

e 时代网络学科推荐教程

E-time Network Discipline Recommend Textbooks

精通 C++ Builder 5 程序设计高级教程

陈周造 陈灿煌 / 编著

阐述 C++ Builder 5 最完整的工具书

教你利用开放且具弹性的数据库链接能力及高效性存取企业级数据库管理系统

MIDAS 提供多层分布运算解决方案，帮助你实现“随时随地取得任何数据”

运用 C++ Builder 5 将繁杂数据变为决策信息

深入介绍可以直接调用 Windows API 达到低阶系统开发应用的目的

附赠光盘内含本书完整范例



中国青年出版社



博硕文化股份有限公司

e 时代网络学科推荐教程
E-time Network Discipline Recommend Textbooks

精通 C++ Builder 5 程序设计高级教程

陈周造 陈灿煌 / 编著



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书由博硕文化股份有限公司授权中国青年出版社独家出版。未经出版者书面许可,任何单位和个人均不得以任何形式复制或传播本书的部分或全部。

版权贸易合同登记号: 01-2000-4169

策 划: 胡守文
王修文
郭 光
责任编辑: 江 颖
何 琼
责任校对: 肖新民

书 名: 《精通 C++Builder 5 程序设计高级教程》
编 著: 陈周造 陈灿煌
出版发行: 中国青年出版社
地址: 北京市东四十二条 21 号 邮政编码: 100708
电话: (010) 64069368 传真: (010) 64053266
印 刷: 北京新丰印刷厂
开 本: 16 开
版 次: 2001 年 2 月北京第 1 版
印 次: 2001 年 2 月第 1 次印刷
印 数: 1-5000
定 价: 79.00 元

内 容 简 介

C++ Builder 提供了一个面向对象、可视化设计的快速应用程序开发环境，使用 C++ Builder 可以用最少的程序代码撰写各类 Windows 应用程序。本书《精通 C++ Builder 5 程序设计高级教程》是一本阐述 C++ Builder 5 最完整的工具书，全书共分 15 章，从可视化设计的整合开发环境入笔，对 C++ Builder 5 的重要功能作了全面介绍，另外，由于数据库的普遍应用，本书着重对数据库程序设计原理、数据库程序设计基础与进阶、数据库维护辅助工具及数据库应用程序设计进行了重点讲解。全书讲解清晰，并辅以程序实例加以说明，是一本极具参考价值的工具书。

目 录

第 1 章 可视化设计的整合开发环境 (IDE)

1-1 程序编辑器 (Code Editor)	2
1-2 程序检索器 (Code Explorer)	7
1-3 表格 (Form)	8
1-4 元件模板 (Component Palette)	9
1-5 对象检视器 (Object Inspector)	12
1-6 快速功能按钮 (Speed Bar)	14
1-7 鼠标右键菜单 (Popup Menus)	14
1-8 除错器 (Debugger)	15
1-9 所有工具视窗皆可 Dockable	29
1-10 在线帮助说明 (On Line Help)	32
1-11 工程程序结构	34
1-12 物体宝库的应用	60

第 2 章 C++ Builder 的基本程序设计原理

2-1 对象的基本观念	65
2-2 C++ Builder 提供的对象	67
2-3 元件的继承	72
2-4 元件的有效范围	76
2-5 建立非可视化元件	80
2-6 文字输入控制元件	81
2-7 选项功能控制元件	91
2-8 信息驱动操作方式	98
2-9 读取鼠标信息	102
2-10 视窗鼠标拖-放程序的撰写	105
2-11 读取键盘信息	108
2-12 读取对象焦点信息	111

第3章 视窗程序的选单设计

3-1 视窗程序的选单设计种类	117
3-2 C++ Builder 提供的选单元件	118
3-3 下拉式选单及右键菜单的设计	119
3-4 Button 和 BitBtn 选单的设计	126
3-5 多页次视窗的设计	128
3-6 如何设计如 Office 97 一般的选单界面	132
3-7 可视化的 VCL 元件皆支持 dock 功能	134
3-8 Action List 元件	138

第4章 视窗与视窗之间的关联与视窗的类别

4-1 什么是视窗	147
4-2 VCL 提供的视窗类别	148
4-3 视窗与视窗之间的关联	149
4-4 动态产生视窗对象	157
4-5 MDI 应用程序的设计	160
4-6 程序 LOGO 视窗的设计	171
4-7 提示信息视窗的应用	175
4-8 标准对话框视窗的应用	182
4-9 可视化的视窗继承	192

