

万水 PowerBuilder
从入门到精通系列

PowerBuilder 8.0

数据窗口 实用编程技术

刘云洲 于德壹 编著



中国水利水电出版社
www.waterpub.com.cn

万水 PowerBuilder 从入门到精通系列

PowerBuilder 8.0 数据窗口 实用编程技术

刘云洲 于德壹 编著

中国水利水电出版社

内 容 提 要

PowerBuilder 是 Sybase 公司推出的优秀的数据库前端开发工具,也是流行的基于客户机/服务器模式的开发工具。2001 年, Sybase 又推出了 PowerBuilder 8.0。

在 PowerBuilder 8.0 中, Sybase 对开发应用程序的模式进行了重大的改变,特别是在应用程序的组织方式和集成开发环境等方面。

本书详细系统地讲述了 PowerBuilder 的核心技术,即是数据窗口技术。其中既有理论方面的精彩介绍,又有大量的开发实例。读者既可以把本书当成开发人员指南,又可以把本书当成用户的参考手册。

特别的,本书适合于对 PowerBuilder 8.0 的集成开发环境和 PowerScript 语言有初步了解的开发人员。

图书在版编目(CIP)数据

PowerBuilder 8.0 数据窗口实用编程技术 / 刘云洲, 于德壹编著. —北京: 中国水利水电出版社, 2001.10

(万水 PowerBuilder 从入门到精通系列)

ISBN 7-5084-0854-3

I. P... II. ①刘...②于... III. 数据库系统 软件工具, PowerBuilder 8.0 程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2001) 第 073229 号

165554/03

书 名	PowerBuilder 8.0 数据窗口实用编程技术
作 者	刘云洲 于德壹 编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 68359286 (万水)、63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
印 刷	北京蓝空印刷厂
规 格	787×1092 毫米 16 开本 25.25 印张 550 千字
版 次	2002 年 1 月第一版 2002 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	38.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

前 言

在计算机的各个应用领域，数据库的开发与设计层出不穷。为了占据这块广阔的市场，各个大型的软件开发公司都推出了相应的数据库前端开发工具。在这些数据库开发工具中，Sybase 公司的 PowerBuilder 一枝独秀，为广大的开发人员所接受。

数据窗口 (DataWindow) 是 PowerBuilder 提供给开发人员快速建立基于数据库应用程序的最强有力的工具，也是 PowerBuilder 与其他面向对象的数据库应用前端开发工具的最主要的区别。它的本质是以自动化的用户/数据库接口为开发人员最大限度地节省时间和精力，并且这种自动化特性并不限制开发人员的主观能动性，开发者能以他所钟爱的方式来运用数据窗口，开发出高质量的应用程序。

在数据窗口技术中，数据窗口对象与数据窗口控件是两个比较重要的方面。其中，数据窗口对象控制了数据窗口如何从数据源中获得数据，以及以什么样的格式来显示这些数据；而数据窗口控件则用来把数据窗口对象和用户界面联系起来，同时提供了访问数据窗口对象中各种控件的属性，以及访问数据窗口从数据源中检索到的数据的能力。

数据窗口对象技术由两部分组成：数据源和显示风格。数据源表示了数据窗口从哪里获取数据，以及以什么样的方式来获取和更新这些数据。PowerBuilder 提供了 5 种类型的数据源，通过这些不同的数据源，我们可以访问任何形式的数据。

而显示风格则表示数据窗口将以什么样的方式来显示这些数据。PowerBuilder 提供了多达 11 种的显示风格，从最简单的 Grid 显示风格到复杂的 Crosstab 显示风格，PowerBuilder 都提供了简单易用的方法，使得开发人员能够以最快的速度完成应用程序的开发。

数据窗口控件和其他的 PowerBuilder 控件具有很多相似的属性，只有通过数据窗口控件，我们才能在窗口中把数据窗口和其他的 PowerBuilder 控件统一起来。更重要的是，数据窗口对象的各种属性以及数据窗口检索到的数据都必须通过数据窗口控件来访问。

