



Programmation événementielle

Java et Swing



L3 Informatique
Semestre 6

Cours donné par
Rédigé par Antoine de ROQUEMAUREL

2014

Table des matières

1	Fenêtres	3
1.1	Boutons	3
2	Table, Arbres et Modèles	4
2.1	Modèles	4
2.2	Rendus	5
2.3	Arbre	5
3	Événements	6
3.1	Exercice 1	6

1

Fenêtres

```
1 public class Exo1 extends JFrame {  
    public Exo1() {  
3         super("Exercice 1");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
5        Container content = getContentPane();  
        content.setBackground(Color.BLUE);  
7  
        setSize(400,600);  
9        setVisible(true);  
    }  
11 public static void main(String[] args) {  
    new Exo1();  
13 }  
}
```

Listing 1.1 – Première fenêtre

```
Object[] options = {"Truc", "Machin", "Chose", "Chouette"};  
2 int op = (JOptionPane.showOptionDialog(this, "Title", "bla bla bla truc", ↵  
    JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null, options);
```

Listing 1.2 – InputDialog

1.1 Boutons

```
JButton b = new JButton("Bouton");  
2 b.setBackground(Color.BLUE);  
b.setToolTipText("Ceci est un bouton");  
4 b.setMnemonic(KeyEvent.B_K);
```

Listing 1.3 – JButton

2

Table, Arbres et Modèles

2.1 Modèles

```
1 public class Model extends AbstractTableModel {
2     public int getRowCount() {
3         return 10;
4     }
5
6     public int getColumnCount() {
7         return 10;
8     }
9
10    public Object getValueAt(int row, int column) {
11        return new Integer((row+1)*(column+1));
12    }
13
14    public boolean isCellEditable(int rowIndex, int columnIndex) {
15        return false;
16    }
17
18    public String getColumnName(int column) {
19        return super.getColumnName(column+1);
20    }
21 }
```

Listing 2.1 – Table de multiplication

```
1 public class Model extends AbstractTableModel {
2     private Object[][] data;
3     public int getRowCount() {
4         return data.length;
5     }
6
7     public int getColumnCount() {
8         return 3;
9     }
10
11    public Object getValueAt(int row, int column) {
12        return data[row][column];
13    }
14
15    public Object setValueAt(Object aValue, int row, int column) {
16        data[row][column] = aValue;
17        fireTableCellUpdated(row, column);
18    }
19
20    public String getColumnName(int column) {
21        String name = "";
22        switch(column) {
```

```

23     case 0:
24         name = "Nom";
25         break;
26     case 1:
27         name = "Prénom";
28         break;
29     case 2:
30         name = "Adress";
31         break;
32     }
33
34     return name;
35 }
36
37 public boolean isCellEditable(int rowindex, int columnindex) {
38     return columnIndex >= 2;
39 }
40
41 }

```

Listing 2.2 – Répertoire

2.2 Rendus

```

1 public class Render extends DefaultTableCellRenderer {
2 }

```

Listing 2.3 – Répertoire

2.3 Arbre

```

1 DefaultMutableTreeNode swing = new DefaultMutableTreeNode("Swing");
2 DefaultMutableTreeNode containers = new DefaultMutableTreeNode("Containers");
3 DefaultMutableTreeNode controls = new DefaultMutableTreeNode("Controls");
4 DefaultMutableTreeNode menus = new DefaultMutableTreeNode("Menus");
5
6 swing.add(containers);
7 swing.add(controls);
8 swing.add(menus);
9
10 JTree jtree = new JTree(swing);

```

Listing 2.4 – Répertoire

3

Événements

3.1 Exercice 1

Donner le triplet widget, event, listener pour les actions utilisateurs suivantes :

Clic sur un bouton simple JButton , MouseEvent, MouseListener

Clic sur le bouton d'icônification d'une fenêtre JFrame, WindowEvent , WindowListener

Clic sur une case à cocher JCheckbox ActionEvent , ActionListener

Appui sur entrée dans un champ de saisie de texte JTextField ,KeyEvent, KeyListener

Clic dans un champ de saisie de texte JTextField MouseEvent, MouseListener

Déplacement du curseur dans un champ texte JTextField,CaretEvent,CaretListener

Sélection d'un élément dans une liste JList,ListSelectionEvent,ListSelectionListener

Déplacement de la souris sur un bouton JButton, MouseEvent, MouseListener