

*Prolećni semestar, 2019/20*

*PREDMET: SE211 KONSTRUISANJE SOFTVERA*

Domaći zadatak: **6**

Ime i prezime: **Aleksa Cekić**

Broj indeksa: **4173**

Datum izrade: **01.04.2021**

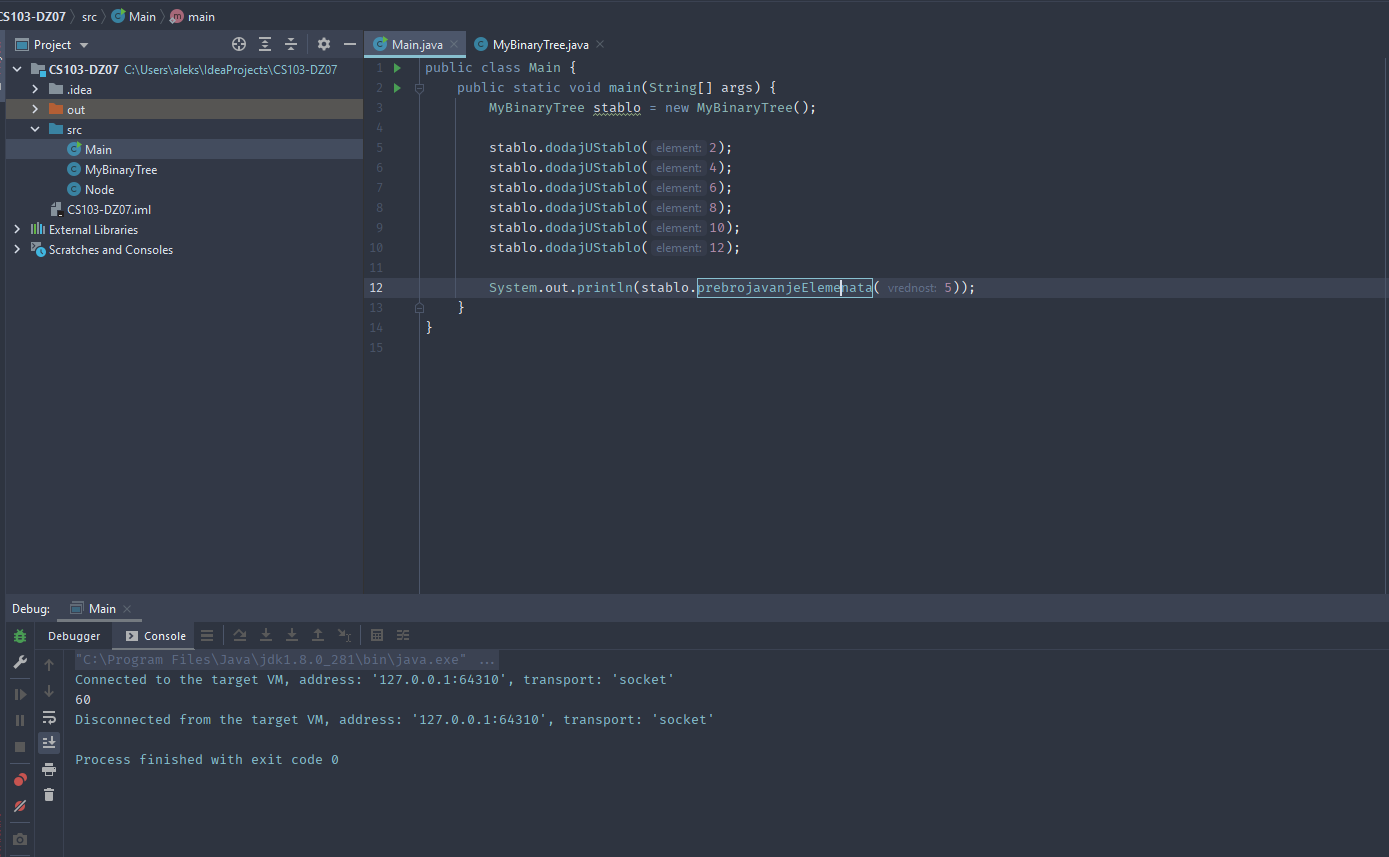
# Tekst domaćeg zadatka:

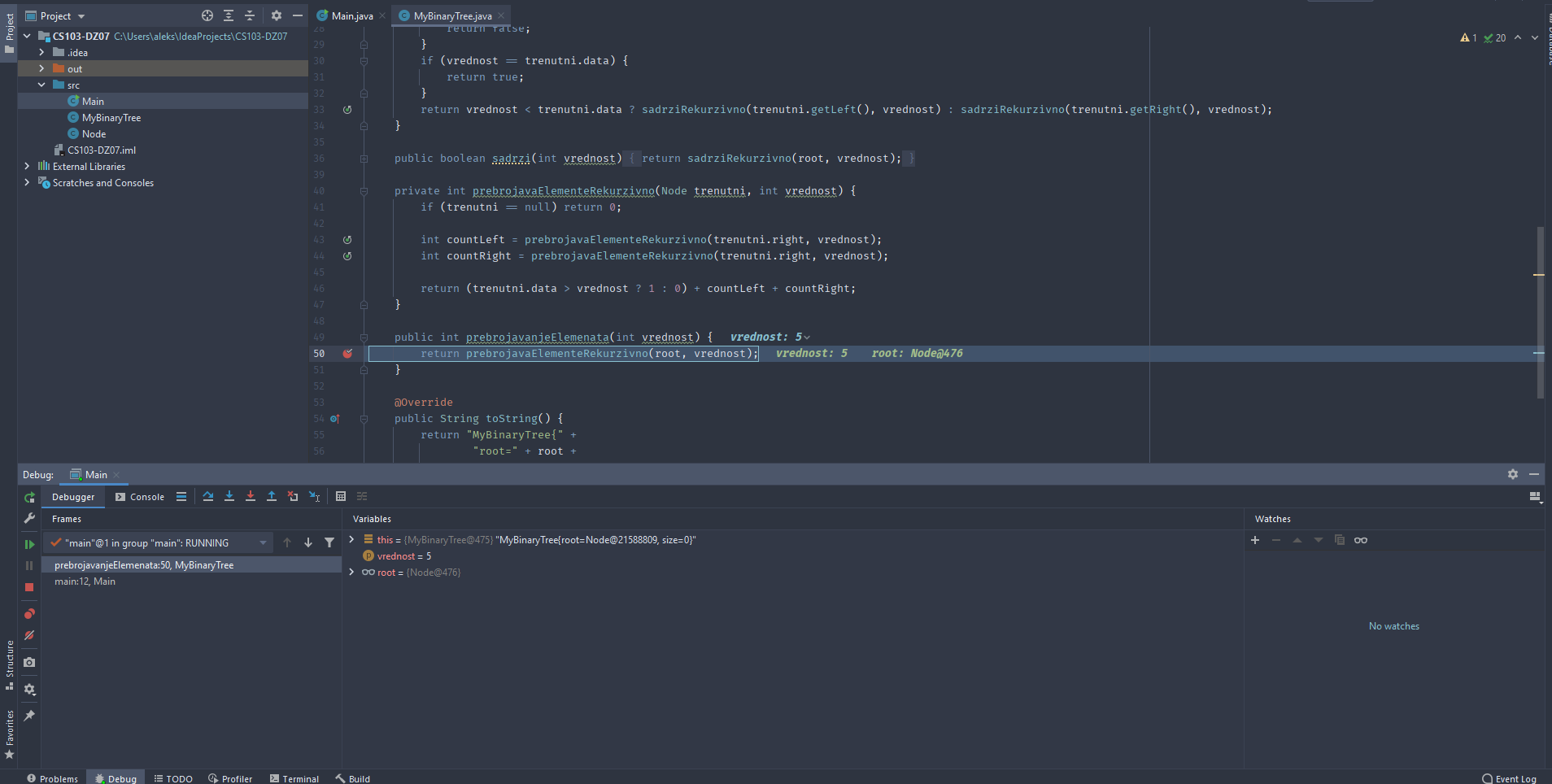
Odabrati jednu debugging tehniku iz predavanja i primeniti na proizvoljnom programu.

