

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

Appello del 31/1/2011

Esercizio 1 (4 punti)

Definire il concetto di ricorsione.

Esercizio 2 (6 punti)

Siano dati i seguenti metodi Java:

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

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

1. Calcolare la complessità in passi base del metodo `f` nei termini del parametro `M` (si distinguano i casi in cui `M` 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 e si esprima $j=2i$).
3. Calcolare la complessità asintotica del metodo `g` nei termini del parametro `N`.

Esercizio 3 (5 punti)

L'azienda vinicola "Regali DiVini" è famosa per le sue preziosissime scatole regalo, contenenti bottiglie del vino prodotto dall'azienda stessa. Il nuovo proprietario ha deciso di rivedere il sistema informativo dell'azienda stessa, informatizzando il procedimento di creazione delle scatole regalo. A tal scopo, ogni bottiglia di vino prodotta dall'azienda viene catalogata all'interno di un computer, memorizzandone il nome del vino, l'annata e la gradazione alcolica. Si scriva una classe `Bottiglia` per l'azienda vinicola "Regali DiVini" 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 la descrizione della bottiglia.
4. Possieda il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Bottiglia` (l'uguaglianza va verificata unicamente sul nome e sull'annata).
5. Implementi l'interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con un oggetto `Bottiglia` passato come parametro (la precedenza va verificata per ordine alfabetico sul nome e, in caso di parità, si procede per annata crescente).

Esercizio 4 (8 punti)

Si scriva una classe `Scatola` che memorizzi le informazioni relative alle scatole regalo ed alle bottiglie ivi incluse. Oltre al nome della scatola ed al prezzo, le bottiglie vanno inseriti all'interno di un vettore. La classe `Scatola` deve inoltre:

1. Presentare un opportuno costruttore (il numero massimo di bottiglie che la scatola è in grado di contenere deve essere passato come parametro).
2. Presentare opportuni metodi che permettano di accedere alle variabili di istanza dell'oggetto.
3. Possedere il metodo `toString` che fornisca la descrizione della scatola (inclusa la descrizione di tutte le bottiglie contenute).
4. Possedere il metodo `aggiungi` che, dato un oggetto `Bottiglia`, lo inserisca all'interno del vettore, controllando che tale inserimento sia possibile.
5. Presentare il metodo `cerca` che, dato il nome di un vino, restituisca l'oggetto `Bottiglia` di tale vino meno giovane, se almeno un esemplare di tale vino è presente nella scatola.

Esercizio 5 (7 punti)

Si scriva un'applicazione per l'azienda vinicola "Regali DiVini" che:

1. Crei un insieme di oggetti `Scatola`.
2. Crei un oggetto `Scatola`, lette da tastiera le informazioni necessarie.
3. Inserisca l'oggetto di cui al punto 2. all'interno dell'insieme di cui al punto 1, verificando che tale insieme non contenga già un oggetto uguale.
4. Crei un oggetto `Bottiglia`, lette da tastiera le informazioni necessarie, e lo inserisca all'interno della scatola di cui al punto 2., controllando che tale inserimento sia possibile.
5. Letto da tastiera il nome di un vino, stampi a video il nome ed il prezzo di tutte le scatole che contengono un vino con tale nome.
6. Stampi a video la descrizione della scatola più costosa tra quelle contenute nell'insieme di cui al punto 1.

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:

- `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	o 2
i<M	M/2+1	o M/2+3/2
sum+=V[++i]	M/2	o M/2+1/2
i++	M/2	o M/2+1/2
Totale	3 + 3M/2	o 9/2 + 3M/2

Domanda 2:

2 assegnamenti	2
j>0	N/2+3/2
sum+=f(V, --j)	N/2+1/2
j--	N/2+1/2
complessità di f	$3N^2/8 + 3N/2 + 9/8$
Totale	$3N^2/8 + 3N + 45/8$

$$\text{complessità di f: } \sum_{\substack{j=0 \\ (j \text{ pari})}}^{N-1} \left(3 + 3 \frac{j}{2} \right) = \sum_{i=0}^{\frac{N-1}{2}} (3 + 3i) = 3 \frac{N+1}{2} + \frac{3(N+1)(N-1)}{2} = 3 \frac{N^2}{8} + 3 \frac{N}{2} + \frac{9}{8}$$

Domanda 3:

Complessità asintotica: $O(N^2)$

Soluzione Esercizio 3

```
class Bottiglia implements Comparable<Bottiglia> {
    private String nome;
    private int annata;
    private float gradazione;

    public Bottiglia(String nome, int annata, float gradazione) {
        this.nome=nome;
        this.annata=annata;
        this.gradazione=gradazione;
    }

    public String getNome() { return nome; }
    public int getAnnata() { return annata; }
    public float getGradazione() { return gradazione; }

    public String toString() {
        return nome + " " + annata + "(" + gradazione + ")";
    }

    public boolean equals(Object o) { return equals((Bottiglia) o); }
    public boolean equals(Bottiglia b) {
        return nome.equals(b.nome)&&(annata==b.annata);
    }

    public int compareTo(Bottiglia b) {
        int ret=this.nome.compareTo(b.nome);
        if(ret==0) ret=this.annata-b.annata;
        return ret;
    }
}
```

Soluzione Esercizio 4

```
import java.util.*;

class Scatola {
    private String nome;
    private float prezzo;
    private Bottiglia bottiglie[];
    private int quante;

    public Scatola(String nome, float prezzo, int n) {
        this.nome=nome;
        this.prezzo=prezzo;
        this.quante=0;
        bottiglie=new Bottiglia[n];
    }

    public String getNome() { return nome; }
    public float getPrezzo() { return prezzo; }
    public String toString() {
        String s=nome + " (" + prezzo + "): ";
        for(int i=0; i<quante; i++) s+=bottiglie[i] + ", ";
        return s;
    }

    public boolean aggiungi(Bottiglia b) {
        if(quante==bottiglie.length) return false;
        bottiglie[quante++]=b;
        return true;
    }

    public Bottiglia cerca(String nome) {
        Bottiglia b=null;
        for(int i=0; i<quante; i++)
            if(bottiglie[i].getNome().equals(nome)&&
                ((b==null)||bottiglie[i].getAnnata(<b.getAnnata())))
                b=bottiglie[i];
        return b;
    }
}
```

Soluzione Esercizio 5

```
import java.util.*;
import fiji.io.*;

class Applicazione {
    public static void main(String[] args) {
        Set<Scatola> s=new TreeSet<Scatola>(); // domanda 1
        Scatola b=new Scatola(Lettore.in.leggiLinea(), Lettore.in.leggiFloat(),
            Lettore.in.leggiInt()); // domanda 2
        if(!s.add(b)) System.out.println("Scatola già esistente!"); // domanda 3
        b.aggiungi(new Bottiglia(Lettore.in.leggiLinea(),Lettore.in.leggiInt(),
            Lettore.in.leggiFloat())); // domanda 4
        String nome=Lettore.in.leggiLinea();
        Scatola c=null;
        for(Scatola box:s) {
            if(box.cerca(nome)!=null)
                System.out.println(box.getNome() + " " + box.getPrezzo());
            if((c==null)||box.getPrezzo(>c.getPrezzo())) c=box;
        } // domanda 5
        System.out.println(c); // domanda 6
    }
}
```