# Esame di Fondamenti di Informatica T-1/T-A Ingegneria Gestionale (A-K)

## Appello del 19/10/2018

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

#### Esercizio 1 (4 punti)

- 1. Definire il concetto di complessità asintotica di un algoritmo.
- 2. Descrivere l'utilizzo degli iteratori per la visita di collezioni in Java.

### Esercizio 2 (2 punti)

Rappresentare in binario il numero **-19,625** supponendo di utilizzare 8 bit per la mantissa (in modulo e segno) ed 8 bit per l'esponente (in complemento a 2). Si riconverta infine il numero rappresentato in base 10, motivando eventuali differenze con il numero originale.

#### Esercizio 3 (5 punti)

Siano dati i seguenti metodi Java:

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

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

- 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).
- 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 q nei termini del parametro N.

#### Esercizio 4 (5 punti)

Gino Foresta è un rocker che si diletta a organizzare eventi musicali presso il suo circolo "Tranquilla 90". In previsione del prossimo festival "Rock Shock" ha intenzione di formare diversi gruppi musicali mandando sul palco i vari musicisti che frequentano la sala prove del circolo. Per questo, Gino ha deciso di informatizzare la gestione dei musicisti invitati memorizzandone il nome, lo strumento suonato e la data di nascita. Si scriva una classe Musicista per Gino Foresta 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 musicista.
- 4. Possieda il metodo equals per stabilire l'uguaglianza con un altro oggetto Musicista (la verifica va fatta su nome e data di nascita).
- 5. Implementi l'interfaccia Comparable, definendo il metodo compareTo per stabilire la precedenza con un oggetto Musicista passato come parametro (per età decrescente e, a parità, in ordine alfabetico per nome).

#### Esercizio 5 (7 punti)

Si scriva una classe Gruppo che registri le informazioni riguardanti i gruppi che si esibiranno sul palco. Per ogni gruppo occorre memorizzare il nome mentre i musicisti caricati vanno memorizzati all'interno di un insieme. La classe Gruppo deve:

- 1. Presentare un opportuno costruttore con parametri (inizialmente, l'insieme dei musicisti 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 del gruppo (inclusa la descrizione di tutti i musicisti).
- 4. Possedere il metodo equals per stabilire l'uguaglianza con un altro oggetto Gruppo (la verifica va effettuata unicamente sul nome).
- 5. Presentare il metodo aggiungi che, dato un oggetto Musicista, lo inserisca all'interno dell'insieme, controllando che tale inserimento sia possibile.
- Possedere il metodo strumento che, data una stringa, indichi se all'interno della gruppo è presente un musicista che suoni tale strumento.
- 7. Possedere il metodo piuAnziano che restituisca il musicista più anziano del gruppo.

#### Esercizio 6 (7 punti)

Si scriva un'applicazione per Gino Foresta che:

- 1. Crei una lista di oggetti Gruppo.
- 2. Crei un oggetto Gruppo, 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 Musicista, lette da tastiera le informazioni necessarie.
- 5. Inserisca l'oggetto di cui al punto 4. all'interno del gruppo di cui al punto 2.
- Letto da tastiera uno strumento, stampi a video il nome di tutti i gruppi che non hanno un musicista che suoni tale strumento).
- 7. Stampi a video la descrizione del musicista più anziano tra tutti i gruppi.

#### Soluzione Esercizio 2

 $-19.625_{10} = 10011.101_2$  quindi la mantissa è (1)1001110, l'esponente  $5_{10} = 00000101$ .

Il numero rappresentato è -19,5, diverso dall'originale poiché quest'ultimo possiede un numero di cifre significative maggiore di 7.

#### Soluzione Esercizio 3

Domanda 1:			Domanda 2:	
2 assegnamenti	2	o 2	2 assegnamenti	2
i>0	N/2 + 1	o(N-1)/2+1	sum+=f(V, j++)	(N + 1)/2
sum+=V[i]	N/2	o(N-1)/2	++j <n< td=""><td>(N + 1)/2</td></n<>	(N + 1)/2
Totale	N + 3	o N + 2	complessità di f	$N^2/4 + 3N/2 + 5/4$
			Totale	$N^2/4 + 5N/2 + 17/4$
complessità. di f: $\sum_{\substack{j=0\\j \text{ pari}}}^{N-1} (j+3) = \sum_{\substack{i=0\\j \text{ pari}}}^{\frac{N-1}{2}} (2i+3) = \frac{(N+1)}{2} \frac{(N-1)}{2} + 3 \frac{N+1}{2} = \frac{N^2}{4} + \frac{3N}{2} + \frac{5}{4}$				
Domanda 3:				

Complessità asintotica:  $O(N^2)$ 

#### Soluzione Esercizio 4

```
class Musicista implements Comparable<Musicista> {
  private String nome, strumento:
  private int q, m, a;
  public Musicista (String nome, String strumento, int g, int m, int a) {
    this.nome = nome:
    this.strumento = strumento;
    this.q = q; this.m = m; this.a = a;
  public String getNome() { return nome; }
  public String getStrumento() { return strumento; }
  public String getData() { return g + "/" + m + "/" + a; }
  public String toString() {
    return nome + ", " + strumento + " (" + getData() + ")";
  public boolean equals(Object o) { return equals((Musicista) o); }
  public boolean equals (Musicista m) {
    return this.nome.equals(m.nome) && this.getData().equals(m.getData());
  public int compareTo(Musicista m) {
    int ret = m.a-this.a;
    if(ret==0) ret = m.m-this.m;
    if(ret==0) ret = m.g-this.g;
   if (ret==0) ret = this.nome.compareTo(m.nome);
    return ret;
```

#### Soluzione Esercizio 5

```
import java.util.*:
class Gruppo {
  private String nome;
  private Set<Musicista> s;
  public Gruppo (String nome) {
    this. nome = nome;
    s = new TreeSet<Musicista>();
  public String getNome() { return nome; }
  public String toString() { return nome + ": " + s;}
  public boolean equals(Object o) { return equals((Gruppo) o); }
  public boolean equals(Gruppo q) { return this.nome.equals(q.nome); }
  public boolean aggiungi(Musicista m) { return s.add(m); }
  public boolean strumento(String s) {
    for (Musicista m: this.s) if (m.getStrumento().equals(s)) return true;
    return false:
  public Musicista piuAnziano() {
    Musicista anziano=null:
    for (Musicista m: s)
      if(anziano==null || m.compareTo(anziano)<0) anziano=m;</pre>
    return anziano:
```

#### Soluzione Esercizio 6

```
import java.util.*;
class Applicazione {
  public static void main(String[] args) {
    List<Gruppo> 1 = new LinkedList<Gruppo>();
    Scanner scanner = new Scanner(System.in);
    Gruppo g = new Gruppo(scanner.nextLine());
    1.add(q);
    Musicista m = new Musicista(scanner.nextLine(), scanner.nextLine(),
         scanner.nextInt(), scanner.nextInt(), scanner.nextInt());
    if(!q.aqqiunqi(m)) System.out.println("Inserimento non avvenuto!");
    String strumento = scanner.nextLine();
    for (Gruppo x: 1)
      if(!x.strumento(strumento)) System.out.println(x.getNome());
    Musicista anziano = null;
    for(Gruppo x: 1) {
      Musicista ag = x.piuAnziano();
      if(anziano==null || ag.compareTo(anziano)<0) anziano=ag;
    System.out.println(anziano.toString());
```