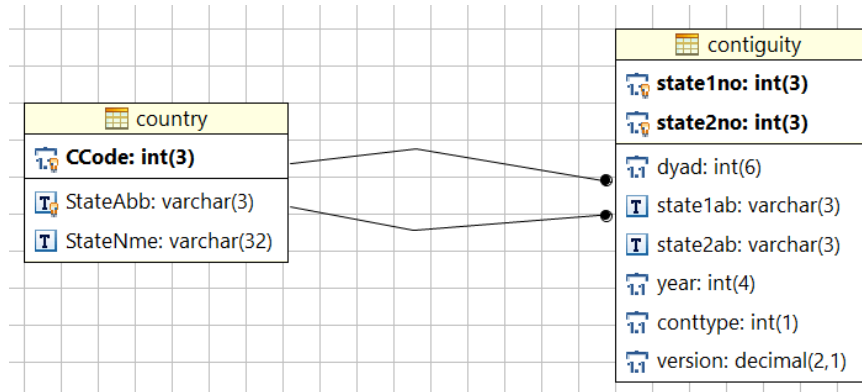


Prova d'esame del 23/01/2017 – Tema unico

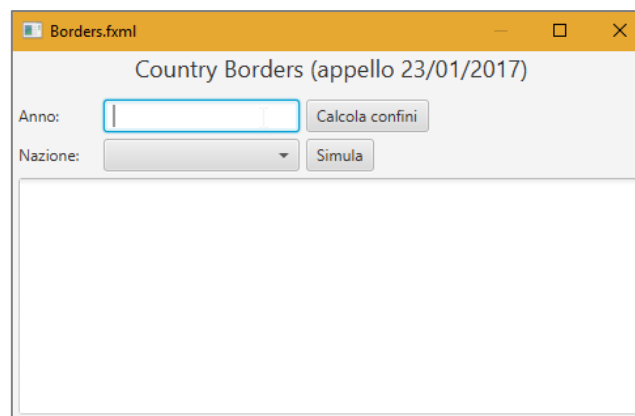
Si consideri il database “country borders”, contenente informazioni sull’evoluzione dei confini tra le diverse nazioni del mondo, al passare degli anni. I dati sono estratti dal progetto “The Correlates of War” <http://www.correlatesofwar.org/>



La tabella *country* riporta tutte le nazioni definite in cui ciascuna nazione è identificata da un numero, da un’abbreviazione univoca di 3 lettere e dal nome completo. La tabella *contiguity* rappresenta, per ogni coppia di stati (*state1no*, *state2no*), il fatto che tali stati fossero confinanti a partire dall’anno *year* (anni compresi tra il 1816 ed il 2006). I tipi di confine sono diversi, ma ci interessa **esclusivamente il confine via terra** (*conttype*=1).

[Nota: la tabella *contiguity2006* non deve essere utilizzata]

Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati, e restituire informazioni utili all’analisi di flussi migratori tra gli stati. L’applicazione dovrà svolgere le seguenti funzioni:



PUNTO 1

- Permettere all’utente di specificare un anno di interesse *x* (tramite casella di testo o menu a tendina) e di richiedere il calcolo dei confini.
- Creare un grafo che rappresenti la situazione dei confini mondiali via terra (*conttype*=1) fino all’anno *x* specificato dall’utente. Le relazioni di confine (via terra) da considerare sono tutte quelle il cui campo *year* sia minore o uguale all’anno specificato. I vertici del grafo devono rappresentare le nazioni per le quali esiste un confine via terra nell’arco di tempo risultante, e gli archi (non orientati, non pesati) devono rappresentare il fatto che due nazioni hanno confinato via terra (*conttype*=1) in quel periodo.
- Stampare l’elenco degli stati (abbreviazione e nome), indicando per ciascuno il numero di stati confinanti secondo il grafo creato. L’elenco deve essere stampato in ordine decrescente del numero di stati confinanti.

PUNTO 2

- a. Simulare la diffusione di una serie di 1000 immigrati, che vengono inseriti in uno stato iniziale (selezionato dall'utente tra quelli presenti nel grafo) e che si spostano tra gli stati secondo le regole enunciate di seguito.
 1. Al tempo iniziale $T=1$, tutti i 1000 immigrati sono posizionati nello stato iniziale.
 2. Ogni volta che degli immigrati entrano in uno stato (sia al tempo $T=1$, che ai tempi successivi), immediatamente il 50% di tali persone diventano "stanziali" (ossia non si muoveranno più dallo stato). Il rimanente 50% (approssimato per difetto) rimane non-stanziale, e si sposterà negli stati confinanti.
 3. Ad ogni nuovo istante di tempo ($T+1$), la popolazione non-stanziale di ciascuno stato si sposta, e si divide in parti uguali (approssimate per difetto) tra tutti gli stati confinanti. Ad esempio, se in uno stato vi sono 40 persone non-stanziali, e vi sono 3 stati confinanti, in ciascuno stato confinante si sposteranno 13 persone ($\lfloor 40/3 \rfloor$) – si assuma che la 40esima persona (il resto della divisione) divenga stanziale.
 4. Come caso particolare, se il numero di persone non-stanziali presenti in uno stato è inferiore al numero di stati confinanti, allora nessuno si sposterà e tutti diverranno stanziali.
 5. Nella scelta degli spostamenti, tutti gli stati confinanti avranno lo stesso numero di persone. È quindi possibile che alcune persone "ritornino indietro" al tempo $T+2$ nello stesso stato in cui erano al tempo T . È un fenomeno normale e non deve essere trattato in nessun modo particolare.
 6. Ad ogni passo di simulazione, il numero di persone ancora non-stanziali si riduce almeno del 50%. Pertanto, ad un certo punto, la simulazione si esaurirà in quanto tutti saranno divenuti stanziali.
- b. Al termine della simulazione, stampare:
 1. Il numero di passi T simulati
 2. Per ogni stato in cui vi sia almeno una persona stanziale, stampare il nome dello stato (abbreviazione + nome) ed il numero di persone presenti. L'elenco deve essere in ordine decrescente di numero di persone.

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.