38 QComboBox和QPlainTextEdit

38.1 QComboBox

QComboBox是表示组合框的类，它提供一个下拉列表供用户选择，同时也带有一个单行编辑框，以接受用户输入的文本。QComboBox下拉列表中的每个列表项，还可以与一个QVariant类型的用户数据相关联。

38.1.1 添加删除列表项

组合框中的列表项可以在Qt设计师中添加或删除，也可以通过QComboBox类的特定方法在代码中添加或删除。每个列表项既可以带有图标，也可以关联用户数据：

```
void QComboBox::addItem(const QString& text, const QVariant& userData =
QVariant());
void QComboBox::addItem(const QIcon& icon, const QString& text, const
QVariant& userData = QVariant());
void QComboBox::addItems(const QStringList& texts);
void QComboBox::clear();
```

38.1.2 获取列表项

QComboBox不提供获取整个列表的功能，但可以获取当前选择项，或通过索引获取指定项：

```
int      QComboBox::currentIndex();                                  // 返回当前项
的索引
QString  QComboBox::currentText();                                   // 返回当前项
的文本
QVariant QComboBox::currentData(int role = Qt::UserRole);            // 返回当前项
的用户数据
QString  QComboBox::itemText(int index);                             // 返回指定项
的文本
QVariant QComboBox::itemData(int index, int role = Qt::UserRole); // 返回指定项
的用户数据
int      QComboBox::count();                                         // 返回项的总
数
```

38.1.3 选择改变信号

当用户在组合框的下拉列表中改变选择项时，QComboBox会发射如下两个信号：

```
void QComboBox::currentIndexChanged(int index);           // 参数为当前选择项的索
引
void QComboBox::currentIndexChanged(const QString& text); // 参数为当前选择项的文
本
```

通过定义与之相连的槽函数，可以针对选择项的变化，执行相应的业务处理。

38.1.4 是否可编辑

QComboBox提供如下方法获取和设置组合框的可编辑属性：

```
bool QComboBox::isEditable() const
void QComboBox::setEditable(bool editable);
```

38.2 QPlainTextEdit

QPlaintTextEdit是表示纯文本编辑框的类，可以显示和编辑多行纯文本内容，同时提供默认的右键菜单功能。

38.2.1 添加删除文本

QPlainTextEdit类提供如下方法用于在代码中添加和删除编辑框中的文本：

```
void QPlainTextEdit::appendPlainText(const QString& text);
void QPlainTextEdit::clear();
```

38.2.2 获取文本

可以一次性获取编辑框中的文本整体：

```
QString QPlainTextEdit::toPlainText() const;
```

也可以逐行获取编辑框中的多行文本：

```
QTextDocument* QPlainTextEdit::document() const;
int            QTextDocument::blockCount() const;
QTextBlock     QTextDocument::findBlockByNumber(int blockNumber) const;
QString        QTextBlock::text() const;
```

38.2.3 是否只读

QPlainTextEdit提供如下方法获取和设置编辑框的只读属性：

```
bool QPlainTextEdit::isReadOnly() const;
void QPlainTextEdit::setReadOnly(bool readonly);
```

38.2.4 默认右键菜单与槽函数

QPlainTextEdit提供默认的右键菜单，并为其中的每个菜单项定义了功能完备的槽函数：

```
void QPlainTextEdit::undo();
void QPlainTextEdit::redo();
void QPlainTextEdit::cut();
void QPlainTextEdit::copy();
void QPlainTextEdit::paste();
void QPlainTextEdit::clear();
void QPlainTextEdit::selectAll();
```

