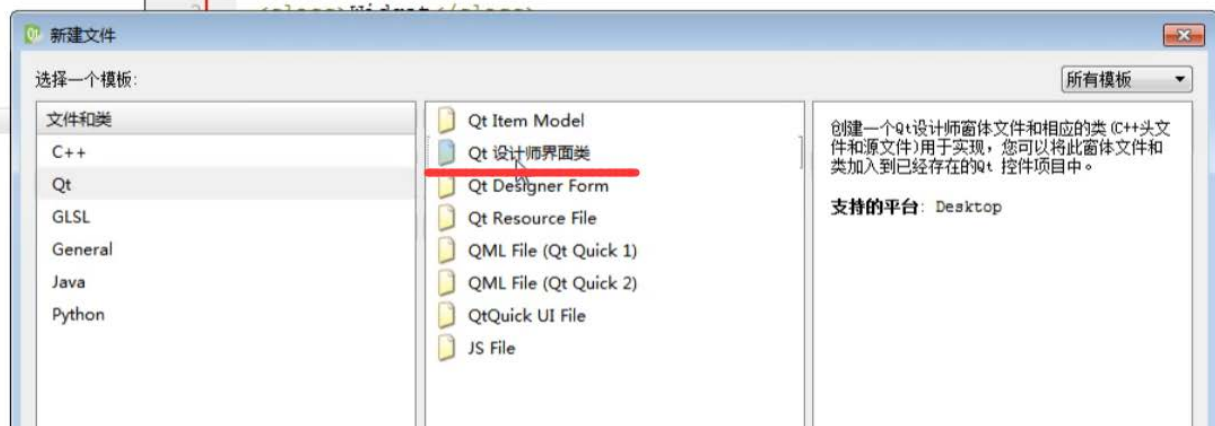


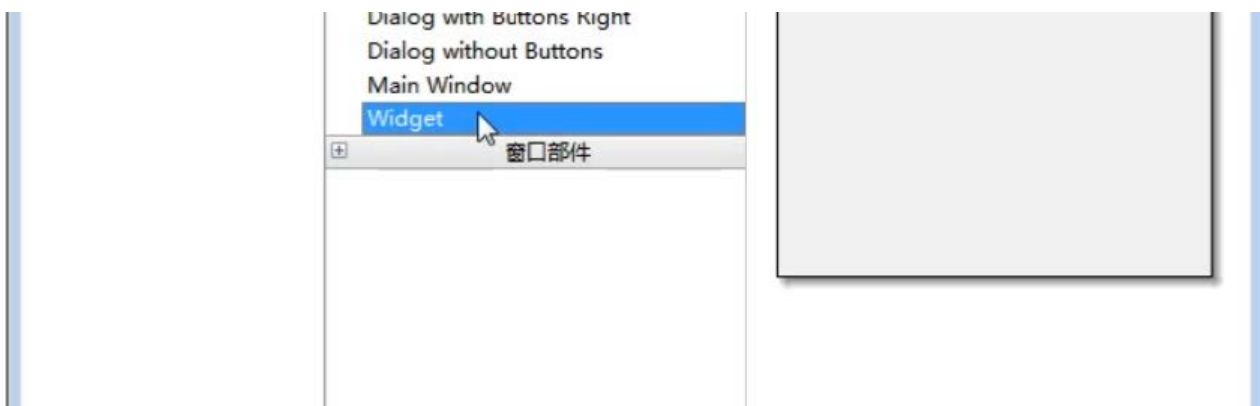
封装一个小控件

第一步

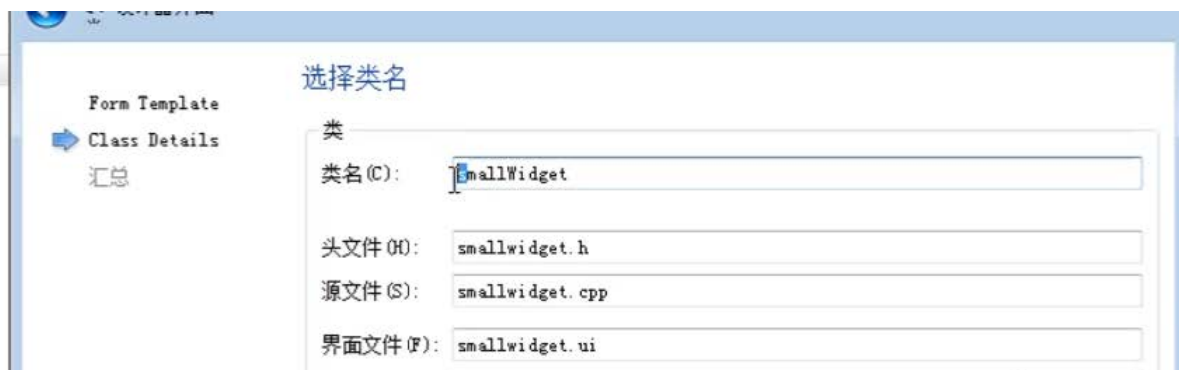
添加设计师界面类



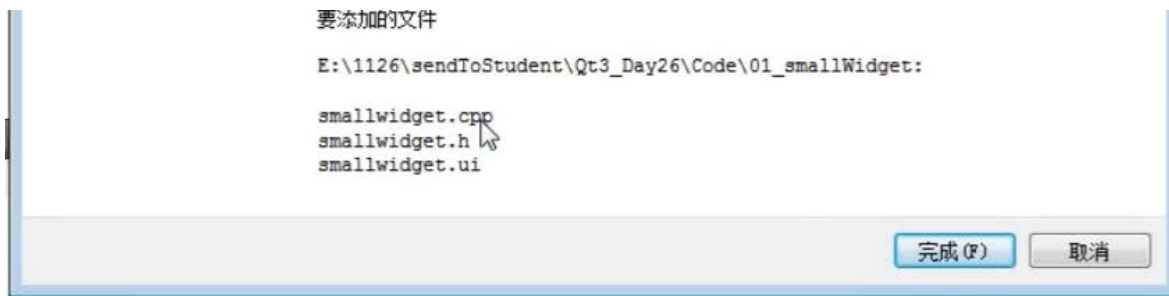
