

## Zadaci

|                                      |           |
|--------------------------------------|-----------|
| <b>Serije .....</b>                  | <b>2</b>  |
| Baza podataka .....                  | 2         |
| Izgled forme .....                   | 3         |
| Rešenje .....                        | 4         |
| <b>Umetnost .....</b>                | <b>6</b>  |
| Baza podataka .....                  | 6         |
| Izgled forme .....                   | 7         |
| Rešenje .....                        | 9         |
| <b>Bolnica.....</b>                  | <b>11</b> |
| Baza podataka .....                  | 11        |
| Izgled forme .....                   | 13        |
| Rešenje .....                        | 15        |
| <b>Igrice.....</b>                   | <b>17</b> |
| Baza podataka .....                  | 17        |
| Izgled forme .....                   | 18        |
| Rešenje .....                        | 20        |
| <b>Humanitarna organizacija.....</b> | <b>22</b> |
| Baza podataka .....                  | 22        |
| Izgled forme .....                   | 23        |
| Rešenje .....                        | 25        |

## **Serije**

### *Baza podataka*

Koristite istu bazu podataka kao i za primer Serije.

## Izgled forme



Po pokretanju forme ListView je prazan.

Klikom na neko od tri dugmeta ListView se popunjava nazivom serije i brojem sezona serija koje odgovaraju striming servisu na koji je kliknuto.

Klikom na neko od polja ListView-a podaci se pojavljuju u TextBox-ovima sa desne strane.

## Rešenje

```
using System;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakSerije
{
    public partial class Form1 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=Striming;Integrated Security=True";
        string selectHbo = "SELECT NazivSerije,BrojSezona FROM Serija WHERE StrimingServisID = 1";
        string selectNetflix = "SELECT NazivSerije,BrojSezona FROM Serija WHERE StrimingServisID = 2";
        string selectShowtime = "SELECT NazivSerije,BrojSezona FROM Serija WHERE StrimingServisID = 3";

        public Form1()
        {
            InitializeComponent();
        }

        private void btnHbo_Click(object sender, EventArgs e)
        {
            lvDrzave.Items.Clear();
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand izaberiSve = new SqlCommand(selectHbo, konekcija);
            konekcija.Open();

            SqlDataReader citac = izaberiSve.ExecuteReader();

            while (citac.Read())
            {
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                prikaz.SubItems.Add(citac[1].ToString());

                lvDrzave.Items.Add(prikaz);
            }
        }

        private void btnNetflix_Click(object sender, EventArgs e)
        {
            lvDrzave.Items.Clear();
            SqlConnection konekcija = new SqlConnection(konekcioniString);

            SqlCommand izaberiSve = new SqlCommand(selectNetflix, konekcija);

            konekcija.Open();

            SqlDataReader citac = izaberiSve.ExecuteReader();

            while (citac.Read())
            {
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                prikaz.SubItems.Add(citac[1].ToString());

                lvDrzave.Items.Add(prikaz);
            }
        }

        private void btnShowtime_Click(object sender, EventArgs e)
        {
            lvDrzave.Items.Clear();
            SqlConnection konekcija = new SqlConnection(konekcioniString);

            SqlCommand izaberiSve = new SqlCommand(selectShowtime, konekcija);

            konekcija.Open();
        }
    }
}
```

```

        SqlDataReader citac = izaberiSve.ExecuteReader();

        lvDrzave.View = View.Details;
        lvDrzave.FullRowSelect = Enabled;
        while (citac.Read())
        {
            ListViewItem prikaz = new ListViewItem(citac[0].ToString());
            prikaz.SubItems.Add(citac[1].ToString());

            lvDrzave.Items.Add(prikaz);
        }
    }

    private void lvDrzave_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (lvDrzave.SelectedItems.Count > 0)
        {
            ListViewItem umetnici = lvDrzave.SelectedItems[0];
            txtSerija.Text = umetnici.SubItems[0].Text;
            txtSezone.Text = umetnici.SubItems[1].Text;
        }
        else
        {
            txtSerija.Clear();
            txtSezone.Clear();
        }
    }
}

```

## Umetnost

### *Baza podataka*

Koristite istu bazu podataka kao i za primer Umetnost.

## Izgled forme

Umetnost

Osvajači Broj osvajača

Bukerova nagrada ▾

Prikaži

| Ime | Godina rođenja |
|-----|----------------|
|-----|----------------|

Forma se sastoji od dva taba.

Na prvom tabu (*Osvajači*) se nalazi comboBox koji je popunjen nazivima nagrada.

dataGridView se popunjava nakon izbora nagrade u comboBox-u i klika na dugme *Prikaži*.

Umetnost

Osvajači Broj osvajača

Sterijina nagrada ▾

Prikaži

|   | Ime              | Godina rođenja |
|---|------------------|----------------|
| ▶ | Petar Božovic    | 1946           |
|   | Zoran Radmilovic | 1933           |
|   | Bora Todorovic   | 1929           |
|   | Mira Banjac      | 1929           |
| * |                  |                |

U dataGridView-u se prikazuju ime i prezime u jednom polju i godina rođenja u drugom polju.

Osvajači su sortirani po godini rođenja, od najmlađeg ka najstarijem.

Na drugom tabu se nalazi numericUpDown sa početnom vrednošću 0, inkrementom (korakom povećavanja/smanjivanja) 1, i maksimalnom vrednošću 10.

Nakon izbora broja unutar numericUpDown-a i klika na dugme *Prikaži* na drugom tabu popunjava se i dataGridView drugog taba.

