Esame di Programmazione Internet

CdL in Informatica per il Management

Data Appello: 27/04/2021 - Appello Prova

Per chi ha superato il parziale: solo esercizio 4 - tempo a disposizione: 1 ora e ½

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 e` 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

Esercizi 1, 2 e 3

[Su programma primo semestre: programmazione procedurale, array, matrici, ricorsione.]

Esercizio 4

Scrivere le classi Java per modellare la seguente situazione. Ogni persona ha un nome. Un lavoratore è una persona della quale si conosce anche numero di matricola, anno di assunzione e stipendio base (double). I lavoratori sono di tre tipi: impiegato, operaio e manager.

Lo stipendio si calcola in modo surreale: lo stipendio di un impiegato si calcola sommando allo stipendio base gli anni di assunzione moltiplicati per 100; per un operaio la regola è la stessa ma il fattore di moltiplicazione è 120; per un manager invece lo stipendio è la somma tra lo stipendio base e un bonus deciso in fase di assunzione.

Implementare una classe Squadra che memorizza un elenco di lavoratori e tutti i loro dati. I dati possono essere importati/esportati da/su file. Formato e tipi dei file sono arbitrari.

La classe Squadra espone i metodi:

- costruttore: crea una squadra vuota;
- popolaDaFile: legge i dati di una squadra da un file e aggiorna i dati della squadra. I
 dati sovrascrivono i precedenti. Il metodo prende in input il percorso di un file o
 direttamente un oggetto File. Gestire errori nell'apertura del file.
- *salvaSuFile*: salva tutti i dati della squadra su file. Il metodo prende in input il percorso di un file o direttamente un oggetto File. Gestire errori nell'apertura del file.
- aggiungiLavoratore: aggiunge un lavoratore alla squadra. Viene assegnato un nuovo numero di matricola; non è richiesto gestire omonimi, possono esserci più lavoratori con lo stesso nome
- calcolaStipendi: restituisce il totale degli stipendi percepiti dalla squadra
- *getSottoSquadra*: prende in input un valore S e restituisce la lista di lavoratori che percepiscono uno stipendio pari o maggiore di S

Per semplicità non è richiesto gestire cancellazione o aggiornamento dei dati di un lavoratore.

Verificare il funzionamento delle classi in una classe di test (con nome arbitrario).