



普通高等教育“十一五”国家级规划教材 计算机系列教材

Qt C++跨平台图形界面 程序设计基础

殷立峰 主 编
房志峰 副主编

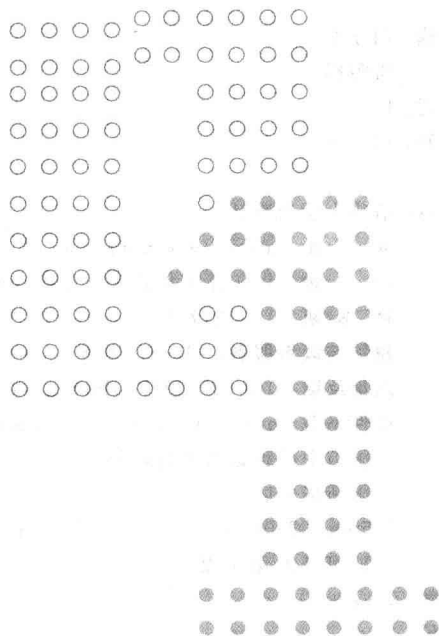


清华大学出版社

计算机系列教材

殷立峰 主编
祁淑霞 房志峰 副主编

Qt C++跨平台图形界面 程序设计基础



清华大学出版社
北京

内 容 简 介

本书是为高等学校面向应用型人才培养编写的 C++ 语言程序设计教材。全书共分 11 章,主要内容包 括 Qt 开发环境介绍、C++ 程序设计基础、类与对象、继承与派生、虚函数与多态、运算符重载、模板和异常处理、输入输出流与命名空间、图形界面程序设计基础、对话框编程、使用 Qt 开发文本编辑器。本书采用 Qt 跨平台 C++ 程序开发框架,结合“案例驱动”编写方式,语法介绍精练,内容深入浅出、循序渐进,程序案例生动易懂,既传授给学生 C++ 语言基本概念和知识,又传授给学生使用 C++ 语言进行跨平台程序设计和图形界面程序设计的基本方法及基本技能。

本书既可以作为高等院校本科及专科 C++ 语言程序设计的教材,又可以作为教师、自学者的参考用书,同时也可以供各类软件开发设计人员学习参考。

本书配有电子教案及相关教学资源,读者可以从网站 www.tup.com.cn 下载。

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

Qt C++跨平台图形界面程序设计基础/殷立峰主编. —北京:清华大学出版社,2014

计算机系列教材

ISBN 978-7-302-34936-5

I. ①Q… II. ①殷… III. ①C 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(2013)第 121325 号



责任编辑:白立军

封面设计:常雪影

责任校对:白蕾

责任印制:杨艳

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:010-62786514

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:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:21

字 数:483 千字

版 次:2014 年 1 月第 1 版

印 次:2014 年 1 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:056543-01

C++ 是优秀的计算机程序设计语言,它的程序设计功能非常强大,我国绝大多数高等院校都把它作为程序设计入门教学的首选。许多经典的 C++ 语言程序设计教材都是基于 Visual C++ 控制台程序设计框架编写,内容主要包括 C++ 语言概述、基本数据类型、运算符和表达式、程序流程控制、数组、函数、指针、结构体、共用体与枚举、类和对象、运算符重载、继承、虚函数与多态、模板、流等 C++ 语言的基本概念和基本知识。这类教材强调的是培养学生理解和掌握 C++ 语言的语法及逻辑规则,强调对 C++ 程序设计语言的字、词、数据、表达式、语句,到函数、类、对象等基本概念知识的掌握。然而多年的教学实践发现,这类教材缺乏对图形界面 C++ 程序设计的介绍,学生要想掌握用 C++ 语言设计图形界面的程序,还必须自学或者参加 Visual C++ 语言课程的学习,而 Visual C++ 语言体系庞大,学起来有一定的困难,造成很多学生学了 C++ 语言,却不能很好地使用它。

本书采用目前广为流行的跨平台 C++ 图形界面应用程序开发框架 Qt 编写,Qt C++ 具有跨操作系统平台强、简单易学的优点。通过本书的学习,不但使学生掌握 C++ 语言的基本概念、基本知识和基本的程序设计方法,更重要的是培养学生具备初步的跨操作系统平台图形界面程序设计的基本技能。本书具备如下特色。

