ΕΡΓΑΣΙΑ ΟΝΤΟΚΕΝΤΡΙΚΟΥ ΠΡΟΓΡ/ΣΜΟΥ 2020-2021

(Παράδοση 4/6/2021)

Την τελευταία χρονιά λόγω των ειδικών συνθηκών covid και της οικονομικής κρίσης, αναπτύχθηκαν πρωτοβουλίες εθελοντών πολιτών για την υποστήριξη ευπαθών ομάδων ή οικογενειών που βρίσκονται σε δυσμενή οικονομική κατάσταση. Στα πλαίσια της εργασίας σας ζητείται να σχεδιάσετε και να υλοποιήσετε ένα απλοποιημένο **Σύστημα Οργάνωσης Εθελοντών-Δωρητών** (**donation**). Το ηλεκτρονικό σύστημα **donation** χρησιμοποιείται για τη συγκέντρωση ειδών από δωρητές (Donators) και αιτημάτων από Επωφελούμενους (Beneficiaries). Το σύστημα θα πρέπει να υποστηρίζει τη διαχείριση χρηστών, διαχείριση ειδών, και καταχώρηση και εξυπηρέτηση αιτημάτων για είδη. Η υλοποίηση μπορεί να γίνει είτε σε Java είτε σε C++ και θα μπορούσε να περιλαμβάνει ενδεικτικά τις κλάσεις που περιγράφονται παρακάτω.
Ο Admin του συστήματος ορίζει τα είδη που μπορούν να παρέχονται από τον οργανισμό. Ένας Δωρητής (Donator) μπορεί να προσφέρει είδη από τη συγκεκριμένη λίστα ειδών. Ένας Επωφελούμενος(Beneficiary) μπορεί να ζητήσει να του δοθούν είδη από τα παρεχόμενα από τον οργανισμό. Εκτός από τα υλικά είδη (πχ γάλα, ζάχαρη κλπ) παρέχονται και υπηρεσίες (πχ ιατρικές, νοσηλευτικές, φύλαξη κλπ). Για τα υλικά υπάρχει περιορισμός ποσοτήτων που εξαρτάται από το μέγεθος της οικογένειας του κάθε Beneficiary. Για τις υπηρεσίες δεν υπάρχει περιορισμός. Για το σκοπό αυτό διατηρούνται για κάθε Beneficiary, το σύνολο των ειδών που του έχουν δοθεί. Ο Admin μπορεί να μηδενίσει (reset) το σύνολο αυτό.
Δεν είναι υποχρεωτικό να ακολουθήσετε απόλυτα την προτεινόμενη δόμηση στην υλοποίησή σας, αλλά θα πρέπει να παρέχετε τις λειτουργίες που περιγράφονται, καθώς και την ιεραρχία των κλάσεων. Θα πρέπει να διατηρήσετε την ονοματολογία που δίνεται στην εκφώνηση. Επίσης μπορείτε να υλοποιήσετε επιπλέον πεδία, μεθόδους ή/και κλάσεις που κρίνετε σκόπιμο για την επίτευξη της ζητούμενης λειτουργικότητας.

Κλάση User: Αναπαριστά έναν χρήστη και χαρακτηρίζεται από το όνομά του name (string) και το κινητό τηλέφωνό του phone (string). Έχει ως υποκλάσεις τις Admin, Donator και Beneficiary. Δεν μπορεί να έχει στιγμιότυπα.

Κλάση Admin: Υποκλάση της User. Αναπαριστά ένα χρήστη που είναι ο διαχειριστής του συστήματος. Διαθέτει επιπλέον πεδίο boolean isAdmin με τιμή true.

Κλάση Donator: Υποκλάση της User. Αναπαριστά ένα χρήστη που είναι δωρητής. Έχει ως πεδίο το offersList, τύπου Offers, το οποίο αναπαριστά τη λίστα ειδών που επιθυμεί να προσφέρει. Οι μέθοδοι της Donator περικλείουν (wrap) μία ή περισσότερες από τις μεθόδους της κλάσης Offers (add, remove, commit…), ώστε ο Donator να μπορεί να ενημερώνει το μέλος offersList. Φτιάξτε wrappers για τις μεθόδους που χρησιμοποιείτε πιο συχνά.