选择空窗口即可



起名



自动生成 .h .cpp .ui

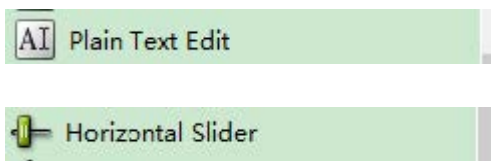


完成后自动定位到ui界面



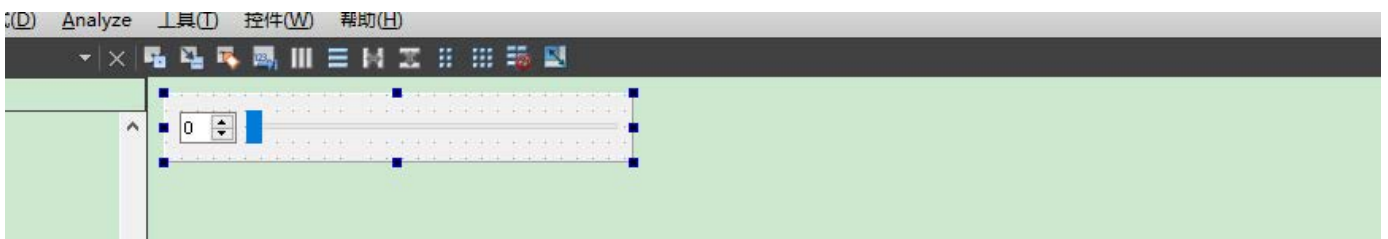
第二步

在设计师界面ui添加控件