Umetnost

Osvajači Broj osvajača

Broj osvajača > 2 Prikaži

|   | Naziv             | Broj osvajača |
|---|-------------------|---------------|
| ► | Gremi             | 5             |
|   | Sterijina nagrada | 4             |
| * |                   |               |

U dataGridView-u drugog taba se ispisuju nazivi nagrada i broj onih koji su ih osvojili, samo za one nagrade koje imaju veći broj osvajača od broja izabranog u numericUpDown-u.



## Rešenje

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakUmetnost
{
    public partial class Form1 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=Umetnost;Integrated Security=True";
        string insertComboBox = "SELECT NagradaID, Naziv FROM Nagrada WHERE NagradaID IN (SELECT
NagradaID FROM Umetnik)";
        string insertGridView1 = "SELECT Ime + ' ' + Prezime AS Ime, YEAR(DatumRodjenja) AS [Godina
rođenja] FROM Umetnik " +
            "WHERE NagradaID = @NagradaID " +
            "ORDER BY YEAR(DatumRodjenja) DESC";
        string insertGridView2 = "SELECT Naziv, COUNT(UmetnikID) AS [Broj osvajača] FROM Nagrada, Umetnik
" +
            "WHERE Umetnik.NagradaID = Nagrada.NagradaID " +
            "GROUP BY Naziv " +
            "HAVING COUNT(UmetnikID) > @Broj" +
            " ORDER BY COUNT(UmetnikID) DESC";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlDataAdapter adapter = new SqlDataAdapter(insertComboBox, konekcija);
            DataTable podaci = new DataTable();
            adapter.Fill(podaci);

            cbxNagrada.DataSource = podaci;
            cbxNagrada.ValueMember = "NagradaID";
            cbxNagrada.DisplayMember = "Naziv";
        }

        private void btnPrikaziOsvajace_Click(object sender, EventArgs e)
        {
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand upis = new SqlCommand(insertGridView1, konekcija);

            upis.Parameters.AddWithValue("@NagradaID", cbxNagrada.SelectedValue.ToString());
            konekcija.Open();

            SqlDataAdapter dataAdapter = new SqlDataAdapter(upis);

            DataTable podaci = new DataTable();
            dataAdapter.Fill(podaci);
            dgvOsvajaci.DataSource = podaci;
            konekcija.Close();
        }

        private void btnPrikazi_Click(object sender, EventArgs e)
        {
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand upis = new SqlCommand(insertGridView2, konekcija);
            upis.Parameters.AddWithValue("@Broj", numBroj.Value);
            konekcija.Open();

            SqlDataAdapter dataAdapter = new SqlDataAdapter(upis);
```

```
DataTable podaci = new DataTable();  
dataAdapter.Fill(podaci);  
dgvNagrade.DataSource = podaci;  
konekcija.Close();
```

```
}
```

```
}
```

```
}
```

# Bolnica

## Baza podataka

```
CREATE DATABASE Bolnica
GO
USE Bolnica
GO

CREATE TABLE Odeljenje (
    OdeljenjeID INT PRIMARY KEY NOT NULL,
    Naziv NVARCHAR(30) NOT NULL,
    BrojKreveta INT
);
GO

CREATE TABLE Doktor (
    DoktorID INT PRIMARY KEY NOT NULL,
    ImeDoktora NVARCHAR(30) NOT NULL,
    PrezimeDoktora NVARCHAR(30) NOT NULL,
    OdeljenjeID INT FOREIGN KEY REFERENCES Odeljenje (OdeljenjeID)
);
GO

CREATE TABLE Pacijent (
    PacijentID INT PRIMARY KEY NOT NULL,
    ImePacijenta NVARCHAR(30) NOT NULL,
    PrezimePacijenta NVARCHAR(30) NOT NULL,
    DatumDolaska DATE NOT NULL,
    OdeljenjeID INT FOREIGN KEY REFERENCES Odeljenje (OdeljenjeID)
);
GO

INSERT INTO Odeljenje VALUES (1, 'Ortopedija', 20);
INSERT INTO Odeljenje VALUES (2, 'Neurologija', 15);
INSERT INTO Odeljenje VALUES (3, 'Hirurgija', 10);
GO

INSERT INTO Doktor VALUES (1, 'Petar', 'Simić', 1);
INSERT INTO Doktor VALUES (2, 'Saša', 'Lajović', 1);
INSERT INTO Doktor VALUES (3, 'Jana', 'Vislavski', 1);
INSERT INTO Doktor VALUES (4, 'Nebojša', 'Tomić', 1);
INSERT INTO Doktor VALUES (5, 'Tijana', 'Borković', 2);
INSERT INTO Doktor VALUES (6, 'Dejan', 'Aleksić', 2);
INSERT INTO Doktor VALUES (7, 'Filip', 'Dobožanov', 2);
INSERT INTO Doktor VALUES (8, 'Marko', 'Subotić', 2);
INSERT INTO Doktor VALUES (9, 'Teodora', 'Vilotić', 3);
INSERT INTO Doktor VALUES (10, 'Tamara', 'Berić', 3);
INSERT INTO Doktor VALUES (11, 'Ognjen', 'Minić', 3);
INSERT INTO Doktor VALUES (12, 'Strahinja', 'Kutina', 3);
INSERT INTO Doktor VALUES (13, 'Milica', 'Slavković', 3);
INSERT INTO Doktor VALUES (14, 'Todor', 'Ratković', 3);
GO

