

Oblikovanje korisničkog sučelja i dijaloga

7/13

Osnovni pojmovi

❑ Sučelja aplikacije

- korisničko sučelje (user interface) – definira interakciju s krajnjim korisnikom
- sučelje sustava, sistemsko sučelje (system interface) – određuje način razmjene informacija s drugim sustavima

❑ Osnovni mehanizmi korisničkih sučelja

- navigacijski mehanizam (navigation mechanism) – osigurava način na koji korisnici određuju što žele napraviti
- ulazni mehanizam (input mechanism) – određuje način prihvata informacija
- izlazni mehanizam (output mechanism) – određuje pružanje informacija korisnicima ili drugim sustavima

❑ Načela oblikovanja korisničkog sučelja

- Raspored (Layout)
- Uvažavanje sadržaja (Content awareness)
- Estetika (Aesthetics)
- Iskustvo korisnika (User experience)
- Dosljednost, konzistentnost (Consistency)
- Lakoća korištenja, minimalni napor korisnika (Minimal user effort)

Raspored, organizacija sučelja

☐ Programska oprema mora imati standardan izgled zaslona.

- područje za navigaciju - izbornici, pregled podataka (najčešće na vrhu)
- područje za prikaz statusa obrade i podataka (najčešće na dnu)
- radno područje - rad s podacima (u sredini)

☐ Polja zaslonske maske moraju biti logički grupirana.

- npr. osobni podaci, podaci o adresama i kontaktima, podaci o članovima obitelji, podaci o školovanju i napredovanju
- raspoređivanje na različite zaslonske maske (forme, prozore)
- grupiranje spremnicima (container) - panelima, tabulatorima
- učahurivanje u vlastite korisničke kontrole (custom control)

☐ Poravnanje

- vertikalno i horizontalno poravnanje kontrola
- lijevo poravnanje polja za unos i (desno) poravnanje njihovih labela
- lijevo poravnanje nizova znakova, desno poravnanje brojeva

Uvažavanje sadržaja

- ☐ **U svakom trenutku mora biti vidljiva informacija o**
 - dijelu obrade (unos, izmjena ...),
 - vrsti prikazanih podataka (osoba, račun, ...),
 - količini podataka (zapis m od n),
 - mogućim akcijama (aktivne kontrole).

- ☐ **Elementi sučelja trebaju se odnositi na poslovne objekte a ne na tehničke aspekte aplikacije.**

- ☐ **Objekti sučelja trebaju oponašati izgled i ponašanje stvarnih objekata koje prikazuju (npr. dokumente)**

Estetika

☐ Preglednost podataka

- Izbjegavati sažimanje kontrola na uskom prostoru
- Minimizirati neiskorišten prostor
- Minimizirati prekrivanje (npr. u slučaju više prikazanih "prozora").

☐ Korišćenje pisma

- Koristiti najviše 4 različite veličine slova na ekranu (web šareniji)
- Koristiti najviše 3 vrste pisma (font) na jednom ekranu
- Koristiti kombinaciju velikih i malih slova (izbjegavati kapitalizaciju)

☐ Upotreba boja

- Koristiti najviše 4 različite boje na ekranu
- Paziti na kontrast – tamni tekst na svijetloj podlozi i obrnuto
- Bojanje neaktivnih kontrola npr. sivo bolje nego sakrivanje
- Ne koristiti plavu boju za tekst (teško se čita)

Prilagođenost korisniku

☐ Prilagođenost iskustvu korisnika

- lakoća učenja za korištenje početnika
- lakoća učenja za korištenje naprednih korisnika
- podrška naprednim korisnicima, npr. ubrzavajućim tipkama (accelerator key)

☐ Korisnike treba navoditi na posluživanje programa

- najčešće interaktivnom pomoći i uputama (uz dosljednost)
- manja potreba za uputama → intuitivnost → kvalitetnije sučelje

☐ Sučelje mora imati ujednačene standardne poruke

- Poruke moraju biti jednostavne, precizne te ovisne o kontekstu.
- Izbjegavati računalski žargon i kratice.
- Prevesti poruke na stranom jeziku (npr., iznimke OS i SUBP)

☐ Svakako ugraditi interaktivnu pomoć ovisnu o kontekstu.

Dosljednost korisničkog sučelja

☐ Dosljednost (konzistentnost) korisničkog sučelja

- kontrole istog tipa svuda jednakog izgleda i ponašanja
- standardne kontrole uvijek na uobičajenom/očekivanom mjestu (gumbi, status, ...)

☐ Najvažniji faktor za pojednostavnjivanje načina korištenja sustava

- omogućuje korisnicima predvidjeti što će se dogoditi
- nakon što svladaju interakciju s jednim dijelom sustava, znat će i s drugim dijelovima.
- skraćuje krivulju učenja

☐ Postavljanje i pridržavanje standarda (slično standardu kodiranja)

- izgled (veličina – omjer, oblik, boja),
- naslovi kontrola,
- ponašanje kontrola
- standardizacija tipki

Minimizacija napora

☐ **Pravilo tri klika (three clicks rule)**

- Treba omogućiti pristup od izbornika do podatka u najviše tri koraka (klika mišem ili tipkovničkih kombinacija)

☐ **Unos se mora obavljati u slijedu kojim su polja fizički poredana.**

- slijeva nadesno i
- postavljanjem *TabIndex* svojstva kontrole

☐ **Korištenje predviđenih (default) vrijednosti**

☐ **Predviđene (default) kontrole**

- Nedestruktivne - npr. "Ne" za brisanje


☐ **Ograničavanje selekcije listama**

- preddefinirani skupovi podataka
- vezani podaci (strani ključevi)


☐ **Ulančavanje procedura (primjer u nastavku)**

Vrste korisničkog sučelja

❑ Sučelje s pojedinačnim dokumentima - Single document interface (SDI)

- sučelje sastavljeno od nezavisno otvorenih i odvojenih formi, od kojih svaka sadrži po jedan "dokument"
- primjer:  ADO\ Artikl, Drzava, DokumentStavka

❑ Sučelje s više dokumenata - Multiple document interface (MDI)

- aplikacija se sastoji od jedne ili više formi sadržanih unutar zajedničkog glavnog prozora
- primjer:  FirmaWin (u nastavku)

❑ Sučelje s tabuliranim dokumentima - Tabbed document interface (TDI)

- različiti dokumenti sadržani unutar jednog prozora, na panelima Tab kontrole
- primjer: VS.NET razvojna okolina

Izbornici i dijalozi

❑ Sučeljem moraju biti podržane različite vrste izbornika

- horizontalni izbornik (menu bar) - uvijek vidljiv, lako dohvatljiv
- padajući (pull-down) i kaskadni - "nevidljivi", ali se opcije daju grupirati
- skočni (pop-up, brzi) - nije očigledno da postoje, skaču na različitim mjestima
- trake s ikonama (toolbar, toolbox, ribbon) - vidljivi, pamtljivi, uz mogući problem dinamičkog osposobljavanja i prikaza/skrivanja ikona

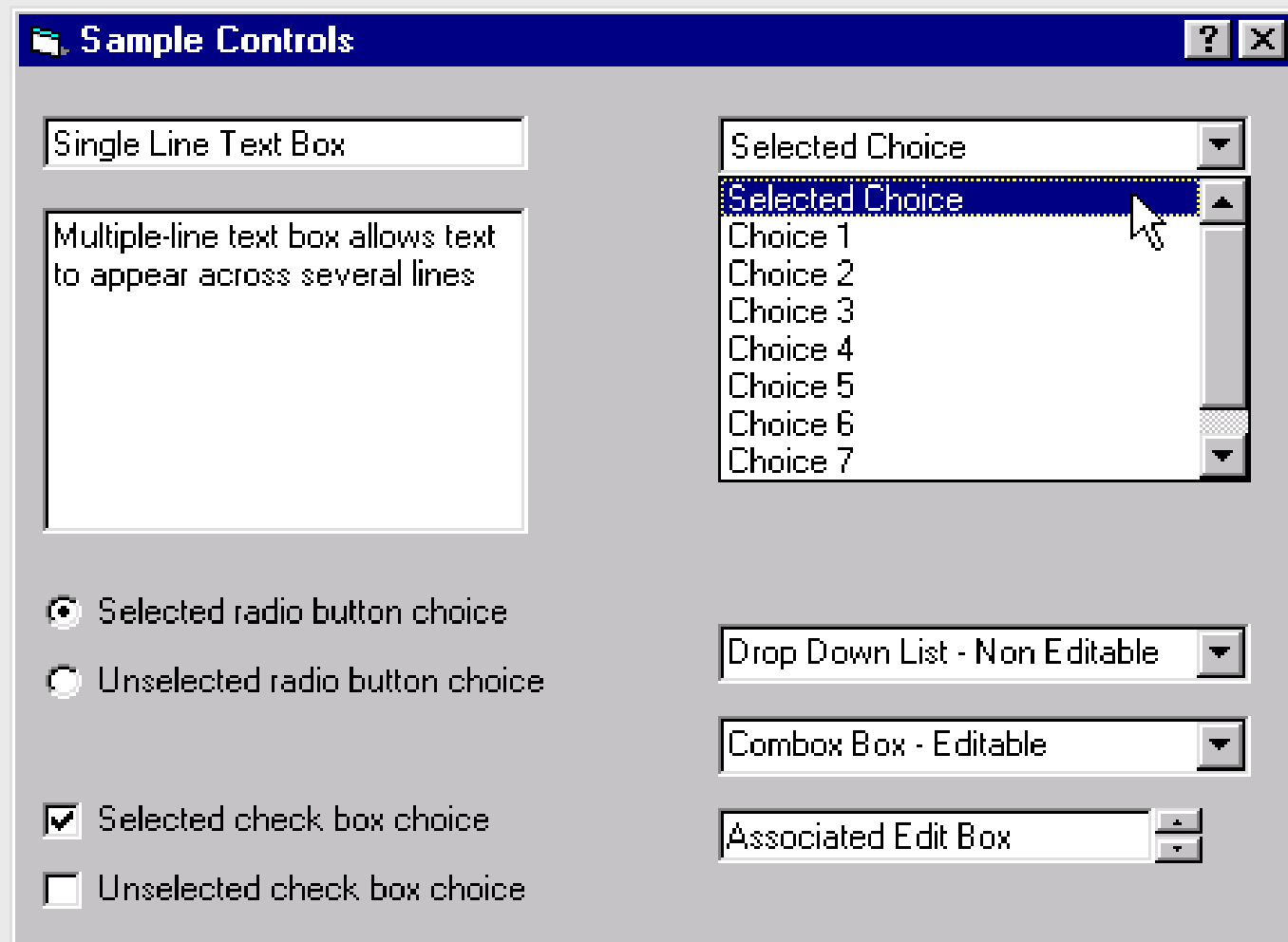
❑ Omogućiti brzi odabir

- ubrzavajućom/funkcijskom tipkom ili slovom opcije izbornika
 - primjer: Alt-Slovo kombinacija pisanjem "&" ispred željenog slova
- Tipke za obavljanje standardnih funkcija moraju biti definirane pažljivo i jednoznačno
 - primjer: F1-Pomoć, F2-Unos, F3-Izmjena, ESC-Opoziv)
 - Unaprijed treba predvidjeti i one za aktiviranje dodatnih funkcija.

