

Resursi

Maja Štula

Ak. God. 2011/2012

DOTNET resursi

- U dotnetu resurs se definira kao neizvršni podaci (*nonexecutable data*) koji se koriste u aplikaciji. To mogu biti podaci tipa poruke o grešci ili podaci koji se koriste u korisničkom sučelju.
- Resursi u dotnetu sadrže podatke u različitim formama, kao što su stringovi, slike, objekti.
- [Pohranjivanje podatka](#) u resurs omogućava mijenjanje podataka bez rekompajliranja cijele aplikacije.

DOTNET

- Resursi se u dotnetu pohranjuju u file-ove sa ekstenzijom .resx (*XML-formatted resource files*) i .resources.
- Visual Studio ima posebne editore za resurse tako da se resursi ne moraju ručno mijenjati. (Winres.exe)
- .resources su CLR binarni file-ovi koji se dobiju iz tekstualnih resursa.
- Generiranje .resources iz tekstualnih file-ova radi se pomoću programa *resgen* (C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin\ResGen.exe).

DOTNET

- Visual Studio projekt omogućava dva načina rada sa resursim.
- Resursi mogu biti linkani u projekt ili mogu biti ugrađeni (*embedded*). Moguće je kombinirati ova dva načina, ali je uobičajenije u jednom projektu koristiti samo jedan način.
- Linkani resursi su datoteke u projektu čiji se sadržaj tijekom kompajliranja dodaje u manifest aplikacije, a aplikacijska resurs datoteka (.resx) sadrži samo relativni put do datoteka na disku.
- Kod ugrađenih resursa sadržaj resursa se sprema u .resx datoteku.
- Resursi se u konačnici ili dodaju u .exe aplikacije (i linkani i ugrađeni), ili se odvajaju u zasebni “satelitski” assembly (pomoću alata Al.exe (Assembly Linker) kojim se kompajliraju .resources datoteke u assembly. “Satelitski” assembly ne smije sadržavati izvršni kod) ili se u aplikaciji direktno učitavaju .resources datoteke.

ResourceManager

- Klasa ResourceManager se koristi za povezivanja sa lokalizacijskim resursima (*culture-specific resources*), te podržava serijalizaciju resursa.
- Metode klase ResourceManager poput GetObject ili GetString mogu dohvatiti određeni resurs s obzirom na lokalizacijske postavke.

C# ikona

- Klasa *Icon* iz imenskog prostora *System.Drawing* predstavlja ikonu.
- Klasa *System.Windows.Forms.Form* ima svojstvo *Icon* preko kojeg se dohvaća i postavlja ikona forme.

```
public Icon Icon {get; set;}
```

- [Primjer](#) postavljanja sistemske ikone upitnika:

```
this.Icon = System.Drawing.SystemIcons.Question;
```

C# ikona

- [Primjer](#) postavljanja vlastite ikone iz resursa:

```
// assembly je resursi_ikona; ikona se zove Icon1.ico
// Build Action ikone treba biti postavljen na Embedded Resource
Icon mojalkona = new
    Icon(GetType().Assembly.GetManifestResourceStream("resursi_ikona.
        Resources.Icon1.ico"));
this.Icon = mojalkona;
```

```
//ili
this.Icon = resursi_ikona.Properties.Resources.Icon1;
```

- Dohvaćanje ugrađenih resursa može se napraviti pomoću metode `GetManifestResourceStream` iz klase `Assembly` koja za skup dohvaća navedeni ugrađeni resurs (`Embedded Resource`) iz skupa.

C# ikona - resursi

- Svojstvo *Build Action* kaže kompajleru što treba napraviti s određenom datotekom.
- Moguće opcije su:
- Embedded Resource – uključi datoteku kao ugrađeni resurs.
- Compile – kompajliraj datoteku.
- Content – ovako označene datoteke mogu se referencirati od Visual Studio setup projekata.
- None – ne radi ništa. Ova se akcija koristi za dokumentaciju ili slične datoteke koje nije potrebno kompajlirati i nije potrebno dodavati u assembly.

