

*Prolećni semestar, 2023/24*

*PREDMET: CS203 – Algoritmi i Strukture Podataka*

Projektni Zadatak 1: Largest Box of 1s

Ime i prezime: **Petar Otovic**

Broj indeksa: **5460**

Datum izrade: **18/4/2024**

Profesor: Milena Bogdanovic

Asistent: Uros Lazarevic

# 1. Uvod u projekat

**1.1 Tekst zadatka:**

Moj projektni zadatak je da napravim algoritam koji ce pronalaziti najveci pravougaonik ili kocku sa svim jedinicama u matrici MxM duzine. Algoritam je napravljen u programskom jeziku Java i izvrsavace se u konzoli aplikacije.

**1.2 Arhitektura zadatka:**

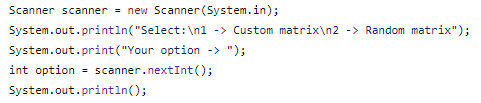
Zadatak je vrlo jednostavan i sastoji se iz 3 fajla ukupno. Glavni program ce se izvrsavati, naravno, u **Main.java** klasi gde se pozivaju metode. Glavna logika celog zadatka se nalazi u fajlu **CustomAlgorithm.java,** gde se nalaze metode i sam algoritam za pronalazenje najveceg pravougaonika popunjenog jednicima. Postoji i fajl Utils.java koji sadrzi pomocne metode za rad sa matricom, kao sto je stampanje matrice i validacija polja matrice.

Aplikacija je zamisljena tako da kada se pokrene program, korisniku se nude 2 opcije. Prva opcija je rucno unosenje matrice, gde ako korisnik izabere ovu opciju, prikazace mu se poruka na konzoli da izabere dimenzije matrice. Dimenzija se unosi samo jednom zato sto je matrica velicine MxM i nakon toga korisnik popunjava polja matrice unosenjem stringa u formatu 1,1,0,1.... gde ce se kasnije obraditi ovaj ulaz i smestiti u odgovarajuca polja matrice. U slucaju neispravnog unosa, program ce baciti Exception. Nakon popunjene matrice, uneta matrica se stampa a nakon toga se izvrsava algoritam za pronalazenje najveceg pravougaonika ili kocke. Nakon izvrsenog algirtma prikazuje se poruka sa najvecim dimenzijama pravougoanika ako postoji. Prikazuje se i vreme izvrsenja algoritma, sto su obicno nanosekunde.

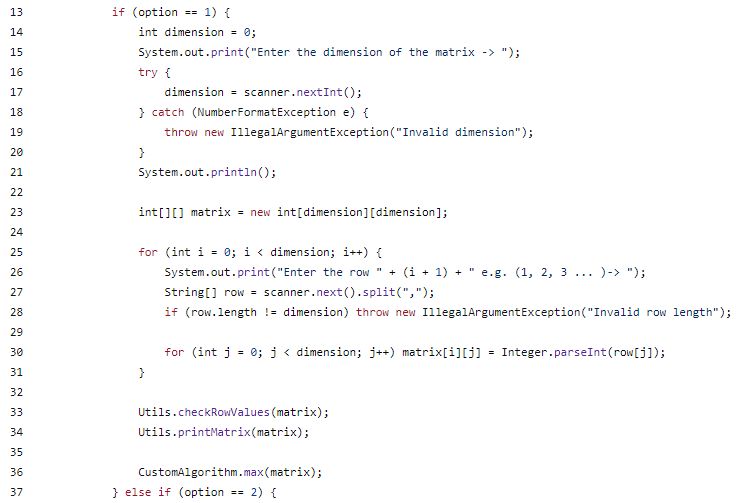
Druga opcija koju moze da izabere korisnik, je opcija da se matrica sama generise. Velicina generisane matrice moze da bude izmedju 5 i 15, ovo ogranicenje je stavljeno zbog preglednosti kada se stampa matrica, medjutim, po potrebi, moze se lako izmeniti u **Main.java** klasi. Nakon genersianja matrica, ona ce se prikazati a zatim prikazati i dimenzija pronadjenog pravougaonika kao i vreme izvrsenja.

# 2. Klase i metode

**- Main.java**



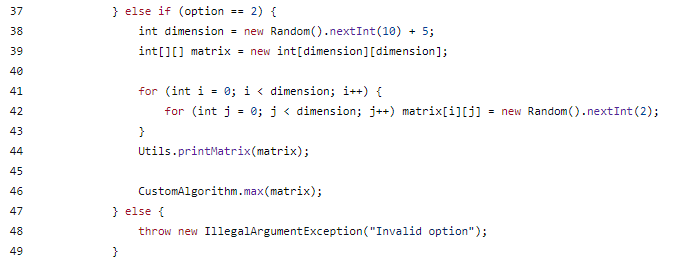
Na pocetku Main.java klase se naravno bira opcija. 1. opcija da korisnik sam unese matricu, 2. da se matrica sama izgenerise.



Ako se izabere opcija jedan, unosi se dimenzija matrice (linija 17), u slucaju neispravnog unosa baca se Exception. Nakon toga sledi deklaracija matrice sa unetim dimenzijama (linija 23). Nakon toga se u prvom for loop-u (linija 25) popunjava jedan po jedan red matrice. U drugom for loop-u se parsira uneta vrednost za dati red matrice i smesta u matricu naravno. Nakon toga, zbog mogucih gresaka u rucnom popunjavanju matrice, mora se proveriti svaka vrednost u matrici u slucaju da nije jednaka 1 ili 0. Poziva se metoda iz klase **Utils, checkRowValues**.

