

IHM avec JAVA

Isabelle Debled-Rennesson
debled@loria.fr

Compléments sur le placement des composants

Auto-évaluation

- Quel est l'algorithme de placement par défaut des
 - JPanel ?

Auto-évaluation

- Quel est l'algorithme de placement par défaut des
 - JPanel ?
 - JFrame ?

Auto-évaluation

- Quel est l'algorithme de placement par défaut des
 - JPanel ?
 - JFrame ?
- Comment modifier un algorithme de placement ?

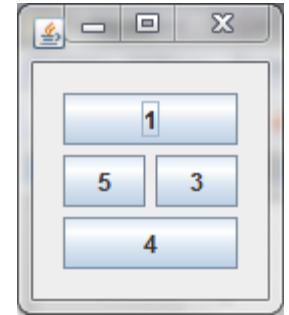
Bordure autour des composants

- Utilisation des classes de javax.swing.border
 - Voir toutes les possibilités à l'adresse <http://docs.oracle.com/javase/tutorial/uiswing/components/border.html#code>
- Méthode setBorder (Border border) sur un JPanel
 - Permet de définir des bordures
 - Exemple :

```
p.setBorder(new EmptyBorder(15,15,15,15));  
//Haut-Gauche-Bas-Droite
```

Bordures – un exemple – dans ExBordure.java

```
class FrameExBordure extends JFrame {  
  
    private JButton b1,b2,b3,b4,b5; //Déclarations  
    private JPanel p;  
  
    public FrameExBordure(String titre){  
        super(titre);  
        p= new JPanel(new BorderLayout(5,5));  
        b1= new JButton("1"); //créations des composants  
        b3= new JButton("3");  
        b4= new JButton("4");  
        b5= new JButton("5");  
        p.setBorder(new EmptyBorder(15,15,15,15)); //Ajoute un bord autour de p  
  
        p.add(b1,BorderLayout.NORTH); //Ajout des boutons dans p  
        p.add(b3,BorderLayout.CENTER);  
        p.add(b4,BorderLayout.SOUTH);  
        p.add(b5,BorderLayout.WEST);  
        this.add(p);  
    }  
}
```



Taille des composants

- **pack ()** : les conteneurs prennent la taille *minimale* pour contenir leur composants, en fonction de la taille *préférée* des composants.

- Pour suggérer ou imposer une autre taille, on retaille par
`setSize(largeur, hauteur)`
`setPreferredSize(dimension)`

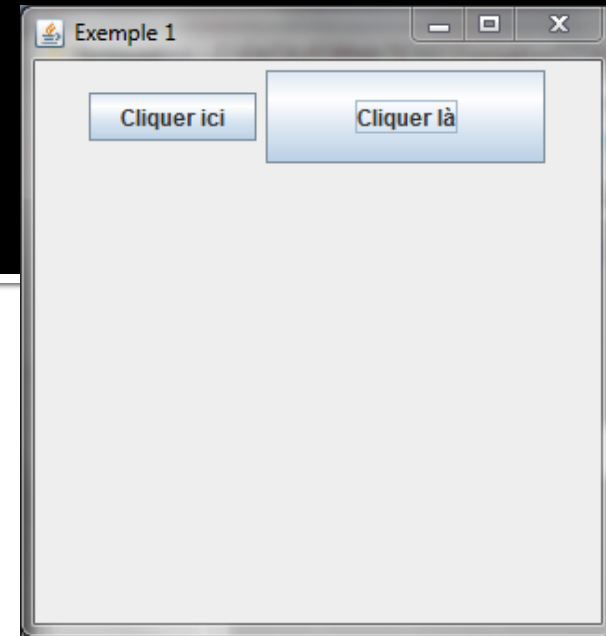
- Ou encore, on réécrit la méthode `getPreferredSize ()` , par exemple

```
public Dimension getPreferredSize() {  
    return new Dimension(100,150);  
}
```


Taille - un exemple -

//...Dans ExTailleBouton.java

```
class FrameExTailleBouton extends JFrame {  
  
    private JButton bout1,bout2;  
    private JPanel p;  
  
    public ExTailleBouton(String titre){  
        super(titre);  
        p= new JPanel();  
  
        bout1=new JButton("Cliquer là");  
        bout1.setPreferredSize(new Dimension(150,50));  
        bout2= new JButton("Cliquer ici");  
  
        p.add(bout2);  
        p.add(bout1);  
        add(p);  
    }  
}
```

