

1.1.6.10. Duyệt tất cả các phần tử của ListView

```
for (int i = 0; i < lvwSanPham.Items.Count; i++)  
{  
    // you code here...  
}
```

Hoặc:

```
foreach (ListViewItem lvi in lvwSanPham.Items)  
{  
    // you code here...  
}
```

Lưu ý: không dùng foreach để xóa phần tử trong ListView

1.2. TreeView

1.2.1. Khái niệm

TreeView là điều khiển cho phép trình bày danh sách phần tử ở dạng cây

Mỗi phần tử của cây được gọi là một node (là đối tượng thuộc lớp TreeNode)

Lớp TreeView thuộc namespace: System.Windows.Forms

1.2.2. Một số thuộc tính

1.2.2.1. CheckBoxes

- ❖ Kiểu dữ liệu: bool
- ❖ Mô tả:
 - Cho phép hiển thị checkbox bên trái mỗi node trong TreeView
 - Giá trị mặc định là false
 - Checkbox được dùng để chọn nhiều node trong TreeView cùng lúc

1.2.2.2. FullRowSelect

- ❖ Kiểu dữ liệu: bool
- ❖ Mô tả:
 - Cho phép tô vùng chọn của node được chọn theo chiều rộng của điều khiển TreeView thay vì theo độ dài nhãn của node
 - Giá trị mặc định là false
 - Thuộc tính FullRowSelect sẽ bị bỏ qua nếu thuộc tính ShowLines = true

1.2.2.3. ImageIndex

- ❖ Kiểu dữ liệu: int
- ❖ Mô tả:
 - Chỉ số của hình ảnh (trong ImageList) được dùng làm hình ảnh mặc định cho các node trong TreeView

1.2.2.4. ImageList

- ❖ Kiểu dữ liệu: ImageList
- ❖ Mô tả:
 - Chứa danh sách các hình ảnh được sử dụng cho các node trong TreeView
 - Giá trị mặc định là null

1.2.2.5. LabelEdit

- ❖ Kiểu dữ liệu: bool
- ❖ Mô tả:
 - Cho phép chỉnh sửa nội dung của nhãn của node trong TreeView khi thực thi (run time)
 - Giá trị mặc định là null

1.2.2.6. Nodes

- ❖ Kiểu dữ liệu: TreeNodeCollection
- ❖ Mô tả:
 - Danh sách các node trong TreeView
 - Mỗi node trong TreeView lại có một danh sách các node con (TreeNodeCollection)

1.2.2.7. Scrollable

- ❖ Kiểu dữ liệu: bool
- ❖ Mô tả:
 - Cho phép hiển thị thanh cuộn trên điều khiển TreeView
 - Giá trị mặc định là false

1.2.2.8. *SelectedNode*

- ❖ Kiểu dữ liệu: `TreeNode`
- ❖ Mô tả:
 - Node được chọn trong `TreeView`
 - Nếu không có node nào được chọn thì `SelectedNode = null`

1.2.2.9. *ShowLines*

- ❖ Kiểu dữ liệu: `bool`
- ❖ Mô tả:
 - Cho phép hiển thị đường kẻ giữa các node trong `TreeView`
 - Giá trị mặc định là `false`

1.2.2.10. *ShowPlusMinus*

- ❖ Kiểu dữ liệu: `bool`
- ❖ Mô tả:
 - Cho phép hiển thị nút có dấu cộng `[+]` và dấu trừ `[-]` dùng để hiện/ẩn các node con
 - Giá trị mặc định là `false`

1.2.2.11. *ShowRootLines*

- ❖ Kiểu dữ liệu: `bool`
- ❖ Mô tả:
 - Cho phép hiển thị đường nối giữa các node cha trong `TreeView`
 - Giá trị mặc định là `false`
 - Nếu `ShowRootLines = false` thì thuộc tính `ShowPlusMinus` sẽ không có tác dụng

1.2.3. Một số phương thức

1.2.3.1. *Phương thức khởi tạo*

- ❖ Cú pháp: `public TreeView()`
- ❖ Mô tả:
 - Dùng để tạo một điều khiển `TreeView`

1.2.3.2. CollapseAll()

- ❖ Cú pháp: `public void CollapseAll()`
- ❖ Mô tả:
 - Thu gọn tất cả node con trong TreeView

1.2.3.3. ExpandAll()

- ❖ Cú pháp: `public void ExpandAll()`
- ❖ Mô tả:
 - Hiển thị tất cả node con trong TreeView

