**Programski prevodioci 1**

Kompajler za Mikrojavu

*Marko Mićović*

*2016/0633*

**Opis Projekta**

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.

Programski prevodilac za Mikrojavu ima četiri osnovne funkcionalnosti: leksičku analizu, sintaksnu analizu, semantičku analizu i generisanje koda.

**Leksički analizator** treba da prepoznaje jezičke lekseme i vrati skup tokena izdvojenih iz izvornog koda, koji se dalje razmatraju u okviru sintaksne analize.

**Sintaksni analizator** ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu formiraju gramatički ispravne sentence.

**Semantički analizator** pronalazi semantičke greške. Formira se na osnovu apstraktnog sintaksnog stabla koje je nastalo kao rezultat sintaksne analize.

**Generator koda** prevodi sintaksno i semantički ispravne programe u izvršni oblik za odabrano izvršno okruženje Mikrojava VM.

**Generisanje, Prevođenje i**

**Pokretanje**

Za generisanje I prevođenje koda prevodioca se koristi ant skripta *build.xml*. Ona sadrži target *compile* koji generiše kod leksera i parsera i prevodi kompletan projekat.

Nakon prevođenja projekta mogu se koristiti sledeće komande.

**Prevođenje mikrojava izvornog koda:**

*java -cp "bin;lib/\*" rs.ac.bg.etf.pp1.Compiler [izvorni fajl] [objektni fajl]*

**Disasembliranje objektnog fajla:**

*java -cp lib/mj-runtime-1.1.jar rs.etf.pp1.mj.runtime.disasm [objektni fajl]*

**Izvršavanje programa:**

*java -cp lib/mj-runtime-1.1.jar rs.etf.pp1.mj.runtime.Run [objektni fajl]*

**Debagovanje programa:**

*java -cp lib/mj-runtime-1.1.jar rs.etf.pp1.mj.runtime.Run -debug [objektni fajl]*

U slučaju da se izvorni i objektni fajl nalaze na putanjama *./test/program.mj* i *./test/program.obj* umesto gore navedenih komandi mogu se koristiti sledeći targeti ant skripte: *runParser*, *disasm*, *runObj*, i *debugObj, respektivno.*

**Opis test primera**

**parser\_sve\_ispravne\_smene** – Testira da li parser podržava sve smene propisane dokumentacijom. U izlazu ovog testa ne sme biti grešaka.

**parser\_oporavak\_od\_gresaka** – Testira da li se parser uspešno oporavlja od grešaka od kojih bi trebalo da se oporavi. U izlazu ovog testa ne sme biti fatalnih grešaka, već se parsiranje treba odraditi do kraja.

**semantic\_sve\_greske\_p1** – Testira da li semantički analizator pronalazi sve semantički nevalidne delove koda (sa izutetkom jedne, videti p2). Ovaj test treba da prijavi ukupno 29 grešaka.

**semantic\_sve\_greske\_p2** – Testira da li semantički analizator prijavljuje greške u potpisu main metode.

**Opis novouvedenih klasa**

**Compiler** – Fasada projekta. Prima putanje izlaznog i ulaznog fajla i vrši prevođenje kroz sve četiri faze.

**SemanticAnalyzer** – Visitor zadužen za semantičku analizu.

**CodeGenerator** – Visitor zadužen za generaciju koda.

**MyTab** – Proširenje klase *Tab* koje sadrži tip *bool* i razne pomoćne metode.

**MyCode** – Proširenje klase *Code* radi dodavanja podrške za *bool* array.

**MyDumpSymbolTableVisitor** – Proširenje klase *DumpSymbolTableVisitor* radi pravilnog ispisivanja *bool* tipa.