

## Test tecnico 2

# Gestore eventi

Stiamo lavorando a un programma che deve gestire eventi (ad esempio concerti, conferenze, spettacoli,...)

Nome progetto e nome epo: **java-gestore-eventi**

Nome package: org.generation.italy.eventi



## MILESTONE 1

La consegna è di creare una classe **Evento** che abbia le seguenti proprietà:

- titolo
- data
- numero di posti in totale
- numero di posti prenotati

Quando si istanzia un nuovo evento questi attributi devono essere tutti valorizzati nel **costruttore**, tranne posti prenotati che va inizializzato a 0.

Inserire il controllo che la data non sia già passata e che il numero di posti totali sia positivo. In caso contrario sollevare opportune eccezioni.

Aggiungere metodi **getter** e **setter** in modo che:

- titolo sia in lettura e in scrittura
- data sia in lettura e scrittura
- numero di posti totale sia solo in lettura
- numero di posti prenotati sia solo in lettura

Vanno inoltre implementati dei **metodi public** che svolgono le seguenti funzioni:

1. **prenota**: aggiunge uno ai posti prenotati. Se l'evento è già passato o non ha posti disponibili deve sollevare un'eccezione.
2. **disdici**: riduce di uno i posti prenotati. Se l'evento è già passato o non ci sono prenotazioni deve sollevare un'eccezione.
3. **l'override** del metodo **toString()** in modo che venga restituita una stringa contenente: data formattata - titolo

Aggiungete eventuali **metodi** (public e private) che vi aiutino a svolgere le funzioni richieste.



## MILESTONE 2

1. Creare una classe **Main** di test, in cui si chiede all'utente di inserire un nuovo evento con tutti i parametri.
2. Dopo che l'evento è stato istanziato, chiedere all'utente se e quante prenotazioni vuole fare e provare ad effettuarle, implementando opportuni controlli e gestendo eventuali eccezioni.
3. Stampare a video il numero di posti prenotati e quelli disponibili
4. Chiedere all'utente se e quanti posti vuole disdire
5. Provare ad effettuare le disdette, implementando opportuni controlli e gestendo eventuali eccezioni
6. Stampare a video il numero di posti prenotati e quelli disponibili



## MILESTONE 3

Creare una classe **ProgrammEventi** con i seguenti **attributi**

- titolo: String
- eventi: List<Evento>

Nel **costruttore** valorizzare il titolo, passato come parametro, e inizializzare la lista di eventi come una nuova ArrayList

Aggiungere i seguenti **metodi**:

- un metodo che aggiunge alla lista un Evento, passato come parametro
- un metodo che restituisce una lista con tutti gli eventi presenti in una certa data
- un metodo che restituisce quanti eventi sono presenti nel programma
- un metodo che svuota la lista di eventi
- un metodo che restituisce una stringa che mostra il titolo del programma e tutti gli eventi ordinati per data nella forma:
  - data1 - titolo1
  - data2 - titolo2
  - data3 - titolo3
  - ...



## MILESTONE 4

Creare una classe **Concerto** che estende **Evento**, che ha anche gli **attributi**:

- ora: **LocalTime**
- prezzo: **BigDecimal**

Aggiungere questi attributi nel **costruttore** e implementarne **getter** e **setter**

Aggiungere i **metodi** per restituire data e ora formattata e prezzo formattato (##,##€)

Fare l'**override** del metodo **toString()** in modo che venga restituita una stringa del tipo:  
data e ora formattata - titolo - prezzo formattato