

Rad sa uskladištenim
procedurama

Baza KorisnikDb

```
CREATE DATABASE KorisnikDb
GO

USE KorisnikDb
GO

CREATE TABLE Korisnik
(
KorisnikId int IDENTITY(1,1) NOT NULL PRIMARY KEY,
Ime nvarchar(30) NOT NULL,
Prezime nvarchar(30) NOT NULL,
Email varchar(50) NOT NULL
CONSTRAINT UQ_Email UNIQUE
CONSTRAINT CHK_Email CHECK (Email LIKE '_%@_%._%'),
Lozinka varchar(15) NOT NULL
)
```

Procedura UbaciKorisnika

```
GO
CREATE PROC UbaciKorisnika
(
@Ime nvarchar(30),
@Prezime nvarchar(50),
>Email nvarchar(50),
@Lozinka nvarchar(50),
@KorisnikId int OUTPUT
)
AS
INSERT INTO Korisnik ( Ime,Prezime, Email, Lozinka )
VALUES ( @Ime, @Prezime, @Email, @Lozinka );
SELECT @KorisnikID = @@IDENTITY;
GO
```

Testiranje procedure

Execute Procedure - [dbo].[UbaciKorisnika]

Select a page: General

Script Help

Parameter	Data Type	Output Parameter	Pass Null Value	Value
@Ime	nvarchar(30)	No	☐	Marko
@Prezime	nvarchar(50)	No	☐	Markovic
@Email	nvarchar(50)	No	☐	marko@gmail.com
@Lozinka	nvarchar(50)	No	☐	marko123
@KorisnikId	int	Yes	☐	

Connection

Server: GORAN-HP

Connection: GORAN-HP\goran

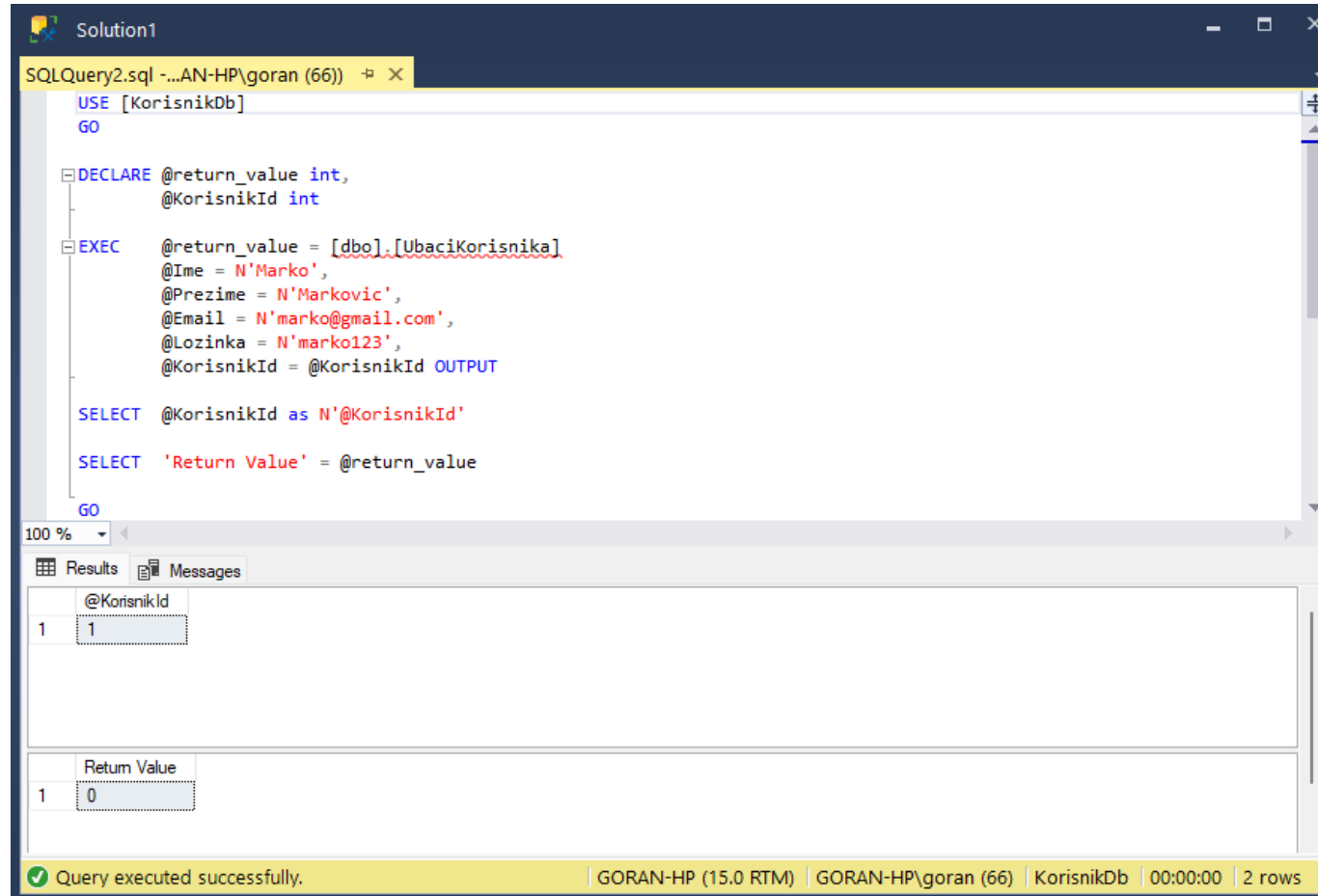
[View connection properties](#)

Progress

Ready

OK Cancel

Testiranje procedure



```
USE [KorisnikDb]
GO

DECLARE @return_value int,
        @KorisnikId int

EXEC @return_value = [dbo].[UbaciKorisnika]
    @Ime = N'Marko',
    @Prezime = N'Markovic',
    @Email = N'marko@gmail.com',
    @Lozinka = N'marko123',
    @KorisnikId = @KorisnikId OUTPUT

SELECT @KorisnikId as N'@KorisnikId'

SELECT 'Return Value' = @return_value

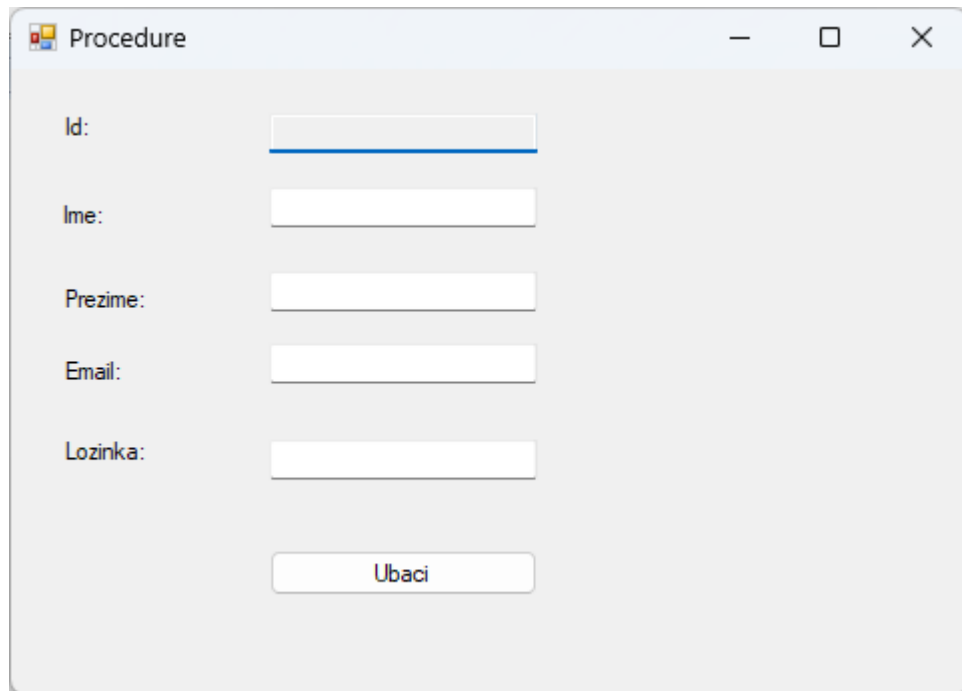
GO
```

	@KorisnikId
1	1

	Return Value
1	0

Query executed successfully. | GORAN-HP (15.0 RTM) | GORAN-HP\goran (66) | KorisnikDb | 00:00:00 | 2 rows

