

Rad sa datotekama i predefiniranim dijalog kontrolama - .NET

Maja Štula

Ak. God. 2011/2012

Dijalog kontrole

- .NET 2.x nudi nekoliko gotovih prozora određenih funkcionalnosti koji se zovu dijalog prozori.
- U WIN32 API programskom modelu su se zvali predefinirani sistemski dijalog prozori. Za razliku od WIN32 API u .NET-u nema razlike između prozora (*window*) i dijalog prozora (*dialog*).
- Dijalog kontrole su prozori za izbor datoteke za otvaranje sa diska (klasa `System.Windows.Forms.OpenFileDialog`), prozori za izbor datoteke za spremanje na disk (klasa `SaveFileDialog`), prozori za izbor fonta (klasa `ColorDialog`), prozori za izbor boje (klasa `FontDialog`) i prozori za izbor direktorija (klasa `FolderBrowserDialog`).

C# rad sa dijalog kontrolama

- U C# dijalog prozorima za otvaranje i pohranu datoteke pristupa se preko [komponenti](#):

OpenFileDialog (klasa
System.Windows.Forms.OpenFileDialog)

SaveFileDialog (klasa
System.Windows.Forms.SaveFileDialog)

C# rad sa datotekama

- Klasa File (System.IO.File) sadrži statičke metode za kreiranje, kopiranje, brisanje i otvaranje datoteke te omogućava kreiranje FileStream (System.IO.FileStream) objekta.
- Klasa FileStream omogućava čitanje iz datoteke, pisanje u datoteku, zatvaranje datoteke. Također omogućava rad i sa drugim sistemskim handlovima tipa file-a kao što su pipe-ovi, standardni ulaz (stdin) i standardni izlaz (stdout).

C# rad sa datotekama

- Čitanje i pisanje može biti sinkrono i asinkrono. Po defaultu su operacije sa datotekama sinkrone.
- [Klasa File](#) sadrži statičke metode za otvaranje datoteke. Povratna vrijednost je instanca FileStream objekta preko kojeg se onda čita ili piše otvorena datoteka.

```
FileStream moj = File.Open(openFileDialog1.FileName,  
    System.IO.FileMode.Open);  
byte[] b = new byte[1024];  
UTF8Encoding temp = new UTF8Encoding(true);  
while (moj.Read(b, 0, b.Length) > 0)  
{  
    richTextBox1.AppendText(temp.GetString(b));  
}
```

Kodiranje

- UTF8Encoding klasa definira objekt koji može sadržavati niz karaktera kodiranih sa UTF-8 kodom.
- Kodiranje je postupak pretvaranja jednog formata zapisa u drugi. Kod tekstualnih datoteka je potrebno definirati kodiranje kojim će se tekstualni sadržaj pretvoriti u binarni sadržaj. Postoji cijeli niz različitih kodova kojima se tekstualni sadržaj pohranjuje na računalo tj. kojima se tekstualni sadržaj pretvara u binarni zapis koji se onda može pohraniti na računalo.

Kodiranje

- Jedan od prvih definiranih kodova je ASCII (1963) kod kojeg se slovo kodira sa 7 bita, te prošireni ASCII sa 8 bita po slovu.
- ISO standard 8859 također definira kodiranje svakog slova sa 8 bita (ISO 8859-2 – Central European), Microsoftovi kodovi (Windows-1250 – Central European) također, te Unicode koji sadrži nekoliko podskupova, jedan od njih je i UTF-8 koji koristi promjenljivi broj bitova za kodiranje slova, za ASCII koristi 8, ali za neke druge kodove koristi više od 8 bitova.
- Npr. [pogledajmo](#) kako će se slovo a i slovo Č zapisati u datoteku na disku ako koristimo UTF-8 formati kodiranja ili ako koristimo ASCII.

C# rad sa datotekama

- `File.Open(String, FileMode)` – otvara datoteku čije je ime navedeno kao prvi argument, drugi argument je `FileMode`.
- `FileMode` je enumeracija kojom se definira način otvaranja datoteke. Može biti `CreateNew`, `Create`, `Open`, `OpenOrCreate`, `Truncate` ili `Append`. Prilikom izvođenja operacije na datoteci operacijski sustav provjerava da li je određene operacija dozvoljena, ukoliko nije, generira se iznimka (*exception*).
- Osim `FileMode` svojstva bitno je svojstvo i `FileAccess` koje je također enumeracija sa vrijednostima `Read`, `Write` i `ReadWrite`.
- Može se eksplicitno pozivati u metodi open oblika `File.Open(String, FileMode, FileAccess)`.

