

Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica
a.a. 2010-11
13 settembre 2011

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 {  
    public 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 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 boolean ordinato(Film[] filmdb)
```

che prende in ingresso il vettore "filmdb" che rappresenta l'archivio della videoteca e controlla se esso è ordinato per anno, in senso crescente.

b) Scrivere il metodo statico:

```
public static Film[] noleggiatiPrima(Film[] filmdb, Noleggio[] noldb,
    int g, int m, int a)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio ed una data rappresentata dai tre interi g (giorno), m (mese) ed a (anno). Il metodo deve ritornare un vettore di Film contenente tutti i film noleggiati prima della data indicata.

c) Scrivere il metodo statico:

```
public static String regista(Film[] filmdb, String t)
```

che prende in ingresso il vettore "filmdb" che rappresenta l'archivio della videoteca e la stringa t, rappresentante il titolo di un film. Il metodo deve ritornare una stringa rappresentante il regista del film avente il titolo indicato oppure "null" nel caso che il film non sia presente nell'archivio.

d) Scrivere il metodo statico:

```
public static int inserisciOrdinato(Film[] filmdb, Noleggio noldb[],
    int co, String t, String r, int a, String[] ca)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio e le informazioni descrittive di un film: codice (co), titolo (t), regia (r), anno (a) e cast (ca). L'archivio "filmdb" passato al metodo deve essere già ordinato per anno crescente. Il metodo crea un nuovo film a partire dalle informazioni fornite e lo inserisce nell'archivio in modo tale da preservarne l'ordinamento. Prima di effettuare l'inserimento, il metodo deve controllare che l'archivio non abbia raggiunto la dimensione massima consentita e che sia effettivamente ordinato. Il metodo ritorna un intero che vale 0 se l'inserimento avviene correttamente, -1 nel caso l'archivio non sia ordinato o -2 nel caso l'archivio sia pieno.

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. Nei film in cui non è presente cast, tale campo va impostato a "null". 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	Nausicaä della Valle del vento	Hayao Miyazaki	1984			
1	Clerks – Commessi	Kevin Smith	1994	Brian O'Halloran Jeff Anderson	Gianni Verdi	9/9/2011
2	Jay & Silent Bob... Fermate Hollywood!	Kevin Smith	2001	Jason Mewes Kevin Smith	Mario Rossi	29/8/2011
3	Ponyo sulla scogliera	Hayao Miyazaki	2008		Mario Rossi	2/9/2011

- controlli se il vettore "videoteca" sia ordinato per anno usando il metodo "ordinato" e stampi l'esito del metodo.
- stampi i titoli del film noleggiati prima del 4/9/2011, usando il metodo "noleggiatiPrima".
- stampi il regista del film dal titolo "Clerks – Commessi", usando il metodo "regista".
- aggiunga all'archivio, utilizzando il metodo "inserisciOrdinato", il film del 1997 "Principessa Mononoke", diretto da "Hayao Miyazaki" ed avente codice "4", poi stampi l'esito del metodo e l'archivio in caso di successo.

Soluzione

```
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[] nolddb) {  
        /* Conteggio dei film presenti nell'archivio */  
        int nFilm = contaFilm(filmdb);  
  
        for (int i = 0; i < nFilm; i++) {  
            filmdb[i].stampaInfo();  
            if (nolddb[i] != null) {  
                System.out.println("\tNoleggiato a " + nolddb[i].utente + " il "  
                    + nolddb[i].giorno + "/" + nolddb[i].mese + "/"  
                    + nolddb[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 boolean ordinato(Film[] filmdb) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);
    /* La variabile ord contiene l'esito del controllo */
    boolean ord = true;
    /*
     * Per controllare che l'archivio sia effettivamente ordinato, basta
     * assicurarsi che ogni elemento non sia minore di quello che lo
     * precede.
     */
    for (int i = 1; i < nFilm && ord; i++) {
        if (filmdb[i].anno < filmdb[i - 1].anno) {
            ord = false;
        }
    }
    return ord;
}

public static Film[] noleggiatiPrima(Film[] filmdb, Noleggio[] noldb,
    int g, int m, int a) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);
    int nRis = 0;
    /* Conteggio degli elementi del vettore risultato */
    for (int i = 0; i < nFilm; i++) {
        if (noldb[i] != null
            && confrontaData(noldb[i].giorno, noldb[i].mese,
                noldb[i].anno, g, m, a) < 0) {
            nRis++;
        }
    }
    /* Creazione del vettore risultato */
    Film[] ris = new Film[nRis];
    /* Popolamento del vettore risultato */
    int k = 0;
    for (int i = 0; i < nFilm; i++) {
        if (noldb[i] != null
            && confrontaData(noldb[i].giorno, noldb[i].mese,
                noldb[i].anno, g, m, a) < 0) {
            ris[k] = filmdb[i];
            k++;
        }
    }
    return ris;
}

public static String regista(Film[] filmdb, String t) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);