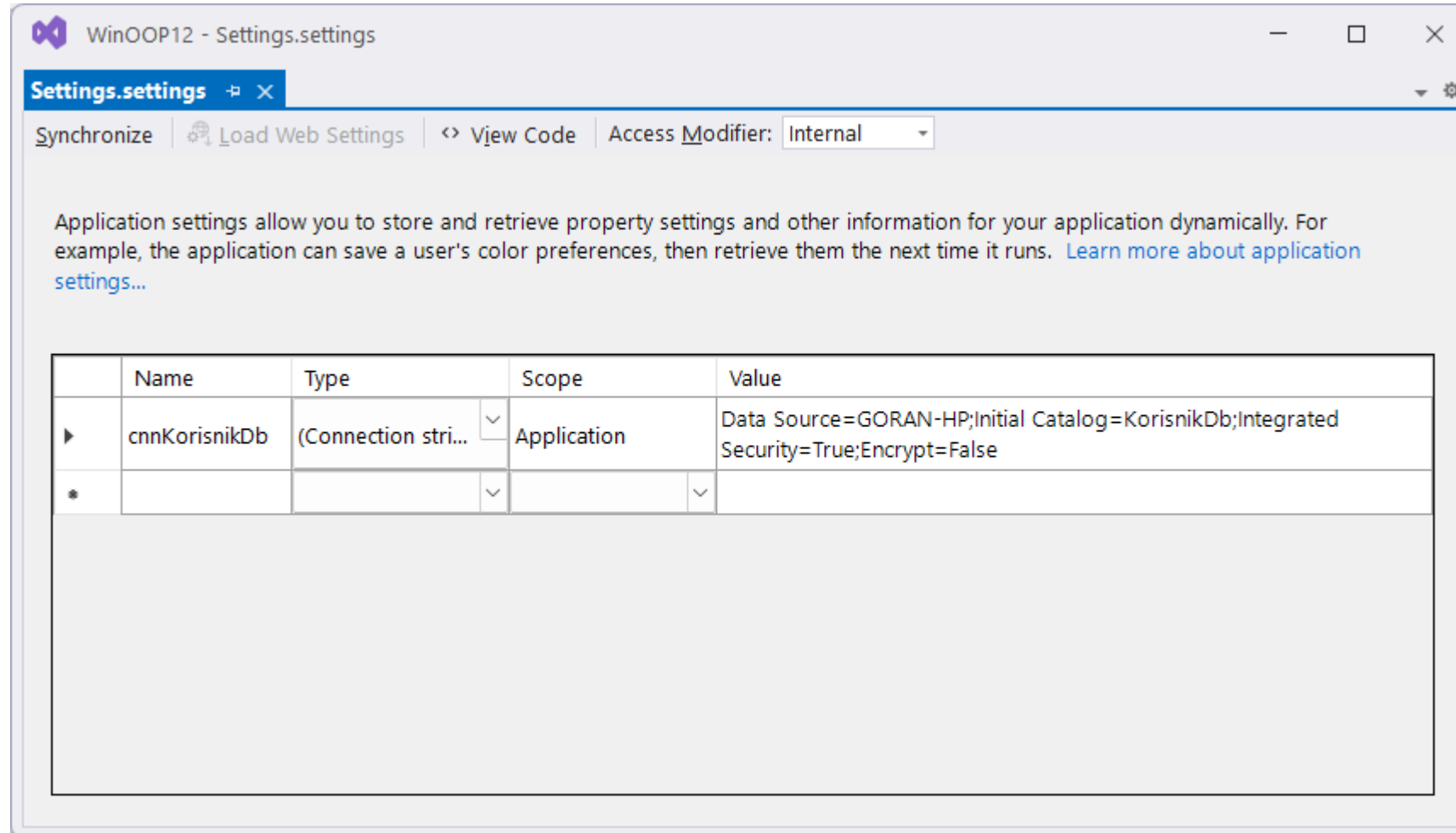
Korisnički interfejs



A screenshot of a user registration form window titled "Procedure". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The form contains five input fields for user information, each preceded by a label: "Id:", "Ime:", "Prezime:", "Email:", and "Lozinka:". The "Id:" field is currently selected, indicated by a blue underline. Below the input fields is a button labeled "Ubaci".

Id:	
Ime:	
Prezime:	
Email:	
Lozinka:	

Upis konekcionog stringa u aplikaciona setovanja



The screenshot shows a web application window titled "WinOOP12 - Settings.settings". The interface includes a toolbar with "Synchronize", "Load Web Settings", "View Code", and an "Access Modifier" dropdown set to "Internal". Below the toolbar, a text block explains that application settings allow storing and retrieving property settings dynamically, with a link to "Learn more about application settings...". A table displays the current settings, with one entry for "cnnKorisnikDb" of type "(Connection stri..." and scope "Application", containing a detailed data source string.

	Name	Type	Scope	Value
▶	cnnKorisnikDb	(Connection stri...	Application	Data Source=GORAN-HP;Initial Catalog=KorisnikDb;Integrated Security=True;Encrypt=False
*				

Klasa Konekcija

```
internal class Konekcija
{
    public static string CnnKorisnikDb =>
        Properties.Settings.Default.cnnKorisnikDb;
}
```

Entitetska klasa Korisnik

```
public class Korisnik
{
    public int KorisnikId { get; set; }

    public string Ime { get; set; }

    public string Prezime { get; set; }

    public string Email { get; set; }

    public string Lozinka { get; set; }
}
```

Metoda UbaciKorisnika

```
public static int UbaciKorisnika(Korisnik k)
{
    try
    {
        using (SqlConnection konekcija = new SqlConnection(Konekcija.CnnKorisnikDb))
        using (SqlCommand komanda = new SqlCommand("UbaciKorisnika", konekcija))
        {
            komanda.CommandType = CommandType.StoredProcedure;

            komanda.Parameters.AddWithValue("@Ime", k.Ime);
            komanda.Parameters.AddWithValue("@Prezime", k.Prezime);
            komanda.Parameters.AddWithValue("@Email", k.Email);
            komanda.Parameters.AddWithValue("@Lozinka", k.Lozinka);

            SqlParameter IdParametar = new SqlParameter("@KorisnikId", SqlDbType.Int);
            IdParametar.Direction = ParameterDirection.Output;
            komanda.Parameters.Add(IdParametar);

            konekcija.Open();
            komanda.ExecuteNonQuery();

            int id = (int)IdParametar.Value;
            return id;
        }
    }
    catch (Exception)
    {
        return -1;
    }
}
```

Validacija korisničkog interfejsa

```
private bool Validacija()
{
    if (string.IsNullOrEmpty(textIme.Text))
    {
        MessageBox.Show("Niste uneli ime");
        textIme.Focus();
        return false;
    }

    if (string.IsNullOrEmpty(textPrezime.Text))
    {
        MessageBox.Show("Niste uneli prezime");
        textPrezime.Focus();
        return false;
    }

    if (string.IsNullOrEmpty(textEmail.Text))
    {
        MessageBox.Show("Niste uneli email");
        textEmail.Focus();
        return false;
    }

    if (string.IsNullOrEmpty(textLozinka.Text))
    {
        MessageBox.Show("Niste uneli lozinku");
        textLozinka.Focus();
        return false;
    }

    return true;
}
```

