Kompajler za MikroJavu

Programski prevodioci 1

Elektrotehnički fakultet Univerziteta u Beogradu

* **OPIS PROJEKTNOG ZADATKA**

Cilj projektnog zadatka je da studente upozna sa osnovnim funkcionalnostima i koracima prilikom prevodjenja programa, u ovom slučaju radi se o pojednostavljenom prevodjenju Java aplikacija.

Projekat prolazi kroz 4 faze prevodjenja, i to, redom:

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

Svaka faza se direktno oslanja na prethodne.

Na osnovu zadate gramatike i specifikacije MikroJava programskog jezika, potrebno je formirati leksičke simbole, gramatiku, a potom i korišćenjem LALR parsera formirati sintaksno stablo, a potom ga kasnije i obići u cilju provere ispravnosti programa.

Takodje, implementirano je prijavljivanje grešaka, tako što se po fazama sakupljaju greške, I potom, ukoliko postoje, ispisuju na kraju parsiranja programa.

* **POKRETANJE PROJEKTA**

Prilikom izvršavanja leksičke i sintaksne analize, potrebno je u korenom direktorijumu pokrenuti **build.xml** fajl, na sledeći način

(desni klik na build.xml -> Run as -> Ant Build -> označiti **compile[default]**)

U nastavku, za izvršavanje semantičke analize i generisanja koda, pokreće se **MJParserTest.java** fajl koji se nalazi u direktorijumu src/rs/ac/bg/etf/pp1. Potrebno je dodati 2 argumenta komandne linije, a to su nazivi test programa koji obavezno mora imati ekstenziju **.mj**, kao i naziv **.obj** fajl programa koji nastaje kao rezultat generisanja koda.

Oba fajla se moraju naći u direktorijumu **test**.

Primer argumenata: test/program.mj test/program.obj

Kada se izvrše svi prethodni koraci, moguće je pokrenuti projekat u više modova:

* Run debug – gde se mogu tačno videti koraci izvršsavanja kao i stanje steka programa
* Run release – klasično izvršavanje programa
* disasm – prikaz mašinskih instrukcija koje saćinjavaju prevedeni program

\* Savet prilikom izrade je da se projekat pokrene na sve načine zbog lakših uočavanja grešaka.

* **PRILOŽENI TESTOVI**

**test301.mj** – javni test za A nivo projekta

**test302.mj** – javni test za nivo B projekta

**program.mj** – modifikovani javni test za nivo B projekta

* **NOVOUVEDENE KLASE**

**CodeGenerator** – klasa koja služi za obilazak sintaksnog stabla i generisanje koda. Implementira VisitorAdaptor.

**CounterVisitor** – klasa koja sa svoje 2 statičke klase FormParamCounter i VarCounter koje su izvedene iz CounterVisitor-a, prebrojavaju formalne parametre i lokalne promenljive metoda.

Implementira VisitorAdaptor.

**MJParserTest** – klasa koja u stvari predstavlja klasu sa main metodom, i ona se pokreće u cilju kompajliranja programa. Main metoda ove klase sadrži argumente naziva ulaznog i izlaznog fajla, kao i da li je potrebno rezultat kompajliranja ispisati u fajl/fajl sa greškama.

Implementira Compiler interfejs, i u njenoj metodi compile vrši sav traženi posao.

**SemanticAnalyzer** – klasa koja se koristi u semantičkoj analizi i slući za obilazak apstraktnog sintaksnog stabla, i dodavanje čvorova u tabelu simbola.

Implementira VisitorAdaptor.

**CompilerError** – klasa koja je data kao deo postavke projekta uz interfejs Compiler. Objekti ove klase u stvari predstvaljaju objekte sa informacijama o različitim tipovima grešaka.