

## Examen machine inf250 (programmation)

Durée : 2h45

Aucun document ni calculatrice autorisés.

Seules domaine disponibles : \*.src, \*.java.com, \*.oracle.com

### Sommaire

Examen machine inf250 (programmation) .....	1
Préambule .....	2
Création du workspace et import du projet.....	2
Présentation des fichiers.....	2
JavaDoc.....	2
Reproduction d'un design .....	4
Gestion des événements.....	6
Restitution .....	8

## Préambule

Nous allons commencer par mettre en place notre espace de travail.

Vous trouverez tous les documents nécessaires à l'url suivante <http://www.src/~jberdjug>.

## Création du workspace et import du projet

Nous allons créer un workspace sur le disque public (E:) puis importer le projet et enfin le renommer :

1. Créer le répertoire « E:\workspace ».
2. Télécharger « exam-inf250-2010-2011-ds2-etu.zip »
3. Importer le projet “exam-inf250-2010-2011-ds2-etu” (exam-inf250-2010-2011-ds2-etu.zip) avec File->Import->General->Existing projects into workspace -> select archive file.
4. Renommer votre projet avec refactor -> rename en exam\_nom où nom est votre nom<sup>1</sup>.

## Présentation des fichiers

Vous disposez de deux paquetages « ihm » et « evenements ».

Dans « ihm » vous trouverez le paquetage « ihm.classes » qui contient la classe « FormsWebAccesApogee » à compléter. Une classe de test « TestFormsWebAccesApogee » est disponible dans le paquetage « ihm.tests ».

Dans « evenements » vous trouverez le paquetage « evenements.classes » qui contient la classe « Polices » à compléter. Une classe de test « TestPolices » est disponible dans le paquetage « evenements.tests ». Je vous conseil des classes internes pour la gestion des événements.

Deux « exécutable » sont fournis : ihm.jar et evenements.jar, ils représentent les solutions attendues.

## JavaDoc

Dans eclipse, pour choisir votre WebBrowser, il vous faut :

1. Sélectionner Windows.
2. Sélectionner preferences.
3. Sélectionner general.
4. Sélectionner WebBrowser.

Pour rappel, le proxy est proxy.src sur le port 3128 et il ne faut pas de proxy pour « \*.src » sous IE et « .src » sur firefox.

Si l'accès à Internet est trop lent, vous trouverez une copie de la JavaDoc à l'URL suivante : <http://docs.ens.src/java/api/>

Pour l'utiliser directement depuis eclipse, il vous faut :

1. Sélectionner JRE System Library
2. Sélectionner rt.jar

---

<sup>1</sup> Le nom du projet ne doit pas contenir d'espaces ou d'accents, si vos fichiers ne se trouvent pas dans « src » appeler un enseignant.

