

INTERFEȚE GRAFICE

1. Noțiuni introductive

Interfețele grafice (Graphical User Interfaces) reprezintă ferestre ce conțin elemente grafice ce permit interacțiunea dintre aplicație și utilizator. Limbajul Java pune la dispoziție numeroase clase pentru implementarea diverselor funcționalități UI, în continuare fiind prezentate două biblioteci pentru realizarea interfețelor grafice: **java.awt** și **java.swing**.

Crearea unei aplicații grafice presupune:

- Crearea unei suprafețe de afișare (cum ar fi o fereastră) pe care vor fi poziționate obiectele grafice (butoane, controale de editare, texte, etc);
- Crearea și poziționarea obiectelor grafice pe suprafața de afișare pe pozițiile corespunzătoare;
- Definirea unor acțiuni care trebuie să se execute în momentul când utilizatorul interacționează cu obiectele grafice ale aplicației;
- „Ascultarea” evenimentelor generate de obiecte în momentul interacțiunii cu utilizatorul și executarea acțiunilor corespunzătoare așa cum au fost ele definite.

Crearea obiectelor grafice în Java nu se afișează automat pe ecran. Pentru a deveni vizibile, aceste obiecte trebuie mai întâi plasate pe o suprafață de afișare, numită **container** (cum ar fi o **fereastră** sau **suprafața unui applet**) și reprezintă o instanță a unei clase obținută prin extensia superclasei *Container*. Componentelor li se va putea vizualiza conținutul doar după ce containerul în care se află este vizibil.

În Java, toate componentele grafice sunt definite în pachetul **java.awt**, iar clasa abstractă **Component** reprezintă superclasa acestora. Suprafețele de afișare (containerele) sunt instanțe ale unor clase construite prin extinderea superclasei **JContainer**. Totuși, nu toate containerele pot apărea direct pe ecran, unele trebuie integrate în alte containere pentru a putea fi afișate.

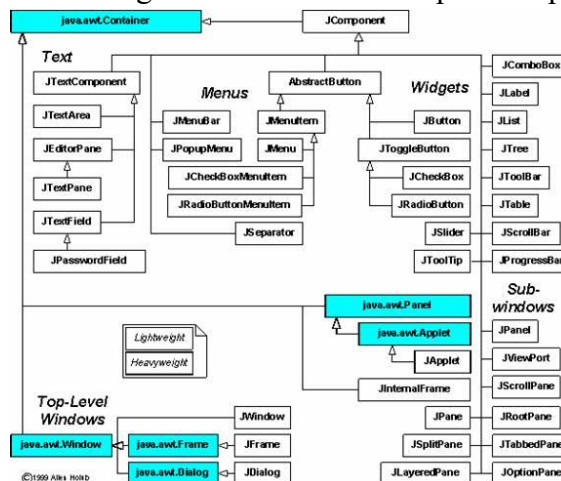


Figura 1. Ierarhia claselor Swing

2. Containere și componente grafice

Elementele grafice sunt de două tipuri:

1. Componente: obiecte care au o reprezentare grafică ce poate fi afișată pe ecran și care poate interacționa cu utilizatorul. Exemple: **JButtons**, **JLabels**, **TextField**.

2. Containere: componente care pot conține alte componente. În Swing, containerele precum **JFrame**, **JPanel**, sunt folosite pentru a conține componente. Componentele se pot aranja într-un anumit fel în interiorul unui container specificând tipul de *layout* dorit (*FlowLayout*, *GridLayout*, *BoxLayout*). Un container poate conține alte containere (subcontainere).

În figura 2 este prezentat un exemplu pentru elementele grafice de bază:

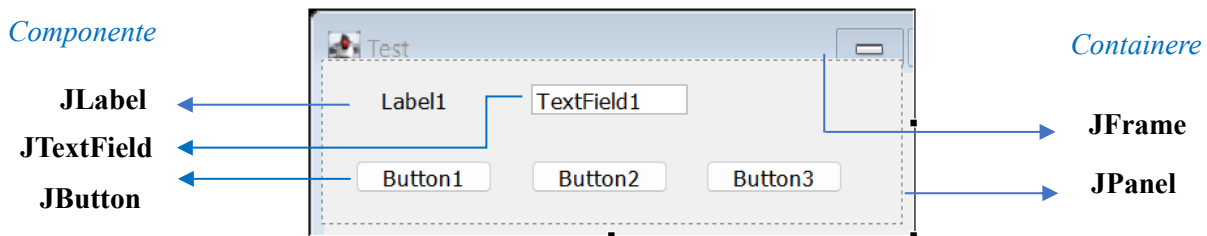


Figura 2. Exemplu de interfață grafică

2.1 Aranjarea componentelor în containere

Pentru a stabili locația componentelor în containere, trebuie specificat modul de aranjare al lor (layout). Există mai multe tipuri de gestionare de aranjare:

- *FlowLayout* (aranjarea implicită pentru **JPanel**) – aranjează elementele unul după altul

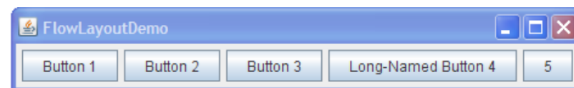


Figura 3. Exemplu de aranjare cu *FlowLayout*

```
FlowLayout experimentLayout = new FlowLayout();
```

...

```
compsToExperiment.setLayout(experimentLayout);  
compsToExperiment.add(new JButton("Button 1"));  
compsToExperiment.add(new JButton("Button 2"));  
compsToExperiment.add(new JButton("Button 3"));  
compsToExperiment.add(new JButton("Long-Named Button 4"));  
compsToExperiment.add(new JButton("5"));
```

- *GridLayout* – plasează componentele într-o matrice specificată prin numărul de linii și de coloane



Figura 4. Exemplu de aranjare cu GridLayout

```
GridLayout experimentLayout = new GridLayout(0,2);
```

...

```
compsToExperiment.setLayout(experimentLayout);
compsToExperiment.add(new JButton("Button 1"));
compsToExperiment.add(new JButton("Button 2"));
compsToExperiment.add(new JButton("Button 3"));
compsToExperiment.add(new JButton("Long-Named Button 4"));
compsToExperiment.add(new JButton("5"));
```

- *BorderLayout* – plasează componential în până la cinci zone diferite: est, vest, sud, nord și centru; spațiul extra este plasat în zona centrală.

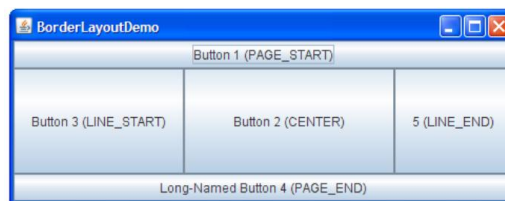


Figura 5. Exemplu de aranjare cu BorderLayout

....

```
Container pane = aFrame.getContentPane();
JButton button = new JButton("Button 1 (PAGE_START)");
pane.add(button, BorderLayout.PAGE_START);
//Make the center component big, since that's the //typical usage of
BorderLayout. button = new JButton("Button 2 (CENTER)");
button.setPreferredSize(new Dimension(200, 100));
pane.add(button, BorderLayout.CENTER);
```

3. Tratarea evenimentelor

3.1 Evenimente

Evenimentele vin de la controalele cu care interacționează utilizatorul. Când definim o interfață utilizator, e nevoie de obicei de o cale pentru a obține informație de la utilizator. Butoanele, meniurile, slider-ele, click-urile pe mouse, etc., generează evenimente atunci când utilizatorul interacționează cu ele. Obiectele eveniment din interfața utilizator sunt transmise de la o sursă de evenimente – cum sunt un buton sau un click pe mouse – la un ascultător de evenimente – o metodă a utilizatorului care va prelucra obiectele eveniment.

Fiecare control pentru intrare (*JButton*, *JSlider*, ...) are nevoie de un ascultător de evenimente.

