

**Πανεπιστήμιο Πληροφορικής
Τμήμα Πληροφορικής
Έτος: 2021 - 2022**



Μάθημα:
«Προγραμματισμός Στο Διαδίκτυο Και Στον Παγκόσμιο Ιστό»
Εργασία: 1η Άσκηση Μαθήματος
Εξάμηνο: 4ο

Ομάδα εργασίας:

Αιμιλιανός Κουρπάς Δανάς Π20100,
Απόστολος Σιαμπάνης Π20173,
Αγγελική Καλδύρη Π20069,
Θεόδωρος Κοξάνογλου Π20094



Εκφώνηση της άσκησης

1η Άσκηση Μαθήματος

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 χρησιμοποιήστε τις μεθόδους που έχετε δημιουργήσει. Χρησιμοποιήστε 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 ελέγχετε ότι το πεδίο Αριθμός Μητρώου είναι αριθμός (ακέραιος), ενώ όλα τα άλλα πεδία είναι αλφαριθμητικά (Strings). Χρησιμοποιήστε 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 Secreteries 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.

Οδηγίες:

- Δημιουργήστε ομάδες (έως 4 άτομα), οι οποίες θα παραμείνουν οι ίδιες μέχρι το τέλος του μαθήματος (τελική εργασία).



-
- Χρησιμοποιήστε το υπόδειγμα συγγραφής εργασιών. Το παραδοτέο θα περιλαμβάνει σε ένα συμπιεσμένο αρχείο, το κείμενο (βάσει του υποδείγματος) και τον κώδικα (σε μορφή project).
 - Η παράδοση θα γίνει μέσω του eclass από έναν φοιτητή για κάθε ομάδα.
-



Περιεχόμενα

Γενική Περιγραφή Προγράμματος	6
1.1 Γενική Περιγραφή της Διεπαφής	6
1.2 Γενική Περιγραφή των Κλάσεων.	6
1.3 Διάγραμμα Κλάσεων και Διεπαφών	8
Κώδικας Προγράμματος	9
2.1 Αναλυτική Περιγραφή Διεπαφής	9
2.1.1 FrontUserCommands	9
2.2 Αναλυτική Περιγραφή Κλάσεων	10
2.2.1 Department	10
2.2.2 User	11
2.2.3 CreateUser	13
2.2.4 Student	14
2.2.5 Professor	15
2.2.6 Secretary	17
2.2.7 Course	18
2.2.8 CreateCourse	19
2.2.9 Grade	19
2.2.10 MainMenu	20
Εκτέλεση Προγράμματος	21
Βιβλιογραφικές Πηγές	24



1. Γενική Περιγραφή Προγράμματος

Ακολουθώντας τις γενικές οδηγίες της εργασίας δημιουργήσαμε ένα Java Project στο Eclipse IDE for Enterprise and Web Developers:

Τίτλος project: JavaWebProject_Part1

Project package: gradepackage

Στο package αυτό δημιουργήσαμε **10 classes** (Course, CreateCourse, CreateUser, Department, Grade, MainMenu, Professor, Secretary, Student, User) και **1 interface** (FrontUserCommands).

Στο πρόγραμμα είναι υλοποιημένες οι βασικές λειτουργίες της τελικής εφαρμογής, καθώς και τα βασικά κελιά των μεθόδων που θα υλοποιηθούν στο δεύτερο μέρος υλοποίησης της εφαρμογής.

Στην πρώτη φάση της εργασίας θεωρούμε ότι η γραμματεία μπορεί να γράψει μαθήματα, φοιτητές και καθηγητές που ανήκουν μόνο σε ένα τμήμα της σχολής στη βάση δεδομένων. Για παράδειγμα: **Department of Informatics**.

1.1 Γενική Περιγραφή της Διεπαφής

- **FrontUsersCommands:** περιλαμβάνει **methods** που δίνουν δικαιώματα στους καθηγητές και στους φοιτητές να δουν τους βαθμούς τους.

1.2 Γενική Περιγραφή των Κλάσεων.

Όπως αναφέραμε υπάρχουν 10 κλάσεις όπου η γενική τους λειτουργία είναι:

- **Department:** καθορίζει το τμήμα που ανήκουν οι φοιτητές, οι καθηγητές, η γραμματεία και τα μαθήματα.
- **User:** είναι abstract class από την οποία υιοθετούν γνωρίσματά της οι φοιτητές, οι καθηγητές και η γραμματεία. Για παράδειγμα και οι τρεις κατηγορίες χρηστών έχουν όνομα, username και password, γνωρίσματα τα οποία υιοθετούν από την κλάση User.
- **Secretary:** είναι η κλάση της γραμματεία. Μέσω αυτής οι γραμματείς μπορούν να κάνουν εγγραφή φοιτητών και καθηγητών σε κάποιο τμήμα, να προβάλλουν στην γραμμή εντολών όλους τους φοιτητές και καθηγητές και τους βαθμούς



όλων των φοιτητών καθώς και να δημιουργήσουν μάθημα ή/και να το συνδέσουν με κάποιον καθηγητή.

