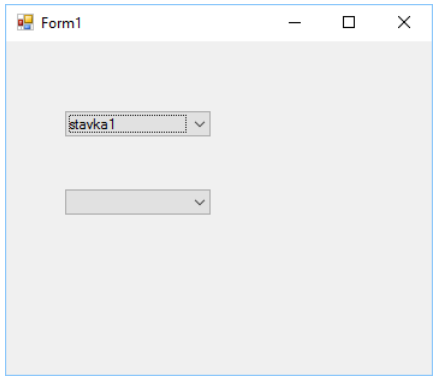
Kontrole ComboBox i ListBox

ComboBox kontrola

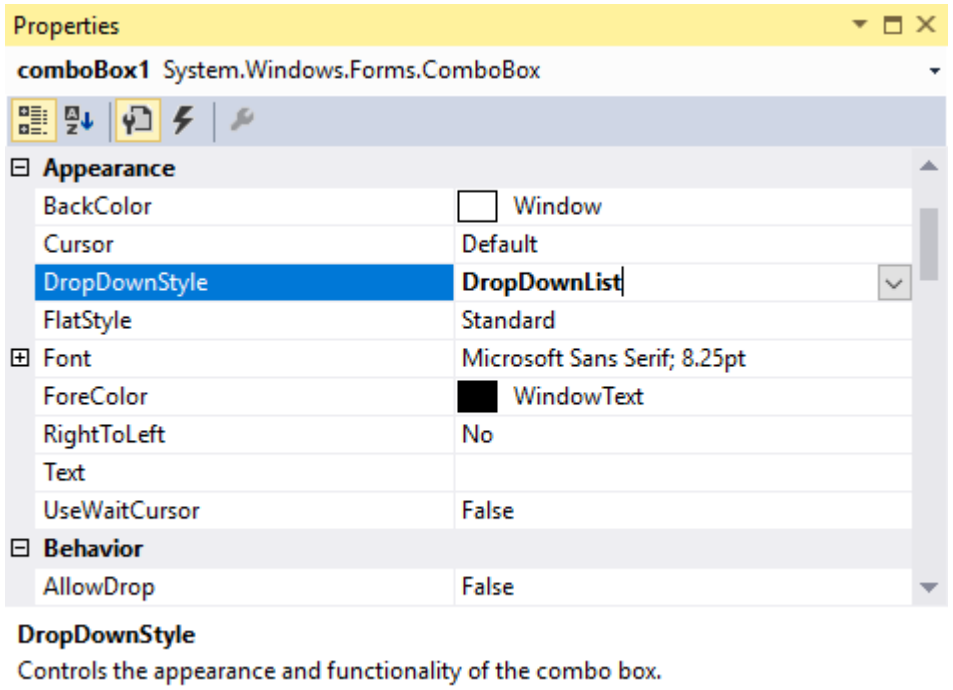
- Svojstvo **Items** daje kolekciju stavki ComboBox kontrole
- **SelectedIndex** svojstvo vraća indeks selektovane stavke kombo boksa
  - int selectedIndex = comboBox1.SelectedIndex;
- **SelectedItem** vraća selektovanu stavku kombo boksa koja je tipa object
  - object selectedItem = comboBox1.SelectedItem;
- Dodavanje stavki u kombo boks:
  - comboBox1.Items.Add(stavka1);
  - comboBox1.Items.AddRange(new object[] {"stavka1","stavka2","stavka3"});

Dodavanje stavki iz koda

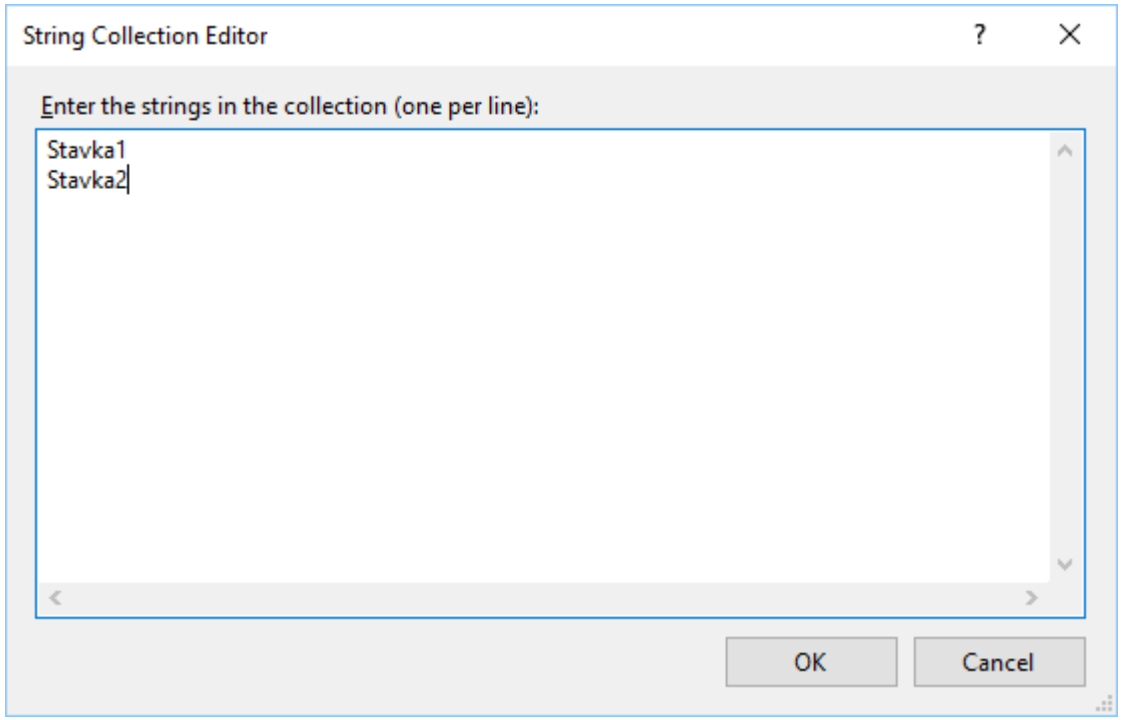
```
private void Form1_Load(object sender, EventArgs e)
{
    comboBox1.Items.Add("stavka1");
    comboBox1.Items.Add("stavka2");
    comboBox1.Items.Add("stavka3");
    comboBox1.SelectedIndex = 0;

    comboBox2.Items.AddRange
        (new object[]{"stavka1","stavka2","stavka3","stavka4"});
}
```



