Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica
a.a. 2012-13
22 febbraio 2013

Testo

Il database di un videonoleggio è costituito da due vettori paralleli. Il primo è denominato “videoteca” e contiene oggetti di tipo “Film” che rappresentano i film presenti nell’archivio. Il secondo vettore è denominato “noleggi” e contiene oggetti di tipo “Noleggio” che rappresentano le informazioni di noleggio di un film. Le informazioni di noleggio del film presente in una determinata posizione nel vettore “videoteca”, si trovano nella corrispondente posizione nel vettore “noleggi”. Nel caso un dato film non sia stato noleggiato, nella sua posizione nel vettore “noleggi” sarà presente un riferimento “null”. Entrambi i vettori hanno dimensione pari alla costante “MAX_FILM” (inizializzata a 256). Se il numero di film contenuti nell’archivio è inferiore a “MAX_FILM”, i primi elementi del vettore conterranno gli oggetti di tipo “Film”, mentre gli altri conterranno riferimenti “null”. Tutti gli elementi “null” del vettore “videoteca” si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Film contiene le informazioni relative ad un film ed un metodo per la stampa a video di queste informazioni:

```
public class Film {

    private int codice;
    public String titolo;
    public String regia;
    public int anno;
    public String[] cast;

    public Film(int co, String t, String r, int a, String[] ca) {
        codice = co;
        titolo = t;
        regia = r;
        anno = a;
        cast = ca;
    }

    public int getCodice() {
        return codice;
    }

    public void stampaInfo() {

        System.out.println("Film n° " + codice + " / " + titolo + " / " + anno
                + " / " + regia);

        if (cast != null && cast.length > 0) {
            System.out.print("\tCast:");
            for (int i = 0; i < cast.length - 1; i++) {
                System.out.print(" " + cast[i] + ",");
            }
            System.out.println(" " + cast[cast.length - 1]);
        }
    }

}
```

La classe Noleggio contiene il nome dell’utente che ha noleggiato un determinato film e la data di noleggio:

```
public class Noleggio {

    public String utente;
    public int giorno;
    public int mese;
    public int anno;

    public Noleggio(String u, int g, int m, int a) {
        utente = u;
        giorno = g;
        mese = m;
        anno = a;
    }
}
```

Si consiglia di procedere nel seguente modo: implementare un metodo e successivamente scrivere la parte del main che utilizza tale metodo, in modo da poterne verificare immediatamente la correttezza.

Le varie operazioni devono essere eseguite sulla porzione significativa dell'archivio, cioè la porzione di "videoteca" che non contiene riferimenti "null".

a) Scrivere il metodo statico:

```
public static int contaFilmAnno(Film[] filmdb, int a)
```

Il metodo deve ritornare il numero di film presenti nel vettore "filmdb" che siano stati girati in un anno successivo a quello specificato dal parametro "a".

b) Scrivere il metodo statico:

```
public static Film[] sostituisciRegia(Film[] filmdb, String oldReg,
        String newReg)
```

Il metodo deve cercare nel vettore "filmdb" i film diretti dal regista "oldReg" e sostituire il campo regia di tali film con "newReg". Il metodo deve inoltre ritornare un vettore contenente i film modificati.

c) Scrivere il metodo statico:

```
public static void ordinaCar(Film[] filmdb, Noleggio[] noldb, char ch)
```

Il metodo deve ordinare, in ordine crescente, il database specificato dai parametri "filmdb" e "noldb" a seconda della somma del numero di occorrenze del carattere "ch" nelle stringhe rappresentanti il titolo, il regista e gli attori del cast del film.

d) Scrivere il metodo statico:

```
public static boolean eliminaRegia(Film[] filmdb, Noleggio[] noldb,
        String reg)
```

Il metodo deve eliminare dal database specificato dai parametri "filmdb" e "noldb" tutti i film del regista "reg", a patto che non siano noleggiati. Se almeno uno dei film è noleggiato, il metodo non deve eliminare nessun film. Il valore di ritorno deve essere true se l'eliminazione avviene con successo, oppure false in caso almeno un film sia noleggiato.

e) Scrivere il metodo main che:

- definisca ed inizializzi i vettori "videoteca" e "noleggi" secondo i valori riportati in tabella e stampi a video l'archivio. La stampa dell'archivio consiste nella stampa delle informazioni di ogni film (usando il metodo "stampaInfo" della classe "Film") e, se il film è noleggiato, deve essere indicato anche l'utente e la data di noleggio.

