

CassetteStoringStockItem.java

```
1: import sale.*;
2: import data.oopl.*;
3: import data.*;
4:
5:
6: /**
7: * StoringStockItem-Implementierung fuuml;r die Videokassette im
8: * Bestand des Kunden.
9: */
10: public class CassetteStoringStockItem extends StockItemImpl
11: {
12:     // attributes ///////////////////////////////
13:     private Object date; // Ausleihdatum
14:     // constructor ///////////////////////////////
15:     /**
16:      * Der Konstruktor reicht die an ihn &uuml;bergebenen Parameter an
17:      * den Konstruktor der Klasse <CODE>StockItemImpl</CODE> weiter.
18:      * Außerdem speichert das Ausleihdatum ab.
19:      */
20:     public CassetteStoringStockItem(String name, Object date)
21:     {
22:         super (name);
23:         this.date = date;
24:     }
25:
26:
27:
28:
29:
30:     // public methods ///////////////////////////////
31:
32:
33: /**
34: * Gibt das Ausleihdatum der speziellen Videokassette zur&uuml;ck.
35: */
36: public Object getDate()
37: {
38:     return date;
39: }
40: }
```

```
1: import sale.*;
2: import sale.stdforms.*;
3:
4:
5: /**
6: * Diese Klasse implementiert den eigentlichen "Verkaufsstand". Hier
7: * kann man Videos gekauft werden.
8: */
9: public class Counter extends SalesPoint
10: {
11:
12:     // constructor ///////////////////////////////
13:
14:     /**
15:      * Konstruktor.
16:      */
17:     public Counter(String name)
18:     {
19:         super(name);
20:     }
21:
22:     // public methods ///////////////////////////////
23:
24:     /**
25:      * Gibt das Standard-FormSheet für diesen Counter zurück.
26:      */
27:     public FormSheet getDefaultFormSheet()
28:     {
29:         return new FormSheet("Menu", new DefaultCounterFormCreator(), false);
30:     }
31:
32: }
```

Customer.java

```
1: import sale.*;
2: import data.*;
3: import data.oimpl.*;
4: import data.events.*;
5: import users.*;
6:
7:
8: /**
9: * Ein registrierter Kunde mitsamt seinem Videobestand.
10: */
11: public class Customer extends User
12: {
13:
14:     // attributes ///////////////////////////////
15:
16:     private String   customerID;           // Kundennummer
17:     private StoringStock customerStoringStock; // ausgeliehene Videos
18:
19:
20:     //////////////////////////////////////////////////
21:
22: /**
23: * Legt einen neuen Kunden an.
24: */
25: public Customer(String customerID)
26: {
27:     super(customerID);
28:
29:     this.customerID = customerID;
30:     customerStoringStock = new StoringStockImpl(customerID,
31:         (CatalogImpl)Shop.getTheShop().getCatalog("Video-Catalog"));
32:     Shop.getTheShop().addStock(customerStoringStock);
33: }
34:
35:     // public methods //////////////////////////////
36:
37: /**
38: * Liefert die Kunden-ID.
39: */
40: /**
41: public String getCustomerID()
42: {
43:     return customerID;
44: }
45:
46: /**
47: * Fügt die Kassette dem Bestand des Kunden hinzu.
48: */
49: public void addVideoCassette(CassetteStoringStockItem cassette)
50: {
51:     customerStoringStock.add(cassette, null);
52: }
53:
54: /**
55: * Löscht die Kassette aus dem Bestand des Kunden.
56: */
57: public void removeVideoCassette(CassetteStoringStockItem cassette)
58: {
59:     try {
60:         customerStoringStock.remove(cassette, null);
61:     } catch(VetoException ve) {
62:     }
63: }
```

```
1: import data.*;
2: import sale.*;
3: import util.swing.*;
4:
5: import java.awt.*;
6: import java.text.*;
7:
8: import javax.swing.*;
9:
10: /**
11:  * Editor f&uuml;r die Geldbetr&auml;ge in den Tabellen.
12: */
13: /*
14: public class DMCellEditor extends DefaultCellEditor
15: {
16:     // attributes ///////////////////////////////
17:     //////////////////////////////////////////////////
18:     private String[] result;
19:     /**
20:      * Erstellt ein neues Objekt der Klasse <CODE>DMCellEditor</CODE>.