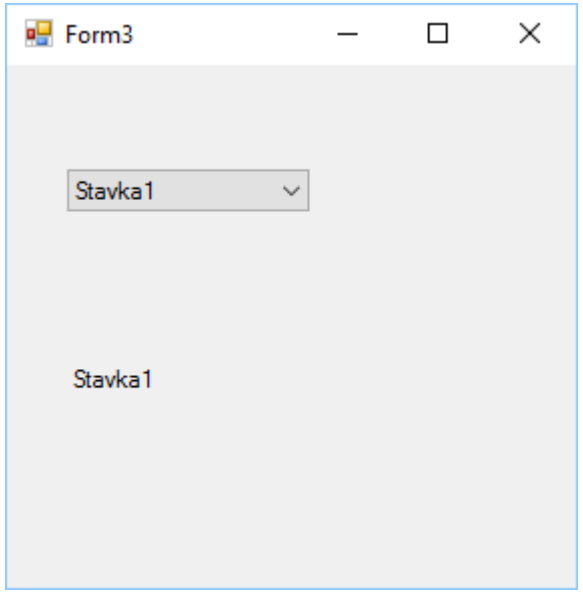
Svojstvo DropDownStyle



Dodavanje stavki u ComboBox u dizajn modu



SelectedIndexChanged događaj

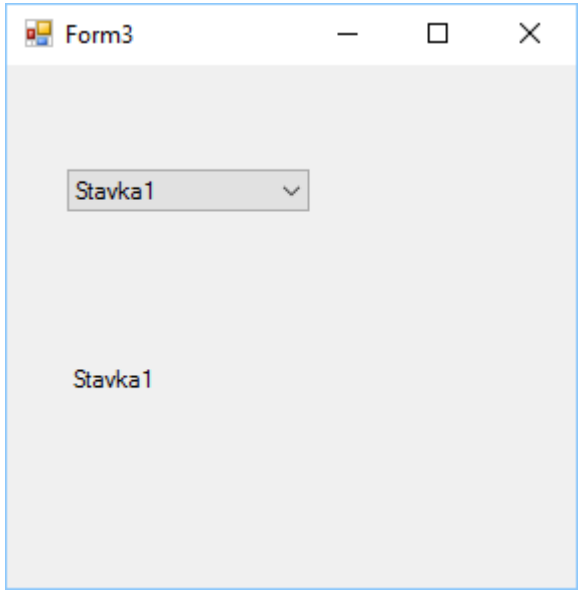



```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    label1.Text = comboBox1.SelectedItem.ToString();
}
```

Povezivanje sa nizom

```
private void Form3_Load(object sender, EventArgs e)
{
    string[] stavke = {"Stavka1","Stavka2","Stavka3","Stavka4" };
    comboBox1.DataSource = stavke;
    comboBox1.SelectedIndex = -1;
}
```

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex > -1)
    {
        label1.Text = comboBox1.SelectedItem.ToString();
    }
    else
    {
        label1.Text = "";
    }
}
```



Povezivanje sa rečnikom-1

```
private void Form4_Load(object sender, EventArgs e)
{
    Dictionary<int, string> recnik = new Dictionary<int, string>();
    recnik.Add(1, "Vrednost1");
    recnik.Add(2, "Vrednost2");
    recnik.Add(3, "Vrednost3");

    BindingSource bs = new BindingSource();
    bs.DataSource = recnik;

    comboBox1.DataSource = bs;

    comboBox1.DisplayMember = "Value";

    comboBox1.SelectedIndex = -1;

