

21 世纪高校规划教材

Delphi & SQL Server 管理系统程序设计

主编 / 陈志延



江西高校出版社

21 世纪高校规划教材

Delphi & SQL Server 管理系统程序设计

主 编 陈志延
副主编 廖志斌 徐照兴
叶晓燕 谢中梅

江西高校出版社

图书在版编目(CIP)数据

Delphi & SQL Server 管理系统程序设计/陈志延主编.
—南昌:江西高校出版社, 2008.8

ISBN 978 - 7 - 81132 - 285 - 9

I .D... II .陈... III .①软件工具 - 程序设计 ②
关系数据库 - 数据库管理系统, SQL Server - 程序设计
IV .TP311.56 TP311.138

中国版本图书馆 CIP 数据核字(2008) 第 119707 号

出版发行	江西高校出版社
社 址	江西省南昌市洪都北大道 96 号
邮 政 编 码	330046
总编室电话	(0791)8504319
销 售 电 话	(0791)8508191
网 址	www.juacp.com
印 刷	江西龙莹印务有限公司
照 排	江西太元科技有限公司照排部
经 销	各地新华书店
开 本	787mm × 1092mm 1/16
印 张	13.125
字 数	319 千字
版 次	2008 年 8 月第 1 版第 1 次印刷
印 数	1 ~ 2500 册
书 号	ISBN 978 - 7 - 81132 - 285 - 9
定 价	22.60 元

版权所有 侵权必究

前 言

Delphi 是目前创建 Windows 应用程序最强大的开发工具之一。它具有 C++ 的强大功能和 Visual Basic 的易用性。友好的开发环境还能够让用户更快、更准确地编写代码。它以可视化的开发环境、简洁明快的编程语言、功能强大的组件、优化的码编辑器、可扩充的数据库访问引擎、强大的多媒体编程功能、丰富的 Internet 应用编程,成为一个目前广泛流行、功能成熟的产品,赢得了越来越多的用户。2002 年 Borland 推出的 Delphi 7 Studio 版本虽然不是最新版本,但却是到目前为止最成功的版本。Delphi 7 Studio 引入了下一代“模型驱动开发 (Model Driven Development)”科技以大幅强化开发生产力。

SQL Server 2000 是微软公司推出的新一代企业级网络数据库产品,是目前世界上大型数据库系统之一,它可以通过对高端硬件平台以及最新网络和存储技术的支持,为大型的 Web 站点和企业级的应用提供可扩展性和高可靠性。

长期以来,编者在进行程序设计和数据库应用的教学过程中,发现这两门课刚上时,学生大多是有兴趣的,但随着课程的深入,学生却越来越难以理解,渐渐地失去了热情,教学效果也越来越差,结果是课程虽然上完了,但学生却没有能力进行甚至是一些简单的数据库管理系统的开发与设计,没有达到教学的起初目的。分析其原因,我们认为有以下几个方面:

(1)现在的“数据库应用”课程,一般都是以 Access、SQL Server 等关系型数据库为主进行讲述的,这种基于二维表格形式的数据库简单易学,因此学生对于数据库表的建立、修改以及一些简单的数据查询容易理解和学习,上手很快,但是对于其他的知识,如视图、存储过程、触发器等数据库对象及一些比较复杂的查询(如多表连接查询、子查询及复合查询等高级查询),由于没有实际应用的基础与经验就难以理解了,特别是对为什么要用、什么时候使用这些数据库对象更是无法明白,因此不论这门课讲了多少,学生也只学了一点点皮毛。

(2)程序设计语言,现在一般以 Visual C、Visual B、Delphi、Java 等为主。可视化程序设计语言极大地减轻了程序设计人员的工作量,使程序的界面设计成为一种轻松的艺术设计工作,但是可视化程序设计依赖于大量的组件进行,内容繁多,在很多的可视化程序设计语言教材中,关于组件的介绍往往占了整个篇幅的 1/5~1/3 以上,这么多的内容,如果没有相当的实践知识与反复训练,真的很难记住,难怪学生学了后面的就忘了前面的,结果学生除了对一些界面的设计比较感兴趣和比较容易理解外,而对于完成特定功能的程序设计,特别是对于一个完整的管理系统的设计就一筹莫展、无从下手了。

本书一改传统的教学方法,采用现在比较流行的“项目教学法”,以学生比较熟悉的“学生成绩管理系统”的一部分为教学主线,结合由美国的神经病学教授 Barrows 在加拿大的 McMaster 大学在 1969 年首创的 PBL (Problem Based Learning) 教学法进行论述,其基本做法是:在练习中学习、在做项目中练习,有关联的内容讲,重要的内容反复练习,难点、重要的内容由浅入深地讲。根据编者多年的实践,证明教学效果良好。



