

## 42 QListWidget等组件的使用

这个案例项目综合使用了列表框组件 (QListWidget)、工具按钮组件 (QToolButton)、工具箱组件 (QToolBox)、标签集组件 (QTabWidget) 和分割器组件 (QSplitter)。

### 42.1 创建项目

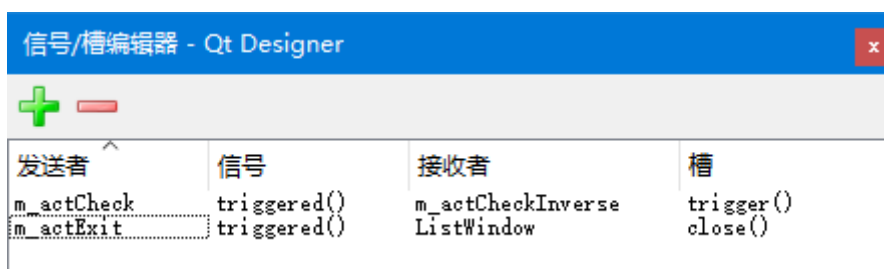
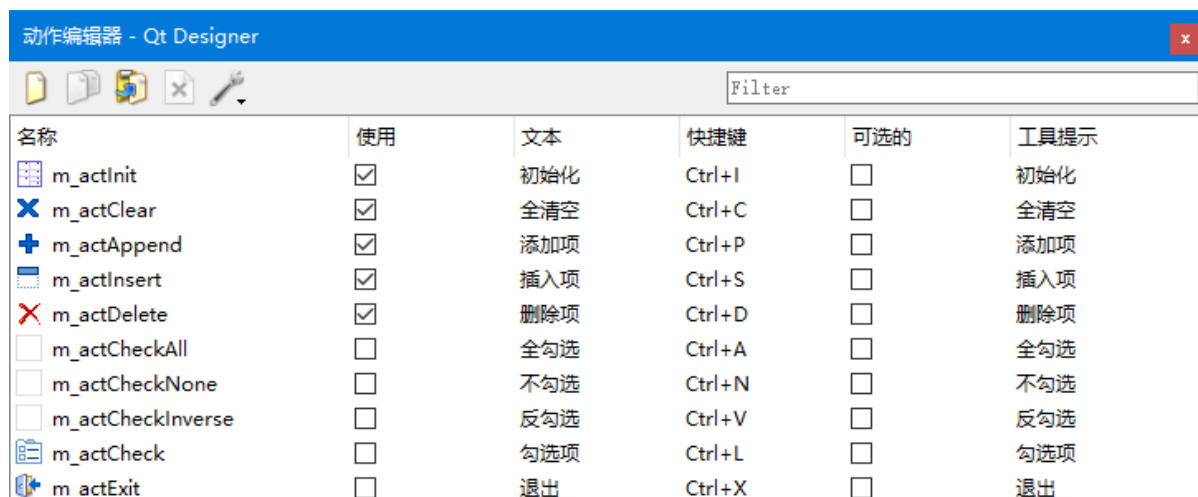
通过QtCreator, 在C:\Users\Minwei\Projects\Qt路径下, 创建名为List的项目。