Sursa evenimentului	Metoda de adăugare a ascultătorului	Metoda ascultătorului
<i>JButton</i> <i>JTextField</i> <i>JMenuItem</i>	<code>addActionListener()</code>	<code>actionPerformed(ActionEvent e)</code>
<i>JSlider</i>	<code>addChangeListener()</code>	<code>stateChanged(ChangeEvent e)</code>
<i>JCheckBox</i>	<code>addItemListener()</code>	<code>itemStateChanged()</code>
Tasta pe componentă	<code>addKeyListener()</code>	<code>keyPressed()</code> , <code>keyReleased()</code> , <code>keyTyped()</code>
Mouse pe componenta	<code>addMouseListener()</code>	<code>mouseClicked()</code> , <code>mouseEntered()</code> , <code>mouseExited()</code> , <code>mousePressed()</code> , <code>mouseReleased()</code>
Mouse mișcat pe componentă	<code>addMouseMotionListener()</code>	<code>mouseMoved()</code> , <code>mouseDragged()</code>
<i>JFrame</i>	<code>addWindowListener()</code>	<code>windowClosing(WindowEvent e)</code>

Tabelul 1. Surse de evenimente, metoda de adăugare a ascultătorului și metoda invocată în ascultător

3.2 Ascultători

Un ascultător este apelat atunci când utilizatorul interacționează cu interfața, ceea ce provoacă un eveniment. Multe dintre evenimente provin de obicei din interfața utilizator, dar ele pot avea și alte surse (Timer). După crearea unui buton, adaugă un ascultător,

```
btn.addActionListener(obiect_ascultator);
```

```
//unde obiect_ascultator este de tipul ButtonListener definit mai jos
```

La click pe buton se invocă metoda *actionPerformed()* definită în clasa obiectului ascultător. Metodei i se transmite ca parametru un obiect *ActionEvent*.

Exemplu de clasă care implementează un ascultător:

```
class ButtonListener implements ActionListener
```

```
{public void actionPerformed(ActionEvent e) {
```

```
//fa ceva cand se apasa butonul, de exemplu ++count; tf.setText(count + ""); }}
```

1. Realizați o aplicație Java cu interfață grafică (GUI) care să conțină o fereastră principală în care utilizatorul poate introduce numele și vârsta sa, folosind două câmpuri text (JTextField). În dreptul fiecărui câmp trebuie să fie afișată o etichetă (JLabel) corespunzătoare celor două câmpuri. În partea inferioară a ferestrei vor fi amplasate două butoane: butonul „Ok” și butonul „Reset”.

```
test1.java  *MainFrame.java  calculator.java
1  package grafica;
2
3  import java.awt.EventQueue;
14
15  public class MainFrame extends JFrame {
16
17      private static final long serialVersionUID = 1L;
18      private JPanel contentPane;
19      private JTextField txtName;
20      private JTextField txtAge;
21      JLabel lblInfo;
22      /**
23       * Launch the application.
24       */
25      public static void main(String[] args) {
26          EventQueue.invokeLater(new Runnable() {
27              public void run() {
28                  try {
29                      MainFrame frame = new MainFrame();
30                      frame.setVisible(true);
31                  } catch (Exception e) {
32                      e.printStackTrace();
33                  }
34              }
35          });
36
37      /**
38       * Create the frame.
39       */
40      public MainFrame() {
41          setTitle("Demo");
42          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
43          setBounds(100, 100, 450, 300);
44          contentPane = new JPanel();
45          contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
46          setContentPane(contentPane);
47          contentPane.setLayout(null);
48
49          JLabel lblNewLabel = new JLabel("Introdu numele");
50          lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 12));
51          lblNewLabel.setBounds(25, 10, 84, 12);
52          contentPane.add(lblNewLabel);
53
54          JLabel lblNewLabel_1 = new JLabel("Introdu varsta");
55          lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 12));
56          lblNewLabel_1.setBounds(25, 54, 84, 16);
57          contentPane.add(lblNewLabel_1);
58
59          txtName = new JTextField();
60          txtName.setBounds(119, 8, 96, 18);
61          contentPane.add(txtName);
62          txtName.setColumns(10);
63          txtAge = new JTextField();
64          txtAge.setBounds(119, 52, 96, 18);
65          contentPane.add(txtAge);
66          txtAge.setColumns(10);
67
```

```

68      JButton btnOk = new JButton("Ok");
69      btnOk.addActionListener(new ActionListener() {
70          public void actionPerformed(ActionEvent e) {
71              String n=txtName.getText();
72              txtAge.getText();
73              int age=Integer.parseInt(txtAge.getText());
74              if(age>=18)
75                  lblInfo.setText(n + " Esti eligil sa votezi!");
76              else
77                  lblInfo.setText(n + " Nu esti eligil sa votezi!");
78          }
79      });
80      btnOk.setFont(new Font("Tahoma", Font.BOLD, 12));
81      btnOk.setBounds(25, 93, 84, 20);
82      contentPane.add(btnOk);
83
84      JButton btnReset = new JButton("Reset");
85      btnReset.addActionListener(new ActionListener() {
86          public void actionPerformed(ActionEvent e) {
87              txtName.setText("");
88              txtAge.setText("");
89              lblInfo.setText("");
90          }
91      });
92      btnReset.setFont(new Font("Tahoma", Font.BOLD, 12));
93      btnReset.setBounds(198, 93, 84, 20);
94      contentPane.add(btnReset);
95
96      lblInfo = new JLabel("");
97      lblInfo.setBounds(89, 185, 193, 12);
98      contentPane.add(lblInfo);
99  }}