本书有些基础知识参考了一些教材的内容,特别是摘录了网上的一些介绍,在此谨表衷心的感谢。

编 者

2008 年 6 月



目 录

第 1 章 Delphi 7 基础知识	1
1.1 Delphi 7 简介	1
1.2 Delphi 7 的特点与功能	3
1.3 对象的基本概念	6
1.4 Delphi 7 集成开发环境的使用	10
第 2 章 SQL Server 2000 数据库基本知识	16
2.1 SQL Server 2000 简介	16
2.2 SQL Server 2000 的安装	18
2.3 企业管理器、查询分析器和联机丛书	27
第 3 章 学生成绩管理系统程序设计	32
3.1 系统需求分析	32
3.2 系统功能模块设计	34
3.3 主窗体设计	35
第 4 章 学生基本信息管理设计	40
4.1 多文档子窗体设计	40
4.2 数据库的设计方法	43
4.3 学生基本信息数据库表的设计	50
4.4 学生基本信息输入界面设计	54
4.5 学生基本信息保存功能设计	57
4.6 学生基本信息查询设计	71
第 5 章 教师基本信息管理设计	73
5.1 教师基本信息内容分析	73
5.2 教师基本信息数据库表的设计	73
5.3 教师基本信息输入界面设计	74
5.4 教师基本信息保存功能设计	80
5.5 教师基本信息查询设计	88
第 6 章 代码管理	90
6.1 窗体整体界面设计	90
6.2 民族代码管理页面设计与功能实现	92
6.3 Delphi 的异常处理	95
6.4 政治面貌代码管理页面设计与功能实现	97
6.5 课程代码管理页面设计与功能实现	101
6.6 通用查询功能的实现	103
第 7 章 教学计划管理模块设计	114



7.1	教学计划信息内容分析	114
7.2	教学计划信息数据库表设计	115
7.3	视图的应用	116
7.4	教学计划信息输入界面设计	120
7.5	教学计划信息查询设计	131
第 8 章	学生成绩输入与查询设计	134
8.1	学生成绩信息内容分析	134
8.2	学生成绩数据库表的设计	135
8.3	学生成绩输入设计	137
8.4	学生成绩保存功能设计	143
8.5	学生成绩查询设计	148
第 9 章	报表设计	155
9.1	初识 Rave	155
9.2	Rave 组件的介绍	156
9.3	设计简单的报表	161
9.4	建立数据报表	164
9.5	预览报表	167
附录 A	Object Pascal 简介	169
附录 B	Delphi 常用函数	186
附录 C	VCL 组件的通用属性、方法和事件	201



第1章 Delphi 7 基础知识

学习目标

- 了解 Delphi 7 的基础知识
- 掌握 Delphi 7 的使用方法
- 熟悉 Delphi 7 的应用环境

1.1 Delphi 7 简介

“初级程序员用 VB,聪明的程序员用 Delphi,真正的程序员用 VC。”这是网上流行的一句说法,是针对程序员进行开发工具选择时的一种概述,这句话不知是谁先说的,但现在网上却很流行,随便到百度或 Google 上一搜,就有很多的条目。

我用这句话作为开场白,首先是要证明大家选择学习 Delphi 是一个正确的选择,希望大家下定决心、沉下心思把 Delphi 学好。当然,大家也可以进行认真地思考,选择一个自己最喜欢、也最符合自己的开发工具。

不过我们认为这句话确实很好地概括了 Delphi 的优点:即易学易用又具有进一步开发的能力。使用 VB 的朋友可别泄气,喜欢 VB 的朋友也别气恼,其实现在的 VB 功能也很强大,而且有微软作后盾,一定会有好的发展。我们也使用了很长时间,现在也一直在使用 VB,不过更多的时间却给了 Delphi,特别是在编写数据库信息管理系统时,Delphi 更是表现出其强大的优势。同庞大的 Visual C++ 相比较,Delphi 简直是个精灵,不但编译速度世界一流,而且到目前为止,几乎还没有 VC 能完成,而 Delphi 完成不了的功能。

1.1.1 Delphi 的发展

要介绍 Delphi 就必须先说说大名鼎鼎的 Borland(宝兰)公司(现在已改名为 Inprise 公司)。Borland 公司的创始人 Philippe Kahn 是持旅游签证和 2000 美元从法国到美国硅谷开办了这个公司,开发出了 Turbo Pascal(当时的售价是 49.95 美元),有意思的是,他当时曾经想为他的公司取名 MIT,麻省理工学院当然不同意,于是就用了 Borland 这个名字。