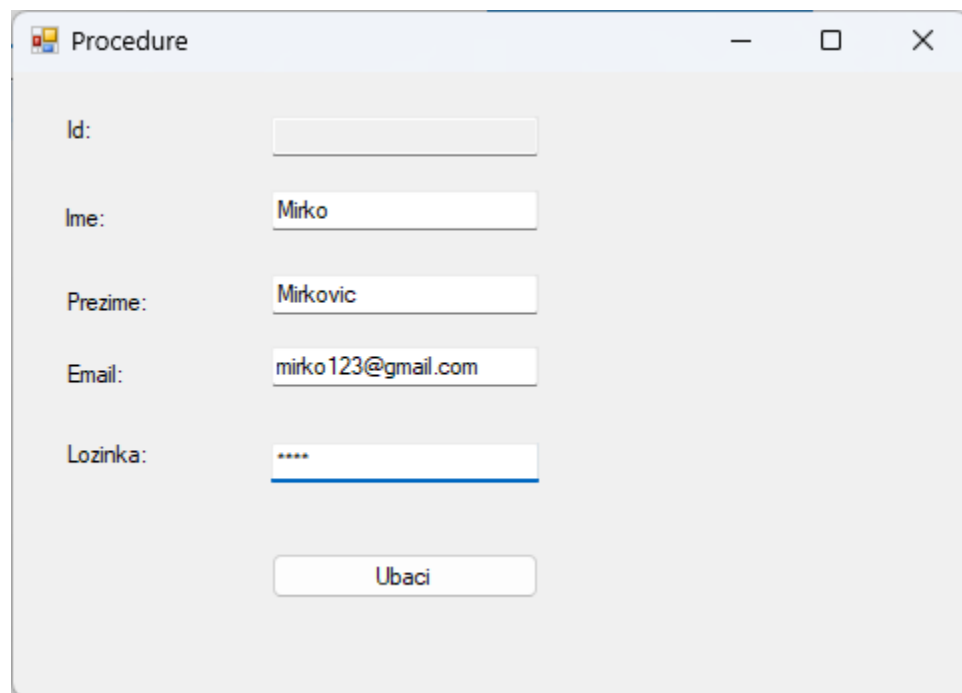
Unos podataka

```
private void button1_Click(object sender, EventArgs e)
{
    if (Validacija())
    {
        Korisnik k1 = new Korisnik
        {
            Ime = textIme.Text,
            Prezime = textPrezime.Text,
            Email = textEmail.Text,
            Lozinka = textLozinka.Text
        };

        int id = UbaciKorisnika(k1);

        if (id == -1)
        {
            MessageBox.Show("Greska pri unosu");
        }
        else
        {
            MessageBox.Show("Ubacen novi korisnik Id: " + id);
        }
    }
}
```

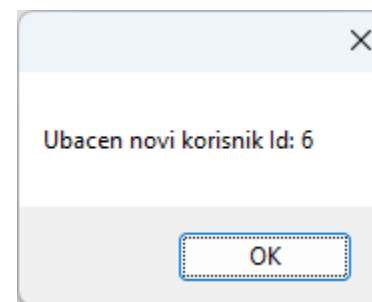
Unos podataka



A screenshot of a software window titled "Procedure". It contains a form for adding a new user. The form has five input fields: "Id:" (empty), "Ime:" (containing "Mirko"), "Prezime:" (containing "Mirkovic"), "Email:" (containing "mirko123@gmail.com"), and "Lozinka:" (containing four asterisks). Below the fields is a button labeled "Ubaci".

Id:	
Ime:	Mirko
Prezime:	Mirkovic
Email:	mirko123@gmail.com
Lozinka:	****

Ubaci

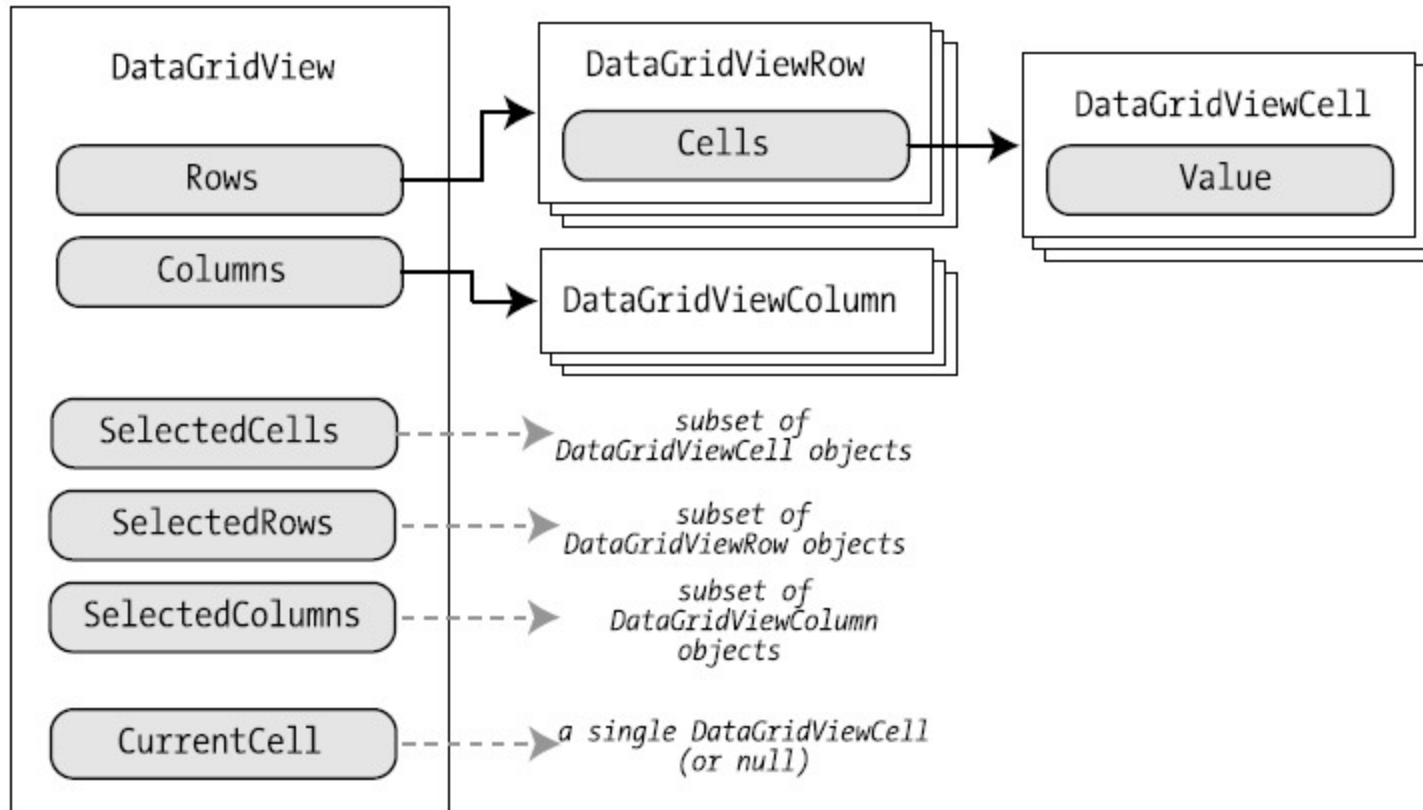


Kontrolle DataGridView i ListView

Uvod u DataGridView

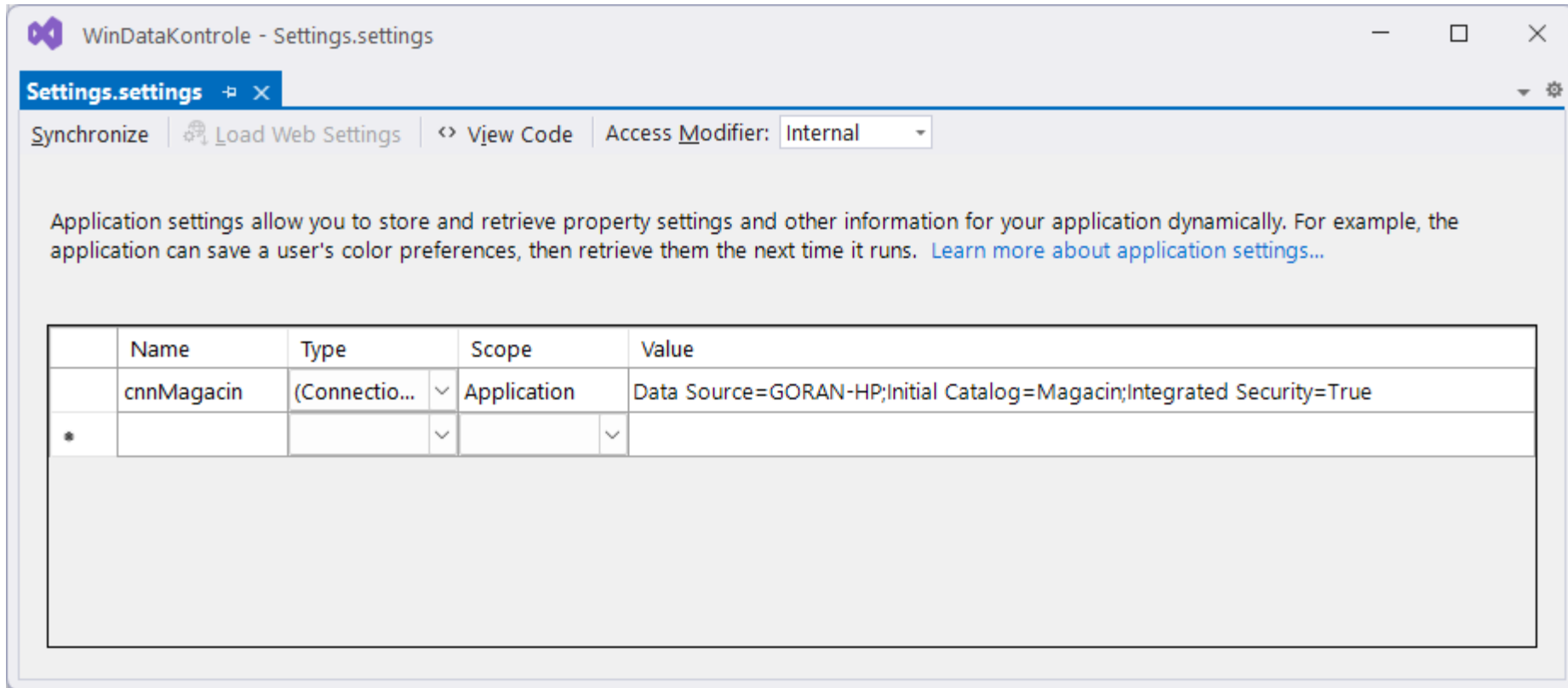
- **DataGridView** je moćna i fleksibilna kontrola, jednostavna za korišćenje, namenjena prikazu tabelarnih podataka
- Može se povezati sa kolekcijom objekata korišćenjem svojstva **DataSource**
- DataGridView automatski kreira kolone za svako javno svojstvo iz izvora podataka
- Svojstvo **Columns** predstavlja kolekciju objekata tipa **DataGridViewColumn**
- Svojstvo **Rows** predstavlja kolekciju objekata tipa **DataGridViewRow**
- Objekat DataGridViewRow sadrži svojstvo **Cells**, koje predstavlja kolekciju objekata tipa **DataGridViewCell**

