Elektrotehnički fakultet u Beogradu

Katedra za računarsku tehniku i informatiku

*Predmet:* Programski prevodioci 1

*Nastavnik:*  dr Dragan Bojić, vanr. prof.

*Asistenti:* Maja Vukasović, mast. inž.

Kristijan Žiža, mast. inž.

*Školska:* 2019/2020.

*Ispitni rok:* Junsko-julski ispitni rok

**Projekat**

**– Kompajler za Mikrojavu –**

*Student:* Magdalena Stanković 0065/2016

*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.

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
* *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.
* *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.

*Komande za generisanje Java koda:*

* Biblioteka lib/Jflex.jar postaviti argumente -d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex (dobija se Yylex.java)
* Biblioteka lib/cup\_v10k.jar postaviti argumente -destdir rs\ac\bg\etf\pp1

-ast rs.ac.bg.etf.pp1.ast -parser MJParser -buildtree ..\spec\mjparser.cup (dobija se MjParser.java i sym.java)

* Disasm: u rs.etf.pp1.mj.runtime.disasm postaviti argumente test\program.obj
* Debug: u rs.etf.pp1.mj.runtime.disasm postaviti argumente -debug test\program.obj
* Run: u rs.etf.pp1.mj.runtime.disasm postaviti argumente test\program.obj

Test primeri – testiraju pojedinačne delove projektnog zadatka.