

Agenda - Object Oriented Programming

Gabor Galazzo (20024195) A.A. 2018-2019 - DiSIT - UniUPO

Agenda è un applicativo per gestire appuntamenti caratterizzati da **data, ora, durata, dove e quando**.

Scelte progettuali:

Appointment:

Al fine di gestire il corretto formato dei campi `Appointment.date` e `Appointment.time` sono state introdotte due costanti statiche di tipo `DateTimeFormatter`:

- `Appointment.APPOINTMENT_DATE_FORMAT`
- `Appointment.APPOINTMENT_TIME_FORMAT`

Tutti i campi di questa classe sono immutabili al fine di prevenire eventuali errori di incapsulamento che porterebbero ad una scorretta alterazione di un appuntamento causando una situazione di incoerenza difficilmente gestibile da `Agenda`

Il metodo `Appointment.getDateTime()` è stato introdotto per agevolare le operazioni sulle date.

Agenda:

Questa classe permette di gestire e manipolare una lista di appuntamenti. All'interno dell'agenda non ci possono essere due o più appuntamenti che collidono, ovvero già definiti per quello stesso orario, tenendo anche conto della durata.

Per modificare un appuntamento bisogna passare per l'agenda che lo contiene in modo tale da evitare che le modifiche possano creare un agenda inconsistente sui suoi vincoli.

Si considerino i metodi `findByDate(String)` e `findByWith(String)` delle agevolazioni di `findByPredicate(Predicate)`

Il metodo `writeAgenda(Writer)` sfrutta la libreria `Gson` per serializzare `appointments` in formato Json.

Il metodo `readAgenda(Reader)` sfrutta la libreria `Gson` per deserializzare un array Json, ne verifica l'integrità, ed ogni errore che si verifica in fase di parsing di un singolo appuntamento (non dell'intero stream) viene incapsulato in un oggetto di tipo `AppointmentParsingException` che, in questo caso, non viene utilizzato come eccezione, ma come bean.

AppointmentParsingException:

Anche se non è mai stata utilizzata come tale questa classe estende `Exception`. E' stata progettata in previsione di un utilizzo concreto: nel caso in cui gli errori di parsing diventino più frequenti e si richiede una gestione diretta di essi.

GUI:

Questa classe fa da interfaccia verso l'agenda per l'utente. Permette l'aggiunta, rimozione e manipolazione degli appuntamenti. Fornisce strumenti di ricerca.

Test:

Per ogni metodo richiesto che dovesse soddisfare dei requisiti specifici presenti nella consegna sono stati creati dei test.

Per ogni classe si effettuano i test di tutti i metodi richiesti o di di quelli a supporto del funzionamento di quelli richiesti.

Tutti i test, tranne quelli relativi all'I/O da file, sono stati effettuati generando dati casuali, al fine di garantire l'integrità dei test.