**Elektrotehnički fakultet Univerzitet u Beogradu**

**Projekat iz predmeta programski prevodioci 1**

Kompajler MicroJava programskog jezika

Andreja Andrejić 2019/0162

Beograd, feb 2023.

# Opis projektnog zadatka

Zadatak samog projekta jeste funkcionalni kompajler za jezik MicroJava. Sam zadatak se može podeliti na 4 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 sadrže 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

* Compiler.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

# Testiranje i pokretanje

Testovi se nalaze u test folderu.

Generisanje potrebnih klasa se radi pokretanjem lexerGen, a zatim default(compile) ant target build-a u build.xml. Pokretanje mikrojava kompajlera se radi pokretanjem pokretanjem ant runObj build-a u build.xml.

Izvršavanje prevedenog mikrojava bajtkoda se radi pokretanjem odgovarajuće klase u mj- runtime arhivi.