



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εξάμηνο 7ο

Τελική εργασία στο μάθημα Τεχνολογία Πολυμέσων

Ονοματεπώνυμο: Αποστολία Χρυσοβαλάντου Σκέντζου

Αριθμός Μητρώου: 03120054

1. Εισαγωγή

Η εφαρμογή Task Manager έχει σχεδιαστεί για τη διαχείριση εργασιών, κατηγοριών, προτεραιοτήτων και υπενθυμίσεων. Η υλοποίηση γίνεται σε Java, αξιοποιώντας το JavaFX για τη δημιουργία του γραφικού περιβάλλοντος και το Jackson για τη σειριοποίηση/αποσειριοποίηση δεδομένων σε JSON. Επιπλέον, η χρήση του Maven για τη διαχείριση εξαρτήσεων και την αυτοματοποίηση της διαδικασίας κατασκευής εξασφαλίζει ένα ευέλικτο και επεκτάσιμο περιβάλλον ανάπτυξης.

2. Γενική Περιγραφή Σχεδίασης

Για την υλοποίηση της εφαρμογής, δημιουργήσαμε ξεχωριστές κλάσεις για τις βασικές οντότητες καθώς και τους αντίστοιχους managers για την διαχείρισή τους. Συνοπτικά, παρουσιάζονται παρακάτω οι κύριες κλάσεις και οι λειτουργίες τους:

Κλάση	Λειτουργία
Category	Αντιπροσωπεύει μια κατηγορία εργασιών.
CategoryManager	Διαχειρίζεται τη συλλογή κατηγοριών. Υποστηρίζει προσθήκη, ενημέρωση και διαγραφή κατηγοριών, καθώς και συγχρονισμό με τις εργασίες που ανήκουν σε αυτές.
Priority	Αντιπροσωπεύει μια προτεραιότητα για μια εργασία.
PriorityManager	Διαχειρίζεται τις προτεραιότητες, επιτρέποντας την προσθήκη, ενημέρωση και διαγραφή τους. Ενημερώνει τις εργασίες ώστε να χρησιμοποιούν την προεπιλεγμένη προτεραιότητα όταν απαιτείται.
Reminder	Αντιπροσωπεύει μια υπενθύμιση για μια εργασία, με πεδία όπως η ημερομηνία υπενθύμισης και ο τύπος υπενθύμισης.
ReminderManager	Διαχειρίζεται τις υπενθυμίσεις. Φορτώνει και αποθηκεύει τις υπενθυμίσεις σε JSON αρχεία, και παρέχει λειτουργίες προσθήκης, ενημέρωσης και διαγραφής υπενθυμίσεων.

Reminder Type	Ορίζει τους διαθέσιμους τύπους υπενθυμίσεων
Task	Αντιπροσωπεύει μια εργασία με πεδία όπως ταυτότητα, τίτλο, περιγραφή, κατηγορία, προτεραιότητα, ημερομηνία λήξης και κατάσταση.
TaskManager	Διαχειρίζεται τις εργασίες. Υποστηρίζει λειτουργίες δημιουργίας, ανάγνωσης, ενημέρωσης και διαγραφής εργασιών, ενημερώνει την κατάσταση των εργασιών και συντονίζει τις υπενθυμίσεις.
TaskStatus	Ορίζει τις πιθανές καταστάσεις μιας εργασίας
JsonUtil	Βοηθητική κλάση για ανάγνωση και εγγραφή δεδομένων σε JSON αρχεία, αξιοποιώντας το Jackson
AppController	Περιέχει τη λογική ελέγχου της εφαρμογής (controller). Αναλαμβάνει τη διαχείριση των δεδομένων μέσω των managers, την ενημέρωση του γραφικού περιβάλλοντος (UI) και την διαχείριση των διαλόγων και των αλληλεπιδράσεων με τον χρήστη.
MainApp	Η κύρια κλάση του γραφικού περιβάλλοντος που δημιουργεί τη διάταξη του UI και αναθέτει τα συμβάντα στον AppController. Φορτώνει τα δεδομένα μέσω των managers και συντονίζει την εκκίνηση και το κλείσιμο της εφαρμογής.

3. Δομή Πληροφοριών στα JSON Αρχεία

Για κάθε βασική οντότητα της εφαρμογής δημιουργήσαμε ξεχωριστό JSON αρχείο:

tasks.json

Περιέχει λίστα αντικειμένων Task με πεδία:

- ✓ **id**: Μοναδικός αναγνωριστικός αριθμός (UUID).
- ✓ **title**: Ο τίτλος της εργασίας.
- ✓ **description**: Η περιγραφή της εργασίας.
- ✓ **category**: Αντικείμενο Category (με τουλάχιστον το όνομα της κατηγορίας).
- ✓ **priority**: Αντικείμενο Priority (π.χ. "Default", "High").
- ✓ **dueDate**: Η ημερομηνία λήξης της εργασίας (σε ISO μορφή).
- ✓ **status**: Η κατάσταση της εργασίας (π.χ. OPEN, COMPLETED, DELAYED).

reminders.json

Περιέχει τις υπενθυμίσεις που σχετίζονται με εργασίες. Κάθε αντικείμενο περιλαμβάνει:

- ✓ **task**: Αναφορά στην εργασία για την οποία έχει οριστεί η υπενθύμιση (μέσω id ή ενσωματωμένου αντικειμένου).

- ✓ **reminderDate:** Η ημερομηνία υπενθύμισης, η οποία ορίζεται ή υπολογίζεται βάσει του τύπου υπενθύμισης.
- ✓ **reminderType:** Ο τύπος της υπενθύμισης (π.χ. ONE_DAY_BEFORE, ONE_WEEK_BEFORE, ONE_MONTH_BEFORE, SPECIFIC_DATE).

categories.json

Περιέχει μια λίστα από κατηγορίες, όπου κάθε αντικείμενο είναι απλά:

- ✓ **name:** Το όνομα της κατηγορίας.

priorities.json

Περιέχει τις προτεραιότητες ως λίστα αντικειμένων, όπου κάθε αντικείμενο έχει:

- ✓ **name:** Το όνομα της προτεραιότητας (π.χ. "Default", "High", "Low").

4. Λειτουργίες που δεν υλοποιήθηκαν

Δεν υπάρχει κάποια λειτουργία που να μην υλοποιήθηκε στην εφαρμογή

5. Παραδοχές

Για τις ειδοποιήσεις θεωρήσαμε σκόπιμο να εμφανίζονται με popup window την ημερομηνία που έχουν οριστεί από τον χρήστη και ύστερα να διαγράφονται.

Επίσης φροντίσαμε η εφαρμογή μας να διαγράφει παλιότερα reminders τα οποία δεν εμφανίστηκαν στον χρήστη, εάν δεν άνοιξε την εφαρμογή την ημέρα της ειδοποίησης.

Θεωρήσαμε σκόπιμο, αφού δεν ζητήθηκε ρητά από την εκφώνηση να μην προσθέσουμε στο γραφικό περιβάλλον κουμπί προσθήκης προτεραιότητας, αλλά μια νέα προτεραιότητα να δημιουργείται μονάχα αν αρχικοποιηθεί από κάποια νέα εργασία.

Εκτέλεση προγράμματος

Μετά την αντιγραφή του φακέλου του project, είναι σκόπιμος ο έλεγχος πως έχει εγκατασταθεί ήδη το Maven και πως γίνεται χρήση της κατάλληλης έκδοσης της Java και της JavaFX. Στη συνέχεια, με ανοιχτό ένα τερματικό στον φάκελο του project είναι απαραίτητη η εκτέλεση της εντολής **mvn clean install** για την εγκατάσταση των εξαρτήσεων και την κατασκευή (build) της εφαρμογής. Μόλις ολοκληρωθεί η κατασκευή, απαιτείται η εκτέλεση της εντολής **mvn javafx:run** για την εκκίνηση της εφαρμογής μέσω του JavaFX Maven Plugin.