Univerzitet u Beogradu

Elektrotehnički fakultet



Najbolje racionalne aproksimacije realnih brojeva

Prvi projektni zadatak

|  |  |
| --- | --- |
| Mentor: | Kandidat: |
| Branko Malešević, prof. dr | Filip Kojić 2023/3297 |

Beograd, Januar 2024.

Sadržaj

[Sadržaj 2](#_Toc150088328)

[1. Postavka projektnog zadatka 3](#_Toc150088329)

[2. Pregled rešenja projektnog zadatka 4](#_Toc150088330)

[3. Testiranje rešenja projektnog zadatka 17](#_Toc150088331)

[Spisak slika 20](#_Toc150088332)

[Literatura 21](#_Toc150088333)

1. Postavka projektnog zadatka

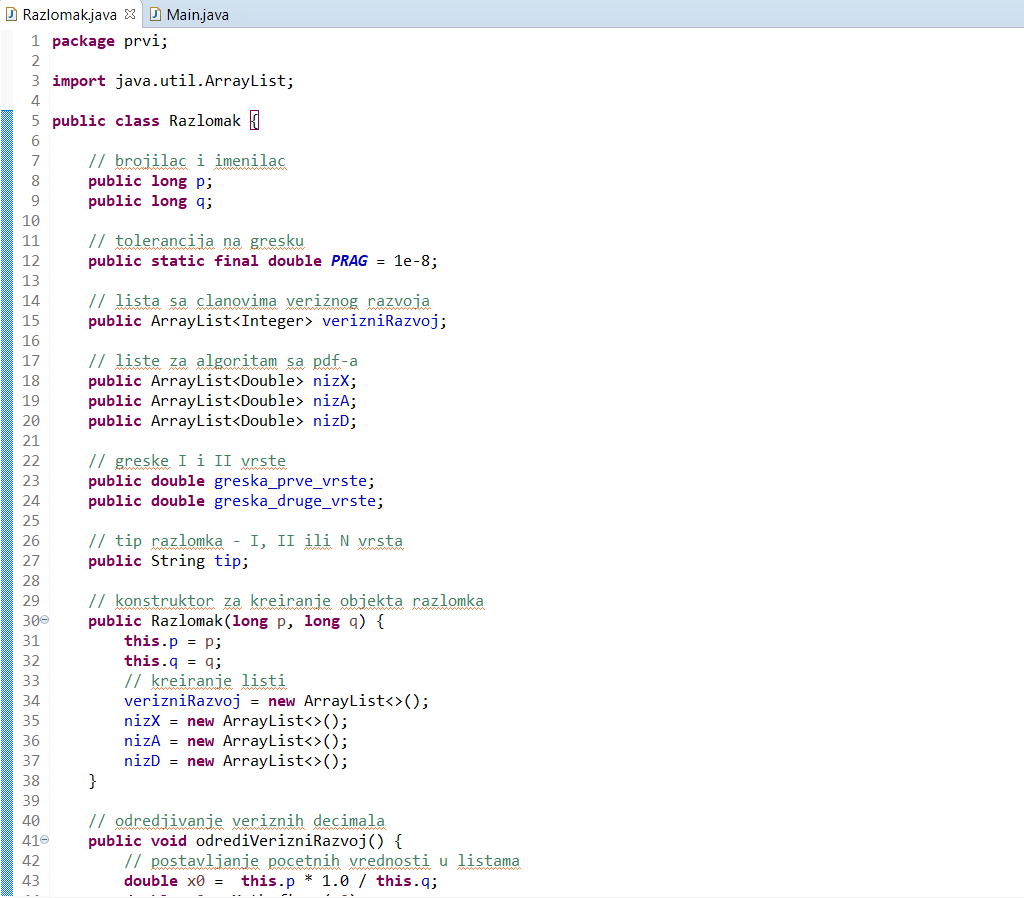
**Projekni zadatak.** Neka je dat pozitivan realan broj α sa konačnim decimalskim zapisom i neka su dati prirodni brojevi n i m, tako da n < m. Formirati niz razlomaka p/q takvih da za imenilac q važi n ≤ q ≤ m (tj. q = n, n + 1, . . . , m) i pri tom imeniocu q pridružujemo brojilac p koji određujemo zaokruživanjem na najbliži priodan broj proizvoda α · q. Predstaviti svaki razlomak p/q u obliku veri- žnog razlomka. U nizu razlomaka p/q izdvojiti:

* najbolje racionalne aproksimacije I vrste
* najbolje racionalne aproksimacije II vrste,
* sortirati sve razlomke p/q po uslovu minimalnosti apsolutne greške |x - p/q| (tj. ε-ranga) .

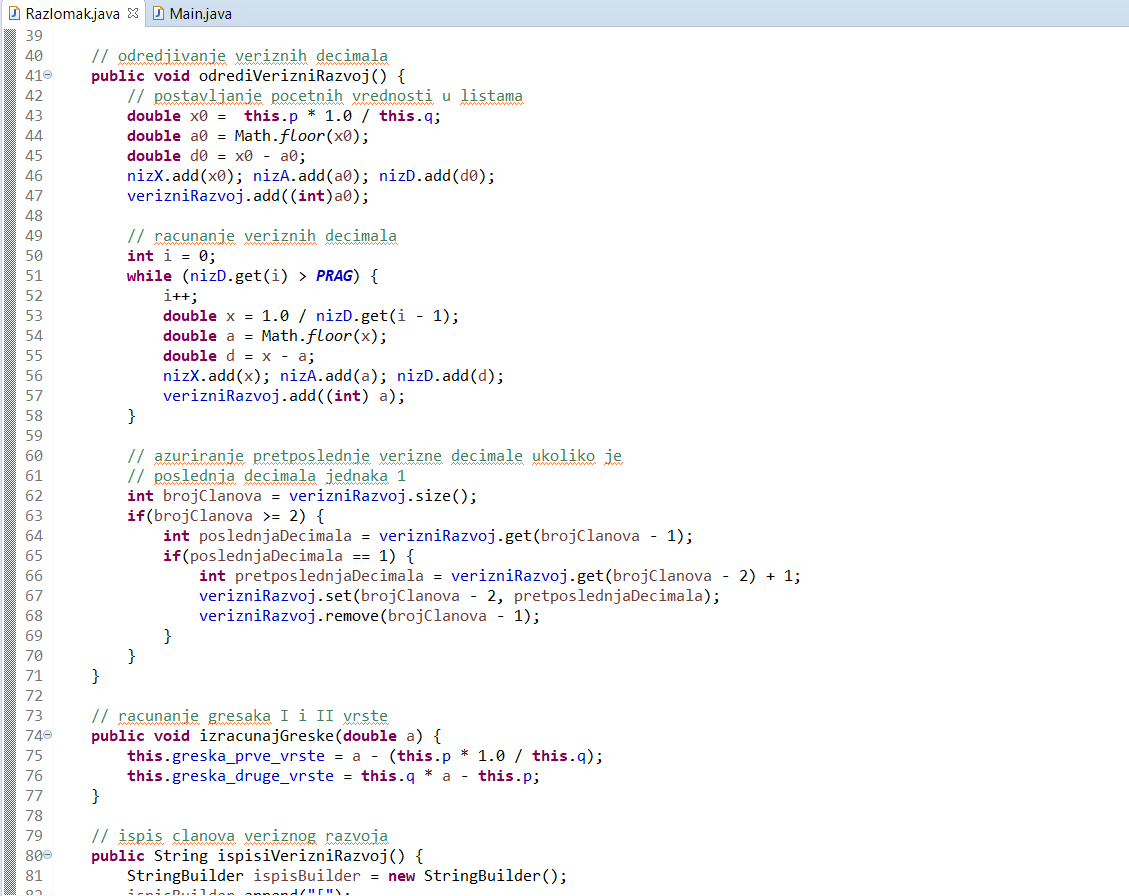
Teorijska osnova za izradu ovog projektnog zadatka data je u materijalima sa predavanja profersora Maleševića. [1]

