Univerzitet u Beogradu, Elektrotehnički fakultet

**MikroJava kompajler**

*MicroJava compiler*

Projekat iz predmeta Programski prevodioci

Student

Andrej Gajić 2018/0303

**Opis projektnog zadatka**

Cilj ovog projekta je realizacija funkcionalnog kompajlera za MikroJavu, upošćenog programskog jezika koji je nastao na Elektrotehničkom fakultetu po ugledu na Javu. Kompajler prevodi ulazni MikroJava fajl (ekstenzija .mj) u izvršivi .obj fajl.

Projekat se sastoji iz 4 faze:

1. Leksička analiza – očitavanje svih simbola koji se pojavljuju u gramatici jezika i njihovo grupisanje u tokene

2. Sintaksna analiza – provera sintakse ulaznog koda koristeći LALR(1) parser. Tokeni koje je izgenerisao leksički analizator se koriste u ovoj fazi da bi se proverilo da li oni formiraju ispravnu gramatiku (da li odgovaraju gramatičkim smenama).

3. Semantička analiza – provera semantike ulaznog koda i građenje tabele simbola pomoću sintaksnog stabla koje je nastalo kao proizvod sintaksne analize. Ova faza se realizuje implementacijom visit() metode za svaki krucijalni čvor u sintaksnom stablu.

4. Generisanje koda – ukoliko je ulazni kod uspešno prošao kroz sintaksnu i semantičku proveru, za njega će biti izgenerisan izvršivi MikroJava bajtkod koristeći MikroJava VM instrukcije.

**Opis komandi**

Osnovni fajl koji se koristi za pokretanje delova realizovanog kompajlera je Ant skripta build.xml.

Moguće komande:

1. lexerGen – pokreće leksički analizator.

2. delete – briše fajlove koji su izgenerisani od strane leksičkog i sintaksnog analizatora.

3. parserGen – pokreće sintaksni analizator.

4. repackage (zavisi od parserGen) – menja nazive paketa izgenerisanih klasa sintaksnog stabla (briše src iz imena paketa).

5. compile (zavisi od repackage) – pokreće sintaksni analizator i radi repackage funkciju.

6. disasm – prevodi .obj MikroJava fajl u čitljivi format MikroJava instrukcija.

7. runObj – pokreće prevedeni kod.

Kompajliranje ulaznog koda se izvršava pozivom main metode Java klase rs.ac.bg.etf.pp1.Compiler, koja zahteva 2 parametra kao argumente komandne linije. Prvi parametar predstavlja putanju do ulaznog .mj fajla, a drugi putanju do izlaznog .obj fajla.

Takođe, prevedni kod se može pokrenuti pozivom main metode Java klase rs.etf.pp1.mj.runtime.Run, i mora joj se dostaviti putanja do .obj fajla koji treba da izvrši.

**Opis novouvedenih klasa**

1. SemanticAnalyser – Java klasa u kojoj se nalazi logika semantičke analize. U ovoj klasi su nadjačane visit() metode svih relevantnih čvorova sintaksnog stabla. U tim metodama se formira tabela simbola na osnovu deklarisanih objekata (konstanti, varijabli, funckija), a takođe se proveravaju i razni semantički uslovi navedeni u tekstu projektnog zadatka.

2. CodeGenerator – Java klasa u kojoj se nalazi logika generisanja koda. Kao i u gore navedenoj klasi, nadjačavaju se visit() metode relevantnih čvorova sintaksnog stabla i generiše se MirkoJava bajtkod.

3. MyTab – ekstenzija klase rs.etf.pp1.symboltable.Tab. U ovoj klasi se uvodi novi tip promenljive – boolean, i takođe se realizuje metoda za formatirani ispis objektnog čvora iz tabele simbola.

4. CounterVisitor – pomoćna Java klasa koja pomaže pri generisanju koda. Sadrži dve statičke klase koje su zadužene za brojanje parametara funkcija radi lakšeg generisanja koda prilikom njene deklaracije.

5. Compiler – glavna Java klasa koja povezuje sve faze projektnog zadatka. Prvenstveno se pokreće leksički analizator, zatim sintaksni, zatim semantički i na kraju kod generator. Ukoliko se u nekoj od faza desi neka greška, ona se ispisuje na standardni izlaz grešaka i program se prekida (ne nastavlja se u naredne faze).

**Opis testova**

1. test1.mj – testira osnovne funkcionalnosti projekta (učitavanje konstanti, globalnih i lokalnih promenljivih, aritmetičke operacije, inkrementiranje, dekrementiranje, ispis, rad sa nizovima).

2. test1err.mj – izbacuje sintaksnu grešku pri aritmetičkim operacijama.

3. test2.mj – testira operator ??

4. test2err.mj – izbacuje semantičku grešku (jedna od vrednosti operatora ?? je char).

5. test3.mj – testira pozive funkcija (sa podrazumevanim parametrima).

6. test3err.mj – izbacuje semantičku grešku prilikom poziva funkcije (nedovoljno parametara).