# Domaći zadatak 06 – Branislav Manojlović kp11
# KI301-DZ06

## Tekst zadatka

Odabrati jednu debuging tehniku iz predavanja i primeniti je na proizvoljnom programu.

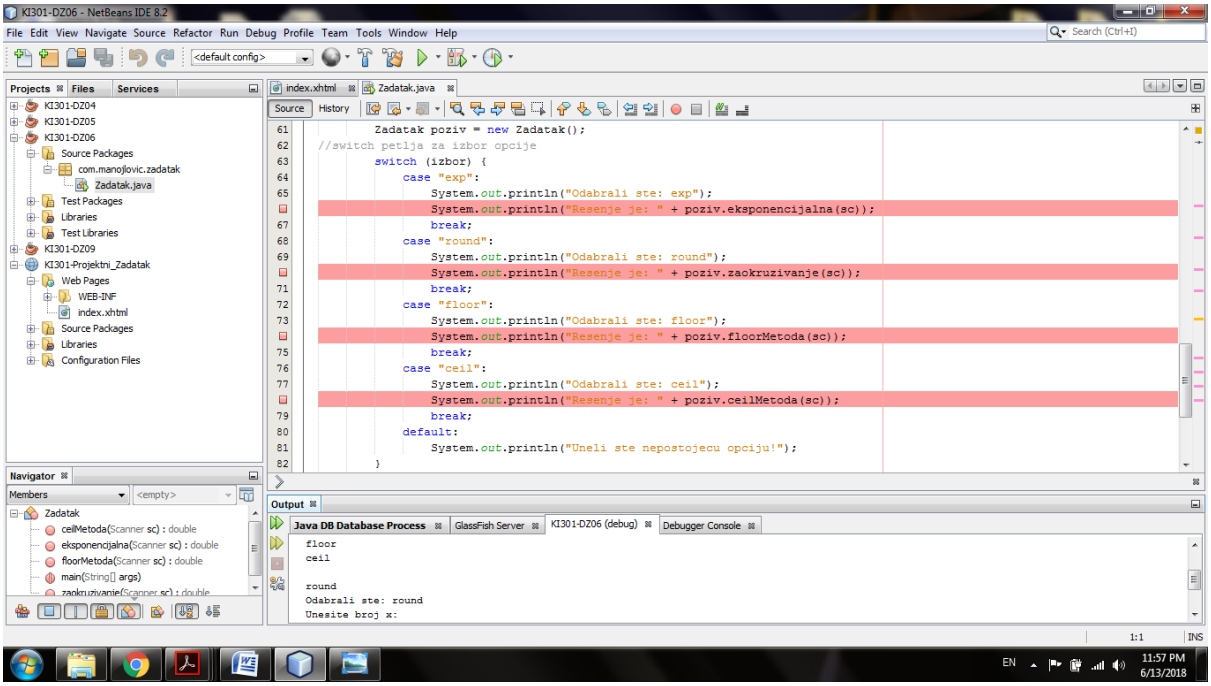
1. Koristiti debbuger u razvojnom okruženju (NetBeans ili Visual Studio zavisno od programskog jezika u kome je pisana aplikacija),
2. Objasniti nedostatke odabranog debbuger-a.

## Rešenje:

1) Ukoliko Java program ne daje izlazne rezultate kao što je planirano i očekivano, moguće je upotrebiti debager kako bi se locirala greška u programskom kodu aplikacije. Kao primer debagovanja iskoristio sam konzolnu Java aplikaciju koja pomoću četiri metode vrši obradu unetih brojeva na način da odgeđuje eksponencijalnu vrednost unetog broja, vrši zaokruživanje broja, vrši zaokruživanje broja na prvu celu gornju vrednost i zaokruživanje broja na prvu celu donju vrednost.

