

```
1  //*****
2  /**
3   * mit VK_ENTER ausgelöster Dialog zur Auswahl einer Leistungsposition sowie
4   * Bearbeiten u. Einfügen
5   */
6  public static void insertRowDialog() {
7      int vvid = (int) IFVorgang.vnr.getValue();
8      String lgr = IFVorgang.TextFieldLGKenn.getText();
9      String lgbez = IFVorgang.areaLG.getText();
10     String pos = IFVorgang.jTextFieldPos.getText();
11     int l = pos.length();          // 4 | 6
12     int df = (l - 4);              //0 | 2
13     String sub1 = pos.substring(0, df); //"" | 5.
14     String sub2 = pos.substring(df, l); //5.01 | 5.01
15     pos = (sub1 + sub2);           // 5.01 | 5.5.01
16     String spos = "";
17     if ("".equals(pos)) {
18         pos = "1.01";
19     } else {
20         Double dpos = Double.valueOf(sub2);
21         dpos = (((dpos * 100) + 1) / 100);
22         spos = (String.format("%.2f", dpos));
23         spos = (spos.replace(',', '.', ''));
24         spos = (sub1 + spos);
25     }
26     double anz = 0;
27     String such = IFVorgang.artSucheLike.getText();
28
29     //Erstellen der Tabelle mit Scrollbalken
30     JScrollPane jScrollPane0 = new javax.swing.JScrollPane();
31     jScrollPane0.setAlignmentY(-1.0F);
32     jScrollPane0.setPreferredSize(new java.awt.Dimension(880, 420));
```

```
33     jTableA = new javax.swing.JTable();
34
35     jTableA.setFont(new java.awt.Font("DejaVu Sans", 1, 13));
36
37     jTableA.setModel(new javax.swing.table.DefaultTableModel(
38         new Object[][]{
39         new String[]{
40             "LID", "ME", "Leistungsbeschreibung", "LE", "EP"}
41     ) {
42         Class[] types = new Class[]{
43             java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.  ↗
44             lang.String.class};
45
46         boolean[] canEdit = new boolean[]{
47             false, true, true, true, true};
48
49         @Override
50         public Class getColumnClass(int columnIndex) {
51             return types[columnIndex];
52         }
53
54         @Override
55         public boolean isCellEditable(int rowIndex, int columnIndex) {
56             return canEdit[columnIndex];
57         }
58     });
59     jTableA.setRowSelectionAllowed(true);           //ACHTUNG GEÄNDERT
60     jTableA.setColumnSelectionAllowed(false);      //ACHTUNG GEÄNDERT
61     //jTableA.setCellSelectionEnabled(true);       //ACHTUNG GEÄNDERT
62
63     jTableA.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
```

```
64     jTableA.setMaximumSize(new java.awt.Dimension(880, 3000));
65
66     jTableA.setMinimumSize(new java.awt.Dimension(440, 1000));
67
68     jTableA.setName("Leistungen"); // NOI18N
69
70     jTableA.setPreferredSize(new java.awt.Dimension(840, 3000));
71
72     jTableA.setRowHeight(52);
73     jTableA.setRowSelectionAllowed(true);
74
75     jScrollPane0.setViewportViewView(jTableA);
76     jScrollPane0.setBorder(new EmptyBorder(10, 10, 10, 10));
77
78     jTableA.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);
79     if (jTableA.getColumnModel().getColumnCount() > 0) {
80         jTableA.getColumnModel().getColumn(0).setMinWidth(40);
81         jTableA.getColumnModel().getColumn(0).setPreferredWidth(60);
82         jTableA.getColumnModel().getColumn(0).setMaxWidth(60);
83         jTableA.getColumnModel().getColumn(1).setMinWidth(40);
84         jTableA.getColumnModel().getColumn(1).setPreferredWidth(60);
85         jTableA.getColumnModel().getColumn(1).setMaxWidth(60);
86         jTableA.getColumnModel().getColumn(2).setMinWidth(400);
87         jTableA.getColumnModel().getColumn(2).setPreferredWidth(430);
88         jTableA.getColumnModel().getColumn(2).setMaxWidth(450);
89         jTableA.getColumnModel().getColumn(3).setMinWidth(40);
90         jTableA.getColumnModel().getColumn(3).setPreferredWidth(60);
91         jTableA.getColumnModel().getColumn(3).setMaxWidth(60);
92         jTableA.getColumnModel().getColumn(4).setMinWidth(80);
93         jTableA.getColumnModel().getColumn(4).setPreferredWidth(80);
94         jTableA.getColumnModel().getColumn(4).setMaxWidth(100);
95
```

```
96     }
97     //MultiLineCellRenderer für Zeilenumbruch im LP Text
98     jTableA.getColumnModel().getColumn(2).setCellRenderer(new MultiLineCellRenderer());
99
100    //CellRenderer für die Textausrichtung in den Spalten
101    DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();