(1) 面向市场应用型人才需求现状,依据 C++ 语言程序设计教学大纲要求,组织编写教学内容,具有很强的针对性。

(2) 把跨操作系统平台程序设计、图形界面程序设计和面向对象的程序设计与 C++ 语言的基本概念和基本知识有机结合,不但传授给学生 C++ 语言的基本概念和基本知识,而且使学生掌握利用 C++ 程序设计语言进行图形界面程序设计开发和跨操作系统平台进行程序设计开发的基本技能。

(3) 采用先提出问题、然后分析解决问题的教学方法,有效提高学生的学习兴趣,引导学生学会利用所学知识分析解决实际问题。

(4) 课时安排恰当,即使没有 C/C++ 语言程序设计基础,也可以通过本书的学习,循序渐进、由浅入深地掌握 C++ 程序设计语言的概念、语法、面向对象程序设计的方法和跨平台图形界面程序设计技能。

(5) 以跨 Windows、Linux 平台,基于 Qt 的 C++ 语言编程为框架,通过案例驱动教学,内容精练、结构紧凑,通俗易懂、重点突出,注重实用和能力的培养,克服了一般 C++ 语言程序设计教科书中学习 C++ 枯燥的缺点,通过生动有趣的案例,激发学生学习兴趣,

让学生自发地喜欢 C++ 程序设计语言,培养学生的程序设计技巧和使用 C++ 程序设计语言分析解决实际问题的动手能力。

由于作者水平有限,书中不足之处在所难免,敬请读者批评指正。

编 者

2013-11-18

F O R E W O R D

第 1 章 走进 Qt	/1
1.1 Qt 简介	/1
1.1.1 认识 Qt	/1
1.1.2 Qt 开发环境的主要构成介绍	/2
1.1.3 使用 Qt 开发 C++ 应用程序的优势	/4
1.2 Qt 的下载、安装与配置	/4
1.2.1 Windows 平台下 Qt 的 C++ 语言开发环境安装与配置	/4
1.2.2 Linux 平台下 Qt 的 C++ 语言开发环境安装与配置	/7
1.3 Qt Creator 集成开发环境	/12
1.3.1 Qt Creator 集成开发环境	/12
1.3.2 Qt Creator 常用菜单功能介绍	/13
1.3.3 使用 Qt 创建项目	/14
1.3.4 Qt 开发环境的使用方法	/18
1.3.5 Qt 项目文件的建立、添加和删除	/18
1.3.6 编辑项目的源程序文件和界面文件	/19
1.3.7 项目编译模式及其配置	/20
1.3.8 编译并链接生成项目文件	/20
1.3.9 纠正编译或连接出现的错误	/20
1.3.10 Qt 工具栏的使用	/21
1.4 Qt Creator 联机帮助系统及其使用	/21
1.4.1 Qt 中如何寻求帮助	/21
1.4.2 帮助文件的打开及使用源代码编辑器	/22
1.4.3 缩小查找范围	/23
1.5 使用 Qt Creator 开发 C++ 语言程序	/24
1.5.1 Windows 平台下使用 Qt 开发 C++ 语言程序	/24
1.5.2 Linux 平台下使用 Qt 开发 C++ 语言程序	/29
1.6 习题	/36

第 2 章 C++ 程序设计基础	/37
2.1 C++ 语言简介	/37
2.1.1 认识 C++	/37
2.1.2 C++ 的标准化	/37
2.2 C++ 源程序的结构	/38
2.2.1 C++ 源程序举例	/38
2.2.2 C++ 源程序的结构	/40
2.2.3 C++ 语言的基本语法成分	/41
2.3 基本数据与表达式	/42
2.3.1 数据类型	/42
2.3.2 常量和变量	/44
2.3.3 运算符与表达式	/46
2.4 C++ 中的输入输出	/47
2.5 程序的控制结构	/50
2.5.1 顺序结构	/50
2.5.2 选择结构	/50
2.5.3 循环结构	/55
2.5.4 跳转语句	/58
2.6 函数	/59
2.6.1 函数	/59
2.6.2 函数的其他特性	/62
2.7 数组与字符串	/65
2.7.1 数组	/65
2.7.2 字符串与 string 类	/67
2.8 指针与引用	/70
2.8.1 指针	/70
2.8.2 引用	/71
2.9 const 修饰符	/76
2.10 动态内存分配	/79
2.11 习题	/82
2.11.1 选择题	/82
2.11.2 填空题	/83