1. Pregled rešenja projektnog zadatka

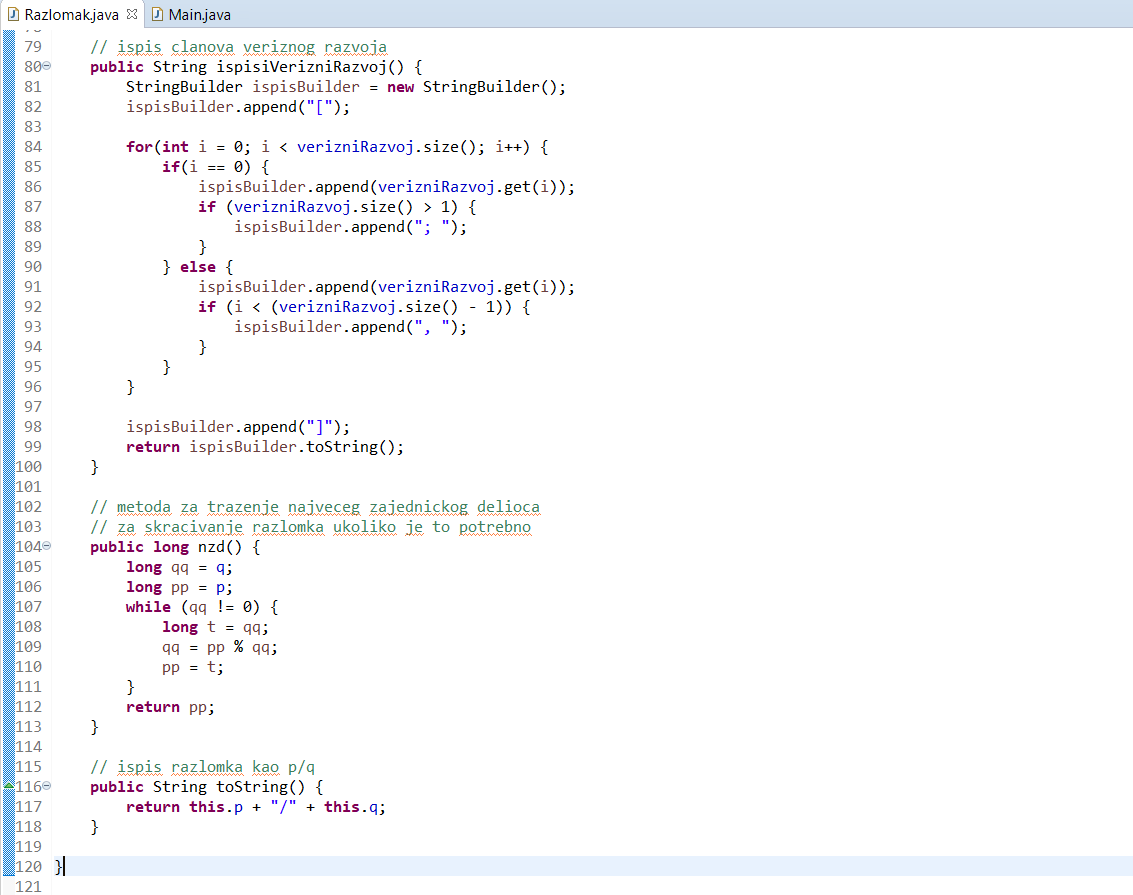
Programski jezik u kome je rađena implementacija ovog projektnog zadatka je Java (JavaSE-13). Implementacija se sastoji iz 2 klase, Razlomak.java i Main.java, čije ćemo detalje ukratko izložiti u nastavku. Razlomak.java je klasa koja služi za predstavljanje objekata koji predstavljaju razlomak formiran iz zadatog realnog broja a. Njena implementacija data je na sledecim slikama:



Slika 2.1. Razlomak.java – deo 1



Slika 2.2. Razlomak.java – deo 2



Slika 2.3. Razlomak.java – deo 3

Klasa Razlomak je Java klasa koja predstavlja racionalni broj sa brojiocem (p) i imeniocem (q). Klasa sadrži različite metode i atribute koji omogućavaju manipulaciju i analizu razlomaka. Evo kratkog opisa svakog dela klase:

1. **Polja:**

* **public long p:** Brojilac razlomka.
* **public long q:** Imenilac razlomka.
* **public static final double PRAG = 1e-8:** Konstanta koja određuje toleranciju na grešku pri izračunavanju verižnih decimala.
* **public ArrayList<Integer> verizniRazvoj:** Lista koja sadrži članove verižnog razvoja razlomka.
* **public ArrayList<Double> nizX, nizA, nizD:** Liste koje se koriste u algoritmu za određivanje verižnih decimala.
* **public double greska\_prve\_vrste, greska\_druge\_vrste:** Vrednosti grešaka I i II vrste pri aproksimaciji realnog broja a sa razlomkom.
* **public String tip:** Tip razlomka (I, II ili N vrsta) u kontekstu aproksimacije realnog broja.

1. **Konstruktor:**

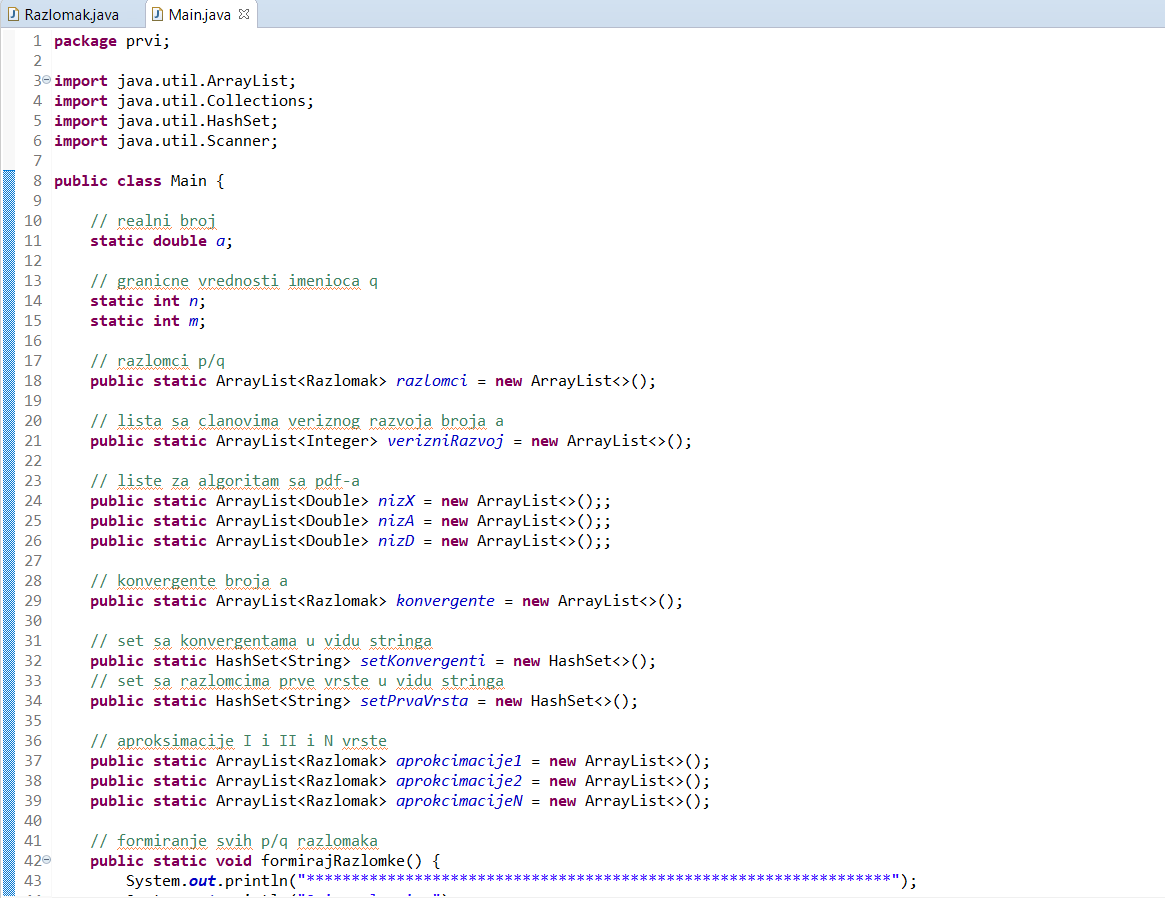
**public Razlomak(long p, long q):** Konstruktor koji inicijalizuje razlomak sa zadatim brojiocem i imeniocem i kreira liste za računanje verižnog razvoja.

