

9. Svojstva (properties)



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Što ćete naučiti u ovom poglavlju?

- Što su svojstva i kada se koriste
- Kako se deklariraju svojstva
- Kako se koristi svojstvo za razmjenu vrijednosti između više formi

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Svojstva (properties)

Svojstva (eng. properties) su opisi objekata u objektnom programiranju, i omogućuju enkapsulaciju podataka na sljedeći način:

- Svojstva pripadaju nekoj klasi i definiraju se za klasu (npr. klasa `textBox` ima svojstvo `Text`). To svojstvo pozivamo sa `textBox1.Text`. Svojstvu se mogu dodijeliti neke vrijednosti, a korisnik može pristupiti tim vrijednostima preko klase (tako da se najprije navede naziv klase, "...", zatim naziv svojstva, npr.

```
textBox1.Text="Neki tekst";
```

- Princip enkapsulacije u objektnom programiranju ne dozvoljava da tom svojstvu pristupimo direktno, nego navođenjem naziva klase. Kada mu pristupimo preko klase, tada možemo manipulirati vrijednostima tog svojstva.
- Do sada smo koristili već ugrađena svojstva kontrola u C#. Međutim, u programu možemo i sami definirati svojstva klase prema potrebi.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Definiranje svojstva

- Npr. ako u aplikaciji imamo više formi, tada je svaka forma zasebna klasa. Unutar jedne klase možemo pristupiti svojstvima i varijablama samo te klase.
- Što ako u prvoj formi unesemo neke podatke (npr. imePrezime), a zatim želimo pozvati drugu formu u kojoj ćemo te podatke ispisati na ekranu?
- Ako su podaci unijeti u jednoj formi (klasi Form1), program u drugoj formi (klasi Form2) zbog enkapsulacije ne zna koje su vrijednosti unesene, i ne može ih ispisati.
- Kako dohvatiti iz jedne klase vrijednosti iz druge klase?
- Odgovor je : kreirati svojstvo (property) putem koje ćemo dohvatiti vrijednost nekih varijabli iz klase.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – rad sa svojstvima

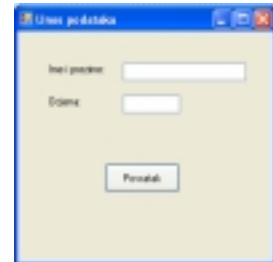
- Otvorite projekt pod nazivom Vježba8. Cilj nam je:
- 1. na prvoj formi kliknuti na dugme "Unos podataka" i pri tome otvoriti drugu formu na kojoj će se napraviti unos podatka o imenu i prezimenu studenta, te ocjeni
- 2. postaviti izbornik s pomoću kojeg ćemo u istoj formi ispisati podatke na ekranu



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – nastavak

- Na prvoj formi, s pomoću kontrole `MenuStrip` napravite dva izbornika: `Ispis`, u kojem u nižoj razini napravite element: `Ispis izvještaja`, te `Kraj rada`, u kojem napravite element: `Kraj`.
- Kreirajte i drugu formu na kojoj ćete dodati ove kontrole za unos podataka kao na slici desno:



Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 - nastavak

- Vratite se u prvu formu (Form1), gdje ćemo u upisati odgovor na događaj `button1_Click` (kada korisnik klikne na dugme "Unos podataka")
- U metodi `button1_Click()` želimo pozvati drugu formu (Form2), pa ćemo upisati naredbu za aktiviranje konstruktora koji će kreirati novi objekt klase Form2
- Naredba za konstruktor izgledat će ovako:

```
Form2 formaUnos = new Form2();
```

Ovo je tip nove forme (novog objekta kojeg kreiramo)

voj poslovnih aplikacija, M

Ovo je naziv nove forme (novog objekta kojeg kreiramo) i može biti proizvoljan

Vježba 8 - nastavak

- U toj metodi treba upisati i naredbu za prikaz nove forme, pa će obje naredbe izgledati ovako:

```
Form2 formaUnos = new Form2();  
formaUnos.ShowDialog();
```

