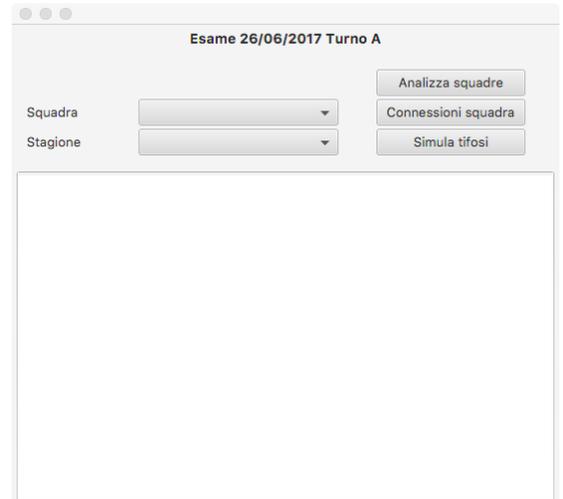


# Prova d'esame del 26/06/2017 – Turno A

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 squadre”.
- Alla pressione del bottone, creare un grafo che rappresenti le partite giocate da ogni coppia di squadre, in tutte le stagioni disponibili. Il grafo dovrà essere non orientato e pesato, con i vertici che rappresentino le squadre che hanno giocato in almeno una stagione, e gli archi il fatto che tali squadre abbiano giocato insieme almeno una volta. Il peso dell'arco tra due team deve rappresentare il *numero di partite* che quelle due squadre hanno giocato (contando sia le partite di andata che di ritorno, in ogni stagione).
- Permettere all'utente di selezionare, da un menu a tendina, una delle squadre presenti nel grafo, e premere il bottone “Connessioni Squadra”.
- Per la squadra selezionata nel menu a tendina, stampare l'elenco delle squadre con cui ha giocato (ed il numero di partite giocate), in ordine decrescente di numero di partite.

## PUNTO 2

- Permettere all'utente di selezionare una stagione dal corrispondente menu a tendina e di premere il bottone “Simula tifosi”.
- Alla pressione del bottone effettuare una simulazione del *numero di tifosi* 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 tifosi  $T$ , con  $T=1000$ .
- Il risultato di ogni partita “simulata” dipenderà dal risultato “storico”, ossia quello presente nel database, corretto da un fattore che tenga conto del numero di tifosi. Infatti, in presenza di un maggior numero di tifosi è più facile vincere. In una ipotetica partita della squadra A contro la squadra B, sia  $T_A$  il numero di tifosi della squadra A e  $T_B$  il numero di tifosi della squadra B. Se  $T_A < T_B$  ( $T_A/T_B$  è un numero compreso tra 0 e 1) simulare che, con una probabilità pari a  $(1-T_A/T_B)$ , la squadra A (quella con meno tifosi) faccia un goal in meno rispetto al risultato storico. Nel caso in cui  $T_A > T_B$ , allora sarà la squadra B ad avere una probabilità  $(1-T_B/T_A)$  di fare un goal in meno.
- Al termine di ogni partita, alcuni dei tifosi della squadra che ha perso decidono di passare alla squadra vincente. La percentuale di tifosi che cambierà squadra sarà proporzionale allo scarto-reti moltiplicato per un fattore  $P$  (si ipotizzi  $P=10$ ). Ad esempio, se una il risultato fosse A-B=3-1, lo scarto reti sarebbe pari a 2, ed il  $2*10=20\%$  dei tifosi della squadra B passerebbe alla squadra A. La squadra che vince la partita non perde alcun tifoso (ma ne guadagnerà). In caso di pareggio, nessun tifoso si sposta.
- Al termine della stagione, si stampi per ciascuna squadra il numero di tifosi 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