```

2. Realizați o interfață grafică pentru o aplicație de tip calculator, similară cu cea din Windows. Interfața trebuie să includă butoane pentru cifre (0-9), butoane pentru operațiile aritmetice (+, -, *, /), o zonă de afișare a rezultatelor, precum și alte butoane funcționale necesare efectuării calculelor aritmetice.

```

1  package grafica;
2
3  import java.awt.EventQueue;
12
13  public class calculator {
14
15      private JFrame CALCULATOR;
16      private JTextField textField;
17      double first;
18      double second;
19      double result;
20      String operation;
21      String answer;
22
23      /**
24       * Launch the application.
25       */
26      public static void main(String[] args) {
27          EventQueue.invokeLater(new Runnable() {
28              public void run() {
29                  try {
30                      calculator window = new calculator();
31                      window.CALCULATOR.setVisible(true);
32                  } catch (Exception e) {
33                      e.printStackTrace();
34                  }
35              }
36          });
37      }

```

```

39  /**
40   * Create the application.
41   */
42  public calculator() {
43      initialize();
44  }
45
46  /**
47   * Initialize the contents of the frame.
48   */
49  private void initialize() {
50      CALCULATOR = new JFrame();
51      CALCULATOR.setTitle("Calculator");
52      CALCULATOR.setBounds(100, 100, 384, 484);
53      CALCULATOR.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
54      CALCULATOR.getContentPane().setLayout(null);
55
56      textField = new JTextField();
57      textField.setFont(new Font("Tahoma", Font.BOLD, 12));
58      textField.setBounds(10, 22, 350, 83);
59      CALCULATOR.getContentPane().add(textField);
60      textField.setColumns(10);
61
62      JButton btnPercent = new JButton("%");
63      btnPercent.setFont(new Font("Tahoma", Font.BOLD, 12));
64      btnPercent.addActionListener(new ActionListener() {
65          public void actionPerformed(ActionEvent e) {
66              first=Double.parseDouble(textField.getText());
67              textField.setText("");
68              operation="%";
69          }
70      });
71
72      btnPercent.setBounds(275, 383, 85, 54);
73      CALCULATOR.getContentPane().add(btnPercent);
74
75      JButton btn7 = new JButton("7");
76      btn7.addActionListener(new ActionListener() {
77          public void actionPerformed(ActionEvent e) {
78              String number=textField.getText()+btn7.getText();
79              textField.setText(number);
80          }
81      });
82      btn7.setFont(new Font("Tahoma", Font.BOLD, 12));
83      btn7.setBounds(10, 198, 85, 54);
84      CALCULATOR.getContentPane().add(btn7);
85
86      JButton btn4 = new JButton("4");
87      btn4.setFont(new Font("Tahoma", Font.BOLD, 12));
88      btn4.addActionListener(new ActionListener() {
89          public void actionPerformed(ActionEvent e) {
90              String number=textField.getText()+btn4.getText();
91              textField.setText(number);
92          }
93      });
94      btn4.setBounds(10, 257, 85, 54);
95      CALCULATOR.getContentPane().add(btn4);
96
97      JButton btn1 = new JButton("1");
98      btn1.addActionListener(new ActionListener() {
99          public void actionPerformed(ActionEvent e) {
100              String number=textField.getText()+btn1.getText();
101              textField.setText(number);
102          }
103      });

