



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εξάμηνο 7ο

**Τελική εργασία στο μάθημα Τεχνολογία Πολυμέσων**

## Περιγραφή Εφαρμογής

Στα πλαίσια της εργασίας θα υλοποιηθεί ένα σύστημα διαχείρισης εργασιών (Task Management System). Η εφαρμογή θα επιτρέπει στον χρήστη την δημιουργία, επεξεργασία, και εποπτεία των διαθέσιμων εργασιών. Επιπλέον, ο χρήστης θα μπορεί να διαχειρίζεται πολλαπλές εργασίες, να ορίζει προτεραιότητες και προθεσμίες, και να λαμβάνει υπενθυμίσεις για επερχόμενες εργασίες.

### A.1. Σχεδιασμός και υλοποίηση λογικής (40%)

Στη συνέχεια περιγράφονται οι δυνατότητες που θα πρέπει να παρέχει η εφαρμογή στον χρήστη σχετικά με την δημιουργία και διαχείριση εργασιών και υπενθυμίσεων.

#### Προσθήκη, τροποποίηση, και διαγραφή εργασιών

Ο χρήστης θα μπορεί να δημιουργεί νέες εργασίες. Οι σχετικές πληροφορίες για μια εργασία θα πρέπει να περιλαμβάνουν: τίτλο, περιγραφή, κατηγορία, προτεραιότητα, προθεσμία ολοκλήρωσης, και κατάσταση. Για λόγους απλότητας η προθεσμία ολοκλήρωσης μιας εργασίας δεν θα περιλαμβάνει ώρα, αλλά θα ορίζεται σε επίπεδο ημέρας (π.χ. 14/12/24)

Η κατάσταση μιας εργασίας θα είναι πάντα μια από τις ακόλουθες: *“Open”*, *“In Progress”*, *“Postponed”*, *“Completed”*, και *“Delayed”*. Για κάθε νέα εργασία η προκαθορισμένη κατάσταση θα είναι *“Open”*. Κατά την αρχικοποίηση της εφαρμογής θα πρέπει να εντοπίζονται οι εργασίες που δεν είναι *“Completed”* και έχει περάσει η προθεσμία ολοκλήρωσης και να αλλάζει αυτόματα η κατάσταση σε *“Delayed”*.

Ο χρήστης θα μπορεί να τροποποιήσει όλα τα στοιχεία μιας εργασίας καθώς και να προχωρήσει στην διαγραφή εργασιών. Στην περίπτωση διαγραφών πρέπει να φροντίσετε να γίνεται η κατάλληλη ενημέρωση των πιθανών υπενθυμίσεων που έχουν οριστεί για την προς διαγραφή εργασία.

### **Προσθήκη, τροποποίηση, και διαγραφή κατηγορίας**

Ο χρήστης θα μπορεί να ορίζει νέες κατηγορίες, δίνοντας το σχετικό όνομα. Επιπλέον, θα μπορεί να τροποποιήσει τον όνομα μιας κατηγορίας. Για λόγους απλότητας θεωρούμε ότι δεν υπάρχουν υποκατηγορίες.

Επίσης, θα μπορεί να διαγράφει μια κατηγορία μαζί με την αυτόματη διαγραφή όλων των εργασιών που ανήκουν σε αυτή. Σε αυτή την περίπτωση θα πρέπει να γίνεται η κατάλληλη ενημέρωση των υπενθυμίσεων για τις εργασίες που διαγράφονται.

### **Προσθήκη, τροποποίηση, και διαγραφή προτεραιότητας**

Η εφαρμογή θα πρέπει να περιλαμβάνει ένα προκαθορισμένο επίπεδο προτεραιότητας με όνομα *“Default”*. Ο χρήστης θα μπορεί να ορίζει νέα επίπεδα προτεραιότητας, δίνοντας το σχετικό όνομα. Επιπλέον, θα μπορεί να τροποποιεί τον όνομα ενός επιπέδου προτεραιότητας καθώς και να διαγράφει επίπεδα προτεραιότητας.

Οι λειτουργίες αλλαγής ονόματος και διαγραφής θα ισχύουν για όλα τα επίπεδα προτεραιότητας εκτός από το προκαθορισμένο. Επίσης, όταν γίνεται διαγραφή ενός επιπέδου προτεραιότητας τότε αυτόματα σε όλες τις εργασίες που ανήκουν στο σχετικό επίπεδο θα πρέπει να ανατίθεται το προκαθορισμένο επίπεδο προτεραιότητας.

