013130

知识点1【Qt创建一个工程】

知识点2【项目介绍】

1、.pro工程文件

2、帮助文档（qtcreater提供了帮助文件）

知识点3【Qt的main介绍】

知识点4【Qt的第一个程序】

1、设置主窗口标题的函数

2、解决按钮乱码

3、窗口简单设置

知识点5【对象树】

建议：从堆区申请空间 而不是从栈区。

知识点6【Qt的坐标体系】（了解）

知识点7【信号和槽机制】（重要）

案例：单击button 关闭主窗口

1、查看QPushButton的信号

2、查看this中槽函数

知识点7【自定义信号和槽】（了解）

注意：

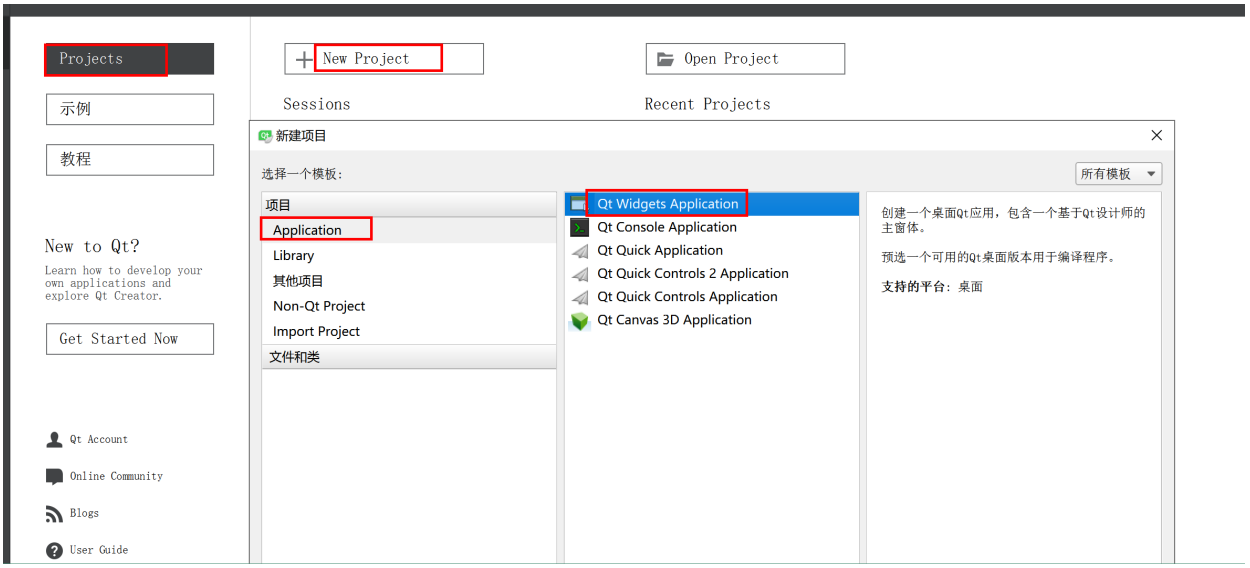
总结：

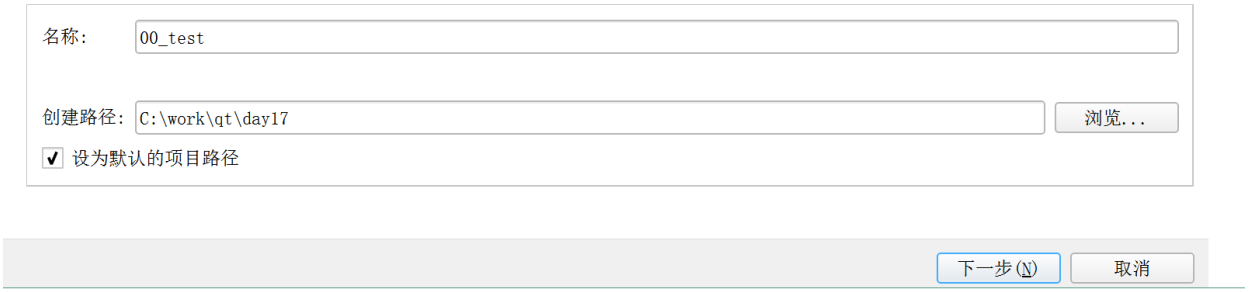
1、一个信号可以和多个槽相连

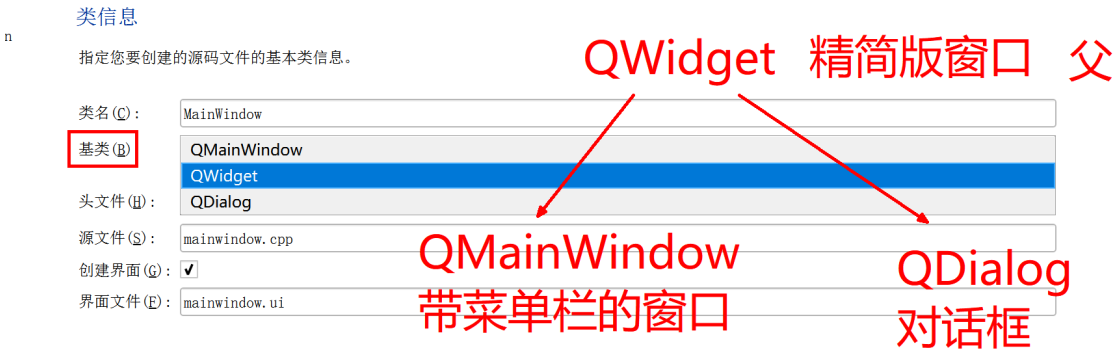
2、多个信号可以连接到一个槽

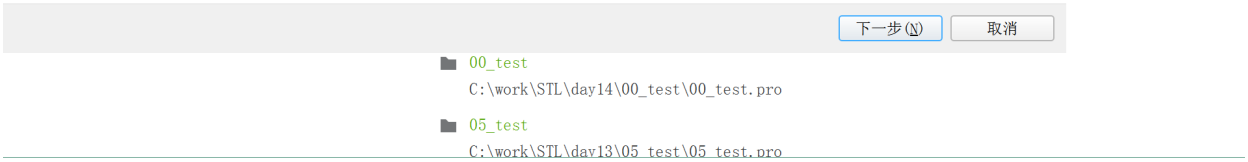
3、一个信号可以连接到另外的一个信号

知识点1【Qt创建一个工程】






类信息
指定您要创建的源码文件的基本类信息。
QWidget 精简版窗口 父
类名(C)：
MainWindow
基类(B)
QMainWindow
QWidget
QDialog
头文件(H)：
源文件(S)：
mainwindow.cpp
创建界面(G)：
界面文件(F)：
mainwindow.ui
QMainWindow
带菜单栏的窗口
QDialog
对话框

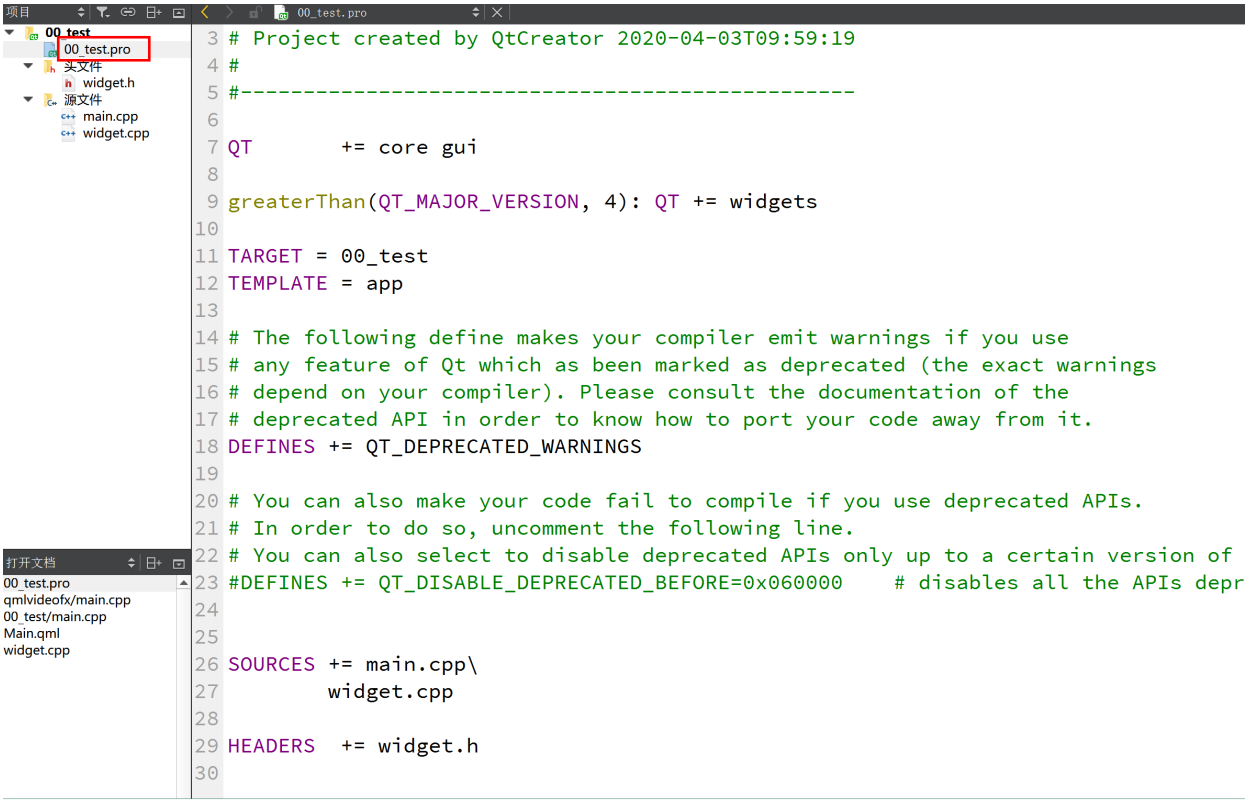

下一步(N)
取消
00_test
C:\work\STL\day14\00_test\00_test.pro
05_test
C:\work\STL\day13\05_test\05_test.pro


