Programarea Aplicațiilor Windows – curs 7

Conf. dr. Cristian CIUREA
Departamentul de Informatică și Cibernetică Economică
Academia de Studii Economice București
cristian.ciurea@ie.ase.ro

Agenda

1. Lucrul cu ferestre multiple (MDI)
2. Controale vizuale complexe (ListView, TreeView)

Lucrul cu ferestre multiple (MDI)

Arhitecturi de aplicații:

- **SDI** – Single Document Interface
- **MDI** – Multiple Document Interface

Arhitectura MDI permite utilizatorilor să lucreze simultan cu mai multe formulare fără să deschidă o nouă instanță a aplicației.

Lucrul cu ferestre multiple (MDI)

Aplicație **MDI** (Multiple Document Interface):

- mai multe ferestre copil;
- o fereastră cadru părinte, container pentru ferestrele copil.

Lucrul cu ferestre multiple (MDI)

Pentru a stabili formularul cadru părinte, se setează proprietatea:

```
this.IsMDIContainer = true;
```

În cadrul formularul părinte, se stabilesc formularele copil:

```
Form2 copil1 = new Form2();
copil1.MdiParent = this;
Form3 copil2 = new Form3();
copil2.MdiParent = this;
```

Lucrul cu ferestre multiple (MDI)

Formularele copil pot fi aranjate:

- în cascadă:

```
this.LayoutMdi(MdiLayout.Cascade);
```

- orizontal:

```
this.LayoutMdi(MdiLayout.TileHorizontal);
```

- vertical:

```
this.LayoutMdi(MdiLayout.TileVertical);
```

Controale vizuale complexe

Vizualizarea informațiilor de pe un formular se realizează sub diverse formate:

- liniar (controlul ListView);
- arborescent (controlul TreeView);
- sub formă de raport;
- listă de icon-uri.

Vizualizarea liniară – controlul ListView

Controlul ListView are următoarele formate de vizualizare:

- Details;
- LargeIcon;
- SmallIcon;
- List.

Vizualizarea liniară – controlul ListView

Pentru adăugarea coloanelor în cadrul controlului ListView se accesează proprietatea *Columns -> Collection* sau direct din cod:

```
listView1.Columns.Add("Observatii");
```

Denumirile coloanelor sunt vizibile doar în formatul de vizualizare Details.

Vizualizarea liniară – controlul ListView

Elementele unui ListView se numesc item-uri și sunt obiecte din clasa ListViewItem.

Controlul ListView deține o colecție de item-uri accesibilă prin proprietatea Items, care la rândul ei conține o colecție de subitem-uri accesibilă prin proprietatea SubItems.

Vizualizarea liniară – controlul ListView

Popularea elementelor unui ListView se realizează:

- **static**, la momentul proiectării aplicației – *Properties* -> *Items* -> *Collections* -> *Add* – pentru fiecare linie adăugată se stabilește textul de afișat (colecțiile Items și SubItems);
- **dinamic**, la momentul execuției, preluând datele dintr-o structură de date.

Vizualizarea liniară – controlul ListView

Dacă obiectul ListView are proprietatea **MultiSelect** pe true, se permite selecția mai multor item-uri în vederea prelucrării multiple a acestora.

Se poate trata evenimentul **ItemActivate** pentru prelucrarea unuia sau mai multor item-uri selectate.

Vizualizarea liniară – controlul ListView

Pentru a putea edita un item direct în ListView, la momentul execuției, se setează proprietatea **LabelEdit** pe true.

Pentru a selecta toată linia, nu doar primul subitem, se setează proprietatea **FullRowSelect** pe true.

Vizualizarea liniară – controlul ListView

Pentru a menține sortată colecția de item-uri a ListView-ului, se setează proprietatea **Sorting** astfel:

```
listView1.Sorting = SortOrder.Ascending;
```

Pentru a permite rearanjarea coloanelor prin drag&drop se setează proprietatea **AllowColumnReorder** pe true.

Vizualizarea arborescentă – TreeView

Controlul TreeView este utilizat pentru afișarea unei colecții în formă arborescentă, cu posibilitatea expandării sau comprimării unor nivele din arbore.

Fiecare nod din TreeView este un obiect din clasa TreeNode. Fiecare TreeNode stochează o colecție Nodes de noduri fii aferenți unui nivel al arborelui. Proprietatea Nodes păstrează colecția de noduri aflată pe nivelul 0 al arborelui.

Vizualizarea arborescentă – TreeView

Principalele **proprietăți** ale TreeView:

- Nodes – colecția de noduri (obiecte TreeNode);
- LabelEdit – indică dacă etichetele nodurilor pot fi editate;
- SelectedNode – indică nodul curent selectat;
- CheckBoxes – indică dacă fiecare nod va fi precedat de câte un checkbox.

Vizualizarea arborescentă – TreeView

Principalele **evenimente** ale TreeView:

- AfterSelect – evenimentul implicit al clasei TreeView care se declanșează imediat după selecția unui nod;
- AfterLabelEdit – imediat după editarea etichetei unui nod (pentru validarea datelor introduse);
- AfterExpand – imediat după expandarea unui nod.

Bibliografie

[1] I. Smeureanu, M. Dârdală, A. Reveiu – *Visual C# .NET*, Editura CISON, București, 2004.

[2] C. Petzold – *Programming Microsoft Windows with C#*, Microsoft Press, 2002.

[3] L. O'Brien, B. Eckel – *Thinking in C#*, Prentice Hall.

[4] J. Richter – *Applied Microsoft .NET Framework Programming*, Microsoft Press, 2002.

[5] http://acs.ase.ro/paw