1.多界面编程：

QWidget\* parentWidget() ---> 获得当前控件的父控件

2.信号和槽

实现原理：qt使用元对象系统机制来实现信号和槽

信号：

无返回值

只声明，不需要实现 ----- signals

发送信号：emit ---如果不发送信号，那么，与它建立的槽函数是不会被调用的

槽：

无返回值

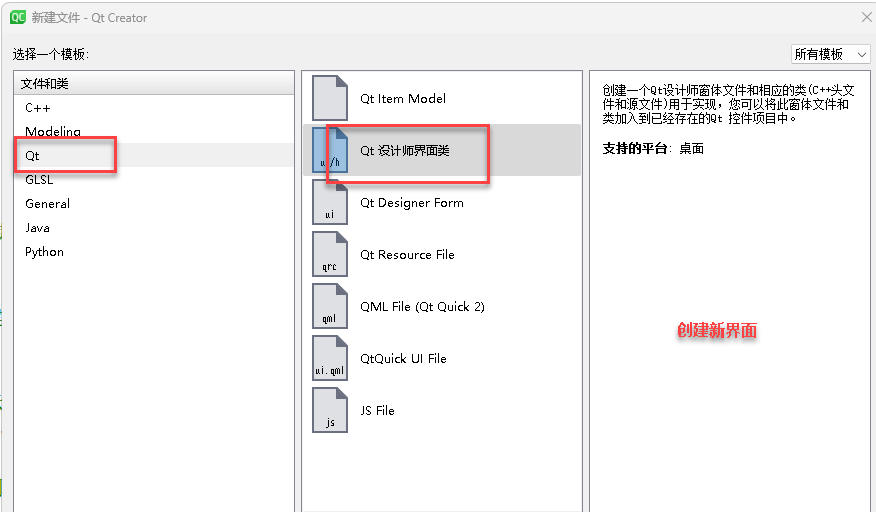
声明并且实现 ---- 成员权限修饰符 slots

通过connect函数来建立信号和槽的连接，默认情况下是连接的

如果要把信号和槽断开连接，那么要使用disconnect

connect(对象一指针，&对象一类名::信号，对象二指针,&对象二类名::槽的名字)

disconnect(对象一指针，&对象一类名::信号，对象二指针,&对象二类名::槽的名字)



#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "twowindow.h" //引入新界面头文件

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

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

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

{

delete ui;

}

//信号和槽的使用：场景一 ---- 执行或者相应用户操作 场景二 --- 两个对象之间的通信

void MainWindow::**on\_pushButton\_clicked**()

{

//1.创建新的界面类

TwoWindow\* tw = new TwoWindow; //创建到堆区，新的界面与当前界面相互独立，没有依赖关系

//2.显示新的界面

tw->show();

//3.关闭或者隐藏当前界面

this->close(); //如果在创建新界面时，没有指定与当前界面的关系，就可以close当前界面

}

1. 在原界面头文件中类中声明signals关键字，声明发送信号函数

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

signals: //定义信号的关键字

void **sendData**(const QString& data); //信号不需要实现，只需要声明

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

1. 在新界面头文件中定义槽函数

#ifndef TWOWINDOW\_H

#define TWOWINDOW\_H

#include <QMainWindow>

namespace **Ui** {

class **TwoWindow**;

}

class **TwoWindow** : public QMainWindow

{

Q\_OBJECT

public:

explicit **TwoWindow**(QWidget \*parent = nullptr);

~***TwoWindow***();

private slots:

void **on\_pushButtonback\_clicked**();

//在对象二中定义槽，如果该槽在类的外部调用，则要把它声明为public

public slots:

void **getData**(const QString& data);

private:

Ui::TwoWindow \*ui;

};

#endif // TWOWINDOW\_H

1. 在第一个界面调用connect接口建立链接,使用emit发射信号

Connect(this, &MainWindow::sendData, tw, &TwoWindow::getData);

void MainWindow::**on\_pushButton\_clicked**()

{

TwoWindow\* tw = new TwoWindow;

connect(this,&MainWindow::sendData,tw,&TwoWindow::getData);

//发射信号 --- emit

emit this->sendData("我是来自上一个界面");

tw->show();

this->close();

}

4.在新界面的槽函数中（接收信号的函数）添加事件

void TwoWindow::**on\_pushButtonback\_clicked**()

{

QWidget\* widget = this->parentWidget(); //该函数返回的是，界面的基类

//将正确返回的基类对象强制称派生类对象

MainWindow\* mw = dynamic\_cast<MainWindow\* >(widget);

mw->show();

//this->hide();

this->close();

}

void TwoWindow::**getData**(const QString &data)

{

ui->label->setText(data);

}