

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον
παγκόσμιο ιστό»

«αριθμός άσκησης»	Δημιουργία βασικών κλάσεων
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Φιλίππαίος Γεώργιος Π20251
	Στεφανουδάκης Μιχάλης Π20250
	Φιλιππίδης Σωτήρης Π21243
Ημερομηνία παράδοσης	



Εκφώνηση της άσκησης

Σε αυτή την άσκηση θα δημιουργήσετε ένα Java project το οποίο θα περιλαμβάνει τις κύριες κλάσεις που θα χρησιμοποιηθούν για την εφαρμογή του βαθμολογίου που περιγράφεται στην τελική εργασία, καθώς και τη βασική λειτουργικότητά τους.

Αναλυτικά Βήματα:

0. (προκαταρκτικά βήματα) Έλεγχος εγκατάστασης και καλής λειτουργίας περιβάλλοντος – εκκίνηση project

0.1. Έλεγχος εγκατάστασης jdk, έλεγχος εγκατάστασης eclipse (ή άλλου IDE της επιλογής σας), έλεγχος ενσωμάτωσης java documentation, έλεγχος δημιουργίας και εκτέλεσης ενός απλού προγράμματος HelloWorld.

0.2. Δημιουργία ενός Java Project με τίτλο JavaBasics.

0.3. Δημιουργία ενός πακέτου με τίτλο mainpackage.

1. Διάγραμμα κλάσεων

1.1. Σχεδιάστε το διάγραμμα κλάσεων (class diagram) στο οποίο θα αναλύονται όλες οι κλάσεις, οι σχέσεις και οι λειτουργίες τους.

(Υπόδειξη: Δημιουργήστε τη βασική κλάση Users από την οποία θα κληρωνόμουν οι κλάσεις Students, Professors, Secretaries. Δημιουργήστε τις κλάσεις Courses και Grades και δημιουργήστε τις κατάλληλες συνθέσεις μεταξύ των κλάσεων. Υλοποιήστε όλες τις απαραίτητες λειτουργίες (μεθόδους) των κλάσεων.

2. Δημιουργία βασικών κλάσεων

2.1. Δημιουργία μίας κλάσης **Users**, η οποία θα περιλαμβάνει τα βασικά χαρακτηριστικά κάθε κατηγορίας χρήστη της εφαρμογής.

2.1.1. Χαρακτηριστικά: όνομα χρήστη (username), όνομα (name), επίθετο (surname), τμήμα (department) και τέλος έναν μετρητή χρηστών (usersCounter) ο οποίος θα ξεκινά από την τιμή 0 και θα αυξάνεται κατά ένα, όταν θα καλείται ο constructor της κλάσης (υπόδειξη: ποιος προσδιοριστής πρέπει να χρησιμοποιηθεί για τη μεταβλητή αυτή;)

2.1.2. Μέθοδοι: Τυπικοί getters και setters για όλες τις μεταβλητές (υπόδειξη: είναι δυνατό να δημιουργηθούν για όλες τις μεταβλητές;)

2.1.3. Δημιουργία τυπικού constructor για την κλάση Users.

3. Δημιουργία υποκλάσης **Students**, η οποία θα χρησιμοποιεί τα χαρακτηριστικά και τις μεθόδους της κλάσης Users αλλά επιπλέον:

3.1. Χαρακτηριστικά: Επιπρόσθετα θα χρησιμοποιεί το χαρακτηριστικό αριθμός μητρώου (registrationNumber). Το χαρακτηριστικό αυτό θα δίδεται μία φορά για κάθε αντικείμενο Student και στη συνέχεια δεν θα μπορεί να αλλάξει.

3.2. Ο constructor θα πρέπει να τροποποιηθεί κατάλληλα ώστε να περιλαμβάνει τον αριθμό μητρώου, τον οποίο θα αρχικοποιεί τη στιγμή της δημιουργίας κάθε αντικειμένου τύπου Student (Υπόδειξη: κάνετε override τον constructor της υπερκλάσης Users)

3.3. Ποια επιπλέον χαρακτηριστικά και ποιες νέες μεθόδους χρειάζεται η κλάση Students; Να δημιουργηθούν αντίστοιχα.



4. Δημιουργία υποκλάσης **Professors**. Ποια επιπλέον χαρακτηριστικά και ποιες νέες μεθόδους χρειάζεται η κλάση **Professors**; Να δημιουργηθούν αντίστοιχα.

5. Δημιουργία υποκλάσης **Secretaries**. Ποια επιπλέον χαρακτηριστικά και ποιες νέες μεθόδους χρειάζεται η κλάση **Professors**; Να δημιουργηθούν αντίστοιχα.

6. Δημιουργία των κλάσεων **Courses** (για την περιγραφή των μαθημάτων) και **Grades** (για την περιγραφή των βαθμολογιών των φοιτητών ανά μάθημα). Για κάθε κλάση να ορίσετε τα απαραίτητα χαρακτηριστικά και μεθόδους που απαιτούνται.

7. Δοκιμή και τεκμηρίωση των βασικών κλάσεων, E/E

7.1. Δημιουργήστε μία κλάση με όνομα **CreateUsers** η οποία θα περιλαμβάνει τη συνάρτηση **main()**. Μέσω της κλάσης αυτής θα δημιουργήσετε αντικείμενα τύπου **Users**, **Students**, **Professors**, **Secretaries**. Δημιουργήστε ένα αντικείμενο από κάθε κλάση και χρησιμοποιείστε τις μεθόδους που έχετε δημιουργήσει. Χρησιμοποιείστε ένα αντικείμενο τύπου **Scanner** ώστε να δημιουργήσετε ένα αντικείμενο τύπου **Students** λαμβάνοντας τα ορίσματα του **constructor** από το πληκτρολόγιο

7.2. Με τη χρήση ενδεικτικών μηνυμάτων, δείξτε τη δημιουργία και τα χαρακτηριστικά των αντικειμένων.

7.3. Προσθέστε κατάλληλη τεκμηρίωση (**documentation**) για τις κλάσεις **Users**, **Students**, **Professors**, **Secretaries** και για όλες μεθόδους απαιτείται.