INSERT INTO Pacijent VALUES (1, 'Slobodan', 'Barać', '2024-10-27', 1);
INSERT INTO Pacijent VALUES (2, 'Milan', 'Jovičić', '2024-10-24', 1);
INSERT INTO Pacijent VALUES (3, 'Nevena', 'Sanićanin', '2024-10-01', 1);
INSERT INTO Pacijent VALUES (4, 'Jasna', 'Škuletić', '2024-9-25', 1);
INSERT INTO Pacijent VALUES (5, 'Igor', 'Pavlović', '2024-10-04', 1);
INSERT INTO Pacijent VALUES (6, 'Danijel', 'Komić', '2024-10-07', 1);
INSERT INTO Pacijent VALUES (7, 'Tatjana', 'Rakić', '2024-9-25', 1);
INSERT INTO Pacijent VALUES (8, 'Simeon', 'Paskaš', '2024-10-02', 1);
INSERT INTO Pacijent VALUES (9, 'Marija', 'Topalov', '2024-9-28', 1);
INSERT INTO Pacijent VALUES (10, 'Matija', 'Babić', '2024-10-03', 1);
INSERT INTO Pacijent VALUES (11, 'Darko', 'Maletić', '2024-9-08', 1);
INSERT INTO Pacijent VALUES (12, 'Nikolina', 'Todorić', '2024-10-26', 1);
```

```
INSERT INTO Pacijent VALUES (13, 'Gordana', 'Jukić', '2024-10-21', 2);
INSERT INTO Pacijent VALUES (14, 'Željko', 'Videkanić', '2024-9-28', 2);
INSERT INTO Pacijent VALUES (15, 'Sara', 'Obradović', '2024-10-02', 2);
INSERT INTO Pacijent VALUES (16, 'Olga', 'Koprivica', '2024-10-09', 2);
INSERT INTO Pacijent VALUES (17, 'Goran', 'Anđelovski', '2024-10-20', 2);
INSERT INTO Pacijent VALUES (18, 'Vladimir', 'Teofilović', '2024-10-28', 2);
INSERT INTO Pacijent VALUES (19, 'Ina', 'Mažić', '2024-10-02', 2);
INSERT INTO Pacijent VALUES (20, 'Miroslav', 'Petković', '2024-10-05', 3);
INSERT INTO Pacijent VALUES (21, 'Anđela', 'Jovanovski', '2024-10-06', 3);
INSERT INTO Pacijent VALUES (22, 'Robert', 'Lukač', '2024-09-28', 3);
```

## Izgled forme

Bolnica

Doktori Pacijenti

ID

Ime i prezime

Odeljenje

Forma se sastoji od dva taba.

Na prvom tabu (*Doktori*) se nalazi listView koji se nakon pokretanja aplikacije popunjava podacima o doktoru u tri polja (DoktorID, ime i prezime, naziv odeljenja).

Bolnica

Doktori Pacijenti

| ID | Ime i prezime    | Odeljenje   |
|----|------------------|-------------|
| 1  | Petar Simic      | Ortopedija  |
| 2  | Saša Lajovic     | Ortopedija  |
| 3  | Jana Vislavski   | Ortopedija  |
| 4  | Nebojša Tomic    | Ortopedija  |
| 5  | Tijana Borkovic  | Neurologija |
| 6  | Dejan Aleksic    | Neurologija |
| 7  | Filip Dobožanov  | Neurologija |
| 8  | Marko Subotic    | Neurologija |
| 9  | Teodora Vilotic  | Hirurgija   |
| 10 | Tamara Beric     | Hirurgija   |
| 11 | Ognjen Minic     | Hirurgija   |
| 12 | Strahinja Kutina | Hirurgija   |
| 13 | Milica Slavkovic | Hirurgija   |

ID

Ime i prezime

Odeljenje

Nakon unosa broja u textBox ID ukoliko se broj poklapa sa nekim od ID-jeva doktora podaci o tom doktoru se dodaju u preostala dva textBox-a, a izabrani doktor se selektuje u listView-u.

Bolnica

Doktori Pacijenti

| ID | Ime i prezime    | Odeljenje   |
|----|------------------|-------------|
| 1  | Petar Simic      | Ortopedija  |
| 2  | Saša Lajovic     | Ortopedija  |
| 3  | Jana Vislavski   | Ortopedija  |
| 4  | Nebojša Tomic    | Ortopedija  |
| 5  | Tijana Borkovic  | Neurologija |
| 6  | Dejan Aleksic    | Neurologija |
| 7  | Filip Dobožanov  | Neurologija |
| 8  | Marko Subotic    | Neurologija |
| 9  | Teodora Vilotic  | Hirurgija   |
| 10 | Tamara Beric     | Hirurgija   |
| 11 | Ognjen Minic     | Hirurgija   |
| 12 | Strahinja Kutina | Hirurgija   |
| 13 | Milica Slavkovic | Hirurgija   |

ID

9

Ime i prezime

Teodora Vilotic

Odeljenje

Hirurgija

Na drugom tabu se nalazi comboBox ispunjen nazivima odeljenja.

Nakon izbora opcije comboBox-a i klika na dugme *Prikaži pacijente* popunjava se dataGridView sa podacima o imenima i prezimenima pacijenata koji se nalaze na tom odeljenju, i broju dana pre koliko su stigli na odeljenje sortirano od onih koji su stigli poslednji ka onima koji su tu najduže.

Bolnica

Doktori Pacijenti

Izaberite odeljenje : Neurologija

Prikaži pacijente

|   | Ime i prezime       | Broj dana |
|---|---------------------|-----------|
| ▶ | Olga Koprivica      | 1         |
|   | Sara Obradovic      | 8         |
|   | Ina Mažic           | 8         |
|   | Vladimir Teofilovic | 10        |
|   | Željko Videkanic    | 10        |
|   | Gordana Jukic       | 17        |
|   | Goran Andelovski    | 18        |

## Rešenje

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakBolnica
{
    public partial class Form1 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=Bolnica;Integrated Security=True";
        string select = "SELECT DoktorID, ImeDoktora + ' ' + PrezimeDoktora AS [Ime i prezime], Naziv
FROM Doktor, Odeljenje " +
        "WHERE Doktor.OdeljenjeID = Odeljenje.OdeljenjeID";
        string insertComboBox = "SELECT OdeljenjeID, Naziv FROM Odeljenje";
        string insertGridView = "SELECT ImePacijenta + ' ' + PrezimePacijenta AS [Ime i prezime],
DATEDIFF(DAY, DatumDolaska, GETDATE()) AS [Broj dana] " +
        "FROM Pacijent WHERE OdeljenjeID = @OdeljenjeID ORDER BY [Broj dana] ASC";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand izaberiSve = new SqlCommand(select, konekcija);
            konekcija.Open();
            SqlDataReader citac = izaberiSve.ExecuteReader();

            while (citac.Read())
            {
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                prikaz.SubItems.Add(citac[1].ToString());
                prikaz.SubItems.Add(citac[2].ToString());
                lvDoktori.Items.Add(prikaz);
            }

            citac.Close();

            SqlDataAdapter adapter = new SqlDataAdapter(insertComboBox, konekcija);
            DataTable podaci = new DataTable();
            adapter.Fill(podaci);

            cbxOdeljenje.DataSource = podaci;
            cbxOdeljenje.ValueMember = "OdeljenjeID";
            cbxOdeljenje.DisplayMember = "Naziv";
        }

        private void txtID_TextChanged(object sender, EventArgs e)
        {
            txtIme.Clear();
            txtOdeljenje.Clear();

            lvDoktori.SelectedIndices.Clear();
            foreach (ListViewItem lv in lvDoktori.Items)
            {
                if (txtID.Text == lv.Text)
                {
                    txtIme.Text = lv.SubItems[1].Text;
                    txtOdeljenje.Text = lv.SubItems[2].Text;
                    lvDoktori.SelectedIndices.Add(lv.Index);
                }
            }
        }
    }
}
```

```
}

private void btnPrikazi_Click(object sender, EventArgs e)
{
    SqlConnection konekcija = new SqlConnection(konekcioniString);
    SqlCommand upis = new SqlCommand(insertGridView, konekcija);
    upis.Parameters.AddWithValue("@OdeljenjeID", cbxOdeljenje.SelectedValue.ToString());

    konekcija.Open();
    SqlDataAdapter adapter = new SqlDataAdapter(upis);
    DataTable casovi = new DataTable();
    adapter.Fill(casovi);
    dgvPacijenti.DataSource = casovi;
}
}
```



