

```

1 package pl.nalazek.komunikator.gui;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import javax.swing.JComponent;
6 import javax.swing.JDialog;
7 import javax.swing.JLabel;
8 import javax.swing.JOptionPane;
9 import javax.swing.JTextField;
10 import javax.swing.SwingUtilities;
11 import javax.swing.event.ListSelectionEvent;
12 import javax.swing.event.ListSelectionListener;
13 import pl.nalazek.komunikator.Program;
14 import pl.nalazek.komunikator.logika.Watek;
15
16 /**
17  * @author Daniel Nalazek
18  * Copyright (C) 2014 Daniel Nalazek
19  */
20
21 /** Klasa reprezentująca zarządcę obsługi panelu rozmowa */
22 public class OknoPanelUzytkownicy implements ActionListener, ListSelectionListener{
23     private Gui gui;
24
25     /** Konstruktor
26      * @param gui Referencja do interfejsu graficznego programu */
27     public OknoPanelUzytkownicy(Gui gui)
28     {
29         this.gui = gui;
30         JDialog.setDefaultLookAndFeelDecorated(true);
31     }
32
33     /** Funkcja realizująca interfejs ActionListener, uruchamiana na skutek zdarzenia przychodzące z panelu
34     użytkownicy */
35     public void actionPerformed(ActionEvent arg0)
36     {
37         if(arg0.getActionCommand() == "Dodaj")
38         {
39             SwingUtilities.invokeLater(new Runnable()
40             {
41                 public void run()
42                 {
43                     String[] dane = oknoDialogoweNowyUzytkownik();
44                     if(dane != null) Program.komponentSterowanie().dodanoUzytkownikaZdalnego(dane);
45                 }
46             });
47         }
48         else if(arg0.getActionCommand() == "Usu\u0144")
49         {
50             if(!gui.listaUzytkownikow.isSelectedEmpty())
51                 (new Watek(){public void run(){
52                     Program.komponentSterowanie().usunietoUzytkownikaZdalnego(gui);
53                 }}).start();
54         }
55         else if(arg0.getActionCommand() == "Po\u0142\u0105cz")
56         {
57             if(!gui.listaUzytkownikow.isSelectedEmpty())
58             {
59                 (new Watek(){public void run(){
60                     Program.komponentSterowanie().zadaniePolaczenia(gui);
61                 }}).start();
62             }
63         }
64
65         else if(arg0.getActionCommand() == "Roz\u0142\u0105cz")
66         {
67             if(!gui.listaUzytkownikow.isSelectedEmpty())
68                 (new Watek(){public void run(){
69                     Program.komponentSterowanie().zadanieRozlaczenia(gui);
70                 }}).start();
71         }
72     }
73
74     /** Funkcja realizująca interfejs ListSelectionListener, uruchamiana na skutek zdarzenia przychodzące z
75     panelu użytkownicy */
76

```

```

77      * @param arg0 Referencja do zdarzenia */
78      public void valueChanged(ListSelectionEvent arg0) {
79          for(int i = 0; i <= arg0.getLastIndex(); i++)
80              {
81                  if(gui.listaUzytkownikow.isSelectedIndex(i))
82                      gui.listaOstatniIndeks = i;
83              }
84
85          Program.komponentSterowanie().zmienionoStanListy(gui);
86      }
87
88      /** Funkcja uruchamia okno dialogowe do wpisywania danych nowego uzytkownika */
89      private String[] oknoDialogoweNowyUzytkownik()
90      {
91
92          JTextField nazwa = new JTextField();
93          JTextField ip = new JTextField();
94          JComponent[] wejscie = new JComponent[]
95              {
96                  new JLabel("Nazwa u\u017cytkownika: "), nazwa,
97                  new JLabel("Adres IP (dla trybu lokalnego \'localhost\'): "), ip
98              };
99          if(JOptionPane.showConfirmDialog(gui, wejscie, "Wprowad\u017a dane", JOptionPane.OK_CANCEL_OPTION) ==
100             JOptionPane.OK_OPTION)
101          {
102              String ipp = ip.getText();
103              boolean zlaWartosc = false;
104
105              if(ipp.length() == 15)
106              {
107                  ipp = ipp.substring(0, 15);
108                  if(ipp.charAt(3) == '.' && ipp.charAt(7) == '.' && ipp.charAt(11) == '.')
109                  {
110                      for(int i = 0; i < ipp.length(); i++ )
111                      {
112                          if(i == 3 || i==7 || i==11) continue;
113                          if(ipp.charAt(i) > 47 && ipp.charAt(i) < 58)
114                              continue;
115                          else zlaWartosc = true;
116                      }
117
118                  }
119                  else zlaWartosc = true;
120              }
121              else zlaWartosc = true;
122              if(nazwa.getText().length() == 0) {
123                  JOptionPane.showMessageDialog(gui, "Wprowad\u017a nazw\u0119
u\u017cytkownika!",
124                  "Wprowadzono b\u0119dne dane", JOptionPane.WARNING_MESSAGE);
125                  return null;
126              }
127
128              if(ipp.matches("localhost")) return new String[] { nazwa.getText(), "localhost" };
129              if(zlaWartosc)
130              {
131                  JOptionPane.showMessageDialog(gui, "Wprowadzono adres ip o b\u0142\u0119dnym formacie."
132                  + "\nPoprawny format to xxx.xxx.xxx.xxx gdzie \'x\' oznacza liczb\u0119",
133                  "Wprowadzono b\u0119dne dane", JOptionPane.WARNING_MESSAGE);
134                  return null;
135              }
136
137              return new String[] { nazwa.getText(), ip.getText() };
138          }
139          return null;
140      }
141
142
143 }
144
145

```