

Ομαδική Εργασία – Τελική Υποβολή

Στην τελική υποβολή καλείστε να υποβάλετε τρία αρχεία:

1. Ένα αρχείο **docx ή pdf** με την πλήρη ανάλυση και σχεδίαση του συστήματος που περιγράφει το σενάριο που σας έχει δοθεί από την αρχή του μαθήματος,
2. ένα **αρχείο Visual Paradigm** και
3. ένα **συμπιεσμένο αρχείο** με τον πηγαίο κώδικα όπως αυτός εξάγεται από το περιβάλλον ανάπτυξης λογισμικού που θα χρησιμοποιήσετε.

Η εργασία πρέπει να είναι πλήρης και περιεκτική. Η εργασία σας θα πρέπει να βασίζεται σε όσα κάναμε κάθε εβδομάδα. Το κείμενο σας θα πρέπει επιπλέον, να περιέχει μια ενότητα «Επιτελική Σύνοψη» στην αρχή, μία ενότητα «Εισαγωγή» και μια ενότητα «Συμπεράσματα» στο τέλος. Θα πρέπει στο τέλος να υπάρχουν δύο Παρατήματα Α και Β. Το Παράρτημα Α με τίτλο «Ομαδική Αυτό-αξιολόγηση» σας δίνεται προς συμπλήρωση στο τέλος του παρόντος εγγράφου. Στο Παράρτημα Β με τίτλο «Λειτουργία Ομάδας και Timesheets» θα αναφέρετε πώς εργάστηκε η ομάδα, εάν έκανε συναντήσεις (οπότε χρειάζονται και οι ατζέντες και τα πρακτικά) καθώς και τα timesheets καθενός μέλους της ομάδας **με το άθροισμα των συνολικών ωρών που εργάστηκε το κάθε μέλος**.

Η «Επιτελική Σύνοψη» είναι πάρα πολύ σημαντική. Πρόκειται για περίληψη της εργασίας και δεν πρέπει να είναι περιγραφή του πίνακα περιεχομένων, αλλά να περιέχει όλα τα βασικά στοιχεία και συμπεράσματα της μελέτης σας.

Θα πρέπει το τελικό σας κείμενο να είναι περιεκτικό και να μην απαιτείται γνώση άλλων πληροφοριών για να γίνει κατανοητό. Δεν θα πρέπει ωστόσο να επαναλάβετε τη θεωρία. Σκεφτείτε πως το κείμενο σας πάει για έγκριση στον πελάτη, ο οποίος δεν γνωρίζει τα περιεχόμενα του από πριν. Άρα πρέπει να εξηγήτε τι θα παρουσιάσετε (π.χ. στην εισαγωγή) και στο τέλος να συνοψίσετε τα συμπεράσματα σας.

Δεν υπάρχει περιορισμός στον αριθμό των σελίδων.

Ενδεικτικά περιεχόμενα:

Η εργασία θα πρέπει να περιλαμβάνει τα παρακάτω:

- 1) Να δημιουργηθεί
 - μια αίτηση συστήματος (system request)
 - μια μελέτη σκοπιμότητας για το νέο ΠΣ της ΕΗΔΕ.

** Για την αίτηση συστήματος θα πρέπει να περιοριστείτε στις πληροφορίες του case study. Ενώ για την μελέτη σκοπιμότητας όπου οι πληροφορίες είναι περιορισμένες (π.χ. απουσιάζουν οικονομικά στοιχεία), θα πρέπει να γίνουν εκτιμήσεις.*

2)

- Να καταγραφούν οι επιχειρηματικές διαδικασίες (ΕΔ) που εντοπίζονται στο case study.
- Να μοντελοποιηθούν οι διαδικασίες, σύμφωνα με όσα αναφέρει το case study για την υπάρχουσα κατάσταση (AS-IS), με χρήση UML διαγραμμάτων δραστηριοτήτων.
- Να μοντελοποιηθούν με τη χρήση UML διαγράμματος δραστηριοτήτων, οι νέες διαδικασίες, όπως θα διαμορφωθούν μετά και την εισαγωγή του νέου πληροφοριακού συστήματος (κατάσταση TO-BE) και να εξηγηθεί πώς αντιμετωπίστηκαν τα προβλήματα που υπήρχαν στις διαδικασίες.
- Να επιλεγεί μία από τις μεθόδους BPA, BPI, BPR και να αιτιολογηθεί η επιλογή.

