

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ

Άσκηση 2^η

Δομή Παραδοτέου

Το παραδοτέο αυτό έχει την εξής δομή:

- ✚ *Main.java*: Η ζητούμενη Main κλάση που περνά από parsing και σημασιολογικό έλεγχο τα αρχεία στα ορισματά γραμμής εντολής της και, αν είναι σωστά, εκτυπώνει τα ζητούμενα offsets (πεδίων και μη overriding μεθόδων) για τις κλάσεις των αρχείων εισόδου.
- ✚ *CreateSymbolTableVisitor.java*: Ο πρώτος Visitor που καλεί η Main ο οποίος δημιουργεί - για την ακρίβεια γεμίζει - το Symbol Table και πιάνει κάποια σημασιολογικά λάθη που μπορούν να πιαστούν χωρίς αυτό να είναι ολοκληρωμένο (πχ duplicate declarations).
- ✚ *SemanticCheckingVisitor.java*: Ο δεύτερος Visitor που καλεί η Main ο οποίος κάνει όλους τους υπόλοιπους σημασιολογικούς ελέγχους χρησιμοποιώντας το (πλεον πλήρες και σωστό) Symbol Table.
- ✚ *VisitorParameterInfo.java*, *ExtendedVisitorParameterInfo.java*: Κλάσεις που για το πέρασμα πληροφορίας μέσω παραμέτρου μεταξύ κλήσεων *visit()* στους Visitors.
- ✚ *VisitorReturnInfo.java*: Κλάση για την επιστροφή πληροφορίας μέσω τιμής επιστροφής κλήσεων *visit()* στους Visitors.
- ✚ *SemanticChecks.java*: Κλάση που περιέχει μόνο στατικές μεθόδους (ουσιαστικά namespace) που κάνουν χρήσιμους σημασιολογικούς ελέγχους (όπως τον έλεγχο για subtyping, etc) που χρησιμοποιούνται σε πολλά σημεία στον δεύτερο Visitor.
- ✚ *SemanticErrors.java*: Κλάση που επίσης περιέχει μόνο στατικές μεθόδους οι οποίες γυρνάνε κατάλληλο String error message ανάλογα το error. Κάθε συνάρτηση επιστρέφει το error μήνυμα σε ένα συγκεκριμένο semantic error δίνοντας πρώτα το context του error (σε ποια γραμμή, σε τι μέρος του κώδικα).
- ✚ *SymbolTable/*: Πακέτο με κλάσεις που συνθέτουν το Symbol Table για το Semantic Checking.
- ✚ *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 clean διαγράφει και τα generated αρχεία-φακέλους)

Τρόπος εκτέλεσης Parser και Semantic Checker:

```
make && java Main [input_file1] [input_file2] ... [input_fileN]
```

Να σημειωθεί ότι όπως και στον javac το όνομα της κλάσης με την μέθοδο *main()* πρέπει να είναι το ίδιο με το όνομα του αρχείου το ίδιο επιβάλλει και η Main κλάση.