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

Appello del 12/6/2015

Esercizio 1 (4 punti)

Discutere le principali differenze esistenti tra gli array e le collezioni in Java.

Esercizio 2 (6 punti)

Siano dati i seguenti metodi Java:

```
public static int f(int V[], int N) {
  int i=N, sum=0;
  while (i-->1)
    sum+=V[--i];
  return sum;
}

public static int g(int V[], int N) {
  int j=N, sum=0;
  for (; j>0; )
    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 3 (5 punti)

Il motoclub "El Guapo" organizza motoraduni ed escursioni in tutta Italia. Visto il successo riscosso dalle proprie iniziative, ha deciso di informatizzare la gestione dei motocicli iscritti al club. A tal scopo, per ogni motociclo sono memorizzati il nome del proprietario, la targa, la cilindrata e l'anno d'immatricolazione. Si scriva una classe `Motociclo` per il motoclub "El Guapo" 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 motociclo.
4. Possieda il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Motociclo` (la verifica va fatta unicamente sulla targa).
5. Implementi l'interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con un oggetto `Motociclo` passato come parametro (in ordine crescente di anno di immatricolazione e, a parità, in ordine decrescente di cilindrata).

Esercizio 4 (8 punti)

Si scriva una classe `Evento` che memorizzi le informazioni riguardanti le iniziative del motoclub. Per ciascun evento occorre memorizzare il titolo, la data di svolgimento, il luogo, mentre i motocicli iscritti all'evento vanno memorizzati all'interno di un insieme. La classe `Evento` deve:

1. Presentare un opportuno costruttore con parametri (inizialmente, all'evento non è iscritto alcun motociclo).
2. Possedere opportuni metodi che permettano di accedere alle variabili d'istanza dell'oggetto.
3. Presentare il metodo `toString` che fornisca la descrizione dell'evento (inclusa la descrizione di tutti i motocicli iscritti).
4. Possedere il metodo `equals` per stabilire l'uguaglianza con un altro oggetto `Evento` (la verifica va effettuata su titolo e data).
5. Presentare il metodo `aggiungi` che, dato un oggetto `Motociclo`, lo inserisca all'interno dell'insieme, controllando che tale inserimento sia possibile.
6. Possedere il metodo `monocilindrata` che indichi se tutti i motocicli che partecipano all'evento sono della stessa cilindrata o meno.
7. Presentare il metodo `vintage` che restituisca il motociclo con l'anno d'immatricolazione minore tra tutti quelli iscritti all'evento.

Esercizio 5 (8 punti)

Si scriva un'applicazione per il motoclub "El Guapo"che:

1. Crei una lista di oggetti `Evento`.
2. Crei un oggetto `Evento`, 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 `Motociclo`, lette da tastiera le informazioni necessarie.
5. Inserisca l'oggetto creato al punto 4. in tutti gli eventi della lista di cui al punto 1., indicando il titolo degli eventi cui il motociclo era già iscritto.
6. Crei un insieme di eventi e vi inserisca tutti gli eventi, tra quelli della lista di cui al punto 1., che vedono partecipare solamente motocicli della stessa cilindrata.
7. Tra tutti gli eventi dell'insieme di cui al punto 6., stampi a video il nome del proprietario del motociclo avente l'anno di immatricolazione meno recente.

Soluzione Esercizio 2

Domanda 1:

2 assegnamenti	2	o 2
`i-->1`	N/2 + 1	o (N + 1)/2
`sum+=V[--i]`	N/2	o (N – 1)/2
Totale	N + 3	o N + 2

Domanda 2:

2 assegnamenti	2
`j>0`	N + 1
`sum+=f(V, --j)`	N
complessità di `f`	$N^2/2 + 2N + 1/2$
Totale	$N^2/2 + 4N + 7/2$

complessità di `f`: $\sum_{\substack{j=0 \\ (j \text{ pari})}}^{N-1} (j+3) + \sum_{\substack{j=0 \\ (j \text{ dispari})}}^{N-1} (j+2) = \sum_{j=0}^{N-1}(j+2) + \sum_{\substack{j=0 \\ (j \text{ pari})}}^{N-1} 1 = \frac{N^2}{2} + 2N + \frac{1}{2}$

Domanda 3:

Complessità asintotica: $O(N^2)$

Soluzione Esercizio 3

```
class Motociclo implements Comparable<Motociclo> {
  private String nome, targa;
  private int cilindrata, anno;

  public Motociclo(String nome, String targa, int cilindrata, int anno) {
    this.nome = nome;
    this.targa = targa;
    this.cilindrata = cilindrata;
    this.anno = anno;
  }

  public String getNome() { return nome; }
  public String getTarga() { return targa; }
  public int getCilindrata() { return cilindrata; }
  public int getAnno() { return anno; }

  public String toString() {
    return nome + ", " + targa + " (" + cilindrata + "): " + anno;
  }

  public boolean equals(Object o) { return equals((Motociclo) o); }
  public boolean equals(Motociclo c) { return this.targa.equals(c.targa); }

  public int compareTo(Motociclo c) {
    int ret = this.anno - c.anno;
    if(ret==0) ret = c.cilindrata - this.cilindrata;
    return ret;
  }
}
```

Soluzione Esercizio 4

```
import java.util.*;
class Evento {
  private Set<Motociclo> s;
  private String titolo, luogo, data;
  public Evento(String titolo, String luogo, String data) {
    this.titolo = titolo;
    this.luogo = luogo;
    this.data = data;
    s = new HashSet<Motociclo>();
  }
  public String getTitolo() { return titolo; }
  public String getLuogo() { return luogo; }
  public String getData() { return data; }
  public String toString() {
    return titolo + ", " + luogo + "( " + data + "):" + s.toString();
  }
  public boolean equals(Object o) { return equals((Evento) o); }
  public boolean equals(Evento e) {
    return this.titolo.equals(e.titolo) && this.data.equals(e.data);
  }
  public boolean aggiungi(Motociclo m) { return s.add(m); }
  public boolean monocilindrata() {
    Motociclo m=null;
    for(Motociclo x: s) {
      if(m==null) m=x;
      else if(x.getCilindrata() != m.getCilindrata()) return false; }
    return true;
  }
  public Motociclo vintage() {
    Motociclo m=null;
    for(Motociclo x: s) if(m==null || x.getAnno()<m.getAnno()) m = x;
    return m;
  }
}
```

Soluzione Esercizio 5

```
import java.util.*;
import fiji.io.*;
class Applicazione {
  public static void main(String[] args) {
    List<Evento> l = new LinkedList<Evento>();
    Scanner scanner = new Scanner(System.in);
    Evento e = new Evento(Lettore.in.leggiLinea(), Lettore.in.leggiLinea(),
        Lettore.in.leggiLinea());
    l.add(e);
    Motociclo m = new Motociclo(Lettore.in.leggiLinea(),
     Lettore.in.leggiLinea(),Lettore.in.leggiInt(), Lettore.in.leggiInt());
    for(Evento x: l) if(!x.aggiungi(m)) System.out.println(x.getTitolo());
    Set<Evento> s = new TreeSet<Evento>();
    for(Evento x: l) if(x.monocilindrata()) s.add(x);
    Motociclo min = null;
    for(Evento x: s) {
      Motociclo v = x.vintage();
      if(min==null || v.getAnno() < min.getAnno())
        min = v;
    }
    System.out.println(min);
  }
}
```