Laborator 12:

Interfeţe grafice în Java

Componente Swing

Întocmit de: Adina Neculai

Îndrumător: Asist. Drd. Gabriel Danciu

11 decembrie 2011

I. NOŢIUNI TEORETICE

Swing este o librărie a limbajului Java, librărie ce este considerată o extensie a lui *AWT*. *Swing* conţine componente noi şi îmbunătăţite care sporesc funcţionalitatea şi înfăţişarea GUI-ului în Java. Pachetul din care vom lua componentele necesare se numeşte *javax.swing*.

Câteva dintre caracteristicile care au fost adăugate în acest pachet sunt următoarele:

- S-au introdus componente diverse cum ar fi: tabele, arbori, slider-e, bare de progres, componente text;
- componentele Swing au *tooltip*-uri plasate deasupra lor. Un tooltip este o fereastră de tipul popup care apare temporar deasupra unei componente atunci când cursorul mouse-ului se află pe componenta respectivă. Acestea sunt utilizate pentru a oferi mai multe informaţii despre componenta în cauză.

Priviţi în imaginea următoare relaţia dintre clasele pachetului *awt* şi clasele pachetului *swing*:

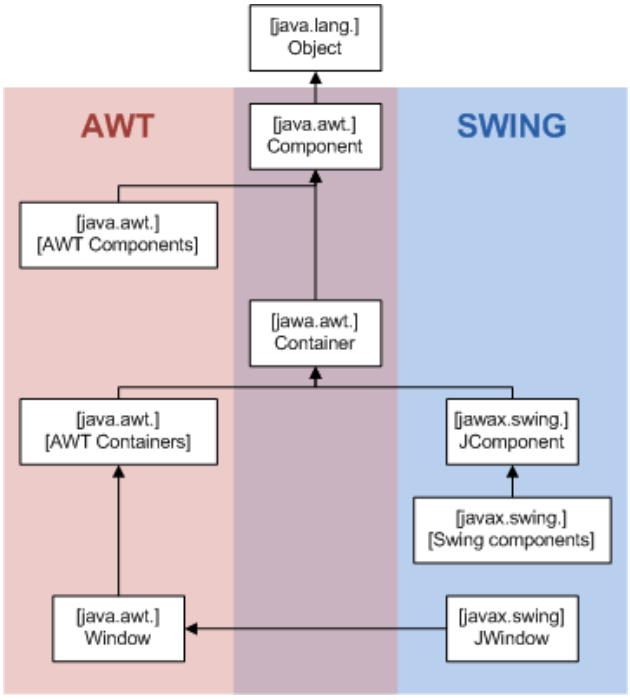

Figura 1: Ierarhia claselor Swing şi relaţia cu clasele AWT

Având în vedere faptul că librăria Swing a fost construită peste AWT, paşii ce trebuie urmaţi pentru a realiza o aplicaţie grafică în Java rămân aceiaşi:

- Crearea unei suprafeţe de afişare pe care vor fi aşezate componentele grafice (butoane, controale de editare, texte, etc);
- Crearea şi aşezarea obiectelor grafice pe suprafaţa de afişare în poziţiile corespunzătoare;
- Definirea unor acţiuni care trebuie să se execute în momentul când utilizatorul interacţionează cu obiectele grafice ale aplicaţiei;
- "Ascultarea" evenimentelor generate de obiecte în momentul interacţiunii cu utilizatorul şi executarea acţiunilor corespunzătoare aşa cum au fost ele definite.

A. Componente grafice Swing

Cele mai folosite componente grafice sunt reprezentate în figura următoare:

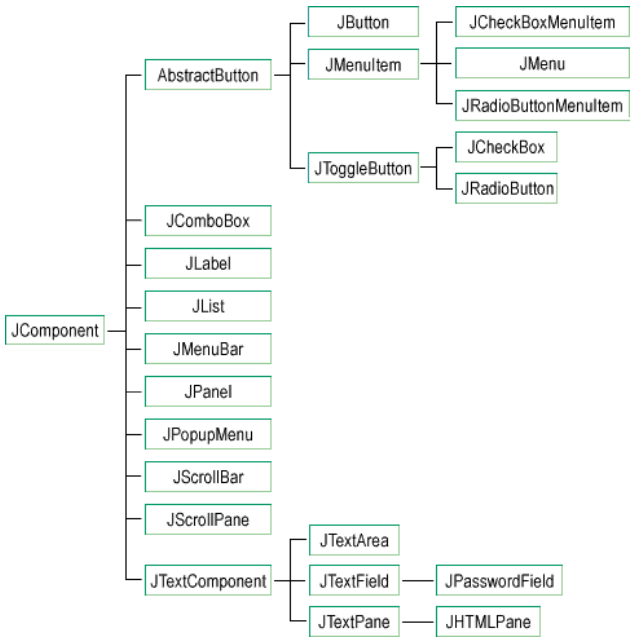

Figura 2: Ierarhia componentelor Swing

B. Componente meniu în Swing

Bara de meniu poate fi creată cu ajutorul clasei **JMenuBar** care permite adăugarea componentelor de tipul *JMenu* pentru a construi meniul propriu-zis.

JMenu este o fereastră popup care conţine componente *JMenuItem* şi care este afişată atunci când utilizatorul selectează o componentă *JMenuBar*. În plus, o componentă *JMenu* poate conţine *JSeparator*.

Clasa **JMenuItem** realizează de fapt un buton din lista care se afişează utilizatorului atunci când acesta selectează o componentă *JMenu*. Clasele sale copil sunt: *JCheckBoxMenuItem, JMenu, JRadioButtonMenuItem*. De ce şi *JMenu*? Pentru că un *JMenu* poate conţine ca *JMenuItem* un submeniu. Nu consider necesar explicarea componentelor *JCheckBoxMenuItem, JRadioButtonMenuItem*. Urmăriţi exemplul prezentat în secţiunea II D.

II. PREZENTAREA LUCRĂRII DE LABORATOR

A. Componenta JLabel

Exemplul de mai jos creează o multitudine de etichete Swing, aranjându-le în moduri diferite. Ce se poate observa este că, spre deosebire de etichetele AWT, cele Swing pot introduce şi imagini pe lângă textul propriu-zis. De asemenea, se vor folosi aceiaşi gestionari de poziţie din pachetul AWT. În acest caz avem *FlowLayout*.

Imaginile se vor adăuga cu ajutorul clasei *ImageIcon* al cărei constructor conţine ca şi parametru un String ce reprezintă calea către imaginea dorită.

Metoda *makeLabel()* returnează un obiect *JLabel*, obiect instanţiat cu şirul de caractere ”Smile” concatenat cu numărul etichetei, iar al doilea parametru reprezintă obiectul imagine. Tot în metoda *makeLabel* se setează poziţia şirului de caractere pe verticală şi pe orizontală cu ajutorul unor numere întregi primite ca parametru. Aceste numere întregi sunt de fapt constante ale clasei *JLabel*.

```
1 import java.awt.*;
2 import javax.swing.*;
3
4 public class LableExample {
5    public static Icon icon = new ImageIcon("resurse/smile.gif");
6
7    public static void main(String[] args) {
```

```
8         JFrame frame = new JFrame("Label Example");
9         frame.setSize(350, 200);
10        frame.setLayout(new FlowLayout());
11
12        JLabel[] labels = new JLabel[9];
13        labels[0] = makeLabel(JLabel.TOP, JLabel.LEFT, 0);
14        labels[1] = makeLabel(JLabel.TOP, JLabel.CENTER, 1);
15        labels[2] = makeLabel(JLabel.TOP, JLabel.RIGHT, 2);
16        labels[3] = makeLabel(JLabel.CENTER, JLabel.LEFT, 3);
17        labels[4] = makeLabel(JLabel.CENTER, JLabel.CENTER, 4);
18        labels[5] = makeLabel(JLabel.CENTER, JLabel.RIGHT, 5);
19        labels[6] = makeLabel(JLabel.BOTTOM, JLabel.LEFT, 6);
20        labels[7] = makeLabel(JLabel.BOTTOM, JLabel.CENTER, 7);
21        labels[8] = makeLabel(JLabel.BOTTOM, JLabel.RIGHT, 8);
22
23        // dezactiveaza eticheta 0
24        labels[0].setEnabled(false);
25
26        // se dezactiveaza eticheta 1. In acest caz, se seteaza alta imagine
27        labels[1].setEnabled(false);
28        labels[1].setDisabledIcon(new ImageIcon("resurse/no.gif"));
29
30        // modifica distanta intre imagine si text la etichetele 2 si 3
31        labels[2].setIconTextGap(15);
32        labels[3].setIconTextGap(0);
33
34        // se adauga etichetele in fereastra
35        for (int i = 0; i < 9; i++) {
36           frame.add(labels[i]);
37        }
38
39        frame.setVisible(true);
40     }
41
42     public static JLabel makeLabel(int vert, int horiz, int contor) {
43        JLabel l = new JLabel("Smile " + contor, icon, SwingConstants.CENTER);
44        l.setVerticalTextPosition(vert);
45        l.setHorizontalTextPosition(horiz);
46        l.setBorder(BorderFactory.createLineBorder(Color.blue));
47        return l;
48     }
49  }
```