第5章 Windows 标准界面元件应用及 C++ Builder 的特殊语法

5-1 何谓 Windows 标准界面元件	197
5-2 图形串列元件 (ImageList)	198
5-3 树状检视元件 (TreeView)	202
5-4 表单检视元件 (ListView)	208
5-5 其他 Win32 控制元件	212
5-6 C++ Builder 的特殊语法	221

第6章 数据库程序设计观念

6-1 客户 / 服务器架构的实际内涵	241
6-2 C++ Builder 的 Two-Tier 客户 / 服务器架构精髓	245

6-3	Two-Tier 数据库程序设计原理	246
6-4	C++ Builder 的 Multi-Tier 架构	248
6-5	Multi-Tier 数据库程序设计原理	254
第 7 章	C++ Builder 与数据库服务器的连接设定	
7-1	数据库连线原理说明	259
7-2	Paradox & dBASE 的连线设定	260
7-3	MS-Foxpro & MS-Access 的连线设定	265
7-4	InterBase 的连线设定	269
7-5	MS-SQL 的连线设定	272
7-6	ORACLE 的连线设定	275
7-7	Informix 的连线设定	277
7-8	使用 ODBC 来连上后端数据库	281
第 8 章	数据库维护辅助工具	
8-1	建立维护数据表及索引 (Database Desktop)	285
8-2	浏览及修改数据库包含的对象 (SQL Explorer)	291
8-3	转移数据库内的数据表 (Data Migration Wizard)	293
8-4	监控查询效率 (SQL Monitor)	294
8-5	SQL 程序产生器 (Visual Query Builder)	295
8-6	图形编辑器 (Image Editor)	296
第 9 章	数据库应用程序设计基础	
9-1	建立第一个数据库应用程序	299
9-2	功能强大的字段编辑器	300
9-3	使用 TTable 元件来设计数据库维护程序	309
9-4	使用 TTable 元件来设计数据库搜索功能	317
9-5	一对多数据表的设定	324
9-6	上网权限及交易资料的控制	325
9-7	何谓 SQL	331
9-8	使用 SQL 语法的数据库程序设计方式	334
9-9	使用储存在后端数据库上的 SQL 程序 (Stored Procedure)	347

9-10	数据库控制元件的应用	354
第 10 章	数据库应用程序设计的进阶观念	
10-1	分析使用 TTable, TQuery, TStoredProc 的效率及差异	369
10-2	数据集 (DataSet) 的应用	377
10-3	TDatabase 的 Isolation Levels	380
10-4	BDE 的 SQL PASSTHRU MODE 参数的重要性	382
10-5	同时存取异质性后端数据库的数据表	385
10-6	数据库程序的错误信息管理	389
10-7	利用 CachedUpdates 功能和 TUpdateSQL 元件来更新多个数据表 产生的查询结果	395
10-8	如何把后端数据库的数据表转成文本文件	400
10-9	统计图表与数据库的结合	404
10-10	商业决策分析应用程序设计	419
10-11	C++Builder 的报表设计方式	429
第 11 章	多人群组开发环境的建置	
11-1	C++Builder 的多人群组开发环境	449
11-2	对象宝库 (Object Repository)	450
11-3	数据字典 (Data Dictionary)	457
11-4	数据模块 (Data Module)	465
第 12 章	Multi-Tier 数据库应用程序设计基础	
12-1	在开始撰写 Multi-Tier 应用程序之前	471
12-2	撰写 Multi-Tier 的基本数据维护程序	479
12-3	如何在 Multi-Tier 的程序中进行 Transaction	483
12-4	如何把 SQL 命令从前端程序传给应用程序服务器执行	496
12-5	如何把查询参数传给应用程序服务器上的 TQuery	511
第 13 章	Multi-Tier 应用程序设计的进阶技巧	
13-1	Single Instance 及 Multiple Instance 的差异	519
13-2	Multi-Tier 的错误处理机制	521
13-3	如何利用 Package 功能来达到 Thin Client 的目的	527

