ELEKTROTEHNIČKI FAKULTET BEOGRAD

**Programski prevodioci 1**

MJ KOMPAJLER

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.

Opis rešenja

Leksičke provere su napisane u fajlu:  ***mjlexer.flex****.*

Implementacija parsera je u fajlu: ***mjparser.cup****.*

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

Korišćene konfiguracije:

**LexerGenerator** - main klasa:JFlex.Main; -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

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

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

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

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

**Pokretanje :**

Odgovarajuća putanja do fajla sa izvornim mikrojava kodom i putanja do izlaznog objektnog fajla zadaju se putem argumenata komandne linije respektivno.  
Potrebno je pozicionirati se u folder MJCompiler gde se nalazi projekat na računaru. U komandnoj liniji je potrebno uneti sledece:

java -cp bin;lib\\* rs.ac.bg.etf.pp1.Compiler test\program.mj test\program.obj > test\izlaz.out 2> test\izlaz.err

Test primeri

***test301.mj:*** *javni test za nivo A*

***ankinTest2.mj:*** *moj test*

***program\_erroneous:*** *moj test sa namernim semantickim greskama*