在 DOS 流行的时代,Borland 的 Turbo 家族:Turbo C、Turbo C++、Turbo Pascal、Turbo Assembler 等对于当时的编程人员是多么的熟悉,Turbo 家族的最大特点是将编辑、编译、调试都统一到集成调试环境(IDE)中,无需先用字处理器编辑好源码再退到 DOS 提示符下进行编译、连接,深得编程人员的喜爱。

随着 Windows 时代的到来,Windows 最早的开发系统只有 Windows SDK(Software Developer Kit),Borland 也很快在其产品中(BC3.0)增加了对 Windows 的支持,提出了 OWL 类库,大大简化了 Windows 编程的难度,但是大多数 DOS 程序员仍然被复杂的 Windows API 函数及使用



的繁琐所困惑,更多的恐怕还是由于对 Windows 的许多新概念不能理解而最终放弃。这时微软率先提出了可视化编程的新思想,并且在 Visual Basic 上实现,很多程序员纷纷转向了 VB。

Borland 为了夺回市场,潜心开发下一代编程平台,Delphi 1.0 的内部项目名称被称为“VB Killer”(VB 杀手)可见其决心,虽然 Delphi 1.0 问世时(1995 年),VB 又升级了一个版本,但 Delphi 1.0 因其优越的性能赢得了多项权威的国际大奖。Borland Delphi 1.0 最大的特点之一是全面更新了 OWL,取而代之的是全新的 Visual Component Library,从此开始了组件(Component)的时代。可能大多数编程者使用的 VB 较早版本是 VB 3.0,它确实好用,但还是解释运行,Delphi 就不同了。Delphi 为编程者提供了真正的编译,除了数据库程序需要附带 BDE 驱动程序一同发布外,其他程序只要一个可执行文件就可以,免除了像 VB 发布程序时需要一大堆的支持文件(大部分是 OCX 文件)。虽然五年过去了,至今还有一些公司用 Delphi 1.0 版为一些要求不高、电脑配置不高的用户开发小型网络终端软件,由此可见 Delphi 的风采。

虽然 Delphi 为 Borland 带来了利润,但股价还是连年下跌。Borland 公司后来调整了发展方向,并易名为 Inprise(中文名称——英博思),让不少迷恋 Borland 的人着实伤心,但 Inprise 为广大 Delphi 编程用户着想,继续开发 Delphi 后续版本,而且在 Delphi 的发行中继续使用“Borland”这个名称(Delphi 5.0 也是如此),也使我们这些 Delphi 追随者感到一丝欣慰。

Delphi 是 Borland 公司推出的一个基于 Windows 平台、以 Pascal 语言为基础的快速应用程序集成开发环境(IDE)。Delphi 使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想,将可视化界面与面向对象技术紧密结合起来,并封装了 Windows 编程的复杂性。Delphi 采用世界领先水平的快速编译器,使开发效率大大提高;Delphi 丰富而高效的数据库驱动技术,使开发基于数据库的应用系统方便而快捷;Delphi 丰富的多媒体支持功能、Web 应用程序的开发功能以及多线程技术,使其成为最好的应用程序开发工具之一。

Delphi 既可开发本地类型的软件,又可开发客户机/服务器类型的程序,并提供丰富的数据库管理工具。它不仅提供了大量的数据库组件,并配有数据库引擎 BDE 和 ADO 访问多种数据库,同时还提供了强大的开发网络数据库能力。

Delphi 采用三层数据管理模式(数据层、对象层、应用程序层),把诸如数据模型、业务规则、窗体、对象等集中存储在对象存储库中。应用程序通过使用 Borland 公司提供的数据库引擎,可以毫无障碍地使用多种数据库,例如 Oracle、Sybase、Informix、Interbase、Microsoft SQL Server、Paradox、Xbase 等。

Delphi 已经成为越来越多的程序员首选的开发工具之一,很多软件公司的产品都是用 Delphi 开发完成,所以 Delphi 程序员目前的市场需求很大。正因如此,全国计算机等级考试于 2008 年新增了 Delphi 类型。

Delphi 7 是 Borland 公司于 2002 年 8 月推出的 Delphi 的版本。和 Delphi 6 以及以前的版本相比,Delphi 7 在集成开发环境 IDE、Web、数据库、编译器、模型生成支持及组件库等很多方面进行了改进,使之更适应开发的要求。虽然现在 Delphi 又向前发展了很多,但 Delphi 7 以其优越的性能仍然被广大编程爱好者使用,而且对于初入编程殿堂的人来说,Delphi 7 也不失为一个最佳的选择之一。



