**Matteo Cartieri 00000000@studenti.uniupo.it**

**Nicholas Ternullo** [**00000000@studenti.uniupo.it**](mailto:20045859@studenti.uniupo.it)

**Progetto: Gestione di un’agenda di appuntamenti**

**Introduzione**

Questo progetto implementa un sistema per la gestione di un insieme di agende. In particolare un’agenda viene riconosciuta univocamente grazie al suo nome. Ogni agenda creata può contenere un insieme di appuntamenti caratterizzati dall’avere la data (dd/MM/yyyy), l’orario (HH:mm), la sua durata (in minuti), il nome della persona con cui si ha l’appuntamento e il luogo presso il quale si terrà.

**Le classi**

La classe principale è la classe Agenda: un oggetto di tipo Agenda(String, int) è caratterizzato da una stringa e un numero intero, in cui la stringa identifica il nome dell’Agenda, e l’intero è la dimensione massima di essa, ovvero il numero massimo di appuntamenti che vi possono essere inseriti all’interno. La classe rappresenta una lista di appuntamenti e permette l'inserimento e l'eliminazione di questi, ma anche la ricerca e la visualizzazione. L'implementazione prevede l'utilizzo di un'ArrayList di oggetti Appuntamento, inoltre la classe implementa l'interfaccia Iterable<Appuntamento> per poter scorrere l'elenco degli appuntamenti contenuti nell'agenda. È possibile anche salvare tutti i suoi dati in un file di testo, oppure crearne una nuova leggendo da esso.

È necessaria anche una classe Agende per raggruppare tutte le agende create; essa permette di effettuare le operazioni di aggiunta, rimozione e visualizzazione dell’elenco delle agende. Per effettuarle vi è un campo ArrayList di agende privato e statico. La funzione di aggiunta di un’agenda lancia un’eccezione se si cerca di aggiungerne una già creata nell’elenco. Anche la rimozione lancia un'eccezione se si cerca di cancellare un’agenda che non è presente.

La classe per l’oggetto di tipo Appuntamento comprende: una data espressa con il formato dd/MM/yyyy, un orario espresso con il formato HH:mm, la sua durata espressa in minuti, il nome della persona con cui si ha l’appuntamento e il luogo presso il quale si terrà. Per i casi in cui l’utente inserisca valori errati nel formato, è stato predisposto un controllo tramite i metodi ValidaData, ValidaOrario, ValidaDurata, ValidaNome e ValidaLuogo. Inoltre, con il metodo isSovrapposto, viene effettuato un ulteriore controllo al momento dell’inserimento dell’Appuntamento nell’Agenda, per verificare che non vi siano contemporaneità (anche parziali) tra diversi Appuntamenti.

Le eccezioni vengono sollevate richiamando la classe GestoreException (che estende la classe Exception), inviando messaggi diversi a seconda di metodi in cui viene richiamata.

**L’interfaccia**

Infine è stato necessario lo sviluppo di un’interfaccia, per accedere a tutte le funzioni dalla linea di comando. Dall’interfaccia si ha a disposizione un menu principale che permette la creazione di una nuova agenda o la gestione di essa. La creazione può avvenire tramite i dati inseriti nel terminale o dalla lettura di quelli di un file. Per quanto riguarda la gestione di un’agenda, vengono permessi l’inserimento e la modifica di appuntamenti, oltre alla ricerca e alla visualizzazione di questi; inoltre permette la scrittura dell’agenda su un file, oppure la cancellazione di questa.