

MJ\_Compajler

A nivo



January 12, 2020

Novosel ivana

0202/2015

*Kratak opis 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.  
Leksički analizator treba da prepoznaje jezičke lekseme i vrati skup tokena izdvojenih iz  
izvornog koda, koji se dalje razmatraju u okviru sintaksne analize. Ukoliko se tokom leksičke  
analize detektuje leksička greška, potrebno je ispisati odgovarajuću poruku na izlaz.  
Sintaksni analizator ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa  
mogu formiraju gramatički ispravne sentence. Tokom parsiranja Mikrojava programa potrebno je  
na odgovarajući način omogućiti i praćenje samog procesa parsiranja na način koji će biti u  
nastavku dokumenta detaljno opisan. Nakon parsiranja sintaksno ispravnih Mikrojava programa  
potrebno je obavestiti korisnika o uspešnosti parsiranja. Ukoliko izvorni kod ima sintaksne greške,  
potrebno je izdati adekvatno objašnjenje o detektovanoj sintaksnoj grešci, izvršiti oporavak i  
nastaviti parsiranje.  
Semantički analizator se formira na osnovu apstraktnog sintaksnog stabla koje je nastalo kao  
rezultat sintaksne analize. Semantička analiza se sprovodi implementacijom metoda za posećivanje  
čvorova apstraktnog sintaksnog stabla. Stablo je formirano na osnovu gramatike implementirane u  
prethodnoj fazi. Ukoliko izvorni kod ima semantičke greške, potrebno je prikazati adekvatnu  
poruku o detektovanoj semantičkoj grešci.  
Generator koda prevodi sintaksno i semantički ispravne programe u izvršni oblik za  
odabrano izvršno okruženje Mikrojava VM. Generisanje koda se implementira na sličan način kao i  
semantička analiza, implementacijom metoda koje posećuju čvorove.

**NIVO A –** potrebno je implementirati generisanje koda za SVE gramatičke smene u nastavku  
(osnovni iskazi, aritmetički izrazi, rad sa nizovima prostih tipova i rad sa funkcijama bez  
parametara):  
Statement := DesignatorStatement ";".  
DesignatorStatement := Designator "=" Expr.  
DesignatorStatement := Designator "++".  
DesignatorStatement := Designator "--".  
Statement := DesignatorStatement ";".  
Statement := "read" "(" Designator ")" ";".  
Statement := "print" "(" Expr [“,” number] ")" ";".  
Expr := ["‐"] Term {Addop Term}.  
Term := Factor {Mulop Factor}.  
Factor := numConst | charConst | "(" Expr ")" | boolConst | "new" Type [ "[" Expr "]" ].  
Factor := Designator [ "(" ")" ].  
Designator := ident ["[" Expr "]" ].  
Addop := "+" | "‐" .  
Mulop := "\*" | "/" | "%"

*Pokretanje programa*

Odabirom iz padajuceg menija biramo sta zelimo da pokrenemo:

-LexerGenerator

-ParserGenerator

-Compile MJCource

- Run za iszvrsavanje programa napisamog I program.mj

Uputstvo: preuzeti template sa sajta predmeta,

-zameniti lib: za tabelu simbola

- zamentiti fajlove za lekser I parser

- zameniti SemanticPass sa SemanticAnalizer

-zameniti CodeGenerator

-zameniti MJParserTest

Ove fajloe iz temlatea zameniti sa odgovarajuci mi predatog projekta.

*Testovi*

Preuzeti sa sajta predmeta, test 301 za A nivo, testira ispravno generisanje koda,

Pristup nizu, sabiranje, mnozenje, I ispis elemenata na standardnom izlazu.

I manji testovi , koji su prilozeni u direktorijumu testovi.