Broadview
www.broadview.com.cn

★Qt：一个跨平台应用程序和UI开发框架★

畅销书
升级版

1. 支持LGPL（GNU Lesser General Public License）协议。
2. 开放了Qt源代码库，并鼓励社区参与。
3. 彻底开放了原来商业平台的代码。
4. 对嵌入式平台支持力度加大。
5. 最后就是不断添加到Qt软件包中的功能，并行计算框架、动画框架、状态机框架、多媒体框架等几个框架为Qt增色不少。

精通 Qt4编程（第2版）

诸多大型项目工程开发经验的凝聚之作

蔡志明 卢传富 李立夏 等编著

电子工业出版社

精通

Qt4编程（第2版）

蔡志明 卢传富 李立夏 等编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书详细介绍了 Qt 的基础知识和 GUI 编程应用，举例翔实，内容全面，基本涵盖了 Qt 编程的各个方面。全书共分 3 篇 25 章，包括 Qt GUI 编程的基础知识（对话框、基础窗口部件、程序主窗口、布局管理）、中级编程（2D 绘图、拖放操作与剪贴板、文件处理、网络编程、多线程、事件机制、数据库及 Qt 风格）和高级应用（XML 应用、模型/视图结构、高级绘图、进程间通信、Qt 插件和脚本应用、多媒体、浏览器、动画及状态机等）。同时，在相关章节也穿插了一些重要的知识点，包括元对象系统、属性系统、对象树机制、信号/槽机制等。

本书体系完整，内容实用，可以作为 Qt 初学者的入门进阶书籍，适合具有一定开发经验的 Qt 程序员作为参考书，也可以作为大中专院校相关专业及培训机构的教材。

图书在版编目（CIP）数据

精通 Qt4 编程 / 蔡志明等编著. —2 版. —北京：电子工业出版社，2011.2
ISBN 978-7-121-12525-6

Ⅰ. ①精… Ⅱ. ①蔡… Ⅲ. ①软件工具—程序设计 Ⅳ. ①TP311.56

中国版本图书馆 CIP 数据核字（2010）第 241779 号

策划编辑：孙学瑛
责任编辑：贾　莉
印　　刷：北京天宇星印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：860×1092　1/16　印张：40.5　字数：1037 千字
印　　次：2011 年 2 月第 1 次印刷
印　　数：4000 册　定价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

本书第1版出版已近三年了，其间Qt也发生了较大的变化。最重要的就是Trolltech公司被Nokia收购，由此改变了Qt发展的趋势。Qt走向了更为开放，功能更丰富，支持的平台更多的道路。

Qt新特点

归纳起来有以下几点：

1．支持LGPL（GNU Lesser General Public License）协议。

我们知道，当一个自由软件使用GPL声明时，该软件的使用者有权重新发布、修改该软件，并得到该软件的源代码；但只要使用者在其程序中使用了该自由软件，或者是使用修改后的软件，那么使用者的程序也必须公布其源代码，同时允许别人发布、修改。也就是说，使用GPL声明下的自由软件开发出来的新软件也一定是自由软件。LGPL是GPL的变种，与GPL的最大不同是，可以私有使用LGPL授权的自由软件，开发出来的新软件可以是私有的而不需要是自由软件。Qt支持LGPL协议后，用户使用开源版Qt获得了更大的自由度。

2．开放了Qt源代码库，并鼓励社区参与。

以前Qt虽是开源代码，但源码库并未开放，现在在http://qt.gitorious.org/可以自由地访问Qt源码库，并可以为Qt做出自己的贡献。Qt的源码库使用分布式版本控制系统Git管理，也可以通过浏览器浏览。这里有Qt，Qt Creator，Qt Jambi等多个项目等着您的参与！开放源代码库，拉近了Qt开发者与用户之间的距离。

3．彻底开放了原来商业平台的代码。

原来只能在商业版中使用的ActiveQt，Oracle数据库驱动等，现在都能在开源版中使用。开源版和商业版不再是两个源码不同的版本，只是遵守的协议不同而已。开源版用户终于能享受全功能Qt了。同时在Windows平台上也开放了Visual Studio插件和相应支持，开源版用户在Windows上除使用MinGW开发外又多了一项选择，使用Visual Studio开发Qt应用。

4．对嵌入式平台支持力度加大。

Qt除了继续支持嵌入式Linux平台外，增加了对Windows CE，Symbian，Maemo 5等嵌入式平台的支持。与此同时，推出了Qt Mobility软件包，对手机上的GPS设备、多媒体、短信、电话等功能提供了较完善的支持，让用户开发手机应用更为便利。同时Qt还对实时操作系统VxWorks

