

Feuille n° 2

Exercice n°1

On va préparer le cadre d'une application de dessin (type `SketcherFrame`) pour qu'il affiche une barre de menus et, en dessous, une zone de texte (`JTextArea`) pouvant être scrollée (on utilisera un `JScrollPane`). Cette zone de texte ne sera pas éditable mais on s'en servira dans un autre TP pour afficher des informations sur les événements se produisant sur les menus.

La barre de menus comportera 3 menus.

1) Le premier menu s'intitule 'Fichier', et présente d'abord 3 items affichant respectivement un simple texte (par ex. Ouvrir), un texte (par ex. Sauver) et une icône, et enfin, une icône seule. Vient ensuite un séparateur, puis un dernier item (par ex. Enregistrer). Pour créer un objet `Icon` (nécessaire au constructeur `JMenuItem` pour faire des items comportant une icône), on peut utiliser `new ImageIcon(String filename)` où `filename` est le nom d'un fichier `.gif` par exemple et contenant une icône.

2) Le second menu s'intitule 'Elements' et présente d'abord 4 items de type boutons radio intitulés Ligne, Rectangle, Cercle et Courbe. Vient ensuite un séparateur, puis un sous-menu intitulé Couleurs qui présentera 3 cases à cocher intitulées Rouge, Vert et Bleu. Faites en sorte que l'on ne puisse sélectionner qu'un seul bouton radio à la fois et qu'une seule case à cocher (on les ajoute à un même `ButtonGroup`).

3) Le dernier menu nommé 'Aide' ne fait rien ici, mais il se trouve complètement à droite dans la barre de menu (introduisez un composant horizontal `Glue` défini dans la classe `Box`).

4) Tous les menus de la barre de menu possèdent une touche mnémonique. On ajoutera également quelques accélérateurs pour les items du premier menu. On pourra aussi ajouter des mnémoniques aux cases à cocher et aux boutons radio du second menu. Veillez à ce qu'il n'y ait pas de conflits sur les lettres utilisées.

Exercice n°2 :

On va implanter un programme analogue à ceux qui ont été vus en cours sur les différents `LayoutManager`, et qui affichaient 6 boutons. Ici, on affichera des `JLabel` qui utiliseront 2 fontes de taille différentes (une assez grosse et une petite comme dans un des exemples du poly). En outre, les chaînes auront des longueurs différentes pour mieux tester l'exemple.

On observera dans chaque cas ce qui se passe quand on agrandit ou rétrécit la fenêtre.

1) Tester un afficheur `FlowLayout` dans une classe `TryFlowLayout` comme en cours.

- 2) Tester un afficheur `BoxLayout` en utilisant un conteneur de type `Box` qui alignera 6 boutons verticalement et sera placé au centre du `JFrame`.
- 3) Tester un afficheur de type `GridLayout`, pour placer 12 labels a) sur une seule ligne ; et b) sur 3 lignes et 4 colonnes.
- 4) Ajouter différents type de bords à vos labels dans l'un des programmes précédents. Les bords peuvent être créés à partir de la classe `BorderFactory`. On testera `BevelBorder`, `TitleBorder`, et `EmptyBorder`, puis on formera des bords composés (double bord) à l'aide de `CompoundBorder`, en associant par exemple un bord vide avec un bord titré.