

Esame di Fondamenti di Informatica L-B Ingegneria Gestionale e dei Processi Gestionali (L-Z)

Appello del 15/7/2008

Esercizio 1 (4 punti)

Gestione ed utilizzo di liste in Java.

Esercizio 2 (6 punti)

Siano date le seguenti funzioni C:

```
int f(int V[], int M, int N) {
    int i=N, sum=0;
    for( i<M; )
        sum+=V[i++];
    return sum;
}

int g(int V[], int N, int M) {
    int j=0, res=1;
    while(j<M) {
        res*=f(V, N, j);
        j++;
    }
    return res;
}
```

1. Calcolare la complessità in passi base della funzione *f* nei termini dei parametri *M* ed *N* (suggerimento: si supponga $M > N$).
2. Calcolare la complessità in passi base della funzione *g* nei termini dei parametri *M* ed *N*.
3. Calcolare la complessità asintotica della funzione *g* nei termini dei parametri *M* ed *N*.

Esercizio 3 (6 punti)

In previsione della prossima apertura, la catena di videonoleggi “Belbostruck” deve approntare il sistema informatico per la gestione dei prestiti dei DVD. A tal scopo, ogni DVD a noleggio è caratterizzato dal titolo, il codice della copia (ogni titolo può essere presente in più copie, ognuna con un codice univoco), la casa di distribuzione e, in caso il DVD sia in prestito, il codice della tessera del cliente che ha noleggiato la copia. Si scriva una classe *DVD* per la catena “Belbostruck” che:

1. Possieda un opportuno costruttore con parametri (inizialmente il DVD non è in prestito).
2. Presenti opportuni metodi che permettano di accedere alle variabili di istanza della classe.
3. Presenti il metodo `toString` che fornisca una descrizione del DVD.
4. Possieda il metodo `equals` per stabilire l’uguaglianza con un altro oggetto *DVD* (l’uguaglianza va verificata sul codice della copia).
5. Implementi l’interfaccia *Comparable*, definendo il metodo `compareTo` per stabilire la precedenza con un *DVD* passato come parametro (l’ordine è quello lessicografico sul titolo e, a parità di titolo, sul codice della copia).

Esercizio 4 (9 punti)

Si scriva una classe *Videoteca* che memorizzi le informazioni relative ai DVD presenti all’interno di ciascun negozio della catena. In particolare, oltre al codice del negozio, i DVD vanno memorizzati in una lista. La classe *Videoteca* deve inoltre:

1. Possedere un opportuno costruttore (inizialmente il negozio non contiene DVD).
2. Presentare un metodo `trovaDVD` che, dato un titolo, restituisca il primo oggetto *DVD* con tale titolo disponibile al prestito, se questo esiste.
3. Possedere un metodo `aggiungiDVD` che, dato un oggetto *DVD*, lo inserisca in coda alla lista.
4. Presentare un metodo `disponibili` che, dato un titolo, restituisca il numero di copie con tale titolo disponibili al noleggio.
5. Possedere il metodo `toString` che restituisca una stringa che fornisca una descrizione del negozio, comprendendo anche i DVD presenti (esclusi quelli noleggiati).
6. Presentare il metodo `equals` per stabilire l’uguaglianza con un altro oggetto *Videoteca* (l’uguaglianza va verificata sul codice del negozio).

Esercizio 5 (7 punti)

Si scriva un’applicazione per la catena “Belbostruck” che:

1. Crei un insieme di oggetti *Videoteca*.
2. Crei un oggetto *Videoteca*, lette da tastiera le informazioni necessarie.
3. Provveda ad inserire all’interno dell’insieme di cui al punto 1. l’oggetto creato al punto 2., controllando che un oggetto uguale non esista già.
4. Letto da tastiera un titolo di film, provveda a stampare il codice della prima copia disponibile di tale DVD all’interno del negozio di cui al punto 2.
5. Letto da tastiera un titolo di film, provveda a stampare il numero totale di copie disponibile di tale DVD all’interno di tutti i negozi della catena.