21:      */
22:     public DMCellEditor (String[] result, String init)
23:     {
24:         super(new JTextField(result, init));
25:         this.result = result;
26:     }
27:
28:     /**
29:      * public methods ///////////////////////////////
30:      */
31: }
32:
33: /**
34:  * Gibt die zu editieren verwendete Komponente korrekt initialisiert
35:  */
36: /**
37:  * ...initialisieren
38:  * zur&uml;ck.
39:  */
40: public Component getTableCellEditorComponent(JTable jTable,
41:     Object value, boolean isSelected, int row, int column)
42: {
43:     // die Komponente holen,
44:     Component component =
45:         super.getTableCellEditorComponent(jTable, value, isSelected,
46:             row, column);
47:
48:     /**
49:      * ...initialisieren
50:      * Shop.getTheShop().getCatalog("DM").toString((Number)value));
51:      */
52:     // ...und zurueckgeben
53:     return component;
54: }
55:
56: /**
57:  * Versucht, den eingegebenen Text als Geldbetrag zu interpretieren.
58: */
59: public Object getCellEditorValue()
60: {
61:     // versuchen, den Text zu parsen und zurueckzugeben
62:     try {
63:         return ((Currency)Shop.getTheShop().getCatalog("DM")).parse(result[0]);
64:     }
65:     catch (ParseException pexc) {
66:         // Text konnte nicht verarbeitet werden, 0 zurueckgeben
67:         return new Integer(0);
68:     }
69: }
70:
71: /**
72:  * Versucht, das Editieren zu beenden und gibt eine Bewertung des
73:  * Erfolges zur&uml;ck.
74: */
75: public boolean stopCellEditing()
76: {
77:     // versuchen den Text zu parseen
78:     try {
79:         ((Currency)Shop.getTheShop().getCatalog("DM")).parse(result[0]);
80:     }
81:     catch (ParseException pexc) {
82:         // Text konnte nicht verarbeitet werden, editieren kann nicht
83:         // beendet werden
84:         return false;
85:     }
86:
87:     // Edittieren kann nach Mauszeabe der Methode der Oberklasse beendet werden
88:     return super.stopCellEditing();
89: }
90: }
```

DefaultCounterFormCreator.java

```

1: import sale.*;
2: import data.oobjimpl.*;
3: import data.stdforms.*;
4:
5: /**
6: * Erzeugt eine Tabelle, sowie notwendige Buttons f&uuml;r die
7: * Angebotsanzeige.
8:
9: */
10: public class DefaultCounterFormCreator extends FormSheetContentCreator
11: {
12:     // constructor ///////////////////////////////
13:
14:     /**
15:      * Konstruktor.
16:      */
17:     public DefaultCounterFormCreator()
18:     {
19:         super();
20:     }
21:
22:
23:     // public methods ///////////////////////////////
24:
25:
26:
27: /**
28: * Erzeugt den <CODE>FormSheetContent</CODE>, also Tabelle(n) und
29: * Buttons. Als Parameter wird das zu bearbeitende <CODE>FormSheet</CODE>
30: * &uuml;bergeben.
31: */
32: public void createFormSheetContent(FormSheet map)
33: {
34:     // heraussuchen des Videobestandes zur Darstellung in der Tabelle
35:     CountingStockImpl cs =
36:         (CountingStockImpl) Shop.getTheShop().getStock("Video-Countingstock");
37:
38:
39:     // erstellen des FormSheets mit der Tabelle des Bestandes als Inhalt
40:     FormSheet fs = SingleTableFormSheet.create("Available Videocassettes",
41:         null, new OfferED(false));
42:
43:     // die Tabelle holen und dem ueberggebenen FormSheet als Komponente zuweisen
44:     map.setComponent(fs.getComponent());
45:
46:     // alle Buttons entfernen
47:     map.removeAllButtons();
48:
49:     map.addButton("rent", 1,
50:         new Sale.Action()
51:             {
52:                 public void doAction(SaleProcess p, SalesPoint s)
53:                 {
54:                     // Code zum Ausfuehren des Verleihvorgangs
55:                     s.runProcess(new RentProcess());
56:                 }
57:             }
58:     );
59:
60:     map.addButton("give back", 2,
61:         new Sale.Action()
62:             {
63:                 public void doAction(SaleProcess p, SalesPoint s)
64:                     {
65:                         // Code zum Ausfuehren des Rueckgabevorgangs
66:                         s.runProcess(new GiveBackProcess());
67:                     }
68:             }
69:     );
70: }
71: }
```

1 DefaultOfficeFormCreator.java

```

1: import data.*;
2: import sale.*;
3: import javax.swing.*;
4: import java.awt.*;
5: /**
6:  * FormSheetContentCreator, der den Inhalt der Standard-FormSheets von
7:  * Office erzeugt
8: */
9: public class DefaultOfficeFormCreator extends FormSheetContentCreator
10: {
11: 
12:     // attributes
13:     // constructor
14:     private Office officeOwner;
15: 
16: 
17: 
18:     /**
19:      * Konstruktor: Erzeugt ein neues Objekt vom Typ
20:      * <CODE>DefaultOfficeFormCreator</CODE>.