8. Έλεγχος εισόδου και χειρισμός εξαιρέσεων

8.1. Κατά τη δημιουργία των αντικειμένων **Students** θα πρέπει να ελέγξετε ότι το πεδίο Αριθμός Μητρώου είναι αριθμός (ακέραιος), ενώ όλα τα άλλα πεδία είναι αλφαριθμητικά (**Strings**). Χρησιμοποιείστε **exceptions** με τη βοήθεια των εντολών **try...catch**.

8.2. Δημιουργήστε ένα νέο **exception** για να χειριστείτε την παραπάνω περίπτωση.

8.3. Προσθέστε κατάλληλη τεκμηρίωση (**documentation**) όπου απαιτείται.

9. Δημιουργία αντικειμένων με τη χρήση αρχείου εισόδου

9.1. Δημιουργήστε ένα **txt** αρχείο το οποίο περιλαμβάνει τα χαρακτηριστικά αντικειμένων τύπου **Students**, έναν **student** ανά γραμμή. Τα διαχωριστικά μεταξύ των χαρακτηριστικών μπορεί να είναι κενά, **tab**, ή κάποιος ειδικός χαρακτήρας π.χ. **“,”** ή **“|”**).

9.2. Δημιουργήστε τα αντίστοιχα αντικείμενα τύπου **Students**, διαβάζοντας τα χαρακτηριστικά τους από αυτό το αρχείο.

9.3. Χρησιμοποιείστε το μηχανισμό εξαιρέσεων για να ελέγξετε για πιθανά σφάλματα ανάγνωσης, εγγραφής, και σωστού τύπου των χαρακτηριστικών.

10. Υλοποίηση επιπλέον λειτουργικότητας

(*Σημείωση: Δεν είναι αναγκαίο σε αυτή τη φάση να υλοποιηθούν πραγματικά οι λειτουργίες που περιγράφονται στα βήματα 9.2 και 9.3. Μπορείτε σε αυτή τη φάση να ορίσετε **abstract** μεθόδους ή η «υλοποίηση» να γίνει με την εμφάνιση απλών μηνυμάτων και όχι με πραγματική υλοποίηση της λειτουργίας, π.χ. «Η λίστα των φοιτητών για το μάθημα X δημιουργήθηκε».)*

10.1. Επεκτείνετε τη λειτουργικότητα της κλάσης **Secretaries** ώστε η γραμματεία να:

10.1.1. Δημιουργεί εγγραφές καθηγητών, φοιτητών και μαθημάτων.

10.1.2. Αναθέτει τους καθηγητές στα μαθήματα.

10.1.3. Δημιουργεί τις σχετικές λίστες των φοιτητών προς βαθμολόγηση.

10.2. Επεκτείνετε τη λειτουργικότητα της κλάσης **Students** ώστε ο κάθε φοιτητής να:

10.2.1. Βλέπει τις βαθμολογίες που του αντιστοιχούν

10.3. Επεκτείνετε τη λειτουργικότητα της κλάσης **Professors** ώστε ο κάθε καθηγητής να:

10.3.1. Θέτει βαθμολογία στις λίστες που του αντιστοιχούν.

10.3.2. Βλέπει τις βαθμολογίες των όλων των φοιτητών αλλά μόνο για τα μαθήματά του.



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

0	Γενική Περιγραφή της λύσης.....	5
0.1	Για την συγκεκριμένη άσκηση χρησιμοποιήθηκε η εφαρμογή eclipse. Καθώς επίσης και οι βιβλιοθήκες jdk της java και τέλος έγινε και απαραίτητος έλεγχος λειτουργίας της εφαρμογής...	5
0.2	Δημιουργήθηκε ένα Java Project με τίτλο JavaBasics	5
0.3	Δημιουργήθηκε ένα πακέτο με τίτλο mainpackage	5
1	Εκκίνηση project.....	5
1.1	Δημιουργήθηκε ένα Java Project με τίτλο JavaBasics.	5
1.2	Δημιουργήθηκε ένα πακέτο με τίτλο mainpackage	5
2	Δημιουργία Βασικών κλάσεων.....	5
3	Κώδικας προγράμματος.....	10
4	UML Diagramm.....	24
5	Βιβλιογραφικές Πηγές.....	25



0 Γενική Περιγραφή της λύσης

0.1 Για την συγκεκριμένη άσκηση χρησιμοποιήθηκε η εφαρμογή eclipse. Καθώς επίσης και οι βιβλιοθήκες jdk της java και τέλος έγινε και απαραίτητος έλεγχος λειτουργίας της εφαρμογής

0.2 Δημιουργήθηκε ένα Java Project με τίτλο JavaBasics

0.3 Δημιουργήθηκε ένα πακέτο με τίτλο mainpackage

1 Εκκίνηση project

1.1 Δημιουργήθηκε ένα Java Project με τίτλο JavaBasics.

1.2 Δημιουργήθηκε ένα πακέτο με τίτλο mainpackage

2 Δημιουργία Βασικών κλάσεων

[2.1]

Για το συγκεκριμένο ερώτημα δημιουργήθηκε μια **abstract** κλάση Users η οποία περιέχει τα ακόλουθα χαρακτηριστικά :

1. name: τύπου String για να αποθηκεύει το όνομα του χρήστη.
2. Surname: τύπου String για να αποθηκεύει το επίθετο του χρήστη.
3. Username: τύπου String για να αποθηκεύει το username του χρήστη.
4. Department: τύπου String για να αποθηκεύει το τμήμα φοίτησης του χρήστη.
5. UsersCounter: τύπου int για να αποθηκεύει το usersCounter του χρήστη.

Επίσης δημιουργήθηκε ένας Constructor:

public Users(String name,String surname,String username,String department) ο οποίος παίρνει σαν ορίσματα όλα τα χαρακτηριστικά της κλάσης Users και δημιουργεί ένα νέο αντικείμενο.

Επίσης δημιουργήθηκαν μέθοδοι get και set για το εκάστοτε χαρακτηριστικό καθώς και εφαρμογή με όνομα show_user για την εμφάνιση των χαρακτηριστικών αυτών.



