

# Kontrole

Maja Štula

Ak. God. 2011/2012

# Kontrole

- Kontrole su prozori specifične funkcionalnosti.
- Microsoft® Windows imaju predefinirane klase s predefiniranim funkcijama za obradu poruka namijenjenih tom prozoru tj. toj kontroli.
- Gotovo sve što se može napraviti sa običnim prozorom može se napraviti i sa kontrolom.
- Windows kontrole su na raspolaganju svim aplikacijama.
- Svaku kontrolu možete isprogramirati i sami, ali korištenjem standardnih kontrola korisničko sučelje je konzistentno među aplikacijama.

# WIN32 API kontrolle

- Animation
  - ComboBox
  - Date and Time Picker
  - Edit
  - Header
  - Image Lists
  - List Box
  - Month Calendar
  - Progress Bar
  - ReBar
  - Scroll Bars
  - Status Bars
  - Tab
  - ToolTip
  - Tree-View
- Button
  - ComboBoxEx
  - Drag List Box
  - Flat Scroll Bar
  - Hot Key
  - IP Address Controls
  - List-View
  - Pager
  - Property Sheets
  - RichEdit
  - Static
  - SysLink
  - Toolbar
  - Trackbar
  - Up-Down

.NET 2.x kontrole

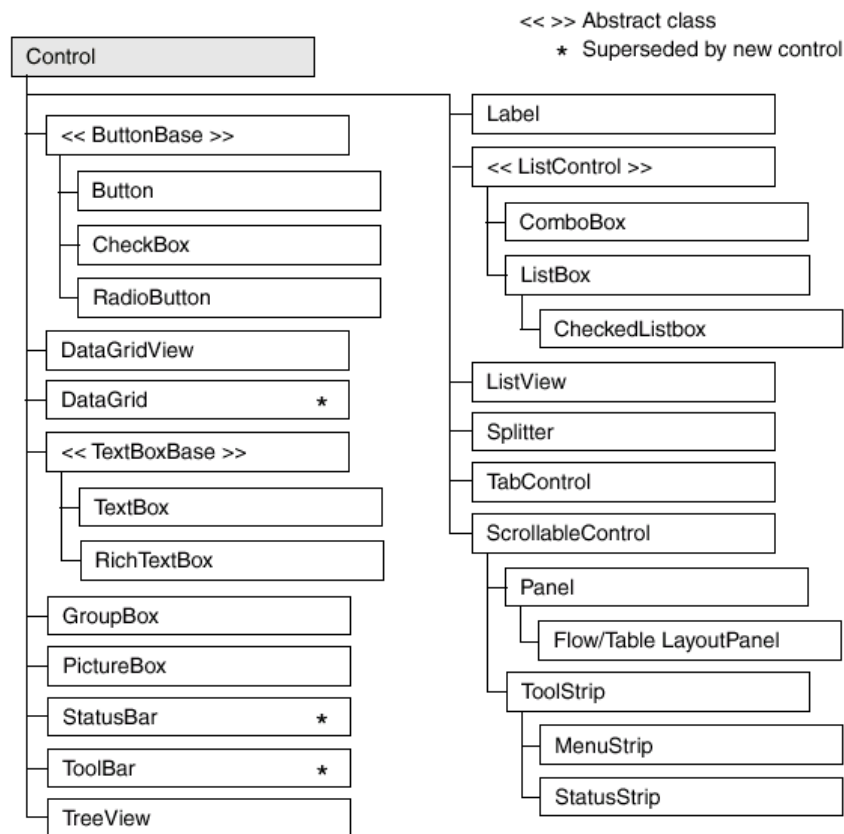
# .NET 2.x kontrole

- Klase smještene u imenskom prostoru `System.Windows.Forms` se mogu grupirati u sljedeće kategorije:
  - Klase `Form`, `Control`, i `UserControl`. Klasa `Control` pruža osnovu funkcionalnost (rad sa ulaznim uređajima, usmjeravanje poruka i sl.) kontrolama koje se prikazuju na formi i samoj formi pa većina klasa nasljeđuje klasu `Control`, direktno ili preko neke već izvedene klase. `Form` klasa se koristi za kreiranje prozora aplikacije. `UserControl` se koristi za stvaranje vlastitih kontrola.
  - Izbornici i trake s botunima (*toolbar*) se kreiraju korištenjem klasa `ToolStrip`, `MenuStrip`, `ContextMenuStrip` i `StatusStrip`.
  - Kontrole su klase sa već isprogramiranim određenim izgledom i funkcionalnostima poput `TextBox` klase, `Button`, itd. i nasljeđuju klasu `Control` (direktno ili preko neke već izvedene klase, npr. `TextBox` -> `TextBoxBase` -> `Control`).