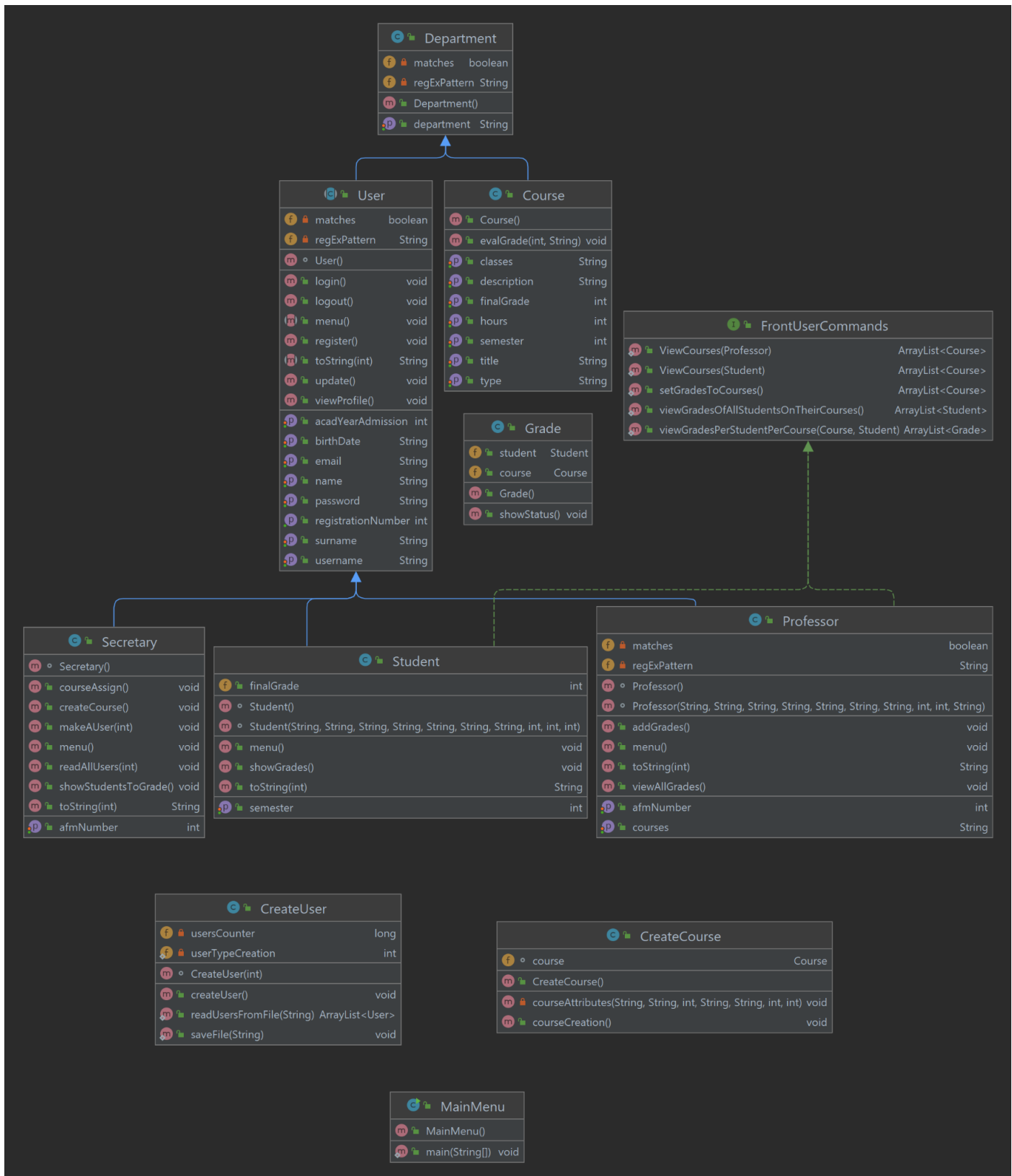
- **Professor:** είναι η κλάση των εκπαιδευτικών. Μέσω αυτής οι εκπαιδευτικοί αποκτούν την δυνατότητα να βλέπουν ποια μαθήματα έχουν αναλάβει, να προσθέτουν βαθμούς στους φοιτητές που έχουν δηλώσει το μάθημα τους και να βλέπουν όλους τους βαθμούς των φοιτητών ανά έτος.
- **Student:** είναι η κλάση των φοιτητών. Οι φοιτητές μπορούν να δουν μόνο τους βαθμούς που έχει περάσει η γραμματεία στα περασμένα μαθήματα.

