Elektrotehnički fakultet u Beogradu Katedra za računarsku tehniku i informatiku

*Predmet:* Programski prevodioci 1

*Student:* Sonja Grubor

*Broj indeksa:* 2018/0260

*Školska:* 2021/2022.

*Datum:* 19.08.2021.

Kompajler za Mikrojavu

**– Propratna dokumentacija –**

# Kratak opis projektnog zadatka

Cilj projektnog zadatka je realizacija kompajlera za programski jezik Mikrojavu. Kompajler omogućava prevodjenje sintaksno i semantički ispravnih Mikrojava programa u Mikrojava bajtkod koji se izvršava na virtuelnoj mašini za Mikrojavu. Sintaksno i semantički ispravni Mikrojava programi su definisani specifikacijom [MJ].

Programski prevodilac za Mikrojavu ima četiri osnovne funkcionalnosti: leksičku analizu, sintaksnu analizu, semantičku analizu i generisanje koda.

# Opis komandi za generisanje Java koda

* Specifikacija leksičkog analizatora smeštena je u fajl MJCompiler/src/spec/mjlexer.lex.
* Specifikacija .lex se transformiše u implementaciju leksera na programskom jeziku Java korišćenjem alata **JFlex sa sajta predmeta.**
* Generisana klasa leksičkog analizatora pripada paketu rs.ac.bg.etf.pp1 u okviru direktorijuma MJCompiler /src.
* Za implementaciju parsera koristi se generator sintaksnih analizatora AST-CUP. AST-CUP generator je lokalno razvijeno proširenje alata CUP za rad sa sintaksnim stablima.
* AST-CUP specifikacija smeštena je u fajl MJCompiler/src/spec/mjparser.cup
* U okviru semantičke analize i faze generisanja koda korišćena je biblioteka symtable.
* Fajl Build.xml korišćen je za jednostavnije povezivanje biblioteka i fajlova projekta kao i za odvajanje odredjenih korisnih funkcionalnosti biblioteka
* Target lexergen koristi se za generisanje Yylex.java klase uy pomoć JFlex alata
* Target repackage se koristi za pozivanje pozivanje parsera i generisanje klasa za svaki neterminal specifiran u .cup fajlu projekta
* Target compile kompajlira celokupan projekat
* Da bi se generisao objektni fajl nad odredjenim testom potrebno je pokrenuti klasu Compiler.java
* Target dissasm se poziva kako bi se video disasemblovani generisani objektni fajl odnosno kod koji je generisan ali je čitljiv korisniku
* Target runObj koristi se kako bi se pokrenulo izvršavanje generisanog koda

# Kratak opis priloženih test primera

## test301.mj

U nastavku teksta su navedeni i opisani zahtevi projektnog zadatka čiju proveru pokriva navedeni test:

* Deklaracija globalnih promenljivih i konstanti
* Korišćenje primitivnih tipova i niyova čiji elementi su nekog primitivnig tipa
* Definicija glavne funkcije od čijeg početka kreće izvršavanje programa
* Deklaracija i korišćenje lokalnih promenljivih unutar main funkcije
* Pristup elementu niza
* Aritmetičke operacije i operacija dodele vrednosti
* Izračunavanje kompleksnih aritmetičkih izraza
* Upotreba sistemskih funkcija print i read

## test302.mj

U nastavku teksta su navedeni i opisani zahtevi projektnog zadatka čiju proveru pokriva navedeni koji ujedino pokriva i sve navedene stavke koje pokriva test301.mj test:

* Definicija više programskih funkcija koje mogu da imaju podrazumevane argumente
* Poziv definisanih funkcija programa i varijacija u navodjenju neobaveznih argumenata kod poziva funkcija
* If-else grananje
* Provera uslova i generisanje skokova u yavisnosti od tačnosti uslova
* Do-while petlja
* Naredba break koja stopira izvršavanje okružujuće petlje
* Naredba continue koja Skače na početak tela koružujuće petlje

# Opis novouvedenih klasa

* *Address.java* – Klasa kojom se predstavlja struktura koja se koristi kod prepravljanja adresa pri skokovima unapred. Sadrži polja adr koje pamti vrednost adrese na kojoj treba da se izvrši prepravka i polje level koje predstavlja nivo ugneždavanja if grananja ili while pelje u zavisnosti o kojoj prepravci je reč.