

9_1. Nizovi, liste i kolekcije



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Što ćete naučiti u ovom poglavlju?

- Što su nizovi i kada se koriste
- Kako se deklariraju nizovi i pohranjuju vrijednosti u nizove
- Što su liste i kada se koriste
- Kako se deklariraju liste i pohranjuju vrijednosti u liste
- Što su kolekcije
- Kako se deklariraju kolekcije i pohranjuju vrijednosti u kolekcije

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Skupovi (kolekcije) klasa

- U .NET-u postoji više načina kako pohraniti više objekata u jednu kolekciju (ili skup) objekata
- Neki od tih načina su:
- Nizovi (Arrays)
 - Liste (List ili ArrayList)
 - Kolekcije (Collection)
- Koriste se kada želimo npr. imena ili plaće više djelatnika pohraniti u jedan zajednički skup objekata i s tim skupom rukovati u aplikaciji (npr. sortirati, mijenjati vrijednosti, brisati, prikazivati, i dr.). Taj skup objekata je indeksiran, tj. za svaki objekt u tom skupu poznat je njegov položaj (koji je on po redu u tom skupu). Indeksi kreću od vrijednosti 0 nadalje, npr.

Djelatnik [0]
Djelatnik [1]
Djelatnik [2]
.... itd.

Klase koje su raspoložive za skupove objekata u .NET-u su: Array, ArrayList, NameValueCollection, StringCollection, Queue, Stack, and BitArray.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Nizovi

Niz (eng. array) je indeksirani skup objekata koji su istog tipa.

Vrste nizova: jednodimenzionalni, višedimenzionalni i ugnježdeni (tj. nizovi niza).
S obzirom da su u C#-u članovi niza objekti, oni mogu imati svoja svojstva i metode.

Kada se deklarira neki niz u C#-u, automatski se kreira objekt u ugrađenoj klasi System.Array (da Vaša aplikacija koristi ovu klasu možete vidjeti u naredbi Using na početku svoje aplikacije – Using System.Array).
Automatski su na raspolaganju ugrađene metode koje vrijede za ovu klasu, a neke od njih su:

```
Clear()  
Copy()  
Sort()  
IndexOf()  
i dr. čiji puni popis možete pronaći u (Liberty, Programming C#,  
O'Reilly, 2006)
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklariranje nizova

- Za deklariranje niza potrebno je upisati naredbu čija je opća sintaksa:
type [] naziv-niza;

type – označava tip niza, npr. string, int, double i dr.

naziv-niza – proizvoljni naziv niza

Npr.

```
string[] imeStudenta;  
int[] ocjenaStudenta;  
Double[] prosjOcjena;
```

Da bismo kreirali instancu niza (tj. ostavili prostor u memoriji za određeni broj članova tog određenog niza) potrebno je upisati ovakvu naredbu:

naziv-niza = new type[dimenzija_niza];

Npr.

```
imeStudenta = new string[5];
```

- Ovim smo otvorili u memoriji prostor za niz koji će se zvati **imeStudenta**, a sadržavat će 5 elemenata niza, koji će biti tipa string (tekstualnog), tj.
- U ovaj niz moći ćemo upisati imena za 5 studenata (indeks niza kreće od 0, 1, 2, 3, 4), dakle za imeStudenta[0], imeStudenta[1],...imeStudenta[4]

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Standardne vrijednosti elemenata niza

Nakon deklaracije, elementi niza kao standardne vrijednosti automatski dobivaju:

- 0, ako je niz numeričkog tipa (int ili double)
- Null, ako je niz tipa reference koji pokazuje na neki objekt, npr. Button ili Form ili dr.

Npr. ako deklariramo niz:

```
ocjena = new int[3];
```

- U memoriji će se ostaviti prostor za 3 ocjene, ocjena[0], ocjena[1], ocjena[2], a svaka će imati početnu vrijednost 0, tj.
ocjena[0]=0, ocjena[1]=0, ocjena[2]=0.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

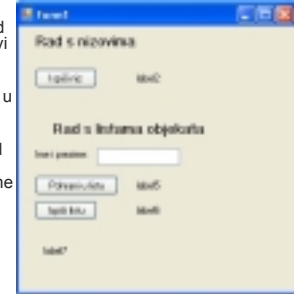
Kada koristiti nizove?

- Nizove se preporuča koristiti kada želite kreirati poznati broj objekata istog tipa i dodijeliti im neke vrijednosti kroz kod programa, npr. ako želite napraviti niz brojeva kojima ćete dodijeliti vrijednosti od 1 do npr. 100, ili npr. niz od 20 županija, ili niz ocjena od 1 do 5 i slično.
- Ne preporuča se koristiti nizove ako:
 - nije poznato koliko će elemenata niza biti,
 - vrijednosti niza treba unijeti korisnik kroz aplikaciju (preko textBox-ova ili dr.) ili upisivati u bazu (u tom slučaju se preporuča koristiti kolekcije)
- Upotreba nizova bit će objašnjena kroz primjer u Vježbi 9_1.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 9_1 – rad sa nizovima, listama i kolekcijama

- Otvorite novi projekt pod nazivom Vježba9_1. Prvi cilj nam je:
- Kreirati u programskom kodu niz i taj niz ispisati u labeli na formi.
- Liste i kolekcije ćemo dodati kasnije, pa zasad napravite samo labelu "Rad s nizovima" i dugme "Ispiši niz".



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklaracija niza i dodjela vrijednosti nizu

- Kliknite na dugme "Ispiši niz" i u metodi `button1_Click()` deklarirajte niz od 5 elemenata koji će sadržavati imena studenata, pa će biti tipa `string`. Deklarirati niz i dodijeliti mu vrijednosti moguće je na više načina
- 1. način: Ako bismo htjeli pojedinačno dodjeljivati vrijednosti za imena studenata, tada bismo u posebnoj naredbi deklarirali niz od 5 elemenata, a u posebnoj naredbi dodijelili vrijednost prvom elementu, drugom, itd. ovako:

```
imeStudenta = new string[5];  
imeStudenta[0] = "Marko";  
imeStudenta[1] = "Janko";  
imeStudenta[2] = "Slavko";  
imeStudenta[3] = "Ante";  
imeStudenta[4] = "Andrija";
```

deklaracija niza od 5 elemenata

Dodjeljivanje vrijednosti elementima niza pojedinačno
- 2. način – deklaracija niza i dodjela vrijednosti u jednoj liniji koda, ovako:

```
string[] imeStudenta = { "Marko", "Janko", "Slavko", "Ante", "Andrija" };
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklaracija niza i dodjela vrijednosti nizu - nastavak

- Za naš primjer koristit ćemo drugi, kraći način, pa će naredbe u metodi `button1_Click()` izgledati ovako:

```
private void button1_Click(object sender, EventArgs e)  
{  
    string[] imeStudenta = { "Marko", "Janko", "Slavko", "Ante", "Andrija" };  
}
```
- Osim što deklariramo niz i dodijelimo mu vrijednosti, potrebno je u toj metodi i ispisati vrijednosti elemenata niza u labeli. Ako želimo ispisati samo jedan element niza, npr. ime prvog studenta, napisali bismo ovakvu naredbu:

```
label2.Text = imeStudenta[0];
```

Rezultat ove naredbe bi bio ispisani tekst: Marko
S obzirom da elemenata niza ima više, da bismo ih sve ispisali, potrebna je petlja iteracije.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Ispis elemenata niza u petlji for i foreach

- Petlja `for` izgledala bi ovako:

```
for (int i = 0; i <= 4; i++)  
{  
    label2.Text += imeStudenta[i] + "\n";  
}
```