[3]

Για το συγκεκριμένο ερώτημα δημιουργήθηκε κλάση με όνομα `Students` η οποία είναι επέκταση της `Users`. Αυτό έχει ως αποτέλεσμα να έχει τα ίδια χαρακτηριστικά με την `Users`.

[3.1]

Δημιουργήθηκε επιπλέον χαρακτηριστικό για την συγκεκριμένη κλάση τύπου `int` με όνομα μεταβλητής `registrationNumber`.

[3.2]

Δημιουργήθηκε constructor ο οποίος δέχεται σαν ορίσματα τα χαρακτηριστικά της κλάσης `Users` συν το `registrationNumber`. Επίσης για να γίνει override ο constructor της `Users` χρησιμοποιείται η εντολή `super`.

[3.3]

Δημιουργήθηκε η επιπλέον μέθοδος με όνομα `Show_Student` και `Show_mitrow`. Στην πρώτη μέθοδος γίνεται εκτύπωση των στοιχείων του φοιτητή και στην δεύτερη ο αριθμός μητρώου του μόνο. Επιπλέον δημιουργήθηκαν `getters` και `setters` για να το χαρακτηριστικό `registrationNumber`.

[4]

Δημιουργήθηκε η κλάση με όνομα `Professors` η οποία είναι προέκταση της `Users`. Επιπλέον χαρακτηριστικό έχει αυτό του μαθήματος, το οποίο είναι κλάσης `Courses`. Δημιουργήθηκε constructor ο οποίος κάνει «`super`» τα χαρακτηριστικά της κλάσης `Users`. Επιπλέον δημιουργήθηκαν 2 μέθοδοι με όνομα `Show_professor` και `Show_mathima`. Η πρώτη εκτυπώνει τα στοιχεία του καθηγητή και η δεύτερη το μάθημα που διδάσκει. Επίσης η κλάση διαθέτει `getters` και `setters` του χαρακτηριστικού `mathimatos`.

[5]

Δημιουργήθηκε η κλάση με όνομα `Secretaries` η οποία είναι προέκταση της `Users`. Επιπλέον χαρακτηριστικό δεν έχει. Δημιουργήθηκε constructor ο οποίος κάνει «`super`» τα χαρακτηριστικά της κλάσης `Users`. Επιπλέον δημιουργήθηκε η μέθοδος με όνομα `Show_grammateia`, η οποία εκτυπώνει τα στοιχεία του προσώπου της γραμματείας.

[6]

Δημιουργήθηκαν κλάσεις με όνομα `Courses` και `Grades`. Η πρώτη για τα μαθήματα της εφαρμογής και η δεύτερη για τους βαθμούς του εκάστοτε φοιτητή. Η κλάσης `Courses` έχει ως χαρακτηριστικά το `courseid` τύπου `int` (κωδικός μαθήματος), `coursename` τύπου `string` (όνομα μαθήματος), `etos` τύπου `int` (έτος μαθήματος). Η κλάσης `Grades` έχει ως χαρακτηριστικά το `foititis` κλάσης `Students`, το μάθημα κλάσης `Courses` και το `vathmos` τύπου `int`. Επιπλέον μέθοδος φτιάχτηκε στην κλάση `Grades` με όνομα `Show_grade`, μέσω της οποίας εκτυπώνεται τα στοιχεία του φοιτητή και ο βαθμός. Τέλος, κατασκευάστηκαν `getters` και `setters` για όλα τα χαρακτηριστικά και στις δυο κλάσεις.



[7.1]

Δημιουργήθηκε κλάσης CreateUsers η οποία περιλαμβάνει την `main(String[] args)`. Στην συνέχεια δημιουργήθηκε αντικείμενο με όνομα `input` τύπου `Scanner` για να εισαγωγή δεδομένων από τον χρήστη.

Ακολουθώντας, μέσω του αντικειμένου `input` εισάγουμε στοιχεία για την δημιουργία αντικειμένου τύπου `Students` και `Users`. Χωρίς την λειτουργία του αντικειμένου `Input` έγινε η δημιουργία αντικειμένων τύπου `Courses`, `Proffessors` και `Secretaries`

[7.2]

Μετά την εκάστοτε δημιουργία αντικειμένου, έγινε χρήση μεθόδου για την εμφάνιση χαρακτηριστικών αυτού

- `Show_User` σειρά 64,
- `Show_student` σειρά 119,
- `Show_proffesor` σειρά 137
- `Show_grammateia` σειρά 142.

[7.3]

Κατά την διάρκεια συγγραφής του κώδικα έχει γίνει τοποθέτηση κατάλληλων σχολίων.

[8.1]-[8.2]

Τόσο στην δημιουργία αντικειμένου `Users` και `Students` χρησιμοποιήθηκε `try-catch` με κατάλληλα `exception` μετά από κάθε έλεγχο εισόδου. Επιπλέον μετά την εισαγωγή στοιχείων από τον χρήστη γίνεται χρησιμοποίηση της μεθόδου `matches` για να γίνει έλεγχος καταχώρησης. Ελέγχεται

- Αν ο αριθμός μητρώου είναι `int` (σειρά 110) (`.matches("[0-9]*$")`)
- Αν τα υπόλοιπα στοιχεία είναι τύπου `strings` (`.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")`)

[8.3]

Κατά την διάρκεια συγγραφής του κώδικα έχει γίνει τοποθέτηση κατάλληλων σχολίων.

[9.1]

Εντός του φακέλου `source` του `project` δημιουργήθηκε αρχείο `txt` με όνομα `Student2` σύμφωνα με τις προδιαγραφές του ερωτήματος 9.1 και διαχωριστικό το «/».

[9.2]

Γίνεται χρήση αντικειμένων τύπου `FileInputStream`, `DataInputStream` και `BufferedReader`. Στην συνέχεια μέσω ενός `loop While` διαβάζεται `line by line` το αρχείο. Σε έναν πίνακα με όνομα `tokens` εισέρχονται τα στοιχεία του `txt` δίνοντας ως `split` το «/». Στην συνέχεια παίρνουμε τα στοιχεία του πίνακα και δημιουργούμε το αντικείμενο τύπου `Students`.



