

Esame di Programmazione Internet

CdL in Informatica per il Management

Data Appello: 27/05/2021 - turno I

Per chi ha superato il parziale: solo esercizio 4 - tempo a disposizione: 1 ora e 1/2

Per gli altri: tutti gli esercizi - tempo a disposizione: 3h

INFORMAZIONI PRELIMINARI

L'esame consiste nello scrivere uno o più file *.java compilabili. Elaborati che **non compilano il codice prodotto non** verranno presi in considerazione.

NON è consentito l'uso di testi e appunti.

La scelta dei nomi dei file, se non espressamente indicati, non è rilevante. I file possono essere salvati in directory diverse.

All'inizio di **ogni** file .java mettete come commento, Nome, Cognome, Num Matricola. Se **non** lo farete il compito sarà annullato. Esempio:

```
/*****  
* Nome: Paolino  
* Cognome: Paperino  
* Num Matricola: 12345678  
* email: paperino@unibo.it  
*****/
```

Se il testo richiede effettuare lettura di file testuali, il candidato deve crearsene alcuni di esempio da solo (usando jEdit come editor e salvando il file con estensione .txt), **rispettando le specifiche date**.

E' **consentito** invocare qualunque metodo delle API di Java.

La prova intende valutare, oltre alla verifica che lo studente ha acquisito **competenze** nell'ambito della programmazione, la capacità del candidato di **comprendere** il testo del compito (i.e., programmi che girano ma **non** fanno quanto richiesto verranno considerati insufficienti), la capacità di inserire **commenti** al codice per facilitare la comprensione dell'elaborato, la capacità di scrivere **codice pulito e ordinato**, la capacità di **ottimizzare** la soluzione (soprattutto in termini di efficienza del programma).

NOTE TECNICHE

Nel caso (frequente) vi si dovesse bloccare il sistema di sviluppo (ex: per loop infinito), potete aprire un terminale (CTRL-ALT-t) e provare scrivere il comando "pkill java" o "pkill jedit". Se si dovesse chiudere jEdit (che è quello che dovrebbe capitare), poi scrivere "jedit &" e il sistema dovrebbe ripartire.

NOTA LOGISTICA

Agli studenti affetti da disturbi quali dislessia, è richiesto di completare tutti gli esercizi in un tempo maggiorato del 30%.

Nota per lo studente: Per sicurezza, nei commenti iniziali si scriva "**studente con DSA**".

TESTO DEL COMPITO

Esercizio 1

Creare un programma che legga da tastiera un valore intero $n > 0$ e successivamente n numeri double che andranno inseriti in un array X . Il programma deve stampare a video la media di tutti gli elementi di X che sono minori del proprio indice in X , ad es. se $n = 5$ e $X = [4.3, -2, 1.5, 6, -5.2]$ il programma deve stampare $(-2 + 1.5 - 5.2) / 3 = -1.9$

NOTA: Se ogni elemento è maggiore o uguale al proprio indice, stampare invece l'elemento avente differenza minima col proprio indice (se sono più di uno stampare il primo). Ad es. se $X = [3, 5.2, 2.03, 10, 4.03]$ deve stampare 2.03

Esercizio 2

Creare un programma che letti da tastiera due numeri N, M carichi e stampi una matrice X di dimensione $N \times M$ con valori interi casuali in $[2, 8]$. Successivamente stampare la matrice prodotto $X \cdot X'$ dove X' è la matrice trasposta di X , ad es. se $N=2, M=3$ e $X = 2, 3, 4$

5, 7, 8

Allora la trasposta è $X' = 2, 5$

3, 7

4, 8

Quindi la matrice prodotto $X \cdot X'$ da stampare è:

$$2 \cdot 2 + 3 \cdot 3 + 4 \cdot 4, \quad 2 \cdot 5 + 3 \cdot 7 + 4 \cdot 8 \quad = \quad 29, \quad 63$$

$$5 \cdot 2 + 7 \cdot 3 + 8 \cdot 4, \quad 5 \cdot 5 + 7 \cdot 7 + 8 \cdot 8 \quad = \quad 63, \quad 138$$

Esercizio 3

Scrivere un metodo ricorsivo che prende in input un vettore di interi, non vuoto, e restituisce l'indice del valore più piccolo contenuto nel vettore.

Se il valore più piccolo è duplicato restituire l'indice maggiore.

Ad esempio:

- *Input: [1, 2, -4, -5, 10, 12, 8] Output: 3*
- *Input: [1, -12, -4, -5, 10, 12, 8] Output: 1*
- *Input: [1, 2, -4, -5, 10, -8] Output: 5*
- *Input: [1, -8, -4, -5, 10, -8] Output: 5*

E' ammesso fare overloading del metodo. Non è ammesso usare metodi ausiliari.

Esercizio 4

Implementare il codice per gestire un semplice quiz da linea di comando. Il quiz si gioca in una sola persona e ha un numero di domande non predefinito.

Un quiz è gestito da una classe `Quiz` che espone i metodi:

- *costruttore*: crea un quiz vuoto; prende in input il percorso del file (o un oggetto `File`) che contiene lo storico delle partite e su cui sarà memorizzato il risultato
- *aggiungiDomanda*: aggiunge domande al quiz; le domande sono di diverso tipo, come spiegato di seguito
- *gioca*: permette all'utente di giocare, mostra cioè tutte le domande in ordine e per ognuna:
 - o chiede la risposta
 - o la controlla
 - o se corretta, aumenta il punteggio del giocatore in base ai punti della domanda
- alla fine della partita mostra il risultato ottenuto e aggiunge le informazioni della partita (nome del giocatore e punteggio) in fondo ad un file che contiene. Tipo e formato del file sono arbitrari. Il percorso del file è stato passato specificato nel costruttore dell'istanza di `Quiz`.

Il quiz prevede tre tipi di domande:

- *Domanda semplice*: il costruttore prende in input il testo della domanda e la risposta corretta; non sono mostrate le possibili risposte; la risposta è di tipo stringa.
- *Domanda rischio*: domanda semplice ma in caso di risposta sbagliata, i punti della domanda vengono sottratti a quelli del concorrente. Anche qui la risposta è una stringa.
- *Domanda a risposta multipla*: il costruttore prende in input il testo della domanda e un vettore con le possibili opzioni e la posizione nel vettore della risposta corretta; la visualizzazione della domanda mostra il testo della domanda e una riga per ogni opzione. La risposta è un valore intero.

Definire le classi e i metodi opportuni per creare le domande e verificare i risultati.

Creare una classe di test che istanzia un quiz, aggiunge le domande e permette ad un utente di giocare. Nota: non è richiesto aggiungere domande in modo interattivo ma si possono usare domande predefinite nella classe di test; l'interazione è richiesta durante la fase di gioco.