102    centerRenderer.setHorizontalAlignment(SwingConstants.CENTER);
103    jTableA.getColumnModel().getColumn(1).setCellRenderer(centerRenderer);
104    jTableA.getColumnModel().getColumn(3).setCellRenderer(centerRenderer);
105    jTableA.getColumnModel().getColumn(4).setCellRenderer(centerRenderer);
106
107    DefaultTableModel model = (DefaultTableModel) jTableA.getModel();
108    model.setRowCount(0);
109    //Tabelle füllen
110    Connection conn = null;
111    String sSQL = "";
112    try {
113        Class.forName("org.postgresql.Driver");
114        conn = DriverManager.getConnection(jhpn, uname, pw);
115        if (IFVorgang.jCheckBox1.isSelected()) {
116            sSQL = "SELECT lid, me, bez, TO_CHAR(le, 'FM99990D00') as le, TO_CHAR(ep, 'FM99990D00')as ep FROM
117                leistungen WHERE vid = " + "" + such + "' ORDER BY pos ASC";
118        } else if (such.matches("\\p{Lu}{3}.*")) {
119            sSQL = "SELECT artnr, me, bez, TO_CHAR(kalk_la, 'FM99990D00') as kalk_la, TO_CHAR(kalk_gp,
120                'FM99990D00')as kalk_gp "
121                + "FROM artikel WHERE sort = " + "" + such + "";
122        } else if (such.matches("\\d*")) {
123            sSQL = "SELECT artnr, me, bez, TO_CHAR(kalk_la, 'FM99990D00') as kalk_la, TO_CHAR(kalk_gp,
124                'FM99990D00')as kalk_gp "
125                + "FROM artikel WHERE artnr = " + "" + such + "";
126        } else if (such.matches("\\p{Alpha}*\\s\\p{Alpha}*")) {
127            String[] suchA = such.split(" ");
```

```
125         String like0 = suchA[0];
126         String like1 = suchA[1];
127         //TO_CHAR(le, 'FM99990D00') AS le, TO_CHAR(lg, 'FM99990D00') AS lg
128         sSQL = "SELECT artnr, me, bez, TO_CHAR(kalk_la, 'FM99990D00') as kalk_la, TO_CHAR(kalk_gp,
129             'FM99990D00')as kalk_gp FROM artikel "
130             + "WHERE bez ILIKE " + "'" + like0 + "%'" + " AND bez ILIKE " + "'" + like1 + "%'" + " ORDER
131             BY bez ASC" + ";";
132     } else {
133         sSQL = "SELECT artnr, me, bez, TO_CHAR(kalk_la, 'FM99990D00') as kalk_la, TO_CHAR(kalk_gp,
134             'FM99990D00')as kalk_gp "
135             + "FROM artikel WHERE bez ILIKE " + "'" + such + "%'";
136     }
137     Statement st = conn.createStatement();
138     ResultSet rs = st.executeQuery(sSQL);
139     while (rs.next()) {
140         String anr = rs.getString(1); //artnr /lid
141         String a = rs.getString(2); //me /me
142         String b = rs.getString(3); //bez /bez
143         String c = rs.getString(4); //le / le
144         String d = rs.getString(5); //ep / ep
145
146         model.addRow(new Object[]{anr, a, b, c, d});
147     }
148     //Tabelle fertig
149     //Erstellen der anderen Elemente-----
150     //Button
151     JButton buttonA = new JButton("LP einfügen");
152     buttonA.setPreferredSize(new java.awt.Dimension(120, 30));
153     JButton buttonC = new JButton("Abbrechen");
154     buttonC.setPreferredSize(new java.awt.Dimension(120, 30));
155     //Label Pos u. Anz
156     Label labelPos = new Label("Position:");
```

```
154     labelPos.setFont(new java.awt.Font("DejaVu Sans", 1, 12));
155     Label labelAnz = new Label("Anzahl:");
156     labelAnz.setFont(new java.awt.Font("DejaVu Sans", 1, 12));
157     //TextFields Pos u. Anz
158     TextField posT = new TextField();
159     posT.setPreferredSize(new java.awt.Dimension(80, 30));
160     posT.setFont(new java.awt.Font("DejaVu Sans", 1, 12));
161     TextField anzT = new TextField();
162     anzT.setPreferredSize(new java.awt.Dimension(80, 30));
163     anzT.setFont(new java.awt.Font("DejaVu Sans", 1, 12));
164     //CheckBox
165     JCheckBox jCheckBox0 = new javax.swing.JCheckBox();
166     jCheckBox0.setFont(new java.awt.Font("DejaVu Sans", 1, 12));
167     jCheckBox0.setText("Vorlage speichern");
168     //Panel1 BorderLayout
169     JPanel panel1 = new JPanel();
170     panel1.setLayout(new BorderLayout());
171     panel1.add(jScrollPane0, BorderLayout.CENTER);
172     //Panel2 BoxLayout
173     JPanel panel2 = new JPanel();
174     panel2.setLayout(new BoxLayout(panel2, BoxLayout.X_AXIS));
175     panel2.setBorder(new EmptyBorder(10, 10, 10, 10));
176     //Einfügen der unteren Elemente ins Panel2
177     panel2.add(labelPos);
178     panel2.add(posT);
179     panel2.add(Box.createHorizontalGlue());
180     panel2.add(labelAnz);
181     panel2.add(anzT);
182     panel2.add(Box.createHorizontalGlue());
183     panel2.add(jCheckBox0);
184     panel2.add(Box.createHorizontalGlue());
185     panel2.add(buttonC);
```

```
186     panel2.add(Box.createHorizontalGlue());
187     panel2.add(buttonA);
188     //Einfügen von Panel2 ins SOUTH Teil von Panel1
189     panel1.add(panel2, BorderLayout.SOUTH);
190     //Erstellen des Dialogs
191     JDialog dialog = new JDialog();
192     dialog.setSize(780, 440);
193     dialog.setLocationRelativeTo(null); //Zentriert
194     dialog.add(panel1); // Einfügen aller Elemente
195     dialog.setModal(true);
196     dialog.setTitle("Leistungsposition bearbeiten u. einfügen");
197     dialog.setDefaultCloseOperation(
198         WindowConstants.DISPOSE_ON_CLOSE);
199     buttonA.addActionListener((ActionEvent e) -> {
200         dialog.dispose();
201     });
202     buttonC.addActionListener((ActionEvent e) -> {
203         posT.setText(""); // Abbrechen Button setzt Wert, der zum Abbruch führt
204         dialog.dispose();
205     });
206     posT.setText(spos);
207     dialog.setVisible(true);
208
209     // Nachdem der Dialog geschlossen wurde: -----
210     //-----
211     pos = posT.getText();
212     if ("".equals(pos)) { // ist true, wenn Abbrechen geklickt wurde
213         return;
214     }
215     String sanz = anzT.getText();
216     int znr = jTableA.getSelectedRow();
217     if (znr == -1) {
```

```
218         JOptionPane.showMessageDialog(null, "Keine Leistung markiert, Bitte Wiederholen!", "Leistung einfügen",
219         JOptionPane.WARNING_MESSAGE);
220         return;
221     }
222     if ("".equals(sanz)) {
223         sanz = JOptionPane.showInputDialog(null, "Anzahl fehlt, Bitte hier eingeben!", "Leistung einfügen",
224         JOptionPane.WARNING_MESSAGE);
225         if (sanz == null) {
226             return;
227         }
228     }
229     String anz1 = (sanz.replace(',', '.', ''));
230     anz = Double.parseDouble(anz1);
231
232     int row = jTableA.getSelectedRow();
233     Object ANR = jTableA.getModel().getValueAt(row, 0);
234     int anr = Integer.valueOf((String) ANR);
235
236     Object ME = model.getValueAt(row, 1);
237     String me = String.valueOf(ME);
238
239     Object BEZ = model.getValueAt(row, 2);
240     String lbez = String.valueOf(BEZ);
241
242     Object LE = model.getValueAt(row, 3);
243     String le = String.valueOf(LE);
244     double dle = Double.parseDouble(le.replace(',', '.', ''));
245     double dlg1 = anz * dle;
246     double dlg = round(dlg1, 2);
247
248     Object EP = model.getValueAt(row, 4);
249     String ep = String.valueOf(EP);
```

```
248     double dep = Double.parseDouble(ep.replace(',', '.'));
249     double dgp1 = anz * dep;
250     double dgp = round(dgp1, 2);
251     sSQL = "INSERT INTO leistungen (vid, lgr, pos, anz, me, bez, ep, gp, le, lg, lgbez, res ) VALUES (" + "" + ↵
vvid + "','" + "" + lgr + "','"
252         + "'" + pos + "','" + "'" + anz + "','" + "'" + me + "','" + "'" + lbez + "','" + "'" + dep + "','" + "'" ↵
+ dgp + "','"
253         + "'" + dle + "','" + "'" + dlgr + "','" + "'" + lgbez + "','" + "'" + anr + "'" + ")";
254     st = conn.createStatement();
255     st.execute(sSQL);
256
257     if (jCheckBox0.isSelected()) {
258         sSQL = "UPDATE artikel SET lgr =" + "" + lgr + "','" + "me =" + "" + me + "','" + "bez =" + "" + lbez + ↵
"',"
259         + "kalk_la =" + "" + dle + "','" + "kalk_gp =" + "" + dep + "" + "WHERE artnr =" + "" + anr + ↵
"";
260         st = conn.createStatement();
261         st.execute(sSQL);
262     }
263     conn.close();
264
265     /* Thread t1 = new Thread() { !!hat grafische Fehldarstellungen beim Neuladen verursacht
266         @Override
267         public void run() { */
268     loadLeistg();
269     /*     }
270     };
271     t1.start();*/
272
273     if (IFVorgang.jCheckBox1.isSelected()) {
274         IFVorgang.artSucheLike.setText(such);
275     } else if (IFVorgang.jCheckBox2.isSelected()) {
```

```
276         IFVorgang.artSucheLike.setText(such);
277     } else {
278         IFVorgang.artSucheLike.setText("");
279     }
280
281     } catch (ClassNotFoundException | SQLException ex) {
282         Logger.getLogger(Vorgang.class.getName()).log(Level.SEVERE, null, ex);
283     }
284 }
285
286 //*****
287
```