❑ Maksimalno 7-12 opcija po izborniku

❑ Konzistentan, pozitivan tekst poruka, punim rečenicama

Elementi grafičkog sučelja za unos podataka



Kriterij za odabir kontrola grafičkog sučelja

☐ Text Box i varijante (npr. RichTextBox)

- unos slobodnog teksta
- tekst može biti maskiran (npr. telefonski broj, registarska oznaka)
- omogućiti unos više redaka i kliznike samo za stvarno višeretkovni tekst (npr. opis, komentar)
- pripaziti na duljinu teksta i tip podatka

☐ Check Box

- binarne vrijednosti, opcionalno "nepoznato"
- mali broj (do 3) unaprijed poznatih i vremenski nepromjenjivih oznaka
- za veći broj oznaka, koristiti listu (List Box) sa selekcijom više stavki

☐ Radio Button (ili Option Button)

- mali broj (do 3) unaprijed poznatih i vremenski nepromjenjivih oznaka
- vrijednosti se međusobno isključuju
- za veći broj oznaka, koristiti listu za odabir jedne vrijednosti (Combo Box)

Kriterij za odabir kontrola grafičkog sučelja

☐ Drop-Down Box (ili Combo Box)

- umjereno velik broj (do nekoliko stotina), međusobno isključivih vrijednosti
- do 20 istovremeno vidljivih vrijednosti pri odabiru
- u slučaju većeg broja zapisa koristiti editabilni *combo* za unos filtra ili kontrolu nadomjestiti posebnom formom za filtriranje i odabir (tzv. lookup)

☐ List Box

- umjereno velik broj (nekoliko desetaka), ne nužno isključivih vrijednosti
- istovremeni prikaz 8-10 vrijednosti
- u slučaju većeg broja zapisa ili zapisa složenije strukture nadomjestiti rešetkom (Grid)

☐ NumericUpDown

- nevelik slijed (nekoliko desetaka) diskretnih vrijednosti
- u slučaju većeg broja vrijednosti, omogućiti izravan ili nadomjestiti listom

☐ Grid – rešetka, mreža, matrica

- kombinacije osnovnih elemenata
- koristiti za prikaz zapisa s malim brojem atributa da se vidi cijeli zapis odjednom

Primjer sučelja poslužiteljske aplikacije

kfertalj (ansi) | PIS - PERSONALNI INFORMACIJSKI SUSTAV | (Sri) 04.12.96 |

OSOBA: Dohv Sljed Preth Unos Iznj Ostalo Lista Rasp Zap ...

Postavljanje uvjeta za dohvrat zapisa

===== (LALIĆ MARIJAN) ===== [(2/3)] ===== 120/261 =====

Sprema (71) (Visoka VII/1)

Br.svjed. (kfertalj (ansi) | PIS - PERSONALNI INFORMACIJSKI SUSTAV | (Čet) 05.12.96 |

Zanimanje (5)

Stamb.stanje (3) ESC-Prihvat
CtrC-Prekid F10-Pomoć

Krvna grupa (= unos uvjeta ===== OSOBA ===== [(1/3)] =====

Zdrav.broj (ID () Prezime (*IĆ) Roditelj ()

Spol () Ime (MARI*) (čiji/čija) ()

Izlazak () JMBG () Br.osobnika ()

Ured za obranu

Osobni VPD

Početak voj.roka Kategorija osobe () ()

Vojska voj.roka Bračno stanje () () Djece ()

Razl.prest.voj.roka Mjesto rođenja () () Dat.rođenja ()

Sudjel. u dom.ratu Općina () ()

Država () ()

Narodnost () ()

Vjera () ()

Državljanstvo 1 ()

Državljanstvo 2 ()

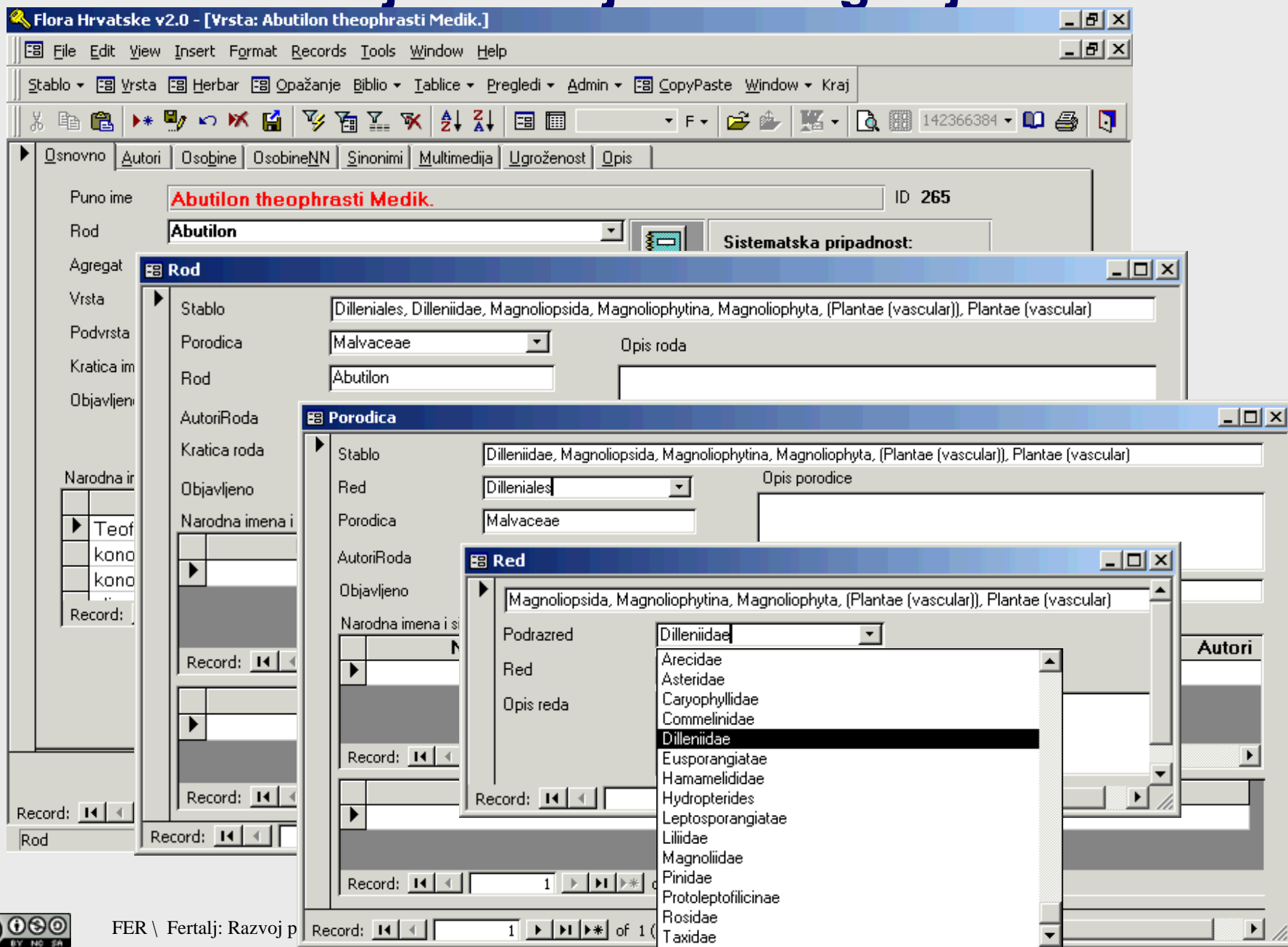
Prebivalište () (ZAGREB) ()

Mjesto stanovanja () () ()

Tel. ()

