Електротехнички факултет

Софтверско инжењерство

**Компајлер за MicroJava**

Мај, 2017

Професор: Драган Бојић Студент:

Асистент: Немања Којић Ђорђе Живановић 0033/2013

# Опис проблема

Програмски преводилац за MicroJava има четири основне функционалности: лексичку анализу, синтаксну анализу, семантичку анализу и генерисање кода.

Лексички анализатор препознаје језичке лексеме и враћа скуп токена издвојених из изворног кода, који се даље разматрају у оквиру синтаксне анализе. Уколико се током лексичке анализе детектује лексичка грешка, исписује одговарајућу поруку на излаз.

Синтаксни анализатор има задатак да утврди да ли издвојени токени из изворног кода програма могу формирају граматички исправне сентенце. Током парсирања MicroJava програма на одговарајући начин се омогућава праћење самог процеса парсирања. Након парсирања синтаксно исправних MicroJava програма корисник добија извештај да ли је парсирање завршено успешно. Уколико изворни код има синтаксне грешке, издаје се адекватно објашњење о детектованој синтаксној грешци, изршава опоравак и наставља парсирање.

Семантички анализатор се добија проширењем функционалности синтаксног анализатора. Семантичка анализа се врши спроводи кроз синтаксно-управљано превођење. Основној граматици, којом је специфициран синтаксни анализатор, додају се атрибути и акције (атрибутивно-транслациона граматика). Такође као код синтаксе издаје се текст грешке ако се деси.

Генератор кода преводи синтаксно и семантички исправне програме у извршни облик за одабрано извршно окружење MicroJava VM. Генерисање кода се имплементира на сличан начин као и семантичка анализа, кроз надоградњу синтаксног анализатора (синтаксно управљано превођење).

# Команде

Све команде се покрећу прављењем нове run конфигурације у Eclipse-у.

1. Лексер генератор: Покреће се JFlex.Main и креира Yylex.java:

Jflex.Main -d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex

1. Лексер тест:
2. Лексер тест: rs.ac.bg.etf.pp1.MJLexerTest се покреће и прослеђује се параметар test/lexer/MJLexerTest1.mj
3. Парсер генератор: java\_cup.Main се покреће, аргументи су:

-destdir src\rs\ac\bg\etf\pp1 -parser MJParser spec\mjparser.cup

1. Парсер тест: rs.ac.bg.etf.pp1.MJParserTest се покреће, параметри су: test/generator/C/Input/dummy.mj (за овај случај) и даље одговарајуће вредности кажу да ли има дебаговања, да ли се гледа само синтакса…
2. Извршавање објектног фајла: rs.etf.pp1.mj.runtime.Run се покреће и параметар је test\generator\program.obj (у овом случају одговарајући објектни фајл)
3. Коришћен је и дисасемблер: rs.etf.pp1.mj.runtime.disasm и аргумент test\generator\program.obj
4. Коришћено је и UnitTest окружење unit.Main са параметрима : -t gen b true -rp true -n -1
5. Коришћен је и : rs.ac.bg.etf.pp1.MJCodeGeneratorTest за поређење излаза који је примао следеће вредности: test/generator/A/Output/MJCodeGenerator1.out

test/generator/A/Output/MJCodeGenerator1.test

# Тест примери

Тест примери се налазе у фајлу test и одговарајућим фолдерима generator, parser(syntax, semantic) и lexer. Сваки има подфолдере за тестирање одговарајућих нивоа пројекта (А, Б, Ц) и за сваки ако име не одговара тесту, онда је дат детаљнији опис теста унутар самог теста. Пројекат је UnitTest-иран и читава имплементација UnitTest-ова налази се у пакету unit.

# Опис новоуведених класа

Уведене класе унутар пакета unit за циљ имају конкретизацију тестова за одговарајућу анализу (лексичку, ..) Направљена је и фабрика UnitTestFactory која има за циљ креирање датих класа за тестирање. UnitTestResult преноси поруко о грешци.Класе унутар фолдера test и унутар пакета rs.ac.bg.etf.pp1 за циљ имају да тестирају да ли се десила лексичка, семантичка/синтаксна и да проверава да ли је исправан резултат извршавања програма. У mjparser.cup-у дефинисана је класа ObjResultWrapper која за циљ има пропагацију резултата кроз синтаксно стабло. Уведена је класа ParserCnt која за циљ има да омота бројање синтаксних симбола и класа TabExt која је проширење Tab-а. Такође нова верзија DumpTableVisitor-а је написана: DumpUpgradeSymbolTableVisitor која проширује функционалности.