和 QNX 提供了社区支持，为平淡无奇的嵌入式图形系统打了一剂强心针。

5．最后就是不断添加到 Qt 软件包中的功能，并行计算框架、动画框架、状态机框架、多媒体框架等几个框架为 Qt 增色不少。

浏览器、XQuery、XPATH、XSLT、多点触摸、图形效果支持提升了 Qt 的能力。当然还有其他功能和性能改进不能一一列出，这都使得 Qt 4 走向了成熟。Qt 4.7 中更是引入了类似于 Windows ZAML 的声明式 UI 编程的功能，该功能命名为 Qt Quick。

总之，在几年前，我们还需要向周围的人介绍什么是 Qt，如今只需要对如何更好地应用 Qt 进行交流。在国内图书市场上，也出现了为数不少 Qt 相关的书籍。为了与时俱进，本书也根据新版的 Qt 4.7 进行了修改，推出了第 2 版，但本版中不包括 Qt Quick 编程介绍。

本书的结构

本书共 25 章，每章讨论一个专题。章节安排上基本采用循序渐进、由浅到深的原则。但最后的高级篇中的章节没有很强的关联，可以按照随意的顺序阅读。每章内容及作者分述如下：

篇章	章　名	作者	内 容 简 介	页码
初级篇	第 1 章　Qt 初步实践	卢传富	建立了第一个较简单的 Qt 应用程序，在 GUI 用户界面中显示一行中文	2
	第 2 章　对话框——QDialog	卢传富	介绍了 Qt 的对话框类 QDialog，实现了一个自定义的登录对话框，举例说明了 Qt 提供的内建对话框类的应用	14
	第3章　基础窗口部件——QWidget	卢传富 蔡志明	首次引入 Qt 设计器的使用，绘制并实现了一个查找文件功能的部件，介绍了 Qt 应用程序中使用 ui 文件的基本方法以及 Qt 样式表；较深入地分析了 Qt 对象模型的一些基本知识，涉及信号和槽机制、Qt 元对象系统、属性系统和对象树机制，以及部件类型和部件的几何布局等内容	36
	第 4 章　程序主窗口——QMainWindow	卢传富	Qt 应用程序的主窗口是由多个部件/组件构成的框架，本章通过一个简单文本编辑器的例子，介绍了主窗口的菜单、工具条、中心部件、锚接部件和状态条，并通过 Qt 设计器绘制和手写代码两种方法实现了简单文本编辑器主窗口界面的排布和管理	87
	第 5 章　布局管理	卢传富	布局管理是 GUI 应用程序编程的一个重要方面。Qt 提供了多种布局管理部件，包括 Qt 布局管理器、分裂器、栈部件、工作空间部件和多文档区部件等。本章一一介绍了这些部件，并举例说明了它们在图形用户界面编程中的应用	124
中级篇	第 6 章　2D 绘图	蔡志明	本章内容较多，包括 Qt 的绘图要素、图形变换与坐标系统、绘图设备、图像处理、图像打印等	158
	第 7 章 Graphics View 框架	蔡志明	讲解 Qt 图形系统的模型视图框架——Graphics View 框架。包括体系结构、坐标系统、编程方法和图形打印	211
	第8章　拖放操作和剪贴板	蔡志明	本章简要地说明了基于 MIME 的拖放操作和剪贴板的使用，关于 Graphics View 框架的拖放操作也在本章	225
	第 9 章　文件处理	蔡志明	介绍了 Qt 的文件处理，包括基于流的文本文件和二进制文件处理，文件信息和目录操作，目录以及文件的变化监控，文件引擎的编写	232

续表

篇章	章　名	作者	内 容 简 介	页码
中级篇	第 10 章　网络	李立夏	介绍了 Qt 的网络处理，包括编写常见的 FTP、HTTP、UDP 和 TCP 程序，以及访问底层网络接口信息和扩展 Qt 网络模块功能的方法	241
	第 11 章　多线程	李立夏	介绍了 Qt 的多线程处理，包括两方面内容：传统的线程操作，以及与 Qt 事件机制相关的操作。这一章还涉及较多的基本概念，并逐一做了介绍	278
	第 12 章　事件处理	李立夏	介绍了 Qt 的事件处理模型，详细介绍了在 Qt 程序设计中处理事件的五种方法，并讨论了如何利用 Qt 事件机制加快用户界面响应速度	308
	第 13 章　数据库	李立夏	介绍了 Qt 的数据库处理，重点介绍了如何在 Qt 中使用 SQL 语句进行数据库操作和如何利用 QSqlTableModel 这类高层次类进行常见的数据库编程	323
高级篇	第 14 章　界面风格	王燕琴 蔡志明	讲解了 Qt 的风格原理，从简单到复杂应用 Qt 自定义风格。Qt 的样式表的原理和应用	354
	第 15 章　XML	蔡志明	对 Qt 的三种 XML 解析方式（DOM、SAX 和基于流的解析）进行了比较和举例。还讲解了如何使用 API 写 XML 文件。XMLPatterns 模块的应用，包括 XSLT、XQuery、XML 模式	413
	第 16 章　模型/视图结构	蔡志明	阐述了 Qt 的模型/视图结构，分别对模型视图的三个组成部分（模型、视图和代理）进行了介绍，演示了如何自定义这些组成部分，并简要说明了拖放以及选中操作	443
	第 17 章　高级绘图	蔡志明	叙述了在 Qt 中如何使用 OpenGL 绘图，对基本的 OpenGL 绘图进行了讲解，介绍了矢量图型文件 SVG 的读写操作	486
	第 18 章　进程与进程间通信	李立夏	介绍进程和进程间通信的知识，重点介绍了 Qt 中桌面环境下基于 D-Bus 的多进程应用程序开发	502
	第 19 章　Qt 插件	蔡志明	说明了 Qt 的插件系统，并对 Qt Designer 插件、数据库插件、风格插件进行了较详细的介绍	528
	第 20 章　脚本——QtScript	蔡志明	这是 Qt 4.3 中引入的最新内容，使得 Qt 能够支持 ECMAScript 脚本。本章简要地举例说明了在 Qt 中如何使用脚本，如何将 C++对象暴露给脚本	546
	第 21 章　国际化	骆艳	本章包括编码的处理，Qt Linguist 的使用步骤，动态语言切换的内容	557
	第 22 章　Qt 单元测试框架	蔡志明	本章阐述了如何使用 QTestLib 框架进行数据测试、GUI 测试和性能测试	568
	第 23 章　动画与状态机框架	蔡志明	简要介绍了动画与状态机框架，并结合两个框架的应用实例进行了讲解	576
	第 24 章　WebKit 与多媒体	蔡志明	介绍了浏览器和 Phonon 多媒体框架，给出了 Phonon 多媒体播放实例的实现过程	595
	第 25 章　嵌入式 Qt	蔡志明	简要地讲解了在 Window Mobile 和嵌入式 Linux 上开发嵌入式图形应用的方法，还给出了嵌入式 Qt 裁剪的方法	607
	附录 A~D	蔡志明	附录中包括 Qt 在 Linux、Windows 上的安装，VS2008、Eclipse、Qt Creator 集成开发环境的使用，qmake 的基本应用，Qt 资源	619

如何获取源代码

由于 Qt 是跨平台的，因此书中的内容应用能够在 Windows、Linux、UNIX 和 Mac OS 上运行，

书中的程序可能在下列平台之一上编写：Windows XP/Vista、Linux（OpenSuSE、Fedora Core 等）。因此书中的屏幕截图可能来源于其中的任何一种操作系统。

要获取本书的源代码，可以访问博文视点资讯有限公司网站获取：

http://www.broadview.com.cn。

致谢

与电子工业出版社孙学瑛编辑是第二次合作了，孙编辑对本书的关心和认真细致的工作，使本书能够及时出版，再次表示诚挚的谢意。

问题反馈

欢迎广大读者和专家对本书提出建议和批评。如果您认为书有错误或对我们有什么建议，可以联系 jsj@phei.com.cn。

蔡志明 卢传富 李立夏

2010 年 9 月 30 日于武汉

目 录

初 级 篇

中 级 篇

高 级 篇

初级篇

第 1 章　Qt 初步实践

作为 Qt 程序开发之旅的第一站，本章将阐述如何使用 Qt 开发一个简单的 GUI 用户界面程序。在这一章，将学习如何建立 Qt 主程序、建立 qmake 工程，还将接触到“信号（signal）”和“槽（slot）”以及“Qt 布局”等基本概念。随着学习的不断深入，将在第 3 章和第 5 章对这些概念进行深入的讲解，并演示它们在 GUI 用户界面设计中的应用。