B. Componentele JTextField şi JTextPassword

Componentele text au fost îmbunătăţite în pachetul Swing. Dacă în pachetul AWT nu era permisă crearea unui câmp text în care să se seteze poziţionarea scrisului, iată că *JTextField* face posibil acest lucru (linia de cod 15 şi 19).

```
1  import javax.swing.*;
2  import java.awt.*;
3
4  public class TextFieldExample {
```

```
5
6     public static void main(String[] args) {
7        JFrame frame = new JFrame("TextField Example");
8
9        JLabel lastNameLabel = new JLabel("Last Name", SwingConstants.LEFT);
10       JLabel firstNameLabel = new JLabel("First Name", SwingConstants.CENTER);
11       JLabel passwordLabel = new JLabel("Password", SwingConstants.RIGHT);
12
13       JTextField lastNameTF = new JTextField(15);
14       lastNameTF.setFont(new Font("bold", Font.BOLD, 12));
15       lastNameTF.setHorizontalAlignment(JTextField.RIGHT);
16
17       JTextField firstNameTF = new JTextField();
18       firstNameTF.setFont(new Font("italic", Font.ITALIC, 13));
19       firstNameTF.setHorizontalAlignment(JTextField.CENTER);
20
21       JPasswordField passwordTF = new JPasswordField();
22
23       JPanel p = new JPanel();
24       p.setLayout(new GridLayout(3, 1));
25
26       p.add(lastNameLabel);
27       p.add(lastNameTF);
28       p.add(firstNameLabel);
29       p.add(firstNameTF);
30       p.add(passwordLabel);
31       p.add(passwordTF);
32
33       frame.add(p);
34
35       frame.pack();
36       frame.setVisible(true);
37    }
38 }
```

C. Componenta JButton

Exemplul prezentat mai jos afişează într-o fereastră trei butoane. Primul conţine doar text, cel de-al doilea conţine doar o imagine, iar cel de-al treilea conţine şi text şi o imagine. Toate cele trei butoane au fost create cu ajutorul clasei *JButton*.

Mai mult, numele apărut pe cel de-al treilea buton are subliniată prima literă, iar la combinaţia de taste ALT şi prima literă butonul este selectat. Acest lucru este posibil datorită metodei *setMnemonic()* a clasei *JButton* care primeşte ca parametru prima literă a numelui de pe buton. Astfel butonul răspunde atât la click-urile de mouse cât şi la combinaţia ALT+prima literă.

Observaţi ce se afişează atunci când plasaţi mouse-ul deasupra butoanelor. Depistaţi liniile de cod care realizează acest lucru.

```
1  import java.awt.*;
2  import javax.swing.*;
3
4  public class ButtonExample {
5
6     public static void main(String args[]) {
7        JFrame frame = new JFrame("JButton Example");
8        frame.setSize(300, 100);
9
10       JPanel panel = new JPanel();
11       panel.setLayout(new FlowLayout(FlowLayout.LEFT));
12
13       JButton cutButton = new JButton("Cut");
14       cutButton.setToolTipText("Tooltip: buton cut");
15       JButton copyButton = new JButton(new ImageIcon("resurse/copy.gif"));
16       copyButton.setToolTipText("Tooltip: buton copy");
17
18       String buttonName = "Paste";
19       String icon = "resurse/paste.gif";
20       final char key = buttonName.charAt(0);
21       JButton pasteButton = new JButton(buttonName, new ImageIcon(icon));
22       pasteButton.setMnemonic(key);
23       pasteButton.setToolTipText("Tooltip: buton paste");
24
25       panel.add(cutButton);
26       panel.add(copyButton);
27       panel.add(pasteButton);
28
29       frame.add(panel, BorderLayout.NORTH);
30
31       frame.setVisible(true);
32    }
33 }
```