1. **Metode:**

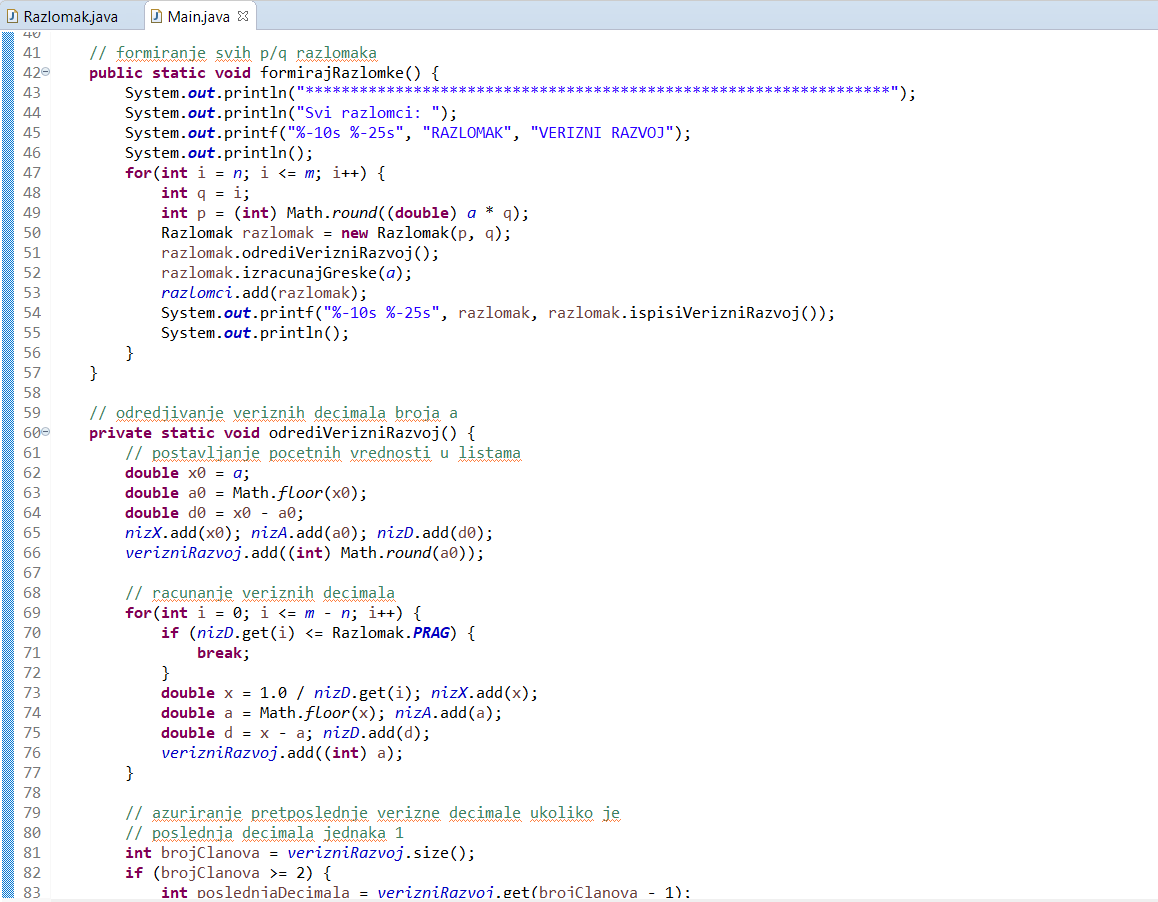
* **public void odrediVerizniRazvoj():** Izračunava verižni razvoj razlomka koristeći algoritam opisan u dokumentaciji.
* **public void izracunajGreske(double a):** Izračunava greške I i II vrste za dati realni broj a prema definicijama datim u dokumentaciji.
* **public String ispisiVerizniRazvoj():** Vraća string reprezentaciju verižnog razvoja razlomka.
* **public long nzd():** Pronalazi najveći zajednički delilac (NZD) brojioca i imenioca kako bi se razlomak skratio, ako je to moguće. Ovo je posebno korisno u situacijama kada je potrebno prepoznati i grupisati identične razlomke unutar liste, čak i kada su oni inicijalno predstavljeni u različitim, ali ekvivalentnim oblicima. Na primer, razlomci 2/4 i 1/2 će nakon skraćivanja biti prepoznati kao isti razlomak (1/2), što omogućava njihovo klasifikovanje u istu kategoriju u analizi koja se izvršava.
* **public String toString():** Vraća string reprezentaciju razlomka u obliku p/q.

Klasa Razlomak.java omogućava detaljnu analizu razlomaka, uključujući njihov verižni razvoj, greške pri aproksimaciji, i mogućnost skraćivanja razlomka. Ove funkcionalnosti su korisne u matematičkim analizama i algoritmima koji zahtevaju rad sa racionalnim brojevima.

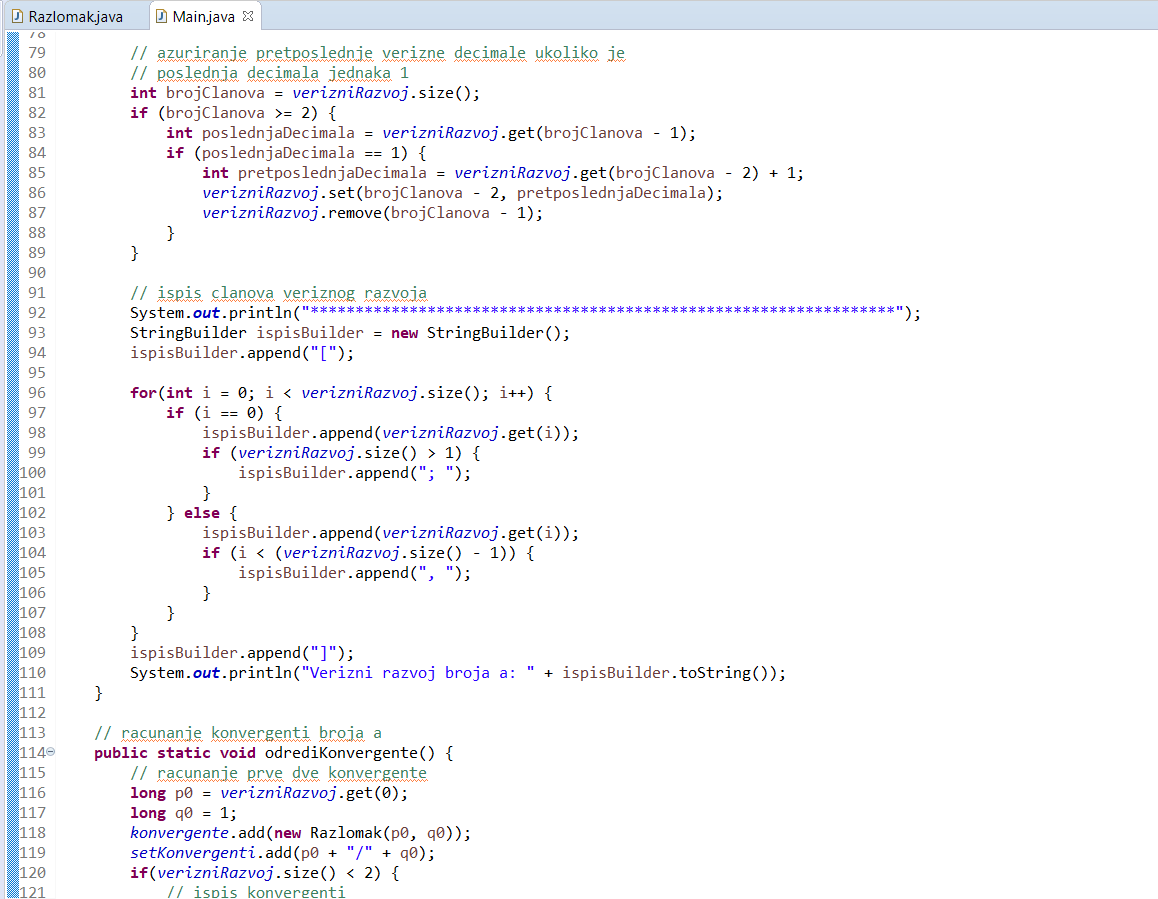
Klasa Main.java je glavni deo Java aplikacije koja se bavi analizom racionalnih aproksimacija realnog broja a. Ova klasa sadrži metode za izračunavanje i klasifikaciju razlomaka u odnosu na njihovu sposobnost da aproksimiraju dati realan broj. Njena implementacija data je na sledećim slikama:



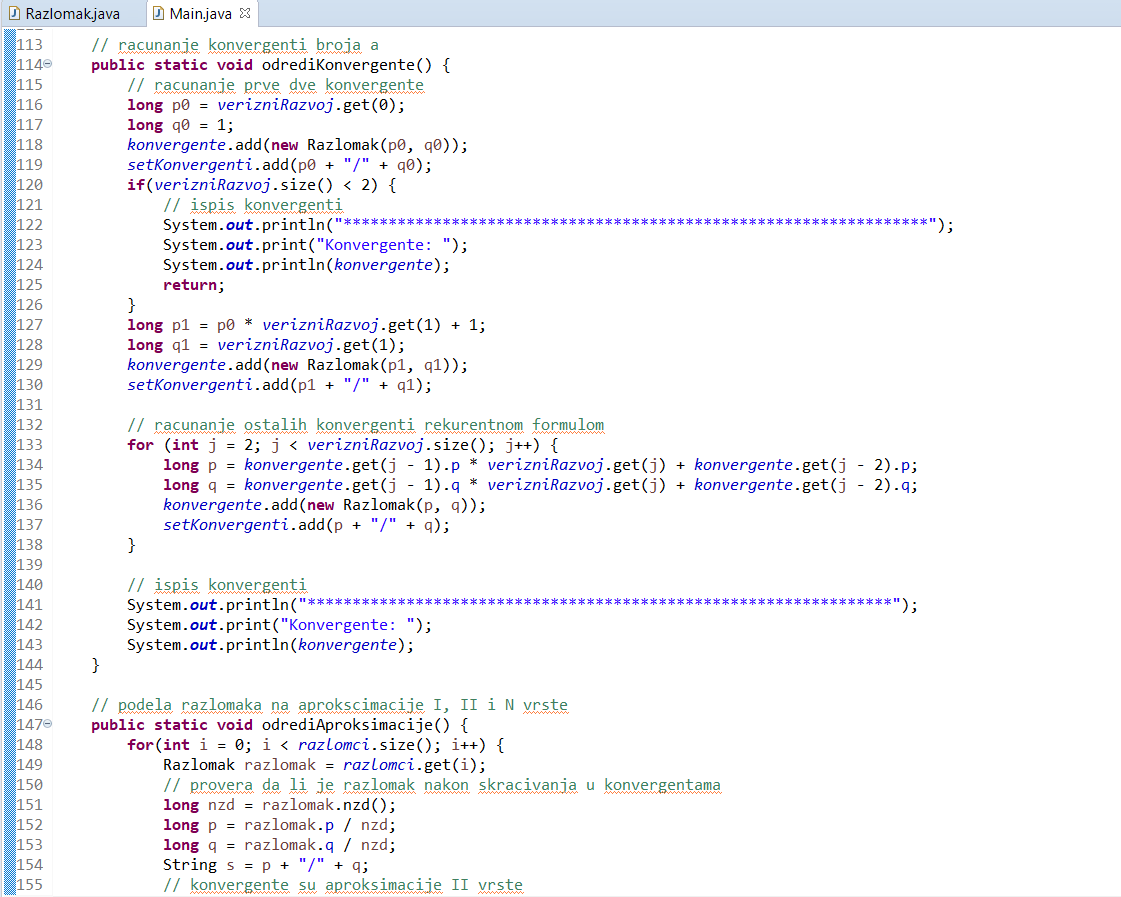
Slika 2.4. Main.java – deo 1



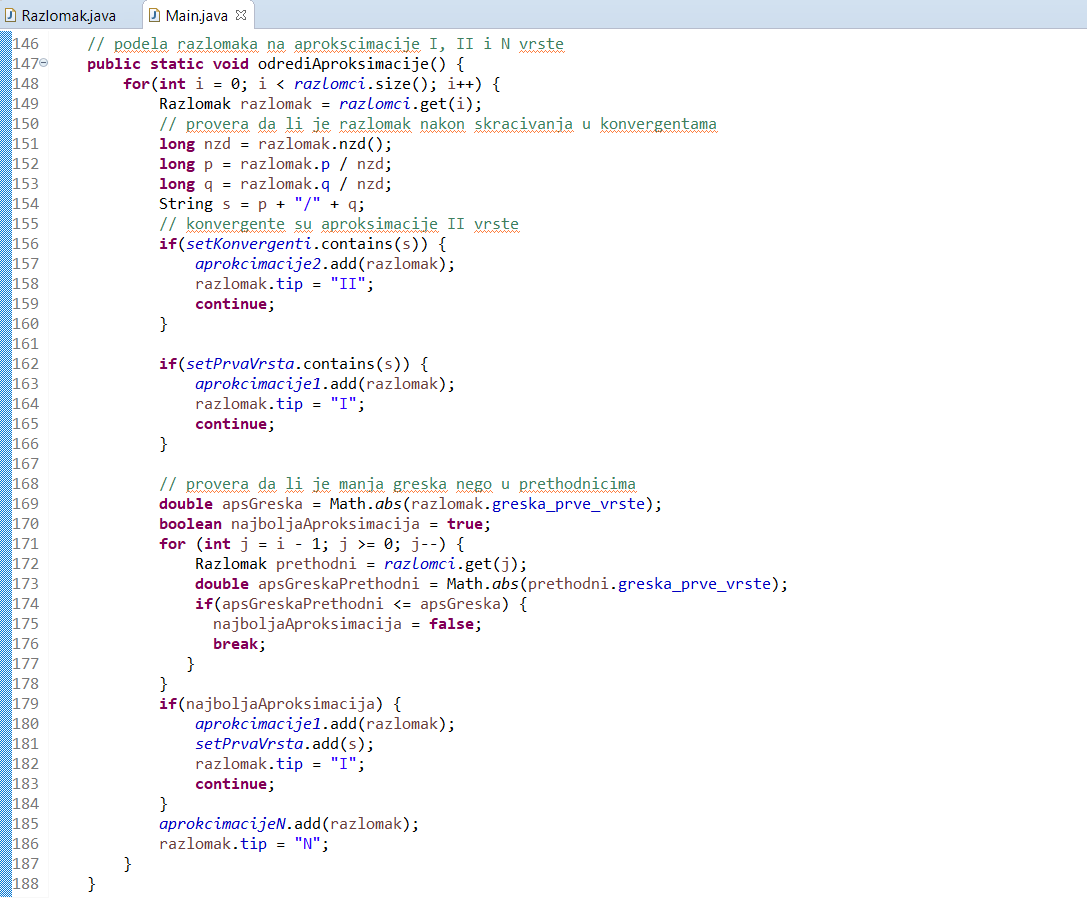
Slika 2.5. Main.java – deo 2



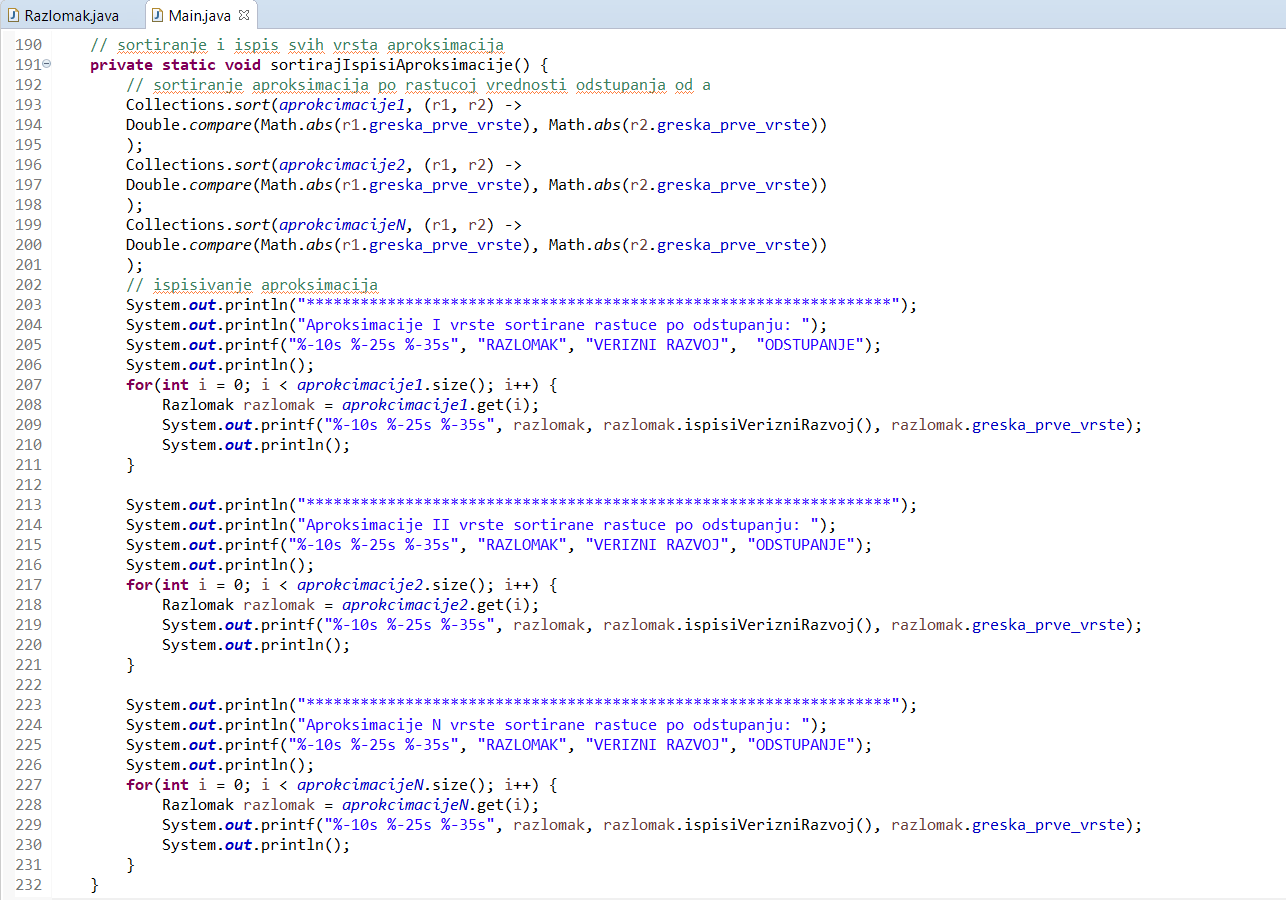
Slika 2.6. Main.java – deo 3



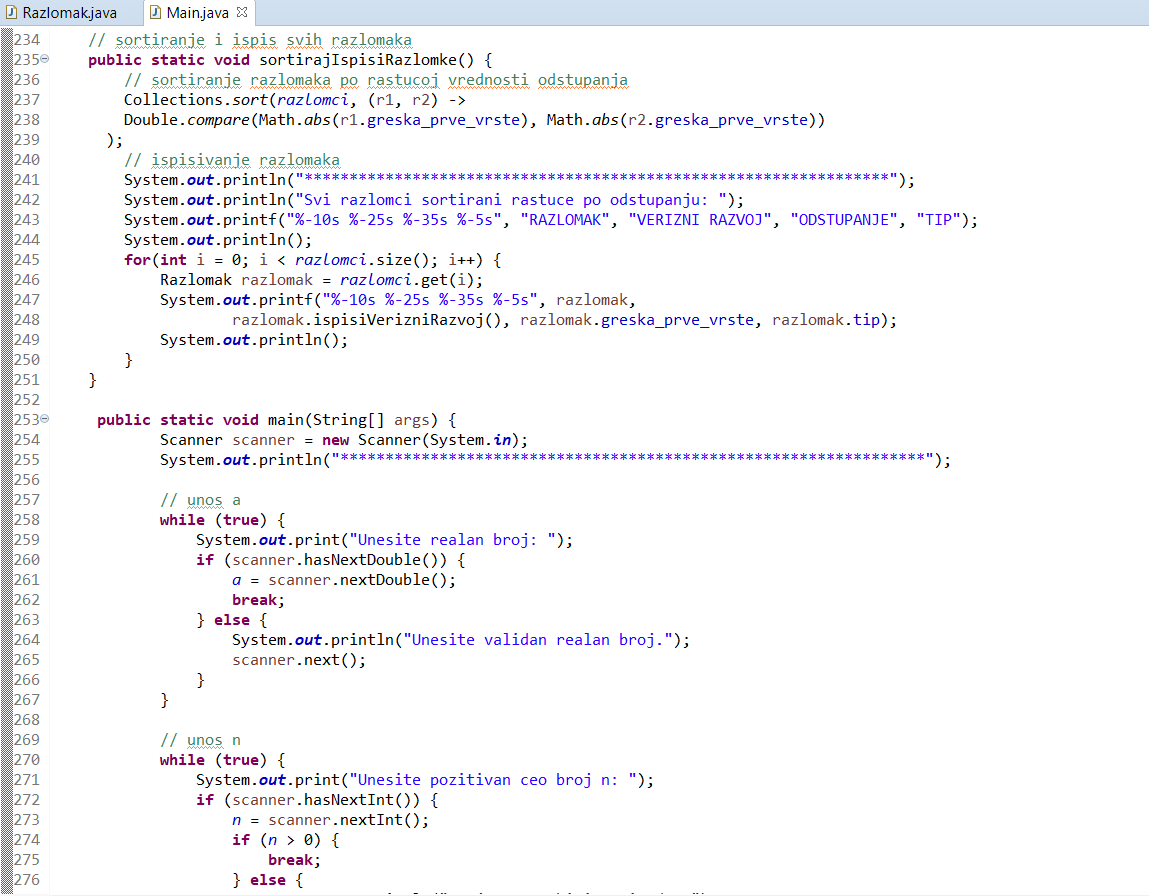
Slika 2.7. Main.java – deo 4



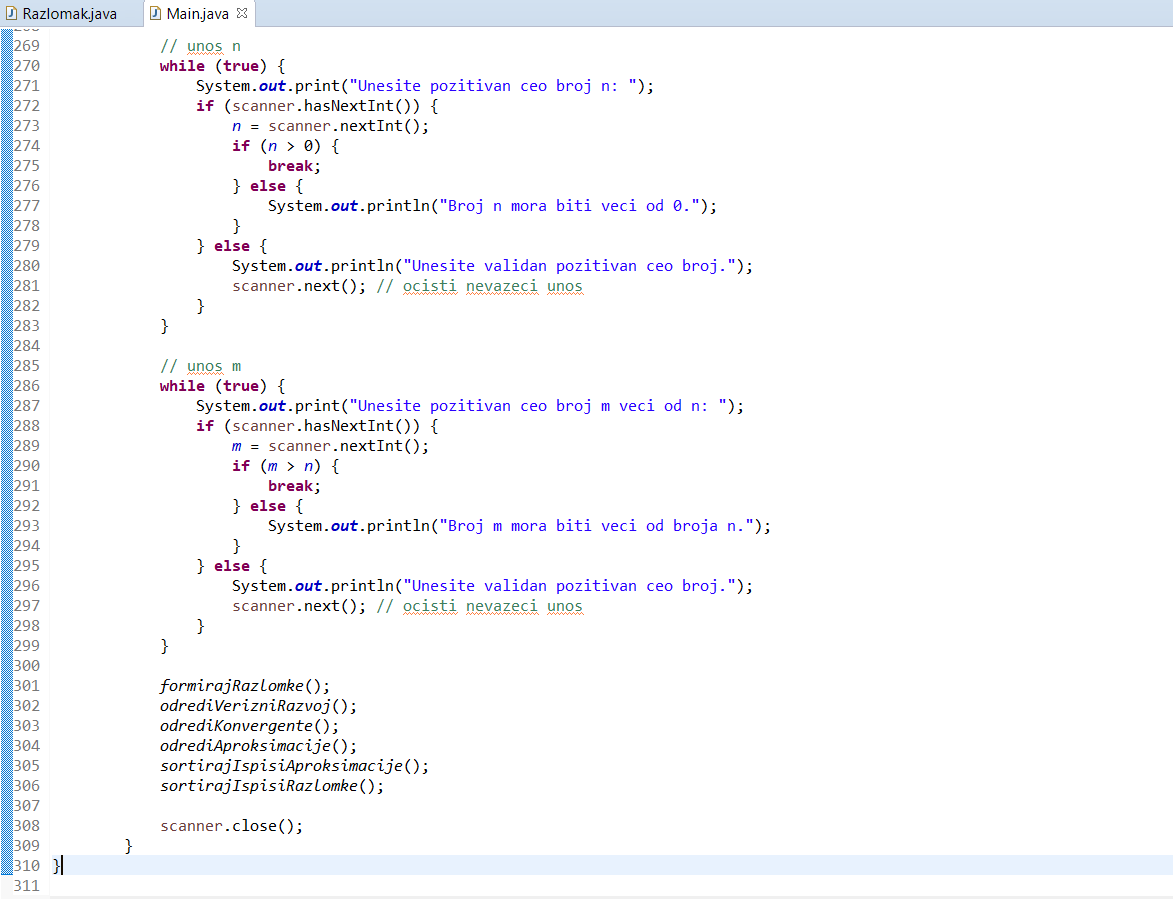
Slika 2.8. Main.java – deo 5



Slika 2.9. Main.java – deo 6



Slika 2.10. Main.java – deo 7



Slika 2.11. Main.java – deo 8

1. **Promenljive klase:**

* **public static double a:** Realan broj koji se aproksimira.
* **public static int n, m:** Donja i gornja granica za imenilac q.
* **public static ArrayList<Razlomak> razlomci:** Lista svih razlomaka koji se formiraju i analiziraju.
* **public static ArrayList<Razlomak> verizniRazvoj, nizX, nizA, nizD**: Liste koje se koriste za izračunavanje verižnog razvoja broja a.