### **Επιπρόσθετες λειτουργίες**

**Ορισμός και διαχείριση υπενθυμίσεων:** Ο χρήστης θα μπορεί να δημιουργεί υπενθυμίσεις για τις εργασίες. Μια υπενθύμιση θα σχετίζεται πάντα με μια εργασία, ενώ για μια εργασία μπορούν να οριστούν πολλαπλές υπενθυμίσεις. Αν μια εργασία έχει κατάσταση *“Completed”* δεν θα υπάρχει η δυνατότητα ορισμού υπενθυμίσεων, ενώ όταν ο χρήστης αλλάζει την κατάσταση μιας εργασίας σε *“Completed”* τότε η εφαρμογή θα διαγράφει αυτόματα όλες τις υπενθυμίσεις σχετίζονται με την συγκεκριμένη εργασία.

Η εφαρμογή θα πρέπει να υποστηρίζει τους ακόλουθους τύπους υπενθυμίσεων: (i) μια ημέρα πριν την προθεσμία, (ii) μια εβδομάδα πριν την προθεσμία, (iii) ένα μήνα πριν την προθεσμία, (iv) συγκεκριμένη ημερομηνία που θα ορίζει ο χρήστης. Θα πρέπει να υλοποιηθούν οι κατάλληλοι έλεγχοι ώστε ο επιλεγμένος τύπος υπενθύμισης να έχει νόημα με βάση την προθεσμία ολοκλήρωσης της εργασίας και στην περίπτωση που υπάρχει θέμα να ενημερώνεται ο χρήστης με το αντίστοιχο μήνυμα. Τέλος, ο χρήστης θα μπορεί να τροποποιεί και να διαγράφει υπενθυμίσεις.

**Αναζήτηση εργασιών.** Ο χρήστης θα έχει τη δυνατότητα αναζήτησης εργασιών με βάση οποιουδήποτε συνδυασμούς των παρακάτω κριτηρίων: τίτλος, κατηγορία, και προτεραιότητα.

## A.2. Αποθήκευση και ανάκτηση πληροφοριών εφαρμογής (10%)

Για την αποθήκευση και ανάκτηση των πληροφοριών της εφαρμογής θα χρησιμοποιηθεί μια λύση βασισμένη στη χρήση αρχείων που θα περιέχουν δεδομένα σε μορφή JSON.

Το JSON (JavaScript Object Notation) είναι μια μορφή αναπαράστασης δεδομένων που χρησιμοποιείται ευρέως για την αποθήκευση και τη μεταφορά δεδομένων. Είναι εύκολο στην ανάγνωση από ανθρώπους και, ταυτόχρονα, κατανοητό από αρκετές γλώσσες προγραμματισμού. Το JSON ακολουθεί μια απλή δομή χρησιμοποιώντας κείμενο για να αναπαραστήσει δεδομένα με μορφή κλειδιών-τιμών. Περισσότερες πληροφορίες μπορείτε να βρείτε στις διαφάνειες του εργαστηρίου στην ενότητα “Java I/O”.

Αρχικά θα πρέπει να αποφασίσετε και να ορίσετε την δική σας οργάνωση (data schema) για τα JSON δεδομένα καθώς και το σύνολο των αρχείων που θα χρησιμοποιηθούν για την αποθήκευση των δεδομένων της εφαρμογής. Τα αρχεία με τα δεδομένα της εφαρμογής θα πρέπει να αποθηκεύονται σε ένα φάκελο με όνομα **“medialab”**.

Στην συνέχεια πρέπει να υλοποιήσετε μέσω των κατάλληλων κλάσεων τις μεθόδους που θα σας επιτρέπουν να γίνεται ανάκτηση των πληροφοριών που έχουν τα αρχεία και αρχικοποίηση των κατάλληλων αντικειμένων στην εφαρμογή σας καθώς και ανανέωση των αρχείων ώστε να μπορεί να διατηρείται η συνολική κατάσταση μεταξύ διαδιαχικών εκτελέσεων της εφαρμογής.

Στη συνέχεια περιγράφουμε την λογική που πρέπει να υλοποιηθεί για την ανάκτηση και ανανέωση των δεδομένων της εφαρμογής.