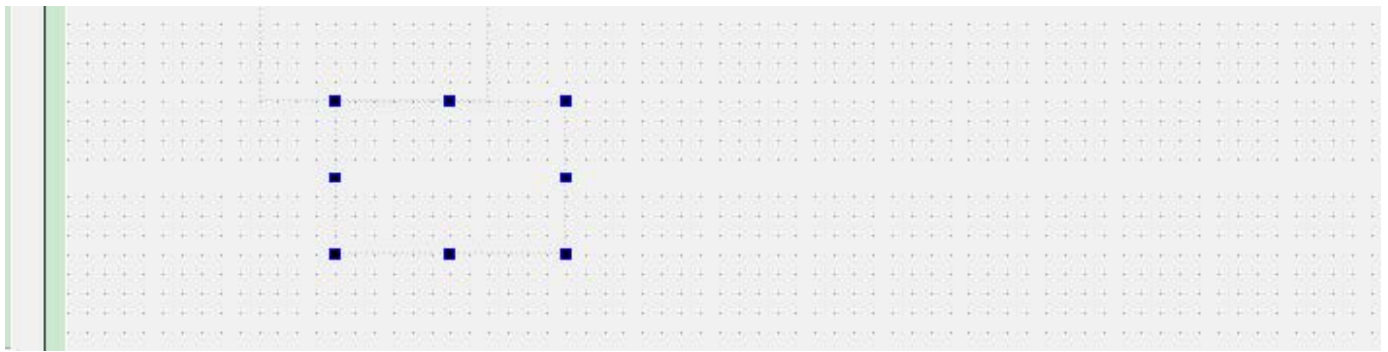


垂直布局

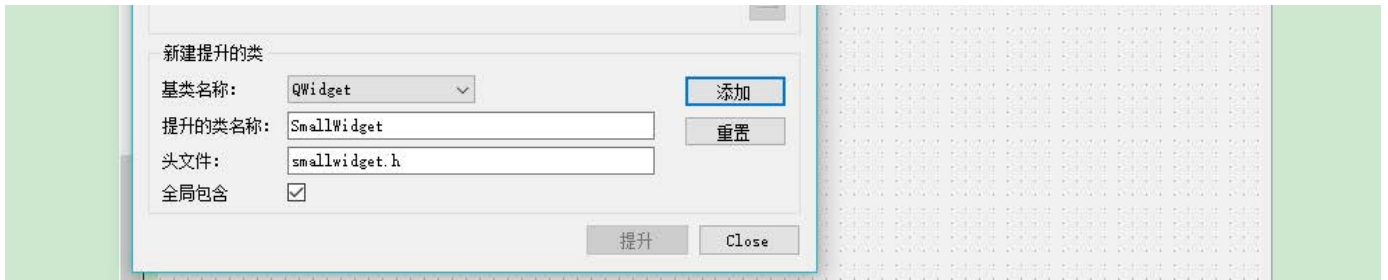
缩放一下



然后，在原本的ui界面添加一块widget

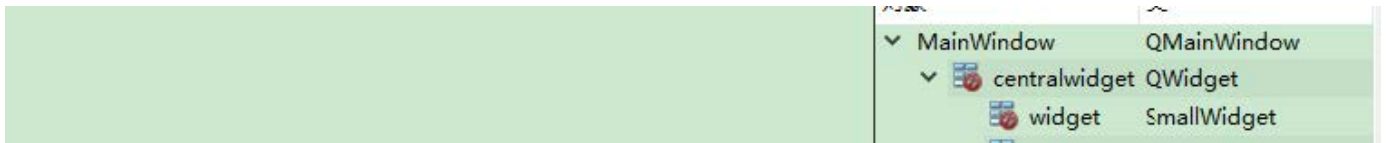
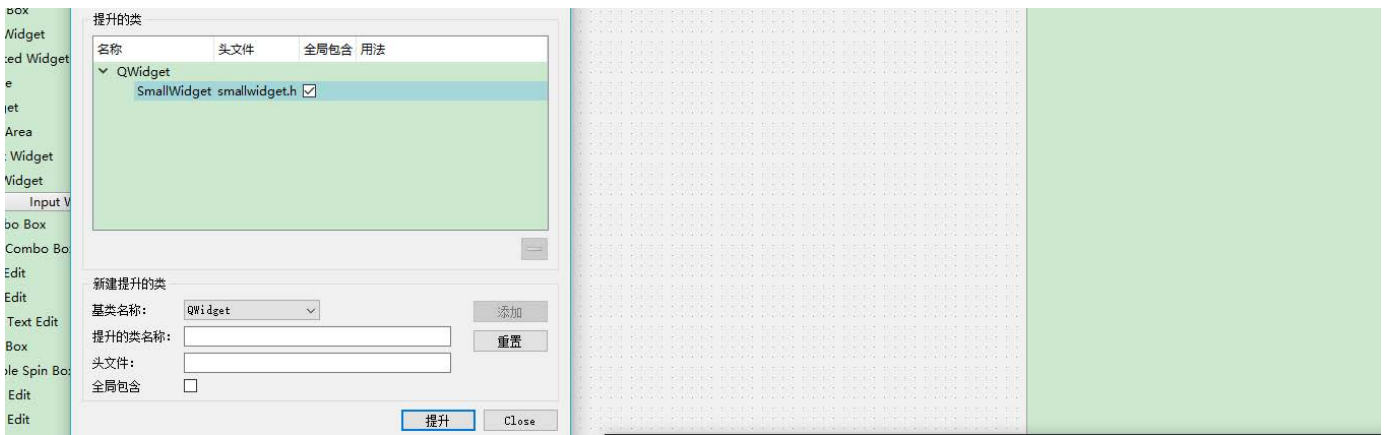


