

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

Appello del 18/12/2007

Esercizio 1 (4 punti)

Gestione ed utilizzo di collezioni ed iteratori in Java.

Esercizio 2 (6 punti)

Siano date le seguenti funzioni C:

```
int f(int V[], int M, int N) {
    int res=0;
    for(i=N; i<=M; i++) {
        if(V[i]>0) res+=V[i];
        else res-=V[i];
    }
    return res;
}

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

1. Calcolare la complessità in passi base (nel caso peggiore) della funzione `f` nei termini dei parametri `M` ed `N`.
2. Calcolare la complessità in passi base (nel caso peggiore) della funzione `g` nei termini del parametro `N`.
3. Calcolare la complessità asintotica della funzione `g` nei termini dei parametri `N`.

Esercizio 3 (6 punti)

Per affrontare il lavoro di impacchettamento dei regali e caricamento delle slitte, Babbo Natale ha deciso di assumere alcuni elfi come lavoratori stagionali. A tal fine, ogni elfo viene caratterizzato, oltre che dal nome e dall'indirizzo di residenza, da una valutazione della sua precedente storia lavorativa (espressa tramite un voto tra 1 e 10) e dal salario richiesto dall'elfo stesso.

Si scriva una classe `Elfo` per Babbo Natale che:

1. Possieda un opportuno costruttore con parametri.
2. Presenti opportuni metodi per accedere alle proprietà dell'elfo.
3. Presenti il metodo `toString` che fornisca una descrizione testuale dell'elfo.
4. Possieda il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Elfo` (l'uguaglianza va verificata sul nome e sull'indirizzo).
5. Implementi l'interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con un altro oggetto `Elfo`; in particolare, un elfo ha precedenza sull'altro se e solo se ha un voto maggiore e richiede un salario inferiore.

Esercizio 4 (8 punti)

Si scriva una classe `Squadra` che memorizzi le informazioni riguardanti le squadre di elfi assegnati alle slitte. Per ogni squadra occorre memorizzare, oltre all'insieme di elfi partecipanti, la targa della slitta ed il budget disponibile. La classe `Squadra` deve inoltre:

1. Possedere un opportuno costruttore (all'inizio la squadra non contiene alcun elfo, quindi il budget iniziale è completamente disponibile).
2. Presenti un metodo `getTarga` che permetta di accedere alla targa della slitta.
3. Presentare un metodo `residuo` che indichi il budget residuo (ovvero decurtato dai salari degli elfi) per la squadra.
4. Possedere un metodo `aggiungi` che, dato un oggetto `Elfo`, lo inserisca all'interno della squadra, se il budget massimo lo permette.
5. Presentare il metodo `toString` che restituisca una descrizione della squadra, compresi tutti gli elfi.
6. Possedere il metodo `capoSlitta` che, dato un oggetto `Elfo`, indichi se tale elfo precede tutti gli elfi della squadra.

Esercizio 5 (6 punti)

Si scriva un'applicazione per Babbo Natale che:

1. Crei un vettore di alcuni oggetti di tipo `Squadra`.
2. Lette da tastiera le informazioni relative ad una nuova squadra, provveda ad inserire tale oggetto in testa al vettore.
3. Lette da tastiera le informazioni relative ad un nuovo elfo, provveda ad inserire tale oggetto all'interno della prima squadra del vettore in grado di accoglierlo, indicando se non esiste alcuna squadra con budget residuo.
4. Lette da tastiera le informazioni relative ad un elfo, stampi su video la targa delle slitte per cui tale elfo può svolgere il ruolo di capo slitta.
5. Stampi su video le informazioni relative a tutte le squadre del vettore.

Per la lettura di dati da tastiera è possibile utilizzare l'oggetto `Letttore.in`, definito all'interno del package `fiji.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	$M - N + 2$
if ($V[i] > 0$)	$M - N + 1$
aggiornamento di res	$M - N + 1$
i++	$M - N + 1$
Totale	$4M - 4N + 7$

Domanda 2:

1 assegnamento	1
while ($j > 0$)	$N + 1$
chiamata di f	N
complessità di f	$2N^2 + 5N$
Totale	$2N^2 + 7N + 2$

Domanda 3:

Complessità asintotica: $O(N^2)$

Soluzione Esercizio 3

```
class Elfo implements Comparable<Elfo> {
    private String nome, indirizzo;
    private int voto, salario;

    public Elfo(String nome, String indirizzo, int voto, int salario) {
        this.nome=nome;
        this.indirizzo=indirizzo;
        this.voto=voto;
        this.salario=salario;
    }

    public String getNome() { return nome; }
    public String getIndirizzo() { return indirizzo; }
    public int getVoto() { return voto; }
    public int getSalario() { return salario; }

    public String toString() {
        return nome+", "+indirizzo+" ("+voto+", "+salario+")";
    }

    public boolean equals(Object o) { return equals((Elfo) o); }
    public boolean equals(Elfo e) {
        return (nome.equals(e.nome)&&indirizzo.equals(e.indirizzo));
    }

    public int compareTo(Elfo e) {
        if((voto>e.voto)&&(salario<e.salario)) return -1;
        if((voto<e.voto)&&(salario>e.salario)) return 1;
        return 0;
    }
}
```

Soluzione Esercizio 4

```
import java.util.*;
class Squadra {
    private String targa;
    private Set<Elfo> s;
    private int budget;

    public Squadra(String targa, int budget) {
        this.targa=targa;
        this.budget=budget;
        this.s=new HashSet<Elfo>();
    }
    public String getTarga() { return targa; }
    public int residuo() { return budget; }

    public boolean aggiungi(Elfo e) {
        if(budget<e.getSalario()) return false;
        s.add(e);
        budget-=e.getSalario();
        return true;
    }
    public boolean capoSlitta(Elfo e) {
        for(Elfo o:s) if(!(e.compareTo(o)<0)) return false;
        return true;
    }

    public String toString() {
        return "Targa: "+targa+"\n"+s+"\nBudget residuo: "+budget;
    }
}
```

Soluzione Esercizio 5

```
import fiji.io.*;
class Applicazione {
    public static void main(String[] args) {
        int n=Lettore.in.leggiInt(), quanti=0;
        Squadra[] v=new Squadra[n]; // domanda 1
        v[0]=new Squadra(Lettore.in.leggiLinea(), Lettore.in.leggiInt());
        quanti++; // domanda 2
        Elfo e=new Elfo(Lettore.in.leggiLinea(), Lettore.in.leggiLinea(),
            Lettore.in.leggiInt(), Lettore.in.leggiInt());
        unsigned int i=0;
        while((i<quanti)&&!(v[i].aggiungi(e))) i++;
        if(i==quanti)
            System.out.println("Nessuna squadra disponibile!"); // domanda 3
        e=new Elfo(Lettore.in.leggiLinea(), Lettore.in.leggiLinea(),
            Lettore.in.leggiInt(), Lettore.in.leggiInt());
        for(i=0; i<quanti; i++)
            if(v[i].capoSlitta(e))
                System.out.println(v[i].getTarga()); // domanda 4
        for(i=0; i<quanti; i++)
            System.out.println(v[i]); // domanda 5
    }
}
```