

悦觀垣巨程 序 设 计

主 编 柴 欣

副 主 编 韩 卫

编 委 武 优 西 史 巧 硕

张 春 英 阿 建 卓

河 北 大 学 出 版 社

河北省普通高校计算机基础教育 教材编审委员会

主任委员 杨建广

副主任委员 李凤翊 魏世泽 崔来堂

委员 鲍继宏 王晨光 刘明生 董爱堂
薛晓萍 柴欣 王兴达

内容简介

本书是学习 悦垣言 语言程序设计的基础教程,全书较为系统地讲述了 悦垣言 语言的基础知识、基本规则及编程方法,在此基础上,对 悦垣言 的面向对象的重要特征如类、封装、继承、多态、~~陈~~流操作等进行讲解。

为了加强基础、注重实践,在内容讲解上采用循序渐进、逐步深入的方法,突出重点,注意将难点分开,使读者易学易懂。

本书可作为大专院校各专业程序设计的正式教材,又可作为研究生计算机基础教育的教材,也可借计算机培训班和读者自学使用。

前 摇 摇 言

悦垣是一种面向对象的程序设计语言,它是在悦语言的基础上发展起来的,几乎包括了悦语言的全部功能,并克服的悦语言的某些不足,是目前使用非常广泛的面向对象的程序设计语言。本书的作者长期从事悦垣语言程序设计课的教学工作,并利用悦垣开发了多个软件项目,因此有着丰富的教学经验和较强的科研能力,对悦垣有着较深入的理解。为了使初学程序设计的读者能够掌握悦垣程序设计语言的使用方法,并初步具备使用悦垣程序设计语言开发应用程序和解决实际问题的能力,作者精选了悦垣的内容,本着加强基础、注重实践、勇于创新、突出应用的原则,力求使本教材达到可读性、适用性和先进性的良好融合。为了便于读者自学,在全书的体系结构和内容上注意了由浅入深、深入浅出、循序渐进的方针。为了提高读者编程技巧,在大部分章节中都提供了典型例题。

本书从软件设计思想出发,将全书分为两个部分,第一部分介绍悦垣的结构化程序设计基础知识,并融和了结构化设计方法。在该部分中,较为系统地讲述了悦垣语言的基础知识、基本规则及编程方法,其中第猿章介绍了程序设计的发展,讲解了结构化程序设计思想和面向对象程序设计思想,还通过一个简单的例子,介绍了悦垣的基本术语和概念,并对悦垣程序的调试和运行环境进行了介绍。第圆-远章讲述了悦垣语言的基本内容,包括:常量、变量、运算符和表达式、各种语句、函数和储类、指针和引用、结构体和联合体等,这些内容也是构成悦垣程序的基础。

第二部分由第苑-员章组成,重点介绍悦垣的面向对象的基本思想及面向对象的设计方法。在这部分中,较系统地讲述了悦垣语言中面向对象的主要特征,如封装、继承、多态、流操作等,这些都是悦垣的核心内容,体现了悦垣语言面向对象的特点。

由于目前的计算机操作系统多为宰圣图形界面平台,因此本书在灾环境下进行过调试和运行。

教师在选用本书作为大学生软件技术基础课程的教材时,可根据实际授课时数取舍其中的章节。由于授课时数的限制,教师可在规定授课时数内重点讲授第一部分的内容,而在后续的选修课程或研究生课程中介绍第二部分的内容,这样可使学生完整地了解悦垣的内容。因此,本书适用于本、专科各专业及硕士研究生的计算机基础教育。

本书由柴欣主编,并负责全书的总体策划与统稿、定稿工作,韩卫任副主编。各章编写分工如下:第猿章由柴欣编写,第圆章由史巧硕编写,第缘章由武优西编写,第苑-员章由韩卫编写。参加本书大纲讨论及部分编写工作的老师还有张春英、阿建卓等。

本书作为河北省统编教材,在编写之初即得到编审委员会多位专家的大力支持,在编写过程中也一直得到各位专家的热心指导与无私帮助,编者在此一并表示衷心的感谢。此外在本书写作时,还参考了大量文献资料,在此也向这些文献资料的作者深表感谢。