点击添加



点击提升

可以看到，此时，右侧窗口的QWidget变成了一开始的ui类名称——SmallWidget



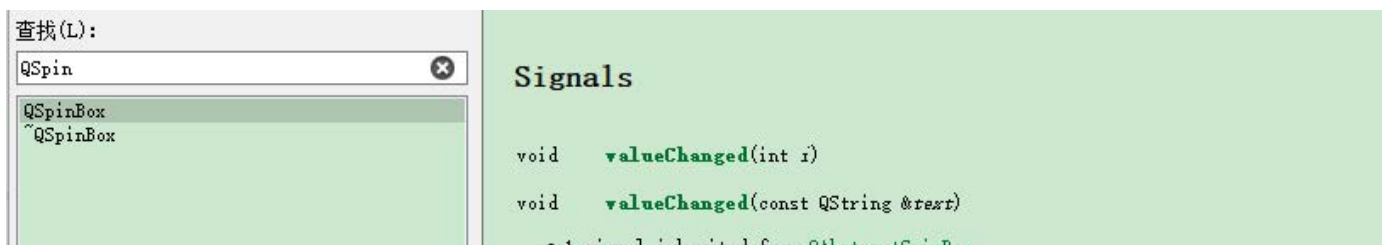
由于此前我们勾选了全局包含

所以我们新建的Widget直接右键也可以进行提升

第三步

由于现在我们都控件并没有达到预期，在代码中进行调整（连接）

通过帮助文档查询包含的信号



我们采用int类型

注意：滚动条只有一个类——QSlider

但是QSlider没有槽函数，我们去看看它的父类



我们选择setValue

```
connect(ui->spinBox, &QSpinBox::valueChanged, ui->horizontalSlider, &QSlider::setValue);
```

但此时依然无法使用因为Slider的信号默认发生了重载

第四步

我们需要用函数指针去选择我们需要的重载版本

```
//在帮助文档中，我们看到SpinBox的信号只有一个，且发生重载，所以编译器不知道我们选择哪个
//因此，我们定义一个指向&QSpinBox::valueChanged的函数指针，且类型为int
//对应我们需要的信号
void (QSpinBox:: *spSignal)(int) = &QSpinBox::valueChanged;
```

```
//在帮助文档中，我们看到SpinBox的信号只有一个，且发生重载，所以编译器不知道我们选择哪个
//因此，我们定义一个指向&QSpinBox::valueChanged的函数指针，且类型为int
//对应我们需要的信号
void (QSpinBox:: *spSignal)(int) = &QSpinBox::valueChanged;

connect(ui->spinBox,spSignal,ui->horizontalSlider,&QSlider::setValue);
//信号 槽函数
connect(ui->horizontalSlider,&QSlider::valueChanged,ui->spinBox,&QSpinBox::setValue);
//信号 槽函数
```