C# učitavanje slike

- Imenski prostor `System.Windows.Forms` sadrži klasu `PictureBox` koja preko svojstva *Image* omogućava učitavanje slike iz datoteke.
- Primjer:

```
try
{
    this.pictureBox1.Image =
        System.Drawing.Image.FromFile("D:\\predavanja\\windows\\pred
        avanja\\dotnet.jpg");
}
catch(OutOfMemoryException greska)
{ MessageBox.Show("Slika se nije učitala"); }
```

Upravljanje iznimkama

- CLR koristi i podržava rukovanje greškama preko iznimaka (exception).
- Sve .NET framework operacije u slučaju problema generiraju iznimku.
- Iznimke omogućavaju bolje upravljanje greškama nego npr. povratne vrijednosti, jer se iznimke ne mogu preskočiti pa se greška ne može propagirati kroz aplikaciju.
- Iznimka je neočekivano ponašanje ili uvjeti koji će dovesti do greške (npr. ne postojanje datoteke koju želite otvoriti). Iznimka se može podignuti zbog greške u samom kôdu aplikacije, ili u kôdu pozvane biblioteke, zbog nedostupnih resursa,...

Upravljanje iznimkama

- Iznimka je objekt klase koja nasljeđuje *Exception* klasu.
- Iznimku podiže dio kôda u kojem se problem javio.
- Iznimka se prosljeđuje na stôg dok je aplikacija ili ne obradi ili dok program ne završi.

Upravljanje iznimkom

- Try-catch naredba omogućava nadzor izvršavanja određenog kôda i preusmjeravanje izvršavanja u slučaju da je nadzirani kôd uzrokovao neku grešku.
- Naredba se sastoji od *try* bloka i jednog ili više *catch* blokova.

```
try
    { nešto }
catch (Exception1 e1)
    { poduzmi nešto }
catch (Exception2 e2)
    { poduzmi nešto drugo }
```

Upravljanje iznimkom

- Kada se desi iznimka, zaustavlja se izvođenje kôda i upravljanje se prenosi na najbliži handler iznimke.
- Zbog toga se može desiti da se preskoči izvršavanje koda koji se obavezno treba izvršiti (npr. zatvaranje otvorene datoteke,...) čak i u slučaju greške.
- Tada se koristi finally blok koda koji će se uvijek izvršiti, bez obzira da li se podigla iznimka ili nije.

```
try
    { nešto }
catch (Exception1 e1)
    { poduzmi nešto }
catch (Exception2 e2)
    { poduzmi nešto drugo }
finally { uvijek se izvrši }
```

Podizanje iznimke

- Podizanje vlastite iznimke radi se naredbom *throw*.

```
Class Employee
{
    private string name;
    public string Name
    {
        get { return name; }
        set {
            if( value == null)
            {
                throw new ArgumentNullException("value");
            }
            name = value; }
    }
}
```

Predefinirane iznimke

Exception type	Base type	Description	Example
Exception	Object	Base class for all exceptions.	None (use a derived class of this exception).
SystemException	Exception	Base class for all runtime-generated errors.	None (use a derived class of this exception).
IndexOutOfRangeException	SystemException	Thrown by the runtime only when an array is indexed improperly.	Indexing an array outside its valid range: <code>arr[arr.Length+1]</code>
NullReferenceException	SystemException	Thrown by the runtime only when a null object is referenced.	<code>object o = null;</code> <code>o.ToString();</code>
AccessViolationException	SystemException	Thrown by the runtime only when invalid memory is accessed.	Occurs when interoperating with unmanaged code or unsafe managed code, and an invalid pointer is used.
InvalidOperationException	SystemException	Thrown by methods when in an invalid state.	Calling <code>Enumerator.GetNext()</code> after removing an <code>Item</code> from the underlying collection.
ArgumentException	SystemException	Base class for all argument exceptions.	None (use a derived class of this exception).
ArgumentNullException	ArgumentException	Thrown by methods that do not allow an argument to be null.	<code>String s = null;</code> <code>"Calculate".IndexOf (s);</code>
ArgumentOutOfRangeException	ArgumentException	Thrown by methods that verify that arguments are in a given range.	<code>String s = "string";</code> <code>s.Chars[9];</code>
ExternalException	SystemException	Base class for exceptions that occur or are targeted at environments outside the runtime.	None (use a derived class of this exception).
ComException	ExternalException	Exception encapsulating COM HRESULT information.	Used in COM interop.
SEHException	ExternalException	Exception encapsulating Win32 structured exception handling information.	Used in unmanaged code interop.

