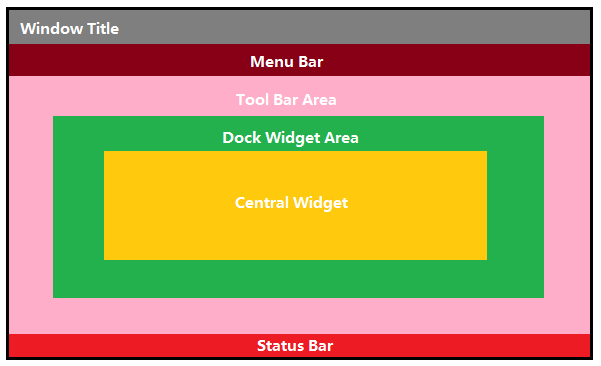
QMainWindow

之前有说到，QMainWindow类主要承担的是菜单栏，工具栏，状态栏，铆接部件，核心部件等窗口部件，如下图所示：



1. 菜单栏

一个主窗口最多只有一个菜单栏，位于主窗口的顶部，标题下面。

* 创建菜单栏，通过QMainWindow类的menubar（）函数获取主窗口菜单栏指针

QMenuBar \* menuBar() const

* 创建菜单，调用QMenu的成员函数addMenu来添加菜单
* QAction是抽象类，当我们把QAction对象添加到菜单中，就显示成菜单项，但是，它并不是一个专门的菜单项类。

QAction\* addMenu(QMenu \* menu)

QMenu\* addMenu(const QString & title)

QMenu\* addMenu(const QIcon & icon, const QString & title)

* 创建菜单项，调用QMenu的成员函数addAction来添加菜单项

QAction\* activeAction() const

QAction\* addAction(const QString & text)

QAction\* addAction(const QIcon & icon, const QString & text)

QAction\* addAction(const QString & text, const QObject \* receiver,

const char \* member, const QKeySequence & shortcut = 0)

QAction\* addAction(const QIcon & icon, const QString & text,

const QObject \* receiver, const char \* member,

const QKeySequence & shortcut = 0)

1. 工具栏

主窗口可以有多个工具栏，可以调整工具栏的位置。

1. 创建工具栏

调用QMainWindow类的addToolBar（）函数获取主窗口的工具条对象，每增加一个工具条都需要调用一次该函数。

1. 插入属于工具栏的选项

通过QToolBar类的addAction函数添加。

1. 移动工具栏

工具栏的停靠区域由QToolBar的allowAreas决定

停靠区域如下：

* + - Qt::LeftToolBarArea 停靠在左侧
    - Qt::RightToolBarArea 停靠在右侧
    - Qt::TopToolBarArea 停靠在顶部
    - Qt::BottomToolBarArea 停靠在底部
    - Qt::AllToolBarAreas 以上四个位置都可停靠

1. 状态栏

主窗口底部的状态栏，最多只有一个。

QStatusBar类

//添加小部件

void addWidget(QWidget \* widget, int stretch = 0)

//插入小部件

int insertWidget(int index, QWidget \* widget, int stretch = 0)

//删除小部件

void removeWidget(QWidget \* widget)

1. 铆接部件

即浮动窗口，能够再窗口中浮动并提取出来，可以有多个。

QDockWidget \* dock = new QDockWidget("标题",this);

addDockWidget(Qt::LeftDockWidgetArea,dock);

dock->setAllowedAreas(Qt::LeftDockWidgetArea | Qt::RightDockWidgetArea | Qt::TopDockWidgetArea); 设置区域范围

1. 核心部件

核心部件是一个应用程序的主体，即主要的操作都在核心部件上面。

如，记事本用的是QTextEdit做核心部件：

QTextEdit \* edit = new QTextEdit(this);

setCentralWidget(edit);

1. 资源文件

资源文件用于将程序运行时所需要的资源以二进制的形式存储于可执行文件内部。如果你将资源以资源文件形式存储，它是会编译到可执行文件内部。

资源文件就是程序中需要使用的图片等资源，如何使用资源文件呢？

再新建文件时，添加Qt Resource File，生成qrc文件，然后为每种资源设置前缀，前缀即资源的分类，之后可以将需要的资源导入。

导入资源文件到项目中的格式

: + 前缀名/文件名

如下所示：

Ui->actionNew->setIcon(QIcon(“:/Image/Luffy.png”));

1. 代码

#include "mainwindow.h"

#include<QMenuBar>

#include<QToolBar>

#include<QStatusBar>

#include<QLabel>

#include<QDockWidget>

#include<QTextEdit>

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

: QMainWindow(parent)

{

resize(600, 400);

//创建菜单栏，只能有一个

QMenuBar \*bar = menuBar();

//将菜单栏放入窗口中

this->setMenuBar(bar);

//创建文件菜单

QMenu \*fileMenu = bar->addMenu("文件");

QMenu \*editMenu = bar->addMenu("编辑");

//添加菜单里的选项

QAction \*newAction = fileMenu->addAction("新建");

//添加分割线

fileMenu->addSeparator();

QAction \*openAction = fileMenu->addAction("打开");

//工具栏， 可以有多个

QToolBar \*toolBar = new QToolBar(this);

addToolBar(Qt::LeftToolBarArea,toolBar);

//只允许左右停靠工具栏

toolBar->setAllowedAreas(Qt::LeftToolBarArea | Qt::RightToolBarArea);

//设置工具栏浮动

toolBar->setFloatable(false);

//工具栏添加菜单项

toolBar->addAction(newAction);

toolBar->addAction(openAction);

//状态栏,只能有一个

QStatusBar \* stBar = statusBar();

setStatusBar(stBar);

//状态栏中显示信息

QLabel \*label = new QLabel("提示信息",this);

//添加提示信息到左侧

stBar->addWidget(label);

//添加提示信息到右侧

QLabel \*label2 = new QLabel("右侧提示信息",this);

stBar->addPermanentWidget(label2);

//铆接部件，浮动窗口,可以有多个

QDockWidget \*dock = new QDockWidget;

//添加到窗口中

addDockWidget(Qt::BottomDockWidgetArea,dock);

//核心部件,只能一个

//文本编辑框

QTextEdit \*edit = new QTextEdit;

setCentralWidget(edit);

}

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

{

}