编译错误：

10:14:27: 为项目QTPrjOne执行步骤 ...

**10:14:27: Could not create directory "C:\Users\JRWX\Desktop\MQL\QTPrjOne\Error in " Util.asciify("build-QTPrjOne-Desktop\_Qt\_5\_8\_0\_MSVC2015\_32bit-Debug")": TypeError: Property 'asciify' of object Core::Internal::UtilsJsExtension(0xbdba328) is not a function"**

Error while building/deploying project QTPrjOne (kit: Desktop Qt 5.8.0 MSVC2015 32bit)

The kit Desktop Qt 5.8.0 MSVC2015 32bit has configuration issues which might be the root cause for this problem.

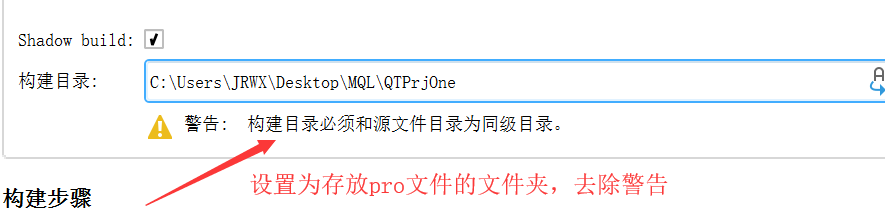
When executing step "qmake"

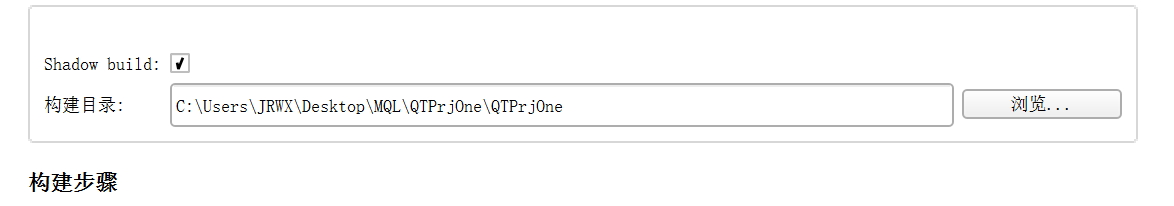
10:14:27: Elapsed time: 00:00.

解决方案：



将上面修改成绝对路径





C++中#ifndef, #define, #endif的作用和使用的注意事项，#开头的都是预编译指令，就是在正式编译之前，编译器做一些预处理的工作。防止重复引入，重复编译（报重定义的错误）。预处理指令是在编译器进行编译之前进行的操作.预处理过程扫描源代码，对其进行初步的转换，产生新的源代码提供给编译器。可见预处理过程先于编译器对源代码进行处理。在很多编程语言中，并没有任何内在的机制来完成如下一些功能：在编译时包含其他源文件、定义宏、根据条件决定编译时是否包含某些代码(防止重复包含某些文件)。要完成这些工作，就需要使用预处理程序。尽管在目前绝大多数编译器都包含了预处理程序，但通常认为它们是独立于编译器的。预处理过程读入源代码，检查包含预处理指令的语句和宏定义，并对源代码进行响应的转换。预处理过程还会删除程序中的注释和多余的空白字符。