21:      */
22: 
23:     public DefaultOfficeFormCreator(Office officeOwner)
24:     {
25:         super();
26:         this.officeOwner = officeOwner;
27:     }
28: 
29: 
30:     /**
31:      * Erzeugt den Inhalt des &uuml;bergebenen FormSheets.
32:      */
33: 
34:     /**
35:      * public void createFormSheetContent (FormSheet fsoToCreate)
36:      {
37:         // JPanel erzeugen und vertikales BoxLayout setzen
38:         JPanel jpFSComponent = new JPanel();
39: 
40:         jpFSComponent.setLayout (new BoxLayout (jpFSComponent, BoxLayout.Y_AXIS));
41: 
42:         // Label mit der aktuellen Zeit ins JPanel einfuegen
43:         jpFSComponent.add(new JLabel("Turn : " + 
44:             Shop.getTheShop().getTimer().getTime()));
45: 
46:         // Geldbestand holen
47:         Stock coinsSlot = Shop.getTheShop().getStock("coin slot");
48: 
49:         // DataBasket des Managers holen
50:         DataBasket db = officeOwner.getBasket();
51: 
52:         // verwendete Waehrung holen
53:         Currency currency = (Currency)Shop.getTheShop().getCatalog("DM");
54: 
55:         // Geldbestand aufsummieren und Label mit der Summe ins JPanel einfuegen
56:         IntegerValue money = (IntegerValue)coinsSlot.sumStock(db,
57:             new CatalogItemValue(), new IntegerValue(0));
58: 
59:         jpFSComponent.add (new JLabel("Money : " + currency.toString(money)));
60: 
61:         // Komponente ins FormSheet einfuegen
62:         fsoToCreate.setComponent (jpFSComponent);
63:     }

```

EditableVideoStockTED.java

```

1: import sale.*;
2: import data.*;
3: import data.oimpl.*;
4: import data.swing.*;
5: import data.events.*;
6: import util.swing.*;
7: import java.util.*;
8: import javax.swing.*;
9: import javax.swing.table.*;
10: import javax.swing.*;
11: import javax.swing.table.*;
12: import javax.swing.*;
13: import javax.swing.*;
14: /**
15: * Ein TableEntryDescriptor zur ausf&uuml;hrlichen Darstellung des
16: * Videobestandes.
17: */
18: public class EditableVideoStockTED extends AbstractTableEntryDescriptor
19: {
20:     ///////////////////////////////////////////////////////////////////
21:     // attributes ///////////////////////////////////////////////////////////////////
22:     private CountingStockImpl videoCountingStock; // darzustellender Bestand
23:     private DatabaseBasket db; // verwendet der Datenkorb
24:     // Variablen zur Formatierung der Darstellung
25:     private TableCellRenderer tcrName; // verantwortet tcrName;
26:     private TableCellRenderer tcrCount; // verantwortet tcrCount;
27:     private TableCellRenderer tcrMoney; // verantwortet tcrMoney;
28:     private TableCellRenderer tcrBasket; // verantwortet tcrBasket;
29:     private TableCellRenderer tcrRecord; // verantwortet tcrRecord;
30:     ///////////////////////////////////////////////////////////////////
31:     // constructor ///////////////////////////////////////////////////////////////////
32:     /**
33:      * Erzeugt ein neues Objekt der Klasse <CODE>EditableVideoStockTED</CODE>.
34:      * Es wird der darzustellende <CODE>CountingStock</CODE> und der zu
35:      * verwendende <CODE>DatabaseBasket</CODE> &uuml;bergeben.
