

Лекція №7. Елементи форм та графічного інтерфейсу в Java

ПРОГРАМУВАННЯ ТА ПРИКЛАДНІ ІНФОРМАЦІЙНІ
СИСТЕМИ

Label

Компонент Label -це просто рядок тексту, оформлений як графічний компонент для розміщення в контейнері. Текст можна поміняти тільки методом доступу `setText(string text)`, але не введенням користувачем з клавіатури або за допомогою миші.

Label конструктори

Label() - порожній об'єкт без тексту;

Label(String text) - об'єкт з текстом text, який притискається до лівого краю компонента;

Label(String text, int alignment) - об'єкт з текстом text і певним розміщенням в компоненті тексту, що задається однією з трьох констант: CENTER, LEFT, RIGHT.

Label методи

`setAlignment(int alignment)` – змінює розміщення тексту відносно контейнеру.

`getText()` – дозволяє отримати текст

`setText(String text)` – дозволяє змінити текст

`getAlignment ()` – отримати розміщення тексту.

Label події

У класі Label відбуваються події класів Component: ComponentEvent, FocusEvent, KeyEvent, MouseEvent.

Button

Компонент Button - це кнопка стандартного для даної графічної системи виду з написом, яка вміє реагувати на натискання кнопки миші

Button конструктори

Button() – створює кнопку без надпису

Button(String label) – створює кнопку з написом label.

Button методи

`getLabel()` – дозволяє отримати значення надпису кнопки

`setLabel(String label)` – дозволяє змінити надпис кнопки

Button події

Крім подій класу Component: ComponentEvent, FocusEvent, KeyEvent, MouseEvent, при впливі на кнопку відбувається подія ActionEvent

Checkbox

Компонент checkbox - це напис праворуч від невеликого квадратика, в якому в деяких графічних системах з'являється галочка після натискання кнопкою миші – компонент переходить в стан (state) on. Після наступного клацання галочка пропадає – це стан off. В інших графічних системах стан on відзначається "вдавленням« квадратика. У компоненті checkbox стани on/off відзначаються логічними значеннями true/false відповідно.

Checkbox конструктори

Checkbox() – створює компонент без надпису,

Checkbox(String label) – створює компонент з надписом label в стані off

Checkbox (String label, boolean state) створює з надписом label і в заданому стані state.

Checkbox методи

`getLabel()` – дозволяє отримати значення надпису,

`setLabel (String label)` – змінює значення надпису

`getState()` – дозволяє отримати стан (true/false),

`setState(boolean state)` – змінює стан

Checkbox події

У класі `Checkbox` відбуваються події класу `Component`: `ComponentEvent`, `FocusEvent`, `KeyEvent`, `MouseEvent`, а при зміні стану кнопки виникає подія `ItemEvent`.

CheckboxGroup

В ситуаціях, коли потрібно вибрати тільки один пункт з декількох, утворюється група так званих радіокнопок (radiobuttons). Вони позначаються звичайно колом або ромбиком, вибір позначається жирною крапкою в колі чи “вдавленням” ромбика.

У бібліотеці AWT радіокнопки не утворюють окремих компонентів. Замість цього декілька компонентів checkbox об'єднуються в групу за допомогою об'єкту класу checkboxGroup.

CheckboxGroup КОНСТРУКТОР

CheckboxGroup () – створює групу радіо кнопок

CheckboxGroup методи

`getSelectedCheckbox()` - повертає вибраний об'єкт `Checkbox`;

`setSelectedCheckbox(Checkbox box)` - задає вибір.

CheckboxGroup

Щоб організувати групу радіокнопок, треба спочатку сформулювати об'єкт класу `CheckboxGroup`, а потім створювати кнопки конструкторами

`Checkbox(String label, CheckboxGroup group, boolean state)`

`Checkbox(String label, boolean state, CheckboxGroup group)`

Ці конструктори ідентичні, просто при записі конструктора можна не думати про порядок проходження його аргументів.

