Univerzitet u Beogradu – Elektrotehnički Fakultet

Računarska Tehnika i Informatika



**Programski Prevodioci 1**

**Kandidat**

Miloš Milisavljević 2015/0588

Beograd, januar 2022. godine

**Kratak opis projekta**

Cilj ovog projekta je realizacija kompajlera za MikroJavu. Kompajler omogućava prevođenje sintaksno i semantički ispravan kod u bajtkod za virtuelnu mašinu MikroJave. Sintaksna specifikacija definiše sintaksna pravila za ovaj jezik, dok Semantička specifikacija definiše semantička pravila.

Zadatak je bio da se napišu Leksička specifikacija, koja opisuje moguće tokene koji se mogu nalaziti u ulaznom fajlu. Zatim Sintaksna specifikacija, koja opisuje pravilnu sintaksu ulaznog fajla. Zatim Semantička specifiacija, koja definiše šta je semantički ispravno napisati u MikroJavi. Na kraju, piše se Generator Koda koji ima zadatak da generiše bajt kod za MikroJavu.

Za Semantičku analizu se pisao Semantički analizator, koji ima zadatak da obiđe sintaksno stablo i proveri da li je ono ispunjava semantička pravila jezika.

**Komande za generisanje koda**

**Leksička analiza**(Otvoriti cmd u src folderu):

java -cp ../lib/JFlex.jar JFlex.Main -d rs\ac\bg\etf\pp1 ..\spec\mjlexer.lex

**Komanda za parser:**

java -cp ../lib/cup\_v10k.jar java\_cup.Main -destdir rs\ac\bg\etf\pp1 -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup

**Pokretanje Compiler.java:**

Preko Run Configuration... se podešavaju argumenti: test/program.mj test/program.obj

**Izvršavanje bajtkoda(Run.class):**

U biblioteci mj-runtime pronaći klasu Run. Za nju definisati, preko Run Configurations..., parametre -debug test\program.obj

**Ispis bajtkoda(disasm.class):**

U biblioteci mj-runtime pronaći klasu disasm. Za nju definisati, preko Run Configurations..., parametre test\program.obj

**Opis test primera**

Pored javnih testova, postoje dva dodatna testa. **Syntax.mj** je test koji ima za zadatak da proveri ispravno ispisivanje sintaksnih grešaka u kodu i da odradi oporavak od grešaka, tako gde je neophodno.

Drugi dodatni test je **semantic.mj** koji ima za zadatak da prikaže ispravne i neispravne semantičke izraze i kako se Compiler snosi sa njima.

**Dodatne pomoćne klase**

* **MyLabel** – Klasa koja je namenjena za upotrebu labela. Sadrži ime labele, adresu na kojoj se labela nalazi i sadrži adresu na kojoj je neophodno da se vrati i upiše skok unapred
* **MyMethod** – Klasa namenjena da čuva Obj za svaku metodu i listu njenih formalnih parametara(funkcionalnost koja nije namenjena za nivo A)
* **MyObj** – Klasa koja nasleđuje klasu Obj, kako bi se uveo tip za labele MyObj.Lab
* **MyStruct** – Klasa koja proširuje klasu Struct. Nove stvari su tip strukture za labele i metoda za jednostavnije definisanje niza
* **MyTab** – Klasa koja proširuje klasu Tab, radi uvođenja tipa za Boolean promenljive