



## 第3章 Qt 5布局管理

### 3.1 分割窗口QSplitter类

### 3.2 停靠窗口QDockWidget类

### 3.3 堆栈窗体QStackedWidget类

### 3.4 基本布局 (QLayout)





## 3.1 分割窗口QSplitter类

介绍一个十分简单的分割窗口功能，整个对话框由三个窗口组成，各个窗口之间的大小可随意拖曳改变，效果如图3.1所示。





## 3.1 分割窗口QSplitter类

(1) 新建Qt Gui应用（详见本书第1章1.3.1节），项目名称为“Splitter”，基类选择“QMainWindow”，取消“创建界面”复选框的选中状态。

(2) 在上述工程的“main.cpp” [文件中添加代码。](#)

(3) 在“main.cpp”文件的开始部分加入以下头文件：

```
#include<Qsplitter>
#include<QTextEdit>
#include <QTextCodec>
```

(4) 运行程序，显示效果如图3.1所示。





## 3.2 停靠窗口QDockWidget类

停靠窗口QDockWidget类也是应用程序中经常用到的，设置停靠窗口的一般流程如下。

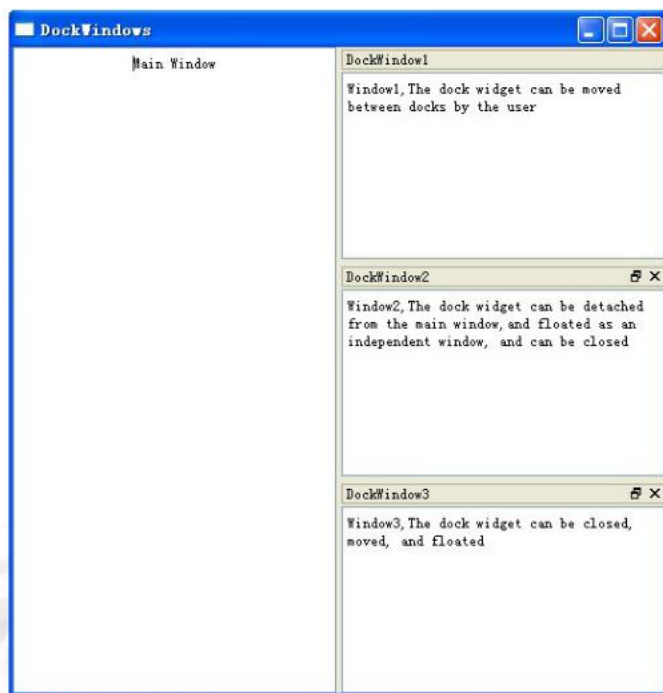
- (1) 创建一个QDockWidget对象的停靠窗体。
- (2) 设置此停靠窗体的属性，通常调用setFeatures()及setAllowedAreas()两种方法。
- (3) 新建一个要插入停靠窗体的控件，常用的有QListWidget和QTextEdit。
- (4) 将控件插入停靠窗体，调用QDockWidget的setWidget()方法。
- (5) 使用addDockWidget()方法在MainWindow中加入此停靠窗体。





## 3.2 停靠窗口QDockWidget类

下面通过一个简单的例子来学习停靠窗口QDockWidget类的使用，窗口1只可在主窗口的左边和右边停靠；窗口2只可在浮动和右部停靠两种状态间切换，并且不可移动；窗口3可实现停靠窗口的各种状态。效果如图3.2所示。





## 3.2 停靠窗口QDockWidget类

(1) 新建Qt Gui应用（详见本书第1章1.3.1节），项目名称为“DockWindows”，基类选择“QMainWindow”，类名命名为“DockWindows”，取消“创建界面”复选框的选中状态，如图3.3所示。







## 3.2 停靠窗口QDockWidget类

(2) DockWindows类说明中只有一个构造函数的说明。代码如下:

```
class DockWindows : public QMainWindow
{
    Q_OBJECT
public:
    DockWindows(QWidget *parent = 0);
    ~DockWindows();
};
```

(3) 打开“dockwindows.cpp”文件, DockWindows类构造函数实现窗口的初始化及功能实现, [具体代码所示。](#)

(4) 在“dockwindows.cpp”文件的开始部分加入以下头文件:

```
#include<QTextEdit>
#include<QDockWidget>
```

(5) 运行程序, 显示效果如图3.2所示。



## 3.3 堆栈窗体QStackedWidget类

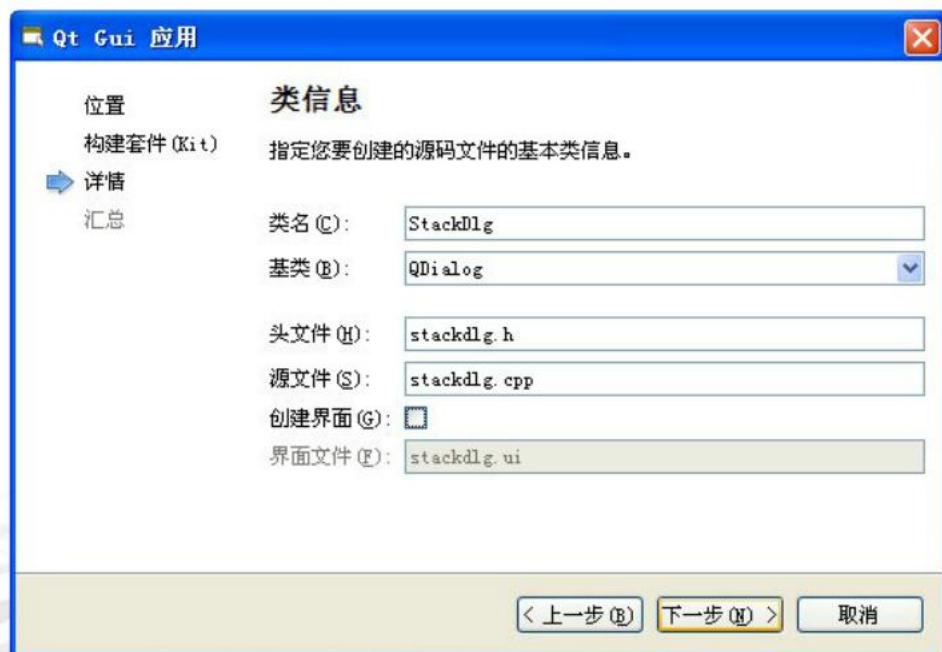
通过一个简单的例子来学习堆栈窗体QStackedWidget类的使用，当选择左侧列表框中不同的选项时，右侧显示所选的不同的窗体。在此使用列表框QListWidget，效果如图3.4所示





## 3.3 堆栈窗体QStackedWidget类

(1) 新建Qt Gui应用，项目名称为“StackedWidget”，基类选择“QDialog”，类名命名为“StackDlg”，取消“创建界面”复选框的选中状态，如图3.5所示。





## 3.3 堆栈窗体QStackedWidget类

(2) 打开“stackdlg.h”文件，添加如下加黑代码：

```
class StackDlg : public QDialog
{
    Q_OBJECT
public:
    StackDlg(QWidget *parent = 0);
    ~StackDlg();
private:
    QListWidget *list;
    QStackedWidget *stack;
    QLabel *label1;
    QLabel *label2;
    QLabel *label3;
};
```

在文件开始部分添加以下头文件：

```
#include <QListWidget>
#include <QStackedWidget>
#include <QLabel>
```



## 3.3 堆栈窗体QStackedWidget类

(3) 打开“stackdlg.cpp”文件，在停靠窗体StackDlg类的构造函数中添加代码。

(4) 在stackdlg.cpp文件的开始部分加入以下头文件：

```
#include <QHBoxLayout>
```

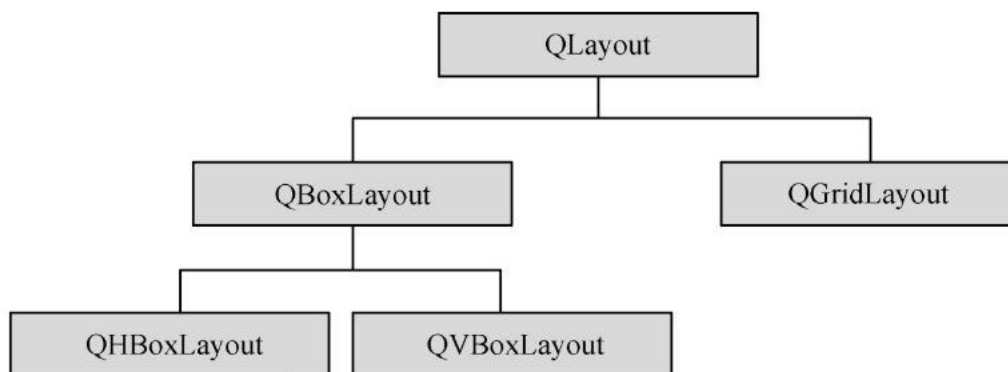
(5) 运行程序，显示效果如图3.4所示。





## 3.4 基本布局 (QLayout)

Qt提供了QHBoxLayout类、QVBoxLayout类及QGridLayout类等的基本布局管理，分别是水平排列布局、垂直排列布局和网格排列布局。它们之间的继承关系如图3.6所示。