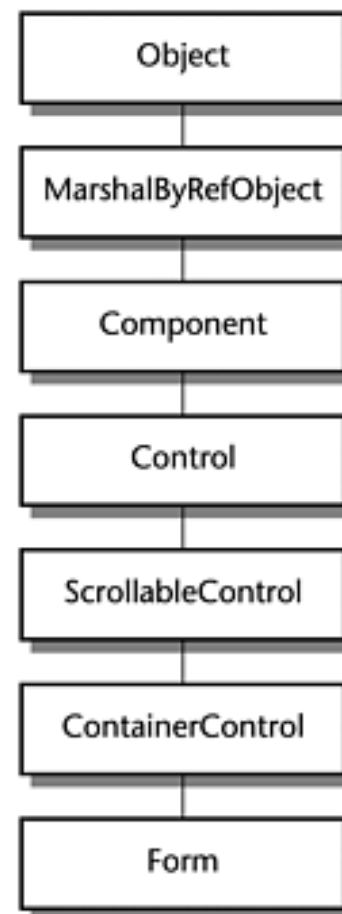
# .NET 2.x kontrole

- Klase za formatiranje (*layout*) prikaza omogućavaju automatsko formatiranje prikaza npr. kontrola na formi i sl. Klasa SplitContainer dijeli površinu prikaza na dva ili više dijelova. TableLayoutPanel omogućava definiranje redaka i stupaca za prikazivanje kontrola kao u tablici. Također nasljeđuju klasu Control.
- Klase za rad sa izvorima podataka (baze, XML datoteke i sl.) poput DataGridView, BindingSource itd., omogućavaju dohvaćanje i prikaz podataka
- Komponente su klase koje ne nasljeđuju klasu Control, ali imaju implementirano grafičko sučelje prema korisniku, poput klasa ToolTip, Help, ErrorProvider i HelpProvider.
- I tu su još i klase običnih dijalog prozora (*Common Dialog Boxes*) za izbor datoteka s diska, izbor tipa slova i sl. poput OpenFileDialog, Font i drugih klasa. MessageBox klasa je također poseban dijalog prozor za prikaz poruka.

# .NET 2.x kontrole



Sa \* su označene klase koje su 2.x zamijenjene sa novim klasama  
DataGrid->DataGridView  
StatusBar->>StatusStrip ..  
ali su zadržane zbog unazadne kompatibilnosti.



# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Prikaz podataka	<a href="#">DataGridView</a> kontrola	DataGridView kontrola pruža prilagodljivu tablicu za prikaz podataka. DataGridView klasa omogućava prilagođavanje, redaka, kolona i okvira tablice.
Povezivanja podataka (Data binding) i navigacija	BindingSource komponenta	Olakšava povezivanje kontrola na formi sa izvorima podataka.
	BindingNavigator kontrola	Pružuje sučelje tipa toolbara za navigaciju i manipulaciju podacima na formi.
Unos i izmjena teksta	TextBox kontrola	Prikazuje tekst unesen u kontrolu prilikom razvoja aplikacije koji korisnik može mijenjati.
	RichTextBox kontrola	Omogućava formatiranje teksta u ( <i>rich-text</i> ) RTF formatu. (Microsoftov <i>cross-platform document interchange</i> format iz 1987)
	MaskedTextBox kontrola	Omogućava ograničavanje formata korisničkog unosa (npr. polje za unos lozinke koje ne prikazuje uneseni karakter nego *).



# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Prikaz sadržaja	Label kontrola	Label kontrola prikazuje tekst koji korisnik ne može direktno mijenjati.
	LinkLabel kontrola	LinkLabel kontrola prikazuje tekst kao Web link i generira događaj kada korisnik klikne na tekst. Obično je to link na drugi prozor ili Web stranicu.
	StatusStrip kontrola	StatusStrip kontrola prikazuje informacije o trenutnom stanju aplikacije u uokvirenom području smještenom obično na dnu glavnog prozora.
	ProgressBar kontrola	ProgressBar kontrola prikazuje trenutno stanje korisničke operacije.
Prikaz Web stranice	WebBrowser kontrola	Omogućava korisniku navigaciju među Web stranicama unutar forme koja koristi tu kontrolu.

# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Izbor elementa iz liste	CheckedListBox kontrola	CheckedListBox kontrola prikazuje pomičnu listu elemenata uz koje je vezana i Checkbox kontrola.
	ComboBox kontrola	Prikazuje padajuću listu elemenata.
	DomainUpDown kontrola	Prikazuje listu tekstualnih elemenata kroz koje se korisnik može pomicati <i>up</i> i <i>down</i> botunima tipkovnice.
	ListBox kontrola	Prikazuje listu tekstualnih i grafičkih elemenata (ikona).
	ListView kontrola	Prikazuje elemente na jedan od četiri načina: samo tekst, tekst sa malim ikonama, tekst sa velikim ikonama i detaljni prikaz.
	NumericUpDown kontrola	Prikazuje listu brojčanih elemenata kroz koje se korisnik može pomicati <i>up</i> i <i>down</i> botunima tipkovnice.
	TreeView kontrola	Prikazuje hijerarhijski organiziranu kolekciju objekata koji se mogu sastojati od teksta sa opcionalnim Checkboxovima ili ikonama.

# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Grafički prikaz	PictureBox kontrola	PictureBox kontrola prikazuje grafički datoteku poput bitmape ili ikone.
Grafička pohrana	ImageList kontrola	Služi kao spremište slika.
Postavljanje vrijednosti	CheckBox kontrola	Sastoji se od kutije koju korisnik može selektirati i labela. Općenito se koristi za postavljanje opcija.
	CheckedListBox kontrola	
	RadioButton kontrola	Botun koji se može uključiti ili isključiti.
	TrackBar kontrola	Omogućava korisniku postavljanje vrijednosti na skali pomicanjem male sličice po skali.
Postavljanje datuma	DateTimePicker kontrola	Grafički prikazan kalendar koji korisniku omogućava izbor datum ili vremena.
	MonthCalendar kontrola	Grafički prikazan kalendar koji korisniku omogućava izbor raspona datuma.

# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Dijalog prozori	ColorDialog kontrola	ColorDialog kontrola prikazuje dijalog prozor za odabir boje elementa sučelja.
	FontDialog kontrola	Prikazuje dijalog prozor za odabir fonta i atributa fonta.
	OpenFileDialog kontrola	Prikazuje dijalog prozor koji korisniku omogućava navigaciju po disku i izbor datoteke.
	PrintDialog kontrola	Prikazuje dijalog prozor koji korisniku omogućava izbor printera i postavljanje atributa printera.
	PrintPreviewDialog kontrola	Prikazuje dijalog prozor koji prikazuje kako će se isprintati komponenta PrintDocument.
	FolderBrowserDialog kontrola	Prikazuje dijalog prozor koji korisniku omogućava pregledavanje, kreiranje i izbor direktorija.
	SaveFileDialog kontrola	Prikazuje dijalog prozor koji korisniku omogućava spremanje datoteke.

# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Izbornik	MenuStrip kontrola	MenuStrip kontrola kreira uobičajeni izbornik.
	ContextMenuStrip kontrola	ContextMenuStrip kontrola kreira uobičajeni izbornik sa dodacima.
Komande	Button kontrola	Pokreće, zaustavlja ili prekida proces.
	LinkLabel kontrola	
	NotifyIcon kontrola	Prikazuje ikonu na <i>taskbaru</i> koja predstavlja aplikaciju koja se izvodi u pozadini.
	ToolStrip kontrola	Kreira toolbar.

# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Korisnička pomoć (help)	HelpProvider komponenta	HelpProvider komponenta pruža pop-up ili online Help za kontrole.
	ToolTip komponenta	Prikazuje pop-up prozor sa kratkim opisom kontrole kada korisnik pozicionira kursor izna kontrole.
Audio	SoundPlayer kontrola	SoundPlayer kontrola svira zvučne datoteke u .wav formatu. Datoteke se mogu učitavati i svirati asinkrono.

# .NET 2.x kontrole

Funkcija	Kontrola	Opis
Grupiranje drugih kontrola	Panel kontrola	Panel kontrola grupira kontrole na neoznačenom okviru sa mogućnošću skroliranja.
	GroupBox kontrola	GroupBox kontrola grupira kontrole na označenom okviru bez mogućnosti skroliranja.
	TabControl kontrola	TabControl kontrola pruža tabulirane stranice za grupiranje objekata.
	SplitContainer kontrola	Pružna dva panela odvojena pokretnom trakom.
	TableLayoutPanel kontrola	Prikazuje panel koji dinamički poravnava sadržaj u mrežu sa redcima i kolonama.
	FlowLayoutPanel kontrola	Prikazuje panel koji dinamički poravnava sadržaj horizontalno ili vertikalno.