13-4	前端程序与应用程序服务器之间的数据传递	534
13-5	可以让你在前端设定 master/detail 的关联	534
13-6	利用 Briefcase 功能来达到 Mobile Client 的目的	539
13-7	TClientDataSet 的 Aggregate 功能	544
13-8	控制前端用户权限	548
13-9	新的 TDataSetProvider	557
13-10	支持 MTS (Microsoft Transaction Server)	562
13-11	支持 CORBA (Common Object Request Broker Architecture)	582
13-12	支持 MIDAS (Multi-tier Distributed Applications Services Suite)	587
第 14 章	撰写互联网应用程序	
14-1	C++ Builder 开发 Web 应用程序的基本原理	603
14-2	撰写第一个 Web 应用程序	605
14-3	数据输入表格的 CGI 应用程序	611
14-4	查询数据库的 CGI 应用程序	616
14-5	ISAPI 及 NSAPI 应用程序的撰写	626
14-6	撰写第一个 ActiveX 应用程序	632
14-7	在 ActiveX 程序中存取远程数据库	637
14-8	如何 Deploy 开发完成的 ActiveX 应用程序	640
14-9	如何在互联网上实作 ActiveX	643
14-10	把 Package 技术应用在 ActiveX 上	645
14-11	利用 TsimpleObjectBroker 来做到 LoadBalancing 及 Fail Over	648
第 15 章	Windows 程序设计进阶技巧	
15-1	在 C++ Builder 的应用程序中调用 Win32 API	653
15-2	撰写及调用 DLL 程序	664
15-3	多国语言的程序开发功能	679
15-4	撰写应用程序的 On Line Help	686
15-5	开发 Windows NT Service 程序	697
15-6	图形处理	703
15-7	多媒体系统的开发	719

15-8 文件处理	721
15-9 如何在 C++ Builder 中控制 Microsoft Office	725
15-10 利用 C++ Builder 撰写 OLE Server	730
15-11 包装 C++ Builder 开发的应用程序	737

可视化设计的整合开发环境 (IDE)

C++ Builder 提供了一个面向对象、可视化设计的快速应用程序开发环境,就是所谓的RAD(Rapid Application Development),这种RAD环境主要是让你能以撰写最少的程序代码来建立高执行效率的Windows应用程序(32位)。C++ Builder 包含的所有RAD开发工具都是整合在IDE(Integrated Development Environment)内,图1-1就是IDE的操作环境,你可以利用这个整合开发环境来帮助你完成整个应用程序的设计,而不需要再依靠其它零碎的工具程序,让你的程序开发环境能够简单一致,提升你整体的工程开发效率。

