**ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU**

Programski prevodioci 1

Kompajler za mikrojavu

Student:

Filip Carević 0065/2017

# Opis zadatka

“Cilj projektnog zadatka je realizacija kompajlera za programski jezik Mikrojavu. Kompajler omogućava prevodjenje sintaksno i semantički ispravnih Mikrojava programa u Mikrojava bajtkod koji se izvršava na virtuelnoj mašini za Mikrojavu. Sintaksno i semantički ispravni Mikrojava programi su definisani specifikacijom [MJ]. Programski prevodilac za Mikrojavu ima četiri osnovne funkcionalnosti: leksičku analizu, sintaksnu analizu, semantičku analizu i generisanje koda.” [Projkat§Uvod.p1]

Prilikom izrade porjekta korišćeni su alati za generisanje leksera (JFlex) I parsera (Cup). Lekser se konstruiše na osnovu specifikacije zadate fajlom “mjlexer.lex”. Parser se konstruiše na osnovu specifikacije zadate fajlom “mjparser.cup”. Proizvod leksičke analize predstavljaju tokeni, njihove vrednosti I klase. Parser prilikom svog rada koristi izgenerisani lekser, I na osnovu prepoznatih toekna konstruiše apstraktno sintaksno stablo. Postorder obilaskom istog stabla, semantički analizator (klasa “SemanticPassVisitor”) vrši semantičke provere specificirane u fajlu [MJ]. U sličaju da je program sintaksno I semantički ispravan, prelazi se u fazu generisanja koda. Ponovnim postorder obilaskom gore pomenutog stable, generator koda (klasa “CodeGenerator”) generiše bajtkod.

# Komande za pokretanje

Neophodno je izvršiti pozicioniranje u src folder.

* generisanje lexera:

java -cp ../lib/JFlex.jar JFlex.Main -d rs\ac\bg\etf\pp1 ..\spec\mjlexer.flex

## generisanje parsera

java -cp ../lib/cup\_v10k.jar java\_cup.Main -destdir rs\ac\bg\etf\pp1 -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup

## prevođenje projekta

javac -cp ..\lib\\* rs\ac\bg\etf\pp1\ast\\*.java rs\ac\bg\etf\pp1\util\\*.java rs\ac\bg\etf\pp1\\*.java

## prevođenje .mj fajla

java -cp ..\lib\\* rs.ac.bg.etf.pp1.MJParserTest mjfilesrc objfiledst

## disassemble

java -cp ..\lib\mj-runtime.jar rs.etf.pp1.mj.runtime.disasm objfilesrc

## run/debug

java -cp ..\lib\mj-runtime.jar rs.etf.pp1.mj.runtime.Run [-debug] objfilesrc

# Ilustracija test primera

* nadovezivanje \_klase\_nizovi.mj – testira greške prilikom ulančavanja pristupa polja objekta u okviru nizova objekata i obratno
* oporavak\_gresaka.mj – ilustruje oporavak parsera prilikom sintaksne analize za sva 3 nivoa
* sintaksno\_ispravan\_kod.mj – ilustruje ispravljen kod testa oporavak\_gresaka.mj
* semantikaA.mj - ilustruje semantički ispravan kod nivoa A
* switch.mj – iliustruje isparavan rad preilikom prevođenja ugnjezdenih switch naredbi
* greskeSintaksa1.mj – ilustruje detektovanje neispravne semantike za :
  + Deklaraciju kontanti
  + Deklaraciju extends klauzule klase
  + Metode bez return iskaza i nekompatibilnonst tipova povratne vrednosti i return Expr-a
  + Nedefinisani tip
  + Nevalidnu dodelu vrednosti
  + Nevalidno inc/dec
  + Nevalidne pozive fja
  + Break/Continue izvan DoWhile i Switch
  + Nevalidan tip parametra za read/print
* greskeSemantika2.mj – ilustruje detektovanje neispravne semantike za :
  + If/ DoWhile uslov
  + Switch promenljivu
  + Nevalidni parametri pri pozivu fje
  + Ternarni operator, kompatibilnost tipova
  + New, indeksiranje i pristup polju

# Opis novih klasa

* SemanticPassVisitor – vrši semantičku analizu i konstrukciju tabele simbola (nasleđuje VisitorAdapter)
* SymbolTablePrinter – vrši ispis tabele simbola (nasleđuje SymbolTableVisitor)
* CodeGenerator – vrši generisanje koda (nasleđuje VisitorAdapter)
* MyStruct – klasa uvedena radi redefinicije metode isAssignableTo iz njene osnovne klase Struct (nasleđuje Struct)