MIANXIANG DUIXIANG CHENGXU SHEJI JICHU

21世纪高等学校计算机科学与技术规划教材

面向对象程序设计基础

周会平　贾丽丽　王　挺　编著

北京邮电大学出版社

21世纪高等学校计算机科学与技术规划教材

面向对象程序设计基础

周会平　贾丽丽　王　挺　编著

UML explained
Design

北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书针对计算机专业教学的需求，介绍了面向对象程序设计的基础知识。

本书以 C++为实现语言，主要介绍了类与对象、类和对象的使用、操作符重载、继承、多态性、异常处理、模板、类库与软件重用等面向对象程序设计的基本概念和设计方法。本书主要介绍面向对象程序设计方法，学习本书需要具备 ANSI C 的基础，并了解高级语言程序设计的基本概念和方法。

本书是作者根据教学实践的经验编写而成，适合作为大学计算机专业和非计算机专业的面向对象程序设计课程的教材，也可供自学的读者使用。

图书在版编目(CIP)数据

面向对象程序设计基础/周会平，贾丽丽，王挺编著. —北京：北京邮电大学出版社，2004

ISBN 7－5635－0837－6

Ⅰ.面... Ⅱ.①周...②贾...③王... Ⅲ.面向对象语言—程序设计—高等学校—教材

Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2004)第 137879 号

书　　名　面向对象程序设计基础

编　　著　周会平　贾丽丽　王　挺

责任编辑　陈露晓

出版发行　北京邮电大学出版社

社　　址　北京市海淀区西土城路 10 号(100876)

电话传真　010－62282185(发行部)　010－62283578(传真)

E-mail　sanwen99@mail.edu.cn

经　　销　各地新华书店

印　　刷　国防科技大学印刷厂印刷

开　　本　787mm×960mm　1/16

印　　张　19.75

字　　数　340 千字

版　　次　2005 年 4 月第 1 版　2005 年 4 月第 1 次印刷

ISBN 7－5635－0837－6/TP・109　　定价　29.50 元

21世纪高等学校计算机科学与技术规划教材

编委会

序

自20世纪80年代以来，高等学校计算机教育发展迅速，计算机教育的内容不断扩展、程度不断加深。特别是近十年来，计算机向高度集成化、网络化和多媒体化发展的速度一日千里；社会信息化不断向纵深发展，各行各业的信息化进程不断加速；计算机应用技术与其他专业的教学、科研工作的结合更加紧密；各学科与以计算机技术为核心的信息技术的融合，促进了计算机学科的发展，各专业对学生的计算机应用能力也有更高和更加具体的要求。

基于近年来计算机学科的发展，以及国家教育部关于计算机基础教学改革的指导思路，我们确立了这套“21世纪高等学校计算机科学与技术规划教材”的编写思想与编写计划。教材是教学过程中的“一剧之本”，是高校计算机教学的首要问题。该套系列教材编写计划的制定凝聚了编委会和作者的心血，是大家多年来计算机学科教学和研究成果的体现，并得到了陈火旺院士的亲自指导与充分肯定。

这套系列教材由北京邮电大学出版社三文工作室精心策划和组织。编写过程中，充分考虑了计算机学科的发展与《计算机学科教学计划》中内容和模块的调整，使得整套教材更具科学性和实用性。整套系列教材体系结构按课程设置进行划分。每册教材均涵盖了相应课程教学大纲所要求的内容，既具备学科设置的合理性，又符合计算机学科发展的需要。从结构上遵循教学认知规律，基本上能够满足不同层次院校、不同教学计划的要求。

各册教材的作者均为多年来从事教学、研究的专家和学者，他们有丰富的教学实践经验，所编写的教材结构严谨、内容充实、层次清晰、概念准确、论理充分、理论联系实际、深入浅出、通俗易懂。

教材建设是一项长期艰巨的系统工程，尤其是计算机科学技术发展迅速、内容更新快，为使教材更新能跟上科学技术的发展，我们将密切关注计算机科学技术的发展新动向，以使我们的教材编写在内容上不断推陈出新、体系上不断发展完善，以适应高校计算机教学的需要。

21世纪高等学校计算机科学与技术规划教材编委会

2005年3月

前　言

C++是当今最流行的一种高效实用的高级程序设计语言，应用十分广泛。C++为面向对象的程序设计方法提供了强有力的支持，因而成为编程人员最广泛使用的工具。学好C++，很容易触类旁通其他程序设计语言，C++架起了通向强大、易用、真正的应用软件开发的桥梁。