}
```

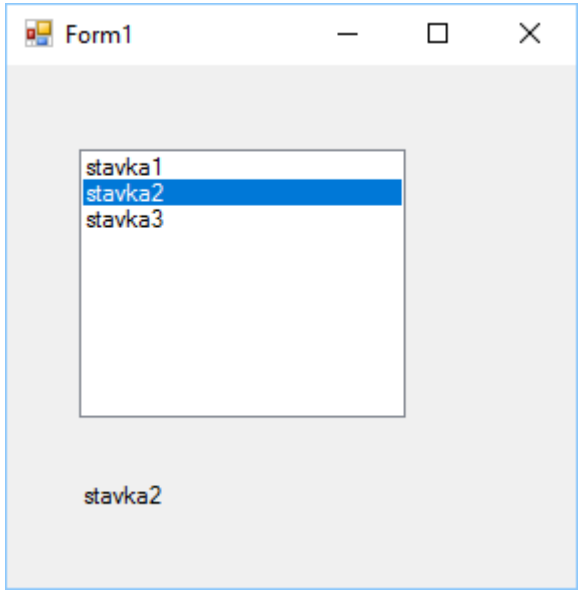
Povezivanje sa rečnikom-2

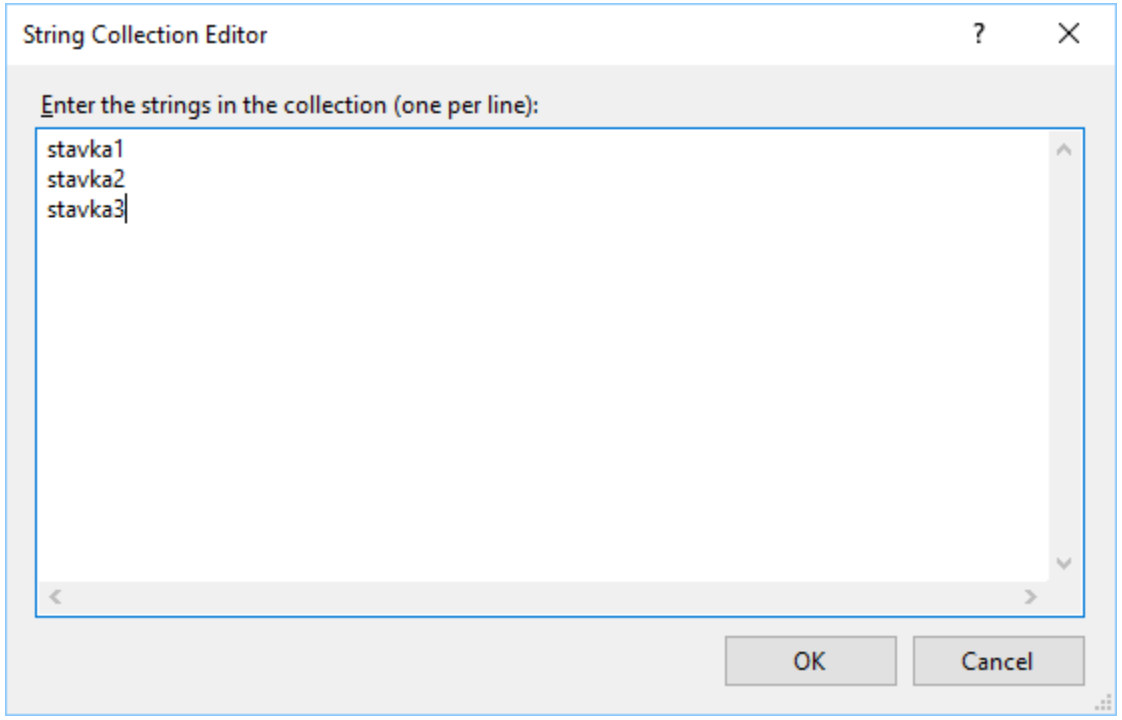
```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex > -1)
    {
        KeyValuePair<int, string> selektovano =
            (KeyValuePair<int, string>)comboBox1.SelectedItem;