38.3 案例

38.3.1 创建项目

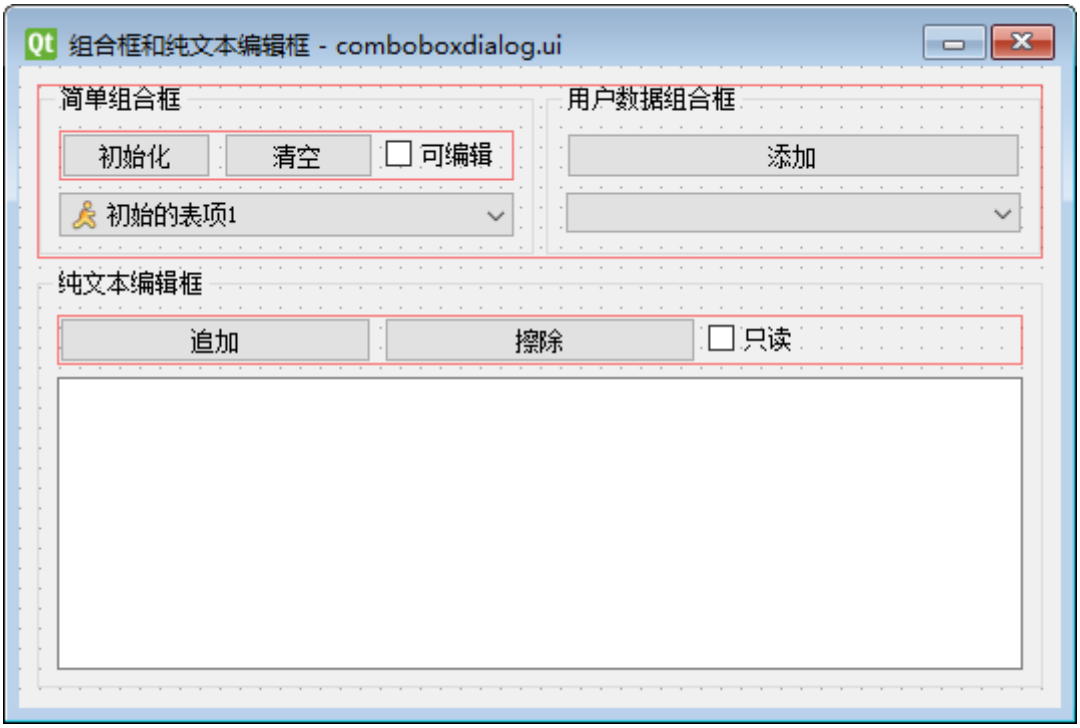
通过QtCreator，在C:\Users\Minwei\Projects\Qt路径下，创建名为ComboBox的项目。