# Igrice

## Baza podataka

```
CREATE DATABASE Igrice
GO
USE Igrice
GO

CREATE TABLE Kompanija (
    KompanijaID INT PRIMARY KEY NOT NULL,
    NazivKompanije NVARCHAR(30)
);

CREATE TABLE Igrica (
    IgricaID INT PRIMARY KEY NOT NULL,
    NazivIgrice NVARCHAR(50),
    DatumIzlaska DATE,
    Ocena FLOAT,
    KompanijaID INT FOREIGN KEY REFERENCES Kompanija (KompanijaID)
);

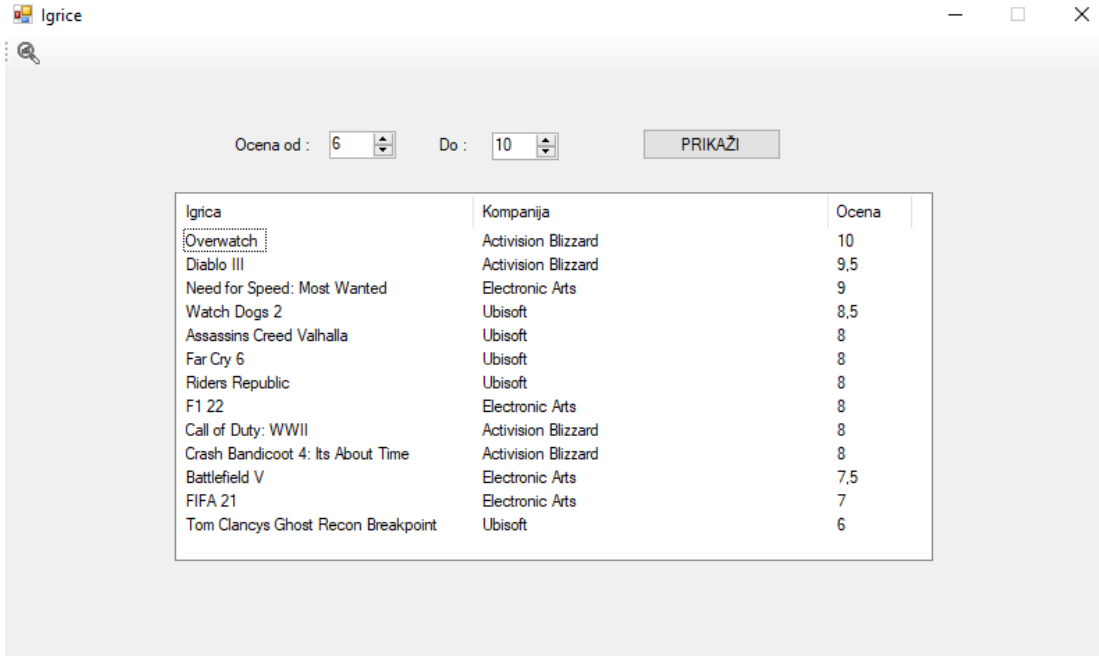
GO

INSERT INTO Kompanija VALUES (1, 'Ubisoft');
INSERT INTO Kompanija VALUES (2, 'Electronic Arts');
INSERT INTO Kompanija VALUES (3, 'Activision Blizzard');

GO

INSERT INTO Igrica VALUES (1, 'Watch Dogs 2', '2016-11-15', 8.5, 1);
INSERT INTO Igrica VALUES (2, 'Tom Clancys Ghost Recon Breakpoint', '2019-10-04', 6, 1);
INSERT INTO Igrica VALUES (3, 'Assassins Creed Valhalla', '2020-11-10', 8, 1);
INSERT INTO Igrica VALUES (4, 'Far Cry 6', '2021-10-07', 8, 1);
INSERT INTO Igrica VALUES (5, 'Riders Republic', '2021-10-28', 8, 1);
INSERT INTO Igrica VALUES (6, 'Need for Speed: Most Wanted', '2012-11-30', 9, 2);
INSERT INTO Igrica VALUES (7, 'Battlefield V', '2018-11-20', 7.5, 2);
INSERT INTO Igrica VALUES (8, 'FIFA 21', '2021-10-28', 7, 2);
INSERT INTO Igrica VALUES (9, 'F1 22', '2022-07-01', 8, 2);
INSERT INTO Igrica VALUES (10, 'Diablo III', '2012-05-15', 9.5, 3);
INSERT INTO Igrica VALUES (11, 'Overwatch', '2016-05-24', 10, 3);
INSERT INTO Igrica VALUES (12, 'Call of Duty: WWII', '2017-11-03', 8, 3);
INSERT INTO Igrica VALUES (13, 'Crash Bandicoot 4: Its About Time', '2020-10-02', 8, 3);
```

