# MicroJava Compiler

### Nikola Pejic

#### Uvod

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. Ukoliko se tokom leksičke analize detektuje leksička greška, potrebno je ispisati odgovarajuću poruku na izlaz.

*Sintaksni analizator* ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu formiraju gramatički ispravne sentence. Tokom parsiranja Mikrojava programa potrebno je na odgovarajući način omogućiti i praćenje samog procesa parsiranja na način koji će biti u nastavku dokumenta detaljno opisan. Nakon parsiranja sintaksno ispravnih Mikrojava programa potrebno je obavestiti korisnika o uspešnosti parsiranja. Ukoliko izvorni kod ima sintaksne greške, potrebno je izdati adekvatno objašnjenje o detektovanoj sintaksnoj grešci, izvršiti oporavak i nastaviti parsiranje.

*Semantički analizator* se formira na osnovu apstraktnog sintaksnog stabla koje je nastalo kao rezultat sintaksne analize. Semantička analiza se sprovodi implementacijom metoda za posećivanje čvorova apstraktnog sintaksnog stabla. Stablo je formirano na osnovu gramatike implementirane u prethodnoj fazi. Ukoliko izvorni kod ima semantičke greške, potrebno je prikazati adekvatnu poruku o detektovanoj semantičkoj grešci.

*Generator koda* prevodi sintaksno i semantički ispravne programe u izvršni oblik za odabrano izvršno okruženje Mikrojava VM. Generisanje koda se implementira na sličan način kao i semantička analiza, implementacijom metoda koje posećuju čvorove.

#### Komande (primeri)

Generisanje leksera

java -jar JFlex.jar -d ..\src\rs\ac\bg\etf\pp1 ..\spec\mjlexer.flex

Generisanje parsera

java -jar cup\_v10k.jar -destdir ..\src\rs\ac\bg\etf\pp1 -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup

Pokretanje MJ programa

java -jar mj-runtime.jar -debug out\program.obj

Dekompajliranje

java -cp mj-runtime.jar rs.etf.pp1.mj.runtime.disasm ..\out\program.obj

#### Test Primeri

Test primeri su podeljeni po fazama projekta (A, B, C) koje testiraju. Takodje, podeljeni su na Green I Red testove, odnosno na one koji prikazuju rad kompajlera u slučaju ispravnog i neispravnog MJ koda.

#### Uvedene klase

* CodeGenerator - Klasa koja se koristi za generisanje koda.
* SemanticAnalyzer - klasa koja se koristi za semantičku analizu koda
* Compiler - klavna klasa koja koristi funkcionalnosti CodeGenerator i SemanticAnalyzer-a za prevodjenje koda.
* CounterVisitor - ova klasa, kao i sve klase koje su izvedene iz nje, koriste se za brojanje pojavljivanja odredjenih elemenata u ulaznom kodu.
* GlobalStuff- sadrži uslužne metode koje se pozivaju iz CodeGenerator i SemanticAnalyzer klasa.