1.2 Delphi 7 的特点与功能

目前的编程语言和编程工具可谓“春色满园,百花齐放”。编程语言方面由以前的汇编、C、Pascal、Basic 等基于过程的编程语言发展到 C++、Object Pascal、Java 等面向对象的编程语言;编程工具也由以前的功能单一、界面呆板的 Turbo C 等发展到具备集成开发环境、功能强大、界面友好的 Delphi、Visual C++、Visual Basic、PowerBuilder、JBuilder 等。但是为什么面对如此多的选择,许多的程序员却对 Delphi 情有独钟呢? Delphi 具有哪些优势呢?

Delphi 最显著的特点就是高效性和稳定性,主要体现在以下 5 个方面:

- 可视化开发环境的性能;
- 编译器的速度和已编译代码的效率;
- 编程语言的功能及其复杂性;
- 数据库结构的灵活性和可扩展性;
- 框架对设计和使用模式的扩充。

1.2.1 可视化开发环境

可视化开发环境通常可分为 3 个部分:编辑器、调试窗口和窗体设计器。与大多数的现代 RAD(快速应用开发)工具一样,这三部分都是协同工作的。当程序员在窗体设计器中工作时,Delphi7 在后台自动为正在窗体中操纵的组件生成代码。程序员还可以自己在编辑器中加入代码来定义应用程序的行为,同时还可以在同一编辑器中通过设置断点和监控点等来调试程序。

总的来说,Delphi 7 的编辑器和其他工具的编辑器类似,但它的 Code Insight 技术却省去了许多输入工作的麻烦。这一技术是建立在编译器信息之上的,而不是基于像 Visual Basic 等使用的类型库,因此应用范围更广泛。

在调试器方面,Delphi 7 不仅可与 Visual Studio 媲美,同时它还具备了许多先进的功能,如远程调试、过程关联、DLL 和包调试、自动本地监控以及 CPU 窗口等。Delphi 7 还支持在调试时随意放置和停靠窗口,并把这一状态保存为命令的桌面设置。由此可见,Delphi 7 的 IDE(集成开发环境)实现了对调试功能的良好支持。

对 RAD 工具(Delphi、Visual C++、Visual Basic、C++ Builder 和 PowerBuilder 等)来说,窗体设计器是一项独特的功能。为一些更为经典的开发环境,都提供了对话编辑器,但却没有将窗体设计器集成到开发流程中。没有窗体设计器将会降低开发工具的整体效率。几年来,Delphi 和 Visual Basic 在完善窗体设计器的功能方面展开了激烈的竞争。它们的新版本功能一个比一个强。Delphi 窗体设计器与众不同之处在于,Delphi 是建立在一个真正面向对象的框架结构基础之上的。这样,你对基类所做的改变都将会传递给所有的派生类。这里涉及的一项关键技术就是 VFI(Visual Form Inheritance),即可视化窗体继承。VFI 技术使你能够动态地继承当前项目或对象库中的任何其他窗体。一旦基窗体发生改变,派生的窗体会立即予以更新。



1.2.2 编译器的速度和已编译代码的效率

快速的编译器可以使用户逐步递进地开发软件,形成经常修改源代码、重新编译、测试、再修改、再编译、再测试……这样一个良好的开发循环,如果编译速度很慢,开发者就不得不分批地修改代码,每次编译前进行多处修改以适应一个低效率的循环过程。提高运行效率、节约运行时间、生成的二进制代码更为短小,其优越性是不言而喻的。

也许 Pascal 编译器最著名的特点就是速度快,而 Delphi 正是建立在这种编译器的基础之上的。事实上,它可能是针对 Windows 的最快的高级语言本地代码编译器。以往速度很慢的 C++ 编译器在近年来取得了很大的进步,增加了链接和各种缓存策略,尤其是在 Visual C++ 和 C++ Builder 中。但即便如此,C++ 的编译器还是比 Delphi 慢了几倍。

编译速度一定能与运行效率成正比吗?当然不是。Delphi 和 C++ Builder 共享同一种编译器后端,因此生成的代码等效于由一个优秀的 C++ 编译器生成的代码。根据最新的可靠评估标准,Visual C++ 在许多场合都被认为在编译速度和生成代码长度方面是最有效的,这得益于一些极为有力地优化措施。

