Univerzitet u Beogradu - Elektrotehnički fakultet

Programski Prevodioci 1



Domaći zadatak

Izveštaj o urađenom domaćem zadatku

|  |  |
| --- | --- |
|  | Student: |
|  | Miloš Ćurčić 2017/0379 |

Beograd, jun 2020.

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 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.

Test primeri

Nazivi fajlova test primera odgovaraju oblastima funkcionalnosti programskog prevodioca koje se testiraju, u okviru komentara test primera je prikazan očekivani izlaz za svaki test primer u slučaju pokretanja. U suprotnom prikazuje greške po redosledu navođenja funkcionalnosti u tekstu samog zadatka.

Novouvedene klase

Novouvedene klase odgovaraju celinama opisanim u samom opisu zadatka gde je ukratko i opisana njihova odgovarajuća funkcionalnost.

Prevođenje i pokretanje rešenja

Ako imate instaliran Apache Ant program kod možete lako prevesti uz pomoću sledeće komande:

* ant -buildFile build.xml compile

Ili uz pomoć grafičkog IDE alatu kao što je IntelliJ:

Graphical user interface, application

Description automatically generated

**Pokretanje parsera**

java -cp 'bin;lib/cup\_v10k.jar;lib/JFlex.jar;lib/log4j-1.2.17.jar;lib/mj-r

untime-1.1.jar;lib/symboltable-1-1.jar' rs.ac.bg.etf.pp1.Compiler test/test301.mj test/program.obj

**Disasembliranje dobijenog fajla**

java -cp 'lib/mj-runtime-1.1.jar' rs.etf.pp1.mj.runtime.disasm test/progra

m.obj

**Pokretanje programa**

java -cp 'lib/mj-runtime-1.1.jar' rs.etf.pp1.mj.runtime.Run test/program.o

bj