Компонент **TreeView** позволяет организовать отображение информации в формате древовидной структуры. Наиболее важными свойствами компонента являются:

* **CheckBoxes** – определяет возможность помечать элементы дерева флажками (компонент **CheckBox**).
* **HideSelection** – определяет, следует ли скрывать выделение элемента дерева, когда компонент **TreeView** теряет фокус.
* **HotTracking** – определяет, следует ли выделять цветом и подчёркиванием (как гиперссылку) элементы дерева при наведении на них курсора.
* **LabelEdit** – определяет возможность редактирования заголовков элементов дерева.
* **Nodes** – определяет состав коллекции элементов дерева.
* **ShowLines** – определяет, следует ли отображать линии-разделители между элементами дерева.
* **ShowPlusMinus** – определяет, следует ли отображать рядом с элементами, содержащими дочерние элементы, кнопку управления раскрытием ветки (подписывается как «+» или «–»).
* **ShowRootLines** – определяет, следует ли отображать линии и кнопку управления раскрытием ветки у корневого элемента дерева.

Элементы дерева **TreeView** являются объектами класса **TreeNode**. Далее приведён пример программного добавления корневого элемента в дерево просмотра **TV1**:

TreeNode NewNode = new TreeNode();

NewNode.Text = "Корневой элемент"; ***// Задаём текст элемента***

TV1.Nodes.Add(NewNode); ***// Добавляем элемент в дерево***

Для добавления нового элемента в дерево также можно было воспользоваться дополнительным конструктором класса **TreeNode**:

TreeNode NewNode = new TreeNode("Корневой элемент");

TV1.Nodes.Add(NewNode);

Кроме того, можно было использовать перегруженный вариант метода **Add**:

TV1.Nodes.Add("Корневой элемент");

Объекты класса **TreeNode** позволяют хранить в себе коллекцию ссылок на объекты этого же класса, которые в дереве **TreeView** отображаются как дочерние элементы. В следующем примере для выделенного в дереве просмотра **TV1** элемента создаётся дочерний элемент:

if (TV1.SelectedNode != null) ***// Проверяем, выделен ли элемент***

TV1.SelectedNode.Nodes.Add("Дочерний элемент");

Здесь следует пояснить, что в свойстве **SelectedNode** компонента **TreeView** хранится ссылка на выделенный элемент дерева.

Удаление из дерева выделенного элемента осуществляется по схеме, аналогичной приведённому примеру:

if (TV1.SelectedNode != null)

TV1.SelectedNode.Remove(); ***// Удаляем элемент из дерева***

Удалить дочерние элементы у выделенного элемента дерева можно следующим образом:

TV1.SelectedNode.Nodes.Clear();

Компонент **TreeView** обладает следующими событиями:

* **AfterCheck** – возникает при установке/снятии флажка у элемента дерева. Второй параметр обработчика данного события имеет тип **TreeViewEventsArgs**, свойство **Node** которого содержит ссылку на помеченный элемент дерева.
* **AfterCollapse** – возникает при сворачивании ветки дерева. Второй параметр обработчика данного события имеет тип **TreeViewEventsArgs**, свойство **Node** которого содержит ссылку на свёрнутый элемент дерева.
* **AfterExpand** – возникает при разворачивании ветки дерева. Второй параметр обработчика данного события имеет тип **TreeViewEventsArgs**, свойство **Node** которого содержит ссылку на развёрнутый элемент дерева.
* **AfterLabelEdit** – возникает при завершении редактирования заголовка элемента дерева. Второй параметр обработчика данного события имеет тип **NodeLabelEditEventsArgs**, который обладает следующим свойствами:
  + **CancelEdit** – установка значения **false** отменяет внесённые изменения;
  + **Label** – содержит новый текст, назначенный элементу;
  + **Node** – содержит ссылку на изменяемый элемент дерева.
* **AfterSelect** – возникает при выборе какого-либо элемента в дереве. При этом в свойстве **SelectedNode** компонента **TreeView** будет храниться ссылка на новый выбранный элемент дерева.
* **BeforeSelect** – возникает перед выбором какого-либо элемента в дереве. При этом в свойстве **SelectedNode** компонента **TreeView** будет храниться ссылка на старый выбранный элемент дерева.
* **DrawNode** – возникает при отрисовке элемента дерева (генерируется только при включенном режиме отрисовки **OwnerDraw**).
* **NodeMouseClick** – возникает при одинарном щелчке по элементу дерева. Второй параметр обработчика данного события имеет тип **TreeNodeMouseClickEventsArgs**, свойства **Node** и **Button** которого позволяют определить, по какому элементу дерева был произведён щелчок и какой кнопкой мыши.
* **NodeMouseDoubleClick** – возникает при двойном щелчке по элементу дерева. Второй параметр обработчика данного события имеет тип **TreeNodeMouseClickEventsArgs**, свойства **Node** и **Button** которого позволяют определить, по какому элементу дерева был произведён щелчок и какой кнопкой мыши.
* **NodeMouseHover** – возникает при движении курсора мыши над элементом дерева.