

Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica
a.a. 2010-11
24 giugno 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 long codice;  
    public String titolo;  
    public String regia;  
    public int anno;  
    public String[] cast;  
  
    public Film(long 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 void ordinaAnno(Film[] filmdb, Noleggio[] noldb)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio e li ordina secondo il campo "anno" della classe Film.

b) Scrivere il metodo statico:

```
public static Film[] trovaFilm(Film[] filmdb, String reg)
```

che prende in ingresso il vettore "filmdb" che rappresenta l'archivio della videoteca e la stringa "reg" e ritorna un vettore di Film contenente soltanto i film che hanno il regista specificato nella stringa "reg".

c) Scrivere il metodo statico:

```
public static int contaDisponibili(Film[] filmdb, Noleggio[] noldb)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio e ritorna il numero di film che non sono attualmente noleggiati.

d) Scrivere il metodo statico:

```
public static int noleggiaFilm(Film[] filmdb, Noleggio[] noldb, long cod, String ut, int g, int m, int a)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio, il codice di un film "cod", l'utente "ut" ed una data espressa da tre interi rappresentanti giorno "g", mese "m", e anno "a". Il metodo deve controllare se il film il cui campo "codice" corrisponde a "cod" è presente nella biblioteca e non è stato noleggiato. In tal caso, si deve effettuare il noleggio di tale film all'utente "ut" utilizzando la data specificata da "g", "m" ed "a". Il metodo deve ritornare "0" se il film viene noleggiato con successo, "-1" se il film non è presente nell'archivio e "-2" se il film è presente ma è già stato noleggiato da un altro utente. Il campo codice deve essere considerato univoco, quindi non è possibile che due film diversi abbiano lo stesso codice.

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	Memento	Christopher Nolan	2000	Guy Pearce Carrie-Anne Moss		
1	Number 23	Joel Schumacher	2007	Jim Carrey		
2	Il cavaliere oscuro	Christopher Nolan	2008	Christian Bale Heath Ledger	Mario Rossi	21/06/2011
3	Equilibrium	Kurt Wimmer	2002	Christian Bale		

- ordini i vettori "videoteca" e "noleggi" utilizzando il metodo "ordinaAnno" e, poi, stampi nuovamente l'archivio.
- stampi l'elenco dei titoli dei film diretti da Christopher Nolan, usando il metodo "trovaFilm".
- stampi il numero di film disponibili per il noleggio, usando il metodo "contaDisponibili".
- effettui il noleggio del film avente codice "3" all'utente "Gianni Verdi" usando come data il 24/06/2011 e stampi l'esito della funzione e l'archivio aggiornato.