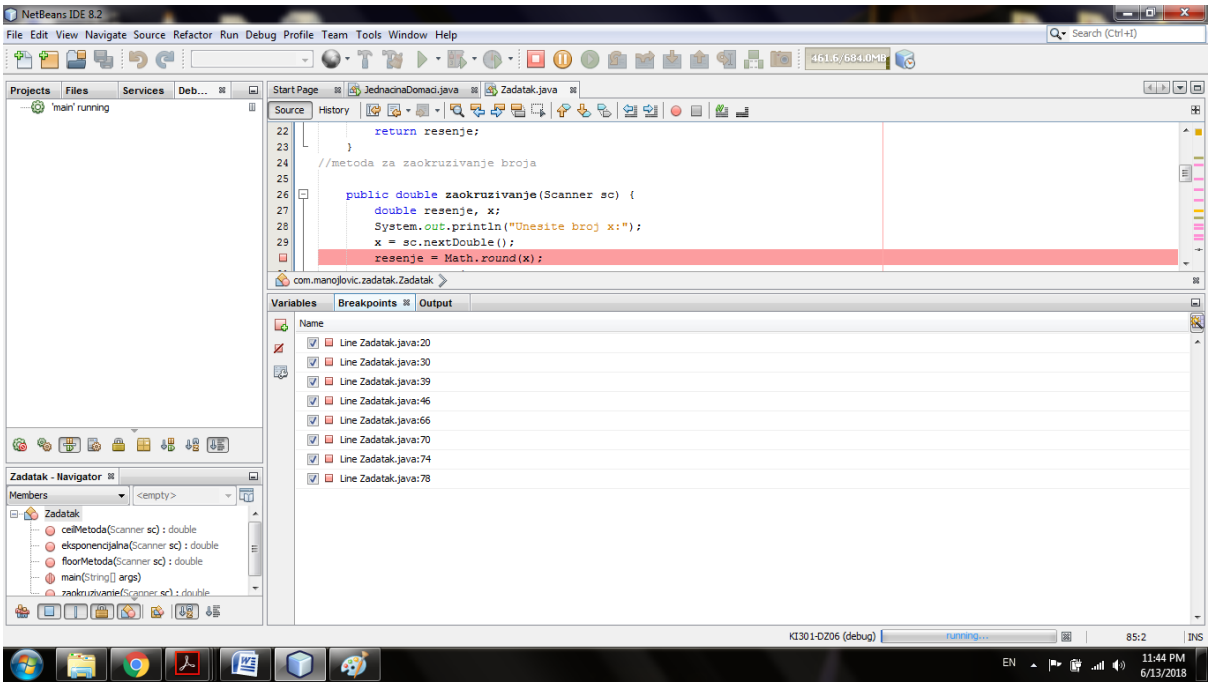
   **Koraci** u debagovanju Java programa, korišćenjem NetBeans debuger-a su sledeći

   1. Prvo postavimo Breakpoints (tačke prekida) na linijama koda za koje sumnjamo da imaju grešku. To radimo levim klikom miša na ivicu radne površine ili desni klik → Breakpoint → Eneabled.



