

# Examen « Interfaces Graphiques »

## Master 1 Informatique Paris Diderot

Jean-Baptiste Yunès

Session 1 - 2011-2012 - Sujet n°2

### Consignes

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

- Pour chaque question, il faut écrire une classe Java correspondante (Classe1, Classe2, etc.);
- Le programme principal est constitué d'une simple classe se contentant d'instancier la classe correspondante :

```
public class Prog {  
    public static void main(String []args) {  
        new Classe1(); // ou new Classe2(); etc...  
    }  
}
```

- Le sujet étant itératif, pour construire la Classe $i+1$ , il suffit de prendre une copie de la Classe $i$  et de 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 !); toutefois pour vous faire une idée correcte, lisez donc l'intégralité du sujet avant de commencer; ceci afin de ne pas vous mettre sur une fausse piste;
- Lorsque vous pensez que vous avez correctement répondu à une question, **faites appel à l'enseignant** en lui faisant signe ou en l'interpellant discrètement; en attendant qu'il vienne (il peut être occupé) passez aux questions suivantes;
- 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 la question, envoyez **immédiatement** le programme Java, en pièce jointe, à l'adresse suivante et en n'oubliant pas d'indiquer, dans le corps du message, vos nom et prénom :

Jean-Baptiste.Yunes@liafa.univ-paris-diderot.fr

## Sujet

1. écrire un programme qui, lorsque exécuté, ouvre un dialogue contenant un message informatif *Bonjour, je suis votre-nom*. Voulez-vous continuer ? et deux boutons Oui et Non (on pensera à utiliser la méthode `show*Dialog` adéquate de la classe `JOptionPane`).
2. ajouter une fonctionnalité de sorte qu'un clic sur le bouton Non quittera l'application et qu'un clic sur le bouton Oui ouvrira une fenêtre contenant un menu `Fichier` contenant un seul item `Quitter` qui lorsque sélectionné permettra de quitter l'application.
3. ajouter un second menu de nom `Dessin` qui contiendra trois `JCheckBoxMenuItem` `Triangle` (sélectionné au départ), `Cercle` (dé-sélectionné au départ) et `Continu` (dé-sélectionné au départ). Faire en sorte que la sélection de l'item `Cercle` dé-sélectionne automatiquement l'item `Triangle` et vice-versa.
4. créer un composant personnalisé (une sous-classe de `JComponent`) dont la fonctionnalité principale sera de remplir l'intégralité de l'espace qui lui est alloué avec sa propre couleur de fond (blanc par défaut). Instancier ce composant et l'insérer dans la partie « principale » de la fenêtre (cette partie devra occuper le plus d'espace possible).
5. modifier le composant personnalisé de sorte qu'un clic dessine avec la couleur noire, un triangle plein (si l'on avait sélectionné `Triangle`) dont les dimensions seront de votre choix ou un cercle (pas un disque ! si l'on avait sélectionné `Cercle`) de diamètre 20 pixels. Les objets dessinés le seront à l'endroit où la souris était placée au moment du clic (le cercle étant centré et le triangle avec attaché par un point). Assurer que les dessins sont conservés dans toutes les conditions de ré-affichage.
6. faire en sorte qu'un clic droit permette de basculer alternativement entre la couleur noire et la couleur rouge pour les dessins (pas ceux déjà existants mais ceux à venir)
7. faire en sorte qu'un dialogue demande confirmation à l'utilisateur lorsqu'il tente de quitter l'application (par le menu ou en fermant la fenêtre).
8. modifier le composant pour prendre en compte le mode `Continu` du menu. Si le mode `Continu` est sélectionné les dessins (triangles ou cercles) se feront en suivant les mouvements de la souris (en commençant par un appui sur le bouton gauche et en terminant au relâchement du bouton).