Qt Widgets Application
Location
Kits
Details
汇总
类信息
指定您要创建的源码文件的基本类信息。
类名(C)：
Widget
窗口类名可以任意
基类(B)：
QWidget
头文件(H)：
widget.h
源文件(S)：
widget.cpp
创建界面(G)：
界面文件(F)：
widget.ui
下一步(N)
取消


项目
widget.cpp
<选择符号>
00_test
00_test.pro
头文件
widget.h
源文件
main.cpp
widget.cpp
#include "widget.h"
工程文件
Widget::Widget(QWidget *parent)
: QWidget(parent)
{
}
Widget::~Widget()
{
}
窗口文件

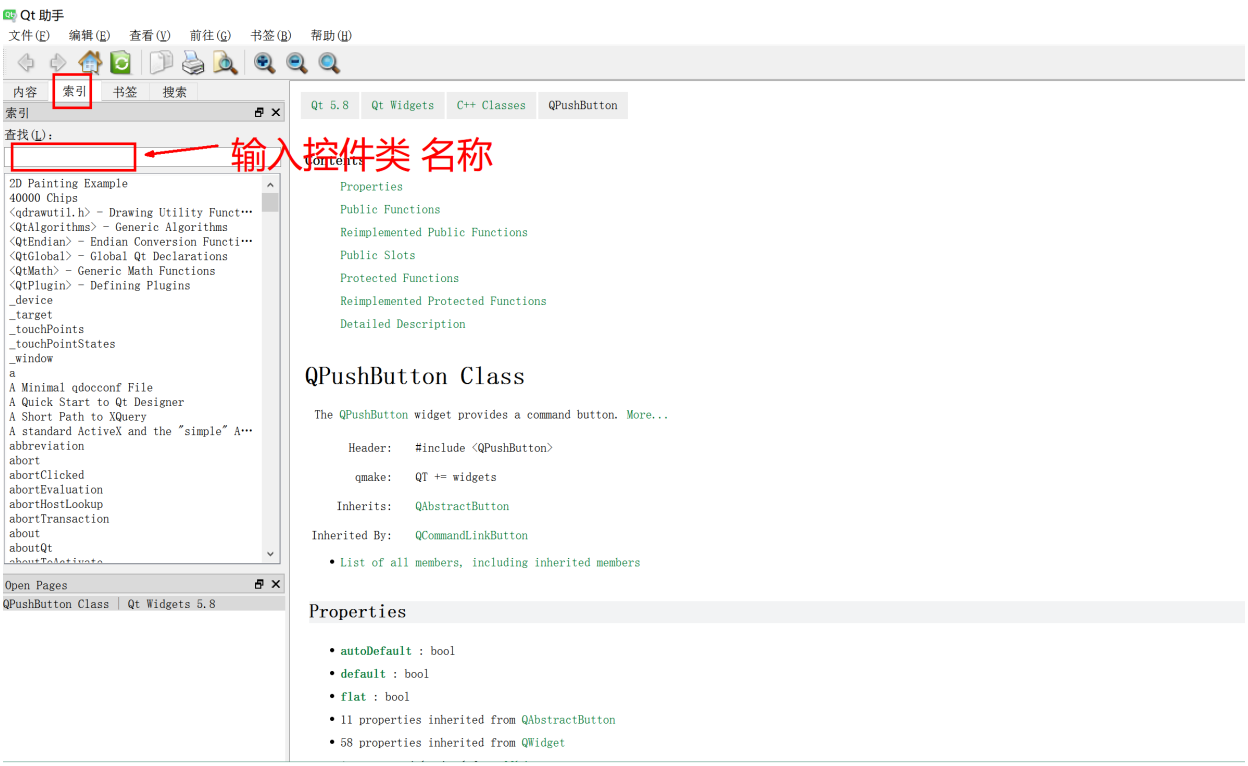
知识点2【项目介绍】

1、.pro工程文件

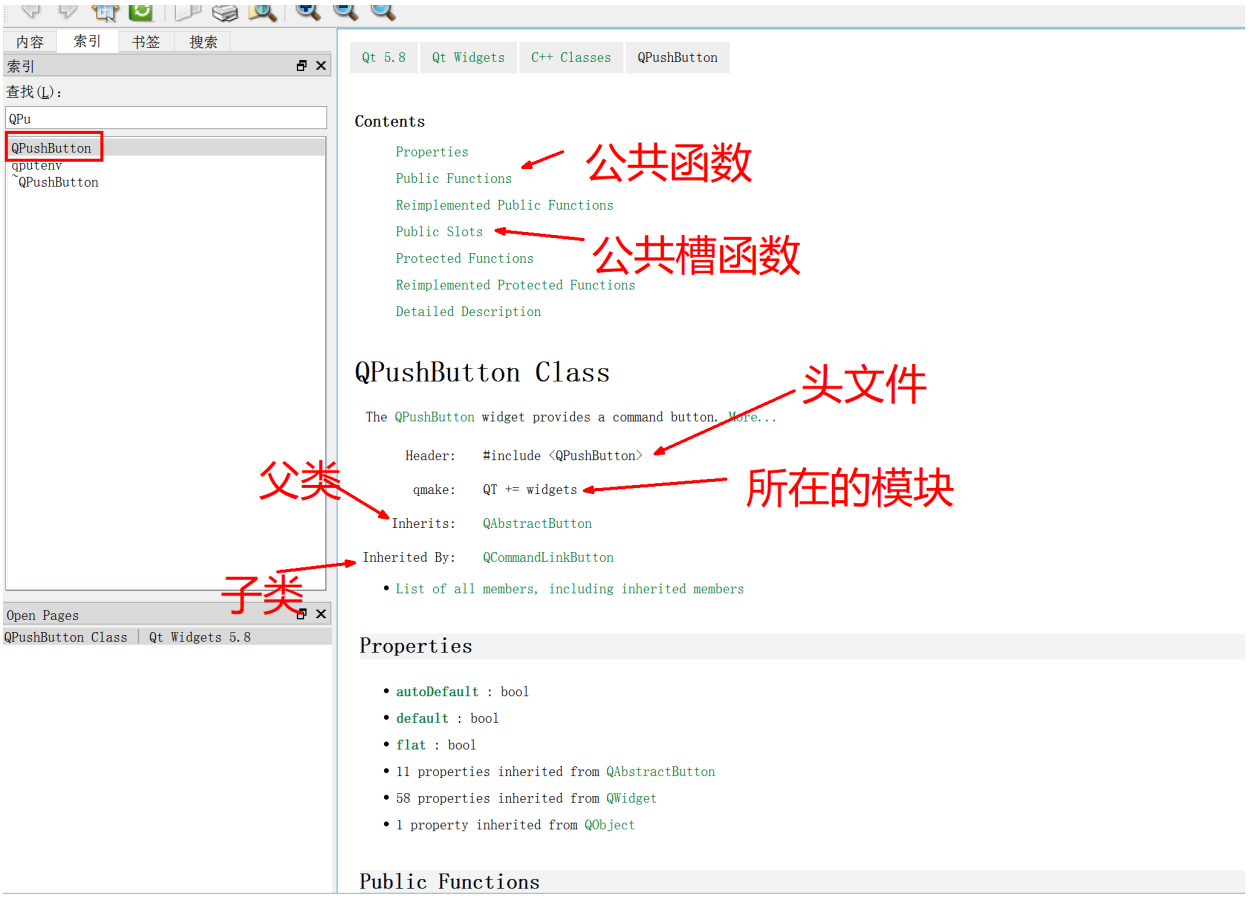


```
QT       += core gui  //包含的模块
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets //大于Qt4版本 才包含widget模块
TARGET = QtFirst  //应用程序名  生成的.exe程序名称
TEMPLATE = app    //模板类型    应用程序模板
SOURCES += main.cpp\   //源文件
        mywidget.cpp
HEADERS  += mywidget.h   //头文件
```