2.11.3 编程题 /84

第3章 类与对象 /85

- 3.1 面向对象程序设计概述 /85
 - 3.1.1 面向对象的基本概念 /85
 - 3.1.2 面向对象的基本特征 /88
 - 3.1.3 面向对象的语言简介 /89
- 3.2 类与对象的定义 /90
 - 3.2.1 类的定义 /90
 - 3.2.2 对象的定义与使用 /97
 - 3.2.3 类的作用域 /101
 - 3.2.4 类的封装性和信息隐藏——公有接口与私有实现的分离 /102
- 3.3 构造函数与析构函数 /104
 - 3.3.1 构造函数 /104
 - 3.3.2 复制构造函数 /113
 - 3.3.3 析构函数 /118
- 3.4 对象的深复制 /120
- 3.5 静态成员 /122
 - 3.5.1 静态数据成员 /122
 - 3.5.2 静态成员函数 /124
- 3.6 常类型 /127
 - 3.6.1 常对象 /127
 - 3.6.2 类的常数据成员 /128
 - 3.6.3 类的常成员函数 /129
- 3.7 友元 /131
 - 3.7.1 友元函数 /131
 - 3.7.2 友元类 /136
 - 3.7.3 友元应用举例 /138
- 3.8 对象数组与类的组合 /140
 - 3.8.1 对象数组 /140
 - 3.8.2 类的组合 /144

3.9	程序举例	/147
3.10	习题	/149
3.10.1	选择题	/149
3.10.2	问答及编程题	/152
第4章 继承与派生 /154		
4.1	单继承	/155
4.1.1	继承的定义	/155
4.1.2	访问控制	/157
4.1.3	重名的成员变量和成员函数	/160
4.1.4	在派生类中访问静态成员	/162
4.1.5	基类的初始化	/163
4.2	多继承	/168
4.2.1	派生类的构造与访问	/168
4.2.2	虚继承	/169
4.3	习题	/170
第5章 虚函数与多态 /171		
5.1	类指针的关系	/172
5.2	静态联编和动态联编	/174
5.3	虚函数	/175
5.4	纯虚函数和抽象类	/178
5.5	习题	/181
第6章 运算符重载 /182		
6.1	运算符重载概述	/183
6.1.1	运算符重载的实质	/183
6.1.2	用友元函数和成员函数重载运算符的 异同	/186
6.1.3	++和--运算符的重载	/188
6.2	习题	/191

第 7 章 模板和异常处理	/192
7.1 模板的概念	/192
7.2 函数模板	/192
7.2.1 函数模板的声明	/192
7.2.2 函数模板的实例化	/193
7.2.3 函数模板应用举例	/195
7.3 类模板	/197
7.3.1 类模板的定义	/197
7.3.2 类模板的实例化	/198
7.3.3 类模板的应用举例	/199
7.4 标准模板库	/202
7.4.1 容器	/203
7.4.2 算法	/206
7.4.3 迭代器	/209
7.5 异常处理	/210
7.5.1 异常处理概述	/210
7.5.2 异常处理的实现	/211
7.5.3 标准库中的异常类型	/216
7.6 习题	/218
7.6.1 选择题	/218
7.6.2 编程题	/218
第 8 章 输入输出流与命名空间	/220
8.1 I/O 流的概念	/220
8.2 标准 I/O 流	/221
8.2.1 标准 I/O 流概述	/221
8.2.2 标准输出	/222
8.2.3 标准输入	/224
8.2.4 重载插入/提取函数	/225
8.3 格式控制	/228
8.3.1 用 ios 成员函数格式化	/228
8.3.2 用操纵算子格式化	/231

8.4	文件处理	/232
8.4.1	文件和流	/232
8.4.2	文件的打开和关闭	/233
8.4.3	文本文件	/234
8.4.4	二进制文件	/235
8.5	命名空间	/238
8.6	习题	/241
第9章 图形界面程序设计基础 /242		
9.1	图形界面程序设计基础知识	/242
9.1.1	C++ 中的对象	/242
9.1.2	Qt C++ 中的窗体	/243
9.1.3	Qt C++ 中的部件和部件类	/243
9.1.4	Qt C++ 中的属性(Properties)窗口	/244
9.2	Qt 的信号和槽	/244
9.3	Qt 的元对象系统	/246
9.4	Qt 命令行法开发图形界面程序	/246
9.5	Qt 中如何实现用户操作的响应	/249
9.6	Qt 中如何实现窗口部件的布局	/250
9.7	习题	/252
第10章 对话框编程 /254		
10.1	代码编程创建对话框	/254
10.2	对话框的可视化设计	/263
10.3	可扩展的对话框	/271
10.4	对话框的动态实现	/279
10.5	Qt 内置的窗口部件和对话框类	/280
10.6	习题	/285
第11章 使用 Qt 开发文本编辑器 /286		
11.1	Qt Creator 的下载和安装	/286
11.2	Qt Creator 开发简单的文本编辑器	/287