Κλάση Beneficiary: Υποκλάση της User. Έχει ένα πεδίο noPersons το οποίο δηλώνει πόσα άτομα έχει η οικογένεια του και έχει default τιμή 1. Διαθέτει πεδίο receivedList, τύπου RequestDonationList, που αναπαριστά τη λίστα των ειδών και των ποσοτήτων που έχει **ήδη** λάβει. Επίσης διαθέτει πεδίο requestsList, τύπου Requests, που αναπαριστά την **τρέχουσα** λίστα των ειδών και των ποσοτήτων που ζητά να του δοθούν. Οι μέθοδοι της Beneficiary είναι wrappers για μία ή περισσότερες από τις μεθόδους των κλάσεων RequestDonationList και Requests (add, remove, commit…), ώστε ο Donator να μπορεί να ενημερώνει τα δύο μέλη του. Φτιάξτε wrappers για τις μεθόδους που χρησιμοποιείτε πιο συχνά.

Κλάση Entity: αντιπροσωπεύει ένα είδος δωρεάς και περιλαμβάνει πεδία name (string), description (string), και id (int), τα οποία είναι αντίστοιχα το όνομα, σύντομη περιγραφήκαι ο κωδικός του είδους αντίστοιχα. Η κλάση Entity δεν μπορεί να έχει στιγμιότυπα. Διαθέτει μέθοδο String getEntityInfo(), που επιστρέφει τις προηγούμενες πληροφορίες, *δήλωση* μεθόδου String getDetails() και μέθοδο String toString() (@override) που θα καλεί τις άλλες δύο, ώστε να τυπώνει κατάλληλα μορφοποιημένες τις συνολικά διαθέσιμες πληροφορίες ενός προϊόντος.

Κλάση Material: Κληρονομεί από την κλάση Entity και έχει επιπλέον πεδία level1, level2 και level3 (double) που αντιπροσωπεύουν τρία επίπεδα ποσοτήτων που μπορεί να δικαιούται κάποιος από ένα material. Level1 είναι η ποσότητα που δικαιούται ένα άτομο, level2 η ποσότητα που δικαιούνται 2-4 άτομα και level3 η ποσότητα που δικαιούνται 5 και περισσότερα άτομα. Αφού δημιουργηθεί αντικείμενο Material, οι τιμές των levels δεν αλλάζουν. Υλοποιεί την getDetails() η οποία επιστρέφει τις επιπλέον πληροφορίες για τα αντικείμενα Material (τα levels και το ότι το αντικείμενο είναι Material).

Κλάση Service: Κληρονομεί από την κλάση Entity. Υλοποιεί την getDetails() και επιστρέφει ως επιπλέον πληροφορία ότι το αντικείμενο είναι Service.

Κλάση RequestDonation: Περιλαμβάνει δύο πεδία, ένα Entity entity και ένα double quantity. Απεικονίζει το αίτημα ή την παροχή για ένα συγκεκριμένο Entity σε συγκεκριμένη ποσότητα quantity. Το id του entity είναι *κλειδί*, δηλαδή δύο αντικείμενα RequestDonation που αφορούν στο ίδιο entity θεωρούνται ταυτόσημα. Προαιρετικά, μπορείτε να το υλοποιήσετε αυτό στο πρόγραμμά σας κάνοντας implement το interface *Comparator*.

Κλάση Organization: Αντιστοιχεί στον οργανισμό που υποστηρίζει το σύστημα donation. Περιλαμβάνει πεδίο name τύπου string, πεδίο admin τύπου Admin, μια λίστα entityList με τα είδη (entity) που μπορούν να διανεμηθούν σε Beneficiaries, μια λίστα με τους δωρητές donatorList και μια λίστα με τους επωφελούμενους beneficiaryList. Επίσης διαθέτει πεδίο currentDonations τύπου RequestDonationList με τις διαθέσιμες προσφορές και τις ποσότητές τους. Περιλαμβάνει μεθόδους για τη διαχείριση και εμφάνιση των παραπάνω:

- setAdmin(), getAdmin(): για την καταχώρηση και την ανάκτηση του Admin του οργανισμού, αντίστοιχα.
- addEntity(): προσθέτει ένα νέο είδος (entity) που μπορεί να διακινηθεί από τον οργανισμό. Γίνεται χειρισμός εξαίρεσης αν υπάρχει ήδη.
- removeEntity(): διαγράφεται ένα Entity όταν παύει να διακινείται από τον οργανισμό. Τη διαγραφή την κάνει μόνο ο Admin.
- insertDonator(): προσθέτει έναν Donator στον οργανισμό. Γίνεται χειρισμός εξαίρεσης, αν υπάρχει ήδη.
- removeDonator(): αφαιρεί έναν Donator από τον οργανισμό.
- insertBeneficiary(): προσθέτει έναν Beneficiary στον οργανισμό. Γίνεται χειρισμός εξαίρεσης, αν υπάρχει ήδη.
- removeBeneficiary (): αφαιρεί έναν Beneficiary από τον οργανισμό.
- listEntities(): Εμφανίζει τις υπάρχουσες κατηγορίες των entity (material, services) και λίστα με όλα τα entity ανά συγκεκριμένη κατηγορία.
- listBeneficiaries(): εμφανίζει τους επωφελούμενους Beneficiaries και την τρέχουσα κατάσταση παροχών που έχουν λάβει ανά είδος.
- listDonators(): εμφανίζει τους δωρητές Donators.
- Προαιρετικά, wrapper μεθόδους για μεθόδους του αντικειμένου currentDonations.

Κλάση RequestDonationList: Αναπαριστά μια συλλογή από αντικείμενα RequestDonation και δίνει μεθόδους για την ενημέρωση της λίστας αυτής[1]. Έχει επομένως μια συλλογή rdEntities με αντικείμενα RequestDonation και μεθόδους:

- get(): Επιστρέφει ένα συγκεκριμένο RequestDonation από τη συλλογή, δίνοντας το id του Entity.
- add(): Μέθοδος η οποία τοποθετεί ένα συγκεκριμένο requestDonation στην rdEntities. Αν το requestDonation αφορά entity που ήδη υπάρχει σε κάποιο requestDonation της συλλογής, γίνεται ενημέρωση της ποσότητας του υπάρχοντος. Αν αφορά entity που δεν υπάρχει σε κάποιο requestDonation, δημιουργεί νέο αντικείμενο και το προσθέτει στη συλλογή. Αν αφορά entity που δεν υπάρχει στην entityList του οργανισμού, εγείρεται εξαίρεση.
- remove(): Αφαιρεί ένα RequestDonation από την rdEntities.
- modify(): Για ένα αντικείμενο στη συλλογή τροποποιεί το quantity.
- monitor(): Εμφανίζει το σύνολο των ειδών που βρίσκεται στη λίστα rdEntities, δηλαδή τα ονόματα των entity και τις ποσότητες που τους αντιστοιχούν.
- reset(): Αφαιρεί όλα τα αντικείμενα από τη λίστα.

Κλάση Requests. Υποκλάση της RequestDonationList. Αναπαριστά το σύνολο των ειδών (Material ή Services) που ζητά ο Beneficiary. Επιπλέον μέθοδοι:

- add(), modify(): Υπερκαλύπτουν τις μεθόδους της υπερκλάσης και τελικά τις καλούν, εφόσον επιτύχουν και οι δύο έλεγχοι παρακάτω: α) η ποσότητα του entity είναι διαθέσιμη στον οργανισμό και β) ο Beneficiary τη δικαιούται (validRequestDonation()). Αν δεν ισχύει το α) θα δημιουργηθεί κατάλληλη εξαίρεση, ενώ αν δεν ισχύει το β) θα εγερθεί κατάλληλη εξαίρεση διαφορετικού τύπου.
- validRequestDonation(): Ελέγχει αν η ποσότητα ενός RequestDonation είναι μέσα στο επιτρεπόμενο όριο με βάση των αριθμό μελών που έχει δηλώσει ο Beneficiary. Ο έλεγχος γίνεται ως εξής: Αν είναι Material τότε ελέγχεται με βάση το πεδίο noPersons σε ποιο Level ανήκει: level1, level2 ή level3. Ακολούθως ελέγχεται αν η συνολική ποσότητα που έχει ήδη λάβει από αυτό το είδος μαζί με την ποσότητα που ζητάει είναι μεγαλύτερη από το όριο. Αν ναι, τότε δεν εγκρίνεται η παροχή. Αν είναι Service δεν γίνεται έλεγχος.
- commit(): Ελέγχει εκ νέου (γιατί;) και εξυπηρετεί τα requestDonation αντικείμενα της rdEntities. Για κάθε requestDonation ελέγχονται τα α, β και εγείρεται κατάλληλη εξαίρεση αν κάποιος έλεγχος αποτύχει. Αν επιτύχουν οι έλεγχοι, αφαιρείται η ποσότητα από το συγκεκριμένο είδος στην currentDonations του οργανισμού, το συγκεκριμένο requestDonation γίνεται remove() από την rdEntities και add() στη receivedList του Beneficiary.