(Σημείωση: Όταν προσθέσουμε τις βάσεις, μία χρήσιμη πρόσθετη λειτουργία για τους φοιτητές θα ήταν η επιλογή των μαθημάτων επιλογής.)

- **CreateUser:** είναι μία κλάση που χρησιμοποιεί επί των πλείστων η γραμματεία για την δημιουργία και την ανάγνωση προφίλ φοιτητών και καθηγητών.
- **Course:** η κλάση των μαθημάτων. Τα αντικείμενά της έχουν όλα τα γνωρίσματα των μαθημάτων της σχολής. Μερικά γνωρίσματα μαθημάτων είναι: η περιγραφή του μαθήματος, οι απαιτήσεις του και το τμήμα στο οποίο ανήκουν.
- **Grade:** μία κλάση που η κύρια λειτουργία της, για τώρα, είναι να δείχνει τον βαθμό των φοιτητών ανά μάθημα.
- **CreateCourse:** μία κλάση, παρόμοια με την CreateUser, δίνει την δυνατότητα στη γραμματεία να δημιουργεί μαθήματα στα τμήματα της σχολής.
- **MainMenu:** η κλάση στην οποία ανήκει η main συνάρτηση. Εμφανίζει το βασικό menu στον χρήστη μέσω του οποίου μπορεί να περιηγηθεί (με το κατάλληλο authentication - φοιτητής, καθηγητής ή γραμματέας) στις επιλογές της εφαρμογής.



1.3 Διάγραμμα Κλάσεων και Διεπαφών



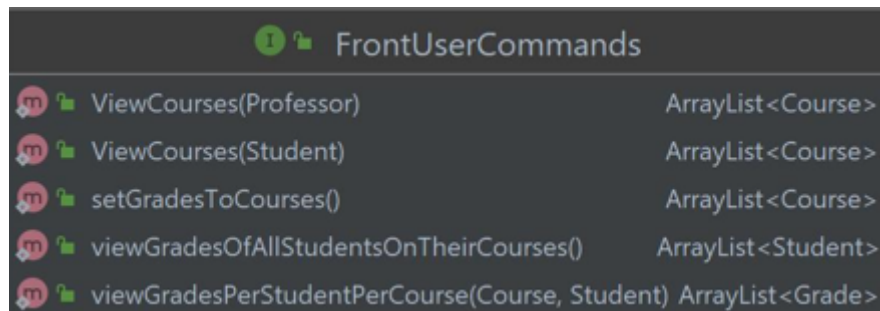
Εικόνα 1: Class Diagram



2. Κώδικας Προγράμματος

2.1 Αναλυτική Περιγραφή Διεπαφής

2.1.1 FrontUserCommands



Εικόνα 2: FrontUserCommands Interface

Στη διεπαφή αυτή έχουμε δημιουργήσει 5 **methods**:

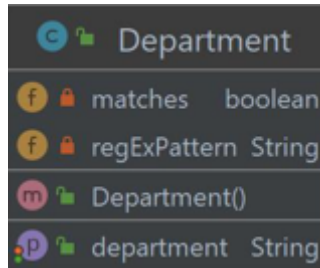
- **ViewCourses(Professor)**: δείχνει τα μαθήματα που έχει αναλάβει ο καθηγητής.
- **ViewCourses(Student)**: δείχνει τα μαθήματα που έχει δηλώσει ο φοιτητής.
- **setGradesToCourse()**: δίνει την δυνατότητα στους εκπαιδευτικούς να προσθέτουν βαθμούς στους φοιτητές.
- **viewGradesOfAllStudentsOnTheirCourses()**: συνάρτηση που χρησιμοποιείται από τα αντικείμενα της κλάσης Professor. Δίνει την δυνατότητα στους εκπαιδευτικούς να βλέπουν όλους τους βαθμούς των φοιτητών στα μαθήματα που τους έχουν ανατεθεί.
- **viewGradesPerStudentPerCourse()**: χρησιμοποιείται από τους καθηγητές και τους φοιτητές. Τους δίνει την δυνατότητα να δουν αναλυτικά την βαθμολογία ενός συγκεκριμένου φοιτητή που έχει δηλώσει ένα συγκεκριμένο μάθημα.



