

```

1  package org.Speedy.SearchPathCount;
2
3  import java.util.ArrayList;
4  import org.gephi.data.attributes.api.AttributeColumn;
5  import org.gephi.data.attributes.api.AttributeRow;
6  import org.gephi.graph.api.Graph;
7  import org.gephi.graph.api.GraphController;
8  import org.gephi.graph.api.GraphModel;
9  import org.gephi.preview.api.PreviewController;
10 import org.gephi.preview.api.PreviewModel;
11 import org.gephi.preview.api.PreviewProperties;
12 import org.netbeans.api.settings.ConvertAsProperties;
13 import org.openide.awt.ActionID;
14 import org.openide.awt.ActionReference;
15 import org.openide.util.Lookup;
16 import org.openide.windows.TopComponent;
17 import org.openide.util.NbBundle.Messages;
18
19 /**
20  * Top component which displays something.
21  */
22 @ConvertAsProperties(
23     dtd = "-//org.Speedy.SearchPathCount//SearchPathCountModual//EN",
24     autostore = false
25 )
26 @TopComponent.Description(
27     preferredID = "SearchPathCountModualTopComponent",
28     //iconBase="SET/PATH/TO/ICON/HERE",
29     persistenceType = TopComponent.PERSISTENCE_ALWAYS
30 )
31 @TopComponent.Registration(mode = "rankingmode", openAtStartup = true)
32 @ActionID(category = "Window", id =
33     "org.Speedy.SearchPathCount.SearchPathCountModualTopComponent")
34 @ActionReference(path = "Menu/Window" /*, position = 333 */)
35 @TopComponent.OpenActionRegistration(
36     displayName = "#CTL_SearchPathCountModualAction",
37     preferredID = "SearchPathCountModualTopComponent"
38 )
39 @Messages({
40     "CTL_SearchPathCountModualAction=SearchPathCountModual",
41     "CTL_SearchPathCountModualTopComponent=SearchPathCountModual Window",
42     "HINT_SearchPathCountModualTopComponent=This is a
43     SearchPathCountModual window"
44 })
45 public final class SearchPathCountModualTopComponent extends TopComponent
46 {
47
48     public SearchPathCountModualTopComponent() {
49         initComponents();
50         setName(Bundle.CTL_SearchPathCountModualTopComponent());
51         setToolTipText(Bundle.HINT_SearchPathCountModualTopComponent());
52         putClientProperty(TopComponent.PROP_MAXIMIZATION_DISABLED,
53             Boolean.TRUE);
54         putClientProperty(TopComponent.PROP_UNDOCKING_DISABLED,
55             Boolean.TRUE);
56
57     }
58
59     /**
60      * This method is called from within the constructor to initialize
61      * the form.

```

```

57      * WARNING: Do NOT modify this code. The content of this method is
58      always
59      * regenerated by the Form Editor.