2、帮助文档（qtcreater提供了帮助文件）



比如：我们查看QPushButton按钮控件



知识点3【Qt的main介绍】

main.cpp

```cpp
#include "widget.h"
#include <QApplication>

int main(int argc, char *argv[])
{
 //QApplication 应用程序类 初始化我们的应用程序
 QApplication a(argc, argv);

 //创建一个窗口控件
  Widget w;

 //显示一个窗口 hide隐藏窗口
 //窗口默认是隐藏的
 w.show();

 //a.exec() 主事件循环（带阻塞 等待用户操作界面）
 return a.exec();
}

```

widget.h

```
#ifndef WIDGET_H
#define WIDGET_H

#include <QWidget>

class Widget : public QWidget
{
 Q_OBJECT//让Widget支持信号和槽机制

public:
 Widget(QWidget *parent = 0);
 ~Widget();
};

#endif // WIDGET_H

```

widget.cpp

```
#include "widget.h"

Widget::Widget(QWidget *parent)
 : QWidget(parent)
{
 //界面的设计是在 窗口控件的构造函数中设计

}

Widget::~Widget()
{

}

```

知识点4【Qt的第一个程序】

1、设置主窗口标题的函数

查找(L):

QWid

Accessibility for QWidget Applications
QWidget
QWidget::DrawChildren
QWidget::DrawWindowBackground
QWidget::IgnoreMask
QWidget::RenderFlag
QWidget::RenderFlags
QWidgetAction
QWidgetItem
QWIDGETSIZE_MAX
~QWidget
~QWidgetAction
~QWidgetItem

Open Pages

QWidget Class | Qt Widgets 5.8

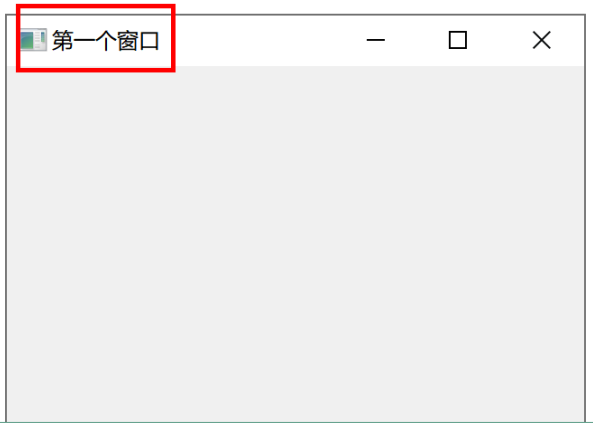
• 14 public functions inherited from QPaintDevice

Public Slots

bool close()
void hide()
void lower()
void raise()
void repaint()
void setDisabled(bool *disable*)
void setEnabled(*bool*)
void setFocus()
void setHidden(bool *hidden*)
void setStyleSheet(const QString &*styleSheet*)
virtual void setVisible(bool *visible*)
void setWindowModified(*bool*)
void setWindowTitle(*const QString &*)
void show()
void showFullScreen()
void showMaximized()
void showMinimized()
void showNormal()
void update()

2、解决按钮乱码

```cpp
Widget::Widget(QWidget *parent)
    : QWidget(parent)
{
    //this  代表当前主窗口
    //设置标题
    this->setWindowTitle("第一个窗口");
}

Widget::~Widget()
{

}
```



3、窗口简单设置

查找(L):

QPush

QPushButton
~QPushButton

Detailed Description

QPushButton Class

The QPushButton widget provides a command button. More...

Header: #include <QPushButton>

qmake: QT += widgets

Inherits: QAbstractButton

Inherited By: QCommandLinkButton

- List of all members, including inherited members

Properties

Public Functions

构造函数

QPushButton(QWidget *parent = Q_NULLPTR)

QPushButton(const QString &text, QWidget *parent = Q_NULLPTR)

QPushButton(const QIcon &icon, const QString &text, QWidget *parent = Q_NULLPTR)

~QPushButton()

widget的构造函数中：

```cpp
Widget::Widget(QWidget *parent)
  : QWidget(parent)
{
  //this 代表当前主窗口
```

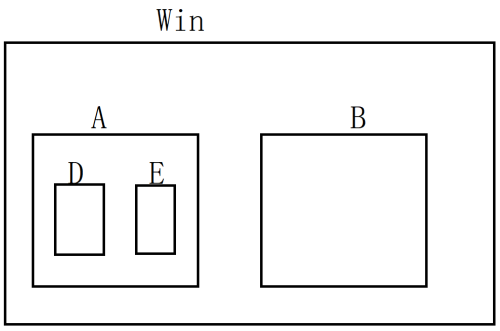
```
5   //设置标题
6   this->setWindowTitle("第一个窗口");
7
8   //固定窗口（不可拖动）
9   //this->setFixedSize(800,600);
10   //设置窗口大小(可拖动)
11   this->resize(800,600);
12
13   //在窗口上放一个按钮
14   //创建一个button控件
15   //parent父对象为this 表明 主窗口 将来接管button
16   //因为我希望将button放到当前窗口中 this代表当前窗口
17   QPushButton *button = new QPushButton("戳我呀",this);
18
19   QPushButton *button1 = new QPushButton("咬我呀",this);
20   //默认控件会显示到主窗口的左上方
21   //移动按钮
22   button1->move(400,300);
23  }
```

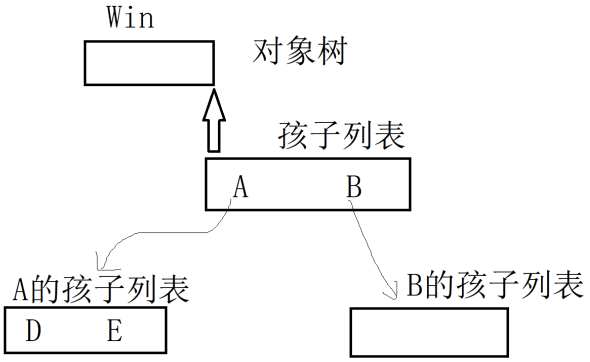
知识点5【对象树】

在创建QObject对象时，可以提供一个其父对象，我们创建的这个QObject对象会自动添加到其父对象的children()列表。

当父对象析构的时候，这个列表中的所有对象也会被析构。（注意，这里的父对象并不是继承意义上的父类！）

我们也可以自己删除子对象，它们会自动从其父对象列表中删除





建议：从堆区申请空间 而不是从栈区。

```
QWidget window;
QPushButton quit("Quit", &window);
```

入栈顺序：window先 quit后 弹栈先调用quit的析构 就会将quit从windos的孩子列表删除，然后windwos调用析构，由于孩子列表中没有对象，就不会再次去释放quit

```
QPushButton quit("Quit");
QWidget window;
quit.setParent(&window);
```

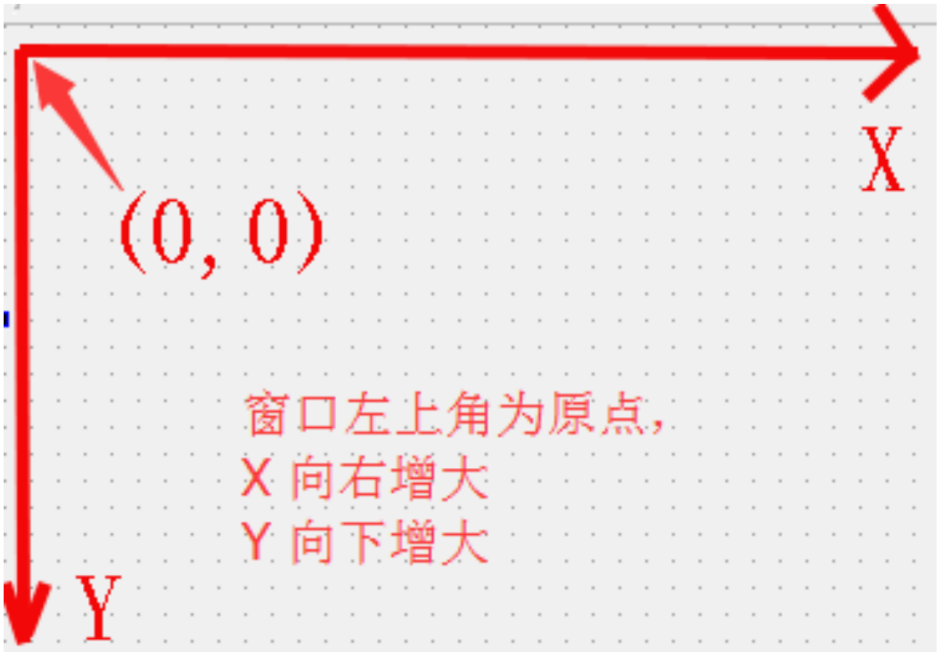
入栈顺序：先入 quit 后window 先调用window析构--->查看window的孩子列表 调用quit的析构。然后quit出栈 也会调用quit的析构。所以quit被调用了两次析构。

在 Qt 中，尽量在构造的时候就指定 parent 对象，并且大胆在堆上创建.

