Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica
a.a. 2010-11
8 luglio 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 {

    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 void ordinaCodice(Film[] filmdb, Noleggio[] noldb, boolean asc)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio e la variabile booleana "asc" che rappresenta la direzione di ordinamento. Il metodo deve ordinare i vettori secondo il campo codice in senso crescente se "asc" è "true" o decrescente altrimenti.

**b) Scrivere il metodo statico:**

```
public static Film[] trovaAttore(Film[] filmdb, String att)
```

che prende in ingresso il vettore "filmdb" che rappresenta l'archivio della videoteca e la stringa "att" e ritorna un vettore di Film contenente soltanto i film che hanno nel cast l'attore specificato dalla stringa "att".

**c) Scrivere il metodo statico:**

```
public static int[] noleggiatiUtente(Film[] filmdb, Noleggio[] noldb,
        String ut)
```

che prende in ingresso i vettori "filmdb" e "noldb" costituenti il database di un videonoleggio e la stringa "ut" e ritorna un vettore di interi. Il vettore ritornato deve contenere gli indici (cioè le posizioni nel vettore "filmdb") dei film noleggiati dall'utente "ut".

**d) Scrivere il metodo statico:**

```
public static boolean aggiungiFilm(Film[] filmdb, String titolo,
        String regia, int anno, String[] cast)
```

che prende in ingresso il vettore "filmdb" che rappresenta l'archivio della videoteca e le informazioni descrittive di un film (le stringhe "titolo" e "regia", l'intero "anno" ed il vettore di stringhe "cast"). Il metodo deve aggiungere nella prima posizione libera del vettore "filmdb" un nuovo film. Le informazioni necessarie per l'inizializzazione del nuovo film sono fornite tramite i parametri del metodo, tranne il codice che, quindi, deve essere calcolato dal metodo stesso. Il campo codice può assumere valori tra 0 e MAX_FILM − 1 (estremi compresi). Il metodo deve cercare il più piccolo valore di codice disponibile ed utilizzarlo per l'inizializzazione del film aggiunto.

**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 |
|---|---|---|---|---|---|---|
| 3 | Poliziotto superpiù | Sergio Corbucci | 1980 | Terence Hill | | |
| 0 | Lo chiamavano Trinità | Enzo Barboni | 1970 | Bud Spencer<br>Terence Hill | Gianni Verdi | 6/7/2011 |
| 2 | Io sto con gli ippopotami | Italo Zingarelli | 1979 | Bud Spencer<br>Terence Hill | Mario Rossi | 6/7/2011 |
| 4 | Bomber | Michele Lupo | 1982 | Bud Spencer<br>Jerry Calà | Mario Rossi | 7/7/2011 |

- ordini i vettori "videoteca" e "noleggi" in senso crescente utilizzando il metodo "ordinaCodice" e, poi, stampi nuovamente l'archivio.
- stampi l'elenco dei titoli dei film che hanno "Bud Spencer" nel cast, usando il metodo "trovaAttore".
- stampi il titolo e la data di noleggio dei film noleggiati da "Mario Rossi", usando il metodo "noleggiatiUtente".
- aggiunga, utilizzando il metodo "aggiungiFilm", il film "Pari e dispari", del 1978, diretto da Sergio Corbucci con Bud Spencer e Terence Hill nel cast e stampi l'esito della funzione e l'archivio aggiornato.

## *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;
        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;
    }


    /*
     * 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 void ordinaCodice(Film[] filmdb, Noleggio[] noldb,
            boolean asc) {
        /* 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++) {
                /*
                 * Per accedere al campo privato codice bisogna usare il metodo
                 * getter getCodice(). La condizione che stabilisce se due
                 * elementi vanno scambiati o no è, quindi:
                 * filmdb[i].getCodice() > filmdb[j].getCodice()
                 * se l'ordinamento è crescente, mentre
                 * filmdb[i].getCodice() < filmdb[j].getCodice()
                 * se invece è decrescente. La condizione che tiene in
                 * considerazione il parametro asc diventa, quindi:
                 */
                if ((asc && filmdb[i].getCodice() > filmdb[j].getCodice())
                        || (!asc && filmdb[i].getCodice() < filmdb[j]
```

```
                            .getCodice())) {
                    /* 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[] trovaAttore(Film[] filmdb, String att) {
        /* Conteggio dei film presenti nell'archivio */
        int nFilm = contaFilm(filmdb);
        int nRis = 0;
        /* Conteggio degli elementi del vettore risultato */
        /* Il primo for scorre tutti i film dell'archivio */
        for (int i = 0; i < nFilm; i++) {
            /*
             * Il for interno scorre i membri del cast del film in esame
             * (filmdb[i]) finché non viene trovato l'attore desiderato o
             * finché non sono stati esaminati tutti gli attori del film
             */
            boolean trovato = false;
            for (int j = 0; j < filmdb[i].cast.length && !trovato; j++) {
                if (filmdb[i].cast[j].equals(att)) {
                    nRis++;
                    trovato = true;
                }
            }
        }
        /* Creazione del vettore risultato */
        Film[] ris = new Film[nRis];
        /* Popolamento del vettore risultato */
        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)) {
                    ris[k] = filmdb[i];
                    k++;
                    trovato = true;
                }
            }
        }
        return ris;
    }

    public static int[] noleggiatiUtente(Film[] filmdb, Noleggio[] noldb,
            String ut) {
        /* 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++) {
            /*
             * Prima di controllare il campo utente dell'oggetto riferito da
             * noldb[i] bisogna assicurarsi che il riferimento non sia nullo, o
             * si avrà un errore a tempo di esecuzione
             */
            if (noldb[i] != null && noldb[i].utente.equals(ut)) {
                nRis++;
            }
        }
        /* Creazione del vettore risultato */
```