Per la lettura di dati da tastiera è possibile utilizzare l’oggetto `Lettore.in`, definito all’interno del package `fi.ji.io`, che possiede i seguenti metodi:

- `char leggiChar()` Legge un singolo carattere.
- `double leggiDouble()` Legge un numero razionale (delimitato da spazi).
- `float leggiFloat()` Legge un numero razionale (delimitato da spazi).
- `int leggiInt()` Legge un intero (delimitato da spazi).
- `String leggiLinea()` Legge una linea di testo.
- `String leggiString()` Legge una parola senza spazi al suo interno.

Soluzione Esercizio 2

Domanda 1:

2 assegnamenti	2
for	$M - N + 1$
sum+=V[i++]	$M - N$
Totale	$3 + 2M - 2N$

Domanda 2:

2 assegnamenti	2
while	$M + 1$
chiamata di f	M
complessità di f	$4M + 2MN - M^2$
j++	M
Totale	$2MN - M^2 + 7M + 3$

Domanda 3:

Complessità asintotica: $O(MN)$

Soluzione Esercizio 3

```
class DVD implements Comparable<DVD> {
    private String titolo, codice, casa, tessera;
    private boolean disponibile;

    public DVD(String titolo, String codice, String casa) {
        this.titolo=titolo;
        this.codice=codice;
        this.casa=casa;
        this.tessera=null;
        disponibile=true;
    }

    public String getTitolo() { return titolo; }
    public String getCodice() { return codice; }
    public String getCasa() { return casa; }
    public String getTessera() { return tessera; }
    public boolean isDisponibile() { return disponibile; }

    public String toString() {
        return titolo+" (" +codice+", "+casa;
    }

    public int compareTo(DVD d) {
        int ret=this.titolo.compareTo(d.titolo);
        if(ret==0) ret=this.codice.compareTo(d.codice);
        return ret;
    }

    public boolean equals(Object o) { return equals((DVD) o); }
    public boolean equals(DVD d) { return (codice.equals(d.codice)); }
}
```

Soluzione Esercizio 4

```
import java.util.*;
class Videoteca {
    private int codice;
    private List<DVD> copie;

    public Videoteca(int codice) {
        this.codice=codice;
        copie=new ArrayList<DVD>();
    }

    public DVD trovaDVD(String titolo) {
        for(DVD d:copie)
            if(d.getTitolo().equals(titolo)&& d.isDisponibile()) return d;
        return null;
    }

    public void aggiungiDVD(DVD d) { copie.add(d); }
    public int disponibili(String titolo) {
        int n=0;
        for(DVD d:copie)
            if(d.getTitolo().equals(titolo)&& d.isDisponibile()) n++;
        return n;
    }

    public String toString() {
        String s="Codice negozio: "+codice+"\n";
        for(DVD d:copie)
            if(d.isDisponibile()) s+=d.toString()+" ";
        return s;
    }

    public boolean equals(Object o) { return equals((Videoteca) o); }
    public boolean equals(Videoteca v) { return (codice==v.codice); }
}
```

Soluzione Esercizio 5

```
import fiji.io.*;
import java.util.*;
class Applicazione {
    public static void main(String[] args) {
        Set<Videoteca> s=new TreeSet<Videoteca>(); // domanda 1
        Videoteca v=new Videoteca(Lettore.in.leggiInt()); // domanda 2
        s.add(v); // domanda 3
        DVD d=v.trovaDVD(Lettore.in.leggiLinea());
        if(d==null) System.out.println("Non ci sono copie disponibili");
        else System.out.println(d.getCodice()); // domanda 4
        String titolo=Lettore.in.leggiLinea();
        int copie=0;
        for(Videoteca n:s)
            copie+=n.disponibili(titolo);
        System.out.println("Copie disponibili: "+copie);
    }
}
```