*CS203 – Algoritmi i strukture podataka*

**Recursive tree**

Profesor: Milena Bogdanovć Student: Aleksa Stanojević, 5726

Asistent: Uroš Lazarević Datum: 31/03/2024



# Uvod

## Postavka Zadatka

Iz predmeta CS203, Algoritmi i strukture podataka, projektni zadatak je iscrtavanje rekurzivnog drveta u JavaFX virtuelnom okruženju. Program je izrađen u programskom jeziku Java uz pomoć JetBrains, IntelliJ IDEA razvojnog okruženja. Program koristi JavaFX grafički korisnički interfejs za iscrtavanje stabla.

Program je zamišljen tako da korisnik ima mogućnost unosa vrednosti dubine stabla kao i podešavanje opcije za animirano crtanje istog. Prevlačenjem mišem stabla preko prozora omogučava se opcija za pozicioniranje stabla bilo gde u prostoru programa.

Korisnik unosi u polje za unos dubinu rekurzivnog stabla. Dubina stabla predstavlja brojčanu vrednost od 0 do maksimalne vrednosti 15. Ukoliko je uneta vrednost ispod nule program će obraditi vrednost 0, dok ako korisnik unese vrednost veću od 15 program će obraditi maksimalnu moguću vrednost dubine 15.



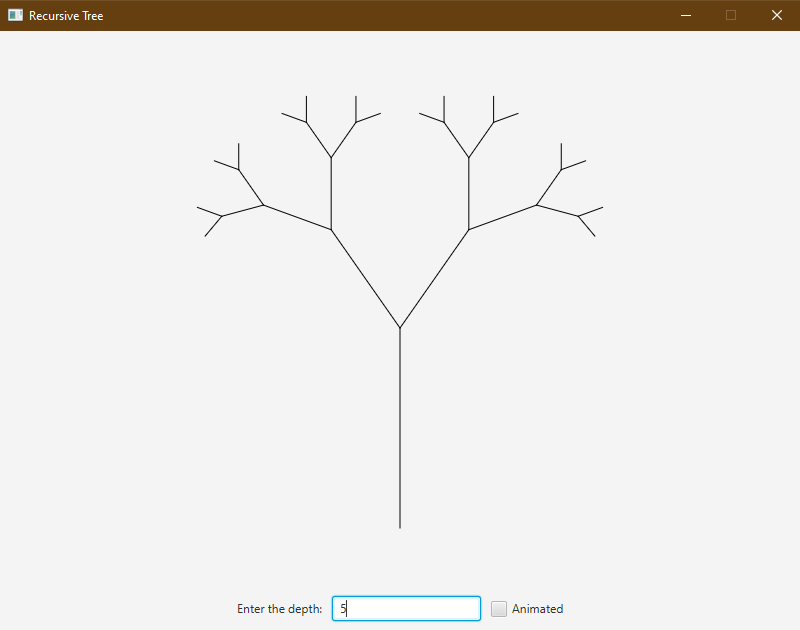
Korisnik ima mogućnost biranja opcije za animirano iscrtavanje stabla. Ukoliko je ova opcija čekirana program će iscrtavati stablo tako što će iterativno crtati granu po granu kako se prolazi kroz rekurziju.



Nakon unošenja vrednosti dubine (ili fokusiranja polja za unos), prilikom pritiska tastera Enter na tastaturi program će započeti iscrtavanje rekurzivnog stabla. Program će prvo očistiti prethodno iscrtano stablo ukoliko postoji, a zatim će iscrtati kompletno stablo ukoliko animacija nije aktivirana, dok će iscrtavati granu po granu u slučaju da je animacija aktivna.