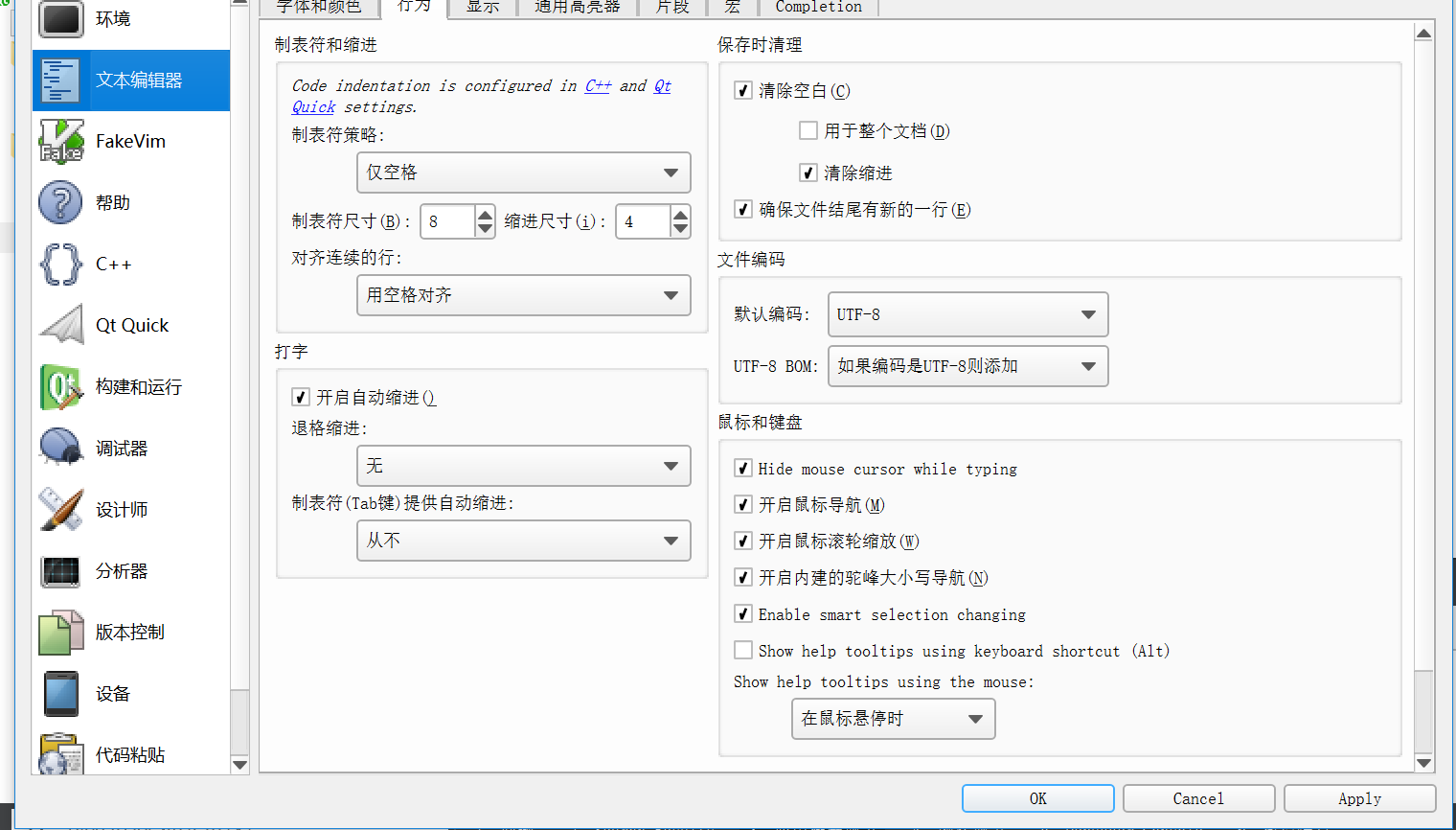
Q\_OBJECT宏，允许类中使用信号和槽的机制

编译错误：

文件是按钮utf-8编码保存的，如下：



并且作了QT Creater的公共设置，如下：



但是在cpp中存在中文时，编译出现错误“常量中有换行符”

解决方案:

原因定位在编译器，vs的编译器不能识别中文。再做了如上公共设置之后，需要再敲几个中文，再编译就好了。

编译好了，但是不管UI上的中文，还是控制台打印的中文，输出还是乱码，就需要在源码里使用编码函数解决了：

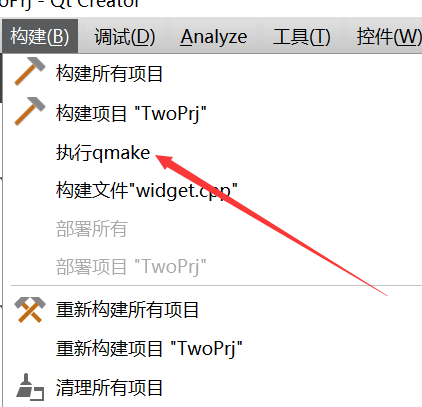
btn->setText(QStringLiteral("自定义按钮"));