36:      */
37:     /**
38:      * public EditableVideoStockTED(CountingStockImpl videoCountingStock,
39:      * DatabaseBasket db)
40:      {
41:          super();
42:          this.videoCountingStock = videoCountingStock;
43:          this.db = db;
44:      }
45:      // Initialisierung der Variablen zu Darstellung der
46:      // einzelnen Formate
47:      tcrMoney = new CurrencyRenderer((Currency)Shop.getTheShop().getCatalog("DM"));
48:      tcrName = new DefaultTableCellRenderer();
49:      tcrCount = new DefaultTableCellRenderer();
50:      tcrrCount = new DefaultTableCellRenderer();
51:      // Ausrichtung f&uuml;r die Darstellung der Anzahl
52:      ((DefaultTableCellRenderer)tcrCount).setHorizontalAlignment(SwingConstants.CENTER);
53:      /**
54:      * Liefert die Anzahl der anzuseigenden Spalten.
55:      */
56:      // Ausrichtung f&uuml;r die Darstellung der Anzahl
57:      ///////////////////////////////////////////////////////////////////
58:      /**
59:      * Liefert die Anzahl der ausgeliehenen Videos ermittelte
60:      */
61:      /**
62:      * public int getColumnCount()
63:      {
64:          return 5;
65:      }
66:      /**
67:      * Liefert die Spaltennamen.
68:      */
69:      /**
70:      * public String getColumnName(int index)
71:      {
72:          return new String[] {"Name", "Buy", "Sell", "Pieces in Stock", "Rent Pieces"}[index];
73:      }
74:      /**
75:      * Legt die Darstellung f&uuml;r die einzelnen Spalten fest.
76:      */
77:      /**
78:          /**
79:          * Legt die Darstellung f&uuml;r die einzelnen Spalten fest.
80:          */
81:          /**
82:          * public TableCellRenderer getCellRenderer(int index)
83:          {
84:              switch (index) {
85:                  case 0: return tcrName;
86:                  case 3: return tcrCount;
87:                  case 4: return tcrMoney;
88:                  default: return tcrName;
89:              }
90:          }
91:          /**
92:          * Wird aufgerufen, wenn <CODE>getCellRenderer()</CODE> oder
93:          * <CODE>getCellEditor() null</CODE> zur&uuml;ckgegeben.
94:          */
95:          /**
96:          * public Class getColumnClass(int index)
97:          {
98:              return null;
99:          }
100:         /**
101:         * Liefer den Zelleninhalt f&uuml;r das &uuml;bergebene Objekt
102:         * und die angegebene Spalte.
103:         */
104:         /**
105:         * public Object getValueAt(Object record, int index)
106:         {
107:             // Videokassette bestimmen
108:             VideoCassette videoCassette =
109:                 (VideoCassette)(data.swing.CountingStockTableModel.Record)
110:                     .record.getDescriptor();
111:             // Anzahl im Automaten ermitteln
112:             int count =
113:                 (data.swing.CountingStockTableModel.Record).getCount();
114:             /**
115:             * Spalteneintrag zurueckgeben
116:             */
117:             switch (index) {
118:                 case 0: return videoCassette.getName();
119:                 /**
120:                 * return ( (QuoteValue)videoCassette.getValue() ).getOffer();
121:                 */
122:                 /**
123:                 * return ( (QuoteValue)videoCassette.getValue() ).getBid();
124:                 */
125:                 /**
126:                 * return new Integer(quoteValue.count);
127:                 */
128:                 /**
129:                 * Anzahl der ausgeliehenen Videos ermittelte
130:                 */
131:             }
132:         }
133:     }
134: }

```

EditableVideoStockTED.java

```

127:     int rented = 0;
128:     try {
129:         Iterator i = videoMachine.getAllCustomer().iterator();
130:         while (i.hasNext()) {
131:             rented = rented +
132:                 ((Customer)i.next()).getStoringStock().countItems(
133:                     videoCassette.getName(), null);
134:         }
135:     } catch (NullPointerException npe) {
136:     } catch (NullPointerException ve) {
137:     }
138:     // und zurueckgeben
139:     return new IntegerValue(rented);
140: }
141: default:
142:     return null;
143: }
144: }

145: /**
146: * Definiert die Spalten, die editierbar sein sollen.
147: */
148: public boolean isElementEditable(Object record, int index)
149: {
150:     return ((index == 1) || (index == 2));
151: }
152: }

153:
154:
155: /**
156: * Liefert f&uuml;r die zu editierenden Spalten den
157: * <CODE>TableCellEditor</CODE> zur&uuml;ck. In diesem Fall
158: * eine Instanz der Klasse <CODE>DMCellEditor</CODE>.