## Izgled forme



Ocena od : 6 Do : 10 PRIKAŽI

| Igrica                             | Kompanija           | Ocena |
|------------------------------------|---------------------|-------|
| Overwatch                          | Activision Blizzard | 10    |
| Diablo III                         | Activision Blizzard | 9,5   |
| Need for Speed: Most Wanted        | Electronic Arts     | 9     |
| Watch Dogs 2                       | Ubisoft             | 8,5   |
| Assassins Creed Valhalla           | Ubisoft             | 8     |
| Far Cry 6                          | Ubisoft             | 8     |
| Riders Republic                    | Ubisoft             | 8     |
| F1 22                              | Electronic Arts     | 8     |
| Call of Duty: WWII                 | Activision Blizzard | 8     |
| Crash Bandicoot 4: Its About Time  | Activision Blizzard | 8     |
| Battlefield V                      | Electronic Arts     | 7,5   |
| FIFA 21                            | Electronic Arts     | 7     |
| Tom Clancys Ghost Recon Breakpoint | Ubisoft             | 6     |

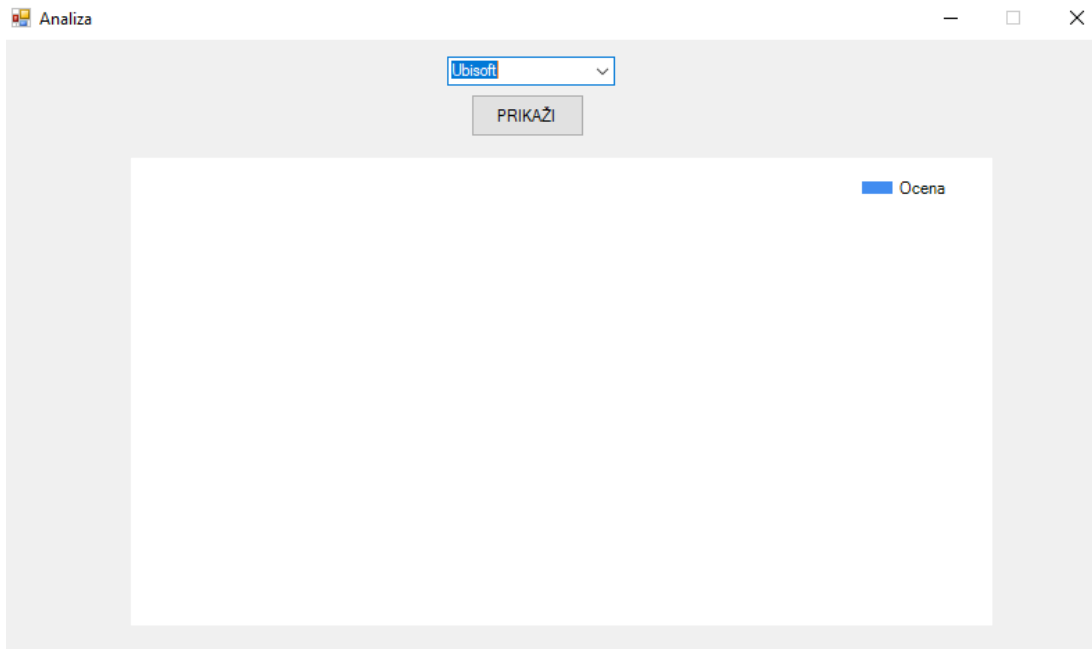
Unutar programa su dve forme.

Na prvoj formi se nalazi ToolStrip sa jednom ikonom.

Pri pokretanju forme listView se popunjava podacima Igrica, Kompanija, Ocena poređano u opadajućem redosledu na osnovu ocene.

Ispod ToolStrip-a su dva numericUpDown-a koji označavaju početnu i krajnju vrednost ocene za koje će se prikazivati igrice unutar ListView-a. Početna vrednost levog numeric-a je 6, a desnog 10. Klikom na dugme Prikaži prikazuju se samo one igrice iz baze koje odgovaraju uslovima da im je ocena veća ili jednaka vrednosti levog numeric-a, a manja ili jedna vrednosti desnog numeric-a.

Klikom na ikonicu unutar ToolStrip-a otvara se druga forma.



