

## C# Forme, windows kontrole

Na početku, kada su se pojavile prve verzije Windowsa, programiranje Windows aplikacija je bilo veoma teško i dostupno samo odabranim profesionalcima. Pojavom vizualnih alata i takozvanog RAD (Rapid Application Development – rapidni razvoj aplikacija) sistema, kreiranje ovakvih aplikacija je značajno pojednostavljeno i postalo je dostupno širokom krugu programera. Svi elementi koji se koriste u programiranju Windows aplikacija su "prefabrikovani" - stoje nam na raspolaganju sa svojom paletom svojstava, metoda i događaja u vidu gotovih klasa. Nama samo ostaje da ih organizujemo na način koji nam je potreban, postavimo im odgovarajuća svojstva i napišemo kod kao odgovor na akcije korisnika – događaji klasa.

Na vrhu hijerarhije elemenata je sama aplikacija, odnosno objekt Application.

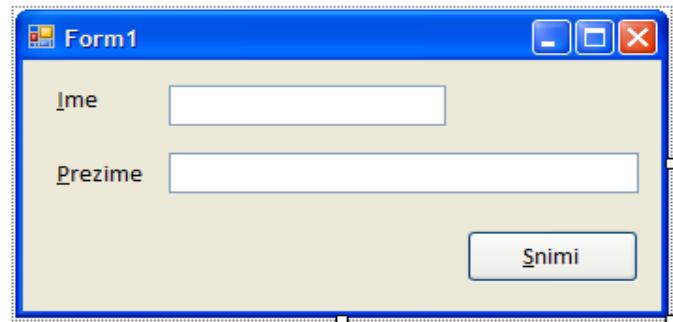
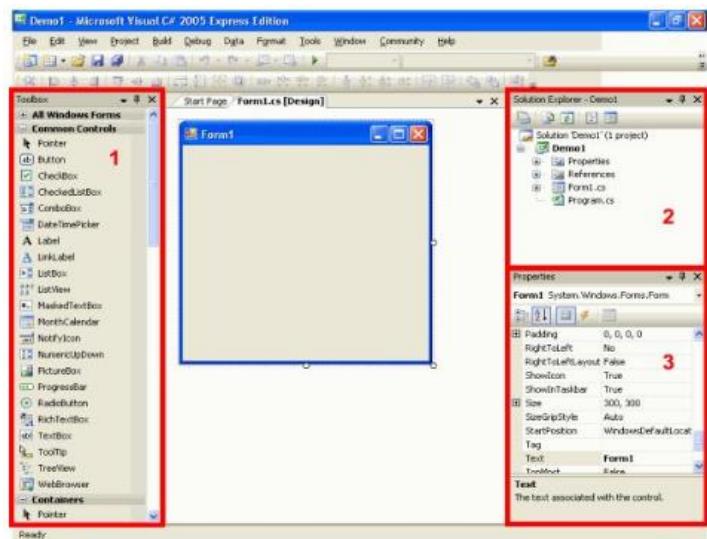
Ispod njega se nalazi bar jedna, a obično i više Windows formi. Svaka od njih sadrži standardne elemente korisničkog interfejsa Windowsa koje nazivamo **kontrole**. Svakom korisniku Windowsa su poznati ovi elementi u koje spadaju **komandna dugmad, liste, padajuće liste, izbor opcija, meniji i tako dalje**.

Svaka Windows aplikacija se sastoji najmanje od jedne forme (formulara), međutim najčešće ih ima nekoliko, pa čak i nekoliko desetina u složenijim aplikacijama. Jasno je da treba naučiti načine kako programski otvoriti i zatvoriti željenu formu, kao i načine postavljanja osobina forme.

Osnove svake windows aplikacije predstavljaju **forme (eng. forms)** koji se po potrebi otvaraju i zatvaraju, zavisno od toka i logike korisničkog okruženja. Na svakoj formi se nalazi više objekata koji predstavljaju korisnički interfejs – windows kontrole. Osnovno za svaku kontrolu je svojstvo Name koje predstavlja njen jedinstveni identifikator u okviru forme na kome se nalazi. Ovo znači da na jednoj formi ne možemo imati dvije ili više kontrola koje se isto zovu.

**Komandno dugme ili samo dugme (button)**, predstavlja najočigledniju i verovatno najjednostavniju kontrolu. Omogućava korisniku da klikom na njega pokrene odgovarajuću akciju – segment programskog koda.

Ključno svojstvo dugmeta je svojstvo **Text** kojim se postavlja natpis na njemu. Generalno, svaka kontrola koja na sebi posjeduje neki natpis ima svojstvo Text.



**Labela (label)** daje mogućnost prikaza teksta na formularu koji može biti samostalan, ali se obično vezuje za neku drugu kontrolu i na taj način opisuje njenu značenje. Najčešće se postavlja uz TextBox kontrolu.

**TextBox** kontrola koja omogućava korisniku da unese podatke koji se dalje koriste u aplikaciji. Relativno složena kontrola koja se veoma često koristi prilikom građenja korisničkog okruženja. Omogućava unos teksta u jednoj ili više linija.

**MessageBox** predstavlja standardni dijalog za poruke u Windowsu. Metoda **Show** prikazuje ovaj dijalog sa ulaznim parametrom koji predstavlja tekst koji treba prikazati.

### CheckBox kontrola

Ova kontrola omogućava korisniku da markira ili demarkira neku opciju. Ako imamo više CheckBox kontrole na formularu, one su međusobno nezavisne – mogu se proizvoljno markirati ili demarkirati. Ključno svojstvo ove kontrole je svojstvo **Checked** koje može imati dvije vrijednosti - **True** (da) i **False** (ne).

