

TD-TP : Interface graphique et gestion des événements

Exercice 1

- 1- Donner la classe permettant d'afficher la fenêtre ci-dessous, sachant que la couleur affichée est donnée via la sélection d'un des boutons radios. De plus, on obtient dans le label, en bas de la fenêtre, la valeur hexadécimale de la couleur choisie



- 2- Ajouter une liste de choix de couleur. Cette liste propose les mêmes couleurs que celles données par les boutons radios. De plus, une sélection d'une couleur dans la liste fait que le bouton radio associé à cette couleur est cochée automatiquement. Si on coche un nouveau radio bouton alors la liste affichera une sélection de la couleur cochée dans les radios boutons



Remarque : Ci-dessous, la méthode permettant de retourner, sous format d'un string, la transformée en hexadécimal d'un entier donné en paramètre à la méthode. Le String retourné contient au moins de 2 chiffres:

```

private String toHex(int val){

    String result = Integer.toHexString(val).toUpperCase();

    if (result.length() == 1) return "0" + result;

    else return result;

}

```

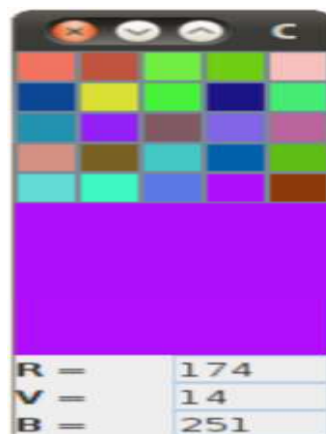
Exercice 2

Donner le code permettant de dessiner dans une fenêtre (voir figure ci-dessous), sachant que le masquage de la fenêtre provoquera la perte du dessin effectué.



Exercice 3

Construisez une application (voir figure ci-dessous) permettant à l'utilisateur de choisir des couleurs et de voir la couleur obtenue ainsi que ses intensités de rouge, vert et bleu



Exercice 4

L'objectif de cet exercice est de construire une application permettant l'affectation d'un ensemble d'étudiants à un professeur d'une matière donnée.

Chaque étudiant peut suivre le cours d'un ou plusieurs professeurs se trouvant dans la liste des professeurs. Chaque professeur peut enseigner à zéro ou plusieurs étudiants.

Construire votre application en permettant des nouveaux enregistrements des professeurs et des étudiants, s'ils ne sont pas déjà inscrits. Votre application doit être capable de lister l'ensemble des professeurs et l'ensemble des étudiants avec un trie. La sélection d'un étudiant permettra d'afficher la liste des professeurs dont il suit leur cours. La sélection d'un professeur permettra d'afficher la liste des étudiants qui suivent son cours.

La suppression d'un étudiant fera supprimer cet étudiant de la liste de tous ses professeurs.

La suppression d'un professeur fera disparaître tout étudiant s'il est son seul professeur.

La recherche d'un professeur (resp. d'un étudiant) doit se faire dynamiquement soit par son identifiant soit par une recherche dans la liste des professeurs (resp. dans la liste des étudiants)

Un double clique sur un étudiant dans la liste des étudiants fera apparaître l'ensemble des informations de l'étudiant cliqué puis en peut faire les modifications qu'il faut et finir par valider cette action. Attention, l'identifiant ne doit pas être modifié lors d'un update.

Les mêmes étapes, dans le cas d'une double clique sur un professeur, doivent se faire pour le cas rafraichissements des informations d'un professeur.

Question : donner une implémentation appropriée à cette application pour répondre aux attentes demandées.

- ➔ Les infos d'un prof : nom, prénom, Cin, GSM, matière enseignée et la liste de ses étudiants.
- ➔ Les infos d'un étudiant : nom, prénom, dateNaissance, idEtudiant, GSM, liste des profs disponibles.
- ➔ Les infos de la fenêtre panelListesProfEtudiant : liste des étudiants, liste des profs. Les deux listes doivent être triées dynamiquement.