由于时间仓促和水平所限,书中难免有不当和欠妥之处,敬请各位专家、读者不吝批评指正。

编 摇 者

圆年 员月

目 录

第 1 章 绪论	(1)
1.1 计算机语言发展概述	(1)
1.2 语言的词法和语法规则	(7)
1.3 简单的 C 程序	(9)
1.4 程序的调试与运行	(10)
第 2 章 数据类型及表达式	(12)
2.1 基本数据类型	(12)
2.2 常量与变量	(14)
2.3 数组	(16)
2.4 运算符与表达式	(18)
2.5 类型转换	(20)
2.6 类型定义	(22)
习 题	(23)
第 3 章 结构化程序设计	(24)
3.1 C 的输入输出流	(24)
3.2 顺序结构及程序举例	(26)
3.3 选择结构的实现	(28)
3.4 循环结构的实现	(30)
3.5 程序设计举例	(32)
习 题	(33)
第 4 章 指针与引用	(34)
4.1 指针	(34)
4.2 引用	(36)
4.3 内存管理	(38)
习 题	(39)
第 5 章 函数与预处理	(40)
5.1 函数的定义	(40)
5.2 函数调用及参数传递机制	(42)
5.3 指针与函数	(44)

第 缘章 摇函数的嵌套调用和递归调用	(缘园)
第 缘章 摇内联函数和重载函数	(缘远)
第 缘章 摇作用域	(缘怨)
第 缘章 摇变量的存储类型	(缘缘)
第 缘章 摇函数的存储类型	(缘贞)
第 缘章 摇编译预处理	(缘源)
第 缘章 摇程序举例	(缘原)
摇习题	(缘园)
第 远章 摇结构体、联合体、枚举类型	(远猿)
第 远章 摇结构体与结构变量	(远猿)
第 远章 摇结构成员的引用	(远愿)
第 远章 摇结构数组	(远贞)
第 远章 摇结构指针	(远猿)
第 远章 摇联合体	(远缘)
第 远章 摇枚举类型	(远愿)
摇习题	(远贞)
第 苑章 摇面向对象设计	(苑缘)
第 苑章 摇面向对象方法的提出	(苑缘)
第 苑章 摇面向对象的主要概念	(苑园)
第 苑章 摇面向对象程序设计语言	(苑园)
第 苑章 摇面向对象分析与设计	(苑贞)
第 苑章 摇面向对象分析与设计举例	(苑贞)
摇习题	(苑园)
第 愿章 摇类与对象	(愿猿)
第 愿章 摇类的构造与封装	(愿猿)
第 愿章 摇类与对象	(愿缘)
第 愿章 摇友元	(愿猿)
第 愿章 摇类模板	(愿缘)
第 愿章 摇程序举例	(愿愿)
摇习题	(愿园)
第 怨章 摇继承与派生	(怨源)
第 怨章 摇派生类	(怨源)
第 怨章 摇派生类的构造函数和析构函数	(怨远)
第 怨章 摇虚函数	(怨园)
第 怨章 摇抽象基类	(怨园)

猿猿猿猿程序举例	(猿猿)
猿猿猿猿	(猿猿)
第猿猿章猿猿运算符重载	(猿猿)
猿猿猿猿重载运算符	(猿猿)
猿猿猿猿类型转换与转换函数	(猿猿)
猿猿猿猿对象赋值与赋值运算符重载	(猿猿)
猿猿猿猿下标运算符与函数调用运算符重载	(猿猿)
猿猿猿猿运算符重载规则	(猿猿)
猿猿猿猿	(猿猿)
第猿猿章猿猿输入输出流	(猿猿)
猿猿猿猿标准输入输出流	(猿猿)
猿猿猿猿解决输入输出流问题	(猿猿)
猿猿猿猿文件输入输出流	(猿猿)
猿猿猿猿	(猿猿)
参考文献	(猿猿)

第 1 章 绪论

1.1 计算机语言发展概述

1.1.1 计算机程序设计语言的发展

计算机之所以能自动进行计算,是因为采用了程序存储的原理,计算机的工作体现为执行程序。程序是控制计算机完成特定功能的一组有序指令的集合,编写程序所使用的语言称为程序设计语言,它是人与计算机之间进行信息交流的工具。从 1946 年世界上诞生第一台计算机起,在短短的 60 余年间,计算机技术迅速发展,程序设计语言的发展从低级到高级,经历了机器语言、汇编语言、高级语言到面向对象语言的多个阶段,具体过程如下。

1.1.1.1 机器语言

计算机能够直接识别和执行的二进制指令(也称机器指令)的集合称为该种计算机的机器语言。早期的计算机程序都是直接使用机器语言编写的,这种语言使用 0 和 1 代码,因此编写出的程序难以理解和记忆,目前已不被人们使用。

1.1.1.2 汇编语言

通过助记符代替 0 和 1 机器指令以利于理解和记忆,由此形成了汇编语言。汇编语言实际上是与机器语言相对应的语言,只是在表示方法上采用了便于记忆的助记符号来代替机器语言相对应的二进制指令代码,因此也称为符号语言。计算机不能直接识别汇编语言,需要编译后才能识别。这种语言的执行效率较高,但由于难以记忆,因此使用较少。

