Τεχνολογία Πολυμέσων

Φοιτητής: Πυλιώτης Αθανάσιος

ΑΜ: 03119050

Περιεχόμενα

Διαχείριση αρχείων JSON

Συνολικά έχουμε 3 αρχεία JSON στον φάκελο medialab της μορφής array με objects. Τα αρχεία αυτά αντιστοιχούν σε 4 διαφορετικά αντικείμενα της εφαρμογής: ένα αρχείο για τα tasks και τα notifications, ένα αρχείο για τα categories και ένα αρχείο για τα priorities. Κάθε JSON εσωτερικά, αν το δούμε πρακτικά, αντιστοιχεί σε μια λίστα με dictionaries, τα οποία έχουν σαν κλειδιά τα στοιχεία του αντικειμένου που αποθηκεύουμε. Για παράδειγμα, για τα categories αποθηκεύουμε το categoryId και το name, για τα priorities το priorityId και το level και για το task όλα τα πεδία του και μια λίστα από notifications. Στο πρόγραμμα μας τα notifications εξαρτώνται από τα tasks και, πρακτικά, κάθε notification εξαρτάται από ένα task. Συνεπώς, στην υλοποίηση μου τα notifications υπάρχουν ως λίστα μέσα στο Task για το οποίο έχουν οριστεί. Για αυτό το λόγο και κατά τη διάρκεια της αποθήκευσης στα JSON αρχεία είναι σαν μια ακόμα λίστα από JSON αντικείμενα.

Συνοπτικά, υπάρχουν μέσα στο φάκελο medialab 3 αρχεία, το tasks.JSON, categories.JSON και priorities.JSON. Παράδειγμα αρχείου:

[{"priorityId":-1,"name":"Default"},{"priorityId":2,"name":"too high"},{"priorityId":3,"name":"low"},{"priorityId":4,"name":"medium"}]

Τέλος, από επιλογή μου, το σύστημα διατηρεί τα IDs και τα χρησιμοποιεί κατά τη διαδικασία αναδημιουργίας των αντικειμένων. Αυτό γίνεται ώστε το σύστημα να ξεκινήσει στην ίδια ακριβώς κατάσταση στην οποία το άφησε ο χρήστης. Με αυτό το τρόπο δεν θα υπάρξει και πρόβλημα με την αλλαγή ID κάθε φορά που ο χρήστης αρχίζει την εφαρμογή.

Διαχείριση αρχείων κώδικα και λειτουργίες

Τα αρχεία κώδικα χωρίζονται σε 5 διαφορετικά κομμάτια. Πρώτον, στην αρχή υπάρχει το pom.xml το οποίο χρησιμοποιείται από το maven για να διαχειρίζεται τα dependencies, το build και τα διάφορα plugins που χρειάζονται προκειμένου να λειτουργήσει η εφαρμογή. Μετά υπάρχει το medialab, το οποίο περιέχει τα JSON αρχεία και στο οποίο έγινε αναφορά παραπάνω, και το JavaDoc, το οποίο περιέχει όλες τις προδιαγραφές για κάθε κλάση που υπάρχει στο σύστημα σε ικανοποιητική ανάλυση. Τέλος, υπάρχει ο φάκελος java με τον κώδικα και ο φάκελος resources με τα FXML και τα CSS αρχεία.

Αρχικά, στα resources υπάρχουν τα views με τα FXMLs, τα οποία περιέχουν τον κώδικα για την απεικόνιση των παραθύρων της εφαρμογής. Τα FXMLs είναι Java FX HTML αρχεία με κάπως διαφορετικά elements από την HTML. Χρησιμοποιούνται κυρίως για απεικόνιση λιστών και πινάκων και για την αλληλεπίδραση μεταξύ χρήστη και εφαρμογής. Τα αρχεία CSS περιέχουν μερικά στοιχεία για την βελτίωση της εμφάνισης της εφαρμογής. Περισσότερα για αυτά παρακάτω στην αναφορά.

Σχετικά με το Java κώδικα, έχουμε το org.taskmanager.taskmanager πακέτο, το οποίο περιέχει τη κλάση mediaLabAssistant που εκκινεί την εφαρμογή και άλλα 4 πακέτα:

- **Controller:** Αυτό το πακέτο περιέχει όλα τα controllers για τα παράθυρα που μπορεί να ανοίξει ο χρήστης στην εφαρμογή. Ο τρόπος με τον οποίο λειτουργεί το JavaFX επιβάλλει κάθε παράθυρο να έχει ένα controller, ο οποίος πρέπει να είναι καινούριο instance controller. Λόγω των αρχών του αντικειμενοστρεφούς προγραμματισμού, επιλέξαμε κάθε controller να επιτελεί συγκεκριμένη λειτουργία και να αντιστοιχεί σε συγκεκριμένο παράθυρο, με πολύ στοχευμένες λειτουργίες. Κάθε διαφορετικό παράθυρο που ανοίγει αντιστοιχεί και σε διαφορετικό controller, σε σύνολο 8.
- **Model:** Αποτελείται από κλάσεις αντικειμένων που αναπαριστούν τα αντικείμενα που θα χρειαστούμε μέσα στην εφαρμογή, συγκεκριμένα τα Task, Priority, Category, Notification και τη βοηθητική κλάση NotificationWrapper. Χρησιμοποιούνται για να διατηρούμε, φορτώνουμε και αποθηκεύουμε μαζικά όλες τις πληροφορίες για ένα από τα προαναφερθέντα αντικείμενα. Επίσης στο Task υπάρχουν βοηθητικές μέθοδοι που επιστρέφουν πληροφορίες για το αντικείμενο.
- **Repository:** Αυτό το πακέτο αποτελείται από 3 repositories, ένα για tasks και notifications, ένα για categories και ένα για priorities. Τα repositories χρησιμοποιούνται για την διαχείριση των λιστών των αντικειμένων κατά τη διάρκεια του παιχνιδιού, περιέχουν λειτουργίες για φόρτωση λιστών που καλούν τα utils κατάλληλα. Η πλειοψηφία των λειτουργιών που εκτελούνται στις λίστες αυτές, αν όχι όλες, γίνονται μέσα στα repositories, καθώς και αναζητήσεις στις λίστες και εξαγωγή χρήσιμων πληροφοριών.
- **Utils:** Τα utils περιέχουν 3 κλάσεις που αλληλοεπιδρούν με τα JSON αρχεία, με 2 μεθόδους η καθεμία, μία για να διαβάζει από το JSON αρχείο και μία για να γράφει στο JSON αρχείο. Πέρα από αυτές, υπάρχει και ένα enumeration util για τη κατάσταση του task. Υπήρχε ένας προβληματισμός για το πακέτο στο οποίο θα έπρεπε να ανήκει το enumeration και επιλέχθηκε το utils καθώς είναι για γενική χρήση.

Γραφική επαφή με χρήστη

Οι επιλογές στη γραφική επαφή ήταν ώστε να γίνει πιο εμφανίσιμη στον χρήστη και χωρίς να του εμφανίζει πολλές πληροφορίες ταυτόχρονα. Στην αρχική σελίδα εμφανίζει μονάχα πληροφορίες για τα tasks στο σύστημα και τα notifications, γενικά στατιστικά για τα tasks, καθώς και κουμπιά για μετακίνηση σε άλλα παράθυρα. Τα παράθυρα για categories, priorities και tasks είναι ξεχωριστά ώστε να καταλαβαίνει εύκολα ο χρήστης που υπάρχει η κάθε λειτουργία. Η μόνη λειτουργία που είναι κάπως πιο κρυμμένη είναι η διαχείριση των notifications, η οποία είναι προσβάσιμη από το παράθυρο διαχείρισης των tasks, καθώς κάθε notification ανταποκρίνεται σε μοναδικό task. Με αυτό το τρόπο η διαχείριση τους είναι πιο εύκολη και ο χρήστης γνωρίζει για ποιο task είναι το notification.

Επιλέχθηκε τα notifications να εμφανίζονται όλα μαζί σε ένα παράθυρο ξεχωριστό κατά την εκκίνησης της εφαρμογής πριν διαγράφονται για να μην υπάρχει spam στον χρήστη αλλά να εμφανίζουν και όλες τις απαραίτητες πληροφορίες. Η παρουσίαση των delayed tasks γίνεται επίσης στην εκκίνηση της εφαρμογής και εμφανίζεται σε ξεχωριστό παράθυρο, καθώς μόνο στην εκκίνηση της εφαρμογής μπορεί το state ενός task να γίνει delayed και μετά ο χρήστης μπορεί να το αλλάξει σε όλες τις υπόλοιπες καταστάσεις.

Το μόνο πρόβλημα που δεν επιλύθηκε είναι που τα παράθυρα για τις ειδοποιήσεις και τα delayed tasks εμφανίζονται πίσω από το βασικό παράθυρο λόγω της σειράς που χρειάζεται να τρέξουν οι εντολές.

Παραδοχές

Παραδοχές που έγιναν για την εργασία που δεν αναγράφονταν ρητά στην εκφώνηση:

- Υπάρχει προ-καταχωρημένη κατηγορία “Uncategorized”, που αντιστοιχεί σε tasks χωρίς κατηγορία. Αντίστοιχα με το Default priority δεν μπορεί να διαγραφεί η ανανεωθεί.
- Δεν γίνει να υπάρξουν 2 κατηγορίες με το ίδιο όνομα ή 2 προτεραιότητες με το ίδιο όνομα, καθώς δεν θα ήταν λογικό να υπάρχουν.
- Τα notifications αποθηκεύονται στο task για το οποίο ορίζονται.
- Ο χρήστης μπορεί να θέτει όποια κατάσταση θέλει σε tasks εκτός από delayed.
- Ο χρήστης μπορεί να θέτει deadline μόνο μετά τη σημερινή μέρα.
- Οι υπενθυμίσεις που έχουν οριστεί για ένα task που διαγράφεται, διαγράφονται με αυτό αφού δεν έχουν πλέον λόγω ύπαρξης.
- Μαζί με τη παρουσίαση του πλήθους των εκπρόθεσμων εργασιών εμφανίζονται και οι ίδιες οι εργασίες.
- Υπάρχει ξεχωριστό παράθυρο για τις CRUD μεθόδους κάθε αντικειμένου για καλύτερη εμφάνιση.

Όλες οι λειτουργικότητες πρέπει να είναι ολοκληρωμένες καθώς και να γίνεται κατάλληλη ανανέωση της εφαρμογής.