Naziv mjesta

Primjer sučelja debelog klijenta

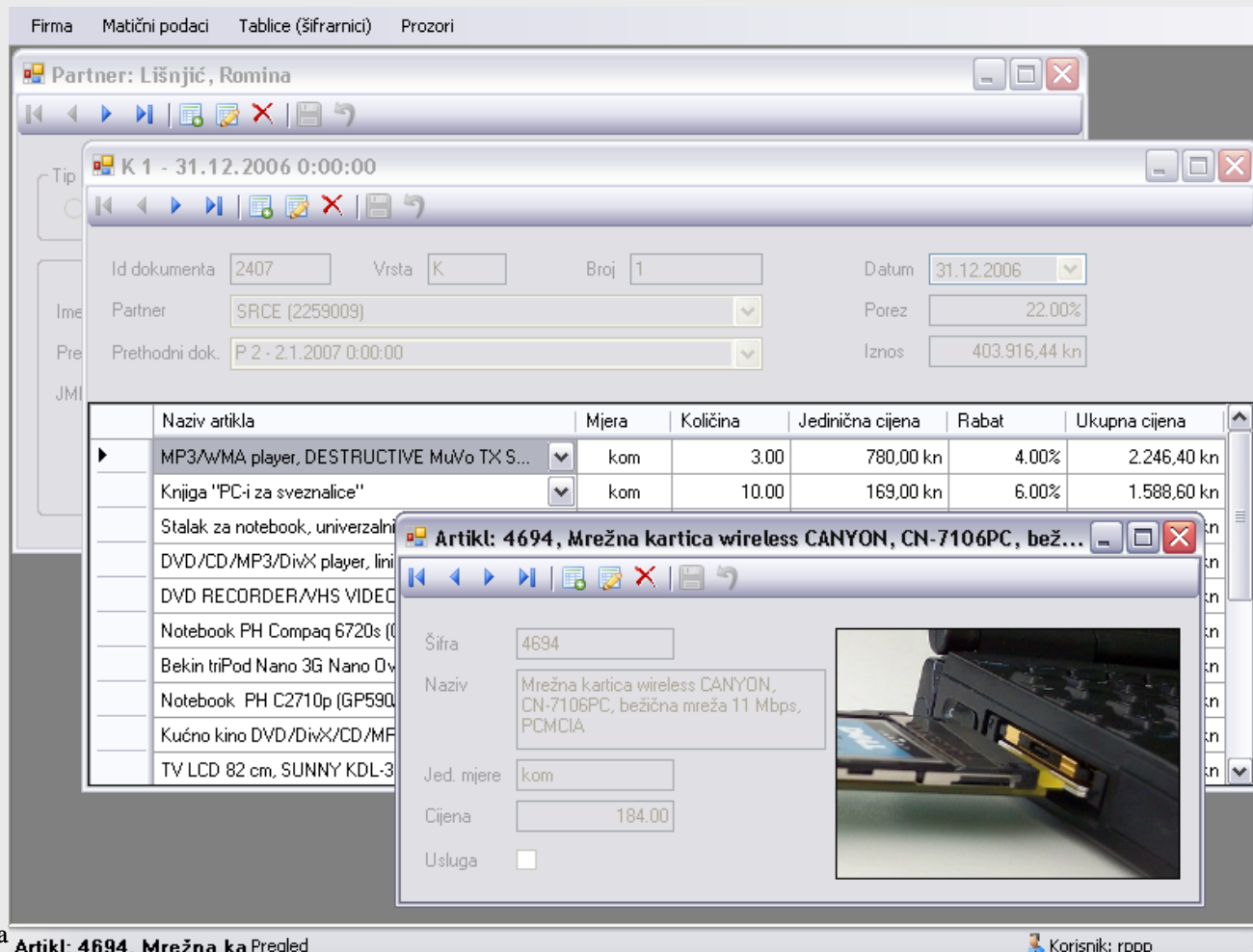


Primjer sučelja aplikacije

MDI korisničko sučelje

❑ MDI (Multiple Document Interface) aplikacija

- aplikacija se sastoji od formi sadržanih unutar zajedničkog glavnog prozora
- izuzetak čine modalne forme koje mogu biti otvorene povrh glavne forme



Forma roditelj i forme djece

❑ Roditelj (*MdiParent*) – glavna forma koja sadrži druge forme

- Svojstvo `IsMdiContainer` postavljeno na *true*
- `MdiChildren` – kolekcija formi djece
- `LayoutMdi` – raspored djece unutar roditeljske forme

❑ Dijete (*MdiChild*) – forma sadržana u formi roditelj

- postavlja se `MdiParent` na roditeljsku formu prije prikazivanja forme

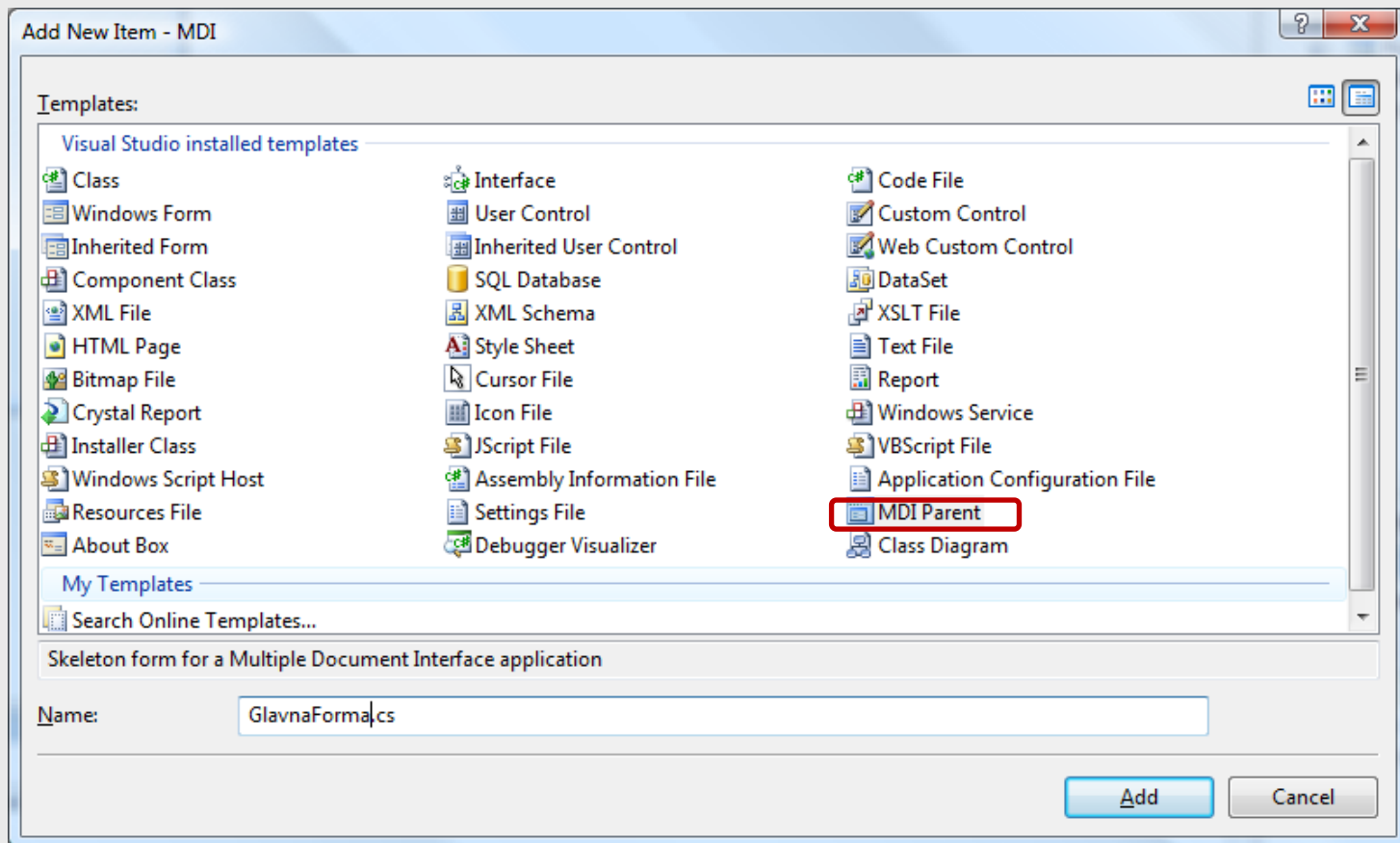
❑ Primjer: FirmaWin

```
private void artiklToolStripMenuItem_Click(object sender, EventArgs e)
{
    ArtiklForm f = new ArtiklForm();
    f.MdiParent = this;
    f.Show();
}
```

Kreiranje MDI roditelja

❑ Formu roditelja je moguće automatski generirati

- Dodavanjem nove datoteke *MDI parent* u projekt
- Stvaraju se izbornik i alatna traka sa standardnim mogućnostima (*open, save, copy, cut, exit, ...*)



Raspored djece unutar roditeljske forme

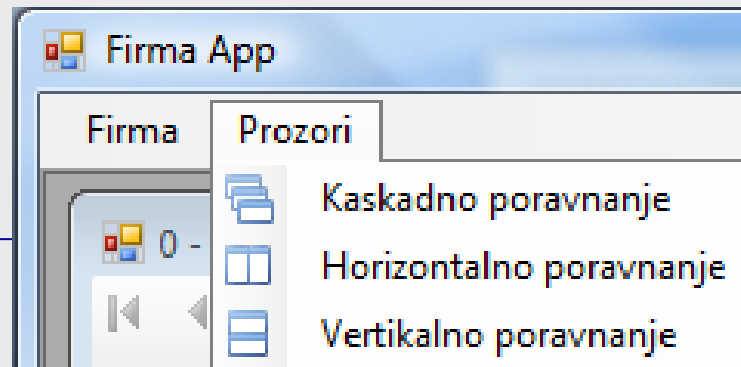
❑ Postupak `LayoutMdi` na roditeljskoj formi, s argumentom

■ enumerator `MdiLayout`

- `Cascade` – kaskadni raspored unutar roditeljske forme
- `TileVertical` – okomito poravnanje unutar roditeljske forme
- `TileHorizontal` – vodoravno poravnanje unutar roditeljske forme
- `ArrangeIcons` – poravnanje ikona formi (minimiziranih formi)

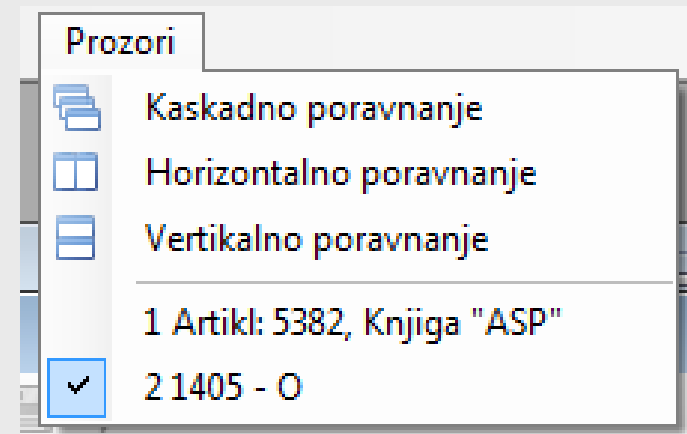
❑ Primjer: **FirmaWin**

```
private void horizontalnoPoravnanjeToolStripMenuItem_Click(  
    object sender, EventArgs e)  
{  
    LayoutMdi (MdiLayout.TileHorizontal) ;  
}
```



