

Compilers

Δεύτερη Σειρά Ασκήσεων (Γλώσσα : Java 7)

Περιεχόμενα :

1. Παραδοχές
 2. Παραδοτέα
 3. Γραμμές Λαθών
-

1. Παραδοχές

Ο έλεγχος γίνεται σε 3 στάδια . Ο πρώτος Visitor αποθηκεύει στο πρώτο επίπεδο του SymbolTable τα ονόματα των κλάσεων. Ο δεύτερος Visitor αποθηκεύει στο δεύτερο επίπεδο του SymbolTable Field και Methods στα scope των αντίστοιχων κλάσεων μέχρι και μεταβλητές των Methods . Ο τρίτος Visitor κάνει το type checking .

Κλάση MySymbolTable :

```
public class MySymbolTable {  
    private String classScope ;  
    private MySymbolTable parent ;  
    private Map<String, MySymbol> map ;
```

...

για την υλοποίηση του SymbolTable χρησιμοποιήθηκε chained δομή όπου κάθε table έχει ένα hashmap από τα ονόματα των κλάσεων/μεταβλητών/συναρτήσεων στα “κελιά ” τους . Υπάρχει επίσης και δείκτης σε γονέα table ώστε να μπορούμε να πάμε προς τα πίσω στα scopes σε περιπτώσεις ελέγχου Overriding και λοιπά .

Κάθε κελί είναι της μορφής :

```
public abstract class MySymbol {  
    protected String name ;  
    protected MySymbolTable myTable ;
```

...

όπου έχει το όνομα του κελιού και δείκτη στο scope του εάν πρόκειται για κλάση/συνάρτηση.

- Δεν έγιναν αλλαγές στη γραμματική .
- Ο “compiler” για κάθε πρόγραμμα σταματάει στο πρώτο λάθος που βρίσκει.

2.Παραδοτέα

Στο φάκελο input :

βρίσκονται τα αρχεία εισόδου που δώθηκαν .

Στο φάκελο mapping :

Βρίσκονται οι 3 κλάσεις που κληρονομούν την MySymbol : Cclass , Function και Var και αποτελούν τα “κελιά” του “πίνακα” MySymbolTable.

Στο φάκελο error :

Βρίσκεται η κλάση LineOfError που χρησιμοποιήθηκε για την εύρεση της γραμμής που βρίσκεται το λάθος .

+Makefile

+ οι 3 Visitors

+ λοιπά αρχεία και φάκελοι που δημιουργούνται από τα εργαλεία .

3. Γραμμές Λαθών

Επειδή η αναγραφή των γραμμών λαθών μπήκαν στον τέλος καθώς δεν γνώριζα οτι αυτό γίνεται εγκαίρως ενδέχεται να υπάρξει λάθος αναγραφή γραμμής .Στα παραδείγματα που δώθηκαν παρ' όλα αυτά οι ενδείξεις γραμμών λάθους ήταν σωστές .