窗口类

基本概念

QMainWindow是一个为用户提供主窗口程序的类，包含一个菜单栏（menu bar）、多个工具栏(tool bars)、多个锚接部件(dock widgets)、一个状态栏(status bar)及一个中心部件(central widget)，是许多应用程序的基础，如文本编辑器，图片编辑器等。

菜单栏

1.重置窗口大小

resize(60,400);

2.菜单栏创建

QMenuBar *bar=menuBar();//创建

QMenuBar默认放在对象树下

setMenuBar(bar);//将菜单栏放进窗口中

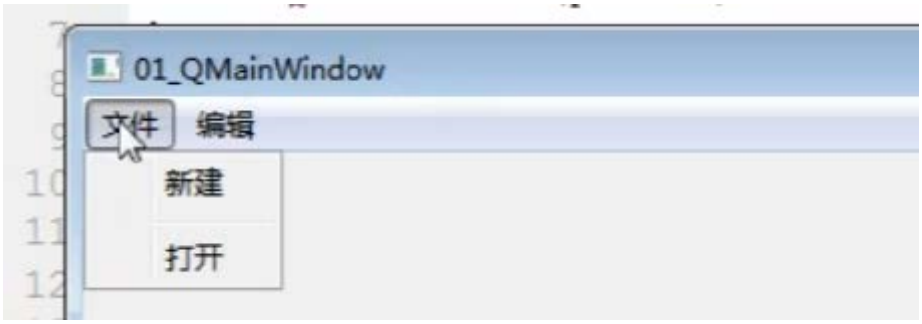
包含头文件

```
#include<QMenuBar>
```

QMenu * fileMenu=bar->addMenu("文件")；//创建菜单

addMenu的返回值是QMenu

```
resize(600,400);
    QMenuBar *bar = menuBar();//创建菜单指针
    setMenuBar(bar);//放进窗口
    QMenu *fileMenu = bar->addMenu("File");
    QMenu *editMenu = bar->addMenu("Editor");
    fileMenu->addAction("Create");//创建菜单项  1
    fileMenu->addSeparator();//添加分割线
    fileMenu->addAction("Open");//创建菜单项   2
```



注意：菜单栏最多只能有一个

工具栏可以有多个

需要包含头文件#include

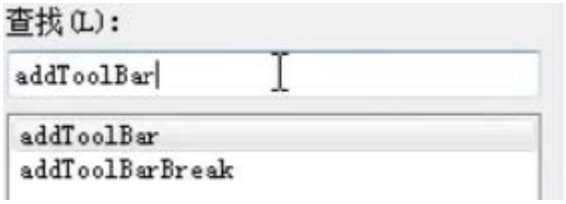
```
//工具栏   可以有多个
QToolBar * toolBar = new QToolBar(this);
addToolBar(toolBar);
```

