

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
Τμήμα Πληροφορικής



Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον  
παγκόσμιο ιστό»

Τίτλος Εργασίας:  
«Εφαρμογή διαχείρισης ραντεβού τακτικών ιατρικών εξετάσεων»

Άσκηση 01	<b>Θεμελιώδεις Κλάσεις και Βασική Λειτουργικότητα</b>
Όνομα φοιτητή – Αρ. Μητρώου	Κωνσταντίνος Αργυρίου – Π19017
	Ιωακείμ Ελ-Χαττάμπ Μπριστογιάννης – Π19048
	Σοφία Μαρκοπούλου – Π19097
	Μυρτώ-Μαρία Σπάθα – Π19156
Ημερομηνία παράδοσης	17/04/2021



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	<a href="#">Εκφώνηση της Άσκησης</a>	2
2	<a href="#">Γενική Περιγραφή της λύσης</a>	2
3	<a href="#">Κώδικας Προγράμματος</a>	2
4	<a href="#">Διάγραμμα Κλάσεων</a>	5
5	<a href="#">Στιγμιότυπα από την εκτέλεση του προγράμματος</a>	5
6	<a href="#">Βιβλιογραφικές Πηγές</a>	6

### 1. Εκφώνηση της άσκησης

[ΑΣΚΗΣΗ-1-2021.pdf](#)

### 2. Γενική Περιγραφή της λύσης

Σκοπός της γενικής εργασίας ήταν η ανάπτυξη μιας εφαρμογής η οποία θα δίνει τη δυνατότητα σε ασθενείς και γιατρούς να προγραμματίζουν ραντεβού μεταξύ τους με τη συμβολή κάποιων διαχειριστών οι οποίοι εισάγουν τους γιατρούς στο σύστημα. Πιο συγκεκριμένα, σκοπός της 1<sup>ης</sup> εργασίας, η οποία αποτελεί και το αντικείμενο της συγκεκριμένης παρουσίασης, ήταν να σχεδιαστούν οι βασικές κλάσεις της εφαρμογής και να εκτελεστούν πρωταρχικές της λειτουργίες. Συνοπτικά, αναπτύχθηκαν οι κλάσεις User (όπου περιλαμβάνονται όλοι οι εγγεγραμμένοι χρήστες της εφαρμογής), Doctor (όπου ανήκουν όλοι οι γιατροί του φορέα<sup>1</sup> για τον οποίο προορίζεται η εφαρμογή), Patient (όπου εντάσσονται οι ασθενείς που έχουν δεχθεί ή πρόκειται να δεχθούν ιατρικές υπηρεσίες), Admin (όπου ορίζεται ο διαχειριστής του συστήματος), Appointment (όπου προγραμματίζονται τα ραντεβού) και CreateUsers (όπου βρίσκεται η εκτελέσιμη συνάρτηση main και είναι το σημείο όπου δημιουργούνται ασθενείς, γιατροί και διαχειριστές).

### 3. Κώδικας προγράμματος

Ακολουθεί η αναλυτική περιγραφή του προγράμματος.

#### 3.1 Κώδικας της κλάσης User

Η κλάση **User** απεικονίζει τους χρήστες κάθε ιδιότητας (γιατροί, ασθενείς, διαχειριστές) της εφαρμογής. Βασικές της λειτουργίες είναι η εγγραφή νέων χρηστών στο σύστημα, η παραχώρηση πρόσβασης σε εγγεγραμμένους χρήστες ώστε να μπορούν να χρησιμοποιήσουν την εφαρμογή και η τροποποίηση των χαρακτηριστικών του προφίλ των χρηστών. Κάθε χρήστης διαθέτει όνομα χρήστη (ψευδώνυμο για την περιήγηση στην εφαρμογή), κωδικό πρόσβασης, όνομα και επώνυμο (username, password, name, surname αντίστοιχα). Η κατάσταση εισόδου του (ενεργός ή



ανενεργός) προσδιορίζεται από τη μεταβλητή login. Το πλήθος των εγγεγραμμένων χρηστών στο σύστημα αποθηκεύεται στη μεταβλητή usersCounter. Αρχικές τιμές σε αυτά τα χαρακτηριστικά δίνονται μέσω του Constructor.