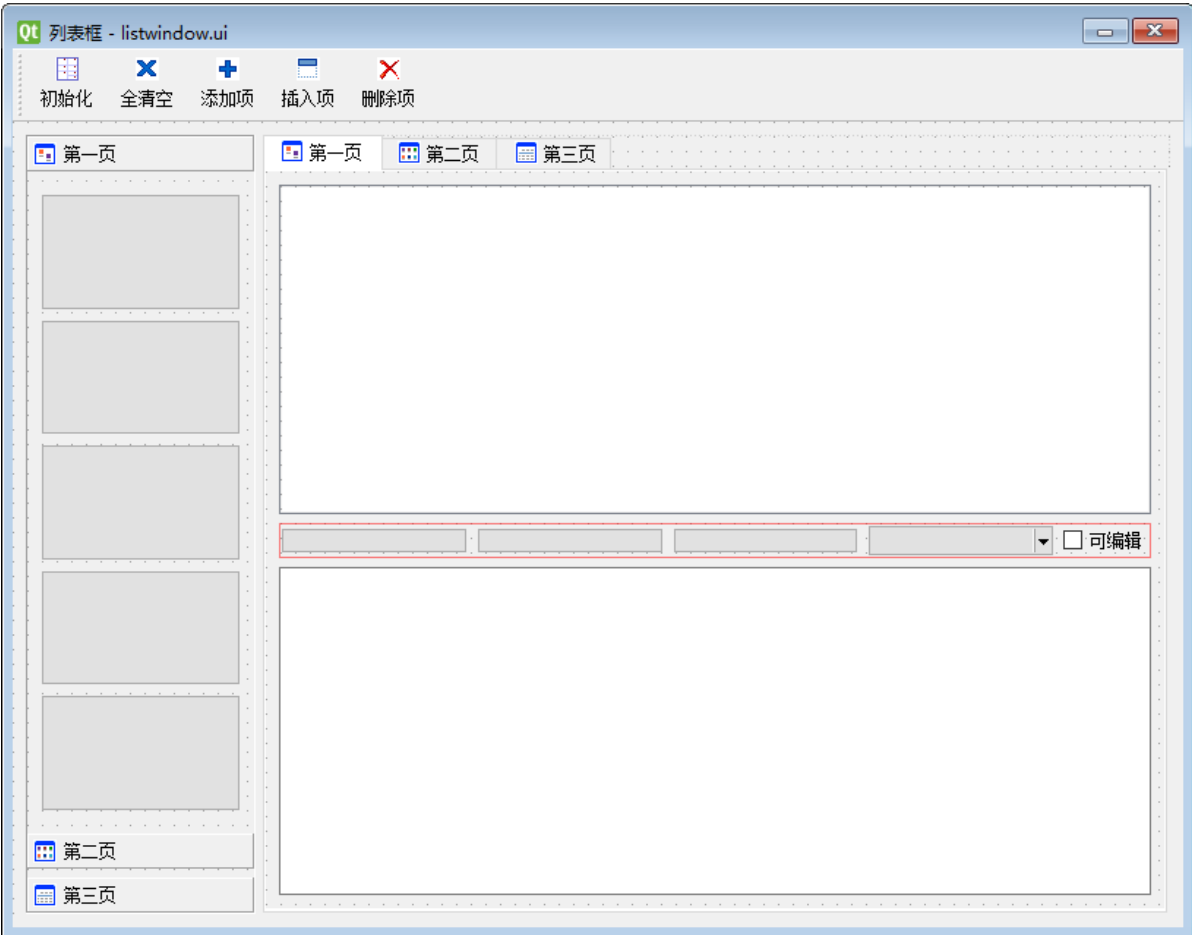
### 42.2 添加资源

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

```
1 <RCC>
2   <qresource prefix="/">
3     <file>images/append.bmp</file>
4     <file>images/check.ico</file>
5     <file>images/clear.bmp</file>
6     <file>images/delete.bmp</file>
7     <file>images/exit.bmp</file>
8     <file>images/init.bmp</file>
9     <file>images/insert.bmp</file>
10    <file>images/item.bmp</file>
11    <file>images/page1.bmp</file>
12    <file>images/page2.bmp</file>
13    <file>images/page3.bmp</file>
14    <file>images/pop.bmp</file>
15  </qresource>
16 </RCC>
```