11.2.1	创建项目 TextEditor	/287
11.2.2	TextEditor 的主窗口、菜单和图标	/289
11.2.3	TextEditor 文件新建、保存和另存为的功能实现	/300
11.2.4	TextEditor 文件的打开、关闭和退出系统的功能实现	/308
11.2.5	TextEditor 文本复制、剪切、粘贴以及撤销的功能实现	/309
11.2.6	TextEditor 文本查找功能的实现	/310
11.2.7	TextEditor 查找和定位函数的方法	/311
11.2.8	TextEditor 中实现状态栏	/316
11.3	习题	/318
附录 A	C++ 关键字	/319
附录 B	C++ 运算符	/320
参考文献		/321

第 1 章 走进 Qt

本章主要内容

- (1) Qt 概要。
- (2) Qt 的下载和安装。
- (3) Qt 开发环境的安装配置。
- (4) Qt 开发环境的使用。
- (5) Qt 控制台程序设计方法。
- (6) Qt 图形界面程序设计方法。

1.1 Qt 简介

1.1.1 认识 Qt

Qt 是 1994 年成立的总部位于挪威奥斯陆的奇趣科技公司(Trolltech)提供的跨平台 C++ 图形界面应用程序开发框架。它既可以开发 GUI 程序,也可开发非 GUI 程序,如控制台工具和服务。它是面向对象的程序开发框架,使用特殊的代码生成扩展(称为元对象编译器(Meta Object Compiler, MOC))以及一些宏,易于扩展,允许组件编程。2008 年,奇趣科技公司被诺基亚公司收购,Qt 也因此成为诺基亚旗下的编程语言工具。2012 年 Qt 被 Digia(总部位于芬兰的 IT 业务供应商)收购。

Qt 使用“一次编写,随处编译”的方式为开发者提供了允许程序开发人员使用 C++ 语言单一源码来构建可以运行在不同平台下的应用程序的不同版本;这些平台包括从 Windows 98, Windows XP, Vista, Win8, Mac OS X, Linux Solaris, HP-UX 以及其他很多基于 X11 的 UNIX。与此同时,作为 Qt 组成部分之一的 Qt/Embedded Linux, 也为嵌入式系统的开发人员搭建了一套完善的窗口系统和开发平台。Qt 单一源程序的多平台兼容性、代码可重用性、丰富的 C++ 方面的性能、高质量的技术支持等特点,使其深受广大 C++ 语言程序员的推崇与爱戴,是时下广为流行的 C++ 语言应用程序开发工具之一。

Qt 具有广泛适应性及良好的可移植性,编写过的 C++ 语言代码,只需在其他不同的操作系统平台中重新编译一遍,即可重复使用。这特别适合于客户要求应用程序能同时运行于不同平台的情况。此外,使用开源许可协议也可以获得 Qt。开源许可协议就是指自由软件、开源软件是自由的、免费的、源代码是开放的,人们可自由下载安装和使用。同时,为了维护作者和贡献者的合法权利,保证这些软件不被一些商业机构或个人窃取,影响软件的发展,开源社区开发出了各种开源(能得到源代码是前提)许可协议,如以任何目的运行此程序的自由;以学习程序工作机制为目的,对程序进行修改的自由;再发行复制件的自由;改进此程序,并公开发布改进的自由等。对于一名开源程序开发人员,从 Qt

那里将获益无穷。

Qt 具有友好的在线帮助文档系统。通过在线文档的帮助,只需轻点鼠标或者简单敲击几下键盘,就可以轻易制作出简单的“Hello World”欢迎对话框,或者甚至是功能更为强大的软件系统。这一点,在众多的软件帮助文档系统中并不多见。然而,帮助文档系统毕竟是为用户提供实用的类库参考为主要目的的,也就是说,它主要是为用户提供准确的“可以如何做”的信息。这对于喜欢举一反三的程序设计人员来说,显然远远不够,因为他们更喜欢知晓为什么要这么做。