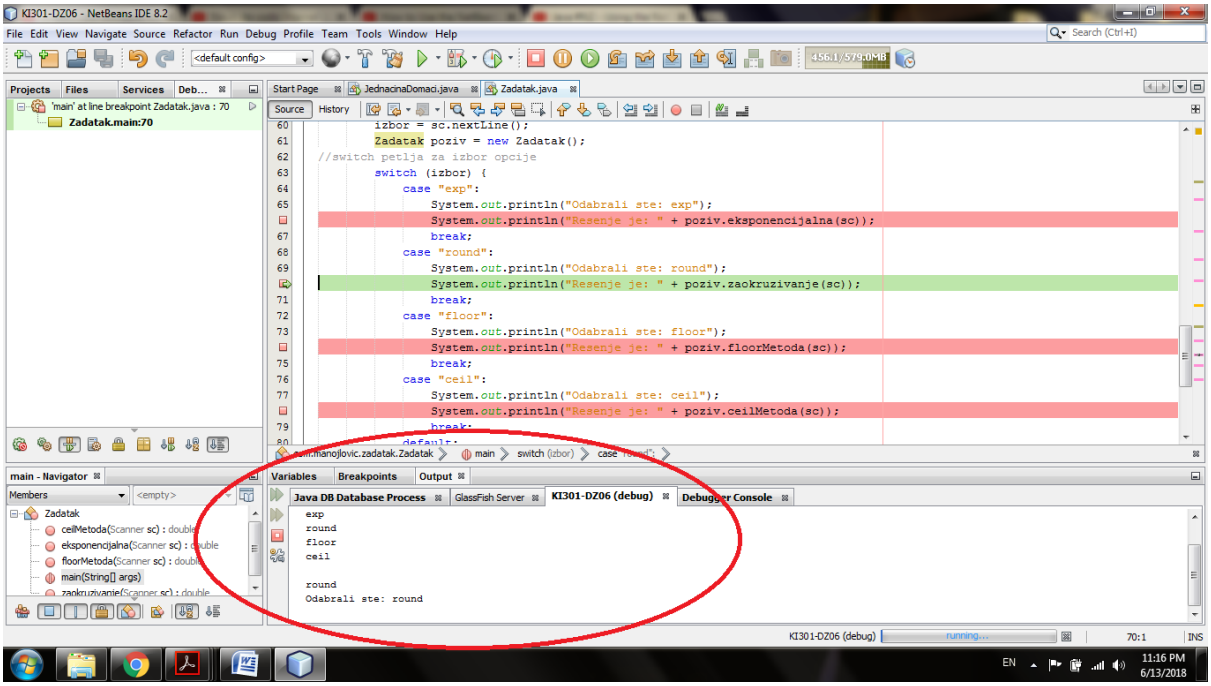

Slika 1. Postavljene Breakpoints

2. Listu svih Breakpoint-a mozemo videti i na kartici „Breakpoints“ u NetBeans-u




Slika 2. Lista postavljenih Breakpoint-a

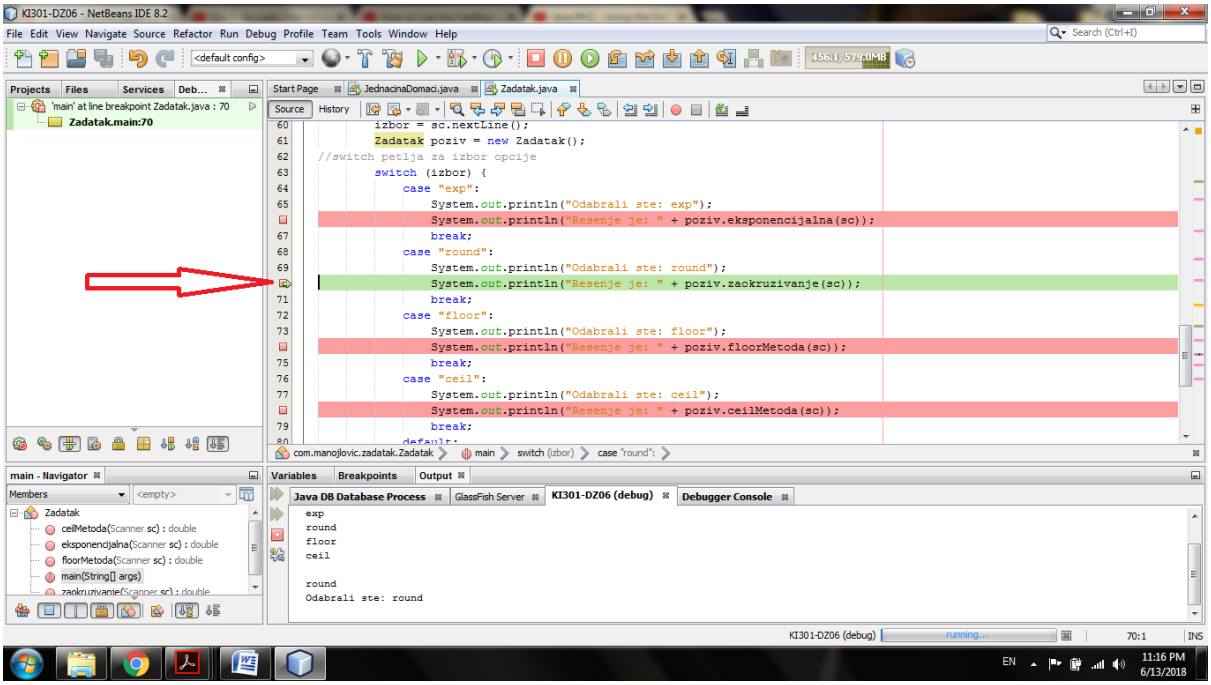
3. Kretanje kroz program i njegovo izvršavanje pratimo u izlaznoj konzoli NetBeans-a




