**Programski prevodioci 1**

**Projekat  
- Kompajler za Mikrojavu -**

**Aleksa Vuković 18/0354**

*Profesor: Dragan Bojić   
Asistenti Maja Vukasović, Mihajlo Ogrizović  
Godina: 2021/2022.*

**Postavka zadatka**

Cilj izrade projektnog zadatka jeste implementacija potpuno funkcionalnog programskog prevodioca za programski jezik Mikrojava. Sama izrada projekta se satoji iz 4 faze: leksicka analiza, sintaksna analiza, semanticka analiza i generisanje koda. Krajnji rezultat jeste generisani objektni fajl koji se moze pokrenuti i testirati njegov rad, koji je pre toga zadovoljio sve uslove i specifikacije jezika Mikrojava.

**Opis komandi**

1. Leksicka analiza - prilikom izrade ove faze projekta, koriscea je JFlex.jar biblioteka, koja na osnovu napisane lex specifikacije (mjlexer.lex) generise Yylex klasu koja vrsi leksicku analizu.

2. Sintaksna analiza - prilikom izrade ove faze projekta, koriscena je cup\_v10k.jar biblioteka, koja na osnovu napisane cup specifikacije (mjparser.cup) izvrsava sintaksnu analizu programa. Na osnovu nje se generisu klase (u ast paketu) koje predstavljaju elemente sintaksnog stabla. Nakon uspesnog parsiranja se generise sintaksno stablo.

3. Semanticka analiza - prilikom izrade ove faze projekta, koriscena je symboltable-1-1.jar biblioteka, koja na osnovu generisanog sintaksnog stabla i njegovim obilaskom izvrsava semanticku analizu programa. Semanticki analizator se integrise sa tabelom simbola u koju se smestaju svi objektni cvorovi po zadatim pravilima koje obilazimo.

4. Generisanje koda - prilikom izrade ove faze projekta, koriscena je mj-runtime-1-1.jar biblioteka, koja na osnovu generisanog sintaksnog stabla, koje je nastalo kao rezultat sintaksne analize i semanticke, provere vrsi generisanje bajtkoda za Mikrojava virtuelnu masinu.

Prevodjenje koda zapocinjemo pokretanjem main metode Compiler klase koja prima 2 argumenta (putanju do ulaznog .mj fajla i putanju na kojoj ce se generisati izlazni .obj fajl).

Testiranje koda sprovodimo pokretanjem main metode Run klase koja prima izlazni .obj fajl uz opcioni -debug argument.

**Testiranje**

U prilozenim test primerima su pokrivene sve smene gramatike koje zadovoljavaju specifikaziju jezika Mikrojava i koje demonstriraju rad sintaksnog i semantickog analizatora uz generisanje ispravnog bajtkoda na samom kraju.

Test 1 prikazuje rad sa promenljivim i konstantama, uz neke osnovne aritmeticke operacije.

Test 2 i 3 detaljno prikazuju rad sa nizovima uz koriscenje aritmetickih operacija, razlicitih nacina pristupa elementu niza i rad sa metodama fiksnih i promenljivih broja argumenata.

Test 4 prikazuje rad sa recordima, pristup elementima recorda.

Test 5 detaljno prikazuje rad sa if-else i do-while, rad sa relacionim i logickim operacijama.

**Novouvedene klase**

CodeGenerator - vrsi generisanje Mikrojava bajtkoda prilikom obilaska sintaksnog stabla

Compiler - pokrece se njegova main metoda koja zapocinje parsiranje i prevodjenje programa (zahteva 2 argumenta)

CounterVisitor - vrsi prebrojavanje formalnih parametara i lokalnih promenljivih (kao sto je pokazano na mini-domacem uz adaptaciju za spostveni program)

RuleVisitor - vrsi prebrojavanje print statement-a i broji deklaracije promenljivih

SemanticAnalyzer - vrsi semanticku analizu programa obilaskom sintaksnog stabla