编译错误：

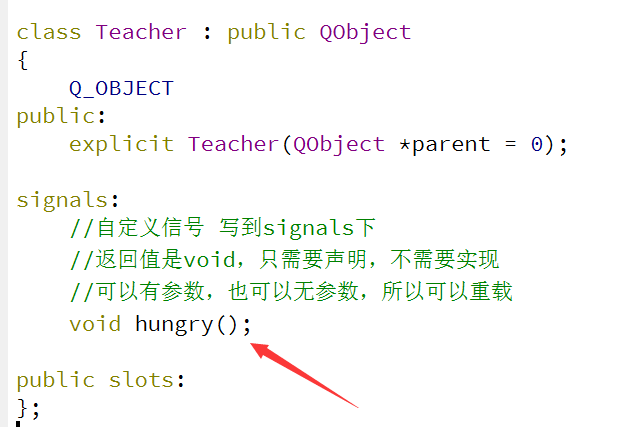
我创建了一个教师类，并且在widget中创建了该对象，编译时

widget.obj:-1: error: LNK2019: 无法解析的外部符号 "public: \_\_thiscall Teacher::Teacher(class QObject \*)" (??0Teacher@@QAE@PAVQObject@@@Z)，该符号在函数 "public: \_\_thiscall Widget::Widget(class QWidget \*)" (??0Widget@@QAE@PAVQWidget@@@Z) 中被引用

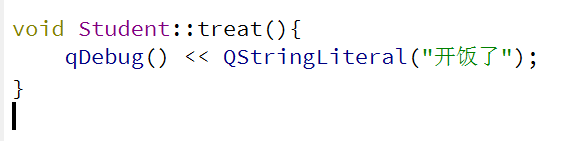
解决方案:

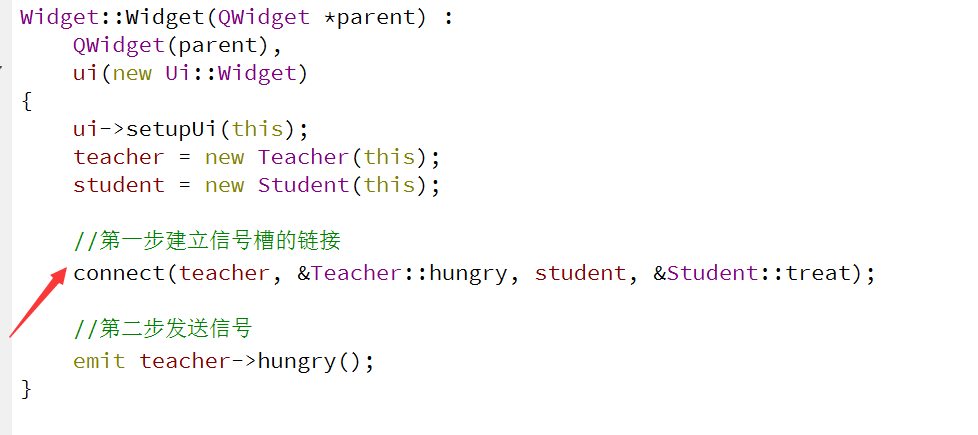


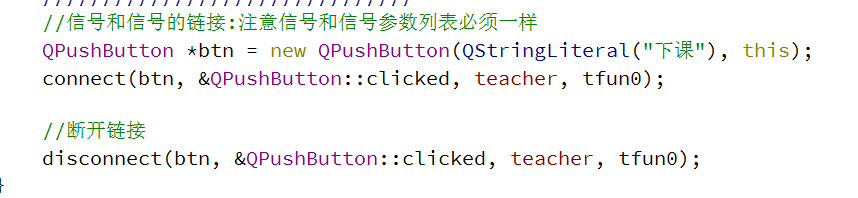
信号槽

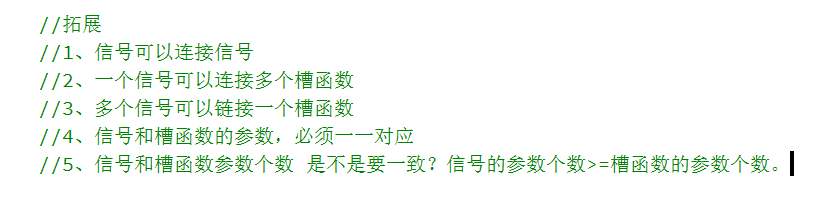






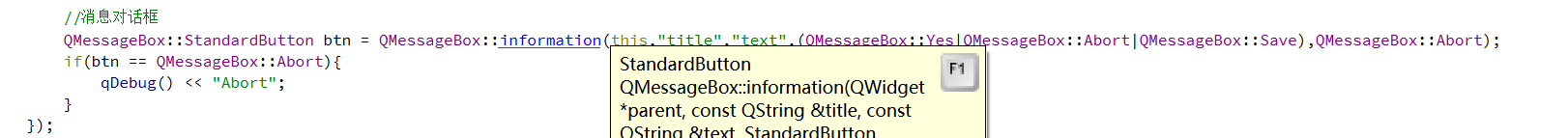














问题：

在可视化设计模板中的控件和运行起来后大小不一致

解决方案：

QT对高分辨率屏幕的显示支持问题，在mian函数中一开始加上#if (QT\_VERSION >= QT\_VERSION\_CHECK(5, 6, 0))  
QCoreApplication::setAttribute(Qt::AA\_EnableHighDpiScaling);  
#endif

void f(vector<int> &v)

{

v[5]; *// A*

v.at[5]; *// B*

}

如果v非空，A行和B行没有任何区别。如果v为空，B行会抛出std::out\_of\_range异常，A行的行为未定义。

c++标准不要求vector<T>::operator[]进行下标越界检查，原因是为了效率，总是强制下标越界检查会增加程序的性能开销。设计vector是用来代替内置数组的，所以效率问题也应该考虑。不过使用operator[]就要自己承担越界风险了。

如果需要下标越界检查，请使用at。但是请注意，这时候的性能也是响应的会受影响，因为越界检查增加了性能的开销。

重载的信号或重载的槽

QStringList strList;

strList.append(QStringLiteral("宝马"));

strList.append(QStringLiteral("奥迪"));

strList.append(QStringLiteral("奔驰"));

ui->comboBox->addItems(strList);

//void currentIndexChanged(int index);

//void currentIndexChanged(const QString &);

void (QComboBox::\*fun)(int) = &QComboBox::currentIndexChanged;

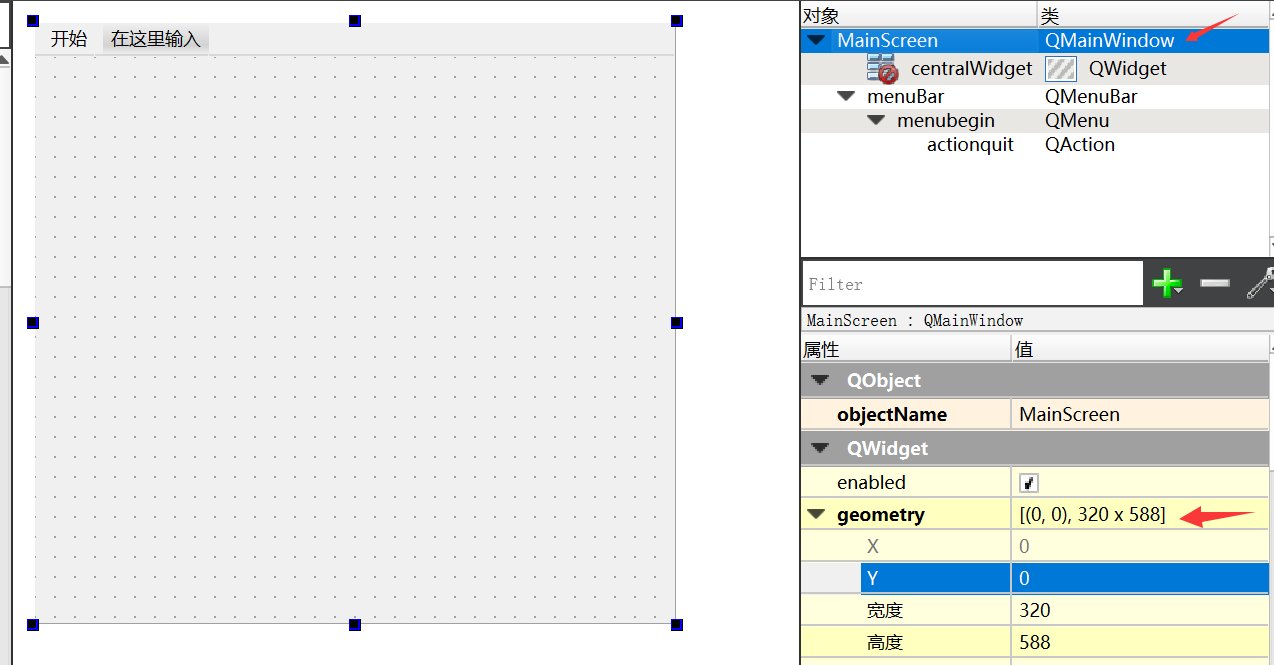
connect(ui->comboBox,fun, [=](int index){

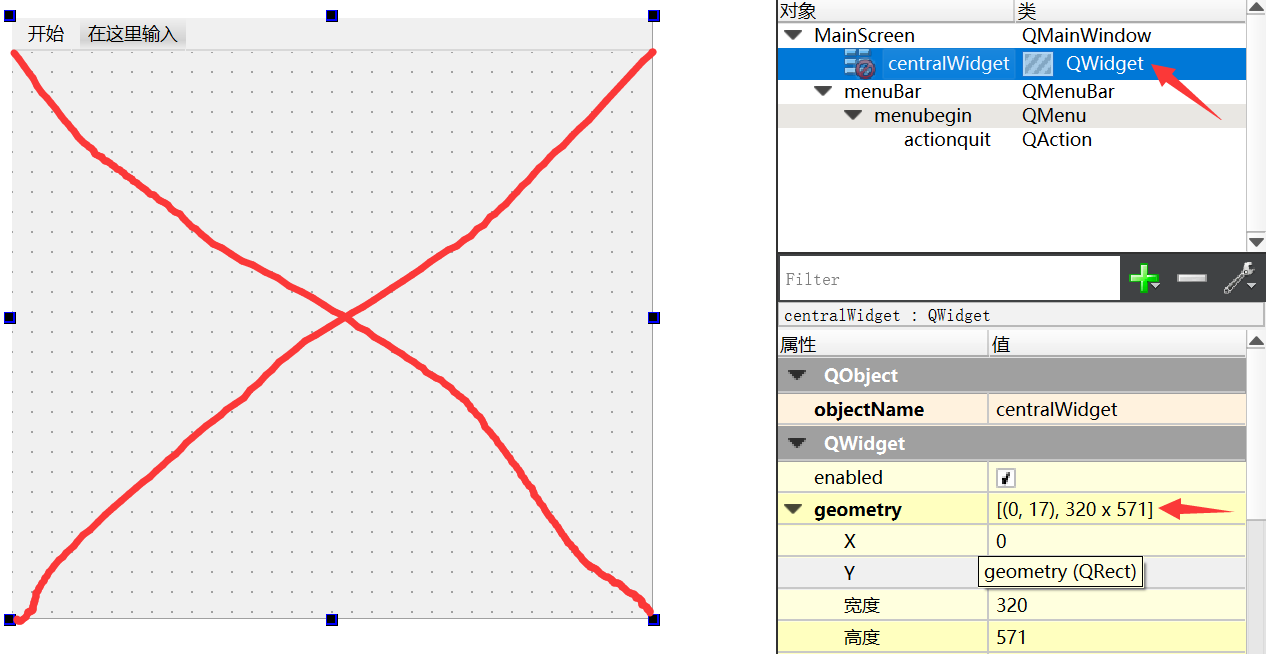
qDebug() << QString::number(index);

