

Fondamenti di Informatica 2 – Prova Scritta

Livello di difficoltà : **medio**

Lo studente è invitato ad analizzare il seguente Tema:

Un Editor di Icone

Una icona è una immaginetta quadrata molto piccola. Il Disegno di icone può essere difficile con strumenti grafici pensati per l'elaborazione di immagini più grandi.

Per questo, si vorrebbe realizzare un piccolo editor di icone. Ogni icona deve essere gestita come una matrice quadrata di pixel, ognuno con un colore (in formato Red-Green-Blue), e per ora si pensa di supportare le dimensioni 16x16, 24x24 e 32x32. L'interfaccia deve soddisfare questi requisiti:

- 1) L'utente può iniziare quando vuole una nuova icona, che è tutta bianca, scegliendo il formato.
- 2) L'utente può selezionare un colore.
- 3) L'utente può scegliere una delle seguenti operazioni:
 - (A) Matita: quando clicco (o trascino il mouse) su un pixel dell'icona, quel pixel si colora col colore (2)
 - (B) Gommino: quando clicco (o trascino il mouse) su un pixel, quel pixel si colora di bianco.
 - (C) Pennello colori : colora col colore (2) tutti i pixel che stanno all'interno di una certa area del pennello.
- 4) Un pannello laterale fornisce tre informazioni (costantemente aggiornate) che sono il valore minimo, medio e massimo dell'intensità di grigio valutata su tutti i pixel. La tonalità di grigio deve essere calcolata con la formula:

$$\text{grigio} = 0.3 * \text{rosso} + 0.59 * \text{verde} + 0.11 \text{ blu}$$

1. Creare un **Workspace Eclipse**. Creare un Progetto **esame**. Dopo aver studiato il problema, implementare in **Java** una interfaccia grafica per l'editor di icone; **del punto (3) è richiesto di implementare soltanto (A) e (B). Il punto (4) è facoltativo**, ma si chiede di tenerne conto di tutto nel Design dei moduli. Si consiglia l'uso delle costanti predefinite nella classe *java.awt.Color* in particolar modo per quanto riguarda il punto (2).
2. In tutti i file, usare il simbolo @author (javadoc), per indicare il proprio nome e cognome.
3. Su foglio protocollo, **a titolo di documentazione e ai fini della valutazione**, lo studente illustri con uno o più semplici schema **UML il ruolo dei design pattern** utilizzati nel progetto dell'interfaccia. E' possibile avvalersi degli schemi generati con ObjectAid, ma in tal caso è richiesto di salvare lo schema sotto forma di immagine PNG.
4. Lo studente può accedere al percorso **/home/etc/FDI2** per recuperare la documentazione Javadoc, **tutti gli esempi di codice presentati al corso** e le **istruzioni di salvataggio dati**. E' Inoltre possibile consultare qualsiasi testo scritto.
5. Alla fine dell'esame, esportare un file zip attraverso la funzionalità **Export...** di eclipse (vedi le **istruzioni di salvataggio dati**) e salvarlo come **/users/studente/esame_N.zip** (ad esempio **/users/studente/esame_20.zip**)

Punteggio (Totale 15+ punti)

- **6+ punti** per l'**architettura** del progetto, della relativa **documentazione UML** e dell'eventuale (gradita) **documentazione javadoc**.
- **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 dedicati all'uso delle librerie grafiche java.

NOTA: Non ci sono vincoli sulla modalità di rappresentazione delle icone, fare scelte semplici!