JDBC的功能及用法等内容。此外还重点介绍JDBC的SQL预处理和存储过程、元数据的操作方法。通过该章的学习，读者可熟练掌握Java对数据库的编程方法，并应用于实际的软件开发中。

本书内容由浅入深，通俗易懂，并采用了大量的实例说明。本书可供具有一定计算机基础知识的读者学习现代程序设计技术。可作为高职、高专、本科院校或计算机培训机构的教材，也可作为计算机爱好者、程序员的自学教材或参考用书。

随书光盘包含了全书所有实例的源代码，以及项目案例的源代码，供读者学习参考使用，所有程序均已经过作者精心的调试。

本书由希赛顾问团顾问葛志春（系统分析员、信息系统项目管理师）主编，西北工业大学聂艳明博士、湖南铁道科技职业技术学院刘志成（高级工程师、系统分析员、希赛顾问团顾问）老师、冯向科老师参编。主要分工如下：葛志春编写了第1、2、3、4、5、6、7、12章；聂艳明博士编写了第8、9、11章，并对全书作了统稿、审核工作；刘志成老师编写了第10、13、14、16章；冯向科老师编写了第15和第17章。希赛网（http://www.csai.cn）扶文奇、周进、肖佳、王勇、史小琴、陈倩、谢顺、王冀等参与了书中的程序代码调试与文字校对工作。

感谢机械工业出版社的陈冀康编辑，他指导了本书的写作工作。感谢本书的主审、希赛顾问团顾问邓子云老师，他为本书的编写提出了许多指导性的意见，并组织审校了全部书稿。借此也向聂艳明博士的妻子商存慧老师和我的妻子童丽春女士致谢，是她们一直在默默地支持我们的工作。正是因为这么多人的大力支持，本书才得以出版。

由于时间仓促和作者的水平有限，书中的错误和不妥之处在所难免，敬请读者批评指正。

有关本书的意见反馈和咨询，读者可在希赛网社区（http://bbs.csai.cn）“书评在线”版块中的“机械工业出版社”栏目中与作者进行交流。

本书相关代码，读者可以在华章网站（http://www.hzbook.com）以及希赛网下载中心（http://data.csai.cn）下载。

葛志春

2007年8月于福州

目　录

第 1 章

Java语言概述

本章知识导学

本章主要介绍Java语言的发展历史、主要特点、Java平台家族，以及Java程序开发基础环境（即JDK和集成开发环境，如开源的Eclipse）。通过本章的学习，读者应该对照其他语言（如C或C++）深刻理解Java语言的特点和优势，并能够熟练地应用和配置Java开发与运行环境JDK，也能够在Eclipse环境下，简单地编辑、调试和运行Java程序代码。

1.1 计算机语言的发展过程

自1946年第一台电子计算机问世以来，计算机已被广泛地应用于生产、生活的各个领域，推动着社会的进步与发展。特别是Internet出现后，传统的信息收集、传输及交换方式发生了革命性的改变。

计算机科学的发展依赖于计算机硬件和软件技术的发展，硬件是计算机的躯体，软件是计算机的灵魂。没有软件，计算机只是一台“裸机”，什么也不能干；有了软件，计算机才有“思想”，才能做相应的事。软件是用计算机语言编写的。计算机语言的发展经历了从机器语言、汇编语言到高级语言的历程，如图1-1所示。

图1-1 计算机语言发展历程

1. 机器语言（Machine Language）

计算机使用的是由“0”和“1”组成的二进制数，二进制编码方式是计算机语言的基础。计算机发明之初，科学家只能用二进制数编制的指令控制计算机运行。每一条计算机指令均由一组“0”、“1”数字，按一定的规则排列组成，若要计算机执行一项简单的任务，需要编写大量的这种指令。这种有规则的二进制数组成的指令集，就是机器语言（也称为指令系统）。不同系列的CPU，具有不同的机器语言，如目前个人计算机中常用AMD公司的系列CPU和Intel公司的系列CPU，具有不同的机器语言。