2.2 Αναλυτική Περιγραφή Κλάσεων

Τα χαρακτηριστικά των κλάσεων είναι τα ακόλουθα:

2.2.1 Department



Εικόνα 3: Department Class

Properties:

- **department** περιλαμβάνει το τμήμα το κάθε προσώπου (είτε student είτε professor είτε secretary)

Constructors:

- **Department()** είναι ο default constructor

Fields:

- **matches**(τύπου boolean)
- **regexPattern**(τύπου string)

που χρησιμοποιούνται για την υλοποίηση του ελέγχου του input του χρήστη.

try-catch errors:

InputMismatchException	Βρίσκεται στο setDepartment και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί μη αλφαβητικός χαρακτήρας.
-------------------------------	--



2.2.2 User

User	
matches	boolean
regexPattern	String
User()	
login()	void
logout()	void
menu()	void
register()	void
toString(int)	String
update()	void
viewProfile()	void
acadYearAdmission	int
birthDate	String
email	String
name	String
password	String
registrationNumber	int
surname	String
username	String

Εικόνα 4: User Class

Η κλάση αυτή κληρονομεί τα χαρακτηριστικά του Department.

Το User είναι μια κλάση τύπου abstract και έχει τα ακόλουθα:

Fields:

- **matches**(τύπου boolean)
- **regexPattern**(τύπου string)

που χρησιμοποιούνται για την υλοποίηση του ελέγχου του input του χρήστη.

Constructors:

- **User()** είναι ο default constructor

Methods:

- **login()**
- **logout()**
- **register()**
- **update()**
- **viewProfile()**



που οι λειτουργίες τους θα υλοποιηθούν στο επόμενο στάδιο εκπόνησης της εργασίας.

try-catch errors:

InputMismatchException	Βρίσκεται στο setName , setSurname και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί μη αλφαβητικός χαρακτήρας.
InputMismatchException	Βρίσκεται στο setUsername και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί οποιοσδήποτε αλφαριθμητικό χαρακτήρα και την κάτω παύλα.
InputMismatchException	Βρίσκεται στο setEmail και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί οποιοσδήποτε σειρά διαφορετική της μορφής διεύθυνσης ενός e-mail.
InputMismatchException	Βρίσκεται στο setPassword και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί οποιοσδήποτε σειρά χαρακτήρων όπου το πλήθος τους είναι μικρότερο 8.
InputMismatchException	Βρίσκεται στο setBirthDate και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί οποιασδήποτε μορφή ημερομηνίας εκτός της YYYY-MM-DD.
InputMismatchException	Βρίσκεται στο setAcadYearAdmission και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί οποιασδήποτε χρονολογία μικρότερη ή ίση με το 1990 είτε μεγαλύτερη από την μεγαλύτερη από την σημερινή χρονολογία.

Οι abstract methods:

- **toString()**
- **menu()**

που χρησιμοποιούνται για τις υποκλάσεις ώστε να εξυπηρετηθούν οι χρήστες της εφαρμογής με την χρήση console.

Properties:



- **username** που περιλαμβάνει το username του χρήστη
- **name** που περιλαμβάνει το όνομα του χρήστη
- **surname** που περιλαμβάνει το επίθετο του χρήστη
- **email** που περιλαμβάνει το email του χρήστη
- **birthDate** που περιλαμβάνει την ημερομηνία γέννησης του χρήστη
- **acadYearAdmission** που περιλαμβάνει το έτος εισαγωγής του χρήστη στο τμήμα
- **password** που περιλαμβάνει τον κωδικό του χρήστη
- **registrationNumber** που περιλαμβάνει τον αριθμό μητρώου του χρήστη. Στη παρούσα φάση του project δεν χρησιμοποιείται η μεταβλητή αυτή σε κάποιο κομμάτι της εφαρμογής, μέχρι να της εντάξουμε βάση δεδομένων όπου θα καθοριστεί και η κατάλληλη μορφή της με βάση τα τμήματα που θα έχει η σχολή.

