

21世纪高等学校规划教材 | 软件工程

建模视角下的 面向对象程序设计

刘鹏远 温 珏 孙宝林 主编
崔洪芳 曾长军 副主编



清华大学出版社

21世纪高等学校规划教材 | 软件工程

A decorative illustration of a plant with several small, five-petaled flowers and long, curved, swirling stems, located in the upper right quadrant of the cover.

建模视角下的 面向对象程序设计

刘鹏远 温 珏 孙宝林 主 编
崔洪芳 曾长军 副主编

清华大学出版社
北京

内 容 简 介

本书是一本集 C++ 语言高级特性和面向对象思想于一身的中级技术指南。在涵盖了 C++ 语言的主要特点(封装、继承、多态)之余,从软件建模的视角出发,引入了针对抽象编程、聚合优先于继承、低耦合、高内聚等面向对象思想的讲述。此外,对困惑程序员已久的程序依赖问题做了深入分析;给出了异步消息通信的原理及实现;引入了架构分析,对循环依赖和 MVC 模式的原理实现及缺陷改良也做了深入论述。本书有别于一般介绍 C++ 语言的书籍,涉及了一些底层原理和编译知识的挖掘理解;不同于纯粹介绍软件模式等面向对象设计方面的书籍,在内容设计上注重由浅入深,实例指导;结合面向对象思想,对各种软件基础模式的原理思想和实现也有介绍。

本书可作为高等院校相关专业高年级本科生、研究生的教材,也可作为软件开发领域工程师的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

建模视角下的面向对象程序设计/刘鹏远,温珏,孙宝林主编. —北京:清华大学出版社,2014

21 世纪高等学校规划教材·软件工程

ISBN 978-7-302-38324-6

I. ①建… II. ①刘… ②温… ③孙… III. ①面向对象语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 236755 号

责任编辑:刘 星 赵晓宁

封面设计:傅瑞学

责任校对:李建庄

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:12.5

字 数:310 千字

版 次:2014 年 11 月第 1 版

印 次:2014 年 11 月第 1 次印刷

印 数:1~2000

定 价:29.00 元

本书编委会

主 编 刘鹏远 温 珏 孙宝林

副主编 崔洪芳 曾长军

委 员 桂 超 胡汉武 李 祥

包 琼 邓沌华 刘 坤

关培超 陈 婕 尤川川

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn



序

所有从事计算机行业的人都离不开 C/C++ 语言的学习,但那种艰难历程相信不少人都终生难忘。虽然国内外的经典 C/C++ 技术书籍流行甚广、读者众多,如《深度探索 C++ 对象模型》、《C++ Effective》、《Thinking in C++》、《C++ Primer》等,但适合在校大学生和入门读者的不多,适合中级程序员水平的也不多,特别是将软件建模的基础模式与编程语言结合起来贯彻面向对象思维进行讲述更为少见。

欣慰地看到本书作者们的辛勤努力。当展开目录时,就有一种不同寻常的感觉。该书关于语言基本要素部分的讲述不足 5 页,但对于类对象、类间关系则花费了 140 页之多来深入阐述,这对于一本首先致用于教学用书来说是一种大胆尝试和创新。

这是一本适合中高级程序员学习和系统回顾掌握复杂特性的技术参考书。能够看到作者们为减缓读者学习坡度做了精心设计,在章节内“巧借联想”多处搭桥;在阐述引用前先指针阐述透彻加以分析对比;介绍软件复用时把面向对象思想原则中最基础和重要的内容通过详例阐明;讲述对象构造、析构时对内存机制分析清晰;讲述两类多态时又将预编译处理、编译绑定、运行时绑定有效结合;在论述应需而变时,从函数指针动态切换、通用指针等 C 语言的一些向高级阶段发展的趋势入手,渐进引入到 C++ 动态多态以不变应万变。诸如此类,可见作者们的精心考量和智慧。

卢炎生

武汉工商学院副校长

原华中科技大学计算机学院副院长

2014 年 7 月

前言

为什么要写这样一本 C++ 技术书籍

本书最开始的想法很简单,即为定制班的学生服务。本书第一作者在 IT 行业从业近十年,是软通动力信息技术(集团)公司的高级系统分析师,同时也是一名高校教师。2010 年 5 月软通动力公司与笔者所在高校签订了 C++ 软件工程师定制班培养协议,从那时起,就有了系列教材出版计划,其中《面向对象 UML 系统分析建模》于 2013 年 9 月已由清华大学出版社出版。