本书阐述了C++面向对象程序设计概念、语法，融会贯通了程序设计方法、程序设计风格以及软件工程的思想。同时摒弃传统C++教材中将C与C++同时讲解的风格，将重点致力于解决C++中的难点，并介绍了一些必要的关于C语言的知识。可以帮助读者快速掌握C++面向对象程序语言，并成为一名真正意义上的程序设计开发人员，而不仅仅是一名将详细设计"翻译"为代码的工作者。

本书是编者在从事多年C与C++程序开发的基础之上，并结合丰富的一线教学经验编写而成的。其内容都是在教学过程中讨论，并在不断收集反馈意见的基础上修改和调整过的。同时在编写过程中我们力求形成如下特色：

1. 教学内容梯度适当。本书章节之间跨度较小，每次只将学习向前推进一小步。在教学讨论的内容基础之上，还加入了很多的信息，使读者顺利地从一个主题进入到另一个主题地学习，并保证在进入下个一主题前能较好地掌握前面已学过的内容。

2. 精选实例，针对性强。本书使用的例子典型易懂，能够更好地帮助读者理解和掌握C++的各个语法点及特性。同时对每个例子的代码长度也做了严格的控制，以方便教学。

3. 重点突出，讲述透彻。本书只介绍最重要的有助于理解C++的内容，删减了大量不常用或较为繁复的知识。便于读者快速顺利地把握C++知识的精华。

4. 注重实践能力的培养。本书提供了丰富的典型例题，并配有难易程度不同的习题，而且适当穿插介绍了一些编程技巧和软件设计经验，使读者能更好地学习和掌握C++程序设计方法。

本书由国防科技大学周会平、贾丽丽、王挺编著，适合作为高等学校面向对象程序设计课程的教材，也可做为程序设计开发人员的入门和提高的书籍。由于编者水平有限，加之时间仓促，书中难免存在缺点和错误，恳请广大读者批评指正。

编　者

2005年1月

目　录

第1章 程序设计基本概念

1.1 计算机系统概述

1.1.1 什么是计算机系统

计算机系统(computer system)是按人的要求接收和存储信息,自动进行数据处理和计算,并输出结果信息的机器系统。计算机是脑力的延伸和扩充,是近代科学的重大成就之一。计算机系统由硬件系统和软件系统组成。前者是借助电、磁、光、机械等原理构成的各种物理部件的有机组合,是系统赖以工作的实体,如CPU、显示器、内存、硬盘和键盘,等等;后者是各种程序和文档,用于指挥全系统按指定的要求进行工作,程序是对计算任务的处理对象和处理规则的描述,而文档是与软件研制、维护和使用相关的资料。随着计算机技术的迅速发展,硬件成本不断地下降,促进了个人计算机的广泛应用和普及。这一趋势也为计算机在信息处理方面提出了更高的要求——设计实现更方便、功能更加强大的应用程序。但是,相对于硬件技术的发展速度,软件的开发技术是明显滞后的。现阶段,软件设计更多地依赖于人的因素,而软件设计的方法、包括程序设计的方法,是提高软件生产率和质量的关键。

1.1.2 计算机硬件

计算机硬件系统是计算机系统快速、可靠、自动工作的基础。计算机硬件就其逻辑功能来说,主要是完成信息变换、信息存储、信息传送和信息处理等功能,它为计算机软件提供具体实现的基础。计算机硬件系统主要由运算器、主存储

器、控制器、输入输出设备、辅助存储等功能部件组成。

1. 运算器

运算器的主要功能是对数据进行算术运算和逻辑运算。整个运算过程是在控制器控制下自动进行的。操作时，运算器从主存储器取得运算数据，经过指令指定的运算处理，所得运算结果或者留在运算器内以备下次运算时使用，或者写入主存储器。

2. 主存储器

主存储器的主要功能是存储二进制信息。它与运算器、控制器等快速部件直接交换信息。从主存储器中应能快速读出信息，并送到其他功能部件中去，或将其他功能部件处理过的信息快速写入主存储器。

3. 控制器

控制器的功能主要是按照机器代码程序的要求，控制计算机各功能部件协调一致地动作，即从主存储器取出程序中的指令，并对该指令进行分析和解释，然后向其他功能部件发出执行该指令所需要的各种时序控制信号，然后再从主存储器取出下一条指令执行，如此连续运行下去，直到程序执行完为止。计算机自动工作的过程就是逐条执行程序中指令的过程。控制器与运算器一起构成中央处理器，中央处理器与主存储器一起构成处理机。

