

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_Tartempion_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, lorsqu'exécuté, permet d'obtenir l'affichage d'une fenêtre similaire à celle de la figure 1. L'ovale plein doit être de couleur rouge et avoir pour dimensions la moitié de la largeur et la hauteur et doit être placé en centre ; ceci même lorsqu'on retaille la fenêtre. La fenêtre doit avoir une taille assez grande par défaut (à vous de choisir une taille).
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. Ajouter un menu intitulé **Formes**. Celui-ci devra contenir trois options intitulées respectivement **Oval**, **Rectangle plein** et **Rectangle**. Ces options doivent être mutuellement exclusives. Le choix d'une de ces options doit permettre de modifier la forme affichée par dans la fenêtre. L'option **Oval** doit être sélectionnée par défaut.  
Indice : `JRadioButtonMenuItem`, `ButtonGroup`, `setSelected()`.
4. Rajouter dans la fenêtre principale un slider permettant de modifier dynamiquement la taille de la forme. La spécification sera un pourcentage : c'est-à-dire qu'une taille de 0 indiquera une forme de largeur/hauteur égale à 0% de la largeur/hauteur de la fenêtre, 50 représentera 50%, etc. La forme devra toujours être centrée. Le slider devra faire apparaître une graduation (à vous de choisir laquelle).  
Indice : `JSlider`, `ChangeListener...`

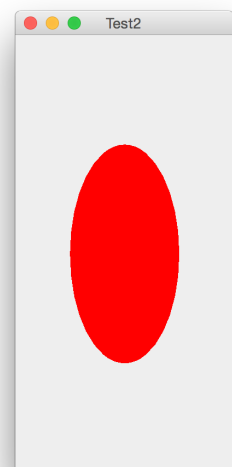


FIGURE 1 – Écran 1

5. Modifier le code de sorte que lorsqu'on clique dans la zone de dessin, un petit (à vous de choisir la taille) carré noir soit rajouté à la position du clic. Ces dessins doivent être accumulés les uns après les autres.  
Indice : `MouseListener`
6. Modifier la fenêtre principale de sorte que le nombre de rectangles créés à l'aide de la souris soit affiché en temps réel dans son coin inférieur 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`.