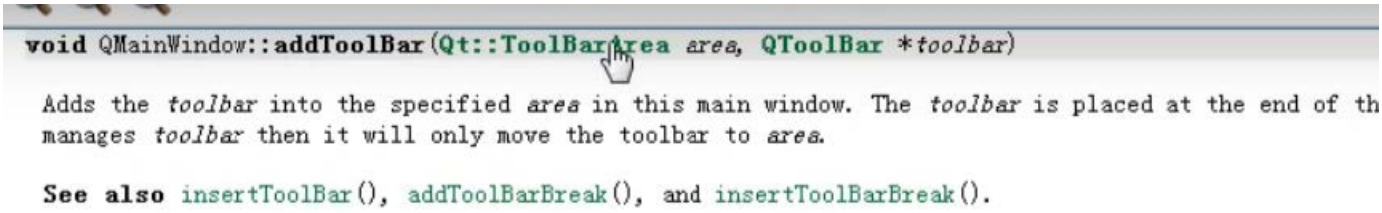
可以看到addToolBar的第三种方式的参数有两个

Qt::toolBarArea //默认停靠区域

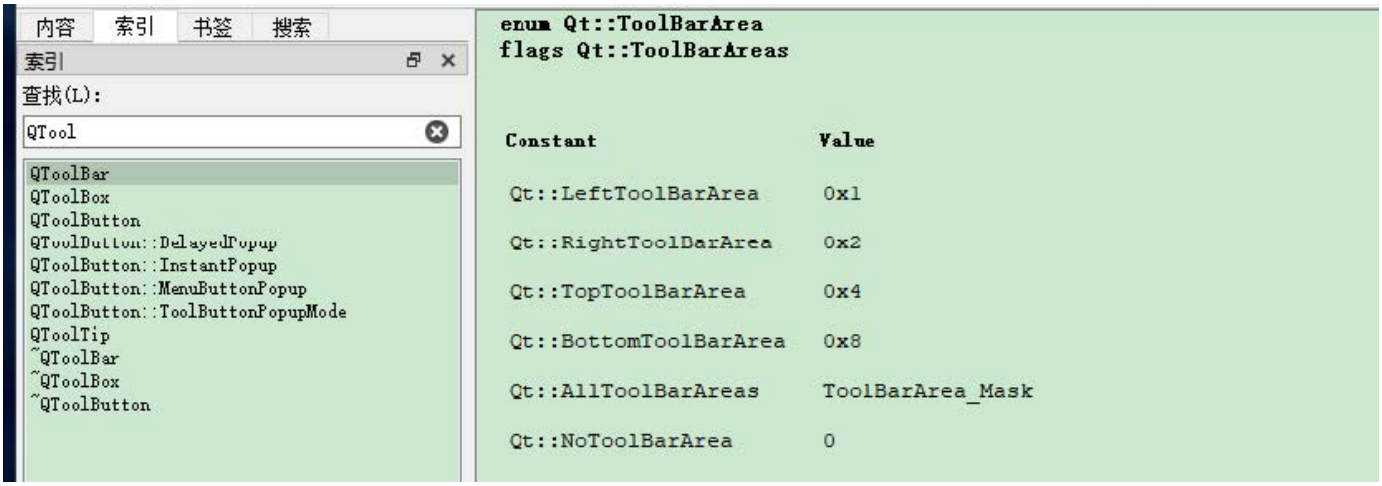
QToolBar *toolbar

通过查询文档找到默认停靠区域的可填写值



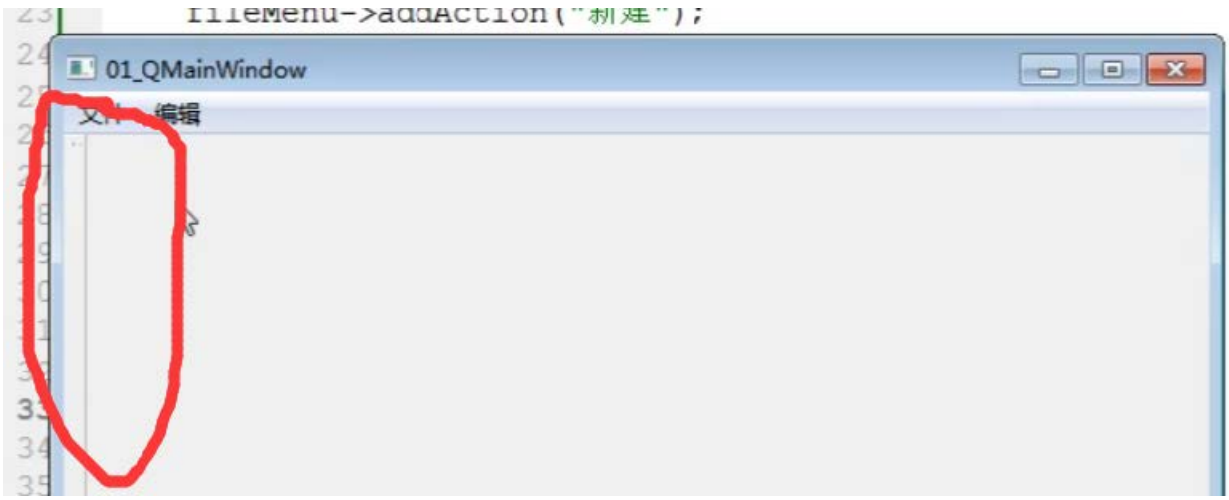


查找之后即可看到Qt::ToolBarArea的枚举值



使用枚举值设置工具栏区域

```
//工具栏  可以有多个
QToolBar * toolBar = new QToolBar(this);
addToolBar(Qt::LeftToolBarArea,toolBar);
```



可以看到左边出现了工具栏（暂未添加内容）

查询QToolBar的帮助文档

```
void    setAllowedAreas(Qt::ToolBarAreas areas)
void    setFloatable(bool floatable)
void    setMovable(bool movable)
void    setOrientation(Qt::Orientation orientation)
```

```
//后期设置 只允许 左右停靠
toolBar->setAllowedAreas( Qt::LeftToolBarArea | Qt::RightToolBarArea );
```

使用或操作符 | 来设置允许左右停靠

```
//设置浮动
toolBar->setFloatable(false);
```

设置移动（总开关）

```
//设置移动 (总开关)
toolBar->setMovable(false);
```

**设置为false之后无法移动（尽管此前已经设置过停靠）

工具栏可以设置内容：

```
//创建菜单项
QAction * newAction = fileMenu->addAction("新建");
```

toolBar->addAction(newAction);

```
//添加分割线