数据窗口控件另外的一个很重要的作用是：数据窗口控件负责维护数据窗口中的各种数据，而这些数据可以是数据源中检索到的原始数据，也可以是用户增加和修改的数据，还可能是用户删除了的数据。通过维护这些数据，PowerBuilder 能够很容易生成更新数据库的 SQL 语句，例如 Update 语句、Insert 语句以及 Delete 语句等。

本书由刘云洲和于德壹主编，另外参加编写的人员还有张玉玲、徐平、邹洁、王艳燕、刘小华、李炎、闫高峰、杨桂莲、田丁、张娟、刘玉树等同志。全书由李琪统稿。

由于时间仓促、作者水平有限，本书错漏之处在所难免，欢迎广大读者批评指正。

编者

2001 年 5 月

目 录

前言

第 1 章 概述	1
1.1 PowerBuilder 简介	1
1.1.1 可视化的编程方法	2
1.1.2 面向对象的编程思想	3
1.1.3 优秀的数据库应用开发工具	4
1.1.4 基于多平台的交叉开发	6
1.1.5 专业的客户机/服务器及组件应用开发工具	7
1.1.6 对 Web 应用的强大支持	9
1.1.7 简洁强大的 PowerScript 语言	10
1.1.8 强大的源码管理功能	11
1.2 PowerBuilder 8.0 的新特性	14
1.2.1 新的用户接口和开发方法	14
1.2.2 新的 Web 特性	15
1.2.3 新的 EAServer 集成特性	16
1.2.4 新的数据库连接特性	16
1.2.5 对双字节字符集 (DBCS) 的支持	17
1.2.6 其他新特性	17
1.3 数据窗口技术简介	18
1.3.1 丰富的数据源	19
1.3.2 多样的显示格式	19
1.3.3 数据窗口表达式	20
1.3.4 数据窗口对象	20
1.3.5 数据窗口控件	21
1.3.6 数据存储对象	22
1.3.7 增强数据窗口对象	22
1.3.8 动态数据窗口技术	23
1.3.9 数据库事务对象	24
1.4 使用数据窗口的一般步骤	25
1.5 小结	25
第 2 章 PowerBuilder 与数据库管理系统	27
2.1 数据库描述文件	27

2.2	数据库接口	28
2.2.1	查看安装的数据库接口	28
2.2.2	通用数据库接口	30
2.2.3	专用数据库接口	32
2.2.4	通用接口和专用接口的比较	33
2.3	创建 ODBC 数据源	34
2.3.1	通过 Windows 的 ODBC 配置工具创建 ODBC 数据源	34
2.3.2	在程序中通过注册表来创建 ODBC 数据源	38
2.4	创建 ODBC 数据库描述文件	40
2.5	小结	41
第 3 章	创建数据窗口对象	42
3.1	理解数据窗口对象	42
3.2	数据窗口对象向导	43
3.2.1	连接数据库	43
3.2.2	选择显示风格	44
3.2.3	选择数据源	45
3.2.4	选择并设计字段	46
3.2.5	设置颜色和边框信息	47
3.2.6	确认先前的设置	47
3.2.7	进入数据窗口画板	48
3.2.8	预览数据窗口	49
3.2.9	保存数据窗口对象	51
3.3	数据窗口对象选项	52
3.3.1	General 选项卡	52
3.3.2	Generation 选项卡	53
3.3.3	Prefixes 选项卡	54
3.4	本章小结	56
第 4 章	使用数据源	57
4.1	Quick Select 数据源	57
4.1.1	理解 Quick Select 数据源	57
4.1.2	使用 Quick Select 数据源	58
4.2	SQL Select 数据源	68
4.2.1	理解 SQL Select 数据源	68
4.2.2	选择数据库表	69
4.2.3	浏览 SQL Select 画板	70

