**Elektrotehnički fakultet u Beogradu**

*Predmet:* Programski prevodioci 1

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

*Asistenti:* mast. inž. Maja Vukasović, mast. inž. Kristijan Žiža

*Školska:* 2019/2020.

**Projekat**

**Kompajler za Mikrojavu**

*Zorana Janković 2016/0143*

**Opis projekta**

Projektom je realizovan kompajler za prilagođenu Mikrojavu. Kompajler se sastoji od 4 osnovna dela:

* Leksički analizator treba da prepoznaje jezike lekseme i vrati skup tokena izdvojenih iz izvornog koda koji se dalje razmatraju u okviru sintaksne analize.
* Sintaksni analizator treba da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu da formiraju gramatički ispravne sentence.
* Semantički analizator se formira na osnovu apstraktnog sintaksnog stabla koje je nastalo kao rezultat sintaksne analize. Treba da detektuje semantičke greške.
* Generator koda prevodi sintaksno i semantičke ispravne programe u izvršni oblik. Kao rezultat se generiše .obj fajl.

**Komande za generisanje koda**

* Lekser Generator:

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

* Parser 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*

* Pokretanje prevodioca (semantička analiza i generisanje koda):

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

* Izvršavanje generisanog objektnog koda:

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

* Disasm (prikazuje izgenerisan obj fajl):

Klasa Disasm: *test/naziv\_testa.obj*

* Sve prethodne stavke osim pokretanja prevodioca je moguće uraditi preko ant-builda, pokretanjem fajla build.xmli selektovanjem željene operacije

**Test primeri**

Postoji više testova za svaku od faza, koji testiraju razne primere i moguće kombinacije date gramatike i generisanja koda. Neispravni testovi su fokusirani na prikazivanje grešaka i način na koji se kompajler nosi sa njima. Ispravni testovi generišu izvršni .obj fajl.

**Novouvedene klase**

* **MJParserTest** – klasa koja predstavlja kompajler – pokreće neophodne klase i metode kako bi se izvršile sve 4 faze kompajlera.
* **SemanticPass** – Visitor koji obilazi apstraktno sintaksno stablo i vrši semantičku analizu.
* **CodeGenerator –** koji obilazi apstraktno sintaksno stablo i vrši generisanje koda.
* **CounterVisitor–** Služi za obilazak formalnih paramatera i lokalnih promenljivih funkcije i njihovo brojanje
* **DesignatorVisitor**– Služi za obilazak podstabla designatora i generisanje koda za potrebne čvorove.