

## Ατομικές Εργασίες Ακαδημαϊκού Έτους 2025-2026

### Εργασία 1<sup>η</sup>: PamakBook

Καλείστε να μοντελοποιήσετε σε αντικειμενοστρεφή κώδικα Java τις οντότητες που απαιτούνται για την υλοποίηση του κάτωθι απλοποιημένου συστήματος κοινωνικής δικτύωσης (PamakBook), καθώς και τις σχέσεις και τις αλληλεπιδράσεις μεταξύ τους. Το δίκτυο θα παρέχει και λειτουργίες που μπορούν δυνητικά να συμβάλλουν στον εντοπισμό διαύλων μετάδοσης ενός ιού σε μια πανδημία, υποθέτοντας ότι δύο 'φίλοι' σε ένα κοινωνικό δίκτυο ενδεχομένως έχουν και αυξημένες κοινωνικές επαφές.

Ένας χρήστης **User** του δημοφιλούς κοινωνικού δικτύου PamakBook χαρακτηρίζεται από το όνομά του και το Email του (με απαιτούμενη μορφή icsxxxxx@uom.edu.gr, ή iisxxxxx@uom.edu.gr, ή daixxxxx@uom.edu.gr – ο αριθμός μητρώου μπορεί να έχει 3, 4 ή 5 ψηφία). Ο κάθε χρήστης διατηρεί μία λίστα με τους άλλους χρήστες του κοινωνικού δικτύου, με τους οποίους είναι «φίλοι». Διατηρεί μία ακόμη λίστα με τις ομάδες (Groups) στις οποίες είναι εγγεγραμμένος.

Κάθε χρήστης (**User**) παρέχει δημόσια τις εξής λειτουργίες-μεθόδους:

- Μέθοδο η οποία δέχεται ως παράμετρο έναν άλλο χρήστη και επιστρέφει true ή false ανάλογα με το αν είναι φίλοι ή όχι. Κάθε χρήστης ΔΕΝ είναι φίλος με τον εαυτό του.
- Μέθοδο η οποία προσθέτει στην λίστα των φίλων έναν άλλο χρήστη. Αυτό γίνεται αφού προηγηθεί έλεγχος για το αν πρόκειται για το ίδιο πρόσωπο ή για χρήστη που είναι ήδη φίλος.
- Μέθοδο η οποία δέχεται ως παράμετρο έναν άλλο χρήστη και επιστρέφει μια λίστα με τους κοινούς φίλους των δύο χρηστών
- Μέθοδο η οποία τυπώνει τους φίλους του χρήστη
- Μέθοδο η οποία τυπώνει τα Groups στα οποία έχει γραφτεί ο χρήστης.
- Μέθοδο εντοπισμού πιθανών μεταδόσεων ιού, η οποία να επιστρέφει μια λίστα με όλους τους φίλους του χρήστη καθώς και όλους τους φίλους αυτών. Στο παράδειγμα εκτέλεσης η λίστα φαίνεται αλφαβητικά ταξινομημένη, αλλά αυτό είναι *προαιρετικό*).

Μια ομάδα (**Group**) χαρακτηρίζεται από το όνομά της, την περιγραφή της και τη λίστα με τα μέλη της. Κάθε ομάδα θα πρέπει να παρέχει δημόσια:

- Μέθοδο η οποία να δέχεται ως παράμετρο έναν χρήστη και να απαντά (true/false) αν είναι μέλος της ομάδας ή όχι.
- Μέθοδο η οποία να δέχεται ως παράμετρο έναν χρήστη και αν δεν είναι ήδη μέλος, να τον προσθέτει στα μέλη της ομάδας.
- Μέθοδο η οποία να τυπώνει τα μέλη της ομάδας

Μια ειδική κατηγορία ομάδας είναι η 'κλειστή' ομάδα (**ClosedGroup**) στην οποία ένας χρήστης μπορεί να εγγραφεί μόνο εφόσον είναι 'φίλος' με κάποιο ήδη υπάρχον μέλος της ομάδας (ο περιορισμός αυτός δεν ισχύει για το πρώτο μέλος μιας κλειστής ομάδας).

- Η μέθοδος που προσθέτει έναν χρήστη στα μέλη μιας κλειστής ομάδας θα πρέπει να ελέγχει αν ο υπό εγγραφή χρήστης είναι φίλος με κάποιο από τα μέλη της ομάδας. Αν ναι, να τον προσθέτει στα μέλη της ομάδας. Αν όχι, να εκτυπώνει μήνυμα αδυναμίας εγγραφής.

Να φτιάξετε μία ξεχωριστή κλάση Main με μέθοδο main στην οποία να κατασκευάσετε τα παρακάτω αντικείμενα και να εκτελέσετε τις κατάλληλες εντολές έτσι ώστε να έχετε στην κονσόλα το ίδιο output όπως παρακάτω:

```
User u1 = new User("Makis", "iis2598@uom.edu.gr");
User u2 = new User("Petros", "ics2524@uom.edu.gr");
User u3 = new User("Maria", "iis2512@uom.edu.gr");
User u4 = new User("Gianna", "iis25133@uom.edu.gr");
User u5 = new User("Nikos", "dai1758@uom.edu.gr");
User u6 = new User("Babis", "ics25104@uom.edu.gr");
User u7 = new User("Stella", "dai1827@uom.edu.gr");
User u8 = new User("Eleni", "ics2586@gmail.com");
```



(Η προσπάθεια δημιουργίας του τελευταίου χρήστη θα πρέπει να αποτύχει παράγοντας ένα μήνυμα της μορφής)

CONSOLE:

User Eleni has not been created. Email format is not acceptable.

```
Group g1 = new Group("WebGurus","A group for web passionates");
ClosedGroup g2 = new ClosedGroup("ExamSolutions","Solutions to common exam
questions");
```

CONSOLE:

Makis and Petros are now friends!