2.2.3 CreateUser

CreateUser		
f	usersCounter	long
f	userTypeCreation	int
m	CreateUser(int)	
m	createUser()	void
m	readUsersFromFile(String)	ArrayList<User>
m	saveFile(String)	void

Εικόνα 5: CreateUser Class

Fields:

- **usersCounter**(τύπου long) που απαριθμεί το πλήθος των χρηστών που υπάρχουν εγγεγραμμένοι στο πρόγραμμα (είναι το άθροισμα των φοιτητών και των καθηγητών)
- **userTypeCreation**(τύπου integer) που βοηθάει για να επιλέξουμε το τι τύπος χρήστη είναι αυτός που πάμε να δημιουργήσουμε (δηλαδή αν θα είναι student ή θα είναι professor)

Constructors:

- **CreateUser(int)** όπου η παράμετρος (int) δηλώνει την επιλογή η οποία έχει γίνει για τον τύπο του χρήστη που θα δημιουργηθεί (δηλαδή αν θα είναι student ή θα είναι professor)

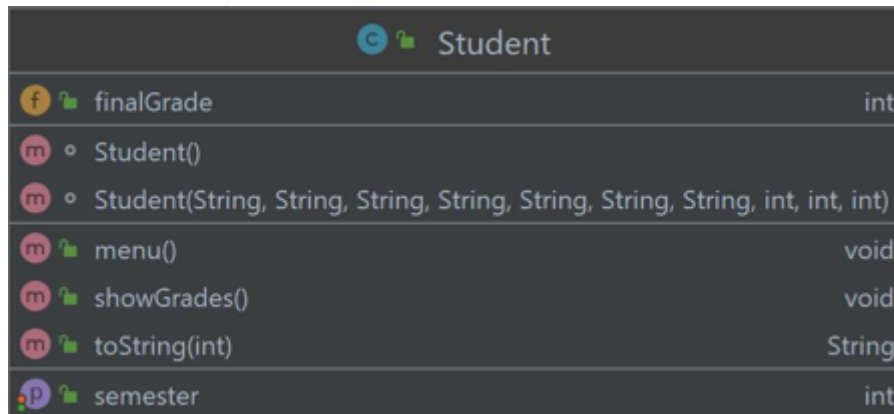
Methods:

- **createUser()** η οποία χρησιμοποιείται για να προσθέσει στο αντικείμενο τα attributes που δίνει ο χρήστης κάνοντας και τους ανάλογους ελέγχους για την εισαγωγή των δεδομένων



- **saveFile()** η οποία αναλαμβάνει την αποθήκευση των αντικειμένων που έχουν δημιουργηθεί (στο αρχείο `student.txt` αν το αντικείμενο είναι τύπου `student` και στο αρχείο `professor.txt` αν το αντικείμενο είναι τύπου `professor`)
- **readUsersFromFile()** που αναλαμβάνει την ανάγνωση των αντικειμένων-χρηστών από το κατάλληλο αρχείο - `students.txt` ή `professors.txt`.

2.2.4 Student



Student		
finalGrade		int
Student()		
Student(String, String, String, String, String, String, String, int, int, int)		
menu()		void
showGrades()		void
toString(int)		String
semester		int

Εικόνα 6: Student Class

Η κλάση αυτή κληρονομεί τα χαρακτηριστικά του User.

Το Student έχει τα ακόλουθα:

Fields:

- **finalGrade** η οποία είναι υπεύθυνη για την τελική βαθμολογία του εκάστοτε φοιτητή (student).

Constructors:

- **Student()** όπου δίνουμε προκαθορισμένες τιμές στα attributes του φοιτητή (student)
- **Student(String, String, ..., int)** που δέχεται ως ορίσματα τα attributes του φοιτητή (student) τα οποία έχει δώσει ο χρήστης.

Methods:

- **menu()** η οποία χρησιμοποιείται για την εμφάνιση του menu στην κονσόλα (console) και παρέχει τις επιλογές στον χρήστη για να περιηγηθεί είτε στις βαθμολογίες του είτε να βγει από την εφαρμογή.
- **showGrades()** όπου εμφανίζει στον χρήστη τις βαθμολογίες του (θα υλοποιηθεί σε επόμενο στάδιο, προς το παρόν εμφανίζει ένα μήνυμα ότι ο χρήστης δεν έχει λάβει κάποια βαθμολογία)
- **toString()** η οποία αναλαμβάνει την επιστροφή ενός επιτρεπτού string με όλα τα attributes του φοιτητή (student). Σε περίπτωση που δεν είναι δυνατή η



υλοποίηση του string αυτού με τα σωστά ορίσματα εμφανίζεται μια εξαίρεση (exception) για λάθος κατάσταση (IllegalStateException).

Properties:

- **semester** που περιλαμβάνει το εξάμηνο που βρίσκεται ο φοιτητής.



try-catch errors:

InputMismatchException	Βρίσκεται στο <code>setSemester</code> και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί αριθμός είτε μικρότερος ή ίσος του 0 είτε μεγαλύτερος του 8.
-------------------------------	--

2.2.5 Professor

Professor		
	<code>matches</code>	<code>boolean</code>
	<code>regExpPattern</code>	<code>String</code>
	<code>Professor()</code>	
	<code>Professor(String, String, String, String, String, String, String, int, int, ArrayList<String>)</code>	
	<code>addGrades()</code>	<code>void</code>
	<code>menu()</code>	<code>void</code>
	<code>toString(int)</code>	<code>String</code>
	<code>viewAllGrades()</code>	<code>void</code>
	<code>afmNumber</code>	<code>int</code>
	<code>courses</code>	<code>ArrayList<String></code>

Εικόνα 7: Professor Class

Η κλάση αυτή κληρονομεί τα χαρακτηριστικά του `User`.

Το `Professor` έχει τα ακόλουθα:

Fields:

- `matches` (τύπου `boolean`)
- `regExpPattern` (τύπου `string`)

που χρησιμοποιούνται για την υλοποίηση του ελέγχου του input του χρήστη.

Constructors:

- `Professor()` όπου δίνουμε προκαθορισμένες τιμές στα attributes του καθηγητή
- `Professor(String, String, ..., int, int)` που δέχεται ως ορίσματα τα attributes του καθηγητή (professor) τα οποία έχει δώσει ο χρήστης κατά την δημιουργία του object - καθηγητής.

Methods:

- `toString()` η οποία αναλαμβάνει την επιστροφή ενός επιτρεπτού string με όλα τα attributes του καθηγητή (professor). Σε περίπτωση που δεν είναι δυνατή η



υλοποίηση του string αυτού με τα σωστά ορίσματα εμφανίζεται μια εξαίρεση (exception) για λάθος κατάσταση (IllegalStateException).

- **menu()** η οποία χρησιμοποιείται για την εμφάνιση του menu στην κονσόλα (console) και παρέχει τις επιλογές στον χρήστη για να περιηγηθεί είτε για να προσθέσει κάποια βαθμολογία σε κάποιο φοιτητή του είτε να κάνει προβολή όλων των βαθμών είτε να βγει από την εφαρμογή,
- **viewAllGrades()** που εμφανίζει όλες τις βαθμολογίες (θα υλοποιηθεί σε επόμενο στάδιο, προς το παρόν εμφανίζει ένα μήνυμα ότι όλοι οι φοιτητές έχουν ένα βαθμό σε ένα ή σε όλα τα μαθήματα).
- **addGrades()** που προσθέτει έναν βαθμό σε κάποιο μάθημα. (Θα υλοποιηθεί σε επόμενο στάδιο. Προς το παρόν εμφανίζει ένα μήνυμα για την εισαγωγή ενός τίτλου μαθήματος, ένα άλλο ότι αυτή είναι η λίστα των φοιτητών που πρέπει να βαθμολογηθεί και ένα τελευταίο μήνυμα ότι οι βαθμοί που εισήγαγε ο χρήστης αποθηκεύτηκαν).

Properties:

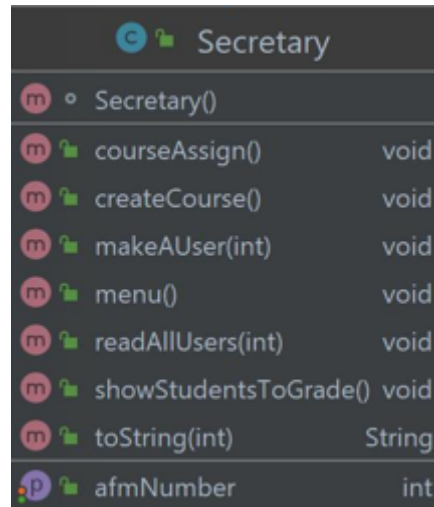
- **afmNumber** περιλαμβάνει τον ΑΦΜ (Αριθμό Φορολογικού Μητρώου) του εκάστοτε καθηγητή
- **courses** είναι τύπου ArrayList και περιέχει τα μαθήματα που διδάσκει ο εκάστοτε καθηγητής

try-catch errors:

InputMismatchException	Βρίσκεται στο setAfmNumber και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί αριθμός είτε μικρότερος ή ίσος του 100000000 είτε μεγαλύτερος του 999999999.
InputMismatchException	Βρίσκεται στο setCourses και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί μη αλφαβητικός χαρακτήρας.



2.2.6 Secretary



Εικόνα 8: Secretary Class

Η κλάση αυτή κληρονομεί τα χαρακτηριστικά του User.