虽然对通常的应用程序开发来说,这些细小的优越性难以被注意到,但如果你正在编写复杂的计算代码,那么它们就会发挥作用。

Visual Basic 的编译技术有点特别。在开发过程中,Visual Basic 以一种集成的方式运作,而且反应相当敏锐。这种编译器速度比较慢,生成的可执行代码的效率也远远不及 Delphi 和 C++ 工具。

Java 是另一种有趣的语言。最新的基于 Java 的工具语言 JBuilder 和 Visual J++ 自称其编译速度能赶得上 Delphi,但是生成代码的执行效率却不尽如人意,因为 Java 是一种集成语言。虽然 Java 在稳步地前进,但在大多数场合,其运行速度却仍与 Delphi 和 C++ 相距甚远。

1.2.3 编程语言的功能及其复杂性

在旁观者的眼里,一种语言的功能和复杂程度是极为重要的,这也是许多人争论的热点。对这个人来说简单的东西,对那个人来说可能很难;对这个人来说功能有限的东西,对另一个人来说却可能是非常完美的。因此,以下几点仅源于编者个人的经验和体会。

从根本上来说,汇编是一种最有力的语言。用它你几乎无所不能。但是,即便是用汇编开发最简单的应用程序,难度也非常大,还可能一无所获。不仅如此,要想在一个小组开发环境中保留一段汇编代码,不管保留多长时间,有时也是根本不可能的。因为代码从一个人传给另一个人,再到下一个人,设计思想和意图越来越不明朗,直到代码看起来如同“天书”。因此,我们对汇编的评价很低,它虽然功能很强大,但对几乎所有的开发者来说都太复杂了。

C++ 是另一种极为有力的语言。在它的潜在功能(如预处理器宏、模板、操作符重载等)的帮助下,你几乎可以使用 C++ 设计你自己的语言。只要合理地使用其丰富的功能选项,就可以开发出简洁直观、易于维护的代码。然而,问题是,许多开发者总滥用这些功能,这就很容易导致发生重大错误。事实上,写出糟糕的 C++ 代码反倒比写出好的 C++ 代码更容易。因为这种语言自己不会朝着好的设计方向前进——这由开发者决定。

Object Pascal 和 Java 给我们的感觉很相似,因为它们很好地把握住了复杂性和功能性的平衡。它们都采取了这样的一种途径,即限制其可用功能以加强开发者的逻辑设计。例如,



两者都避免了完全面向对象但却容易被滥用的多重继承的观念,而是实现了一个执行多重接口功能的类。两者都不支持美观却危险的操作符加载。两者都有一些强大的功能,诸如异常处理、运行期类型信息和生存期内存自我管理字符串。同时,两种语言都不是由专门的编委会写出来的,而是来自于单个组织中对这种语言有着共同理解的个人或小组。

1.2.4 帮助的使用

作为一种性能优越的编程工具,Delphi 7 同样也提供了功能强大的帮助功能,你可以随时随地取得需要的帮助信息。下面列举了使用帮助系统的一些小知识,供读者参考。

1. 当遇到问题时,可以尝试按下[F1]键,Delphi 会根据当前的焦点而启动相应的帮助,一般情况下 Delphi 都会准确的定位。

Delphi 的联机帮助具有智能判断的功能。如果当前位置在对象编辑器中的某个属性中时,按下[F1]键后则会自动显示该属性的帮助内容;如果当前显示的是 Delphi 系统的某个对话框,按下[F1]键后则会显示该对话框的有关帮助内容;如果弹出了某个菜单,这时按下[F1]键,则会显示该菜单命令的有关帮助内容。

2. 如果需要浏览系统的一些帮助内容,比如说对象 Pascal 语言,可以通过菜单 Help→Delphi Help 命令打开“帮助主题:Delphi Help”对话框,在目录标签中展开相关的条目,进行系统地学习。

3. 在使用 Delphi 编程的过程中,常常要使用到 Windows API 函数,然而通过菜单 Help→Delphi Help 命令打开的“帮助主题:Delphi Help”对话框中,在“索引”选项卡中可能查不到有关的帮助信息,比如 Windows API 函数 Message Beep 的有关帮助信息。

一种解决的办法就是在代码编辑器中输入 Windows API 函数的名称,然后将光标定位到该名称中,接着按下[F1]键。

另一种办法就是可以通过菜单栏中的 Help→Windows SDK 直接打开帮助文件 Win32.hlp,在其中的“索引”选项卡中查找。

在使用一些 Windows API 函数时,可能会出现一条错误信息,比如:“[Error]Unit1.pas (31):Undeclared identifier:‘OpenPrinter’”。这表明应该将该 API 函数声明所在的单元包含在 uses 语句中。至于是哪一个单元,可以按下该 API 函数的帮助内容中的“QuickInfo”按钮,查看 Import Library 中指定的是哪个库文件。