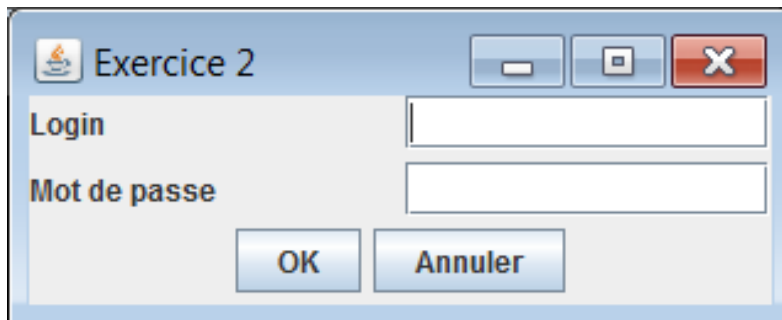


Modification des composants

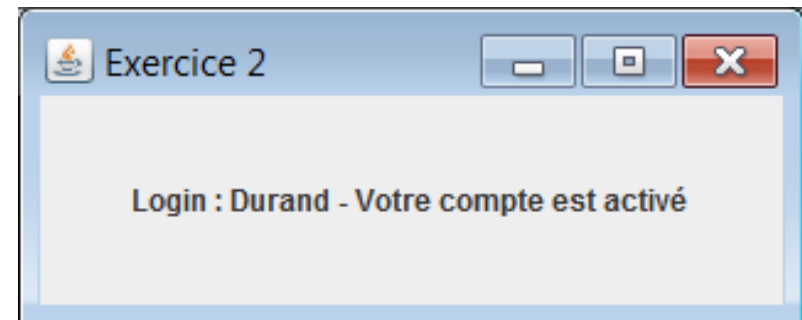
- Un composant peut être ajouté (**add()**) ou enlevé d'un autre composant (**remove()**) ou encore retaillé (**resize()**)
 - Dans ce cas, il est *invalidé*, on obtient l'état par **isValid()**
- Un composant invalidé est réorganisé (et réaffiché) à la prochaine retaille, ou par :
 - **validate()**
 - réorganise un composant invalidé, et le marque comme valide;
 - ne fait rien sur un composant valide (même si un descendant est invalidé)
- Pour l'afficher, nécessité d'appeler la méthode **repaint()**

A faire aujourd'hui

1. Finir le TP8 (calculatrice)
2. Construire l'IG ci-dessous à gauche et gérer les événements suivants :
 - si l'utilisateur appuie sur le bouton Annuler, les textes des zones de saisie disparaissent
 - Si l'utilisateur appuie sur le bouton ok, l'interface se transforme en tenant compte des textes saisis comme ci-dessous à droite



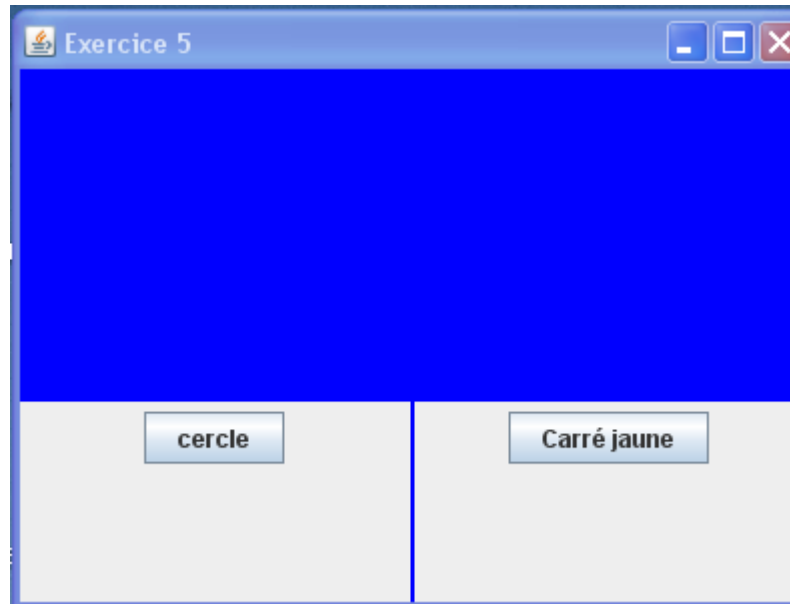
A screenshot of a Windows-style application window titled "Exercice 2". The window contains a login form with two text input fields. The first field is labeled "Login" and the second is labeled "Mot de passe". Below the fields are two buttons: "OK" and "Annuler". The window has standard minimize, maximize, and close buttons in the title bar.



