Β. **Αξιολόγηση Λογισμικού**

Για την αξιολόγηση της πορείας του λογισμικού θα χρησιμοποιήσουμε μετρικές επιπέδου κλάσης, οι οποίες μας δίνουν πληροφορίες για:

* **το μέγεθος του κώδικα**

Η μέτρηση του μεγέθους του προγράμματος δε γραμμές κώδικα, μας δίνει πληροφορίες για το χρόνο και την προσπάθεια που έχει γίνει, προκειμένου το έργο να φτάσει στο εκάστοτε σημείο, στο οποίο λαμβάνεται η μέτρηση. Ενώ συνιστά τον πιο εύκολο τρόπο κοστολόγησης, δεν ανταποκρίνεται (πάντα) στην πραγματικότητα, κυρίως όσον αφορά στην ποιότητα, αλλά και τις συνθήκες μέσα από τις οποίες προέκυψε η τρέχουσα κατάσταση του κώδικα. Σίγουρα, οι γραμμές κώδικα, όντως αποτυπώνουν τον βαθμό προσπάθειας, όπως σε περιπτώσεις ελέγχου και εντοπισμού επίμονων σφαλμάτων. Από την άλλη μεριά, ένα σύνολο εντολών με δομές ελέγχου, σίγουρα χρειάζεται περισσότερο χρόνο υλοποίησης (σκέψη και συγγραφή) από ό,τι οι απλές εντολές εκτύπωσης της πορείας της εκτέλεσης του κώδικα. Οπότε κατά καιρούς χρησιμοποιήθηκαν εναλλακτικοί τρόποι υπολογισμού του μεγέθους του προγράμματος (συνδυασμός συναρτήσεων, τελεστών, μεταβλητών κλπ.). Αυτό που φαίνεται να ισχύει, τελικά, είναι το γεγονός ότι oi οι ίδιες πρακτικές δυσκολίες που συνηγορούν στην αποφυγή χρήσης του πλήθους γραμμών κώδικα, υπάρχουν και στις περιπτώσεις εναλλακτικών μονάδων μέτρησης.

Εμείς, στο πλαίσιο της εργασίας, θα χρησιμοποιήσουμε τη μονάδα μέτρησης **LOC (Lines of Code)**.

* **την συνεκτικότητα**

Η συνεκτικότητα μιας ενότητας (module), π.χ. κλάσης, του προγράμματός μας αντικατοπτρίζει τον βαθμό στον οποίο τα στοιχεία που βρίσκονται μέσα σε αυτή την ενότητα σχετίζονται μεταξύ τους. Κατά κανόνα, ενότητες με υψηλή συνεκτικότητα, είναι περισσότερο ισχυρές, πιο κατανοητές, πιο εύκολα συντηρήσιμες και επαναχρησιμοποιήσιμες. Αντίθετα, σε ενότητες που παρατηρείται χαμηλή συνεκτικότητα η κατανόηση και συντήρησή τους είναι δύσκολη, όπως και η επαναχρησιμοποίησή τους. Γενικά, κλάσεις με χαμηλή συνεκτικότητα, μπορούν να διαιρεθούν σε περισσότερες από μία κλάσεις, οι οποίες θα έχουν υψηλότερη συνεκτικότητα. Ως μονάδα μέτρησης, θα χρησιμοποιήσουμε την **LCOM (Lack of Cohesion of Methods)**, η οποία μας δείχνει αν έχω μεγάλη ή μικρή έλλειψη συνεκτικότητας μεταξύ των μεθόδων μιας κλάσης (Δύο μέθοδοι είναι συνεκτικές, εάν τα σύνολα των μελών δεδομένων που χρησιμοποιούν έχουν κοινά στοιχεία).

* **τη σύζευξη**

Η σύζευξη εκφράζει τον βαθμό αλληλεξάρτησης μεταξύ των ενοτήτων (π.χ. κλάσεων) ενός προγράμματος. Όσο λιγότερες είναι οι αλληλεξαρτήσεις μιας κλάσης, τόσο πιο εύκολος είναι ο έλεγχος, η τροποποίηση και η επαναχρησιμοποίησή της. Η σύζευξη είναι αντιστρόφως ανάλογη προς την συνεκτικότητα. Υψηλή συνεκτικότητα συνεπάγεται χαμηλή σύζευξη και το αντίστροφο. Θα χρησιμοποιήσουμε τη μονάδα μέτρησης **CBO (Coupling Between Objects)**. Αρχικά, η CBO υπολογιζόταν ως το άθροισμα του πλήθους των κλάσεων στις οποίες αναφέρεται μια κλάση αι των κλάσεων οι οποίες αναφέρονται σε αυτήν, ενώ τελευταία η πιο συνήθης ερμηνεία της αναφέρεται στον αριθμό των κλάσεων που παρέχουν σε μία κλάση Α τις απαραίτητες πληροφορίες (δεδομένα ή λειτουργίες) ώστε να ολοκληρώνονται οι μέθοδοί της.

* **Πολυπλοκότητα**

Για τον υπολογισμό της πολυπλοκότητας θα χρησιμοποιήσουμε τη μονάδα μέτρησης **WMC (Weighted Method Complexity)**. Χρησιμοποιούμε τον αριθμό των μεθόδων και την πολυπλοκότητά τους ως δείκτης της προσπάθειας που απαιτείται για την υλοποίηση και τον έλεγχο μιας κλάσης. Όσο αυξάνει ο αριθμός των μεθόδων μιας κλάσης τόσο πιο περίπλοκο γίνεται το δένδρο κληρονομικότητας και τόσο αυξάνει η συσχέτιση της κλάσης με μία συγκεκριμένη εφαρμογή, με συνέπεια να περιορίζεται η δυνατότητα επαναχρησιμοποίησης της κλάσης. Για τους λόγους αυτούς, η τιμή της μετρικής WMC θα πρέπει να παραμένει κατά το δυνατόν χαμηλή.