

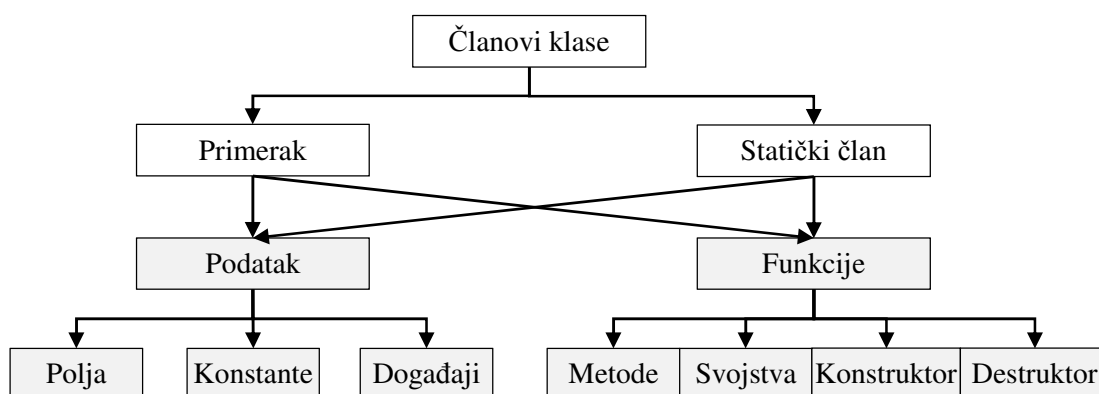
4.4. KLASSE I WIN FORME

U ovom poglavlju pokazaće se kako se formiraju klase i uključuju u projekat. Sama klasa predstavlja kompleksan tip podataka, slika 25., odnosno kompleksnu strukturu koja može da se satoji od podataka raličitog tipa (*string*, broj: *int*, *float*,...; *char*, *niz*). Drugim rečima, može se reći da klasa predstavlja uošteni šablon na osnovu koga se grade objekti. Komponente klase su atributi – članovi klase (*members*) i akcije – metode (*methods*). Uz pomoć klase mogu se formirati, kao što je već rečeno, virtualni objekti davanjem vrednosti atributima i korišćenjem definisanih metoda.

4.4.1. KLASA U C#

Već je pomenuto da se na osnovu klase se formiraju objekti. Objekti sadrže podatke i metode za manipulisanje i pristup podacima, pri čemu sama klasa određuje koje podatke mora da sadrži svaki konkretan objekat. Naravno, sama klasa ne sadrži nikakve podatke.

Za naziv konkretnog objekta, u literaturi se može naći naziv primerak ili instanca. Kada se formira klasa, u stvari se uređuju njeni članovi u entiteti koji se opisuju atributima. Različiti entitet mogu da imaju iste ili različite vrednosti definisanih atributa. Dok strukture predstavljaju vrednosni tip podataka, klase daje veću fleksibilnost i više mogućnosti s obzirom da klase podržavaju nasleđivanje. Ovo u stvari znači da su klase referentni tip podataka. Klase, slika 25., predstavljaju osnovne i značajne objekte u C# programskom jeziku i rad sa njima je jednostavniji nego u programskom jeziku C++, mada postoji i velika sličnost.



Slika 25. Struktura klase u C#

Da bi se shvatila klasa u programiranju, najbolje je posmatrati primere iz realnog sveta, pa će se shodno tome posmatrati objekat knjiga. Knjiga se kao objekat opisuju sledeće osobine: naziv, autor, žanr, izdavač, godina izdanja i slično. Klasa koja će opisati knjigu treba da opiše sva svojstva koje neka knjiga opisuje, ali ta klasa ne predstavlja ni jednu određenu knjigu. Kako, u realnom svetu, postoje različiti primerci knjiga od različitih autora, izdavača to znači da bi svaki primerak knjige bio prepoznat kao primerak klase Knjiga.

Kada se stvara klas, moramo se imati u vidu, da se stvara novi tip podataka. Kao takav on predstavlja kompleksan tip podataka. Kompleksni tipovi podataka se sastoje od jednostavnih tipova podataka. Između vitičastih zagrada koje označavaju početak i kraj klase (bloka naredbi) upisuju se članovi i metode te klase.

