

Esercitazione - Classi (307)

Fondamenti 2 - Corso di Laurea in Informatica

Esercizio 1: Realizzare la classe Prenotazione che contenga almeno i seguenti campi:

- int numero;
- int posti;
- string codice; dove è possibile avere 10 codici di prenotazioni: “a0001”, “b0002”, “c0003”, “d0004”, “e0005”, “f0006”, “g0007”, “h0008”, “i0009”, “z0010”.

La scelta dei metodi da inserire nella classe Prenotazione è libera.

Realizzare la classe GestorePrenotazioni che permetta di gestire le prenotazioni.

La classe deve contenere al suo interno un campo di tipo `list<Prenotazione*` > e i seguenti metodi:

- `bool aggiungiPrenotazione(int posti, string codice)`; aggiunge una prenotazione alla lista di prenotazioni. Una prenotazione può essere aggiunta solo se c'è abbastanza posto per quella determinata prenotazione, dove il numero di posti disponibili è il seguente: 1 posto per il codice “a0001”, 2 posti per il codice “b0002”, 3 posti per il codice “c0003”, ecc. Se la prenotazione può essere aggiunta, il metodo crea una nuova prenotazione con un codice casuale e univoco, e restituisce `true`. Se la prenotazione non può essere aggiunta il metodo restituisce `false`.
- `bool rimuoviPrenotazione(int numero)`; rimuove una prenotazione dalla lista. Restituisce `true` se la prenotazione è presente, `false` altrimenti.
- `void stampaPrenotazioni(string codice) const`; stampa tutte le prenotazioni con quel codice.
- `void ordinaPrenotazioni()`; ordina le prenotazioni come segue:
 - Prima tutte le prenotazioni il cui codice inizia con la lettera a (non importa l'ordine).
 - Poi tutte le prenotazioni il cui codice inizia con la lettera b (non importa l'ordine).
 - Poi tutte le prenotazioni il cui codice inizia con la lettera c (non importa l'ordine).

- Infine, tutte le altre prenotazioni in ordine di numero.

Infine, realizzare un main di prova che permetta l'interazione con la classe GestorePrenotazioni. La scelta del menu è libera, ma deve essere possibile usare tutti i metodi della classe GestorePrenotazioni.

Esercizio 2: Realizzare una classe CodaPrioritaria che gestisca una coda di numeri. La classe deve contenere un campo privato `list<int>` e i seguenti metodi:

- `void aggiungi(int n)`; se `n` è 0 il numero non deve essere aggiunto.
- `int prossimoNumero() const`;
- `void rimuoviProssimoNumero()`;
- `int size() const`;

La coda ha il seguente ordine di priorità:

- i numeri positivi precedono i negativi
- tra i numeri positivi, i pari precedono i dispari
- tra i numeri negativi, i dispari precedono i pari
- a parità delle altre condizioni, il numero più piccolo (in valore assoluto) precede il più grande. Ad esempio, un possibile ordine è: 2 4 1 3 5 -1 -3 -6 -8

Esercizio 3: Realizzare la classe Partita che contenga i seguenti campi privati: `string squadra1`; `string squadra2`; `int golSquadra1`; `int golSquadra2`; `string arbitro`.

Realizzare la classe Mondiale contenente un campo privato `list<Partita>` e i seguenti metodi:

- `void aggiungi(Partita)`; aggiunge un elemento all'interno della lista.
- `int metodo1()`; si consideri come `M` la media dei gol segnati in tutte le partite (utilizzare la divisione intera). Si consideri come `P` la partita con meno gol segnati tra quelle con un numero di gol segnati $\geq M$ (nel caso in cui più partite soddisfino la condizione prendere la prima in ordine di apparizione nella lista). Il metodo restituisce i gol della partita `P`. Se non sono presenti partite, restituire -1.
- `int metodo2()`; restituisce il numero di arbitri che non hanno mai arbitrato più di una partita della stessa squadra. Se non sono presenti partite, restituire -1.
- `int metodo3()`; restituisce il numero di arbitri che hanno arbitrato solo partite finite con più di 2 gol di differenza. Se non sono presenti partite, restituire -1.

- `int metodo4()`; Data la seguente definizione: Una squadra S1 domina una squadra S2 se:
 - S1 vince una partita in trasferta contro S2 con almeno due gol di scarto, oppure
 - S1 domina S3 e S3 domina S2.

Il metodo restituisce il numero di squadre dominate dalla squadra1 della prima partita della lista. Se non sono presenti partite, restituire -1.

Note: Una squadra S domina sempre sé stessa.