**ПРАКТИЧЕСКАЯ РАБОТА**

**ЭЛЕМЕНТЫ УПРАВЛЕНИЯ**

***RadioButton***

Компонент RadioButton представляет собой переключатель, состояние которого зависит от состояния других переключателей (компонентов RadioButton). Обычно компоненты RadioButton объединяют в группу (достигается это путем размещения нескольких компонентов в поле компонента GroupBox). В каждый момент времени только один из переключателей группы может находиться в выбранном состоянии (возможна ситуация, когда ни один из переключателей не выбран). Состояние компонентов, принадлежащих одной группе, не зависит от состояния компонентов, принадлежащих другой группе.

Свойства компонента приведены в табл. 3.5.

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| Text | Текст, который находится справа от переключателя |
| Checked | Состояние, внешний вид переключателя. Если переключатель выбран, то значение свойства Checked равно True; если не выбран, то значение свойства Checked равно False |
| TextAllign | Положение текста в поле отображения текста. Текст может располагаться в центре поля (MiddleCenter), быть прижат к левой (MiddleLeft) или правой (MiddleRight) границе. Можно задать и другие способы размеще­ния текста надписи (TopLeft. TopCenter. TopRight, BottomLeft, BottomCenter, BottomRight) |
| CheckAllign | Положение кнопки в поле компонента. Кнопка может быть прижата к левой верхней границе (TopLeft), прижата к левой границе и находиться на  равном расстоянии от верхней и нижней границ поля компонента (Middleleft). Есть и другие варианты размещения кнопки в поле компонента |
| Enabled | Свойство позволяет сделать переключатель недоступным (False) |
| Visible | Свойство позволяет скрыть (False) переключатель |
| AutcCheck | Свойство определяет, должно ли автоматически изменяться состояние переключателя в результате щелчка на его изображении. По умолчанию значение равно True |
| FlatStyle | Стиль переключателя. Переключатель может быть обычным (Standard), плоским (Flat) или "всплывающим" (Popup). Стиль переключателя опре­деляет его поведение при позиционировании указателя мыши на изобра­жении кнопки |
| Appearance | Определяет вид переключателя. Переключатель может выглядеть обыч­ным образом (Normal) или как кнопка (Button) |
| linage | Картинка, которая отображается в попе компонента |
| ImageAlign | Положение картинки в поле компонента. Картинка может располагаться в центре (MiddleCenter), быть прижата к левой (Middleleft) или правой (MiddleRight) границе. Можно задать и **другие** способы размещения кар­тинки на кнопке (TopLeft. TopCenter. TopRight, BottomLeft, BottomCenter. BottomRight) |
| InageList | Набор картинок, используемых для обозначения различных состояний переключателя. Представляет собой объект типа imageList. Чтобы за­дать значение свойства, в форму приложения надо добавить компонент ImageList |
| Image Index | Номер (индекс) картинки из набора Imagelist, которая отображается в попе компонента |

Состояние переключателя изменяется в результате щелчка на его изображении (если значение свойства AutoCheck равно True). При этом возникает событие CheckedChanged, а затем событие Click. Если значение свойства AutoCheck равно False, то в результате щелчка на переключателе возникает событие Click, а затем, если процедура обработки этого события изменит состояние кнопки, возникает событие CheckedChanged.

Программа "Фото" (ее форма приведена ниже, а текст — в листинге 3.6) демонстрирует использование компонента RadioButton. Программа вычисляет стоимость заказа печати фотографий, в зависимости от их размера. Значения свойств компонентов RadioButton приведены в табл. 3.6. Следует обратить внимание: на форму сначала надо поместить компонент GroupBox, затем — компоненты RadioButton. Помимо события Click, возникающего при щелчке на кнопке OK, в программе обрабатывается событие Click компонентов RadioButton. Функция обработки этого события (одна для всех компонентов) очищает поле отображения результата и устанавливает курсор в поле ввода/редактирования.