Slika 3. Prikaz izlazne konzole i ispis komunikacije sa korisnikom

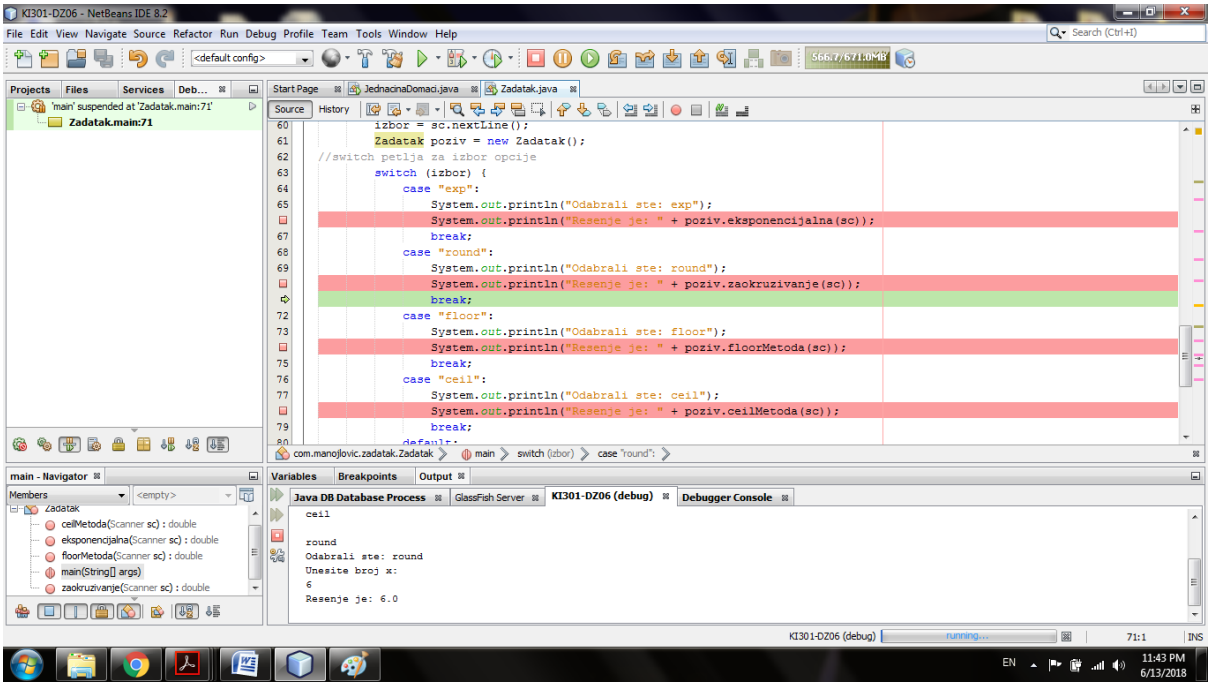
4. Sledeće pokrenemo debager klikom na dugme „Debug project“ ili desnim klikom na programski java fajl koji zelimo da debagujemo → i klik na Debug File opciju.