Šematski prikaz strukture DataGridView kontrole



Pristup vrednosti ćelije vrši se preko:
`dataGridView.Rows[i].Cells[j].Value`

Setovanja aplikacije



WinDataKontrol - Settings.settings

Settings.settings

Synchronize Load Web Settings View Code Access Modifier: Internal

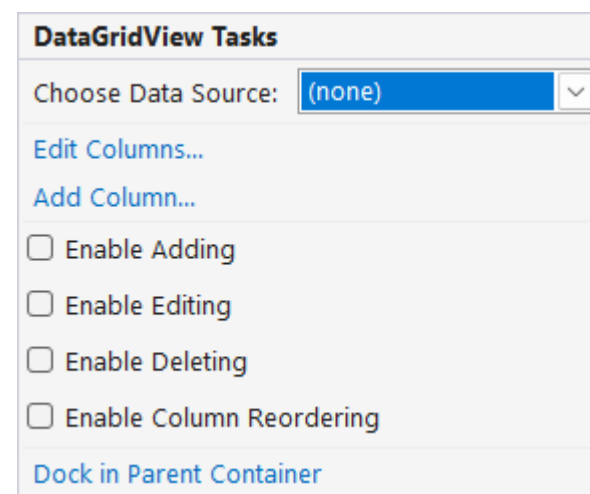
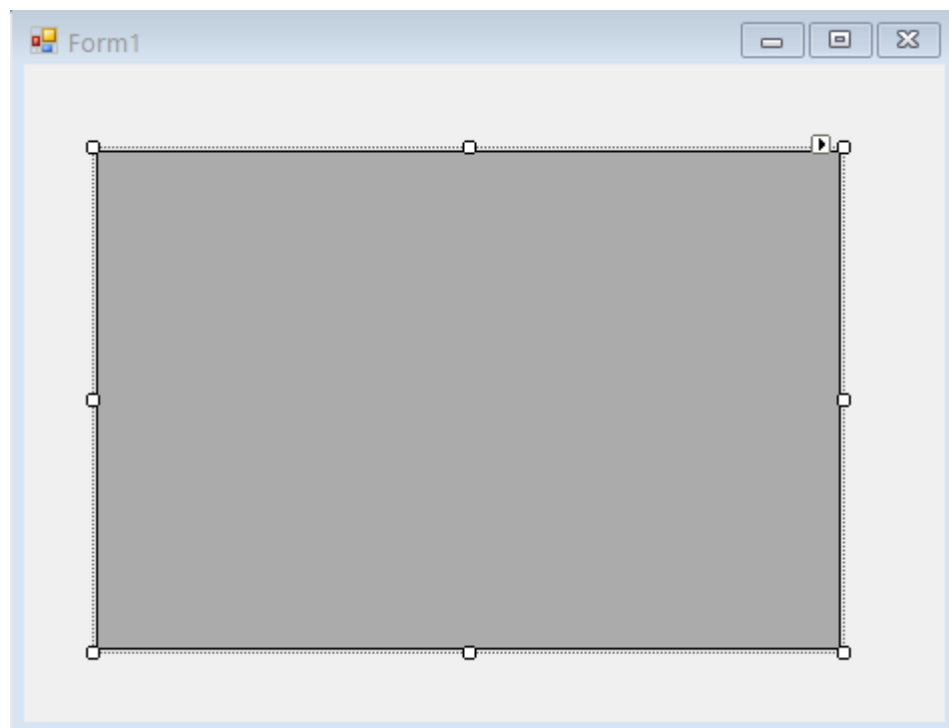
Application settings allow you to store and retrieve property settings and other information for your application dynamically. For example, the application can save a user's color preferences, then retrieve them the next time it runs. [Learn more about application settings...](#)

	Name	Type	Scope	Value
	cnnMagacin	(Connectio...	Application	Data Source=GORAN-HP;Initial Catalog=Magacin;Integrated Security=True
*				

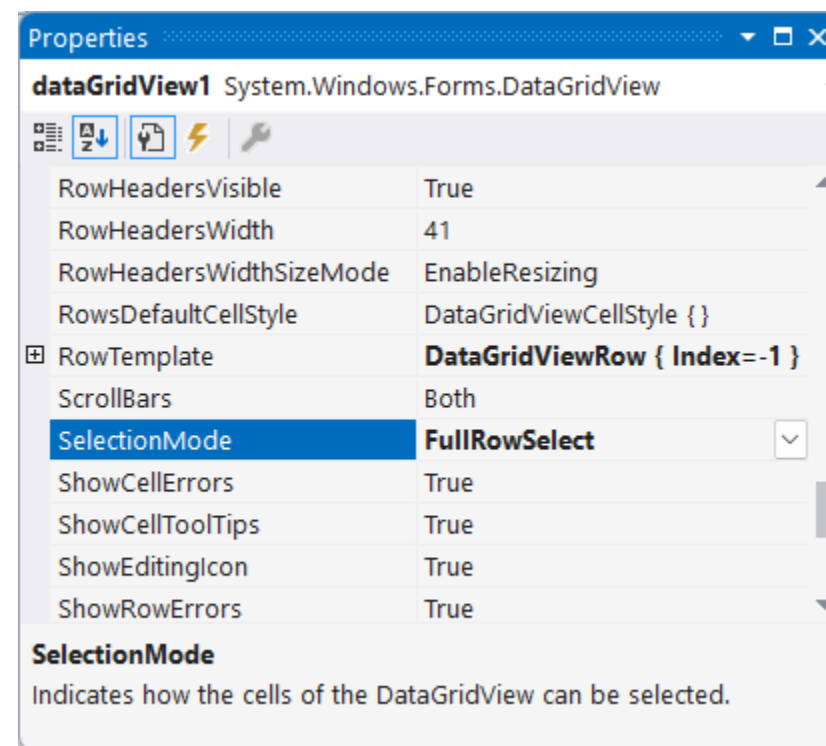
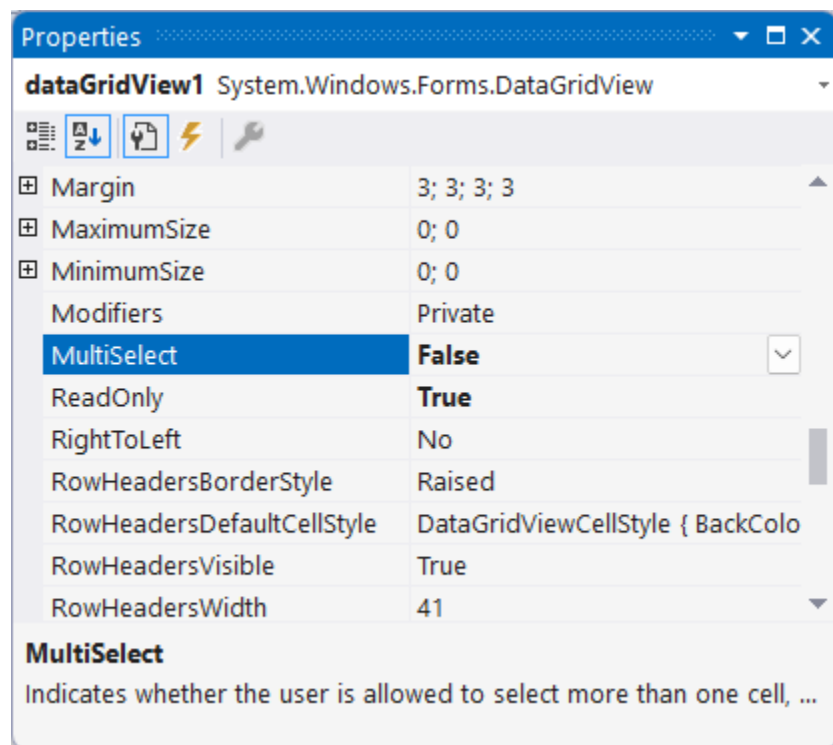
Klasa Konekcija