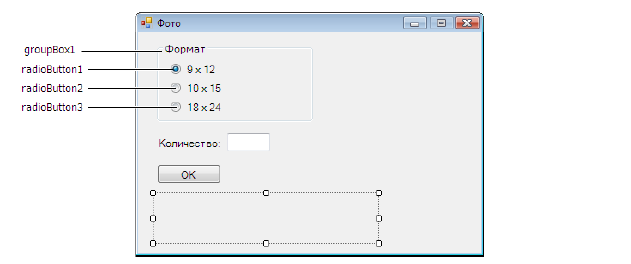
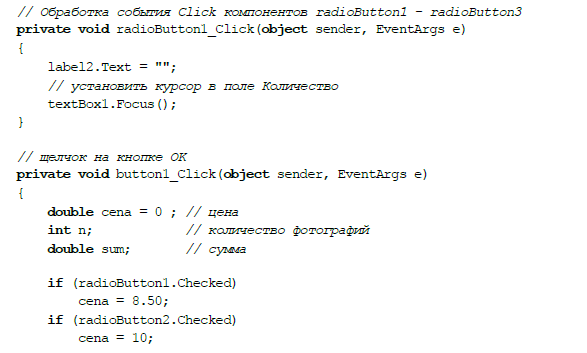


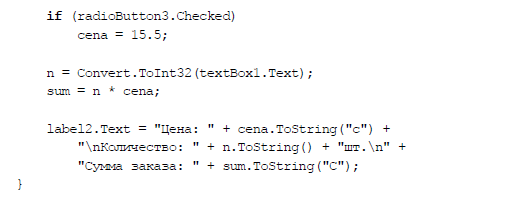
Рис. 3.11. Форма программы "Фото"

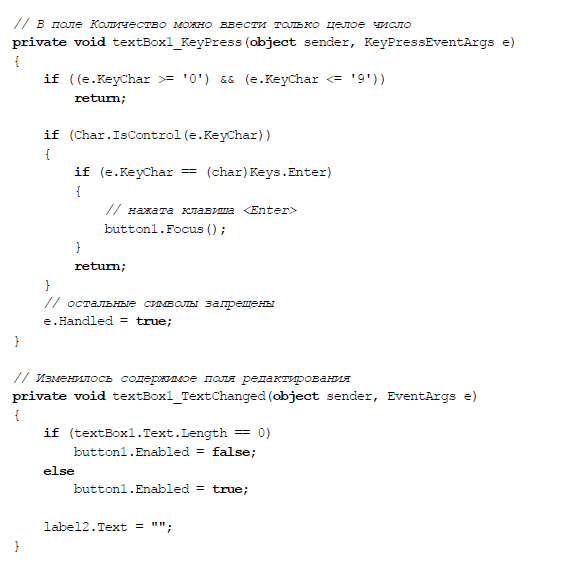
Таблица 3.6. Значения свойств компонентов RadioButton

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| radioButtonl. Text | 9x12 |
| radioButtonl.Checked | true |
| radioButton2.Texc | 10x15 |
| radioButton3.Text | 18x24 |

Листинг. Фото (функции обработки событий)







GroupBox

Компонент GroupBox представляет собой контейнер для других компонентов. Обычно он используется для объединения в группы компонентов RadioButton по функциональному признаку.

Свойства компонента GroupBox приведены в табл. 3.7.