4.2.4	使用 SQL Select 画板.....	72
4.2.5	检索参数.....	85
4.2.6	定义连接 (Join)	91
4.2.7	定义联合 (Union)	100
4.3	Query 数据源	104
4.3.1	理解 Query 数据源	104
4.3.2	创建 Query 对象.....	106
4.3.3	创建 Query 数据源	107
4.4	Stored Procedure 数据源.....	108
4.5	External 数据源.....	115
4.5.1	理解 External 数据源	115
4.5.2	定义 External 数据源	116
4.5.3	使用 External 数据源	117
4.6	小结.....	119
第 5 章	使用显示风格	121
5.1	Freeform 显示风格.....	121
5.2	Grid 显示风格	122
5.3	Tabular 显示风格	122
5.4	Label 显示风格	123
5.5	RichText 显示风格	128
5.6	N-Up 显示风格	131
5.7	Crosstab 显示风格	133
5.8	Group 显示风格	139
5.9	Graph 显示风格	145
5.10	OLE 2.0 显示风格.....	149
5.11	Composite 显示风格	156
5.12	小结.....	157
第 6 章	设计数据窗口对象.....	158
6.1	数据窗口画板简介	158
6.1.1	“Design” 视图.....	159
6.1.2	“Properties” 视图	162
6.1.3	“Control List” 视图.....	162
6.1.4	“Preview” 视图	162
6.1.5	“Column Specifications” 视图.....	162
6.1.6	“Data” 视图.....	163

6.2	数据窗口对象的属性页	165
6.3	设置字段对象的属性	166
6.3.1	添加和删除字段对象	166
6.3.2	字段对象的显示格式	167
6.3.3	字段对象的编辑样式	170
6.3.4	字段的有效性规则	175
6.4	操纵数据窗口对象的行	175
6.4.1	定义过滤条件	176
6.4.2	定义排序字段	176
6.4.3	定义检索条件	177
6.5	数据窗口的更新属性	178
6.5.1	“Where Clause for Update/Delete” 组框	179
6.5.2	“Key Modification” 组框	180
6.5.3	“Identity Column” 下拉列表框	180
6.6	预览数据窗口对象	180
6.7	数据窗口的其他设置	182
6.7.1	压缩重复值	182
6.7.2	压缩字段间空间	183
6.8	小结	184
第 7 章	数据窗口增强对象	185
7.1	理解数据窗口增强对象	185
7.2	数据窗口增强对象的作用	185
7.3	数据窗口增强对象详解	186
7.3.1	Button 对象	186
7.3.2	Text 对象	196
7.3.3	Picture 对象	196
7.3.4	Line 对象	196
7.3.5	Oval 对象	196
7.3.6	Rectangle 对象	197
7.3.7	Round Rectangle 对象	197
7.3.8	Column 对象	197
7.3.9	Computed Field 对象	197
7.3.10	Graph 对象	201
7.3.11	Report 对象	216
7.3.12	Group Box 对象	218

7.3.13	OLE Object 对象.....	218
7.3.14	OLE Database Blob 对象.....	219
7.4	小结.....	222
第 8 章	数据窗口控件.....	223
8.1	理解数据窗口控件.....	223
8.2	创建数据窗口控件.....	224
8.3	数据窗口控件的属性.....	225
8.4	数据窗口控件的事件.....	226
8.4.1	检索与更新事件.....	226
8.4.2	单击事件.....	229
8.4.3	错误处理事件.....	229
8.4.4	记录处理事件.....	230
8.4.5	打印事件.....	231
8.5	数据窗口控件的函数.....	232
8.5.1	几个重要的概念.....	232
8.5.2	数据库数据操纵函数.....	236
8.5.3	缓冲区数据操纵函数.....	239
8.5.4	数据行滚动函数.....	241
8.5.5	数据行操纵函数.....	244
8.5.6	可编辑控件函数.....	246
8.5.7	数据项操纵函数.....	247
8.5.8	有效性规则函数.....	248
8.5.9	数据共享函数.....	249
8.5.10	打印函数.....	251
8.5.11	页面生成函数.....	253
8.6	数据窗口控件使用示例.....	254
8.6.1	DWExample 示例程序的外观.....	254
8.6.2	创建应用程序及窗口对象.....	254
8.6.3	创建一个数据窗口对象.....	256
8.6.4	往窗口对象上放置各种控件.....	257
8.6.5	添加连接数据库和断开数据库的函数.....	258
8.6.6	编写命令按钮的 Clicked 事件.....	260
8.6.7	编写数据窗口控件的 DBError 事件.....	265
8.7	小结.....	266