### 42.3 设计界面





对象	类
▼ ListWindow	QMainWindow
▼ m_central	QWidget
▼ m_splitter	QSplitter
▼ m_pages	QToolBox
▼ m_page1	QWidget
m_btnAppend	QToolButton
m_btnClear	QToolButton
m_btnDelete	QToolButton
m_btnInit	QToolButton
m_btnInsert	QToolButton
m_page2	QWidget
m_page3	QWidget
▼ m_tabs	QTabWidget
▼ m_tab1	QWidget
▼ m_layoutTab1Hor	QHBoxLayout
m_btnCheck	QToolButton
m_btnCheckAll	QToolButton
m_btnCheckInverse	QToolButton
m_btnCheckNone	QToolButton
m_checkEditable	QCheckBox
m_editChanged	QPlainTextEdit
m_list	QListWidget
m_tab2	QWidget
m_tab3	QWidget
m_toolBar	QToolBar
m_actInit	QAction
m_actClear	QAction
m_actAppend	QAction
m_actInsert	QAction
m_actDelete	QAction

C:\Users\Minwei\Projects\Qt\List\listwindow.ui:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3 <class>Listwindow</class>
4 <widget class="QMainWindow" name="Listwindow">
5 <property name="geometry">
6 <rect>
7 <x>0</x>
8 <y>0</y>
9 <width>800</width>
10 <height>600</height>
11 </rect>
12 </property>
13 <property name="windowTitle">
14 <string>列表框</string>
15 </property>
16 <widget class="QWidget" name="m_central">
17 <layout class="QHBoxLayout" name="m_layoutHor">
18 <item>

```

```

19 <widget class="QSplitter" name="m_splitter">
20 <property name="orientation">
21 <enum>Qt::Horizontal</enum>
22 </property>
23 <widget class="QToolBox" name="m_pages">
24 <property name="frameShape">
25 <enum>QFrame::Panel</enum>
26 </property>
27 <property name="frameShadow">
28 <enum>QFrame::Sunken</enum>
29 </property>
30 <property name="currentIndex">
31 <number>0</number>
32 </property>
33 <widget class="QWidget" name="m_page1">
34 <property name="geometry">
35 <rect>
36 <x>0</x>
37 <y>0</y>
38 <width>155</width>
39 <height>440</height>
40 </rect>
41 </property>
42 <attribute name="icon">
43 <iconset resource="List.qrc">
44
45 <normaloff>:/images/page1.bmp</normaloff>:/images/page1.bmp</iconset>
46 </attribute>
47 <attribute name="label">
48 <string>第一页</string>
49 </attribute>
50 <layout class="QVBoxLayout" name="m_layoutPage1">
51 <item>
52 <widget class="QPushButton" name="m_btnInit">
53 <property name="sizePolicy">
54 <sizepolicy hsize="Preferred" vsize="Preferred">
55 <horstretch>0</horstretch>
56 <verstretch>0</verstretch>
57 </sizepolicy>
58 </property>
59 <property name="toolButtonStyle">
60 <enum>Qt::ToolButtonTextOnly</enum>
61 </property>
62 </widget>
63 </item>
64 <item>
65 <widget class="QPushButton" name="m_btnClear">
66 <property name="sizePolicy">
67 <sizepolicy hsize="Preferred" vsize="Preferred">
68 <horstretch>0</horstretch>
69 <verstretch>0</verstretch>
70 </sizepolicy>
71 </property>
72 <property name="toolButtonStyle">
73 <enum>Qt::ToolButtonTextOnly</enum>

```

```

74     </widget>
75 </item>
76 <item>
77     <widget class="QPushButton" name="m_btnAppend">
78         <property name="sizePolicy">
79             <sizepolicy hstretch="Preferred" vstretch="Preferred">
80                 <horstretch>0</horstretch>
81                 <verstretch>0</verstretch>
82             </sizepolicy>
83         </property>
84         <property name="toolButtonStyle">
85             <enum>Qt::ToolButtonTextOnly</enum>
86         </property>
87     </widget>
88 </item>
89 <item>
90     <widget class="QPushButton" name="m_btnInsert">
91         <property name="sizePolicy">
92             <sizepolicy hstretch="Preferred" vstretch="Preferred">
93                 <horstretch>0</horstretch>
94                 <verstretch>0</verstretch>
95             </sizepolicy>
96         </property>
97         <property name="toolButtonStyle">
98             <enum>Qt::ToolButtonTextOnly</enum>
99         </property>
100    </widget>
101 </item>
102 <item>
103     <widget class="QPushButton" name="m_btnDelete">
104         <property name="sizePolicy">
105             <sizepolicy hstretch="Preferred" vstretch="Preferred">
106                 <horstretch>0</horstretch>
107                 <verstretch>0</verstretch>
108             </sizepolicy>
109         </property>
110         <property name="toolButtonStyle">
111             <enum>Qt::ToolButtonTextOnly</enum>
112         </property>
113     </widget>
114 </item>
115 </layout>
116 </widget>
117 <widget class="QWidget" name="m_page2">
118     <property name="geometry">
119         <rect>
120             <x>0</x>
121             <y>0</y>
122             <width>155</width>
123             <height>440</height>
124         </rect>
125     </property>
126     <attribute name="icon">
127         <iconset resource="List.qrc">
128
<normaloff>:/images/page2.bmp</normaloff>:/images/page2.bmp</iconset>

```