4.4.2. PRIMER KLASA U C#

Kao ilustraciju za rad sa klasama u C#, uradiće se win forma, slika 26 i tri klase: *Knjiga1*, *Knjiga2* i *Udzbenik*. Klase *Knjiga1* i *Knjiga2* su po strukturi iste (u realnom programiranju to se NIKADA ne radi), a ovde služe za ilustraciju različitog načina kodiranja kako bi se uočile razlike. Treća klasa *Udzbenik* je specifična jer nasleđuje klasu *Knjiga1*. U samoj formi su: 8 labela (L1 – L8, slika 26.), 9 textbox – ova (*Naziv* – T1, *Ime* – T2, *Prezime* – T3, *Tiraz* – T4, *Izdavac* – T5, *Godina* – T6, *Tekst1* – T7, *Tekst2* – T8 i *Tekst3* – T9, slika 26. i dva komanda dugmeta: *Izlaz* – D1 i *Dodati* – D2, slika 26.

U formi je definisano svojstvo **KeyPreview** na **True**, tako da se u događaju **KeyPressDown** dodaje procedure za prekid rada u programu – PK 25. Dugme za izlaz (**Izlaz**) ima istu funkciju kao i pritisak tastera Esc, što znači da je potrebno ispisati kod u događaju **Click**, PK – 26.

U tabeli T – 9., dati su postavljeni parametri za sve objekte sa slike 26.

PK – 25. Programski kod za zatvaranje forme kada se pritisne taster **Esc – KeyDown**

```

1 private void PrimerKlase_KeyDown(object sender, KeyEventArgs e)
2 {
3     if (e.KeyCode == Keys.Escape)
4         Application.Exit();
5 }

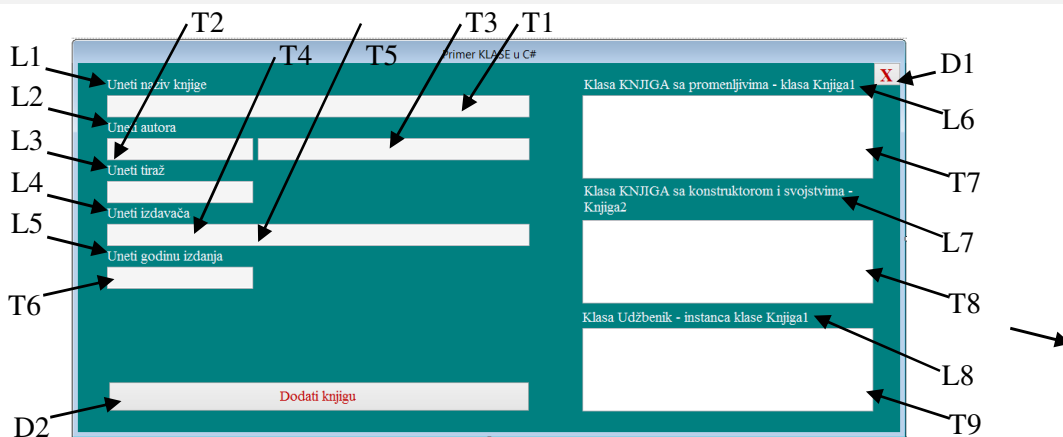
```

PK – 26. Programski kod za zatvaranje forme kada se pritisne dugme (D1) – **Izlaz**

```

1 private void Izlaz_Click(object sender, KeyEventArgs e)
2 {
3     Application.Exit();
4 }

```



Slika 26. Forma za rad sa klasama

Tabela T – 9. Podešena svojstva fome i njenih objekata za primer sa slike 26

Svojstvo	Vrednost
	Label (L1)
Location	34, 18
Text	Uneti naziv knjige

Svojstvo	Vrednost
	Label (L2)
Location	34, 74
Text	Uneti autora

Svojstvo	Vrednost
	Label (L3)
Location	34, 130
Text	Uneti tiraž

Svojstvo	Vrednost
	Label (L4)
Location	34, 186
Text	Uneti izdavača

Svojstvo	Vrednost
	Label (L5)
Location	34, 242
Text	Uneti godinu izdanja

Svojstvo	Vrednost
	Label (L6)
Location	657, 18
Text	Klasa KNJIGA sa promenljivima - klasa Knjiga1

Svojstvo	Vrednost
	Label (L7)
Location	657,157
AutoSize	False
Size	382, 45

Svojstvo	Vrednost
	Label (L8)
Location	657, 322
AutoSize	False
Size	306, 21

Text	Uneti godinu izdanja
------	-----------------------------

Text	Klasa KNJIGA sa promenljivima - klasa Knjiga1
------	--

	Textbox (T1)
(Name)	Naslov
BackColor	WhiteSmoke
ForeColor	Teal
Location	38, 42
MultiLine	False
Size	552, 29

	Textbox (T2)
(Name)	Ime
BackColor	WhiteSmoke
ForeColor	Teal
Location	38, 98
MultiLine	False
Size	191, 29

Nastavak tabele T – 9.

