

Exemple 04 : Interface graphique avec Swing

Java offre plusieurs bibliothèques (standard ou non-standard) pour la conception et la réalisation des interfaces graphiques et conviviale, permettant aux utilisateurs d'interagir avec l'application d'une manière simple et efficace.

Tout de même, il faut faire attention à ne pas mélanger le code associé aux interfaces graphiques (fenêtres, boîte de dialogue, boutons, zones de textes, icônes, etc.) aux codes associés à la logique métiers (le domaine d'application). De telle sorte de garder le code source bien structuré.

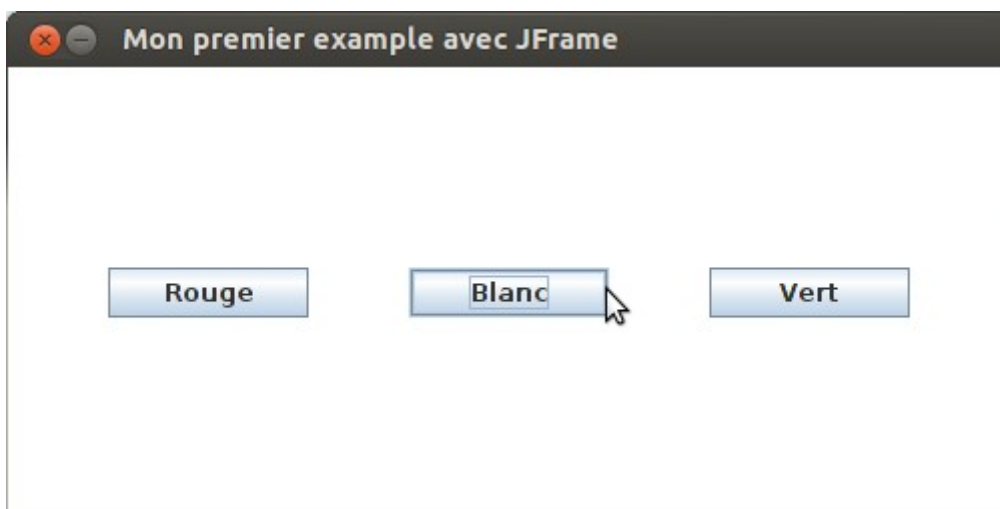
N.B. :

Pour le T.P. à réaliser, il est évident qu'une représentation graphique de la circulation des véhicules sera mieux placée qu'une représentation console vu la nature du problème à résoudre.

Le premier programme : Changer la couleur de la fenêtre

Dans ce premier programme, on va voir comment créer une interface graphique constituée d'une fenêtre contenant trois boutons : Rouge, Blanc et Vert. Le clic sur un bouton sera l'effet de changer la couleur de la fenêtre. Il faut toujours garder à l'esprit de séparer la logique métier de l'interface graphique. Ce programme est constitué de deux classes : **MainExample07** (contenant uniquement la méthode `main()`) et la classe : **MyFrame** (contenant l'implémentation nécessaire pour la création de l'interface et le traitement des événements associés aux trois boutons).

L'interface est représentée par la figure suivante :



MyFrame.java :

```
public class MyFrame extends JFrame{
    private static final long serialVersionUID = 1L;

    private static JFrame instance = null; // (singleton)

    private JButton bRouge;
    private JButton bBlanc;
    private JButton bVert;

    /**
     * Constructeur private : interdiction d'instancier en dehors de la classe
     */
    private MyFrame(){
        super("Mon premier exemple avec JFrame");
        initContents();
    }

    /**
     * Pour initialiser le contenu de la fenetre
     */
    private void initContents(){
        this.bRouge = new JButton("Rouge");
        this.bBlanc = new JButton("Blanc");
        this.bVert = new JButton("Vert");

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setResizable(false);

        manageLyaout();
        manageEvents();
    }

    /**
     * Pour la disposition des composants graphiques (3 boutons) sur le frame
     */
    private void manageLyaout(){
        this.setLayout(null);

        bRouge.setBounds(50, 100, 100, 25); // setBounds (x, y, w, h)
        bBlanc.setBounds(200, 100, 100, 25);
        bVert.setBounds(350, 100, 100, 25);

        this.add(bRouge);
        this.add(bBlanc);
        this.add(bVert);

        this.setSize(500, 225);
        this.setLocationRelativeTo(null);
    }

    /**
     * Pour programmer les differents evenements
     */
    private void manageEvents(){
        ActionListener actionListenerAllButtons = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                handleActionEvents(evt);
            }
        };
    }
};
```

```

        bRouge.addActionListener(actionListenerAllButtons);
        bBlanc.addActionListener(actionListenerAllButtons);
        bVert.addActionListener(actionListenerAllButtons);
    }

    /**
     * le traitement de l'evenement(ActionEvent<br/>
     * @param evt : l'evenement capte par l'application
     */
    private void handleActionEvents(ActionEvent evt){
        Color color = this.getContentPane().getBackground();
        if (evt.getSource().equals(bRouge)){
            color = Color.RED;
        }
        else if (evt.getSource().equals(bBlanc)){
            color = Color.WHITE;
        }
        else if (evt.getSource().equals(bVert)){
            color = Color.GREEN;
        }

        this.getContentPane().setBackground(color);
    }

    /**
     * permet de creer une instance unique et l'affiche avec setVisible(true)
     */
    public static void createAndShow(){
        if (instance == null){
            instance = new MyFrame();
        }

        instance.setVisible(true);
    }
}

```

MainExample07.java:

```

package main;

import userinterface.MyFrame;

public class MainExample07{
    public static void main(String[] args){
        MyFrame.createAndShow();
    }
}

```

Le deuxième programme : Résolution de l'équation $ax+b = 0$

Veillez analyser (lire attentivement et avec réflexion) le code sources (séparation entre le modèle de la présentation).

Récapitulatif

L'objectif n'est pas uniquement de faire des interfaces graphiques (Mais, en plus, de bien structurer le code sources et mettre chaque chose à la place qui lui convient)

Voir aussi GroupLayout pour la gestion de disposition.