知识点6【Qt的坐标体系】（了解）

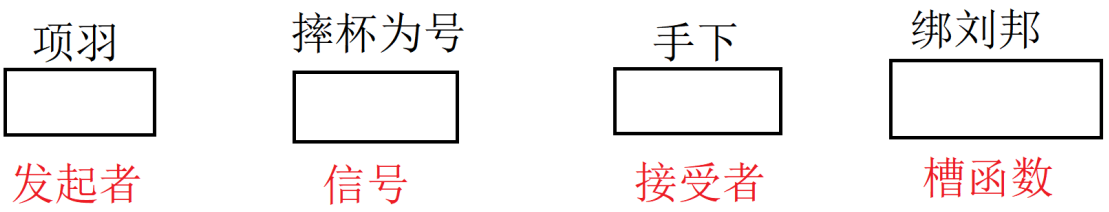
坐标体系：

以左上角为原点（0,0），X向右增加，Y向下增加。



对于嵌套窗口，其坐标是**相对于父窗口**来说的。

知识点7【信号和槽机制】（重要）



1、事先约定

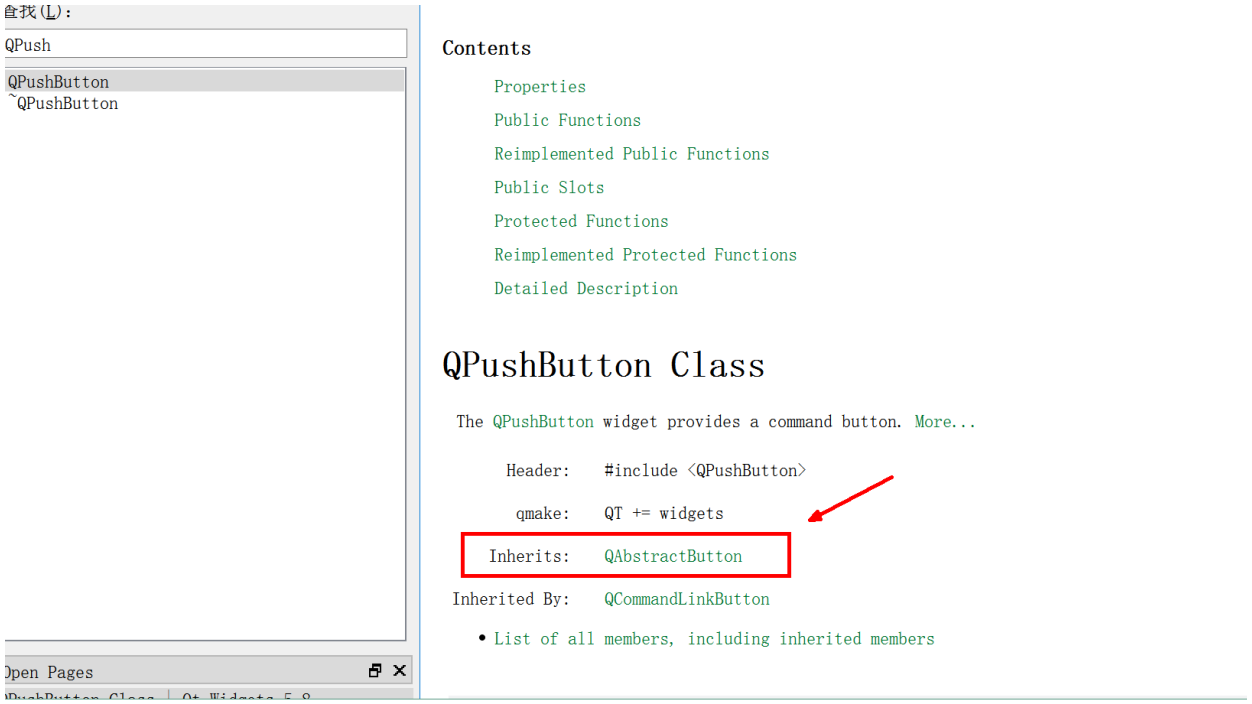
connect(发起者，信号，接收者，槽函数);

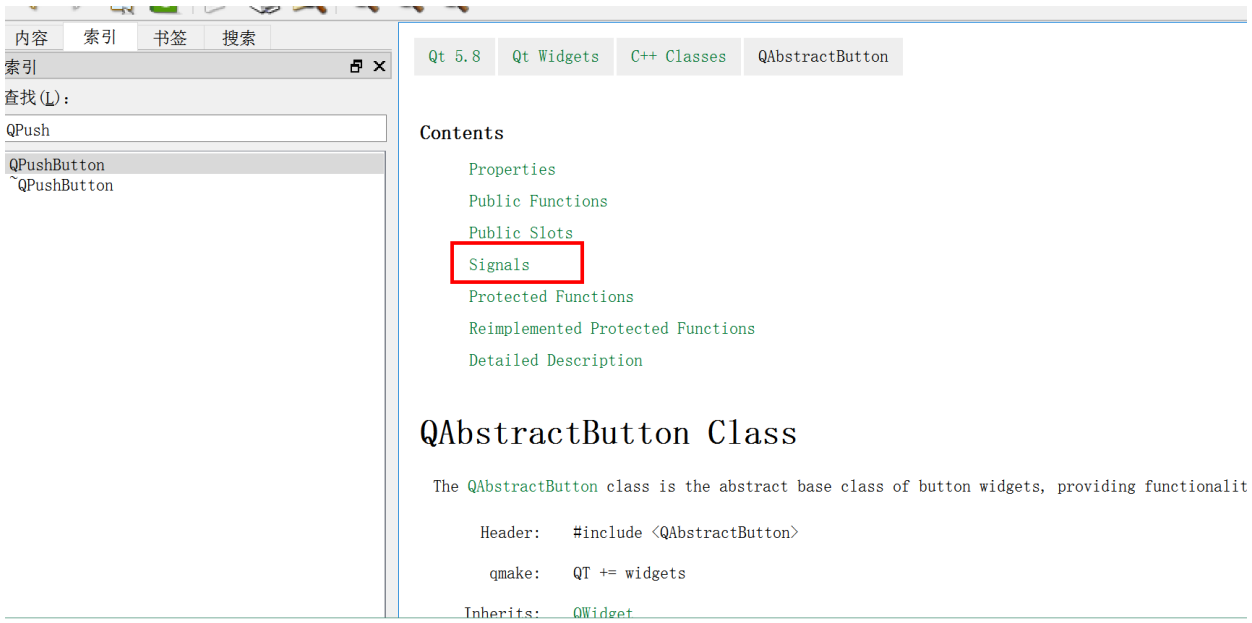
如果：信号发出　　接受者 执行 槽函数

2、信号 发生--->接受者 执行 槽函数

案例：单击button 关闭主窗口

1、查看QPushButton的信号


查找(L):
QPush
QPushButton
~QPushButton
Open Pages
Contents
Properties
Public Functions
Reimplemented Public Functions
Public Slots
Protected Functions
Reimplemented Protected Functions
Detailed Description
QPushButton Class
The QPushButton widget provides a command button. More...
Header: #include <QPushButton>
qmake: QT += widgets
Inherits: QAbstractButton
Inherited By: QCommandLinkButton
List of all members, including inherited members


内容
索引
书签
搜索
索引
查找(L):
QPush
QPushButton
~QPushButton
Qt 5.8
Qt Widgets
C++ Classes
QAbstractButton
Contents
Properties
Public Functions
Public Slots
Signals
Protected Functions
Reimplemented Protected Functions
Detailed Description
QAbstractButton Class
The QAbstractButton class is the abstract base class of button widgets, providing functionalit
Header: #include <QAbstractButton>
qmake: QT += widgets
Inherits: QWidget

Signals

void **clicked**(bool *checked* = false)

void **pressed**()

void **released**()

void **toggled**(bool *checked*)

- 3 signals inherited from QWidget
- 2 signals inherited from QObject

单击信号

Protected Functions

virtual void **checkStateSet**()

virtual bool **hitButton**(const QPoint &*pos*) const

2、查看this中槽函数

内容 索引 书签 搜索

索引

查找(L):

Widget

Accessibility for QWidget Applications
QWidget
QWidget::DrawChildren
QWidget::DrawWindowBackground
QWidget::IgnoreMask
QWidget::RenderFlag
QWidget::RenderFlags
QWidgetAction
QWidgetItem
QWIDGETSIZE_MAX
QWidget
QWidgetAction
QWidgetItem

Open Pages

Widget Class | Qt Widgets 5.8

Qt 5.8 | Qt Widgets | C++ Classes | QWidget

Contents

槽函数

Public Slots

```
bool    close()
void    hide()
void    lower()
void    raise()
void    repaint()
void    setDisabled(bool disable)
void    setEnabled(bool)
void    setFocus()
void    setHidden(bool hidden)
void    setStyleSheet(const QString &styleSheet)
```

