

Cognome _____
Matricola _____

Nome _____
Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica - a.a. 2014
19 Settembre 2014

Testo

Il database di un bar è costituito da due vettori paralleli. Il primo è denominato “tables” e contiene oggetti di tipo “Tavolo” che rappresentano i tavoli presenti all'interno di un locale. Il secondo vettore è denominato “orders” e contiene oggetti di tipo “Ordine” che rappresentano le informazioni di ogni singolo ordine di un tavolo. Ad ogni tavolo può corrispondere più di un ordine, in quel caso le informazioni del tavolo saranno replicate.

Per ogni tavolo presente nella posizione *i*-esima del vettore “tables” si troveranno le informazioni relative ad un ordine nella corrispondente posizione del vettore “orders”. Nel caso che il tavolo in posizione *i*-esima non abbia alcun ordine associato, nella posizione corrispondente nel vettore “orders” sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante “MAX_ELEM” (inizializzata a 1024). Se il numero di tavoli contenuti nell’archivio è inferiore a “MAX_ELEM”, i primi elementi del vettore conterranno gli oggetti di tipo “Tavolo”, mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore “tables” si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Tavolo contiene le informazioni relative ad un tavolo:

```
public class Tavolo {
    private int id;
    public String nomeCliente;
    public String cognomeCliente;
    public String posizione;

    public Tavolo(int myId, String nome, String cognome, String posizione){
        this.id = myId;
        this.nomeCliente = nome;
        this.cognomeCliente = cognome;
        this.posizione = posizione;
    }

    public int getId(){
        return id;
    }

    public String toString(){
        return "Tav. #" + id + " (" + posizione + ") "
            + nomeCliente + " " + cognomeCliente;
    }
}
```

La classe Ordine contiene le informazioni relative ai singoli ordini di un tavolo.

```
public class Ordine {

    static private int numeroProgressivo = 0;
    private int numeroOrdine;

    public String prodotto;
    public int numero;
    public double prezzo;
    public boolean servito;
    public String orario;

    public Ordine(String product, int quantity, double price, String time, boolean served){
        numeroOrdine = numeroProgressivo++;
        prodotto = product;
        numero = quantity;
        prezzo = price;
        orario = time;
        servito = served;
    }
}
```

```

    public String toString(){
        return "[ord." + numeroOrdine + "] "
            + "num. " + numero + " " + prodotto;
    }
}

```

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 "tables" che non contiene riferimenti "null".

A) Scrivere il metodo statico:

```
public static String converti(String orario)
```

Il metodo riceve come parametro la Stringa orario interpretabile nelle 12h, ovvero nei due formati *hh:mm-AM* o *hh:mm-PM*. Tale stringa deve essere convertita nel formato 24h del tipo *hh:mm* e restituita come parametro di ritorno.

B) Scrivere il metodo statico:

```
public static void ordinaArrivo(Tavolo[] tavoli, Ordine[] ordini)
```

Il metodo deve ordinare, nel vettore "tavoli", gli elementi in maniera decrescente, usando come criterio il numero di consonanti che compongono il cognome del cliente. L'archivio deve poi essere lasciato in uno stato consistente.

C) Scrivere il metodo statico:

```
public static void stampaOrdini(Tavolo[] tavoli, Ordine[] ordini, String pos)
```

Il metodo deve stampare tutti gli ordini non nulli che sono stati effettuati per i tavoli che si trovano nella posizione indicata da *pos*. Per ogni ordine si dovrà stampare il nome del prodotto, l'orario ed il costo. Ogni stampa avrà quantità unitaria, ovvero se un certo prodotto è stato ordinato specificando una generica quantità N, si dovranno fare N stampe distinte.

D) Scrivere il metodo statico:

```
public static boolean dupOrdini(Tavolo[] tavoli, Ordine[] ordini, int tid1, int tid2)
```

Il metodo deve innanzitutto verificare la correttezza dei parametri di ingresso: se almeno uno tra *tid1* e *tid2* non esiste, il metodo deve restituire false e non effettuare operazioni. In caso contrario il metodo restituirà true e provvederà a duplicare gli ordini del tavolo con id *tid1* ed assegnarli (in aggiunta) al tavolo con id *tid2*. In dettaglio, il metodo provvederà a creare nuovi record nell'archivio, in numero uguale a quello degli ordini presenti per il tavolo individuato da *tid1*. Nei nuovi record il tavolo è individuato da *tid2* mentre gli ordini sono copie degli ordini presenti per il tavolo individuato da *tid1*. Per semplicità si supponga che è sempre possibile aggiungere nuovi record all'archivio.

E) Scrivere il metodo main che:

- definisca ed inizializzi i vettori "tables" e "orders" secondo i valori riportati in tabella. La stampa dell'archivio, ove richiesta, consiste nello stampare le informazioni di ogni tavolo e gli ordini associati (se ve ne sono). Si utilizzino correttamente i relativi metodi toString() implementati nelle due classi. Per ogni ordine si stampi anche l'orario.

Id	Nome e Cognome	Posizione	OrarioOrdine	Prodotto	Quantità	Costo	Servito
0	Mario Rossi	Ristopub	08:56-AM	Acqua	1	2	Sì
1	Pietro Rossi	Pizzeria					
0	Mario Rossi	Ristopub	09:05-PM	Birra	2	3.50	Sì
2	Gio' Verdini	Pizzeria	10:55-PM	Vino	2	10	Sì
2	Gio' Verdini	Pizzeria	09:30-PM	Acqua	3	2	Sì

- Avvalendosi del metodo al punto A stampi tutti gli ordini con l'informazione dell'orario convertito.
- Ordini l'intero archivio utilizzando il metodo del punto B e stampi a video l'archivio prima e dopo l'ordinamento.
- Utilizzando il metodo C, stampi le informazioni relative ai tavoli con posizione *Pizzeria*.
- Utilizzando il metodo del punto D, prende gli ordini del tavolo con id 0 e li duplichi nel tavolo con id 2. Ripeta l'operazione tentando di copiare gli ordini del tavolo con id 5 nel tavolo con id 2. Al termine di ogni operazione, se essa è avvenuta con successo si stampi l'archivio aggiornato o viceversa si stampi un messaggio di errore.