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

Poděšavanje DataGridView kontrole



Podešavanje DataGridView kontrole



Svojstva i metode DataGridView kontrole

- **DataSource** - koristi se za povezivanje DataGridView kontrole sa izvorom podataka (lista objekata, DataTable, BindingSource)
- **SelectedRows** daje kolekciju trenutno selektovanih redova u DataGridView kontroli. Koristi se kada je SelectionMode podešen na selekciju celog reda
- **ClearSelection()** metoda uklanja selekciju sa svih redova i ćelija u DataGridView kontroli
- **SelectionChanged** događaj se aktivira svaki put kada se promeni selekcija u DataGridView kontroli
- **Focused** je logičko svojstvo koje označava da li DataGridView kontrola trenutno ima fokus

Entitetska klasa Proizvod

```
public class Proizvod
{
    public int ProizvodId { get; set; }
    public int KategorijaId { get; set; }
    public string Naziv { get; set; }
    public decimal Cena { get; set; }
    public string Opis { get; set; }

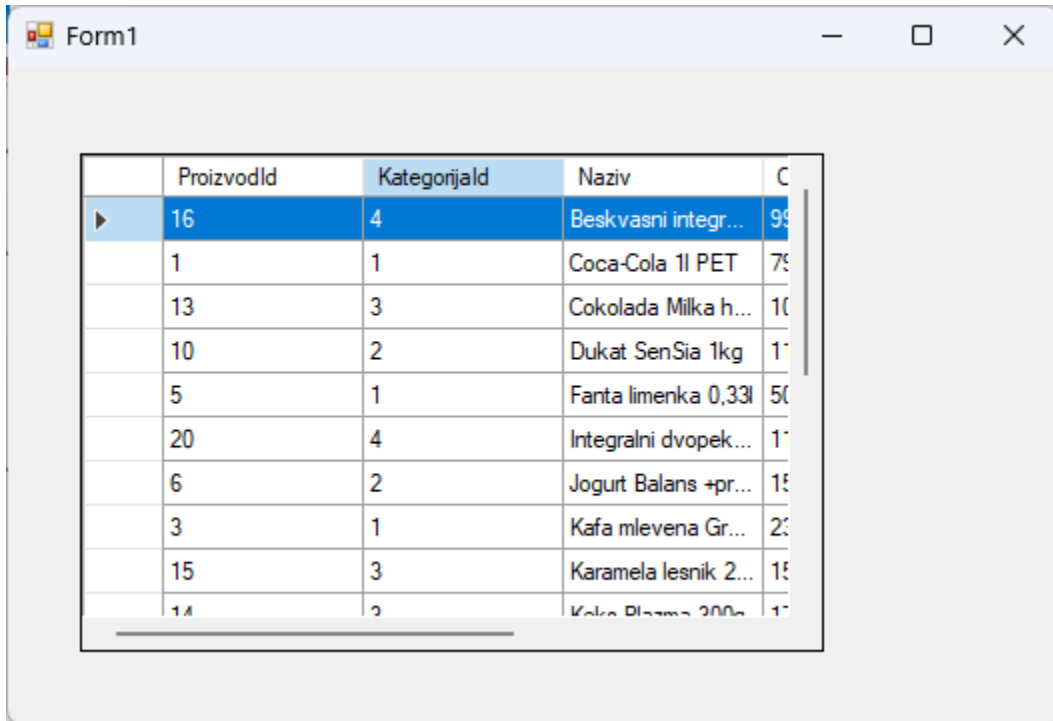
    public override string ToString()
    {
        return Naziv;
    }
}
```

Metoda VратиProizvode() statička klasa ProizvodDal

```
public static List<Proizvod> VратиProizvode()
{
    List<Proizvod> listaProizvoda = new List<Proizvod>();
    using (SqlConnection konekcija = new SqlConnection(Konekcija.CnnMagacin))
    using (SqlCommand komanda = new SqlCommand("SELECT * FROM Proizvod ORDER BY Naziv", konekcija))
    {
        try
        {
            konekcija.Open();
            using (SqlDataReader dr = komanda.ExecuteReader())
            {
                while (dr.Read())
                {
                    Proizvod p1 = new Proizvod
                    {
                        ProizvodId = dr.GetInt32(0),
                        KategorijaId = dr.GetInt32(1),
                        Naziv = dr.GetString(2),
                        Cena = dr.GetDecimal(3),
                        Opis = dr.IsDBNull(4) ? "" : dr.GetString(4)
                    };
                    listaProizvoda.Add(p1);
                }
            }
            return listaProizvoda;
        }
        catch (Exception)
        {
            return null;
        }
    }
}
```

Prikaz podataka u DataGridView kontroli

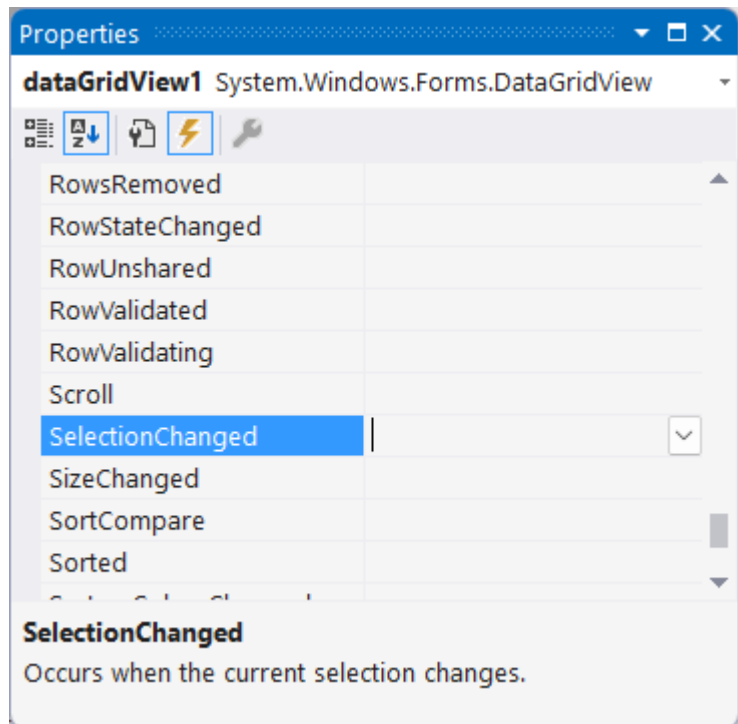
```
private void Form1_Load(object sender, EventArgs e)
{
    dataGridView1.DataSource = ProizvodDal.VratiProizvode();
}
```



The screenshot shows a Windows Form titled "Form1" with a standard Windows window border. Inside the form, there is a DataGridView control displaying a table of product data. The table has four columns: "ProizvodId", "KategorijaId", "Naziv", and "Cena". The first row is selected, highlighting the product with ID 16, category 4, name "Beskvasni integr...", and price 99. The table contains 10 rows of data in total.

ProizvodId	KategorijaId	Naziv	Cena
16	4	Beskvasni integr...	99
1	1	Coca-Cola 1l PET	79
13	3	Cokolada Milka h...	10
10	2	Dukat SenSia 1kg	11
5	1	Fanta limenka 0,33l	50
20	4	Integralni dvopek...	11
6	2	Jogurt Balans +pr...	15
3	1	Kafa mlevena Gr...	20
15	3	Karamela lesnik 2...	15
14	2	Kola Plasma 200ml	11

Događaj SelectionChanged



Event handler koji se poziva **svaki put kada se promeni selekcija** u DataGridView-u.