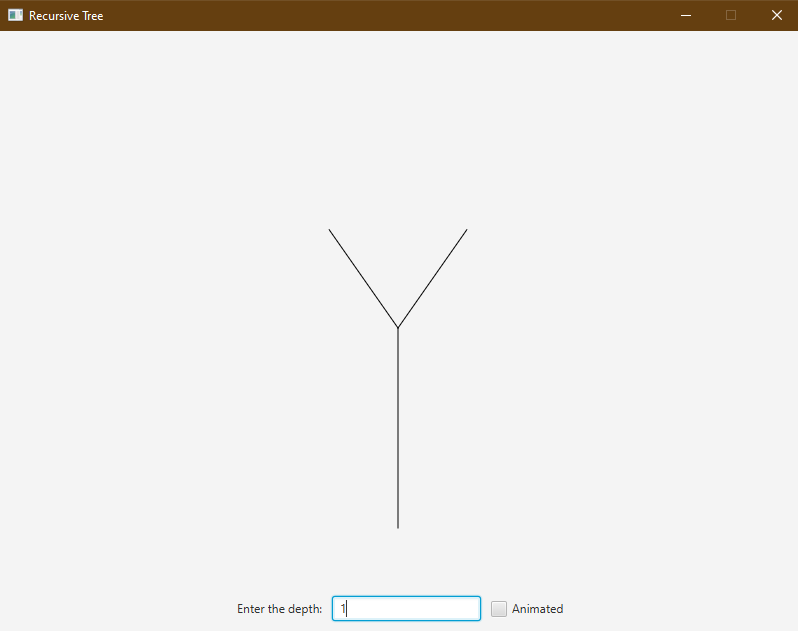
## Prikaz rezultata

Rešenje projektnog zadatka je program ***“Recursive Tree”***. Kada se program pokrene korisnicima se prikazuje prozor koji služi za iscrtavanje stabla i interakcije sa grafičkim korisničkim interfejsom. Početna podešavanja prikazuju drvo za dubinu rekurzije 5.

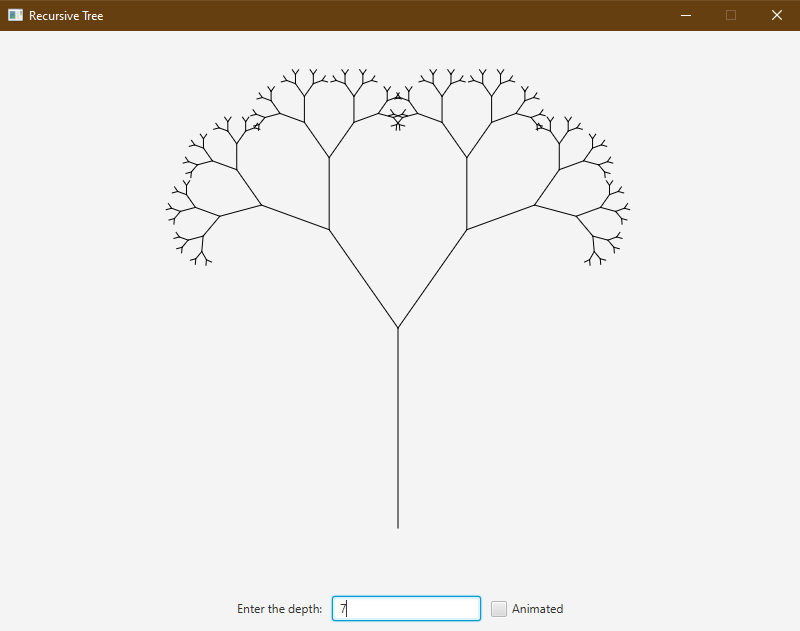


1. *Glavni Prozor*

Nakon što korisnik unese validnu vrednost u polje za unos dubine, program iscrtava rekurzivno stablo za datu dubinu.