### Radio Button kontrola

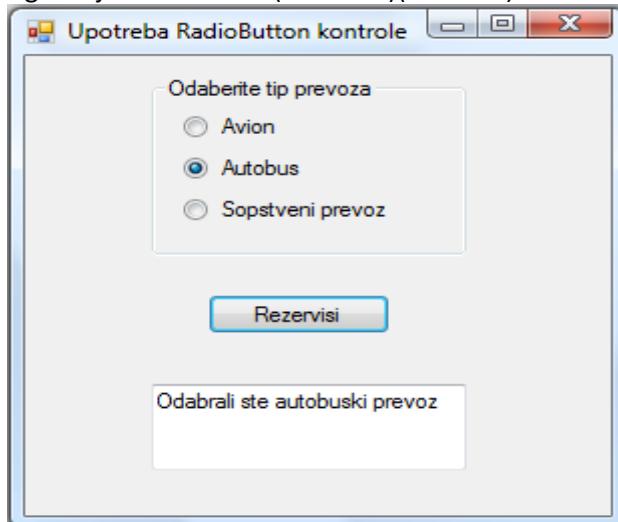
Ova kontrola je slična **CheckBox** kontroli, ali u ovom slučaju možemo markirati samo jednu kontrolu u okviru iste grupe. Otud i naziv "radio dugme", sjetite se starih radio aparata gdje se preko mehaničkih dugmadi birao frekventni opseg. Pritiskom na bilo koje dugme, prethodno uključeno se isključuje – samo jedno može biti upaljeno. Pod grupom radio dugmića se smatraju svi koji su postavljeni na isti **kontejner** kontrola. Očigledni kontejner kontrola je formular, ali ih ima još nekoliko na raspaganju.

### List Box kontrola

Ovo je složenija kontrola, koja se sastoji od više stavki. Na primer, može predstavljati listu predmeta u školi. Korisnik može izabrati predmet i na primer upisati ocjene i slično.

### Zadatak:

Napraviti windows aplikaciju, koja će imati jednu formu sa tri radio dugmeta, smještenih u jedan group box, zatim jedno dugme i jedan text box (multiline)(*Slika 1.1*):

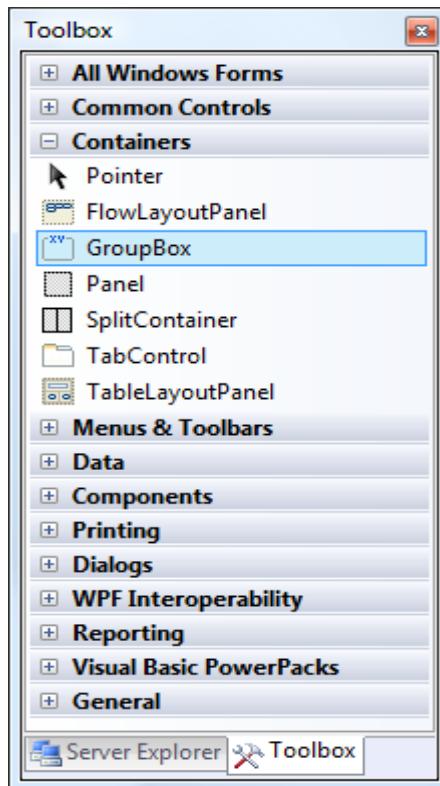


*Slika 1.1: Izgled programa*

Pritiskom na dugme "Rezervisi", u text box-u se ispisuje odgovarajuća poruka korisniku. Samo jedno radio dugme u jednom trenutku može biti selektovano.

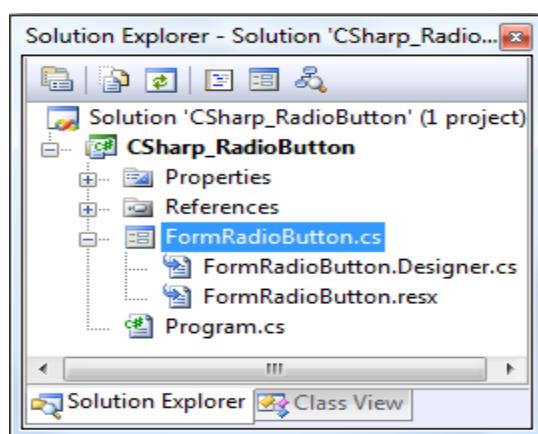
**Rješenje:**

Radio dugmad se dodaju na formu direktnim prevlačenjem iz toolbox-a. Pre toga, neophodno je da se na formu postavi **GroupBox** kontrola, a tek onda u nju da se smjeste radio dugmad. GroupBox se dodaje prevlačenjem iz toolbox-a, i to iz grupe "**Containers**" (*Slika 1.2*):



*Slika 1.2: Mjesto GroupBox-a u okviru Toolbox-a*

Prozor Solution Explorer-a treba da izgleda kao na narednoj slici (*Slika 1.3*):



*Slika 1.3: Solution Explorer*

Kod za dugme “Rezervisi”:

```
private void buttonRezervisi_Click(object sender, EventArgs e)
{
    // brisanje sadrzaja text box-a
    textBox1.Clear();
    // property Checked radio dugmeta vraca true, ako je ono cekirano
    // ili false, ako nije cekirano
    if (radioButtonAvion.Checked)
        textBox1.Text = "Odabrali ste avionski prevoz";
    else if (radioButtonAutobus.Checked)
        textBox1.Text = "Odabrali ste autobuski prevoz";
    else if (radioButtonSopstveni.Checked)
        textBox1.Text = "Odabrali ste sopstveni prevoz";
    else
    {
        // u slucaju da nista nije cekirano
        MessageBox.Show("Odaberite tip prevoza");
        return;
    }
}
```