Brisanje selekcije pri bindovanju

```
private void Form1_Load(object sender, EventArgs e)
{
    dataGridView1.DataSource = ProizvodDal.VratiProizvode();
    dataGridView1.ClearSelection();
}
```

Iščitavanje selektovanog reda DataGridView kontrole - SelectedRows

```
private void dataGridView1_SelectionChanged(object sender, EventArgs e)
{
    if (!dataGridView1.Focused)
        return;

    if (dataGridView1.SelectedRows.Count == 0)
        return;

    DataGridViewRow selektovaniProizvod = dataGridView1.SelectedRows[0];

    StringBuilder sb = new StringBuilder();

    sb.AppendLine(selektovaniProizvod.Cells["Naziv"].Value?.ToString() ?? string.Empty);
    sb.AppendLine(selektovaniProizvod.Cells["Cena"].Value?.ToString() ?? string.Empty);
    richTextBox1.Text = sb.ToString();
}
```

Iščitavanje selektovanog reda DataGridView kontrole

The screenshot shows a Windows Form titled "Form1" with a DataGridView control. The DataGridView contains the following data:

	ProizvodId	KategorijaId	Naziv	Cena
▶	16	4	Beskrvasni integr...	99.99
	1	1	Coca-Cola 1l PET	79.99
	13	3	Cokolada Milka h...	10.99
	10	2	Dukat SenSia 1kg	11.99
	5	1	Fanta limenka 0,33l	50.99
	20	4	Integralni dvopek...	11.99

Below the DataGridView, a text box displays the details of the selected row (row 1):

Beskrvasni integralni hleb Maxi 500g
99.99

Klasa ProizvodView

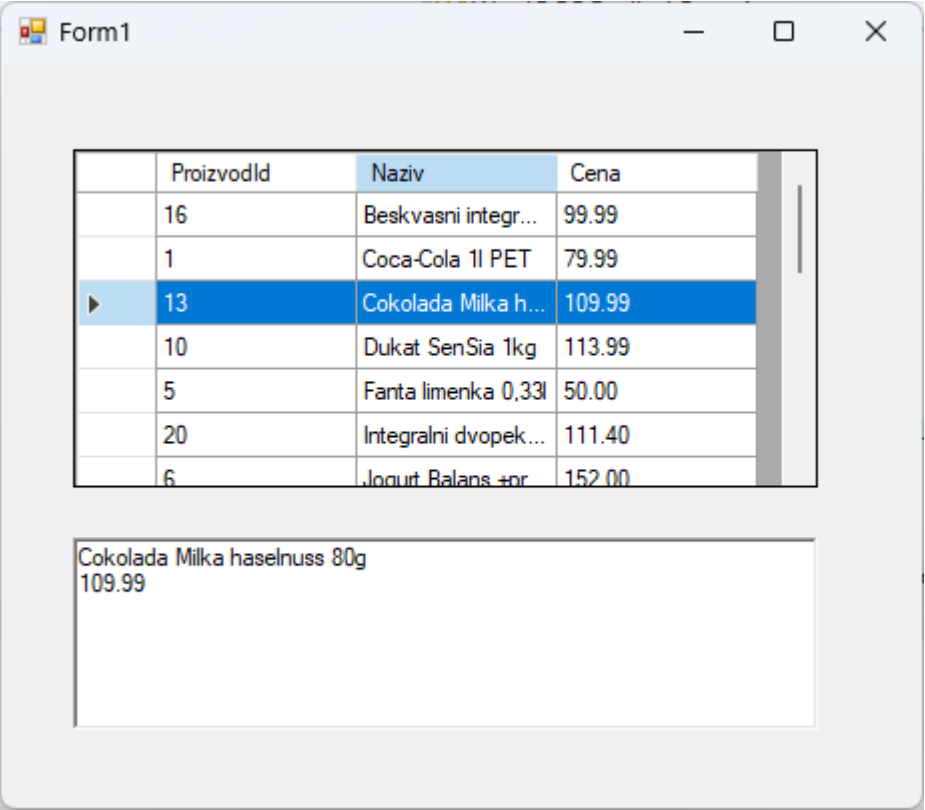
```
public class ProizvodView
{
    public int ProizvodId { get; set; }
    public string Naziv { get; set; }
    public decimal Cena { get; set; }
}
```

Modifikacija metode Form1_Load

```
private void Form1_Load(object sender, EventArgs e)
{
    //dataGridView1.DataSource = ProizvodDal.VratiProizvode();
    List<Proizvod> listaProizvoda = ProizvodDal.VratiProizvode();
    IEnumerable<ProizvodView> lista1 = listaProizvoda
        .Select(p => new ProizvodView { ProizvodId = p.ProizvodId, Naziv = p.Naziv, Cena = p.Cena });
    dataGridView1.DataSource = lista1.ToList();

    dataGridView1.ClearSelection();
}
```

Prikaz određenih kolona tabele



The screenshot shows a Windows application window titled "Form1". Inside the window, there is a table with four columns: "ProizvodId", "Naziv", and "Cena". The table contains seven rows of data. The third row, with "ProizvodId" 13 and "Naziv" "Cokolada Milka h...", is selected and highlighted in blue. Below the table, there is a text box containing the text "Cokolada Milka haselnuss 80g" and "109.99".

ProizvodId	Naziv	Cena
16	Beskrvasni integr...	99.99
1	Coca-Cola 1l PET	79.99
13	Cokolada Milka h...	109.99
10	Dukat SenSia 1kg	113.99
5	Fanta limenka 0,33l	50.00
20	Integralni dvopek...	111.40
6	Jogurt Balans +pr	152.00

