Elektrotehnički fakulet u Beogradu

Katedra za računarsku tehniku i informatiku

Avgust 2019. godine

**Kompajler za Mikrojavu**

Programski prevodioci 1

Nastavnici: Student:

Prof. dr. Dragan Bojić Dodov Nikola

As. Maja Vukasović 2015/0395

As. Kristijan Žiža

Postavka zadatka

Cilj projektnog zadatka iz predmeta Programski Prevodioci 1, na odseku za Računarsku tehniku i informatiku, je 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, 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.

Alati korišćeni prilikom izrade projekta

* ***lexer.flex*** – fajl koji vrši leksičke provere;
* ***parser.cup***– fajl koji vrši semantiške provere i parseriranje;
* ***build.xml*** – skripta za generisanje leksera, parsera, generatora koda;

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

Testovi

***TestB1.mj*** – ispravan kod koji testira funkcionalnosti za nivo B

***TestERROR.mj*** – sintaksno i semantičko neispravan kod koji prikazuje oporavak od grešaka

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

***Test302.mj*** – javni test za nivo B