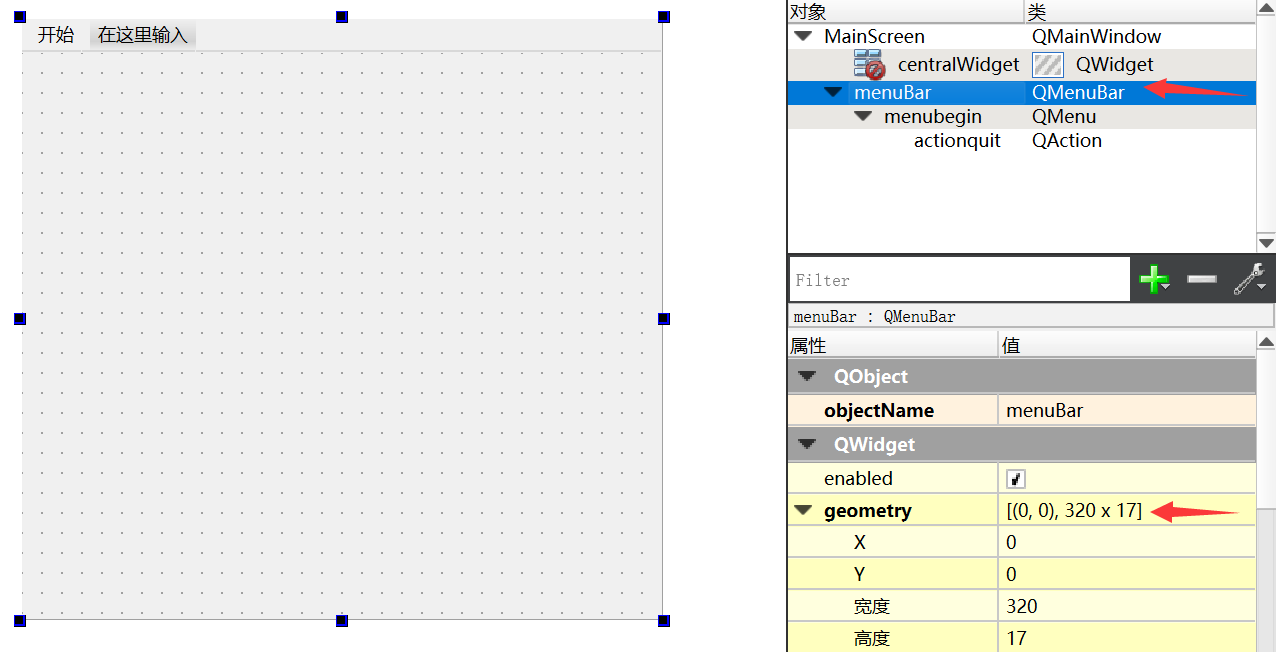
});

问题：

使用派生自QMainWindow的窗口







说明：



QFont font ( “Microsoft YaHei”, 10, 75); //第一个属性是字体（微软雅黑），第二个是大小，第三个是加粗（权重是75）

ui->label->setFont(font);

常见权重

QFont::Light - 25 高亮

QFont::Normal - 50 正常

QFont::DemiBold - 63 半粗体

QFont::Bold - 75 粗体

QFont::Black - 87 黑体

————————————————

控件样式：

//将QLabel设置自定义样式

label->setStyleSheet("QLabel{border-width: 1px 1px 1px 1px;"

"border-style: solid;"

"border-color: rgb(255,0,0);"

"background-color: rgba(213, 213, 213, 125);"

"border-radius:5px;}");

//将按钮设置为“不规则样式”

this->setStyleSheet("QPushButton{border:0px;}");