

Simulazione d'esame del 14/06/2017

Si consideri il database `artsmia.sql` (<https://github.com/artsmia/collection>) contenente informazioni su opere e mostre del Minneapolis Institute of Art, in Minnesota, USA. Il database è strutturato secondo il diagramma ER illustrato nella pagina seguente.



Si intende costruire un'applicazione JavaFX che permetta di interrogare tale base dati, visualizzando informazioni sulle mostre e simulando un flusso di visitatori. In particolare, l'applicazione dovrà svolgere le seguenti funzioni:

PUNTO 1

- Popolare il menù a tendina dell'interfaccia grafica con tutti gli anni in cui sono state organizzate delle mostre (campo *begin* nella tabella *exhibitions*).
- Facendo click sul pulsante *Crea grafo*, l'applicazione deve creare un grafo orientato e non pesato i cui nodi rappresentano le mostre iniziate a partire dall'anno selezionato (tabella *exhibitions*) e tale per cui due mostre sono collegate se sono temporalmente successive (utilizzare la data di inizio), ma hanno almeno un anno in comune. Utilizzare i campi *begin* e *end* della tabella *exhibitions* per ottenere l'informazione sugli anni di inizio e fine di ciascuna mostra.

Esemio: Se mostra A [2012-2015] e mostra B [2013-2017], A a B sono collegate.

- In seguito, utilizzare l'area di testo nell'interfaccia grafica per rispondere alle seguenti domande:
 - Il grafo è fortemente connesso?
 - Quale tra le mostre selezionate ha il maggior numero di opere d'arte in esposizione?

PUNTO 2

- L'utente inserisce nell'interfaccia grafica il numero N di studenti da utilizzare nella simulazione.
- Si implementi la seguente simulazione: N studenti d'arte scelgono in modo casuale una stessa mostra a cui partecipare, con il solo vincolo che l'anno di inizio coincida con quello selezionato nel menù a tendina. Successivamente, in ciascun passo della simulazione, ogni studente sceglie in modo casuale la successiva mostra a cui assistere tra quelle raggiungibili a partire da quella corrente (utilizzare il grafo creato al primo punto).
- Al termine della simulazione, visualizzare la classifica (in ordine decrescente) del numero di opere viste da ciascuno studente, ipotizzando che partecipando ad una mostra lo studente veda tutte le opere esposte.
- Nota: ai fini della simulazione si è interessati solo al numero di opere viste e non è necessario visualizzare i dettagli di ciascuna opera. Inoltre, se la stessa opera è presente in più mostre a cui lo studente partecipa deve essere contata una sola volta.

Nella realizzazione del codice, si lavori a partire dalle classi (Bean e DAO, FXML) e dal database contenuti nel progetto di base. È ovviamente permesso aggiungere o modificare classi e metodi.

Tutti i possibili errori di immissione, validazione dati, accesso al database, ed algoritmici devono essere gestiti, non sono ammesse eccezioni generate dal programma.

Le tabelle **artists**, **objects** e **exhibitions** contengono rispettivamente informazioni su artisti, opere del museo e mostre (sia presenti, che passate). La tabella **authorship** mette in relazione artisti ed opere, mentre la tabella **exhibitions_objects** elenca per ciascuna mostra le opere esposte.