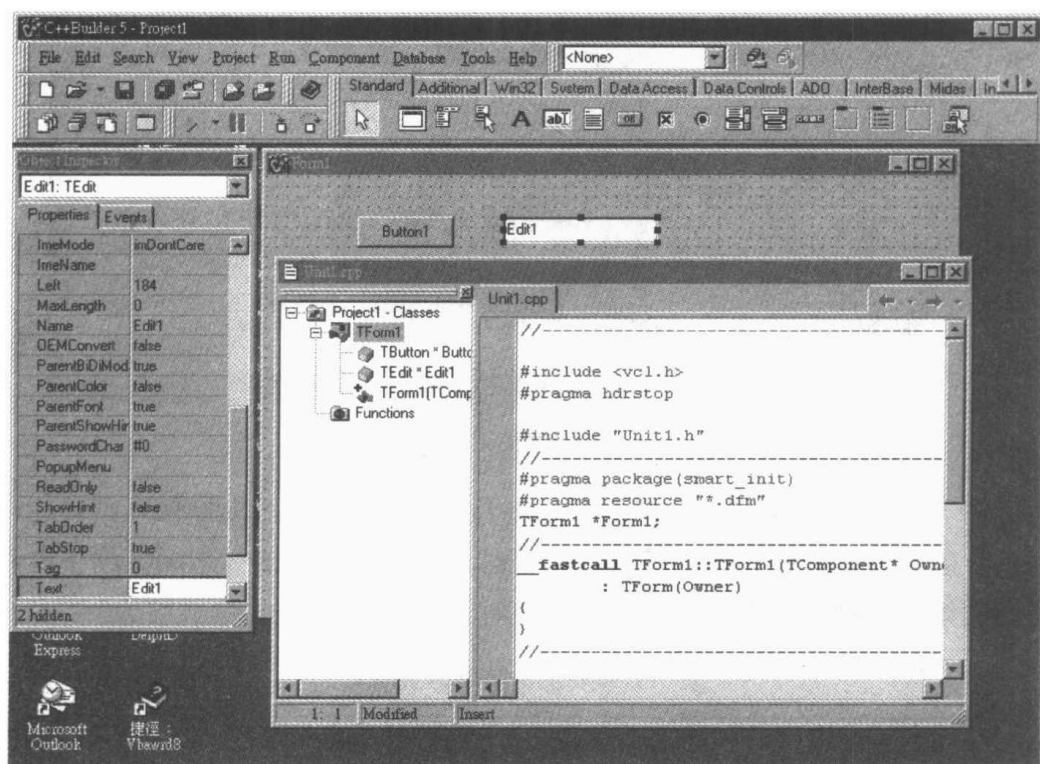


图 1-1 整合开发环境(IDE)

基本上，C++ Builder 的 IDE 环境包含有一些要素，分别是程序编辑器(Code Editor)、程序检索器(Code Explorer)、表格(Form)、元件模板(Component Palette)、对象检视器(Object Inspector)、鼠标右键菜单 (Popup Menus)、对象浏览器(Object Browser)、工程管理员(Project Manager)、快速功能按钮(Speed Bar)、除错器(Debugger)、在线帮助说明(On Line Help)。下面将针对 IDE 的每一个组成要素做一番说明：

1-1 程序编辑器(Code Editor)

这是一个具有完整功能的程序编辑器，你可以在上面同时编辑好几个原程序文件，它会将关键字(keyword)以粗体来表示，或是以不同的字体颜色来区别代表不同意义的程序区块(参考图 1-2)。你可以通过主选单的[File]选项或是快速功能按钮，来维护编辑器内的原程序代码(打开旧文件、存文件、另存新文件等)。



图 1-2 程序编辑器功能

它也提供一整组的编辑功能键(请参考表 1-1)，可以让你熟练地操作编辑功能，增加程序撰写效率，因为许多程序设计师在撰写程序时，不习惯操作鼠标来编辑程序内容，所以这些编辑用的组合按键对于资深的程序设计师而言，应该非常重要。

表 1-1 编辑器的功能键说明表

功能键	意 义
Ctrl+Left	将光标移到字的最左边字符
Ctrl+Right	将光标移到字的最右边字符

功能键	意 义
Ctrl+End	将光标移到程序的结尾
Ctrl+K,B	设定标示区块的开头
Ctrl+K,K	设定标示区块的结尾
Ctrl+K,R	读取某个文件插入程序中
Ctrl+K,W	将标示区块的程序代码写入文件中
Ctrl+K,P	将标示区块的程序代码输出到打印机
Ctrl+K,I	将标示区块的程序代码往后缩一格空白
Ctrl+K,U	将标示区块的程序代码往前推一格空白
Ctrl+K,C	复制标示区块的程序代码(没有用到系统的剪贴板)
Ctrl+K,V	复制标示区块的程序代码(没有用到系统的剪贴板)
Ctrl+C	复制标示区块的程序代码到系统的剪贴板
Ctrl+V	将剪贴板内的资料复制到编辑器
Ctrl+X	复制标示区块的程序代码到系统的剪贴板, 并且删除原标示区块的程序代码
Del	删除光标所在的字符或是标示区块的程序代码
Ctrl-K,0,1,2,3.....9	设定标签(每页程序最多 10 个)
Ctrl-Q,1,2,3,4.....9	将光标移到标签处
Ctrl-F	搜索字符串
Ctrl-R	搜索并取代字符串