从实际教学和工程指导所需来看,为何不能够集中课时将关注点集中在 C++ 的中高级特性上,少言甚至滤去有关输入输出、基本语言特性部分;为何不能将两类多态的原理讲透彻,让学生理解多态中蕴含的软件复用的发展;为何不能结合软件设计模式让学生理解拥抱变化的内涵;为何不能结合面向对象思想精髓让学生在初始编程时就养成测试驱动和 MVC 分层、针对抽象不针对具象的基本素养?每一个 IT 人,都有着技术的狂热和理想,自己不例外更希望能适当总结一些单纯、适用的东西给这些年轻的 IT 从业者,希望他们能在商业项目忙碌之余,手头上能有一本给予适当启发的参考书。

自 C++ 语言诞生以来,面向对象程序设计成为主流,但 C++ 语言学习的高难度使得它在程序员市场在 21 世纪初面临着以 Java 语言为代表的第二代面向对象程序设计语言的强烈竞争。自 Visual Basic、Delphi 等客户前端开发工具出现后,C/C++ 语言逐步退出桌面端开发工具市场,而退守深植根于嵌入式和后台交换控制领域。1998 年 Java 语言诞生,以其前后台通吃地全面性席卷全球,并迅速迎来 Web 开发时代,软件迎来 C/S 模式向 B/S 模式的深刻变革。一时间,在桌面开发工具市场上,Java 的 AWT/SWING 所向披靡,摧枯拉朽般消灭了 Visual Basic、Delphi、PowerBuilder 等开发工具;在 Web 开发工具市场,J2EE 的 JSP+Servlet 的 Model2 模型迅速成为市场主流压倒 ASP 框架。更主要的是,Java 语言一改程序员严格的内存分配、初始化、释放自律负责的态度,程序员不再需要关注内存细节,极大降低了编程难度;Java 首创虚拟机即时编译,使得字节码的编译结果得以在不同虚拟机平台上再次动态编译执行成为可能,从根本上解决了应用程序的跨 OS 平台移植难题——此举从根本上将程序员繁重的移植任务交给了 Java 语言本身,即由 JDK 的拥有者负责不同版本虚拟机的支持解释。但不论 Java 如何强势,在 TIOBE 编程语言排行榜独占鳌头近十年,但在企业后台交换市场和一些严格强调安全性、并发性、效率性能的行业市场,C/C++ 以其专业性地位牢不可破。纵观 20 世纪 80 年代以来的编程语言市场,C/C++ 语言该体系遥遥领先于其他编程语言,即便 Java 语言最强大时期也从未占据 20% 以上份额;最近七八年来,随着移动互联和手游的兴起,C++ 以其杰出的底层控制和语言效率王者归来,与 C 语言、Java 语言一起牢牢占据着 TIOBE 编程语言排行榜的三甲。

在程序员初入行市场薪资水平上,C/C++程序员工资一般高于Java程序员,近年来对C/C++程序员的追逐饥渴热度直线上升,且与Java程序员工资差距呈现不断扩大趋势。作为企业任职的高级系统分析师和高校教师双重身份的我们,不禁疑惑,究竟是怎么了,那些20世纪80~90年代大学生们学习C/C++的狂热都到哪里去了?

问题还是出在Java身上,它把程序员惯坏了:它让身为程序员应具有的严格、规范和全局性思维不再。越来越多的人认识到,学生必须掌握基础,必须牢固地理解内存和OS机制,而不能什么都交给虚拟机。数据结构、算法课程,更应植根于C/C++本身特性,而并非各种STL、JDK的教学上。

另外,Java语言和J2EE框架的流行又极大地普及了设计模式和软件建模的应用。那么反过来,是否可以将这种模式建构系统的观点应用在C++语言中的教学中,让从业者、初学者能从浩如烟海的C++知识点中汲取到那最宝贵的百分之十面向对象思想?笔者教学和实施项目之余,常常想着这样的念头如何付诸实施。2013年9月,经过3年认真删减留下的精简版的《面向对象UML系统分析建模》终于出版了,该书有别于传统讲述UML和RUP模型的工具书,反而更似一本将建模与设计模式结合的技术专著,得到众多褒扬之余,也下定决心再写一本C++与建模结合的书,让学习C++语言的人能够感受到面向对象之美,体验设计模式之灵活神奇。

