

## Test tecnico 2

# Gestore eventi

Nome progetto Visual Studio: **GestoreEventi**

Nome repo: **csharp-gestore-eventi**



### Contesto e problema da risolvere:

Immaginate di lavorare in una software house, che ha diversi clienti. Vi è stato commissionato da parte della vostra azienda la creazione di un **gestionale eventi** per eventi come concerti, conferenze, spettacoli,... per un suo cliente. Il cliente necessita di un semplice programma senza interfaccia grafica (ossia che venga eseguito in console o terminale) che si occupa di:

- Memorizzare e tenere traccia di tutti gli eventi in futuro che ha programmato
- Poter gestire le prenotazioni e le disdette delle sue conferenze e tenere traccia quindi dei posti prenotati e di quelli disponibili per un dato evento
- Poter gestire un intero programma di Eventi (ossia tenere traccia di tutti gli eventi che afferiscono ad serie di Conferenze)

# MILESTONE 1

Per prima cosa è necessario creare una classe **Evento** che abbia i seguenti **attributi**:

- titolo
- data
- capienza massima dell'evento
- numero di posti prenotati

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

- titolo sia in lettura e in scrittura
- data sia in lettura e scrittura
- numero di posti per la capienza massima sia solo in lettura
- numero di posti prenotati sia solo in lettura

ai setters inserire gli opportuni controlli in modo che la data non sia già passata, che il titolo non sia vuoto e che la capienza massima di posti sia un numero positivo. In caso contrario sollevare opportune eccezioni.

Dichiarare un **costruttore** che prenda come parametri il titolo, la data e i posti a disposizione e inizializza gli opportuni attributi facendo uso dei metodi e controlli già fatti. Per l'attributo posti prenotati invece si occupa di inizializzarlo lui a 0.

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

1. **PrenotaPosti**: aggiunge i posti passati come parametro ai posti prenotati. Se l'evento è già passato o non ha posti o non ha più posti disponibili deve sollevare un'eccezione.
2. **DisdiciPosti**: riduce del i posti prenotati del numero di posti indicati come parametro. Se l'evento è già passato o non ci sono i posti da disdire sufficienti, deve sollevare un'eccezione.
3. **l'override** del metodo **ToString()** in modo che venga restituita una stringa contenente: data formattata – titolo

Le eccezioni possono essere generiche (Exception) o usate quelle già messe a disposizione da C#, ma aggiungete un eventuale messaggio chiaro per comunicare che cosa è successo.

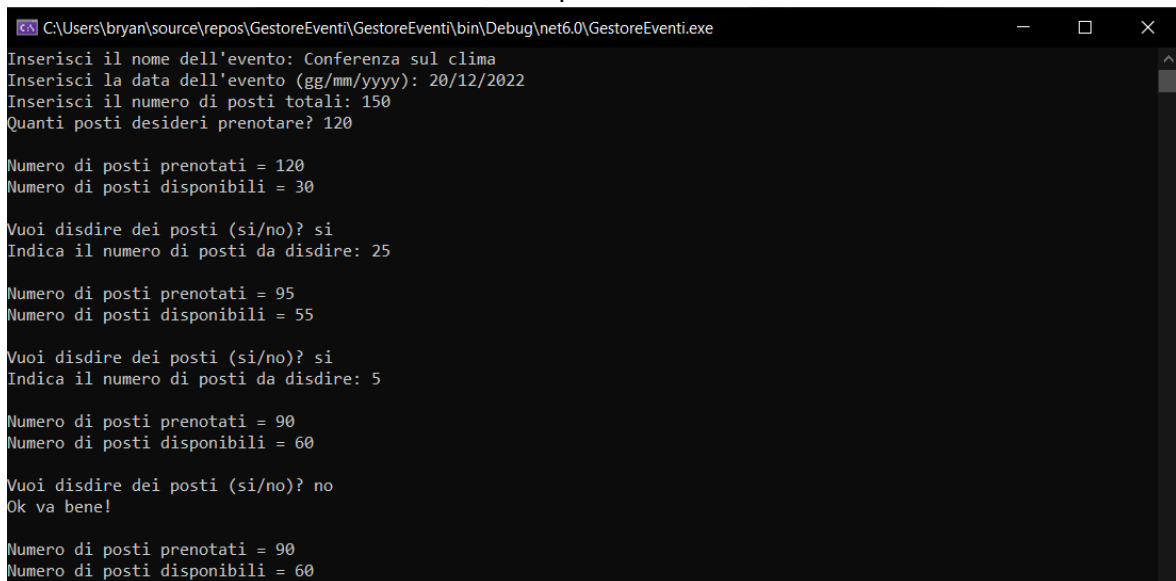
Vi ricordo che man mano che andrete avanti con le altre milestones, potrete aggiungere più avanti altri eventuali **metodi** (public e private) che vi aiutino a svolgere le funzioni richieste se ritenete necessari!

## MILESTONE 2

1. Nel vostro programma principale Program.cs, chiedete all'utente di inserire un nuovo evento con tutti i parametri richiesti dal costruttore.
2. Dopo che l'evento è stato istanziato, chiedete all'utente se e quante prenotazioni vuole fare e provare ad effettuarle.
3. Stampare a video il numero di posti prenotati e quelli disponibili.
4. Ora chiedere all'utente fintanto che lo desidera, **se e quanti** posti vuole disdire. Ogni volta che disdice dei posti, stampare i posti residui e quelli prenotati.

**Attenzione:** In questa prima fase non è necessario gestire le eventuali eccezioni da Program.cs che potrebbero insorgere, eventualmente il programma si blocca se qualcosa va storto che voi avevate già previsto. Piuttosto pensate bene alla logica del vostro programma principale e della vostra classe Evento pensando bene alle responsabilità dei metodi e alla visibilità di attributi e metodi.

Questo dovrebbe essere il risultato fino a qui:



```
C:\Users\bryan\source\repos\GestoreEventi\GestoreEventi\bin\Debug\net6.0\GestoreEventi.exe
Inserisci il nome dell'evento: Conferenza sul clima
Inserisci la data dell'evento (gg/mm/yyyy): 20/12/2022
Inserisci il numero di posti totali: 150
Quanti posti desideri prenotare? 120

Numero di posti prenotati = 120
Numero di posti disponibili = 30

Vuoi disdire dei posti (si/no)? si
Indica il numero di posti da disdire: 25

Numero di posti prenotati = 95
Numero di posti disponibili = 55

Vuoi disdire dei posti (si/no)? si
Indica il numero di posti da disdire: 5

Numero di posti prenotati = 90
Numero di posti disponibili = 60

Vuoi disdire dei posti (si/no)? no
Ok va bene!

Numero di posti prenotati = 90
Numero di posti disponibili = 60
```

## MILESTONE 3

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

- string Titolo
- List<Evento> eventi

Nel **costruttore** valorizzare il titolo, passato come parametro, e inizializzate la lista di eventi come una nuova Lista vuota di eventi.

Aggiungete poi i seguenti **metodi**:

- un metodo che aggiunge alla lista del programma eventi un Evento, passato come parametro al metodo.
- un metodo che restituisce una lista di eventi con tutti gli eventi presenti in una certa data.
- un metodo statico che si occupa, presa una lista di eventi, di stamparla in Console, o ancora meglio vi restituisca la rappresentazione in stringa della vostra lista di eventi.
- un metodo che restituisce quanti eventi sono presenti nel programma eventi attualmente.
- un metodo che svuota la lista di eventi.
- un metodo che restituisce una stringa che mostra il titolo del programma e tutti gli eventi aggiunti alla lista. Come nell'esempio qui sotto:

Nome programma evento:

data1 - titolo1

data2 - titolo2

data3 - titolo3

...

## MILESTONE 4

Una volta completata la classe **ProgrammaEventi**, usatela nel vostro programma principale Program.cs per creare un nuovo programma di Eventi che l'utente vuole organizzare, chiedendogli qual è il titolo del suo programma eventi.

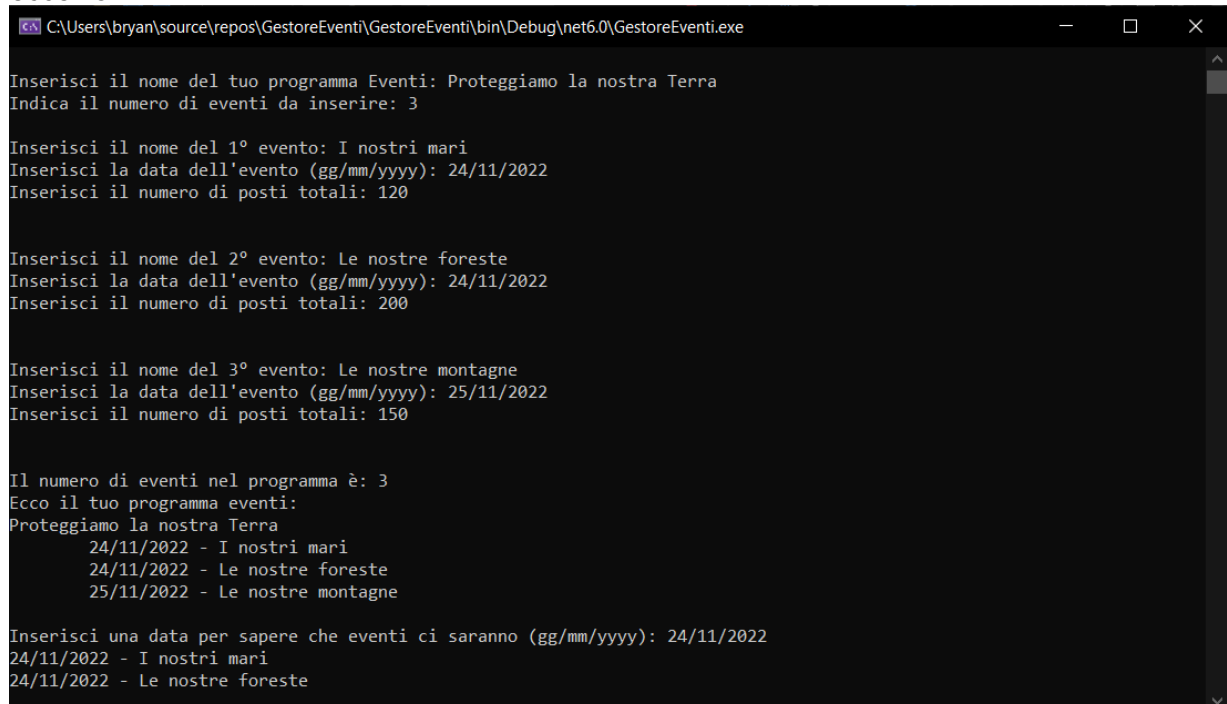
Chiedete poi al vostro utente quanti eventi vuole aggiungere, e fategli inserire ad uno ad uno tutti gli eventi necessari chiedendo man mano tutte le informazioni richieste all'utente.

**Attenzione:** qui si gestite eventuali eccezioni lanciate dagli eventi creati, in questo caso il programma informa l'utente dell'errore e non aggiunge l'evento al programma eventi (o meglio alla lista di Eventi del programmaEventi), ma comunque chiederà in continuazione all'utente di inserire eventi fintanto che non raggiunge il numero di eventi specificato inizialmente dall'utente.

Una volta compilati tutti gli eventi:

1. Stampate il numero di eventi presenti nel vostro programma eventi
2. Stampate la lista di eventi inseriti nel vostro programma usando il metodo già fatto
3. Chiedere all'utente una data e stampate tutti gli eventi in quella data. Usate il metodo che vi restituisce una lista di eventi in una data dichiarata e create un **metodo statico** che si occupa di stampare una lista di eventi che gli arriva. Passate dunque la lista di eventi in data al metodo statico, per poterla stampare.
4. Eliminate tutti gli eventi dal vostro programma.

Questo dovrebbe essere il risultato finale del vostro programma se non sono accadute eccezioni:



```
C:\Users\bryan\source\repos\GestoreEventi\GestoreEventi\bin\Debug\net6.0\GestoreEventi.exe

Inserisci il nome del tuo programma Eventi: Proteggiamo la nostra Terra
Indica il numero di eventi da inserire: 3

Inserisci il nome del 1° evento: I nostri mari
Inserisci la data dell'evento (gg/mm/yyyy): 24/11/2022
Inserisci il numero di posti totali: 120

Inserisci il nome del 2° evento: Le nostre foreste
Inserisci la data dell'evento (gg/mm/yyyy): 24/11/2022
Inserisci il numero di posti totali: 200

Inserisci il nome del 3° evento: Le nostre montagne
Inserisci la data dell'evento (gg/mm/yyyy): 25/11/2022
Inserisci il numero di posti totali: 150

Il numero di eventi nel programma è: 3
Ecco il tuo programma eventi:
Proteggiamo la nostra Terra
    24/11/2022 - I nostri mari
    24/11/2022 - Le nostre foreste
    25/11/2022 - Le nostre montagne

Inserisci una data per sapere che eventi ci saranno (gg/mm/yyyy): 24/11/2022
24/11/2022 - I nostri mari
24/11/2022 - Le nostre foreste
```

Se ci dovessero essere eccezioni il programma vi avvisa su qual è l'errore commesso e vi richiede di reinserire l'evento.

# BONUS

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

- **relatore**: string
- **prezzo**: double

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

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

Fate l'**override** del metodo **ToString()** in modo che venga restituita una stringa del tipo: data - titolo - relatore - prezzo formattato.

A questo punto provate ad aggiungere al vostro programma oltre che dei semplici eventi anche delle vere e proprie conferenze (potete fare a meno di svuotare la lista precedentemente creata, commentando il metodo svuota Lista di Eventi).

A questo punto se provate a stampare la vostra lista di eventi con l'apposito metodo che avete fatto nella classe **Evento**, se lo avete fatto bene avvalendovi del **ToString()** per stampare gli eventi, noterete come il metodo **ToString()** di **Evento** e di **Conferenza** vengono correttamente chiamati a seconda che nel programma eventi ci sia un evento normale o una conferenza.

Questo dovrebbe essere l'output finale del vostro programma:

```
---- BONUS ----

Aggiungiamo anche un conferenza!
Inserisci il nome della conferenza: Intervento di Fondazione Terra
Inserisci la data del conferenza (gg/mm/yyyy): 26/11/2022
Inserisci il numero di posti per la conferenza: 200
Inserisci il relatore della conferenza: Pierpaolo Ricci
Inserisci il prezzo del biglietto della conferenza: 12

Ecco il tuo programma eventi con anche le Conferenze:
Proteggiamo la nsotra Terra
  24/11/2022 - I nostri mari
  24/11/2022 - Le nostre foreste
  25/11/2022 - Le nostre montagne
  26/11/2022 - Intervento di Fondazione Terra - Pierpaolo Ricci - 12,00 euro

C:\Users\bryan\source\repos\GestoreEventi\GestoreEventi\bin\Debug\net6.0\GestoreEventi.exe (process 9384) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```