Универзитет у Београду

Електротехнички факултет

Одсек СИ

A picture containing text, watch

Description automatically generated

Програмски преводиоци 1

Пројектни задатак – извештај

Студент:

Александра Богићевић 0390/17

Београд, школска година 2020/2021

**Опис поставке задатка**

Циљ пројектног задатка је реализација комапјлера за програмски језик Микројаву. Компајлер омогућава превођење синтаксно и семантички исправних Микројава програма у Микројава бајткод који се извршава на виртуелној машини за Микројаву.

Програмски преводилац за Микројаву 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. Лексички анализатор је имплементиран унутар фајла *mjlexer.flex*.

Синтаксни анализатор има задатак да утврди да ли издвојени токени из изворног кода програма могу да формирају граматички исправне секвенце на основу граматике из фајла *mjparser.cup*.

Семантички анализатор се форира на основу апстрактног синтаксног стабла које је настало као резултат синтаксне анализе. Семантичка анализа се спроводи имплементацијом метода за посећивање чворова апстрактног синтаксног стабла унутат фајла *SemanticAnalyzer.java*.

Генератор кода преводи синтаксно и семантички исправне програме у извршни облик за одабрано извршно окружење Микројава ВМ. Генератор кода је дат у фајлу *CodeGenerator.java*.

**Опис команди за генерисање јава кода алатима, превођење кода компајлером, покретање и тестирање решења**

**Генерисање лексера**:

1. Начин:

Покретањем таргета *lexerGen* из *build.xml* фајла.

1. Начин:

Из командне линије, командом:

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

**Генерисање парсера:**

1. Начин:

Покретањем таргета parserGen из *build.xml* фајла.

1. Начин:

Из командне линије, командом:

java -cp lib/cup\_v10k.jar java\_cup.Main -destdir rs\ac\bg\etf\pp1 -dump\_states -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup >izlaz.out 2>izlaz.err

**Компајлирање:**

1. Начин:

Покретањем таргета *compile* из build.xml фајла.

1. Наичин:

test\program.mj test\program.obj (локални фајл MJTestParser.java)

**Испис бајткода:**

1. Начин:

Покретањем таргета *disasm* из build.xml фајла.

1. Начин:

test\program.obj (библиотека mj-runtime-1.1.jar класа disasm.class)

**Извршавање бајткода:**

1. Начин:

Покретањем таргета *runObj* из build.xml фајла.

1. Начин:

-debug test\program.obj (библиотека mj-runtime-1.1.jar класа Run.class)

**Опис приложених тест примера**

Тестови се налазе у тест фолдеру. Тест под називом *test301* тестира потребне функционалности.

**Опис новоуведених класа**

* SemanticAnalyzer.java – семантичка анализа и генерисање табеле симбола
* CodeGenerator.java – генерисање МЈ бајткода
* МЈParserTest.java – за компајлирање
* MyVariable.java – представља апстракцију променљиве и константе, користи се у семантичкој анализи како би се променљиве и константе убацивале у табелу симбола.
* MyMethod.java – представља апстракцију метода.
* MySymbolTableVisitor.java – проширује могућност исписа за bool тип компоненте при испису табеле симбола.
* MyCond.java – користи се у фази генерисања кода; у њој чувају вредности потребне за попуњавање адреса приликом скокова унапред