- Αρχικοποίηση εφαρμογής: Θα πρέπει να γίνεται η ανάκτηση του συνόλου των πληροφοριών που βρίσκονται στα JSON αρχεία και ταυτόχρονα η αρχικοποίηση των αντίστοιχων αντικειμένων στην εφαρμογή σας.
- Εκτέλεση εφαρμογής: Η εφαρμογή θα χρησιμοποιεί τις πληροφορίες κατάστασης που έχουν ανακτηθεί στην μνήμη του προγράμματος κατά την αρχικοποίηση. Όλες οι λειτουργίες που σχετίζονται με τις εργασίες και τις υπενθυμίσεις που διαχειρίζεται η εφαρμογή θα εκτελούνται με βάση τις πληροφορίες που υπάρχουν στην μνήμη του προγράμματος.
- Τερματισμός εφαρμογής: Η ανανέωση των JSON αρχείων με τις πληροφορίες κατάστασης του συστήματος θα γίνεται αποκλειστικά πριν τον τερματισμό της εφαρμογής. Η υλοποίηση θα πρέπει να αποθηκεύει στα αντίστοιχα JSON αρχεία την συνολική κατάσταση της εφαρμογής κατά την στιγμή του τερματισμού.

### A.3. Δημιουργία γραφικής διεπαφής (30%)

Θα πρέπει να σχεδιάσετε και να υλοποιήσετε την κατάλληλη γραφική διεπαφή χρήστη (Graphical User Interface - GUI) χρησιμοποιώντας το πλαίσιο JavaFX [1][2].

**Σημείωση:** Στη συνέχεια παρουσιάζονται οι βασικές προδιαγραφές για την γραφική διεπαφή, για όλες τις λεπτομέρειες της τελικής υλοποίησης μπορείτε να κάνετε όποιες επιλογές θέλετε σχετικά με την εμφάνιση και τη γενικότερη αλληλεπίδραση του χρήστη με την εφαρμογή, χωρίς καμία επίπτωση στην τελική βαθμολογία. Για παράδειγμα, μπορείτε να επιλέξετε μια απλή απεικόνιση για τα διάφορα στοιχεία ή να συνδυάσετε διάφορα χαρακτηριστικά από το JavaFX ώστε να δημιουργήσετε ένα αποτέλεσμα που να αντιστοιχεί σε μια σύγχρονη εφαρμογή. Σε κάθε περίπτωση, δεν υπάρχει λόγος να κάνετε πολύπλοκο το συγκεκριμένο μέρος της εργασίας.

Αρχικά κατά την εκκίνηση της εφαρμογής αν υπάρχουν εργασίες που είναι σε κατάσταση “Delayed” θα πρέπει να ενημερώνεται ο χρήστης με ένα κατάλληλο popup παράθυρο σχετικά με το πλήθος των εκπρόθεσμων εργασιών.

Σχετικά με την δημιουργία της γραφικής διεπαφής θα πρέπει να ακολουθήσετε τις παρακάτω γενικές οδηγίες:

- Δημιουργήστε το κεντρικό «παράθυρο» της εφαρμογής με τίτλο “MediaLab Assistant” και ορίστε τις κατάλληλες διαστάσεις.
- Χωρίστε το παράθυρο σε δύο βασικά μέρη.
- Στο πάνω μέρος της οθόνης θα εμφανίζονται συγκεντρωτικές πληροφορίες που θα πρέπει να ανανεώνονται αντίστοιχα με βάση τις ενέργειες του χρήστη. Οι πληροφορίες περιλαμβάνουν: (i) συνολικό αριθμό εργασιών ανεξαρτήτως κατάστασης, (ii) αριθμό εργασιών με κατάσταση “Completed”, (iii) αριθμό εργασιών με κατάσταση “Delayed”, και (iv) αριθμό εργασιών με προθεσμία ολοκλήρωσης εντός 7 ημερών.
- Στο άλλο μέρος της οθόνης θα πρέπει να υλοποιηθούν οι διάφορες λειτουργίες που πρέπει να υποστηρίζει η γραφική διεπαφή. Υπάρχει πλήρης ελευθερία για τον τρόπο υλοποίησης των συγκεκριμένων λειτουργιών, ωστόσο, θα πρέπει να φροντίσετε να γίνεται κατάλληλη ενημέρωση στις πληροφορίες της γραφικής διεπαφής ανάλογα με τις ενέργειες του χρήστη.