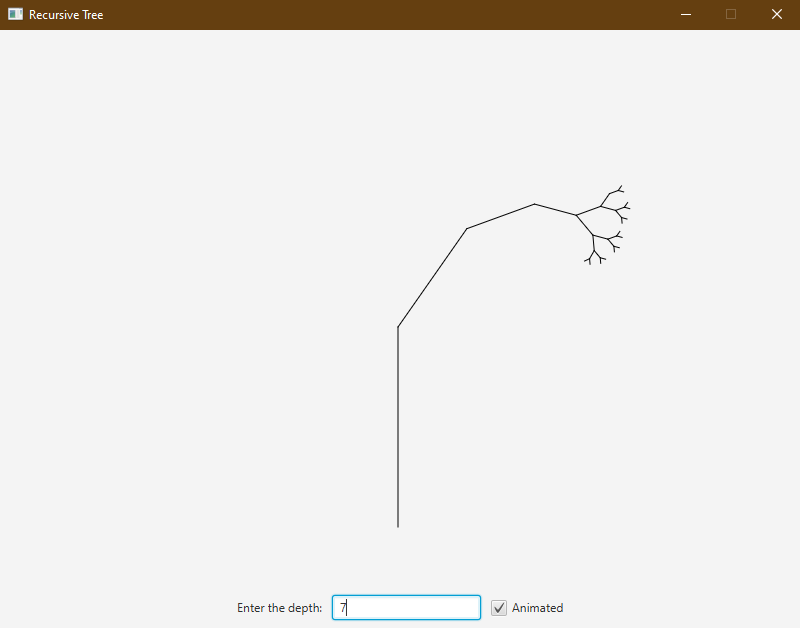


1. *Rekurzivno drvo za vrednost 1*

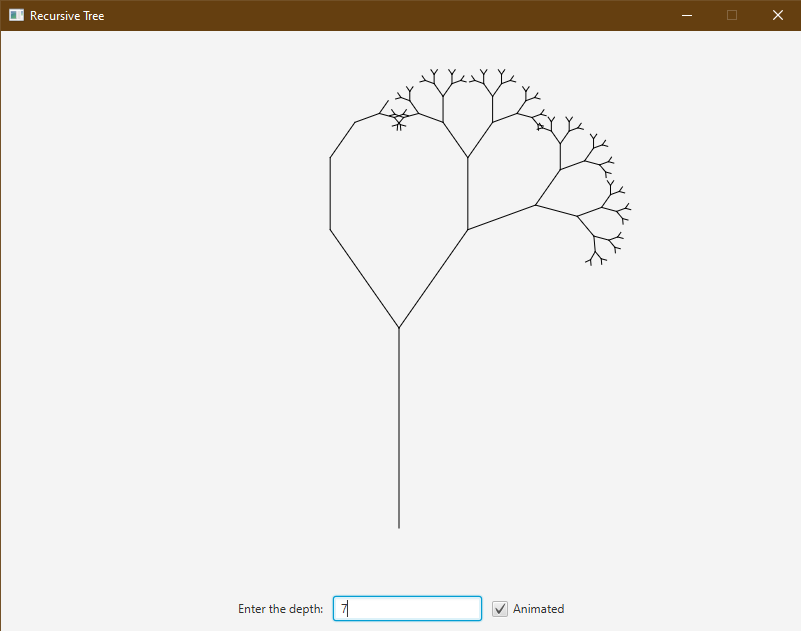


1. *Rekurzivno drvo za vrednost 7*

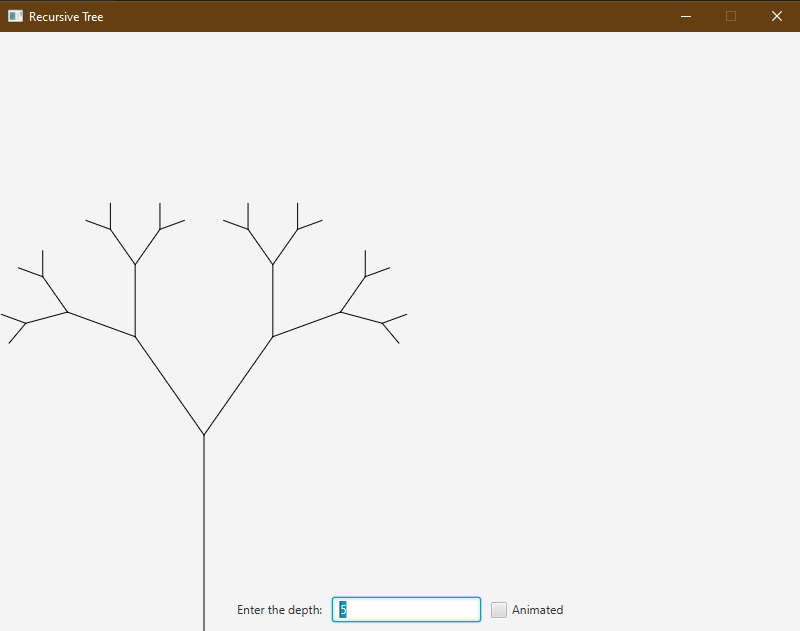
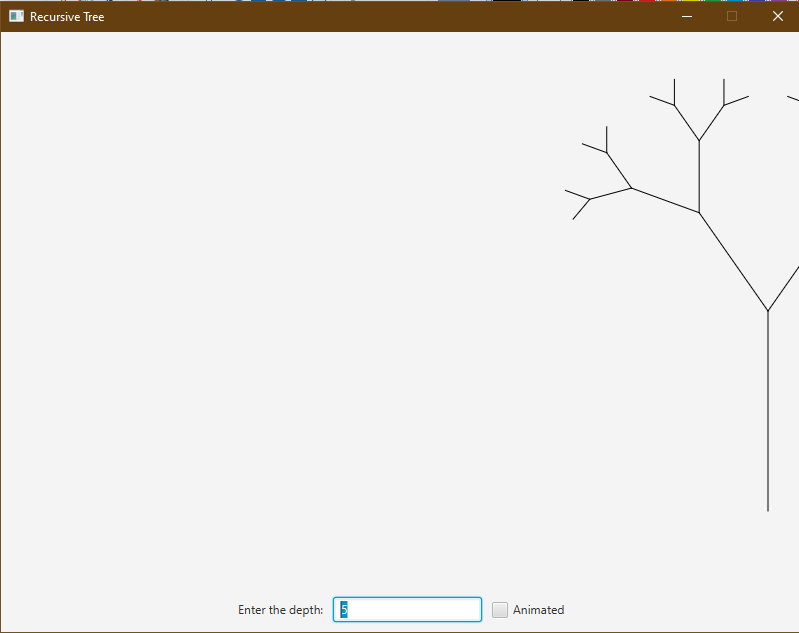
U slučaju da korisnik čekira opciju animiranog crtanja rekurzivnog drveta, program će početi da iscrtava drvo granu po granu sve dok ne završi kompletno iscrtavanje drveta.



1. *Animirano iscrtavanje vreme: t*

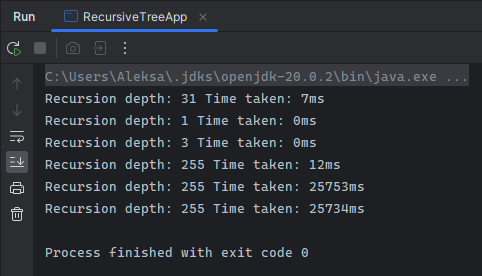


1. *Animirano iscrtavanje vreme: t+1*

Prevlačenjem rekurzivnog drveta po prozoru programa, stablo menja svoju poziciju prateći miš. 

1. *Pomeranje rekurzivnog drveta*

Takođe se u konzoli ispisuje dubina rekurzije kao i vreme izvršenja iscrtavanja rekurzivnog drveta.



1. *Prikaz konzole*

# Arhitektura

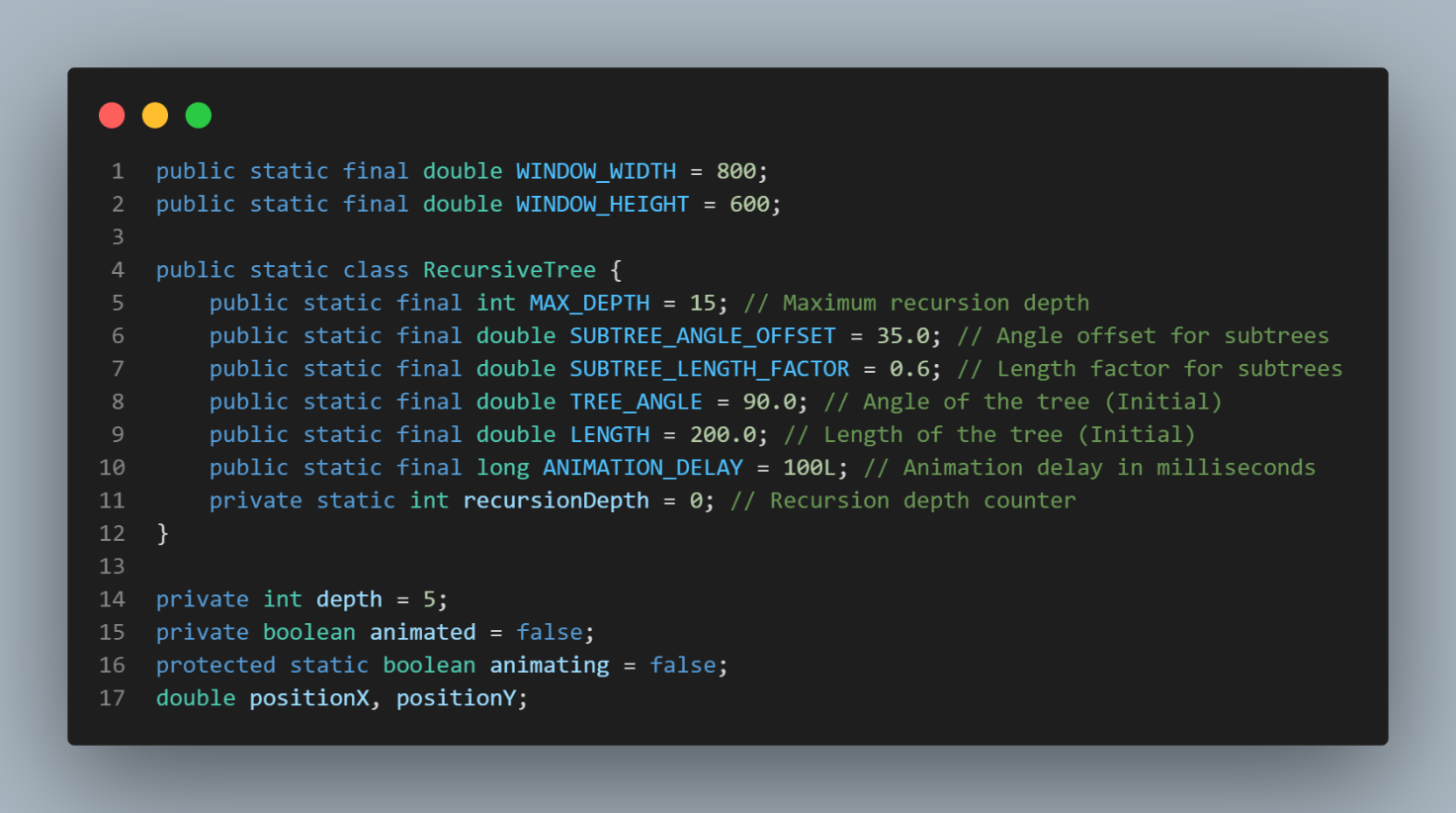
## Arhitektura igre

Kao što je već rečeno, program je izrađen u programskom jeziku Java korišćenjem JetBrains, IntelliJ IDEA razvojnog okruženja.

## Podešavanje programa pomoću globalnih promenljivih

Pre početka rada programa podešavamo veličinu prozora pomoću vrednosti WINDOW\_WIDTH i WINDOW\_HEIGHT. To su finalne promenljive što znači da se promenljive ne mogu modifikovati tokom izrade igre. Nakon toga pravimo vrednosti klase RekursiveTree, to su statične konstantne vrednosti MAX\_DEPTH,SUBTREE\_ANGLE\_OFFSET,SUBTREE\_LENGTH\_FACTOR,TREE\_ANGLE,LENGTH,ANIMATION\_DELAY.

Podešavamo vrednosti programa depth kao dubinu koju korisnik unosi, animated kao vrednost da li je program animiran ili ne, animating kao vrednost da li je u toku animacija programa ili ne kao i positionX i positionY kao trenutne pozicije stabla na prozoru.

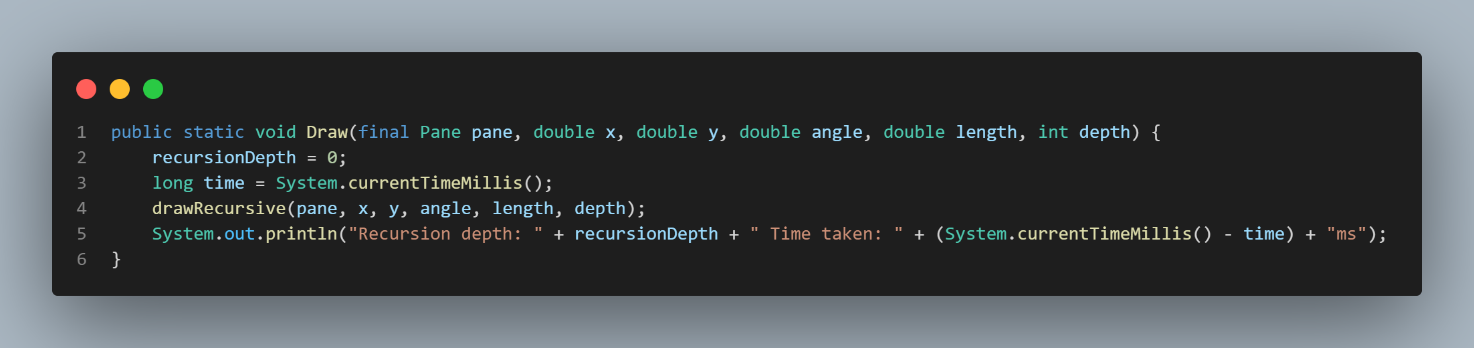


1. *Podešavanje globalnih promenljivih pre početka programa*

## Klasa RecursiveTree

Kako bi se olakšao rad programa konstruisana je pomoćna klasa RecursiveTree. Ona služi za lakšu obradu rekurzivnog iscrtavanja rekurzivnog stabla sa parametrima.

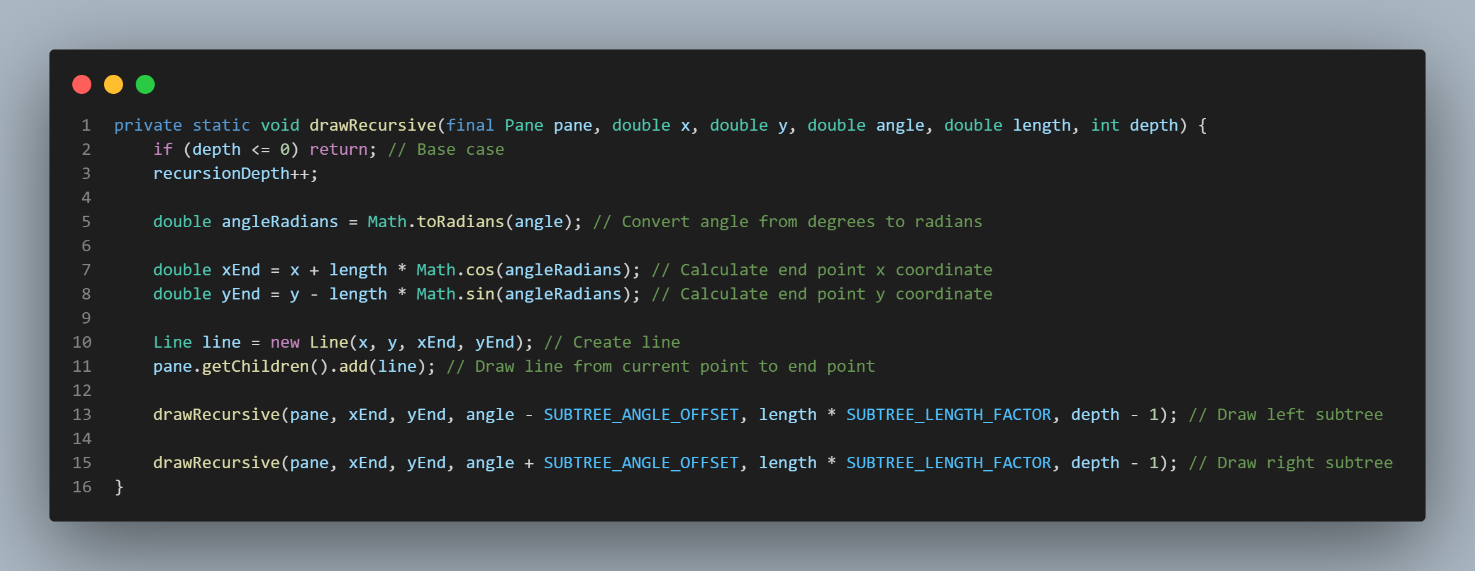
Metoda Draw služi za pozivanje rekurzivne metode drawRecursive. Ona pre svega resetuje vrednost dubine rekurzije na 0, zatim uzima trenutno vreme statičnom metodom System.currentTimeMillis(), nakon čega poziva metodu drawRecursive (sa parametrima pane,x,y,angle,length,depth) i na kraju štampa poruku o dubini rekurzije kao i vremenu potrebnom za iscrtavanje stabla.



1. *Draw metoda*

Metoda drawRecursive služi za rekurzivno iscrtavanje stabla. Kao parametre metoda prihvata pane na kome se iscrtava stablo, x i y poziciju početne tačke iscrtavanja grane na zadatom pane-u, angle kao vrednost ugla pod kojim se grana iscrtava u odnosu na normalu tačke početne pozicije, length kao dužinu grane, kao i depth koja označava dubinu subgrana prethodne grane.

Metoda pre svega proverava da li je dubina manja ili jednaka nuli kako bi prekinula rekurzivne pozive (bazni slučaj). Nakon toga metoda inkrementuje brojač rekurzivne dubine, nakon čega računa ugao u radianima tako što koristi matematičku metodu statične klase Math.toRadians(degrees). Nakon izračunavanja ugla u radianima računa se krajnja tačka grane formulom . Zatim se iscrtava linija korišćenjem klase Line koja kao parametre prima x,y kao početnu tačku line i x1,x2 kao krajnju tačku linije koja se zatim dodaje kao child objekat na prosleđeni pane. Nakon svega toga dva puta se poziva rekurzivna metoda drawRecursive, jedna za levu granu podgrane dok je druga za desnu granu. Parametri ova dva poziva rekurzivne metode su uglavnom isti dok se samo ugao za levu granu oduzima sa vrednosti SUBTREE\_ANGLE\_OFFSET kako bismo rotirali sledeću levu granu za ugao te vrednosti dok za desnu stranu dodajemo tu vrednost za ugao. Što se tiče dužine grane, potrebno je obe grane pomnožiti sa vrednosti SUBTREE\_LENGTH\_FACTOR kako bi sve dublje I dublje grane bile sve kraće kako je I definisano u zahtevima projekta. Na samom kraju parametar depth(dubina) potrebno je umanjiti za 1 kako bismo ispratili dubinu grananja i kako bi sprečili beskonačnu rekurzivnu petlju (stack overflow).