    /* Ricerca del film desiderato */
    String ris = null;
    for (int i = 0; i < nFilm && ris == null; i++) {
        if (filmdb[i].titolo.equals(t)) {
            ris = filmdb[i].regia;
        }
    }

    return ris;
}

public static int inserisciOrdinato(Film[] filmdb, Noleggio noldb[],
    int co, String t, String r, int a, String[] ca) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);
    /* Variabile per il valore di ritorno */
    int ret;
    /* Controllo dell'ordinamento dell'archivio */
    if (!ordinato(filmdb)) {

```

```

    ret = -1;
} else if (nFilm == MAX_FILM) {
    ret = -2;
} else {
    /* Ricerca della posizione in cui inserire il film */
    /*
    * La variabile pos conterrà la posizione in cui inserire il film.
    * Viene inizializzata con il valore massimo (nFilm, inserimento in
    * fondo), poi si cerca nell'array la corretta posizione dove
    * inserire il film. Tale posizione corrisponde a quella del primo
    * film con anno superiore ad "a". È importante assicurarsi di
    * terminare il ciclo una volta trovata la posizione giusta. Questo
    * è garantito dalla condizione di guardia (pos == nFilm) che fa sì
    * che appena viene cambiato il valore della variabile pos, il
    * ciclo termini. Se non viene trovato nessun film in archivio con
    * anno superiore ad "a", il nuovo film dovrà essere inserito in
    * fondo, ciò è già assicurato dal fatto che la variabile viene
    * inizializzata con il valore nFilm.
    */
    int pos = nFilm;
    for (int i = 0; i < nFilm && pos == nFilm; i++) {
        if (filmdb[i].anno > a) {
            pos = i;
        }
    }
    ret = 0;
    /*
    * I film dalla posizione pos in poi devono essere spostati in
    * avanti di una posizione, prima di poter effettuare
    * l'inserimento. È importante notare che si parte dal fondo, in
    * modo tale da evitare di sovrascrivere i film successivi.
    */
    for (int i = nFilm; i > pos; i--) {
        filmdb[i] = filmdb[i - 1];
        nolddb[i] = nolddb[i - 1];
    }
    /*
    * Il film viene inserito e le sue informazioni di noleggio vengono
    * azzerate.
    */
    filmdb[pos] = new Film(co, t, r, a, ca);
    nolddb[pos] = null;
}
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, "Nausicaä della Valle del vento",
        "Hayao Miyazaki", 1984, null);

    videoteca[1] = new Film(1, "Clerks - Commessi", "Kevin Smith", 1994,
        new String[] { "Brian O'Halloran", "Jeff Anderson" });

    videoteca[2] = new Film(2, "Jay & Silent Bob... Fermate Hollywood!",
        "Kevin Smith", 2001, new String[] { "Jason Mewes",
        "Kevin Smith" });

    videoteca[3] = new Film(3, "Ponyo sulla scogliera", "Hayao Miyazaki",
        2008, null);

    noleggi[1] = new Noleggio("Gianni Verdi", 9, 9, 2011);
    noleggi[2] = new Noleggio("Mario Rossi", 29, 8, 2011);
    noleggi[3] = new Noleggio("Mario Rossi", 2, 9, 2011);

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

    boolean ris = ordinato(videoteca);
}

```

```

System.out.println();
if (ris) {
    System.out.println("L'archivio è ordinato");
} else {
    System.out.println("L'archivio non è ordinato");
}

Film[] ris2 = noleggiatiPrima(videoteca, noleggi, 4, 9, 2011);
System.out.println();
System.out.println("Film noleggiati prima del 4/9/2011: ");
for (int i = 0; i < ris2.length; i++) {
    System.out.println(ris2[i].titolo);
}

String ris3 = regista(videoteca, "Clerks - Commessi");
System.out.println();
if (ris3 == null) {
    System.out.println("Il film non è presente in archivio");
} else {
    System.out.println("Il regista di \"Clerks - Commessi\" è " +
ris3);
}

int ris4 = inserisciOrdinato(videoteca, noleggi, 4,
    "Principessa Mononoke", "Hayao Miyazaki", 1997, null);
System.out.println();
if (ris4 == -2) {
    System.out.println("L'archivio è pieno");
} else if (ris4 == -1) {
    System.out.println("L'archivio non è ordinato");
} else if (ris4 == 0) {
    System.out.println("Film inserito con successo.");
    System.out.println("Archivio dopo l'inserimento:");
    stampaArchivio(videoteca, noleggi);
}
}
}
}

```