**Elektrotehnički fakultet u Beogradu**

*Predmet:* Programski prevodioci 1

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

*Asistenti:* dipl. ing. Nemanja Kojić, dipl. ing. Maja Vukasović

*Školska:* 2017/2018.

**Projekat**

**Kompajler za Mikrojavu**

*Autor:* Marija Maričić 0457/2014

**Opis projekta**

* Projektom je realizovan kompajler za prilagodjenu Mikrojavu.
* Kompajler se sastoji od 4 osnovna dela: leksicka analiza, sintaksna analiza, semanticka naaliza i generisanje koda
* Leksicka analiza sluzi da se prepoznaju jezicke lekseme i vrati skup tokena. Koricen flex.
* Sintaksna analiza proverava da li tokeni odgovaraju gramatici jezika. Realizovan oporavak od nekih gresaka. Koriscen cup. Na kraju generise apstraktno sintaksno stablo koje odgovara programu.
* Semanticka analiza obilazi izgenerisano apstraktno sintaksno stablo metodom BottomUp i proverava semantiku.
* Generisanje koda se izvrsava ukoliko nije bilo gresaka u prethodnim fazama. On generise izvrsni obj kod

**Komande za generisanje koda**

* Lekser Generator:

Klasa JFlex.Main: *-d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex*

* Parer Generator:

java\_cup.Main: *-destdir rs\ac\bg\etf\pp1 -dump\_states -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup*

* Prevodjenje testa (Semanticka analiza i generisanje koda):

Klasa Compiler, argumenti: *test/naziv\_testa.mj naziv\_testa.obj*

* Izvrsavanje izgenerisanog objektnog koda:

Klasa Run, argumenti*: [-debug]* *naziv\_testa.obj*

(debug opcija se ubacuje ukoliko zelite debug Run)

* Disasm (prikazuje izgenerisan obj fajl):

Klasa Disasm: *naziv\_testa.obj*

**Test primeri**

Svaki test primer testira ono sto mu i samo ime kaze. Neispravni testovi su fokusirani na prikazivanje gresaka i nacin na koji se kompajler nosi sa njima. Ispravni testovi generisu izvrsni obj fajl. Test „sintaksnaOporavak“ prikazuje kako se parser oporavlja od sintaksnih gresaka i onda u semantickoj analizi ne detektuje nijednu gresku.

**Novouvedene klase**

* **Compiler** – klasa koja predstavlja kompajler – ona kreira i pokrece neophodne klase i metode kako bi se izvrsile sve 4 faze kompajlera.
* **SemanticAnalyzer** – Visitor koji obilazi apstraktno sintaksno stablo i vrsi semanticku analizu.
* **CodeGenerator –** Visitor namenjen obilasku sintaksnog stabla radi generisanja koda.
* **CustomTableDumpVisitor –** Sluzi za obilazak i ispis tabele simbola – Override-ovan da bi podrzao tip Bool i sprecio rekurziju kod klasa
* **VirtualTableGenerator –** Pomocna klasa koja se koristi za pravljenje tabele virtuelnih funkcija.