4. 输入设备

输入设备的功能主要是将用户信息(数据、程序等)变换为计算机能识别和处理的信息形式。输入设备种类很多，如纸带阅读机、软磁盘机、汉字输入设备、键盘输入设备等。它们的工作特点是将人工编制的程序和原始数据，在某种媒介物上以二进制编码形式来表现，如纸带上穿孔和不穿孔分别表示“1”和“0”；磁表面上磁化方向不同以区别“1”和“0”等。载有信息的媒介物通过相应的输入设备，将信息变换成电信号为计算机接收，并存入存储器。

5. 输出设备

输出设备主要是将计算机中二进制信息变换为用户所需要并能识别的信息形式。输出设备种类很多，如打印机、凿孔输出机、汉字输出设备、绘图仪、显示终端等。它们的工作特点与输入设备正好相反，是将计算机中二进制信息经过相应变换，成为用户需要的信息形式，记录在媒介物上或显示出来供用户使用。输出的信息形式多为十进制数字、字符、图形、表格等。

6. 辅助存储器

辅助存储器主要是存储主存储器难以容纳、又为程序执行所需要的大量文件信息。它的特点是存储容量很大，存储成本很低，但存取速度较慢。它不能直接与中央处理器交换信息。辅助存储器一般为磁盘机、磁带机和光盘机等。

1.1.3　计算机软件

光有硬件计算机还不能工作，要使计算机能解决各种实际问题，还必须有软件的支持。软件是用户与硬件之间的接口界面。使用计算机就必须针对待解的问题拟定算法，用计算机所能识别的语言对有关的数据和算法进行描述，即编写程序。用户主要是通过软件与计算机进行交互。软件是计算机系统中的指挥者，它规定计算机系统的工作，包括各项计算任务内部的工作内容和工作流程，以及各项任务之间的调度和协调。软件是计算机系统结构设计的重要依据。为了方便用户，在设计计算机系统时，必须全面考虑软件与硬件的结合，以及用户的要求和软件的要求。

计算机软件通常分为三大类：系统软件、支撑软件和应用软件。

1. 系统软件

系统软件居于计算机系统中最靠近硬件的一层。其他软件一般都通过系统软件发挥作用。它与具体的应用领域无关，如编译程序和操作系统等。操作系统(operating system)负责管理系统的各种资源、控制程序的执行。语言处理程序(language processor)包括汇编程序与各种高级语言的解释程序和编译程序，其任务是将使用汇编语言或高级语言编写的源程序翻译成能被计算机硬件直接识别和执行的机器语言。没有语言处理程序的支持，用户采用汇编语言和高级语言编写的应用软件就无法被计算机识别。在任何计算机系统的设计中，系统软件都要优先予以考虑。

2. 支撑软件

支撑软件是用于支撑其他软件的开发和维护的软件。随着计算机科学技术的发展，软件的开发和维护代价在整个计算机系统中所占的比重很大，远远超过硬件。因此，支撑软件的研究具有重要意义，直接促进软件的发展。当然，编译程序、操作系统等系统软件也可算作支撑软件。但是，70年代中期和后期发展起来的软件支援环境可看成为现代支撑软件的代表，主要包括环境数据库、各种接口软件和工具组。三者形成一个整体，协同支援其他软件的开发。

3. 应用软件

应用软件是特定应用领域专用的软件。例如，人口普查用的软件就是一种应用软件。对于具体的应用领域，应用软件的质量往往成为影响实际效果的决定性因素。70 年代出现的嵌入式应用，其相应软件的复杂程度高，开发工作量大，促进了软件的发展。模拟应用导致模拟语言(SIMULA)的出现。应用软件的作用越来越大。

上述分类不是绝对的，而是互相交叉和变化的。有些软件如编译程序和操作系统，既可看作是系统软件，又可看作是支撑软件。它们在一个系统中是系统软件，而在另一个系统中却是支撑软件；也可以在同一系统中既是系统软件，又是支撑软件。系统软件和应用软件之间也有类似情况。有的软件如数据库管理系统、网络软件和图形软件，原来算作应用软件，后来又被看作为系统软件。而且系统软件、支撑软件和应用软件三者的开发技术基本相同。因此，这三者既有分工，又有结合，并不截然分开。

1.2 程序设计基本概念

1.2.1 问题求解过程

在计算机中，一切信息处理都要受程序的控制，数值数据是如此，非数值数据也是如此。因此任何问题求解(problem solving)最终要通过执行程序来完成。根据传统的(面向过程)程序设计方法，把计算机的应用需求转变为可在计算机上运行的程序，一般要经历问题定义、算法设计、程序编码、程序测试等步骤。