第 9 章 数据存储对象	267
9.1 理解数据存储对象	267
9.2 添加数据存储对象	267
9.3 数据存储对象的属性	270
9.3.1 ClassDefinition 属性	270
9.3.2 DataObject 属性	270
9.3.3 Object 属性	270
9.4 使用数据存储对象举例	271
9.4.1 进行后台数据库处理	272
9.4.2 以多种视图显示相同的数据	274
9.4.3 代替嵌入式 SQL 语句来操纵表中的数据	279
9.4.4 在分布式应用中代替客户机存取数据	284
9.5 使用数据存储对象的注意事项	284
9.6 小结	285
第 10 章 数据窗口表达式	286
10.1 数据窗口表达式的应用场合	286
10.2 数据窗口表达式运算符	287
10.2.1 算术运算符	287
10.2.2 关系运算符	288
10.2.3 连接运算符	291
10.2.4 逻辑运算符	291
10.2.5 运算符优先级	291
10.3 编辑数据窗口表达式	292
10.4 测试数据窗口表达式的值	294
10.4.1 使用 Describe 函数来测试数据窗口表达式的值	295
10.4.2 用当前的数据来测试条件表达式	296
10.5 数据窗口表达式使用举例	298
10.5.1 定义计算字段	298
10.5.2 定义过滤器	298
10.5.3 定义有效性规则	299
10.6 小结	300
第 11 章 数据窗口表达式函数	301
11.1 理解数据窗口表达式函数	301
11.1.1 使用数据窗口表达式函数的场合	301
11.1.2 数据窗口表达式函数的返回值	301

11.1.3	数据窗口表达式函数中的聚集函数	302
11.1.4	使用用户自定义的数据窗口表达式函数	302
11.1.5	数字显示格式的本地化	302
11.2	聚集函数	303
11.2.1	Avg 函数	304
11.2.2	Count 函数	306
11.2.3	CumulativePercent 函数	308
11.2.4	CumulativeSum 函数	310
11.2.5	First 函数	312
11.2.6	Large 函数	313
11.2.7	Last 函数	315
11.2.8	Max 函数	317
11.2.9	Median 函数	318
11.3	类型检查与转换函数	320
11.3.1	Integer 函数	320
11.3.2	IsDate 函数	321
11.4	数据窗口信息函数	321
11.4.1	CurrentRow 函数	322
11.4.2	Describe 函数	323
11.4.3	GetRow 函数	325
11.4.4	GetText 函数	325
11.5	时间日期函数	326
11.5.1	Date 函数	327
11.5.2	DateTime 函数	328
11.5.3	Day 函数	328
11.5.4	DayName 函数	329
11.5.5	DayNumber 函数	329
11.5.6	DaysAfter 函数	330
11.5.7	Hour 函数	330
11.6	字符串操作函数	331
11.6.1	Fill 函数	331
11.6.2	WordCap 函数	332
11.7	交叉表函数	332
11.7.1	CrosstabAvg 函数	333
11.7.2	CrosstabCount 函数	337

11.7.3	CrosstabMax 函数	339
11.7.4	CrosstabMin 函数	340
11.7.5	CrosstabSum 函数	342
11.8	数值运算函数	343
11.8.1	Abs 函数	344
11.8.2	Exp 函数	344
11.8.3	Fact 函数	344
11.8.4	Int 函数	345
11.9	其他函数	345
11.9.1	If 函数	346
11.9.2	Case 函数	347
11.10	小结	348
第 12 章	使用动态数据窗口	349
12.1	属性表达式	349
12.1.1	理解属性表达式	349
12.1.2	使用 DataWindow Syntax 工具	351
12.1.3	使用 PowerBuilder Help 联机帮助	353
12.2	访问数据窗口中的数据	354
12.2.1	SetItem 函数	354
12.2.2	GetItem 系列函数	355
12.2.3	属性表达式	357
12.2.4	两种方法的比较	358
12.3	Describe 函数	358
12.3.1	语法定义	359
12.3.2	使用方法	359
12.3.3	用法举例	361
12.4	Modify 函数	362
12.4.1	语法定义	362
12.4.2	使用方法	363
12.4.3	典型应用	369
12.5	Create 函数	369
12.5.1	语法定义	370
12.5.2	使用方法	370
12.5.3	用法举例	371
12.6	SyntaxFromSQL 函数	374