Όσον αφορά στις μεθόδους, αρχικά ορίζονται Setters για τα χαρακτηριστικά username, password, name, surname και Getters για τα username, name, surname, usersCounter. Επίσης, ορίζεται η συνάρτηση login. Η λειτουργία της είναι να δίνει τιμή true στη μεταβλητή login σε περίπτωση που το όνομα χρήστη και ο κωδικός πρόσβασης του εγγεγραμμένου χρήστη που θέλει να χρησιμοποιήσει την εφαρμογή είναι σωστά. Σε αντίθετη περίπτωση, γίνεται χρήση της εξαίρεσης LoginFailure (η LoginFailure κληρονομεί από την Exception). Τέλος, ορίζονται οι συναρτήσεις logout (δίνει την τιμή false στη login), isLoggedIn (επιστρέφει την τιμή της login) και increaseUsers (αυξάνει την τιμή της userCounter κατά 1).

### 3.2 Κώδικας της κλάσης Doctor

Η κλάση **Doctor** απεικονίζει τους γιατρούς που είναι εγγεγραμμένοι στην εφαρμογή. Κάθε γιατρός έχει όλα τα χαρακτηριστικά ενός χρήστη (κληρονομεί από την κλάση **User**) και επιπλέον χαρακτηρίζεται από την ειδικότητά του specialty. Διαθέτει επίσης ένα μηνιαίο ωρολόγιο πρόγραμμα ραντεβού schedule (ξεκινώντας από την τρέχουσα ημερομηνία) και μια λίστα times (περιέχει όλες τις ημέρες του προγράμματος schedule). Τα χαρακτηριστικά αυτά αρχικοποιούνται μέσω του Constructor.

Όσον αφορά στις μεθόδους, αρχικά ορίζεται ένας Setter για το specialty και Getters για τα specialty, schedule. Επιπλέον, ορίζονται οι availability (εκτυπώνει τη λέξη 'closed' δίπλα από την ημερομηνία και ώρα, αν η συγκεκριμένη ημερομηνία και ώρα δεν είναι διαθέσιμη για ραντεβού), generateSchedule (προσθέτει 30 μέρες στη λίστα times, ξεκινώντας από την τωρινή ημερομηνία), generateTimes (προσθέτει στη λίστα schedule τις ημέρες της λίστας times προσθέτοντας σε κάθε ημέρα τις εργάσιμες ώρες του φορέα) και generateDoctorSchedule (εκτελεί τις μεθόδους generateSchedule και generateTimes). Ακολουθούν οι μέθοδοι setDnd (κατοχυρώνει μια ημερομηνία και ώρα για έναν εικονικό ασθενή<sup>ii</sup> σε περίπτωση που ο γιατρός θέλει να εξασφαλίσει ότι δεν θα υπάρχει ραντεβού την συγκεκριμένη ώρα και εκτελεί τη συνάρτηση makeAppointment γι' αυτόν τον ασθενή –βλ. 3.5) και replaceDnd (αντικαθιστά τον εικονικό ασθενή με έναν πραγματικό ασθενή σε περίπτωση που γιατρός θέλει να ορίσει ένα ραντεβού τη συγκεκριμένη ώρα). Τέλος, ορίζονται οι συναρτήσεις program (εκτελεί τη συνάρτηση showAllAppointments –βλ. 3.5), searchAppointment (εκτελεί τη συνάρτηση showAppointment –βλ. 3.5) και removeAppointment (εκτελεί τη συνάρτηση removeAppointment –βλ. 3.5).