4. 在帮助文件打开的情况下,如果要浏览相关内容,可以按下工具栏中的“>>”按钮或“<<”按钮,通常这样是在同一个主题中进行浏览。有些帮助文件中有 Topic Group,通过它可以新打开一个主题列表窗口,这样可以非常方便地浏览一个完整主题的帮助内容。此外,通过一些 See Also 热链接,也可以查看相关的帮助内容。

5. 在组件栏上右击,通过快捷菜单的 Help 命令,可以查看有关组件的帮助内容。

6. 在 Delphi 集成开发环境中,通过菜单栏中的 Help→Customize 命令可以打开一个 Open Help 窗口,在该窗口中可以对帮助文件进行管理,其中包括帮助内容、帮助索引、链接位置和帮助工程文件等的管理。实际上,该窗口运行的是 oh.exe,该文件位于目录“…\Borland\Delphi 7\Bin”中。



1.3 对象的基本概念

为了理解面向对象的理论,必须从最基本的概念入手,通过对最基本的概念的掌握,来真正认识面向对象方法的作用。这一节我们介绍面向对象理论中的几个最基本的部件。

1.3.1 对象的定义

对象是面向对象系统中的核心概念,如果不能正确地认识和定义它,就无法掌握面向对象的理论。

首先需要搞清楚的第一个问题是,什么是对象?对象具有两方面的含义,即在现实世界中的含义和在计算机世界中的含义。

在我们所生活的现实世界中,“对象”无处不在。在我们身边存在的一切事物都是对象,例如,一粒米、一本书、一个人、一所学校,甚至一个地球,这些都是对象。除去这些可以触及的事物是对象之外,还有一些无法整体触及的抽象事件,例如,一次演出、一场球赛、一次借书等过程,也都是对象。

一个对象既可以非常简单,又可以非常复杂,复杂的对象往往可以是由若干个简单的对象组合而成。

所有的这些对象,除去它们都是现实世界中所存在的事物之外,它们都还具有各自不同的特征。例如一粒米,首先它是一粒米这样一个客观存在的东西(也可以说是实体,即实实在在存在的物体),汉语中称作“米”,其次它还具有颜色、体积、形状、重量等属性,再次它还可以被食用,可以用做原料等功能用途。也就是说一粒米是一个具有自身状态和自身功能的客观存在。再例如一个人,首先它也是一个客观实体,具有一个名字来标识,其次它具有性别、年龄、身高、体重等这些体现他自身状态的特征;再次他还具有一些技能,例如会说英语、会修电器等。

通过上面的这些举例我们可以对“对象”下一个定义,即对象是现实世界中的一个实体,它具有如下特性:

- 有一个名字以区别于其他对象;
- 有一组状态用来描述它的某些特征;
- 有一组操作,每一个操作决定对象的一种功能或行为;
- 对象的操作可分为两类:一类是自身所承受的操作,一类是施加于其他对象的操作。

对象是其自身所具有的状态特征及可以对这些状态施加的操作结合在一起所构成的独立实体。

例如,有一个人名叫张三,性别男,身高 1.78m、体重 65kg、民族汉,会修电视机,能够教计算机课,能够当班主任。下面我们来描述这个对象:

对象名: 张三

对象的状态:

性别: 男

身高: 1.78m

体重: 65kg



对象的功能(可做的操作):

回答身高

回答体重

回答性别 属于自身所承受的操作

回答民族

修理电视机

教计算机课 属于施加于其他对象的操作

当班主任

在计算机世界中我们可以把对象看成是存储器中一个可标识的区域,它能保存固定或可变数目的数值(或数值的集合)。

1.3.2 对象的划分

我们在解决实际问题时,首先要对这个问题进行分析,分析一下要解决的这个问题包含什么成分,即包含哪几个对象,每一个对象有什么作用,对象之间又有什么关系。

要设计一个面向对象的程序,对象的确定和划分是非常重要的,对象的确定与划分得是否适宜直接影响到所编制的程序的质量。如何才能使对象确定与划分得合理,这就需要对问题进行全面、细致的分析。值得注意的是,对象的确定与划分并没有唯一的标准,它还依赖于设置对象的目的和所需进行的操作。例如,我们要模拟一个学校系统,模拟的目的不同,所选择的对象就不同。