fileMenu->addSeparator();
QAction * openAction = fileMenu->addAction("打开");
```

toolBar->addAction(openAction);

```
//工具栏中可以设置内容
toolBar->addAction(newAction);
toolBar->addAction(openAction);
```

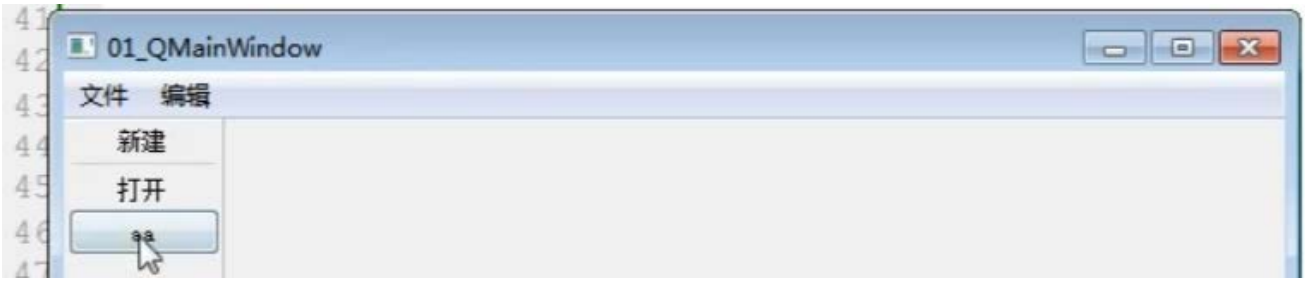
如此一来，便实现了两个菜单项公用在了工具栏。

```
//工具栏
    QToolBar * toolBar = new QToolBar(this);
    addToolBar(Qt::LeftToolBarArea,toolBar);
    toolBar->addAction("Create");//创建工具栏选项
    toolBar->setAllowedAreas(Qt::LeftToolBarArea | Qt::RightToolBarArea);//后期设置只允许左右停靠
    toolBar->setFloatable(false);//设置停靠为FALSE
//    toolBar->setMovable(false);//总开关（移动）
    QAction * newAction = fileMenu->addAction("new");//新建选项new
    QAction * openAction = fileMenu->addAction("open");//新建选项open
    toolBar->addAction(newAction);//放进工具栏
    fileMenu->addSeparator();//插入分割线
    toolBar->addAction(openAction);//放进工具栏
```

在创建好的工具栏中添加控件

1. #include
2. 添加按钮

```
//工具栏中添加控件
QPushButton * btn = new QPushButton("aa" , this);
toolBar->addWidget(btn);
```

3.
4. 运行查看
5. 

可以看到，工具栏既可以添加菜单，又可以添加按钮

状态栏

最多一个

```
QStatusBar * stBar =statusBar();
setStatusBar(stBar);//设置到窗口中（只能有一个）
```

关键字：QStatusBar , setStatusBar (创建和设置)

预览：

现在状态栏什么也没写，所以是空的

接下来为状态栏添加标签控件

1. 包含头文件-->#include
2. QLabel * label = new QLabel ("提示的信息 " ， this);//添加标签
3. stBar->addWidget(label); //将标签放进状态栏

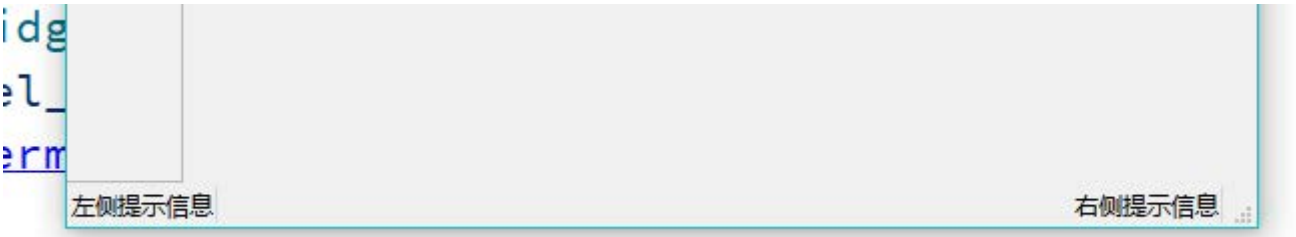
```
//状态栏
    QStatusBar * stBar =statusBar();
    setStatusBar(stBar);//设置到窗口中
    QLabel *label_1 = new QLabel("左侧提示信息",this);//添加标签
    stBar->addWidget(label_1);//将标签放进状态栏
```

预览：

右侧也可以添加状态栏信息

```
QLabel *label_2 = new QLabel("右侧提示信息",this);//添加标签
    stBar->addPermanentWidget(label_2);//将标签放进状态栏
```

预览效果：

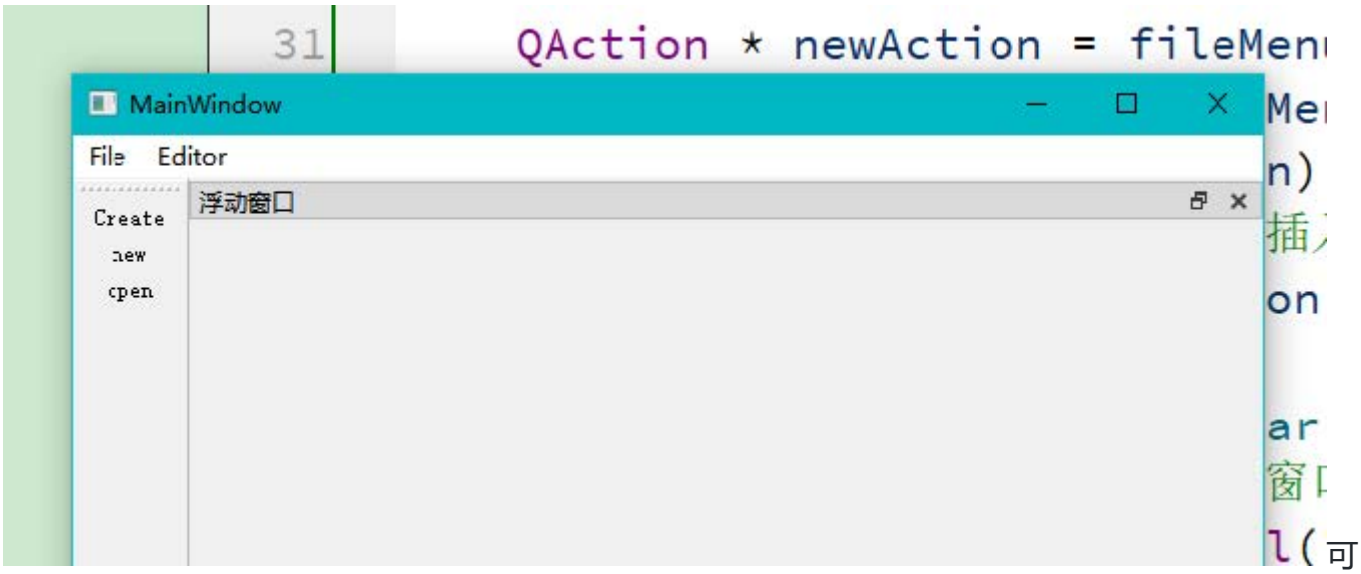


铆接部件（浮动窗口）

关键字：QDockWidget

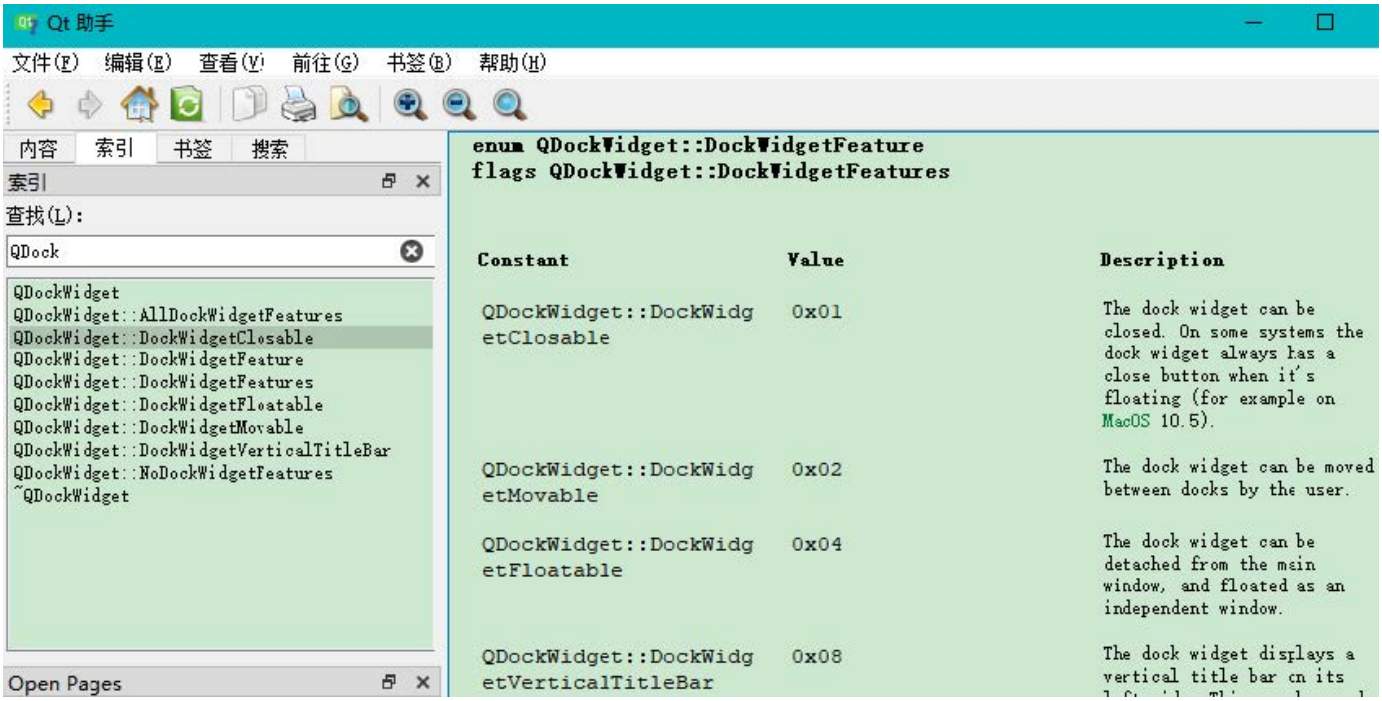
```
//铆接部件（浮动窗口）
    QDockWidget *dockWidget = new  QDockWidget("浮动窗口",this);
    addDockWidget(Qt::BottomDockWidgetArea,dockWidget);//设置停靠在下面
```