Qt 作为一个著名的跨平台程序开发框架,拥有直观、强大的 API(应用程序接口),很多公司更愿意把 Qt 用于单一平台的软件开发上。Adobe PhotoShop Album 就是用 Qt 编写的面向大众市场的 Windows 应用程序的一个例子。市场中很多功能完善的软件系统,如三维动画工具、数字电影处理软件、自动化电路设计系统(用于芯片设计)、油气资源勘探、金融服务以及医学成像等,都可以基于 Qt 构建而成。基于 Qt 编写的 Windows 应用软件产品,在不使用 native API 前提下,只需通过重新编译,就可以轻松地在 Mac OS X 和 Linux 世界中开拓出新的市场。

可以基于多种许可协议获得 Qt 的使用权。Qt 有商业版和开源版两大类版本,其中商业版主要用于商业软件开发。它们提供传统商业软件发行版并且提供在协议有效期内的免费升级和技术支持服务。如果想构建商业应用程序,那么从奇趣科技公司购买一个 Qt 的商业许可协议或者使用 LGPL 协议。但如果只想构建一些开源程序,那么就可以使用基于 GPL 的 Qt 开源版本。KDE 是 K 桌面环境(Kool Desktop Environment)的缩写,是一种著名的在 UNIX、Linux 以及 FreeBSD 等操作系统上面运行的自由图形工作环境,它采用的就是 Trolltech 公司所开发的 Qt 程序库。KDE 和 Gnome 都是 Linux 操作系统上最流行的桌面环境系统。如今蓬勃发展的 KDE 桌面环境和丰富强大的 Qt 开发功能,进一步展示了 Qt 的无限发展潜力和令人期待的远景。

除了 Qt 预先定义好的数百个程序员可以拿来直接使用的类,还有很多扩展 Qt 应用范围和功能的其他软件,其中相当多的软件可由另外一些公司或者开源社区提供。对于可用的 Qt 额外软件的列表清单,可以查阅 Digia 公司的网站获得。众多开发人员也有他们自己的网站,他们会把自己写的一些用于娱乐方面的、有趣的或者是有用的非官方代码放在那里。Qt 还建立了一个维护良好并且内容丰富的用户社区,供 Qt 用户交流。

本书后续章节围绕如何使用 Qt 编写图形用户界面程序这一中心,从 C++ 语言的基本语法开始,按照循序渐进,由浅入深的原则,从手写代码的方式入手,生动、全面地阐述使用 Qt 编写 C++ 语言图形界面应用程序时所需的基本概念、设计思路和设计方法。读者通过学习这些知识就可以写出实用的图形界面应用程序。

1.1.2 Qt 开发环境的主要构成介绍

截至目前,Qt 已经由一个简单的图形工具包演变成长为具有事实标准意义的应用程序开发框架。成长为一个综合的软件开发环境,它的主要构成简要介绍如下。

(1) GCC: 是 GNU C Compiler 的英文缩写(其中 GNU 是类似 UNIX 的操作系统,

是由一系列应用程序、系统库和开发工具构成的软件集合,包括用于资源分配和硬件管理的内核),是 Linux 系统下程序的编译器。GCC 最初只是一个 C 语言编译器,随着众多自由开发者的加入和 GCC 自身的发展,如今的 GCC 已经是一个包含 C、C++、Ada、Object C 和 Java 等众多语言的编译器了。所以,GCC 也由原来的 GNU C Compiler 变为 GNU Compiler Collection。也就是 GNU 编译器家族的意思。当然,如今的 GCC 借助于它的特性,具有了在一个平台下编译另一个平台代码的交叉编译器功能。

(2) MinGW: 是 Windows 平台下的 GCC 移植版,是可以自由使用及发布的 Windows 特定头文件和使用 GNU 工具集导入库的集合,允许在 GNU/Linux 和 Windows 平台生成本地的 Windows 程序而不需要第三方 C 运行时库。MinGW 本身也存在多个分支,可以在 Windows, Linux 中编译能在 Windows 中执行的二进制代码。

(3) Qt Creator: 是 Qt 的集成开发环境,能够跨平台运行,支持的系统包括 Linux (32 位及 64 位)、Mac OS X 以及 Windows。包括项目生成向导、高级的 C++ 代码编辑器、编译器、调试器、图形设计器及类的工具、集成了 Qt Designer、Qt Assistant、Qt Linguist、图形化的 GDB 调试前端,集成 qmake 构建工具等。开发人员利用 Qt 这个应用程序框架,能更加快速及轻易地完成开发任务。