3. Dans les propriétés de rt.jar sélectionner la JavaDoc Location (<http://docs.ens.src/java/api/>).

## Reproduction d'un design

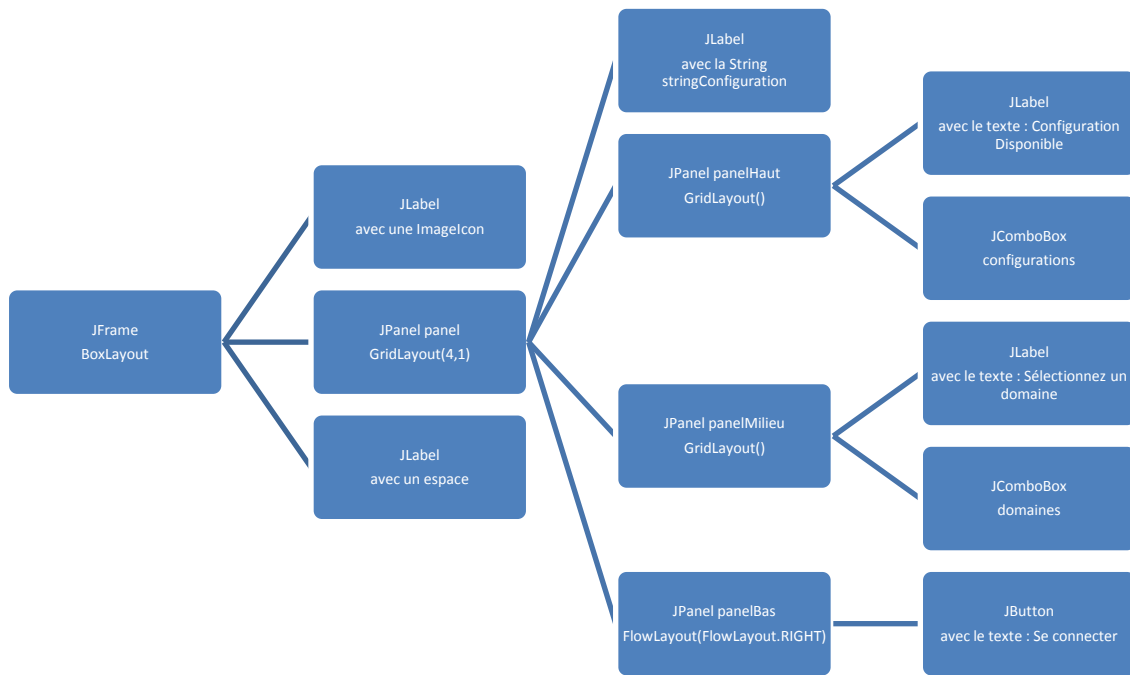


Figure 1 Design à reproduire

Reproduire dans la classe « ihm.classes.FormsWebAccesApogee.java » le design précédent, sachant que :

1. La JFrame utilise comme LayoutManager un BorderLayout vous pouvez vous inspirer de la documentation pour savoir comment rajouter un composant : <http://download.oracle.com/javase/tutorial/uiswing/layout/box.html>.
2. Pour ajouter une icône sur un label il faut d'abord fabriquer une « ImageIcon », l'URL est déjà fournie.
3. La classe « `BorderFactory` » permet de fabriquer des bordures, vous devrez fabriquer des bordures de type ligne et de types titrées.
4. Le texte « Choisissez la configuration et le domaine auxquels vous souhaitez vous connecter » a une couleur définie par rouge 0, vert 0, bleu 204 ; la police est « Dialog », le style est gras et la taille 16.
5. Les textes des JComboBox sont déjà fournis.
6. La couleur de fond des JComboBox est blanche.
7. La fenêtre est non redimensionnable.

L'exécutable ihm.jar est ce que vous devez obtenir.