Nakon toga matrica se stampa isto metodom iz klase **Utils,** printMatrix, koja stampa matricu.

Na kraju naravno, poziva se algoritam koji pronalazi najveci pravougaonik, metoda **max** (imenovanje metoda intuitivnost level 100) iz klase **CustomAlgorithm.**



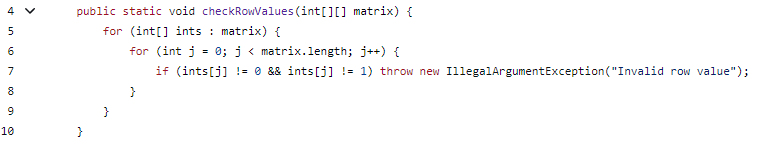
U slucaju da je izabrana opcija 2, tj. Opcija gde ce program sam napraviti matricu, generise se nasumicno dimezija matrica (linija 38). Nakon toga sledi popunjavanje matrice nulama i jedinicama (linija 42).

Nakon toga matrica se prikazuje, kako bi vizuelizovali generisanu matricu (linija 44). I na kraju se poziva algoritam za pronalazenje najveceg pravougaonika (linija 46).

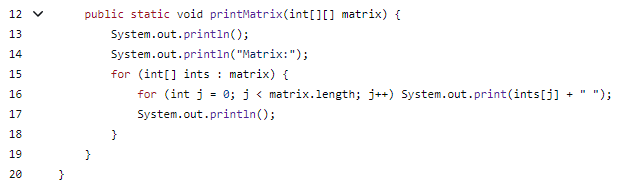
U slucaju da je broj izabrane opcija neispravan bacice se Exception.

* **Utils.java:**

Ovo je klasa koja sadrzi 2 pomocne metode.



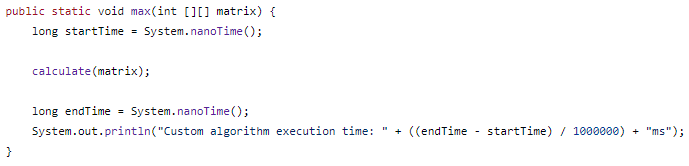
Prva metoda je za proveru svakog polja matrice, tj. Provera da li je vrednost polja 1 ili 0. U slucaju da nije baca se Exception.



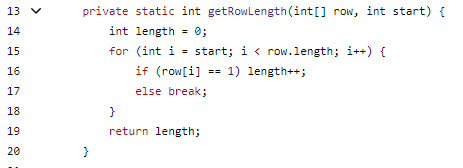
Druga metoda je jednostavna metoda za stampanje matrice.

* **CustomAlgorithm.java:**

Ova klasa ima 3 staticke metode, od kojih su 2 **private** i jedna je **public.**

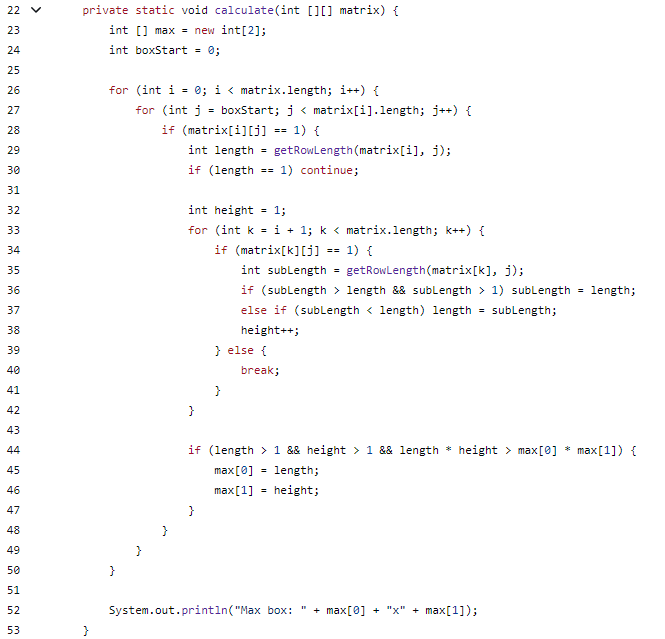


Ova public metoda poziva jednu funkciju za izvrsavanje algoritma, **calculate,** i prikazuje vreme izvrsavanja algoritma u milisekundama.



Prva **private** metoda je metoda koja malazi duzinu jedinica u jednom redu, pocevsi od zadatog indeksa. Na kraju vraca pronadjenu duzinu.

Druga **private** metoda je metoda koja pronalazi najveci pravougaonik, i ovo je glavna funckija.



Metoda je zamisljena tako da prolazi kroz svaki red matrice i nalazi najvecu duzinu jedinica u redu. U slucaju da se nadje duzina od najmanje 2 jedinice u nekom redu, uzece se pronadjena duzina i traziti ista duzina u redu ispod. U slucaju da je nadjena odgovarajuca duzina i u redu ispod, nastavlja se red ispod ponovo itd, sve dok se ne naidje na red koji nema odgovarajucu duzinu sa jednicama. U slucaju da je nova pronadjena dimezija pravougaonika veca od postojece, ona se pamti.