Analisi e Progettazione del Software
Prova scritta del 7 febbraio 2017

Si vuole progettare una classe C++ per la gestione di una partita di calcio e della sua evoluzione in termini di formazioni. Una partita è caratterizzata dalle squadre che giocano (una in casa e una ospite), dalla data e dal luogo dove si svolge. Della partita vengono memorizzati anche i giocatori che partecipano e le sostituzioni che avvengono. Le operazioni della classe sono le seguenti:

- $InserisciGiocatore(s : Squadra, g : Giocatore)$:
  Il giocatore g viene inserito nella formazione della squadra s.
  Precondizioni: la partita non è iniziata ed s è una delle due squadre; il giocatore g è nella rosa della squadra s; se g è portiere, non è stato ancora inserito un portiere per s; se g è l'11-esimo giocatore, deve essere stato inserito un portiere per s oppure g è portiere.

- $IniziaPartita()$:
  Viene registrato che la partita ha inizio.
  Precondizioni: La partita non è ancora iniziata; le formazioni sono complete, cioè 11 con giocatori inseriti per squadra (di cui un portiere).

- $RegistraSostituzione(g_1 : Giocatore, g_2 : Giocatore, s : Squadra, m : intero)$:
  Viene registrata la sostituzione del giocatore g_1 con il giocatore g_2 per la squadra s al minuto m.
  Precondizioni: la partita è in corso; nessuna sostituzione è stata registrata in un minuto successivo ad m; il giocatore g_1 è in campo per la squadra s; il giocatore g_2 è nella rosa della squadra s (non in campo o già uscito) ed ha lo stesso ruolo di g_1.

- $TerminaPartita()$:
  La partita termina, nessuna operazione può più essere eseguita sulla partita.
  Precondizioni: la partita è iniziata.

Si assumano già disponibili le classi `Squadra` e `Giocatore` delle quali non si conosce la definizione. Si sa solo che la classe `Giocatore` ha il metodo `Ruolo` che restituisce un valore tra `portiere`, `difensore`, `centrocampista`, e `attaccante`, e il metodo `VediSquadra` che restituisce un puntatore alla squadra in cui il giocatore milita.

Esercizio 1 (punti 5) Si disegni il diagramma UML delle classi per l'applicazione, che comprenda anche le responsabilità delle associazioni.

Esercizio 2 (punti 4) Si scriva la definizione della classe `Partita`.

Esercizio 3 (punti 12) Si scrivano le definizioni dei metodi della classe `Partita` che corrispondono alle operazioni sopra elencate e i selettori che si ritengono opportuni. Si gestiscano le precondizioni tramite il lancio di opportune eccezioni.

Esercizio 4 (punti 4) Si scriva l'operatore di output della classe `Partita`, in modo che stampi anche tutti i dati ad essa collegati. Si assumano già disponibili gli operatori di output per le classi `Squadra` e `Giocatore`.

Esercizio 5 (punti 5) Si scriva una funzione esterna che riceva come parametro un vettore di partite ed un giocatore e restituisca il numero di volte che il giocatore è entrato nel secondo tempo (minuto successivo al 46°).