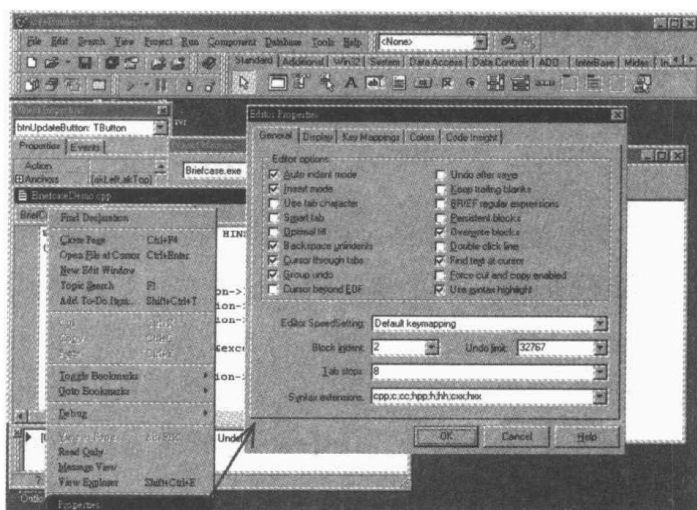


图 1-3 编辑器的参数设定

当然,你也可以自己定义程序编辑器的一些设定值,如字体显示的颜色、粗细、大小,或编辑器附带的一些功能。你只要点选右键菜单的[Properties]选项, C++ Builder 将会弹出程序编辑器的参数设定视窗让你修改(请参考图 1-3),整个参数选项共分成三类,分别放在 Editor、Display、Colors 三个页次内,由于选项太多,限于篇幅缘故,所以无法一一列举,请自行测试。

在 C++ Builder 中具有一项 Code Insight 的功能,这个 Code Insight 功能又分有四个子功能,分别是 Code templates、Code completion、Code parameters、Tool-tip expression evaluation 等,下面是这些子功能的使用说明:

1. **Code templates:** 这个功能会把 Object C++语法中,所有的命令语句列在一个下拉式的小视窗内(如 CASE、WHILE、FOR 等命令),帮助程序设计师在忘记 Object C++ 的命令语法时,可以不用再翻阅手册,直接在程序编辑器中得到帮助。你可以按下 [CTRL]+J 键来启动 Code templates,当你启动这项功能之后,你可以直接用鼠标去点选想要取得的命令格式,或是用键盘输入命令名称,再按下 [Enter] 键来取得命令格式(请参考图 1-4)。

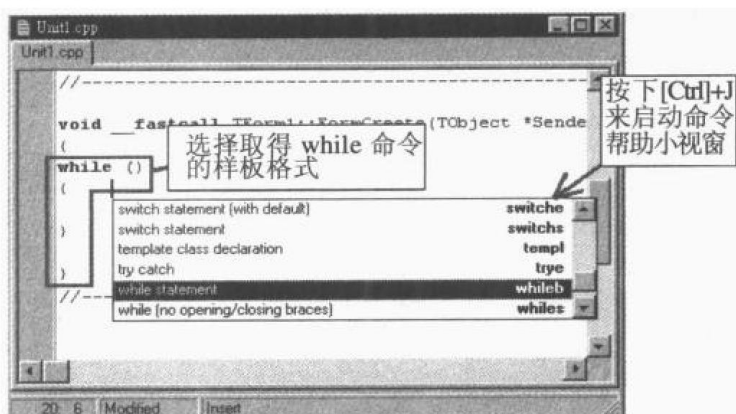


图 1-4 编辑器的 Code templates 功能

2. **Code completion:** 在你撰写程序的过程当中,有可能声明许多变量、对象、函数、程序等,而当你想键入这些声明的资料时,如果忘记了这些声明资料的名称,你还必须翻到最前面去把声明资料的名称复制到下面来,当这些声明资料多达上百个时,你就会到感觉头昏脑胀、两眼昏花,不知道想要的资料在那里,而这项功能将可以帮助你解决这个困难。你可以按下 [Ctrl]+space 键来启动 Code completion 视窗,此视窗会列出你目前所在 Unit 内声明的所有变量、对象、函数、程序名称,帮助你取得这些又臭又长的变量名称(请参考图 1-5)。

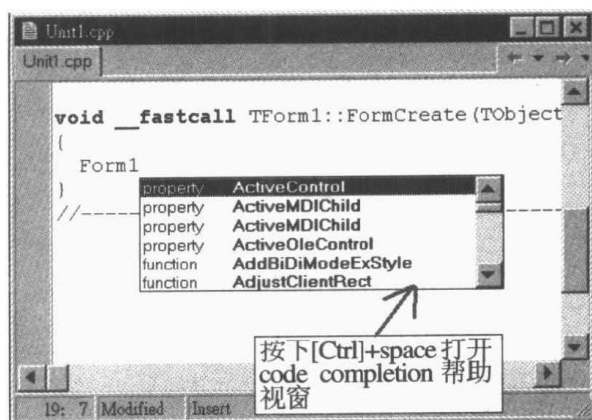


图 1-5 编辑器的 Code completion 功能

如果你输入的是对象名称，那么你一定接着会输入此对象的某个属性或方法，此时你只要在对象名称后面键入逗号，不到两秒钟的时间 Code completion 视窗将会把此对象包含的所有属性及方法找出来列在视窗内，你可以按下[Enter]键或用鼠标点选来取得需要的属性或方法名称，请参考图 1-6。