书的主要内容有哪些

全书共8章,第1和第2章扼要回顾C语言并介绍C++语言的主要特点(封装、继承、多态),对功能分解和逐步分层求精做了深入阐述,对软件复用做了归纳,对容易混淆的变量/函数声明、变量/函数定义做了清晰的区分,对多文件工程常见的头文件重复包含问题给出了原理上的分析和解决方法;第3章将指针和引用一起做了深入的分析比对;第4章对类和对象做了全面的阐述(类结构、类的分析识别过程、对象初始化、析构);第5章全面深入地阐述了两类多态原理,对函数间的重载、覆盖、隐藏关系做了精确深入的剖析,并引入了针对抽象编程这一关键面向对象思想;第6章对类间的4种关系(继承、聚合、关联、依赖)做了清晰描述,结合实例引入了聚合优先于继承以及低耦合高内聚等面向对象思想的讲述,对困惑C++程序员的程序依赖问题做了深入挖掘分析,最后详尽阐述了消息通信机制,给出了异步消息通信的原理及实现;第7章对内存泄漏、运算符重载、友元、抽象类以及virtual“三虚”做了全面细致的分析阐述;第8章对循环依赖问题、架构分析以及MVC原理实现与缺陷改良做了深入论述,最后引入了一个综合应用了针对抽象编程、依赖倒置、聚合优先于继承等面向对象思想的实例。

谁需要本书

学习和了解了一些C++语言特性,但却始终无法真正迈进面向对象大门的程序员可能需要本书;绝大多数在面向对象领域里刻苦攻读、努力实践、却迟迟不能看到美好回报的程序员可能需要本书;一些准备学习设计模式、UML软件建模、软件重构和软件体系结构知识的中高级软件工程师可能需要本书。

分工与答谢

笔者都是从业 IT 多年的软件工程师、项目经理或高校教师。孙宝林、崔洪芳和曾长军负责编写第 1 章；桂超、胡汉武和李祥负责编写第 2 章；关培超、刘坤和邓沌华负责编写第 3 章；包琼、陈婕和尤川川负责编写第 7 章；刘鹏远和温珏负责其余章节的编写及全书统稿。

在本书编写过程中，要特别感谢戴志锋和曾宇容等人，是他们在长期教学实践中给予真知灼见的指导。要向王虹致以崇高敬意，本书体例编写、文字组织得到了他的精心指导；要向薛吉宝、蒋国银等领导致谢，是他们的坚强领导和有力组织使得该书从签订合同到付梓出版仅耗时半年多时间；要向软通动力信息技术(集团)的陈友华、李江波、苑永超、汪亚军、袁盐成等同事们致意，感谢他们的宝贵意见和建议，使该书更能适应 IT 企业员工的所急所想，更具有实用性；最后，还要感谢华中科技大学的陈传波、云南大学的李彤，以及南湖地区计算机联盟、湖北省高校计算机学会联盟等单位的支持，没有他们的不吝赐教与帮助，本书难以如此顺利的出版。

反馈

本书作者是普通的程序员，也是高校教师，水平有限，书中错误在所难免。欢迎同行和读者就本书的内容、文字、体例不吝赐教。笔者的电子邮箱为 waynewendy@126.com。

编 者

2014 年 9 月

目 录

第 1 章 绪论	1
1.1 结构化程序设计的特点	1
1.1.1 水平功能分解	1
1.1.2 垂直逐步分层求精	2
1.2 C++的特点	3
1.2.1 全面兼容 C	3
1.2.2 强类型检查	12
1.2.3 封装	13
1.2.4 继承	14
1.2.5 软件复用	15
1.2.6 多态	16
第 2 章 语言基本要素	19
2.1 基本编程语句	19
2.2 运算与表达式	21
2.3 声明与定义	23
2.3.1 变量定义	23
2.3.2 变量声明	24
2.3.3 函数定义	25
2.3.4 函数声明	25
2.3.5 类声明与定义	26
第 3 章 指针与引用	29
3.1 值传递	29
3.1.1 赋值兼容性检查	29
3.1.2 值传递时机	31
3.2 指针	32
3.2.1 定义	32
3.2.2 值	32
3.2.3 型与值	33
3.2.4 值传递	34
3.3 多重指针	36