12.6.1	语法定义	374
12.6.2	使用方法	375
12.6.3	用法举例	377
12.7	增强检索功能	380
12.7.1	QueryMode 属性	380
12.7.2	QuerySort 属性	383
12.7.3	QueryClear 属性	384
12.8	利用库函数来生成数据窗口对象	384
12.8.1	PowerBuilder 库函数简介	384
12.8.2	利用库函数生成数据窗口对象的一般步骤	388
12.9	小结	389

第 1 章 概述

Sybase 推出的 PowerBuilder 是功能强大的数据库应用程序和客户机/服务器应用程序的快速开发工具。自 1991 年 6 月 PowerBuilder 1.0 版发行后, Sybase 先后推出了多个版本, 今年又推出 PowerBuilder 8.0。在这个版本中, PowerBuilder 增加和改进了许多功能, 例如用户可以在多个项目之间随意切换, 而不用反复打开和关闭这些项目以及增强的 Web 特性、新的数据库连接特性等。

PowerBuilder 具有很多的优良特性, 例如可视化的面向对象的开发环境、多平台下的交叉设计、对众多数据库管理系统和通用数据库接口的支持, 以及对分布式计算和网络应用的强有力支持, 都使得 PowerBuilder 成为日益流行的数据库应用开发工具。可以这么说, PowerBuilder 集成了世界上应用程序开发方面最先进的技术, 学习 PowerBuilder 必将提高我们的应用开发效率, 使我们成为一名优秀的程序员。

在 PowerBuilder 中, Sybase 设计了一种简洁而功能强大的面向对象语言——PowerScript, 它使得开发者能非常轻松地书写各种函数和事件处理程序。PowerScript 脚本语言的风格既类似于比较简单的 BASIC 语言, 又类似于比较高级的 C/C++ 语言, 这种特性使得它有很好的用户基础。使用各种编程语言的用户都会在 PowerScript 语言中找到熟悉的东西。这门高级的 4GL 语言支持许多编程语言的特性, 例如支持用户自定义的函数和对象, 支持嵌入式的 SQL 语句, 支持面向对象的思想等。

而数据窗口 (DataWindow) 技术则是 PowerBuilder 序列产品区别并领先于其他数据库应用开发工具的核心技术之一。通过数据窗口, 程序设计者既可以非常轻松地访问各种底层数据库, 又能开发出各种符合用户要求的图形界面来操纵数据库中的数据。

对开发者来讲, 数据窗口充当了数据库和用户的接口。数据窗口能够从众多的数据源中获取所需的数据, 然后以丰富而专用的风格显示给用户。数据窗口既是操作数据库的行家里手, 又是开发友好美观图形用户界面 (GUI) 的专家。掌握了数据窗口技术, 也就差不多掌握了 PowerBuilder 的精髓。

本章的目的是简要介绍一下 PowerBuilder 的特点以及 PowerBuilder 8.0 的新特性, 然后着重讨论数据窗口的概念。希望不管是刚入门的新手, 还是已经比较熟悉 PowerBuilder 的专业人士, 都能从本章中找到一些有用的信息。

1.1 PowerBuilder 简介

作为一种数据库应用程序和客户机/服务器应用程序的开发工具, PowerBuilder 具有许多

优秀的特性。正是这些特性使得 PowerBuilder 成为了一种优秀的应用开发工具，从而吸引了众多的用户投身于 PowerBuilder 的学习和使用中。

本小节将简要介绍一下 PowerBuilder 的主要特点，说明使 PowerBuilder 成为一种优秀应用开发工具的那些技术。

1.1.1 可视化的编程方法

使用 PowerBuilder，开发人员可以在它的可视化集成开发环境（IDE）中以一种快速简洁的方式来开发应用程序。通过简单的拖放技术，程序员就能设计出复杂的 Windows 应用程序，从而满足用户的要求。