Constructors:

- **Secretary()** ορίζει τις default τιμές του object secretary

Methods:

- **courseAssign()** αναθέτει μαθήματα στους καθηγητές
- **createCourse()** δημιουργεί νέο μάθημα
- **makeAUser(int)** δημιουργεί νέο χρήστη (φοιτητή ή καθηγητή)
- **menu()** το menu που εμφανίζεται στον χρήστη όταν αυτός συνδέεται με ιδιότητες γραμματείας
- **readAllUsers(int)** εμφανίζει όλους τους χρήστες που έχουν δημιουργεί
- **showStudentsToGrades()** δείχνει τους φοιτητές που χρειάζονται βαθμολόγηση
- **toString(int)** επιστρέφει στη συνάρτηση από την οποία καλέσθηκε ένα string με τα ορίσματα του secretary

Properties:

- **afmNumber** το οποίο παίρνει τον αριθμό Α.Φ.Μ του πολίτη

try-catch errors:

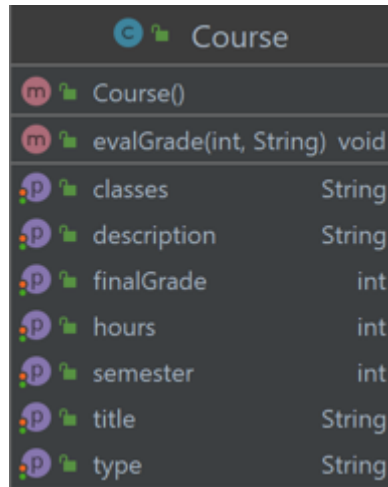
InputMismatchException

Βρίσκεται στο **setAfmNumber** και σταματάει το πρόγραμμα σε περίπτωση που εισαχθεί αριθμός είτε μικρότερος ή ίσος του 100000000 είτε μεγαλύτερος του 999999999.





2.2.7 Course



Course	
Course()	
evalGrade(int, String)	void
classes	String
description	String
finalGrade	int
hours	int
semester	int
title	String
type	String

Εικόνα 9: Course Class

Constructors:

- **Course()** είναι ο default constructor

Methods:

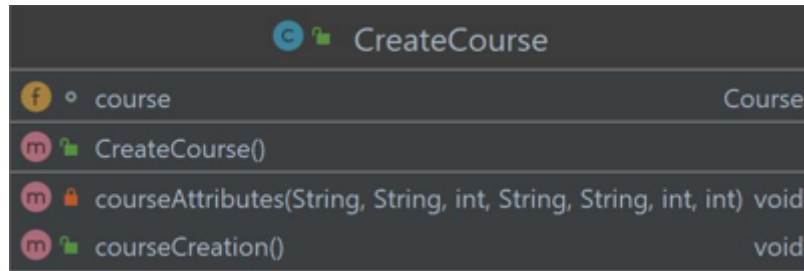
- **evalGrade(int, String)** υπολογίζει τον βαθμό τους που αναθέτει ο καθηγητής στον φοιτητή

Properties:

- **classes** αποθηκεύει τις αίθουσες που πραγματοποιούνται τα τμήματα
- **description** αποθηκεύει την περιγραφή του μαθήματος
- **finalGrade** αποθηκεύει τον τελικό βαθμό του φοιτητή
- **hours** αποθηκεύει τις εβδομαδιαίες ώρες διδασκαλίας
- **semester** αποθηκεύει το εξάμηνο που διδάσκεται το μάθημα
- **title** αποθηκεύει τον τίτλο του μαθήματος
- **type** αποθηκεύει τον τύπο του μαθήματος (υποχρεωτικό ή μη)



2.2.8 CreateCourse



Εικόνα 10: CreateCourse Class

Field:

- **course** είναι object της κλάσης **Course**

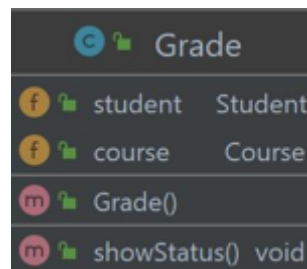
Constructors:

- **CreateCourse()** είναι ο default constructor

Methods:

- **courseAttributes()** αποθηκεύει τα fields του νέου μαθήματος
- **courseCreation()** δημιουργούμε μέσα από αυτή τη μέθοδο ένα νέο μάθημα (αντικείμενο του Course)

2.2.9 Grade



Εικόνα 11: Grade Class

Fields:

- **student** αντικείμενο τύπου **Student**
- **course** αντικείμενο τύπου **Course**

