Компонент **ListView** представляет собой список, позволяющий организовать сложную структуру для своих элементов, а также задать настройки их отображения при визуальном представлении. Основные свойства компонента **ListView**:

* **View** – определяет режим отображения элементов в списке:
  + **LargeIcon** – большие иконки;
  + **SmallIcon** – маленькие иконки;
  + **List** – маленькие иконки списком;
  + **Details** – подробный режим;
  + **Tile** – большие снимки.
* **Activation** – определяет реакцию на ввод:
  + **Standart** – реакция определяется установками операционной системы;
  + **OneClick** – реакция после одинарного щелчка кнопкой мыши;
  + **TwoClick** – реакция после двойного щелчка кнопкой мыши.
* **AllowColumnReorder** – для режима списка (**Details**) предоставляет пользователю возможность менять порядок колонок в таблице свойств.
* **AutoArrange** – определяет для режимов маленьких иконок (**SmallIcon**) и больших иконок (**LargeIcon**), нужно ли автоматически сортировать элементы списка.
* **CheckBoxes** – определяет, будут ли отображаться флажки (компонент **CheckBox**) напротив элементов списка.
* **Columns** – определяет набор колонок, отображаемых в подробном режиме (**Details**).
* **FullRowSelect** – определяет для подробного режима, следует ли выделять не только заголовок элемента, но и всю строку, содержащую дополнительную информацию об элементе в так называемых подчинённых элементах (**SubItems**), отображаемых в колонках, состав которых был задан в свойстве **Columns**.
* **Groups** – определяет группы элементов списка и их состав.
* **HeaderStyle** – определяет стиль заголовков колонок:
  + **Clickable** – заголовки колонок работают, как кнопки (доступен щелчок);
  + **NonClickable** – по заголовку колонки нельзя щелкнуть, как по кнопке;
  + **None** – заголовки у колонок отсутствуют.
* **HideSelection** – определяет, следует ли скрывать выделение, когда с элемента управления снимается фокус.
* **HotTracking** – определяет, следует ли при событии наведения на элемент изменять иконку стрелки курсора на указывающую руку.
* **Items** – определяет набор элементов списка.
* **LabelEdit** – определяет, можно ли редактировать заголовки элементов.
* **LabelWrap** – определяет, можно ли разбивать длинные заголовки элементов на несколько строк.
* **LargeImageList** – определяет набор иконок для больших картинок в виде компонента **ImageList**. Картинки из набора могут быть назначены элементам списка по индексу.
* **MultiSelect** – определяет, следует ли разрешить множественное выделение в списке.
* **ShowGroups** – определяет, следует ли отображать группы.
* **SmallImageList** – определяет набор иконок для маленьких картинок.

Поскольку свойство **Items** компонента **ListView** представляет собой коллекцию, он обладает методом **Add**, который позволяет программно добавить новый элемент в список:

LV1.Items.Add("Заголовок элемента");

Элементы списка являются объектами класса **ListViewItem**. В связи с этим бывает удобно предварительно создавать объект, представляющий новый элемент списка, настраивать его свойства и только после этого добавлять созданный элемент в список:

ListViewItem NewItem = new ListViewItem("Мой элемент");

NewItem.Group = LV1.Groups[0]; ***// Определяем группу элемента***

LV1.Items.Add(NewItem); ***// Добавляем элемент в список***

Группы элементов также представляют собой объекты, но принадлежащие к классу **ListViewGroup**. Соответственно, группа также может быть предварительно создана в качестве отдельного объекта, а уже затем добавлена в коллекцию групп компонента **ListView** при помощи метода **Add**:

ListViewGroup NewGroup = new ListViewGroup("Моя группа");

LV1.Groups.Add(NewGroup); ***// Добавляем группу в список групп***

При создании нового объекта, представляющего элемент списка **ListView**, его группа может быть определена в параметрах конструктора:

ListViewItem NewItem = new ListViewItem("Элемент", NewGroup);

LV1.Items.Add(NewItem);

Помимо элементов списка и их групп, коллекцией также является набор выделенных пользователем элементов списка, на который ссылается свойство **SelectedItems** компонента **ListView**. Перебор элементов данной коллекции может быть осуществлён в цикле **foreach**:

foreach (ListViewItem i in LV1.SelectedItems)

MessageBox.Show(i.Text);

Удаление элемента списка может быть осуществлено при помощи метода **Remove**. В следующем примере из коллекции элементов списка удаляется первый по индексу элемент:

if (LV1.Items.Count > 0) ***// Проверяем, не пуст ли список***

LV1.Items.Remove(LV1.Items[0]); ***// Удаляем первый элемент***

Удаление элемента из списка может быть осуществлено не только по ссылке на элемент, но и непосредственно по индексу этого элемента. Для этого следует использовать метод **RemoveAt** коллекции элементов.

if (LV1.Items.Count > 5) ***// Проверяем возможность удаления***

LV1.Items.RemoveAt(5); ***// Удаляем пятый элемент из списка***

В числе событий компонента **ListView** следует отметить следующие:

* **SelectedIndexChanged** – возникает при изменении индекса выделенного элемента;
* **SearchForVirtualItem** – возникает при наборе названия элемента на клавиатуре при его поиске в списке;
* **VirtualItemsSelectionRangeChanged** – возникает при изменении набора выделенных элементов;
* **ItemActivate** – возникает при активации элемента способом, заданным в свойстве **Activation** компонента **ListView**;
* **ItemCheck** – возникает перед тем, как у элемента списка изменится свойство **Checked**;
* **ItemChecked** – возникает, когда какой-либо из элементов списка помечается флажком;
* **ItemDrag** – возникает, когда пользователь начинает тянуть элемент списка мышкой;
* **ItemMouseHover** – возникает, когда курсор мыши движется над поверхностью элемента списка;
* **ItemSelectionChanged** – возникает при изменении выделенного элемента в списке;
* **DrawItem** – возникает при перерисовке элемента в режиме **OwnerDraw** (одноимённому свойству должно быть присвоено значение **true**);
* **DrawSubItem** – возникает при отрисовке подчинённого элемента из коллекции **SubItems**;
* **DrawColumnHeader** – возникает при отрисовке заголовка колонки;
* **ColumnClick** – возникает при щелчке по заголовку колонки;
* **ColumnWidthChanged** – возникает при изменении ширины колонки.