# C# tekstualne kontrole

- Osnovna funkcionalnost tekstualnih kontrola je omogućiti korisniku aplikacije unos/promjenu teksta.
- Osnovno svojstvo, s obzirom na funkcionalnost kontrole, je Text (tipa String). Ovo svojstvo sadrži tekst koji je unesen u kontrolu.
- [Primjer](#) dohvaćanja teksta iz kontrole



# C# list kontrola

- Osnovna funkcionalnost list kontrole je omogućiti korisniku aplikacije izbor jednog (ili više) elemenata od ponuđenih.
- Kada kontrola sadržava u sebi sadrži više drugih elemenata, kao što je slučaj sa ListBox kontrolom, ti su elementi pohranjeni u svojstvu kontrole koje može sadržavati kolekciju elemenata.
- Svojstva klase ListBox Items, SelectedItems, i SelectedIndices omogućavaju pristup trima kolekcijama koje se koriste u ovoj klasi:
  1. ListBox.ObjectCollection (System.Windows.Forms.ListBox.ObjectCollection) – sadrži sve elemente sadržane u ListBox kontroli.
  2. ListBox.SelectedObjectCollection - sadrži sve selektirane elemente u ListBox kontroli.
  3. ListBox.SelectedIndexCollection - sadrži indekse selektiranih elemenata u ListBox.SelectedObjectCollection.

# C# list kontrola

- Ova svojstva su implementacije tri sučelja, `IList`, `ICollection` i `IEnumerable`, koja sadrže metode potrebne za rad sa kolekcijama elemenata.
- Ova su sučelja definirana u imenskom prostoru `System.Collections` koji sadrži i sučelja i klase za različite kolekcija objekata (lista, red, stog, hash tablice i sl.).

# C# list kontrola

- Dodavanje elementa u listu:  
`listBox1.Items.Add(textBox1.Text);`
- Brisanje čitavog sadržaja liste:  
`listBox1.Items.Clear();`
- Dohvaćanje selektiranih elemenata:  

```
ListBox.SelectedIndexCollection svi = listBox1.SelectedIndices;  
for(int i=0; i<svi.Count;i++)  
{  
    listBox2.Items.Add(listBox1.Items[svi[i]]);  
}
```
- Omogućavanje korisniku selektiranje više elemenata odjednom radi se postavljanjem svojstva ListBox kontrole `SelectionMode` na `MultiSimple` ili `MultiExtended`.

# C# kombo kontrola

- Dodavanje elementa u kombo kontrolu:  
`comboBox1.Items.Add(textBox1.Text);`
- Brisanje čitavog sadržaja kombo kontrolu :  
`comboBox1.Items.Clear();`
- Dohvaćanje selektiranog elemenata:  
`String jedan = (String) comboBox1.SelectedItem;`  
`listBox2.Items.Add(jedan);`

# C# TreeView kontrola

- TreeView kontrola se koristi za prikaz kolekcije elemenata u hijerarhijskom obliku, svaki element je objekta tipa klase TreeNode.
- Elementi su u TreeView kontroli sadržani u svojstvu Nodes koji je kolekcija TreeNode objekata.
- Svaki element TreeNode može opet biti kolekcija TreeNode elemenata tako da hijerarhijski TreeView kontrola može ići proizvoljno duboko.
- Čvorovi mogu pored labele sadržavati i sliku, smještenu u ImageList svojstvu.
- Primjer dohvaćanja direktorija u TreeView kontrolu →

# C# TreeView kontrola

```
private void button5_Click(object sender, EventArgs e)
{
    String disk = "E:\\";
    treeView1.Nodes.Clear();
    treeView1.Nodes.Add(disk);
    napuni(disk, treeView1.Nodes[0]);
}
public void napuni(string disk, TreeNode parentNode)
{
    string[] directoryArray = Directory.GetDirectories(disk);
    try
    {
        if (directoryArray.Length != 0)
        {
            foreach (string directory in directoryArray)
            {
                substringDirectory = directory.Substring(directory.LastIndexOf("\\") + 1, directory.Length - directory.LastIndexOf("\\") - 1);
                TreeNode myNode = new TreeNode(substringDirectory);
                parentNode.Nodes.Add(myNode);
                napuni(directory, myNode);
            }
        }
    }
    catch (UnauthorizedAccessException)
    {
        parentNode.Nodes.Add("Access denied");
    }
}
```

# C# TreeView kontrola

- Dodavanje slika se radi preko ImageList svojstva na TreeView kontroli.
- Sve TreeNode elementi koji se dodaju na kontrolu po defaultu postavljaju prvu sliku iz ImageList svojstva iz TreeView kontrole na koju se smještaju , što se može promijeniti postavljanjem ImageIndex svojstva na TreeNode elementu.