[9.3]

Κατά την διάρκεια δημιουργίας του πίνακα γίνεται έλεγχος με την μέθοδο `.matches`. Αν κάτι δεν έχει καταχωρηθεί στο αρχείο `txt` σωστά, γίνεται μέσω ενός `Exception` εκτύπωση μηνύματος λάθους. Επιπλέον, αν κάτι πάει λάθος με τα `stream` του αρχείου μέσω `try-catch` γίνεται εκτύπωση μηνύματος λάθους.

[10.1.1]

- Καλείται η μέθοδος **Dimiourgia_kathighi** στέλνοντας ως παραμέτρους μια λίστα με τους καθηγητές και τα μαθήματα που έχουν δημιουργηθεί (`lista_kathigitwn`, `lista_mathimatwn`). Στην συνέχεια η συνάρτηση `Dimiourgia_kathighi` στην κλάση `Secretaries` δημιουργεί αντικείμενο `Professor` και το εισάγει στην λίστα καθηγητών. Για την δημιουργία του αντικείμενου `Professor` χρειάζεται και το αντίστοιχο μάθημα του. Μάθημα το οποίο το κάνουμε αναζήτηση στην `lista_mathimatwn` με βάση το μάθημα που δόθηκε από το πληκτρολόγιο και της μεθόδου `findUsingEnhancedForLoop`.

- Καλείται η μέθοδος **Dimiourgia_foithth** στέλνοντας ως παραμέτρους μια λίστα με τους φοιτητές που έχουν δημιουργηθεί (`lista_foithitwn`). Στην συνέχεια η μέθοδος `Dimiourgia_foithth` στην κλάση `Secretaries` δημιουργεί αντικείμενο `Students` και το προσθέτει στην λίστα φοιτητών.

- Καλείται η μέθοδος **Dimiourgia_mathimatos** στέλνοντας ως παραμέτρους μια λίστα με τους μαθημάτων που έχουν δημιουργηθεί (`lista_mathimatwn`). Στην συνέχεια η μέθοδος `Dimiourgia_mathimatos` στην κλάση `Secretaries` δημιουργεί αντικείμενο `Courses` και το προσθέτει στην λίστα μαθημάτων.

[10.1.2]

Καλείται η μέθοδος **Anathesi_mathimatos** στέλνοντας ως παραμέτρους μια λίστα με τους καθηγητές και τα μαθήματα που έχουν δημιουργηθεί (`lista_kathigitwn`, `lista_mathimatwn`). Στην συνέχεια εκτυπώνεται κατάλληλο μήνυμα.

[10.1.3]

Στην κλάση `CreateUsers` και συγκεκριμένα στην σειρά 18 δημιουργείται λίστα φοιτητών που πρόκειται να δοθούν στην γραμματεία για βαθμολόγηση μέσω της μεθόδου `vathmologish_foithitwn` που βρίσκεται στην κλάση `secretaries`.

[10.2.1]

Έχει δημιουργηθεί μέθοδος στην κλάση `Students` με όνομα `emfanish_vathmologias` η οποία στην παρούσα εκτυπώνει μόνο κατάλληλο μήνυμα. Επιπλέον δέχεται ως ορίσματα τις λίστες μαθητών και τις λίστες βαθμών.

[10.3.1]

Έχει δημιουργηθεί μέθοδος στην κλάση `Professors` με όνομα `emfanish_vathmologias` η οποία στην παρούσα εκτυπώνει μόνο κατάλληλο μήνυμα. Επιπλέον δέχεται ως ορίσματα τις λίστες μαθητών και τις λίστες βαθμών.



[10.3.3]

Έχει δημιουργηθεί μέθοδος στην κλάση Professors με όνομα `emfanish_vathμου_gia_mathima` η οποία στην παρούσα εκτυπώνει μόνο κατάλληλο μήνυμα.



3. Κώδικας προγράμματος

CLASS: USERS

```
package mainpackage;
//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ USERS ΓΙΑ ΕΡΩΤΗΜΑ 2,1
public class Users {
//ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΓΙΑ ΕΡΩΤΗΜΑ 2.1.1
    private String name;
    private String surname;
    private String username;
    private String department;
    static int usersCounter=0;

    public Users(String name2, String surname2, String username2, String department2) {
        //CONSTUCTOR ΓΙΑ ΕΡΩΤΗΜΑ 2.1.3
        this.name=name2;
        this.surname=surname2;
        this.username=username2;
        this.department=department2;
        this.usersCounter++;
    }

    void Show_User() {
        System.out.println("Ο Χρήστης είναι ο " + name + "-" + surname + "-" +
        username + "Στο τμήμα" + department );
    }

    //ΜΕΘΟΔΟΙ GETTERS + SETTERS 2.1.2
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSurname() {
        return surname;
    }

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getDepartment() {
        return department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    public static int getUsersCounter() {
        return usersCounter;
    }

    public static void setUsersCounter(int usersCounter) {
        Users.usersCounter = usersCounter;
    }
}
```



CLASS: STUDENTS

```
package mainpackage;

import java.util.List;
//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ ΓΙΑ ΕΡΩΤΗΜΑ 3
class Students extends Users {

    //ΕΡΩΤΗΜΑ 3.1. ΕΙΝΑΙ FINAL ΔΙΟΤΙ ΔΕΝ ΘΑ ΑΛΛΑΖΕΙ ΠΟΤΕ
    final int registrationNumber ;

    public Students(String name,String surname,String username,String department,int
registrationNumber) {
        //ΕΡΩΤΗΜΑ 3.2 CONSTRUCTOR
        super(name,surname,username,department);
        this.registrationNumber=registrationNumber;
    }

    //ΕΡΩΤΗΜΑ 3.3. ΜΕΘΟΔΟΣ ΓΙΑ ΝΑ ΕΜΦΑΝΙΖΟΝΤΑΙ ΟΙ ΦΟΙΤΗΤΕΣ
    public void Show_student() {
        System.out.println("Ο Φοιτητής είναι ο " + getName() + "Με επίθετο"
            + getSurname() +"Στο τμήμα" + getDepartment());
    }

    public void Show_mitrwo() {
        System.out.println("Αριθμός Μητρώου " + registrationNumber );
    }

    public int getRegistrationNumber() {
        return registrationNumber;
    }

    //ΕΡΩΤΗΜΑ 10.2.1
    public void emfanish_vathmologias(List<Students> studentList, List<Grades>
lista_vathmwn) {

        // sunarthsh gia na vlepoun oi foitites tis vatmologies
        System.out.println("Εδω οι φοιτητες θα βλεπουν τις βαθμολογιες τους " );
    }
}
```