5. Kada se debager pokrene, praktično se i program pokrene u debug modu i radi sve dok ne stane na prvom Breakpoint-u koji smo označili u programu. Kada se to desi program privremeno stane sa izvrsavanjem i programer moze da pregleda programski kod tražeći grešku u datoj liniji.



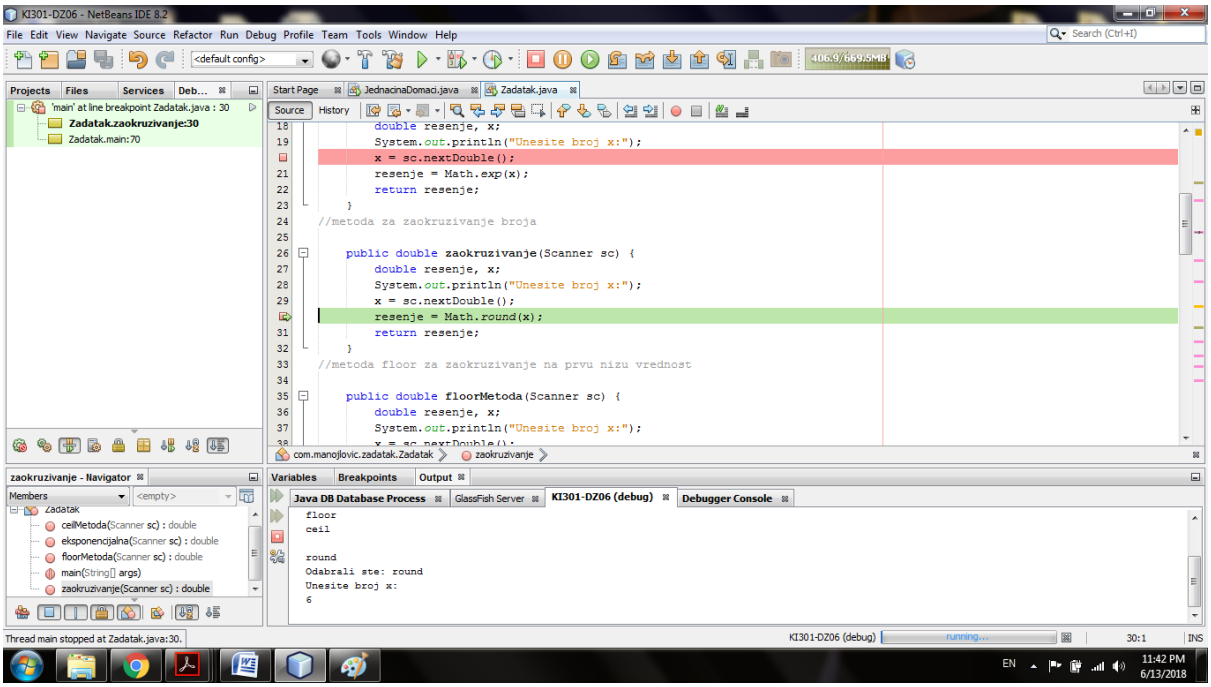
Slika 4. Prikaz programa koji je stao na prvom Breakpoint-u

6. Sledeće, kliknem na dugme „Step Over“ ukoliko želim da nastavim da prolazim kroz kod liniju po liniju, kako bih debagovao program tražeći problem. Prelaz je označen zelenom linijom na slici 5. Ukoliko želim da sa jednog Breakpoint-a odmah prođem kroz celu metodu, mogu da kliknem na dugme „Step Out“.