### 3.3 Κώδικας της κλάσης Patient

Η κλάση **Patient** απεικονίζει τους ασθενείς που είναι εγγεγραμμένοι στην εφαρμογή. Κάθε ασθενής έχει όλα τα χαρακτηριστικά ενός χρήστη (κληρονομεί από την κλάση **User**) και επιπλέον χαρακτηρίζεται από το ΑΜΚΑ του amka. Διαθέτει επίσης μια λογική μεταβλητή isLock η οποία λαμβάνει την τιμή true αν ένας ασθενής είναι εικονικός<sup>ii</sup>. Τα χαρακτηριστικά αυτά αρχικοποιούνται μέσω του Constructor.

Όσον αφορά στις μεθόδους, ορίζονται οι getAmka (επιστρέφει το ΑΜΚΑ), searchAppointmentByDoctorName (εκτελεί τη συνάρτηση showAppointmentsByDoctorName –βλ. 3.5), searchAppointmentByDoctorSpecialty (εκτελεί τη συνάρτηση showAppointmentsBySpecialty -



βλ. 3.5), showScheduledAppointments (εκτελεί τη συνάρτηση showPatientAppointment –βλ. 3.5) και showAppointmentsHistory (εκτελεί τη συνάρτηση showPatientHistory –βλ. 3.5).

### 3.4 Κώδικας της κλάσης Admin

Η κλάση **Admin** απεικονίζει τους διαχειριστές του συστήματος της εφαρμογής. Βασική της λειτουργία είναι η εισαγωγή ενός γιατρού στο σύστημα. Κάθε διαχειριστής έχει όλα τα χαρακτηριστικά ενός χρήστη (κληρονομεί από την κλάση **User**) και επιπλέον διαθέτει μια λίστα doctors με τους γιατρούς που έχει εισάγει ο ίδιος. Οι μέθοδοι που διαθέτει είναι ο constructor, addDoctor (εισάγει ένα γιατρό στη doctors) και removeDoctor (διαγράφει ένα γιατρό από τη doctors).

### 3.5 Κώδικας της κλάσης Appointment

Η κλάση **Appointment** απεικονίζει τις ενέργειες που είναι απαραίτητο να πραγματοποιηθούν ώστε να προγραμματιστεί ένα ραντεβού. Διαθέτει έναν πίνακα κατακερματισμού appointments με το πρόγραμμα των ραντεβού για κάθε γιατρό και έναν πίνακα κατακερματισμού doctors με όλους τους γιατρούς. Τα χαρακτηριστικά αυτά αρχικοποιούνται μέσω του Constructor.

Όσον αφορά στις μεθόδους, ορίζονται οι makeAppointment (προσθέτει ασθενή, ημερομηνία και ώρα στο πρόγραμμα του καθορισμένου γιατρού και το ραντεβού στην appointments), removeAppointment (αφαιρεί ένα ραντεβού από το πρόγραμμα του καθορισμένου γιατρού σε περίπτωση που αυτό είναι προγραμματισμένο για τουλάχιστον 3 μέρες αργότερα από την τωρινή ημερομηνία –μόνο ο γιατρός έχει δικαιοδοσία να διαγράψει ραντεβού, γι' αυτό και πρώτα εξετάζεται αν αυτός είναι συνδεδεμένος), showAllAppointments (εκτυπώνει το μηνιαίο ωρολόγιο πρόγραμμα του γιατρού), showAppointment (εξετάζει αν υπάρχει κάποιο διαθέσιμο ραντεβού για τον καθορισμένο γιατρό σε συγκεκριμένη ημερομηνία και ώρα), showPatientAppointment (εμφανίζει όλα τα επερχόμενα ραντεβού ενός συγκεκριμένου ασθενή), showPatientHistory (εμφανίζει όλα τα παρελθοντικά ραντεβού ενός ασθενή), showAppointmentsByDoctorName (εμφανίζει όλα τα ραντεβού ενός γιατρού, μετά από αναζήτηση του ονόματός του από τον ασθενή), showAppointmentsBySpecialty (εμφανίζει όλα τα ραντεβού όλων των γιατρών μιας ειδικότητας και τα ονόματά τους).