1.1.1.3 高级语言

机器语言和汇编语言是面向机器的语言,高级语言采用更接近自然语言的命令或语句,使用高级语言编程,一般不必了解计算机的指令系统和硬件结构,只需掌握解题方法和高级语言的语法规则,就可以编写程序。高级语言在设计程序时着眼于问题域中的过程,因此它是一种面向过程的语言。对于高级语言,人们更容易理解和记忆,这也给编程带来很大方便,但它与自然语言还是有较大差别的。

1.1.1.4 面向对象语言

面向对象语言是比面向过程语言更高级的一种高级语言。面向对象语言的出现改变了编程者的思维方式,使设计程序的出发点由着眼于问题域中的过程转向着眼于问题域中的对象及其相互关系,这种转变更加符合人们对客观事物的认识。因此,面向对象语言更接近于自然语言,是人们对于客观事物更高层次的抽象。

目前,世界上已经设计和实现的计算机语言有上千种之多,但实际被人们广泛使用的不过数十种。

1.1.2 程序设计的发展历程

回顾程序设计发展的历史,大体上可以划分为如下几个不同的时期。

20 世纪 50 年代的程序都是用指令代码或汇编语言来编写的,这种程序的设计相当麻

烦 编写和调试一个稍大一点的程序常常要花费很长的时间 培养一个熟练的程序员更需经过长期的训练和实习 这种局面严重影响了计算机的普及与应用。

20世纪 50年代高级语言的出现大大简化了程序设计 缩短了解题周期 因此显示出强大的生命力。此后 编制程序已不再是软件专业人员才能做的事了 一般工程技术人员花上较短的时间学习 也可以使用计算机解题。这个时期 随着计算机的应用日益广泛地渗透到各学科和技术领域 也发展了一系列不同风格的、为不同对象服务的程序设计语言。其中较为著名的有 BASIC、FORTRAN、ALGOL、COBOL、PASCAL、LISP、PROLOG 等十几种语言。高级语言的蓬勃兴起 使得编译和形式语言理论相应日趋完善 这是该时期的主要特征。但就整个程序设计方法而言 并无实质性的改进。

自 20世纪 50年代末到 60年代初 出现了大型软件系统 如操作系统、数据库 这给程序设计带来了新的问题。大型系统的研制需要花费大量的资金和人力 可是研制出来的产品却是可靠性差 错误多 且不易维护和修改。一个大型操作系统有时需要几千人年的工作量 而所获得的系统又常常会隐藏着几百甚至几千个错误。当时 人们称这种现象为“软件危机”。

“软件危机”震动了软件行业 程序设计的传统习惯和工作方式导致了不清晰的程序结构 使得程序的可靠性难以保障。另一方面 程序设计工具的严重缺乏也使得大型系统的开发陷入困境。此时人们开始重新审视程序设计中的一些最基本的问题。例如 程序的基本组成部分是什么 应该用什么样的方法来设计程序 如何保证程序设计正确 程序设计的主要方法和技术应如何规范等等。

1969年 尼克劳斯·梅特罗提出了结构化程序设计的概念 强调了从程序结构和风格上来研究程序设计。经过几年的探索和实践 结构化程序设计的应用确实取得了成效 用结构化程序设计的方法编写出来的程序不仅结构良好 易写易读 而且其正确性易于证明。

20世纪 60年代末结构化设计方法得到了很大的发展 詹姆斯·尼古拉·沃西提出了“算法 数据结构 越程序设计”的程序设计方法 他将软件划分成若干个可单独命名和编址的部分 它们被称为模块 模块化使软件能够有效地被管理和维护 能够有效地分解和处理复杂问题。到 70年代 模块化程序设计方法普遍被人们接受。

虽然几十年来结构化程序设计技术得到了广泛的使用 但有些问题仍未得到很好的解决。由于软件开发是对问题的求解过程 从认识论角度看 软件开发过程包括人们对要解决问题及相关事物的认识和基于认识所进行的描述。而结构化设计方法不能直接反映出人类认识问题的过程。另外 结构化设计方法中 程序模块和数据结构是松散地耦合在一起的 因此 当应用程序比较复杂时 容易出错 难以维护。随着计算机软件的发展 软件系统越来越复杂、庞大 结构化程序设计方法已显得力不从心。

在 20世纪 80年代 人们又提出了面向对象的程序设计方法。这种方法直接对问题域中的客观事物建造分析模型中的对象 使对象的描述与客观事物一致 保持问题域中的单个事物及事物之间的关系的原貌 而面向对象的语言可以直接描述问题域中的对象及其相互关系。用面向对象的程序设计方法 可以使人们对复杂系统的认识过程与系统的程序设计与实现过程尽可能地一致 这是因为它从客观世界中所存在的事物出发 比较符合人们的思维方式。因此 目前这种面向对象的程序设计模式正逐渐取代“数据结构 算法”的面向过程的程序设计模式。

面向对象程序设计概述

面向对象方法的起源

结构化程序设计技术虽已使用了几十年,但如下问题仍未得到很好的解决。

首先,面向过程的设计方法与人们习惯的思维方法仍然存在一定的差距,所以很难自然、准确地反映真实世界。因而用此方法开发出来的软件,有时很难保证其质量,甚至需要进行重新开发。

其次,结构化程序设计在方法实现中只突出了实现功能的操作方法(模块),而被操作的数据(变量)处于实现功能的从属地位,即程序模块和数据结构是松散地耦合在一起的。因此当应用程序比较复杂时,容易出错,难以维护。

由于上述缺陷,结构化程序设计方法已不能满足现代化软件开发的要求,一种全新的软件开发技术应运而生,这就是面向对象的程序设计(即面向对象编程,简称OOP)。

20世纪70年代,在软件开发中各种概念和方法积累的基础上,就如何超越程序的复杂性障碍,如何在计算机系统中自然地表示客观世界等问题,人们提出了面向对象的程序设计方法。面向对象的方法不再将问题分解为过程,而是将问题分解为对象。对象将自己的属性和方法封装成一个整体,供程序设计者使用。对象之间的相互作用则通过消息传递来实现。用面向对象的程序设计方法,可以使人们对复杂系统的认识过程与系统的程序设计与实现过程尽可能地一致。目前,这种“对象+消息”的面向对象的程序设计模式正逐渐取代“数据结构+算法”的面向过程的程序设计模式。

面向对象语言的发展

早在20世纪70年代,就出现了最早的面向对象语言——Smalltalk语言,具有了类和对象的概念。随后又推出了纯面向对象设计语言,如80年代美国卡内基梅隆大学研究中心推出了C++,它完整地体现并进一步丰富了面向对象的概念,还开发了配套的工具环境,为其最终实用化奠定基础。但由于当时人们已经接受并广泛应用结构化设计理论,而不能一下子完全接受面向对象程序设计的所有思想等诸多原因,这些语言并没有能够广泛流行起来。后来人们开始对流行的语言进行面向对象的扩充,曾经推出过许多版本的版本,而主要成功的代表是在当时流行的程序设计语言——C语言的基础上开发的C++语言。这时面向对象语言已形成几大类别:一类是纯面向对象的语言,如C++和Java;另一类是混合型的面向对象语言,如C#和Python;还有一类是与人工智能语言结合形成的,如Prolog和Lisp以及适合网络应用的Perl语言等。

C++语言是由AT&T公司贝尔实验室的Bjarne Stroustrup博士开发的,它的创作灵感来源于计算机语言多方面成果的凝聚,特别是C语言、Pascal语言、ALGOL语言、Lisp语言、Modula-2语言(以面向对象为核心的语言)和Smalltalk(以面向对象为核心的语言),同时也借鉴了Basic语言。C++的名字是由Richard Stalman提出的,到1989年被确定。

C++是一门高效、实用的混合型程序设计语言,它包括两部分内容:一是C++基础部分,它是以C语言为核心的。另一部分是C++面向对象特性部分,是C++对C语言的扩充部分。这样它既支持面向对象程序设计方法,又支持结构化程序设计方法。同时,广泛的应用基础和丰富的开发环境的支持,也使面向对象设计得到很快普及。

猿说坦语言对面向对象方法的支持

悦坦语言是悦语言的扩展,其基础部分除了一些细微的差别外,可以说是悦语言的超集,它保留了悦语言功能强、效率高、风格简洁、适合于大多数的系统程序设计任务等优点,使得悦坦与悦之间取得了兼容,因此,在过去的软件开发中积累的大量悦的库函数和实用程序都可在悦坦中应用。

另外,悦坦语言通过对悦的扩充,克服了原有悦语言的缺点,特别是引进了面向对象语言的要素,使得悦坦语言成为一种面向对象的程序设计语言,它对面向对象程序设计方法的支持体现在如下几个方面。

(1) 支持数据封装

在悦坦语言中,类是支持数据封装的工具,对象是数据封装的实现。在封装中还提供了一种对数据访问的控制机制,使得有些数据被隐藏在封装体内,因此具有隐蔽性。封装体与外界进行信息交换是通过操作接口进行的。这种访问控制机制体现在类的成员中可以有公有成员、私有成员和保护成员。