Κλάση Offers: Υποκλάση της RequestDonationList. Αναπαριστά το σύνολο των ειδών που προτίθεται να συνεισφέρει ο Donator. Επιπλέον μέθοδος:

- commit(): Ενημερώνει τα currentDonations του οργανισμού με τις προσφορές που περιέχονται στη λίστα rdEntities. Αν αυτό ολοκληρωθεί επιτυχώς, διαγράφει τα περιεχόμενα της λίστας rdEntities.

Κλάση Menu: αναπαριστά το μενού της εφαρμογής και περιλαμβάνει μεθόδους για την εκτύπωση των επιλογών, τον χειρισμό της εισόδου του χρήστη και την πλοήγηση στο μενού.

[1] Όπου απαιτείται συλλογή από αντικείμενα RequestDonation, κάθε στοιχείο της συλλογής θα αφορά διαφορετικό entity. Μια συλλογή μπορεί να υλοποιηθεί ως List ή Set.

Κλάσεις Εξαιρέσεων: θα πρέπει να υλοποιηθούν κατάλληλες κλάσεις εξαιρέσεων και να γίνεται χειρισμός τους σε περίπτωση μη επιτρεπτών ενεργειών, αντί να γίνονται εξαντλητικοί έλεγχοι if…else για την ορθή λειτουργία του προγράμματος. Ενδεικτικές περιπτώσεις εξαιρέσεων:

- Αν ζητηθεί από έναν Beneficiary ποσότητα από ένα entity που υπερβαίνει την προσφερόμενη ποσότητα ή αρνητική ποσότητα.
- Αν ζητηθεί από έναν Beneficiary ποσότητα από ένα entity που υπερβαίνει το όριο της ποσότητας που δικαιούται.
- Αν εισαχθεί από τον Admin ένα είδος που υπάρχει στον οργανισμό.
- Αν προστεθεί ένα entity με τον ίδιο κωδικό. Αν γίνει εισαγωγή μη έγκυρης επιλογής στο μενού κοκ.

Λειτουργικότητα Εφαρμογής

Διαμορφώστε κατάλληλο μενού σε command line όπου αρχικά θα ζητείται το κινητό τηλέφωνο. Το σύστημα με αυτόν το τρόπο ελέγχει και ταυτοποιεί υφιστάμενο χρήστη. Αν ο χρήστης δεν είναι εγγεγραμμένος, επομένως δεν είναι Admin ούτε βρίσκεται στη λίστα δωρητών donatorList ή επωφελούμενων beneficiaryList, τότε η εφαρμογή των ρωτά αν ενδιαφέρεται να εγγραφεί και του δίνεται η επιλογή να εγγραφεί ως Δωρητής ή ως Επωφελούμενος. Στη συνέχεια του εμφανίζεται το εξής μενού επιλογών, ανάλογα αν είναι Donator, Beneficiary ή Admin:

Donator: Εμφανίζει χαιρετισμό και τα στοιχεία του χρήστη, το όνομα του Οργανισμού και το όνομα Donator. Στη συνέχεια:

1. **Add Offer** (Πλοήγηση στις επιλογές για donation): Εμφανίζονται αριθμημένες οι δύο κατηγορίες (1. Material, 2. Services). Δίπλα σε κάθε είδος αναγράφεται σε παρένθεση η τρέχουσα ποσότητα του παρεχόμενου είδους ή υπηρεσίας στον οργανισμό.
   a. Επιλέγοντας μια κατηγορία, εμφανίζεται αριθμημένη λίστα με τα παρεχόμενα είδη (material) ή υπηρεσίες (services) της κατηγορίας αυτής.
      i. Επιλέγοντας ένα από τα παρεχόμενα είδη (material ή services) εμφανίζονται οι συνολικές πληροφορίες του και ερωτάται ο donator αν θέλει να το προσφέρει (y/n). Αν ναι, συμπληρώνεται η προσφερόμενη ποσότητα (material) ή προσφερόμενες ώρες (service) και καλείται η **add** ().
2. **Show Offers**: Καλούνται οι αντίστοιχες μέθοδοι της Offers για τον συγκεκριμένο Donator. Αρχικά εμφανίζεται αριθμημένη λίστα με τα RequestDonation που προσφέρει ο Donator, διαφορετικά μήνυμα ότι δεν έχει προσφορές αυτή τη στιγμή.
   a. Επιλέγοντας μια γραμμή παροχής RequestDonation ο χρήστης έχει τη δυνατότητα:
      i. Να διαγράψει την παροχή αυτή.
      ii. Να τροποποιήσει την παροχή αλλάζοντας την ποσότητά της.
   b. Επιλογή καθαρισμού όλων των παροχών.
   c. Commit (*η επιλογή υπάρχει ξανά και στο κύριο μενού του χρήστη*)
3. **Commit** (ολοκλήρωση καταχώρησης δωρεάς): Καλείται η commit() του Donator και τυπώνεται κατάλληλο μήνυμα.
4. **Back:** Επιστρέφει κάθε φορά ένα επίπεδο πιο πάνω στο μενού.
5. **Logout:** Αποσύνδεση χρήστη και ερώτημα για σύνδεση άλλου χρήστη.
6. **Exit:** Έξοδος από πρόγραμμα.

