Elektrotehnički fakultet u Beogradu

Katedra za računarsku tehniku i informatiku

*Predmet:* Programski prevodioci 1

*Školska: 2018/2019.*

**Projekat**

**– Kompajler za Mikrojavu –**

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

*Asistenti:* dipl. Ing. Kristijan Ziza, dipl. ing. Maja Vukasović

*Student:* Tamara Stevanovic 2014/0255

**Opis Projekta**

Cilj projektnog zadatka je realizacija kompajlera za programski jezik Mikrojavu koji omogućava prevodjenje sintaksno i semantički ispravnih Mikrojava programa u Mikrojava bajtkod. Dobijeni bajtkod se zatim izvršava pomoću (na) virtuelnoj mašini za Mikrojavu.

Programski prevodilac za Mikrojavu se sastoji iz četiri faze: leksičku analizu, sintaksnu analizu, semantičku analizu i generisanje koda.

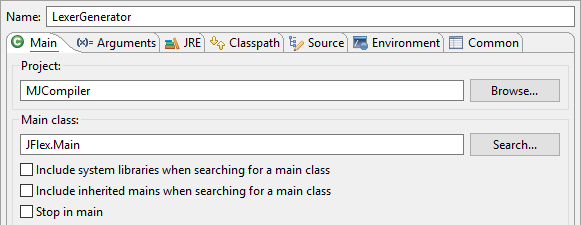
Leksički analizator prepoznaje jezičke lekseme I vraća skup tokena izdvojenih ih izvornog koda. Sintaksni analizator utvrđuje da li izdvojeni tokeni mogu formirati gramatički ispravne rečenice.

Izlaz sintaksne analize predstavlja apstraktno stablo koje je ulaz za semantičku analizu I na osnovu koga se formira semantički analizator. Posećivanjem čvorova apstraktnog sintaksnog stable metodama za posećivanje čvorova semantički analizator provera da li je došlo do semantičke greške.

Generator koda prevodi sintaksno I semantički ispravne program u izvršni kod za Mikrojava virtuelnu mašinu.

**Opis komandi**

Konfiguracija za generisanje lexera:



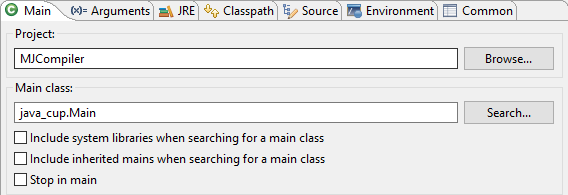
**Argumenti**: -d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex

Working directory: Default

Koristeći biblioteku JFlex.jar, mjlexer.flex fajl prevodimo u Yylex.java.

Argumenti su –d, putanja ka loakciji gde želimo da napravimo Yylex.java I .flex fajl.

Konfiguracija za generisanje parsera:

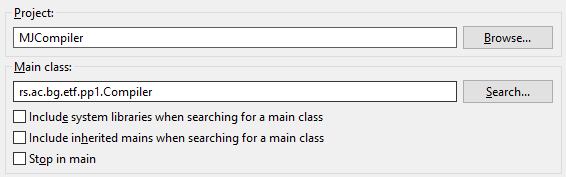


**Argumenti**: -destdir rs\ac\bg\etf\pp1 -dump\_states -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup

Working directory: Other: ${workspace\_loc:MJCompiler\src}

Koristeći biblioteku cup\_v10k.jar, mjparser.cup fajl prevodimo u sym.java I MJParser.java fajlove.

Argumenti su: –destdir (putanja ka lokaciji gde želimo da nam se stvore sym I MJParser fajlovi), -dump\_states (ispisuje stanja automata), -parser MJParser (ime parser fajla),-ast , putanja ka lokaciji gde želimo generisati niz klasa za reprezentaciju sintaksnog stabla parsera, vizitora i vizitor adaptera, -buildtree pravi apstraktno sintaksno stablo, I .cup fajl koji se koristi za prevođenje.



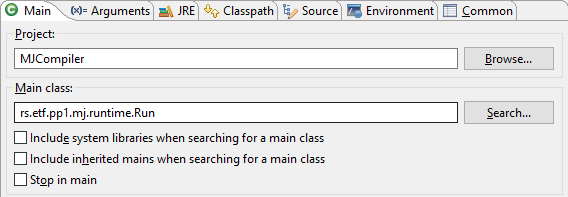
Pomoću konfiguracije sa prethodne slike je mogu da se pokrene izvršavanje Compile klase.

**Argumenti**: test\programJavni.mj test\program.obj

Working directory: Default

Prvi argument je izvorni fajl sa programom, a drugi je izlazni obj fajl koji predstavlja bajtkod za MikroJava virtuelnu mašinu.

Pokretanje Mikrojava virtuelne mašine sa bajtkodom:

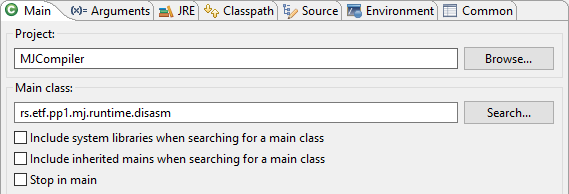


**Argumenti**: -debug test\program.obj

Working directory: Default

-debug nam omogućava da vidimo svaki komandu koja se izvršava I stanje expr steka posle svake instrukcije, a drugi argument je lokacija .obj fajla

Pokretanje Disasemblera:



**Argumenti**: test\program.obj

Working directory: Default

Pokrećemo disasembler – prikazaće nam codesize, datasize, mainPC I sve instrukcije koje će se izvršiti, čak iako dođe do errora.

Argument je izvorni obj fajl.

**Opis test primera**

Uz rešenje su priložena četiri testa. Testirano je sledeće:

1. Javni test za nivo AOporavak od greške
2. Test koji proverava sve funkcionalnosti i smene iz sintaksne analize
3. Program koji vrši oporavke od grešaka načinjenih u sintaksnoj analizi (navedeni su svi primeri za koje je u tekstu projekta nevedeno da mora da postoje oporavci)
4. Detektovanje grešaka prilikom semantičke analize

**Opis novih klasa**

MJTest.java – nalazi se u test direktorijumu u paketu rs.ac.bg.etf.pp1 – služi za proveru leksičke analize.

CounterVisitor – služi za prebrojavanje lokalnih promenljivih u okviru neke metode.

**Opis alternativnog načina pokretanja**

Zarad lakšeg pokretanja klasa pri promenama projekta, napravljena je I ***build.xml*** skripta u kojoj se pomoću različitih target-a pokreću tražene klase.

Izgled fajla:



