université PARIS DIDEROT PARIS 7

EXAMEN INTERFACES GRAPHIQUES 25/05/2015 12^{00}-13^{30}

Positionner impérativement vos mobiles en mode « avion ».
Aucun document ou support autre que le sujet ou la documentation de l'API Java n'est autorisé.
L'API est consultable à l'adresse `http://lucien/`.
L'écran ou les fichiers du voisin ne sont pas des supports autorisés.

Attention : l'écran de votre session est entièrement enregistré durant l'examen. En cas de suspicion de fraude et uniquement dans ce cas, nous nous réservons le droit de l'examiner. L'enregistrement sera détruit un mois après l'examen.

Consignes (à lire attentivement)

Cet examen se déroulant sur machine il est demandé de procéder de la façon suivante :

- Pour chaque question numéro n, il faut écrire une classe Java dont le nom **doit** comprendre nom de famille et n. Par exemple, pour un étudiant qui s'appellerait **tartempion** qui résout la question n°3, la classe doit s'appeler `Classe_Tertempion_3` ;
- Le programme principal répondant à la question n°3 commencera donc par :

```
public class Classe_Tartempion_3 {
  public static void main(String []args) {
    ...
```

- Le sujet étant itératif, pour répondre à la question $n+1$ et construire la classe $n+1$, il suffit de prendre une **copie** de la classe n (réponse à la question n), de la **renommer** et la **modifier**.
- Notez qu'il n'y a pas de piège, qu'il n'est pas demandé de réaliser des choses très complexes, faites donc (presque) au plus simple (mais pas n'importe quoi tout de même !) ; toutefois pour vous faire une idée correcte, lisez donc l'intégralité du sujet avant de commencer ; ceci afin de ne pas vous positionner sur une fausse piste ;
- Lorsque vous pensez que vous avez correctement répondu à une question, **faites appel à un enseignant présent** en lui faisant signe ou en l'interpellant discrètement ; en attendant qu'il vienne (il peut être occupé) passez à la question suivante, mais insistez pour qu'il vienne ; n'attendez pas la fin de l'examen !
- L'enseignant **viendra valider** votre réponse en testant lui-même le programme (il sera maître de la souris et du clavier à cet instant) et en examinant votre code si nécessaire.
- Si l'enseignant valide votre réponse à une question, copiez immédiatement le fichier source dans le répertoire `/info/ens/yunes/IG`.

Sujet

1. Écrire un programme qui, lorsque exécuté, permet d'obtenir l'affichage d'une fenêtre ressemblant à celle de la figure 1. Les composants présents sont des labels dont le texte doit être placé respectivement à gauche, au milieu et à droite (y compris lorsqu'on retaille la fenêtre). Il faut aussi reproduire approximativement les couleurs de la figure, à savoir, gris foncé, gris moyen et gris clair. Il est conseillé de créer une classe qui lorsqu'instanciée permettra d'obtenir un `JPanel` contentant les trois `JLabels`.

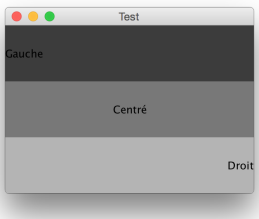

FIGURE 1 – Écran 1

2. Ajouter une barre de menu avec un menu intitulé `Fichier`. Le menu contiendra une option `Quitter` qui permettra de terminer l'application et à laquelle sera associée un raccourci clavier adéquat.
Indice : `setAccelerator`.

3. Modifier la fenêtre principale de sorte qu'on y trouve deux onglets `Panneaux` et `Bouton`. Le premier faisant apparaître un panneau contenant les trois labels (comme en première question) et le second un panneau contenant un simple bouton. Voir figure 2. Le bouton affichera le texte `Clique-moi!`.
Indice : `JTabbedPane`.

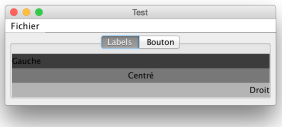

FIGURE 2 – Écran 2

4. Faire en sorte que lorsqu'on clique sur le bouton, un dialogue modal apparaisse affichant `Tout va bien`.
   Indice : `JOptionPane`.

5. Rajouter dans le tabbedpane un panneau intitulé `Rectangles` qui lorsqu'il est présent à l'écran permet d'afficher toutes les $\frac{1}{2}$ secondes un rectangle rouge. Le rectangle aura une taille aléatoire et sera aussi localisée aléatoirement dans l'espace du panneau (quelque soit celle-ci).
   Indice : `Random.nextInt()`, `Graphics.fillRect()`, `Thread`
6. Modifier le panneau `Rectangles` de sorte que le nombre de rectangles générés soit affiché en temps réel dans son coin supérieur droit.
7. Modifier l'application de sorte que d'une exécution à l'autre, le nombre de rectangles générés soit conservé.
   Indice : `java.util.prefs.Preferences`.
8. Internationaliser l'application de sorte que tous les textes puissent être modifiés sans avoir à recompiler l'application. Fournir deux localisations différentes.
   Indice : `ResourceBundle`.