Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica - a.a. 2015
15 Giugno 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 chamata, 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 int contaChiamatePerse(Chiamata[] chiamate)
```

Il metodo deve contare il numero totale delle chiamate perse. Per chiamata persa si intende una chiamata ricevuta (attributo ricevuta = true, R/E = R nella tabella) con durata pari a "00:00:00".

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 la durata per elemento della chiamata corrispondente. Se la durata della chiamata è uguale vengono prima i contatti che hanno le chiamate ricevute. I contatti con chiamate nulle vanno in testa al vettore. 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 € 0,003 al secondo senza scatto alla risposta, il metodo deve calcolare il costo totale di tutte le chiamate effettuate (attributo ricevuta = false, R/E = E nella tabella).

D. Scrivere il metodo statico:

```
public static boolean rimuoviContatto(Contatto[] contatti, Chiamata[] chiamate,
                                                                    String tel)
```

Il metodo deve eliminare dal database, specificato dai parametri "contatti" e "chiamate", tutti i contatti e le relative chiamate corrispondenti ad un determinato numero di telefono; restituire true o false a seconda del fatto che sia stato eliminato almeno un contatto e 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, 32	Pisa	56127	050642687	R	09/10/2014	17:04	00:00:00
2	Mario Ramarri	Via Lenin, 4	Roma	00149	0623476	E	09/11/2014	07:45	00:00:00
2	Mario Ramarri	Via Lenin, 4	Roma	00149	0623476	E	25/12/2014	12:03	00:01:43

- Avvalendosi del metodo al punto A stampi a video il numero di chiamate perse.
- 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.
- Elimini il contatto con telefono 0623476, utilizzando il metodo del punto D. Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.

Soluzione:

```
public class Appello_20150615 {

	public static final int MAX_ELEM = 1024;

	public static int contaContatti(Contatto[] contatti){
		int count = 0;
		while (count < MAX_ELEM && contatti[count] != null) {
			count++;
		}
		return count;
	}

	/*
	 * A.
	 * Il metodo deve contare il numero totale delle chiamate perse.
	 * Per chiamata persa si intende una chiamata ricevuta
	 * (attributo ricevuta = true, R/E = R nella tabella) con durata pari a “00:00:00”.
	 */
	public static int contaChiamatePerse(Contatto[] contatti, Chiamata[] chiamate) {
		int count = 0;
		int n = contaContatti(contatti);
		for (int i=0; i<n; i++) {
			// Nota: E' possibile utilizzare anche il metodo convertiDurata().
			if (chiamate[i] != null && chiamate[i].ricevuta && chiama-
te[i].durata.equals("00:00:00")) {
				count++;
			}
		}
		return count;
	}

	// scambia l'elemento in posizione i-esima con quello in posizione j-esima
	private static void scambiaChiamate(Chiamata[] v, int i, int j){
		Chiamata tmp = v[i];
		v[i] = v[j];
		v[j] = tmp;
	}

	// scambia l'elemento in posizione i-esima con quello in posizione j-esima
	private static void scambiaContatti(Contatto[] v, int i, int j){
		Contatto tmp = v[i];
		v[i] = v[j];
		v[j] = tmp;
	}

	/* B.
	 * Il metodo deve ordinare, nel vettore “contacts”, gli elementi in maniera crescente,
	 * usando come criterio la durata per elemento della chiamata corrispondente.
	 * Se la durata della chiamata è uguale vengono prima i contatti che hanno le chiamate rice-
vute.
	 * I contatti con chiamate nulle vanno in testa al vettore.
	 * Il metodo deve mantenere la corrispondenza iniziale tra contatti e chiamate.
	 */
	public static void ordinaContatti(Contatto[] contatti, Chiamata[] chiamate){
		int n = contaContatti(contatti);
		for (int i = 0; i < n-1; i++) {
			for (int j=i+1; j < n; j++) {
				// Nota: E' possibile utilizzare anche il metodo convertiDurata().
				if (
					chiamate[j] == null ||
					chiamate[i] != null && chiama-
te[j].durata.compareToIgnoreCase(chiamate[i].durata) < 0 ||
					chiamate[i] != null && chiama-
te[j].durata.compareToIgnoreCase(chiamate[i].durata) == 0 &&
					chiamate[j].ricevuta == true
				)                                {
					scambiaContatti(contatti, i, j);
					scambiaChiamate(chiamate, i, j);
				}
			}
		}
	}

	public static int convertiDurata(String duration){
		int idxOra = duration.indexOf(':', 0);
		String ora = duration.substring(0, idxOra);

		int idxMinuti = duration.indexOf(':', idxOra + 1);
```

```
                String minuti = duration.substring(idxOra + 1, idxMinuti);

                String secondi = duration.substring(idxMinuti + 1);

                return Integer.parseInt(ora)*3600 + Integer.parseInt(minuti)*60 + Inte-
ger.parseInt(secondi);
        }