在 PowerBuilder 的 IDE 中开发应用，诸如窗口、菜单、文本、按钮等 Windows 的标准控件都不必在 Script 程序中显式地创建。这些控件的布局、颜色、字体、大小等属性都可以在 PowerBuilder 的集成开发环境中简单完成。

一个应用程序通常需要有多个窗口协同工作，共同完成用户所要求的任务。根据业务逻辑的不同，这些窗口有不同的弹出顺序、不同的外观及不同的功能。我们在开发一个应用系统时，首先要在集成开发环境中生成应用需要的那些窗口，然后确定这些窗口的大小、外观、标题等各种各样的属性。

为了让这些窗口和用户进行交互，我们还需要为这些窗口添加各种各样的 Windows 控件，并且设置这些控件的属性。最重要的是，为了让这些窗口具有一定的功能，我们需要为它们和它们所包含的控件书写各种事件处理程序，来处理用户的输入、系统的变化、错误的产生等多种事件。

利用 PowerBuilder 的 IDE，所有的这些工作都可以轻松完成。

通过 PowerBuilder 的菜单，我们可以很方便地为应用程序生成一个窗口对象。然后，我们就可以在窗口画板的 Properties 视图中设置该窗口的属性。所有这些属性的设置都通过 PowerBuilder 的友好界面简单地完成。

下一步要做的是，在窗口画板的 Layout 视图中设计该窗口所需要的各种控件。我们可以从 PainterBar 的控件面板上拖动所需的控件到 Layout 视图里，然后放置在该窗口的窗体上。选中这些控件或者窗体本身，我们就可以在 Properties 视图里设置它们的属性了。

设置完了这些控件的初始属性后，我们就可以在它们的事件列表中选中并双击某个事件，PowerBuilder 则自动进入了窗口画板的 Script 视图中，我们可以在该视图中直接编写各种事件处理程序。

不仅仅是窗口，其他许多的系统对象和用户对象，例如菜单、用户对象、结构和函数等，都可以在 PowerBuilder 的集成开发环境界面中得以简单快速的设计。

正因为如此，开发人员所要考虑的一切都可以在 PowerBuilder 的各个画板和视图中轻松完成。我们不用去过问许多和应用功能无关的细节，只需集中精力去讨论应用程序的需求。PowerBuilder 能充分缩短从设计需求到开发产品的时间，使我们的编码和实现周期变得非常

短，从而降低了系统的总体成本。

在 Windows 应用程序的开发方面，PowerBuilder 支持基本上所有的 Windows 编程特性，例如事件驱动的编程风格，多进程和多线程的概念，图形化用户界面（GUI），丰富强大的应用编程接口（API）等。这使得 PowerBuilder 在流行的 Windows 操作系统平台上得到了用户的普遍赞赏，取得了用户的信任。

1.1.2 面向对象的编程思想

PowerBuilder 不仅支持面向对象的思想，并且还把面向对象的技术和方法充分运用到 PowerBuilder 的各方面。与面向对象技术相结合，使得 PowerBuilder 能够从面向对象的技术中获得不断向前发展的动力。

面向对象的技术主要有两个方面的优秀思想：

第一个方面是它主张把客观事物的数据和对这些数据的操作封装在一切抽象到计算机系统中。这和结构化思想中把数据和操作分离开来看待的做法有本质差别。在面向对象的思想中，这些数据叫做属性，而使用这些数据的操作叫做方法。

通过这种封装，面向对象的技术把某个对象内部的实现细节隐藏了起来，而表现给外部的是一个一个的接口。只有通过这些接口，对象外部的其他对象才能和这个对象进行交互，才能间接地使用对象内部的数据。因为对数据的使用是通过对象自身的方法来实现的，所以这就保证了这些数据的正确性、一致性和完整性。

于是，我们在设计一个系统时可以采用一种比较清晰的思路来进行。首先，我们要抽象和设计好系统中的各个小型对象，定义它们的数据，实现它们的方法，把各种细节都包装起来，并对外界隐蔽。然后，再利用这些小型对象搭建成中型的对象，利用中型对象搭建大型的对象，直到最后生成整个系统。