1.2.3.4. GetNodeCount(bool includeSubTrees)

- ❖ Cú pháp: `public int GetNodeCount(bool includeSubTrees)`
- ❖ Mô tả:
 - Đếm số lượng node trong TreeView
 - Tham số: includeSubTrees (true: đếm cả các node của cây con, false: không đếm các node của cây con)

1.2.4. Một số sự kiện

1.2.4.1. AfterSelect

Xảy ra sau khi node trong TreeView được chọn

1.2.4.2. AfterCollapse

Xảy ra sau khi node trong TreeView được thu gọn

1.2.4.3. AfterExpand

Xảy ra sau khi node trong TreeView được mở rộng

1.2.4.4. Click

Xảy ra khi click vào điều khiển TreeView

1.2.5. Lớp TreeNode

TreeNode là lớp đối tượng thể hiện cho một node trong TreeView. Hay nói cách khác, mỗi node trong TreeView có kiểu dữ liệu là TreeNode.

Lớp TreeNode thuộc namespace `Sytem.Windows.Forms`.

1.2.5.1. Một số thuộc tính

❖ Checked

- Kiểu dữ liệu: bool
- Có giá trị true nếu node được check vào checkbox, ngược lại có giá trị false (mặc định)

❖ FirstNode

- Kiểu dữ liệu: TreeNode
- Là node đầu tiên trong danh sách các node con của node hiện hành. Nếu node hiện hành không có node con nào thì FirstNode = null

❖ FullPath

- Kiểu dữ liệu: string
- Là chuỗi bao gồm các nhãn của các node dẫn đến node hiện hành bắt đầu từ node gốc của TreeView. Ngăn cách giữa các nhãn của mỗi node là dấu “\”. Ví dụ: “Country\VietNam\TPHoChiMinh”, trong đó node gốc của TreeView là Country, node hiện hành là TPHoChiMinh.

❖ ImageIndex

- Kiểu dữ liệu: int
- Chỉ số của hình ảnh (trong ImageList) của node khi chưa được chọn

❖ Index

- Kiểu dữ liệu: int
- Chỉ số của node trong cây con

❖ LastNode

- Kiểu dữ liệu: TreeNode
- Là node cuối cùng trong danh sách các node con của node hiện hành. Nếu node hiện hành không có node con nào thì LastNode = null

❖ Level

- Kiểu dữ liệu: int

- Độ sâu của node trong cây. Node gốc có Level = 0

❖ NextNode

- Kiểu dữ liệu: TreeNode
- Là node kế tiếp (cùng node cha) của node hiện hành. Nếu node hiện hành không có node kế tiếp nào thì NextNode = null

❖ Nodes

- Kiểu dữ liệu: TreeNodeCollection
- Danh sách các node con của node hiện hành

❖ Parent

- Kiểu dữ liệu: TreeNode
- Là node cha của node hiện hành. Nếu node hiện hành là node gốc thì Parent = null

❖ PrevNode

- Kiểu dữ liệu: TreeNode
- Là node kế trước (cùng node cha) của node hiện hành. Nếu node hiện hành không có node kế trước nào thì PrevNode = null

❖ SelectedImageIndex

- Kiểu dữ liệu: int
- Chỉ số của hình ảnh (trong ImageList) của node khi được chọn

❖ Text

- Kiểu dữ liệu: string
- Nội dung (văn bản) được hiển thị của node

1.2.5.2. Một số phương thức

❖ Phương thức khởi tạo

- `public TreeNode()`
- `public TreeNode(string text)`
- `public TreeNode(string text, int imageIndex, int selectedImageIndex)`

❖ `public void Collapse()`

Thu gọn node hiện hành và các node con của node hiện hành

❖ `public void Collapse(bool ignoreChildren)`

Nếu `ignoreChildren = true` thì thu gọn node hiện hành, không thu gọn các node con của node hiện hành. Ngược lại giống với phương thức `Collapse()`

