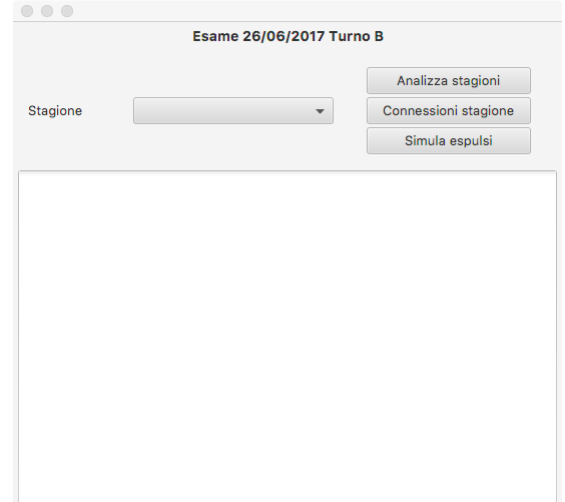


# Prova d'esame del 26/06/2017 – Turno B

Si consideri il database “serie\_a”, contenente i risultati di tutte le partite di calcio della Serie A italiana, tra la stagione 2002/2003 e la stagione 2016/2017. Il database (tratto dal sito <http://www.football-data.co.uk/italym.php>) è strutturato secondo il diagramma ER della pagina seguente.

Si intende costruire un'applicazione JavaFX che permetta di interrogare tale base dati, e calcolare informazioni a proposito delle gare disputate. L'applicazione dovrà svolgere le seguenti funzioni:



## PUNTO 1

- Permettere all'utente di selezionare il bottone “Analizza stagioni”.
- Alla pressione del bottone, creare un grafo che rappresenti per tutte le stagioni, la continuità della presenza di ciascuna squadra da una stagione all'altra. Il grafo dovrà essere non orientato e pesato, con i vertici che rappresentino le stagioni presenti nel database, e gli archi il fatto in tale coppia di stagioni ci siano delle squadre comuni (ossia squadre che hanno giocato in entrambe le stagioni). Il peso dell'arco tra due stagioni deve rappresentare il *numero di squadre* che hanno giocato nelle due stagioni.
- Permettere all'utente di selezionare, da un menu a tendina, una delle stagioni presenti nel grafo, e premere il bottone “Connessioni Stagione”.
- Per la stagione selezionata nel menu a tendina, stampare l'elenco delle stagioni aventi squadre comuni con quella selezionata, indicando il numero di squadre comuni ed ordinando il risultato per stagioni con data crescente.

## PUNTO 2

- Permettere all'utente di selezionare una stagione dal corrispondente menu a tendina e di premere il bottone “Simula espulsi”.
- Alla pressione del bottone effettuare una simulazione del *numero di espulsi* appartenenti alle diverse squadre. Il simulatore deve considerare tutte le partite giocate nella stagione, in ordine crescente di data. (Nota: il grafo creato al primo punto non risulta utile ai fini della simulazione e non è necessario costruirne un altro).
- All'inizio della simulazione, si ipotizzi che ciascuna squadra abbia lo stesso numero di giocatori  $G$ , con  $G=11$ .
- Il risultato di ogni partita “simulata” dipenderà dal risultato “storico”, ossia quello presente nel database, corretto da un fattore che tiene conto dei giocatori presenti in campo. In una ipotetica partita della squadra A contro la squadra B, sia  $G_A$  il numero di giocatori della squadra A e  $G_B$  il numero di giocatori della squadra B. Se  $G_A < G_B$  ( $G_A/G_B$  è un numero compreso tra 0 e 1) simulare che, con una probabilità pari a  $(1-G_A/G_B)$ , la squadra A (quella con meno giocatori), faccia un goal in meno rispetto al risultato storico. Nel caso in cui  $G_A > G_B$ , allora sarà la squadra B ad avere una probabilità  $(1-G_B/G_A)$  di fare un goal in meno.
- Al termine di ogni partita, esiste una probabilità pari a  $P$  (si ipotizzi  $P=20\%$ ) che un giocatore *della squadra perdente* venga squalificato per un certo numero di partite. I giocatori della squadra vincente non vengono mai squalificati. Il numero di giornate per cui il giocatore rimane sospeso è un numero casuale compreso tra 1 e 4. Dopo le giornate di squalifica, il giocatore ritorna regolarmente in squadra.
- Al termine della stagione, si stampi, per ciascuna squadra, il numero di giornate totali di squalifica (ossia la somma di tutte le giornate di squalifica di tutti i propri giocatori) ed il numero di punti in classifica, considerando +3 punti per ogni partita vinta, +1 in caso di pareggio, 0 in caso di perdita.

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.

Legenda (tabella matches):

- Season = Season year  
(foreign key to table seasons)
- Div = League Division
- Date = Match Date
- HomeTeam = Home Team  
(foreign key to table teams)
- AwayTeam = Away Team  
(foreign key to table teams)
- FTHG = Full Time Home Team Goals
- FTAG = Full Time Away Team Goals
- FTR = Full Time Result  
(H=Home Win, D=Draw, A=Away Win)
- HTHG = Half Time<sup>(\*)</sup> Home Team Goals
- HTAG = Half Time<sup>(\*)</sup> Away Team Goals
- HTR = Half Time<sup>(\*)</sup> Result  
(H=Home Win, D=Draw, A=Away Win)

(\*) tranne quanto la partita è assegnata a tavolino

Parametri aggiuntivi (non presenti in tutte le stagioni):

- HS = Home Team Shots
- AS = Away Team Shots
- HST = Home Team Shots on Target
- AST = Away Team Shots on Target
- HHW = Home Team Hit Woodwork
- AHW = Away Team Hit Woodwork
- HC = Home Team Corners
- AC = Away Team Corners
- HF = Home Team Fouls Committed
- AF = Away Team Fouls Committed
- HO = Home Team Offsides
- AO = Away Team Offsides
- HY = Home Team Yellow Cards
- AY = Away Team Yellow Cards
- HR = Home Team Red Cards
- AR = Away Team Red Cards