	Textbox (T3)
(Name)	Prezime
BackColor	WhiteSmoke
ForeColor	Teal
Location	235, 98
MultiLine	False
Size	355, 29

	Textbox (T4)
(Name)	Tiraz
BackColor	WhiteSmoke
ForeColor	Teal
Location	38, 154
MultiLine	False
Size	191, 29

	Textbox (T5)
(Name)	Izdavac
BackColor	WhiteSmoke
ForeColor	Teal
Location	38, 210
MultiLine	False
Size	552, 29

	Textbox (T6)
(Name)	Godina
BackColor	WhiteSmoke
ForeColor	Teal
Location	38, 266
MultiLine	False
Size	191, 29

	Textbox (T7)
(Name)	Tekst1
BackColor	WhiteSmoke
ForeColor	Teal
Location	659,42
MultiLine	True
Size	380, 109

	Textbox (T8)
(Name)	Tekst2
BackColor	WhiteSmoke
ForeColor	Teal
Location	659,205
MultiLine	True
Size	380, 109

	Textbox (T9)
(Name)	Tekst3
BackColor	WhiteSmoke
ForeColor	Teal
Location	659,42
MultiLine	True
Size	380, 109

	Forma
(Name)	PrimerKlase
AutoScaleMode	None
BackColor	Teal
ControlBox	False
Font	Times New Roman, Regual, 14pt
ForeColor	GhostWhite
KeyPreview	True
Size	1090, 521
StartPosition	Center Screen
Text	Primer KLASE u C#

	Button (D1)
(Name)	Izlaz
ForeColor	Red
Location	1039,0
Size	36,31
Text	X

	Button (D2)
(Name)	Dodati
ForeColor	Red
Location	40, 416
Size	550, 39
Text	Dodati knjigu

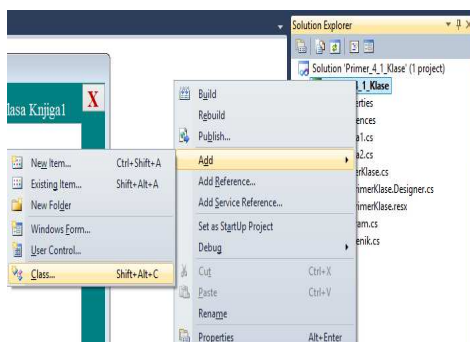
Kada se formira forma, sa svim željenim objektima, naredni korak je da se formiraju klase. Klase u *C#* se dodaju tako što se u *Solution Explorer* – u na ime projekta klikne desnim tasterom miša, pa se pokaže na *Add* i na ponuđenom meniju se izabere *Class*, slika 27 ili se u isto vreme pritisnu tasteri *Alt + Shift + C*.

Nakon ove operacije, pojaviće se meni dijalog prozor za dodavanje novog objekta, slika 28., u kome treba izabrati tip objekta (u ovom slučaju *Class*) i dati ime tom objektu. Ovde treba voditi računa da se, pri davanju imena objekta, ne izbriše nastavak *.cs*.