A screenshot of the same application window after a successful login. The text "Login : Durand - Votre compte est activé" is displayed in the center of the window. The window title and standard buttons remain the same.

Exercice 3

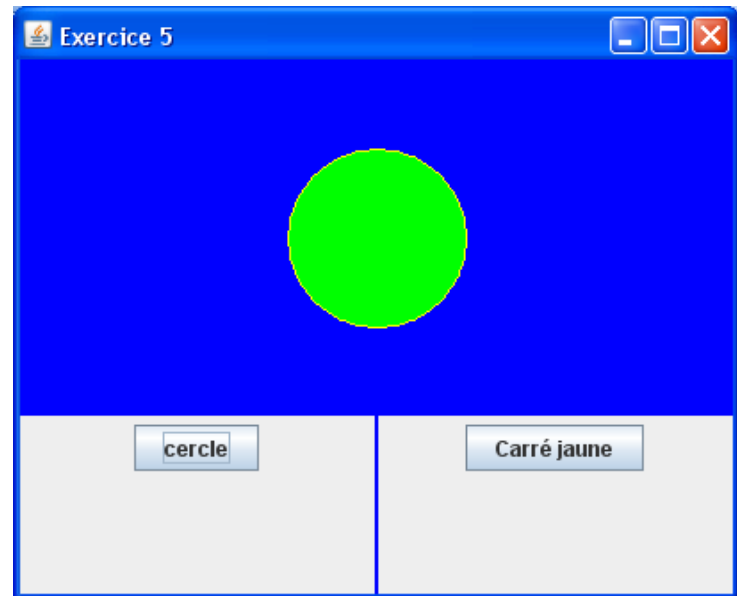
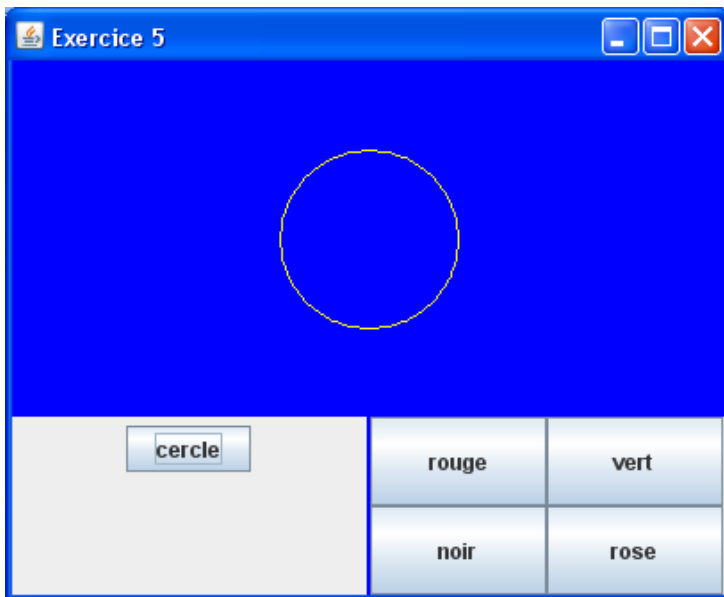
- Donner l'arborescence des composants de l'interface graphique suivante



- Donner le code de cette interface graphique, de plus quand l'utilisateur appuie sur *cercle*, un cercle jaune est dessiné et sur *carré jaune*, un carré jaune est dessiné

Exercice 4

Faire évoluer l'IG de la manière suivante : Quand l'utilisateur appuie sur *cercle*, un cercle se dessine mais le bouton de droite disparaît et l'interface évolue comme ci-dessous :



- Et ensuite après le choix d'une couleur de remplissage, l'interface redevient la même qu'initialement.
- Quelle modification dans l'arborescence ? Code ?