(4) Qt Designer: 是一个功能强大的 GUI 布局与窗体构造器,能够在所有支持平台上,以本地化的视图外观与认知,快速开发高性能的用户界面。它是 Qt 用来设计应用程序图形界面的工具。

(5) Qt Assistant: 又称为 Qt 助手,是一个可以完全自定义、能重新组织的帮助文件或文档浏览器,它能与基于 Qt 的应用程序一起运行。开发人员使用它能加快文档的处理过程。

(6) QT Library: 是一个拥有超过 400 个 C++ 类,同时类的数量还在不断扩展的类库。它封装了用于端到端应用程序开发所需要的所有基础结构,包括成熟的对象模型的优秀的 Qt 应用程序接口,以及内容丰富的集合类,具有图形用户界面编程、布局设计、数据库编程、网络、XML、国际化、OpenGL 等支持功能。

(7) QT Linguist: 又称为 Qt 语言家,是用来消除国际化程序设计流程中障碍的工具。借助这个工具,开发人员可把应用程序的翻译转换外包给非技术性翻译人员,从而可增加精确度,大大加快软件本地化速度。

(8) Qt dev-tools: 包含了 Qt Assistant 及 Qt Linguist 等工具,因此不需要单独安装这两个工具。

(9) Qt doc: 是帮助文档,包含 Qt 中各个类库的详细说明以及丰富的例子程序,可以使用 Qt Assistant 工具来打开阅读。

(10) Qt Qtconfig: 是配置 Qt 环境的一个对话框,一般默认就行了,很少有必要去更改。

(11) Qt demos: 是 Qt 示例程序的集合,包含很多可以运行起来的可执行文件以及源代码。帮助大家快速学习和掌握 Qt 程序开发技巧。

(12) Qt/Embedded: 是一个完整的包含 GUI 和基于 Linux 的嵌入式平台开发工具。

(13) QWT: 全称是 Qt Widgets for Technical Applications,是一个基于 LGPL 版权

协议的第三方类库,可生成各种统计图。它为具有技术专业背景的程序提供 GUI 组件和一组实用类,其目标是以基于 2D 方式的窗体部件来显示数据,数据源以数值、数组或一组浮点数等方式提供,输出方式可以是 Curves(曲线)、Slider(滚动条)、Dials(圆盘)、Compasses(仪表盘)等。该工具库基于 Qt 开发,所以也继承了 Qt 的跨平台特性。

1.1.3 使用 Qt 开发 C++ 应用程序的优势

MFC 的汉语意思是微软基础类(Microsoft Foundation Classes),同 Qt 类似,也是一种应用程序框架,它随着微软公司 Visual C++ 开发工具发布,自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后开始,到目前已经发展到版本 12.0(截至 2013 年 3 月),并且发布了中文版。该类库提供一组通用的可重用的类库供开发人员使用。是 Windows 操作环境中比较流行的 C++ 语言开发平台。

Qt 同 X Window 上的 Motif、Openwin、GTK 等图形界面库和 Windows 平台上的 MFC、OWL、VCL、ATL 是同类型的东西,但与它们相比,Qt 具有下列优点。

(1) 优良的跨平台特性。具有跨操作系统平台优势,Qt 支持 Microsoft Windows 95/98、Microsoft Windows NT、Linux、Solaris、SunOS、HP-UX、Digital UNIX(OSF/1、Tru64)、Irix、FreeBSD、BSD/OS、SCO、AIX、OS390、QNX 等操作系统环境下的 C++ 语言图形界面应用程序开发。

(2) 兼容性强。Qt 支持 C++ 语言“一次编写,随处编译”的方式,允许程序员使用 C++ 语言单一源码来构建可以运行在不同操作系统平台下的应用程序的不同版本。

(3) 面向对象的特性体现比 MFC 明显。就面向对象这一点来讲,Qt 的良好封装机制使得 Qt 的模块化程度非常高,可重用性较好,对于用户开发程序来说非常方便。

(4) 简单易学。语法结构简单清晰,Qt 通过提供一种称为信号(signals)/槽(slots)的安全类型来替代回调函数(callback),使程序各部件之间的协同工作变得十分简单。代码写起来比较优雅,简单易学。