159: */
160: public TableCellEditor getCellEditor(int index)
161: {
162:     // Preise zu editieren?
163:     if (index == 1 || index == 2)
164:         return new DMCellEditor(new String[1], "");
165:     // restliche Felder
166:     else
167:         return super.getCellEditor(index);
168: }
169: }

170:
171: /**
172: * &Uuml;bertr&uuml;gt den eingegebenen Wert in das jeweilige
173: * Objekt.
174: */
175: public void setValueAt(Object record, int index, Object value)
176: {
177:     VideoCassette videoCassette =
178:         (VideoCassette)((data.swing.CountingStockTableModel.Record)
179:             record).getDescriptor();
180:     try {
181:         VideoCassette videoCassetteToEdit =
182:             (VideoCassette)videoCassette.getCatalog(null).get(
183:                 videoCassette.getName(), db, true);
184:         quoteValue.quoteValue = (QuoteValue)videoCassetteToEdit.getValue();
185:         quoteValue.quoteValue.setValue(new QuoteValue.quoteValue());
186:     }
187:     if (index == 1)
188:         videoCassetteToEdit.setValue(new QuoteValue.quoteValue());
189: }

```

GiveBackProcess.java

```

1: import sale.*;
2: import sale.stdforms.*;
3: import data.*;
4: import data.oobjimpl.*;
5: import data.stdforms.*;
6: import data.events.*;
7: import users.*;
8: import log.*;
9:
10: import java.util.*;
11: import java.text.*;
12: import java.lang.*;
13: import java.io.*;
14:
15: import javax.swing.JTextField;
16: import javax.swing.JPanel;
17: import javax.swing.JOptionPane;
18: import javax.swing.BoxLayout;
19: import javax.swing.JLabel;
20:
21:
22: /**
23:  * 
24:  * R&uuml;ckgabeProzess, der von einem registrierten Kunden
25:  * am Automaten durchgef&uuml;hrt werden kann.
26:  */
27: public class GiveBackProcess extends SaleProcess
28: {
29:   // attributes
30:   /*
31:    * Gates
32:    */
33:   protected UIGate capabilityGate;
34:   protected UIGate selectionGate;
35:   protected UIGate giveRestGate;
36:
37:   // Transitions
38:   protected Transition toSelectionTransition;
39:   protected Transition toGetMoneyTransition;
40:
41:   // verwendete Waehrung
42:   protected Currency myCurrency;
43:
44:   // rueckzuhender Betrag
45:   private int topPayBackValue;
46:
47:   // gezahlter Betrag (Verkaufspreis des Videos)
48:   private IntegerValue paidValue;
49:
50:   // Kunde, der zurueckgeben moechte
51:   private Customer customer;
52:
53:   // Verleihpreis eines Videos pro Tag
54:   private int rentPrice = 300;
55:
56:   /**
57:    * constructor
58:    */
59:   /*
60:    * Erzeugt ein neues Objekt der Klasse GiveBackProcess.
61:    */
62:   public GiveBackProcess()
63:   {
64:     super ("GiveBackProcess");
65:   }
66:   //////////////////////////////////////////////////////////////////
67:   /**
68:    * 
69:    * * Baut die Oberfl&uuml;che f&uuml;r den R&uuml;ckgabevorgang auf.