Тільки одна радіокнопка в групі може мати стан `state = true`

Приклад

```
import java.awt.*;
import java.awt.event.*;
class SimpleComp extends Frame{
    SimpleComp(String s){
        super(s);
        setLayout(null);
        Font f = new Font("Serif", Font.BOLD, 15);
        setFont(f);
        Label l1 = new Label("Выберите товар:", Label.CENTER);
        l1.setBounds(0, 50, 120, 30); add(l1);
        Label l2 = new Label("Выберите способ оплаты:");
        l2.setBounds(160, 50, 200, 30); add(l2);
        Checkbox chl = new Checkbox("Книги");
        chl.setBounds(20, 90, 100, 30); add(chl);
        Checkbox ch2 = new Checkbox("Диски");
        ch2.setBounds(20, 120, 100, 30); add(ch2);
        Checkbox ch3 = new Checkbox("Игрушки");
        ch3.setBounds(20, 150, 100, 30); add(ch3);
    }
}
```

```
CheckboxGroup grp = new CheckboxGroup();
Checkbox chg1 = new Checkbox("Почтовым переводом", grp, true);
chg1.setBounds(170, 90, 200, 30); add(chg1);
Checkbox chg2 = new Checkbox("Кредитной картой", grp, false);
chg2.setBounds(170, 120, 200, 30); add(chg2);
Button b1 = new Button("Продолжить");
b1.setBounds( 30, 220, 100, 30); add(b1);
Button b2 = new Button("Отменить");
b2.setBounds(140, 220, 100, 30); add(b2);
Button b3 = new Button("Выйти");
b3.setBounds(250, 220, 100, 30); add(b3);
setSize(400, 300);
setVisible(true);
```

```
public static void main(String[] args){
    Frame f = new SimpleComp (" Простые компоненты");
    f.addWindowListener (new WindowAdapter(){
        public void windowClosing(WindowEvent ev){
            System.exit(0);
        }
    });
}
```

Choice

Компонент choice - це список, що розкривається, один, вибраний, пункт (item) якого видно в полі, а інші з'являються при натисканні кнопкою миші на невеликій кнопці праворуч від поля компонента.

Choice конструктор

Спочатку конструктором `Choice()` створюється порожній список.

Потім, методом `add(String text)`, в список додаються нові пункти з текстом `text`. Вони розташовуються в порядку написання методів `add()` і нумеруються від нуля.

Choice методи

`insert(String text, int position)` - вставляє новий пункт у потрібне місце

`select(String text)` - вибір пункту по значенню

`select(int position)` – вибір пункту по позиції

`remove(String text)` - видаляє один пункт зі списку по значенню

`remove(int position)` - видаляє один пункт зі списку по позиції

`removeAll()` - видаляє всі пункти відразу.

`getItemCount()` – повертає число пунктів в списку.

З'ясувати, який пункт знаходиться в позиції `pos` можна методом `getItem(int pos)`, що повертає рядок.

`getSelectedIndex()` – визначення вибраного пункту, повертає позицію цього пункту,

`getSelectedItem()` – визначення вибраного пункту, повертає виділений рядок.

Choice події

У класі Choice відбуваються події класу Component: ComponentEvent, FocusEvent, KeyEvent, MouseEvent, а при виборі пункту виникає подія ItemEvent.

List

Компонент List - це список зі смугою прокрутки, в якому можна виділити один або декілька пунктів.

Кількість видимих на екрані пунктів визначається конструктором списку і розміром компонента.

List конструктори

List() - створює порожній список з чотирма видимими пунктами;

List(int rows) - створює порожній список з rows видимими пунктами;

List(int rows, boolean multiple) - створює порожній список в якому можна відзначити кілька пунктів, якщо multiple == true.

List методи

Після створення об'єкту в список додаються пункти з текстом item:

`add(String item)` - додає новий пункт в кінець списку;

`add(String item, int position)` - додає новий пункт в позицію position.

Позиції нумеруються по порядку, починаючи з нуля.

Видалити пункт можна методами `remove (String item)`, `remove (int position)`, `removeAll ()`.

`replaceAll(String newItem, int pos)` - дозволяє замінити текст пункту в позиції pos.

`getItemCount()` - повертає кількість пунктів у списку

Виділений пункт можна отримати методом `getSelectedItem()`, а його позицію - методом `getSelectedIndex()`.

Якщо список дозволяє здійснити множинний вибір, то виділені пункти у вигляді масиву типу `string []` можна отримати методом `getSelectedItems()`, позиції виділених пунктів у вигляді масиву типу `int []` - методом `getSelectedIndexes()`.

Крім цих необхідних методів клас List містить безліч інших, що дозволяють маніпулювати пунктами списку і отримувати його характеристики.

List події

Крім подій класу Component: `ComponentEvent`, `FocusEvent`, `KeyEvent`, `MouseEvent`, при подвійному натисканні кнопкою миші на вибраному пункті відбувається подія `ActionEvent`.