若模拟的目的是为了收集学生的学习情况,那么我们可以设置对象来分别表示学生、教师、教材、课程、教室、图书馆的图书等。

若模拟的目的是为了收集学生在学校的所有学习和生活情况,那么我们在原有的基础上还需增加学生宿舍、学生食堂,以及医院、文体设施、课外活动等作为要考察的对象。

在面向对象程序设计中,对象的选择是绝对关键的一步。若对象选择得适宜,既可以便于程序进行扩充,又可以为以后的其他应用提供基础。如何进行对象的选择,并没有固定的方法,它依赖于设计人员的经验和技巧,但对现实世界事物的把握有一个最基本的原则,就是寻求一个大系统中事物的共性,将所有具有共性的系统成分确定为一种对象。

1.3.3 对象的状态

一个对象之所以能在现实世界中独立存在,是因为它具有自身的状态——即自身所具有的那些特征。由于这些状态的存在,使其能对自身及对外界对象施加操作。当然一个对象的这些状态并不是完全用来直接为外界服务的,但它们本身是能够为外界服务的基础。

例如,一个人的五脏六腑是他的内部状态,作为人这个对象还具有一些诸如身高、体重等外部状态。所有这些状态中有一些并不是用来直接向外界提供服务的,如那些内部状态,但它们体现了人的身体状况,若没有好的身体,人是不能很好地为外界服务的,因此这些内部状态是基础。

在面向对象系统中,一个对象的状态是通过“域”来描述的,也称为私有存储单元,这些私有存储单元只能由它自己的操作来进行处理。



1.3.4 对象的特性

对象从形式上看是系统程序员、应用程序员或用户所定义的抽象数据类型的变量,当用户定义了一个对象,就创造出了具有丰富内涵的新的抽象数据类型。它是构成和支撑整个面向对象程序设计(OOP)最重要的细胞与基石。它具有下述三个特性。

1. 模块独立性

从逻辑上看,一个对象是独立存在的模块。从外部看这模块,只需了解它具有哪些功能,至于它如何实现这些功能和使用哪些局部数据来完成它们,则都“隐蔽”在模块内部,这个特性有点像大家比较熟悉的“黑箱”。这意味着模块内部状态不因外界的干扰而改变,也不会波及其他模块;模块间的依赖性极小或几乎没有;各模块可独立为系统所组合选用,也可被程序员重用,而不必担心波及或破坏其他模块。

2. 动态连接性

客观世界中各式各样的对象,并不是孤立存在的,它们之间是有联系的,正是它们之间的相互作用、联系和连接,才构成了世间各种不同的系统。同时,在面向对象程序设计中,通过消息激活机制,把对象之间动态联系连接在一起,使整个机体运转起来,便称为对象的连接性。

3. 易维护性

由于对象的功能被“隐蔽”,好像核桃的壳保护内部的核桃仁一样保护对象的内部,所以修改、完善功能及其实现的细节都被局限于该对象的内部,不会波及外部,这就使得对象和整个系统变得非常容易维护。

1.3.5 消息

对象与对象之间并不是彼此孤立的,它们之间存在着联系,在面向对象系统中,对象之间的联系是通过消息来传递的。

1. 什么是消息

我们通过一个具体的例子来叙述,例如我们在上面举的例子,一个对象为人,名字叫张三,他是一个计算机讲师,他会修理电视机,也可以上计算机课、也可以担任班主任。除去这些向他人提供的服务外,他也要接收其他对象的服务,如吃饭、穿衣、娱乐等。他要吃饭但不需要也不可能自己去种地,他要穿衣但不需要也不可能自己去织布,他就要请求他人来帮助解决这些问题,这里的“请求”便是一个人与其他人进行交互的手段。同样,他什么时候修电视机,什么时候上课,需要在得到其他对象的请求后才进行。因此,在面向对象技术中这个“请求”本身就是发送的“消息”。在日常生活中除去“请求”以外,还有“命令”,如上级对下级的命令等,这些“命令”也是一种“消息”。

因此,消息是对象之间相互请求或相互协作的途径,是要求某个对象执行其中某个功能操作的规格的说明。

通常,我们把发送消息的对象称为发送者,接收消息的对象称为接收者。对象间的联系,只能通过传送消息来进行。对象也只有在收到消息时,才被激活,被激活后的对象代码将“知道”如何去操作它的私有数据,去完成所发送的消息要求的功能。

