Corso di Laurea in Ingegneria Informatica

Prova scritta di *Fondamenti di informatica e laboratorio* – Modulo 2

8 novembre 2019

**Tempo a disposizione: 2 ore**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Cognome e nome:*** |  | ***Matricola:*** |  | ***Corso:*** | ***◊ A  ◊ B*** |

In base alle specifiche riportate di seguito, si implementi in Java un sistema di supporto all’emissione dei biglietti per le partite dei campionati di calcio italiani. Il sistema (implementato in una classe *Sistema*) gestisce informazioni sulle partite (istanze della classe *Partita*), sui biglietti emessi (istanze della classe *Biglietto*) e sulle persone che acquistano un biglietto (istanze della classe *Persona*).

Ogni *Partita* è identificata da un *codice* e caratterizzata dal *nome* della squadra di casa, dal *nome* della squadra ospite e dalla *data* in cui si gioca la partita. Un *Biglietto* è identificato dal *numero di serie* e caratterizzato dal *codice* della partita, dal *numero di posto*, dal *settore* e dal *codice fiscale* della persona che lo ha acquistato. Una *Persona* è identificata da un *codice fiscale* e caratterizzata da *data* e *luogo di nascita*.

Si implementino in Java le classi *Partita*, *Biglietto* e *Persona*. Oltre a scrivere eventuali metodi che si ritengono necessari per implementare l’applicazione, occorre fornire almeno i seguenti metodi nella classe *Sistema*:

* 1. *public Biglietto* ***emettiBiglietto****(Partita pa, Persona pe, String settore)*. Il metodo riceve in input una Partita, una Persona e il nome di un settore (per es. curva, tribuna, etc.) e restituisce un Biglietto con: *(i)* un numero di serie calcolato come il numero di serie del biglietto emesso precedentemente più 1; *(ii)* un numero di posto scelto a caso tra i posti che nel settore passato come parametro risultano liberi al momento dell’acquisto (si supponga di avere 1000 posti per settore); *(iii)* tutti gli altri elementi che contraddistinguono un biglietto come descritto sopra. Inoltre, il biglietto può essere emesso solo se la stessa Persona non ha già acquistato un altro biglietto per la stessa Partita, e se esistono posti liberi nel settore passato come parametro, altrimenti il metodo restituisce *null*.
  2. *public ArrayList<Biglietto>* ***listaBiglietti****(Partita p)*. Il metodo restituisce la lista dei biglietti venduti per la partita *p* passata come parametro, ordinati in modo crescente rispetto alla data di nascita degli acquirenti.
  3. *public HashMap<String, Integer>* ***statistica****(Partita p)*. Il metodo restituisce una *HashMap*, in cui per ogni settore dello stadio in cui si è giocata la Partita *p* e in cui è stato venduto almeno un biglietto, è riportato il numero di biglietti venduti.
  4. *public String* ***bestTeam****()*. Il metodo restituisce il nome della squadra per cui sono stati venduti più biglietti. Nel caso di più squadre con lo stesso numero di biglietti venduti, restituisce la prima in base all’ordine lessicografico dei nomi delle squadre.

ESEMPIO - Si supponga che:

siano state giocate le seguenti partite

* { codice=”A1”, casa=”Juventus”, ospite=”Inter”, data=”d101” }
* { codice=”A2”, casa=”Juventus”, ospite=”Genoa” , data=”d102”}
* { codice=”B”, casa=”Cosenza”, ospite=”Cremonese”, data=”d101” }
* { codice=”C”, casa=”Rende”, ospite=”Reggina”, data=”d104” }

siano stati venduti i seguenti biglietti

* { num=100, codPartita=”A1”, posto=34, settore=”tribuna”, CF=”CF1” }
* { num=101, codPartita=”A1”, posto=12, settore=”curva”, CF=”CF2” }
* { num=102, codPartita=”A1”, posto=26, settore=”tribuna”, CF=”CF3” }
* { num=103, codPartita=”B”, posto=16, settore=”tribuna”, CF=”CF4” }
* { num=104, codPartita=”B”, posto=6, settore=”curva”, CF=”CF5” }
* { num=105, codPartita=”B”, posto=36, settore=”curva”, CF=”CF6” }
* { num=106, codPartita=”C”, posto=21, settore=”tribuna”, CF=”CF7” }
* { num=107, codPartita=”C”, posto=22, settore=”tribuna”, CF=”CF8” }
* { num=108, codPartita=”C”, posto=42, settore=”curva”, CF=”CF9” }
* { num=109, codPartita=”A2”, posto=34, settore=”tribuna”, CF=”CF1” }
* { num=110, codPartita=”A2”, posto=12, settore=”curva”, CF=”CF2” }
* { num=111, codPartita=”A2”, posto=26, settore=”tribuna”, CF=”CF3” }

le seguenti persone siano presenti nel sistema, avendo acquistato qualche biglietto:

* { CF=”CF1”, data=”d1”, citta=”Torino” }
* { CF=”CF2”, data=”d2”, citta=”Torino” }
* { CF=”CF3”, data=”d3”, citta=”Milano” }
* { CF=”CF4”, data=”d4”, citta=”Cosenza” }
* { CF=”CF5”, data=”d5”, citta=”Cosenza” }
* { CF=”CF6”, data=”d6”, citta=”Cosenza” }
* { CF=”CF7”, data=”d7”, citta=”Cosenza” }
* { CF=”CF8”, data=”d8”, citta=”Reggio Calabria” }
* { CF=”CF9”, data=”d9”, citta=”Reggio Calabria” }

Allora:

* Se *pa* è la partita con codice *”A1”, pe* è la persona con codice fiscale *”CF10”* e *settore=”curva”*, il metodo ***emettiBiglietto****(pa, pe, settore)* restituisce un biglietto con numero di serie *112* (in quanto l’ultimo biglietto emesso ha numero di serie *111*), numero di posto *33* (in quanto nel settore *curva* per la partita *A1* è stato occupato solo il posto *12*), codice partita *A1*, settore *curva*, codice fiscale *CF10*. Il biglietto è stato emesso in quanto la persona *CF10* non ha già acquistato altri biglietti per la partita *A1*, e anche perché c’è almeno un posto disponibile in *curva*.
* Se *pa* è la partita con codice *”A1”,* il metodo ***listaBiglietti****(pa)* restituisce la lista di biglietti identificata dai numeri <100,101,102> se assumiamo che la data di nascita del cliente (con codice fiscale) *CF1* sia inferiore a quelle del cliente *CF2*, a sua volta inferiore a quella del cliente *CF3*.
* Se *p* è la partita con codice *”B”*, il metodo ***statistica****(p)* restituisce la HashMap *{ (curva,2), (tribuna,1) }*, in quanto per la partita *B* sono stati venduti due biglietti di curva e uno di tribuna.
* Il metodo ***bestTeam****()* restituisce “Juventus” perché la Juventus è la squadra per cui sono stati venduti più biglietti. Ne sono stati infatti venduti 6 per la Juventus, e 3 per ogni altra squadra.