(2) 支持继承性

悦坦语言允许单继承和多继承。继承是面向对象语言的重要特性。一个类可以根据需要生成它的派生类,派生类还可以再生成它的派生类。派生类继承了基类成员,另外它还可以定义自己的成员。继承是实现抽象和共享的一种机制。

(3) 支持多态性

悦坦语言支持多态性表现在:

① 悦坦语言允许函数重载和运算符重载;

② 悦坦语言可以定义虚函数,通过它来支持动态联编。动态联编是多态性的一个重要特征。

猿说坦语言开发环境的发展

随着悦坦语言逐渐成为业界标准,这种新的面向对象程序设计语言迅速成为程序员最广泛使用的工具。众多悦坦语言的开发环境也不断地推出,竞争十分激烈。猿年,月集世公司开发了猿说坦,而后又推出了月集世悦坦版本。配公司于年代中期在配说坦的基础上开发了配悦坦,同时引进了配库,完善了源代码。以前这些版本都是依赖于环境,或在宰圣增下的模式运行。不久,配公司推出的配悦坦,即宰圣增版本,它是配公司推出的第一个真正的基于宰圣增环境下的可视化的集成开发环境,它将编辑、编译、链接和执行集成为一体。从宰圣增版本以后,配公司决定不再将更多的努力花的支持猿位编程上,虽然宰圣增悦坦仍提供对猿位的支持,但从圆版本以后,宰圣增悦坦更多地用来创建猿位程序。在版本上,配公司没有推出猿版本,版本号直接从圆跳到源,这样,宰圣增悦坦与配的版本号取得一致。由于配流行明显地影响了产品的设计,在源版本中,宰圣增悦坦引进了为配编程而设计的新类库。缘版本也增加了一些新类,但注意力更多地集中在改善产品的界面上,以提供更好的在线帮助系统、更高级的宏能力和对在开发者组内进行类和其他代码共享的支持。远版本在功能上有了进一步的改进。宰圣增悦坦经历远

会导致语法错误。关于上述关键字的含义,在本书中会逐渐讲述。

表 员原圆京瑶悦垣巨关键字表

葬燥	遭燥	糟藻	糟则	糟贼	糟贼藻
凿藻贼	凿燥	凿燥藻	藻藻	藻皂	藻藻址
枣藻贼	枣则	早燥	蚤	蚤贼	蚤早
泽藻则	泽藻址	泽藻贼	泽藻崖	泽藻枣	泽藻糟
泽藻贼	泽藻燥	贼藻藻藻	怨藻址	怨泽藻崖	增藻
增藻藻	憎藻藻	遭藻造	糟燥燥	糟藻泽	糟燥贼 糟藻贼
凿藻藻藻	凿藻藻皂 凿藻藻 糟藻藻贼	藻藻藻藻藻贼	枣藻藻	枣藻藻崖	蚤藻藻藻
皂燥藻藻	灶燥藻藻 灶燥藻	灶燥	燥燥燥燥则	责燥燥燥	责燥燥藻藻崖
责藻藻	贼燥燥燥燥藻 糟藻藻 泽燥燥藻 糟藻藻贼	贼燥藻藻	贼燥藻藻藻	贼燥	贼燥增
贼藻	贼增	贼燥藻崖	贼燥藻藻藻	怎藻早	增燥藻造
憎燥则 贼					

猿运算符

运算符是一些用来进行某种操作的单词,它实际上是系统预定义的函数名。这些函数作用于被操作的对象上将获得一个结果值。运算符是由一个或多个字符组成的单词。

悦垣巨语言的运算符除了包含了悦语言中的运算符外,还增加了一些新的运算符。悦垣巨语言的运算符可以重载。

源分隔符

分隔符被称为程序中的标点符号,它是用来分隔单词与程序正文的,表示某个程序实体的结束和另一个程序实体的开始。

悦垣巨常用的分隔符有以下几个:

- ①空格:用作单词之间的分隔符。
- ②逗号:用作变量之间或对象之间的分隔符,或用作函数的多个参数之间的分隔符。
- ③冒号:用作语句标号与语句间的分隔符以及泽藻藻语句中糟藻藻约整数型表达式跃与语句序列之间的分隔符。
- ④花括号:用来为函数体、复合语句等定界。

缘常量

悦垣巨语言中,常量有数字常量、字符常量和字符串常量。程序中的常量经常用符号常量来表示。关键字糟燥贼用来定义各种不同类型的常量。

远注释符

在程序中注释起到对程序的注解和说明的作用,目的是为了便于程序的阅读和分析。