70:    * 
71:    * * 
72:    * protected void setupMachine()
73:    * {
74:    *   // verwendete Waehrung ermittelten
75:    *   myCurrency = (Currency)Shop.getTheShop().getCatalog ("DM");
76:    *
77:    *   // Videokatalog
78:    *   final CatalogImpl videoCatalog =
79:    *     (CatalogImpl)Shop.getTheShop().getCatalog ("Video-Catalog");
80:
81:   // Gate zur Kundennummer-Abfrage anleeren
82:   capabilityGate = new UIGate(null, null);
83:
84:   // Formsheet fuer Eingabe der Kundennummer
85:   final TextInputForm tif = new TextInputForm ("Customer-ID",
86:     "Customer-ID",
87:     "");
88:   // Auswahl-Gate anlegen
89:   selectionGate = new UIGate(null, null);
90:
91:   // zu verwendenden Datenkorb ermitteln
92:   final DataBasket db = getBasket();
93:
94:   // Transition zum Selection Gate
95:   toSelectionTransition = new Transition()
96:   {
97:     public Gate perform (SaleProcess power, User usr)
98:     {
99:       // zu verwendenden Bestand ermitteln
100:      StoringStock ss = customer.getStoringStock();
101:
102:      // am Gate darzustellendes FormSheet
103:      TwoTableFormSheet ttfs = TwoTableFormSheet.create(
104:        "Give Back", // Titel des FormSheets
105:        ss, // Quell-StoringStock
106:        db, // Ziel-DataBasket
107:        selectionGate // Gate, an dem das FormSheet darzustellen ist
108:      );
109:
110:      // FormSheetContentCreator am Auswahl-Gate anmelden
111:      ttfs.addContentCreator (new FormSheetContentCreator ()
112:      {
113:        protected void createFormSheetContent (FormSheet fs)
114:        {
115:          // alle vorhandenen Buttons entfernen
116:          fs.removeAllButtons ();
117:
118:          // neuen "OK"-Button einbauen
119:          fs.addButton ("Ok", 100, new sale.Action ()
120:          {
121:            public void doAction (SaleProcess p, SalesPoint sp)
122:            {
123:              // Transition zum Berahlen als naechste Transition setzen
124:              selectionGate.setNextTransition (toGetMoneyTransition);
125:
126:            }
127:          });
128:        }
129:      });
130:    }
131:  }
132:
133:  /**
134:   * Transition zum Berahlen als naechste Transition setzen
135:   */
136:  protected void setNextTransition (Transition next)
137:  {
138:    selectionGate.setNextTransition (next);
139:  }
140:
141:  /**
142:   * 
143:   */
144:  protected void doAction (SaleProcess p, SalesPoint sp)
145:  {
146:    // 
147:  }
148:
149:  /**
150:   * 
151:   */
152:  protected void removeButtons (FormSheet fs)
153:  {
154:    fs.removeAllButtons ();
155:  }
156:
157:  /**
158:   * 
159:   */
160:  protected void addButtons (FormSheet fs)
161:  {
162:    fs.addButton ("Ok", 100, new sale.Action ())
163:    {
164:      public void doAction (SaleProcess p, SalesPoint sp)
165:      {
166:        // 
167:      }
168:    });
169:  }
170:
171:  /**
172:   * 
173:   */
174:  protected void createFormSheetContent (FormSheet fs)
175:  {
176:    fs.removeAllButtons ();
177:  }
178:
179:  /**
180:   * 
181:   */
182:  protected void addContentCreator (FormSheetContentCreator creator)
183:  {
184:    fs.addContentCreator (creator);
185:  }
186:
187:  /**
188:   * 
189:   */
190:  protected void setNextTransition (Transition next)
191:  {
192:    selectionGate.setNextTransition (next);
193:  }
194:
195:  /**
196:   * 
197:   */
198:  protected void doAction (SaleProcess p, SalesPoint sp)
199:  {
200:    // 
201:  }
202:
203:  /**
204:   * 
205:   */
206:  protected void removeAllButtons (FormSheet fs)
207:  {
208:    fs.removeAllButtons ();
209:  }
210:
211:  /**
212:   * 
213:   */
214:  protected void addButton (FormSheet fs)
215:  {
216:    fs.addButton ("Ok", 100, new sale.Action ())
217:    {
218:      public void doAction (SaleProcess p, SalesPoint sp)
219:      {
220:        // 
221:      }
222:    });
223:  }
224:
225:  /**
226:   * 
227:   */
228:  protected void setNextTransition (Transition next)
229:  {
230:    selectionGate.setNextTransition (next);
231:  }
232:
233:  /**
234:   * 
235:   */
236:  protected void doAction (SaleProcess p, SalesPoint sp)
237:  {
238:    // 
239:  }
240:
241:  /**
242:   * 
243:   */
244:  protected void removeButtons (FormSheet fs)
245:  {
246:    fs.removeAllButtons ();
247:  }
248:
249:  /**
250:   * 
251:   */
252:  protected void addButtons (FormSheet fs)
253:  {
254:    fs.addButton ("Ok", 100, new sale.Action ())
255:    {
256:      public void doAction (SaleProcess p, SalesPoint sp)
257:      {
258:        // 
259:      }
260:    });
261:  }
262:
263:  /**
264:   * 
265:   */
266:  protected void createFormSheetContent (FormSheet fs)
267:  {
268:    fs.removeAllButtons ();
269:  }
270:
271:  /**
272:   * 
273:   */
274:  protected void addContentCreator (FormSheetContentCreator creator)
275:  {
276:    fs.addContentCreator (creator);
277:  }
278:
279:  /**
280:   * 
281:   */
282:  protected void setNextTransition (Transition next)
283:  {
284:    selectionGate.setNextTransition (next);
285:  }
286:
287:  /**
288:   * 
289:   */
290:  protected void doAction (SaleProcess p, SalesPoint sp)
291:  {
292:    // 
293:  }
294:
295:  /**
296:   * 
297:   */
298:  protected void removeAllButtons (FormSheet fs)
299:  {
300:    fs.removeAllButtons ();
301:  }
302:
303:  /**
304:   * 
305:   */
306:  protected void addButton (FormSheet fs)
307:  {
308:    fs.addButton ("Ok", 100, new sale.Action ())
309:    {
310:      public void doAction (SaleProcess p, SalesPoint sp)
311:      {
312:        // 
313:      }
314:    });
315:  }
316:
317:  /**
318:   * 
319:   */
320:  protected void setNextTransition (Transition next)
321:  {
322:    selectionGate.setNextTransition (next);
323:  }
324:
325:  /**
326:   * 
327:   */
328:  protected void doAction (SaleProcess p, SalesPoint sp)
329:  {
330:    // 
331:  }
332:
333:  /**
334:   * 
335:   */
336:  protected void removeButtons (FormSheet fs)
337:  {
338:    fs.removeAllButtons ();
339:  }
340:
341:  /**
342:   * 
343:   */
344:  protected void addButtons (FormSheet fs)
345:  {
346:    fs.addButton ("Ok", 100, new sale.Action ())
347:    {
348:      public void doAction (SaleProcess p, SalesPoint sp)
349:      {
350:        // 
351:      }
352:    });
353:  }
354:
355:  /**
356:   * 
357:   */
358:  protected void createFormSheetContent (FormSheet fs)
359:  {
360:    fs.removeAllButtons ();
361:  }
362:
363:  /**
364:   * 
365:   */
366:  protected void addContentCreator (FormSheetContentCreator creator)
367:  {
368:    fs.addContentCreator (creator);
369:  }
370:
371:  /**
372:   * 
373:   */
374:  protected void setNextTransition (Transition next)
375:  {
376:    selectionGate.setNextTransition (next);
377:  }
378:
379:  /**
380:   * 
381:   */
382:  protected void doAction (SaleProcess p, SalesPoint sp)
383:  {
384:    // 
385:  }
386:
387:  /**
388:   * 
389:   */
390:  protected void removeAllButtons (FormSheet fs)
391:  {
392:    fs.removeAllButtons ();
393:  }
394:
395:  /**
396:   * 
397:   */
398:  protected void addButton (FormSheet fs)
399:  {
400:    fs.addButton ("Ok", 100, new sale.Action ())
401:    {
402:      public void doAction (SaleProcess p, SalesPoint sp)
403:      {
404:        // 
405:      }
406:    });
407:  }
408:
409:  /**
410:   * 
411:   */
412:  protected void setNextTransition (Transition next)
413:  {
414:    selectionGate.setNextTransition (next);
415:  }
416:
417:  /**
418:   * 
419:   */
420:  protected void doAction (SaleProcess p, SalesPoint sp)
421:  {
422:    // 
423:  }
424:
425:  /**
426:   * 
427:   */
428:  protected void removeButtons (FormSheet fs)
429:  {
430:    fs.removeAllButtons ();
431:  }
432:
433:  /**
434:   * 
435:   */
436:  protected void addButtons (FormSheet fs)
437:  {
438:    fs.addButton ("Ok", 100, new sale.Action ())
439:    {
440:      public void doAction (SaleProcess p, SalesPoint sp)
441:      {
442:        // 
443:      }
444:    });
445:  }
446:
447:  /**
448:   * 
449:   */
450:  protected void createFormSheetContent (FormSheet fs)
451:  {
452:    fs.removeAllButtons ();
453:  }
454:
455:  /**
456:   * 
457:   */
458:  protected void addContentCreator (FormSheetContentCreator creator)