** Δε χρειάζεται να δείξετε αντικείμενα και Πληροφοριακά Συστήματα στα διαγράμματα. Να περιοριστείτε στις πληροφορίες που δίνονται στην εκφώνηση. Εάν θα πρέπει οπωσδήποτε να κάνετε κάποια επιπλέον παραδοχή αυτή θα πρέπει να διατυπωθεί ρητά.*

3)

- Να γίνει επιλογή μεθόδων συλλογής πληροφοριών.
- Να γίνει καταγραφή των user stories και των epics, με βάση τις πληροφορίες που παρουσιάζονται στο case study. Στη συνέχεια να δημιουργήσετε το αντίστοιχο backlog λαμβάνοντας υπόψη τις εξαρτήσεις μεταξύ των user stories αλλά και των προτεραιοτήτων τους.
- Να γίνει καταγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων.

4) Να δημιουργηθεί το διάγραμμα UML περιπτώσεων χρήσης για το νέο ΠΣ.

5) Να γραφούν οι λεκτικές περιγραφές των ΠΧ του διαγράμματος χρήσεων που παρουσιάσατε προηγουμένως. Οι λεκτικές περιγραφές θα πρέπει να συνοδεύονται από ενδεικτικές (mock up) οθόνες.

** Κάθε μέλος της ομάδας θα αναλάβει να γράψει λεκτικές περιγραφές και mock up οθόνες για 2 ΠΧ του διαγράμματος. Αν το διάγραμμά σας περιλαμβάνει συνολικά λιγότερες ΠΧ από τα μέλη*2, τότε θα ανατεθούν αναλογικά (όσο είναι αυτό εφικτό) στα μέλη της ομάδας.*

6) Να καταγραφεί το μοντέλο του πεδίου του προβλήματος (domain model), απεικονίζοντας τις κλάσεις και τις σχέσεις μεταξύ τους. Θα πρέπει να χρησιμοποιηθεί η μέθοδος ανάλυσης κειμένου (text analysis) στις λεκτικές περιγραφές όλων των ΠΧ του συστήματος που παρουσιάσατε προηγουμένως. Στην απάντησή σας, εκτός από το τελικό μοντέλο θα πρέπει επίσης να συμπεριλάβετε τον τρόπο εργασίας σας καθώς και τα σημαντικότερα ενδιάμεσα αποτελέσματα σύμφωνα με τις αρχές της ανάλυσης κειμένου (δηλ. αρχική λίστα ουσιαστικών, ποια αποκλείστηκαν και γιατί κλπ). (Αναμένεται να περιέχονται στο domain model και όσα πεδία έχουν ανακαλυφθεί ήδη από τις λεκτικές περιγραφές).

7) Να δημιουργηθούν 2 διαγράμματα αντικειμένων που να δείχνουν βασικά στοιχεία του διαγράμματος κλάσεων.

8) Να αναθεωρηθούν οι λεκτικές περιγραφές των περιπτώσεων χρήσεων στο στάδιο της σχεδίασης, επισημαίνοντας τα σημεία που διαφοροποιήθηκαν με **highlight**.

9) Να δημιουργηθούν τα διαγράμματα ακολουθίας των περιπτώσεων χρήσης του σεναρίου. Τα διαγράμματα ακολουθίας θα πρέπει να περιλαμβάνουν τη βασική αλλά και τις εναλλακτικές ροές για τις αντίστοιχες περιπτώσεις χρήσης. Επιπλέον, θα πρέπει στα διαγράμματα αυτά να προσδιορίζετε λεπτομερώς τα μηνύματα που ανταλλάσσονται, ώστε να προκύπτει ποια από αυτά είναι λειτουργίες (με τις παραμέτρους τους, αν υπάρχουν) και ποια ενέργειες/εντολές του χρήστη.

10) Να εντοπιστούν τα στοιχεία που προέκυψαν από την κατασκευή των διαγραμμάτων ακολουθίας και την τελική μορφή των λεκτικών περιγραφών. Να εμπλουτιστεί το διάγραμμα κλάσεων με τα παραπάνω στοιχεία. Το εμπλουτισμένο διάγραμμα κλάσεων θα πρέπει να περιλαμβάνει λεπτομερώς οτιδήποτε μπορεί να εξαχθεί από το στάδιο της σχεδίασης.

11) Υλοποίηση.

- Βασιζόμενοι στα αποτελέσματα των προηγούμενων θεμάτων προχωρήστε στην υλοποίηση των κλάσεων του συστήματος με έμφαση στη λειτουργικότητα των συσχετίσεων των κλάσεων (ανταλλαγή μηνυμάτων).
- Για λόγους δοκιμής/επίδειξης της λειτουργικότητας του προγράμματος θα αναπτύξετε στη μέθοδο `main()` κατάλληλο πρόγραμμα οδήγησης (μέσω εντολών δημιουργίας αντικειμένων, συνδέσεων μεταξύ αυτών και κλήσης μεθόδων επί των αντικειμένων αυτών).
- Μπορείτε επίσης να υλοποιήσετε ένα απλό `text menu` το οποίο να εκτελεί βασικές λειτουργίες.
- Αυτό που μας ενδιαφέρει κατά βάση είναι η επίδειξη της ανταλλαγής μηνυμάτων και της χρήσης Καταλόγων!
- Δεν χρειάζεται να φτιάξετε Βάση Δεδομένων και GUI (είναι προαιρετικό).