Soluzione

```
public class Esame {  
  
    public static final int MAX_FILM = 256;  
  
    public static int contaFilm(Film[] filmdb) {  
        int nFilm = 0;  
        for (int i = 0; i < MAX_FILM; i++) {  
            if (filmdb[i] != null) {  
                nFilm++;  
            }  
        }  
        /*  
        * Per evitare di scorrere l'intero array, il conteggio può essere  
        * realizzato anche nel seguente modo (considerando che non ci devono  
        * essere elementi nulli in mezzo agli elementi validi):  
        */  
        /*  
        int nFilm = 0;  
        while (nFilm < MAX_FILM && filmdb[nFilm] != null) {  
            nFilm++;  
        }  
        */  
        return nFilm;  
    }  
  
    public static void ordinaAnno(Film[] filmdb, Noleggio[] noldb) {  
        /* 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 (filmdb[i].anno > filmdb[j].anno) {  
                    /* Scambio i due film nelle posizioni i e j */  
                    Film tmp = filmdb[i];  
                    filmdb[i] = filmdb[j];  
                    filmdb[j] = tmp;  
                    /*  
                    * Per mantenere la consistenza dei dati, vanno scambiati  
                    * anche i corrispondenti elementi del vettore parallelo  
                    */  
                    Noleggio tmp2 = noldb[i];  
                    noldb[i] = noldb[j];  
                    noldb[j] = tmp2;  
                }  
            }  
        }  
    }  
  
    public static Film[] trovaFilm(Film[] filmdb, String reg) {  
        /* Conteggio dei film presenti nell'archivio */  
        int nFilm = contaFilm(filmdb);  
  
        /* Conteggio del numero di risultati della ricerca */  
        int nRis = 0;  
        for (int i = 0; i < nFilm; i++) {  
            if (filmdb[i].regia.equals(reg)) {  
                nRis++;  
            }  
        }  
  
        /*  
        * Una volta noto il numero di risultati, è possibile allocare un  
        * vettore di dimensione adeguata  
        */  
        Film[] ris = new Film[nRis];  
  
        /* Popolamento del vettore di risultati */  
        int j = 0;  
        for (int i = 0; i < nFilm; i++) {
```

```

        if (filmdb[i].regia.equals(reg)) {
            ris[j] = filmdb[i];
            j++;
        }
    }
    return ris;
}

public static int contaDisponibili(Film[] filmdb, Noleggio[] nolddb) {
    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);

    /* Conteggio dei film non noleggiati */
    int nDisp = 0;
    for (int i = 0; i < nFilm; i++) {
        if (nolddb[i] == null) {
            nDisp++;
        }
    }

    return nDisp;
}

public static int noleggiaFilm(Film[] filmdb, Noleggio[] nolddb, long cod,
    String ut, int g, int m, int a) {

    /* Conteggio dei film presenti nell'archivio */
    int nFilm = contaFilm(filmdb);

    /*
     * Ricerca dell'indice del libro da prestare. La variabile idx che
     * conterrà l'indice viene inizializzata col valore -1 a rappresentare
     * il fatto che l'indice non è stato trovato. Il ciclo termina quando
     * sono stati esaminati tutti i film presenti nell'array o quando viene
     * trovato l'indice.
     */
    int idx = -1;
    for (int i = 0; i < nFilm && idx == -1; i++) {
        if (filmdb[i].codice == cod) {
            idx = i;
        }
    }

    /*
     * Effettuata la ricerca, si possono valutare i vari casi. Se idx è
     * rimasto pari a -1, nessun film con il codice adeguato è stato
     * trovato. Se, invece, è stato trovato un film, idx conterrà la sua
     * posizione nell'array filmdb e le sue informazioni di noleggio si
     * troveranno in nolddb[idx].
     */

    int ret; // valore di ritorno
    if (idx == -1) { // codice non trovato
        ret = -1;
    } else { // codice trovato
        if (nolddb[idx] != null) { // film già noleggiato
            ret = -2;
        } else { // film disponibile
            ret = 0;
            /* Bisogna creare un nuovo oggetto Noleggio */
            nolddb[idx] = new Noleggio(ut, g, m, a);
        }
    }

    return ret;
}

/*
 * 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);
        }
    }
}

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

    String[] tmp = new String[] { "Guy Pearce", "Carrie-Anne Moss" };
    videoteca[0] = new Film(0, "Memento", "Christopher Nolan", 2000, tmp);
    /*
    * Si può evitare di usare la variabile tmp usando la new direttamente
    * nei parametri del costruttore:
    */
    /*
    videoteca[0] = new Film(0, "Memento", "Christopher Nolan", 2000,
        new String[] { "Guy Pearce", "Carrie-Anne Moss" });
    */
    tmp = new String[] { "Jim Carrey" };
    videoteca[1] = new Film(1, "Number 23", "Joel Schumacher", 2007, tmp);

    tmp = new String[] { "Christian Bale", "Heath Ledger" };
    videoteca[2] = new Film(2, "Il cavaliere oscuro", "Christopher Nolan",
        2008, tmp);

    tmp = new String[] { "Christian Bale" };
    videoteca[3] = new Film(3, "Equilibrium", "Kurt Wimmer", 2002, tmp);

    noleggi[2] = new Noleggio("Mario Rossi", 21, 6, 2011);

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

    ordinaAnno(videoteca, noleggi);
    System.out.println();
    System.out.println("Archivio dopo l'ordinamento:");
    stampaArchivio(videoteca, noleggi);

    Film[] ris = trovaFilm(videoteca, "Christopher Nolan");
    System.out.println();
    System.out.println("I film di Christopher Nolan sono:");
    for (int i = 0; i < ris.length; i++) {
        System.out.println(ris[i].titolo);
    }

    int ret = contaDisponibili(videoteca, noleggi);
    System.out.println();
    System.out.println("Sono disponibili per il noleggio ancora " + ret
        + " film");

    ret = noleggiaFilm(videoteca, noleggi, 3, "Gianni Verdi", 24, 6, 2011);
    System.out.println();
    if (ret == -1) {
        System.out.println("Il codice specificato non è valido");
    } else if (ret == -2) {
        System.out.println("Il film non è disponibile");
    } else {
        System.out.println("Il film è stato prestato con successo");
    }
}

```

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