建立：信号和槽函数的关系使用connect

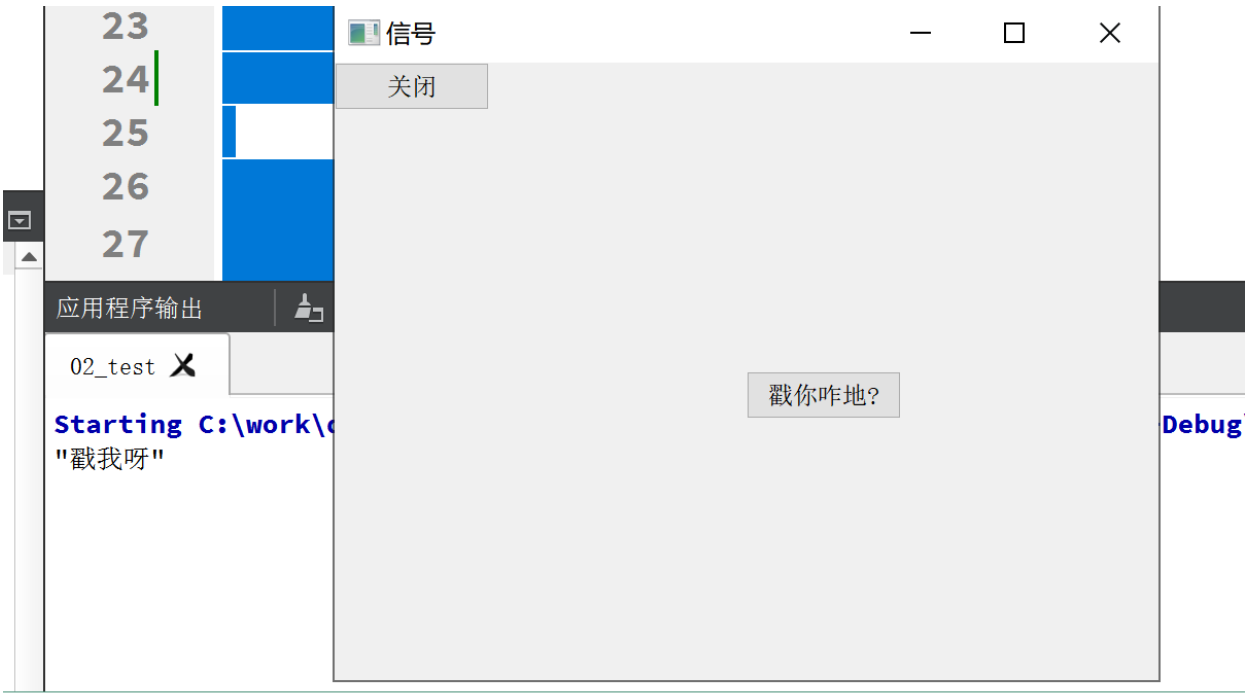
```
connect(信号的发起者,信号,接受者,操作);
```

widget的构造函数中

```
Widget::Widget(QWidget *parent)
 : QWidget(parent)
{
 //设置窗口标题
 this->setWindowTitle("信号");
 //设置窗口大小
 this->resize(800,600);
 //创建一个按钮
 QPushButton *btn1=new QPushButton("关闭",this);
  //需求：单击btn1 关闭主窗口
  //信号的发起者btn1 发出信号 主窗口this 关闭（槽函数）
  connect(btn1, &QPushButton::clicked, this, &QWidget::close);


  //lambda表达式
  QPushButton *btn2 =new QPushButton("戳我呀",this);
  btn2->move(400,300);
  connect(btn2, &QPushButton::clicked, [=](){
  //获取按钮上的文本
```

```
20  QString text = btn2->text();
21  qDebug()<<text <<endl;
22
23  //设置按钮的文本
24  btn2->setText("戳你咋地?");
25  } );
26 }
```

