

Esame di Fondamenti di Informatica T-1/T-A

Ingegneria Gestionale (A-K)

Appello del 16/9/2016

NOTA: Per il superamento dell'esame è **necessario** ottenere la sufficienza nello svolgimento dell'Esercizio 1.

Esercizio 1 (4 punti)

1. Discutere il concetto di interfaccia in Java, evidenziandone le differenze con una classe astratta.
2. Descrivere la gerarchia di memorie all'interno di un elaboratore.

Esercizio 2 (2 punti)

Rappresentare in binario il numero **35,75** supponendo di utilizzare 8 bit per la mantissa (in modulo e segno) ed 8 bit per l'esponente (in complemento a 2). Si riconverta infine il numero rappresentato in base 10, motivando eventuali differenze con il numero originale.

Esercizio 3 (5 punti)

Siano dati i seguenti metodi Java:

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

public static int g(int V[], int N) {
    int j=0, sum=0;
    while(j<N)
        sum+=f(V, ++j);
    return sum;
}
```

1. Calcolare la complessità in passi base del metodo `f` nei termini del parametro `N` (si distinguano i casi in cui `N` assume valori pari da quelli in cui assume valori dispari).
2. Calcolare la complessità in passi base del metodo `g` nei termini del parametro `N` (si supponga `N` dispari).
3. Calcolare la complessità asintotica del metodo `g` nei termini del parametro `N`.

Esercizio 4 (5 punti)

Christian Secchi è un appassionato cinefilo e possiede una vasta collezione di DVD. In previsione del suo prossimo trasloco, Christian ha deciso di gestire le informazioni relative ai DVD posseduti all'interno di un calcolatore. In particolare, per ogni attore occorre registrare il nome e il cognome, l'anno di nascita e la nazionalità. Si scriva una classe `Attore` per Christian Secchi che:

1. Possieda un opportuno costruttore con parametri.
2. Presenti opportuni metodi che permettano di accedere alle variabili d'istanza dell'oggetto.
3. Presenti il metodo `toString` che fornisca una descrizione dell'attore.
4. Possieda il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Attore` (la verifica va fatta su nome, cognome e anno di nascita).
5. Implementi l'interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con un oggetto `Attore` passato come parametro (in ordine alfabetico per cognome e nome e, a parità, per anno di nascita crescente).

Esercizio 5 (7 punti)

Si scriva una classe `DVD` che memorizzi le informazioni riguardanti un singolo DVD della collezione. Per ogni film occorre memorizzare il titolo, il nome del regista, l'anno di produzione, mentre gli attori del cast vanno inseriti all'interno di un insieme. La classe `DVD` deve:

1. Presentare un opportuno costruttore con parametri (inizialmente, il film non contiene attori).
2. Possedere opportuni metodi che permettano di accedere alle variabili d'istanza dell'oggetto.
3. Presentare il metodo `toString` che fornisca la descrizione del DVD (inclusa la descrizione di tutti gli attori del cast).
4. Possedere il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `DVD` (la verifica va effettuata sul titolo e l'anno di produzione).
5. Presentare il metodo `aggiungi` che, dato un oggetto `Attore`, lo inserisca all'interno dell'insieme, controllando che tale inserimento sia possibile.
6. Possedere il metodo `attore` che, data una stringa rappresentante il nome e il cognome di un attore, indichi se il cast contiene almeno un attore con tale nome e cognome.
7. Presentare il metodo `attori` che, dato il nome di una nazione, restituisca un insieme contenente gli attori del cast con tale nazionalità.

Esercizio 6 (8 punti)

Si scriva un'applicazione per Christian Secchi che:

1. Crei una lista di oggetti `DVD`.
2. Crei un oggetto `DVD`, lette da tastiera le informazioni necessarie.
3. Inserisca l'oggetto di cui al punto 2. in coda alla lista di cui al punto 1.
4. Crei un oggetto `Attore`, lette da tastiera le informazioni necessarie.
5. Inserisca l'oggetto di cui al punto 4. all'interno del `DVD` di cui al punto 2.
6. Letti da tastiera il nome e il cognome di un attore, stampi a video il titolo di tutti i film interpretati da tale attore all'interno della lista di cui al punto 1.
7. Letta da tastiera una nazionalità, stampi a video il nome e il cognome dell'attore di tale nazionalità più anziano tra quelli della lista di cui al punto 1.

Soluzione Esercizio 2

$35,75_{10} = 100011,11_2$ quindi la mantissa è **(0)1000111**, l'esponente $6_{10} = 00000110$.

Il numero rappresentato è diverso dall'originale poiché quest'ultimo possiede un numero di cifre significative maggiore di 7.

Soluzione Esercizio 3

Domanda 1:

2 assegnamenti	2	$O(2)$
$i++<N$	$N/2 + 1$	$O(N/2 + 1)$
$sum+=V[i]$	$N/2$	$O(N/2)$
$++i$	$N/2$	$O(N/2)$

Totale $3N/2 + 3$ $O(3N/2 + 9/2)$

$$\text{complessità di } f: \sum_{j \text{ pari}}^N \left(\frac{3j}{2} + 3 \right) + \sum_{j \text{ dispari}}^N \left(\frac{3j}{2} + \frac{9}{2} \right) = \sum_{j=1}^N \left(\frac{3j}{2} + 3 \right) + \sum_{j \text{ dispari}}^N \frac{3}{2} = \frac{3N(N+1)}{4} + 3N + 3 \frac{N+1}{4} = \frac{3N^2}{4} + \frac{9N}{2} + \frac{3}{4}$$

Domanda 2:

2 assegnamenti	2
$j<N$	$N+1$
$sum+=f(V, ++j)$	N
complessità di f	$3N^2/4 + 9N/2 + 3/4$

Totale $3N^2/4 + 13N/2 + 15/4$

Domanda 3:

Complessità asintotica: $O(N^2)$

Soluzione Esercizio 4

```
class Attore implements Comparable<Attore> {
    private String nome, cognome, nazione;
    private int anno;

    public Attore(String nome, String cognome, int anno, String nazione) {
        this.nome = nome;
        this.cognome = cognome;
        this.anno = anno;
        this.nazione = nazione;
    }

    public String getNome() { return nome + " " + cognome; }
    public int getAnno() { return anno; }
    public String getNazione() { return nazione; }

    public String toString() {
        return getNome() + ": " + anno + " (" + nazione + ")";
    }

    public boolean equals(Object o) { return equals((Attore) o); }
    public boolean equals(Attore a) {
        return nome.equals(a.nome) && cognome.equals(a.cognome) && anno==a.anno;
    }

    public int compareTo(Attore a) {
        int ret = cognome.compareTo(a.cognome);
        if(ret==0) ret = nome.compareTo(a.nome);
        if(ret==0) ret = anno-a.anno;
        return ret;
    }
}
```

Soluzione Esercizio 5

```
import java.util.*;
class DVD {
    private String titolo, regista;
    private int anno;
    private Set<Attore> s;
    public DVD(String titolo, String regista, int anno) {
        this.titolo = titolo;
        this.regista = regista;
        this.anno = anno;
        s = new HashSet<Attore>();
    }
    public String getTitolo() { return titolo; }
    public String getRegista() { return regista; }
    public int getAnno() { return anno; }
    public String toString() {
        return titolo + "(" + regista + "), " + anno + ":" + s;
    }
    public boolean equals(Object o) { return equals((DVD) o); }
    public boolean equals(DVD d) {
        return titolo.equals(d.titolo) && anno==d.anno;
    }
    public boolean aggiungi(Attore a) { return s.add(a); }
    public boolean attore(String s) {
        for(Attore a: this.s) if(a.getNome().equals(s)) return true;
        return false;
    }
    public Set<Attore> attori(String nazione) {
        Set<Attore> x = new TreeSet<Attore>();
        for(Attore a: s)
            if(a.getNazione().equals(nazione)) x.add(a);
        return x;
    }
}
```

Soluzione Esercizio 6

```
import java.util.*;
class Applicazione {
    public static void main(String[] args) {
        List<DVD> l = new ArrayList<DVD>();
        Scanner t = new Scanner(System.in);
        DVD d = new DVD(t.nextLine(), t.nextLine(), t.nextInt());
        l.add(d);
        Attore a = new Attore(t.nextLine(), t.nextLine(), t.nextInt(),
            t.nextLine());
        d.aggiungi(a);
        String nome = t.nextLine(), cognome = t.nextLine();
        for(DVD x: l)
            if(x.attore(nome+" "+cognome)) System.out.println(x.getTitolo());
        String nazione = t.nextLine();
        Attore anziano = null;
        for(DVD x: l) {
            Set<Attore> s=x.attori(nazione);
            for(Attore y: s)
                if(anziano==null || y.getAnno()<anziano.getAnno()) anziano=y;
        }
        System.out.println(anziano.getNome());
    }
}
```