|  |  |
| --- | --- |
|  | **ELEKTROTEHNIČKI FAKULTET**  **UNIVERZITETA U BEOGRADU**  Bulevar kralja Aleksandra 73, PF 35-54, 11120 Beograd, Srbija  Tel 011/3218-436, Faks 011/3218-433, Račun 840-1438666-48 |

Kompajler za Mikrojavu

Programski prevodioci 1

Student:

Uroš Jokić 2017/0517

Beograd, septembar

## 1. Postavka 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.

*Generator parsera prihvata .cup fajl u kome je specificirana gramatika jezika. Tokom izvršavanja koda parsera, dobijamo informacije o svim sintaksno neispravnim programima, uspješno se parsiraju samo potpuno ispravni.*

* 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.

*Semantički analizator je realizovan kroz klasu SemanticAnalyzer. Pokretanjem dobijamo informacije o eventualnim semantičkim greškama u programu.*

* 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.

*Semantički analizator je realizovan kroz klasu CodeGenerator. Pokretanjem dobijamo objektni fajl program.obj čijim disasembliranjem možemo dobiti generisani kod za Mikrojava VM.*

*Pokretanjem semantički ispravnog programa, dobijamo ispis na konzoli output-a programa.*

## 2. Opis komandi za pokretanje rešenja

2.1 Pokretanje iz Comandne linije

* GENERISANJE LEKSERA

java -jar lib/JFlex.jar -d ./src/rs/ac/bg/etf/pp1 ./spec/mjlexer.lex

* GENERISANJE PARSERA

cd src

java -jar ../lib/cup\_v10k.jar -destdir ./rs/ac/bg/etf/pp1 -ast rs.ac.bg.etf.pp1.ast -parser Parser -buildtree ../spec/mjparser.cup

cd ..

* KOMPAJLIRANJE

javac -cp .:lib/\* ./src/rs/ac/bg/etf/pp1/\*.java ./src/rs/ac/bg/etf/pp1/ast/\*.java ./src/rs/ac/bg/pp1/util/\*.java

* PREVODJENJE MIKROJAVA KODA

java -cp .:bin:lib/\* rs.ac.bg.etf.pp1.Compiler test/ulaz.mj test/izlaz.obj

* IZVRSAVANJE MIKROJAVA PROGRAMA

java -cp ./lib/mj-runtime.jar rs.etf.pp1.mj.runtime.Run ./test/program.obj

2.2 Pokretanje iz programskog okruzenja Eclipse

Prije pokretanja iz programskog okruzenja eclipse, potrebno je instalirati *ant* alat I podesiti Path sistemsku varijablu da pokazuje na bin folder instalacije *ant* alata.

Pri pokretanju okruzenja pozicioniramo Workspace alata nan as workspace I ucitamo projekat. Projekat se pokrece za sve faze osim izvrsavanja programa preko

*Desni klik na Compile.java -> Run as -> Java aplication*

*Desni klik na build.xml -> Run as -> Ant Build -> cekiranjem odgovarajuce konfiguracije*

Za pokretanje izvrsavanja mikrojava programa potrebno je dodati novu konfiguraciju

*Desni klik na Compile.java -> Run as -> Run Configurations -> New launch configuration ->*

*Project: MJCompiler*

*Main class: rs.etf.pp1.mj.runtime.Run*

A zatim pomocu *Desni klik na Compile.java -> Run as -> Run Configurations -> Naziv nase prethodno kreirane konfiguracije*