

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'une fenêtre : sans **aucune** décoration, dont la taille est exactement la moitié de celle de l'écran (en hauteur et en largeur), placée exactement au centre de l'écran.
Indice : `Toolkit.getDefaultToolkit().getScreenSize()`
2. Modifier les caractéristiques suivantes de la solution précédente : fond rouge, avec au centre un label affichant votre nom à l'aide d'une police de caractères de taille > 64 ; comme dans la figure 1.
Indice : `Component.getFont()`.
3. Rajouter sous le label un bouton contenant le texte **Ok** qui, lorsqu'on clique dessus, permet : de fermer la fenêtre obtenue précédemment, et de faire 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 **Question 4, Examen IG 2014** 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). Installer un **BorderLayout** dans le panneau principal de la frame. Instancier le composant créé et l'insérer dans la partie centrale de la fenêtre.

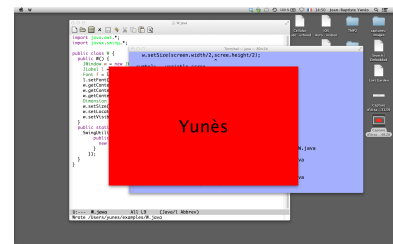


FIGURE 1 – Écran 1

6. Créer un menu **Jeu** (entre les deux autres) contenant une option **Jouer** de sorte que lorsqu'on la sélectionne un bouton apparaît dans le composant précédent à une position tirée au hasard. Le texte affiché par le bouton devra être **Attrape-moi!**. À chaque fois que le choisit l'option **Jouer** la position du bouton est modifiée au hasard. Le raccourci **ALT-J** devra permettre d'obtenir la fonctionnalité sans dérouler le menu. Attention à ce que le bouton 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 au « nord » de celle-ci un label avec le texte **Score: 0 points** au démarrage ; et de sorte que si l'on clique sur le bouton **Attrape-moi!**, le score doit être incrémenté de 1. On ajoutera le mnémonique **A** permettant de déclencher l'action du bouton (mode triche!).
Indice : `setMnemonic()`.
8. Faire en sorte que lorsque la souris s'approche trop près du bouton (à moins de 5 pixels de l'un de ses bords) alors le bouton se déplace à nouveau à une autre position tirée au hasard. Modifier aussi la position du bouton lorsqu'on arrive à cliquer dessus.
Attention : le bouton doit toujours être entièrement visible et doit pouvoir se placer a-priori dans n'importe quelle partie du composant quelque soit la taille de celui-ci.
9. Modifier l'application de sorte que d'une exécution à l'autre, le score soit conservé.
Indice : `java.util.prefs.Preferences`.