## 3.4 基本布局 (QLayout)

`addWidget()` 方法用于向布局中加入需要布局的控件，`addWidget()` 的函数原型如下：

```
void addWidget
(
    QWidget *widget,           //需要插入的控件对象
    int fromRow,               //插入的行
    int fromColumn,            //插入的列
    int rowSpan,               //表示占用的行数
    int columnSpan,            //表示占用的列数
    Qt::Alignment alignment=0  //描述各个控件的对齐方式
)
```

`addLayout ()` 方法用于向布局中加入需要布局的子布局，`addLayout ()` 的函数原型如下：

```
void addLayout
(
    QLayout *layout,           //表示需要插入的子布局对象
    int row,                   //插入的起始行
    int column,                //插入的起始列
    int rowSpan,               //表示占用的行数
    int columnSpan,            //表示占用的列数
    Qt::Alignment alignment=0  //指定对齐方式
)
```



## 3.4 基本布局 (QLayout)

本节将通过实现一个“用户基本资料修改”的功能表来介绍如何使用基本布局管理，如QHBoxLayout类、QVBoxLayout类及QGridLayout类，效果如图3.7所示。

The screenshot shows a Qt application window titled "UserInfo". The window contains a form with the following elements:

- 用户名:** A single-line text input field.
- 姓名:** A single-line text input field.
- 性别:** A dropdown menu with "女" (Female) selected.
- 头像:** A circular icon of a lifebuoy and a "更新" (Update) button.
- 个人说明:** A large multi-line text area.
- 部门:** A large multi-line text area.
- 年龄:** A single-line text input field.
- 备注:** A single-line text input field.
- Buttons:** "确定" (OK) and "取消" (Cancel) buttons at the bottom right.



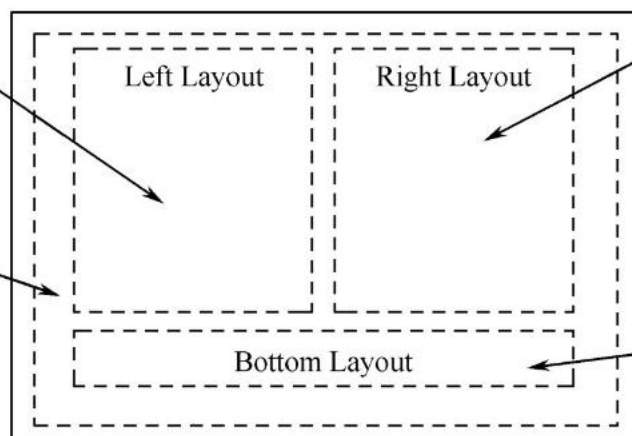


## 3.4 基本布局 (QLayout)

本实例共用到四个布局管理器，分别是LeftLayout、RightLayout、BottomLayout和MainLayout，其布局框架如图3.8所示。

左部的布局管理器，  
包含若干Label和  
LineEdit，可用一个  
表格布局管理器  
QGridLayout实现

Main Layout主布局  
管理器，可用一个  
QGridLayout实现



右部的布局管理器，可  
用一个QVBoxLayout实现

底部的布局管理器，包  
含两个按钮，可用一个  
QHBoxLayout实现



## 3.4 基本布局 (QLayout)

(1) 新建Qt Gui应用，项目名称为“UserInfo”，基类选择“QDialog”，取消“创建界面”复选框的选中状态。

(2) 打开“dialog.h”头文件，在头文件中声明对话框中的各个控件。[添加代码所示。](#)

添加如下的头文件：

```
#include <QLabel>
#include <QLineEdit>
#include <QComboBox>
#include <QTextEdit>
#include <QGridLayout>
```



## 3.4 基本布局 (QLayout)

(3) 打开“dialog.cpp”文件，在类Dialog的构造函数中添加代码。

(4) 在“dialog.cpp”文件的开始部分加入以下头文件：

```
#include<QLabel>
#include<QLineEdit>
#include<QComboBox>
#include<QPushButton>
#include<QFrame>
#include<QGridLayout>
#include<QPixmap>
#include<QHBoxLayout>
```

(5) 选择“构建”→“构建项目"UserInfo"”命令，为了能够在界面上显示头像图片，请将事先准备好的图片312.png复制到D:\Qt\CH3\CH304\build-UserInfo-Desktop\_Qt\_5\_0\_2\_MinGW\_32bit-Debug目录下。