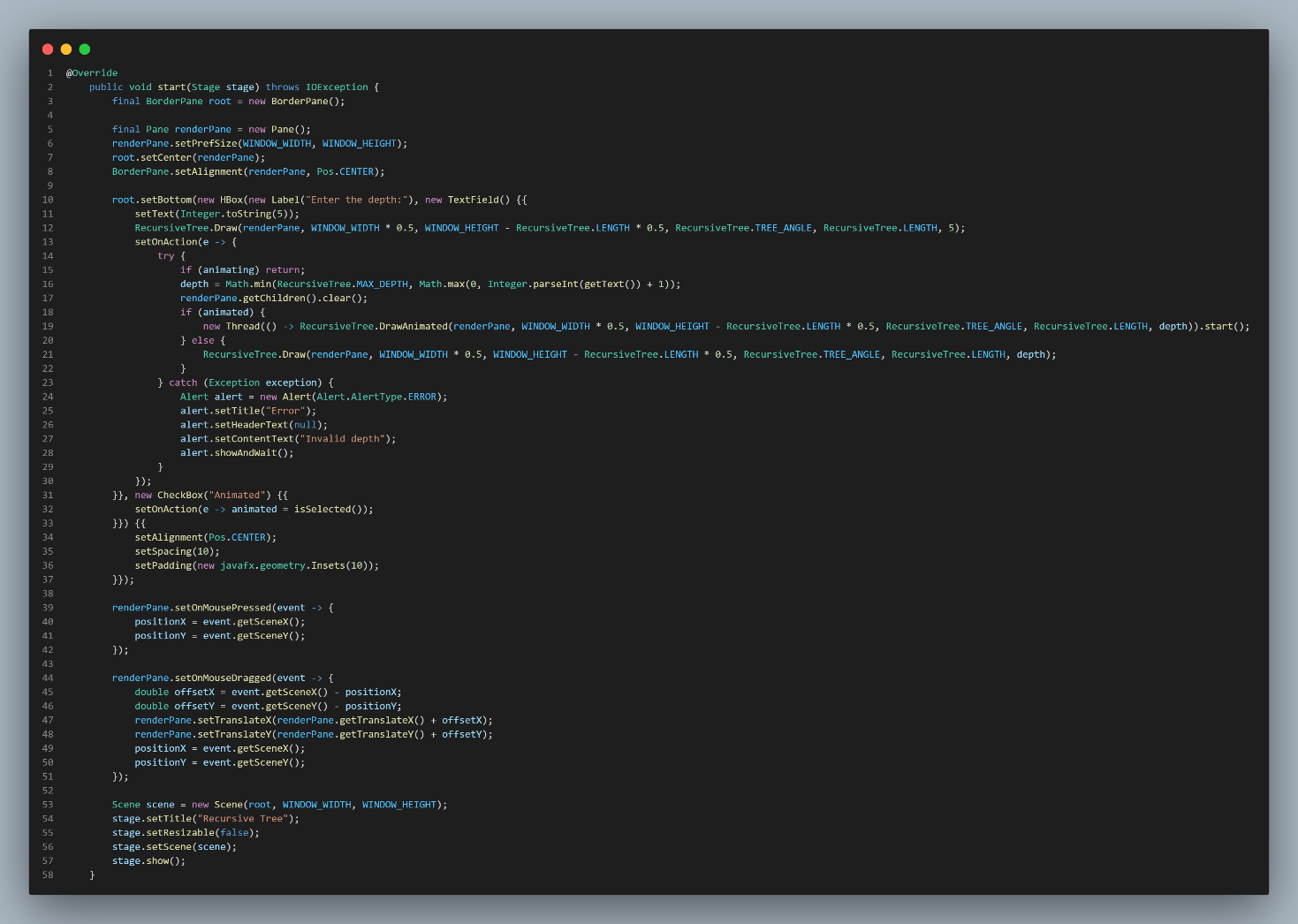
1. *drawRecursive metoda*

Što se tiče metoda DrawAnimated i drawAnimatedRecursive princip je uglavnom isti. Kod metode drawAnimatedRecursive smo liniju pane.getChildren().add(line) zamenili linijom Platform.runLater(() -> pane.getChildren().add(line)) kako bismo nagovestili javaFX thread-u da je potrebno dodati novu liniju tek nakon završetka svih prethodnih obrada FX funkcionalnosti. Takođe smo dodali Thread.sleep(ANIMATION DELAY) kako prilikom svakog rekurzivnog poziva uspavali thread na određeno vreme kako bismo simulirali animirano iscrtavanje grane po granu.



1. *Animirano iscrtavanje*

U glavnom programu (odnosno start metodi javaFX aplikacije) podešavamo BorderPane kao root parent node naše aplikacije. Kreiramo novi Pane kao renderPane na kome će se iscrtavati naše stablo. Podešavamo veličinu pejna i stavljamo ga u sam centar naše aplikacije. Zatim pravimo HBox koji postavljamo u donji deo naše aplikacije i njega podešavamo na sledeći način, dodajemo labelu za tekst unosa dubine, dodajemo TextField koje služi za unos vrednosti dubine. Polje za unos dubine podešavamo tako što podešavamo početni tekst na 5, zatim pozivamo našu rekurzivnu metodu za iscrtavanje rekurzivnog drveta, nakon čega podešavamo akciju za promenu vrednosti polja. Kada polje promeni vrednost potrebno je pročitati vrednost iz polja kao dubinu stabla, zatim očistiti prozor za iscrtavanje (prethodnu sliku) nakon čega proveravamo da li je potrebno pozvati animirano ili obično iscrtavanje stabla. Sve ovo se nalazi u try bloku kako bismo uhvatili bilo kakvu grešku prilikom izvršenja programa. Ako se greška uhvati, prikazaće se poruka sa tekstom „Invalid Depth“. Vraćamo se na Hbox i dodajemo CheckBox kako bismo pratili promene vrednosti animiranog programa i podešavamo akciju na promenu tako što menjamo boolean vrednost za praćenje animacija. Pozicioniramo CheckBox i time završavamo podešavanja Hboxa. Nakon dodavanja eventa za miš-a kreiramo novu scenu kojoj prosleđujemo kreirani root, podešavamo naziv, podešavamo fiksnu veličinu kao i pozivamo prikaz scene.



