Elektrotehnički fakulet – Beograd

**Programski prevodioci 1**

Kompajler za Mikrojavu

Nastavnici: Student:

Prof. dr. Dragan Bojić Stefan Krivokapić

As. Maja Vukasović 2015/0312

As. Kristijan Žiža

Postavka zadatka

Cilj projektnog zadatka jeste realizacija kompajlera za programski jezik Mikrojava. Kompajler omogućava prevodjenje sintaksno i semantički ispravnih Mikrojava programa u Mikrojava bajtkod koji se izvršava na virtuelnoj mašini za Mikrojavu. 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, ispisuje se odgovarajuća poruka na glavni izlaz.

**Sintaksni analizator**ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu formiraju gramatički ispravne rečemoce (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 obavijestiti korisnika o uspješnosti 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, semantički analizator ispisuje odgovarajuću poruku na glavni izlaz.

**Generator koda** prevodi sintaksno i semantički ispravne programe u izvršni oblik za odabrano izvršno okruženje Mikrojava VM. Generisanje koda se vrši obilaženjem sintaksnog stable slično kao u semantičkom analizatoru, pa se samim tim implementacija svodi, kao i kod semantičke analize, na implementiranje metoda visit koje obrađuju odgovarajuće čvorove stable.

Opis rešenja

Leksičke provjere su napisane u fajlu: mj***lexer.flex****.*

Gramatika jezika je u fajlu: mj***parser.cup****.*

Testovi se nalaze u folderu ***test****.*

Korišćene konfiguracije:

**LexerGenerator** - main klasa:JFlex.Main; -d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex

* java -jar lib\JFlex.jar -d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex

**ParserGenerator** - main klasa: java\_cup.Main ; -destdir rs\ac\bg\etf\pp1 -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup

* java -jar lib\cup\_v10k.jar -destdir src\rs\ac\bg\etf\pp1 -parser MJParser spec\mjparser.cup

**MJTest -** main klasa: rs.ac.bg.etf.pp1.ParserTest

* java -cp .;bin;test;lib\log4j-1.2.17.jar;lib\JFlex.jar rs.ac.bg.etf.pp1.MJTest

**Compiler** - main klasa: rs.ac.bg.etf.pp1. Compiler; test\program.mj test\program.obj

* java -cp .;bin;tests;lib\cup\_v10k.jar;lib\log4j-1.2.17.jar;lib\symboltable.jar;lib\mj-runtime.jar rs.ac.bg.etf.pp1.Compiler test\program.mj test\program.obj

**Disasm** - main klasa: rs.etf.pp1.mj.runtime.disasm; test\program.obj

* java -cp lib\mj-runtime.jar rs.etf.pp1.mj.runtime.disasm test\program.ob

**Run** - main klasa: rs.etf.pp1.mj.runtime.Run; -debug test\program.obj

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

Test primeri

***Test1.mj*** – Sintaksno i semantički ispravan kod koji testira nivo A funkcionalnosti sa fokusom na nizove, inkrementiranje i operaciju sabirajuće dodjele

***Test2.mj*** – Sintaksno i semantički ispravan kod koji testira nivo A i funkcije funkcionalnosti

***Test3.mj*** – Sintaksno neispravan kod koji prikazuje oporavak od grešaka za nivo A

***Test301.mj*** – javni test za nivo A