```

129     </attribute>
130     <attribute name="label">
131         <string>第二页</string>
132     </attribute>
133 </widget>
134 <widget class="QWidget" name="m_page3">
135     <property name="geometry">
136         <rect>
137             <x>0</x>
138             <y>0</y>
139             <width>155</width>
140             <height>440</height>
141         </rect>
142     </property>
143     <attribute name="icon">
144         <iconset resource="List.qrc">
145
146 <normaloff>:/images/page3.bmp</normaloff>:/images/page3.bmp</iconset>
147     </attribute>
148     <attribute name="label">
149         <string>第三页</string>
150     </attribute>
151 </widget>
152 </widget>
153 <widget class="QTabWidget" name="m_tabs">
154     <property name="currentIndex">
155         <number>0</number>
156     </property>
157     <widget class="QWidget" name="m_tab1">
158         <attribute name="icon">
159             <iconset resource="List.qrc">
160
161 <normaloff>:/images/page1.bmp</normaloff>:/images/page1.bmp</iconset>
162     </attribute>
163     <attribute name="title">
164         <string>第一页</string>
165     </attribute>
166     <layout class="QVBoxLayout" name="m_layoutTab1">
167         <item>
168             <widget class="QListWidget" name="m_list">
169                 <property name="contextMenuPolicy">
170                     <enum>Qt::CustomContextMenu</enum>
171                 </property>
172             </widget>
173         </item>
174         <item>
175             <layout class="QHBoxLayout" name="m_layoutTab1Hor">
176                 <item>
177                     <widget class="QToolButton" name="m_btnCheckAll">
178                         <property name="sizePolicy">
179                             <sizepolicy hsize="Preferred" vsize="Fixed">
180                                 <horstretch>0</horstretch>
181                                 <verstretch>0</verstretch>
182                             </sizepolicy>
183                         </property>
184                         <property name="toolButtonStyle">

```

```
183         <enum>Qt::ToolButtonTextOnly</enum>
184     </property>
185 </widget>
186 </item>
187 <item>
188     <widget class="QToolButton" name="m_btnCheckNone">
189         <property name="sizePolicy">
190             <sizepolicy hsizeType="Preferred" vsizeType="Fixed">
191                 <horstretch>0</horstretch>
192                 <verstretch>0</verstretch>
193             </sizepolicy>
194         </property>
195         <property name="toolButtonStyle">
196             <enum>Qt::ToolButtonTextOnly</enum>
197         </property>
198     </widget>
199 </item>
200 <item>
201     <widget class="QToolButton" name="m_btnCheckInverse">
202         <property name="sizePolicy">
203             <sizepolicy hsizeType="Preferred" vsizeType="Fixed">
204                 <horstretch>0</horstretch>
205                 <verstretch>0</verstretch>
206             </sizepolicy>
207         </property>
208         <property name="toolButtonStyle">
209             <enum>Qt::ToolButtonTextOnly</enum>
210         </property>
211     </widget>
212 </item>
213 <item>
214     <widget class="QToolButton" name="m_btnCheck">
215         <property name="sizePolicy">
216             <sizepolicy hsizeType="Preferred" vsizeType="Fixed">
217                 <horstretch>0</horstretch>
218                 <verstretch>0</verstretch>
219             </sizepolicy>
220         </property>
221         <property name="popupMode">
222             <enum>QToolButton::MenuButtonPopup</enum>
223         </property>
224         <property name="toolButtonStyle">
225             <enum>Qt::ToolButtonTextBesideIcon</enum>
226         </property>
227     </widget>
228 </item>
229 <item>
230     <widget class="QCheckBox" name="m_checkEditable">
231         <property name="sizePolicy">
232             <sizepolicy hsizeType="Fixed" vsizeType="Fixed">
233                 <horstretch>0</horstretch>
234                 <verstretch>0</verstretch>
235             </sizepolicy>
236         </property>
237         <property name="text">
238             <string>可编辑</string>
```