D. Componente Meniu

Clasa *MenuExample* afişează o fereastră ce conţine o bară de meniu cu două componente meniu. Prima componentă meniu mai conţine, însă, alte obiecte meniu şi un submeniu.

În acest exemplu, meniul este afişat pe verticală datorită metodei *setLayout()* care este apelată de către componenta *JMenuBar* cu parametrul *GridLayout* în linia de cod 19.

Observaţi că asemenea componentelor *JButton* şi *JLabel*, şi *JMenuItem* poate conţine imagini. De asemenea, meniul prezintă separatori între componentele diferite ale unui *JMenu* (liniile de cod 36, 50 şi 58).

```
1  import java.awt.GridLayout;
2  import javax.swing.*;
3
4  public class MenuExample {
5
```

```
6     public static void main(String s[]) {
7        JFrame frame;
8        JMenuBar menuBar;
9        JMenu menu, submenu;
10       JMenuItem menuItem;
11       JRadioButtonMenuItem rbMenuItem;
12       JCheckBoxMenuItem cbMenuItem;
13
14       frame = new JFrame("Menu example");
15       frame.setSize(200, 200);
16
17       // se creeaza bara de meniu
18       menuBar = new JMenuBar();
19       menuBar.setLayout(new GridLayout(2,1));
20
21       // se construieste primul meniu
22       menu = new JMenu("Un Meniu");
23       menuBar.add(menu);
24
25       //un grup de componente JMenuItems
26       menuItem = new JMenuItem("Doar text");
27       menu.add(menuItem);
28
29       menuItem = new JMenuItem("Si text si imagine", new ImageIcon("resurse/rolling.gif"));
30       menu.add(menuItem);
31
32       menuItem = new JMenuItem(new ImageIcon("resurse/rolling.gif"));
33       menu.add(menuItem);
34
35       // un grup de componente radio-buton
36       menu.addSeparator(); //adauga la sfarsitul grupului de componente anterioare o linie orizontala,
             considerata separator
37       ButtonGroup group = new ButtonGroup();
38
39       rbMenuItem = new JRadioButtonMenuItem("o componenta radio-button");
40       rbMenuItem.setSelected(true);
41       //butonul radio se adauga grupului si meniului
42       group.add(rbMenuItem);
43       menu.add(rbMenuItem);
44
45       rbMenuItem = new JRadioButtonMenuItem("alta componenta radio-buton");
46       group.add(rbMenuItem);
47       menu.add(rbMenuItem);
48
49       // un grup de componente check-box
50       menu.addSeparator();
51       cbMenuItem = new JCheckBoxMenuItem("o componenta check-box");
52       menu.add(cbMenuItem);
53
54       cbMenuItem = new JCheckBoxMenuItem("alta componenta check-box");
55       menu.add(cbMenuItem);
56
57       // un submeniu
58       menu.addSeparator();
59       submenu = new JMenu("Submeniu");
60
61       menuItem = new JMenuItem("alt obiect in submeniu");
62       submenu.add(menuItem);
63
```

```
64          menu.add(submenu);
65
66          // se construieste al doilea meniu ce se adauga in bara de meniu
67          menu = new JMenu("Alt Meniu");
68          menuItem = new JMenuItem("obiect in Alt Meniu");
69          menu.add(menuItem);
70          menuBar.add(menu);
71
72          //se ataseaza ferestrei bara de meniu
73          frame.setJMenuBar(menuBar);
74
75          frame.pack();
76          frame.setVisible(true);
77      }
78  }
```

III. TEMĂ

1. Realizaţi cerinţele 2 şi 3 din *Laborator10.pdf*. Folosiţi componente din pachetul *javax.swing*. Consultaţi figura din secţiunea I A şi API-ul.