CLASS: PROFESSORS

```
package mainpackage;

import java.util.List;
//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ ΕΡΩΤΗΜΑ 4
class Professors extends Users {
    //ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ ΚΛΑΣΗΣ COURSES ΓΙΑ ΝΑ ΔΗΛΩΝΕΙ ΣΕ ΠΟΙΟ/Α ΜΑΘΗΜΑΤΑ ΕΙΝΑΙ Ο
    ΚΑΘΗΓΗΤΗΣ
    public Courses mathimatos;

    public Professors(String name, String surname, String username, String
department, Courses mathimatos) {
        //CONSTRUCTOR
        super(name, surname, username, department);
        this.mathimatos = mathimatos;
    }
    // ΜΕΘΟΔΟΣ ΓΙΑ ΕΜΦΑΝΙΣΗ ΚΑΘΗΓΗΤΗ
    public void Show_professor() {
        System.out.println("Ο Καθηγητής είναι ο " + getName() + "Με
επίθετο"
            + getSurname() + "Στο τμήμα" + getDepartment());
    }
    public void Show_mathima() {
        System.out.println("Διδάσκει " + mathimatos);
    }
    public Courses getMathimatos() {
        return mathimatos;
    }
    public void setMathimatos(Courses mathimatos) {
        this.mathimatos = mathimatos;
    }
    // 10.3.1
    public void emfanish_vathmologias(List<Students> lista_student,
List<Courses> courseList) {
        // sunarthsh gia na vathmologei o kathigiths
        System.out.println("Εδώ ο καθηγητής θα βαθμολογεί τα μαθηματα που
του εχουν ανατεθει");
    }
    // 10.3.2
    public void emfanish_vathmou_gia_mathima(List<Students>
lista_student, List<Courses> courseList) {

        // μέθοδος που προοριζεται για να γινεται εμφανιση βαθμων
        μονο οσως μαθηματων διδάσκει ο εκαστοτε καθηγητής
        System.out.println("μέθοδος που προοριζεται για να γινεται
εμφανιση βαθμων μονο οσως μαθηματων διδάσκει ο εκαστοτε καθηγητής");

    }

}
```



CLASS: GRADES

```
package mainpackage;
//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ ΒΑΘΜΩΝ ΓΙΑ ΕΡΩΤΗΣΗ 6
public class Grades {
    //ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΛΑΣΗΣ
    Students foitiths ;
    Courses mathima ;
    int vathmos;

    public Grades(Students foitiths , Courses mathima , int vathmos) {
        this.foitiths = foitiths;
        this.mathima = mathima;
        this.vathmos = vathmos;
    }
    //ΜΕΘΟΔΟΣ ΓΙΑ ΕΜΦΑΝΙΣΗ ΒΑΘΜΟΥ
    void Show_grade(Students foitiths , Courses mathima , int vathmos) {
        System.out.println("Oi vathmoi tou" + foitiths + "gia to mathima" +
mathima.getCoursename() + "einai" + vathmos);
    }

    // GETTERS + SETTERS ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ
    public Students getFoitiths() {
        return foitiths;
    }

    public void setFoitiths(Students foitiths) {
        this.foitiths = foitiths;
    }

    public Courses getMathima() {
        return mathima;
    }

    public void setMathima(Courses mathima) {
        this.mathima = mathima;
    }

    public int getVathmos() {
        return vathmos;
    }

    public void setVathmos(int vathmos) {
        this.vathmos = vathmos;
    }
}
```



CLASS: COURSES

```
package mainpackage;

//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ ΓΙΑ ΕΡΩΤΗΜΑ 6
public class Courses {
    //ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΑΘΗΜΑΤΟΣ
    private int courseid;
    private String coursename;
    private int etos ;

    public Courses(int courseid , String coursename , int etos ) {

        this.courseid = courseid ;
        this.coursename = coursename;
        this.etos = etos ;
    }

    //GETTERS + SETTERS
    public int getCourseid() {
        return courseid;
    }

    public void setCourseid(int courseid) {
        this.courseid = courseid;
    }

    public String getCoursename() {
        return coursename;
    }

    public void setCoursename(String coursename) {
        this.coursename = coursename;
    }

    public int getEtos() {
        return etos;
    }

    public void setEtos(int etos) {
        this.etos = etos;
    }

}
```