1. 包含头文件#include
2. QDockWidget * dockWidget =new QDockWidget("浮动" ， this);
3. addDockWidget(Qt : : BottomDockWidgetArea，dockWidget); //设置停靠在下面
4. 预览：



以拖动该窗口

查询帮助文档，获取枚举值



设置后期停靠

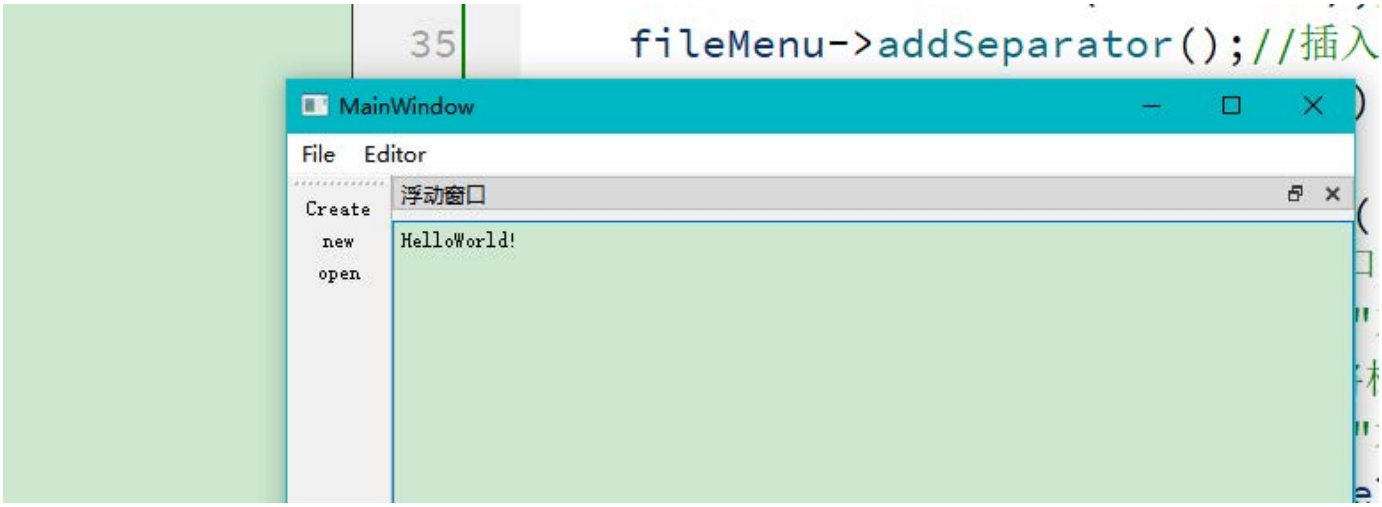
```
//设置后期停靠区域，只允许上下
dockWidget->setAllowedAreas( Qt::TopDockWidgetArea | Qt::BottomDockWidgetArea );
```

设置中心部件

只能有一个

```
//中心部件
    QTextEdit *edit = new QTextEdit(this);
    setCentralWidget(edit);//放置中心部件
```