哦豁，万事大吉

控件功能添加

我们想添加两个小功能：

第一个

点击之后定位到进度条一半

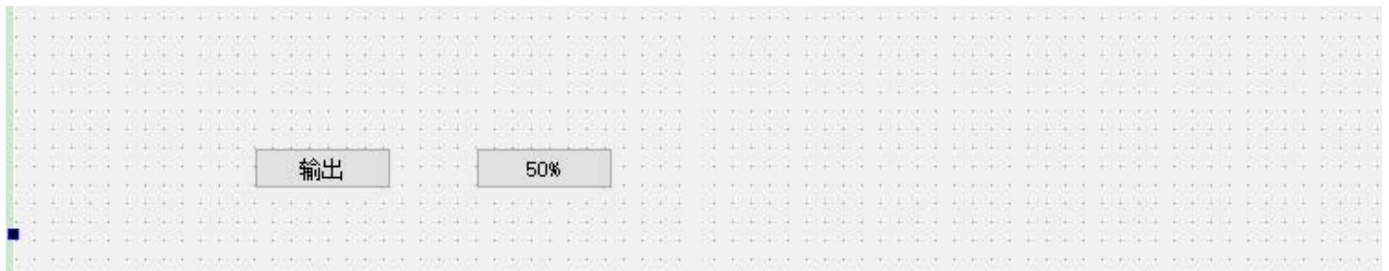
第二个

点击之后qDebug输出进度数值

开干

第一步

在我们自己的ui中添加两个按钮



第二步

在smallwidget.h中声明函数

```
#ifndef SMALLWIDGET_H
#define SMALLWIDGET_H

#include <QWidget>

namespace Ui {
class SmallWidget;
}

class SmallWidget : public QWidget
{
    Q_OBJECT

public:
    explicit SmallWidget(QWidget *parent = nullptr);
    ~SmallWidget();
    //设置数字
    void setNum(int num);

    //获取数字
    int getNum();
};
```

```
private:
    Ui::SmallWidget *ui;
};

#endif // SMALLWIDGET_H
```

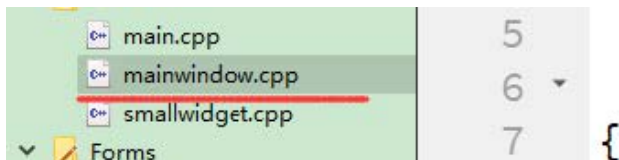
第三步

SmallWidget.cpp中的

```
//设置数字
void SmallWidget::setNum(int num)
{
    ui->spinBox->setValue(num);
}
//获得数字
int SmallWidget::getNum()
{
    return ui->spinBox->value();
}
```

第四步

我们在窗口文件中编辑



```
: QMainWindow(parent)
, ui(new Ui::MainWindow)
```

```
//获得数字
ui->setupUi(this);
connect(ui->pushButton1,&QPushButton::clicked,[=]() {
    .....
    qDebug()<<ui->widget->getNum();
});
//设置数字
connect(ui->pushButton2,&QPushButton::clicked,[=]() {
    .....
    ui->widget->setNum(50);
});
```

查看结果

```

ui(new Ui::MainWindow)

获得数字
->setupUi(this);
connect(ui->pushButton1,&QPushButton::clicked,
        qDebug()<<ui->widget->getNum()
        );
设置数字
connect(ui->pushButton2,&QPushButton::clicked,
        qDebug()<<ui->widget->setNum(50)
        );

```



21:21:47: Starting C:\Users\Sakura\Desktop\QT_Code\build-Controls_He-Desktop_Qt_5_12_10_MinGW_64_bit-Debug\debug\Controls_He.e

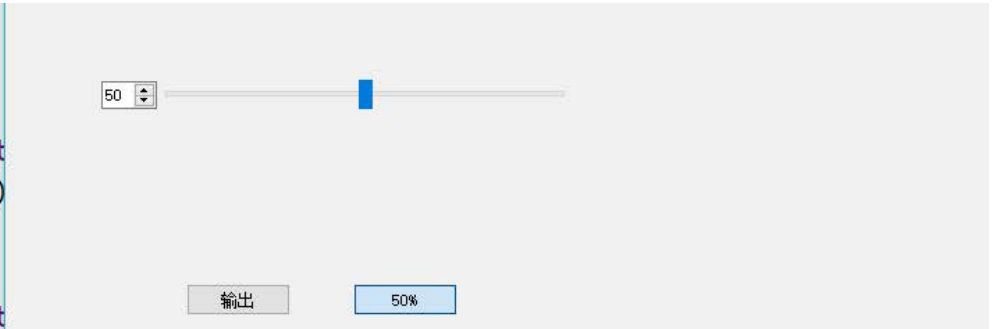
21
Fully register Qt versions and kits? To do this later, select Options > Kits > Qt Versions > Link with Qt.

```

this);
pushButton1,&QPushButton::clicked,
(ui->widget->getNum()

pushButton2,&QPushButton::clicked,
(ui->widget->setNum(50)

```



总结:

- 1.1 添加新文件 - Qt - 设计师界面类 (.h .cpp .ui)
- 1.2 .ui中 设计 QSpinBox和QSlider 两个控件
- 1.3 Widget中使用自定义控件，拖拽一个Widget，点击提升为，点击添加，点击提升
- 1.4 实现功能，改变数字，滑动条跟着移动，信号槽监听。
- 1.5 提供 getNum 和 setNum对外接口