Znak `+="` stavlja se kako bi u labelu uvijek ispisivala nova vrijednost kao dodatak staroj, a ne samo zadnja.
Znak `\n` označava prijelaz u novi red nakon svake vrijednosti.
- Petlja `foreach` izgledala bi ovako:

```
foreach (string ime in imeStudenta)  
{  
    label2.Text += ime + "\n";  
}
```

U petlji `foreach` uvodi se nova varijabla `ime` tipa `string` koja će preuzeti pojedinačne elemente niza.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Sortiranje i ispis elemenata niza

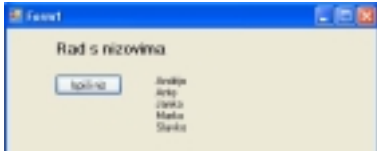
- Za naš primjer koristit ćemo petlju `foreach`.
- Prije ispisa u labelu, sortirat ćemo elemente niza po abecednom redoslijedu, i to s pomoću naredbe `Array.Sort(naziv_niza)`; pa će cijela metoda izgledati ovako:

```
private void button1_Click(object sender, EventArgs e)  
{  
    string[] imeStudenta = { "Marko", "Janko", "Slavko", "Ante", "Andrija" };  
    Array.Sort(imeStudenta);  
    foreach (string ime in imeStudenta)  
    {  
        label2.Text += ime + "\n";  
    }  
}
```
- Pohranite i debug-irajte aplikaciju. Prilikom izvršavanja kliknite na dugme "Ispiši niz". Kao rezultat ispisat će se niz imena ovako:

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Izvršavanje aplikacije za rad s nizovima

- Kada kliknemo na dugme "Ispiši niz" pojavit će sljedeći rezultat ispisa (imena će biti sortirana po abecedi):



- Osim sortiranja, na nizovima se mogu primijeniti i druge metode, o kojima više pogledajte u (Liberty, Programming C#, O'Reilly, 2006)

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Popunjavanje niza brojevima

- Ako bismo niz htjeli popuniti brojčanim vrijednostima, tada se to može napraviti u petlji for, npr. na sljedeći način:

```
for (int i = 0; i <= 4; i++)  
{  
    imeStudenta[i] = i.ToString();  
}
```

Kada bismo ispisali ovaj niz, rezultat bi bio:

0
1
2
3
4

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Liste

- Lista (eng. List) je skup objekata istog tipa koji se pohranjuju u zajednički niz objekata (ArrayList) i mogu se dodavati bez da unaprijed znamo dimenziju (broj elemenata) u nizu
- Pogodna je za upotrebu ako se vrijednosti objekata učitavaju od strane korisnika kroz formu (npr. iz textBox-a)

Sintaksa za deklaraciju liste:

- **List<type>** djelatnici = new List<type>();

Npr.

```
List<string> imeStudenta = new List<string>();
```

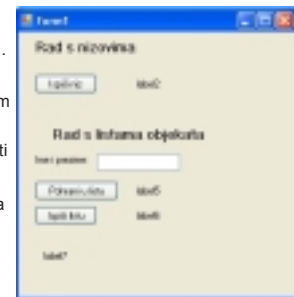
Popunjavanje liste vrijednostima

- Ako se u deklariranu listu imeStudenta žele pohraniti vrijednosti koje korisnik unese u textBox1, naredba je sljedeća:
imeStudenta.Add(textBox1.Text);

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 9_1 – nastavak – rad s listama

- Nastavit ćemo projekt pod nazivom Vježba9_1. Nakon rada s nizovima, drugi cilj nam je:
- Kreirati listu pod nazivom **studenti**, dodjeljivati joj vrijednosti koje korisnik unese u textBox i ispisati elemente liste
- Za te potrebe dodajte kontrole za rad s listama objekata prikazane na slici desno.