Η γραφική διεπαφή θα πρέπει να υποστηρίζει τις παρακάτω λειτουργίες:

- Διαχείριση εργασιών: η εφαρμογή πρέπει να παρουσιάζει τις διαθέσιμες εργασίες ανά κατηγορία. Επίσης θα πρέπει να υπάρχει η δυνατότητα ο χρήστης να ορίζει νέα εργασία, να τροποποιεί μια εργασία και να διαγράφει κάποια εργασία. Η υλοποίηση πρέπει να γίνει σύμφωνα με την αντίστοιχη λογική από την ενότητα A.1.

- Διαχείριση κατηγοριών: η εφαρμογή πρέπει να παρουσιάζει την λίστα με τις κατηγορίες. Ακόμη θα πρέπει ο χρήστης να μπορεί να ορίζει νέες κατηγορίες, να αλλάζει το όνομα μιας κατηγορίας και να διαγράφει κατηγορίες. Η υλοποίηση πρέπει να γίνει σύμφωνα με την αντίστοιχη λογική από την ενότητα A.1.
- Διαχείριση επιπέδων προτεραιότητας: η εφαρμογή πρέπει να παρουσιάζει τα διαθέσιμα επίπεδα προτεραιότητας. Ο χρήστης θα πρέπει να έχει την δυνατότητα να τροποποιεί το όνομα των επιπέδων προτεραιότητας και να διαγράφει επίπεδα προτεραιότητας. Η υλοποίηση πρέπει να γίνει σύμφωνα με την αντίστοιχη λογική από την ενότητα A.1.
- Διαχείριση υπενθυμίσεων: η εφαρμογή πρέπει να παρουσιάζει όλες τις ενεργές υπενθυμίσεις. Ο χρήστης θα πρέπει να έχει την δυνατότητα να τροποποιεί μια υπενθύμιση καθώς και να διαγράφει υπενθυμίσεις. Η υλοποίηση πρέπει να γίνει σύμφωνα με την αντίστοιχη λογική από την ενότητα A.1.
- Αναζήτηση εργασιών: θα πρέπει να υπάρχει η αντίστοιχη φόρμα που θα επιτρέπει την αναζήτηση εργασιών με βάση τις προδιαγραφές από την ενότητα A.1. Τα αποτελέσματα θα πρέπει να περιλαμβάνουν: τίτλο, επίπεδο προτεραιότητας, κατηγορία και προθεσμία ολοκλήρωσης.

Τέλος, θα πρέπει η υλοποίηση της γραφικής σας διεπαφής κατά τον τερματισμό της εφαρμογής να εξασφαλίζει την ανανέωση των πληροφοριών του συστήματος στα σχετικά αρχεία, σύμφωνα με την διαδικασία που περιγράφεται στην ενότητα A.2.

#### **A.4. Λοιπές απαιτήσεις (20%)**

- Η υλοποίηση θα πρέπει να ακολουθεί τις αρχές σχεδίασης του αντικειμενοστραφούς προγραμματισμού (OOP design principles).
- Σε μια κλάση της επιλογής σας θα πρέπει κάθε public μέθοδος που περιέχει να είναι τεκμηριωμένη σύμφωνα με τις προδιαγραφές του εργαλείου javadoc [3].

**Σημείωση:** Για οτιδήποτε δεν είναι ξεκάθαρο από την εκφώνηση μπορείτε να κάνετε τις δικές σας παραδοχές και υποθέσεις. Η εκφώνηση περιγράφει τις βασικές απαιτήσεις για την εφαρμογή, ωστόσο μπορείτε να κάνετε δικές σας σχεδιαστικές παραδοχές προσπαθώντας να κάνετε πιο ρεαλιστική την εφαρμογή, χωρίς παράλληλα να κάνετε πολύπλοκη την υλοποίηση.

## Παραδοτέα

- Το project (του IDE της επιλογής σας) με τον κώδικα υλοποίησης της εφαρμογής.
- Μια σύντομη (το πολύ 3 σελίδες) αναφορά που θα περιέχει μια γενική περιγραφή του σχεδιασμού της υλοποίησης και την περιγραφή του σχεδιασμού σας σχετικά με την δομή της πληροφορίας στα διαφορετικά JSON αρχεία. Επίσης, θα αναφέρετε όποια λειτουργικότητα δεν έχετε υλοποιήσει καθώς και όποιες παραδοχές έχετε κάνει. Σε καμία περίπτωση δεν θα πρέπει να συμπεριλάβετε τμήματα κώδικα.

## Αναφορές

- [1] <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>
- [2] [https://docs.oracle.com/javafx/2/get\\_started/jfxpub-get\\_started.htm](https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm)
- [3] <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>