

Cognome _____

Nome _____

Matricola _____

Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica - a.a. 2016/2017
21 Novembre 2017

Testo

Il database del sistema di gestione delle prenotazioni dei voli di una compagnia aerea è costituito da due vettori paralleli. Il primo è denominato "flights" e contiene oggetti di tipo "Volo" che rappresentano i voli presenti all'interno della banca dati della compagnia. Il secondo vettore è denominato "reservations" e contiene oggetti di tipo "Prenotazione" che rappresentano le informazioni relative ad ogni prenotazione effettuata relativa ad un determinato volo. Ad ogni volo può corrispondere più di una prenotazione, in quel caso le informazioni del volo saranno replicate. Per ogni volo presente nella posizione *i*-esima del vettore "flights", le informazioni relative ad una prenotazione si troveranno nella corrispondente posizione del vettore "reservations". Nel caso in cui il volo in posizione *i*-esima non abbia alcuna prenotazione associata, nella posizione corrispondente nel vettore "reservations" sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante "MAX_ELEM" (inizializzata a 1024). Se il numero dei voli contenuti nell'archivio è inferiore a "MAX_ELEM", i primi elementi del vettore conterranno gli oggetti di tipo "Volo", mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore "flights" si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Volo contiene le informazioni relative ad un volo:

```
public class Volo {  
  
    private String codice;           public int capienza;  
    public String data;             public String tragitto;  
  
    public Volo(String code, int capacity, String date, String travel) {  
        codice = code;             capacita = capacity;  
        data = date;               tragitto = travel;  
    }  
  
    public String getCodice() {  
        return codice;  
    }  
  
    public String toString() {  
        return ("#" + codice + ": " + data + " - " +  
            tragitto + " ( " + capienza + " )");  
    }  
}
```

La classe Prenotazione contiene le informazioni relative alle singole prenotazioni relative ad un determinato volo.

```
public class Prenotazione {  
  
    private static int numeroProgressivo = 0;  
    private int numero;               public String data;           public String passeggero;  
    public double prezzo;             public int posto;  
  
    public Prenotazione(String date, String passenger, double price, int sit) {  
        data = date;                 passeggero = passenger;  
        prezzo = price;              posto = sit;                   numero = ++numeroProgressivo;  
    }  
  
    public Prenotazione(int number, String date, String passenger, double price, int sit) {  
        data = date;                 passeggero = passenger;  
        prezzo = price;              posto = sit;                   numero = number;  
    }  
  
    public int getNum() { return numero; }  
  
    public void print() { System.out.println("(" + data + ") " + passeggero); }  
  
    public String toString() {  
        return ( "[" + numero + "]" + data + " - " + passeggero +  
            " - " + prezzo + " € (" + posto + ")");  
    }  
}
```

Si consiglia di procedere implementando un metodo e successivamente la parte del main che utilizza tale metodo.

Le varie operazioni devono essere eseguite sulla porzione significativa dell'archivio, cioè la porzione di "flights" che non contiene riferimenti *null*. Se si ha la necessità di convertire una stringa in intero, si può utilizzare la funzione di libreria *Integer.parseInt(s)* che converte la stringa *s* in un intero restituito come risultato.

A. Scrivere il metodo statico:

```
public static int contaVoliCapienza(Volo[] voli, int capienza)
```

Il metodo deve contare il numero totale dei voli che abbiano capienza superiore a quella passata come parametro.

B. Scrivere il metodo statico:

```
public static void ordinaVoliPrenotazioni(Volo[] voli, Prenotazione[] prenotazioni)
```

Il metodo deve ordinare, nell'array parallelo, gli elementi in maniera decrescente usando come criterio la data del volo ed eventualmente la data della prenotazione associata considerando le prenotazioni nulle aventi data di prenotazione inferiore tutte le altre prenotazioni nello stesso volo.

C. Scrivere il metodo statico:

```
public static double calcolaIncassoAR(Volo[] voli, Prenotazione[] prenotazioni)
```

Il metodo deve calcolare l'incasso totale delle sole prenotazioni di Andata e Ritorno (AR).

D. Scrivere il metodo statico:

```
public static boolean cancellaVoli(Volo[] voli, Prenotazione[] prenotazioni,
                                   String dest)
```

Il metodo deve eliminare dal database, specificato dai parametri "voli" e "prenotazioni", tutti i voli che hanno la destinazione uguale a quella passata come parametro; restituire **true** o **false** a seconda del fatto che sia stato eliminato almeno un volo e mantenere l'archivio in uno stato consistente.

E. Scrivere il metodo main che:

definisca ed inizializzi i vettori "flights" e "reservations" secondo i valori in tabella inizializzando a *null* o a zero i campi non presenti. La stampa dell'archivio consiste nello stampare le informazioni di ogni volo e delle relative prenotazioni (se ve ne sono). Si utilizzino correttamente i relativi metodi *toString()* implementati nelle due classi.

Codice	Data (gg/mm/aaaa)	Tragitto	Capacità	Nr.	Data (gg/mm/aaaa)	Passeggero	Prezzo (€)	Posto
PB160101	01/01/2016	Pisa-Barcellona	200	2	31/12/2015	Marco Rossi	59,99	75
PR170110	10/01/2017	Pisa-Roma	100					
PM170320	20/03/2017	Pisa-Milano	150	6	15/02/2017	Ugo Bianchi	199,99	40
MP170612	12/06/2017	Milano-Pisa	300	5	20/03/2017	Franco Verdi	70,5	150
MP170612	12/06/2017	Milano-Pisa	300	7	15/02/2017	Ugo Bianchi	141,00	23
PB160101	01/01/2016	Pisa-Barcellona	200	1	09/09/2015	Luca Neri	59,99	200
PM170320	20/03/2017	Pisa-Milano	150	3	20/03/2016	Carla Celeste	19,00	100
PM170320	20/03/2017	Pisa-Milano	150	4	20/03/2017	Franco Verdi	19,00	99

- Avvalendosi del metodo al punto A stampi a video il numero dei voli con capienza superiore a 100 posti.
- Ordini l'intero archivio utilizzando il metodo del punto B e stampi a video l'archivio prima e dopo l'ordinamento.
- Avvalendosi del metodo al punto C stampi a video l'incasso totale delle prenotazioni AR.
- Utilizzando il metodo al punto D, rimuova tutti i voli con destinazione Roma. Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.