

Java - Programmation orientée objet

Table des matières

5 Composants graphiques - suite	2
5.1 Les composants personnalisés	2
5.2 La méthode <code>paintComponent</code>	3
5.3 Les images	3

Chapitre 5

Composants graphiques - suite

5.1 Les composants personnalisés

En utilisant l'héritage, il est possible de personnaliser ses composants graphiques. La méthode qui définit ce qu'il se passe au moment de l'affichage d'un composant s'appelle `void paintComponent(Graphics p)`. C'est une méthode de la classe `JComponent` dont hérite tous les composants graphiques.

Exemple 5.1.1 Dans l'exemple suivant, on redéfinit la méthode `paintComponent` d'une classe héritant de `JPanel`.

```
public class MyPanel extends JPanel {
    public MyPanel() {
        //On appelle le constructeur parent.
        super();
    }

    @Override
    public void paintComponent(Graphics pinceau) {
        //Appel la méthode paintComponent de la classe mère, afin de gérer
        //par exemple l'affichage du Background
        super.paintComponent(pinceau);

        //Défini ensuite l'affichage spécifique à ce composant: On affiche la
        //chaîne de caractère "Bonjour":
        pinceau.setColor(this.getForeground());
        pinceau.drawString("Bonjour !", 10, 20);
    }
}
```

L'objet `Graphics` en paramètre de `paintComponent` sert à afficher des choses. Cet objet dispose de plusieurs méthodes, parmi lesquelles :

1. `drawString` : Pour afficher du texte
2. `drawLine` : Pour afficher une ligne
3. `drawRect` : Pour afficher un rectangle
4. `drawOval` : Pour afficher une ellipse
5. `drawRect` : Pour afficher un rectangle rempli

5.2 La méthode `paintComponent`

On appelle jamais directement la méthode `paintComponent`. Cette dernière est appelée automatiquement quand un réaffichage du composant est nécessaire, par exemple quand il est redimensionné, ou encore quand on le déplace sur l'écran. Si malgré tout l'utilisateur juge que la méthode `paintComponent` doit être appelée à un moment donné, il peut forcer cet appel avec la méthode `repaint` d'un composant.

5.3 Les images

Il est également possible d'afficher des images au format jpeg, gif ou png, avec la méthode `drawImage` de l'objet `Graphics`, et à l'aide de la classe `Image`.

Exemple 5.3.1 Dans l'exemple suivant, on redéfinit la méthode `paintComponent` d'une classe héritant de `JPanel`.

```
public class MyPanel extends JPanel {
    private Image pingouin;
    public MyPanel() {
        super();
        //On charge l'image en mémoire
        pingouin = Toolkit.getDefaultToolkit().getImage("pingouin.png");
    }

    @Override
    public void paintComponent(Graphics pinceau) {
        //Appel la méthode paintComponent de la classe mère, afin de gérer
        //par exemple l'affichage du Background
        super.paintComponent(pinceau);

        //On affiche ensuite l'image, en définissant sa position d'affichage,
        //et en dernier paramètre le composant sur lequel on doit afficher
        //l'image.
        pinceau.drawImage(pingouin, 10, 20, this);
    }
}
```