

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, 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 **yunes** qui résout la question n°3, la classe doit s'appeler `Classe_Yunes_3` ;
- Le programme principal répondant à la question n°3 commencera donc par :

```
public class Classe_Yunes_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 l'enseignant** 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 le fichier source dans le répertoire `/info/ens/yunes/IG`

Sujet

1. Écrire un programme qui, lorsque exécuté, permet d'obtenir l'affichage d'un dialogue contenant un label (affichant le texte **Mot de passe:**), une zone de saisie de texte et deux boutons (affichant **Ok** et **Cancel**).
Indice : `JOptionPane`.
2. Modifier le programme précédent de sorte que si l'utilisateur clique sur le bouton **Cancel** l'application s'arrête simplement. Si un mot de passe est saisi mais n'est pas le bon (le bon mot de passe est le mot **secret**), l'application affiche un dialogue indiquant que le mot de passe est le mauvais, puis lorsque l'utilisateur a lu le message, redemande (au maximum trois fois) le mot de passe via le dialogue de départ.
3. Modifier le programme de sorte que si le mot de passe est correctement saisi il fait apparaître une fenêtre principale avec décoration dont la barre de titre contient votre nom de famille, la barre de menu contient un menu **Fichier** avec une option **Quitter** qui permet de quitter l'application, et un autre menu **Aide** avec une option **À propos...**
4. Lorsque l'utilisateur choisit l'option **À propos...**, un dialogue doit apparaître affichant : le message **Cette application magnifique (version 0.9) est l'œuvre de votre nom**. et un bouton permettant de fermer ce dialogue.
Attention : à tout instant il ne doit y avoir au plus qu'un seul dialogue !
5. Créer un composant personnalisé dont la fonctionnalité principale sera de remplir l'intégralité de l'espace qui lui est alloué avec sa propre couleur de fond (un gris clair de votre choix) et qui aura comme taille par défaut 300x300. Installer un `BorderLayout` dans le panneau principal de la frame. Instancier le composant créé et l'insérer dans la partie centrale de la fenêtre.
Indice : `getPreferredSize`

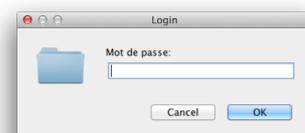


FIGURE 1 – Écran 1

6. Créer un menu **Jeu** (entre les deux autres) contenant une option **Jouer** de sorte que lorsqu'on la sélectionne un label apparaît dans le composant précédent à une position tirée au hasard. Le texte affiché par le label devra être **Je suis là**. À chaque fois que le choisit l'option **Jouer** la position du label est modifiée au hasard. Le raccourci **ALT-J** devra permettre d'obtenir la fonctionnalité sans dérouler le menu. Attention à ce que le label soit toujours entièrement visible à l'écran.
Aide : Pour gérer le positionnement manuel il faut utiliser un `null` layout, puis penser à forcer le dessin lorsque nécessaire.
Indice : `java.util.Random`, `repaint()`, `setLocation()`, `setSize()`, `setBounds()`, `setAccelerator()`.
7. Ajouter dans la fenêtre principale à l'« est » de celle-ci une barre de défilement verticale et au « sud » une barre de défilement horizontale. Les barres de défilement devront être paramétrées de sorte que la position relative du curseur de la barre reflète la position relative du label dans la fenêtre (y compris quand le label est déplacé via l'option du menu).
Indice : `JScrollBar`, `setValue`.
8. Faire en sorte que lorsqu'on déplace le curseur d'une barre de défilement, la position du label soit modifiée en conséquence.
Indice : `AdjustmentListener`.
9. Assurer que le fonctionnement est correct quand bien même on retaille la fenêtre principale (les examinateurs seront cléments dans les cas dégénérés, i.e. fenêtre vraiment trop petite par exemple ; mais intraitables pour les cas ordinaires).
10. Modifier l'application de sorte que d'une exécution à l'autre, la position du label soit conservée.
Indice : `java.util.prefs.Preferences`.