由于高层对象不必过问底层对象的实现细节，不必担心底层对象数据的正确性，所以系统设计的复杂性大大地降低，同时保证了系统的正确性和稳定性。这是面向对象技术如此具有魅力的原因之一。

另一个方面面向对象的技术支持继承的概念，子对象不仅可以继承父对象的数据和方法，还可以拥有自己独有的数据和方法。这样，具有很多相似特性和某些不同特性的对象就都可以去继承同一个祖先，该祖先实现它们公共的那些属性和方法，而这个祖先的各个子孙则实现那些不同的属性和方法。

因此，我们不必为每个子对象都实现它们共有的那些属性和方法，而只需在它们的祖先中实现一次。这极大地增加了程序代码的重用性，降低了代码的编写总量，简化了系统的实现。并且，我们可以使用前人遗留下来的对象，实现了人类智慧的共享。

除了这两方面的主要特性外，面向对象的技术还有其他诱人的思想。例如，重载允许子对象重新实现父对象的方法，虚函数允许对父对象方法的调用在运行时去调用了对象的方法，多态性允许相同的方法具有不同的参数和不同的实现等。

面向对象的思想已经渗透到了计算机科学的各个领域。在软件工程领域中,面向对象的思想体现在面向对象的分析(OOA)、面向对象的设计(OOD)、对象建模、面向对象的编程(OOP)以及组件技术等各方面。所有的这些技术都体现了面向对象思想的巨大生命力,体现了它的优越性。

PowerBuilder 完全支持面向对象的思想。在 PowerBuilder 中,各种系统对象,大到应用、窗口、菜单,小到各种控件、队列、缓冲区,都是一个一个的对象。用户还可以定义自己的对象,来满足系统对象不能满足的要求。这些对象都拥有自己的属性、方法和事件,既能和对象之外的其他对象进行交互,又能对外封闭地处理各种内部事务,保证自身数据的正确性和完整性。

利用 PowerBuilder 进行应用程序的开发就是一个执行面向对象思想和技术的过程。首先我们需要一个全局范围的应用对象,然后在去设计各种较小的对象,例如窗口、菜单等以及各种用户对象,然后再利用这些较小的对象生成较大的对象,最后生成整个应用。

另外,PowerBuilder 提供的 PowerScript 语言也是一种面向对象的编程语言。该语言支持面向对象的众多特性,例如支持继承、封装、重载、多态、异常处理等面向对象的基本概念。这使得 PowerScript 能够顺应当代程序设计语言的潮流,能够最大地降低应用系统的复杂性。用 PowerScript 开发出来的程序具有良好的组织结构,这使得程序员对代码的管理和重用更加简单和方便。

总之,面向对象的思想在 PowerBuilder 中得到了充分的体现,PowerBuilder 也从面向对象的思想中获取了强大的生命力。学习 PowerBuilder,我们可以学习和运用到很多面向对象的知识,培养我们面向对象的思考方式。

1.1.3 优秀的数据库应用开发工具

在数据库应用程序开发方面,PowerBuilder 也处于领先的地位。和其他数据库应用开发工具(例如 Delphi、Visual Basic 等)一样,PowerBuilder 能通过一些简单的操作,建立与数据库的连接,并且对数据库的数据进行检索、增加、修改和删除。

PowerBuilder 支持众多的数据库管理系统,提供了多种数据库接口,使得用户不必直接去使用这些 DBMS 所提供的 API 就可以访问数据库。

专用接口是 PowerBuilder 针对大型的关系型数据库管理系统而专门设计的。设计专用接口的目的,是为了利用这些 DBMS 提供的 API 来提高访问数据库的速度。

除了提供基于各个 DBMS 的专用数据库接口外,PowerBuilder 还提供了通用的数据库接口,例如 ODBC (Open Database Connectivity, 开放数据库互连)、JDBC (Java Database Connectivity, Java 数据库互连)。

由于 ODBC 和 JDBC 的通用性,使得它们的性能受到了-定的影响,在访问数据库的时候速度不是很快,特别是对大型的远程数据库管理系统。

对于小型的本地数据库管理系统,PowerBuilder 不向用户提供针对它们的专用数据库接