CLASS: SECRETARIES

```
package mainpackage;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ ΓΙΑ ΤΗΝ ΓΡΑΜΜΑΤΕΙΑ ΓΙΑ ΕΡΩΤΗΜΑ 5
class Secretaries extends Users {

    Scanner input = new Scanner(System.in);

    public Secretaries(String name, String surname, String username, String
department) {
        super(name, surname, username, department);
    }

    public void Show_Grammateia() {
        System.out.println("Ο/Η ΓΡΑΜΜΑΤΕΑΣ είναι ο/η " + getName() + "Με
επίθετο"
            + getSurname() +"Στο τμήμα" + getDepartment());
    }
    //ΜΕΘΟΔΟΣ ΓΙΑ ΝΑ ΒΡΙΣΚΕΙ ΜΑΘΗΜΑ ΑΠΟ ΜΙΑ ΛΙΣΤΑ ΜΑΘΗΜΑΤΩΝ
    public Courses findUsingEnhancedForLoop(String mathima_kathigiti,
List<Courses> courseList) {

        for (Courses cours : courseList) {
            if (cours.getCoursename().equals(mathima_kathigiti)) {
                return cours;
            }
        }
        return null;
    }

    public List<Professors> Dimiourgia_kathigith(List<Professors>
professorList, List<Courses> courseList) {
        try {
            System.out.println("Δώστε το όνομα του Καθηγητή: ");
            String name_kathigiti = input.nextLine();
            if (name_kathigiti.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

            } else {
                throw new Exception("το όνομα δεν είναι σωστό");
            }

            System.out.println("Δώστε το επίθετο του Καθηγητή ");
            String surname_kathigiti = input.nextLine();
            if (surname_kathigiti.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

            } else {
                throw new Exception("το επίθετο δεν είναι σωστό");
            }
        }
    }
}
```



```
System.out.println("Δώστε το username του Καθηγητή: ");
String username_kathigiti = input.nextLine();
if (username_kathigiti.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

} else {
    throw new Exception("το username δεν είναι σωστό");
}

System.out.println("Δώστε το τμήμα που διδάσκει: ");
String department_kathigiti = input.nextLine();
if (department_kathigiti.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {
} else {
    throw new Exception("το τμήμα δεν είναι σωστό");
}

System.out.println("Δώστε το μάθημα που διδάσκει: ");
String mathima_kathigiti = input.nextLine();

Courses mathima = findUsingEnhancedForLoop(mathima_kathigiti,
courseList);

Professors kathigiths = new Professors(name_kathigiti,
surname_kathigiti, username_kathigiti,
    department_kathigiti, mathima);
professorList.add(kathigiths);

} catch (Exception e) {
    System.out.println("General Exception occurred" + e);
}
return professorList;
}

////////////////////////////////////
////////////////////////////////////

public List<Students> Dimiourgia_foithth(List<Students> lista_student) {
    try {

        System.out.println("Δώστε το όνομα του Φοιτητή: ");
        String names_student = input.nextLine();
        if (names_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

        } else {
            throw new Exception("το όνομα δεν είναι σωστό");
        }

        System.out.println("Δώστε το επίθετο του Φοιτητή ");
        String surname_student = input.nextLine();
        if (surname_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

        } else {
            throw new Exception("το επίθετο δεν είναι σωστό");
        }
    }
}
```




```
System.out.println("Δώστε το username του Φοιτητή: ");
String username_student = input.nextLine();
if (username_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

} else {
    throw new Exception("το username δεν είναι σωστό");
}

System.out.println("Δώστε το τμήμα που φοιτεί: ");
String department_student = input.nextLine();
if (department_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

} else {
    throw new Exception("το τμήμα δεν είναι σωστό");
}

System.out.println("Δώστε το registration number: ");
String registrationNumber = input.nextLine();
if (registrationNumber.matches("[0-9]*$")) {

} else {
    throw new Exception("Ο Αριθμός Μητρώου δεν είναι
σωστός");
}

Students foithths = new Students(names_student,
surname_student, username_student, department_student,
    Integer.parseInt(registrationNumber));
lista_student.add(foithths);

} catch (Exception e) {
    System.out.println("General Exception occurred" + e);
}
return lista_student;
}

public List<Courses> Dimiourgia_mathimatos(List<Courses> lista_mathimatwn)
{
    try {

        System.out.println("Δώστε τον Κωδικό του Μαθήματος: ");
        String course_id = input.nextLine();
        if (course_id.matches("[0-9]*$")) {
        } else {
            throw new Exception("ο κωδικός δεν είναι σωστός");
        }

        System.out.println("Δώστε το όνομα του Μαθήματος: ");
        String onoma_mathimatos = input.nextLine();
        if (onoma_mathimatos.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

        } else {
            throw new Exception("το όνομα δεν είναι σωστό");
        }
    }
}
```



```
        System.out.println("Δώστε το έτος του Μαθήματος: ");
        String etos_mathimatos = input.nextLine();
        if (etos_mathimatos.matches("[0-9]*$")) {
        } else {
            throw new Exception("το έτος δεν είναι σωστό");
        }

        Courses neo_mathima = new
Courses(Integer.parseInt(course_id), onoma_mathimatos,
        Integer.parseInt(etos_mathimatos));
        lista_mathimatwn.add(neo_mathima);

    } catch (Exception e) {
        System.out.println("General Exception occurred" + e);
    }
    return lista_mathimatwn;
}

// 10.1.2
public void Anathesh_mathimatos(List<Professors> professorList,
List<Courses> courseList) {
    System.out.println("Εδώ θα γίνεται η ανάθεση μαθήματος");
    // sunarthsh gia anathesi mathimatwn
    // mesw enos for kai katalilhs anazhths stis listes tha ginetai h
anathesi
}
// 10.1.3
public void vathmologish_foititwn(List<Students>
lista_student,List<Grades> lista_vathmwn) {
    System.out.println("Εδώ θα γίνεται η βαθμολογηση φοιτητών");
    // sunarthsh gia vathmologish foititwn
    // pairnei thn lista kai kanei tis katalililes anatheseis

}

}
```



