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

Appello del 17/1/2020

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

Esercizio 1 (4 punti)

- 1. Discutere il concetto di ereditarietà.
- Illustrare le fasi di realizzazione di un programma Java (compresa l'esecuzione del programma stesso).

Esercizio 2 (2 punti)

Rappresentare in binario il numero **-21,625** 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 M, int N) {
   int i=M, sum=0;
   while(i<N)
      sum+=V[++i];
   return sum;
}

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

- 1. Calcolare la complessità in passi base del metodo f nei termini dei parametri M e N (si distinguano i casi in cui M assume valori minori di N da quelli in cui assume valori maggiori o uguali).
- 2. Calcolare la complessità in passi base del metodo g nei termini dei parametri M e N (si supponga M maggiore di N).
- 3. Calcolare la complessità asintotica del metodo q nei termini dei parametri M e N.

Esercizio 4 (5 punti)

Anche Giada, sorella di Safiria, è una ragazza dai grandi talenti. In particolare, la sua passione è il softball e il suo sogno è giocare in serie A1 nella Virtus. Nel frattempo, organizza tornei tra le classi della sua scuola. Per gestire il tutto, Giada ha deciso di registrare le informazioni su giocatrici e squadre all'interno di un calcolatore. Per questo, per ogni giocatrice occorre memorizzare il nome e il cognome, la data di nascita e il ruolo in campo (es. "ricevitore", "lanciatore", "terza base"). Si scriva una classe Giocatrice per Giada 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 della giocatrice.
- 4. Possieda il metodo equals per stabilire l'uguaglianza con un altro oggetto Giocatrice (la verifica va fatta su nome e cognome e data di nascita).
- 5. Implementi l'interfaccia Comparable, definendo il metodo compareTo per stabilire la precedenza con un oggetto Giocatrice passato come parametro (per età crescente e, a parità, in ordine alfabetico per cognome e nome).

Esercizio 5 (7 punti)

Si scriva una classe Squadra che registri le informazioni riguardanti le squadre che partecipano al torneo. Per ogni squadra occorre memorizzare la classe (es. "5C", "3A"), mentre le giocatrici vanno inserite all'interno di una lista. La classe Squadra deve:

- 1. Presentare un opportuno costruttore con parametri (inizialmente, la lista di giocatrici è vuota).
- 2. Possedere opportuni metodi che permettano di accedere alle variabili d'istanza dell'oggetto.
- Presentare il metodo toString che fornisca la descrizione della squadra (inclusa la descrizione di tutte le sue giocatrici).
- 4. Possedere il metodo equals per stabilire l'uguaglianza con un altro oggetto Squadra (la verifica va effettuata unicamente sulla classe).
- Presentare il metodo aggiungi che, dato un oggetto Giocatrice, lo inserisca in testa alla lista.
- 6. Possedere il metodo ruolo che, dato un ruolo in campo, indichi se nella squadra esiste o meno una giocatrice avente tale ruolo.
- 7. Possedere il metodo piuGiovani che, data una data, restituisca l'insieme delle giocatrici della squadra nate prima di tale data.

Esercizio 6 (7 punti)

Si scriva un'applicazione per Giada che:

- 1. Crei un insieme di oggetti Squadra.
- 2. Crei un oggetto Squadra, lette da tastiera le informazioni necessarie.
- 3. Inserisca l'oggetto di cui al punto 2. all'interno dell'insieme di cui al punto 1, controllando che tale inserimento sia possibile.
- 4. Crei un oggetto Giocatrice, lette da tastiera le informazioni necessarie.
- 5. Inserisca l'oggetto di cui al punto 4. all'interno della squadra di cui al punto 2.
- 6. Stampi a video la descrizione di tutte le squadre che non possiedono lanciatori.
- 7. Stampi a video la descrizione di tutte le giocatrici nate prima del 1/1/2008.

Soluzione Esercizio 2

 $-21,625_{10} = 10101,101_2$ quindi la mantissa è (1)1010110, l'esponente $5_{10} = 00000101$.

Il numero rappresentato \tilde{e} –21,5, diverso dall'originale poiché quest'ultimo necessita di 9 cifre binarie per la mantissa.

Soluzione Esercizio 3

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

C 1 ''

Complessità asintotica: O(MN)

Soluzione Esercizio 4

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

Soluzione Esercizio 5

```
import java.util.*;
class Squadra {
  private String classe:
  private List<Giocatrice> 1;
  public Squadra(String classe) {
    this.classe = classe;
    l = new ArrayList<Giocatrice>();
  public String getClasse() { return classe; }
  public String toString() { return classe + ": " + 1; }
  public boolean equals(Object o) { return equals((Squadra) o); }
  public boolean equals(Squadra s) { return classe.equals(s.classe); }
  public void aggiungi(Giocatrice g) { 1.add(0,g); }
  public boolean ruolo(String ruolo) {
    for (Giocatrice q: 1)
      if(g.getRuolo().equals(ruolo)) return true;
    return false;
  public Set<Giocatrice> piuGiovani(int g, int m, int a) {
    Set<Giocatrice> s = new TreeSet<Giocatrice>();
    Giocatrice x = new Giocatrice("", "", "", q, m, a);
    for (Giocatrice z: 1) if (z.compareTo(x)<0) s.add(z):
    return s;
```

Soluzione Esercizio 6

```
import java.util.*;
class Applicazione {
  public static void main(String[] args) {
    Set<Squadra> s = new HashSet<Squadra>();
    Scanner scanner = new Scanner(System.in);
    Squadra q = new Squadra(scanner.next());
    if(!s.add(q)) System.out.println("Inserimento non avvenuto!");
    Giocatrice c = new Giocatrice (scanner.nextLine(),
        scanner.nextLine(), scanner.nextLine(),
        scanner.nextInt(), scanner.nextInt());
    q.aggiungi(c);
    for(Squadra x: s)
        if(!x.ruolo("lanciatore")) System.out.println(x);
    for(Squadra x: s)
        System.out.println(x.piuGiovani(1, 1, 2008));
  }
}
```