1.1　第一个 Qt 程序

在这一节，学习创建第一个较简单的 Qt 应用程序。在这个程序中，用户界面将显示一行中文“同一个世界，同一个梦想！”通过这个程序，将学会使用两种手段建立 Qt 应用程序：KDevelop 集成开发环境和 vim 编辑器。

1.1.1　建立主程序

首先，看一下第一个 Qt GUI 应用程序 hello 的源代码，其内容如下所示（为了便于读者查阅相关源代码，代码的第一行注释了该文件在源代码中的路径）。

```
// chapter01/hello/src/hello.cpp.
#include <QtGui/QApplication>
#include <QtGui/QWidget>
#include <QtGui/QLabel>
#include <QtCore/QTextCodec>
int main(int argc, char* argv[])
{
    QApplication app(argc, argv);
    QTextCodec::setCodecForTr(QTextCodec::codecForName("gb18030"));
    QWidget* pWidget = new QWidget;
    QLabel label(pWidget);
    label.setText(QObject::tr("同一个世界，同一个梦想！"));
    pWidget->show();
    return app.exec();
}
```

这个程序的功能是在一个窗口中显示“同一个世界，同一个梦想！”，运行效果如图 1-1 所示。

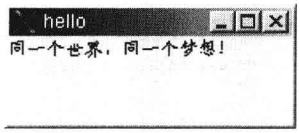

图 1-1 第一个 Qt 程序

注意

窗口（Window）和窗口部件（Widget）

本书中，多次使用了窗口和窗口部件的概念。把一个图形用户界面称为窗口，它往往具有标题栏、窗口边框（frame）、能够通过鼠标拖动和改变大小等特性，最典型的窗口就是对话框。例如，第一个 Qt 应用程序的用户界面就是一个窗口。当文中使用窗口的时候，就是特指这种情况。

一般的，窗口部件是对所有图形用户界面的统称，它既可以作为单独的窗口出现，也可以出现在一个窗口的内部。

1.1.2 建立工程

现在暂不分析第一个 Qt 应用程序是如何运行的，而是先为它建立 qmake 工程，然后进行调试和运行。

在 Linux 系统中，可以有多种方法输入、编辑上述 Qt 程序，此处将使用 vim 文本编辑器和 KDevelop 工具建立上述主程序。

1. 在 vim 中建立 Qt 应用程序

在 vim 中建立 Qt 应用程序，步骤如下（如果读者觉得在 vim 文本编辑器中编辑、调试以及运行 Qt 应用程序比较麻烦，可以直接跳过这一部分，而选择在 KDevelop 集成开发环境中建立该 Qt 应用程序）：

01 打开控制台程序 konsole，将当前目录切换到相应的路径下，执行控制台命令“mkdir hello”建立 hello 目录，执行 “cd hello”命令进入该目录。

02 在控制台执行“vim hello.cpp”命令（如果文件 hello.cpp 已经存在，则打开文件；否则新建 hello.cpp 文件），打开 vim 文本编辑器。

03 进入 vim 编辑器后按“i”键（即打开 vim 编辑器的修改编辑功能），然后输入第一个 Qt 应用程序 hello 的源代码。

04 按 Esc 键，退出 vim 文本编辑器的编辑功能。

05 最后，在 vim 文本编辑器的命令行输入命令“:wq”，按回车键后 vim 将保存文件并退出。

现在，第一个 Qt 应用程序 hello 已经输入到 vim 文本编辑器中了。关于 vim 编辑器的使用读者可查阅相关帮助文档，在此不再赘述。

2. 在 KDevelop 中建立 Qt 应用程序

KDevelop 是集编辑、编译、调试和运行 C++程序等诸工具于一身的应用程序集成开发环境。

在 KDevelop 中建立 Qt 应用程序并进行编译、调试、运行都是比较简单的，但需要建立 KDevelop 工程（这和使用 VC++进行应用程序开发是类似的），对于初学者来说过程有些复杂。具体步骤如下：

01 打开 KDevelop 后，选择菜单“工程”|“新建工程”，如图 1-2 所示。

02 在“建立新工程”对话框的“所有工程”选项卡中，选择“C++ | QMake project | Basic Qt 4 Application”，选择或者输入存放位置（例如，“/home/lcf/book/chapter01”），输入应用程序名称“hello”（KDevelop 将会在/home/lcf/book/chapter01/路径下建立 hello 目录），单击“下一步”按钮，如图 1-3 所示。