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklaracija liste

- S obzirom da listu studenti želimo koristiti u više metoda naše forme, nećemo je deklarirati unutar jedne metode, nego za cijelu formu, tj. odmah ispod naredbi kojima se kreira klasa forme upisat ćemo deklaraciju liste:

```
namespace vjezba9  
{  
    public partial class Form1 : Form  
    {  
        .....  
        List<string> studenti = new List<string>();  
        .....  
    }  
}
```

Deklaracija liste (upisana u klasu forme, ali izvan svih metoda forme, kako bi se lista mogla koristiti u više metoda)

List je tipa string, jer će se u njezine elemente upisivati tekstualne vrijednosti za imena studenata, ali se u aplikaciji mogu kreirati i liste raznih drugih tipova (int, double, button, itd.)

Dodjeljivanje vrijednosti liste napraviti ćemo u metodi koja se aktivira na klik miša na dugme "Pohrani u listu".

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Dodjeljivanje vrijednosti elementima liste

- Želimo listu puniti vrijednostima koje korisnik unese u textBox1, tj. nakon što korisnik unese ime studenta u textBox1 i klikne na tipku "Pohrani u listu", želimo u listu **studenti** pohraniti unesenu vrijednost za novi element liste.
- Da bismo to učinili, potrebno je dvaput kliknuti na dugme "Pohrani u listu" kako bismo isprogramirali odgovor na događaj. U metodi button2_click upisat ćemo sljedeću naredbu:

```
studenti.Add(textBox1.Text);
```

- Ova naredba dodat će novi element liste i u njega pohraniti vrijednost iz textBoxa1.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Metode koje se mogu koristiti na listi

- Na listi je moguće koristiti ugrađene metode koje mogu npr.
 - dodati novi element liste (Add)
 - Prebrojati koliko je elemenata uneseno u listu (Count)
 - Prikazati koliko elemenata stane u listu (Capacity)
 - Obrisati listu (Clear)
 - i dr.
- U metodi `button1_click` osim dodavanja elementa liste, prebrojat ćemo i koliko je elemenata liste dodano, te koliki je kapacitet liste. Te ćemo podatke upisati u labele (`label6` i `label7`).
- Upotreba metoda `Count` i `Capacity`:
`label5.Text = "Uneseno je " + studenti.Count + " imena";`
`label7.Text = "Kapacitet liste je " + studenti.Capacity;`

Kapacitet liste se mijenja dinamički ovisno o broju unesenih elemenata u listu. Najprije se u memoriji otvara mjesta za 4 elementa, a ako se unese 4 elementa, ostavlja se 8 itd. kapacitet se automatski povećava.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Ispis elemenata liste

- Cijela metoda `button2_click()` izgleda ovako:
`private void button2_Click(object sender, EventArgs e)`

```
{
    studenti.Add(textBox1.Text);
    label5.Text = "Uneseno je " + studenti.Count + " imena";
    label7.Text = "Kapacitet liste je " + studenti.Capacity;
    textBox1.Text = "";
}
```

Naredba kojom će se isprazniti `textBox1.Text`

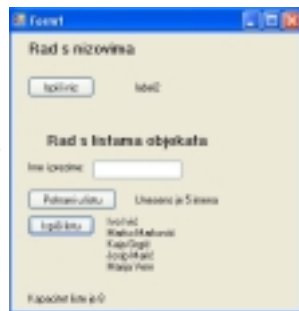
Da bismo ispisali elemente liste, potrebno je dvaput kliknuti na dugme "Ispiši listu", kako bismo se pozicionirali u metodu `button3_click()`. U njoj treba ispisati elemente liste u petlji iteracije `foreach`:

```
foreach (string student in studenti)
{
    label6.Text += student + "\n";
}
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pokretanje aplikacije za rad s listama

- Pokrenite debug-iranje aplikacije. Unesite nekoliko vrijednosti u `textBox1`, nakon unosa svake kliknite na tipku "Pohrani u listu." U `label5` će biti ispisano koliko je elemenata liste uneseno. Zatim kliknite na dugme "Ispiši listu". Prikazat će se rezultat kao na slici:



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Kolekcije

- Kolekcija (eng. collection) je generička klasa objekata koja se može koristiti za pohranjivanje i rukovanje skupom objekata istog tipa, a korespondira sa skupom klasa zvanim `CollectionBase`.
- Sadrži metode za dodavanje, uklanjanje elemenata, promjenu vrijednosti elemenata i dr.
- Elementi kolekcije su također indeksirani, indeks im počinje od 0 nadalje.
- Koriste se slično kao i liste.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklariranje kolekcije

- Za deklariranje kolekcije koristi se sintaksa:
`Collection<type> naziv_kolekcije = new Collection<type>();`
Npr. ako želimo deklarirati kolekciju `djelatnici`, tada bismo upisali:
`Collection<string> djelatnici = new Collection<string>();`
- Za dodavanje elemenata kolekciji koristi se metoda `Add`, npr. ako bismo željeli dodati 3 djelatnika u kolekciju, napisali bismo:
`Djelatnici.Add("Marko Marković");`
`Djelatnici.Add("Ivo Ivic");`
`Djelatnici.Add("Mara Marić");`
- Za brisanje nekog djelatnika iz liste napisali bismo:
`Djelatnici.Remove("Marko Marković");`
- Za brisanje cijele kolekcije napisali bismo:
`Djelatnici.Clear();`

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Ispis kolekcije

- Za ispis kolekcije na formu (npr. u labelu) koristi se petlja `foreach` na isti način kao i kod ispisa liste:

```
foreach (string djelatnik in djelatnici)
{
    label2.Text += djelatnik + "\n";
}
```

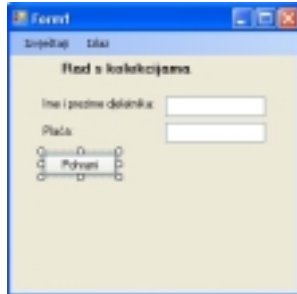
- Za prebrojavanje elemenata kolekcije koristi se (kao i kod liste) metoda `Count`, npr. ako u labelu 5 želimo ispisati koliko je elemenata u nekoj kolekciji:

```
Label5.Text= Djelatnici.Count;
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 9_2 – rad s kolekcijama kreirane klase

- U C# Express-u otvorite novu aplikaciju, nazovite je vježba9_2, te na formu dodajte ove kontrole kao na slici desno.
- Cilj vježbe je kreirati kolekciju djelatnika koja će biti tipa klase djelatnika. Klasa djelatnika će imati svojstva: imeDjelatnika i placaDjelatnika. Tim svojstvima treba unositi vrijednosti putem TextBox-ova u jednoj formi, a zatim u putem izbornika, napraviti izvještaj u drugoj formi (Form2)



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Kreiranje vlastite klase

- Da bismo za nekog djelatnika vodili podatke o imenu i prezimenu u plaći, potrebno je kreirati vlastitu klasu koju ćemo nazvati Djelatnik. Klasa Djelatnik imat će dva svojstva:
 - imePrezime (tipa string),
 - placa (tipa double).
- Klasu ćemo kreirati prema sintaksi za kreiranje klase navedenoj u poglavlju 8 (Klase, objekti), na sljedeći način:
 - Otvoriti unutar iste aplikacije u C#-u posebnu datoteku u koju ćemo upisati kod za kreiranje klase. Ta klasa će se moći koristiti u cijeloj aplikaciji.
 - Aktivirajte naredbu izbornika Project/Add New Item, pa u prozoru za izbor objekata izaberite CodeFile (jer ne otvaramo novu formu, nego novu datoteku za pisanje programskog koda). Kao naziv datoteke upišite Djelatnik.cs
 - U novu praznu datoteku Djelatnik.cs upišite ove naredbe za kreiranje nove klase:

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Kreiranje vlastite klase - nastavak

```
public class Djelatnik
{
    private string _imePrezime;
    public string ImePrezime
    {
        get { return _imePrezime; }
        set { _imePrezime = value; }
    }
    private double _placa;
    public double Placa
    {
        get { return _placa; }
        set { _placa = value; }
    }
    public Djelatnik(string imePrezime, double placa)
    {
        this.ImePrezime = imePrezime;
        this.Placa = placa;
    }
}
```

deklarirana klasa Djelatnik

svojstvo ImePrezime u klasi Djelatnik

svojstvo Placa u klasi Djelatnik

Metoda kojom se proslijedene vrijednosti za imePrezime i placa spremaju u svojstva te klase

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Kreiranje kolekcije klasa kao svojstva forme

- U datoteci Djelatnik.cs sada imamo kreiranu klasu Djelatnik sa svojstvima.
- Da bismo mogli tu klasu koristiti za više djelatnika, kreirat ćemo kolekciju klasa, i to kao svojstvo, jer je želimo prenositi kroz više formi iste aplikacije:
 - Ta će kolekcija sadržavati skup unesenih djelatnika, dakle ona će biti tipa <Djelatnik>.
 - Budući da je treba koristiti u više metoda i više formi, kreirat ćemo je za cijelu formu Form1 kao svojstvo (kasnije i u Form2), tako da u Form1.cs, odmah ispod deklaracije klase Form1 upišemo deklaraciju za kolekciju, ovako:

```
public partial class Form1 : Form
{
    private Collection<Djelatnik> _djelatnici = new
    Collection<Djelatnik>();
    public Collection<Djelatnik> Djelatnici
    {
        get { return _djelatnici; }
        set { _djelatnici = value; }
    }
}
```

ovo sve upisati u istoj liniji

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Unos vrijednosti u svojstva klase Djelatnik

- Da bismo u tako deklarirana svojstva unijeli vrijednosti koje korisnik unese u TextBox-ove, dvaput ćemo kliknuti na dugme "Pohrani" i u metodi Button1_Click() upisati naredbe za unos vrijednosti u klasu Djelatnik.
- Za te potrebe, moramo kreirati instancu klase djelatnik, nazvat ćemo je "d", i ta instanca može poprimiti vrijednosti unesene u TextBox-ove, tako ih da ih proslijedimo u zagradi. Naredbe izgledaju ovako:

```
private void button1_Click(object sender, EventArgs e)
{
    Djelatnik d = new Djelatnik(textBox1.Text,
    Convert.ToDouble(textBox2.Text));
    Djelatnici.Add(d);

    label4.Text = "Uneseno je " + Djelatnici.Count + " djelatnika.";
    textBox1.Text = "";
}
```

Kreira se instanca d klase Djelatnik, i njoj se proslijeduju u zagradi vrijednosti unesene u textBox1 i textBox2. Vrijednost iz textBox1 će se pohraniti u svojstvo imePrezime, a iz textBox2 u svojstvo placa.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pozivanje druge forme za ispis izvještaja

- S obzirom da izvještaj želimo ispisati u drugoj formi, kreirat ćemo u istoj aplikaciji formu Form2, i iz Form1 dvaput kliknuti na izbornik Izvještaji/ispis plaća da bismo došli u metodu u kojoj treba upisati naredbe za kreiranje nove instance forme 2 (nazvat ćemo je "Izvještaji", i proslijediti joj vrijednosti iz kolekcije Djelatnici ovako:

```
private void ispisPlacaToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 izvjestaji = new Form2();
    izvjestaji.Djelatnici = this.Djelatnici;

    izvjestaji.Show();
}
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Ispis izvještaja u Form2

- U Form2 najprije ćemo dodati kontrole labela za ispis podataka, kao na slici desno.
- Želimo da se prilikom učitavanja forme 2, u label4 i label5 upisuju podaci o imenu i prezimenu i plaći djelatnika.
- Dvaput kliknite na neko prazno mjesto na formi 2 i programskom kodu treba upisati naredbe koje slijede:



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklariranje kolekcije drugoj formi

- Najprije na vrhu forme2, prije svih metoda, na početku klase Form2 treba deklarirati kolekciju:
private Collection<Djelatnik> _djelatnici;
public Collection<Djelatnik> Djelatnici
{
 get { return _djelatnici; }
 set { _djelatnici = value; }
}
- Nakon toga, u metodi Form2_Load() treba upisati sljedeće naredbe za ispis svojstava svih djelatnika iz kolekcije:

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Ispis svojstava u drugoj formi

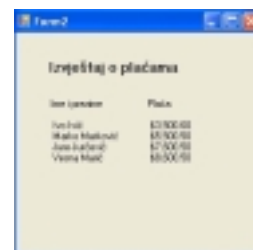
U petlji foreach se za svakog djelatnika u kolekciji Djelatnici ispisuju podaci. Prije petlje dobro je isprazniti tekst u labelama, kako se ne bi vidjele na ekranu oznake "label4" i "label5"