* **public static ArrayList<Razlomak> konvergente**: Lista konvergenti broja a.
* **public static HashSet<String>** **setKonvergenti, setPrvaVrsta:** Skupovi setKonvergenti i setPrvaVrsta se koriste za efikasno praćenje i klasifikaciju razlomaka koji su već identifikovani kao konvergente ili kao najbolje aproksimacije prve vrste. Korišćenjem string reprezentacija razlomaka, program može brzo da proveri da li je trenutni razlomak već poznat i da ga svrsta u odgovarajuću kategoriju bez potrebe za ponovnim izračunavanjem ili poređenjem..
* **public static ArrayList<Razlomak** **aprokcimacije1**, **aprokcimacije2**, **aprokcimacijeN**: Liste aproksimacija I, II i N vrste.

1. **Metode:**

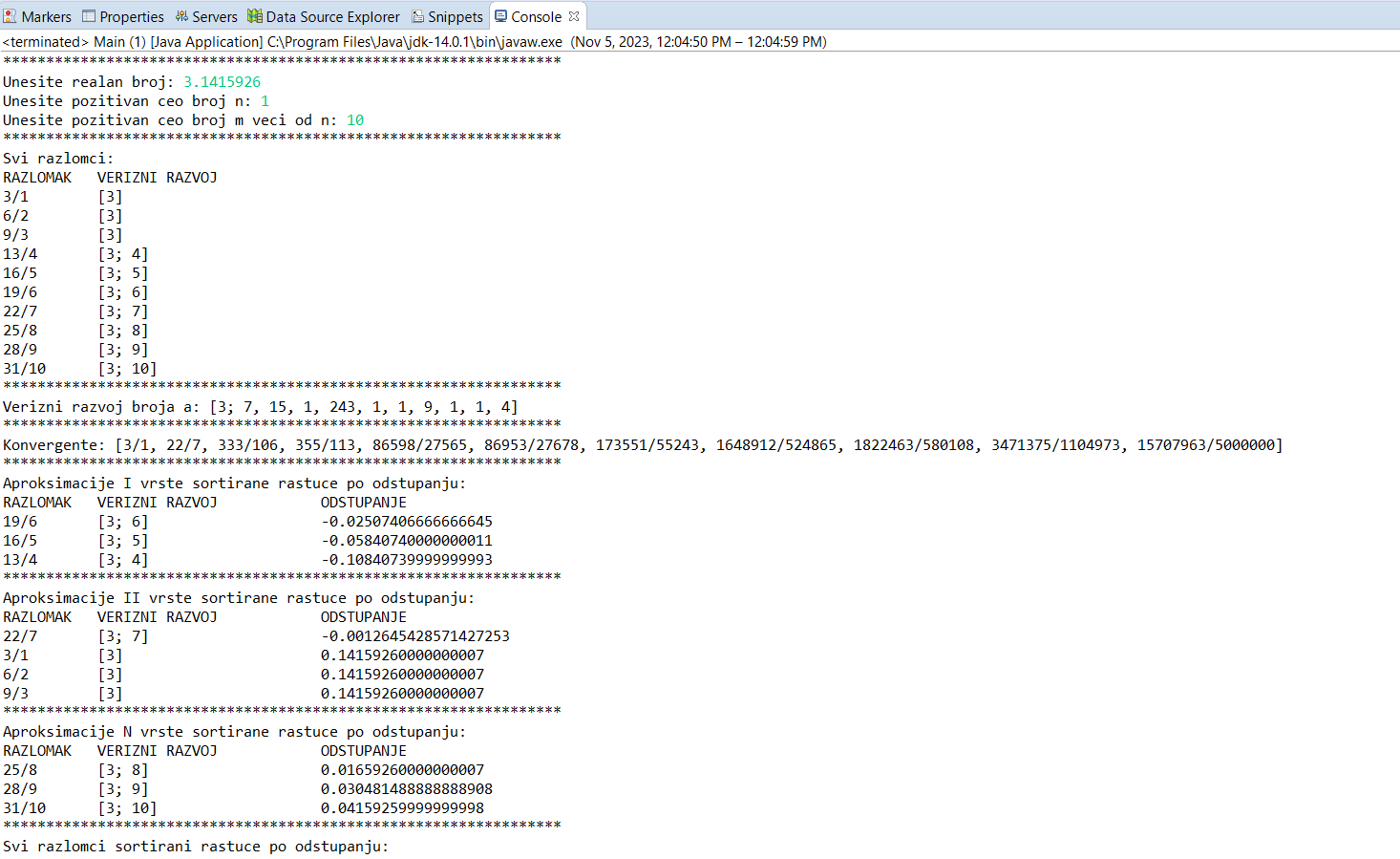
* **public static void formirajRazlomke():** Formira sve razlomke p/q u datom opsegu i izračunava njihov verižni razvoj i greške.
* **public static void odrediVerizniRazvoj():** Izračunava verižni razvoj realnog broja a.
* **public static void odrediKonvergente():** Izračunava konvergente za realan broj a koristeći verižni razvoj.
* **public static void odrediAproksimacije():** Klasifikuje razlomke u aproksimacije I, II ili N vrste na osnovu njihovih grešaka i odnosa sa konvergentima.
* **public static void sortirajIspisiAproksimacije():** Rastuće sortira aproksimacije po veličini greške prve vrste i ispisuje ih.
* **public static void sortirajIspisiRazlomke():** Rastuće sortira sve razlomke po veličini greške prve vrste i ispisuje ih zajedno sa njihovim tipom.
* **public static void main(String[] args):** Ova metoda je ulazna tačka programa. Ona upravlja unosom korisnika za realan broj a i granice n i m, a zatim poziva ostale metode da izvrše izračunavanja i ispišu rezultate.