CLASS: CreateUsers

```
package mainpackage;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.io.*;
//ΔΗΜΙΟΥΡΓΙΑ ΚΛΑΣΗΣ ΓΙΑ ΕΡΩΤΗΜΑ 7,1
public class CreateUsers {

    public static void main(String[] args) {
        //ΛΙΣΤΕΣ ΜΕ ANTIKEimena
        List<Users> lista_xrhstwn = new ArrayList<Users>();
        List<Students> lista_foititwn = new ArrayList<Students>();
        List<Secretaries> lista_gramateias = new ArrayList<Secretaries>();
        List<Professors> lista_kathigitwn = new ArrayList<Professors>();
        List<Grades> lista_vathmwn = new ArrayList<Grades>();
        List<Courses> lista_mathimatwn = new ArrayList<Courses>();

        // ΔΗΜΙΟΥΡΓΙΑ USER ΜΕΣΩ ΜΕΣΩ SCANNER ΚΑΙ TRY-CATCH. ΕΡΩΤΗΜΑ 7.1
        Scanner input = new Scanner(System.in);
        try {

            System.out.println("Δώστε το όνομα του Χρήστη: ");
            String name = input.nextLine();
            if (name.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

            } else {
                throw new Exception("το όνομα δεν είναι σωστό");
            }

            System.out.println("Δώστε το επίθετο του Χρήστη ");
            String surname = input.nextLine();
            if (surname.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

            } else {
                throw new Exception("το επίθετο δεν είναι σωστό");
            }

            System.out.println("Δώστε το username του χρήστη : ");
            String username = input.nextLine();
            if (username.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

            } else {
```



```
        throw new Exception("το username δεν είναι σωστό");
    }

    System.out.println("Δώστε το τμήμα που ανοίκει: ");
    String department = input.nextLine();
    if (department.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

    } else {
        throw new Exception("το τμήμα δεν είναι σωστό");
    }

    Users xrhsths = new Users(name, surname, username,
department);

    lista_xrhstwn.add(xrhsths);
    //ΕΜΦΑΝΙΣΗ ΧΡΗΣΤΗ ΓΙΑ ΕΡΩΤΗΜΑ 7.2
    xrhsths.Show_User();

} catch (Exception e) {
    System.out.println("General Exception occurred" + e);
} finally {
    System.out.println("Η Εγγραφή του Χρήστη τελείωσε!");
}

// ΔΗΜΙΟΥΡΓΙΑ STUDENT ΜΕΣΩ ΜΕΣΩ SCANNER ΚΑΙ TRY-CATCH. ΕΡΩΤΗΜΑ 7.1
//TRY-CATCH ΓΙΑ ΕΡΩΤΗΜΑ 8.1
//ΜΕ ΤΙΣ ΜΕΘΟΔΟΥΣ SHOW ΤΗΣ ΕΚΑΣΤΟΤΕ ΚΛΑΣΗΣ ΓΙΝΕΤΑΙ ΕΜΦΑΝΙΣΗ ΤΩΝ
ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ
    try {

        System.out.println("Δώστε το όνομα του Φοιτητή: ");
        String names_student = input.nextLine();
        if (names_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

        } else {
            throw new Exception("το όνομα δεν είναι σωστό");
        }

        System.out.println("Δώστε το επίθετο του Φοιτητή ");
        String surname_student = input.nextLine();
        if (surname_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

        } else {
            throw new Exception("το επίθετο δεν είναι σωστό");
        }

        System.out.println("Δώστε το username του Φοιτητή: ");
        String username_student = input.nextLine();
        if (username_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

        } else {
            throw new Exception("το username δεν είναι σωστό");
        }

        System.out.println("Δώστε το τμήμα που φοιτεί: ");
```



```
String department_student = input.nextLine();
if (department_student.matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$")) {

} else {
    throw new Exception("το τμήμα δεν είναι σωστό");
}
System.out.println("Δώστε το registration number: ");
String registrationNumber = input.nextLine();
if (registrationNumber.matches("[0-9]*$")) {

} else {
    throw new Exception("Ο Αριθμός Μητρώου δεν είναι
σωστός");
}

Students foithths = new Students(names_student,
surname_student, username_student, department_student,
