



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εξάμηνο 7ο

Τελική εργασία στο μάθημα Τεχνολογία Πολυμέσων

Δήμητρα Σεφεριάδη

ΑΜ: 03120131

➤ *Γενική Περιγραφή Σχεδιασμού και Υλοποίησης*

Η εφαρμογή υλοποιεί ένα σύστημα διαχείρισης εργασιών (Task Management System) χρησιμοποιώντας JavaFX για τη γραφική διεπαφή και ένα ενιαίο αρχείο JSON για την αποθήκευση των δεδομένων. Στόχος είναι η δημιουργία, επεξεργασία και διαγραφή εργασιών (tasks), καθώς και η διαχείριση κατηγοριών, επιπέδων προτεραιότητας και υπενθυμίσεων.

Η λογική της εφαρμογής διαχωρίζεται σε πακέτα, ώστε να διατηρείται καθαρή η αρχιτεκτονική και ο κώδικας να είναι ευκολότερος στη συντήρηση. Ακολουθούν τα πακέτα και οι κλάσεις που έχουν υλοποιηθεί:

1. *application*

Περιέχει το αρχείο εκκίνησης της εφαρμογής

- *Main.java*

Το σημείο εκκίνησης της εφαρμογής (main method). Φορτώνει την αρχική σκηνή (Stage) και ξεκινά το JavaFX application thread.

2. *data*

Διαχειρίζεται την ανάγνωση και εγγραφή δεδομένων στο αρχείο JSON.

Ουσιαστικά εδώ υλοποιείται η πρόσβαση στα δεδομένα (Data Access Layer).

- *DataManager.java*

Κύρια κλάση για φόρτωση/αποθήκευση των δεδομένων από και προς το medialab_data.json. Περιλαμβάνει μεθόδους που διαβάζουν και αναλύουν (parse) το JSON, δημιουργούν τα αντίστοιχα αντικείμενα (Tasks, Reminders κ.λπ.), καθώς και μεθόδους για να μετατρέπουν τα αντικείμενα σε JSON μορφή πριν την αποθήκευση.

- *LocalDateAdapter.java*

Βοηθητική κλάση για την προσαρμογή των ημερομηνιών σε JSON.

- *ObservableListDeserializer.java*

Βοηθητική κλάση για την προσαρμογή των υπενθυμίσεων σε JSON.

3. *logic*

Υλοποιεί τη λογική της εφαρμογής, δηλαδή ο χειρισμός των αντικειμένων (Tasks, Categories, Priorities, Reminders) με βάση τους κανόνες που ορίζονται από την εκφώνηση.

- *TaskManager.java*

Η κεντρική κλάση που διαχειρίζεται τη συλλογή των εργασιών, των κατηγοριών και των προτεραιοτήτων. Παρέχει μεθόδους για προσθήκη, διαγραφή, ενημέρωση, αναζήτηση, καθώς και μεθόδους που ελέγχουν τις καταστάσεις των εργασιών (π.χ. αλλαγή σε "Delayed" όταν ξεπεραστεί η προθεσμία).

4. *model*

Περιλαμβάνει τις κλάσεις-οντότητες (entities) που αντιπροσωπεύουν τα δεδομένα της εφαρμογής.

- *Task.java*

Αναπαριστά μια εργασία, με τα πεδία id, title, description, category, priority, deadline, status και reminders. Περιέχει getter/setter μεθόδους και την απαραίτητη λογική για το κάθε αντικείμενο.

- *Reminder.java*

Αναπαριστά μία υπενθύμιση με τα πεδία type και reminderDate. Περιέχει getter/setter μεθόδους και την απαραίτητη λογική για το κάθε αντικείμενο.

5. *gui*

Το πακέτο που περιέχει τους JavaFX Controllers και διαχειρίζεται την αλληλεπίδραση χρήστη (User Interface - UI). Συνδέει τα στοιχεία της διεπαφής (Views) με τις κλάσεις λογικής (logic/model).

- *MainController.java*

Ελέγχει το κύριο παράθυρο της εφαρμογής, φορτώνει τις λίστες εργασιών, κατηγοριών και προτεραιοτήτων και παρέχει λειτουργίες για την προσθήκη, διαγραφή και επεξεργασία. Υλοποιεί επίσης τη λογική για την αναζήτηση και την εμφάνιση των στατιστικών (π.χ. Completed Tasks, Delayed Tasks).

- *CategoriesController.java*

Διαχειρίζεται το παράθυρο που αφορά τη δημιουργία, επεξεργασία και διαγραφή των κατηγοριών.

- *PrioritiesController.java*

Διαχειρίζεται το παράθυρο που αφορά τη δημιουργία, επεξεργασία και διαγραφή των προτεραιοτήτων.

- *ReminderController.java*

Επιτρέπει τη δημιουργία, επεξεργασία ή διαγραφή των υπενθυμίσεων που σχετίζονται με μια εργασία.

- *TaskController.java*

Διαχειρίζεται τη φόρμα προσθήκης ή επεξεργασίας μιας συγκεκριμένης εργασίας (Task). Εδώ περιλαμβάνονται χειριστές (event handlers) για αποθήκευση, ακύρωση, διαγραφή κ.ά.

6. *view*

Περιέχει τα αρχεία FXML που ορίζουν τη διάταξη (layout) των φορμών/παραθύρων της εφαρμογής.

- *AddTaskView.fxml*
- *CategoriesView.fxml*
- *MainView.fxml*
- *PrioritiesView.fxml*
- *Reminder.fxml*
- *TaskView.fxml*

Κάθε FXML αρχείο περιγράφει τα γραφικά στοιχεία που αντιστοιχούν στον αντίστοιχο Controller.

➤ Δομή Δεδομένων στα JSON Αρχεία

Όλα τα δεδομένα αποθηκεύονται σε ένα ενιαίο αρχείο JSON με την παρακάτω δομή:

- *priorities*: Λίστα με τα επίπεδα προτεραιότητας.
- *categories*: Λίστα με τις κατηγορίες εργασιών.
- *tasks*: Λίστα με τις εργασίες.

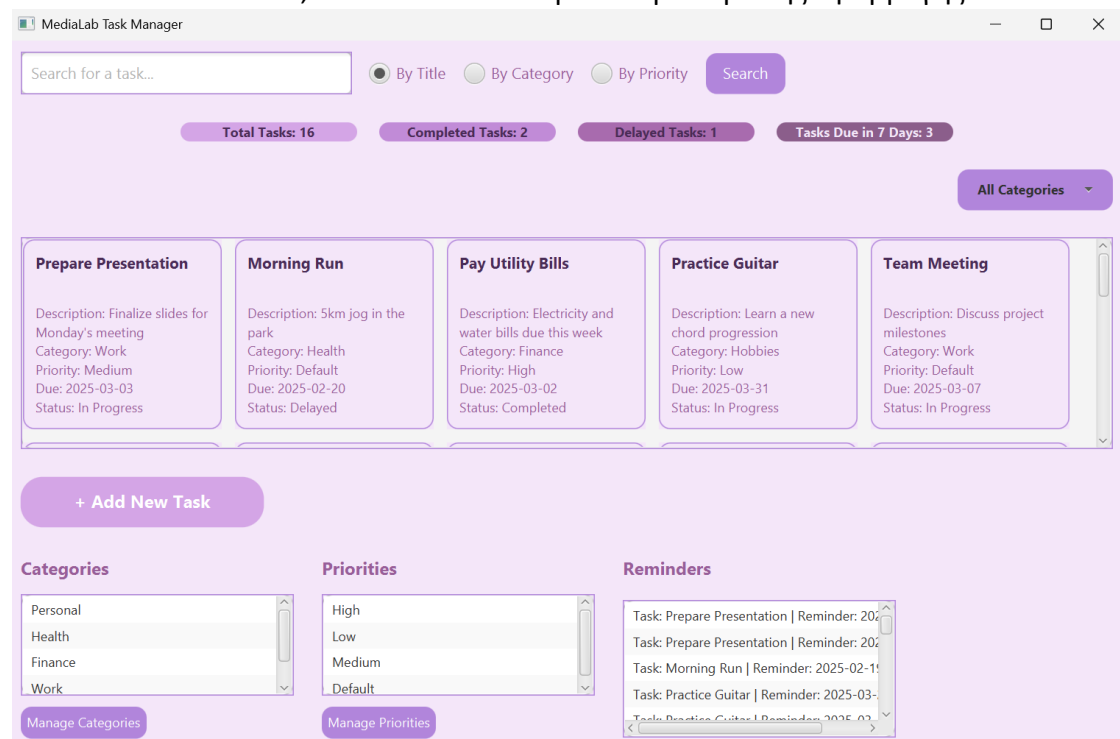
Κατά την εκκίνηση της εφαρμογής, η κλάση DataManager διαβάζει το αρχείο *medialab_data.json* και φορτώνει τα δεδομένα σε αντίστοιχες λίστες. Κατά τη λειτουργία της εφαρμογής, οι αλλαγές γίνονται στη μνήμη, δηλαδή άμεσα στον TaskManager και κατά τον τερματισμό καλείται η μέθοδος *stor()*, που αποθηκεύει τα περιεχόμενα του TaskManager ξανά στο JSON αρχείο, αντικαθιστώντας τα παλιά δεδομένα.

➤ Λειτουργικότητες – Παραδοχές

Έχουν υλοποιηθεί όλες οι λειτουργικότητες ακριβώς όπως αναγράφονται στην εκφώνηση. Οι παραδοχές που έχουμε κάνει είναι οι εξής:

- Για να προσθέσουμε ένα Task πρέπει να περιέχει Title, Description, Category και DueDate. Για κάθε νέο Task η προκαθορισμένη κατάσταση του επιπέδου προτεραιότητας είναι Default και του Status είναι Open, αλλά φυσικά μπορούν να τροποποιηθούν.
- Για να προσθέσουμε Reminders σε ένα Task πρέπει να το έχουμε δημιουργήσει πρώτα.
- Όσον αφορά τα Reminders δεν μπορούμε να ορίσουμε υπενθύμιση μετά το DueDate του Task, ούτε για ημερομηνίες πριν την σημερινή. Ο χρήστης σε αυτή την περίπτωση ειδοποιείται με κατάλληλο μήνυμα.
- Ικανοποιούνται, σε όλα τα στάδια της εφαρμογής, έλεγχοι για να μην μπορείς να προσθέσεις κάποιο αντικείμενο με κενό κάποιο πεδίο του ή με ίδιο όνομα με αντικείμενο που υπάρχει ήδη στην εφαρμογή.

Ενδεικτικά, ακολουθεί το κεντρικό παράθυρο της εφαρμογής:



Και κάποια άλλα στιγμιότυπα:

