

# Corso di Laurea in Informatica

## Prova Scritta di Metodologie di Programmazione - Primo Canale

Sapienza Università di Roma

16 Settembre 2021

Durante l'esame non è consentito l'utilizzo di alcunché. Non è consentito inoltre l'utilizzo della matita o di penne il cui colore sia diverso dal nero o dal blu. Il ritiro dalla prova equivale al mancato superamento dell'esame. Le pagine che non sono considerate utili alla valutazione vanno **barrate**.

Nome e Cognome:

Matricola:

Domande	1	2	3	4	Totale
Punti	4	2	18	8	33
Valutazione					

- (4 punti) Descrivere il meccanismo dell'overloading e overriding. Produrre un esempio **minimale** scritto in Java che concretizzi quanto spiegato.
- (2 punti) Per ogni costrutto iterativo, indicare il numero di volte per il quale viene eseguito il suo corpo. Se non diversamente indicato, si assume che la variabile contatore non venga modificata all'interno del corpo di ciascun costrutto iterativo.
  - `for(int i = 8; i >= -3; i = i + 2){...}`
  - `for(int i = 0; i < 10; i--){...}`
  - `for(int i = 10; i > 0; i++){...}`
  - `for(int i = -10; i >= 0; i--){...}`
  - `for(int i = -2; i >= 0; i--){...}`
  - `for(int i = -8; i <= 3; i = i - 4){...}`
  - `for (int k = 0; k < 20; k+=2) {  
    if (k - 3 == 1) break;  
}`

3. Si vuole realizzare un programma in Java che rispecchi il "Gioco delle sedie musicali". Tale gioco ha le seguenti caratteristiche:

- $N$  partecipanti si contendono  $N - 1$  sedie
- All'inizio di ogni turno del gioco, tutti i giocatori sono in piedi
- Viene fatta partire la musica per un numero casuale di secondi che va da 1 a 20
- Viene fatta terminare la musica e i giocatori si dovranno sedere nelle  $N - 1$  sedie disponibili. Colui che non ci riuscirà verrà eliminato
- Si ripete il turno, eliminando una sedia, finché non rimarrà un solo giocatore il quale sarà il vincitore della partita

Dal punto di vista tecnico, la lista dei partecipanti viene fornita attraverso un file di testo in cui viene riportato: il codice identificativo del giocatore, il nome, il cognome e l'età. Ciascun campo è separato dal carattere ";".

Un esempio di contenuto del file è il seguente:

```
1;Marco;Neri;35
2;Giulio;Ponte;25
3;Maria;Verdi;30
4;Paola;Rossi;18
...
```

Il programma inoltre:

- (4 punti) Dovrà decretare il numero di giocatori in base a quanti sono presenti all'interno del file, impostando coerentemente il numero di sedie necessarie per lo svolgimento del gioco
- (5 punti) Dovrà simulare l'esecuzione della musica generando un numero casuale e sospendendo l'esecuzione del programma tramite l'istruzione `Thread.sleep(int mills)`, dove *mills* è l'equivalente in millisecondi del tempo di esecuzione della musica precedentemente generato
- (5 punti) Dovrà simulare l'eliminazione del giocatore rimasto in piedi in ciascun turno scegliendolo in modo casuale. L'esclusione di tale giocatore dalla partita deve essere riportata all'interno del programma rimuovendolo dalla corrispondente struttura dati
- (4 punti) Non appena vi sarà rimasto un solo giocatore all'interno della partita, il programma dovrà stampare a video le informazioni relative a tale giocatore, terminando successivamente il programma.

**NB:** è necessario un approccio orientato agli oggetti. Si assume che l'input fornito dal file sia sempre corretto sia nella forma che nel contenuto.

4. Realizzare un programma Java il quale, dato un file di testo:

- (2 punti) Legga il suo contenuto
- (4 punti) Crei un secondo file di testo con la lista delle parole uniche, senza ripetizioni
- (2 punti) Stampi a video il numero di parole presenti nel file di input e il numero di parole uniche trovate

**NB:** la scelta della struttura dati più opportuna è una parte fondamentale ai fini della valutazione dell'esercizio. Si assume inoltre che le parole siano sempre e solo divise da uno spazio.