        label1.Text = selektovano.Key + " " + selektovano.Value;
    }
    else
    {
        label1.Text = "";
    }
}
```

Svojstva ListBox kontrole

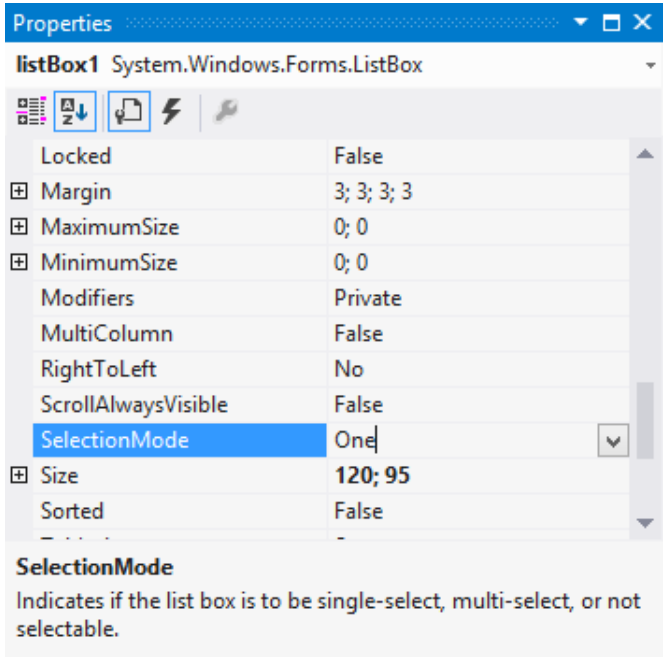
- **SelectedIndices** svojstvo vraća kolekciju selektovanih indeksa
- **SelectedIndex**
- **SelectionMode** svojstvo određuje koliko stavki može biti selektovanano u ListBox kontroli
  - MultiSimple omogućava selektovanje više stavki
  - MultiExtended omogućava korišćenje tastera SHIFT
- **SelectedItems** vraća kolekciju selektovanih stavki
- **SelectedItem**

Dodavanje stavki



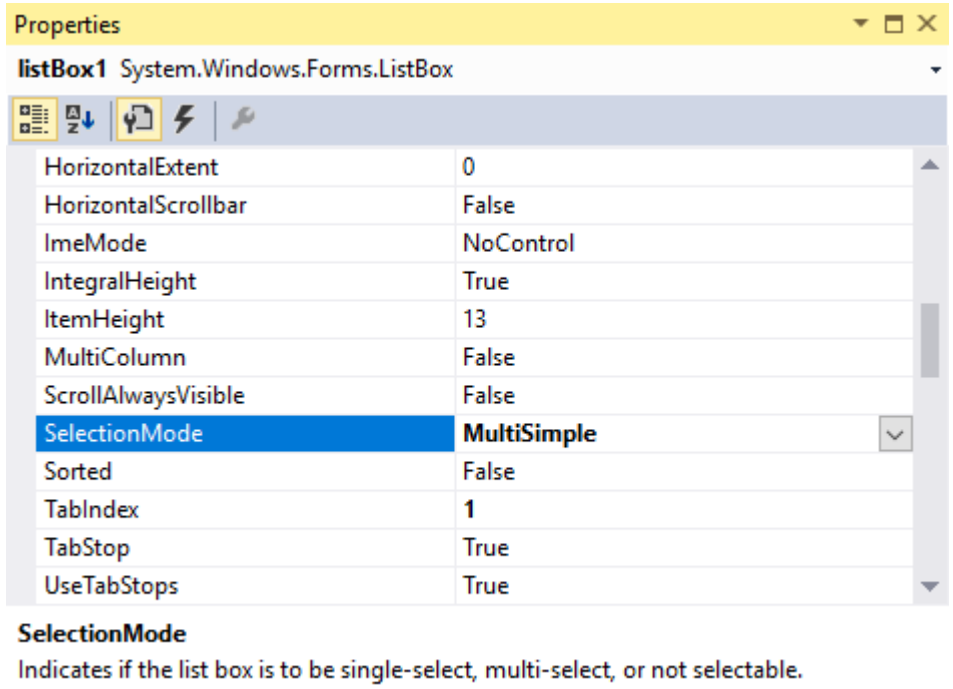


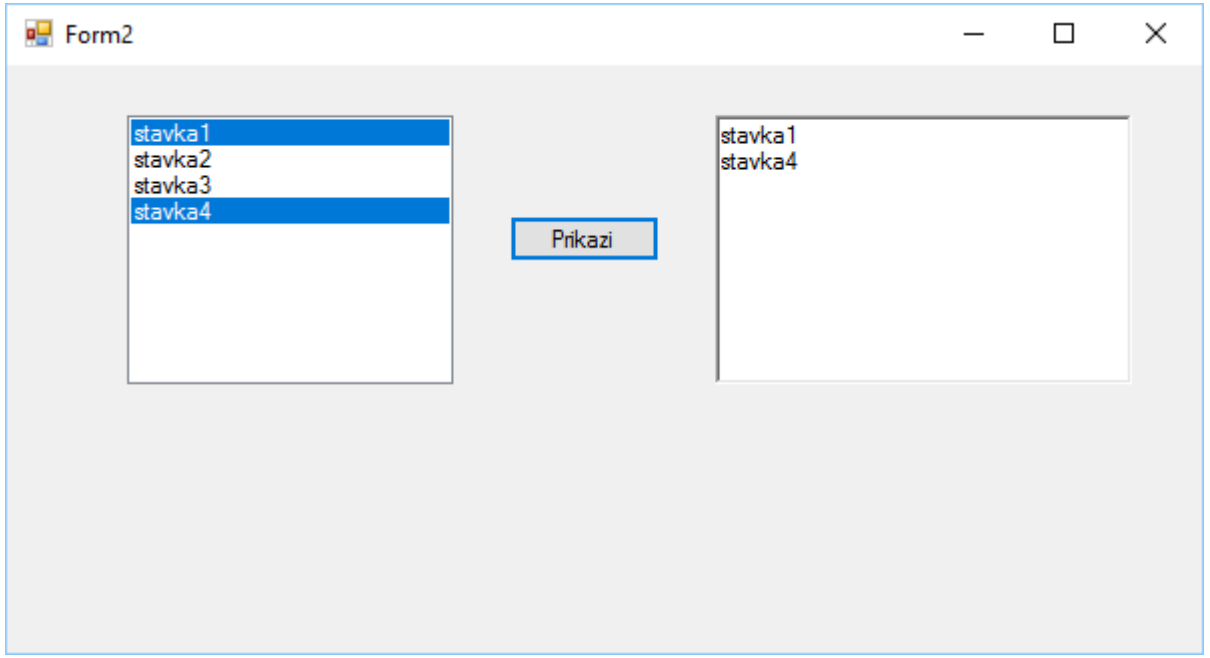
Događaj SelectedIndexChanged