Codice	Titolo	Regia	Anno	Cast	Utente	Data
0	Jackie Brown	Quentin Tarantino	1997	Pam Grier Samuel L. Jackson		
1	Pulp Fiction	Quentin Tarantino	1994	John Travolta Samuel L. Jackson		
2	Cloud Atlas	Larry e Andy Wachowski	2012	Tom Hanks Hugo Weaving	Gianni Verdi	01/02/2013
3	Matrix	Larry e Andy Wachowski	1999	Keanu Reeves Hugo Weaving	Mario Rossi	30/01/2013

- stampi il numero di film girati dopo il 1995 usando il metodo "contaFilmAnno".
- sostituisca il campo regia dei film diretti da "Larry e Andy Wachowski" con "Lana e Andy Wachowski" usando il metodo "sostituisciRegia", stampi il titolo dei film modificati e stampi l'archivio aggiornato.
- ordini l'archivio a seconda del numero di occorrenze del carattere "o" utilizzando il metodo "ordinaCar" e stampi a video l'archivio ordinato.
- elimini i film di Quentin Tarantino dall'archivio utilizzando il metodo "eliminaRegia" e stampi a video l'esito e, in caso di esito positivo, l'archivio aggiornato.

Soluzione

```
/*
 * Nome
 * Cognome
 * Matricola
 * Postazione
 */

public class Esame {

	public static final int MAX_FILM = 256;

	/*
	 * Questo metodo conta i film presenti nell'archivio. Non è richiesto dal
	 * testo, ma semplifica la scrittura degli altri metodi.
	 */
	public static int contaFilm(Film[] filmdb) {
		int nFilm = 0;
		while (nFilm < MAX_FILM && filmdb[nFilm] != null) {
			nFilm++;
		}
		return nFilm;
	}

	/*
	 * Questo metodo stampa le informazioni relative a tutti i film presenti
	 * nell'archivio e, per i film noleggiati, indica l'utente che li ha
	 * noleggiati e la data di noleggio. Non è richiesto dal testo, ma
	 * semplifica la scrittura del main.
	 */
	public static void stampaArchivio(Film[] filmdb, Noleggio[] noldb) {
		/* Conteggio dei film presenti nell'archivio */
		int nFilm = contaFilm(filmdb);

		for (int i = 0; i < nFilm; i++) {
			filmdb[i].stampaInfo();
			if (noldb[i] != null) {
				System.out.println("\tNoleggiato a " + noldb[i].utente + " il "
						+ noldb[i].giorno + "/" + noldb[i].mese + "/"
						+ noldb[i].anno);
			}
```

```
		}
}

public static int contaFilmAnno(Film[] filmdb, int a) {
		int nFilm = contaFilm(filmdb);

		int conta = 0;
		for (int i = 0; i < nFilm; i++) {
			if (filmdb[i].anno >= a) {
				conta++;
			}
		}

		return conta;
}

public static Film[] sostituisciRegia(Film[] filmdb, String oldReg,
			String newReg) {
		int nFilm = contaFilm(filmdb);

		int nRis = 0;
		for (int i = 0; i < nFilm; i++) {
			if (filmdb[i].regia.equals(oldReg)) {
				nRis++;
			}
		}

		Film[] ris = new Film[nRis];

		int k = 0;
		for (int i = 0; i < nFilm; i++) {
			if (filmdb[i].regia.equals(oldReg)) {
				filmdb[i].regia = newReg;
				ris[k] = filmdb[i];
				k++;
			}
		}

		return ris;
}
```

```
public static int contaOccorrenze(Film f, char ch) {
	int occ = 0;

	for (int j = 0; j < f.titolo.length(); j++) {
		if (f.titolo.charAt(j) == ch) {
			occ++;
		}
	}

	for (int j = 0; j < f.regia.length(); j++) {
		if (f.regia.charAt(j) == ch) {
			occ++;
		}
	}

	for (int i = 0; i < f.cast.length; i++) {
		for (int j = 0; j < f.cast[i].length(); j++) {
			if (f.cast[i].charAt(j) == ch) {
				occ++;
			}
		}
	}
	return occ;
}

public static void ordinaCar(Film[] filmdb, Noleggio[] noldb, char ch) {
	/* Conteggio dei film presenti nell'archivio */
	int nFilm = contaFilm(filmdb);
	/* Ordinamento */
	for (int i = 0; i < nFilm - 1; i++) {
		for (int j = i + 1; j < nFilm; j++) {
			if (contaOccorrenze(filmdb[i], ch) > contaOccorrenze(filmdb[j],