C# rad sa datotekama

- Pored klase FileStream postoje i druge klase za rad sa streamom podataka pa je tako sa tekstualnom datotekom lakše koristiti klasu StreamReader (System.IO.StreamReader).

```
using (StreamReader sr = File.OpenText(openFileDialog1.FileName))
{
    string s = "";
    while ((s = sr.ReadLine()) != null)
    {
        richTextBox1.AppendText(s);
        richTextBox1.AppendText("\n");
    }
}
```

- Naredba *using* definira doseg (scope) na kraju kojeg će objekt biti oslobođen (uništen).

C# rad sa datotekama

- Ili jednostavnije i preporučeno korištenje i metoda iz klase File za dohvaćanje sadržaja datoteke:
if (openFileDialog1.ShowDialog() == DialogResult.OK)
String dohvacen = File.ReadAllText(openFileDialog1.FileName);
- File.ReadAllText(String) – otvori tekstualnu datoteku, pročitaj sve linije do kraja datoteke i zatvori je.
- File.ReadAllText(String, Encoding) – otvori tekstualnu datoteku, pročitaj sve linije do kraja datoteke koristeći navedeni tip kodiranja u drugom argumentu metode i zatvori je.

C# rad sa datotekama

- Imenski prostor `System.Runtime.Serialization` sadrži klase za serijalizaciju i deserijalizaciju objekata. Serijalizacija je postupak pretvaranja objekta u niz bajtova koji se onda mogu pohraniti u datoteku. Deserijalizacija je obrnuti postupak.
- Imenski prostor `System.Runtime.Serialization.Formatters` sadrži klase za formatiranje objekata prilikom serijalizacije i deserijalizacije.

```
private void kodirajfile(object sender, EventArgs e)
{
    FileStream moj = File.Open(saveFileDialog2.FileName, System.IO.FileMode.Create);
    Class1 mojobjekt = new Class1();
    mojobjekt.pravi = richTextBox1.Text;
    BinaryFormatter b=new BinaryFormatter(); (System.Runtime.Serialization.Formatters.Binary)
    b.Serialize(moj,mojobjekt);
    moj.Close();
}
```

[Serializable] ← atribut

```
public class Class1
{
    public int bez = 22;
    public string pravi;
}
```

Atribut

- C# ima mehanizam za definiranje deklarativnih tagova koji se zovu atributi i koji pružaju neke dodatne informacije u kodu.
- Npr. deklarativne informacije o kodu su modifikatori *public*, *protected*, *internal*, *private*.
- C# atributi omogućavaju definiranje novih tipova deklarativnih informacija.

- PRIMJER:

```
using System;
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Struct)]
public class Author : Attribute
{
    public Author(string name) { this.name = name; version = 1.0; }
    public double version;
    string name;
}
private void dohvatiAtributToolStripMenuItem_Click(object sender, EventArgs e)
{
    Attribute[] ispis = Attribute.GetCustomAttributes(typeof(Class1));
    foreach(Attribute attr in ispis)
    {
        if (attr is Author)
        {
            Author a = (Author)attr; MessageBox.Show(a.GetName());
            MessageBox.Show( a.version.ToString()); } }
}
```

WINFX rad sa datotekama

- Za rad sa standardnim dijalog prozorima za otvaranja i spremanje datoteka potrebno je uključiti imenski prostor Microsoft.Win32.

- Primjer:

```
void OtvoriDijalog(object sender, RoutedEventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.FileName = "Dokument"; // Default file name
    dlg.DefaultExt = ".txt"; // Default file extension
    dlg.Filter = "Tekst file (.txt)|*.txt"; // Filter files by extension
    Nullable<bool> result = dlg.ShowDialog();
    // Process open file dialog box results
    if (result == true)
    {
        // Open document
        using (StreamReader sr = File.OpenText(dlg.FileName))
        {
            string s = "";
            while ((s = sr.ReadLine()) != null)
            {
                richTextBox1.AppendText(s);
                richTextBox1.AppendText("\n");
            }
        }
    }
}
```