2. 消息具有三个性质



- 同一对象可接收不同形式的多个消息,产生不同的响应;
- 相同形式的消息可以送给不同对象,所做出的响应可以是截然不同的;
- 消息的发送可以不考虑具体的接收者,对象可以响应消息,也可以对消息不予理会,对消息的响应并不是必须的。

3. 公有消息和私有消息

在面向对象系统中,消息分为两类:即公有消息和私有消息。

到底哪些消息是公有消息,哪些消息是私有消息,需要有一个明确的规定。

若有一批消息同属于一个对象,其中有一部分是由外界对象直接向它发送的,称之为公有消息;还有一部分则是它自己向本身发送的,这些消息是不对外开放的,外界不必了解它,称之为私有消息。

外界对象向此对象发送消息时只能发送公有消息,而不能发送私有消息,私有消息是由其自身发送的。

4. 特定于对象的消息

所谓特定于对象的消息是指将所有能支持此对象可接受消息的函数集中在一起,形成一个大消息,称为特定于对象的消息,这些消息让对象执行这个函数而不管它可能做什么及怎么做。

5. 消息可分为三种类型

- 可以返回对象的内部状态的消息;
- 可以改变对象的内部状态的消息;
- 可以做一些特定的操作,改变系统状态的消息。

6. 消息序列

在日常生活中,人与人之间的交流,并不是只向一个人请求帮助或发号施令,也不是只向别人发出一次请求或命令。

例如,有一个经理,他一上班就给他的下属们分配了如下任务:

老王,今天去银行跑贷款;

老张,中午去找 XX 客户签合同;

小王,上午去参加产品展示会;

小刘,把这份文件送到主管部门;

林秘书,今天将年度总结写出来;

.....

这就是一个非常自然的消息序列。在这个序列中的各个消息如何实施,即如何签合同、如何送文件、如何写总结等,都不是经理的事情,而是具体执行这些指令的人的事情。

在面向对象系统中就是用消息序列来表述要解决的一系列问题的。

1.3.6 类

类是面向对象系统中最重要概念,面向对象程序设计中的所有操作都归结为对类的操作。在面向对象程序设计中正因为引入了类的概念,使得它与传统的结构化程序设计明显地区分开来。

1. 类的定义



像其他概念一样,我们首先要讨论的是,什么是类?

在日常生活中,我们经常听到“类”这个术语,它是对一组客观对象的抽象,它将该组对象所具有的共同特征(包括结构特征和行为特征)集中起来,以说明该组对象的能力和性质。

例如“人类”这个词就抽象着所有人的共同之处。

在计算机世界的系统构成上,类形成了一个具有特定功能的模块和一种代码共享的手段。它为程序员提供了一种工具,使得他们可以方便地建立所需要的任何类型,并可以像一般类型那样方便地使用。

2. 类与实例的关系

类的概念在前面已经给出,组成类的对象均为此类的实例。

类与实例之间的关系可以看成是抽象与具体的关系。例如我们看到一个苹果,就会说:“这是一个苹果”,把它变成我们面向对象系统的语言来叙述:“这是一个苹果类的实例”,再例如,李四是一个学生,学生是一个类,而李四作为一个具体的对象,是学生类的一个实例。

类是多个实例的综合抽象,而实例又是类的个体实物。

对于同一类的不同实例之间,必定具有如下特点:

- 相同的操作集合;
- 相同的属性集合;
- 不同的对象名。

3. 类的确定与描述

类的确定是采用归纳的方法,从对所遇到的对象的分析中,归纳出共同的特征来确定一个类。

例如我们可以通过对许多马的归纳总结,抽象出各种马的共同特征,从而得到马类。

在实际工作中,单纯地描述一个独立的对象如何工作是没有任何意义的,往往是描述一个类的工作。

1.4 Delphi 7 集成开发环境的使用

集成开发环境(Integrate Development Environment,以下简称 IDE)是指通过单一的控制面板访问所有的开发工具:编辑器、调试器、对象管理器、编译器、实时分析器、图形信号分析器等。当前流行的开发工具,比如 Delphi、Visual C++、Visual Basic、C++ Builder 和 PowerBuilder 等都给程序员提供了集成式开发环境,极大地提高了程序员的开发效率,缩短了程序的开发时间。在本节中介绍一下 Delphi 7 中的 IDE。

1.4.1 认识集成开发环境

Delphi 7 的 IDE(如图 1-1 所示)主要包括 7 个部分:主窗口、组件面板、工具栏、窗体设计器、代码编辑器、对象观察器(object inspector)和代码浏览器。使用 IDE,软件开发人员可以很方便地完成创建、调试、修改应用程序等各种操作。下面将分别对它们予以介绍。