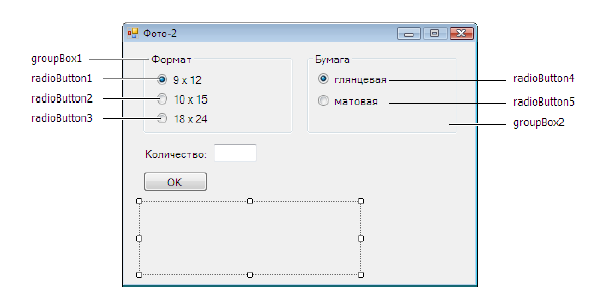
Таблица 3.7. Свойства компонента GroupBox

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| Texr | Заголовок — текст, поясняющий назначение компонентов, которые находятся в поле компонента GroupBox |
| Enabled | Позволяет управлять доступом к компонентам, находящимся в поле |
|  | (на поверхности) компонента GroupBox. Если значение свойства равно False, то все находящиеся в поле GroupBox компоненты недоступны |
| Visible | Позволяет скрыть (сделать невидимым) компонент GroupBox и все компоненты, которые находятся на его поверхности |

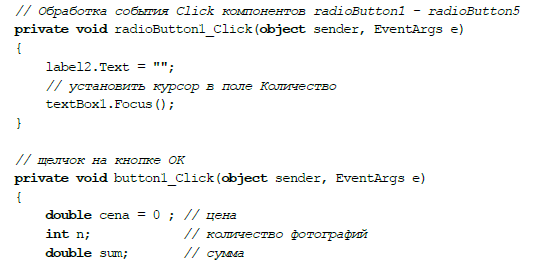
Следующая программа (ее форма приведена на рис. 3.13, а текст - в листинге 3.7) демонстрирует использование компонентов GroupBox и RadioButton. Компоненты RadioButton принадлежат к двум разным группам (находятся в полях разных компонентов GroupBox). Это позволяет установить в выбранное состояние два переключателя одновременно, по одному в каждой группе. Значения свойств компонентов GroupBox и RadioButton приведены в табл. 3.8.

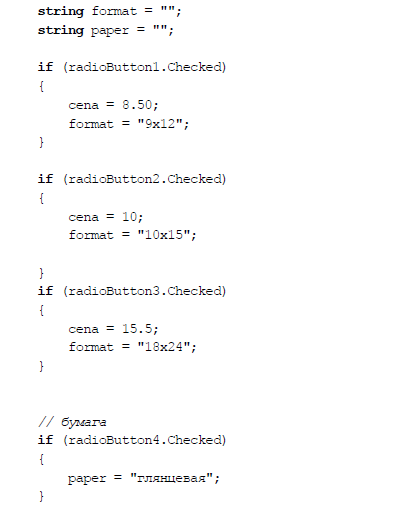
Таблица 3.8. Значения свойств компонентов

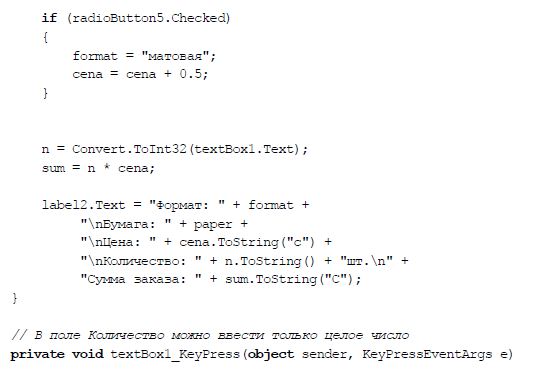
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Компонент** | **Свойство** | **Значение** |  | |
| groupBoxl | Text | Формат |  | |
| radioButtonl | Text | 9x12 |  | |
| Checked | true |  | |
| radioButton2 | Text | 10x15 | |
| radioButton3 | Text | 18x24 | |
| groupBox2 | Text | Бумага | |
| radioButton4 | Text | глянцевая | |
| Checked | true | |
| radioButton5 | Text | матовая | |



ЛИСТИНГ. ФОТО-2 (Компоненты GroupBox и RadioButton)









**ComboBox**

Компонент **ComboBox** представляет собой комбинацию поля редактирования и списка, что позволяет вводить данные путем набора на клавиатуре или выбора значения в списке.

Свойства компонента приведены в табл. 3.9.