运行结果：

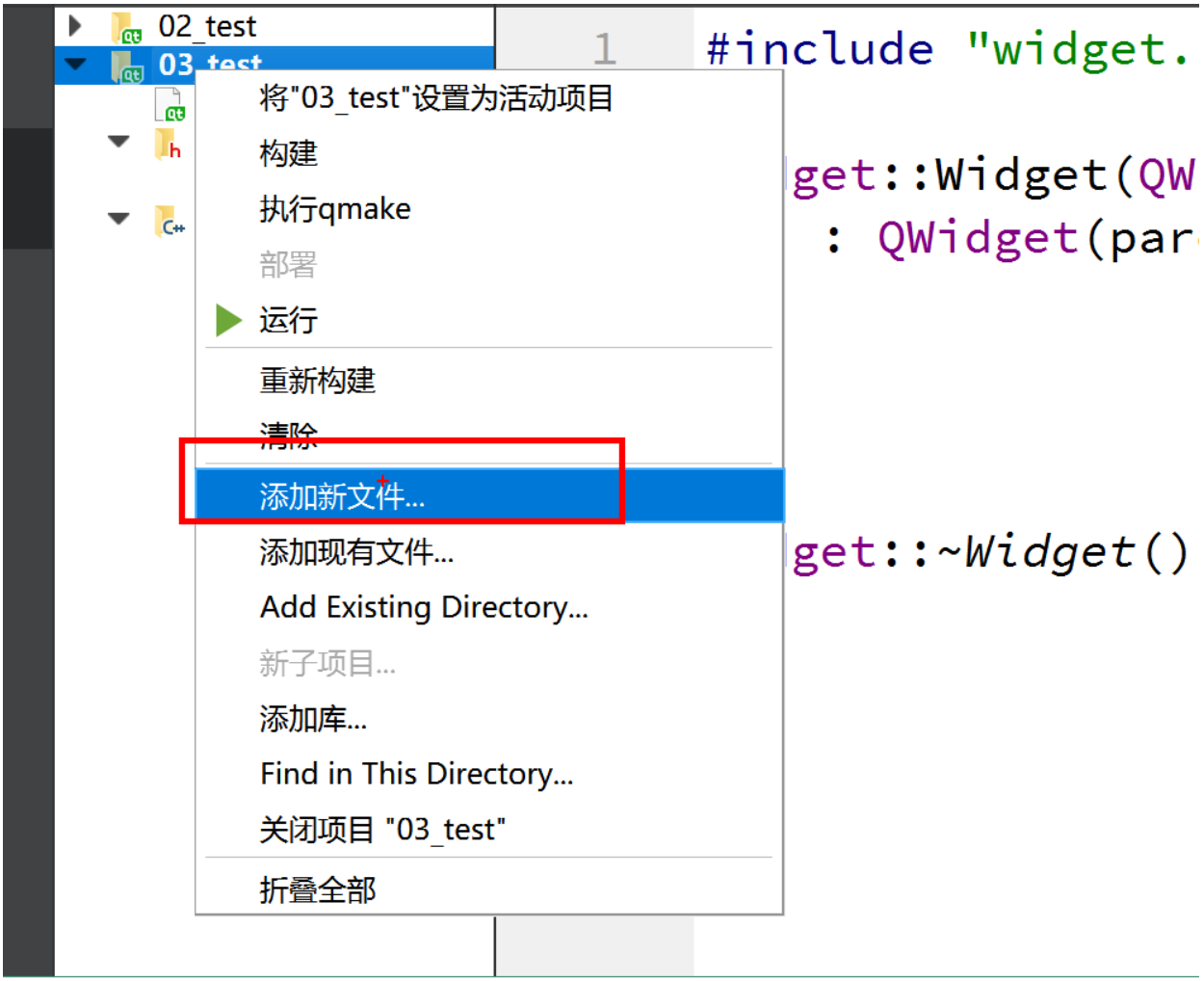


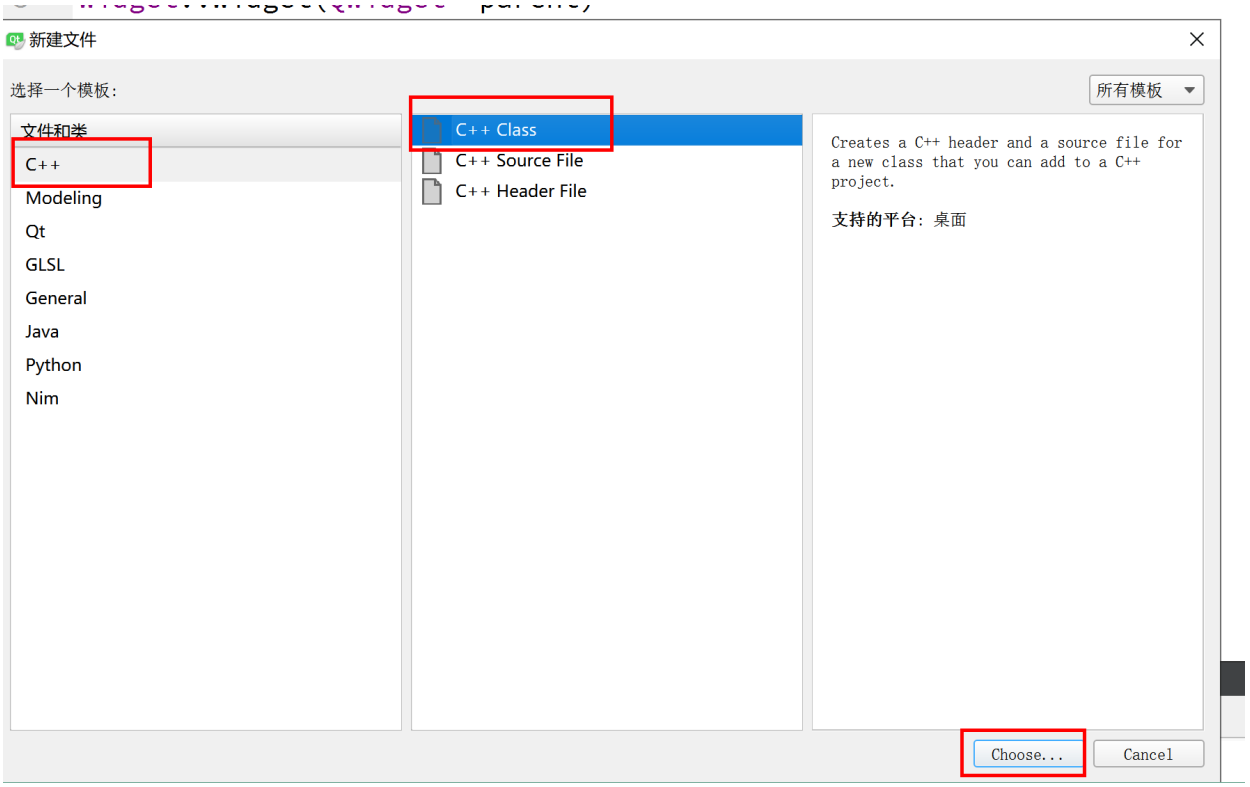
知识点7【自定义信号和槽】（了解）

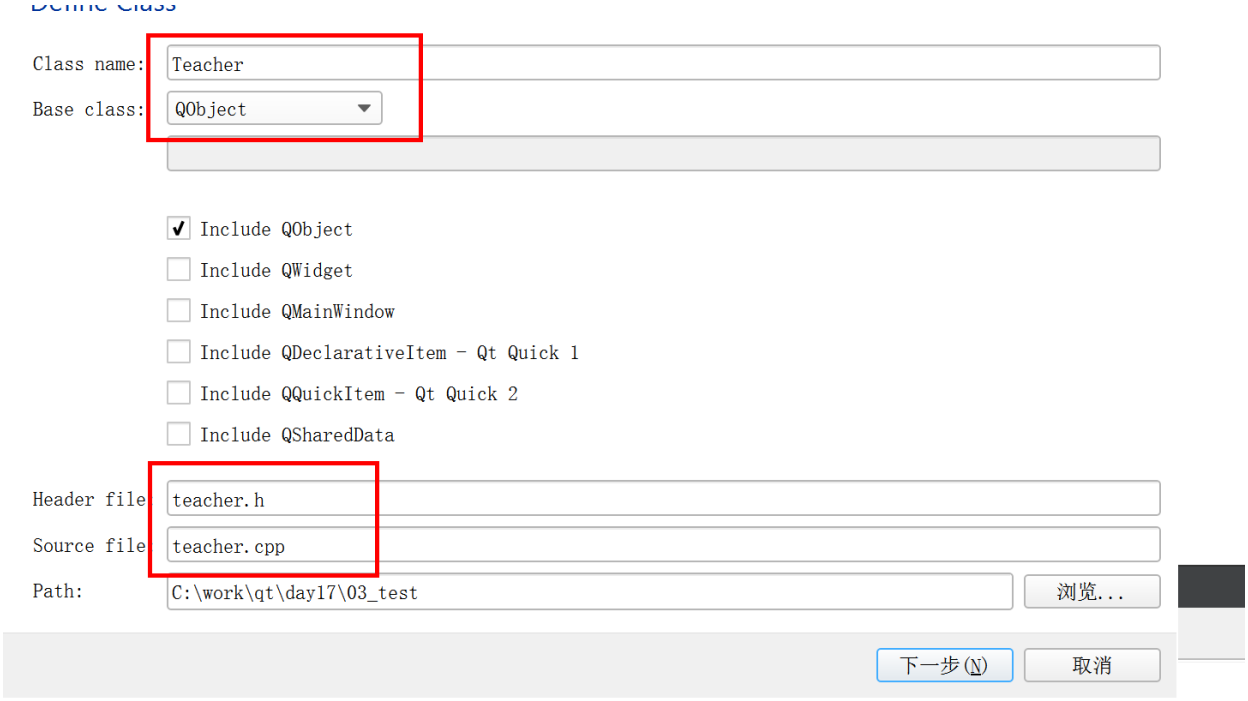
老师--->饿了信号　学生--->请老师吃饭

设计两个类：老师类 学生类

添加类的步骤：


02_test
03_test
将"03_test"设置为活动项目
构建
执行qmake
部署
运行
重新构建
清除
添加新文件...
添加现有文件...
Add Existing Directory...
新子项目...
添加库...
Find in This Directory...
关闭项目 "03_test"
折叠全部
1
#include "widget.
get::Widget(QW
: QWidget(par
get::~Widget()


新建文件
选择一个模板:
所有模板
文件和类
C++
Modeling
Qt
GLSL
General
Java
Python
Nim
C++ Class
C++ Source File
C++ Header File
Creates a C++ header and a source file for a new class that you can add to a C++ project.
支持的平台: 桌面
Choose...
Cancel



注意：

1、定义信号的规则：在signals下方

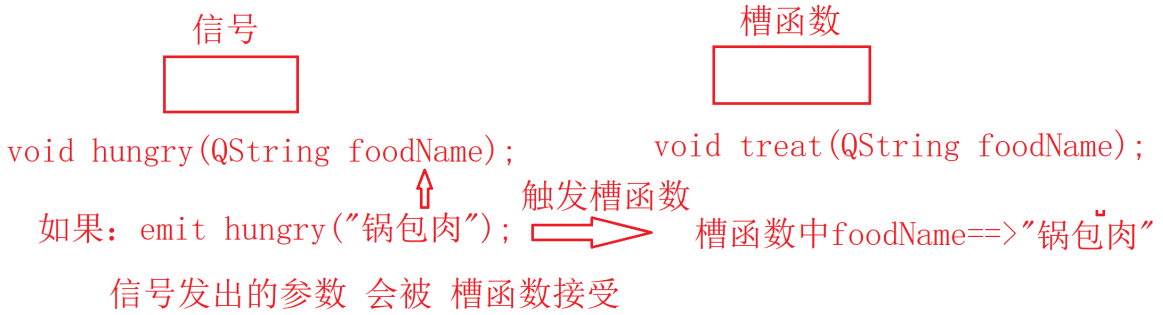
返回值类型为void　只需声明 不用实现　可以有参数 可以重载

2、定义槽函数的规则：public slots:

返回值类型为void　需要声明 需要实现　可以有参数 可以重载

3、用户可以使用emit 发出信号

```
emit tea->hungry();
```



案例：

student.h

```
#ifndef STUDENT_H
#define STUDENT_H

```

```
#include <QObject>

class Student : public QObject
{
 Q_OBJECT
public:
  explicit Student(QObject *parent = 0);

 signals:

 public slots:
  //返回值类型为void 需要声明 需要实现 可以有参数 可以重载
  void treat();
  void treat(QString foodName);
 };

 #endif // STUDENT_H

```

student.cpp

```
#include "student.h"
#include<QDebug>
Student::Student(QObject *parent) : QObject(parent)
{

}

void Student::treat()
{
  qDebug()<<"请老师吃饭了"<<endl;
 }

 void Student::treat(QString foodName)
 {
  qDebug()<<"请老师吃饭了"<<foodName<<endl;
 }

```

teacher.h

```
#ifndef TEACHER_H
#define TEACHER_H

```

```
#include <QObject>
#include<QString>
class Teacher : public QObject
{
 Q_OBJECT
public:
  explicit Teacher(QObject *parent = 0);

 signals://自定义信号函数
  //定义信号的规则：返回值类型void 只需声明 不用实现 可以有参数 可以重载
  void hungry();
  void hungry(QString foodName);
 public slots://自定义槽函数
 };

 #endif // TEACHER_H

```

teacher.cpp

```
#include "teacher.h"

Teacher::Teacher(QObject *parent) : QObject(parent)
{

}

