

Cognome _____

Nome _____

Matricola _____

Postazione PC _____

Corso di Laurea in Ingegneria Gestionale

Esame di Informatica - a.a. 2015

6 Luglio 2015

Testo

Il database del sistema di gestione delle chiamate di uno Smart-Phone è costituito da due vettori paralleli. Il primo è denominato "contacts" e contiene oggetti di tipo "Contatto" che rappresentano i contatti presenti all'interno della rubrica dello Smart-Phone. Il secondo vettore è denominato "calls" e contiene oggetti di tipo "Chiamata" che rappresentano le informazioni relative ad ogni chiamata effettuata ad un determinato contatto. Ad ogni contatto può corrispondere più di una chiamata, in quel caso le informazioni del contatto saranno replicate.

Per ogni contatto presente nella posizione *i*-esima del vettore "contacts", le informazioni relative ad una chiamata si troveranno nella corrispondente posizione del vettore "orders". Nel caso in cui il contatto in posizione *i*-esima non abbia alcuna chiamata associata, nella posizione corrispondente nel vettore "calls" sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante "MAX_ELEM" (inizializzata a 1024). Se il numero di contatti contenuti nell'archivio è inferiore a "MAX_ELEM", i primi elementi del vettore conterranno gli oggetti di tipo "Contatto", mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore "contacts" si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Contatto contiene le informazioni relative ad un contatto:

```
public class Contatto {
    private static int idProgressivo = 0;
    private int id;
    public String nome;
    public String cognome;
    public String indirizzo;
    public String citta;
    public String cap;
    public String telefono;

    public Contatto(int myId, String name, String surname,
        String address, String city, String zipcode, String phone) {
        id = myId;    nome = name;    cognome = surname;    indirizzo = address;
        citta = city; cap = zipcode; telefono = phone;
    }

    public int getId() {
        return id;
    }

    public String toString() {
        return "# " + id + ": " + telefono + " - " + nome + " " + cognome + " - " +
            indirizzo + " - " + citta + " - " + cap;
    }
}
```

La classe Chamata contiene le informazioni relative alle singole chiamate ad un contatto.

```
public class Chiamata {
    private static int numeroProgressivo = 0;
    private int numero;
    boolean ricevuta;
    public String data;
    public String ora;
    public String durata;

    public Chiamata(boolean incoming, String date, String hour, String duration) {
        numero = numeroProgressivo++; ricevuta = incoming;
        data = date;    ora = hour;    durata = duration;
    }

    public int getNumero() {
        return numero;
    }

    public String toString() {
        return ((ricevuta)? "<< ": ">> ") + data + " - " + ora + " - " + durata;
    }
}
```

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 "contacts" che non contiene riferimenti "null".

A. Scrivere il metodo statico:

```
public static Contatto[] contattiDa(Contatto[] contatti, Chiamata[] chiamate,
                                    String city)
```

Il metodo deve restituire un nuovo array contenente solo i contatti di una città specificata dal parametro "city". Tale array non deve contenere duplicati e la dimensione deve essere la minima possibile. Se non vi sono contatti che soddisfano il criterio, il metodo deve restituire un riferimento a "null".

B. Scrivere il metodo statico:

```
public static void ordinaContatti(Contatto[] contatti, Chiamata[] chiamate)
```

Il metodo deve ordinare, nel vettore "contacts", gli elementi in maniera crescente, usando come criterio il CAP per elemento del contatto. Se il CAP è uguale vengono prima i contatti che hanno le chiamate nulle, poi quelli con chiamate ricevute ed infine quelli con chiamate effettuate. Il metodo deve mantenere la corrispondenza iniziale tra contatti e chiamate. 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.

C. Scrivere il metodo statico:

```
public static double costoChiamate(Contatto[] contatti, Chiamata[] chiamate)
```

Supponendo un piano tariffario di € 0,5 al minuto con scatto alla risposta pari ad € 0,1; il metodo deve calcolare il costo totale di tutte le chiamate effettuate (attributo ricevuta = false, R/E = E nella tabella) arrotondando i secondi non nulli al minuto successivo.

D. Scrivere il metodo statico:

```
public static boolean effettuaChiamata(Contatto[] contatti, Chiamata[] chiamate,
                                        String tel, String date, String hour, String duration)
```

Il metodo deve inserire nel database, specificato dai parametri "contatti" e "chiamate", una chiamata di una determinata durata ("duration"), effettuata ad una determinata data ("date") ed ora ("hour"), verso il primo contatto trovato, individuato per mezzo del numero di telefono ("tel"), che non abbia una chiamata associata oppure duplicando il contatto in caso contrario. Il metodo deve restituire "true" oppure "false" a seconda del fatto che sia stato trovato almeno un contatto associato al numero di telefono nel database. Se non esiste nessun contatto associato al numero di telefono ("tel"), l'inserimento non deve essere effettuato. L'inserimento deve mantenere l'archivio in uno stato consistente.

E. Scrivere il metodo main che:

definisca ed inizializzi i vettori "contacts" e "calls" secondo i valori in tabella. La stampa dell'archivio consiste nello stampare le informazioni di ogni contatto e le chiamate associate (se ve ne sono). Si utilizzino correttamente i relativi metodi toString() implementati nelle due classi.

Id	Nome e Cognome	Indirizzo	Città	Cap	Telefono	R/E	Data (gg/mm/aaaa)	Ora (hh:mm)	Durata (hh:mm:ss)
0	Mario Rossi	Piazza Cairoli, 3	Pisa	56124	050576904	R	01/01/2013	00:00	00:00:00
1	Pietro Rossi	Via Roma, 31	Pisa	56127	050642687				
2	Mario Ramarri	Via Lenin, 4	Roma	00149	0623476	R	03/02/2014	09:45	00:00:55
3	Giovanni Verdi	Via Pollione, 5	Chieti	66100	08712278	R	02/04/2014	10:21	00:01:23
3	Giovanni Verdi	Via Pollione, 5	Chieti	66100	08712278	E	07/07/2014	15:32	01:33:45
1	Pietro Rossi	Via Roma, 31	Pisa	56127	050642687	R	09/10/2014	17:04	00:00:00

- Avvalendosi del metodo al punto A, si ottenga l'array dei contatti di Pisa e lo si stampi a video.
- 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 il costo di tutte le chiamate effettuate.
- Avvalendosi del metodo al punto D. Si effettui la chiamata al numero "050642687", di durata pari a "01:23:45" in data "23/06/2015" alle "23:43". Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.