机器语言是计算机唯一能识别并直接执行的语言，与汇编语言或高级语言相比，其执行效率高。但其可读性差，不易记忆；编写程序既难又繁，容易出错；程序调试和修改难度巨大，不容易掌握和使用。此外，因为机器语言直接依赖于中央处理器，所以用某种机器语言编写的程序只能在相应的计算机上执行，无法在其他型号的计算机上执行，也就是说，可移植性差。

2. 汇编语言（Assemble Language）

为了减轻使用机器语言编程的痛苦，20世纪50年代初，出现了汇编语言。汇编语言用比较容易识别、记忆的助记符替代特定的二进制串。下面是几条Intel80x86的汇编指令：

ADD AX，BX ；表示将寄存器AX 和 BX 中的内容相加，结果保存在寄存器AX中。

SUB AX , NUM ；表示将寄存器AX中的内容减去NUM，结果保存在寄存器AX中。

MOV AX , NUM ；表示把数NUM保存在寄存器AX中。

通过这种助记符，人们就能较容易地读懂程序，调试和维护也更方便了。但这些助记符号计算机无法识别，需要一个专门的程序将其翻译成机器语言，这种翻译程序被称为汇编程序。

汇编语言的一条汇编指令对应一条机器指令，与机器语言性质上是一样的，只是表示方式做了改进，其可移植性与机器语言一样不好。总之，汇编语言是符号化的机器语言，执行效率仍接近于机器语言，因此，汇编语言至今仍是一种常用的软件开发工具。

3. 高级语言

尽管汇编语言比机器语言方便，但汇编语言仍然具有许多不便之处，程序编写的效率远远不能满足需要。1954年，第一个高级语言——FORTRAN问世了。高级语言是一种用能表达各种意义的“词”和“数学公式”按一定的“语法规则”编写程序的语言，也称为高级程序设计语言或算法语言。半个多世纪以来，有几百种高级语言问世，影响较大、使用较普遍的有FORTRAN、A LGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、Visual C++、Visual Basic 、Delphi、Java等。高级语言的发展也经历了从早期语言到结构化程序设计语言、面向对象程序设计语言的过程。

高级语言与自然语言和数学表达式相当接近，不依赖于计算机型号，通用性较好。高级语言的使用，大大提高了程序编写的效率和程序的可读性。

与汇编语言一样，计算机无法直接识别和执行高级语言，必须翻译成等价的机器语言程序（称为目标程序）才能执行，如图1-2所示。高级语言源程序翻译成机器语言程序的方法有“解释”和“编译”两种。解释方法采用边解释边执行的方法，如早期的BASIC 语言即采用解释方法，在执行BASIC 源程序时，解释一条BASIC语句，执行一条语句。编译方法采用相应语言的编译程序，先把源程序编译成指定机型的机器语言目标程序，然后再把目标程序和各种标准库函数连接装配成完整的目标程序，在相应的机型上执行。如C、C++、Visual C++及Visual Basic等均采用编译的方法。编译方法比解释方法更具有效率。

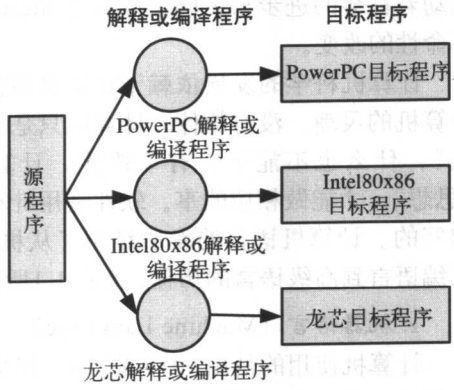


图1-2 高级语言程序翻译

4. 结构化程序设计（Structural Programming）语言

高级语言编写程序的编写效率虽然比汇编语言高，但随着计算机硬件技术的日益发展，人们对大型、复杂的软件需求量剧增，而同时因缺乏科学规范、系统规划与测试，程序含有过多错误而无法使用，甚至带来巨大损失。20世纪60年代中后期“软件危机”的爆发，使人们认识到大型程序的编制不同于小程序。“软件危机”的解决一方面需要对程序设计方法、程序的正确性和软件的可靠性等问题进行深入研究，另一方面需要对软件的编制、测试、维护和管理方法进行深入研究。