## Gestion des événements

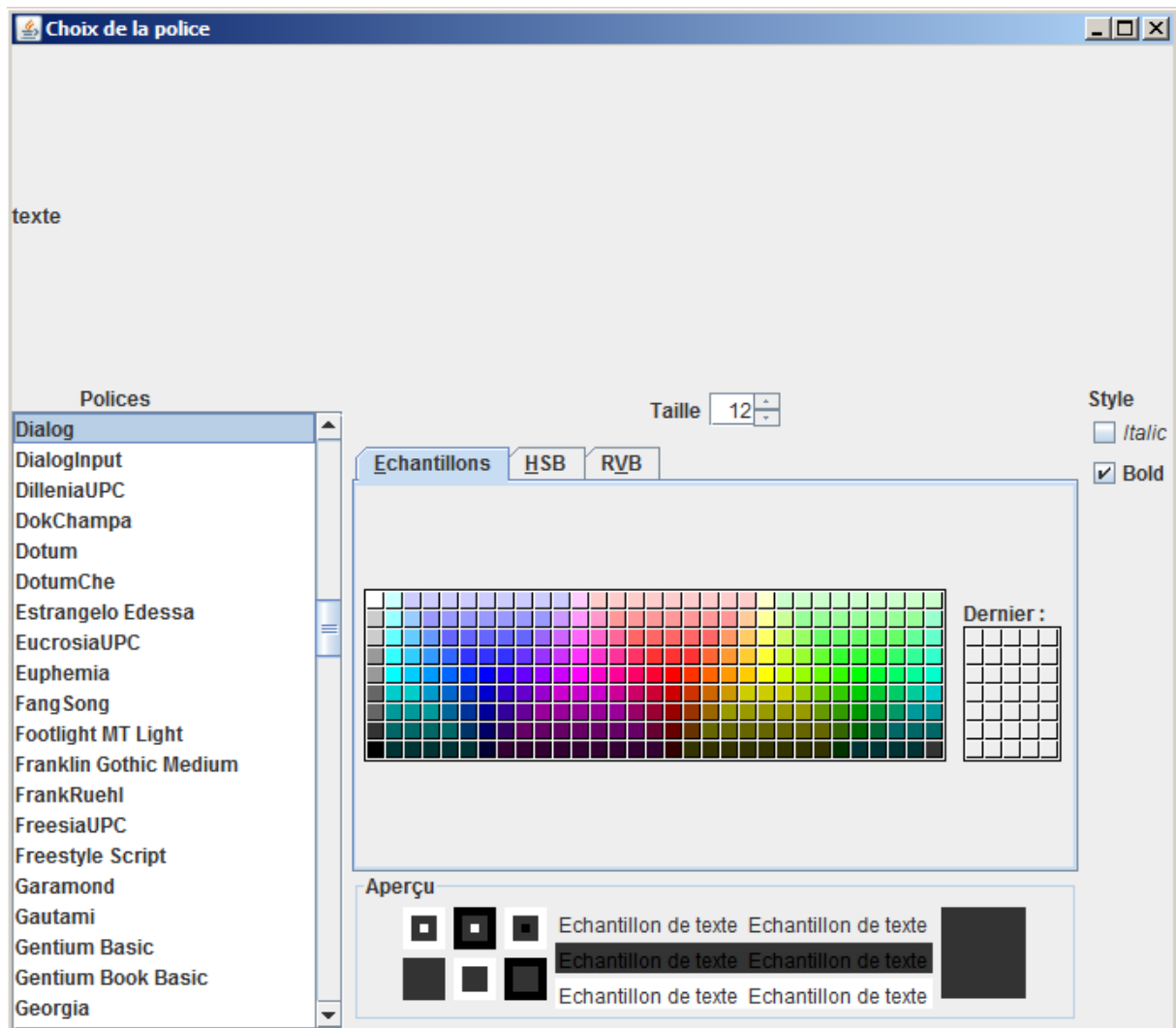


Figure 2 Gestion des événements

Le but est de modifier la mise en forme du JLabel labelTexte en utilisant la JList (listePolices), le JColorChooser (colorChooser), le JSpinner(spinnerTaille) et les JCheckBox (boxItalic, boxBold).

Vous aurez besoin des interfaces ActionListener, d'un ChangeListener (et d'un MouseListener).

Pour la JList vous pouvez au choix implémenter l'interface MouseListener ou hériter de la classe MouseAdapter qui a déjà implémenté l'interface MouseListener. La méthode à utiliser est celle qui correspond au clic.

En tout, vous devez avoir au moins trois écouteurs et cinq composants munis d'un écouteur.

L'exécutable evenements.jar représente ce que vous devez obtenir.

### Remarques :

- L'écouter du JColorChooser doit-être placé sur son model :  
`this.colorChooser.getSelectionModel().addXXXListener(...)`
- Vous pouvez récupérer l'item sélectionné de la JList avec : `this.listPolices.getSelectedValue()`.

- La documentation du constructeur de Font vous rappelle la composition des styles.
- Vous n'avez pas besoin de sélectionner la source des événements.
- Je vous conseil de regrouper le code de la mise en page du label dans la méthode update() et de faire appel à cette dernière dans les méthodes de vos interfaces.

## Restitution

Pour restituer votre travail vous devez exporter votre projet puis le déposer par FTP (ftp-exam.src) dans « jberdjug/examS2 ».

Pour l'export :

1. Sélectionner le projet (exam\_nom)
2. Rafraichir F5
3. Bouton droit-> general -> archive file e:\nom.zip (nom étant votre nom)

Pour le FTP :

1. Se connecter sur ftp-exam.src avec votre login et votre mot de passe.
2. Se placer chez « jberdjug », EXAM\_PROG.
3. Déposer votre export.
4. Vérifier la taille 135 ko.