

# ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ

## Άσκηση 3<sup>η</sup>

Το παραδοτέο αυτό έχει χτιστεί πάνω από το 2<sup>ο</sup> παραδοτέο (δηλαδή κάνει και σημασιολογική ανάλυση πριν παράξει LLVM κώδικα).

### Δομή Παραδοτέου

- ✚ *Main.java*: Η ζητούμενη Main κλάση που περνά από parsing και σημασιολογικό έλεγχο τα αρχεία στα ορισματά γραμμής εντολής της και, αν είναι σωστά, παράγει ισοδύναμο LLVM κώδικα σε αρχεία με το ίδιο όνομα αλλά με *.ll* file extension.
- ✚ *SymbolTable/*: Πακέτο με κλάσεις που συνθέτουν το Symbol Table.
- ✚ *SemanticAnalysis/*: Πακέτο με κλάσεις που αφορούν την δημιουργία («γέμισμα») του Symbol Table και τον σημασιολογικό έλεγχο της εισόδου.
- ✚ *LLVMCodeGeneration/*: Πακέτο με κλάσεις που αφορούν την παραγωγή LLVM κώδικα:
  - *FileWriter.java*: Κλάση που γράφει γραμμικά (κάνοντας append) σε ένα αρχείο (χρησιμοποιείται ως γραμμικό buffer για την παραγωγή του κώδικα).
  - *LLVMNameGenerator.java*: Κλάση που επιστρέφει εγγυημένα μοναδικά ονόματα μεταβλητών και labels (με δυνατότητα reset των counters για κάθε ξεχωριστό «namespace» / συνάρτηση).
  - *LLVMCodeGeneratingVisitor.java*: Ο Visitor (3<sup>ος</sup> συνολικά) που παράγει τον LLVM κώδικα για ένα σημασιολογικά σωστό αρχείο χρησιμοποιώντας τις πληροφορίες του Symbol Table και κάνοντας visit το Syntax Tree του MiniJava προγράμματος εισόδου.
  - *LLVMCodeGenerating.java*: Namespace για συναρτήσεις που παράγουν LLVM κώδικα.
- ✚ *Util/*: Πακέτο με κλάσεις γενικού σκοπού όπως η *MyPair* και οι κλάσεις που χρησιμοποιούνται ως return και parameter types στους Visitors.
- ✚ *MiniJavaType/*: Πακέτο με μία κλάση και ένα enumeration που κωδικοποιούν τους τύπους της *MiniJava* και μεθόδους επί αυτών.
- ✚ *Resources/*: Εδώ πρέπει να βρίσκονται τα *.jar* αρχεία για τον Javacc και JTB.
- ✚ *MiniJava.jj*: Η δοσμένη γραμματική για την παραγωγή του parser.
- ✚ *Makefile*: το make cleaner διαγράφει και τα generated αρχεία-φακέλους, ενώ το make clean μόνο τα *.class* αρχεία
- ✚ *runInput.sh*: Bash script που ομαδοποιεί την μεταγλώττιση και εκτέλεση ενός MiniJava προγράμματος (MiniJava -> LLVM -> executable -> εκτέλεση)

Τρόπος εκτέλεσης: `make && java Main [file1] [file2] ... [fileN]`

Output: `file1.ll, file2.ll, ..., fileN.ll`