1. 创建项目 注意事项

注意：路径名不支持中文和空格

QWidget是QMainWindow和QDialog的基类。

QWidget只是一个窗口，没有具体的功能

QMainWindow实现菜单栏，状态栏

QDialog实现弹出对话框

项目管理

添加到版本控制系统

Git vss svn都是用于团队开发，项目的版本管理。

二．Main.cpp初始化分析

#include “myWidget.h”

#include<QApplication> 头文件 应用程序

//程序入口， argc命令行变量数量， argv命令行变量数组

Int main(int argc, char \*argv[])

{

QAppliaction a(argc, argv); //应用程序对象a，Qt中有且只有一个应用程序对象

MyWidget w; //MyWidget的基类是QWidget，窗口对象

w.show(); //窗口默认是不会弹出的，调用show()弹出

return a.exec(); //进入消息循环机制，窗口能够一直保持

}

* QApplication应用程序类
  + 管理图形用户界面应用程序的控制流和主要设置。
  + Qt的整个后台管理，它**包含主事件循环**，在其中来自窗口系统和其它资源的**所有事件处理和调度**。它也处理**应用程序的初始化和结束**，并且**提供对话管理**。
  + 有且只存在一个QApplication 对象。
* a.exec()

**程序进入消息循环**，等待对用户输入进行响应。这里main()把控制权转交给Qt，Qt完成事件处理工作，当应用程序退出的时候exec()的值就会返回。**在exec()中，Qt接受并处理用户和系统的事件并且把它们传递给适当的窗口部件。**

三．Pro配置文件分析

**.pro就是工程文件(project)，它是qmake自动生成的用于生产makefile的配置文件**

QT += core gui //Qt包含的模块

greaterThan(QT\_MAJOR\_VERSION,4)：QT += widgets //大于4的版本，会包含widget模块

TARGET = 01\_Qt\_First //生成的.exe可执行文件的名称

TEMPLATE = app //模板，应用程序

SOURCES += main.cpp\

Mywidget.app //源文件

HEADERS += mywidget.h //头文件

* 模板变量告诉qmake为这个应用程序生成哪种makefile。下面是可供使用的选择：**TEMPLATE** = app
  + app -建立一个应用程序的makefile。这是默认值，所以如果模板没有被指定，这个将被使用。
  + lib - 建立一个库的makefile。
  + vcapp - 建立一个应用程序的VisualStudio项目文件。
  + vclib - 建立一个库的VisualStudio项目文件。
  + subdirs -这是一个特殊的模板，它可以创建一个能够进入特定目录并且为一个项目文件生成makefile并且为它调用make的makefile。
* #配置信息

CONFIG用来告诉qmake关于应用程序的配置信息。

CONFIG += c++11 //使用c++11的特性

在这里使用“+=”，是因为我们添加我们的配置选项到任何一个已经存在中。这样做比使用“=”那样替换已经指定的所有选项更安全。

四．Mywidget.h文件分析

#ifndef MYWIDGET\_H //防止mywidget.h头文件重复编译

#define MYWIDGET\_H

#include<QWidget> //基类

Class MyWidget : public QWidget //继承基类QWidget

{

Q\_OBJECT //宏，写上就支持Qt中的信号和槽机制

}

Public:

MyWidget(QWidget \* parent = 0);

~MyWidget();

#endif //MYWIDGET\_H

五．命名规则

类名

一个单词，首字母大写

多个单词，单词和单词之间 ，首字母大写

函数和变量

一个单词 首字母小写

多个单词，单词与单词之间，首字母大写

六．快捷键

运行 ctrl + R

编译 ctrl + B

查找 ctrl + F

帮助文档 F1

字体缩放 ctrl + 鼠标滚轮

自动对齐 ctrl + i

整行移动 ctrl + shift + ↑或者↓

同名的.h和.cpp切换 F4