1968年，E.W.Dijkstra首先提出“GOTO语句有害”论点，引起了人们对程序设计方法讨论的普遍重视。程序设计方法学在这场讨论中逐渐产生和形成。程序设计方法学是一门讨论程序性质、设计理论和方法的学科。它包含的内容比较丰富，如：结构化程序设计、程序的

正确性证明、程序变换、程序的形式说明与推导以及自动程序设计等。

在程序设计方法学中，结构化程序设计占有重要的地位，可以说，程序设计方法学是在结构化程序设计的基础上逐步发展和完善的。结构化程序设计是一种程序设计的原则和方法。它讨论了如何避免使用GOTO语句；如何将大规模、复杂的流程图转换成一种标准的形式，使得它们能够用几种标准的控制结构（顺序、分支和循环）通过重复和嵌套来表示。结构化程序设计思想采用“自顶向下、逐步求精”的方法，避免被具体的细节所缠绕，降低难度，直到恰当的时机，才考虑实现的细节，从而有效地将复杂的程序系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。

按结构化程序设计的要求设计出的高级程序设计语言称为结构化程序设计语言。利用结构化程序设计语言，或者说按结构化程序设计思想编写出来的程序称为结构化程序。结构化程序具有结构清晰、容易理解、容易修改、容易验证等特点。

1970年，瑞士计算机科学家Niklaus.Wirth开发了第一个结构化程序设计语言——Pascal语言，标志着结构化程序设计时期的开始。Pascal语言的简洁明了以及丰富的数据结构，为程序员提供了极大的方便性与灵活性，同时它特别适合微计算机系统，因此大受欢迎，并迅速走红。结构化程序设计方法也在整个20世纪70年代的软件开发中占绝对统治地位。除Pascal语言外，常见的结构化程序设计语言还有C、FORTRAN、True BASIC等。

但是，到了20世纪70年代末期，随着计算机应用领域的不断扩大，对软件技术的要求越来越高，结构化程序设计语言和结构化程序设计方法又无法满足用户需求的变化了，其缺点也日益显露出来：

（1）代码的可重用性差。随着软件规模的逐渐庞大，代码重用成了提高程序设计效率的关键；但采用传统的结构化设计模式，程序员每进行一个新系统的开发，几乎都要从零开始，这中间需要做大量重复、繁琐的工作。

（2）可维护性差。结构化程序是由大量的过程（函数、子程序）组成的；随着软件规模逐渐庞大，程序变得越来越复杂，过程（函数、子程序）越来越多，相互间的耦合越来越高，它们变得难以管理；当某个业务有所变化时必须对大量的程序进行修改和调试。

（3）稳定性差。结构化程序要求模块独立，并通过过程（函数、子程序）的概念来实现。但这一概念狭隘、稳定性有限，在大型软件开发过程中，数据的不一致性问题仍然存在。

（4）难以实现。在结构化程序中，代码和数据是分离的，正如Niklaus.Wirth的定义：结构化程序＝算法＋数据结构。例如在C语言中，代码单位为函数，而数据单位称为结构，函数和结构没有结合在一起。然而，函数和数据结构并不能充分地模拟现实世界。人的思维焦点通常是在于事物和实体，以及它们的属性和活动，比如说当考虑会计部门的应用程序时，我们会考虑下列内容：

- 出纳支付工资；
- 职工出具凭证；
- 财务主管批准支付；
- 出纳记账。

但实际应用中，要决定如何通过数据结构、变量和函数来实现这个应用程序却是很困难的。

5. 面向对象（Object Oriented）语言的产生

结构化程序设计方法与语言是面向过程的，存在较多的缺点，同时程序的执行是流水线式的，在一个模块被执行完成前，不能干别的事，也无法动态地改变程序的执行方向。这和人们日常认识、处理事物的方式不一致。人们认为客观世界是由各种各样的对象（或称实体、