```

```

103 btn1.setFont(new Font("Tahoma", Font.BOLD, 12));
104 btn1.setBounds(10, 321, 85, 54);
105 CALCULATOR.getContentPane().add(btn1);
106
107 JButton btn0 = new JButton("0");
108 btn0.setFont(new Font("Tahoma", Font.BOLD, 12));
109 btn0.addActionListener(new ActionListener() {
110     public void actionPerformed(ActionEvent e) {
111         String number=textField.getText()+btn0.getText();
112         textField.setText(number);
113     }
114 });
115 btn0.setBounds(10, 383, 85, 54);
116 CALCULATOR.getContentPane().add(btn0);
117
118 JButton btnBack = new JButton("\uF0E7");
119 btnBack.addActionListener(new ActionListener() {
120     public void actionPerformed(ActionEvent e) {
121         String backSpace=null;
122         if(textField.getText().length()>0);
123         {
124             StringBuilder str=new StringBuilder(textField.getText());
125             str.deleteCharAt(textField.getText().length()-1);
126             backSpace=str.toString();
127             textField.setText(backSpace);
128         }
129     }
130 });
131 btnBack.setFont(new Font("Windings", Font.BOLD, 12));
132 btnBack.setBounds(10, 134, 85, 54);
133 CALCULATOR.getContentPane().add(btnBack);
134

```

```

136 JButton btn8 = new JButton("8");
137 btn8.addActionListener(new ActionListener() {
138     public void actionPerformed(ActionEvent e) {
139         String number=textField.getText()+btn8.getText();
140         textField.setText(number);
141     }
142 });
143 btn8.setFont(new Font("Tahoma", Font.BOLD, 12));
144 btn8.setBounds(97, 198, 85, 54);
145 CALCULATOR.getContentPane().add(btn8);
146
147 JButton btn5 = new JButton("5");
148 btn5.addActionListener(new ActionListener() {
149     public void actionPerformed(ActionEvent e) {
150         String number=textField.getText()+btn5.getText();
151         textField.setText(number);
152     }
153 });
154 btn5.setFont(new Font("Tahoma", Font.BOLD, 12));
155 btn5.setBounds(97, 257, 85, 54);
156 CALCULATOR.getContentPane().add(btn5);
157
158 JButton btn2 = new JButton("2");
159 btn2.setFont(new Font("Tahoma", Font.BOLD, 12));
160 btn2.addActionListener(new ActionListener() {
161     public void actionPerformed(ActionEvent e) {
162         String number=textField.getText()+btn2.getText();
163         textField.setText(number);
164     }
165 });

```



```

166 btn2.setBounds(97, 321, 85, 54);
167 CALCULATOR.getContentPane().add(btn2);
168
169 JButton btn00 = new JButton("00");
170 btn00.setFont(new Font("Tahoma", Font.BOLD, 12));
171 btn00.addActionListener(new ActionListener() {
172     public void actionPerformed(ActionEvent e) {
173         String number=textField.getText()+btn00.getText();
174         textField.setText(number);
175     }
176 });
177 btn00.setBounds(188, 134, 85, 54);
178 CALCULATOR.getContentPane().add(btn00);
179
180 JButton btnClear = new JButton("C");
181 btnClear.addActionListener(new ActionListener() {
182     public void actionPerformed(ActionEvent e) {
183         textField.setText(null);
184     }
185 });
186 btnClear.setFont(new Font("Tahoma", Font.BOLD, 12));
187 btnClear.setBounds(97, 134, 85, 54);
188 CALCULATOR.getContentPane().add(btnClear);
189
190 JButton btn9 = new JButton("9");
191 btn9.addActionListener(new ActionListener() {
192     public void actionPerformed(ActionEvent e) {
193         String number=textField.getText()+btn9.getText();
194         textField.setText(number);
195     }
196 });