❖ `public void Expand()`

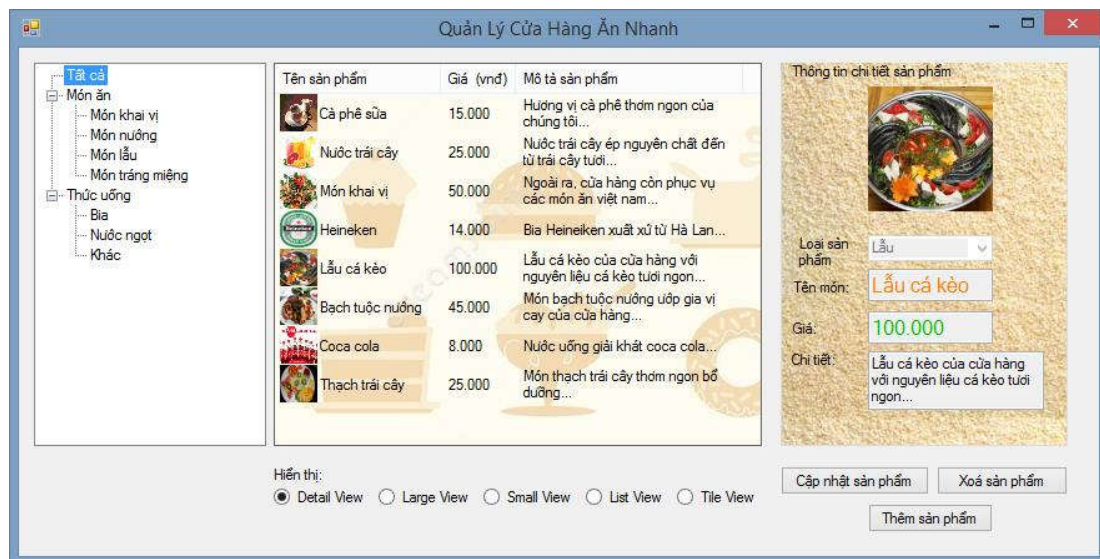
Mở rộng node hiện hành, không mở rộng các node con của node hiện hành

❖ `public void ExpandAll()`

Mở rộng node hiện hành và tất cả node con của node hiện hành

1.2.6. Ví dụ minh họa

Phát triển từ ứng dụng “*Quản Lý Cửa Hàng Ăn Nhanh*” với giao diện như sau:

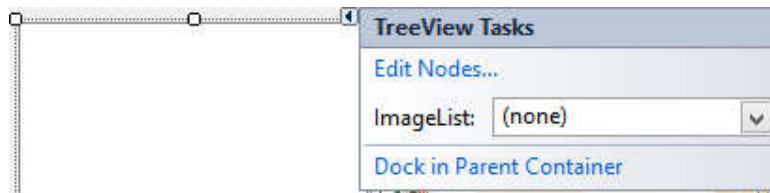


1.2.6.1. Tạo TreeView (design)

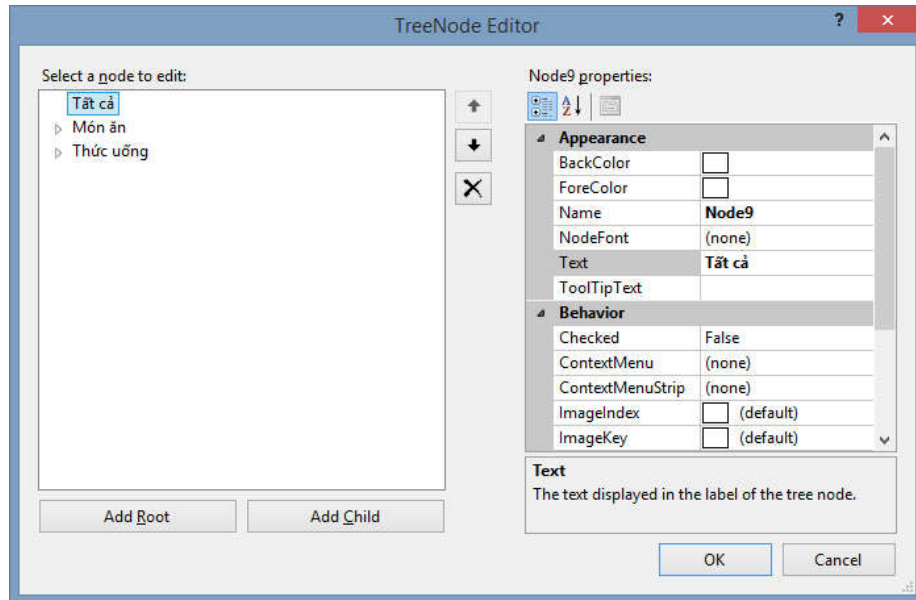
❖ Kéo điều khiển TreeView từ hộp công cụ (Toolbox) vào Form và đặt tên cho điều khiển TreeView. Ví dụ trong ứng dụng TreeView có tên là `treLoaiSanPham`.

