

## 4. Debug-iranje aplikacija u C#-u

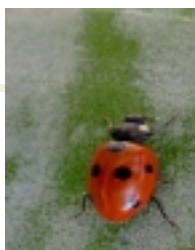


## Što ćete naučiti u ovom poglavlju

- Što je debug-iranje
- Koji se alati za debug koriste
- Kako se radi s točkama prekida
- Kako se koriste alati za kretanje kroz kod po koracima

## Što je debug-iranje?

= Pronalaženje grešaka i nedostataka u programskom kodu



Naziv potječe od stvarnog događaja – kod jednog od prvih računala "Mark I" dogodio se kvar u radu. Nakon dugotrajne potrage za greškom programeri su otkrili da je kvar izazvala jedna buba koja se uvukla u računalo. Otada se svaka greška ili nedostatak u radu računala zove "bug" (Barker, 2007).

## Vrste grešaka u aplikacijama

Postoje tri vrste grešaka koje se mogu pojaviti u aplikacijama:

1. **Sintaksne greške** – formalne greške u jeziku
2. **Semantičke greške** – program ne radi ono za što je namijenjen
3. **Greške izuzetka** – pojavljuju se prilikom korištenja programa, neizbježne su, a uzroci mogu biti različiti

## Sintaksne greške

- osnovna vrsta grešaka, nastaje pogreškom u jeziku, tj. u pogrešno upisanom programskom kodu (npr. nedostatak neke zagrade, nepravilno upisana ključna riječ naredbe ili dr.)
- Ove greške prilikom debug-iranja budu prikazane u prozoru s popisom grešaka (Error list)

Primjer:

- **Private void button1\_click(object sender, EventArgs e)**  
**{**  
    **iznosPoreza = vrijednostProizvoda \* stopaPoreza;**  
- Nedostaje vitičasta zagrada koja označava kraj funkcije button1\_click
- **int ocjenaStudenta** – nedostaje ";" na kraju naredbe
- **Duble iznosPoreza;** – krivo napisan tip varijable

## Semantičke greške

- Kod njih su naredbe formalno ispravno napisane, pa ih je teže pronaći od sintaksnih grešaka (ne vide se u prozoru Error list prilikom debug-iranja)
- To su greške značenja – program ne radi ono što bi trebao raditi, nego nešto drugo
- Mogu biti uzrokovane logičkim pogreškama u programskom kodu ili podacima koje unese korisnik
- One se vide tek prilikom izvršavanja programa (kada otkrijemo da program ne radi što bi trebao raditi, ali još uvijek nije jasno što treba ispraviti u programskom kodu)

## Greške izuzetaka

- Javljaju se, kao i semantičke greške, tek prilikom izvršavanja programa
- Izuzeci (eng. exceptions) će nastati bez obzira što programer poduzme da ih spriječi (neizbježne su)
- Uzrok može biti različit, ponekad je u semantičkim greškama, jer se logika programa ne uspije ostvariti zbog nekih specifičnih podataka koje korisnik unese
- Rukovanje ovim greškama (eng. exception handling) obrađeno je u posebnom poglavlju

Razvoj poslovnih aplikacija, M.Zekić-Sušac

7

## Alati za debugiranje - Build i Debug

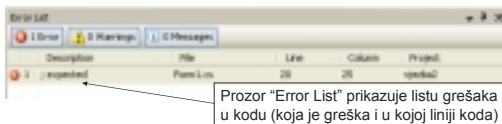
- Build – alat koji prevodi program ali ga ne pokreće (ako ima grešaka prikazuje ih u Error List prozoru) - koristi se za provjeru sintakasnih grešaka
  - pokretanje alata: pomoću gl. izbornika ili F6
- Debug – alat koji prevodi program, prikazuje listu grešaka u Error list prozoru, i pokreće program – koristi se za provjeru i sintakasnih i ostalih dviju vrsta grešaka
  - pokretanje alata: pomoću gl. izbornika ili F5

Razvoj poslovnih aplikacija, M.Zekić-Sušac

8

## Lista grešaka (Error list)

- Lista grešaka (Error list) je alat koji prikazuje sve greške u kodu koje su spriječile da se program izvrši
- Najčešće prikazuje sintaksne greške, ali prikazuje i "rupe" u kodu, tj. ako je nešto zaboravljeno, npr. ako se nekoj deklariranoj varijabli ne dodijeli vrijednost

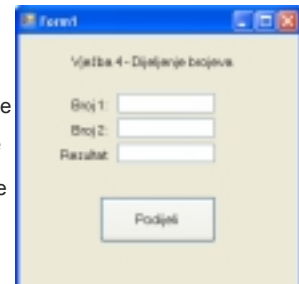


Razvoj poslovnih aplikacija, M.Zekić-Sušac

9

## Vježba 4 – Dijeljenje brojeva

- Otvorite u C# Express-u novi projekt (Windows aplikaciju) pod nazivom: Vježba4
- Na formi kreirajte kontrole *label*, te jednu kontrolu *button*, na gumbu upišite "Klikni" (vidi Sliku)
- Dvaput kliknite na dugme kako biste došli u programski kod i upisali odgovor na događaj *button1\_click*