60      */
61      // <editor-fold defaultstate="collapsed" desc="Generated Code">
62      private void initComponents() {
63          jLabel1 = new javax.swing.JLabel();
64          jLabel2 = new javax.swing.JLabel();
65          jComboBox1 = new javax.swing.JComboBox();
66          jComboBox2 = new javax.swing.JComboBox();
67          jButton1 = new javax.swing.JButton();
68          jButton3 = new javax.swing.JButton();
69
70          org.openide.awt.Mnemonics.setLocalizedText(jLabel1,
71          org.openide.util.NbBundle.getMessage(SearchPathCountModualTopComp
72          onent.class, "SearchPathCountModualTopComponent.jLabel1.text"));
73          // NOI18N
74
75          org.openide.awt.Mnemonics.setLocalizedText(jLabel2,
76          org.openide.util.NbBundle.getMessage(SearchPathCountModualTopComp
77          onent.class, "SearchPathCountModualTopComponent.jLabel2.text"));
78          // NOI18N
79
80          jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new
81          String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
82
83          jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new
84          String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
85
86          org.openide.awt.Mnemonics.setLocalizedText(jButton1,
87          org.openide.util.NbBundle.getMessage(SearchPathCountModualTopComp
88          onent.class, "SearchPathCountModualTopComponent.jButton1.text"));
89          // NOI18N
90          jButton1.addActionListener(new java.awt.event.ActionListener() {
91              public void actionPerformed(java.awt.event.ActionEvent evt) {
92                  jButton1ActionPerformed(evt);
93              }
94          });
95
96          org.openide.awt.Mnemonics.setLocalizedText(jButton3,
97          org.openide.util.NbBundle.getMessage(SearchPathCountModualTopComp
98          onent.class, "SearchPathCountModualTopComponent.jButton3.text"));
99          // NOI18N
100         jButton3.addActionListener(new java.awt.event.ActionListener() {
101             public void actionPerformed(java.awt.event.ActionEvent evt) {
102                 jButton3ActionPerformed(evt);
103             }
104         });
105
106         javax.swing.GroupLayout layout = new
107         javax.swing.GroupLayout(this);
108         this.setLayout(layout);
109         layout.setHorizontalGroup(
110             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
111             LEADING)
112             .addGroup(layout.createSequentialGroup()
113                 .addGap(61, 61, 61)
114                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
115                 Alignment.LEADING)
116                     .addComponent(jLabel1)
117                     .addComponent(jLabel2)
118                     .addComponent(jComboBox1, javax.swing.GroupLayout.DEFAULT_
119                     SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_
120                     _SIZE)
121                     .addComponent(jComboBox2, javax.swing.GroupLayout.DEFAULT_
122                     SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_
123                     _SIZE)
124                     .addComponent(jButton1)
125                     .addComponent(jButton3))
126                 .addContainerGap(61, Short.MAX_VALUE))
127         );
128     }
129
130     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
131         // TODO add your handling code here:
132     }
133
134     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
135         // TODO add your handling code here:
136     }
137 }
138
139 // End of code generation.

```

```

99             .addComponent(jLabel1)
100            .addComponent(jLabel2)
101
102            .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING, false)
103                .addGroup(layout.createSequentialGroup()
104                    .addComponent(jComboBox1,
105                        GroupLayout.PREFERRED_SIZE,
106                        GroupLayout.DEFAULT_SIZE,
107                        GroupLayout.PREFERRED_SIZE)
108
109                    .addPreferredGap(GroupLayout.Style.ComponentPlacement.RELATED,
110                        GroupLayout.DEFAULT_SIZE,
111                        Short.MAX_VALUE)
112                    .addComponent(jButton3))
113                .addGroup(layout.createSequentialGroup()
114                    .addComponent(jComboBox2,
115                        GroupLayout.PREFERRED_SIZE,
116                        GroupLayout.DEFAULT_SIZE,
117                        GroupLayout.PREFERRED_SIZE)
118                    .addGap(72, 72, 72)
119                    .addComponent(jButton1)))
120            .addContainerGap(160, Short.MAX_VALUE)
121        );
122        layout.setVerticalGroup(
123            layout.createParallelGroup(GroupLayout.Alignment.LEADING)
124                .addGroup(layout.createSequentialGroup()