(5) 丰富的 API 支持功能: Qt 包含了 250 个以上的 C++ 类,提供基于模板的 collections、serialization、file、I/O device、directory management、date/time 类。提供正则表达式的处理功能。支持 2D/3D 图形渲染,支持 OpenGL 大量的开发文档,支持 XML,支持 Webkit 引擎的集成,可以实现本地界面与 Web 内容的无缝集成。

1.2 Qt 的下载、安装与配置

1.2.1 Windows 平台下 Qt 的 C++ 语言开发环境安装与配置

1. 下载与安装

首先从 Qt 的开源版本网站下载开发程序,网址是 <http://Qt-project.org/downloads>。Qt 官方网站上有各种操作系统环境中使用的不同的 Qt 二进制程序安装包,既有开源的

MinGW 预编译的版本,也有给 VC2003/2005/2008 预编译的商用版本,而且一直在更新,商业版 Qt 需要在 Digia 官网下载。这里下载安装的是一个 Windows XP 环境下开源的版本: Qt-sdk-win-opensource-2010.05.exe。这是一个集成的 (IDE) 编程环境,包括 QtCreator、MinGW(gcc)、Qt Designer 等。

运行下载的程序: Qt-sdk-win-opensource-2010.05.exe,按照向导一步一步做下去就完成 Qt 开发程序的安装。安装时可以选择是否创建桌面快捷方式。安装完成后就能在开始菜单中找到相关的 Qt 开发程序。

2. 环境配置

要使用 Qt 命令行 (Qt Command Prompt) 方式编辑、编译、链接和运行 C++ 程序,必须首先进行开发环境的配置。下面以 Qt-sdk-win-opensource-2010.05 版本的 Qt 为例,介绍命令行方式开发 Qt 图形界面 C++ 语言程序的环境配置方法。

Qt 命令行方式开发 Qt 图形界面 C++ 语言程序的开发环境配置很简单。做法是将 Qt 的 bin 目录添加到环境变量中,方法是: 右击桌面上“我的电脑”图标,弹出的菜单如图 1-1 所示,在弹出的菜单中选择“属性”,弹出如图 1-2 所示的对话框;接下来在弹出的对话框中选择“高级”选项卡,弹出如图 1-3 所示的选项卡;然后单击“环境变量”按钮,弹出如图 1-4 所示的对话框。

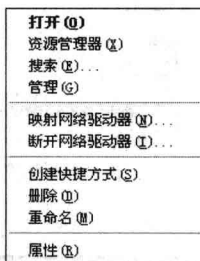


图 1-1 快捷菜单

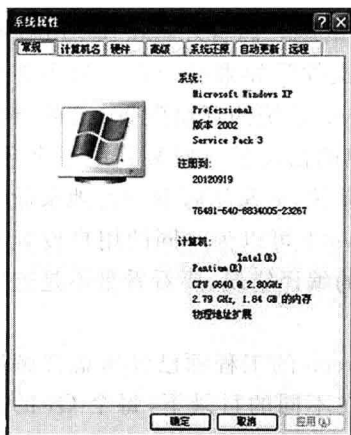


图 1-2 “系统属性”对话框

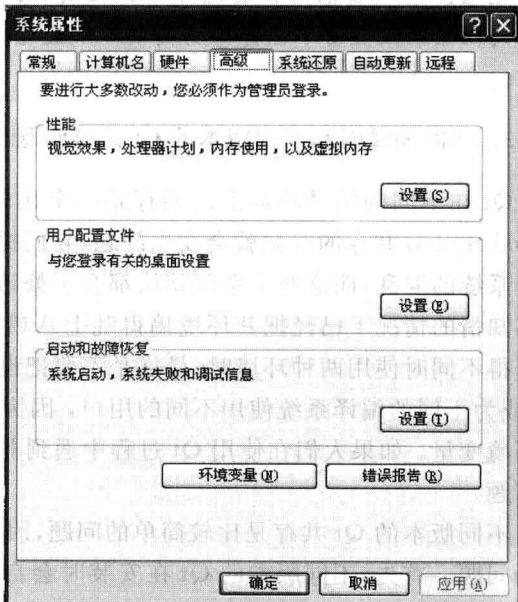


图 1-3 “高级”选项卡

然后在“系统变量”列表中单击选中名为 Path 的环境变量,然后单击“编辑”按钮,弹出如图 1-5 所示的对话框,在“变量值”单行文本输入框的后面加上英文的分号,然后再加