Klasa Main koristi klasu Razlomak za reprezentaciju i manipulaciju razlomaka, kao i za izračunavanje verižnog razvoja i grešaka aproksimacija. Sve u svemu, klasa Main je glavni deo aplikacije koja koordinira proces aproksimacije realnog broja i prikazuje rezultate korisniku.

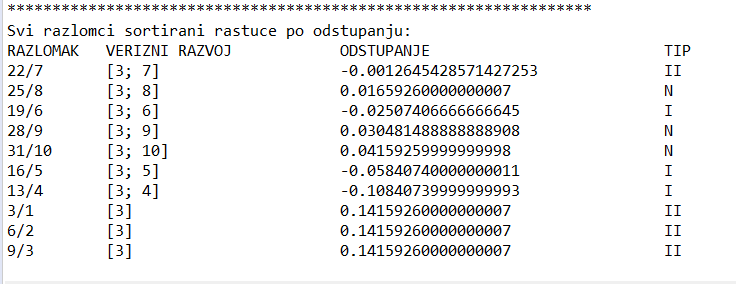
1. Testiranje rešenja projektnog zadatka

U nastavku će biti izloženo testiranje rešenja projektnog zadatka sa tri različita seta ulaznih podataka.

1. **Test primer 1:**

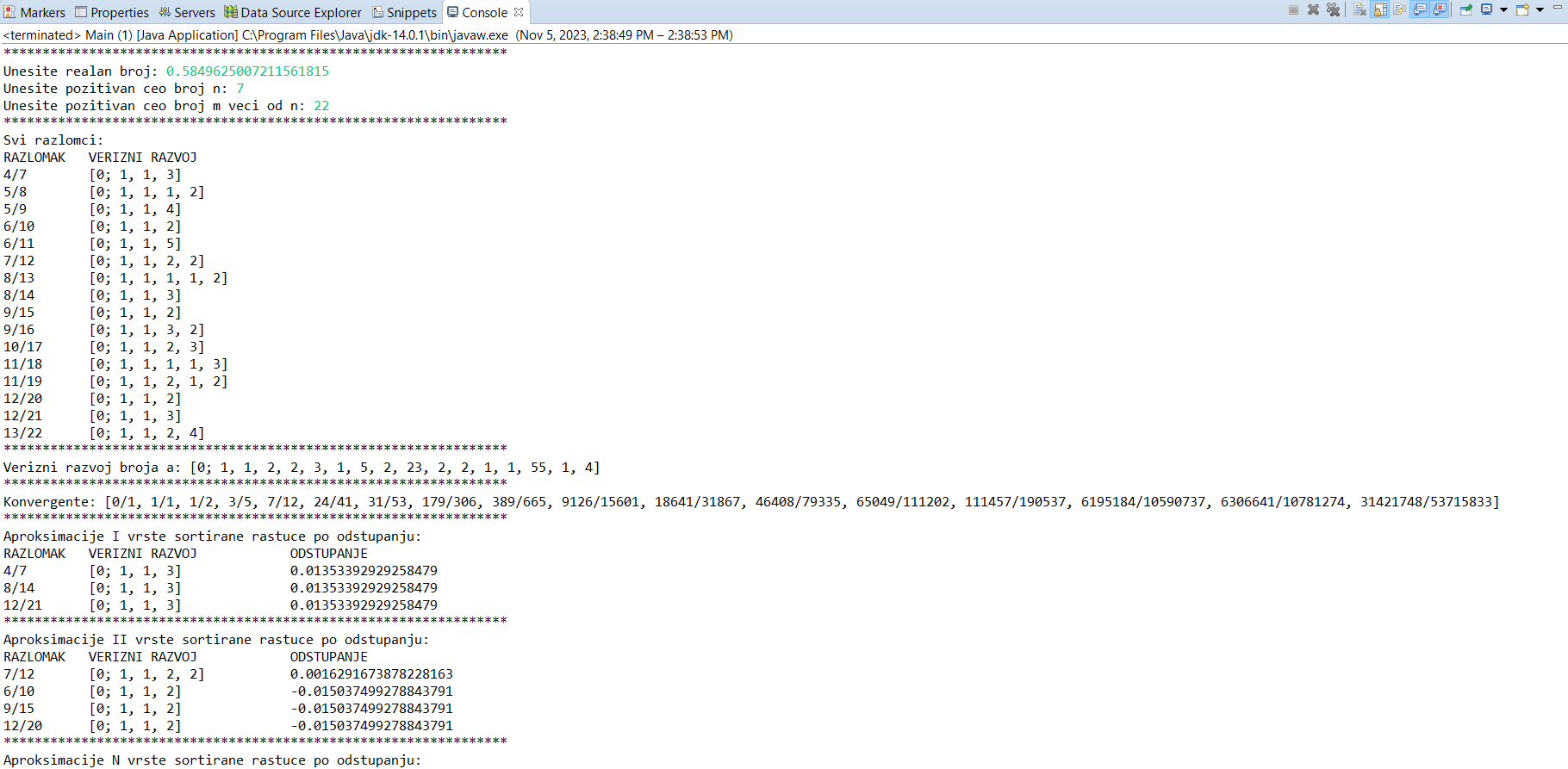


Slika 3.1. Test primer 1 – deo 1

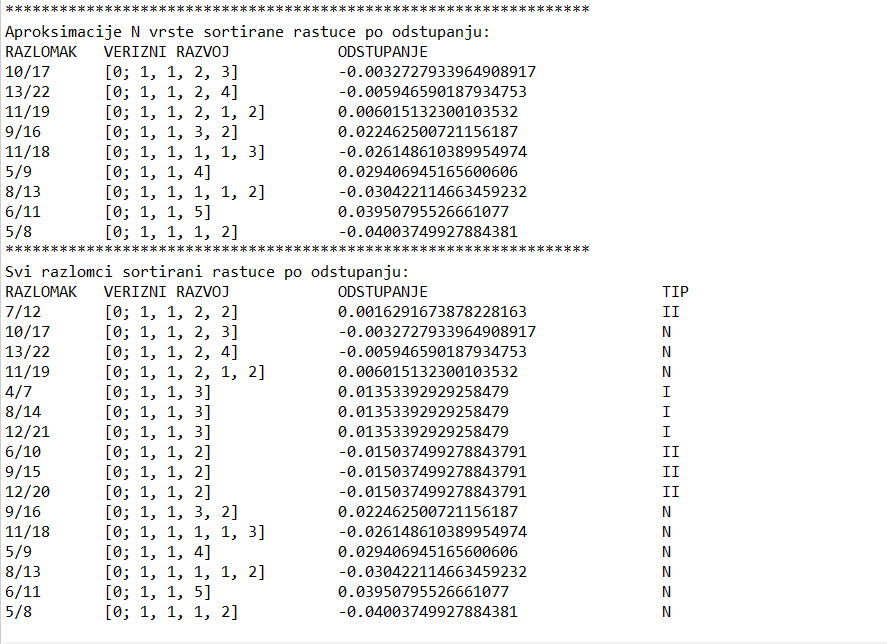


Slika 3.2. Test primer 1 – deo 2

1. **Test primer 2:**

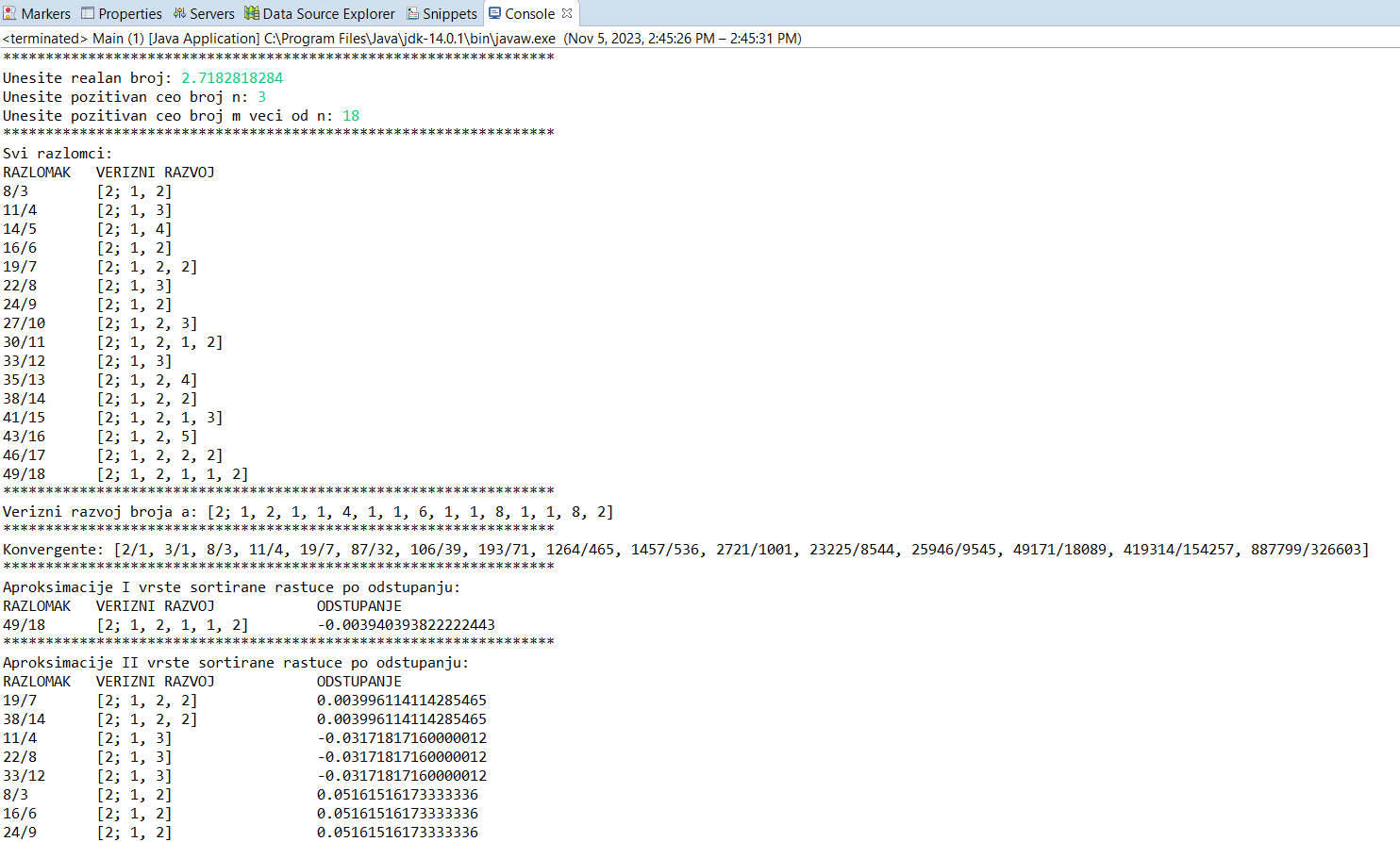


Slika 3.3. Test primer 2 – deo 1

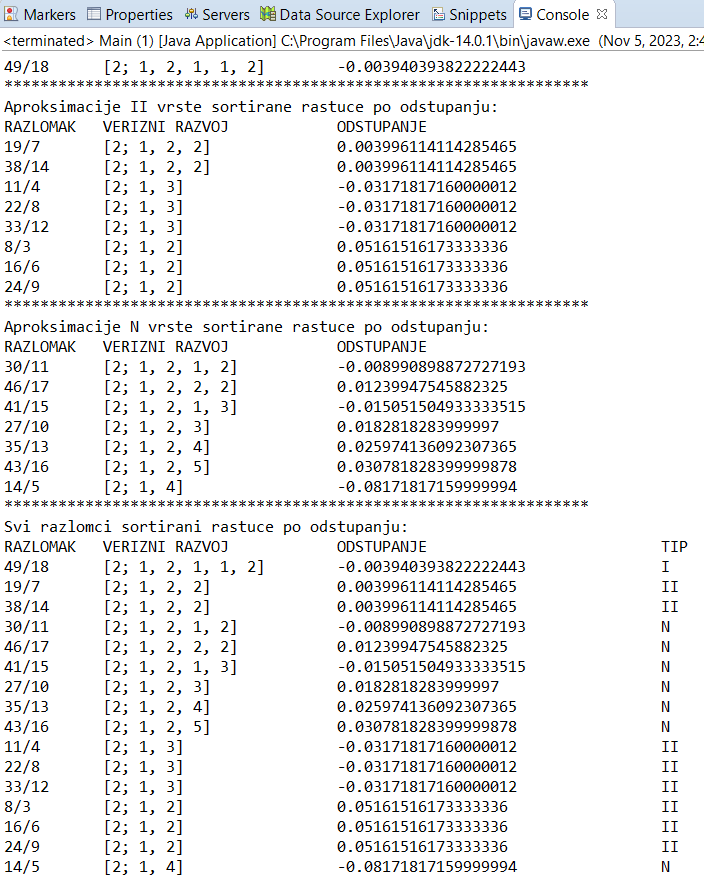


Slika 3.4. Test primer 2 – deo 2

1. **Test primer 3:**



Slika 3.5. Test primer 3 – deo 1



Slika 3.6. Test primer 3 – deo 2

Spisak slika

