

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

Ingegneria Gestionale (A-K)

Appello del 11/7/2014

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 classe astratta.
2. Descrivere la codifica dei numeri interi all'interno di un elaboratore elettronico.

Esercizio 2 (2 punti)

Convertire in binario i numeri **-97** e **-32**, supponendo di utilizzare una rappresentazione a 8 bit in complemento a 2. Si esegua infine la somma dei due numeri, riconvertendo il risultato in base 10, motivando eventuali differenze tra il risultato ottenuto e quello atteso.

Esercizio 3 (5 punti)

Siano dati i seguenti metodi Java:

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

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

1. Calcolare la complessità in passi base del metodo `f` nei termini dei parametri `M` e `N` (si distinguano i casi in cui `N` assume valori minori di `M` da quelli in cui assume valori maggiori o uguali a `M`).
2. Calcolare la complessità in passi base del metodo `g` nei termini dei parametri `M` e `N` (si supponga `N` maggiore di `M`).
3. Calcolare la complessità asintotica del metodo `g` nei termini dei parametri `M` e `N`.

Esercizio 4 (5 punti)

L'istituto "Luz" dello stato caraibico di St. Vincent si occupa del restauro e dell'archiviazione dei film prodotti con fondi statali. Per questo vuole informatizzare la gestione dei film e degli attori che vi recitano. Le informazioni relative a ogni attore comprendono, oltre al nome e al cognome, la data di nascita e l'anno di debutto nel mondo del cinema. Si scriva una classe `Attore` che:

1. Possieda un opportuno costruttore con parametri.
2. Presenti opportuni metodi che permettano di accedere alle variabili di 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 e cognome).
5. Implementi l'interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con un oggetto `Attore` passato come parametro (in ordine decrescente di età e, a parità, per ordine alfabetico di cognome e nome).

Esercizio 5 (7 punti)

Si scriva una classe `Film` che memorizzi le informazioni relative a un film dell'istituto "Luz". Per ogni film occorre memorizzare il titolo, il cognome del regista, l'anno di produzione e gli attori che vi recitano (all'interno di un insieme). La classe `Film` deve:

1. Presentare un opportuno costruttore con parametri (inizialmente, l'insieme degli attori è vuoto).
2. Possedere opportuni metodi che permettano di accedere alle variabili di istanza dell'oggetto.
3. Presentare il metodo `toString` che fornisca la descrizione del film (inclusa la descrizione di tutti gli attori).
4. Possedere il metodo `aggiungi` che, dato un oggetto `Attore`, lo inserisca all'interno dell'insieme, controllando che tale inserimento sia possibile.
5. Presentare il metodo `attori` che restituisca l'insieme degli attori che recitano nel film che hanno debuttato prima dell'anno passato come parametro al metodo.
6. Possedere il metodo `cerca` che, dato il nome e il cognome di un attore, indichi se tale attore recita o meno nel film.
7. Possedere il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Film` (la verifica va fatta sul nome e sull'anno di produzione del film).

Esercizio 6 (9 punti)

Si scriva un'applicazione per l'istituto "Luz" che:

1. Crei una lista di oggetti `Film`.
2. Crei un oggetto `Film`, lette da tastiera le informazioni necessarie.
3. Inserisca l'oggetto di cui al punto 2. nella prima posizione della lista di cui al punto 1.
4. Crei un oggetto `Attore`, lette da tastiera le informazioni necessarie.
5. Inserisca l'attore creato al punto 4. tra quelli del film di cui al punto 2, controllando che tale inserimento sia possibile.
6. Letto da tastiera il cognome di un regista, stampi a video il numero di film diretti da tale regista all'interno della lista di cui al punto 1.
7. Stampi a video il titolo di tutti i film, all'interno della lista di cui al punto 1., nei quali l'attore di cui al punto 4. ha recitato.
8. Stampi a video le informazioni dell'attore più anziano tra quelli che recitano nel film di cui al punto 2. e che hanno debuttato prima del 2013.

Soluzione Esercizio 2

$$\begin{aligned} -97_{10} &= 10011111_2 \\ -32_{10} &= 11100000_2 \end{aligned}$$

$$01111111_2 = 127_{10}$$

Il numero è diverso dal risultato atteso (-129), in quanto quest'ultimo non è rappresentabile con soli 8 bit.

$$\begin{array}{r} 10011111+ \\ \underline{11100000} \\ 01111111 \end{array}$$

Soluzione Esercizio 3

Domanda 1:

2 assegnamenti	2	o 2
--i>N	M-N	o 1
sum+=V[i]	M-N-1	o 0
Totale	2M-2N+1	o 3

Domanda 2:

2 assegnamenti	2
sum+=f(V, N, ++j)	M
j<M	M
complessità di f	$2MN - M^2$
Totale	$2MN - M^2 + 2M + 2$

$$\text{complessità di f: } \sum_{j=1}^M (2N - 2j + 1) = 2MN - M(M + 1) + M = 2MN - M^2$$

Domanda 3:

Complessità asintotica: $O(MN)$

Soluzione Esercizio 4

```
class Attore implements Comparable<Attore> {
    private String nome, cognome;
    private int g, m, a, deb;

    public Attore(String nome, String cognome, int g, int m, int a, int deb) {
        this.nome=nome;
        this.cognome=cognome;
        this.g=g; this.m=m; this.a=a;
        this.deb=deb;
    }

    public String getNome() { return nome; }
    public String getCognome() { return cognome; }
    public String getData() { return g+"/"+m+"/"+a; }
    public int getDebutto() { return deb; }

    public String toString() {
        return nome + " " + cognome + ": " + getData() + " - " + deb;
    }

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

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

Soluzione Esercizio 5

```
import java.util.*;
class Film {
    private String titolo, regista;
    private int anno;
    private Set<Attore> s;
    public Film(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 + " (" + anno + "): " + regista + "; " + s.toString();
    }
    public boolean aggiungi(Attore a) { return s.add(a); }
    public Set<Attore> attori(int anno) {
        Set<Attore> attori = new TreeSet<Attore>();
        for(Attore a:s) if(a.getDebutto()<anno) attori.add(a);
        return attori;
    }
    public boolean cerca(String nome, String cognome) {
        for(Attore a: s)
            if(a.getNome().equals(nome) && a.getCognome().equals(cognome))
                return true;
        return false;
    }
    public boolean equals(Object o) { return equals((Film) o); }
    public boolean equals(Film f) {
        return titolo.equals(f.titolo) && anno==f.anno;
    }
}
```

Soluzione Esercizio 6

```
import java.util.*;
class Applicazione {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Film> l=new ArrayList<Film>();
        Film f = new Film(scanner.nextLine(), scanner.nextLine(),
            scanner.nextInt());
        l.add(0, f);
        Attore a = new Attore(scanner.nextLine(), scanner.nextLine(),
            scanner.nextInt(), scanner.nextInt(), scanner.nextInt(),
            scanner.nextInt());
        f.aggiungi(a);
        String regista = scanner.nextLine();
        int quanti = 0;
        for(Film x: l) if(f.getRegista().equals(regista)) quanti++;
        System.out.println(quanti);
        for(Film x:l) if(x.cerca(a.getNome(), a.getCognome()))
            System.out.println(x.getTitolo());
        Attore max = null;
        Set<Attore> s = f.attori(2013);
        for(Attore x: s) if(max == null || x.compareTo(max) < 0) max = x;
        System.out.println(max);
    }
}
```