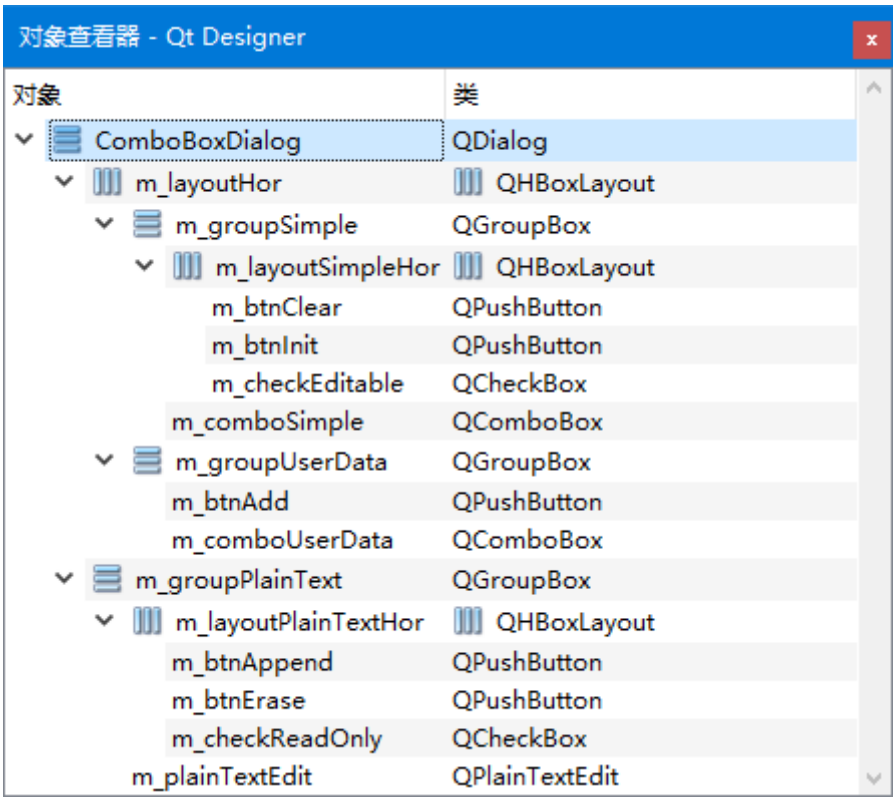
38.3.2 添加资源

C:\Users\Minwei\Projects\Qt\ComboBox\ComboBox.qrc：

```
<RCC>
    <qresource prefix="/">
        <file>images/aim.ico</file>
        <file>images/unit.ico</file>
    </qresource>
</RCC>

```

38.3.3 设计界面





C:\Users\Minwei\Projects\Qt\ComboBox\comboboxdialog.ui：

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>ComboBoxDialog</class>
 <widget class="QDialog" name="ComboBoxDialog">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>520</width>
    <height>320</height>
   </rect>
  </property>
  <property name="windowTitle">
   <string>组合框和纯文本编辑框</string>
  </property>
  <layout class="QVBoxLayout" name="m_layoutVer">
   <item>
    <layout class="QHBoxLayout" name="m_layoutHor">
     <item>
      <widget class="QGroupBox" name="m_groupSimple">
       <property name="title">
        <string>简单组合框</string>
       </property>
       <layout class="QVBoxLayout" name="m_layoutSimpleVer">
        <item>
         <layout class="QHBoxLayout" name="m_layoutSimpleHor">
          <item>
           <widget class="QPushButton" name="m_btnInit">
            <property name="text">
             <string>初始化</string>
            </property>
```

```
           </widget>
          </item>
          <item>
           <widget class="QPushButton" name="m_btnClear">
            <property name="text">
             <string>清空</string>
            </property>
           </widget>
          </item>
          <item>
           <widget class="QCheckBox" name="m_checkEditable">
            <property name="text">
             <string>可编辑</string>
            </property>
           </widget>
          </item>
         </layout>
        </item>
        <item>
         <widget class="QComboBox" name="m_comboSimple">
          <item>
           <property name="text">
            <string>初始的表项1</string>
           </property>
           <property name="icon">
            <iconset resource="ComboBox.qrc">

<normaloff>:/images/aim.ico</normaloff>:/images/aim.ico</iconset>
           </property>
          </item>
          <item>
           <property name="text">
            <string>初始的表项2</string>
           </property>
           <property name="icon">
            <iconset resource="ComboBox.qrc">

<normaloff>:/images/aim.ico</normaloff>:/images/aim.ico</iconset>
           </property>
          </item>
          <item>
           <property name="text">
            <string>初始的表项3</string>
           </property>
           <property name="icon">
            <iconset resource="ComboBox.qrc">

<normaloff>:/images/aim.ico</normaloff>:/images/aim.ico</iconset>
           </property>
          </item>
         </widget>
        </item>
       </layout>
      </widget>
     </item>
     <item>
```

```
      <widget class="QGroupBox" name="m_groupUserData">
       <property name="title">
        <string>用户数据组合框</string>
       </property>
       <layout class="QVBoxLayout" name="m_layoutUserDataVer">
        <item>
         <widget class="QPushButton" name="m_btnAdd">
          <property name="text">
           <string>添加</string>
          </property>
         </widget>
        </item>
        <item>
         <widget class="QComboBox" name="m_comboUserData"/>
        </item>
       </layout>
      </widget>
     </item>
    </layout>
   </item>
   <item>
    <widget class="QGroupBox" name="m_groupPlainText">
     <property name="title">
      <string>纯文本编辑框</string>
     </property>
     <layout class="QVBoxLayout" name="m_layoutPlainTextVer">
      <item>
       <layout class="QHBoxLayout" name="m_layoutPlainTextHor">
        <item>
         <widget class="QPushButton" name="m_btnAppend">
          <property name="text">
           <string>追加</string>
          </property>
         </widget>
        </item>
        <item>
         <widget class="QPushButton" name="m_btnErase">
          <property name="text">
           <string>擦除</string>
          </property>
         </widget>
        </item>
        <item>
         <widget class="QCheckBox" name="m_checkReadOnly">
          <property name="text">
           <string>只读</string>
          </property>
         </widget>
        </item>
       </layout>
      </item>
      <item>
       <widget class="QPlainTextEdit" name="m_plainTextEdit"/>
      </item>
     </layout>
    </widget>
```