Razvoj poslovnih aplikacija, M.Zekić-Sušac

10

## Vježba 4 - nastavak

- u kod programa upišite sljedeće:
 

```
private void button1_Click(object sender, EventArgs e)
{
    double broj1;
    double broj2;
    double rezultat;
    // sprečavanje dijeljenja s nulom
    if (broj2 == 0)
    {
        MessageBox.Show("Broj2 mora biti pozitivan. Unesite ponovo pozitivan broj.", "Poruka",
        MessageBoxButtons.OK);
    }
}
```

Upis komentara u program s pomoću znakova "//"

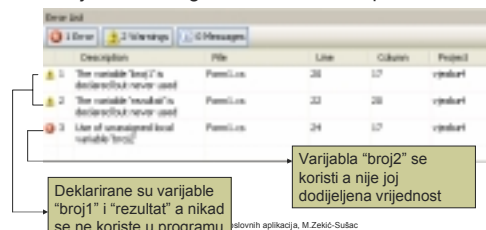
Struktura "if-then"

Razvoj poslovnih aplikacija, M.Zekić-Sušac

11

## Vježba 4 - nastavak

- U prethodnim naredbama ima grešaka, pa se prilikom aktiviranja naredbe Build (ili F6) pojavljuje sljedeća lista grešaka u Error List prozoru:



Razvoj poslovnih aplikacija, M.Zekić-Sušac

12

## Vježba 4 - nastavak

- Da bismo otklonili grešku 1, najjednostavnije je dvaput kliknuti na tu grešku u prozoru Error List i alat nas pozicionira u liniju greške. Iza te linije pritisnut ćemo Enter i upisati ovu naredbu kojom ćemo dodijeliti vrijednost varijabli "broj1":  
`broj1 = Convert.ToDouble(textBox1.Text);`
- Na isti način otklonite grešku 2. Upišite liniju:  
`broj2 = Convert.ToDouble(textBox2.Text);`
- Ponovo pokrenite alat prevođenje programa tipkom F6 ili naredbom Build – pojavit će se samo još jedna greška na popisu grešaka  
(The variable "rezultat" is declared but never used)

Razvoj poslovnih aplikacija, M.Zekić-Sušac

13

## Vježba 4 - nastavak

- Da bismo iskoristili varijablu "rezultat", upisat ćemo dvije naredbe s kojima ćemo podijeliti dva učitana broja, rezultat dijeljenja upisati u var."rezultat" i ispisati u kontroli textBox3
- Naredbe treba upisati iza strukture if-then (nakon što je zatvorena vitičasta zagrada za tu strukturu):  
`rezultat = broj1 / broj2;`  
`textBox3.Text = Convert.ToString(rezultat);`

