**Elektrotehnički fakultet**

**Univerzitet u Beogradu**



**Projekat iz predmeta Programski Prevodioci 1**

**Kompajler za programski jezik mikrojava**

Igor Stevanović 2018/0458

Beograd, Jun 2023.

### Opis zadatka:

Cilj projekta je da se napravi program odnosno “kompajler(compiler)” za školski programski jezik mikrojava, sastoji se iz sledećih celina:

1. Leksički analizator – Uloga leksičkog analizatora je da prihvati fajl sa izvornim kodom na jeziku mikrojava I podeli ga na tokene. Ovo se implementira pisanjem .flex specifikacije. Tokeni se vraćaju eksplicitnim pozivom metode next\_token()
2. Sintaksni analizator – Ovo je parser kompajlera, implementira se pisanjem gramatičkih specifikacija u .cup fajlu prema generatoru sintaksnih analizatora AST-CUP. Generator funkcioniše koristeći LALR(1) gramatiku.
3. Semantički analizator – U procesu provere semantičke korektnosti izvornog koda, pregledava se apstraktno sintaksno stablo (koje je stvoreno u prethodnom koraku) i kreira se tabela simbola. Za ovaj korak se koristi symboltable.jar.

1. Generator koda – Nakon što se izvrši semantička analiza, ponovo se pregledava apstraktno sintaksno stablo i koristeći prethodno generisanu tabelu simbola, generiše se bajtkod koji je kompatibilan sa MJVM

### Komande:

### Pokretanje svih delova programa se radi preko build.xml fajla selektovanjem određenih targeta u istom i pokretanjem glavne klase:

### Za lexer – build.xml – target:lexerGen

### Za parser – build.xml – target:compile

### Onda se pokreće MJParserTest.java koji daje izveštaj sintaksne I semantičke analize I generiše objektni fajl programa – program.mj ( u program.obj)

### Za generisanje koda – build.xml – target:runObj

### Test primeri(opis):

### AssignmentTest.mj – testira implementaciju DesignatorAssign odnosno dodele vrednosti promenljivi, konstanti, elemenata niza, kao I implementaciju inkrementiranja I dekrementiranja

### ReadPrintTest.mj – testira implementaciju učitavanja sa ulaza I ispisa na standardni izlaz

### SyntaxErrorTest.mj – pokazuje implementaciju oporavka od sintaksne greške, tj. izveštaja

### SemanticErrorTest.mj – pokazuje implementaciju izveštaja semantičke greške

### test301.mj – javni test dostupan na sajtu predmeta kao glavni za A nivo, pokazuje veliku većinu implementacije celog kompajlera

### test302.mj – javni test dostupan na sajtu predmeta kao glavni za B nivo, pokazuje veliku većinu implementacije celog kompajlera

### Opis klasa:

### SemanticAnalzyer.java – klasa za semanticku obradu

### CodeGenerator.java – klasa koja služi za generisanja bajtkoda

### SymbolTableVisitorExtended.java – pomoćna klasa za proširavanje ispisa tabele simbola

### TabExtended.java – pomoćna klasa koja proširuje početnu tabelu simbola sa bool tipom

### Utils.java – pomoćna klasa koja pomaže pri ispisu i koja kreira Struct Array tip za zadati tip.

### MethodManager.java – pomoćna klasa koja sadrži provere oko main methode. Nju nasledjuju klase MethodSyntaxParsingManager.java i MethodCodeGenerationManager.java

### ControlFlowSyntaxParsingManager,java – pomoćna klasa koja sadrži logiku koja je potrebna pri sintaksnom parsiranju kontrolnih tokova(if, while, map)

### DeclarationSyntaxParserManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri sintaksnom parsiranju deklaracija

### DesignatorStatementSyntaxParsingManager.java - pomoćna klasa koja sadrži logiku koja je potrebna pri sintaksnom parsiranju dodele, inkrementa i dekrementa

### ExprSyntaxParsingManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri sintaksnom parsiranju izraza

### MethodSyntaxParsingManager.java - pomoćna klasa koja sadrži logiku koja je potrebna pri sintaksnom parsiranju metoda, nasledjuje već pomenutu klasu MethodManager.java

### StatementSyntaxParsingManager.java - pomoćna klasa koja sadrži logiku koja je potrebna pri sintaksnom parsiranju read i print

### ControlFlowCodeGenerationManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri generisanju koda za kontrolne tokove(if, while, map)

### DesignatorStatementCodeGenerationManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri generisanju koda za pristup elementu niza ili matrice

### ExprCodeGenerationManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri generisanju koda za izraza

### MethodCodeGenerationManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri generisanju koda za metode, nasledjuje već pomenutu klasu MethodManager.java

### StatementCodeGenerationManager.java – pomoćna klasa koja sadrži logiku koja je potrebna pri generisanju koda za read i print