Таблица 3.9. Свойства компонента ComboBox

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| DropDownStyle | Вид компонента: DropDown — поле ввода и раскрывающийся список; Simple — поле ввода и список; DropDownList — раскры­вающийся список |
| Text | Текст, находящийся в поле ввода/редактирования (для компонен­тов типа DropDown и Simple) |
| Items | Элементы списка — коллекция строк |
| Items.Count | Количество элементов списка |
| Items .Selectedlndex | Номер элемента, выбранного в списке. Если ни один из элементов списка не выбран, то значение свойства равно -1 |
| Sorted | Признак необходимости автоматической сортировки (True) списка после добавления очередного элемента |
| MaxDropDownltems | Количество отображаемых элементов в раскрытом списке. Если количество элементов списка больше чем MaxDropDovnltem.s. то появляется вертикальная полоса прокрутки |
| Location | Положение компонента на поверхности формы |
| Size | Размер компонента без (для компонентов типа DropDown. и DropDownList) или с учетом (для компонента типа Simple) раз­мера области списка или области ввода |
| DropDownWidth | Ширина области списка |
| Font | Шрифт, используемый для отображения содержимого поля редак­тирования и элементов списка |

Список, отображаемый в поле компонента, можно сформировать во время создания формы или во время работы программы. Чтобы сформировать список во время создания формы, надо в окне Properties выбрать свойство Items, щелкнуть на кнопке с тремя точками (она находится поле значения строке свойства Items) и в появившемся окне String Collection Editor (рис. 3.15) ввести элементы списка.

Чтобы сформировать список во время работы программы, надо применить метод Add к свойству Items. Например, следующий фрагмент кода формирует упорядо-ченный по алфавиту список.

comboBox1->Items->Add("пластик");

comboBox1->Items->Add("алюминий");

comboBox1->Items->Add("соломка");

comboBox1->Items->Add("текстиль");

comboBox1->Items->Add("бамбук");

comboBox1->Sorted = true;

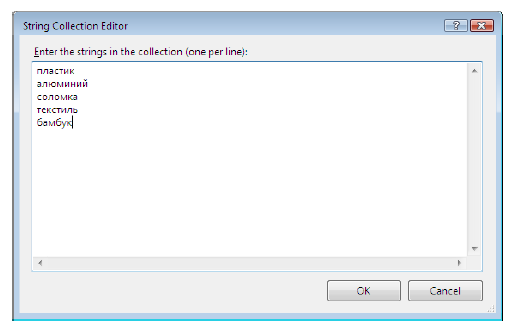


Рис. 3.15. Формирование списка компонента ComboBox во время создания формы

Программа "Жалюзи" демонстрирует использование компонента ComboBox для ввода данных. Форма программы приведена на рис. 3.16, текст — в листинге 3.8. Настройку компонента ComboBox выполняет конструктор формы — свойству SelectedIndex присваивает значение "минус один" (значения свойств компонента ComboBox приведены в табл. 3.10). Поэтому при появлении формы на экране название материала не отображается (ни один из элементов списка не выбран). Кроме того, кнопка OK становится доступной только после выбора материала и ввода размеров жалюзи. Доступностью кнопки управляет функция обработки события TextChanged полей редактирования (одна функция обрабатывает событие TextChanged обоих компонентов).

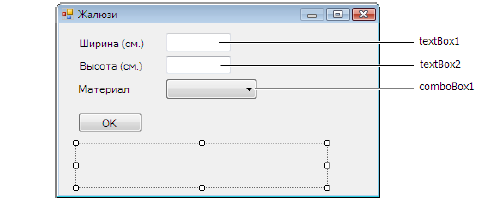


Рис. 3.16. Форма программы "Жалюзи"

Таблица 3.10. Значения свойств компонента ComboBox

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| ComboBox З. DropDownStyle | DropDownList |
| ComboBox З.Sorted | false |

Листинг. Жалюзи (компонент ComboBox)