```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    label1.Text = listBox1.SelectedItem.ToString();
}
```

ListBox kontrola - MultiSimple

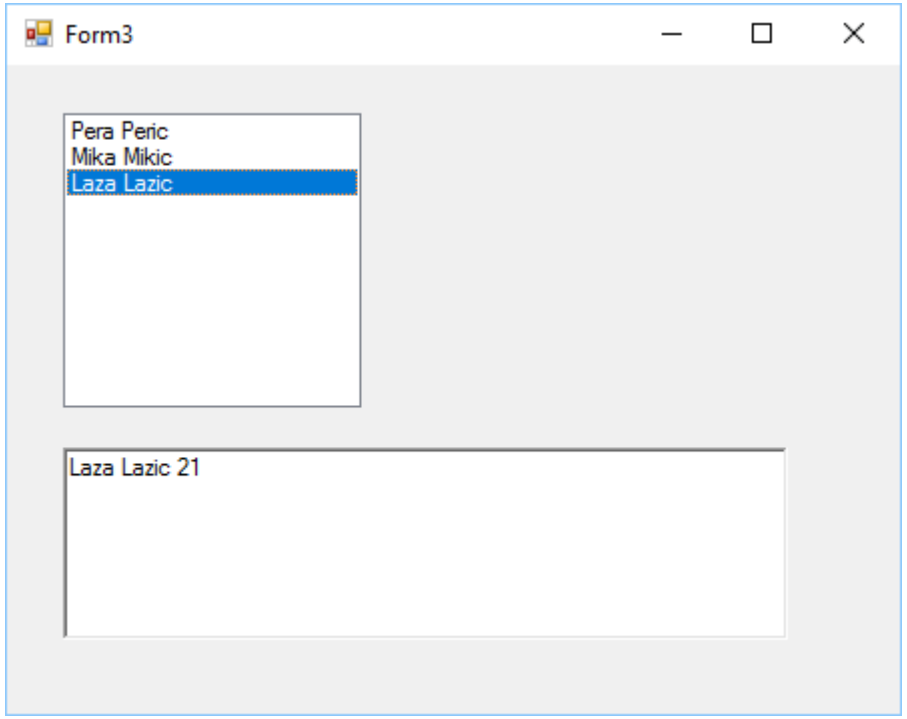




ListBox kontrola sa višestrukom selekcijom

```
private void button1_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();

    foreach (string stavka in listBox1.SelectedItems)
    {
        richTextBox1.AppendText(stavka + "\n");
    }
}
```

Korisnički interfejs



Klasa Osoba

```
class Osoba
{
    public int OsobaId { get; set; }
    public string Ime { get; set; }

    public string Prezime { get; set; }
    public int Starost { get; set; }

    public string PunoIme
    {
        get
        {
            return Ime + " " + Prezime;
        }
    }
    //public override string ToString()
    //{
    //    return Ime + " " + Prezime;
    //}
}
```

Load procedura forme

```
private void Form3_Load(object sender, EventArgs e)
{
    Osoba os1 = new Osoba {OsobaId=1, Ime="Pera", Prezime="Peric", Starost=24 };
    Osoba os2 = new Osoba { OsobaId = 2, Ime = "Mika", Prezime = "Mikic", Starost = 22 };
    Osoba os3 = new Osoba { OsobaId = 3, Ime = "Laza", Prezime = "Lazic", Starost = 21 };

    listBox1.Items.Add(os1);
    listBox1.Items.Add(os2);
    listBox1.Items.Add(os3);

    listBox1.DisplayMember = "PunoIme";
    listBox1.ValueMember = "OsobaId";
    listBox1.SelectedIndex = -1;
}
```

Čitanje selektovane stavke

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex >-1)
    {
        Osoba selektovano = (Osoba)listBox1.SelectedItem;
        richTextBox1.Text = selektovano.Ime + " " + selektovano.Prezime + " " + selektovano.Starost;
    }
    else
    {
        richTextBox1.Clear();
    }
}
```

Pitanje 1

Neka je comboBox1 instanca kontrole ComboBox. Nakon izvršavanja naredbe:
string selektovano = comboBox1.SelectedItem;

a. promenljiva selektovano dobija vrednost koja je indeks selektovane stavke
b. promenljiva selektovano dobija vrednost koja je vrednost selektovane stavke
c. generiše se izuzetak jer nije izvršeno odgovarajuće kastovanje

Odgovor: c

Pitanje 2

Stavke ComboBox kontrole su tipa:

a. string
b. object
c. double

Odgovor: b

Pitanje 3

Ako želimo da osiguramo da nijedna stavka ComboBox kontrole ne bude selektovana, onda pišemo:

a. comboBox1.SelectedIndex = 0;
b. comboBox1.SelectedIndex = -1;
c. comboBox1.SelectedItem = -1;

Odgovor: b

Pitanje 4

Neka je instanca ListBox kontrole pod nazivom listBox1 povezana sa generičkom listom objekata klase Osoba pomoću svojstva DataSource. Ako je PunoIme svojstvo klase Osoba, na koji način se vrednost ovog svojstva prikazuje na mestu stavke ListBox kontrole

a. listBox1. BindingMember = "PunoIme";
b. listBox1.DisplayMember = "PunoIme";
c. listBox1.ValueMember = "PunoIme";
d. listBox1.ValueMember = PunoIme;

Odgovor: b

Pitanje 5

Kreirana je WindowsForms aplikacija u kojoj se koristi ListBox kontrola u režimu jednostruke selekcije. U Load procedure forme napisan je sledeći kod:

```
private void Form1_Load(object sender, EventArgs e)
{
   string[] stavke = { "stavka1", "stavka2", "stavka3", "stavka4" };
   listBox1.DataSource = stavke;
}
```

Pokretanjem forme:

a. neće biti selektovana ni jedna stavka
b. biće selektovana prva stavka
c. biće selektovana poslednja stavka

Odgovor: b

Pitanje 6

Način selekcije stavki u ListBox kontroli definiše se korišćenjem:

a. svojstva SelectionMode
b. svojstva SelectionType
c. svojstva Mode

Odgovor: a

Uvod u ADO.NET

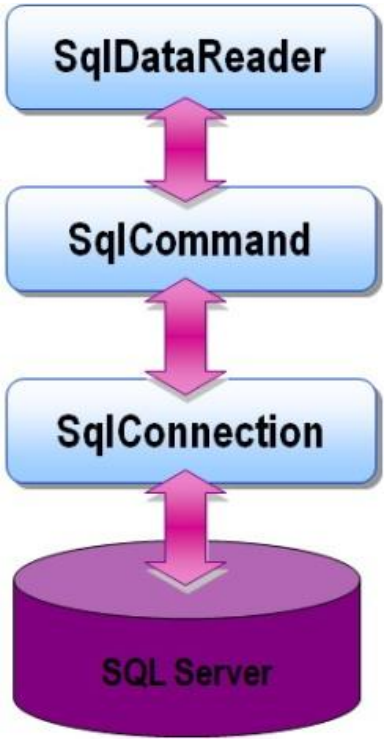
ADO.NET

- ADO.NET je skup klasa za rad sa podacima
- .NET snabdevači podataka su klase koje obezbeđuju mogućnost konektovanja na izvor podataka
  - SQL Server snabdevač podataka
  - OLE DB snabdevač podataka
  - ostali snabdevači podataka
- Prostori imena za rad sa podacima su
  - System.Data
  - System.Data.SqlClient
  - System.Data.OleDb
  - System.Data.SqlTypes
  - System.Xml

Konektovani scenario

- Resursi se uzimaju sa servera sve dok se konekcija ne zatvori
- Korisnik je konstantno povezan na izvor podataka
- Podaci su ažurni
- Konkurentnost se lakše kontroliše
- Mora postojati konstantna mrežna konekacija

Korišćenje ADO.NET klasa u konektovanom scenariju

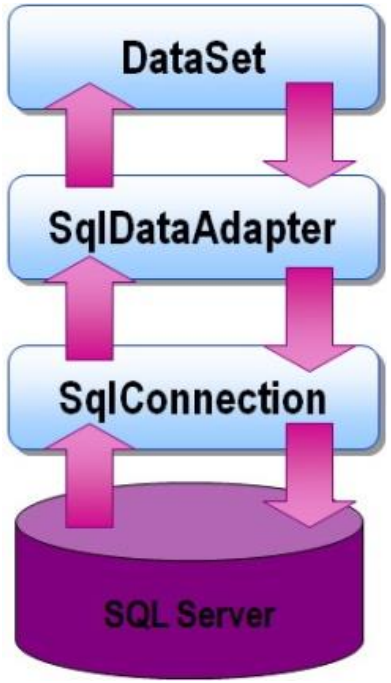


- Otvaranje konekcije
- Izvršavanje komande
- Zatvaranje konekcije

Diskonektovani scenario

- U diskonektovanom scenariju podskup podataka iz baze podataka se kopira lokalno
- Dok se korisnik nalazi u diskonektovanom radu ostali korisnici mogu da koriste konekciju
- Diskonektovani rad povećava skalabilnost aplikacije
- Podaci nisu uvek ažurni
- Kada se podacima iz lokalne kopije podataka ažurira baza može doći do konflikta

Korišćenje ADO.NET klasa u diskonektovanom scenariju



- Otvaranje konekcije
- Punjenje DataSet-a
- Zatvaranje konekcije
- Obrada podataka u DataSet-u
- Otvaranje konekcije
- Ažuriranje izvora podataka
- Zatvaranje konekcije

Konekcije

- Pre bilo kakvog rada sa bazom podataka potrebno je kreirati a zatim otvoriti konekciju
- U ADO.NET se kreira objekat klase XxxConnection
- System.Data.SqlClient.SqlConnection omogućava kreiranje konekcije na SQL Server bazu podataka
- System.Data.OleDb.OleDbConnection omogućava kreiranje konekcije na svaki izvor podataka sa pridruženim OLE DB provajderom

SqlConnection objekat

- Svaki SqlConnection objekat je izveden iz klase DbConnection koja se nalazi u prostoru imena System.Data.Common
- Klasa DbConnection predstavlja konekciju na bazu podataka

```
public abstract class DbConnection : Component, IDbConnection, IDisposable
```

Svojstva klase DbConnection

- **ConnectionString** svojstvo definiše string koji se koristi za otvaranje konekcije
- Svojstvo **DataSource** specificira ime database servera na koji se vrši konekcija
- Svojstvo **Database** specificira ime baze na koju se vrši konekcija
- Svojstvo **State** specificira stanje konekcije
- Svostvo **ConnectionTimeout** specificira vreme dozvoljeno za uspostavljenje konekcije nakon čega se generiše poruka o grešci

Metode i događaji klase DbConnection

- Metoda **Open()** otvara konekciju na bazu podataka koristeći vrednosti specificirane u konekcionom stringu
- Metoda **Close()** zatvara konekciju sa bazom podataka i oslobađa sistemske resurse
- Metoda **BeginTransaction**() otpočinje transakciju nad bazom
- Događaj **StateChange** se trigeruje kada se stanje konekcije menja (npr. iz otvorenog u zatvoreno)