Beneficiary: Εμφανίζει χαιρετισμό, τα στοιχεία του χρήστη, το όνομα του Οργανισμού και το όνομα του Beneficiary.
Στη συνέχεια:

1. **Add Request** (Πλοήγηση στις επιλογές για request): όπως η Add Offer, αλλά για requests. Θα πρέπει να γίνει χειρισμός των εξαιρέσεων της add() και εκτύπωση μηνυμάτων.
2. **Show Requests:** όπως η Show Offers, αλλά για requests. Θα πρέπει να γίνει χειρισμός των εξαιρέσεων της modify() και εκτύπωση μηνυμάτων.
3. **Commit:** Καλείται η commit() του Beneficiary. Γίνεται χειρισμός εξαιρέσεων, ώστε να τυπωθούν κατάλληλα διαγνωστικά μηνύματα και να ενημερωθεί ο χρήστης ποια αιτήματά του έχουν ικανοποιηθεί επιτυχώς και ποια δεν πληρούν τις προϋποθέσεις.
4. **Back:** Επιστρέφει κάθε φορά ένα επίπεδο πιο πάνω στο μενού.
5. **Logout:** Αποσύνδεση χρήστη και ερώτημα για σύνδεση άλλου χρήστη.
6. **Exit:** Έξοδος από πρόγραμμα.

Admin: Εμφανίζει χαιρετισμό, τα στοιχεία του χρήστη και ένδειξη ότι είναι Admin. Στη συνέχεια:

1. **View:** Εμφανίζονται αριθμημένες οι δύο κατηγορίες (1. Material, 2. Services). Δίπλα σε κάθε είδος αναγράφεται σε παρένθεση η τρέχουσα ποσότητα του παρεχόμενου είδους ή υπηρεσίας στον οργανισμό.

    a. Επιλέγοντας μια κατηγορία, εμφανίζεται αριθμημένη λίστα με τα παρεχόμενα είδη (material) ή υπηρεσίες (services) της κατηγορίας αυτής.

        i. Επιλέγοντας ένα από τα παρεχόμενα είδη (material ή services) εμφανίζονται οι συνολικές πληροφορίες του.
2. **Monitor Organization:**
    a. **List Beneficiaries**: Καλείται η μέθοδος και εμφανίζεται αριθμημένη λίστα με τους Beneficiaries. Επιλέγοντας έναν Beneficiary:
        i. Εμφανίζεται η τρέχουσα κατάσταση των συνολικών παροχών που έχει πάρει.
        ii. Δίνεται η δυνατότητα να γίνει καθαρισμός του **receivedList** για τον συγκεκριμένο Beneficiary.
        iii. Δίνεται η δυνατότητα να διαγράψει τον Beneficiary.
    b. **List Donators**: Καλείται η μέθοδος και εμφανίζεται αριθμημένη λίστα με τους Donators. Επιλέγοντας έναν Donator:
        i. Εμφανίζεται η τρέχουσα κατάσταση των παροχών που προτίθεται να προσφέρει ο Donator.
        ii. Δίνεται η δυνατότητα να διαγράψει τον Donator.
    c. **Reset Beneficiaries Lists**: Για ΟΛΟΥΣ τους Beneficiaries γίνεται καθαρισμός του **receivedList του κάθε ενός.**
3. **Back**: Επιστρέφει κάθε φορά ένα επίπεδο πιο πάνω στο μενού.
4. **Logout:** Αποσύνδεση χρήστη και ερώτημα για σύνδεση άλλου χρήστη.
5. **Exit:** Έξοδος από πρόγραμμα.

