

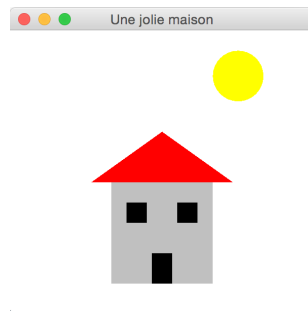
# TP 9 : Les Interfaces Graphiques

christina.boura@uvsq.fr, stephane.lopes@uvsq.fr

15 avril 2016

## 1 Dessiner une maison

Le but de cet exercice est de créer une fenêtre et dessiner dedans la maison suivante :



Vous pouvez réaliser votre programme en utilisant les trois classes ci-dessous :

1. Une classe **Panneau** **extends** **JPanel** qui sera le conteneur dans lequel vous allez dessiner la maison. Le constructeur de cette classe doit préciser la taille de votre panneau et définir la couleur de celui-ci comme blanche. Cette classe contiendra ensuite une seule méthode

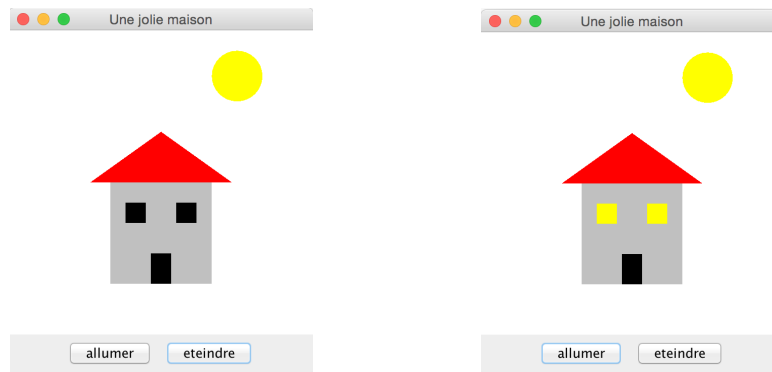
```
public void paintComponent(Graphics g)
```

dont le rôle sera de réaliser le dessin. Utilisez les méthodes de la classe **Graphics** qui vous semblent utiles.

2. Une classe **Fenetre** **extends** **JFrame** qui sera votre conteneur de premier niveau pour placer les autres composants (en réalité juste le panneau). Dans le constructeur de cette classe vous devez donner un titre à votre fenêtre et préciser sa taille. Vous devez ensuite créer un panneau, c'est-à-dire un conteneur intermédiaire dans lequel vous allez dessiner la maison. Utilisez pour cela une instance de la classe **Panneau**.
3. Une classe de test qui contiendra juste la méthode **main** du programme.

## 2 Allumer la lumière dans la maison avec un clic

On veut maintenant ajouter deux boutons en bas de notre fenêtre principale, comme on peut le voir à la figure suivante. Quand l'utilisateur appuie sur le bouton **allumer**, la lumière doit s'allumer dans la maison. De la même façon, quand l'utilisateur appuie sur le bouton **eteindre**, la lumière doit s'éteindre.



Pour réaliser ceci, vous pouvez modifier les trois classes que vous avez créé pour l'exercice précédente. Les principales modifications à faire sont les suivantes :

- Déclarez une variable booléenne d'instance `lumiere` dans votre classe `Panneau`. Modifiez votre méthode `paintComponent` de façon que si `lumiere` vaut `true` les deux fenêtres sont peintes en jaune, tandis que si elle vaut `false` elles sont peintes en noir.
- Votre classe `Fenetre` contiendra maintenant deux conteneurs : Un conteneur de type `Panneau` comme avant qui contiendra le dessin et un conteneur pour contenir les deux boutons. Pensez donc à créer les deux boutons et à les ajouter à ce deuxième conteneur.
- La classe `Fenetre` doit implémenter l'interface `ActionListener` afin qu'on puisse ajouter un écouteur pour les deux boutons. Chaque classe qui implémente l'interface `ActionListener` doit implémenter la méthode

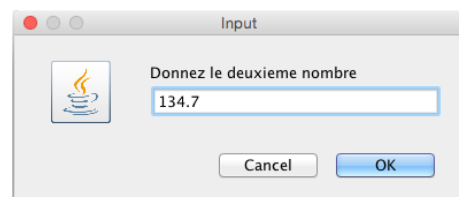
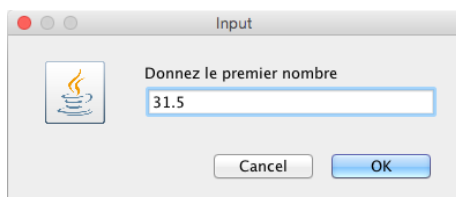
```
public void actionPerformed(ActionEvent e)
```

qui est la seule méthode de cette interface. Cette méthode doit mettre la variable `lumiere` de la classe `Panneau` à jour selon le bouton qui a été appuyé par l'utilisateur et repeindre éventuellement le panneau.

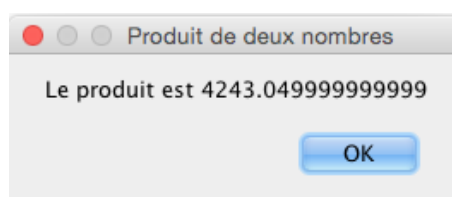
### 3 Créer des fenêtres de dialogue

On veut créer ici un petit programme qui demande à l'utilisateur d'entrer par le clavier deux nombres réels et qui affiche ensuite leur produit à l'écran. Nous allons utiliser pour cela des boîtes de dialogue pour la saisie ainsi que pour l'affichage du résultat.

Au démarrage du programme une boîte comme celle de l'image doit s'afficher et l'utilisateur doit entrer le premier nombre dans le champ texte. Quand l'utilisateur clique sur le bouton OK une deuxième boîte doit s'afficher demandant à l'utilisateur le deuxième entier.



Finalement, quand l'utilisateur appuie sur le bouton OK une fenêtre doit apparaître affichant le produit de ces deux nombres, comme dans l'image ci-dessous.



Pour réaliser ceci vous devez utiliser la classe `JOptionPane` et sa méthode statique `showInputDialog(String s)` comme par exemple ici

```
String s = JOptionPane.showInputDialog("Entrez quelque chose ici");
```

Pour afficher une boîte de dialogue contenant le résultat vous pouvez utiliser la méthode statique `showMessageDialog()` de la même classe.