3.4	void * 指针	36
3.4.1	定义	37
3.4.2	用途	37
3.5	函数指针	39
3.6	const 修饰符	40
3.7	数组	42
3.7.1	一维数组	43
3.7.2	多维数组	45
3.7.3	数组指针	46
3.7.4	字符串常量与字符数组	47
3.7.5	越界检查	48
3.8	引用	49
3.8.1	定义	49
3.8.2	引用传递	51
3.8.3	引用的意义	54
第4章	类与对象	55
4.1	类与结构体	55
4.2	类的分析识别过程	56
4.3	访问控制符	61
4.3.1	类内/外部与授权访问	62
4.3.2	public 访问属性	64
4.3.3	private 访问属性	64
4.3.4	protected 访问属性	65
4.4	静态变量	67
4.4.1	静态、全局、局部	67
4.4.2	类的静态成员	69
4.5	初始化与构造	70
4.5.1	变量声明、定义与初始化	70
4.5.2	单类构造	71
4.5.3	继承构造	79
4.5.4	成员初始化表	81
4.6	析构	83
4.6.1	内存区域	84
4.6.2	生命期与作用域	85
4.6.3	内存分配/销毁	85
4.6.4	单类析构	86
4.6.5	继承结构析构	90
4.6.6	组合析构	91

4.6.7 虚析构	97
第 5 章 多态	100
5.1 静态多态	100
5.1.1 重载	100
5.1.2 模板	101
5.2 动态多态	105
5.3 函数间关系	109
5.3.1 重载	110
5.3.2 覆盖	110
5.3.3 隐藏	111
5.4 针对抽象编程	114
第 6 章 类间关系	118
6.1 继承关系	118
6.1.1 IS-A 的软件复用含义	118
6.1.2 继承改写子类成员访问控制符	119
6.1.3 继承带来的麻烦与问题	123
6.1.4 开闭原则	130
6.2 使用关系	132
6.2.1 关联	133
6.2.2 聚集	134
6.2.3 依赖	135
6.2.4 聚合优先于继承	136
6.3 程序依赖问题	137
6.3.1 向前引用	137
6.3.2 弱类型依赖	139
6.3.3 双向依赖	141
6.4 低耦合与高内聚	142
6.4.1 低耦合模式	142
6.4.2 高内聚模式	144
6.5 消息通信机制	145
6.5.1 同异步消息范畴	145
6.5.2 多进程间异步消息通信机制	146
6.5.3 单宿主机单进程内异步消息通信应用范畴	151
第 7 章 杂项	153
7.1 内存泄漏	153
7.1.1 第一类内存泄漏	153

7.1.2	第二类内存泄漏	153
7.2	运算符重载	155
7.2.1	普通运算符重载	155
7.2.2	成员运算符重载	160
7.2.3	转换构造与运算符重载	161
7.3	友元	164
7.3.1	友元函数	164
7.3.2	友元类	167
7.4	抽象类与纯抽象类	168
7.4.1	抽象类	168
7.4.2	纯抽象类	169
7.5	virtual“三虚”	171
第8章	高级面向对象思想	172
8.1	循环依赖问题	172
8.2	架构分析	172
8.3	MVC 模式	177
8.3.1	模式设计目的	178
8.3.2	模式基本结构	179
8.3.3	模式的不足	180
8.4	应需而变实例	180
参考文献		184

第 1 章

绪论

本章将简要介绍 C++ 语言,其间涉及一些 C 语言特性的归纳概括。

本章重点:

多文件工程;

水平分解与垂直分层;

多态。

难点:多态。

1.1 结构化程序设计的特点

结构化,即为“功能分解、逐步求精”。如果将结构化编程的思想引入软件建模领域,那么功能分解,即子系统划分的原则;逐步求精,即分层的原则。上述两个过程,一般都是先进行水平方向上的功能分解,再对某一功能点进一步进行垂直方向上的逐步求精。

1.1.1 水平功能分解

功能分解即水平地划分功能点集合,使得能够分而治之地逐步解决大的问题。

不可能单独用一个模型来反映整个系统的任何侧面。软件系统是复杂的,对于软件模型的任意一个侧面不可能用一个模型来反映所有内容,需要把问题分解为不同的子模型,分别处理这些模型,相对独立但又互相联系,综合起来构成了此侧面的一个完整的模型。从方法论的角度来说,当一个系统比较大时,采用分治的策略将其划分为较小规模的多个系统,从而容易分别去思考分析解决问题的方法,即运用子系统划分的思想方法。

以学生管理系统为例,初步分析认为在一个大学校园中,学生管理系统应具有以下功能:

- 选课管理;
- 成绩浏览;
- 课程网上评教;
- 班级管理;
- 班级 BBS。

又以模拟 QQ 的短消息通信软件为例,可以初步做水平划分认为有以下功能:

- 接收阅读消息功能;