Τέλος, εφοδιάστε τη `main` με κατάλληλες εντολές εκτύπωσης που θα πληροφορούν για την ενέργεια που μόλις εκτελέστηκε επιτυχώς (π.χ. «New appointment has been created»). Προσθέστε στις κλάσεις των αντικειμένων που χρησιμοποιήθηκαν στη `main`, μέθοδο `printData()` που θα εμφανίζει τις τιμές των ιδιοτήτων κάθε αντικειμένου. Στο τέλος της `main` καλέστε μέσω κατάλληλων `for loops` τη μέθοδο `printData()` στα αντικείμενα Καταλόγων.

Σημειώστε ότι εάν χρειαστεί να υλοποιήσετε οποιαδήποτε μέθοδο ή γνώρισμα (ιδιότητα, μεταβλητή) που δεν περιγράφεται στο διάγραμμα κλάσεων ή ακολουθίας, θα πρέπει να εξηγήσετε (με σύντομο και περιεκτικό σχόλιο εντός του κώδικα) γιατί προέκυψε αυτή η ανάγκη και γιατί δεν τεκμηριώνεται στα προηγούμενα διαγράμματα.

Στο κείμενο της εργασίας σας **θα πρέπει να αναφέρετε** εάν ο κώδικας είναι πλήρης και εκτελείται σωστά ή όχι, καθώς και οτιδήποτε άλλο θεωρείτε χρήσιμο.

Η κύρια κλάση της εφαρμογής είναι η `CarOps.java`, η οποία όταν εκτελείται (κλήση της `main` μεθόδου) εκτελεί την ακόλουθη λειτουργικότητα (ή κάποια αντίστοιχη με βάση τη δική σας σχεδίαση):

- Κατασκευάζει 3 αντικείμενα της κλάσης `ΕργασίαΕπισκευής` και τα τοποθετεί σε μια λίστα αντικειμένων τύπου `ΕργασίαΕπισκευής`. Τα αντικείμενα που δημιουργούνται έχουν τα εξής δεδομένα:
 - `ΕργασίαΕπισκευής1`("Αλλαγή λαδιών", 20 ευρώ)
 - `ΕργασίαΕπισκευής2`("Αλλαγή φίλτρου καμπίνας", 5 ευρώ)
 - `ΕργασίαΕπισκευής3`("Συντήρηση φρένων", 30 ευρώ)
- Κατασκευάζει 6 αντικείμενα της κλάσης `ΤύποςΑνταλλακτικού` και τα τοποθετεί σε μια λίστα αντικειμένων τύπου `ΤύποςΑνταλλακτικού`. Τα αντικείμενα που δημιουργούνται έχουν τα εξής δεδομένα:
 - `ΤύποςΑνταλλακτικού1`("Συσκευασία λαδιών 4lt", 30€)
 - `ΤύποςΑνταλλακτικού2`("Φίλτρο λαδιού", 20€)
 - `ΤύποςΑνταλλακτικού3`("Φίλτρο καμπίνας", 30€)
 - `ΤύποςΑνταλλακτικού4`("Τακάκι φρένων εμπρός τροχού", 5€)
 - `ΤύποςΑνταλλακτικού5`("Τακάκι φρένων πίσω τροχού", 5€)
 - `ΤύποςΑνταλλακτικού6`("Υγρό φρένων", 10 ευρώ)

