Univerzitet u Beogradu

Elektrotehnički fakultet

**Kompajler za Mirkojavu**

Igor Stevanoski 2018/0190

Beograd

2023

**Uvod**

Cilj projektnog zadatka je realizacija kompajlera za programski jezik Mikrojava. Projekat se sastoji iz 4 dela: Leksička analiza, Sintaksna analiza, Semantička analiza i Generisanje koda.

Leksičku analizu vrši Lekser, čija je uloga da ulazni tekstualni fajl podeli na manje jedinice ( tokene) koje olakšavaju nadalje rad parsera.

Sintaksnu analizu vrši Parser, koji na osnovu ulaznih tokena proverava da li je dati kod u skladu sa opisanom gramatikom jezika. Proizvodi apstraktno sintaksno stablo koje se dalje prosledjuje na semantičku obradu.

Semantičku analizu vrši semantički analizator, koji na osnovu prosledjenog apstraktnog sintaksnog stabla proverava da li je dati kod u skladu sa semantikom jezika.

Generisanje koda se vrši na kraju za kod koji je sintaksno I semantički ispravan. Produkt ove faze je objektni fajl koji se može izvršavati na MikroJava virtuelnoj mašini.

**Upotreba alata i pokretanje programa**

Projekat se pravi pomoću build.xml fajla koji u sebi sadrzi skripte za generisanje pojedinačnih delova. Lekser se automatski generiše pomoću jflex alata koji zahteva prethodno napisanu specifikaciju za samu leksičku analizu u vidu .flex fajla ( u ovom projektu je to mjlexer.flex ). Parser se automatski generiše pomoću jcup alata koji zahteva prethodno napisanu specifikaciju za sintaksnu analizu u vidu .cup fajla ( u ovom projektu je to mjparser.cup). Zatim se sintaksno ispravan program šalje dalje na semantičku proveru ( u ovom projektu je to posao klase SemanticAnalyzer.java) i na samom kraju generisanje koda ( u ovom projektu je to posao klase CodeGenerator.java) preko runobj target-a u build.xml fajlu.

Provera da li je rešenje sintaksno i semantički ispravno vrši se pokretanjem main metode klase Compiler.java, koja produkuje objektni fajl. Sadržaj objektnog fajla se može proveriti disasm opcijom build.xml fajla, a isti se može izvršiti opcijom runObj.

**Opis priloženih test primera**

Sintaksna analiza: sadrži 4 test primera, od kojih je samo prvi ispravan, a ostala tri imaju grešku.

Prvi test ( testParser1.mj) pokriva sve smene koje odobrava data gramatika.

Drugi test ( testParserF1.mj - F od failure ) prikazuje oporavak od grešaka pri deklaraciji promenljivih I konstanti, kao I grešku u izrazu za dodelu vrednosti.

Treći test ( testParserF2.mj ) prikazuje rad parsera pri nepotpunoj deklaraciji klase.

Četvrti test ( testParserF3.mj ) prikazuje rad parsera pri nepotpunom iskazu.

Semantička analiza: sadrži 5 test primera, od kojih je samo prvi ispravan a ostala četiri imaju grešku.

Prvi test ( testSemantic1.mj ) ispravan kod za sve tražene funkcionalnosti semantičke obrade.

Drugi test ( testSemanticF1.mj ) prikazuje grešku pri upotrebi nedeklarisane promenljive.

Treći test ( testSemanticF2.mj ) prikazuje grešku pri sabiranju nekompatibilnih tipova.

Četvrti test ( testSemanticF3.mj ) prikazuje grešku pri redefinisanju konstante.

Peti test ( testSemanticF4.mj ) prikazuje grešku pri pokušaju menjanja vrednosti konstante.

Generisanje koda: sadrži 3 test primera.

Prvi test ( testCodeGenerator1.mj ) prikazuje rad promenljivih sa svim operacijama.

Drugi test ( testCodeGenerator2.mj ) prikazuje rad sa nizovima i svim tipovima podataka.

Treći test ( testCodeGenerator3.mj ) prikazuje rad izrazom oblika: “[ DesignatorList ] = array;” .

**Novouvedene klase**

Pored klasa koje su opisane tekstom projektnog zadatka nema novouvedenih klasa.