TEHNIKE VIZUELNOG PROGRAMIRANJA – C#

MDI/SDI, tipični grafički elementi
Osnove GDI+

MDI APLIKACIJE

- MDI kontejner je forma glavne aplikacije u kojoj se otvaraju, "žive" i nestaju forme koje su namenjene obradi dokumenata.
- Jedna MDI aplikacija može istovremeno da radi sa više dokumenata (Word, Excel,...). Nasuprot MDI aplikacijama stoje SDI aplikacije.

- **`IsMdiContainer`**
  - Svojstvo koje definiše glavnu formu MDI aplikacije
- Postoji relacija roditelj-dete između glavne forme aplikacije i formi koje nastaju u toj aplikaciji.
- Primer kreirnja forme u MDI aplikaciji:

```
Frm1 dlg = new Frm1();
dlg.MdiParent = this;
dlg.Show();
```

- Moguća je obrada više formi u kontejneru.

```
foreach(Form frm in this.MdiParent.MdiChildren){}
```

MENIJI, TOOLBAR, STATUSBAR

GLAVNI I KONTEKSNI MENI

- Prikazuje listu akcija/komandi koje se mogu izvršavati.
- Grupisani u podmenije
- Kontrole koje obezbeđuju aplikaciji rad sa menijima su:
  - `MenuStrip`
  - `ContextMenuStrip`
- Objekti su tipa:
  - `ToolStripMenuItem`
  - `ToolStripTextBox`
  - `ToolStripComboBox`
  - `ToolStripSeparator`

FORMIRANJE GLAVNOG MENIJA

- Obično koristeći dizajner
- Programsko kreiranje
  - `menuStrip1 = new System.Windows.Forms.MenuStrip();`
- Kreiranje kolekcije stavki koja se dodaje ovoj kontroli. `Items` je tipa `_System.Windows.Forms.ToolStripMenuItem`
  - `menuStrip1.Items.Add (prvaGrupaStavkiToolStripMenuItem);`
  - `menuStrip1.Items.Add (drugaGrupaStavkiToolStripMenuItem);`
- Dodavanje standardnih svojstava
  - `menuStrip1.Location = new System.Drawing.Point(0, 0);`
  - `menuStrip1.Name = "menuStrip1";`
  - `menuStrip1.Size = new System.Drawing.Size(598, 24);`
  - `menuStrip1.TabIndex = 0;`
  - `menuStrip1.Text = "menuStrip1";`

17:42

FORMIRANJE PODMENIJA

- Na gotovo identičan način

17:42

```
// kreiranje
stavka11ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
prvaGrupaStavkiToolStripMenuItem.DropDownItems.Add (stavka11ToolStripMenuItem);
prvaGrupaStavkiToolStripMenuItem.DropDownItems.Add (stavka12ToolStripMenuItem);
prvaGrupaStavkiToolStripMenuItem.DropDownItems.Add
   (******************************);
//
prvaGrupaStavkiToolStripMenuItem.Name = "prvaGrupaStavkiToolStripMenuItem";
prvaGrupaStavkiToolStripMenuItem.Size = new System.Drawing.Size(105, 20);
prvaGrupaStavkiToolStripMenuItem.Text = "PrvaGrupaStavki";
// standardna svojstva
stavka11ToolStripMenuItem.Name = "stavka11ToolStripMenuItem";
stavka11ToolStripMenuItem.Size = new System.Drawing.Size(152, 22);
stavka11ToolStripMenuItem.Text = "Stavka 1 - 1";
```

Aktiviranje stavki menija

- Mišem
- Tastaturom
  - **Alt** postavlja glavni meni u fokus.
  - Ako je neka stavka u meniju prikazana tako da je neko slovo podvučeno u toj stavci, onda se ista aktivira izborom odgovarajućeg tastera.
  - Skraćena komanda menija se postiže dodavanjem znaka **"&"** ispred slova koje se želi podvući i iskoristiti za skraćeno pozivanje.

Dodavanje prečica stavkama

- Koristi se svojstvo `ShortcutKeys` stavke menija:
- **ShortcutKeys = Keys.Control | Keys.U**

17:42

DODAVANJE METODE ZA DOGAĐAJE

- Vrši se kao dodavanje rukovaoca događajima za bilo koju drugu kontrolu (očekivano)

```
stavka11.Click += new System.EventHandler (
                  stavka11ToolStripMenuItem_Click );
```

Primer 1 a.

- Dodati formi jedan meni sa dva podmenija i po dve stavke u svakom od podmenija.
- Dodati skraćene komande za podmenije.
- Dodati skraćene komande za stavke u podmeniju.
- Za po jednu stavku u svakom od podmenija obezbediti hvatanje događaja.

17:42

KONTEKSNI MENIJI

- Svojstvo **ContextMenuStrip** pripada kontrolama ili formama.
- Posebno napravljen meni se može vezati za neku određenu kontrolu ili formu i to je konteksni meni tj. meni koji zavisi od toga na šta se kliknulo desnim tasterom miša.

KREIRANJE KONTEKSNOG MENIJA

- Klasa nije ista kao kod glavnog menija

```
contextMS = new ContextMenuStrip();
```

- Dodavanje stavki u meni

```
contextMS.Items.Add(ToolStripMenuItem);
```

- Standardna svojstva

```
contextMS.Name = "contextMenuStrip1";
contextMS= new System.Drawing.Size(87, 54);
**********
```

17:42

Primer 1 b.

- Formi dodati jednu kontrolu ListBox sa par stavki.
- Dodati dva konteksna menija sa po dve stavke i vezati ih za formu odnosno listBox.
- Obezbediti reagovanje na po jednu stavku iz svakog konteksnog menija.

MDI APLIKACIJE I MENIJI

- Meniji za glavni prozor i prozore dokumenata su odvojeni.
- Kada nema otvorenih dokumenata vidljiv je samo meni *mdicontainer* prozora.
- Kada je neki dokument otvoren vidljive su stavke menija glavne forme i menija forme koji su otvoreni.

Primer 1 c.

- Dodati istoj aplikaciji Form2 i proizvoljan meni sa jednim podmenijem i sa dve stavke u njemu. Dodati rukovanje dogadjajem za neku stavku u ovom meniju.
- Testirati izgled stavki u meniju kada se otvaraju dokumenti.
- Šta se događa ako postoje dva različita dokumenta sa različitim menijima?

TOOLBAR

- Kreiranje i naziv klase: `toolStrip1 = new Forms.ToolStrip();`
- Stavke u toolbaru su tipa:

  `ToolStripItem` *što je apstraktna klasa iz koje se izvode posebno:*

  1. `ToolStripButton`
  2. `ToolStripSplitButton`
  3. `ToolStripSeparator`
  4. `ToolStripComboBox`
  5. `ToolStripProgressBar`
  6. **********

17:42

PRIMER 1 D.

- Dodati toolbar sa dva dugmeta koji treba da zamene dve stavke iz menija.
- Dodati meniju i druge kontrole koje stoje na raspolaganju

VEZIVANJE ZA DOGAĐAJE

- Svaka stavka u toolbaru može da pravi svoje rukovaoce događajima koji su tipični toj kontroli. Na primer:

- toolStripButton1.Click += new System.EventHandler(

  this.toolStripButton1_Click);

STATUS BAR

- Kreiranje i naziv klase:

```
statusStrip1 = new Forms.StatusStrip()
```

- Standardna svojstva:

```
tatusStrip1.Location = new System.Drawing.Point(0, 373);
statusStrip1.Name = "statusStrip1";
statusStrip1.Size = new System.Drawing.Size(598, 22);
statusStrip1.TabIndex = 4;
statusStrip1.Text = "statusStrip1";
```

- Objekti koji čine status nasleđuju apst. klasu `ToolStripItem`:
  - `ToolStripStatusLabel`
  - `ToolStripProgressBar`
  - `ToolStripDropDownButton`
  - `ToolStripSplitButton`

PRIMER 1 E.

- Istoj aplikaciji dodati status bar za glavnu formu i za forme otvorenih dokumenata. U status baru glavne forme prikazati koliko ima otvorenih dokumenata a u status baru svakog dokumenta prikazati vreme.

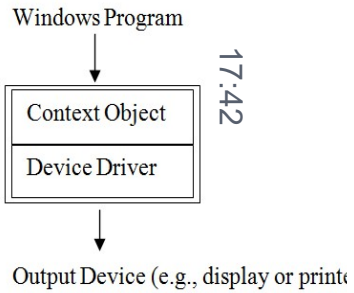
17:42

GDI

15

KONTEKST UREĐAJA

- Definiše u Windows programima pristup izlaznim uređajima. Na primer:
  - Displej (monitor)
  - Printer
- Ovaj objekat za grafički izlaz nazivamo
  - **KONTEKST UREĐAJA ili *device context* odnosno skraćeno**
- DC čini izlazne uređaje nezavisnim za programe (programere).
- Programi postaju celine koje rade nezavisno od vrste štampača ili displeja tj. grafičke karte ili ekrana.
- Generalno, biblioteke koje su zadužene za rad sa DC se nazivaju GDI (Graphical Device Interface).
- U .Net-u, ovaj GDI interfejs je značajno unapređen i označava se kao GDI+.
- .Net GDI+ klase uključuju neke važne Windows API metode za kreiranje grafičkih klasa.



PROSTORI IMENA ZA KLASE I STRUKTURE

- Prostor imena **`System.Drawing`** sadrži većinu klasa, struktura i nabrojivih lista koji obezbeđuju osnovne funkcije crtanja.

- `System.Darwing` sadrži druge prostore imena specijalizovane za neke celine. Na primer:
  - `System.Drawing.Imaging` za obradu slika,
  - `System.Drawing.Design` za dizajn kontrola..,
  - `System.Design.Text` za kontrolu fontova,
  - `System.Drawing.Printing` za kontrolu štampe.
- Na primer, tri važne .Net strukture koje se često koriste su:
  1. `Point,`
  2. `Size`
  3. `Rectangle.`

17:42

- Struktura `Point` ima dva svojstva: `X`, `Y` koja su celobrojne vrednosti.
- Za predstavljanje tačaka koristi se i struktura `PointF` gde su `X` i `Y` `float` tipa.
- Slično važi za `Size` i `Rectangle` imaju ekvivalentne strukture `SizeF` odnosno `RectangleF`.
- Struktura `Size` ima svojstva
  - `Width`
  - `Height`
- Struktura za opis pravougaonika - `Rectangle` može biti specificirana skupom vrednosti
  - `Top`, `Left`, `Bottom`, `Right`;
  - `Top`, `Left`, `Width`,`Height`;
  - `Location` (`Point`), `Size`.
- Postoji takođe i klasa `Region` koja se može koristiti za definisanje proizvoljnih kompleksnih oblika.
- Klasa `Color` ima definisane standardne boje u statičkim elementima, kao na primer `Color.Red`.
- Kada je potrebno definisati tačno neku boju koristi se metoda
  - `FromArgb` pomoću koje se zadaju posebno tri komponente boje
  - `Color.FromArgb(255,255,0)` ; generiše žutu boju

GRAFIČKI OBJEKAT

- **DC** u GDI+ je objekat klase `System.Drawing.Graphics`.
- Sva crtanja, oblici kao na primer pravougaonici, linije, tekst zahtevaju kreiranje DC objekta.
- Tipičan kod za kreiranje DC objekta je:
- **`Graphics dc = this.CreateGraphics();`**

PAINT MESSAGE

- Windows generiše `PAINT` kada je potrebno da neki prozor bude iscrtan. To može biti prvi put pri pojavi prozora ili pri osvežavanju prikaza.
- .Net framework poziva `Paint( )` *handler* (metoda koja odgovara na događaj) kada prozor treba da bude ponovo iscrtan.
- Dakle, operativni sistem ne memoriše izgled prozora. Iscrtavanje prozora se inicira porukama operativnog sistema.
- Sve vizuelne kontrole nasleđuju klasu *Control* u kojoj je realizovana metona `OnPaint` i to kao virtuelna. Dakle, ova metoda može da bude predefinisana u izvedenim klasama.
- Ovu metodu poseduje i `Form` klasa.

PRIMER:

- Na formi nacrtati liniju koja spaja gornji levi ugao sa donjim desnim uglom kada uradite dvostruki klik mišem.
  - U klasi Graphics se nalaze metode za iscrtavanje. Među njima biramo jednu koju koristimo za ovaj primer *DrawLine*

```
private void dblclk(object sender, System.EventArgs e)
{
        Point pt1 = new Point( 0, 0 );

        // nije dobro ako koristimo...
        // Point pt2 = new Point( this.Size );
        // ...jer se odnosi na celu formu
        Point pt2 = new Point( this.ClientSize );
        Graphics g = this.CreateGraphics();
        g.DrawLine( Pens.Blue, pt1, pt2 );
        g.Dispose();
}
```

17:42

Važno

- `Kada` kreirate grafički objekat obavezno morate osloboditi zauzete resurse. Ovo izvodite pozivom metode **`Dispose`.**
- Dodavanje metode za obradu događaja *Paint* se izvodi kao dodavanje bilo kog drugog događaja.
- Metoda za obradu događaja `Paint` uzima DC preko parametra `PaintEventArgs` tako da se može jednostavno koristiti DC.
- `Graphics` – DC objekat
- `ClipRectangle` – pravougaonik koji se iscrtava (`e.Graphics.VisibleClipBounds`)

Primer

- Rešiti problem iz prethodnog primera iscrtavanjem na ovaj događaj. Samo početno iscrtavanje obaviti na dvostruki klik!

GRAFIČKI OBJEKTI

Pen

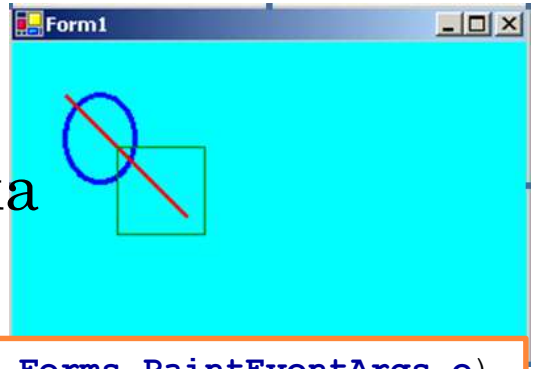
- Definiše olovku sa kojom se crta. Tri osnovna atributa:
  - Debljina
  - Boja
  - Stil
- Važno: Ne zaboravite da oslobodite resurse svakog graf. objekta, pa i olovke.

Brush

- Klasa koja služi za popunjavanje grafičkih površina.
- Da bi se napravio neki objekat ove klase koriste se izvedene kalse
  - `SolidBrush`
  - `TextureBrush`
  - `LinearGradientBrush`

PRIMER

Napisati kod koji će iscrtaviti tri oblika nalik slici.




```
private void Form1_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
{
   //---first create a device context
   // Graphics dc = this.CreateGraphics(); normally but Paint
   // is given a DC through the PaintEventArgs parameter
   Graphics dc = e.Graphics;

   Pen bluePen = new Pen(Color.Blue, 3);  // 3 pixels wide
   dc.DrawEllipse(bluePen, 30,30,40,50);  //30,30 is top,left, 40=width,
   50=height
   Pen redPen = new Pen(Color.Red, 2);
   dc.DrawLine(redPen, 30,30,100,100);
   Pen greenPen = new Pen(Color.Green);
   dc.DrawRectangle(greenPen,60,60,50,50);
}
```

- Modifikovati prethodni primer tako da se pravougaonik popunjava koristeći četkicu *LinearGradientBrush* sa prelivom boja od Crvene od Žute.