Prikaz aktivnih formi u izborniku

- ❑ Dorada opcije izbornika *Prozor* horizontalnim separatorom i listom otvorenih formi
 - Klik na naziv forme aktualizira odgovarajuću formu
 - Zatvaranje forme uklanja njezin naziv iz liste
- ❑ `this.MainMenuStrip` – referenca glavne izborničke trake forme
- ❑ Svojstvo `MdiWindowListItem` kontrole `MenuStrip` postavlja se na element izbornika u kojem želimo prikazati aktivne forme
 - `menuStrip1.MdiWindowListItem = prozoriToolStripMenuItem;`
 - Potrebno je ukloniti ručno (u kodu) separator ukoliko nema otvorenih formi
 - `menuStrip1.MdiWindowListItem.DropDownOpening += MdiWindowListItem_DropDownOpening;`



Spajanje izbornika glavne forme i forme djeteta

❑ Glavna forma može udružiti izbornik djeteta u zajednički izbornik

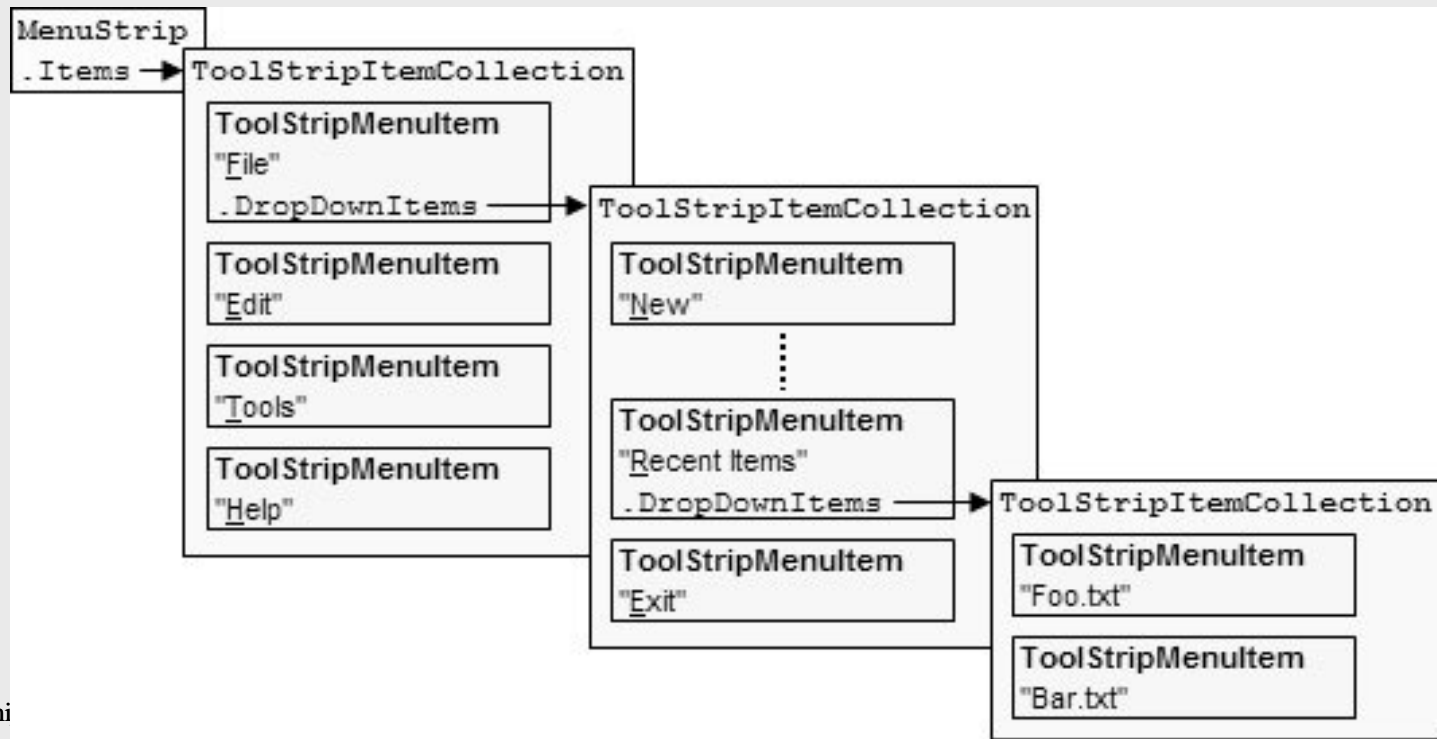
- Obavlja se prilikom maksimiziranja forme djeteta unutar glavne forme
- Svojstvo glavne forme `MainMenuStrip` postavlja se na izbornik forme
- `this.MainMenuStrip = this.menuStrip1;`

❑ Spajanje pojedinih elemenata izbornika

- Svojstvo `MergeAction` elementa izbornika *ToolStripMenuItem*
 - Način na koji se izbornici spajaju
 - `enum MergeAction {Append, Insert, Replace, Remove, MatchOnly}`
- Svojstvo `MergeIndex` elemenata izbornika *ToolStripMenuItem*
 - Pozicija elementa u zajedničkom izborniku

Izbornik

- ❑ ***MenuStrip*** - kontrola koja predstavlja korijen sustava izbornika
 - Sadrži kolekciju *Items* tipa *ToolStripItemCollection*
- ❑ **Element izbornika može biti tipa (desni klik – Convert to)**
 - *ToolStripMenuItem* – element za odabir, podrazumijevano se dodaje
 - može sadržavati podelemente (svojstvo *DropDownItems*)
 - *ToolStripComboBox* – padajuća lista
 - *ToolStripTextBox* – tekst



Izbornik zavisan o kontekstu

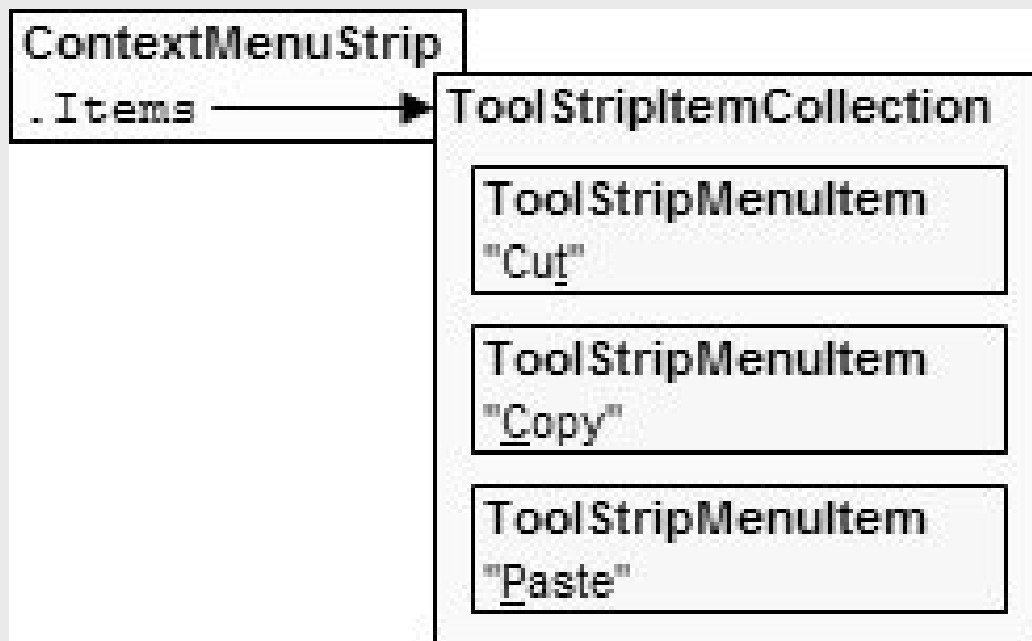
❑ **ContextMenuStrip**

- izbornik zavisan o kontekstu (brzi izbornik, skočni izbornik, izbornik prečice)
- sadrži kolekciju *Items* tipa *ToolStripItemCollection*
- može imati hijerarhiju opcija ali nema horizontalnu početnu komponentu

❑ **Svaka forma i kontrola može imati najviše jedan izbornik zavisan o kontekstu**

- *ContextMenuStrip* svojstvo forme ili kontrole

❑ **Primjer** 📁 (kasnije)



Statusna traka

❑ Kontrola **StatusStrip** – predstavlja statusnu traku (*status bar*)

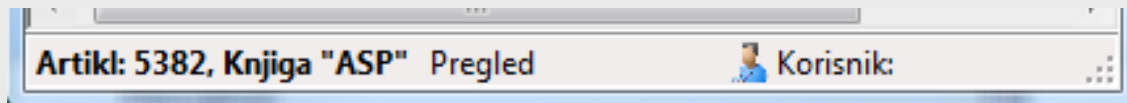
❑ Može sadržavati različite kontrole (desni klik – Items)

- *ToolStripProgressBar*

- *ToolStripStatusLabel*

- Prikaz teksta, slike ili oboje

- ...



❑ Svojstva **StatusStrip**

- *SizingGrip* – prikaz oznake mogućeg razvlačenja u donjem desnom kutu

- *SizeGripStyle* – vrsta *SizingGrip* oznake (Auto, Show, Hide)

❑ Svojstva kontrole **ToolStripStatusLabel**

- *Image*, *ImageAlign* – prikaz slike i položaj slike unutar statusne trake

- *Text* – prikaz teksta

- *DisplayStyle* – prikaz slike/teksta ili oboje

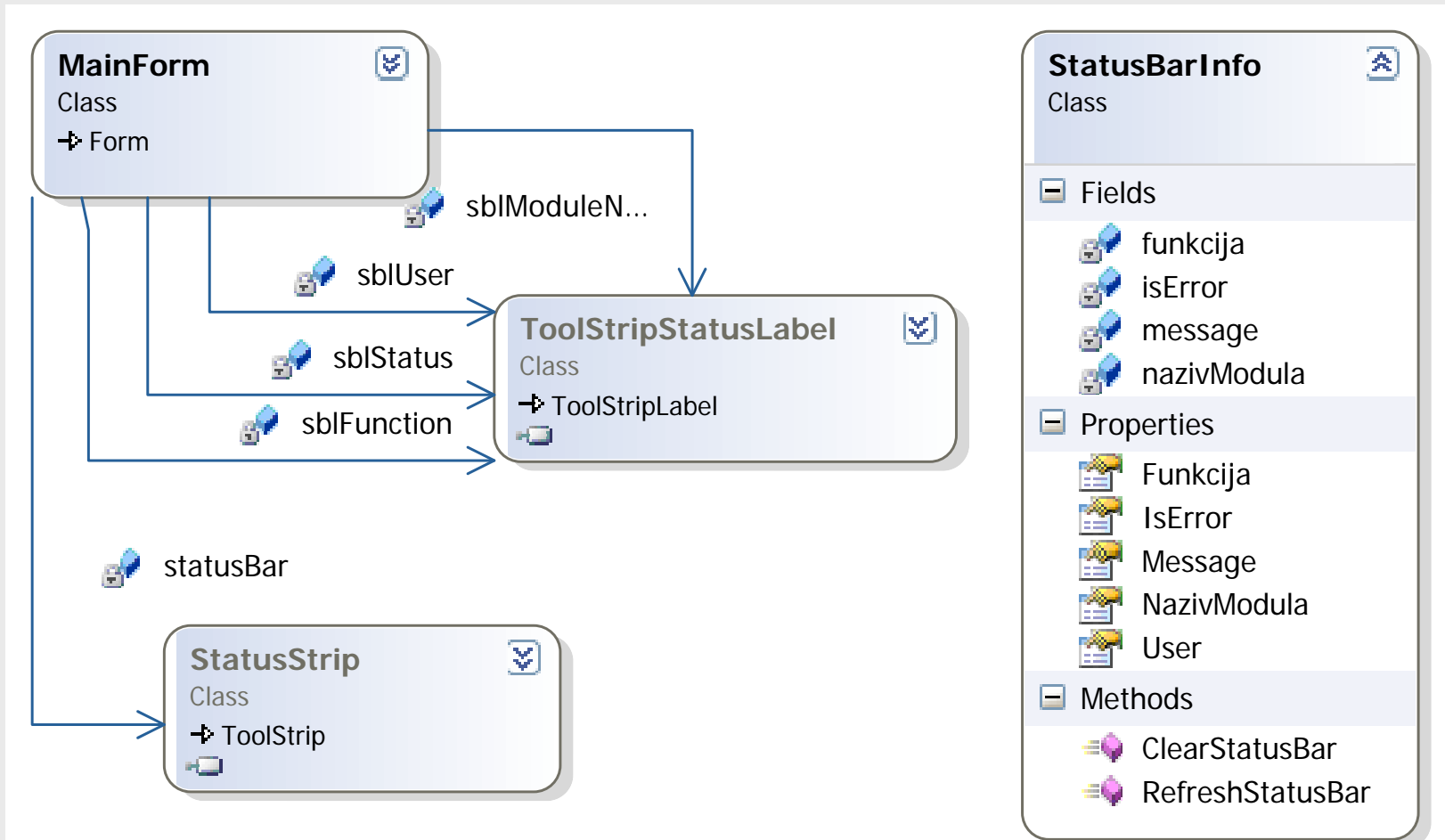
- *Visible* – zastavica vidljivosti

- *Spring* – specificira popunjava li kontrola statusnu liniju koliko je moguće (*=true*) ili samo unutar zadane širine

Primjer statusne trake

❑ Primjer Firma – MainForm

- statusBar – statusna traka koja sadrži labele
- ToolStripStatusLabel sbIModuleName, sbIUser, sbIStatus;



Prikaz elemenata statusne trake

- ❑ **Za prikaz pojedine labele koristi se sadržaj instance vlastitog razreda `StatusBarInfo` (Firma \Core – `StatusBarInfo`)**
 - svojstva `Message`, `NazivModula`, `IsError`, `User`, ...
 - postupak `ClearStatusBar` – inicijalizira statusnu traku
 - postupak `RefreshStatusBar` – osvježava traku navedenim sadržajem

- ❑ **Postupak `StatusBarInfo.RefreshStatusBar`**
 - poziva `SetStatusBar` postupak razreda `MainForm` koji postavlja vrijednosti statusne trake

```
public void RefreshStatusBar() {  
    if (MainForm.Instance != null)  
        MainForm.Instance.SetStatusBar(this);  
}
```

Postavljanje elemenata statusne trake

❑ Primjer: FirmaWin - MainForm

- osvježavanje statusa temeljem predane informacije

```
public void SetStatusBar(StatusBarInfo info)
{
    if (info != null)
    {
        sblModuleName.Text = info.NazivModula;
        sblFunction.Text = info.Funkcija;
        sblStatus.Text = info.Message;
        sblUser.Text = info.User;
    }
}
```

❑ Primjer: FirmaWin - BaseForm

- informiranje o kontekstu obrade pri dodavanju/izmjeni/brisanju

```
public virtual void New()
{
    ...
    StatusBar.Funkcija = "Novi unos";
    StatusBar.Message = string.Empty;
    StatusBar.IsError = false;
}
```

Prikaz informacije o aktualnom zapisu

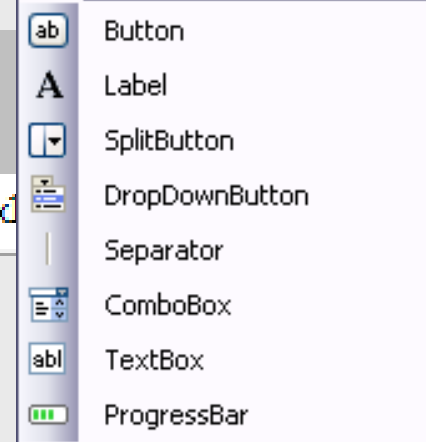
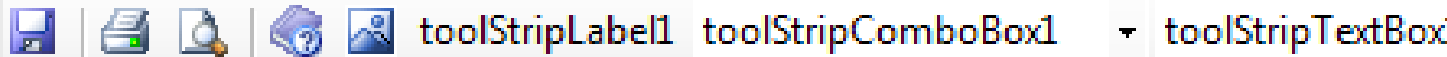
❑ Postavljanje naslova forme kad se promijeni aktualni artikl

```
private void artiklBindingSource_CurrentItemChanged(  
    object sender, EventArgs e)  
{  
    this.Text = "Artikl";  
    if (artiklBindingSource.Current != null)  
    {  
        this.Text = "Artikl: "  
            + ((Artikl)artiklBindingSource.Current).ToString();  
    }  
}
```

❑ Izazove promjenu statusne trake

```
private void BaseForm_TextChanged(object sender, EventArgs e)  
{  
    ...  
    StatusBar.NazivModula = this.Text;  
}
```

Alatna traka



❑ **ToolStrip** – predstavlja alatnu traku (*toolbar*)

- Osnovica za *MenuStrip*, *ContextMenuStrip* i *StatusStrip*

❑ **Može sadržavati različite kontrole (izvedene iz *ToolStripItem*):**

- *ToolStripComboBox* , *ToolStripTextBox* , *ToolStripButton* , ...

❑ **Svojstva**

- `Appearance: Flat, Normal` (enum `ToolBarAppearance`)
- `Divider` – traka crtom odijeljena od ostalog dijela forme
- `TextAlign` – smještaj i vidljivost teksta (enum `ToolBarTextAlign`)
- `Wrappable` – raspored `ToolBarButton` kontrola
- `Buttons`: lista `ToolStripButton` kontrola

❑ **Događaji:**

- `ButtonClick`, `ButtonDropDown`, ...

❑ **Svojstva *ToolStripButton* kontrole**

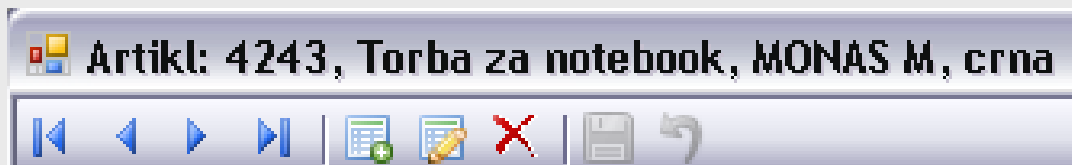
- `Enabled`, `Text`, `ToolTipText`, ...
- `DisplayStyle` (enum `ToolStripItemDisplayStyle`)

Vlastita alatna traka

❑ Potreba za vlastitom alatnom trakom

- Standardna alatna traka (*ToolStrip*) uvijek je u stanju za uređivanje (*Edit*)
- Vlastita alatna traka – Dodajemo gumb za uređivanje (ikona olovke)
- U ovisnosti o stanju (uređivanje ili čitanje) prikazujemo ostale gumbe

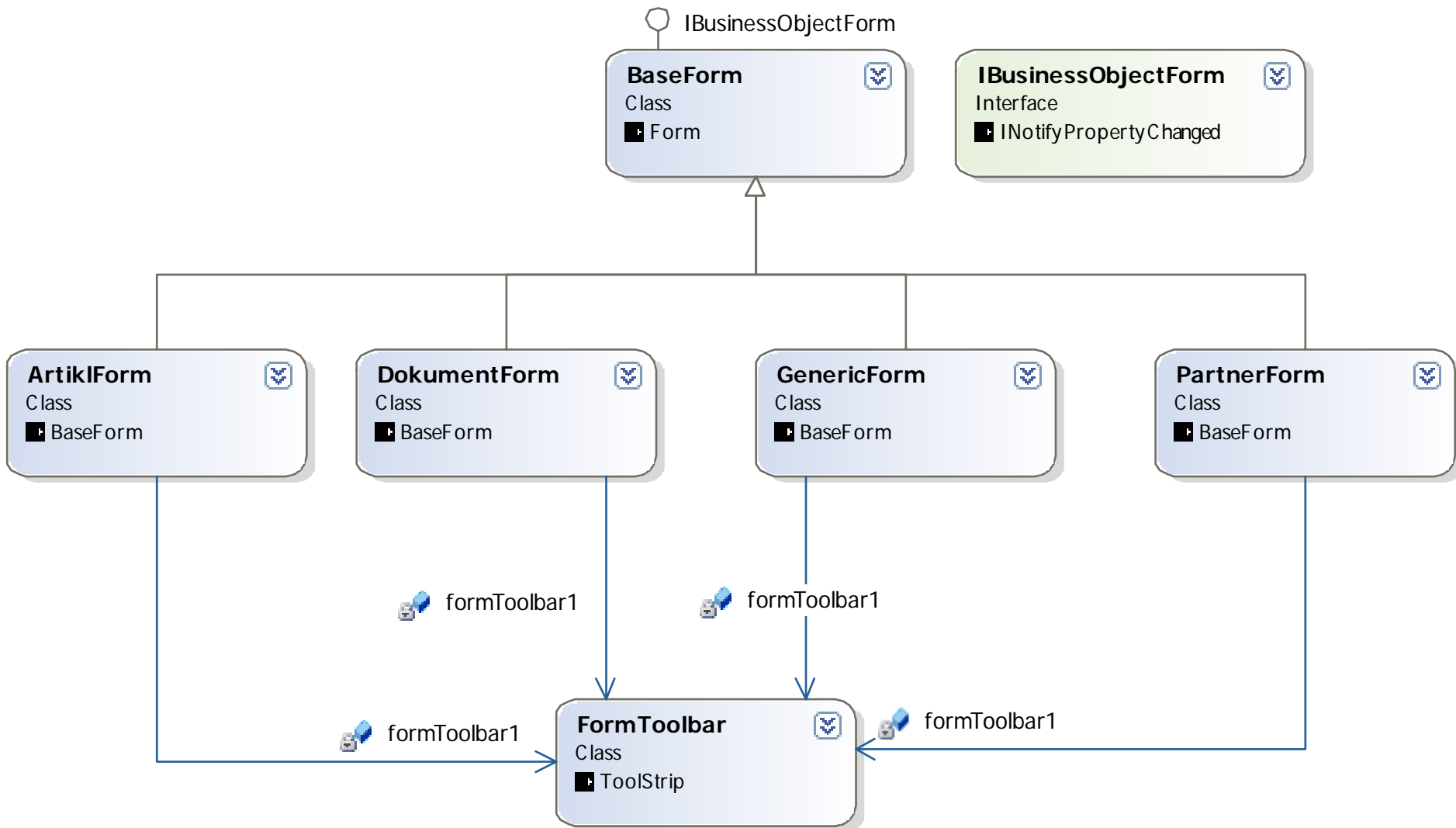
❑ Primjer: FirmaWin – FormToolbar



❑ Vlastita alatna traka

- Izvedena je iz razreda *ToolStrip*
 - `public partial class FormToolbar : ToolStrip`
- Sadrži formu `f` s kojom komunicira (forma na kojoj alatna traka prikazuje)
 - `private IBusinessObjectForm f;`
- Postupak `InitButtons` – postavlja rukovatelje događajem klik na gumbe
- Postupak `RefreshToolbar` – osvježava alatnu traku (prikazuje trenutno aktualne elemente alatne trake)

Organizacija formi s alatnom trakom



Dinamička prilagodba alatne trake

❑ Postupak InitButtons i obrada događaja

```
private void InitButtons() {  
    btnFirst.Click += new EventHandler(btnFirst_Click);  
    ...  
}  
  
private void btnFirst_Click(object sender, EventArgs e)  
{  
    if (f != null)  
        f.First();  
}
```

❑ RefreshToolbar – Prilagodba alatne trake











```
public void RefreshToolbar()  
{  
    this.Enabled = f != null;  
    btnFirst.Enabled = f != null ? f.CanDoFirst : false;  
    btnPrevious.Enabled = f != null ? f.CanDoPrevious : false;  
    ...  
}
```

IBusinessObjectForm










Interface

→ INotifyPropertyChanged

Properties

-  CanDoCancelChanges
-  CanDoDelete
-  CanDoEdit
-  CanDoFirst
-  CanDoLast
-  CanDoNew
-  CanDoNext
-  CanDoPrevious
-  CanDoSaveChanges
-  InEditMode

Methods

-  CancelChanges
-  Delete
-  Edit
-  First
-  Last
-  New
-  Next
-  Previous
-  SaveChanges

Events

-  NeedToolbarRefresh

Događaji pri unosu s tipkovnice

❑ Događaji pridruženi kontrolama i formi, obavljaju se redom

- **KeyDown**
 - *Delegate KeyEventHandler, event arguments KeyEventArgs*
 - pritisnuta je (bilo koja) tipka
- **KeyPress**
 - *Delegate KeyPressEventHandler, event arguments KeyPressEventArgs*
 - pritisnuta je tipka koja predstavlja ASCII znak
- **KeyUp**
 - *Delegate KeyEventHandler, event arguments KeyEventArgs*
 - otpuštena je prethodno pritisnuta tipka

❑ Enumerator **Keys** – nabraja oznake tipki

- `enum Keys { Home, PageUp, PageDown, Left, Right, A, B, C, F1, F2, ...}`

❑ Unos podataka s tipkovnice može biti preduhitren prije nego li ga obradi kontrola u koju se tipka

- Formi se postavlja svojstvo `KeyPreview=true`
- primjer `MainForm_KeyDown`

Argumenti rukovatelja događaja tipkovnice

❑ Svojstva razreda *EventArgs*

- `Alt` – pritisnuta tipka Alt
- `Control` – pritisnuta tipka Control
- `Shift` – pritisnuta tipka Shift
- `Handled` – *bool* oznaka da je događaj obrađen (uklanja "eho")
- `KeyCode` – oznaka tipke kao vrijednost enumeratora `Keys` (npr. `K`)
- `KeyData` – oznaka tipke s informacijom o modifikatoru (npr. `K`, `Control`)
- `KeyValue` – brojučana oznaka tipke (npr. 75)
- `Modifiers` – vrijednost `Keys` za pritisnuti modifikator (npr. `Control`)

❑ Svojstva razreda *KeyPressEventArgs*

- `KeyChar` – pritisnuti osnovni znak (npr. za `SHIFT+K` ima vrijednost veliko `K`)
- `Handled` – *bool* oznaka da je događaj obrađen

Obrada događaja pri unosu s tipkovnice

❑ Primjer: FirmaWin – BaseForm

- Obrada *KeyDown* događaja (*BaseForm*), pozivom odgovarajućih postupaka

```
private void BaseForm_KeyDown(object sender, KeyEventArgs e) {  
    e.Handled = true;  
    switch (e.KeyCode) {  
        case Keys.Home: First(); break;  
        case Keys.PageUp: Previous(); break;  
        case Keys.PageDown: Next(); break;  
        case Keys.End: Last(); break;  
        case Keys.F1: ShowHelp(); break;  
        case Keys.F2: New(); break;  
        case Keys.F3: Edit(); break;  
        case Keys.F4: e.Handled = false; break;  
        case Keys.F5: Zoom(); break;  
        case Keys.F6: EditDetails(); break;  
        case Keys.F7: Delete(); break;  
        case Keys.F10: // pohrana, povrat odabranog zapisa  
        ...  
    }
```

Kontrola *DataGridView*

❑ Mrežna/tablična obrada podataka s različitih vrsta izvora

- Liste (*IList*), skupovi podataka (*DataSet*, *DataTable*), *bindingsource*

❑ Svojstva *DataSource* i *DataMember*

- *DataSource* – izvor s kojeg se prikazuju podaci
- *DataMember* – naziv liste ili tablice izvora podataka koji se prikazuju

❑ Primjer: FirmaWin – DokumentForm

	Naziv artikla	Mjera	Količina	Jedinična cijena	Rabat	Ukupna cijena
▶	MP3/wMA player, DESTRUCT... ▼	kom	3.00	780,00 kn	4.00%	2.246,40 kn
	Knjiga "PC-i za sveznalice" ▼	kom	10.00	169,00 kn	6.00%	1.588,60 kn
	Stalak za notebook, univerzalni,... ▼	kom	5.00	380,00 kn	9.00%	1.729,00 kn
	DVD/CD/MP3/DivX player, linij... ▼	kom	8.00	549,00 kn	10.00%	3.952,80 kn
	DVD RECORDER/VHS VIDEO... ▼	kom	5.00	2.049,00 kn	4.00%	9.835,20 kn
	Notebook PH Compaq 6720s (... ▼	kom	1.00	4.553,00 kn	3.00%	4.416,41 kn
	Bekin triPod Nano 3G Nano Ov... ▼	kom	1.00	73,00 kn	1.00%	72,27 kn
	Notebook PH C2710p (GP590... ▼	kom	3.00	12.514,00 kn	8.00%	34.538,64 kn
	Kućno kino DVD/DivX/CD/MP... ▼	kom	15.00	2.069,00 kn	10.00%	27.931,50 kn

K 1 – 31.12.2006 0:00:00 Pregled



Korisnik: rppp

Članovi *DataGridView*

□ Svojstva

- Columns – kolekcija stupaca (*DataGridViewColumnCollection*)
 - SortMode svojstvo stupca – postavljanje načina sortiranja prilikom klika na zaglavlje (npr. *DataGridViewColumnSortMode.NotSortable*)
- Rows – kolekcija redaka (*DataGridViewRowCollection*)
- SelectedCells – kolekcija označenih ćelija (*DataGridViewSelectedCellCollection*)
- ColumnHeadersVisible, RowHeadersVisible – vidljivost zaglavlja
- CurrentCell, CurrentRow – trenutno aktivna ćelija/redak
- DefaultCellStyle – definira izgled ćelije
- ReadOnly – oznaka da je dozvoljeno uređivanje

□ Indeksir

- [rowIndex, columnIndex] – dohvat ćelije u retku indeksa rowIndex i stupcu indeksa columnIndex

□ Postupci

- BeginEdit, EndEdit – započimanje / završetak promjena

Događaji i stupci *DataGridView*

□ Događaji

- `CurrentCellChanged` – promjena fokusa aktivne ćelie
- `CellBeginEdit`, `CellEndEdit` – okida se pri započinjanju/završavanju promjena u ćeliji
- `CellValueChanged` – okida se nakon završetka promjena unutar ćelije
 - argument `DataGridViewCellEventArgs` – podaci (svojstva `ColumnIndex`, `RowIndex`)

□ Stupac *DataGridView* kontrole (iz *DataGridViewColumn*) može biti

- Tekstovni element (*`DataGridViewTextBoxColumn`*)
- Padajući izbornik (*`DataGridViewComboBoxColumn`*)
- Slika (*`DataGridViewImageColumn`*)
- ...

