29-01-2014

Fondamenti di Informatica 2 – Prova Scritta

Nome: …..... Cognome: …..... Matricola: ..... Postazione N. : .....

Sviluppare una semplice Interfaccia Grafica che serva a confrontare il Tempo di Esecuzione di alcune tipiche operazioni effettuate su collezioni di elementi. L'Utente, scelto uno o più operatori e selezionato a piacere un numero N da 1 a 100000, vede a schermo il Tempo di necessario ad ognuno degli operatori a confronto per eseguire su un insieme di N elementi estratti casualmente.

Attualmente, i confronti che l'interfaccia deve supportare sono i seguenti :

- **Inserimento di elementi col metodo add in (A) un ArrayList<int> e (B) una LinkedList<int>**
- **Ordinamento di (A) un ArrayList<int> con Collections.sort() e (B) un vettore int[] con Arrays.sort()**
- **Ordinamento di un ArrayList<int> (A) con Collections.sort() e (B) con una implementazione adeguata di Bubble Sort.**
- **Ordinamento di (A) un ArrayList<String> con Collections.sort() e (B) una LinkedList<String> con Collections.sort(). Usare stringhe costruite a partire da vettori di 6 caratteri, ognuno preso casualmente nell'intervallo [a-z].**
- **Ricerca in una ArrayList<int> preordinata (A) con il metodo indexof e (B) con il metodo Collections.binarySearch(). In tutti i casi si ricerchi come elemento quello fornito da Collections.max().**
- **Ricerca in una lista usando il metodo indexof in (A) una ArrayList<int> (B) in una LinkedList<int>. In tutti i casi si ricerchi come elemento quello fornito da Collections.max().**

Il calcolo del tempo di esecuzione DEVE essere effettuato usando la funzione static System.nanoTime(). Sull'interfaccia i tempi devono essere formattati in millisecondi fornendo almeno 3 cifre decimali. Es. 74,128 ms

Si tenga infine presente che le modalità di confronto che si vorranno aggiungere in futuro sono davvero tante.

1. Creare un Workspace **Eclipse**. Creare un Progetto **esame**. Dopo aver studiato il problema, implementare in **Java** una possibile soluzione modulare e ad oggetti.
2. Su foglio protocollo, **a titolo di documentazione e ai fini della valutazione**, si realizzi uno schema UML sintetico che metta in luce le relazioni che intercorrono tra i moduli implementati. E' possibile utilizzare ObjectAID UML, ma in quel caso è obbligatorio esportare gli schemi UML in formato immagine png. ***E' necessario*** **inoltre ad utilizzare la documetnazione Javadoc nel codice dove lo si ritenga opportuno**.
3. Lo studente può accedere al percorso **/home/etc/FDI2** per recuperare la documentazione **Javadoc**, i cosiddetti **esempi forniti** e altro materiale utile. E' inoltre possibile consultare qualsiasi testo scritto.
4. Alla fine dell'esame, esportare un file zip attraverso la funzionalità **Export...** di eclipse (vedi le **istruzioni di salvataggio dati**) e salvarlo come **/home/esm/esame_N/esame_N.zip** (ad esempio **/home/esm/esame_20/esame_20.zip**)

Punteggio (Totale 15+ punti)

- **6+ punti** per **l'architettura** del progetto, con particolare enfasi alla struttura del Modello.
- **3 punti** per la corretta **implementazione** in **Java** delle funzionalità del programma.
- **3 punti** in merito alla **validità** di implementazione interna ad **ogni singola classe**.
- **3 punti** sono assegnati sulla base dell'utilizzo dei componenti Java indicati nel testo.