一．对话框QDialog

之前有说到，QDialog类负责对话框部件，继承于QWidget类。对话框一般用于实现用户交互，弹出一个顶层窗口。很多不适合放在主窗口的功能都可以放在对话框中。

QDialog类及其子类对于parent指针有特别的设置：如果parent为NULL，那么该对话框独立作为一个顶层窗口；若parent不为NULL，则该对话框作为其父组件的子对话框，而非顶层窗口。

**顶层窗口与非顶层窗口的区别在于，顶层窗口在任务栏会有自己的位置，而非顶层窗口则会共享其父组件的位置。**

二．对话框分为模态对话框和非模态对话框：

模态对话框：阻塞同一程序的其他窗口操作，仅限于同一程序的界面位置。

实现：

应用程序级别的模态

当该种模态的对话框出现时，用户必须首先对对话框进行交互，直到关闭对话框，然后才能访问程序中其他的窗口。

QDialog::exec()现应用程序级别的模态对话框

窗口级别的模态

该模态仅仅阻塞与对话框关联的窗口，但是依然允许用户与程序中其它窗口交互。窗口级别的模态尤其适用于多窗口模式。

QDialog::open()实现窗口级别的模态对话框

非模态对话框：可以同时操作同一程序的其他窗口。

使用QDialog::show()实现非模态对话框

QDialog dialog(this);

dialog.setWindowTitle(tr("Hello, dialog!"));

dialog.show();

注意：使用show()时，对话框有时会闪现，然后就消失了。这是因为当dialog建立在栈上时，show()函数返回，MainWindow::open()函数结束，则dialog超出作用域被析构了，因此对话框消失了。

解决：将dialog建立在堆上。

QDialog \*dialog = new QDialog(this);

这样做，也有问题，若主窗口不关闭，那么该对话框占用的内存就一直不会释放，怎么处理呢？

设置dialog的属性，当dialog对话框关闭时，自动销毁对话框。

QDialog \*dialog = new QDialog;

dialog->setAttribute(Qt::WA\_DeleteOnClose);

三．标准对话框

Qt中内置了很多标准的对话框，因为开发中，有很多功能都是软件通用的，所以不需要单独开发，直接使用就可以了。

Qt 的内置对话框大致分为以下几类：

* QColorDialog： 选择颜色；
* QFileDialog： 选择文件或者目录；
* QFontDialog： 选择字体；
* QInputDialog： 允许用户输入一个值，并将其值返回；
* QMessageBox： 模态对话框，用于显示信息、询问问题等；
* QPageSetupDialog： 为打印机提供纸张相关的选项；
* QPrintDialog： 打印机配置；
* QPrintPreviewDialog：打印预览；
* QProgressDialog： 显示操作过程。

四．QMessageBox对话框

QMessageBox用于显示消息提示，是最常用的标准对话框。

1. 错误

提示错误信息的，参数如下所示：

StandardButton critical(QWidget \* parent,

const QString & title,

const QString & text,

StandardButtons buttons = Ok,

StandardButton defaultButton = NoButton)：

1. 信息提示

弹出信息提示。

StandardButton information(QWidget \* parent,

const QString & title,

const QString & text,

StandardButtons buttons = Ok,

StandardButton defaultButton = NoButton)

（三）询问

与用户交互，询问用户如何操作？

StandardButton question(QWidget \* parent,

const QString & title,

const QString & text,

StandardButtons buttons = StandardButtons( Yes | No ),

StandardButton defaultButton = NoButton)

（四）警告

StandardButton warning(QWidget \* parent,

const QString & title,

const QString & text,

StandardButtons buttons = Ok,

StandardButton defaultButton = NoButton)

五．代码

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include<QDebug>

#include<QDialog>

#include<QMessageBox>

#include<QColorDialog>

#include<QFileDialog>

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

//点击新建菜单项，弹出对话框

connect(ui->actionnew, &QAction::triggered, this, [=](){

qDebug() << "弹出对话框！";

// //对话框：模态对话框（不可操作其他窗口）非模态对话框（可操作其他窗口）

// QDialog dlg;

// dlg.resize(200, 100);

// //模态对话框

// dlg.exec(); //阻塞后面的操作

// //非模态对话框

// //QDialog dlg2; //创建在栈上，一闪而过

// QDialog \*dlg2 = new QDialog(this);

// dlg2->resize(200,100);

// dlg2->show();

//使用标准对话框 QMessageBox

//错误

//QMessageBox::critical(this, "错误！","critical");

//信息提示

//QMessageBox::information(this, "信息！","message");

//询问

// if(QMessageBox::Save == QMessageBox::question(this,"询问","保存吗？", QMessageBox::Save | QMessageBox::Cancel))

// {

// qDebug() << "点击的是保存";

// }

// else

// {

// qDebug() << "点击的是取消";

// }

//颜色对话框

QColor color = QColorDialog::getColor(QColor(255,0,0));

qDebug() << color.red() << color.green() << color.blue();

//文件对话框

QFileDialog::getOpenFileName(this, "打开文件","./", "\*.txt");

});

}

MainWindow::~***MainWindow***()

{

delete ui;

}