1. *Start metoda*

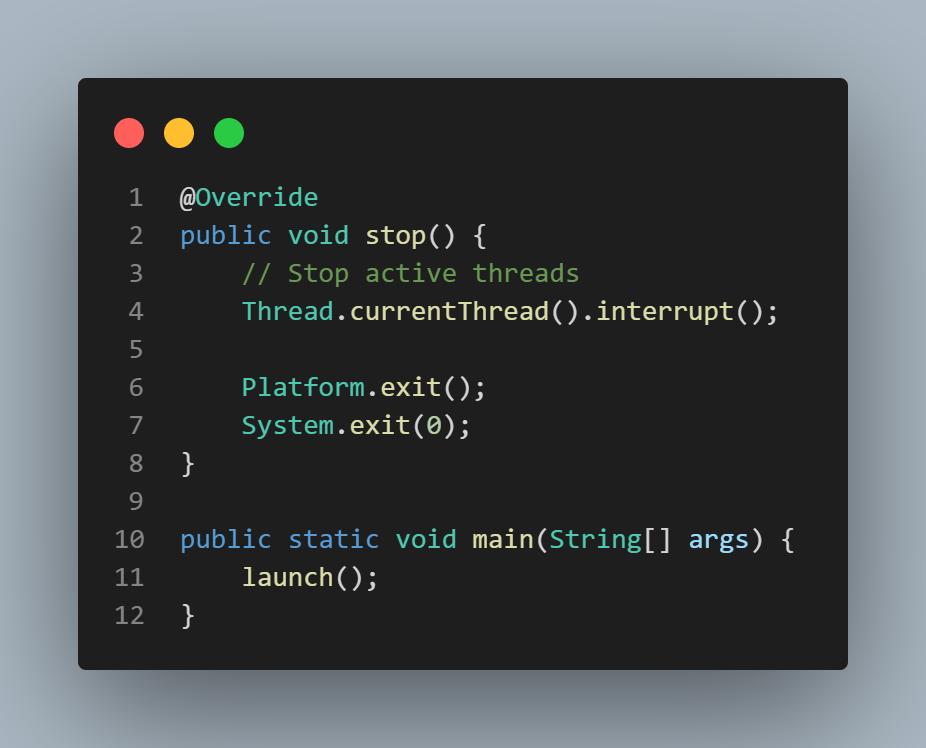
Akcije za osluškivanje radnji sa mišem rade po sledećem principu. Ukoliko je korisnik kliknuo mišem uzimamo poziciju klika miša na trenutnoj sceni. Kada korisnik drži klik miša i prevlači mišem po površini programa obradjuje se onMouseDragged metoda koja uzima offset tako što oduzima trenutnu poziciju miša sa pozicijom prethodno upamćenih koordinata istog. Nakon računanja pomeraja, renderPane-u se podešava pozicija (korišćenjem metode setTranslate) tako što uzimamo trenutnu poziciju pane-a (getTranslate) i dodavanjem vrednosti prethodno izračunatog pomeraja. Kako bi sve ovo funkcionisalo potrebno je na samom kraju skladištiti trenutnu poziciju miša za sledeću iteraciju.

**

1. *Akcije miša*

Pri samom kraju potrebno je definisati stop metodu javaFX programa u kojoj zaustavljamo sve trenutno aktivne tredove(niti), gasimo platformu kao i kompletan program.

Koristimo podrazumevani ulaz aplikacije main kako bismo pozvali pokretanje javaFX programa.



1. *Main*

# Zaključak

Program koji je izrađen za projekat zadovoljava osnovne zahteve zadatka. Tokom izrade zadatka utvrdio sam prethodno stečeno znanje. Poboljšanje programa bi bilo u samom UI-u kao i u dodatnim funkcionalnostima na sam zahtev klijenata.

# Reference

* <http://lams.metropolitan.ac.rs/>
* Book: Java Programming: Comprehensive Concepts and Techniques by Herbert Schildt
* [Drawing a tree using recursion](https://stackoverflow.com/questions/30016085/drawing-a-tree-using-recursion)
* [Recursion Trees and the Master Method](https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/lec20.html)
* [CSE120 Sp17 Lab 11 - Recursive Tree](https://courses.cs.washington.edu/courses/cse120/17sp/labs/11/tree.html)
* [Recursion Trees](https://lodev.org/cgtutor/recursiontrees.html)