Ubisoft

PRIKAŽI

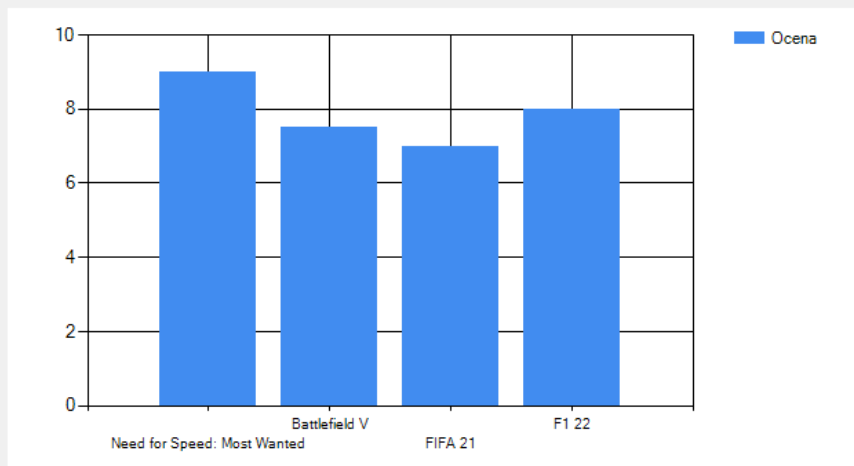
Ocena

Na drugoj formi se nalaze ComboBox, Button i Chart.

Klikom na dugme Prikaži na drugoj formi se Chart popunjava kolonama, na x osi se nalaze nazivi igrice a na y osi ocena.

Electronic Arts ▾

PRIKAŽI



## Rešenje

```
using System;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakIgrice
{
    public partial class Form1 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=Igrice;Integrated Security=True";
        string select = "SELECT NazivIgrice, NazivKompanije, Ocena " +
            "FROM Igrica INNER JOIN Kompanija ON Igrica.KompanijaID = Kompanija.KompanijaID ORDER BY
Ocena DESC";
        string select2 = "SELECT NazivIgrice, NazivKompanije, Ocena " +
            "FROM Igrica INNER JOIN Kompanija ON Igrica.KompanijaID = Kompanija.KompanijaID " +
            "WHERE Ocena >= @Od AND Ocena <= @Do ORDER BY Ocena DESC";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand izaberiSve = new SqlCommand(select, konekcija);
            konekcija.Open();
            SqlDataReader citac = izaberiSve.ExecuteReader();
            while (citac.Read())
            {
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                prikaz.SubItems.Add(citac[1].ToString());
                prikaz.SubItems.Add(citac[2].ToString());
                lvIgrice.Items.Add(prikaz);
            }
        }

        private void btnPrikazi_Click(object sender, EventArgs e)
        {
            lvIgrice.Items.Clear();
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand izaberiSve = new SqlCommand(select2, konekcija);
            izaberiSve.Parameters.AddWithValue("@Od", numOd.Value.ToString());
            izaberiSve.Parameters.AddWithValue("@Do", numDo.Value.ToString());
            konekcija.Open();
            SqlDataReader citac = izaberiSve.ExecuteReader();
            while (citac.Read())
            {
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                prikaz.SubItems.Add(citac[1].ToString());
                prikaz.SubItems.Add(citac[2].ToString());
                lvIgrice.Items.Add(prikaz);
            }
        }

        private void tsbAnaliza_Click(object sender, EventArgs e)
        {
            Form2 forma = new Form2();
            forma.Show();
        }
    }
}
```

```
using System;
```

```

using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakIgrice
{
    public partial class Form2 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=Igrice;Integrated Security=True";
        string insertComboBox = "SELECT KompanijaID, NazivKompanije FROM Kompanija";
        string insertChart = "SELECT NazivIgrice, Ocena FROM Igrica WHERE KompanijaID = @KompanijaID";

        public Form2()
        {
            InitializeComponent();

            private void Form2_Load(object sender, EventArgs e)
            {
                SqlConnection konekcija = new SqlConnection(konekcioniString);
                SqlDataAdapter adapter = new SqlDataAdapter(insertComboBox, konekcija);
                DataTable podaci = new DataTable();
                adapter.Fill(podaci);

                cbxKompanije.DataSource = podaci;
                cbxKompanije.ValueMember = "KompanijaID";
                cbxKompanije.DisplayMember = "NazivKompanije";
            }

            private void btnAnaliza_Click(object sender, EventArgs e)
            {
                SqlConnection konekcija = new SqlConnection(konekcioniString);
                SqlCommand upis = new SqlCommand(insertChart, konekcija);
                upis.Parameters.AddWithValue("@KompanijaID", cbxKompanije.SelectedValue.ToString());
                konekcija.Open();
                SqlDataAdapter adapter = new SqlDataAdapter(upis);
                konekcija.Close();
                chIgrice.DataSource = adapter;
                chIgrice.Series[0].XValueMember = "NazivIgrice";
                chIgrice.Series[0].YValueMembers = "Ocena";
                chIgrice.DataBind();
            }
        }
    }
}

```

# Humanitarna organizacija

## *Baza podataka*

```
CREATE DATABASE HumanitarnaOrganizacija
GO
USE HumanitarnaOrganizacija
GO

CREATE TABLE Drzava (
    DrzavaID INT PRIMARY KEY NOT NULL,
    Naziv NVARCHAR(50) NOT NULL
);

CREATE TABLE Donacija (
    DonacijaID INT PRIMARY KEY NOT NULL,
    Iznos INT NOT NULL,
    Datum DATE NOT NULL,
    DrzavaID INT FOREIGN KEY REFERENCES Drzava(DrzavaID)
);

Podatke kopirati sa eitsombor.rs/baza.html
```

## Izgled forme

Humanitarna organizacija

Analiza

☒ Evropa ☒ Severna Amerika **PRIKAŽI**

| Država   | Datum      | Iznos |
|----------|------------|-------|
| Cma Gora | 2023-03-20 | 31931 |
| Švedska  | 2023-03-18 | 46234 |
| Kanada   | 2023-03-17 | 37705 |
| Švedska  | 2023-03-14 | 23511 |
| Srbija   | 2023-03-13 | 35214 |
| Kanada   | 2023-03-11 | 26891 |
| Cma Gora | 2023-03-07 | 20855 |
| Kanada   | 2023-03-05 | 21157 |
| Švedska  | 2023-03-05 | 45282 |
| Kanada   | 2023-03-03 | 15272 |

**IZADI**

Po pokretanju forme ListView je popunjen podacima o svim donacijama u formatu Država – Datum – Iznos, poređano po datumu od najskorijeg ka najdaljem. Oba checkBox-a su obeležena.