[Slika 2.1. Razlomak.java – deo 1 4](#_Toc150088311)

[Slika 2.2. Razlomak.java – deo 2 5](#_Toc150088312)

[Slika 2.3. Razlomak.java – deo 3 6](#_Toc150088313)

[Slika 2.4. Main.java – deo 1 8](#_Toc150088314)

[Slika 2.5. Main.java – deo 2 9](#_Toc150088315)

[Slika 2.6. Main.java – deo 3 10](#_Toc150088316)

[Slika 2.7. Main.java – deo 4 11](#_Toc150088317)

[Slika 2.8. Main.java – deo 5 12](#_Toc150088318)

[Slika 2.9. Main.java – deo 6 13](#_Toc150088319)

[Slika 2.10. Main.java – deo 7 14](#_Toc150088320)

[Slika 2.11. Main.java – deo 8 15](#_Toc150088321)

[Slika 3.1. Test primer 1 – deo 1 17](#_Toc150088322)

[Slika 3.2. Test primer 1 – deo 2 17](#_Toc150088323)

[Slika 3.3. Test primer 2 – deo 1 18](#_Toc150088324)

[Slika 3.4. Test primer 2 – deo 2 18](#_Toc150088325)

[Slika 3.5. Test primer 3 – deo 1 19](#_Toc150088326)

[Slika 3.6. Test primer 3 – deo 2 19](#_Toc150088327)

Literatura

1. Branko J. Malešević, Najbolje racionalne aproksimacije (2023).pdf