

BEPT - Milestone 3

Gestore Eventi

Nome repo: **java-gestore-eventi**

Obiettivi didattici

Devi saper fare:

- Creare un programma in JAVA eseguibile da Terminale
- Saper tipizzare correttamente una variabile
- Conoscere le strutture dati e relativi metodi
- Accedere ai valori in lettura e scrittura
- Iterare array e liste
- Creare una classe e istanziare oggetti
- Sfruttare al meglio i principi di OOP

Requisiti

Bisogna conoscere:

- Fundamentals Java
- Concetti di OOP:
 - Classi e Oggetti
 - Costruttore
 - Attributi
 - Incapsulamento
 - Ereditarietà
 - Composizione
 - Polimorfismo
 - Metodi e attributi statici
 - Interfacce
 - Classi astratte
- Java Collections
- Gestione date

Consegna

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

Step 1

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 mostrare i dovuti avvisi all'utente

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 restituire un messaggio di avviso.
2. **disdici**: riduce di uno i posti prenotati. Se l'evento è già passato o non ci sono prenotazioni restituisce un messaggio di avviso.
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.

Step 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
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
6. Stampare a video il numero di posti prenotati e quelli disponibili

Step 3

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

- ora (LocalTime)
- prezzo

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*

Step 4 (Bonus opzionale)

Creare una classe **ProgrammaEventi** con i seguenti attributi:

- titolo: String (*una variabile di istanza chiamata titolo, di tipo String*)
- eventi: List<Evento> (*una variabile di istanza chiamata eventi, di tipo "implementatore di" 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
 - ...