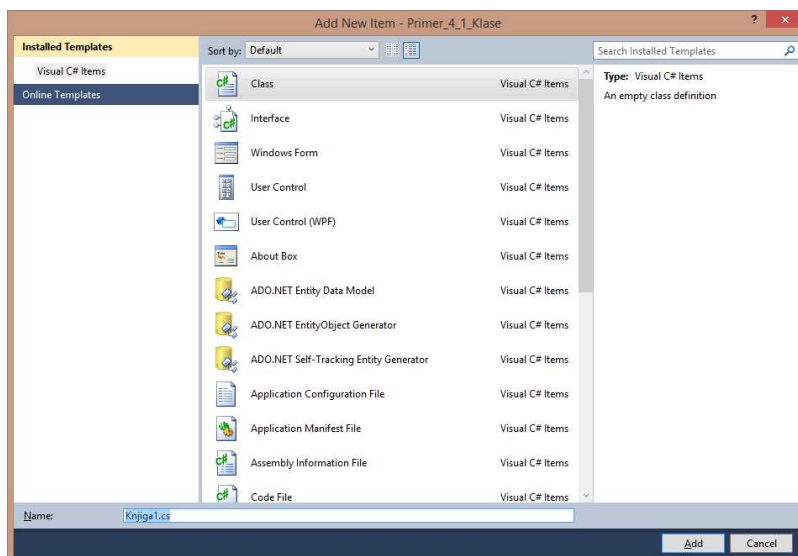
Kada se da naziv klasi, i klikne na *Add* odmah se otvara editor za pisanje programskog koda, PK – 27. Sada je potrebno definisati sve potrebne atribute koji će definisati klasu. Potreban kod se upisuje od linije 5. i u kodovima PK – 28, PK – 29 i PK 30, prikazan je način kako se formiraju klase.

U ovom primeru, formiraće se klase:

- **Knjiga1** sa atributima (PK – 28):
 - *Naslov (string); Ime (string); Prezime (string); Tiraz (int); Izdavac (string); Godina (int)* i
 - funkcijom *Autor()* koja vraća ime i prezime autora.
- **Knjiga2** sa atributima (PK – 29):
 - *Naslov (string); Ime (string); Prezime (string); Tiraz (int); Izdavac (string); Godina (int)* ,
 - funkcijom *Autor()* koja vraća ime i prezime autora i
 - konstruktorom *Knjiga2()*.
- **Udzbenik** (PK – 30);
 - koji nasleđuje klasu *Knjiga1* i
 - i atributom *Predmet* koji ima default vrednost “*Uvod u VP*”.



Slika 27. Postupak za dodavanje nove klase u projekat



Slika 28. Postupak za dodavanje nove klase (*Knjiga1.cs*) u projekat

PK – 27. Početni izgled editora za definisanje klase *Knjiga1*

```
1 namespace Primer_4_1_Klase
2 {
3     class Knjiga1
4     {
5
6     }
7 }
```

PK – 28. Programski kod za formiranje klase *Knjiga1*

```
1 namespace Primer_4_1_Klase
2 {
3     class Knjiga1
4     {
5         public string Naslov;
6         public string Ime;
7         public string Prezime;
8         public int Tiraz;
9         public string Izdavac;
10        public int Godina;
11        public string Autor()
12        {
13            return Ime + " " + Prezime;
14        }
15    }
16 }
```

PK – 29. Programski kod za formiranje klase *Knjiga2*

```
1 namespace Primer_4_1_Klase
2 {
3     class Knjiga2
4     {
5         public string Naslov { get; set; }
6         public string Ime { get; set; }
7         public string Prezime { get; set; }
8         public int Tiraz { get; set; }
9         public string Izdavac { get; set; }
10        public int Godina { get; set; }
11        public string Autor()
12        {
13            return Ime + " " + Prezime;
14        }
15        public Knjiga2(string qNaslov, string qIme, string qPrezime, int
16                        qTiraz, string qIzdavac, int qGodina)
17        {
18            Naslov = qNaslov;
19            Ime = qIme;
20            Prezime = qPrezime;
21            Tiraz = qTiraz;
22            Izdavac = qIzdavac;
23            Godina = qGodina;
24        }
25    }
26 }
```

PK – 30. Programski kod za formiranje klase *Udzbenik*

```
1 namespace Primer_4_1_Klase
```

```

2  {
3      class Udzbenik : Knjiga1
4      {
5          private string _udzbenik="Uvod u VP";
6          public string Predmet
7          {
8              get
9              {
10                 return _udzbenik;
11             }
12             set
13             {
14                 _udzbenik = value;
15             }
16         }
17     }
18 }

```

Klasa *Knjiga1* – PK – 28., sadži javne promenjive (linije 5 – 10) i samo sa njima su izgrađena, takozvana, **fields** (polja) klase. Pored polja, klasa sadži samo jednu metodu, **Autor()**, (linije: 11 – 14) koja vraća vrednost ime i prezime autora knjige. I pored toga što se ne vidi, da ova klasa, ima konstruktor, ona ga u stvari ima, kao podrazumevan i kao takav, on je tokom pisanja naredbi nevidljiv. Ovaj, podrazumevani konstruktor se dodaje programskom kodu tokom kompajliranja. Drugim rečima, kad god se izostavi konstruktor, kompajler ga dodaje jedan prazan bez ikakvih parametara. Klasa ne sadži podrazumevani konstruktor jedino ako se napiše barem jedan konstruktor u klasi, bez obzira da li on traži atribute ili ne.