Constructor:

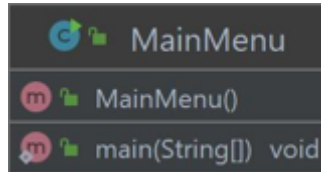
- **Grade()** είναι ο default constructor

Method:

- **showStatus()** δείχνει την κατάσταση του βαθμού ενός φοιτητή σε ένα μάθημα



2.2.10 MainMenu



Εικόνα 12: MainMenu Class

Constructor:

- **MainMenu()** είναι ο default constructor

Method:

- **main(String[])** είναι η βασική συνάρτηση του προγράμματος. Αυτή περιλαμβάνει και το βασικό menu της εφαρμογής



3. Εκτέλεση Προγράμματος

1. Όταν ο χρήστης τρέξει την εφαρμογή στη console, του εμφανίζεται ένα menu επιλογών:

```
Welcome to the grade system of our University.  
1. Sign in as a student  
2. Sign in as a professor  
3. Sign in as a secretary  
4. Exit  
  
Please select an option: (we accept numbers only)
```

Επιλογές 1: Αρχικό Μενού Επιλογών

2. Ανάλογα με την ιδιότητα που έχει στο πανεπιστήμιο, επιλέγει και την κατάλληλη επιλογή. (Στο πρώτο στάδιο της εργασίας λειτουργεί ουσιαστικά μόνο η επιλογή του γραμματέα)

```
You logged in as a student  
  
1. Show All Grades  
2. Exit the app  
  
Please select an option: (we accept numbers only)
```

Επιλογές 2: Αν συνδεθεί ο χρήστης ως φοιτητής

```
You logged in as a professor  
  
1. Add Grades to your students based on the course  
2. Show all Grades  
3. Exit the app  
  
Please select an option: (we accept numbers only)
```

Επιλογές 3: Αν συνδεθεί ο χρήστης ως καθηγητής



You logged in as a secretary

1. Create a Student
2. Read all Students
3. Create a Professor
4. Read all professors
5. Create a Course
6. Exit the app

Please select an option: (we accept numbers only)

Επιλογές 4: Αν συνδεθεί ο χρήστης ως γραμματέας

3. Αν ο γραμματέας επιλέξει τις επιλογές 1 ή 3, μπορεί να δημιουργήσει έναν νέο φοιτητή ή καθηγητή αντίστοιχα, συμπληρώνοντας κάθε φορά τα ορίσματα που του ζητούνται στην console:

```
Department:
Informatics
Name:

Invalid name! Please try again...
Nick
Surname:
Mike
Username:
```

Επιλογές 5: Συμπληρώνοντας τα στοιχεία ενός χρήστη

Παρατηρούμε τα μηνύματα λάθος εισαγωγής που έχουν προστεθεί σε κάθε όρισμα ώστε ο χρήστης να είναι έγκυρος στη μελλοντική βάση δεδομένων που θα προσθέσουμε.

4. Αν επιλέξει τις επιλογές 2 ή 4 εκτυπώνεται στον terminal οι εγγεγραμμένοι φοιτητές ή καθηγητές που υπάρχουν στο σύστημα/βάση δεδομένων.

```
Please select an option: (we accept numbers only)
2
You are going to read all students profile
Student = [Department = Informatics, Name = Apostolis, Surname = Siampanis, Username = Apostoli13, Email = apostolis@gmail.com, Pa:
```

Επιλογές 6: Αν επιλέξει ο χρήστης φοιτητές



```
Please select an option: (we accept numbers only)
```

```
4
```

```
You are going to create a student profile
```

```
Professor = [Department = Informatics, Name = Theo, Surname = Koxanoglou, Username = theo12, Email = theokoxan@gmail.com, Password = hj
```

Επιλογές 7: Αν επιλέξει ο χρήστης καθηγητές

5. Όταν θελήσει ο χρήστης να κλείσει την εφαρμογή, επιλέγει την επιλογή 4:

```
4. Exit
```

```
Please select an option: (we accept numbers only)
```

```
4
```

```
You will exit the app. See you soon!
```

Επιλογές 8: Κλείσιμο Εφαρμογής



Βιβλιογραφικές Πηγές

1. **ΕΡΓΑΣΤΗΡΙΑΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ (Java)**, Retrieve 18/04/2022 from <https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMB117&openDir=/201102161747563cexy8tj>
2. **ΠΑΡΟΥΣΙΑΣΕΙΣ**, Retrive 19/04/2022 from <https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMB117&openDir=/20110216172215d4kp21g1>