

Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica
a.a. 2011-12
17 settembre 2012

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 regista;  
    public int anno;  
    public String[] cast;  
  
    public Film(int co, String t, String r, int a, String[] ca) {  
        codice = co;  
        titolo = t;  
        regista = r;  
        anno = a;  
        cast = ca;  
    }  
  
    public int getCodice() {  
        return codice;  
    }  
  
    public void stampaInfo() {  
        System.out.println("Film n° " + codice + " / " + titolo + " / " + anno  
            + " / " + regista);  
  
        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 Film[] trovaNonAtt(Film[] filmdb, String att)
```

Il metodo deve ritornare un vettore di Film contenente i film del vettore "filmdb" che non hanno l'attore att tra i membri del cast. Nel caso non ci siano film che soddisfino questi criteri, il metodo deve ritornare null.

b) Scrivere il metodo statico:

```
public static int restituiscePrima(Film[] filmdb, Noleggio[] noleggi, int g, int m, int a)
```

Il metodo deve rendere di nuovo disponibili per il noleggio tutti i film del database specificato dai parametri "filmdb" e "noleggi" che siano stati noleggiati prima della data indicata dai parametri "g", "m" ed "a" (rispettivamente: giorno, mese ed anno). Il metodo deve ritornare il numero di film restituiti.

c) Scrivere il metodo statico:

```
public static void ordinaCast(Film[] filmdb, Noleggio[] noleggi)
```

Il metodo deve ordinare, in ordine crescente, il database specificato dai parametri "filmdb" e "noleggi" a seconda della somma delle lunghezze delle stringhe rappresentanti gli attori del cast.

d) Scrivere il metodo statico:

```
public static boolean eliminaVecchio(Film[] filmdb, Noleggio[] noleggi)
```

Il metodo deve eliminare dal database specificato dai parametri "filmdb" e "noleggi" il film con anno minore. Il valore di ritorno deve essere true se l'eliminazione avviene con successo, oppure false se il database è vuoto e quindi non è possibile eliminare alcun film.

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	Bianco, rosso e Verdone	Carlo Verdone	1981	Carlo Verdone Irina Sanpiter	Mario Rossi	15/09/2012
1	Il dittatore dello stato libero di Bananas	Woody Allen	1971	Woody Allen Louise Lasser		
3	7 chili in 7 giorni	Luca Verdone	1986	Renato Pozzetto Carlo Verdone	Mario Rossi	13/09/2012
2	Grandi magazzini	Franco Castellano	1986	Lino Banfi Renato Pozzetto	Gianni Verdi	11/09/2012

- stampi i titoli dei film che non hanno "Carlo Verdone" tra gli attori del cast, utilizzando il metodo "trovaNonAtt".
- utilizzi il metodo "restituiscePrima" per rendere disponibili per il noleggio i film noleggiati prima del 14/09/2012 e stampi il numero di film restituiti e l'archivio aggiornato.
- ordini l'archivio utilizzando il metodo "ordinaCast" e stampi a video l'archivio ordinato.
- elimini il film più vecchio dall'archivio utilizzando il metodo "eliminaVecchio" 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);
```

```
            }
```

```

    }
}

/*
 * Questo metodo confronta due date, ciascuna espressa da tre interi che
 * indicano giorno, mese ed anno. Ritorna un valore: positivo se la prima
 * data è maggiore della seconda; negativo se la prima data è minore della
 * seconda; zero se le date sono uguali. Non è richiesto dal testo, ma
 * semplifica la scrittura dei metodi.
 */
public static int confrontaData(int g1, int m1, int a1, int g2, int m2,
                               int a2) {
    /*
     * Prima si confrontano gli anni. La differenza tra gli anni (qualora
     * sia diversa da 0) delle due date costituisce anche il valore di
     * ritorno, perché se è positiva allora la prima data è maggiore, mentre
     * se negativa la prima data è minore.
     */
    int ret = a1 - a2;
    /*
     * Nel caso in cui l'anno sia lo stesso, allora il confronto avviene sul
     * mese in maniera simile a quanto già visto.
     */
    if (ret == 0) {
        ret = m1 - m2;
        /*
         * Nel caso in cui anche il mese sia lo stesso, allora il confronto
         * avviene sul giorno.
         */
        if (ret == 0) {
            ret = g1 - g2;
        }
    }
    return ret;
}

public static Film[] trovaNonAtt(Film[] filmdb, String att) {
    int nFilm = contaFilm(filmdb);

    int nRis = 0;
    for (int i = 0; i < nFilm; i++) {