Za razliku od klase *Knjiga1*, klasa *Knjiga2* (PK – 29.), umesto polja koristi **auto – property** (auto – svojstva), linije: 5 – 10. Ovaj način obezbeđuje pisanje bezbednijeg programskog koda. U stvari, to predstavlja enkapsulaciju. U ovom kodu svojstva se mogu i čitati (**get;**) i upisivati (**set;**). Pored definisanih svojstava, ova klasa ima i konstruktor, koji zahteva šest atributa (linije 15 – 23). Kao što se vidi, iz linije 15., konstruktor je specijalna metoda kojoj je dato isto ime kao i klasi. Pošto je ovde definisan konstruktor, drugi joj neće, tokom kompajliranja, biti dodeljen. Ovo znači da prilikom inicijalizacije klase moraju se imati uneseni atributi, inače klasa se ne možete inicijalizovati. U realnim projektima, uobičajeno je, da se formira još jedan konstruktor koji ne zahteva nikakve atribute, tako da se klasa može pozvati i bez atributa.

Treća klasa, koja je formirana, je klasa *Udzbenik* (PK – 30.). Ova klasa je naslednik klase *Knjiga1* (polimorfizam). Kao što se vidi iz linije 3., nasleđivanje klase se vrši dodavanjem dvotačke (:) iza naziva (*Udzbenik*) klase koja se formira i dodavanje naziva klase (*Knjiga1*) koju nasleđuje.

Nasleđivanjem klase *Knjiga1*, automatski je klasa *Udzbenik* nasledila sva svojstva i metod klase *Knjiga1*. Naravno, u novoj klasi može se dodati novi atribut (ili više atributa), **Predmet**, linija 6, kojoj je dodeljena podrazumevana vrednost: “Uvod u VP”, linije 5 – 16.

Formiranje primeraka, na osnovu klase, prikazano je u kodu PK – 31. Formiranje primerka se izvodi na osnovu vrednosti *textbox* – ova, slika 26. Prvo se ispituje da li je uneta bilo koja vrednost u te *textbox* – ove. Za tu namenu, uvode se promenljive *provera1* – *provera6* i u te promenljive se smeštaju unete vrednosti u *textbox* – ove. Provera, da li su unete vrednosti, se izvodi u liniji 9., koristeći funkciju: **string.IsNullOrEmpty(string)**. Ukoliko je uslov iz linije 9. istinit, generiše se poruka da moraju biti uneti svi podaci iz forme, linija 10., u suprotnom se kreiraju primerci za svaku klasu.

Primerak klase, *Knjiga1*, se formira u liniji 14. Znači, prvo se pozove klasa, čiji primerak se formira, *Knjiga1*, upiše se ime primerka, *knjiga1*, pa se iza znaka jednakosti (=) unese ključna reč **new** i konstruktor klase *Knjiga1()*.

Dodeljivanje vrednosti atributima primerka klase, *knjiga1*, se izvodi tako što se napiše naziv primerka, tačka (.) i naziv atributa, a posle znaka jednakosti (=) upiše se vrednost koja se dodeljuje, linije od 15 – 20 i od 37 – 42. U linijama 18, 20, 28, 40 i 42; je upotrebljena funkcija **Int32.Parse(string)**, koja navedenu *string* promenljivu, u

zagradama (*string1*), konvertuje u celobrojnu vrednost. Ovo je iz razloga što je tip atributa *Tiraz* i *Godina*, u klasi *Knjiga1*, celobrojna vrednost (*int*), a tipovi podataka koji se unose u formu su tipa *string*, pa je potrebno konvertovanje. Na isti način, kao što se dodaju, tako se i preuzimaju vrednosti atributa iz primerka klase, navodeći naziv primerka, tačku (.) i naziv atributa.