```
private void Form2_Load(object sender, EventArgs e)
{
    label4.Text = "";
    label5.Text = "";
    foreach (Djelatnik djelatnik in this.Djelatnici)
    {
        label4.Text += djelatnik.ImePrezime + "\n";
        label5.Text += djelatnik.Placa.ToString("c");
    }
}
```

Pri tome je:
Djelatnik = tip (klasa koju smo deklarirali u posebnoj .cs datoteci)
djelatnik = instanca tipa Djelatnik
Djelatnici = kolekcija koja je deklarirana kao svojstvo forme

Rezultat ispisa

- Pohranite aplikaciju i pokrenite debug. U prvoj formi u textBox-ove upišite podatke za četiri djelatnika (svaki put kliknite na dugme "Pohrani"). Zatim kliknite na izbornik "Izvještaji" i "Ispis plaća". Na ekranu će se pojaviti druga forma, s ispisom kao na slici:



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Zaključak

- Ako želimo više objekata pohraniti u jedan zajednički skup objekata, gdje će objekti biti indeksirani, koristit ćemo ili nizove, ili liste ili kolekcije
- Nizove, liste i kolekcije najprije treba deklarirati, a zatim im dodijeliti vrijednosti
- Ako se vrijednosti liste ili kolekcije učitavaju iz forme, vrijednosti se pohranjuju s pomoću petlje foreach u kojoj se deklarira neka instanca koja predstavlja element te liste ili kolekcije
- Ako želimo elemente liste ili kolekcije koristiti kroz više formi neke aplikacije, potrebno je:
 - U posebnoj datoteci te aplikacije kreirati vlastitu klasu objekata (npr. Djelatnik)
 - U svakoj formi aplikacije deklarirati svojstva koja su tipa kolekcije klase
 - U metodi koja treba ispisati ili učitati vrijednosti svojstvima koristiti petlju foreach u kojoj se deklarira instanca klase

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pitanja za ponavljanje

- Što je niz?
- Kada se preporuča upotreba niza?
- Koje vrste nizova postoje?
- Koja je opća sintaksa za deklaraciju niza?
- Koja je sintaksa ako želite deklarirati niz i kreirati instancu niza (u jednoj naredbi)?
- Deklarirajte niz i kreirajte njegovu instancu koja će se zvati **nazivArtikla**, a sadržavat će 10 elemenata niza, koji će biti tipa string (tekstualnog) – sve u jednoj naredbi
- Što će napraviti naredba: **imeStudenta = new string[5];** ?
- Ispišite elemente niza koji je deklariran ovom naredbom i njihove standardne (default) vrijednosti:
student = new string[5];
- Deklarirajte niz Ocjene koji će sadržavati cjelobrojne vrijednosti za 5 ocjena. Zatim napišite u for petlji naredbe kojima će se dodjeljivati vrijednosti svim elementima niza. Neke im se u petlji dodijele vrijednosti od 1 do 5.
- Kojom naredbom ćete sortirati elemente niza "studenti"?

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pitanja za ponavljanje - nastavak

11. Deklarirajte niz koji će sadržavati cjelobrojne vrijednosti za 5 ocjena, i u istoj naredbi im dodijelite vrijednosti od 1 do 5. Zatim napišite u foreach petlji naredbu kojom će se ispisati vrijednosti svih 5 elemenata tog niza u labeli label1 (svaki element u novi red labela).
12. Što su liste?
13. Kada se preporuča upotreba lista?
