**Univerzitet u Beogradu**



**Projekat iz predmeta Programski prevodioci 1**

**Kompajler za Mikrojavu**

**Jana Milovanović 2019/0292**

Beograd, januar 2023. god.

Opis zadatka

**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 Mikrojava virtuelnoj masini.

Sintaksno i semantički ispravni Mikrojava programi su definisani specifikacijom [MJ].

Programski prevodilac za Mikrojavu ima 4 osnovne funkcionalnosti:

* leksičku analizu
* sintaksnu analizu
* semantičku analizu
* generisanje koda.

**Leksički analizator** prepoznaje jezičke lekseme i vraća skup tokena izdvojenih iz izvornog koda, koji se dalje razmatraju u okviru sintaksne analize. Ukoliko se tokom leksičke analize detektuje leksička greška, ispisaće se odgovarajuća poruka na izlaz.

**Sintaksni analizator** utvrđuje da li izdvojeni tokeni iz izvornog koda programa mogu formiraju gramatički ispravne sentence. Nakon parsiranja sintaksno ispravnih Mikrojava programa obaveštava se korisnik o uspešnosti parsiranja. Ukoliko izvorni kod ima sintaksne greške, izdaje se adekvatno objašnjenje o detektovanoj sintaksnoj grešci, izvršava oporavak i nastavlja 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, prikazuje se adekvatna poruka o detektovanoj semantičkoj grešci.

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

Opis komandi za generisanje java koda, prevodjenje koda kompajlerom i pokretanje i testiranje resenja

Graphical user interface, text, application

Description automatically generated

**Generisanje JAVA koda** .(slika1):

Java kod se generise pomocu ant alata. Desnim klikom na build.xml fajl odabirom opcije Run As->Ant Build->compile generisace se klase Yylex I MJParser.

Graphical user interface, text, application, email

Description automatically generated

**Prevodjenje koda kompajlerom**.(slika2) :

Glavni program kompajlera predstavlja klasa Compiler.java u kojoj se redom pokreću sva 4 koraka odnosno leksička, sintaksna, semantička analiza i na kraju generisanje MicroJava koda. Kao rezultat se dobija .obj fajl. Ovoj klasi treba proslediti kao prvi argument putanju do fajla sa izvornim mikrojava kodom, a kao drugi argument putanju do .obj fajla.

Graphical user interface, text, application, email

Description automatically generated

**Pokretanje I tesitranje resenja** .(slika3) :

U folderu test nalaze se testovi za testiranje projekta. U paketu my\_test.correct nalaze se potpuno ispravni testotovi. U paketu my\_test.incorrect nalaze se testovi koji imaju sintaksne ili semanticke greske I kao rezultat njihovog pokretanja dobijamo informacije koje su se greske desile.

Pokretanje i testiranje resenja se vrsi desnim klikom na build.xml fajl odabirom opcije Run As->Ant Build->runObj dobijamo u konzoli rezultat izvrsenja naseg mikrojava koda. Fajl input.txt koristi se za unos podataka koji su potrebni nasem programu.

**Novouvedene klase**

# MySimbolTable.java:

Uveli smo klasu **rs.ac.bg.etf.pp1.MySimbolTable** koja proširuje klasu ***Tab*** iz *rs.ac.bg.etf.pp1.symboltable* paketa na takav način da se uvek pri pozivanju njene *init()* metode u tabelu simbola veštački ubacuje i boolean tip podataka koji inače nije bio predefinisan u osnovnoj ***Tab*** klasi.

# MyDumpSymbolTableVisitor.java:

Uveli smo klasu **rs.ac.bg.etf.pp1.MyDumpSymbolTableVisitor** koja proširuje klasu **DumpSymbolTableVisitor** iz *rs.ac.bg.etf.pp1.symboltable* da bi se omogucili tekstualni ispis i bool tipa.

# Conditions.java:

Uveli smo klasu **rs.ac.bg.etg.pp1.Conditions.java** koja predstavlja jedan Condition neterminal, odnosno stablo condition term i condition factor-a.