125                    .addContainerGap()
126                    .addComponent(jLabel1)
127                    .addPreferredGap(GroupLayout.Style.ComponentPlacement.RELATED)
128                    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
129                        .addComponent(jComboBox1,
130                            GroupLayout.PREFERRED_SIZE,
131                            GroupLayout.DEFAULT_SIZE,
132                            GroupLayout.PREFERRED_SIZE)
133                        .addComponent(jButton3))
134                    .addGap(18, 18, 18)
135                    .addComponent(jLabel2)
136                    .addPreferredGap(GroupLayout.Style.ComponentPlacement.RELATED)
137                    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
138                        .addComponent(jComboBox2,
139                            GroupLayout.PREFERRED_SIZE,
140                            GroupLayout.DEFAULT_SIZE,
141                            GroupLayout.PREFERRED_SIZE)
142                        .addComponent(jButton1))
143                    .addContainerGap(185, Short.MAX_VALUE))
144        );
145    }
146    // </editor-fold>
147
148    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
149    {

```

```

132 GraphController graphController =
    Lookup.getDefault().lookup(GraphController.class);
133 GraphModel model = graphController.getModel();
134 Graph graph = model.getGraph();
135 PreviewController previewController =
    Lookup.getDefault().lookup(PreviewController.class);
136 PreviewModel previewModel =
    previewController.getModel(graph.getGraphModel().getWorkspace());
137 PreviewProperties previewProperties = previewModel.getProperties();
138 AttributeColumn inDegree = previewProperties.getValue("In-Degree");
139 AttributeColumn outDegree = previewProperties.getValue("Out-Degree");
140 ArrayList<String> in = new ArrayList<String>();
141 ArrayList<String> out = new ArrayList<String>();
142 for(int i = 0; i<= graph.getNodeCount(); i++ )
143 {
144     AttributeRow temp = (AttributeRow) graph.getNode(i).getAttributes();
145     if( temp.getValue(inDegree).equals(0) )
146     {
147         in.add((graph.getNode(i).getNodeData().getId()));
148     }
149     else if( temp.getValue(inDegree).equals(0))
150     {
151         out.add((graph.getNode(i).getNodeData().getId()));
152     }
153 }
154 }
155
156
157 }
158
159 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
160 {
161     GraphController graphController =
        Lookup.getDefault().lookup(GraphController.class);
162     GraphModel model = graphController.getModel();
163     Graph graph = model.getGraph();
164     SearchPathCountAnalisys.Process(jComboBox1.getSelectedItem(),
        jComboBox2.getSelectedItem(), graph );
165 }
166
167 // Variables declaration - do not modify
168 private javax.swing.JButton jButton1;
169 private javax.swing.JButton jButton3;
170 private javax.swing.JComboBox jComboBox1;
171 private javax.swing.JComboBox jComboBox2;
172 private javax.swing.JLabel jLabel1;
173 private javax.swing.JLabel jLabel2;
174 // End of variables declaration
175 @Override
176 public void componentOpened() {
177     // TODO add custom code on component opening
178 }
179
180 @Override
181 public void componentClosed() {
182     // TODO add custom code on component closing
183 }
184
185 void writeProperties(java.util.Properties p) {
186     // better to version settings since initial version as advocated
    at

```

```
187         // http://wiki.apidesign.org/wiki/PropertyFiles
188         p.setProperty("version", "1.0");
189         // TODO store your settings
190     }
191
192     void readProperties(java.util.Properties p) {
193         String version = p.getProperty("version");
194         // TODO read your settings according to their version
195     }
196 }
197
198
199 package org.Speedy.SearchPathCount;
200
201 import org.gephi.data.attributes.api.AttributeColumn;
202 import org.gephi.graph.api.Edge;
203 import org.gephi.graph.api.EdgeIterator;
204 import org.gephi.graph.api.Graph;
205 import org.gephi.graph.api.Node;
206 import org.gephi.preview.api.PreviewController;
207 import org.gephi.preview.api.PreviewModel;
208 import org.gephi.preview.api.PreviewProperties;
209 import org.openide.util.Lookup;
210
211 public class SearchPathCountAnalysis {
212     public static void Process(Node source, Node sink, Graph inGraph)
213     {
214         EdgeIterator test = inGraph.getEdges(source).iterator();
215         if( source.getId() == sink.getId())
216         {
217             return;
218         }
219
220         else
221         {
222             while(test.hasNext())
223             {
224                 Edge temp = test.next();
225                 temp.setWeight(temp.getWeight()+1);
226                 Process(temp.getTarget(), sink, inGraph);
227             }
228         }
229     }
230 }
231 }
232
```