```
    </item>
   </layout>
  </widget>
  <resources>
   <include location="ComboBox.qrc"/>
  </resources>
  <connections/>
 </ui>
```

38.3.4 实现功能

C:\Users\Minwei\Projects\Qt\ComboBox\comboboxdialog.h：

```
#ifndef COMBOBOXDIALOG_H
#define COMBOBOXDIALOG_H

#include <QDialog>

QT_BEGIN_NAMESPACE
namespace Ui { class ComboBoxDialog; }
QT_END_NAMESPACE

class ComboBoxDialog : public QDialog
{
    Q_OBJECT

public:
    ComboBoxDialog(QWidget *parent = nullptr);
    ~ComboBoxDialog();

private slots:
    void on_m_btnInit_clicked();
    void on_m_btnClear_clicked();
    void on_m_checkEditable_clicked(bool checked);
    void on_m_comboSimple_currentIndexChanged(const QString &arg1);

    void on_m_btnAdd_clicked();
    void on_m_comboUserData_currentIndexChanged(const QString &arg1);

    void on_m_btnAppend_clicked();
    void on_m_btnErase_clicked();
    void on_m_checkReadOnly_clicked(bool checked);

private:
    Ui::ComboBoxDialog *ui;
};

#endif // COMBOBOXDIALOG_H
```

C:\Users\Minwei\Projects\Qt\ComboBox\comboboxdialog.cpp：

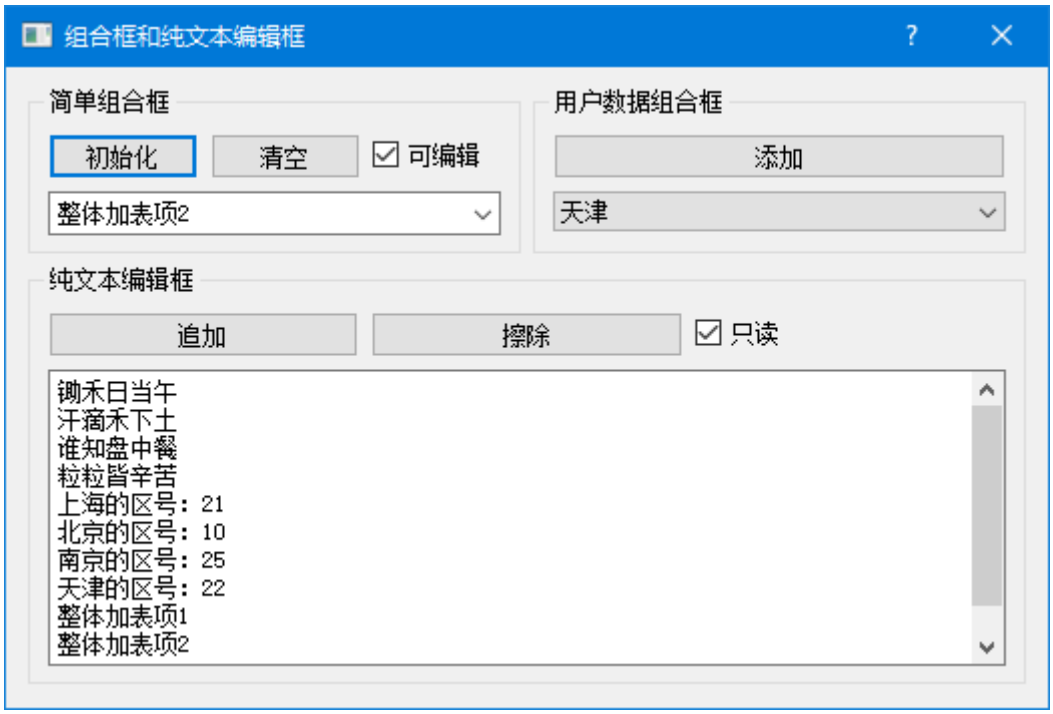
```
#include <QTextBlock>