        /*
         * C.
         * Supponendo un piano tariffario di € 0.003 al secondo senza scatto alla risposta,
         * il metodo deve calcolare il costo totale di tutte le chiamate effettuate
         * (attributo ricevuta = false, R/E = E nella tabella).
         */
        public static double costoChiamate(Contatto[] contatti, Chiamata[] chiamate) {
                int n = contaContatti(contatti);
                double costo = 0.0;

                for (int i=0; i<n; i++) {
                        if (chiamate[i] != null && !chiamate[i].ricevuta) {
                                costo += 0.003 * convertiDurata(chiamate[i].durata);
                        }
                }
                return costo;
        }

        /*
         * D.
         * Il metodo deve eliminare dal database, specificato dai parametri “contatti” e “chiamate”,
         * tutti i contatti e le relative chiamate corrispondenti ad un determinato numero di tele-
fono;
         * restituire true o false a seconda del fatto che sia stato eliminato almeno un contatto e
         * mantenere l'archivio in uno stato consistente.
         */
        public static boolean rimuoviContatto(Contatto[] contatti, Chiamata[] chiamate, String tel)
{
                int n = contaContatti(contatti);
                int ordDel = 0;

                for(int i=0; i < n; i++) {
                        if(contatti[i].telefono.equals(tel)) {
                                contatti[i] = null;
                                chiamate[i] = null;
                                ordDel++;
                        }
                }

                if(ordDel > 0) {
                        // Nota: si è fatta almeno una elminazione percui bisogna fare una compatta-
zione
                        for(int i=0; i<n-1; i++) {
                                for(int j=i+1; j<n; j++) {
                                        if(contatti[i] == null && chiamate[j]!= null){
                                                scambiaContatti(contatti, i, j);
                                                scambiaChiamate(chiamate, i, j);
                                        }
                                }
                        }
                        return true;
                }
                else
                        return false;
        }

        public static void stampaDB(Contatto[] contatti, Chiamata[] chiamate){
                int n = contaContatti(contatti);
                for (int i=0; i < n; i++){
                        if (chiamate[i] != null){
                                System.out.println(contatti[i] + " " + chiamate[i]);
                        }
                        else {
                                System.out.println(contatti[i]);
                        }
                }
        }

        /*
         * E.
         */
        public static void main(String[] args) {
```

```
		Contatto[] contacts = new Contatto[MAX_ELEM];
		Chiamata[] calls = new Chiamata[MAX_ELEM];

		contacts[0] = new Contatto(0, "Mario", "Rossi", "Piazza Cairoli, 3", "Pisa", "56124",
"050576904");
		contacts[1] = new Contatto(1, "Pietro", "Rossi", "Via Roma, 31", "Pisa", "56127",
"050642687");
		contacts[2] = new Contatto(2, "Mario", "Ramarri", "Via Lenin, 4", "Roma", "00149",
"0623476");
		contacts[3] = new Contatto(3, "Giovanni", "Verdi", "Via Pollione, 4", "Chieti",
"66100", "08712278");
		contacts[4] = new Contatto(3, "Giovanni", "Verdi", "Via Pollione, 4", "Chieti",
"66100", "08712278");
		contacts[5] = new Contatto(1, "Pietro", "Rossi", "Via Roma, 31", "Pisa", "56127",
"050642687");
		contacts[6] = new Contatto(2, "Mario", "Ramarri", "Via Lenin, 4", "Roma", "00149",
"0623476");
		contacts[7] = new Contatto(2, "Mario", "Ramarri", "Via Lenin, 4", "Roma", "00149",
"0623476");

		calls[0] = new Chiamata(true, "01/01/2013", "00:00", "00:00:00");
		calls[2] = new Chiamata(true, "03/02/2014", "09:45", "00:00:55");
		calls[3] = new Chiamata(true, "02/04/2014", "10:21", "00:01:23");
		calls[4] = new Chiamata(false, "07/07/2014", "15:32", "01:33:45");
		calls[5] = new Chiamata(true, "09/10/2014", "17:04", "00:00:00");
		calls[6] = new Chiamata(false, "09/11/2014", "07:45", "00:00:00");
		calls[7] = new Chiamata(false, "25/12/2014", "12:03", "00:01:43");

		System.out.println("\nA.");
		System.out.print("Il numero totale delle chiamate perse è: ");
		System.out.println(contaChiamatePerse(contacts, calls));

		System.out.println("\nB.");
		System.out.println("\nDatabase PRIMA dell'ordinamento:");
		stampaDB(contacts, calls);
		ordinaContatti(contacts, calls);
		System.out.println("\nDatabase DOPO dell'ordinamento:");
		stampaDB(contacts, calls);

		System.out.println("\nC.");
		System.out.print("Il costo totale dei tutte le chiamate effettuate è: € ");
		System.out.println(costoChiamate(contacts, calls));

		System.out.println("\nD.");
		if (rimuoviContatto(contacts, calls, "0623476")) {
			System.out.println("Le chiamate rimanenti sono:");
			stampaDB(contacts, calls);
		}
		else{
			System.out.println("Nessuna eliminazione.");
		}
	}

}
```