```

```

        boolean trovato = false;
        for (int j = 0; j < filmdb[i].cast.length && !trovato; j++) {
            if (filmdb[i].cast[j].equals(att)) {
                trovato = true;
            }
        }
        if (!trovato) {
            nRis++;
        }
    }

    Film[] ris = null;

    if (nRis > 0) {
        ris = new Film[nRis];
    }

    int k = 0;

    for (int i = 0; i < nFilm; i++) {
        boolean trovato = false;
        for (int j = 0; j < filmdb[i].cast.length && !trovato; j++) {
            if (filmdb[i].cast[j].equals(att)) {
                trovato = true;
            }
        }
        if (!trovato) {
            ris[k] = filmdb[i];
            k++;
        }
    }

    return ris;
}

```

```

public static int restituisciPrima(Film[] filmdb, Noleggio[] noldb, int g,
    int m, int a) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);
    int nRis = 0;
    for (int i = 0; i < nFilm; i++) {

```

```

        if (nolddb[i] != null
            && confrontaData(nolddb[i].giorno, nolddb[i].mese,
                            nolddb[i].anno, g, m, a) < 0) {
                nRis++;
                nolddb[i] = null;
            }
    }
    return nRis;
}

```

```

public static void ordinaCast(Film[] filmdb, Noleggio[] nolddb) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);
    /* Ordinamento */
    for (int i = 0; i < nFilm - 1; i++) {
        int somma_i = 0;
        for (int k = 0; k < filmdb[i].cast.length; k++) {
            somma_i += filmdb[i].cast[k].length();
        }

        for (int j = i + 1; j < nFilm; j++) {
            int somma_j = 0;
            for (int k = 0; k < filmdb[j].cast.length; k++) {
                somma_j += filmdb[j].cast[k].length();
            }

            if (somma_i > somma_j) {
                Film tmp = filmdb[i];
                filmdb[i] = filmdb[j];
                filmdb[j] = tmp;
                Noleggio tmp2 = nolddb[i];
                nolddb[i] = nolddb[j];
                nolddb[j] = tmp2;

                somma_i = somma_j;
            }
        }
    }
}

```

```

public static boolean eliminaVecchio(Film[] filmdb, Noleggio[] nolddb) {

```

```

int nFilm = contaFilm(filmdb);
boolean ret = false;
if (nFilm > 0) {
    int vecchio = 0;
    for (int i = 1; i < nFilm; i++) {
        if (filmdb[i].anno < filmdb[vecchio].anno) {
            vecchio = i;
        }
    }
    filmdb[vecchio] = filmdb[nFilm - 1];
    noldb[vecchio] = noldb[nFilm - 1];
    filmdb[nFilm - 1] = null;
    noldb[nFilm - 1] = null;
    ret = true;
}
return ret;
}

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

    videoteca[0] = new Film(0, "Bianco, rosso e Verdone", "Carlo Verdone",
        1981, new String[] { "Carlo Verdone", "Irina Sanpiter" });

    videoteca[1] = new Film(1,
        "Il dittatore dello stato libero di Bananas", "Woody Allen",
        1971, new String[] { "Woody Allen", "Louise Lasser" });

    videoteca[2] = new Film(3, "7 chili in 7 giorni", "Luca Verdone", 1986,
        new String[] { "Renato Pozzetto", "Carlo Verdone" });

    videoteca[3] = new Film(2, "Grandi magazzini", "Franco Castellano",
        1986, new String[] { "Lino Banfi", "Renato Pozzetto" });

    noleggi[0] = new Noleggio("Mario Rossi", 15, 9, 2012);
    noleggi[2] = new Noleggio("Mario Rossi", 13, 9, 2012);
    noleggi[3] = new Noleggio("Gianni Verdi", 11, 9, 2012);

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

```

```
System.out.println();
```

```
Film[] ris = trovaNonAtt(videoteca, "Carlo Verdone");
```

```
if (ris != null) {
```

```
    System.out
```

```
        .println("I film il cui non è presente Carlo Verdone sono:");
```

```
    for (int i = 0; i < ris.length; i++) {
```

```
        System.out.println(ris[i].titolo);
```

```
    }
```

```
} else {
```

```
    System.out.println("In tutti i film è presente Carlo Verdone");
```

```
}
```

```
System.out.println();
```

```
ordinaCast(videoteca, noleggi);
```

```
System.out.println("Archivio ordinato:");
```

```
stampaArchivio(videoteca, noleggi);
```

```
System.out.println();
```

```
int ris2 = restituisciPrima(videoteca, noleggi, 14, 9, 2012);
```

```
System.out.println("Numero di libri restituiti: " + ris2);
```

```
System.out.println("Archivio aggiornato: ");
```

```
stampaArchivio(videoteca, noleggi);
```

```
System.out.println();
```

```
if (eliminaVecchio(videoteca, noleggi)) {
```

```
    System.out.println("Film eliminato correttamente ");
```

```
    System.out.println("Archivio aggiornato: ");
```

```
    stampaArchivio(videoteca, noleggi);
```

```
} else {
```

```
    System.out.println("Archivio vuoto!");
```

```
}
```

```
}
```

```
}
```