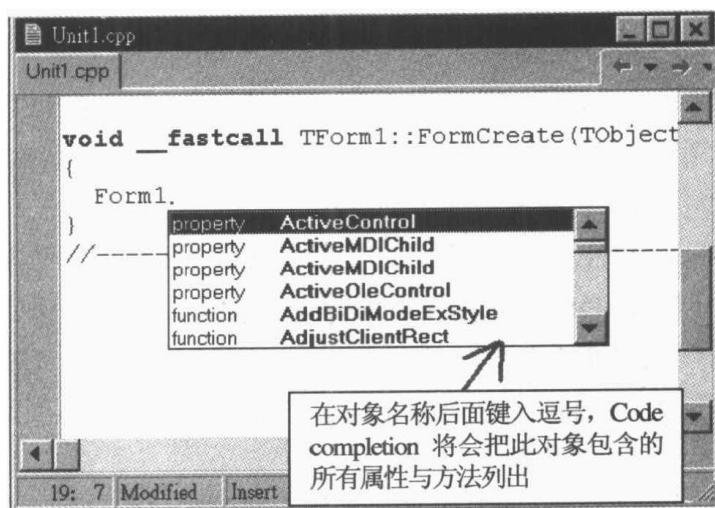


图 1-6 编辑器的 Code completion 功能

3. Code parameters: 这项功能是与 Code completion 互相搭配的，当你输入对象名称之后，可以启动 Code completion 来取得对象的方法名称，可是这些对象方法都是函数或程序格式，因此你还必须知道它们包含有哪几个参数及参数的数据类型为何。一般传统的做法是翻阅使用手册或直接查询 On Line Help，在 C++ Builder 中你可以在对象方法后面键入“左括号”来启动 Code parameters，它将会把查询到的参数格式以 Hint 的方式显示出来，请你参考图 1-7。

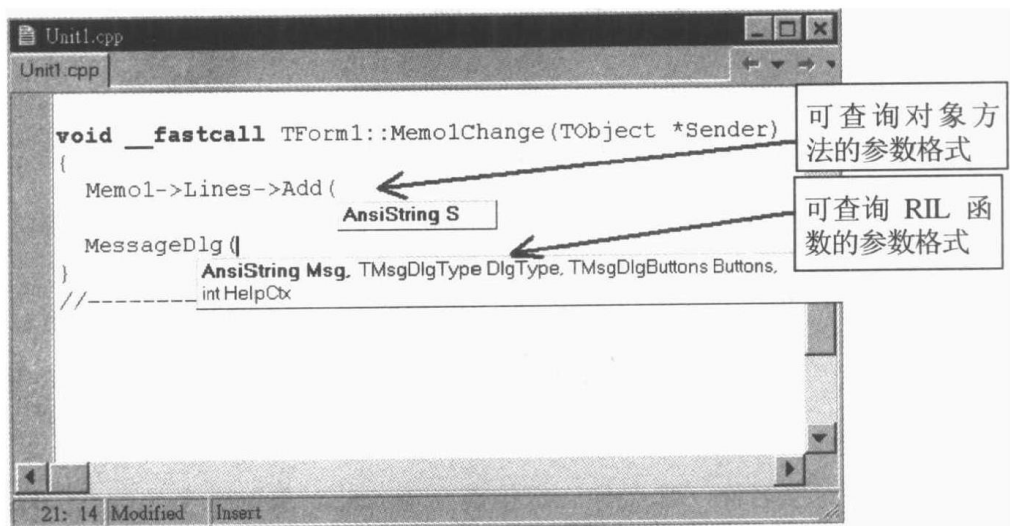


图 1-7 编辑器的 Code parameters 功能

Code parameters 不但可以查询到对象方法的参数格式，还有执行时期函数库(RTL)的函数及程序、自行定义的函数及程序、对象产生的事件程序等等，Code parameters 都可以查到这些函数或程序的参数格式。