#include "comboboxdialog.h"
```

```cpp
#include "ui_comboboxdialog.h"

ComboBoxDialog::ComboBoxDialog(QWidget *parent)
    : QDialog(parent)
    , ui(new Ui::ComboBoxDialog)
{
    ui->setupUi(this);
}

ComboBoxDialog::~ComboBoxDialog()
{
    delete ui;
}

void ComboBoxDialog::on_m_btnInit_clicked()
{
    for (int i = 1; i <= 3; ++i)
        ui->m_comboSimple->addItem(QString::asprintf("无图标表项%d", i));

    for (int i = 1; i <= 3; ++i)
        ui->m_comboSimple->addItem(QIcon(":/images/unit.ico"),
            QString::asprintf("有图标表项%d", i));

    QStringList items;
    items << "整体加表项1" << "整体加表项2" << "整体加表项3";
    ui->m_comboSimple->addItems(items);
}

void ComboBoxDialog::on_m_btnClear_clicked()
{
    ui->m_comboSimple->clear();
}

void ComboBoxDialog::on_m_checkEditable_clicked(bool checked)
{
    ui->m_comboSimple->setEditable(checked);
}

void ComboBoxDialog::on_m_comboSimple_currentIndexChanged(const QString
&arg1)
{
    if (!arg1.isEmpty())
        ui->m_plainTextEdit->appendPlainText(arg1);
}

void ComboBoxDialog::on_m_btnAdd_clicked()
{
    ui->m_comboUserData->clear();

    QMap<QString, int> cityZone;
    cityZone["北京"] = 10;
    cityZone["广州"] = 20;
    cityZone["上海"] = 21;
    cityZone["天津"] = 22;
    cityZone["重庆"] = 23;
    cityZone["沈阳"] = 24;
```

```
    cityZone["南京"] = 25;
    cityZone["武汉"] = 27;
    cityZone["成都"] = 28;

    for (QString const& city : cityZone.keys())
        ui->m_comboUserData->addItem(city, cityZone[city]);
}

void ComboBoxDialog::on_m_comboUserData_currentIndexChanged(const QString
&arg1)
{
    if (!arg1.isEmpty())
    {
        QVariant zone = ui->m_comboUserData->currentData();
        ui->m_plainTextEdit->appendPlainText(arg1 + "的区号：" +
zone.toString());
    }
}

void ComboBoxDialog::on_m_btnAppend_clicked()
{
    QTextDocument* doc = ui->m_plainTextEdit->document();
    int blockCount = doc->blockCount();

    for(int blockNumber = 0; blockNumber < blockCount; ++blockNumber)
        ui->m_comboSimple->addItem(QIcon(":/images/aim.ico"),
            doc->findBlockByNumber(blockNumber).text());
}

void ComboBoxDialog::on_m_btnErase_clicked()
{
    ui->m_plainTextEdit->clear();
}

void ComboBoxDialog::on_m_checkReadOnly_clicked(bool checked)
{
    ui->m_plainTextEdit->setReadOnly(checked);
}
```

38.3.5 测试验证

运行效果如图所示：


组合框和纯文本编辑框
简单组合框
初始化
清空
可编辑
整体加表项2
用户数据组合框
添加
天津
纯文本编辑框
追加
擦除
只读
锄禾日当午
汗滴禾下土
谁知盘中餐
粒粒皆辛苦
上海的区号：21
北京的区号：10
南京的区号：25
天津的区号：22
整体加表项1
整体加表项2