

Cognome _____

Nome _____

Matricola _____

Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica - a.a. 2018/2019
18 Settembre 2019

Testo

Il database del sistema di gestione delle officine di una compagnia di autoriparazioni è costituito da due vettori paralleli. Il primo è denominato “garages” e contiene oggetti di tipo “Officina” che rappresentano le officine presenti all'interno della banca dati della compagnia. Il secondo vettore è denominato “repairs” e contiene oggetti di tipo “Riparazione” che rappresentano le informazioni relative ad ogni riparazione effettuata o da effettuare in una determinata officina. Ad ogni officina può corrispondere più di una riparazione, in quel caso le informazioni dell’officina saranno replicate. Per ogni officina presente nella posizione *i*-esima del vettore “garages”, le informazioni relative ad una riparazione si troveranno nella corrispondente posizione del vettore “repairs”. Nel caso in cui l’officina in posizione *i*-esima non abbia alcuna riparazione associata, nella posizione corrispondente nel vettore “repairs” sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante “MAX_ELEM” (inizializzata a 1024). Se il numero delle officine contenute nell’archivio è inferiore a “MAX_ELEM”, i primi elementi del vettore conterranno gli oggetti di tipo “Officina”, mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore “garages” si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Officina contiene le informazioni relative ad una officina:

```
public class Officina {  
  
    private String codice;    public String citta;  
    public String indirizzo;  public String cap;  
  
    public Officina(String code, String city, String address, String zip) {  
        codice = code;        citta = city;  
        indirizzo = address;   cap = zip;  
    }  
  
    public String getCodice() {  
        return codice;  
    }  
  
    public String toString() {  
        return ("#" + codice + ": " + indirizzo + " - " + citta + " (" + cap + ")");  
    }  
}
```

La classe Riparazione contiene le informazioni relative alle riparazioni relative ad una officina.

```
public class Riparazione {  
  
    private static int numeroProgressivo = 0;  
    private int numero;                public String descrizione;        public double prezzo;  
    public String dataEntrata;          public String dataUscita;  
  
    public Riparazione(String description, double price, String entranceDate, String exitDate) {  
        descrizione = description;    prezzo = price;  
        dataEntrata = entranceDate;   dataUscita = exitDate;  
        numero = ++numeroProgressivo;  
    }  
  
    public Riparazione(int number, String description, double price, String entranceDate, String exitDate) {  
        descrizione = description;    prezzo = price;  
        dataEntrata = entranceDate;   dataUscita = exitDate;  
        numero = number;  
    }  
  
    public int getNum() {  
        return numero;  
    }  
  
    public String toString() {  
        String s = "[" + numero + "] " + descrizione + " (€ " + prezzo + ") " + ": " + dataEntrata;  
        if (dataUscita != null) { s += " - " + dataUscita; }  
        return s;  
    }  
}
```

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 "garages" che non contiene riferimenti *null*. Se si ha la necessità di convertire una stringa in un *long* oppure in un *int*, si possono utilizzare le funzioni di libreria *Long.parseLong(s)* ed *Integer.parseInt(s)* che convertono la stringa *s* rispettivamente in un *long* oppure in un *int* restituiti come risultato.

A. Scrivere il metodo statico:

```
public static int contaOfficine(Officina[] officine, Riparazione[] riparazioni)
```

Il metodo deve contare il numero totale di officine gestite che abbiano almeno una riparazione in corso d'opera.

B. Scrivere il metodo statico:

```
public static void ordinaDB(Officina[] officine, Riparazione[] riparazioni)
```

Il metodo deve ordinare, nell'array parallelo, gli elementi in maniera crescente usando come criterio il prezzo di una riparazione. Le officine senza riparazione associata vanno poste in fondo all'array.

C. Scrivere il metodo statico:

```
public static int contaGiorni(Officina[] officine, Riparazione[] riparazioni)
```

Il metodo deve contare il numero di giorni impiegati per effettuare la riparazione più veloce. Per semplicità si assuma che tutte le riparazioni siano state effettuate nello stesso anno solare.

D. Scrivere il metodo statico:

```
public static boolean rimuoviRiparazioni(Officina[] officine, Riparazione[] riparazioni, int anno)
```

Il metodo deve eliminare dal database tutte le riparazioni effettuate in un determinato anno passato come parametro; restituire **true** o **false** a seconda del fatto che sia stata eliminata almeno una riparazione e mantenere l'archivio in uno stato consistente evitando di eliminare definitivamente le officine senza riparazioni associate.

E. Scrivere il metodo main che:

definisca ed inizializzi i vettori "garages" e "repairs" 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 officina e delle relative riparazioni effettuate o da effettuare (se ve ne sono). Si utilizzino correttamente i relativi costruttori e metodi *toString()* implementati nelle due classi.

Codice Officina	Città	Indirizzo	CAP	Numero Riparazione	Descrizione	Prezzo (€)	Data Entrata (gg/mm/aaaa)	Data Uscita (gg/mm/aaaa)
PICA95	Pisa	Via Cattaneo, 95	56125	2	Marmitta	180,00	01/05/2018	10/05/2018
GEMO9	Genova	Via Morin, 9	16129					
FIBA3	Firenze	Via dei Bardi, 3	50125	6	Frizione	340,00	10/06/2018	18/06/2018
PIDI2	Pisa	Via Diotisalvi, 2	56122	5	Motore	710,00	05/06/2018	
PIDI2	Pisa	Via Diotisalvi, 2	56122	7	Frizione	280,00	11/06/2018	21/06/2018
PICA95	Pisa	Via Cattaneo, 95	56125	1	Semiassa	543,00	28/04/2018	31/06/2018
FIBA3	Firenze	Via dei Bardi, 3	50125	3	Trasmissione	311,00	18/05/2018	28/06/2018
FIBA3	Firenze	Via dei Bardi, 3	50125	4	Motore	60,00	01/06/2018	

- Avvalendosi del metodo al punto A, stampi a video il numero totale di officine gestite che abbiano almeno una riparazione in corso d'opera.
- Ordini l'intero archivio utilizzando il metodo al punto B e stampi a video l'archivio dopo l'ordinamento.
- Avvalendosi del metodo al punto C, stampi a video il numero di giorni impiegati per effettuare la riparazione più veloce.
- Avvalendosi del metodo al punto D, rimuova dall'archivio tutte le riparazioni effettuate nel 2018. Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.
- Si copi l'output della Console nel file soluzioni.txt nel progetto Eclipse.