

Αναφορά Εξαμηνιαίας Εργασίας

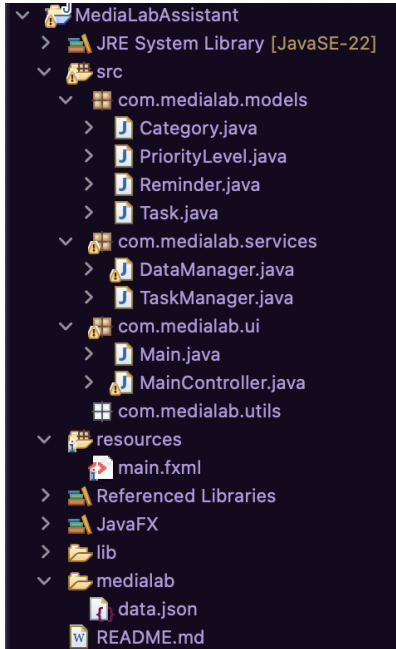
Τεχνολογία Πολυμέσων

ΑΜ: el20034

Όνομα: Φίλιππος Μαρντιροσιάν

Σχεδιασμός Υλοποίησης

Σκελετός του *project*:



Αρχικά, υπάρχουν οι εξής φάκελοι (με κώδικα υλοποίησης): `src`, `resources`, `medialab`.

Ο φάκελος “**src**” περιέχει τα πακέτα `com.medialab.models`, `com.medialab.services`, `com.medialab.ui`.

Ο φάκελος “**resources**” περιέχει το αρχείο `main.fxml`.

Ο φάκελος “**medialab**” περιέχει το μοναδικό `data.json` (περισσότερα για αυτό στη συνέχεια).

Όσον αφορά τα πακέτα:

Το πακέτο `com.medialab.models` περιεχει **κλάσεις που αναπαριστούν τις βασικές οντότητες** της εφαρμογής.

Το πακέτο `com.medialab.services` περιέχει κλάσεις που είναι υπεύθυνες για τη **διαχείριση της λογικής της εφαρμογής** και τη διαχείριση δεδομένων.

Το πακέτο `com.medialab.ui` περιέχει κλάσεις που διαχειρίζονται τη **διεπαφή χρήστη και τη λογική της αλληλεπίδρασης**

Δομή Πληροφορίας στο JSON αρχείο

Έχω επιλέξει να αποθηκεύω την πληροφορία της εφαρμογής (Tasks, Categories, Priorities, Reminders) σε ένα **μοναδικό** αρχείο JSON.

Εξήγηση δομής:

Το αρχείο JSON περιέχει τρία βασικά τμήματα:

1. **tasks** (Λίστα με εργασίες)

- Κάθε εργασία (task) έχει τίτλο (title), κατάσταση (status), προτεραιότητα (priority), και προαιρετικά:
- Κατηγορία (category), αν η εργασία ανήκει σε συγκεκριμένη ομάδα.

- Προθεσμία (deadline), αν έχει συγκεκριμένη ημερομηνία λήξης.
 - Υπενθυμίσεις (reminders), που περιλαμβάνουν ημερομηνία και μήνυμα.
2. **categories** (Λίστα με κατηγορίες εργασιών)
- Περιέχει τις διαθέσιμες κατηγορίες (title), όπως category1, category2.
3. **priorities** (Λίστα με επίπεδα προτεραιότητας)
- Κάθε προτεραιότητα έχει ένα τίτλο (title), όπως Default, priority1, priority2, etc.

Προσοχή: Έχω επιλέξει να αποθηκεύσω ξεχωριστά τις κατηγορίες (**Categories**) και τις προτεραιότητες (**Priorities**) αντί να τις ενσωματώσω μέσα στις εργασίες (**Tasks**). Αυτό συμβαίνει γιατί θεωρώ ότι μια κατηγορία μπορεί να υπάρχει αυτόνομα, χωρίς να είναι απαραίτητα συνδεδεμένη με κάποια συγκεκριμένη εργασία. Το ίδιο ισχύει και για τις προτεραιότητες, καθώς μπορούν να υπάρχουν ανεξάρτητα από τα tasks που τις χρησιμοποιούν.

Ωστόσο, οι υπενθυμίσεις (**Reminders**) αποθηκεύονται μέσα στις εργασίες, καθώς – σύμφωνα με την εκφώνηση – «Μια υπενθύμιση θα σχετίζεται πάντα με μια εργασία». Αυτό σημαίνει ότι κάθε υπενθύμιση είναι άμεσα συνδεδεμένη με ένα task και δεν μπορεί να υπάρξει από μόνη της.

Παράδειγμα δομής:

```
{
  "tasks": [ // The main list that stores all tasks
    {
      "title": "task1", // The title of the task (task1)
      "status": "Open", // The current status of the task (e.g., Open, Completed, Postponed etc.)
      "priority": "priority1", // The priority level assigned to this task (priority1)
      "reminders": [] // The list of reminders for this task. It's empty because no reminders exist.
    },
    {
      "title": "task3",
      "status": "Open",
      "priority": "priority2",
      "deadline": "2025-02-08", // The due date for completing this task (YYYY-MM-DD format)
      "reminders": [ // A list of reminder objects related to this task
        {
          "date": "2025-02-07", // The date when this reminder will trigger (one day before the deadline)
          "message": "reminder1" // The (unique) message or note associated with this reminder
        }
      ]
    }
  ],
  "categories": [ // The list of available categories for tasks
    {
      "title": "category1" // The title of the first category (category1)
    },
    {
      "title": "category2" // The title of the second category (category2)
    }
  ],
  "priorities": [ // The list of available priority levels
    {
      "title": "Default" // The title of the first priority level (Default - always exists)
    },
    {
      "title": "priority2" // The title of the second priority level (priority2)
    }
  ]
}
```

```
]
}
```

Έχουν υλοποιηθεί όλες οι λειτουργικότητες.

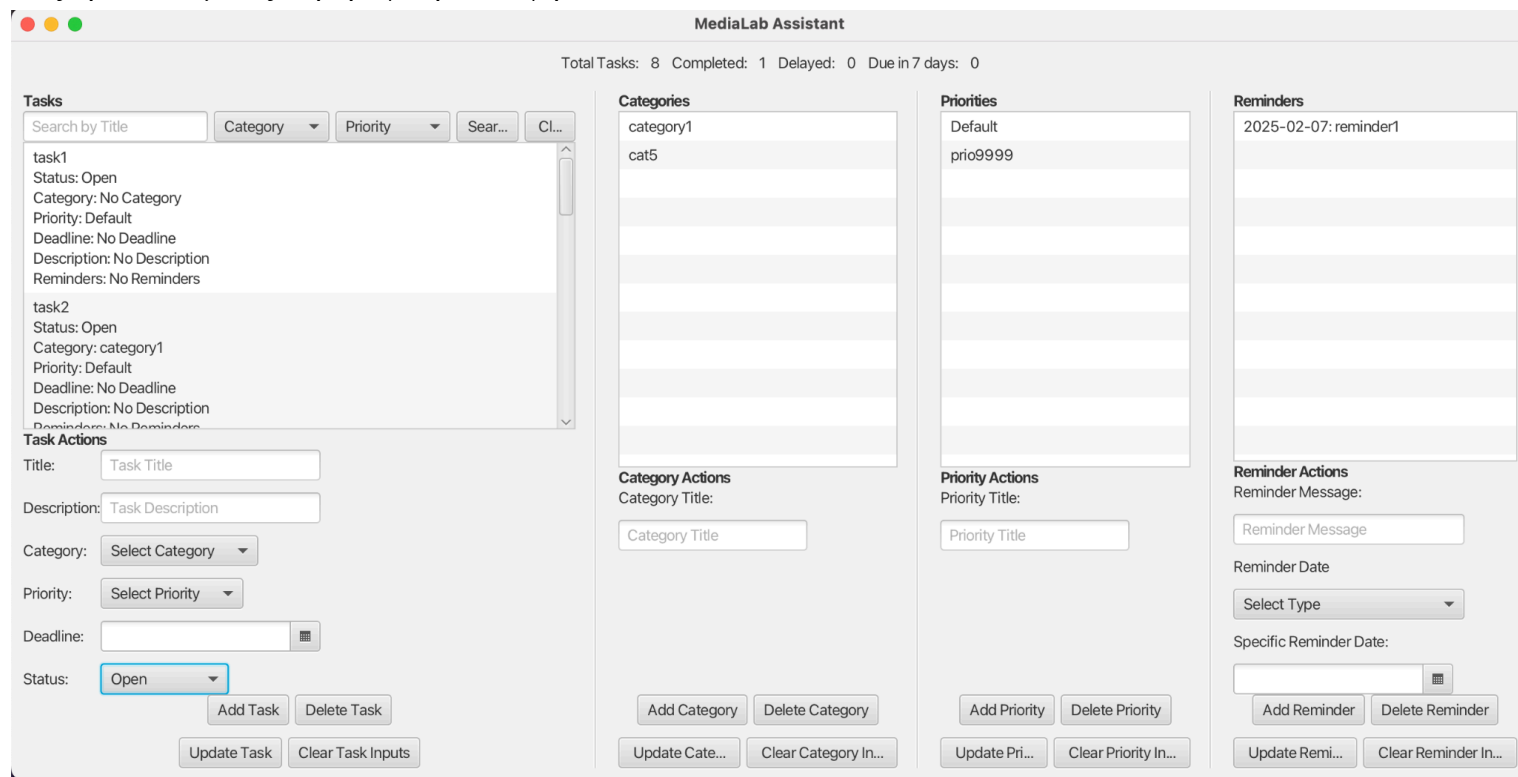
Τεκμηρίωση javadoc

Σημειώνεται ότι η κλάση της επιλογής μου (στην οποία κάθε public μέθοδος που περιέχει είναι τεκμηριωμένη σύμφωνα με τις προδιαγραφές του εργαλείου **javadoc**) είναι η κλάση **DataManager**, η οποία περιέχει τις εξής δύο public μεθόδους: **saveData**, και **loadData**.

Παραδοχές

1. Μια Εργασία μπορεί να ανήκει σε το πολύ μία Κατηγορία.
2. Κάθε Υπενθύμιση αποτελείται από μια ημερομηνία, και από ένα μήνυμα. Για ευκολία διαχείρισης των υπενθυμίσεων, έχω θεωρήσει πως τα **μηνύματα** είναι μοναδικά (**unique**).

Πώς πρέπει να μοιάζει η Γραφική Διεπαφή:



Σύντομη περιγραφή του τρόπου λειτουργίας:

Υπάρχουν 4 στήλες για τα Tasks, Categories, Priorities, Reminders αντίστοιχα.

Στην κάθε μια παρουσιάζεται η λίστα, και από κάτω τα αντίστοιχα Actions.

Για οποιοδήποτε Action, θα πρέπει πρώτα να συμπληρωθούν οι αντίστοιχες φόρμες, κι έπειτα να πατηθεί το κουμπί για το Action.

Λεπτά σημεία για διευκόλυνση στο Testing:

- Για προσθήκη Reminder, θα πρέπει πρώτα να έχει επιλεγθεί το Task, με το οποίο θα συσχετιστεί το Reminder. (Αντιθέτως, οι Κατηγορίες και οι Προτεραιότητες είναι ανεξάρτητες οντότητες και μπορούν να δημιουργηθούν χωρίς επιλογή κάποιου Task).
- Στην αλλαγή ενός Task ("Update Task"), αλλάζουν μόνο τα πεδία που έχουν τροποποιηθεί (δηλαδή αυτά που έχουν μη κενή τιμή όταν πατηθεί το κουμπί), με σκοπό να μην πρέπει ο χρήστης να ξαναγράψει όλα τα πεδία από την αρχή - πράγμα κουραστικό.