Baza Magacin

```
CREATE DATABASE Magacin
GO

USE Magacin
GO

CREATE TABLE Kategorija
(
    KategorijaId int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    NazivKategorije nvarchar(50)  NOT NULL UNIQUE,
    OpisKategorije nvarchar(100)  NOT NULL
)

CREATE TABLE Proizvod
(
ProizvodId int IDENTITY(1,1) NOT NULL PRIMARY KEY,
KategorijaId int NOT NULL FOREIGN KEY REFERENCES Kategorija(KategorijaId),
NazivProizvoda nvarchar(50)  NOT NULL,
Cena decimal(12,3) NOT NULL,
KolicinaNaLageru int NOT NULL,
)
```

Konekcija kod Windows autentifikacije

```
using System.Data.SqlClient;
```

```
SqlConnection con = new SqlConnection();
con.ConnectionString = "konekcioni string";

SqlConnection con = new SqlConnection("konekcioni string");
```

Windows autentifikacija:

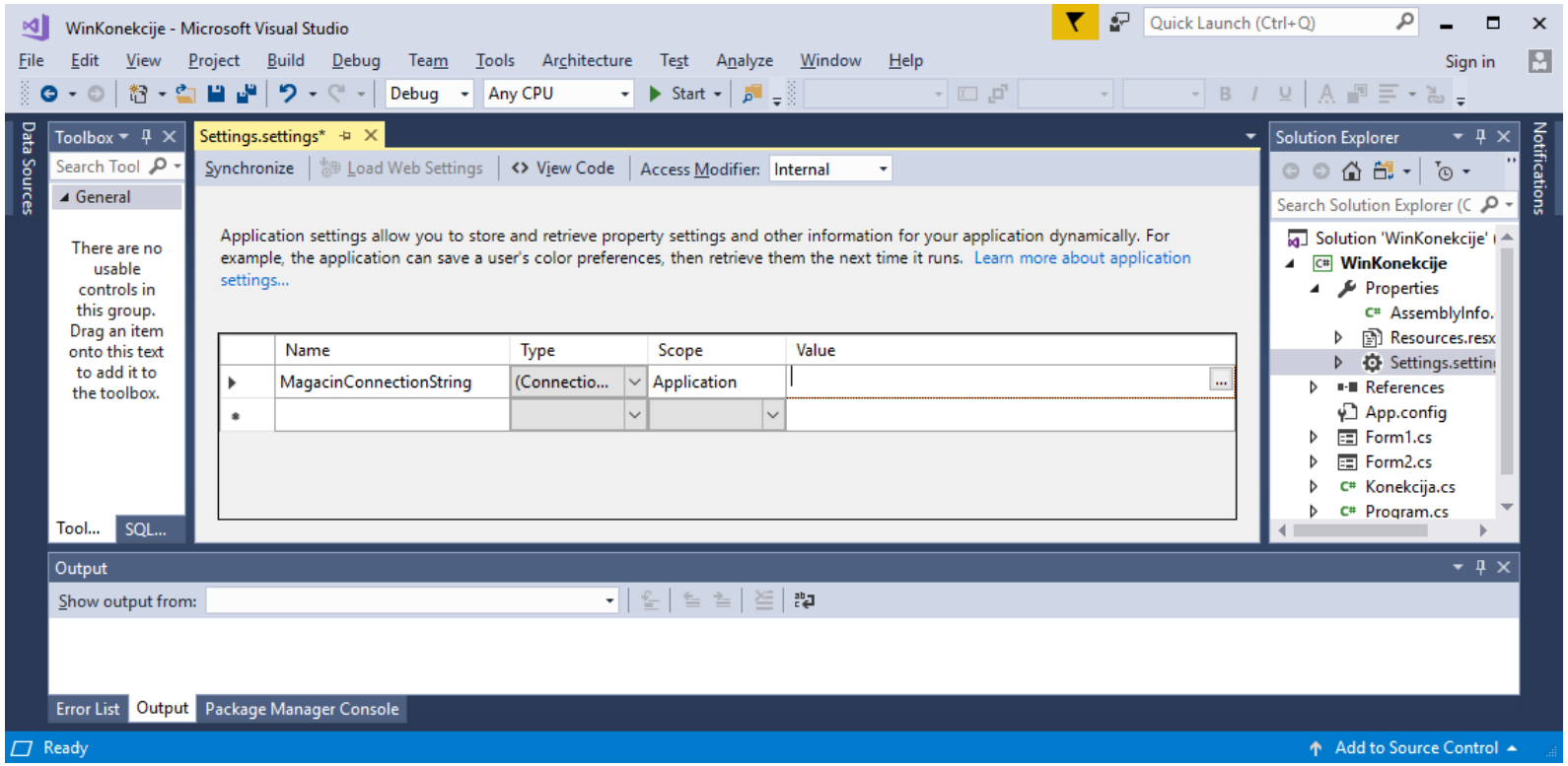
```
string konekcioniString = @"Data Source=(local)\SqlExpress;
Initial Catalog=Magacin;Integrated Security=true";
```

