

Progetto per la prova finale del corso di Paradigmi di Programmazione a.a. 2022/2023

Dopo la lettura del testo del progetto abbiamo deciso di svilupparlo basandoci su tre differenti classi:

- **Agende**: classe che rappresenta il sistema di agende.
- **Agenda**: classe che rappresenta un'agenda di appuntamenti.
- **Appuntamento**: classe che rappresenta l'appuntamento.

Abbiamo deciso di iniziare con la stesura iniziale della classe Appuntamento in modo tale da riuscire a sviluppare e testare in seguito l'agenda e le agende.

Inizialmente la classe Appuntamenti aveva tutti i dati (data, orario, durata, nome e luogo) nel formato di stringhe ma in seguito vedendo la complessità che stava prendendo il codice abbiamo deciso di farle diventare:

- **LocalDate** con il formato "gg-MM-yyyy" per la data.
- **LocalTime** con il formato "HH-mm" per l'orario.
- **Int** per la durata.

Il costruttore dell'oggetto Appuntamento e i setter prenderanno comunque stringhe in input in modo da vedere se esse rispettino le eccezioni, in caso passassero i controlli verranno poi convertite nel formato giusto in base al tipo di dato. Le funzioni che abbiamo in questa classe sono le seguenti:

- *public void controllo(Agenda agenda)*: esso andrà a verificare se l'appuntamento non è nello stesso giorno e stesso orario di un altro all'interno dell'agenda e in seguito viene richiamato un metodo privato *sovrapponeOrario(Appuntamento app)* che controlla se l'appuntamento si sovrappone a livello di durata, inizio e fine con un altro all'interno dell'agenda, tiene conto anche del caso limite in cui un appuntamento inizi in un giorno e finisca in quello seguente.
- *public int compareTo(Appuntamento other)*: è un Override della funzione *compareTo()* dell'interfaccia Comparable che ci permetterà di comparare i diversi appuntamenti per metterli in ordine.

Passiamo adesso con la stesura della classe Agenda dove anch'essa nei costruttori e nel setter del nome prende delle stringhe che devono rispettare un'eccezione. Le funzioni principali di questa classe sono:

- *public void scriviSuFile(String filepath)*: essa scrive l'agenda sul file specificato dalla variabile filepath.
- *public void inserisciAppuntamento(Appuntamento newApp)*: inserisce un nuovo appuntamento all'interno dell'agenda solo se rispetta i vari controlli.

- *public void aggiornaAppuntamento(Appuntamento app, String data, String orario, String durata, String nome, String luogo):* aggiorna i dati di un appuntamento esistente solo se i nuovi dati rispettano anch'essi i controlli.
- *public void cercaAppuntamenti(String str_cerca):* data una stringa che può rappresentare la data o il nome si andrà a rappresentare gli appuntamenti nell'agenda di essi se esistono.
- *public void ordinaAppuntamenti():* essa va ad ordinare gli appuntamenti dell'agenda.

Inoltre, in questa classe c'è la presenza dell'interfaccia Iterable che permette di iterare sull'agenda i vari appuntamenti e le funzioni di essa sono due: la prima controlla se ci sono ancora degli appuntamenti e la seconda ci ritorna l'appuntamento seguente.

Concludiamo in fine con la classe Agende, il quale compito principale sarà creare agende con tutti nomi unici e in caso eliminarle. Le funzioni di questa saranno:

- *public Agenda creaAgenda(String nome):* va a creare una nuova agenda se rispetta i controlli riguardanti il nome, il quale sia unico.
- *public void eliminaAgenda(Agenda agenda):* elimina un'agenda all'interno del sistema delle agende.
- *public Agenda creaAgendaDaFile(String nomeA, String filepath):* crea una nuova agenda andando a prendere gli appuntamenti dal file descritto dalla variabile filepath.

Date queste tre classi abbiamo testato tutte le funzioni e creato una Console utente in grado di utilizzarle tutte senza mai che si interrompesse l'esecuzione andando a catturare le eccezioni.