- Κατασκευάζει 2 αντικείμενα της κλάσης Επισκευή και τα συνδέει με κατάλληλα αντικείμενα που αντιπροσωπεύουν τις εργασίες επισκευής και τις χρήσεις ανταλλακτικών που πραγματοποιήθηκαν στα πλαίσια των επισκευών. Στη συνέχεια τα προσθέτει σε μια λίστα αντικειμένων. Τα αντικείμενα που δημιουργούνται έχουν τα εξής δεδομένα:
 - ο Επισκευή1: περιλαμβάνει εκτέλεση των εργασιών ΕργασίαΕπισκευής1 και ΕργασίαΕπισκευής2 και χρήση ενός (1) τεμαχίου καθενός από τα ανταλλακτικά ΤύποςΑνταλλακτικού1, ΤύποςΑνταλλακτικού2 και ΤύποςΑνταλλακτικού3. Η διάρκεια της επισκευής είναι 1 μέρα.
 - ο Επισκευή2: περιλαμβάνει εκτέλεση της ΕργασίαΕπισκευής3 και χρήση 4 τεμαχίων του ΤύποςΑνταλλακτικού4, 4 τεμαχίων του ΤύποςΑνταλλακτικού5 και 1 τεμαχίου του ΤύποςΑνταλλακτικού6. Η διάρκεια της επισκευής είναι 2 μέρες.
- Εκτυπώνει στην οθόνη τα στοιχεία Εργασιών Επισκευής, Τύπων Ανταλλακτικών και Επισκευών, διατρέχοντας τις παραπάνω λίστες αντικειμένων με χρήση δομών επανάληψης. Συγκεκριμένα:
 - ο Για κάθε Εργασία Επισκευής εμφανίζει στην ίδια γραμμή την ονομασία και το κόστος της,
 - ο Για κάθε Τύπο Ανταλλακτικού εμφανίζει στην ίδια γραμμή την ονομασία και το κόστος ανά τεμάχιο,
 - ο Για κάθε Επισκευή εμφανίζει στην ίδια γραμμή τη διάρκεια επισκευής σε ημέρες και το συνολικό κόστος επισκευής. Θα πρέπει να σημειωθεί ότι το συνολικό κόστος επισκευής υπολογίζεται με χρήση της μεθόδου getTotalCost() που ανήκει στην κλάση Επισκευή. Η μέθοδος υπολογίζει το κόστος ως άθροισμα του κόστους εργασιών και του κόστους ανταλλακτικών που χρησιμοποιήθηκαν στην επισκευή. Το κόστος κάθε χρήσης ανταλλακτικού προκύπτει ως γινόμενο του κόστους του συγκεκριμένου τύπου ανταλλακτικού με την ποσότητα που χρησιμοποιήθηκε στην επισκευή. Όσον αφορά το κόστος εργασιών, προκύπτει αθροίζοντας τα επιμέρους κόστη των εργασιών που εκτελέστηκαν στα πλαίσια της επισκευής.

Σημειώσεις:

- Για τη δημιουργία των UML διαγραμμάτων θα πρέπει να χρησιμοποιήσετε το Visual Paradigm. Ωστόσο, θυμίζουμε πως τα διαγράμματα θα πρέπει να εισαχθούν στο κείμενο και ως εικόνες.
- Ως εργαλείο ανάπτυξης του κώδικα θα πρέπει να χρησιμοποιήσετε το Eclipse (www.eclipse.org). Θα πρέπει να παραδοθεί συμπίεσμένος ο φάκελος που περιέχει το αντίστοιχο project με τον πηγαίο κώδικα.
- Όλες οι παραδοχές που θα χρειαστεί να γίνουν θα πρέπει να καταγραφούν.

ΠΡΟΣΟΧΗ: Τυχόν αντιγραφή οδηγεί αυτόματα σε μηδενισμό ΟΛΩΝ των μελών ΟΛΩΝ των ομάδων, ανεξάρτητα από το ποιος αντέγραψε από ποιόν.

Παράρτημα Α: Ομαδική αυτο-αξιολόγηση

Θα πρέπει κάθε ομάδα να συμπληρώσει τους παρακάτω πίνακες και στοιχεία:

Αυτο-αξιολόγηση εργασίας:

- Βαθμός εργασίας ως προς την προσπάθεια (1-10):
- Βαθμός εργασίας ως προς το τελικό αποτέλεσμα (1-10):
- Τελικός βαθμός της ομαδικής εργασίας σας (1-10):

Επιχειρηματολογήστε για τους λόγους που θα πρέπει να πάρετε αυτόν τον βαθμό (έως 150 λέξεις – όχι παραπάνω).

Συγκριτική συνεισφορά κάθε μέλους στο τελικό αποτέλεσμα:

Ονοματεπώνυμο	Συνεισφορά (%)
Σύνολο	100%

Το άθροισμα όλων των επιμέρους (ατομικών) συνεισφορών θα πρέπει να είναι 100%. Συνεπώς πρόκειται για σχετική συνεισφορά. Εάν συμφωνήσετε στα ποσοστά, πιθανώς να υπάρξει ανάλογη διαφοροποίηση του βαθμού των μελών της ομάδας. Σε περίπτωση που διαφωνείτε ως προς τα σχετικά ποσοστά, μπορείτε να μην συμπληρώσετε τον πίνακα, οπότε και όλα τα μέλη της ομάδας θα πάρουν τον ίδιο βαθμό.

Παράρτημα Β: Λειτουργία Ομάδας και Timesheets

Θα συμπληρωθεί σύμφωνα με τις οδηγίες που δίνονται πιο πάνω.