Makis and Nikos are now friends!

Nikos and Babis are now friends!

Maria and Gianna are now friends!

Maria and Petros are now friends!

Gianna and Babis are now friends!

Nikos and Maria are now friends!

Makis and Babis are now friends!

Nikos and Petros are now friends!

Stella and Makis are now friends!

\*\*\*\*\*

Common friends of Nikos and Gianna

\*\*\*\*\*

1: Babis

2: Maria

-----

\*\*\*\*\*

Common friends of Makis and Nikos

\*\*\*\*\*

1: Babis

2: Petros

-----

\*\*\*\*\*

Friends of Makis

\*\*\*\*\*

1: Babis

2: Nikos

3: Petros

4: Stella

-----

\*\*\*\*\*

Friends of Maria

\*\*\*\*\*

1: Gianna

2: Nikos

3: Petros

-----

Gianna has been successfully enrolled in group WebGurus

Maria has been successfully enrolled in group WebGurus

Petros has been successfully enrolled in group WebGurus

Gianna has been successfully enrolled in group ExamSolutions

FAILED: Nikos cannot be enrolled in group ExamSolutions

Babis has been successfully enrolled in group ExamSolutions

Nikos has been successfully enrolled in group ExamSolutions

\*\*\*\*\*

Groups that Gianna has been enrolled in

\*\*\*\*\*

1: WebGurus

2: ExamSolutions

-----

\*\*\*\*\*

Members of group WebGurus



\*\*\*\*\*

1: Gianna  
2: Maria  
3: Petros

-----

\*\*\*\*\*

Members of group ExamSolutions

\*\*\*\*\*

1: Babis  
2: Gianna  
3: Nikos

-----

\*\*\*\*\*

Gianna has been infected. The following users have to be tested

\*\*\*\*\*

Babis  
Makis  
Maria  
Nikos  
Petros

-----

(όπου υπάρχει εκτύπωση ονομάτων ή ομάδων, η σειρά εμφάνισης δεν παίζει ρόλο. Κάθε όνομα θα πρέπει ωστόσο να μην εμφανίζεται περισσότερες από μία φορές)

Τα προγράμματα θα αξιολογηθούν με βάση τα παρακάτω κριτήρια:

- **Αντικειμενοστρέφεια** – Περιλαμβάνει το πρόγραμμα τις απαιτούμενες κλάσεις και σχέσεις μεταξύ τους για τη μοντελοποίηση των οντοτήτων που περιγράφονται;
- **Λειτουργικότητα** – Έχουν υλοποιηθεί σωστά οι μέθοδοι που ζητούνται; Εκτελείται ορθά η μέθοδος main();
- **Γενική εντύπωση** – Είναι το πρόγραμμα σωστά δομημένο?
- **Σφάλματα** – μεταγλωττίζεται το πρόγραμμα? Υπάρχουν προφανή σφάλματα?
- **Εσωτερική τεκμηρίωση** – Είναι εύκολα κατανοητός ο κώδικας? Χρησιμοποιείτε κατάλληλα ονόματα κλάσεων/ιδιοτήτων/μεθόδων? Υπάρχουν σχόλια (στα σημεία που απαιτούνται)?
- **Αναγνωσιμότητα** – Χρησιμοποιείται στοίχιση που να ανακλά τη δομή του κώδικα? Είναι εύκολη η ανάγνωση του κώδικα? Υπάρχουν κενά που να διαχωρίζουν τα κύρια τμήματα του κώδικα?

**Σημείωση περί αντιγραφής:** Η συγκεκριμένη εργασία εμπίπτει στις ατομικές εργασίες του μαθήματος. Κατά συνέπεια θεωρείται ότι είναι **προϊόν ατομικής προσπάθειας**. Για τον λόγο αυτό, εργασίες που θα παρουσιάζουν ενδείξεις αντιγραφής τμημάτων κώδικα θα μηδενίζονται (για όλους τους εμπλεκόμενους).

**Σημείωση περί προθεσμίας και τρόπου υποβολής:** Εργασίες που θα υποβληθούν μετά την προθεσμία δεν θα γίνουν δεκτές. Εργασίες που θα υποβληθούν εκτός της πλατφόρμας eClass δεν θα γίνουν δεκτές.

### Οδηγίες Υποβολής

Υποβάλλετε συμπιεσμένο ολόκληρο το Eclipse project στην αντίστοιχη εργασία του eClass. Το όνομα του Eclipse Project θα πρέπει να έχει τη μορφή:

**Erwnymo\_Onoma\_ArithmosErgasias**

π.χ. **Papadopoulos\_Nikolaos\_1**