- Primjer:

```
ImageList myImageList = new ImageList();  
myImageList.Images.Add(Image.FromFile("Image1.bmp"));  
myImageList.Images.Add(Image.FromFile("Image2.bmp"));  
treeView1.ImageList = myImageList;
```

# NotifyIcon kontrola

- Kontrola NotifyIcon prikazuje ikonu na *taskbaru* koja predstavlja aplikaciju.
- Svojstvo Icon se koristi za definiranje ikone, ako nije postavljeno ikona se neće prikazati na *taskbaru*.
- Kontrola se dodaje na glavnu formu aplikacije.
- Svojstvo ContextMenuStrip koristi se za dodavanje ContextMenuStrip izbornika na kontrolu.



# C# kontrole za formatiranje prikaza

- Klase za formatiranje (*layout*) prikaza omogućavaju automatsko formatiranje prikaza.
- Npr. `TableLayoutPanel` kontrola omogućava definiranje redaka i stupaca za prikazivanje kontrola kao u tablici. Poravnanje sadržaja kontrole radi se pri razvoju aplikacije (design time) i prilikom izvođenja aplikacije (run time). Kontrola se dinamički prilagođava stanju aplikacije, npr. promjenama veličine prozora.
- Na kontrolu se može smjestiti bilo koja `Windows.Forms` kontrola, uključujući i još jednu instancu `TableLayoutPanel` kontrole.

# C# ToolTip kontrola (komponenta)

- ToolTip kontrola prikazuje prozor sa kratkim opisom kontrole kada korisnik pomakne miša iznad kontrole. Koristi se za savjete korisniku o namjeni kontrole.
- ToolTip prozor se ne prikazuje kada je kontrola deaktivirana. Ukoliko je postavljeno svojstvo ShowAlways tada se ToolTip prozor uvijek prikazuje.
- Za sve kontrole na formi se obično koristi samo jedna ToolTip kontrola.
- ToolTip se povezuje sa kontrolama pozivom SetToolTip metode na ToolTip komponenti. Npr.  
`this.toolTip1.SetToolTip(this.button1, "Ovo je botun");`  
odnosno preko grafičkog sučelja postavlja se preko svojstva ToolTip on ToolTip1

# C# kontrole

- Kada se želi omogućiti formi, koja sadrži kontrole, prihvaćanje ulaza sa tipkovnice koristi se svojstvo KeyPreview iz klase Form.
- Svojstvo KeyPreview dohvaća i postavlja vrijednost koja određuje hoće li forma dobivati događaje sa tipkovnice prije no što se događaji prosljede kontroli na formi koja ima fokus. Ukoliko je svojstvo false neće se prosljeđivati formi. Kada je svojstvo postavljeno u true forma će primiti događaje KeyPress, KeyDown i KeyUp.
- Nakon što forma obradi događaje tipkovnice oni se prosljeđuju kontroli koja ima fokus. Ukoliko se događaji tipkovnice žele obraditi samo u formi, bez prosljeđivanja kontroli u event handleru forme za događaje tipkovnice treba staviti sljedeći kod:

KeyPressArgs.Handled = true

[PRIMJER:](#)

# Modalni i nemodalni prozori u .NET-u

# C#

- Kreiranje modalne forme u C# radi se pozivom metode ShowDialog.
- PRIMJER:  
Form1 novaforma= new Form1();  
novaforma.ShowDialog(); // Modalna forma  
novaforma.Show(); // Nemodalna forma

# WINFX

- Kreiranje modalnog prozora u WINFX-u radi se pozivom metode ShowDialog.

- PRIMJER:

```
void prvafunkcija(object sender, RoutedEventArgs e)
{
    Window1 window = new Window1();
    window.ShowDialog(); // Modalni
}
void drugafunkcija(object sender, RoutedEventArgs e)
{
    Window1 window = new Window1();
    window.Show(); // Nemodalni
}
```