Klikom na dugme *Prikaži* proverava se koji od dva checkBox-a je obeležen. U zavisnosti od toga u ListView-u se prikazuju podaci koji odgovaraju checkBox-ovima (Evropa – sve sem SAD i Kanade, Severna Amerika – samo SAD i Kanada, oba – sve, nijedan – prazan ListView).

Humanitarna organizacija

Analiza

☐ Evropa ☒ Severna Amerika **PRIKAŽI**

| Država | Datum      | Iznos |
|--------|------------|-------|
| Kanada | 2023-03-17 | 37705 |
| Kanada | 2023-03-11 | 26891 |
| Kanada | 2023-03-05 | 21157 |
| Kanada | 2023-03-03 | 15272 |
| Kanada | 2023-02-27 | 32094 |
| Kanada | 2023-02-17 | 28396 |
| SAD    | 2023-02-16 | 19468 |
| SAD    | 2023-02-13 | 18635 |
| Kanada | 2023-02-01 | 47462 |
| Kanada | 2023-01-29 | 43400 |

**IZADI**

Klikom na dugme *Izadi* forma se zatvara.

Klikom na dugme *Analiza* u ToolStrip-u otvara se druga forma.

Analiza

Izaberite državu :

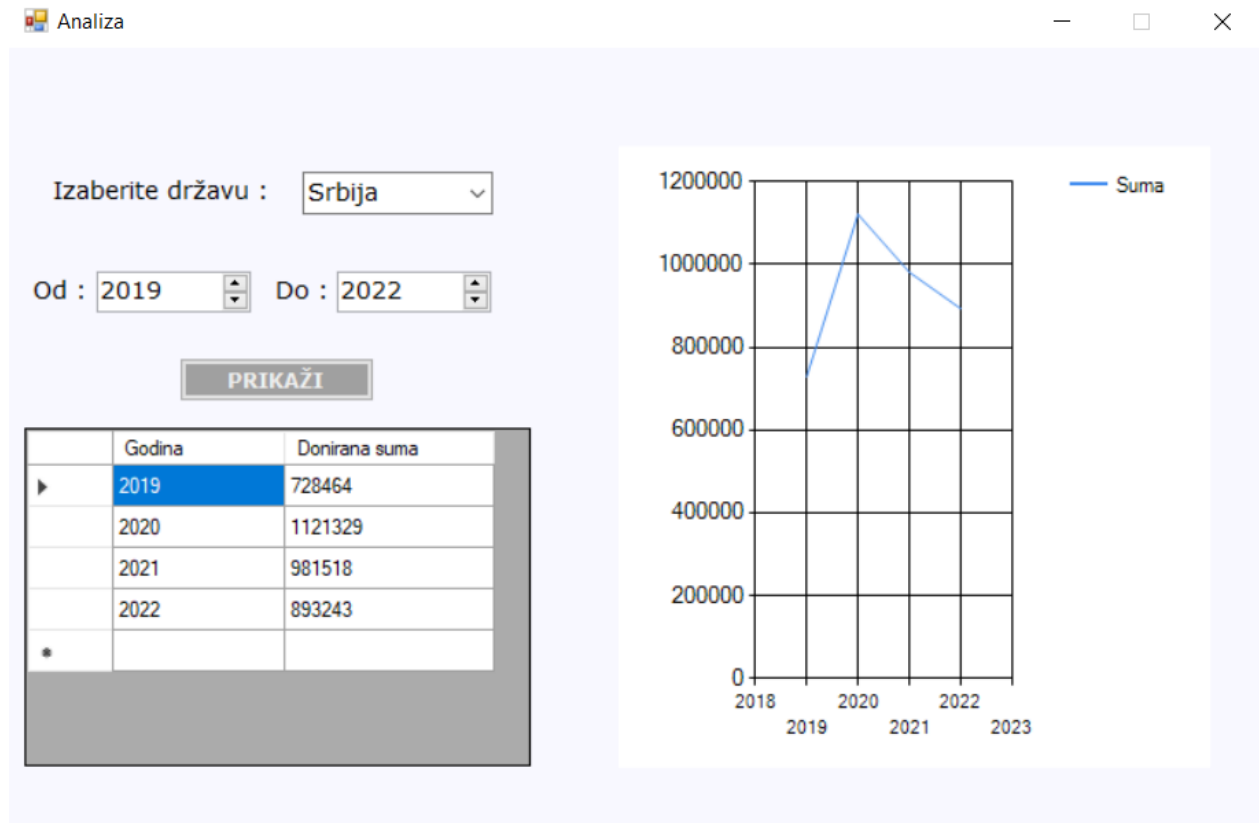
Od :  Do :

Na drugoj formi se nalazi ComboBox popunjen nazivima država iz baze podataka.

NumericUpDown-i predstavljaju godine u kojima su izvršene donacije. Mogućnosti vrednosti unutar NumericUpDown-a su od 2019 do 2023.

Klikom na dugme *Prikaži* DataGridView se popunjava podacima o izabranoj državi u formatu Godina – Ukupna suma svih donacija.

Istovremeno se ti podaci prikazuju i na grafiku linijskog tipa (Line).





## Rešenje

```
using System;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakHumanitarnaOrganizacija
{
    public partial class Form1 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=HumanitarnaOrganizacija;Integrated Security=True";
        string insertListView1 = "SELECT Naziv, CAST(Datum AS nvarchar), Iznos FROM Drzava RIGHT JOIN
Donacija ON Drzava.DrzavaID = Donacija.DrzavaID ORDER BY Datum DESC";
        string insertListView2 = "SELECT Naziv, CAST(Datum AS nvarchar), Iznos FROM Drzava, Donacija
WHERE Drzava.DrzavaID = Donacija.DrzavaID AND Donacija.DrzavaID NOT IN (2,3) ORDER BY Datum DESC";
        string insertListView3 = "SELECT Naziv, CAST(Datum AS nvarchar), Iznos FROM Drzava, Donacija
WHERE Drzava.DrzavaID = Donacija.DrzavaID AND Donacija.DrzavaID IN (2, 3) ORDER BY Datum DESC";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            SqlConnection konekcija = new SqlConnection(konekcioniString);
            SqlCommand select = new SqlCommand(insertListView1, konekcija);

            konekcija.Open();

            SqlDataReader citac = select.ExecuteReader();

            while (citac.Read())
            {
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                prikaz.SubItems.Add(citac[1].ToString());
                prikaz.SubItems.Add(citac[2].ToString());
                lvDonacije.Items.Add(prikaz);
            }
        }

        private void btnPrikazi_Click(object sender, EventArgs e)
        {
            lvDonacije.Items.Clear();
            if (cbEvropa.Checked && cbSAmerika.Checked)
            {
                SqlConnection konekcija = new SqlConnection(konekcioniString);
                SqlCommand select = new SqlCommand(insertListView1, konekcija);

                konekcija.Open();

                SqlDataReader citac = select.ExecuteReader();

                while (citac.Read())
                {
                    ListViewItem prikaz = new ListViewItem(citac[0].ToString());
                    prikaz.SubItems.Add(citac[1].ToString());
                    prikaz.SubItems.Add(citac[2].ToString());
                    lvDonacije.Items.Add(prikaz);
                }
            }

            else if (cbEvropa.Checked && cbSAmerika.Checked == false)
            {
                SqlConnection konekcija = new SqlConnection(konekcioniString);
                SqlCommand select = new SqlCommand(insertListView2, konekcija);
```

```

        konekcija.Open();

        SqlDataReader citac = select.ExecuteReader();

        while (citac.Read())
        {
            ListViewItem prikaz = new ListViewItem(citac[0].ToString());
            prikaz.SubItems.Add(citac[1].ToString());
            prikaz.SubItems.Add(citac[2].ToString());
            lvDonacije.Items.Add(prikaz);
        }
    }
    else if (cbEvropa.Checked == false && cbSAmerika.Checked)
    {
        SqlConnection konekcija = new SqlConnection(konekcioniString);
        SqlCommand select = new SqlCommand(insertListView3, konekcija);

        konekcija.Open();

        SqlDataReader citac = select.ExecuteReader();

        while (citac.Read())
        {
            ListViewItem prikaz = new ListViewItem(citac[0].ToString());
            prikaz.SubItems.Add(citac[1].ToString());
            prikaz.SubItems.Add(citac[2].ToString());
            lvDonacije.Items.Add(prikaz);
        }
    }
    else
    {
        lvDonacije.Items.Clear();
    }
}

private void tsbAnaliza_Click(object sender, EventArgs e)
{
    Form2 forma = new Form2();
    forma.Show();
}

private void btnIzadji_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ZadatakHumanitarnaOrganizacija
{
    public partial class Form2 : Form
    {
        string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial
Catalog=HumanitarnaOrganizacija;Integrated Security=True";

        string insertComboBox = "SELECT DrzavaID, Naziv FROM Drzava";
        string insertGridView = "Select YEAR(Datum) as Godina, SUM(Iznos) as [Donirana suma] FROM Drzava
INNER JOIN Donacija ON Drzava.DrzavaID = Donacija.DrzavaID WHERE Year(Datum)>=@pocetna" +
        " AND Year(Datum) <= @zavrsna AND Drzava.DrzavaID = @drzava GROUP BY YEAR(Datum)";
    }
}

```

```

public Form2()
{
    InitializeComponent();
}

private void Form2_Load(object sender, EventArgs e)
{
    SqlConnection konekcija = new SqlConnection(konekcioniString);
    SqlDataAdapter adapter = new SqlDataAdapter(insertComboBox, konekcija);
    DataTable podaci = new DataTable();
    adapter.Fill(podaci);
    cbxDrzave.DataSource = podaci;
    cbxDrzave.ValueMember = "DrzavaID";
    cbxDrzave.DisplayMember = "Naziv";
}

private void btnPrikazi_Click(object sender, EventArgs e)
{
    SqlConnection konekcija = new SqlConnection(konekcioniString);
    SqlCommand upis = new SqlCommand(insertGridView, konekcija);
    upis.Parameters.AddWithValue("@drzava", cbxDrzave.SelectedValue.ToString());
    upis.Parameters.AddWithValue("@pocetna", numOd.Value);
    upis.Parameters.AddWithValue("@zavrsna", numDo.Value);

    konekcija.Open();
    SqlDataAdapter dataAdapter = new SqlDataAdapter(upis);
    DataTable podaci = new DataTable();
    dataAdapter.Fill(podaci);
    dgvPodaci.DataSource = podaci;
    konekcija.Close();

    chLinija.DataSource = dataAdapter;
    chLinija.Series[0].YValueMembers = "Donirana suma";
    chLinija.Series[0].XValueMember = "Godina";
    chLinija.DataBind();
}
}

```