1. 包含头文件#include
2. QTextEdit * edit =new QTextEdit(this);
3. setCentralWidget(edit); //放置中心部件
4. 预览：



可以看到，添加中心部件之后，浮动窗口变成我们设置好的下方

总结QMainWindow

菜单栏最多一个(set)

```
QMenuBar *bar = menuBar();//创建
setMenuBar(bar);    //设置菜单栏
QMenu *fileMenu = bar->addMenu("文件");//创建菜单--->文件选项
QAction #newAction = fileMenu->addAction("新建");//创建菜单项
fileMenu->addSeparator();   //添加分割线
```

工具栏可以有多个(add)

```
QToolBar * toolbar =new QToolBar(this);//创建工具栏
addTolBar("默认停靠区域",toolBar);    //区域枚举值请查阅帮助文档
```

状态栏只能有一个(set)

```
QStatusBar * stBar = statusBar();   //创建状态栏对象
setStatusBar(stBar);                //设置到窗口中
QLabel * label = new QLabel( "提示信息" , this);  //创建标签
stBar->addWidget(label);            //设置到窗口中（左侧）
```

铆接部件（浮动窗口）（add）

```
QDockWidget
addDockWidget( 默认停靠区域 ， 浮动窗口指针)
设置后期停靠区域
```

中心部件（set）

```
#include<QTextEdit>
QTextEdit * edit =new QTextEdit(this);    //创建文本框中心部件
setCentralWidget( edit );                      //放置中心部件
```

附录（完整代码）

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include<QMenu>
#include<QToolBar>
#include<QDebug>
#include<QStatusBar>
#include<QLabel>
#include<QDockWidget>
#include<QTextEdit>
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    resize(600,400);
    QMenuBar *bar = menuBar();//创建菜单指针
    setMenuBar(bar);//放进窗口
    //菜单栏
    QMenu *fileMenu = bar->addMenu("File");
    QMenu *editMenu = bar->addMenu("Editor");
    fileMenu->addAction("Create");
    fileMenu->addSeparator();//添加分割线
    fileMenu->addAction("Open");
    //工具栏
    QToolBar * toolBar = new QToolBar(this);
    addToolBar(Qt::LeftToolBarArea,toolBar);
    toolBar->addAction("Create");//创建工具栏选项
    toolBar->setAllowedAreas(Qt::LeftToolBarArea | Qt::RightToolBarArea);//后期设置只允许左右停靠
    toolBar->setFloatable(false);//设置停靠为FALSE
//    toolBar->setMovable(false);//总开关（移动）
    QAction * newAction = fileMenu->addAction("new");//新建选项new
    QAction * openAction = fileMenu->addAction("open");//新建选项open
    toolBar->addAction(newAction);//放进工具栏
    fileMenu->addSeparator();//插入分割线
    toolBar->addAction(openAction);//放进工具栏
```

```
    //状态栏
    QStatusBar * stBar =statusBar();
    setStatusBar(stBar);//设置到窗口中（只能有一个）
    QLabel *label_1 = new QLabel("左侧提示信息",this);//添加标签
    stBar->addWidget(label_1);//将标签放进状态栏
    QLabel *label_2 = new QLabel("右侧提示信息",this);//添加标签
    stBar->addPermanentWidget(label_2);//将标签放进状态栏
    //铆接部件（浮动窗口）
    QDockWidget *dockWidget = new  QDockWidget("浮动窗口",this);
    addDockWidget(Qt::BottomDockWidgetArea,dockWidget);//设置停靠在下面
    dockWidget->setAllowedAreas(Qt::TopDockWidgetArea | Qt::BottomDockWidgetArea);//设置后期停靠（上，下）
    //中心部件
    QTextEdit *edit = new QTextEdit(this);
    setCentralWidget(edit);//放置中心部件

}

MainWindow::~MainWindow()
{
    delete ui;
}
```