14. Nabrojite barem 4 metode koje se mogu koristiti na elementima liste.
15. Koja je sintaksa za deklaraciju liste?
16. Deklarirajte listu koja će sadržavati nazive artikala.
17. Deklarirajte listu koja će sadržavati cijene artikala.
18. Što će napraviti naredba: imeStudenta.Add(textBox1.Text); ?
19. Napišite naredbu kojom će se iz vrijednost unesena u textBox1 pohraniti u listu ocjenaStudenta koja je cjelobrojnog tipa.
20. Ako je već prije deklarirana lista "imeStudenta", kojom naredbom biste ispisali u labelu koliko elemenata ima ta lista?

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pitanja za ponavljanje - nastavak

21. Što će napraviti ova petlja u nekoj metodi:
foreach (string student in studenti)
{
 label6.Text += student + "\n";
}
22. Deklarirajte listu "cijenaArtikla" realnog tipa u koju iz textBox1 učitajte i pohranite vrijednost.
23. Deklarirajte listu "cijenaArtikla" realnog tipa i u petlji foreach u textBox1 ispišite vrijednosti svih elemenata liste.
24. Što radi naredba nazivListe.Add();
25. Što radi naredba nazivListe.Capacity();
26. Što radi naredba nazivListe.Count();
27. Što radi naredba nazivListe.Clear();
28. Ako neku listu želimo koristiti u svim metodama neke forme, gdje ćemo upisati deklaraciju liste?

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pitanja za ponavljanje - nastavak

29. Što su kolekcije?
30. Koja je sintaksa za deklaraciju kolekcije?
31. Deklarirajte kolekciju "imeDjelatnika" i u nju dodajte ime za jednog djelatnika koji se zove "Luka Marković".
32. Deklarirajte kolekciju "sifraDjelatnika" cjelobrojnog tipa i u nju dodajte šifru za jednog djelatnika "1010".
33. Kojom naredbom biste iz liste "imeDjelatnika" obrisali djelatnika "Luka Marković"?
34. U petlji foreach ispišite sva imena djelatnika iz liste "imeDjelatnika".
35. U petlji foreach ispišite sve plaće djelatnika iz liste "imeDjelatnika".
36. Ako želimo kreirati vlastitu klasu, koja će se koristiti u svim formama aplikacije, u kojoj ćemo datoteci to napraviti?
37. Deklarirajte vlastitu klasu pod nazivom "Artikal". Klasa neka ima samo dva svojstva: nazivArtikla i cijenaArtikla.
38. Napišite metodu kojom se proslijedene vrijednosti za nazivArtikla i cijenaArtikla spremaju u svojstva te klase.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pitanja za ponavljanje - nastavak

39. Što će napraviti ove naredbe:

```
private Collection<Djelatnik> _djelatnici = new  
Collection<Djelatnik>();  
public Collection<Djelatnik> Djelatnici  
{  
    get { return _djelatnici; }  
    set { _djelatnici = value; }  
}
```
40. Što će napraviti ova naredba:

```
Djelatnik d = new Djelatnik(textBox1.Text,  
Convert.ToDouble(textBox2.Text));
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Literatura

- J. Liberty, Programming C#, Fourth edition, O'Reilly and Associates, 2005.
- Prijevod: C# programiranje, A. Dragosavljević (ur.), Dobar plan, 2005.
- Predavanja i zadaci na web stranici kolegija
- Web izvori

Razvoj poslovnih aplikacija, M.Zekić-Sušac