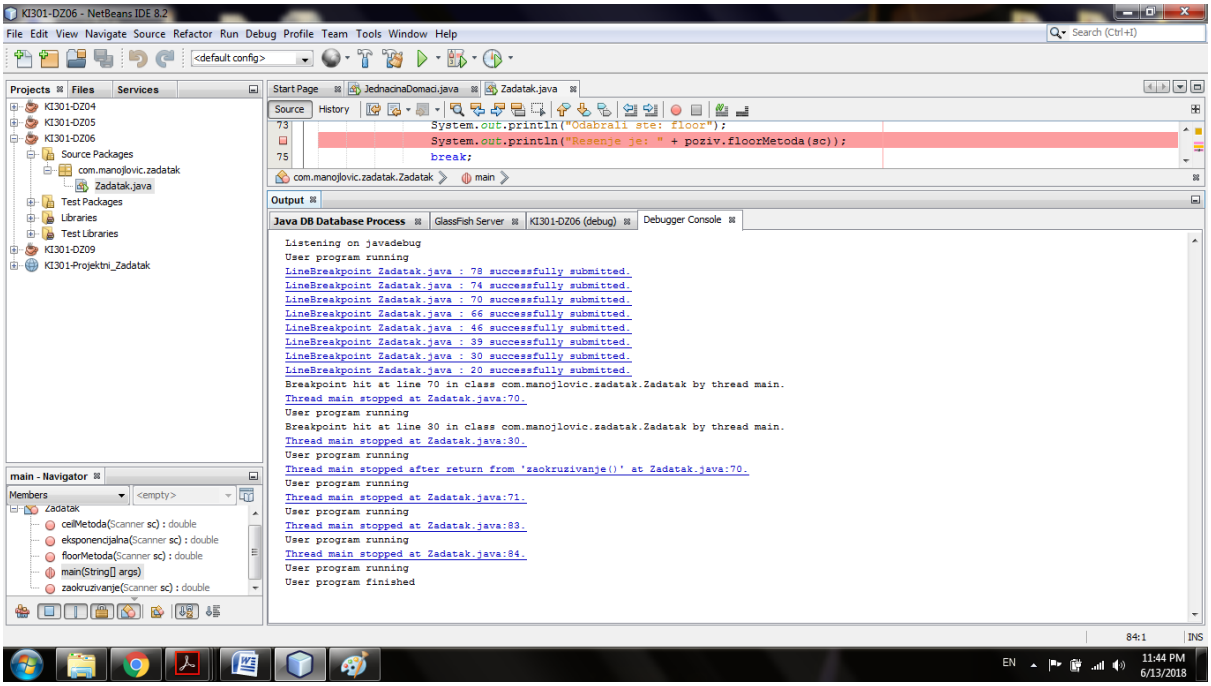
Slika 5. Prelazak na sledeću liniju koda klikom na dugme Step Over

7. Ukoliko zelimo da sa jednog Breakpoint-a predjem odmah na sledeci, kliknem dugme „Continue“ i tada se program u debug modu izvrsava sve do sledeceg Breakpoint-a.




Slika 6. Prelazak na sledeci Breakpoint klikom na dugme „Continue“

8. Svaki korak koji učinimo prilikom debagovanja programskog fajla možemo pratiti na konzoli za debagovanje (Debugger Console). Prolazimo kroz kod Breakpoint po Breakpoint i polako ispravljamo bug po bug dok ne uklonimo sve greške u kodu na koje naiđemo.




Slika 7. Ispis na Debugger Console

Korišćenje debagera nam omogućava da prolazimo kroz programski kod liniju po liniju i da polako pregledamo šta se dešava i da tako uočimo gde se javlja greška ili problem.

2) Nedostaci upotrebe ovog debagera su u sledećem:
   - Ovaj debuger omogućava samo da se locira mesto u kodu na kome se bug nalazi, ali ga ne rešava,
   - Debagovanje programa koji hendluje, tj. radi sa više niti je veoma komplikovano kada se radi korak po korak,
   - Debagovanje programa u real-time modu je praktično nemoguće sa debagovanjem korak-po-korak. Ovde su primer programi za prepoznavanje oblika kakvi se koriste u video kamerama i kod njih se ovo obično rešava korišćenjem već prethodno snimljenog video materijala.
   - Ovaj debager nije pogodan za velike programe i projekte sa puno klasa, puno linija koda i puno aktivnih učesnika. Zato se upotreba debugera preporučuje samo za manje projekte.

**MSc Branislav Manojlović**
**Novi Sad 13.06.2018.**