Integer.parseInt(registrationNumber));
//ΕΜΦΑΝΙΣΗ ΦΟΙΤΗΤΗ ΕΡΩΤΗΜΑ 7.2
foithths.Show_student();
lista_foithitwn.add(foithths); //ΠΡΟΣΘΗΚΗ ΦΟΙΤΗΤΗ ΣΤΗΝ ΛΙΣΤΑ
ΦΟΙΤΗΤΩΝ

foithths.emfanish_vathmologias(lista_foithitwn,
lista_vathmwn); //ΕΡΩΤΗΜΑ 10,2,1. ΕΜΦΑΝΙΣΗ ΒΑΘΜΟΛΟΓΙΩΝ

} catch (Exception e) {
    System.out.println("General Exception occurred" + e);
} finally {
    System.out.println("Η Εγγραφή του Φοιτητή τελείωσε!");
}

//ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΜΑΘΗΜΑΤΟΣ ΚΑΙ ΕΙΣΑΓΩΓΗ ΤΟΥ ΣΤΗΝ ΛΙΣΤΑ
Courses agglika = new Courses(0, "agglika", 3);
lista_mathimatwn.add(agglika);

// ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΚΑΘΗΓΗΤΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΛΙΣΤΑ ΕΡΩΤΗΜΑ 7.1
Professors kathigitis = new Professors("Γιώργος", "Φιλιππαίος",
"filippaios", "Πληροφορικής", agglika);
lista_kathigitwn.add(kathigitis);
kathigitis.Show_professor(); //ΕΜΦΑΝΙΣΗ ΓΙΑ ΕΡΩΤΗΜΑ 7.2

// ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΓΡΑΜΜΑΤΕΑ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΛΙΣΤΑ ΕΡΩΤΗΜΑ 7.1
Secretaries grammateia = new Secretaries("Κική", "Βουγιουκλάκη",
"kiki", "Πληροφορικής");
lista_grammateias.add(grammateia);
grammateia.Show_Grammateia(); //ΕΜΦΑΝΙΣΗ ΓΙΑ ΕΡΩΤΗΜΑ 7.2

////////////////////////////////////
////////////////////////////////////
//ΕΝΤΟΛΕΣ ΓΙΑ ΤΑ ΕΡΩΤΗΜΑΤΑ 9 ΚΑΘΩΣ ΚΑΙ TRY CATCH ΓΙΑ ΕΡΩΤΗΜΑ 9.3
// 9.1, 9.2..
try {
```



```
        FileInputStream fstream = new
FileInputStream("/src/mainpackage/Student.txt");
        DataInputStream in = new DataInputStream(fstream);
        BufferedReader br = new BufferedReader(new
InputStreamReader(in));
        String strLine;
        while ((strLine = br.readLine()) != null) {
            String[] tokens = strLine.split("/");
            int number = Integer.parseInt(tokens[4]);

            if ((tokens[0].matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$"))) {
            } else {
                throw new Exception("το όνομα δεν είναι σωστό");
            }
            if ((tokens[1].matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$"))) {
            } else {
                throw new Exception("το επίθετο δεν είναι
σωστό");
            }
            if ((tokens[2].matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$"))) {
            } else {
                throw new Exception("το username δεν είναι
σωστό");
            }
            if ((tokens[3].matches("[α-ωΑ-Ωα-ζΑ-Ζ]*$"))) {
            } else {
                throw new Exception("το Τμήμα δεν είναι σωστό");
            }
            if (tokens[4].matches("[0-9]*$")) {
            } else {
                throw new Exception("Ο Αριθμός Μητρώου δεν είναι
σωστός");
            }

            Students record = new Students(tokens[0], tokens[1],
tokens[2], tokens[3], number);
            System.out.println("Η Εγγραφή του Φοιτητή τελείωσε!" +
record.getRegistrationNumber());
        }
        in.close();
    } catch (Exception e) {
        System.err.println("Κάτι πήγε λάθος: " + e.getMessage());
    }

    //////////////////////////////////////
    //////////////////////////////////////
    // ΕΝΤΟΛΕΣ ΓΙΑ ΕΡΩΤΗΜΑΤΑ ΣΤΗΝ ΠΑΡΑΓΡΑΦΟ 10.0

    grammateia.Dimiourgia_kathigith(lista_kathigitwn,lista_mathimatwn);
//10.1.1
    grammateia.Dimiourgia_foithth(lista_foititwn); //10,1,1
```



```
        grammateia.Dimiourgia_mathimatos(lista_mathimatwn); //10,1,1
        grammateia.Anathesh_mathimatos(lista_kathigitwn,lista_mathimatwn);
//10,1,2
        grammateia.vathmologish_foititwn(lista_foititwn,lista_vathmwn);
//10,1,3

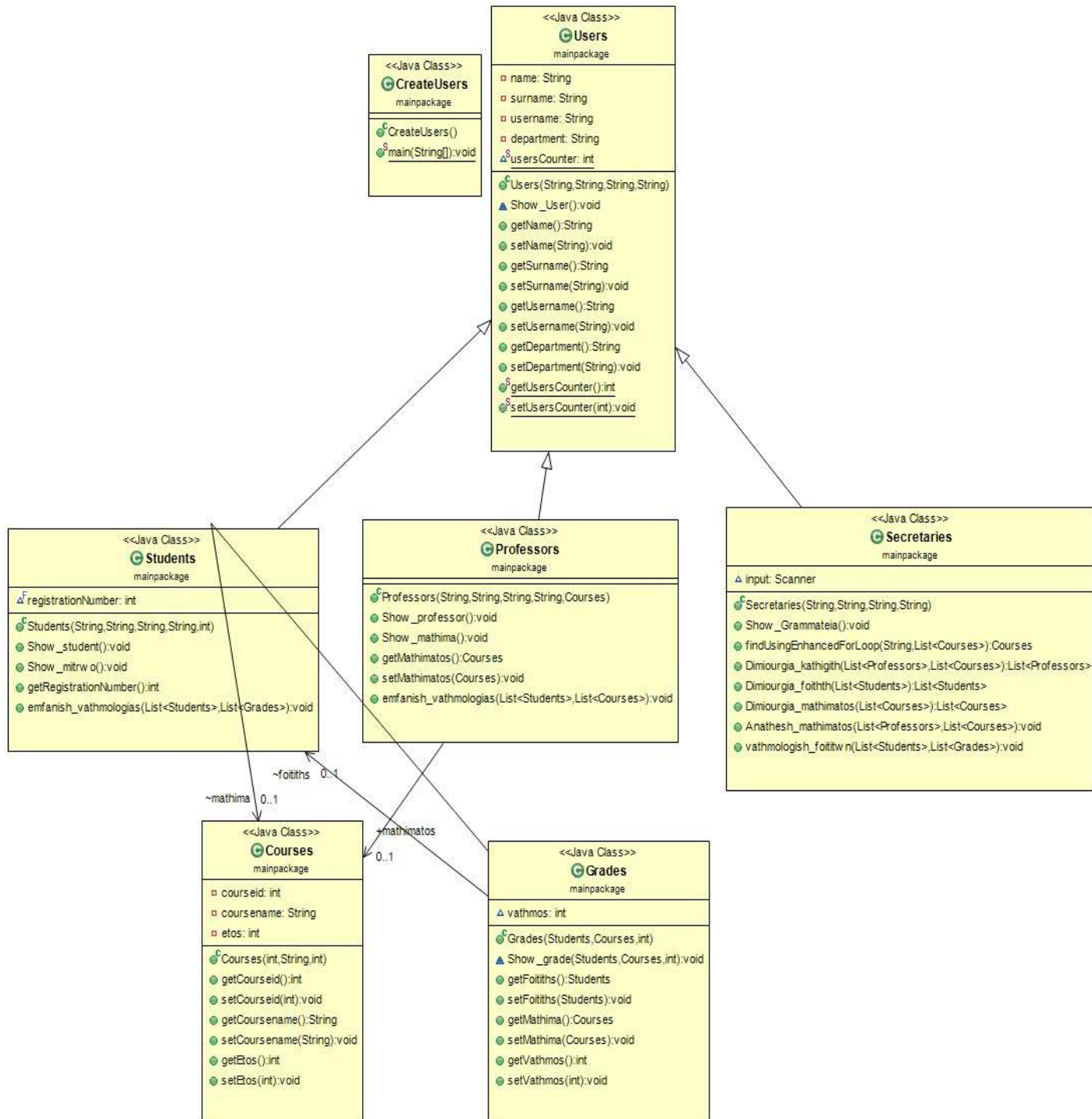
        kathigitis.emfanish_vathmologias(lista_foititwn, lista_mathimatwn);
//10,3,1

    }

}
```



4 UML Diagramm





5 Βιβλιογραφικές Πηγές

- 1) Σημειώσεις μαθήματος από gunett.
- 2) <https://stackoverflow.com/>
- 3) <https://www.w3schools.com/>
- 4) <https://www.geeksforgeeks.org/>