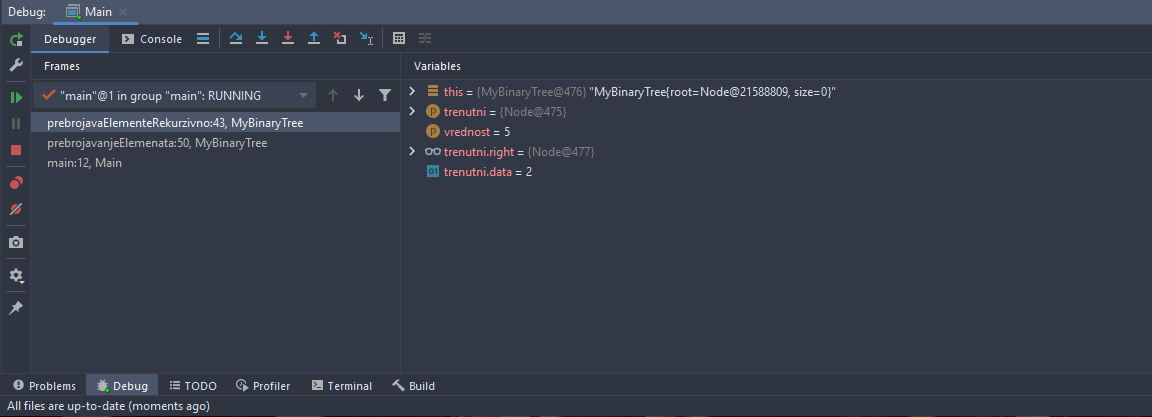
1. Koristiti debbuger u razvojnom okruženju (NetBeans ili Visual Studio zavisno od programskog jezika u kome je pisana aplikacija).
2. Objasniti nedostatke odabranog debbuger-a.
3. Detektovati i dokumentovati sintaksne greške u programskom kodu.
4. Prikazati runtime greške ukoliko postoje u odabranom programu.

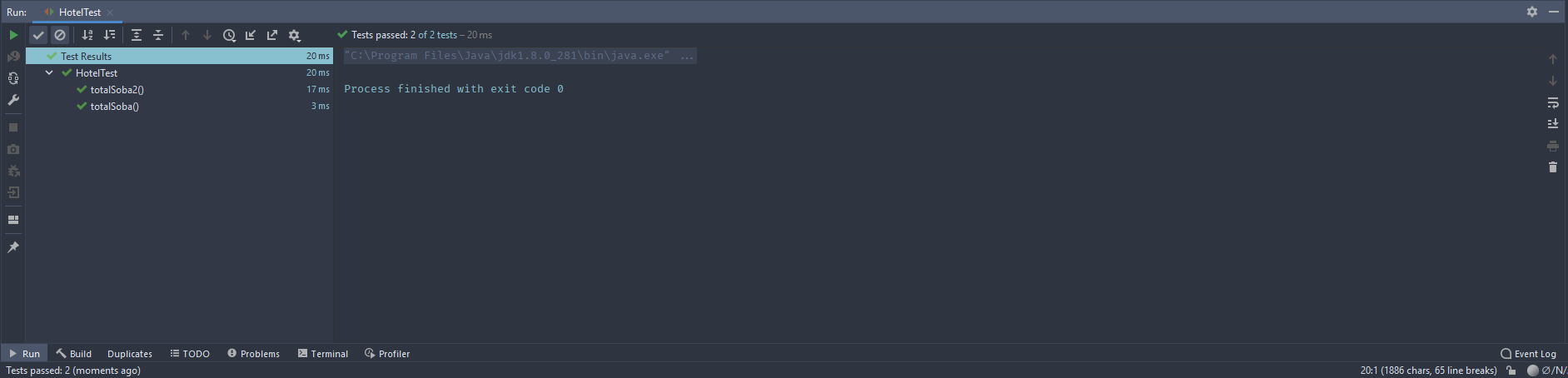
# Rešenje zadatka:

1. Za ovaj domaći zadatak sam iskoristio aplikaciju koja je bila napisana za domaći zadatak iz CS130. Poenta zadatka je da kreira binarno stablo i kako ga kreira tako da ubacuje elemente sa korenove leve ili desne strane u zavisnosti od toga da li je broj koji se ubacuje veći ili manji od korena. Metoda prebrojavanjeElemenata mora da preboji koliko elemenata binarnog stabla ima veću vrednost od zadate vrednosti. U našem slučaju vidimo da postoji negde logički bug u programu zato što na izlazu kaže da je broj 60.









1. Koristio sam IntelliJ-ev debugger koji je veoma dobar, dozvoljava bilo gde da se postave breakpoints kroz aplikaciju i korisnik može da isprati kako kod aplikacija zapravo funkcioniše. Intellij ima skup alata koje supportaju debuggovanje u aplikaciji što ga čini jako dobrim. Pored postavljanja breakpointa, IntelliJ nudi korisniku da inspektuje promenljive, evaulira ekspresije i jos dosta drugih debugging alata. Jedini nedostatak koji bih ja opisao je to što je namenjen samo za Java programski jezik.
2. Ukoliko bi postojale sintaksne greške program se ne bi kompajlirao
3. Što se tiče runtime grešaka ovaj program nije imao ovakvih grešaka.

# Korišćena literatura: (opciono)