U liniji 28., je dat kod za definisanje primerka klase *Knjiga2*, sa konstruktorom *Knjiga2(...)*. Parametri (...), koji se navode u zagradi, se moraju navesti u redosledu koji je dat u PK – 29 linija 15.

Na kraju u liniji 36, PK – 31. dat je način formiranja primerka *udzbenik*, klase *Udzbenik*.

PK – 31. Programski kod za događaj *Event* dugmeta *Dodati*

```
1 private void Dodati_Click(object sender, EventArgs e)
2 {
3     string provera1 =Naslov.Text.Trim();
4     string provera2 = Ime.Text.Trim();
5     string provera3 = Prezime.Text.Trim();
6     string provera4 = Tiraz.Text.Trim();
7     string provera5 = Izdavac.Text.Trim();
8     string provera6 = Godina.Text.Trim();
9     if (string.IsNullOrEmpty(provera1) &&
10         string.IsNullOrEmpty(provera2) &&
11         string.IsNullOrEmpty(provera3) &&
12         string.IsNullOrEmpty(provera4) &&
13         string.IsNullOrEmpty(provera5) &&
14         string.IsNullOrEmpty(provera6))
15         MessageBox.Show("Marate uneti sve podatke!!!");
16     else
17     {
18         //Klasa Knjiga1
19         Knjiga1 knjiga1 = new Knjiga1();
20         knjiga1.Naslov = Naslov.Text.Trim();
21         knjiga1.Ime = Ime.Text.Trim();
22         knjiga1.Prezime = Prezime.Text.Trim();
23         knjiga1.Tiraz= Int32.Parse(Tiraz.Text.Trim());
24         knjiga1.Izdavac= Izdavac.Text.Trim();
25         knjiga1.Godina = Int32.Parse(Godina.Text.Trim());
26         Tekst1.Text = "";
27         Tekst1.Text += knjiga1.Naslov;
28         Tekst1.Text += "\r\n"+knjiga1.Autor();
29         Tekst1.Text += "\r\nTiraž " +knjiga1.Tiraz.ToString();
30         Tekst1.Text += "\r\nIzdavač " + knjiga1.Izdavac;
31         Tekst1.Text += "\r\nGodina izdanja " + knjiga1.Godina.ToString();
32         //Klasa Knjiga2
33         Knjiga2 knjiga2 = new Knjiga2(Naslov.Text.Trim(), Ime.Text.Trim(),
34             Prezime.Text.Trim(),
35             Int32.Parse(Tiraz.Text.Trim()),
36             Izdavac.Text.Trim(),
37             Int32.Parse(Godina.Text.Trim()));
38
39         Tekst2.Text = "";
40         Tekst2.Text += knjiga2.Naslov;
41         Tekst2.Text += "\r\n"+knjiga2.Autor();
42         Tekst2.Text += "\r\nTiraž " +knjiga2.Tiraz.ToString();
43         Tekst2.Text += "\r\nIzdavač " + knjiga2.Izdavac;
44         Tekst2.Text += "\r\nGodina izdanja " + knjiga2.Godina.ToString();
45         //Klasa Udžbenik
46         Udzbenik udzbenik = new Udzbenik();
47         udzbenik.Naslov = Naslov.Text.Trim();
```

```

38     udzbenik.Ime = Ime.Text.Trim();
39     udzbenik.Prezime = Prezime.Text.Trim();
40     udzbenik.Tiraz = Int32.Parse(Tiraz.Text.Trim());
41     udzbenik.Izdavac = Izdavac.Text.Trim();
42     udzbenik.Godina = Int32.Parse(Godina.Text.Trim());
43     Tekst3.Text = "";
44     Tekst3.Text += udzbenik.Naslov;
45     Tekst3.Text += "\r\n" + udzbenik.Autor();
46     Tekst3.Text += "\r\nTiraž " + udzbenik.Tiraz.ToString();
47     Tekst3.Text += "\r\nIzdavač " + udzbenik.Izdavac;
48     Tekst3.Text += "\r\nGodina izdanja " + udzbenik.Godina.ToString();
49     Tekst3.Text += "\r\nUdžbenik iz predmeta " + udzbenik.Predmet;
50     }
51 }

```

U programskim linijama od 21 – 26, 29 – 34 i 43 – 49, u *textbox* – ovima se upisuju vrednosti iz atributa primeraka formiranih klasa.