Cokolada Milka haselnuss 80g
109.99

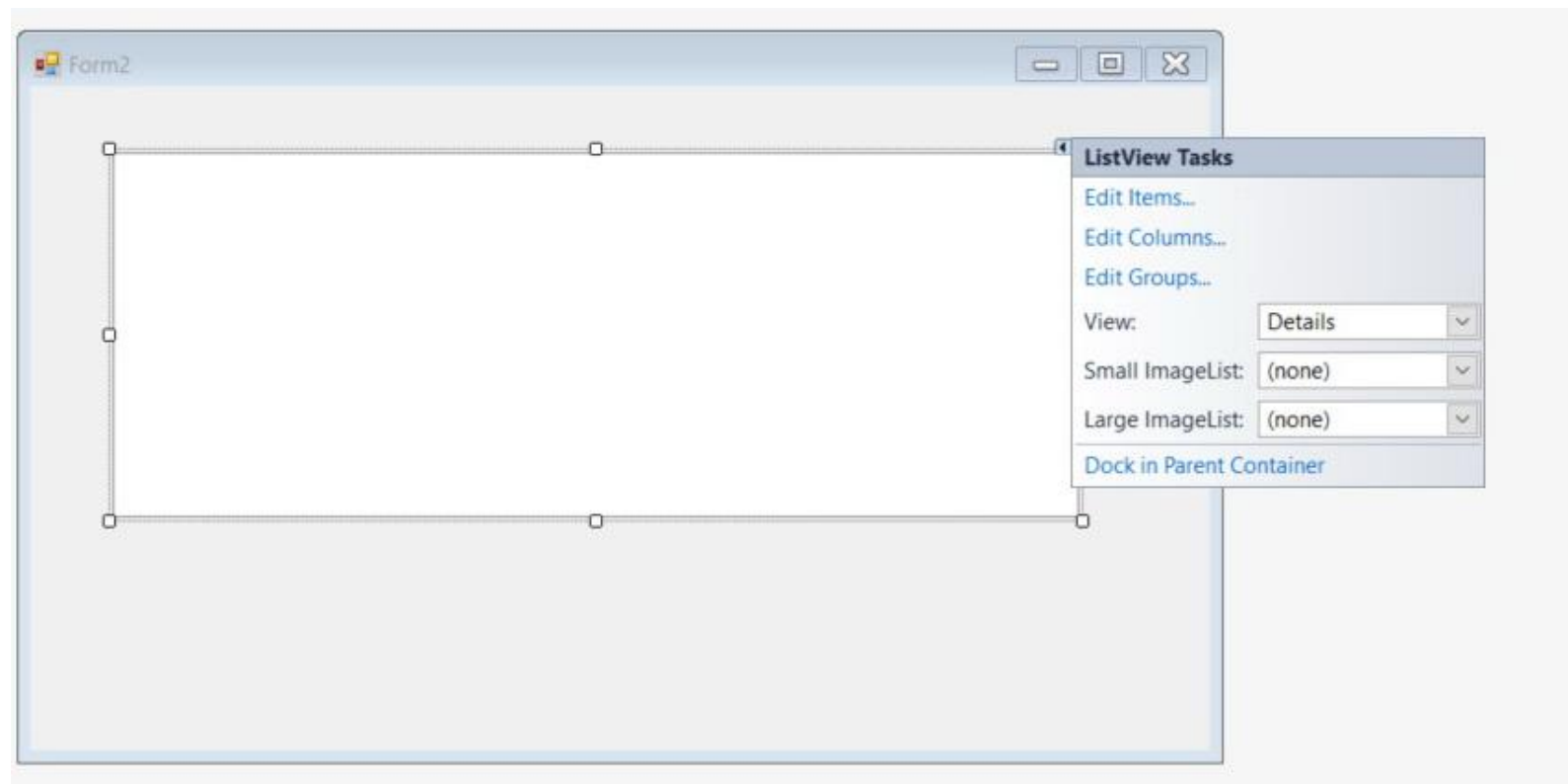
ListView kontrola

- **ListView** omogućava da se prikaže lista stavki kojima se mogu pridružiti slike kao u File Explorer-u
- Stavke se mogu prikazati u jednom od sledeća četiri režima prikaza:
 - prikaz sa velikim ikonama (Large Icon)
 - prikaz sa malim ikonama (Small Icon)
 - lista (List)
 - detaljni prikaz (Details)
- Stavke ListView kontrole su objekti klase **ListViewItem**

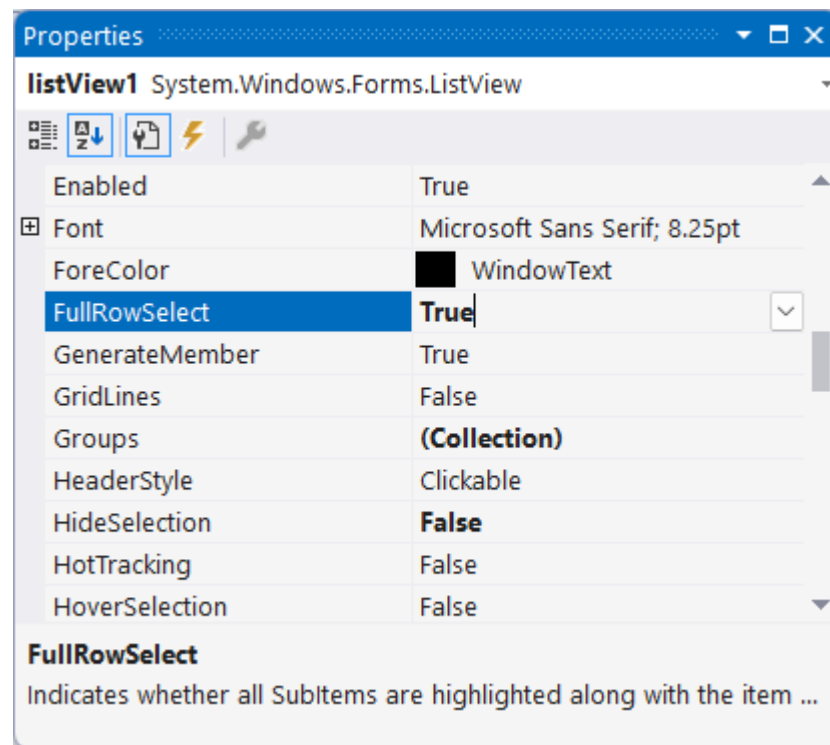
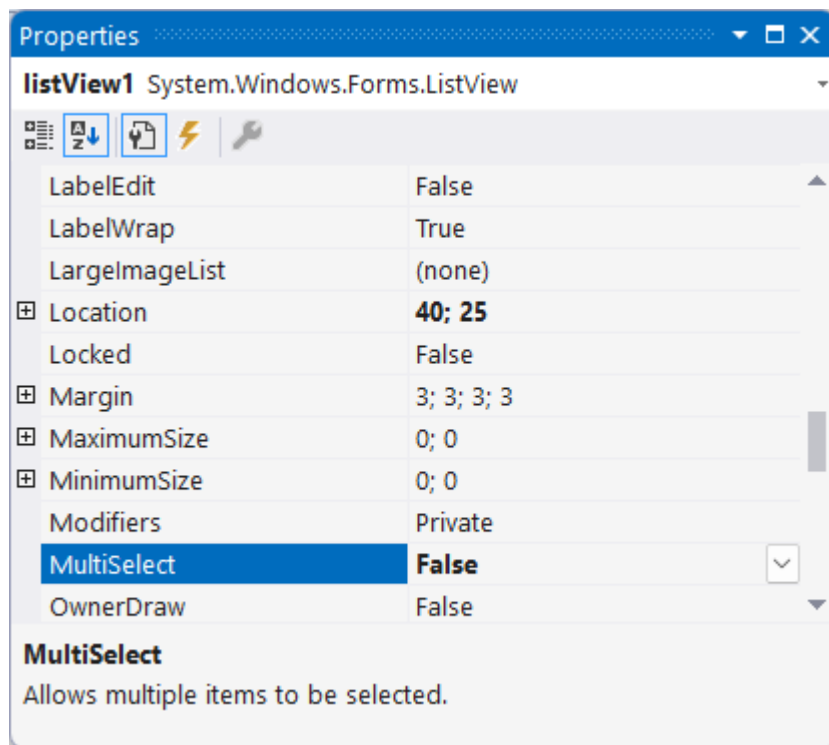
Svojstva ListView kontrole

- **Columns** – sadrži kolekciju kolona koje se prikazuju u Details režimu
- **Items** – sadrži kolekciju objekata klase **ListViewItem** koji se prikazuju u ListView kontroli
- **LargeImageList** – određuje ImageList iz koje se koriste ikone kada je svojstvo View = LargeIcon
- **SmallImageList** – određuje ImageList iz koje se koriste ikone kada je svojstvo View = SmallIcon
- **View** – određuje način prikaza ListView kontrole (LargeIcon, SmallIcon, List, Details)