### 3.6 Κώδικας της κλάσης CreateUsers

Η κλάση **CreateUsers** είναι η κλάση στην οποία υλοποιείται η main συνάρτηση του προγράμματος. Οι ενέργειες που εκτελούνται είναι η δημιουργία ενός ασθενή και ενός γιατρού καθώς και η δημιουργία ενός ασθενή του οποίου τα στοιχεία εξάγονται από ένα αρχείο Patient.txt. Επιπρόσθετες μέθοδοι που ορίζονται είναι οι makePatient (δημιουργία αντικειμένου τύπου Patient μετά από εισαγωγή των στοιχείων του από το πληκτρολόγιο και εκτέλεση της μεθόδου login), makeDoctor (δημιουργία αντικειμένου τύπου Doctor μετά από εισαγωγή των στοιχείων του από το πληκτρολόγιο και εκτέλεση της μεθόδου login) και makePatientFromFile (δημιουργία αντικειμένου τύπου Patient μετά από εξαγωγή των στοιχείων του από το αρχείο Patient.txt).

**ΠΡΟΣΟΧΗ!!!** Σε περίπτωση που η μέθοδος makePatientFromFile δεν εκτελείται σωστά, δοκιμάστε να εισάγετε την απόλυτη διαδρομή του αρχείου Patient.txt μέσα στο όρισμα της new File().



## 4. Διάγραμμα Κλάσεων

[class\\_diagram.pdf](#)

## 5. Στιγμιότυπα από την εκτέλεση του προγράμματος

```
Adding a patient...
Give me a username:
kwstas534
Give me a password:
kwstas?23#
Give me a name:
kwstantinos
Give me a surname:
papadopoulos
Give me a amka:
9123123130
Please login:
username:
kwstas534
password:
kwstas?23#
Logged in!!
```

Εικόνα 5.1

```
Adding a doctor....
Give me a username:
mitsos
Give me a password:
mitsos#@
Give me a name:
dimitrhs
Give me a surname:
papadimitriou
Give me specialty:
nurse
Please login:
username:
mitsos
password:
mitsos#@
Logged in!!
```

Εικόνα 5.2



## 6. Βιβλιογραφικές Πηγές

1. *Class HashMap<K, V>*. (n.d.). Ανακτήθηκε από:  
<https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>
2. *Class HashSet<E>*. (n.d.). Ανακτήθηκε από:  
<https://docs.oracle.com/javase/7/docs/api/java/util/HashSet.html>
3. *Class Date*. (n.d.). Ανακτήθηκε από:  
<https://docs.oracle.com/javase/7/docs/api/java/util/Date.html>
4. Garg, V. *HashMap in Java with Examples*. (2020). Ανακτήθηκε από:  
<https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/>
5. Singh, D. *HashSet in Java*. (2020). Ανακτήθηκε από:  
<https://www.geeksforgeeks.org/hashset-in-java/>
6. Landup, D. *How to Get Current Date and Time in Java*. (n.d.). Ανακτήθηκε από:  
<https://stackabuse.com/how-to-get-current-date-and-time-in-java/>
7. Gupta, L. *Java – Get Next and Previous Date*. (n.d.). Ανακτήθηκε από:  
<https://howtodoinjava.com/java/date-time/java8-next-previous-date/>

---

<sup>i</sup> Φορέας: το νοσοκομείο, ιατρικό κέντρο, κλπ. για το οποίο προορίζεται η εφαρμογή.

<sup>ii</sup> Εικονικός Ασθενής: ορίζεται ένας εικονικός ασθενής για κάθε εργάσιμη ώρα (σύμφωνα με το πρόγραμμα του φορέα) που ο γιατρός δεν είναι διαθέσιμος, ώστε να φαίνεται ότι υπάρχει ήδη ένα ραντεβού στη συγκεκριμένη ώρα.