```
        int[] ris = new int[nRis];
        /* Popolamento del vettore risultato */
        int k = 0;
        for (int i = 0; i < nFilm; i++) {
            if (noldb[i] != null && noldb[i].utente.equals(ut)) {
                ris[k] = i;
                k++;
            }
        }
        return ris;
    }

    public static boolean aggiungiFilm(Film[] filmdb, String titolo,
            String regia, int anno, String[] cast) {
        /* Conteggio dei film presenti nell'archivio */
        int nFilm = contaFilm(filmdb);
        boolean ret;
        if (nFilm >= MAX_FILM) {
            ret = false;
        } else {
            /* Ricerca del primo codice libero (sarà memorizzato in codice). */
            int codice = -1;
            /*
             * Si esaminano i possibili valori di codice (da 0 a MAX_FILM) e si
             * controlla che ciascuno di essi sia presente. Appena viene
             * trovato un valore di codice che non sia già in uso, il ciclo
             * viene interrotto (la variabile "codice" conterrà un valore
             * diverso da -1) e si usa quel valore per l'inserimento. Dato che
             * il numero di film è minore di MAX_FILM, sicuramente sarà trovato
             * un valore di codice disponibile, quindi non son necessari altri
             * controlli.
             */
            for (int cod = 0; cod < MAX_FILM && codice == -1; cod++) {
                boolean presente = false;
                /*
                 * Si scorrono tutti i film e si controlla il codice candidato
                 * (cod) con il campo codice dei film presenti nella videoteca.
                 * Se questo viene trovato si interrompe il ciclo mettendo a
                 * true la variabile "presente".
                 */
                for (int i = 0; i < nFilm && !presente; i++) {
                    if (cod == filmdb[i].getCodice()) {
                        presente = true;
                    }
                }
                /*
                 * Se il codice candidato (cod) non è stato trovato in nessun
                 * film, allora presente sarà false ed il codice può essere
                 * usato per l'inserimento
                 */
                if (!presente) {
                    codice = cod;
                }
            }
            filmdb[nFilm] = new Film(codice, titolo, regia, anno, cast);
            ret = true;
        }
        return ret;
    }

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

        String[] tmp = new String[] { "Terence Hill" };
        videoteca[0] = new Film(3, "Poliziotto superpiù", "Sergio Corbucci",
                1980, tmp);
        /*
         * Si può evitare di usare la variabile tmp usando la new direttamente
         * nei parametri del costruttore:
         */
```

```
        /*
        videoteca[0] = new Film(3, "Poliziotto superpiù", "Sergio Corbucci",
                1980, new String[] { "Terence Hill" });
        */
        tmp = new String[] { "Bud Spencer", "Terence Hill" };
        videoteca[1] = new Film(0, "Lo chiamavano Trinità", "Enzo Barboni",
                1970, tmp);

        tmp = new String[] { "Bud Spencer", "Terence Hill" };
        videoteca[2] = new Film(2, "Io sto con gli ippopotami",
                "Italo Zingarelli",    1979, tmp);

        tmp = new String[] { "Bud Spencer", "Jerry Calà" };
        videoteca[3] = new Film(4, "Bomber", "Michele Lupo", 1982, tmp);

        noleggi[1] = new Noleggio("Gianni Verdi", 6, 7, 2011);
        noleggi[2] = new Noleggio("Mario Rossi", 6, 7, 2011);
        noleggi[3] = new Noleggio("Mario Rossi", 7, 7, 2011);

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

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

        Film[] ris = trovaAttore(videoteca, "Bud Spencer");
        System.out.println();
        System.out.println("Film con Bud Spencer:");
        for (int i = 0; i < ris.length; i++) {
            System.out.println(ris[i].titolo);
        }

        int[] ris2 = noleggiatiUtente(videoteca, noleggi, "Mario Rossi");
        System.out.println();
        System.out.println("Film noleggiati da Mario Rossi:");
        for (int i = 0; i < ris2.length; i++) {
            int idx = ris2[i];
            System.out.println(videoteca[idx].titolo + " / noleggiato il "
                    + noleggi[idx].giorno + "/" + noleggi[idx].mese + "/"
                    + noleggi[idx].anno);
        }

        tmp = new String[] { "Bud Spencer", "Terence Hill" };
        boolean ret = aggiungiFilm(videoteca, "Pari e dispari",
                "Sergio Corbucci", 1978, tmp);
        System.out.println();
        if (!ret) {
            System.out.println("L'archivio è pieno");
        } else {
            System.out.println("Il film è stato aggiunto con successo.");
            System.out.println("Archivio dopo l'aggiunta:");
            stampaArchivio(videoteca, noleggi);
        }
    }

}
```