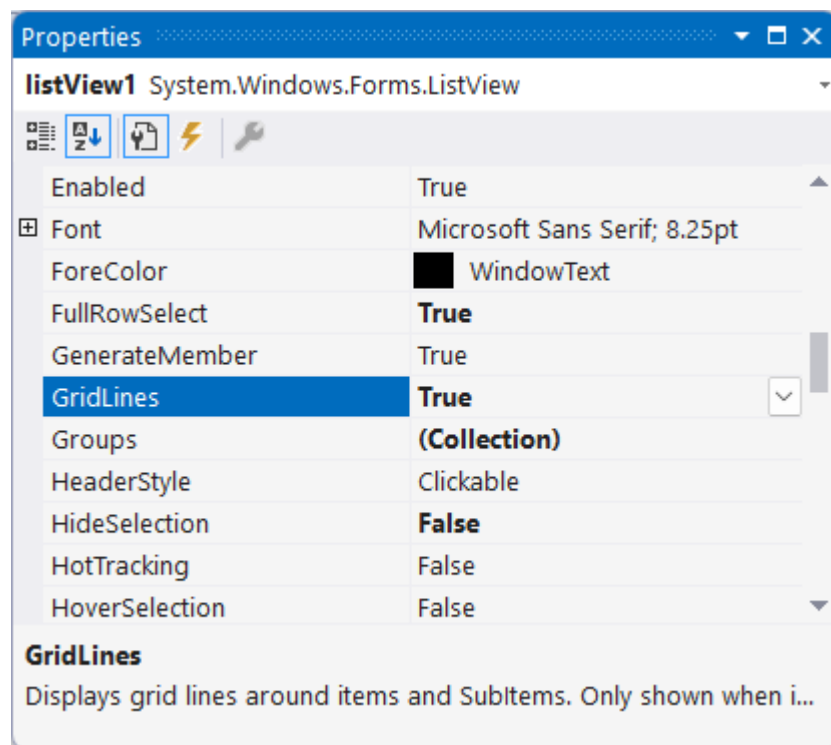
ListView kontrola u režimu detalja



Podešavanje ListView kontrole

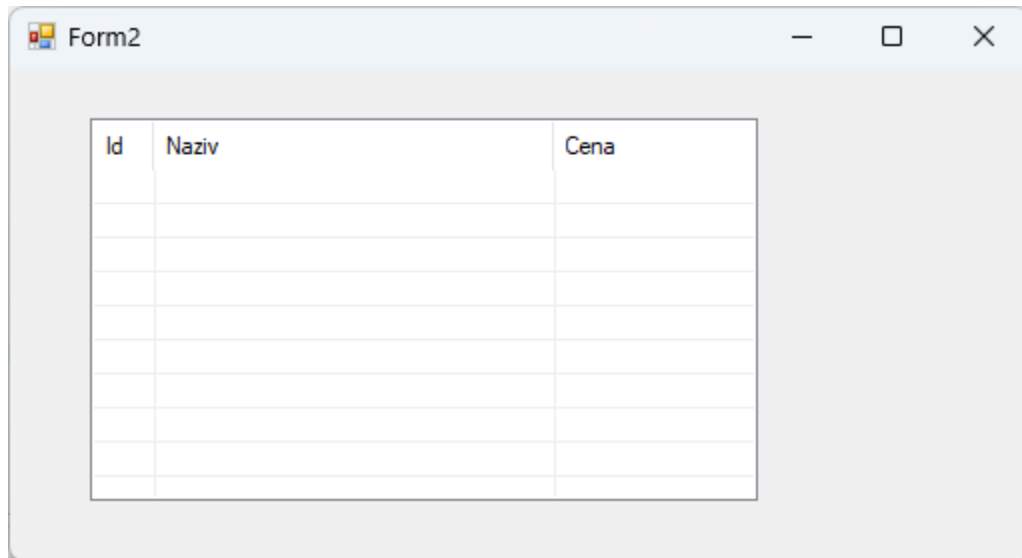


Prikaz linija grida kod ListView kontrole



Definisanje kolona ListView kontrole

```
private void Form2_Load(object sender, EventArgs e)
{
    listView1.Columns.Add("Id", 30);
    listView1.Columns.Add("Naziv", 200);
    listView1.Columns.Add("Cena", 120);
}
```



Prikaz podataka iz baze u ListView kontroli

```
private void PrikaziProizvode()
{
    listView1.Items.Clear();

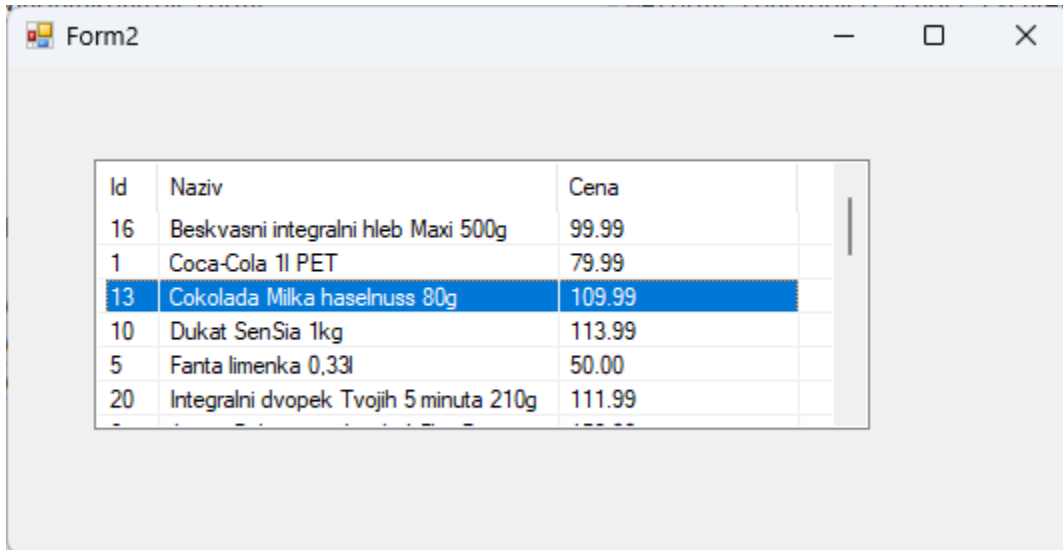
    foreach (Proizvod p in ProizvodDal.VratiProizvode())
    {
        ListViewItem lvi = new ListViewItem(p.ProizvodId.ToString());
        lvi.SubItems.Add(p.Naziv);
        lvi.SubItems.Add(p.Cena.ToString());
        listView1.Items.Add(lvi);
    }
}
```

Prva kolona u ListView-u se postavlja preko svojstva Text, dok se ostale kolone dodaju kroz kolekciju SubItems

```
string id   = lvi.SubItems[0].Text; // prva kolona
string naziv = lvi.SubItems[1].Text; // druga kolona
string cena  = lvi.SubItems[2].Text; // treća kolona
```

Load procedura forme

```
private void Form2_Load(object sender, EventArgs e)
{
    listView1.Columns.Add("Id", 30);
    listView1.Columns.Add("Naziv", 200);
    listView1.Columns.Add("Cena", 120);
    PrikaziProizvode();
}
```



Id	Naziv	Cena
16	Beskrvasni integralni hleb Maxi 500g	99.99
1	Coca-Cola 1l PET	79.99
13	Cokolada Milka haselnuss 80g	109.99
10	Dukat SenSia 1kg	113.99
5	Fanta limenka 0,33l	50.00
20	Integralni dvopek Tvojih 5 minuta 210g	111.99

Iščitavanje selektovane stavke

```
private void listView1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        ListViewItem sel = listView1.SelectedItems[0];

        string id = sel.Text;
        string naziv = sel.SubItems[1].Text;
        string cena = sel.SubItems[2].Text;

        richTextBox1.Text = $"Id: {id} | Naziv: {naziv} | Cena: {cena}";
    }
}
```

Iščitavanje selektovane stavke

Form2

Id	Naziv	Cena
16	Beskvasni integralni hleb Maxi 500g	99.99
1	Coca-Cola 1l PET	79.99
13	Cokolada Milka haselnuss 80g	109.99
10	Dukat SenSia 1kg	113.99
5	Fanta limenka 0,33l	50.00
20	Integralni dvopek Tvojih 5 minuta 210g	111.40
6	Jogurt Balans +probiotik 1.5kg Pet	152.00
3	Kafa mlevena Grand Gold 200g	239.99
15	Karamela limenka 200ml	152.99

Id: 10 | Naziv: Dukat SenSia 1kg | Cena: 113.99

Pitanje 1

Svojstvo Columns objekta klase DataGridView daje:

- a. Kolekciju objekata klase DataColumn
- b. Kolekciju objekata klase GridViewColumn
- c. Kolekciju objekata klase DataGridViewColumn

Odgovor: c

Pitanje 2

Tip selekcije u DataGridView kontroli definiše se njenim svojstvom:

- a. SelectionMode
- b. SelectionType
- c. SelectionUnit

Odgovor: a

Pitanje 3

DataGridView kontrola čije je svojstvo Name dataDataGridView1, ima svojstvo MultiSelect postavljeno na vrednost false i svojstvo SelectionMode na vrednost FullRowSelect.

Na koji način se pristupa selektovanom redu:

- a. `DataGridViewRow selVrsta = dataDataGridView1.SelectedRows;`
- b. `DataGridViewRow selVrsta = dataDataGridView1.SelectedRows[0];`
- c. `DataGridViewRow selVrsta = dataDataGridView1.SelectedItem;`

Odgovor: b

Pitanje 4

U kontroli ListView

- a. stavke se uvek prikazuju na isti način
- b. način prikazivanja stavki definise se svojstvom View
- c. način prikazivanja stavki definise se svojstvom Icon
- d. način prikazivanja stavki definise se svojstvom ImageMode

Odgovor: b

Pitanje 5

Stavke ListView kontrole su objekti klase:

- a. ListViewItem
- b. ListItem
- c. ListNode

Odgovor: a

Pitanje 6

Da bismo zabranili selektovanje više od jedne stavke unutar ListView kontrole:

- a. svojstvo Multiple postavimo na False
- b. svojstvo MultiSelect postavimo na False
- c. svojstvo SingleSelect postavimo na vrednost True

Odgovor: b

Pitanje 7

Koji događaj kontrole DataGridView se pokreće kada dođe do promene selekcije (reda, kolone ili ćelije)?

- a. CellClick
- b. CellValueChanged
- c. SelectionChanged

Odgovor: c

Pitanje 8

Koji događaj kontrole ListView se pokreće kada se promeni indeks selektovane stavke?

- a. SelectedItemChanged
- b. SelectedIndexChanged
- c. ItemActivate

Odgovor: b