Kompajler za *MicroJava*

Filip Stojsavljevic 2019/0656

Sadržaj

[Opis postavke zadatka 3](#_Toc126492141)

[Kratak opis 3](#_Toc126492142)

[Opis faza projekta 3](#_Toc126492143)

[Opis komandi i klasa 4](#_Toc126492144)

[Opis korišćenih klasa 4](#_Toc126492145)

[Opis komandi 4](#_Toc126492146)

[Komande korišćene u build.xml 4](#_Toc126492147)

[Komande za pokretanje preko komandne linije 4](#_Toc126492148)

[Opis testova 5](#_Toc126492149)

# Opis postavke zadatka

## Kratak opis

Cilj ovog projekta je bio da se upoznamo na praktičan način sa funkcionisanjem jednog programskog prevodioca kroz izradu istog. Programski jezik za koji se izrađivao programski prevodilac je *MicroJava*, kreiran upravo u svrhe učenja.

Ideja je bila da prolazimo kroz faze izrade pravog kompajlera, na manjem nivou, i da se na taj način upoznamo sa svim funkcionalnostima.

## Opis faza projekta

U projektu su od značaja sledeće faze:

1. Leksička analiza
2. Sintaksna analiza
3. Semantička analiza
4. Generisanje koda

**Leksička analiza** obuhvata prepoznavanje teksta kroz korišćenje *JFlex* alata.

Sintaksna analiza obuhvata generisanje parsera koristeći CUP alat. Uz to generisano je i sintaksno stablo koje na slikovit način prikazuje rad našeg parsera.

Semantička analiza predstavlja prepoznavanje svih eventualnih grešaka ili nedozvoljenih programskih paradigmi u programskom kodu. Takođe je rađeno i popunjavanje tabele simbola, čija implementacija nam je bila dostupna

Generisanje koda je poslednja faza u izradi ovog projekta, i predstavlja generisanje objektnog koda za *MicroJava* jezik. Uz ovo koristili smo i olakšice nuđene u klasi *Code*.

# Opis komandi i klasa

## Opis korišćenih klasa

Potrebno je napisati sledeće klase:

* Yylex.java – predstavlja lekser koji skenira tekst i generiše tokene
* MJParser.java – radi parsiranje teksta našeg programskog koda
* SemanticAnalyzer.java – radi semantičku analizu teksta i popunjavanje tabele simbola
* CodeGenerator.java – radi generisanje koda

## Opis komandi

### Komande korišćene u build.xml

Sledeće komande su od značaja u build.xml fajlu, koji služi za pokretanje i prevođenje programskog koda na asemblerski kod:

* *Repackage* – premeštanje svih generisanih klasa u zaseban paket
* *Compile* – prevod svih potrebnih i napisanih klasa
* *RunObj* – pokretanje programa

Ostale funkcije su zavisne od gore navedenih i nisu neophodne.

### Komande za pokretanje preko komandne linije

Odbrana može da se izvede i preko ANT alata, ali i iz komandne linije, neophodne su sledeće komande:

* Prenos izlaza sa standardnog na .out ili .err izlazne fajlove

java Compiler.class program.mj program.obj 1>izlaz.out 2>izlaz.err

* Prepisivanje prevedenog u .obj fajla

java Compiler.class program.mj program.obj

# Opis testova

Testovi služe da se proveri valjanost rešenja i da li se program izvršava na korektan način. Postoje testovi za sve faze, ali od značaja su oni koji rade testiranje svih faza, javni testovi, koji su podeljeni po nivoima izrade.

Testovi 301 i 302 su za A i B nivo, ti testovi su realizovani budući da je ovaj projekat izrađen za B nivo. Ovi testovi testiraju funkcionalnosti poput deklarisanja promenljivih i globalnih metoda, njihovo pozivanje i menjanje vrednosti, testiranje uslovnih skokova i petlji.