```

239         </property>
240     </widget>
241 </item>
242 </layout>
243 </item>
244 <item>
245     <widget class="QPlainTextEdit" name="m_editChanged">
246         <property name="readOnly">
247             <bool>true</bool>
248         </property>
249     </widget>
250 </item>
251 </layout>
252 </widget>
253 <widget class="Qwidget" name="m_tab2">
254     <attribute name="icon">
255         <iconset resource="List.qrc">
256             <normaloff>:/images/page2.bmp</normaloff>:/images/page2.bmp</iconset>
257         </attribute>
258         <attribute name="title">
259             <string>第二页</string>
260         </attribute>
261     </widget>
262 <widget class="Qwidget" name="m_tab3">
263     <attribute name="icon">
264         <iconset resource="List.qrc">
265             <normaloff>:/images/page3.bmp</normaloff>:/images/page3.bmp</iconset>
266         </attribute>
267         <attribute name="title">
268             <string>第三页</string>
269         </attribute>
270     </widget>
271 </widget>
272 </widget>
273 </item>
274 </layout>
275 </widget>
276 <widget class="QToolBar" name="m_toolBar">
277     <property name="windowTitle">
278         <string>工具栏</string>
279     </property>
280     <property name="toolButtonStyle">
281         <enum>Qt::ToolButtonTextUnderIcon</enum>
282     </property>
283     <attribute name="toolBarArea">
284         <enum>TopToolBarArea</enum>
285     </attribute>
286     <attribute name="toolBarBreak">
287         <bool>>false</bool>
288     </attribute>
289     <addaction name="m_actInit"/>
290     <addaction name="m_actClear"/>
291     <addaction name="m_actAppend"/>
292     <addaction name="m_actInsert"/>

```



```
293     <addaction name="m_actDelete"/>
294 </widget>
295 <action name="m_actInit">
296     <property name="icon">
297         <iconset resource="List.qrc">
298             <normaloff>:/images/init.bmp</normaloff>:/images/init.bmp</iconset>
299         </property>
300     <property name="text">
301         <string>初始化</string>
302     </property>
303     <property name="toolTip">
304         <string>初始化</string>
305     </property>
306     <property name="shortcut">
307         <string>Ctrl+I</string>
308     </property>
309 </action>
310 <action name="m_actClear">
311     <property name="icon">
312         <iconset resource="List.qrc">
313             <normaloff>:/images/clear.bmp</normaloff>:/images/clear.bmp</iconset>
314         </property>
315     <property name="text">
316         <string>全清空</string>
317     </property>
318     <property name="toolTip">
319         <string>全清空</string>
320     </property>
321     <property name="shortcut">
322         <string>Ctrl+C</string>
323     </property>
324 </action>
325 <action name="m_actAppend">
326     <property name="icon">
327         <iconset resource="List.qrc">
328             <normaloff>:/images/append.bmp</normaloff>:/images/append.bmp</iconset>
329         </property>
330     <property name="text">
331         <string>添加项</string>
332     </property>
333     <property name="toolTip">
334         <string>添加项</string>
335     </property>
336     <property name="shortcut">
337         <string>Ctrl+P</string>
338     </property>
339 </action>
340 <action name="m_actInsert">
341     <property name="icon">
342         <iconset resource="List.qrc">
343             <normaloff>:/images/insert.bmp</normaloff>:/images/insert.bmp</iconset>
344         </property>
345     <property name="text">
346         <string>插入项</string>
```

```
347     </property>
348     <property name="toolTip">
349         <string>插入项</string>
350     </property>
351     <property name="shortcut">
352         <string>Ctrl+S</string>
353     </property>
354 </action>
355 <action name="m_actDelete">
356     <property name="icon">
357         <iconset resource="List.qrc">
358             <normaloff>:/images/delete.bmp</normaloff>:/images/delete.bmp</iconset>
359     </property>
360     <property name="text">
361         <string>删除项</string>
362     </property>
363     <property name="toolTip">
364         <string>删除项</string>
365     </property>
366     <property name="shortcut">
367         <string>Ctrl+D</string>
368     </property>
369 </action>
370 <action name="m_actCheckAll">
371     <property name="text">
372         <string>全勾选</string>
373     </property>
374     <property name="toolTip">
375         <string>全勾选</string>
376     </property>
377     <property name="shortcut">
378         <string>Ctrl+A</string>
379     </property>
380 </action>
381 <action name="m_actCheckNone">
382     <property name="text">
383         <string>不勾选</string>
384     </property>
385     <property name="toolTip">
386         <string>不勾选</string>
387     </property>
388     <property name="shortcut">
389         <string>Ctrl+N</string>
390     </property>
391 </action>
392 <action name="m_actCheckInverse">
393     <property name="text">
394         <string>反勾选</string>
395     </property>
396     <property name="toolTip">
397         <string>反勾选</string>
398     </property>
399     <property name="shortcut">
400         <string>Ctrl+V</string>
401     </property>
```

```
402 </action>
403 <action name="m_actCheck">
404   <property name="icon">
405     <iconset resource="List.qrc">
406       <normaloff>:/images/item.bmp</normaloff>:/images/item.bmp</iconset>
407     </property>
408     <property name="text">
409       <string>勾选项</string>
410     </property>
411     <property name="toolTip">
412       <string>勾选项</string>
413     </property>
414     <property name="shortcut">
415       <string>Ctrl+L</string>
416     </property>
417   </action>
418 <action name="m_actExit">
419   <property name="icon">
420     <iconset resource="List.qrc">
421       <normaloff>:/images/exit.bmp</normaloff>:/images/exit.bmp</iconset>
422     </property>
423     <property name="text">
424       <string>退出</string>
425     </property>
426     <property name="toolTip">
427       <string>退出</string>
428     </property>
429     <property name="shortcut">
430       <string>Ctrl+X</string>
431     </property>
432   </action>
433 </widget>
434 <resources>
435   <include location="List.qrc"/>
436 </resources>
437 <connections>
438   <connection>
439     <sender>m_actCheck</sender>
440     <signal>triggered</signal>
441     <receiver>m_actCheckInverse</receiver>
442     <slot>trigger</slot>
443     <hints>
444       <hint type="sourcelabel">
445         <x>-1</x>
446         <y>-1</y>
447       </hint>
448       <hint type="destinationlabel">
449         <x>-1</x>
450         <y>-1</y>
451       </hint>
452     </hints>
453   </connection>
454   <connection>
455     <sender>m_actExit</sender>
456     <signal>triggered</signal>
457     <receiver>ListWindow</receiver>
```

```

458     <slot>close()</slot>
459     <hints>
460         <hint type="sourceLabel">
461             <x>-1</x>
462             <y>-1</y>
463         </hint>
464         <hint type="destinationLabel">
465             <x>399</x>
466             <y>299</y>
467         </hint>
468     </hints>
469 </connection>
470 </connections>
471 </ui>

```

## 42.4 实现功能

C:\Users\Minwei\Projects\Qt\List\listwindow.h:

```

1  #ifndef LISTWINDOW_H
2  #define LISTWINDOW_H
3
4  #include <QMainWindow>
5  #include <QListWidgetItem>
6
7  QT_BEGIN_NAMESPACE
8  namespace Ui { class Listwindow; }
9  QT_END_NAMESPACE
10
11 class Listwindow : public QMainWindow
12 {
13     Q_OBJECT
14
15 public:
16     Listwindow(QWidget *parent = nullptr);
17     ~Listwindow();
18
19 private slots:
20     void on_m_actInit_triggered();
21     void on_m_actClear_triggered();
22     void on_m_actAppend_triggered();
23     void on_m_actInsert_triggered();
24     void on_m_actDelete_triggered();
25
26     void on_m_actCheckAll_triggered();
27     void on_m_actCheckNone_triggered();
28     void on_m_actCheckInverse_triggered();
29
30     void on_m_checkEditable_clicked(bool checked);
31
32     void on_m_list_currentItemChanged(QListWidgetItem *current,
33     QListWidgetItem *previous);
34     void on_m_list_customContextMenuRequested(const QPoint &pos);
35     void on_m_pages_currentChanged(int index);

```

```

36     void on_m_tabs_currentChanged(int index);
37
38 private:
39     Ui::Listwindow *ui;
40 };
41
42 #endif // LISTWINDOW_H

```

C:\Users\Minwei\Projects\Qt\List\listwindow.cpp:

```

1  #include <QMenu>
2
3  #include "listwindow.h"
4  #include "ui_listwindow.h"
5
6  Listwindow::Listwindow(QWidget *parent)
7      : QMainWindow(parent)
8      , ui(new Ui::Listwindow)
9  {
10     ui->setupUi(this);
11
12     ui->m_btnInit->setDefaultAction(ui->m_actInit);
13     ui->m_btnClear->setDefaultAction(ui->m_actClear);
14     ui->m_btnAppend->setDefaultAction(ui->m_actAppend);
15     ui->m_btnInsert->setDefaultAction(ui->m_actInsert);
16     ui->m_btnDelete->setDefaultAction(ui->m_actDelete);
17
18     ui->m_btnCheckAll->setDefaultAction(ui->m_actCheckAll);
19     ui->m_btnCheckNone->setDefaultAction(ui->m_actCheckNone);
20     ui->m_btnCheckInverse->setDefaultAction(ui->m_actCheckInverse);
21     ui->m_btnCheck->setDefaultAction(ui->m_actCheck);
22
23     QMenu* menuCheck = new QMenu(this);
24     menuCheck->addAction(ui->m_actCheckAll);
25     menuCheck->addAction(ui->m_actCheckNone);
26     menuCheck->addAction(ui->m_actCheckInverse);
27     ui->m_btnCheck->setMenu(menuCheck);
28
29     ui->m_toolBar->addSeparator();
30
31     QPushButton* btnCheck = new QPushButton(this);
32     btnCheck->setPopupMode(QPushButton::InstantPopup);
33     btnCheck->setToolButtonStyle(Qt::ToolButtonTextUnderIcon);
34     btnCheck->setDefaultAction(ui->m_actCheck);
35     btnCheck->setMenu(menuCheck);
36     ui->m_toolBar->addWidget(btnCheck);
37
38     ui->m_toolBar->addAction(ui->m_actExit);
39 }
40
41 Listwindow::~Listwindow()
42 {
43     delete ui;
44 }
45

```

```

46 void Listwindow::on_m_actInit_triggered()
47 {
48     for (int i = 1; i <= 3; ++i)
49     {
50         QListWidgetItem* item = new QListWidgetItem(
51             QIcon(":/images/check.ico"), QString::asprintf("初始列表项%d",
52             i));
53         item->setCheckState(Qt::Checked);
54         if (ui->m_checkEditable->isChecked())
55             item->setFlags(item->flags() | Qt::ItemIsEditable);
56
57         ui->m_list->addItem(item);
58     }
59 }
60
61 void Listwindow::on_m_actClear_triggered()
62 {
63     ui->m_list->clear();
64 }
65
66 void Listwindow::on_m_actAppend_triggered()
67 {
68     QListWidgetItem* item = new QListWidgetItem(
69         QIcon(":/images/check.ico"), "添加列表项");
70
71     item->setCheckState(Qt::Unchecked);
72     if (ui->m_checkEditable->isChecked())
73         item->setFlags(item->flags() | Qt::ItemIsEditable);
74
75     ui->m_list->addItem(item);
76 }
77
78 void Listwindow::on_m_actInsert_triggered()
79 {
80     QListWidgetItem* item = new QListWidgetItem(
81         QIcon(":/images/check.ico"), "插入列表项");
82
83     item->setCheckState(Qt::Unchecked);
84     if (ui->m_checkEditable->isChecked())
85         item->setFlags(item->flags() | Qt::ItemIsEditable);
86
87     ui->m_list->insertItem(ui->m_list->currentRow(), item);
88 }
89
90 void Listwindow::on_m_actDelete_triggered()
91 {
92     delete ui->m_list->takeItem(ui->m_list->currentRow());
93 }
94
95 void Listwindow::on_m_actCheckAll_triggered()
96 {
97     int nitems = ui->m_list->count();
98
99     for (int i = 0; i < nitems; ++i)
100         ui->m_list->item(i)->setCheckState(Qt::Checked);

```

```

101 }
102
103 void Listwindow::on_m_actCheckNone_triggered()
104 {
105     int nitems = ui->m_list->count();
106
107     for (int i = 0; i < nitems; ++i)
108         ui->m_list->item(i)->setCheckState(Qt::Unchecked);
109 }
110
111 void Listwindow::on_m_actCheckInverse_triggered()
112 {
113     int nitems = ui->m_list->count();
114
115     for (int i = 0; i < nitems; ++i)
116         ui->m_list->item(i)->setCheckState(
117             ui->m_list->item(i)->checkState() == Qt::Checked ?
118             Qt::Unchecked : Qt::Checked);
119 }
120
121 void Listwindow::on_m_checkEditable_clicked(bool checked)
122 {
123     int nitems = ui->m_list->count();
124
125     for (int i = 0; i < nitems; ++i)
126     {
127         QListWidgetItem* item = ui->m_list->item(i);
128
129         if (checked)
130             item->setFlags(item->flags() | Qt::ItemIsEditable);
131         else
132             item->setFlags(item->flags() & ~Qt::ItemIsEditable);
133     }
134 }
135
136 void Listwindow::on_m_list_currentItemChanged(QListWidgetItem *current,
137 QListWidgetItem *previous)
138 {
139     QString text = "[";
140
141     if (previous)
142         text += previous->text();
143
144     text += "]->[";
145
146     if (current)
147         text += current->text();
148
149     text += "];";
150
151     ui->m_editChanged->appendPlainText(text);
152 }
153 void Listwindow::on_m_list_customContextMenuRequested(const QPoint &pos)
154 {
155     Q_UNUSED(pos)

```

```
156
157     QMenu* menuPopup = new QMenu(this);
158
159     menuPopup->addAction(ui->m_actInit);
160     menuPopup->addAction(ui->m_actClear);
161     menuPopup->addAction(ui->m_actAppend);
162     menuPopup->addAction(ui->m_actInsert);
163     menuPopup->addAction(ui->m_actDelete);
164
165     menuPopup->addSeparator();
166
167     menuPopup->addAction(ui->m_actCheckAll);
168     menuPopup->addAction(ui->m_actCheckNone);
169     menuPopup->addAction(ui->m_actCheckInverse);
170
171     menuPopup->exec(QCursor::pos());
172
173     delete menuPopup;
174 }
175
176 void Listwindow::on_m_pages_currentChanged(int index)
177 {
178     ui->m_tabs->setCurrentIndex(index);
179 }
180
181 void Listwindow::on_m_tabs_currentChanged(int index)
182 {
183     ui->m_pages->setCurrentIndex(index);
184 }
```

## 42.5 测试验证

运行效果如图所示:



