

## Esercitazione su I/O – Stream di caratteri

E' richiesto di implementare un parser per un report di eventi sismici generato da INGV

<http://terremoti.ingv.it/>

L'elenco di eventi è presente nel file di testo allegato "query".

Il file è strutturato in modo che ogni riga riporti un evento. In particolare, per ciascun evento vengono indicati i valori delle proprietà che lo caratterizzano presentati nel seguente ordine e con il carattere di delimitazione '|':

```
#EventID|Time|Latitude|Longitude|Depth/Km|Author|Catalog|Contributor|ContributorID|MagType|Magnitude|MagAuthor|EventLocationName
```

Nello specifico si richiede di implementare le seguenti classi:

- La classe **EQEvent** che rappresenta l'evento e che include tutti gli attributi ad esso relativi. Scegliere per gli attributi il nome indicato sopra rispettando le convenzioni di Java per i nomi (per `depth/km` si può usare `depthKm`) e il tipo di dato che si ritiene conveniente. Può essere utile implementare i getter e i setter per tutti gli attributi. Prestare particolare attenzione alla gestione dell'attributo **Time**\*
- La classe **EQReport** che rappresenta il report ed è caratterizzata da un nome e da una opportuna struttura dati per realizzare la collezione. La classe presenta i seguenti metodi:
  - `public void addEvent(EQEvent e)` – per aggiungere un nuovo evento alla collezione;
  - `public void sort()` - per ordinare gli eventi secondo l'ordine dettato da EventID crescente
  - `public void sort(Comparator<EQEvent> c)` – per ordinare gli eventi secondo un ordine definito da un comparatore passato come parametro
  - `public static EQReport readFromTextFile(String filename)` – per ottenere un report mediante lettura di un file di testo il cui nome è passato come parametro
  - `public static void printToTextFile(EQReport eqr, String filename)` – per scrivere un report passato come parametro su un file di testo il cui nome è passato come parametro
- la classe **MagnitudeComparator** implementa un comparatore (`Comparator<EQEvent>`) che consente di ordinare il report per magnitudo (ordine crescente).

Come risultato dell'esercitazione, si chiede di stampare su un nuovo file di testo l'elenco di eventi in ordine crescente di magnitudo, mantenendo per il file prodotto la convenzione di formato definita dal file di input.

Viene fornito come allegato la classe **EQTest** che contiene il metodo main.

\*per la gestione della proprietà **Time** si consiglia di consultare la documentazione Java relativamente alle seguenti classi:

- `java.time.LocalDateTime`
- `java.time.Instant`
- `java.time.ZoneId`