Elektrotehnički fakultet

Univerzitet u Beogradu

Projekat iz predmeta programski prevodioci 1

Kompajler microjava programskog jezika

Djordje Dimitrijević 2018/0489

Beograd, feb 2022.

Opis projektnog zadatka

Zadatak samog projekta jeste funkcionalni kompajler za jezik microjava. Sam zadatak se može podeliti na 4 manje celine:

1. Lekser – Procesira ulazni izvorni kod i tokenizuje ga. U slučaju da naiđe na nevalidni token prijavljuje grešku i nastavlja dalje sa procesiranjem. Tokeni sadrze tekst unutar sebe. Tokeni se specificiraju unutar mjlexer.flex. Ovakav kod se dalje predaje parseru.
2. Parser (Sintaksni analizator) – Prima tokenizovani izvorni kod od leksera i vrši sintaksnu analizu generišući apstraktno sintaksno stablo nad tim kodom služeći se LALR(1) gramatikom jezika zadatom unutar mjparser.cup fajla. Obilazi AST prema postorder redosledu.

3. Semantički analizator – Obilaskom apstraktnog sintaksnog stabla (generisanom u prošloj celini) proverava semantičku korektnost izvornog koda i generiše tabelu simbola. Za generisanje tabele simbola se koristi symboltable.jar. Semanticki zahtevi su definisani u opisu projektnog zadatka

4. Generator koda – Ponovnim obilaskom nakon semantičke analize i služenjem prethodno generisanom tabelom simbola generiše se instrukcije za bajtkod za izvršno okruženje za MJVM (Micro Java Virtual Machine)

Opis klasa rešenja

- MJCompiler.java – Pokreće sve celine kompajlera i ispisuje njihov output i greške
- SemanticAnalyzer.java – Radi semantičku analizu i generiše tabelu simbola
- CodeGenerator.java – Generiše MJ bajtkod za MJVM
- CondObj.java - pamti sve adrese koje su potrebne za jedan CONDITION, u obilasku AST stavljeni su dummy skokovi koji se kasnije menjaju na prave adrese.
- JmpReplace.java - sadrzi tri informacije, adresa na kojoj je instrukcija, gde treba da se skoci I koja operacija je u pitanju
- FunctionParams.java - pamti parametre za jednu funkciju

Testiranje i pokretanje

Testovi se nalaze u test folderu.
Generisanje potrebnih klasa se radi pokretanjem default ant target build-a u build.xml.
Pokretanje mikrojava kompajlera se radi pokretanjem main metode sa odgovarajućim argumentima komandne linije klase MJCompiler.
Izvršavanje prevedenog mikrojava bajtkoda se radi pokretanjem odgovarajuće klase u mj-runtime arhivi.