1. 问题定义

问题定义(problem definition)的目的是明确要解决的问题，写出求解问题的“规格说明”(specifications)。其内容应包括：用户要求的输入/输出的数据及其形式、求解问题的数学模型或对数据处理的需求、程序的运行环境等。在软件的开发过程中，需求分析完成的就是问题定义，即明确拟开发的软件的功能需求。

2. 算法设计

算法设计(algorithm design)是指把问题的数学模型或处理需求转化为计算机的解题步骤。算法设计的好坏直接影响着程序的质量。对于较大型软件的开发来说,设计是一个更为复杂和更加重要的阶段,通常还进一步分为概要设计和详细设计两个阶段。其中在概要设计阶段,主要是根据软件需求规格说明建立目标软件系统的总体结构、设计全局数据结构,规定设计约束,制定组装测试计划等等;而在详细设计阶段,主要是逐步细化概要设计所生成的各个模块,并详细描述程序模块的内部细节(数据结构、算法、工作流程等等),形成可编程的程序模块。

3. 程序编码

程序编码(coding)的主要任务,是用选定的某种程序设计语言将前一步设计出来的算法实现为能在计算机上运行的程序。在软件开发过程中,编码的工作是严格根据详细设计规格说明而进行的,所以,软件的设计应当尽可能做到详细、正确和完整。

4. 测试和调试

测试和调试(testing and debugging)的主要目的在于发现(通过测试)和纠正(通过调试)程序中的错误。只有经测试合格的程序,才能交付用户使用。在软件开发过程中,通过对编写的程序进行调试和测试,以验证程序与详细设计文档的一致性,从而确保程序实现需求规格说明规定功能,即解决了所定义的问题。

1.2.2　算法与程序

1. 算法

算法是求解问题类的机械的、统一的方法,它由有限多个步骤组成,对于问题类中的每个给定的具体问题,机械地执行这些步骤就可以得到问题的解答。算法的这种特性,使得计算不仅可以由人,而且可以由计算机来完成。用计算机解决问题的过程可以分成三个阶段:分析问题、设计算法和实现算法。让计算机解决某一个问题之前,必须先对问题进行分析,提出解决问题的办法,然后建立此问题的计算步骤,最后在计算机上进行实现。

算法具有下列特征:

(1) 有穷性:一个算法在执行有穷个计算步骤后必须终止。

(2) 确定性:一个算法给出的每一个计算步骤必须是精确定义的、无二义性的。

(3) 能行性:算法中要执行的每一个计算步骤都可以在有限时间内做完。

(4) 输入:要求输入一个或多个输入信息,有的算法可能不要求输入,这些输入取自某一特定集合。

(5) 输出:一个算法一般有一个或多个输出信息。

算法与计算机没有必然的关系,人们可以用多种方法来描述算法。主要有以下三种:

(1)文字描述。

文字描述即用自然语言(汉语、英语等)描述算法。采取这种描述方法,可以使得算法易读易理解。例如,下面是求解两个整数整商的算法的文字描述:

① 读入两个整数,即被除数和除数;

② 如果除数等于 0,则输出除数为 0 的错误信息;

③ 否则,计算被除数和除数的整商,并输出计算结果。

(2)图形描述。

我们可以采取图形描述的方法来描述算法,主要包括流程图(也称为框图)、盒图(也称为 N-S 图),PAD 图等。在这里主要介绍流程图这种描述方法。流程图是对算法逻辑顺序的图形描述。如用长方形表示计算公式,用菱形框表示条件判断等等。此方法形象、清晰、画法简单、格式自由,可以不涉及太多的机器细节或程序细节,其主要弱点是计算机很难直接识别。

流程图采用一些图框表示各种操作,形象直观,易于理解。ANSI(American National Standard Institute, 美国国家标准化协会)规定了一些常用的流程图符号,已为世界各国程序工作者普遍采用。流程图也有不同的表示形式。主要的流程图符号如图 1-1 所示:

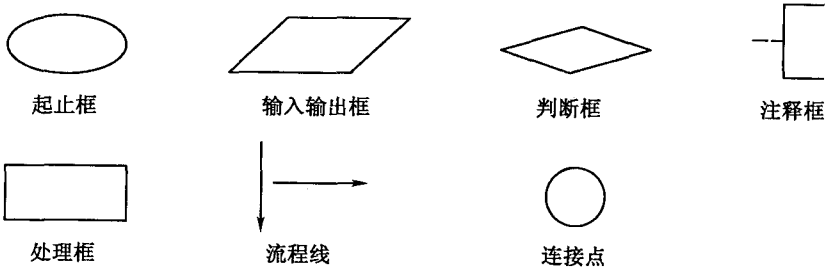

图 1-1 流程图符号