```

widget构造函数：

```
#include "widget.h"
#include"teacher.h"
#include"student.h"
#include<QPushButton>
#define N 0
Widget::Widget(QWidget *parent)
  : QWidget(parent)
{
 //设置主窗口大小
  this->resize(800,600);
  //实例化一个老师
  Teacher *tea = new Teacher(this);
  //实例化一个学生
  Student *stu = new Student(this);
```

```
#if N
 //connect建立老师和学生的关系
 void (Teacher:: *p1)() = &Teacher::hungry;
 void (Student:: *p2)() = &Student::treat;
 connect(tea,p1, stu, p2);
#else
 //报错的原因：hungry信号被重载了
 //通过函数指针 明确那个信号
 void (Teacher:: *p1)(QString foodName) = &Teacher::hungry;
 void (Student:: *p2)(QString foodName) = &Student::treat;
 connect(tea,p1, stu, p2);
#endif
 //下课了 老师饿了
 QPushButton *btn = new QPushButton("下课", this);
 connect(btn,&QPushButton::clicked,[=](){
#if N
 //下课函数：让老师发出饿的信号
 emit tea->hungry();
#else
 //下课函数：让老师发出饿的信号
 emit tea->hungry("锅包肉");
#endif
 });

}

Widget::~Widget()
{

}

```

运行结果：

总结:

1、一个信号可以和多个槽相连

如果是这种情况，这些槽会一个接一个的被调用，但是它们的调用顺序是不确定的。

2、多个信号可以连接到一个槽

只要任意一个信号发出，这个槽就会被调用。

3、一个信号可以连接到另外的一个信号

当第一个信号发出时，第二个信号被发出。除此之外，这种信号-信号的形式和信号-槽的形式没有什么区别。

4、槽可以被取消链接

这种情况并不经常出现，因为当一个对象delete之后，Qt自动取消所有连接到这个对象上面的槽。

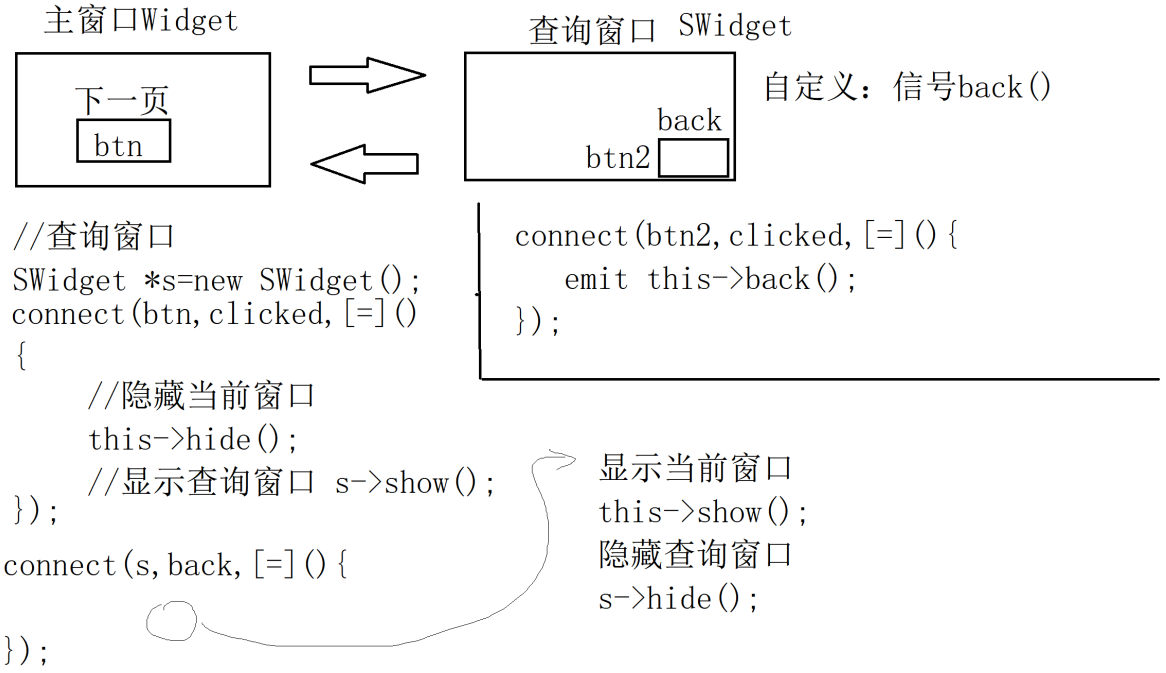
5、Qt4版本的信号槽写法

```
connect(tea,SIGNAL(hungry()), stu, SLOT(treat()))
```

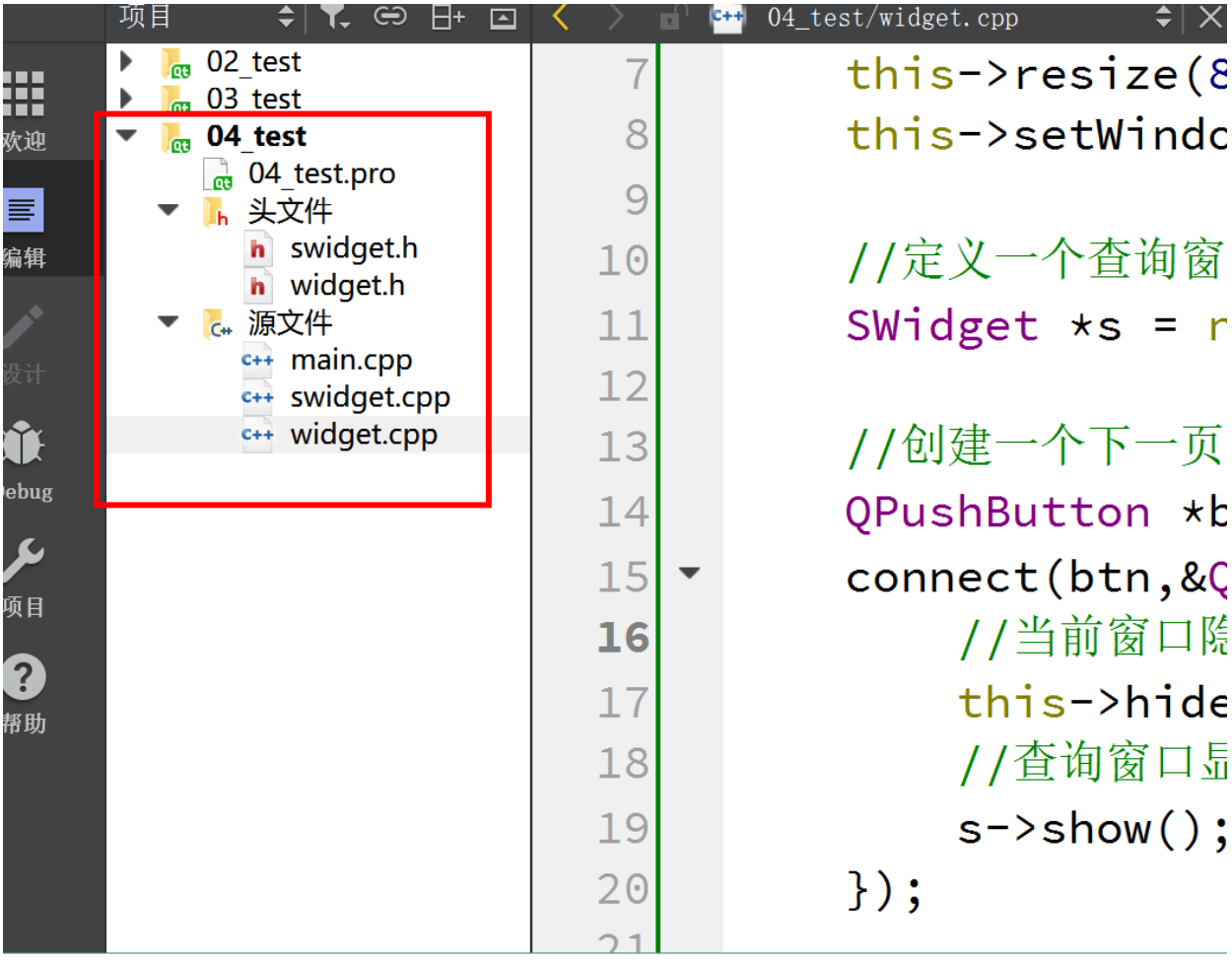
6、Qt5版本的信号槽写法

```
connect(tea, &Teacher::hungry, stu, &Student::treat)
```

知识点8【窗口切换】



工程文件：



swidget.h

```
#ifndef SWIDGET_H
#define SWIDGET_H

```

```
#include <QWidget>

class SWidget : public QWidget
{
 Q_OBJECT
public:
  explicit SWidget(QWidget *parent = 0);

 signals:
  void back();//回退信号
 public slots:
 };

 #endif // SWIDGET_H

