

C# Događaji

Katedra za informatiku
Fakultet tehničkih nauka
Univerzitet u Novom Sadu

Primer

- U toku izvršavanja određuje se vrednost promenljive koja se ispisuje

```
class Test {  
    public string S {get; set;}  
  
    public void Ispis()  
    {  
        Console.WriteLine(S) ;  
    }  
}  
  
Test t = new Test() ;  
t.S = "Petar" ;  
t.Ispis() ;
```

Delegati - Primer

- U toku izvršavanja određuje se koja metoda će obaviti ispisivanje
- Definisane su dve metode za ispis, u trenutku kompajliranja se ne zna koja metoda treba da se poziva

```
private void IspisMalimSlovima(string s)
{
    Console.WriteLine(s.ToLower());
}
```

```
private void IspisVelikimSlovima(string s)
{
    Console.WriteLine(s.ToUpper());
}
```

Delegati - Primer

- Delegat je kao referenca na metodu
- Svojstvo OI “pokazuje” na metodu koja će izvršiti ispis
 - zaglavlje metode mora biti u skladu sa delegatom ObradivacIspisa
- U toku izvršavanja se postavlja vrednost promenljive OI

```
class Test
{
    public delegate void ObradivacIspisa(string tekst);
    public ObradivacIspisa OI { get; set; }
    public void Ispis(String s)
    {
        if (OI != null)
            OI(s);
    }
}

Test t = new Test();
t.OI = IspisVelikimSlovima;
t.Ispis("Tekst");
```

Događaji

- Mehanizam kojim objekat obaveštava korisnika objekta da je izvršena određena operacija nad objektom
- Implementiraju se korišćenjem delegata
 - Korišćenjem ključne reči **event** obezbeđuje se enkapsulacija delegata - korisnik objekta može da reaguje na događaj, ali ne i da sam izaziva događaj
- U trenutku kada se događaj desi, klasa obezbeđuje poziv metode, a korisnik klase određuje koja se konkretno metoda poziva i obrađuje događaj

Događaji - Primer

- Objekat obaveštava korisnika da je došlo do promene teksta
- Pri promeni teksta, pozvaće se metoda koju određuje korisnik objekta

```
class Test
{
    private string s;
    public string S
    {
        get { return s; }
        set
        {
            s = value;
            if (PromenaTeksta != null)
                PromenaTeksta(this);
        }
    }
    public delegate void ObradjivacPromeneTeksta(Test test);
    public event ObradjivacPromeneTeksta PromenaTeksta;
}
```

Događaji - Primer

```
Test t = new Test();  
t.PromenaTeksta += ObradaPromene;  
t.S = "Petar"; //nakon ovog reda pozvace  
                //se metoda ObradaPromene  
private void ObradaPromene(Test t)  
{  
    Console.WriteLine(  
        "Izvrшена promena. Novi tekst: "  
        + t.S);  
}
```