❖ Thêm node vào điều khiển TreeView:

- Nhấn chuột vào hình tam giác góc trên bên phải của điều khiển TreeView, chọn `Edit Nodes...`

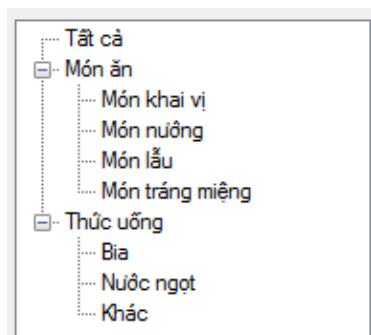


Trong hộp thoại TreeNode Editor, nhấn vào nút:



- Add Root: để thêm node gốc vào treeview
- Add Child: để thêm node con vào node được chọn trong treeview

❖ Kết quả:



1.2.6.2. Thêm Node vào TreeView (lập trình)

❖ Thêm node cha

```
// 1. Tạo đối tượng TreeNode với nhãn của node
TreeNode tn = new TreeNode("Tất cả");
// 2. Thêm đối tượng treenode vào treeview
treLoaiSanPham.Nodes.Add(tn);
```

Tương tự thêm các node “Thức ăn”, “Món uống” vào TreeView

❖ Thêm node con

- Thêm node “Món khai vị” vào node “Món ăn”

```
// 1. Lấy node Món ăn (index = 1) trong treeview
TreeNode tnRoot = treLoaiSanPham.Nodes[1];
// 2. Tạo node Món khai vị
TreeNode tnChild = new TreeNode("Món khai vị");
// 3. Thêm node Món khai vị vào node Món ăn
tnRoot.Nodes.Add(tnChild);
```

- Thêm node có nhãn được nhập từ TextBox txtLoaiSanPham vào node được chọn trong TreeView

```
// 1. Lấy node được chọn trong treeview
TreeNode tnRoot = treLoaiSanPham.SelectedNode;
// 2. Tạo node mới
TreeNode tnChild = new TreeNode(txtLoaiSanPham.Text);
// 3. Thêm node mới vào node được chọn
tnRoot.Nodes.Add(tnChild);
```

1.2.6.3. Truy xuất đến Node trong TreeView

❖ Lấy node được chọn

```
// Lấy node được chọn trong treeview
TreeNode tnSel = treLoaiSanPham.SelectedNode;
```

❖ Lấy node “Món ăn”

```
// Lấy node Món ăn (index = 1) trong treeview
TreeNode tnRoot = treLoaiSanPham.Nodes[1];
```

❖ Lấy node “Món lẩu”

```
// Lấy node Món ăn (index = 1) trong treeview
TreeNode tnRoot = treLoaiSanPham.Nodes[1];
// Lấy node Món lẩu (index = 2) trong node Món ăn
TreeNode tnChild = tnRoot.Nodes[2];
```

Hoặc

```
TreeNode tn = treLoaiSanPham.Nodes[1].Nodes[2];
```

1.2.6.4. Cập nhật Node

Ví dụ: cập nhật nhãn của node được chọn trong TreeView bằng nội dung được nhập từ TextBox txtLoaiSanPham

```
// 1. Lấy node được chọn trong treeview
TreeNode tnSel = treLoaiSanPham.SelectedNode;
// 2. Cập nhật nhãn
tnSel.Text = txtLoaiSanPham.Text;
```

1.2.6.5. Xóa Node khỏi Treeview

Ví dụ: xóa node được chọn trong TreeView

```
// 1. Lấy node được chọn trong treeview
TreeNode tnSel = treLoaiSanPham.SelectedNode;
// 2. Xóa node
treLoaiSanPham.Nodes.Remove(tnSel);
```

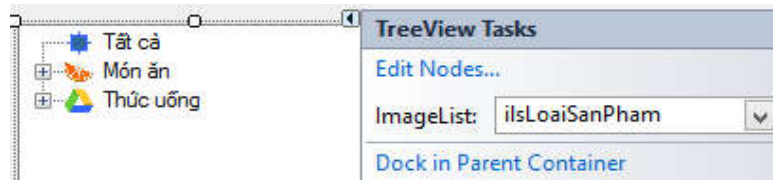
1.2.6.6. Thêm hình ảnh vào trước mỗi Node trong Treeview

Để có thể hiện thị hình ảnh trước mỗi node trong TreeView thì ta sẽ sử dụng điều khiển ImageList chứa danh sách các hình ảnh cho các node trong TreeView.