C# kursor

- Klasa `System.Windows.Forms.Form` ima svojstvo *Cursor* preko kojeg se dohvaća i postavlja kursor forme.

```
public Cursor Cursor {get; set;}
```

- [Primjer](#) postavljanja sistemskog kursor pješčanog sata:

```
this.Cursor = Cursors.WaitCursor;
```


.NET 3.x

- Klasa `System.Windows.Window` ima svojstvo *Icon* preko kojeg se dohvaća i postavlja ikona prozora. [PRIMJER](#)

```
public ImageSource Icon { get; set; }
```

- Klasa `System.Windows.Media.ImageSource` je apstraktna klasa, ali zato možete preko klase `System.Windows.Media.Imaging.IconBitmapDecoder` učitati ikonu i dodijeliti je prozoru.

```
IconBitmapDecoder ibd = new IconBitmapDecoder(new  
    Uri(@"C:\Program Files\Microsoft Visual Studio  
    8\VC\VCWizards\AppWiz\Generic\Application\templates\1033\root.i  
    co", UriKind.RelativeOrAbsolute), BitmapCreateOptions.None,  
    BitmapCacheOption.Default);  
this.Icon = ibd.Frames[0];
```

.NET 3.x

- Windows Presentation Foundation (WPF) samostalne aplikacije (*standalone applications*) imaju dva tipa ikona:
 - Ikonu vezanu uz čitavi assembly, koja se specificira preko *ApplicationIcon* svojstva projekta. Ova se ikona koristi kao desktop ikona.
 - Ikonu za svaki prozor specificiranu preko svojstva *Icon* svakog prozora. Ta se ikona smješta na naslovu prozora (*caption* ili *title bar*) u task bar-u i sl.
- WPF prozor uvijek ima ikonu. Ukoliko nije postavljeno svojstvo *Icon* prozora, WPF prikazuje ikonu prozora prema sljedećim pravilima:
 - Ako je definirana, koristi se ikona assembly-ja.
 - Ako nije definirana ikona assembly-ja koristi se default-na Microsoft Windows ikona.
- Svojstvo *Icon* prozora ne može se postavljati ukoliko se prozor prikazuje u pretraživaču.

.NET 3.x učitavanje slike

// Create source

```
BitmapImage myBitmapImage = new BitmapImage();
```

// BitmapImage.UriSource must be in aBeginInit/EndInit block

```
myBitmapImage.BeginInit();
```

```
myBitmapImage.UriSource = new
```

```
    Uri(@"D:\predavanja\windows\predavanja\dotnet.jpg");
```

```
myBitmapImage.EndInit();
```

//set image source

```
this.image1.Source = myBitmapImage;
```

.NET 3.x učitavanje slike

- Klasa Uri iz imenskog prostora System (assembly system.dll) pruža objekt za predstavljanje i dohvat URI-ja (*uniform resource identifier*).
- URI predstavlja resurs koji aplikacija može dohvatiti na intranetu ili Internetu.

```
Uri siteUri = new Uri("http://www.fesb.hr/");
```

```
Uri siteUri = new Uri(@"D:\predavanja\windows\predavanja\dotnet.jpg");
```

```
Uri siteUri = new Uri("D:\\predavanja\\windows\\predavanja\\dotnet.jpg");
```