Razvoj poslovnih aplikacija, M.Zekić-Sušac

14

## Vježba 4 - nastavak

- Cijela metoda button1\_click sada izgleda ovako:  

```
private void button1_Click(object sender, EventArgs e)
{
    double broj1;
    broj1 = Convert.ToDouble(textBox1.Text);
    double broj2;
    broj2 = Convert.ToDouble(textBox2.Text);
    double rezultat;
    // sprečavanje dijeljenja s nulom
    if (broj2 == 0)
    {
        MessageBox.Show("Broj2 mora biti pozitivan. Unesite ponovo pozitivan broj.", "Poruka", MessageBoxButtons.OK);
    }
    rezultat = broj1 / broj2;
    textBox3.Text = Convert.ToString(rezultat);
}
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

15

## Vježba 4 - nastavak

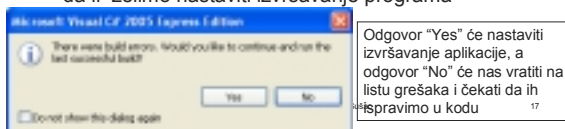
- Pohranite i ponovo pokrenite prevođenje programa (Build).
- Kojih grešaka sada nema u programu?  
**Sintaksnih.**
- Da bismo provjerili da li ima semantičkih grešaka i grešaka izuzetaka, potrebno je i pokrenuti aplikaciju.
- Iz razvojne okoline C# Express aplikaciju možemo prevesti i pokrenuti naredbom **Debug (ili tipkom F5)**. **Pokrenite Debug da biste provjerili ispravnost aplikacije.**

Razvoj poslovnih aplikacija, M.Zekić-Sušac

16

## Debugiranje programa

- Prilikom pokretanja Debug alata, programer prevodi i pokreće program kako bi testirao njegovu logičku ispravnost (tj. da li radi ono za što je namijenjen)
- Ovdje se također mogu pojaviti nove greške, koje treba popraviti prije davanja aplikacije na korištenje korisnicima
- Ako se pojave greške kod debugiranja, pojavljuje se poruka na ekranu da postoje greške, te pitanje da li želimo nastaviti izvršavanje programa



17

## Režim prekida vs. režim izvršavanja

- U nekim je situacijama potrebno prilikom izvršavanja aplikacije (debugiranja) zaustaviti ili pauzirati izvršavanje. Ovaj se proces prilikom debugiranja zove - **režim prekida**.
- Režim prekida omogućava da se postupno (korak po korak) ide po naredbama koda i ispravljaju semantičke i logičke greške, ali i da se ispituju različiti aspekti aplikacije mijenjanjem nekih naredbi
- Kako postići rad u režimu prekida?
- Jedan od načina je naredbom: **Debug / Break All** prilikom debugiranja

Razvoj poslovnih aplikacija, M.Zekić-Sušac

18

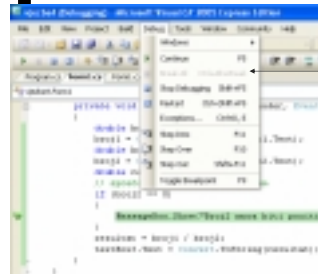
## Rad u režimu prekida

- Da bismo isprobali rad u režimu prekida, pokrenite Debug u aplikaciji vježba4
- Prilikom izvršavanja programa unesite 0 za vrijednost varijable broj2, kako bi se pojavila poruka. U MessageBox-u nemojte kliknuti na dugme OK
- Kliknite mišem u prozor C# Express-a i u izborniku izaberite naredbu **Debug / Break All**. Alat će pauzirati izvršavanje, vratit će nas u kod programa, a linija koda koja je zadnja izvršena će biti označena drugom bojom (u našem slučaju to je linija u kojoj je pozvana metoda MessageBox)

Razvoj poslovnih aplikacija, M.Zekić-Sušac

19

## Rad u režimu prekida - nastavak



U izborniku Debug sada možemo ili nastaviti izvršavanje aplikacije (Continue) ili zaustaviti debugiranje (Stop Debugging)

Označena linija u kojoj je zaustavljeno izvršavanje

Razvoj poslovnih aplikacija, M.Zekić-Sušac

20

## Rad sa tačkama prekida

- Drugi način kako aplikaciju možemo prekinuti prilikom izvršavanja je postavljanje **tačka prekida** na mjestima gdje želimo da se program zaustavi (gdje želimo kontrolirati rad programa, tj. vrijednosti nekih varijabli u tijeku izvođenja programa i otkriti eventualne semantičke ili logičke greške)

Kako postaviti tačke prekida?

- kada se nalazimo u liniji kod u kojoj želimo prekinuti aplikaciju prilikom izvršavanja, aktivirati naredbu **Debug / Toggle Breakpoint** (ili tipku **F9**), ili
- Pomoću miša kliknuti na sivo područje lijevo od linije koda u kojoj želimo prekinuti aplikaciju

Razvoj poslovnih aplikacija, M.Zekić-Sušac

21

## Rad sa tačkama prekida - nastavak

- U oba slučaja pojavit će se crvene točke kraj linije, koje označavaju tačke prekida



Točke prekida – prilikom debugiranja program će kod izvršavanja aplikacije u tim linijama pauzirati izvršavanje, i tada ćemo moći provjeriti vrijednosti koje su poprile varijable

Razvoj poslovnih aplikacija, M.Zekić-Sušac

22

## Brisanje i onespособljavanje tačkama prekida

- Brisanje tačkama prekida – ako nam više nisu potrebne, tačke prekida brišu se na isti način kao što se postavljaju – ponovnim klikom miša na crvenu tačku kraj linije, ili s pomoću izbornika Debug / Toggle Breakpoint
- Onespособljavanje – ako nam trenutno nisu potrebne, ali će nam trebati kasnije, tačke prekida možemo onespособiti desnim klikom na crvenu tačku i naredbom Disable Breakpoint

Razvoj poslovnih aplikacija, M.Zekić-Sušac

23

## Vježba 4 - nastavak

- U aplikaciji Vježba4 postavite nekoliko tačkama prekida (kao na str.22 ovog predavanja)
- Aktivirajte Debug, zatim unesite vrijednost 2 za broj1, te 3 za broj2, kliknite na dugme "Podijeli"
- Kad program prekine izvršavanje aplikacije u prvoj tački, nastavite dalje naredbom Debug /Continue (ili F5)
- Kad se program zaustavi u drugoj tački prekida, prijedite u linijama koda mišem preko naziva varijable "broj1". Vidjet ćete da će alat IntelliSense prikazati da je trenutna vrijednost varijable jednaka 2. Provjerite trenutnu vrijednost i varijable broj2, kao i svojstva TextBox1.Text i TextBox2.Text

Razvoj poslovnih aplikacija, M.Zekić-Sušac

24

## Prikaz i promjena vrijednosti varijabli u režimu prekida

- Osim što se u režimu prekida može vidjeti vrijednost varijabli, ona se može ovdje i promijeniti po potrebi klikom na dio gdje je prikazana vrijednost
- U aplikaciji Vježba4 pokušajte promijeniti vrijednost varijabli Broj2 u režimu prekida, npr. postavite je na 0, i pratite daljnje debugiranje aplikacije.

Razvoj poslovnih aplikacija, M.Zekić-Sušac

25

## Prozor Locals

- Vrijednosti lokalnih varijabli (koje program koristi lokalno, samo za trenutnu formu), mogu se vidjeti i u prozoru Locals
- Prozor Locals se aktivira za vrijeme debugiranja aplikacije, naredbom Debug / Windows / Locals

Name	Value
this	(Vježba4.Form1, Text: Form1)
sender	(Text = "Podijeli")
e	(X = 59 Y = 12 Button = Left)
broj1	2.0
broj2	3.0
rezultat	0.0

Razvoj poslovnih aplikacija, M.Zekić-Sušac

26

## Prozor Immediate

- Ovaj prozor iskusniji programeri koriste za prikaz i promjenu vrijednosti varijabli i naredbi u programu, kako bi lakše pronašli neke semantičke greške
- Prozor se aktivira naredbom Debug / Windows / Immediate
- Da bi se dobila vrijednost neke varijable, u prozor je potrebno upisati "?" naziv varijable
- Da bi se promijenila vrijednost nekoj varijabli u tom prozoru, potrebno je upisati naziv varijable, "=" i novu vrijednost

