Univerzitet u Beogradu - Elektrotehnički fakultet

Programsi prevodioci 1 (13S114PP1)



Projekat – Kompajler za Mikrojavu

Izveštaj o urađenom projektnom zadatku

|  |  |
| --- | --- |
| Nastavnik: |  |
| dr Dragan Bojić, redovni profesor  Asistenti:  mast. inž. Maja Vukasović  mast. inž. Mihajlo Ogrizović | Student:  Stefan Branković 0253/2019 |

Beograd,februar 2023.

Sadržaj

[Sadržaj 2](#_Toc124881392)

[1. Opis postavke zadtka 3](#_Toc124881393)

[2. Opis komandi 4](#_Toc124881394)

[4. Opis test primera 5](#_Toc124881395)

[5. Opis novouvedenih klasa 6](#_Toc124881396)

1. Opis postavke zadtka

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

**Sintaksni analizator** ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu Da formiraju gramatički ispravne sentence.

**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.

**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.

1. Opis komandi

Generisanje leksera:

* Pokretanje build.xml fajla ANT alatom i izabrati **lexerGen** za target

Generisanje parsera:

* Pokretanje build.xml fajla ANT alatom i izabrati **compile** za target

Semantička analiza i prevođenje koda:

* Pokretanje main funkcije klase Compiler gde se pozivaju obilasci semantičkog analizatora i generatora koda

Pokretanje prevedenog koda:

- java -cp lib/mj-runtime-1.1.jar rs.etf.pp1.mj.runtime.Run [-debug] <putanja do .obj fajla>

- opcioni debug flag služi za ispisivanje steka

- u slučaju da nije moguće pokrenuti iz komandne linije dodati

<redirector input="input.txt"></redirector> u runObj target u build.xml fajlu (ispod arg taga)

1. Opis test primera

U folderu test se nalaze svi ulazni test fajlovi sa ekstenzijom .mj, kao i odgovarajući izlazni fajlovi koji su rezultat testiranja, sa istim imenom kao ulazni fajl, ali sa ekstenzijom .out za standardni izlaz i .err za izlaz greške.

Test *test1.mj* služi za testiranje faze leksičke analize i detekciju ispravnih i pogrešnih leksema.

Test *syntax\_errors.mj* služi za testiranje sintaksne analize i detektovanje grešaka uz oporovak od istih.

Test *test3.mj* služi za testiranje faze semantičke analize i detektovanje korišćenih simbola uz proveru kontekstnih uslova.

Test *test4.mj* služi za testiranje faze generisanja koda.

Test *pr1.mj* testira sve faze programa kroz primer citanja i menjanja atributa klasnih objekata.

Testovi *programA.mj, programB.mj* i *programC.mj* testiraju svaki konstrukt iz respektivnih faza na odgovarajući način (sve tri faze)

Testovi *test301.mj*, *test302.mj* i *test303.mj* su javni testovi koji testiranju sve faze projekta za sva tri nivoa.

1. Opis novouvedenih klasa

Nije korišćena nijedna nova klasa sem onih koji su navedene u samoj specifikaciji projektnog zadatka.