□ Neka svojstva elemenata *DataGridViewColumn*

- `HeaderText`, `ReadOnly`, `Visible`, `DefaultCellStyle`, ...
- Svojstva elementa *`DataGridViewComboBoxColumn`*
 - `DataSource`, `DisplayMember`, `ValueMember`, ...

Primjer obrade događaja kontrole *DataGridView*

❑ Primjer: FirmaWin – DokumentForm

- Obrada događaja `CellValueChanged`
- Pri promjeni artikla unutar stavke mijenja se i cijena stavke

```
private void stavkaDataGridView_CellValueChanged(object sender,
    DataGridViewCellEventArgs e) {
    if (e.RowIndex == -1) return;
    if (e.ColumnIndex == 0 || e.ColumnIndex == 1) {
        object sifArtikla =
            stavkaDataGridView[e.ColumnIndex, e.RowIndex].Value;
        if (sifArtikla != null) {
            Artikal a = artiklBll.Fetch((int)sifArtikla);
            if (a != null) {
                Stavka s = stavkaBindingSource.Current as Stavka;
                s.JedCijArtikla = a.CijArtikla;
                s.JedMjereArtikla = a.JedMjere;
            }
        }
    }
}
```


Izgled ćelije – *DataGridViewCellStyle*

❑ *DataGridViewCellStyle* – izgled ćelije *DataGridView* kontrole

- Može se primijeniti i na određeni stupac *DataGridView* kontrole

❑ Svojstva

- `Alignment` – poravnanje sadržaja ćelije
- `Format` – format teksta (*string*) ako se radi o tekstualnoj ćeliji
- `Font` – font tekstualne ćelije
- `ForeColor`, `BackColor` – boja teksta, boja pozadine ćelije
- `SelectionForeColor`, `SelectionBackColor` – boja teksta, boja pozadine označene ćelije
- `Padding` – razmak između sadržaja i okvira ćelije

❑ Primjer: FirmaWin – DokumentForm

```
dataGridViewCellStyle1.Alignment =  
    DataGridViewContentAlignment.MiddleCenter;  
dataGridViewTextBoxColumn7.DefaultCellStyle = dataGridViewCellStyle1;
```

Ulančavanje procedura

☐ Programska oprema mora slijediti poslovne procese.

- Gdje god je to moguće, treba smanjiti broj postupaka za jedan poslovni proces da korisnici ne bi ponavljali iste akcije.

☐ Kada u sustavu ne postoji potreban vezani podatak korisnik je prisiljen ...

- poništiti do tog trenutka unesene podatke,
- proći kroz nekoliko izbornika do šifarnika, unijeti novu šifru,
- vratiti se na mjesto gdje je šifra bila potrebna,
- ponoviti unos prije poništenih podataka i tek tada ih pohraniti.

☐ Rješenje: na polju za unos vrijednosti stranog ključa treba omogućiti

- otvaranje “prozora” s listom za pregled i odabir zahtijevanog podatka uz prikaz svih zapisa u odgovarajućoj tablici (*lookup*)
- otvaranje liste za pregled uz prikaz ograničenog skupa zapisa na temelju prethodno postavljenog uvjeta
- aktiviranje funkcije za unos ili izmjenu vezanog podatka
- aktiviranje čitavog modula za obradu vezanih podataka (*zoom*)

Povezivanje formi pri prijenosu vrijednosti stranog ključa

❑ Primjer: FirmaWin – Dokument (u stanju za uređivanje!)

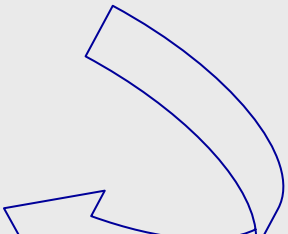
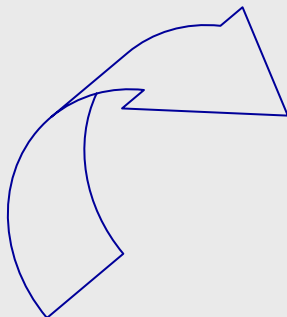
- Postupak Zoom osnovne forme preopterećen u stavkama dokumenta
- Pritiskom tipke F5 otvara se forma Artikl
- Pritiskom tipke F10 odabrani artikl sprema se u stavku dokumenta

```
if (InEditMode &&
    stavkaDataGridView.SelectedCells[0].ColumnIndex == 0 && ...) {
ArtiklForm f = new ArtiklForm();
if (f.ShowDialog() == DialogResult.OK) {
    artiklBindingSource.DataSource = artiklBll.FetchAll();
    ...
    if (stavkaDataGridView.Rows...IsNewRow) { // SKRAĆENO
        s = new Stavka();
    } else
        s = (Stavka)stavkaBindingSource.Current;
    ...

    s.SifArtikla = a.SifArtikla;
    s.NazArtikla = a.NazArtikla;
}
```

Povezivanje formi pri automatizaciji poslovnog procesa

❑ Primjer: FirmaWin - DokumentWizard



Id dokumenta **Vrsta** **Broj** **Datum**

Partner **Porez**

Prethodni dok. **Iznos**

	Naziv artikla	Mjera	Količina	Jedinična cijena	Rabat	Ukupna cijena	
	RAM, 512 MB, SDRAM, PC-133, brand name	▼	kom	1.00	407,00 kn	10.00%	366,30 kn
▶	Torba za notebook, MONAS M, crna	▼	kom	1.00	283,00 kn	0.00%	283,00 kn
*		▼					

Poslovni partner

Tip partnera
☐ Tvrtka ☒ Osoba

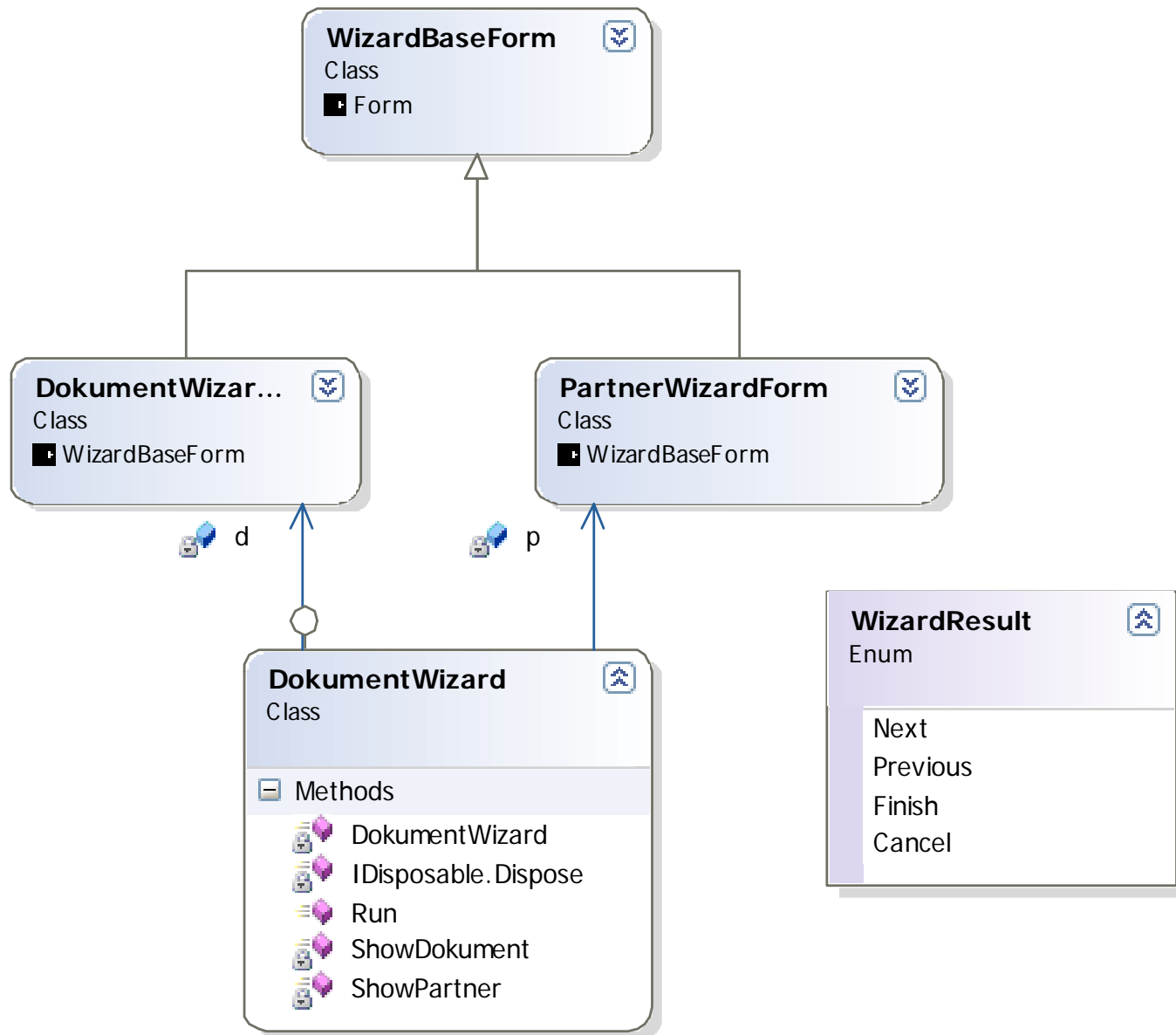
Ime
Prezime
JMBG

Adresa
Mjesto

Buttons: Odustani, Prethodni korak, Završi

Buttons: Novi partner, Odustani, Prethodni korak, Sljedeći korak

Dijagram razreda ulančavanja zaslonskih maski



Koraci povezivanja

```
private PartnerWizardForm p;  
private void ShowPartner()  
{  
    if (p == null)  
        p = new PartnerWizardForm();  
    WizardResult rez = p.ShowWizard();  
    if (rez == WizardResult.Next)  
    {  
        ShowDokument();  
    }  
}
```

```
private void ShowDokument()  
{  
    if (d == null)  
        d = new DokumentWizardForm();  
  
    d.SetPartner(p.Partner);  
    WizardResult rez = d.ShowWizard();  
    if (rez == WizardResult.Previous)  
    {  
        ShowPartner();  
    }  
}
```

Vlastiti dijalozi

❑ Svojstva forme

- `AcceptButton` – gumb koji će biti aktiviran pritiskom na Enter
 - npr. `this.AcceptButton = this.buttonOK;`
- `CancelButton` - gumb koji će biti aktiviran pritiskom na Escape

❑ Svojstva gumba

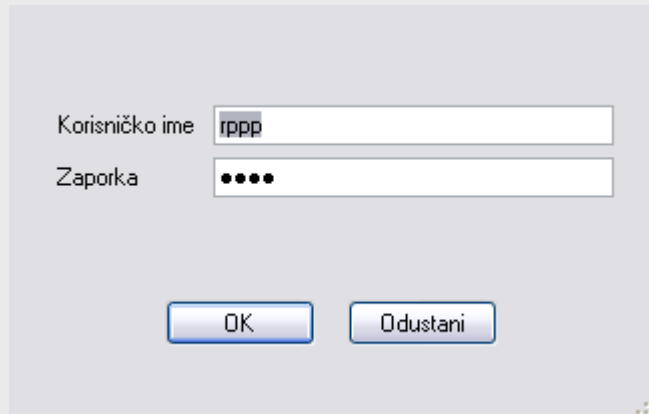
- `DialogResult` – neka od vrijednosti enumeratora `DialogResult`
 - `Abort`, `Cancel`, `Ignore`, `No`, `None`, `OK`, `Retry`, `Yes`
- na MODALNOJ formi, bilo koja postavka izuzev `None` automatski zatvara formu, pa se u obradi događaja može izostaviti `this.Close()`

```
this.buttonClose.DialogResult =  
    System.Windows.Forms.DialogResult.Cancel;  
...  
void buttonCancel_Click(object sender, EventArgs e)  
{  
    //DialogResult = DialogResult.Cancel;  
    //Close();  
}
```

Dijalog prijave korisnika

❏ Primjer: FirmaWin – LoginForm

- `textBoxZaporka.PasswordChar = "*" ;`
- `FormBorderStyle`
 - `MaximizeBox, MinimizeBox, ControlBox`



```
private SecurityBllProvider sec = new SecurityBllProvider();
private void buttonOk_Click(object sender, EventArgs e)
{
    if (sec.IsAuthenticated(textBoxKorisnik.Text,
                            textBoxZaporka.Text))
    {
        DialogResult = DialogResult.OK;
        Close();
    }
    else
    {
        MessageBox.Show("Krivo korisničko ime ili zaporka.",
                        "Neuspjela autorizacija",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```


Naredba *using*

❑ Naredba using

- definira doseg na kraju kojeg će upotrijebljeni objekt biti uništen (disposed)
- nema veze s direktivom using za uvođenje prostora imena

❑ Primjer: FirmaWin – StatusBusy

```
public class StatusBusy : IDisposable
{
    private Cursor c;
    public StatusBusy()
    {
        this.c = Cursor.Current;
        Cursor.Current = Cursors.WaitCursor;
    }
}
```

```
using (new StatusBusy())
{
    ArtiklForm f = new ArtiklForm();
    f.MdiParent = this;
    f.Show();
}
```

Kontrola MaskedTextBox

❑ Kontrola MaskedTextBox

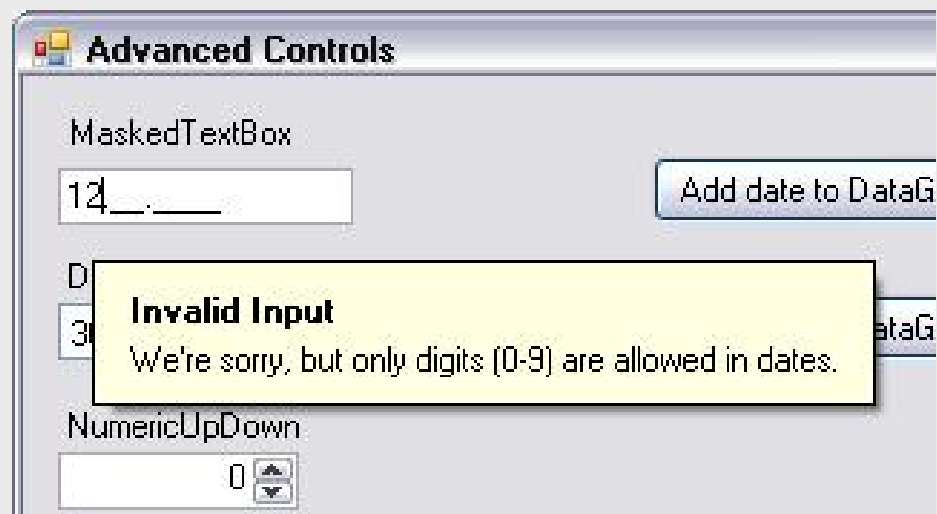
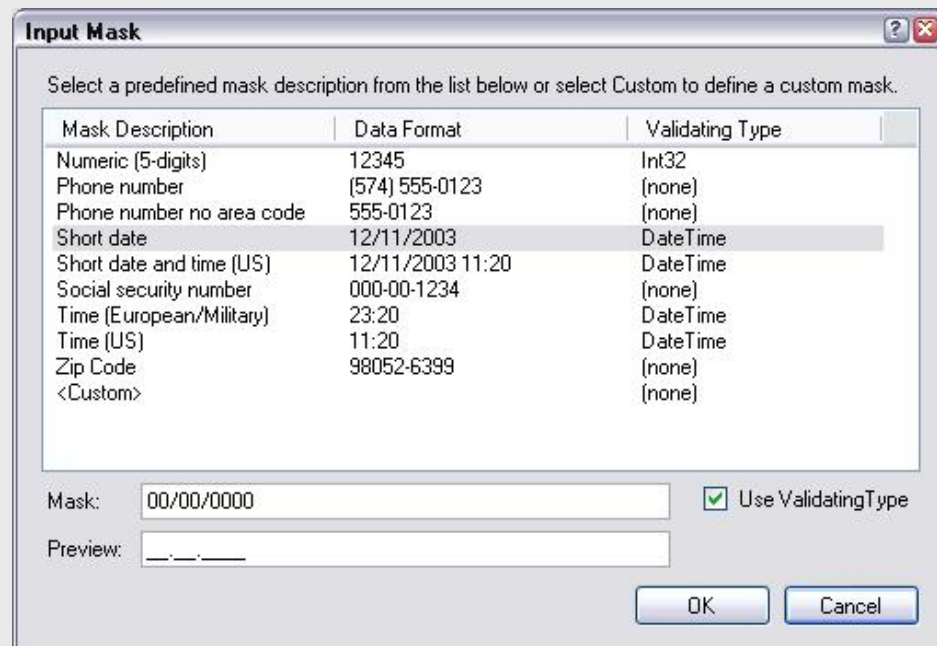
- Koristi masku za ograničenje unosa pojedinačne vrijednosti
- Zamjenjuje regularne izraze
- Postoje predefinirane maske
- Prilikom pogrešnog unosa je moguće grešku dojaviti pop-up prozorom i/ili zvučnim signalom

❑ Svojstva


- Mask – string maske
- BeepOnError – zastavica dojave

❑ Događaj

- MaskInputRejected – pogrešan unos



Kontrola *Splitter*

- ❑ Predstavlja razdjelnik za promjenu veličine na nju prislonjenih (docked) kontrola
- ❑ Svojstva
 - `BorderStyle` – obrub (`Fixed3D`, `FixedSingle`, `None`)
 - `MinSize` - minimalna udaljenost između Splitter kontrole i ruba spremnika na koji je kontrola prislonjena
 - `MinExtra` – minimalna udaljenost između Splitter kontrole i suprotnog ruba spremnika ili najbliže kontrole prislonjene na suprotnu stranu
 - `Cursor` – vrsta značke koja se pojavljuje prolaskom miša iznad kontrole
- ❑ Događaji
 - `SplitterMoving` – pomicanje u tijeku
 - `SplitterMoved` – pomicanje završeno
- ❑ Primjer:  Arhitekture (kasnije)

Reference

☐ Scott W. Ambler: User Interface Design Tips, Techniques, and Principles

- <http://www.ambysoft.com/essays/userInterfaceDesign.html>

☐ Papers and Essays by Jakob Nielsen

- <http://www.useit.com/papers/>

☐ Human-Computer Interfaces

- <http://www.cs.umd.edu/hcil/>

☐ GUI testing checklist

- <http://members.tripod.com/bazman/download/checklist.pdf>