```
Immediate Window
? broj1
2.0
broj2 = 4
4.0
```

Razvoj poslovnih aplikacija, M.Zekić-Sušac

27

## Prolaz kroz kod po koracima

### Naredbe Debug / Step Into, Step Over i Step Out of

- Omogućuju prolaženje kroz programski kod po koracima ako želimo pratiti što se događa u svakoj liniji koda i zaustavljati izvršavanje
- Prozor Debug / Windows / Watch omogućuje praćenje varijabli kroz cijelu aplikaciju

Razvoj poslovnih aplikacija, M.Zekić-Sušac

28

## Izvršavanje aplikacije od strane korisnika

- Korisnik prilikom izvršavanja aplikacije pokreće samo izvršnu (exe) inačicu aplikacije
- Izvršnu inačicu aplikacije C# Express kreira nakon debugiranja, i sprema u posebnu podmapu /bin/debug u mapi projekta
- Za Vježbu 4 – pronađite izvršnu (exe) inačicu aplikacije Vježba4.exe i pokrenite je, te testirajte ispravnost aplikacije.
- Prilikom pokretanja izvršne inačice **nije omogućeno** ispravljanje grešaka i ulaženje u programski kod aplikacije

Razvoj poslovnih aplikacija, M.Zekić-Sušac

29

## Zaključak

- Postoji više načina prevođenja i izvršavanja aplikacije
- Naredba (alat) Build omogućuje samo prevođenje programa uz prikaz popisa sintaksnih grešaka
- Naredba (alat) Debug omogućuje i prevođenje i izvršavanje aplikacije, te praćenje rada programa po linijama koda (režim prekida) kako bi se pronašle semantičke i logičke greške

Razvoj poslovnih aplikacija, M.Zekić-Sušac

30

## Pitanja za ponavljanje

1. Što je debugiranje?
2. Koje su tri glavne vrste grešaka kod programiranja (izrade) aplikacije?
3. Koje se dvije grešaka mogu ispraviti (debugirati)?
4. Koje su greške neizbježne i ne mogu se ispraviti?
5. Objasnite što su semantičke greške.
6. O kakvoj vrsti greške se radi u primjeru:  
`Int ocjenaStudenta=2;  
MessageBox.Show(Convert.ToString(ocjenaStudenta))`
7. O kakvoj vrsti greške se radi u primjeru:  
`Int ocjenaStudenta;  
MessageBox.Show(Convert.ToString(ocjenaStudenta));`
8. Što omogućuje alat Build?
9. Što omogućuje alat Debug?

## Pitanja za ponavljanje - nastavak

10. Što je režim prekida i kada se može koristiti?
11. Kako se postavljaju točke prekida?
12. Navedite dva prozora koja prikazuju vrijednosti varijabli u režimu prekida?
13. Kojom naredbom možemo nastaviti debugiranje programa ako se nalazimo u režimu prekida?
14. Kako se zove prozor u kojem se prikazuje popis grešaka nakon prevođenja programa?

## 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