Data Source - ime SQL server-a
Initial Catalog – ime baze sa kojom se radi

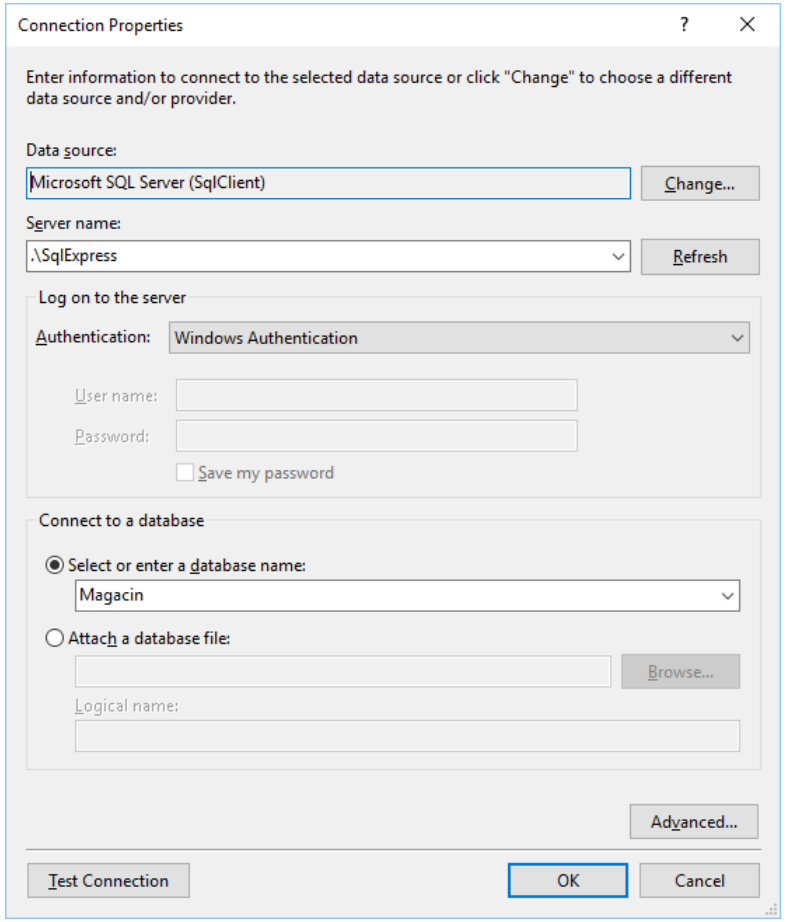
Konekcija kod SQLServer autentifikacije

```
//SQL Server autentifikacija
string konekcioniString = @"Data Source=(local)\SqlExpress;
Initial Catalog=Magacin;User ID=sa;Password=****";
```

Upis konekcionog stringa u aplikaciona setovanja -1



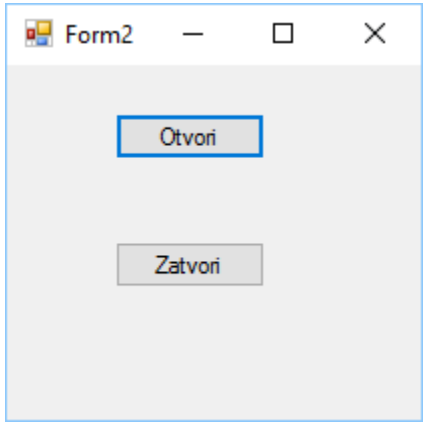
Upis konekcionog stringa u aplikaciona setovanja -1



Kreiranje konekcije

```
static class Konekcija
{
    public static string cnnMagacin =
Properties.Settings.Default.MagacinConnectionString;
}
```

Primer uspostavljanja konekcije sa bazom



Napomena: primer ilustruje samo princip otvaranja i zatvaranja konekcije.
Vreme trajanja otvorene konekcije ne sme da zavisi od želje korisnika.

Otvaranje konekcije

```
private SqlConnection konekcija = new SqlConnection(Konekcija.cnnMagacin);
```

```
private void buttonOtvori_Click(object sender, EventArgs e)
{
    if (konekcija.State == ConnectionState.Open)
    {
        MessageBox.Show("Konekcija je vec otvorena", "Poruka");
    }
    else
    {
        konekcija.Open();
        MessageBox.Show(konekcija.State.ToString(), "Stanje konekcije");
    }
}
```

Zatvaranje konekcije

```
private void buttonZatvori_Click(object sender, EventArgs e)
{
    if (konekcija.State == ConnectionState.Closed)
    {
        MessageBox.Show("Konekcija je vec zatvorena");
    }
    else
    {
        konekcija.Close();
        MessageBox.Show(konekcija.State.ToString(), "Stanje konekcije");
    }
}
```

Pitanje 1

Pri konektovanom scenariju

a. resursi se uzimaju a servera kada se konekcija zatvori
b. resursi se uzimaju sa servera dok je konekcija otvorena
c. kreira se lokalna kopija podataka iz baze

Odgovor: b

Pitanje 2

Za uspostavljanje konekcije na SQl Server bazu podataka koristi se objekat klase

a. SqlConnection
b. OleDbConnection
c. SqlServerConnection

Odgovor: a

Pitanje 3

Atribut Data Source u konekcionom stringu definiše :

a. bazu podataka sa kojom se uspostavlja konekcija
b. tabelu u bazi sa kojom se uspostavlja konekcija
c. database server sa kojim se uspostavlja konekcija

Odgovor: c

Pitanje 4

Pri diskonektovanom scenariju:

a. kreira se lokalna kopija podataka iz baze
b. korisnik je stalno povezan na izvor podataka

Odgovor: a

Pitanje 5

Da bi se koristio SQL Server.NET snabdevač podataka potrebno je ulkjučiti prostor imena:

a. System.Data.SqlTypes
b. System.Data.SqlClient
c. System.DataSqlServer

Odgovor: b