

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

Ingegneria Gestionale (A-K)

Appello del 17/2/2017

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

Esercizio 1 (4 punti)

1. Illustrare i componenti interni di una CPU.
2. Descrivere i tipi primitivi del linguaggio Java.

Esercizio 2 (2 punti)

Convertire in binario i numeri **-59** e **-68**, 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 N) {
    int i=0, sum=0;
    while(++i<N)
        sum+=V[++i];
    return sum;
}

public static int g(int V[], int N) {
    int j=0, sum=0;
    for(; j<=N; ++j)
        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)

Tempi sempre più duri per il rettore dell'Università "La Sorbolona": la protesta degli studenti, sempre capitanata dall'irriducibile Joseph Palmier, ora si è spostata sull'esercizio delle biblioteche, portando alla necessità di una riorganizzazione del sistema di gestione dei prestiti di libri. Per questo, il rettore ha deciso di gestire all'interno di un calcolatore le informazioni relative ai libri conservati nelle varie biblioteche. In particolare, per ogni libro occorre registrare il titolo, il nome dell'autore e l'anno di pubblicazione. Si scriva una classe `Libro` per il rettore de "La Sorbolona" 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 del libro.
4. Possieda il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Libro` (la verifica va fatta esclusivamente sul titolo).
5. Implementi l'interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con un oggetto `Libro` passato come parametro (per ordine alfabetico del nome dell'autore e, a parità, per anno di pubblicazione crescente).

Esercizio 5 (7 punti)

Si scriva una classe `Biblioteca` che memorizzi le informazioni riguardanti una singola biblioteca de "La Sorbolona". Per ogni biblioteca occorre memorizzare il nome e l'indirizzo, mentre i libri vanno inseriti all'interno di un insieme. La classe `Biblioteca` deve:

1. Presentare un opportuno costruttore con parametri (inizialmente, l'elenco dei libri è vuoto).
2. Possedere opportuni metodi che permettano di accedere alle variabili d'istanza dell'oggetto.
3. Presentare il metodo `toString` che fornisca la descrizione della biblioteca (inclusa la descrizione di tutti i libri conservati).
4. Possedere il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Biblioteca` (la verifica va effettuata esclusivamente sul nome).
5. Presentare il metodo `aggiungi` che, dato un oggetto `Libro`, lo inserisca all'interno dell'insieme, controllando che tale inserimento sia possibile.
6. Possedere il metodo `autore` che, dato il nome di un autore, indichi il numero di libri di tale autore contenuti nella biblioteca.
7. Possedere il metodo `totale` che restituisca il numero totale di libri contenuti nella biblioteca.

Esercizio 6 (7 punti)

Si scriva un'applicazione per il rettore de "La Sorbolona" che:

1. Crei una lista di oggetti `Biblioteca`.
2. Crei un oggetto `Biblioteca`, 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 `Libro`, lette da tastiera le informazioni necessarie.
5. Inserisca l'oggetto di cui al punto 4. all'interno della biblioteca di cui al punto 2., controllando che tale inserimento sia possibile.
6. Letto da tastiera il nome di un autore, stampi a video l'indirizzo di tutte le biblioteche, all'interno della lista di cui al punto 1., che contengono libri di tale autore.
7. Stampi a video la descrizione della biblioteca che contiene più libri tra quelli della lista di cui al punto 1.

Soluzione Esercizio 2

$$-59_{10} = 11000101_2$$
$$-68_{10} = 10111100_2$$

$$10000001_2 = -127_{10}$$

Il numero rappresentato è uguale all'originale.

Soluzione Esercizio 3

Domanda 1:			Domanda 2:		
2 assegnamenti	2	o 2	2 assegnamenti	2	
i++<N	N/2 + 1	o (N + 1)/2	j<N	(N + 1)/2 + 1	
sum+=V[++i]	N/2	o (N - 1)/2	sum+=f(V, ++j)	(N + 1)/2	
Totale	N + 3	o N+ 2	++j	(N + 1)/2	
			complessità di f	N ² /4 + 3N/2 + 5/4	
			Totale	N ² /4 + 3N + 23/4	

$$\text{complessità di f: } \sum_{j=1}^N (j + 2) = \sum_{\substack{j=1 \\ j \text{ dispari}}}^{\frac{N-1}{2}} (2j + 3) = \frac{N-1}{2} \frac{N+1}{2} + 3 \frac{N+1}{2} = \frac{N^2}{4} + \frac{3N}{2} + \frac{5}{4}$$

Domanda 3:
Complessità asintotica: $O(N^2)$

Soluzione Esercizio 4

```
class Libro implements Comparable<Libro> {
    private String titolo, autore;
    private int anno;

    public Libro(String titolo, String autore, int anno) {
        this.titolo = titolo;
        this.autore = autore;
        this.anno = anno;
    }

    public String getTitolo() { return titolo; }
    public String getAutore() { return autore; }
    public int getAnno() { return anno; }

    public String toString() {
        return titolo + " (" + autore + "): " + anno;
    }

    public boolean equals(Object o) { return equals((Libro) o); }
    public boolean equals(Libro l) { return titolo.equals(l.titolo); }

    public int compareTo(Libro l) {
        int ret = this.autore.compareTo(l.autore);
        if(ret==0) ret = this.anno - l.anno;
        return ret;
    }
}
```

Soluzione Esercizio 5

```
import java.util.*;

class Biblioteca {
    private String nome, indirizzo;
    private Set<Libro> s;

    public Biblioteca(String nome, String indirizzo) {
        this.nome = nome;
        this.indirizzo = indirizzo;
        s = new TreeSet<Libro>();
    }

    public String getNome() { return nome; }
    public String getIndirizzo() { return indirizzo; }

    public String toString() { return nome + " (" + indirizzo + ": " + s; }

    public boolean equals(Object o) { return equals((Biblioteca) o); }
    public boolean equals(Biblioteca b){ return nome.equals(b.nome); }

    public boolean aggiungi(Libro l) { return s.add(l); }

    public int autore(String a) {
        int n = 0;
        for(Libro l: s) if(l.getAutore().equals(a)) n++;
        return n;
    }

    public int totale() { return s.size(); }
}
```

Soluzione Esercizio 6

```
import java.util.*;

class Applicazione {
    public static void main(String[] args) {
        List<Biblioteca> l = new ArrayList<Biblioteca>();
        Scanner scanner = new Scanner(System.in);
        Biblioteca b = new Biblioteca(scanner.nextLine(),
            scanner.nextLine());

        l.add(b);
        Libro t = new Libro(scanner.nextLine(), scanner.nextLine(),
            scanner.nextInt());
        if(!b.aggiungi(t)) System.out.println("Inserimento non riuscito!");
        String nome = scanner.nextLine();
        for(Biblioteca x: l)
            if(x.autore(nome)>0) System.out.println(x.getIndirizzo());
        Biblioteca max = null;
        for(Biblioteca x: l)
            if(max==null || x.totale()>max.totale())max=x;
        System.out.println(max);
    }
}
```