4. Tool-tip expression evaluation: 这项功能应该是 Code Insight 四项功能里，最厉害的一项。传统在程序设计除错时，你必须把要被除错的变量名称加入到 Watch 视窗，然后一步一步执行来观察变量内容的变化。可是这种除错方式具有两项缺失，第一、Watch 视窗可以一次显示在屏幕上的变量个数是有限的，当你加入好几十个变量到 Watch 视窗内时，你必须用鼠标慢慢地翻页寻找想要观察的变量；第二、有些 Local 的变量在某个程序下是有效的，当跳出此程序后，这些 Local 变量就变成无效，你必须删除它们，另外再加上目前所在程序的 Local 变量，不断地重复这些工作，增加了不少除错时的工作负担。

Tool-tip expression evaluation 就是用来解决你上述的除错问题。你只要在程序除错状态下，将鼠标指标指向想要观察的变量，即可马上启动了 Tool-tip expression evaluation 的功能，它将会把此变量目前的值以 Hint 的方式显示出来，请你参考图 1-8。

当然有了 Tool-tip expression evaluation 功能，并不代表你就可以完全放弃原先使用的除错工具，如图 1-8 中的数组变量，如果你想知道第 52 个数组节点的值，你不可能在上面数到第 52 个来查看变量的内容，这种状况还是必须借助传统的除错工具来帮助你除错。因此，Tool-tip expression evaluation 是用来更加强补足 C++ Builder 的除错功能，而不是用来取代旧有的除错功能。本章的 1-8 节将会针对 C++ Builder 提供的所有除错功能一一说明。

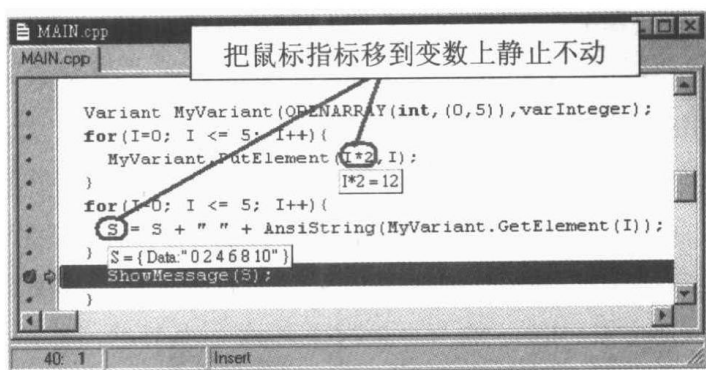


图 1-8 编辑器的 Tool-tip expression evaluation 功能

1-2 程序检索器(Code Explorer)

当你开启 C++ Builder 时, 你是否发现程序编辑器(Code Editor)左半边有一个树状结构的视窗, 没错! 这个视窗就是 C++ Builder 的 Code Explorer 工具视窗(如图 1-9 所示)。

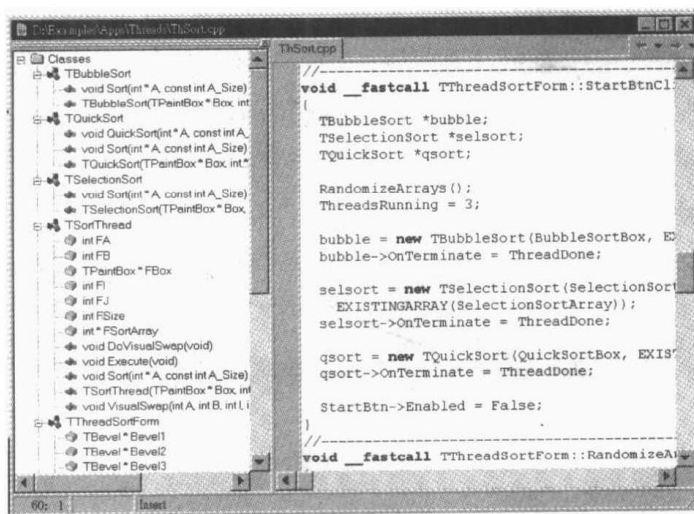


图 1-9

Code Explorer 对于 C++ Builder 的程序设计师而言, 是一项早就该提供的功能, 由于 Windows 的程序写法属于事件驱动的方式, 也就是说大部分的程序代码都会写在对象的事件内, 若是遇到程序要修改时, 你就必须先 Focus 到元件上, 然后打开 Object Inspector, 找到想要修改的事件程序, 双击就会找到事件程序的程序段落, 若是包含的对象不多那还好, 如果遇到程序很大或是对象又很多时, 光是找程序就晕头转向了。有了 Code Explorer, 这些问题都将迎刃而解。