Приклад

```
import java.awt.*;
import java.awt.event.*;
class ListTest extends Frame{
    ListTest(String s){
        super(s);
        setLayout(null);
        setFont(new Font("Serif", Font.BOLD, 15));
        Label l1 = new Label("Выберите товар:", Label.CENTER);
        l1.setBounds(0, 50, 120, 30); add (l1);
        Label l2 = new Label("Выберите способ оплаты:");
        l2.setBounds(170, 50, 200, 30); add(l2);
        List l = new List(2, true);
        l.add("Книги");
        l.add("Диски");
        l.add("Игрушки");
        l.setBounds(20, 90, 100, 40); add(l);
        Choice ch = new Choice();
        ch.add("Почтовым переводом");
        ch.add("Кредитной картой");
        ch.setBounds(170, 90, 200,30); add(ch);
    }
}
```

```
    Button b1 = new Button("Продолжить");
    b1.setBounds(30, 150, 100, 30); add(b1);
    Button b2 = new Button("Отменить");
    b2.setBounds(140, 150, 100, 30); add(b2);
    Button b3 = new Button("Выйти");
    b3.setBounds(250, 150, 100, 30); add(b3);
    setSize(400, 200); setVisible(true);
}
public static void main(String[] args){
    Frame f = new ListTest(" Простые компоненты");
    f.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent ev){
            System.exit(0);
        }
    });
}
}
```

TextField

Компонент TextField - це поле для введення одного рядка тексту. Ширина поля вимірюється в колонках (column). Ширина колонки - це середня ширина символу в шрифті, яким вводиться текст. Натискання клавіші <Enter> закінчує введення і служить сигналом до початку обробки введеного тексту, тобто при цьому відбувається подія ActionEvent.

TextField конструктори

`TextField()` - створює порожнє поле шириною в одну колонку;

`TextField(int columns)` - створює порожнє поле з числом колонок `columns`;

`TextField(String text)` - створює поле з текстом `text`;

`TextField(String text, int columns)` - створює поле з текстом `text` і числом колонок `columns`.

TextField методи

`getColumns()` – повертає кількість колонок в текстовому полі

`setColumns(int col)` – змінює кількість колонок в текстовому полі.

Різновид поля вводу - поле для введення пароля. В такому полі замість символів, що вводять з'являється якийсь особливий символ, найчастіше зірочка.

Дане поле введення отримують виконанням методу `setEchoChar(char echo)`. Аргумент `echo` - це символ, який буде з'являтися в поле. Перевірити, чи встановлений символ, можна логічним методом `echoCharisSet()`, отримати символ - методом `getEchoChar()`.

Щоб повернути поле введення в звичайний стан, досить виконати метод `setEchoChar(0)`.

TextField події

Крім подій класу Component: ComponentEvent, FocusEvent, KeyEvent, MouseEvent, при зміні тексту користувачем відбувається подія TextEvent, а при натисненні клавіші <Enter> - подія ActionEvent.

TextArea

Компонент TextArea - це область введення з довільним числом рядків. Натискання клавіші <Enter> просто переводить курсор в початок наступного рядка. В області введення можуть бути встановлені лінійки прокрутки, одна чи обидві.

TextArea конструктори

TextArea (String text, int rows, int columns, int scrollbars) - створює область введення з текстом text, числом видимих рядків rows, числом колонок columns, і завданням смуг. прокрутки scrollbars однієї з чотирьох констант: SCROLLBARS_NONE, SCROLLBARS_HORIZONTAL_ONLY, SCROLLBARS_VERTICAL_ONLY, SCROLLBARS_BOTH.

TextArea (String text, int rows, int columns) - присутні обидві смуги прокрутки;

TextArea (int rows, int columns) - в поле порожній рядок;

TextArea (String text) - розміри встановлює контейнер;

TextArea () - конструктор за умовчанням.

TextArea методи

`append (String text)` - додає текст `text` в кінець вже введеного тексту;

`insert (String text, int pos)` – вставляє текст в зазначену позицію `pos`;

`replaceRange (String text, int begin, int end)` - видаляє текст починаючи з індексу `begin` включно по `end` виключно, і поміщає замість нього текст `text`.

Інші методи дозволяють змінити і отримати кількість видимих рядків.

TextArea події

Крім Подій класу Component:
ComponentEvent, FocusEvent, KeyEvent,
MouseEvent, при зміні тексту користувачем
відбувається подія TextEvent.