Κλάση Main

Στη main αρχικά θα δημιουργήσετε ένα αντικείμενο Organization και θα εισάγετε 6 Entities: 3 Material (π.χ. milk, sugar, rice) και 3 Services (MedicalSupport, NurserySupport, BabySitting)), έτσι ώστε να είναι αντιπροσωπευτικά. Επίσης να δημιουργήσετε έναν Admin, δύο Beneficiaries και έναν Donator. Επίσης για έναν Beneficiary από τους δύο, να εισάγετε ένα αίτημα (request) με περισσότερα από δυο είδη από τα Entities (Material & Services). Αντίστοιχα εισάγετε παροχές από τον Donator.

Πέραν των παραπάνω παραδειγμάτων θα πρέπει για την επίδειξη της λειτουργικότητας του κώδικα να έχετε προετοιμάσει και δοκιμάσει ικανό πλήθος δεδομένων, ώστε να είναι διαθέσιμα σε κάθε εκτέλεση της εφαρμογής. Θα πρέπει τα δεδομένα που εισάγετε να δίνουν

τη δυνατότητα ελέγχου πολλών σεναρίων (διαγραφή request, τροποποίηση donation, προσπάθεια εισαγωγής ποσότητας σε request που δεν εγκρίνεται για το Beneficiary, Reset κοκ).

- **Bonus +20%:** Υλοποιήστε το πρόγραμμα και σε 2η γλώσσα (Java ή C++).
- ΔΙΕΥΚΡΙΝΗΣΗ: **Όσοι χρωστούν μόνο το project C++**, είτε διότι χρωστούν το παλιό μάθημα Οντοκεντρικός Προγραμματισμός ΙΙ είτε διότι έδωσαν το project Java άλλη χρονιά, θα πρέπει να υλοποιήσουν την εργασία αποκλειστικά σε C++.

ΑΞΙΟΛΟΓΗΣΗ

Η βαθμολόγηση πέρα από την σωστή εκτέλεση του προγράμματος όπως περιγράφεται παραπάνω θα βασιστεί στην σωστή εφαρμογή των εννοιών οντοκεντρικού προγραμματισμού που διδάσκονται στο μάθημα:

1. Σωστή δήλωση των κλάσεων και ιεράρχησή τους όπως ζητείται.
2. Τήρηση της αρχής της απόκρυψης και της ενθυλάκωσης: Αποφυγή χρήσεως public όπου δεν είναι απαραίτητο και χρήση getters και setters για πρόσβαση στα πεδία των κλάσεων.
3. Υπερκάλυψη μεθόδων και πολυμορφική χρήση τους όπου χρειάζεται.
4. Χρήση Static μεταβλητών/μεθόδων για γενική πληροφορία που αφορά κάθε κλάση.
5. Έμφαση στην αντικειμενοστρέφεια και την δυνατότητα επαναχρησιμοποίησης κώδικα.
6. Λίστες / Πίνακες
7. Ποιότητα κώδικα, αναγνωσιμότητα και επαρκής σχολιασμός
8. Η έξοδος του προγράμματος να είναι καλά μορφοποιημένη και εύκολα ερμηνεύσιμη από τον χρήστη

ΠΑΡΑΔΟΣΗ

- Οι εργασίες θα πρέπει να παραδοθούν στο e-Class μέχρι και την Παρασκευή 4/6/2021 (23:59).
- Οι εργασίες είναι ομαδικές (τουλάχιστον 2 και το πολύ 3 άτομα).
- Για κάθε εργασία παραδίδετε ***αναφορά*** που περιλαμβάνει:
  1. Τα στοιχεία (ΟΝΟΜΑΤΑ - ΑΜ - email) των μελών της ομάδας
  2. Σύνδεσμο προς προς τον κώδικα που έχετε υλοποιήσει (σε κάποια file sharing υπηρεσία ή private code repo), ώστε να ληφθεί υπόψη.
  3. Τον κώδικα στο τέλος της αναφοράς, ανά αρχείο, με κατάλληλη μορφοποίηση (font 11, μονό διάστιχο) Σε περίπτωση διαφοράς, ως οριστικός θα ληφθεί υπόψη ο κώδικας στην αναφορά. Χωρίς τα (2) και (3) η εργασία δεν εξετάζεται.
  4. Διάγραμμα κλάσεων (UML).
  5. Σύντομη περιγραφή της υλοποίησης σας με τεκμηρίωση των σχεδιαστικών αποφάσεων και της υλοποίησης για κάθε κλάση.

ΔΙΑΔΙΚΑΣΙΑ ΕΞΕΤΑΣΗΣ

Η διαδικασία εξέτασης θα ανακοινωθεί.

Καλή Επιτυχία!