



Τμήμα Εφαρμοσμένης Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Τεχνολογία Συστημάτων Λογισμικού»

## 1<sup>η</sup> Εργασία 2023-2024

### «Εντοπισμός Code Smells ή/και Παραβιάσεων Αρχών Σχεδίασης, Refactorings, Git»

#### Θέμα

Η εργασία συνίσταται στον εντοπισμό οσμών κώδικα (code smells) ή/και παραβιάσεων Αρχών Σχεδίασης (π.χ. παραβίαση της Αρχής Μοναδικής Αρμοδιότητας) για επιλεγμένο έργο λογισμικού ανοικτού κώδικα, στην εφαρμογή αναδομήσεων για την αφαίρεση οσμών κώδικα ή τη συμμόρφωση με αρχές σχεδίασης, στην υποβολή του κώδικα που προκύπτει μετά από κάθε αναδόμηση σε ένα αποθετήριο λογισμικού και την παρακολούθηση μέσω μετρικών λογισμικού της μεταβολής της ποιότητας.

Το παραδοτέο της εργασίας είναι μία αναφορά σε μορφή pdf με βάση τα βήματα για την υλοποίηση της εργασίας που αναλύονται στη συνέχεια:

#### Βήμα 1ο.

Αντικείμενο της ανάλυσης θα είναι η τελευταία έκδοση ενός έργου ανοικτού κώδικα Java που μπορείτε να επιλέξετε ελεύθερα. Διατρέχοντας τον κώδικα είτε με τη χρήση κατάλληλων εργαλείων (π.χ. SmellDetectorMerger που έχει διατεθεί στην πλατφόρμα eclass) θα προβείτε στον εντοπισμό οσμών κώδικα (code smells) ή και παραβιάσεων αρχών σχεδίασης

**Επιμέρους παραδοτέο:** Στην αναφορά θα καταγράψετε 5 οσμές κώδικα ή/και παραβιάσεις Αρχών Σχεδίασης που εντοπίστηκαν μαζί με μια σύντομη περιγραφή της φύσης του κάθε προβλήματος και των επιπτώσεων που μπορεί να έχει στη συντηρησιμότητα του συστήματος.

#### Βήμα 2ο.

Μελετώντας τη σχετική βιβλιογραφία ή/και πηγές διαδικτύου θα εφαρμόσετε κατάλληλες αναδομήσεις στον πηγαίο κώδικα για την απομάκρυνση των οσμών που εντοπίστηκαν ή τη συμμόρφωση με τις Αρχές Σχεδίασης όπου εντοπίστηκαν παραβιάσεις. Για παράδειγμα, θα εφαρμόσετε την αναδόμηση 'Extract Method' (εξαγωγή μεθόδου) για την απομάκρυνση μιας οσμής τύπου 'Long Method' (μεγάλη μέθοδος) – η ίδια αναδόμηση μπορεί να εφαρμοστεί για τη συμμόρφωση με την Αρχή της Μοναδικής Αρμοδιότητας.

**Επιμέρους παραδοτέο:** Στην αναφορά θα περιγράψετε εν συντομία τις αναδομήσεις που εφαρμόσατε παραθέτοντας τα αντίστοιχα τμήματα κώδικα που τροποποιήθηκαν.

#### Βήμα 3ο.

Ο κώδικας που θα προκύπτει μετά την εφαρμογή κάθε αναδόμησης θα υποβάλλεται σε ένα αποθετήριο GitHub το οποίο θα δημιουργήσετε για το σκοπό αυτό – 5 αναδομήσεις -> 5 αναθεωρήσεις (revisions)

**Επιμέρους παραδοτέο:** Το URL του GitHub repository

#### Βήμα 4ο.

Με τη χρήση μετρικών λογισμικού θα αποτυπώσετε την εξέλιξη της ποιότητας του κώδικα κατά την εφαρμογή των αναδομήσεων. Η επιλογή των κατάλληλων μετρικών είναι στη ευχέρειά σας. Θα πρέπει να επιλέξετε μετρικές που αρμόζουν στο είδος των αναδομήσεων που εφαρμόσατε. Δεν είναι υποχρεωτικό κάθε αναδόμηση να επηρεάζει όλες τις επιλεγμένες μετρικές. Για τον υπολογισμό των μετρικών μπορείτε να αξιοποιήσετε οποιοδήποτε εργαλείο υπολογισμού τους (π.χ. για Java μπορείτε να χρησιμοποιήσετε το εργαλείο «Metric Calculator με GUI» που διατίθεται στο eclass).

**Επιμέρους παραδοτέο:** Πίνακας με τις τιμές των επιμέρους μετρικών για την αρχική έκδοση όσο και τις εκδόσεις του λογισμικού μετά την εφαρμογή κάθε αναδόμησης, ως εξής:



Project Name						
	Αρχική Έκδοση	Έκδοση 1 (μετά την 1 <sup>η</sup> αναδόμηση)	Έκδοση 2 (μετά τη 2 <sup>η</sup> αναδόμηση)	Έκδοση 3 (μετά την 3 <sup>η</sup> αναδόμηση)	Έκδοση 4 (μετά την 4 <sup>η</sup> αναδόμηση)	Έκδοση 5 (μετά την 5 <sup>η</sup> αναδόμηση)
Μετρική 1 (π.χ. LOC)						
Μετρική 2 (π.χ. CBO)						
Μετρική 3 (π.χ. LCOM)						
...						

Σχολιάστε τις μεταβολές των μετρικών σε σχέση με τις αναδομήσεις που πραγματοποιήσατε.