03 设置“工程选项”，在此输入 Qt 4 的 qmake 和 Qt 设计器的绝对路径，直接单击“下一步”按钮，如图 1-4 所示。

04 设置“版本控制系统”，略过，单击“下一步”按钮，如图 1-5 所示。

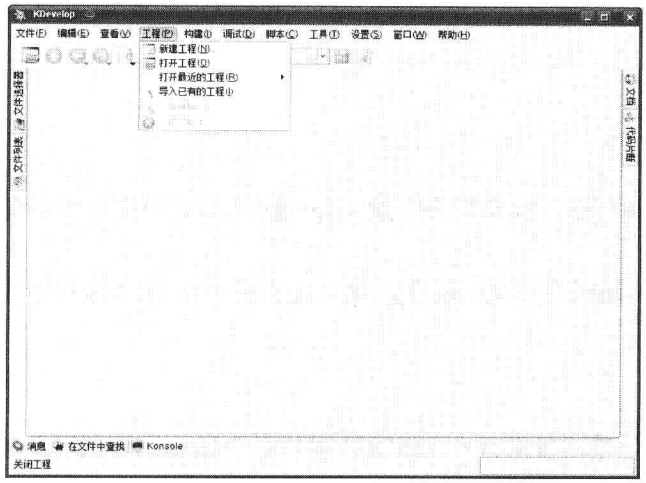

图 1-2 新建 KDevelop 工程

图 1-3 建立 hello 工程

图 1-4 设置工程选项

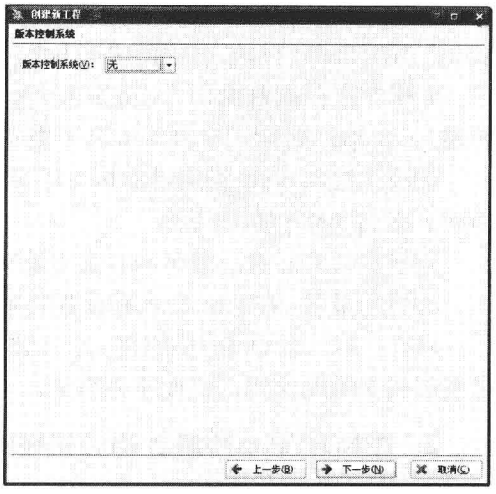

图 1-5 设置版本控制

05 在“h 文件的模板”选项（如图 1-6 所示）中，可以设置头文件 .h 的格式（在此省略）；单击“下一步”按钮进入“cpp 文件的模板”选项卡（如图 1-7 所示），与“h 文件的模板”类似。

06 最后，单击“完成”按钮，KDevelop 会自动生成一个标准的 C++主程序。在此，编辑修改为第一个 Qt 应用程序 hello 的源代码，如图 1-8 所示。

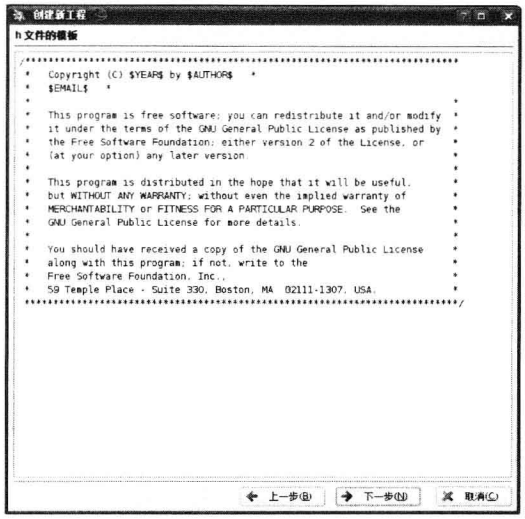

图 1-6 设置头文件模板

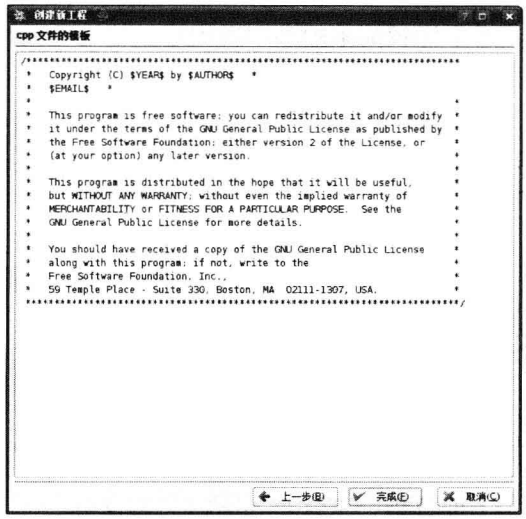

图 1-7 设置实现文件模板

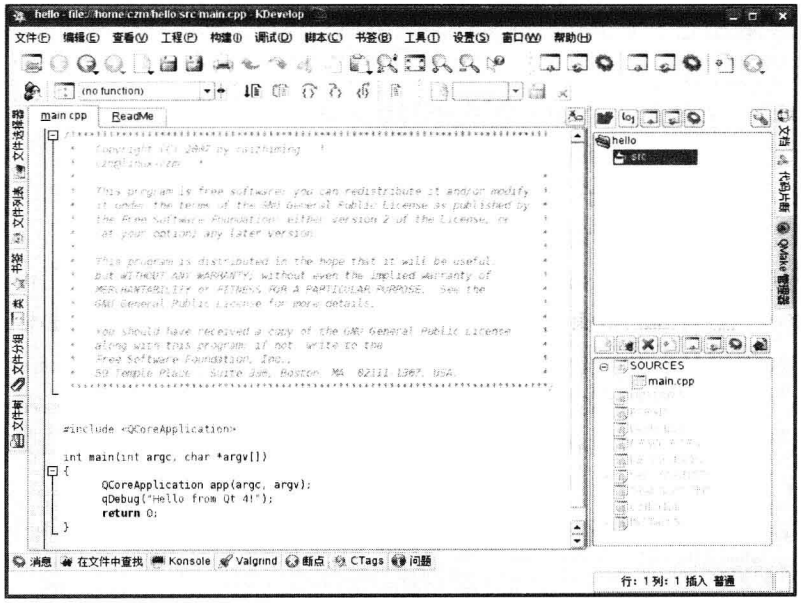

图 1-8 第一个 KDevelop 工程

到此，在 KDevelop 中已经建立了一个 KDevelop 工程，并且输入了第一个 Qt 应用程序 hello。

接下来，建立 Qt 应用程序的 qmake 工程文件。

有两种方法建立 qmake 工程：自动生成和手动建立。下面分别描述如何使用这两种方法建立应用程序 hello 的 qmake 工程。

3. 自动建立 qmake 工程

对于比较简单的小应用程序，使用 qmake 命令自动建立的 qmake 工程完全可以满足需要。

前面，已经在 vim 文本编辑器中输入了第一个 Qt 应用程序 hello 的源代码，现在为它建立相应的 qmake 工程。

首先，在控制台 konsole 中将当前目录切换到 hello.cpp 文件所在的目录，运行“qmake –project”命令。此时 Qt 的 qmake 工具将在当前目录下自动生成应用程序 hello 的工程文件 hello.pro，其内容如下。

```
TEMPLATE = app
TARGET =
DEPENDPATH += .
INCLUDEPATH += .
# Input
SOURCES += hello.cpp
```

下面分析一下这个 qmake 工程文件。

前 3 行文本是 qmake 工具自动添加的注释文本，它描述了该 qmake 工程文件是由 qmake 工具自动生成的，以及文件生成的时间和 qmake 工具的版本号。

变量 TEMPLATE 描述了为建立目标文件而采用何种模板，即生成何种形式的 Makefile 文件。qmake 工具定义了 5 种模板：

- **应用程序 app**，为建立一个 Qt 应用程序创建 Makefile 文件；
- **库 lib**，为建立应用程序库而创建 Makefile 文件；
- **子工程 subdirs**，为建立子目录下的目标文件创建一个 Makefile 文件，子目录通过变量 SUBDIRS 指定（子目录下的工程文件也需要指出使用何种模板）；
- **VC 应用程序 vcapp**，为 Visual Studio 生成一个应用程序工程，仅仅用于 Windows 操作系统。
- **VC 库 vclib**，为 Visual Studio 生成一个应用程序库工程，仅仅用于 Windows 操作系统。

由于第一个 Qt 程序是一个可直接执行的应用程序，因此采用“应用程序 app”模板。

变量 TARGET 描述了目标文件的名称，即生成的应用程序的名字。qmake 工具自动生成的 qmake 工程文件采用默认方式（TARGET 的值为空），即应用程序的名字采用工程文件 hello.pro 所在的文件夹的名字 hello。

变量 DEPENDPATH 描述了建立应用程序所依赖的其他文件所在的路径。hello.pro 工程文件中将该值设置为当前目录。

变量 INCLUDEPATH 描述了编译该工程时编译器需要搜索的#include 路径。hello.pro 工程文件中将该值设置为当前目录。

变量 SOURCES 选项告诉编译器，源代码文件的路径（相对于工程文件 hello.pro 的位置）及其