

Σπίθας Ευάγγελος

Στην εργασία χρησιμοποιώ 2 visitors. Τον InformationStoringVisitor με τον οποίο αποθηκεύω καταλλήλες πληροφορίες για κάθε κλάση, μέθοδο και μεταβλητή και κάνω καποιους ελέγχους. Και τον TypeCheckVisitor ο οποίος κάνει όλους τους υπόλοιπους ελέγχους. Για την αποθήκευση των πληροφοριών χρησιμοποιώ 3 κλάσεις. Την ClassInfo, την MethodInfo και την VariableInfo.

InformationStoringVisitor

Στον πρώτο visitor αποθηκεύω όλες της πληροφορίες. Για κάθε κλάση που επισκέπτομαι δημιουργώ ένα Map στο οποίο φυλλάω της μεταβλητές και ένα στο οποίο φυλλάω όλες τις μεθόδους. Για κάθε μέθοδο δημιουργώ επίσης ένα map για τα ορίσματα και ένα για τις μεταβλητές. Κάθε μεταβλητή ή όρισμα που διαβάζω τα αποθηκεύω σε μία κλάση τύπου VariableInfo, στην οποία φυλάω το όνομα και τον τύπο της μεταβλητής.

Οι έλεγχοι που κάνω είναι όταν μία κλάση κληρονομεί μία άλλη. Σε αυτή τη περίπτωση κοιτάω αν υπάρχει ορισμένη η κλάση που κληρονομείται και αν ναι τότε για κάθε μέθοδο που έχει οριστεί σε κάποιο πρόγονο θα κάνω σύγκριση των ορισμάτων και τύπου επιστροφής.

TypeCheckVisitor

Με τον 2ο visitor κάνω όλους τους υπόλοιπους ελέγχους. Σε κάθε πρωτεύουσα έκφραση επιστρέφω τον τύπο της. Αν είναι identifier ψάχνω να βρω αν έχει οριστεί στη τρέχουσα μέθοδο ή κλάση και στη συνέχεια σε όλες τις γονεϊκές κλάσεις. Αν έχει οριστεί τότε με ανάλογο τρόπο παίρνω και τον τύπο της. Στις υπόλοιπες εκφράσεις ελέγχω αν οι τύποι των τελεστών είναι συμβατοί και επιστρέφω την κατάλληλη τιμή που επιστρέφει η έκφραση. Στην περίπτωση της κλήσης συναρτήσεων αρχικά θα υπολογίσω τον τύπο της έκφρασης αριστερά της τελείας. Έπειτα θα ελέγξω αν ο identifier έχει οριστεί για τον συγκεκριμένο τύπο και στη συνέχεια κάνω έλεγχο για τα ορίσματα. Ο έλεγχος των ορισμάτων θα γίνει με τη συνάρτηση acceptArgs. Στις αναθέσεις κάνω έλεγχο για το αν είναι ίδιου τύπου ο identifier με την έκφραση ενώ στα if και while statement ελέγχω για το αν η συνθήκη είναι boolean. Τέλος στο println έλεγχω αν η τιμή προς εκτύπωση είναι ακέραιος. Στις μεθόδους κοιτάω αν η εφραση επιστροφής είναι ίδιου τύπου με τον τύπο επιστροφής. Στις περιπτώσεις που πρόκειται για μη πρωτεύων τύπο θα πρέπει να γίνει έλεγχος για τον αν πρόκειται για υποτύπο, το οποίο υλοποιείται με την συνάρτηση isSubType της ClassInfo

Ο λόγος που επέλεξα LinkedHashMaps είναι επειδή θα πρέπει να γίνονται συνεχώς συγκρίσεις για τους τύπους των μεταβλητών. Με τα maps μπορώ να παίρνω τον τύπο κάθε μεταβλητής, με βάση το όνομα της, άμεσα. Επίσης αφού είναι LinkedHashMaps άμα περάσω τα στοιχεία τους σε μία λίστα θα είναι με τη σειρά εισαγωγής. Αυτό εξυπηρετεί για την εκτύπωση των offset που πρέπει να γίνεται με τη σειρά που βρίσκονται και συνεπώς με τη σειρά που διαβάστηκαν και εισάχθηκαν.

IR_GeneratorVisitor

Για την άσκηση χρησιμοποίησα έναν visitor, τον IR_GeneratorVisitor. Αυτός ο visitor αρχικά παράγει τους V_table με βάση τα offset που υπολογίζονται από τον InformationStoringVisitor, τον οποίο είχα υλοποιήσει στα πλαίσια της προηγούμενης άσκησης. Έπειτα θα παράξει κώδικα για κάθε συνάσταση.

Όταν γίνεται make θα παραχθεί ένας φάκελος με το όνομα outputs όπου θα μπουν όλα τα .ll αρχεία που θα δημιουργηθούν. Με την εντολή make clean θα διαγραφεί αυτός ο φάκελος.