 - ovo je nova naredba koja omogućava da se druga forma prikaže, ali da korisnik ne može prijeći klikom miša nazad u prvu formu dok ne zatvori drugu
- U drugoj formi (Form2) kreirali smo kontrole za unos podataka za ime i prezime, te Ocjenu. Premjestite se u Form2, te postavite pokazivač u `button1_Click()`. U ovoj metodi trebamo upisati naredbe kojima će se vrijednosti koje korisniku unese pohraniti u varijable.
- Kada bismo imali samo jednu formu, i jednu metodu, tada bi te naredbe izgledale ovako:

```
string imePrezimeStudenta = textBox1.Text;  
int ocjenaStudenta = Convert.ToInt32(textBox2.Text);
```
- Međutim, tada koristimo samo privatne varijable čije vrijednosti može dohvatiti samo ta metoda i ne možemo ih ispisati u drugoj formi. Dakle, ovaj način ne funkcionira ako imamo više formi.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Enkapsulacija pomoću svojstva

- Jedno rješenje je: varijable `imePrezimeStudenta` i `ocjenaStudenta` deklarirati kao `public` na razini cijele aplikacije (u metodi `Main()`). Tada bi sve forme mogle dohvatiti te javne varijable. Međutim, to rješenje nije preporučljivo.
- Zašto? Jer narušava princip enkapsulacije. Da bismo zadržali enkapsulaciju podataka (da varijable ostanu privatne za klasu), tada koristimo svojstva (properties) koje definiramo za klasu i preko tih svojstva možemo dohvaćati vrijednosti
- Kako definirati novo svojstvo? Unutar klase (a prije metoda koje opisuju ponašanje klase) navesti deklaraciju svojstva.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Deklaracija svojstva

Opći oblik naredbi za deklaraciju novog svojstva za neku klasu je - na primjeru za svojstvo Hour (Liberty, 2005):

```
private int _hour;  
public int Hour  
{  
    get  
    {  
        return _hour;  
    }  
    set  
    {  
        _hour = value;  
    }  
}
```

Definiramo privatnu varijablu čiji naziv počinje s "_", npr. "_hour"

Definiramo javno svojstvo čiji naziv počinje s velikim slovom naziva varijable npr. "Hour" - tako naglašavamo razliku između privatne varijable i javnog svojstva

Naredbom "get" definiramo da će se vrijednost tog svojstva vraćati (moći dohvatiti) kroz varijablu _hour

Naredbom "set" definiramo da će se privatnoj varijabli _hour moći dodijeliti vrijednost tog svojstva

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba8 - Deklaracija svojstva

- Ako u prvoj formi (Form1) za dohvaćanje imena i prezimena studenta kreiramo novo svojstvo, deklaracija će izgledati ovako:

```
private string _imePrezimeStudenta;  
public string ImePrezimeStudenta  
{  
    get  
    {  
        return _imePrezimeStudenta;  
    }  
    set  
    {  
        _imePrezimeStudenta = value;  
    }  
}
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba8 - Deklaracija svojstva

U prethodnom primjeru smo definirali svojstvo `ImePrezimeStudenta` i dodijelili ga klasi Form1. To znači da će naša prva forma (koja je instanca klase Form1):

- naslijediti to svojstvo od klase Form1
- moći dohvatiti vrijednost tog svojstva u privatnoj varijabli `_imePrezimeStudenta`
- Moći dodijeliti vrijednost tom svojstvu putem privatne varijable `_imePrezimeStudenta`

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba8 - Deklaracija svojstva

- Na isti način trebamo dodijeliti i svojstvo `OcjenaStudenta` klasi `Form1`. Ispod prethodne deklaracije za svojstvo `ImePrezimeStudenta` upišite naredbe za deklaraciju svojstva `OcjenaStudenta` (ispravite tip koji ne treba biti `string`, nego `int`):

```
private int _ocjenaStudenta;  
public int OcjenaStudenta  
{  
    get { return _ocjenaStudenta; }  
    set { _ocjenaStudenta = value; }  
}
```

Vitičaste zagrade mogu biti u istom ili u drugom retku neke naredbe. Zbog štednje prostora na slajdu napisane su u istom retku.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba8 - Deklaracija svojstva

- Budući da ćemo ista svojstva: `ImePrezimeStudenta` i `OcjenaStudenta` trebati koristiti i u drugoj formi, prethodne dvije deklaracije svojstava koje smo napravili kopirajte u programski kod za `Form2` (priđite u drugu formu `Form2`, postavite pokazivač na programski kod forme, te na mjesto ispod deklaracije forme `Form2`, a iznad metoda), ovako:

```
public partial class Form2 : Form  
{  
    private string _imePrezimeStudenta;  
    public string ImePrezimeStudenta  
    {  
        get { return _imePrezimeStudenta; }  
        set { _imePrezimeStudenta = value; }  
    }  
    private int _ocjenaStudenta;  
    public int OcjenaStudenta  
    {  
        get { return _ocjenaStudenta; }  
        set { _ocjenaStudenta = value; }  
    }  
}
```

Deklaracija svojstava kopirana u Form2

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – Upotreba svojstava kroz više formi

- Da bismo javna svojstva koja smo deklarirali mogli koristiti u obje forme, u programski kod prve forme (`Form1.cs`) u metodi `button1_Click` treba dodati ovu naredbu čiji je opći oblik sljedeći:

```
this.naziv_svojstva = naziv_forme.naziv_svojstva;
```

U našem primjeru iz Vježbe 8 naredba bi bila:

```
this.ImePrezimeStudenta = formaUnos.ImePrezimeStudenta;
```

Na taj način kažemo da će vrijednost svojstva `ImePrezimeStudenta` iz forme `formaUnos` dodijeliti istom svojstvu, ali u formi u kojoj se nalazimo (`this`). To znači se vrijednosti svojstava iz dviju formi mogu razmjenjivati. Na isti način treba reći da se i vrijednost svojstva `OcjenaStudenta` dodjeljuje iz forme `formaUnos`. Cijela metoda `button1_click` u prvoj formi izgleda ovako:

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – Upotreba svojstava kroz više formi

```
private void button1_Click(object sender, EventArgs e)  
{  
    Form2 formaUnos = new Form2();  
    formaUnos.ShowDialog();  
}
```

```
this.ImePrezimeStudenta = formaUnos.ImePrezimeStudenta;  
this.OcjenaStudenta = formaUnos.OcjenaStudenta  
}
```

Međutim, još uvijek ne znamo koje vrijednosti za ime i prezime, te ocjenu je korisnik unio. Unos će se obavljati u drugoj formi (`Form2`), pa ćemo u toj formi upisati naredbe kojima ćemo pohraniti vrijednosti koje korisnik unese.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – dodjela vrijednosti svojstvu u formi Form2

Nakon deklaracije svojstava za obje forme, potrebno je u `Form2.cs` upisati naredbe kojima će se vrijednosti koje korisnik unese u `textBox1` i `textBox2` pohraniti u svojstva (a ne u privatne varijable). Dakle, umjesto u varijablu, sadržaj `textBox-a 1` pohranjujemo u svojstvo `ImePrezimeStudenta`, koje je definirano za klasu `Form2`. Naredbe koje treba upisati u `button1_Click()` – koje će se izvršiti kada korisnik klikne na dugme "Povratak" su:

```
private void button1_Click(object sender, EventArgs e)  
{  
    this.ImePrezimeStudenta = textBox1.Text;  
    this.OcjenaStudenta = Convert.ToInt32(textBox2.Text);  
    this.Close();  
}
```

"this" se ovdje u svim trima naredbama koristi da označi instancu klase u kojoj se nalazimo (a nalazimo se u klasi `Form2`). Tako kažemo da se svojstvu `ImePrezimeStudenta` koje pripada ovoj formi u kojoj se nalazimo, dodijeli vrijednost iz `textBox1.Text`. Na isti način se dodjeljuje i vrijednost svojstvu `OcjenaStudenta`.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – dohvaćanje vrijednosti svojstva u formi Form1

- U formi `Form2` smo dodijelili vrijednost svojstvima. Kako sada te vrijednosti koristiti u prvoj formi (`Form1`)? U našoj vježbi želimo u prvoj formi ispisati na zaslon (u `labelu`) ime i prezime studenta, te ocjenu, koje su učitane u `Form2`. Da bismo ispisali te vrijednosti, moramo im pristupiti preko svojstava koje smo deklarirali za obje forme.
- Premjestite se u programski kod prve forme (`Form1.cs`)
- Kada smo predvidjeli da se izvrši ispis podataka? Kada korisnik klikne na izbornik "Ispis", "Ispis izvještaja".
- To znači da naredbe za ispis treba upisati u metodi koja se kreira ako u dizajnu forme `Form1` dvaput kliknemo na izbornik "Ispis izvještaja". Učinite to. Pozicionirali ste se u metodu: `ispisIzvjestajaToolStripMenuItem_Click()`.
- U toj metodi upišite naredbu kojom ćemo u `labeli (label4)` ispisati vrijednosti svojstava: `ImePrezimeStudenta` i `OcjenaStudenta`

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – ispis vrijednosti svojstva u formi Form1

- U ovu metodu upišite sljedeće naredbe:

```
private void IspisIzvjestajaToolStripMenuItem_Click(object sender, EventArgs e)
{
    label3.Text = this.ImePrezimeStudenta;
    label4.Text = Convert.ToString(this.OcjenaStudenta);
}
```
- U oznaku label3 će se ispisati ime i prezime, a u oznaku label4 ocjena studenta prethodno pretvorena u tekst.
- Riječ "this" ovdje označava da se radi o svojstvu koje pripada instanci klase u kojoj se nalazimo, iako se u principu može i izostaviti, ali se preporuča navođenje riječi "this" zbog toga da ako koristimo i privatnu varijablu s istim nazivom npr. ImePrezimeStudenta, da razlikujemo svojstvo od varijable.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Vježba 8 – ispis vrijednosti svojstva u formi Form1

- Pohranite sve datoteke i debug-irajte aplikaciju. Klikom na dugme "Unos podataka" unesite neku vrijednost za ime i prezime studenta, te za ocjenu.
- Kliknite na "Povratak", te u aktivirajte u prvoj formi izbornik "Ispis" "Ispis izvještaja". Provjerite da li su se podaci ispisali na zaslonu.

Zadaci za dodatnu vježbu:

- Napravite i treću formu u istoj aplikaciji, te da se nakon klika na "Ispis izvještaja" podaci ispisuju u toj trećoj formi
- Napravite unos još jednog podatka o studentu (Spol) i ispišite zajedno s imenom i prezimenom, te ocjenom
- U istoj aplikaciji napravite rukovanje izuzecima u metodi u kojoj se unose podaci

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Zaključak

- Svojstva su važan način za razmjenu vrijednosti između različitih klasa
- Ako želimo dohvatiti neku vrijednost koja pripada instanci druge klase, potrebno je deklarirati svojstvo koje će pripadati i jednoj i drugoj klasi
- Svojstvo se deklarira preko get i set naredbi
- U deklaraciji se navodi naziv javnog svojstva i naziv privatne varijable (ako će se vrijednost svojstva mijenjati u nekoj metodi)
- Vrijednost svojstva iz druge klase dohvaćamo naredbom:

```
this.naziv_svojstva = naziv_forme.naziv_svojstva;
```

Nakon ove naredbe možemo rukovati s vrijednošću tog svojstva u klasi u kojoj se nalazimo (iako je vrijednost dodijeljena u drugoj klasi)

Razvoj poslovnih aplikacija, M.Zekić-Sušac

Pitanja za ponavljanje

- Što su svojstva?
- Koji je opći oblik deklaracije svojstva?
- Deklarirajte svojstvo "ImeDjelatnika" tekstualnog tipa
- Deklarirajte svojstvo "PlacaDjelatnika" kao realni broj
- Kojom biste naredbom prikazali novu formu na način da korisnik ne može kliknuti na prethodnu formu dok ne zatvori trenutno aktivnu formu?
- Kojom biste naredbom podatak koji korisnik unese u textBox1.Text pohranili u svojstvo Spol koje je cjelobrojnog tipa?
- Ako se nalazite u formi Form1, kojom biste naredbom dohvatili vrijednost svojstva Spol koje pripada formi Form2?
- Kojom biste naredbom ispisali cjelobrojnu vrijednost svojstva Spol u 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.
- S. Barker, Visual C# 2005 Express, Mihailo J. Šolajić (ur.), Kompjuter biblioteka, 2007.
- Predavanja i zadaci na web stranici kolegija
- Web izvori

Razvoj poslovnih aplikacija, M.Zekić-Sušac