Ví dụ, trong ứng dụng sử dụng ImageList có tên ilsLoaiSanPham cho TreeView treLoaiSanPham.

❖ Gán ImageList cho TreeView

- Design:

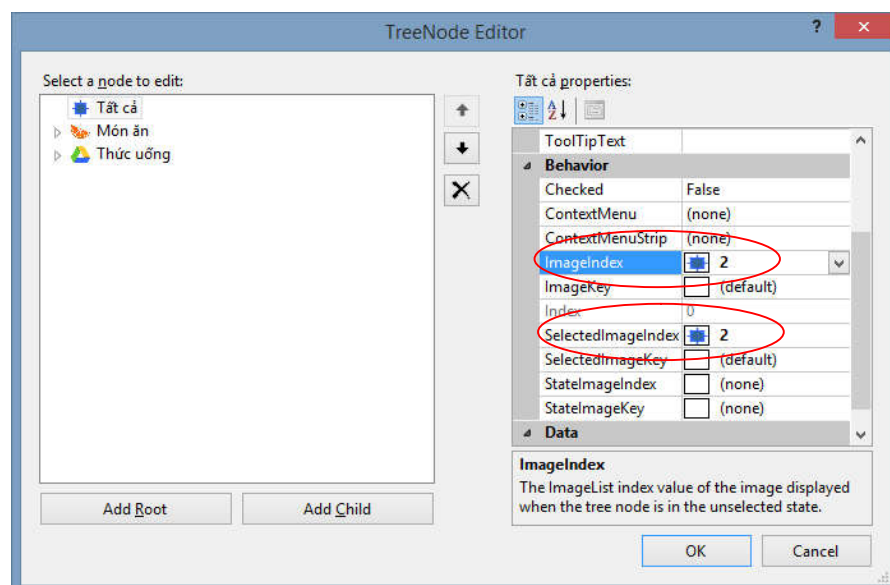


- Hoặc code:

```
treLoaiSanPham.ImageList = ilsLoaiSanPham;
```

❖ Gán hình ảnh cho các Node

- Design:



- Code:

Ví dụ: gán hình ảnh cho node “Tất cả”

```
// 1. Lấy node Tất cả (index = 0)
TreeNode tn = treLoaiSanPham.Nodes[0];
// 2. Gán hình ảnh có index = 2 trong imagelist cho node
// 2.1. Gán hình ảnh khi node không được chọn
tn.ImageIndex = 2;
// 2.2. Gán hình ảnh khi node được chọn
tn.SelectedIndex = 2;
```

1.2.6.7. Xử lý khi nhấn chuột vào Node trong TreeView

Lưu ý: để làm việc này, ta sẽ viết code xử lý sự kiện AfterSelect của TreeView

Ví dụ: Hiển thị danh sách các sản phẩm theo loại sản phẩm được chọn trong TreeView.

Ghi chú: Để biết sản phẩm loại nào, ta sử dụng thuộc tính Tag của ListViewItem và TreeNode để gán loại sản phẩm.

```
private void treLoaiSanPham_AfterSelect(object sender, TreeViewEventArgs e)
{
    if (treLoaiSanPham.SelectedNode.Tag != null)
    {
        // xoá hết sản phẩm ở lvwSanPham
        lvwSanPham.Items.Clear();

        if (treLoaiSanPham.SelectedNode.Tag.ToString() == "-1")
        {
            // thêm vào lvwSanPham tất cả các phần tử
            // của mảng lviAll
            foreach (ListViewItem lviItem in lviAll)
            {
                lvwSanPham.Items.Add((ListViewItem)lviItem.Clone());
            }
        }
        else
        {
            // tìm các sản phẩm có cùng loại (tag) với treeview
            // node (loại sản phẩm) và thêm vào lvwSanPham
            foreach (ListViewItem lviItem in lviAll)
            {
                if (lviItem.Tag.ToString() ==
                    treLoaiSanPham.SelectedNode.Tag.ToString())
                {
                    lvwSanPham.Items.Add((ListViewItem)
                                            lviItem.Clone());
                }
            }
        }
    }
}
```