```

```

197 btn9.setFont(new Font("Tahoma", Font.BOLD, 12));
198 btn9.setBounds(188, 198, 85, 54);
199 CALCULATOR.getContentPane().add(btn9);
200
201 JButton btn6 = new JButton("6");
202 btn6.addActionListener(new ActionListener() {
203     public void actionPerformed(ActionEvent e) {
204         String number=textField.getText()+btn6.getText();
205         textField.setText(number);
206     }
207 });
208 btn6.setFont(new Font("Tahoma", Font.BOLD, 12));
209 btn6.setBounds(188, 257, 85, 54);
210 CALCULATOR.getContentPane().add(btn6);
211
212 JButton btn3 = new JButton("3");
213 btn3.addActionListener(new ActionListener() {
214     public void actionPerformed(ActionEvent e) {
215         String number=textField.getText()+btn3.getText();
216         textField.setText(number);
217     }
218 });
219 btn3.setFont(new Font("Tahoma", Font.BOLD, 12));
220 btn3.setBounds(188, 321, 85, 54);
221 CALCULATOR.getContentPane().add(btn3);
222 JButton btnDot = new JButton(".");
223 btnDot.addActionListener(new ActionListener() {
224     public void actionPerformed(ActionEvent e) {
225         String number=textField.getText()+btnDot.getText();
226         textField.setText(number);
227     }
228 });

```

```

229 btnDot.setFont(new Font("Tahoma", Font.BOLD, 12));
230 btnDot.setBounds(97, 383, 85, 54);
231 CALCULATOR.getContentPane().add(btnDot);
232
233 JButton btnPlus = new JButton("+");
234 btnPlus.addActionListener(new ActionListener() {
235     public void actionPerformed(ActionEvent e) {
236         first=Double.parseDouble(textField.getText());
237         textField.setText("");
238         operation="+";
239     }
240 });
241 btnPlus.setFont(new Font("Tahoma", Font.BOLD, 12));
242 btnPlus.setBounds(275, 134, 85, 54);
243 CALCULATOR.getContentPane().add(btnPlus);
244
245 JButton btnMinus = new JButton("-");
246 btnMinus.addActionListener(new ActionListener() {
247     public void actionPerformed(ActionEvent e) {
248         first=Double.parseDouble(textField.getText());
249         textField.setText("");
250         operation="-";
251     }
252 });
253 btnMinus.setFont(new Font("Tahoma", Font.BOLD, 12));
254 btnMinus.setBounds(275, 198, 85, 54);
255 CALCULATOR.getContentPane().add(btnMinus);

```

```

257 JButton btnMul = new JButton("*");
258 btnMul.addActionListener(new ActionListener() {
259     public void actionPerformed(ActionEvent e) {
260         first=Double.parseDouble(textField.getText());
261         textField.setText("");
262         operation="*";
263     }
264 });
265 btnMul.setFont(new Font("Tahoma", Font.BOLD, 12));
266 btnMul.setBounds(275, 257, 85, 54);
267 CALCULATOR.getContentPane().add(btnMul);
268
269 JButton btnDiv = new JButton("/");
270 btnDiv.addActionListener(new ActionListener() {
271     public void actionPerformed(ActionEvent e) {
272         first=Double.parseDouble(textField.getText());
273         textField.setText("");
274         operation="/";
275     }
276 });
277 btnDiv.setFont(new Font("Tahoma", Font.BOLD, 12));
278 btnDiv.setBounds(275, 321, 85, 54);
279 CALCULATOR.getContentPane().add(btnDiv);
280
281 JButton btnEgal = new JButton("=");
282 btnEgal.addActionListener(new ActionListener() {
283     public void actionPerformed(ActionEvent e) {
284         String answer;
285         second=Double.parseDouble(textField.getText());
286         if (operation=="+")
287         {
288             result=first+second;

```

```

288         result=first+second;
289         answer=String.format("%.2f", result);
290         textField.setText(answer);
291     }
292     }
293     else if (operation=="-")
294     {
295         result=first-second;
296         answer=String.format("%.2f", result);
297         textField.setText(answer);
298     }
299     }
300     else if (operation=="*")
301     {
302         result=first*second;
303         answer=String.format("%.2f", result);
304         textField.setText(answer);
305     }
306     }
307     }
308     else if (operation=="/")
309     {
310         result=first/second;
311         answer=String.format("%.2f", result);
312         textField.setText(answer);
313     }
314     }
315     }
316     });
317
318     btnEgal.setFont(new Font("Tahoma", Font.BOLD, 12));
319     btnEgal.setBounds(188, 383, 85, 54);
320     CALCULATOR.getContentPane().add(btnEgal);
321 }
322
323 }

```