- 发送点对点消息功能；
- 增、改联系人简介功能；
- 群发消息功能；
- 群邮件功能。

1.1.2 垂直逐步分层求精

逐步求精：垂直划分问题聚焦功能点，使得对功能点的认识从表面深入到里层，完成对陌生事物从抽象到具体的认识过程。

从结构化程序设计的“逐步求精”来看，对一个问题的解决分析是逐步深入的。随着不断深入，软件系统也在衍生划分出新的层次，每个层次只关注于本层应着力解决的相关问题。以内聚性来看，每层是一个功能相对独立的内聚体。

面对复杂事物，需要由表及里。例如，对模拟 QQ 的短消息通信软件，需要对 1.1.1 节列出的所有功能点进一步做逐步求精，即垂直分层划分。为节约篇幅，这里选取点对点短消息发送这个功能点，显然，至少可纵向分为两层：

- (1) 提供输入和输出显示的正文框。
- (2) 后台消息发送处理。

对于前台的正文框已经分析足够清楚，但对于第二层还需要进一步求精。

- 需要解析得出该头像联系人活跃的 QQ 进程以便连接；
- 需要解析得出该头像联系人活跃的 QQ 进程所在计算机的 IP 地址以便通信；
- 需要解析得出该头像联系人活跃的 QQ 进程所在计算机的网卡 MAC 地址以便底层数据链路通信。

按照上述理解，将该功能以分层的形式对应于 TCP/IP 分层模型，如图 1-1 所示。



图 1-1 TCP/IP 模型图对应于相应分层形式

小结:

子系统划分是将复杂事物分而治之的水平划分方法。分层是逐步求精的纵向划分方法,是人们观测和思考世界的基本原则和方法,也是结构化程序设计的基本思想。

需要指出的是,不能以功能分解作为最开始的分析建模步骤,否则不同模块可能产生很强耦合去操作同一个数据,参见 8.2 节。

1.2 C++的特点

C++(C PLUS PLUS)名字本身就说明它是 C 语言的超集,即继承和保持了 C 语言的一切特性以保证全面的兼容性。总体来看,C++ 语言新增了一系列重要的特性:如为了改善编译器的“宽容”,C++ 将比 C 更严格执行编译类型检查(强类型检查);为减少无序和不安全的函数间耦合,C++ 将全局平面化的函数组织到类这样一个域中,通过给予数据封装,使得函数间错综复杂不受限的耦合改良为相关函数的在类内部的数据耦合、功能耦合;为提高软件复用性,提供了继承这样一个强有力的机制来模拟现实世界中普遍存在的一般-特殊关系;为提供软件模型的灵活性,提供的两类多态机制等。

1.2.1 全面兼容 C

1. 函数驱动

无论 C 或 C++ 程序,都以函数驱动的形式执行,即从一个函数的内部的执行流程,到因为函数调用跳转到被调用函数内部执行另一个流程。当被调函数执行结束时,需要返回结果并回至函数调用发生处下一句继续原来的处理。那么当发生函数调用时,系统执行点从调用函数跳转到被调函数内部之前,需要先保存现场方便将来继续调用函数流程处理(该过程也称为保存断点,包括保存实参、返回地址)。

由于函数调用经常是嵌套式的,即可能 $a \rightarrow b \rightarrow c$ 的形式,那么返回时应该是从 $c \rightarrow b' \rightarrow a'$ 的形式。 b' 表示 b 调用 c 函数的 b 内的下一句, a' 表示 a 调用 b 函数的 a 内的下一句。从上面的保存和返回次序看出这是一个典型的后进先出次序(先保存 a ,再 b ,再 c ,先返回 c ,再 b ,再 a),因此 OS 使用系统堆栈来保存函数调用发生时的断点。

每当进入一个函数,就要在系统堆栈中分配该函数体中的形参变量和局部变量的内存(自右向左)。一旦调用某函数,立即压栈保存返回地址,然后控制转移到被调函数内部。

进入被调函数内部,在系统堆栈中继续分配形参变量和局部变量内存,当遇到 `return` 语句或函数结尾的“`}`”时,处理结果立即保存在 `eax` 寄存器中(若有返回值),并 `pop` 销毁分配的形参和局部变量,最后 `pop` 的双字单元就是前面保存的返回地址,系统回到原来的被调函数继续后面的语句执行。

下面的程序段为例:

```
void main()  
{  
    L1: int x = 0;  
    L2: int y = f(x);  
    L3: while( ... )
```