```

swidget.cpp

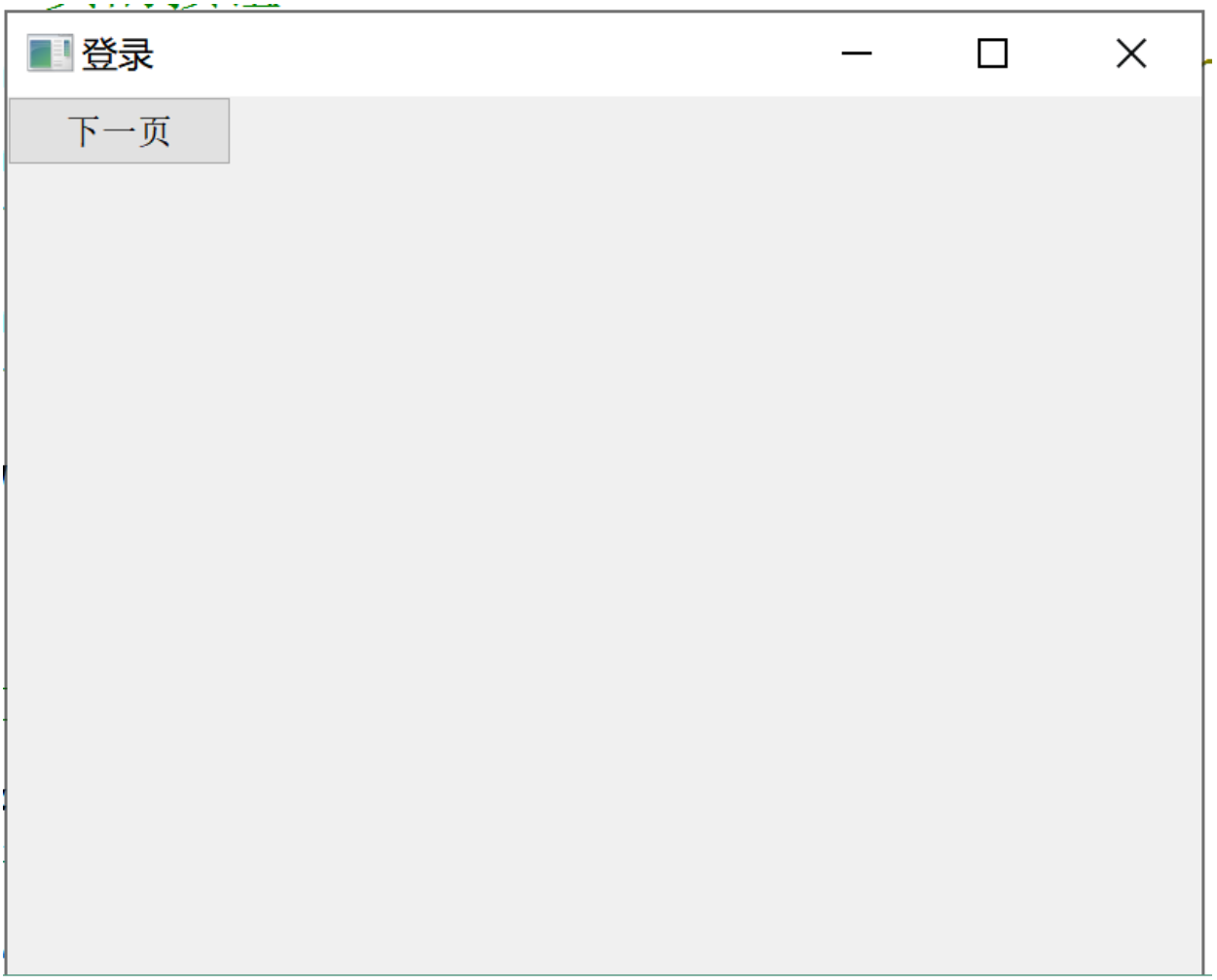
```
#include "swidget.h"
#include<QPushButton>
SWidget::SWidget(QWidget *parent) : QWidget(parent)
{
 //设置查询窗口的大小
 this->resize(800,600);
 this->setWindowTitle("查询窗口");

 //定义一个回退按钮
  QPushButton *btn = new QPushButton("back",this);

  //当按下back 就发出一个back信号
  connect(btn,&QPushButton::clicked,[=](){
  emit this->back();
  });
 }

```

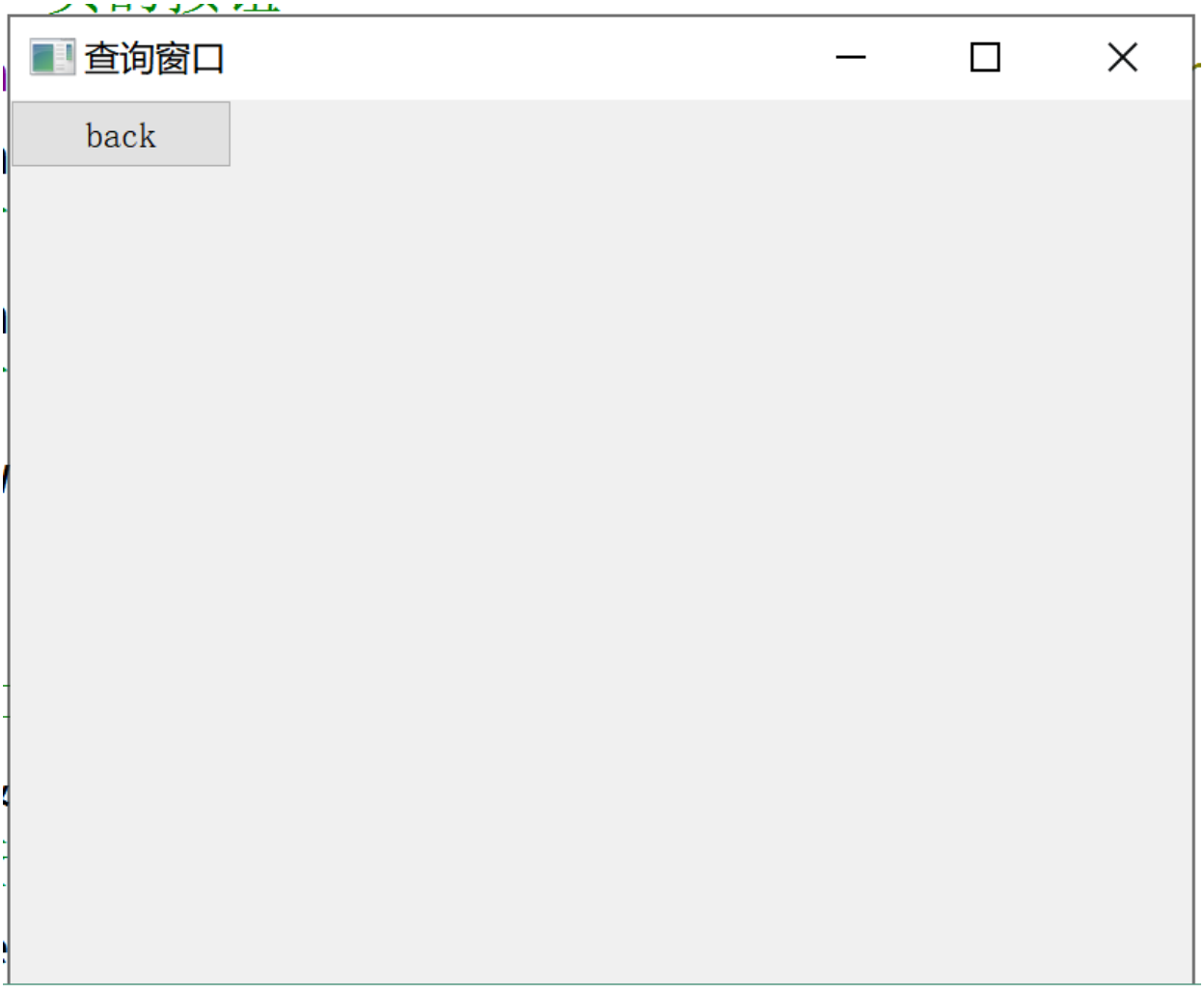
widget.cpp

```
#include "widget.h"
#include <QPushButton>
#include"swidget.h"
Widget::Widget(QWidget *parent)
  : QWidget(parent)
{
```

```
7   this->resize(800,600);
8   this->setWindowTitle("登录");
9
10   //定义一个查询窗口
11   SWidget *s = new SWidget();
12
13   //创建一个下一页的按钮
14   QPushButton *btn = new QPushButton("下一页",this);
15   connect(btn,&QPushButton::clicked,[=](){
16   //当前窗口隐藏
17   this->hide();
18   //查询窗口显示
19   s->show();
20   });
21
22   //监测查询窗口s的回退信号
23   connect(s,&SWidget::back,[=](){
24   //隐藏查询窗口
25   s->hide();
26   //显示当前窗口
27   this->show();
28   });
29
30  }
31
32  Widget::~Widget()
33  {
34
35  }
36
```

运行结果：


登录
下一页


查询窗口
back