					ch)) {
				Film tmp = filmdb[i];
				filmdb[i] = filmdb[j];
				filmdb[j] = tmp;
				Noleggio tmp2 = noldb[i];
				noldb[i] = noldb[j];
				noldb[j] = tmp2;
			}
		}
```

```
        }
}

// soluzione "bella" anche se non efficiente
public static boolean eliminaRegia(Film[] filmdb, Noleggio[] noldb,
                String reg) {
        int nFilm = contaFilm(filmdb);
        boolean noleggiato = false;
        for (int i = 0; i < nFilm; i++) {
                if (filmdb[i].regia.equals(reg) && noldb[i] != null) {
                        noleggiato = true;
                }
        }
        if (!noleggiato) {
                for (int i = 0; i < nFilm; i++) {
                        if (filmdb[i].regia.equals(reg)) {
                                filmdb[i] = null;
                        }
                }

                for (int i = 0; i < nFilm - 1; i++) {
                        for (int j = 0; j < nFilm; j++) {
                                if(filmdb[i] == null && filmdb[j] != null) {
                                        Film tmp = filmdb[i];
                                        filmdb[i] = filmdb[j];
                                        filmdb[j] = tmp;
                                        Noleggio tmp2 = noldb[i];
                                        noldb[i] = noldb[j];
                                        noldb[j] = tmp2;
                                }
                        }
                }
        }
        return !noleggiato;
}

public static void main(String[] args) {
        Film videoteca[] = new Film[MAX_FILM];
        Noleggio noleggi[] = new Noleggio[MAX_FILM];

        videoteca[0] = new Film(0, "Jackie Brown", "Quentin Tarantino", 1997,
```

```
                    new String[] { "Pam Grier", "Samuel L. Jackson" });

videoteca[1] = new Film(1, "Pulp Fiction", "Quentin Tarantino", 1994,
                    new String[] { "John Travolta", "Samuel L. Jackson" });

videoteca[2] = new Film(2, "Cloud Atlas", "Larry e Andy Wachowski",
                    2012, new String[] { "Tom Hanks", "Hugo Weaving" });

videoteca[3] = new Film(3, "Matrix", "Larry e Andy Wachowski", 1999,
                    new String[] { "Keanu Reeves", "Hugo Weaving" });

noleggi[2] = new Noleggio("Mario Rossi", 1, 2, 2013);
noleggi[3] = new Noleggio("Gianni Verdi", 30, 1, 2013);

System.out.println("Archivio film:");
stampaArchivio(videoteca, noleggi);

System.out.println();

System.out.println("Nell'archivio sono presenti "
                    + contaFilmAnno(videoteca, 1995) + " girati dopo il 1995");

System.out.println();

Film[] ris = sostituisciRegia(videoteca, "Larry e Andy Wachowski",
                    "Lana e Andy Wachowski");

System.out.println("Sono stati aggiornati  i seguenti film:");
for (int i = 0; i < ris.length; i++) {
            System.out.println(ris[i].titolo);
}
System.out.println("Archivio aggiornato:");
stampaArchivio(videoteca, noleggi);

System.out.println();

ordinaCar(videoteca, noleggi, 'o');
System.out.println("Archivio ordinato:");
stampaArchivio(videoteca, noleggi);

System.out.println();
```

```
            if (eliminaRegia(videoteca, noleggi, "Quentin Tarantino")) {
                System.out.println("Film eliminati correttamente.");
                System.out.println("Archivio aggiornato: ");
                stampaArchivio(videoteca, noleggi);
            } else {
                System.out.println("Impossibile eliminare i film.");
            }

        }
}
```