459:  {
460:    fs.addContentCreator (creator);
461:  }
462:
463:  /**
464:   * 
465:   */
466:  protected void setNextTransition (Transition next)
467:  {
468:    selectionGate.setNextTransition (next);
469:  }
470:
471:  /**
472:   * 
473:   */
474:  protected void doAction (SaleProcess p, SalesPoint sp)
475:  {
476:    // 
477:  }
478:
479:  /**
480:   * 
481:   */
482:  protected void removeAllButtons (FormSheet fs)
483:  {
484:    fs.removeAllButtons ();
485:  }
486:
487:  /**
488:   * 
489:   */
490:  protected void addButton (FormSheet fs)
491:  {
492:    fs.addButton ("Ok", 100, new sale.Action ())
493:    {
494:      public void doAction (SaleProcess p, SalesPoint sp)
495:      {
496:        // 
497:      }
498:    });
499:  }
500:
501:  /**
502:   * 
503:   */
504:  protected void setNextTransition (Transition next)
505:  {
506:    selectionGate.setNextTransition (next);
507:  }
508:
509:  /**
510:   * 
511:   */
512:  protected void doAction (SaleProcess p, SalesPoint sp)
513:  {
514:    // 
515:  }
516:
517:  /**
518:   * 
519:   */
520:  protected void removeButtons (FormSheet fs)
521:  {
522:    fs.removeAllButtons ();
523:  }
524:
525:  /**
526:   * 
527:   */
528:  protected void addButtons (FormSheet fs)
529:  {
530:    fs.addButton ("Ok", 100, new sale.Action ())
531:    {
532:      public void doAction (SaleProcess p, SalesPoint sp)
533:      {
534:        // 
535:      }
536:    });
537:  }
538:
539:  /**
540:   * 
541:   */
542:  protected void createFormSheetContent (FormSheet fs)
543:  {
544:    fs.removeAllButtons ();
545:  }
546:
547:  /**
548:   * 
549:   */
550:  protected void addContentCreator (FormSheetContentCreator creator)
551:  {
552:    fs.addContentCreator (creator);
553:  }
554:
555:  /**
556:   * 
557:   */
558:  protected void setNextTransition (Transition next)
559:  {
560:    selectionGate.setNextTransition (next);
561:  }
562:
563:  /**
564:   * 
565:   */
566:  protected void doAction (SaleProcess p, SalesPoint sp)
567:  {
568:    // 
569:  }
570:
571:  /**
572:   * 
573:   */
574:  protected void removeAllButtons (FormSheet fs)
575:  {
576:    fs.removeAllButtons ();
577:  }
578:
579:  /**
580:   * 
581:   */
582:  protected void addButton (FormSheet fs)
583:  {
584:    fs.addButton ("Ok", 100, new sale.Action ())
585:    {
586:      public void doAction (SaleProcess p, SalesPoint sp)
587:      {
588:        // 
589:      }
590:    });
591:  }
592:
593:  /**
594:   * 
595:   */
596:  protected void setNextTransition (Transition next)
597:  {
598:    selectionGate.setNextTransition (next);
599:  }
600:
601:  /**
602:   * 
603:   */
604:  protected void doAction (SaleProcess p, SalesPoint sp)
605:  {
606:    // 
607:  }
608:
609:  /**
610:   * 
611:   */
612:  protected void removeButtons (FormSheet fs)
613:  {
614:    fs.removeAllButtons ();
615:  }
616:
617:  /**
618:   * 
619:   */
619:  protected void addButtons (FormSheet fs)
620:  {
621:    fs.addButton ("Ok", 100, new sale.Action ())
622:    {
623:      public void doAction (SaleProcess p, SalesPoint sp)
624:      {
625:        // 
626:      }
627:    });
628:  }
629:
630:  /**
631:   * 
632:   */
633:  protected void createFormSheetContent (FormSheet fs)
634:  {
635:    fs.removeAllButtons ();
636:  }
637:
638:  /**
639:   * 
640:   */
641:  protected void addContentCreator (FormSheetContentCreator creator)
642:  {
643:    fs.addContentCreator (creator);
644:  }
645:
646:  /**
647:   * 
648:   */
649:  protected void setNextTransition (Transition next)
650:  {
651:    selectionGate.setNextTransition (next);
652:  }
653:
654:  /**
655:   * 
656:   */
657:  protected void doAction (SaleProcess p, SalesPoint sp)
658:  {
659:    // 
660:  }
661:
662:  /**
663:   * 
664:   */
665:  protected void removeAllButtons (FormSheet fs)
666:  {
667:    fs.removeAllButtons ();
668:  }
669:
670:  /**
671:   * 
672:   */
673:  protected void addButton (FormSheet fs)
674:  {
675:    fs.addButton ("Ok", 100, new sale.Action ())
676:    {
677:      public void doAction (SaleProcess p, SalesPoint sp)
678:      {
679:        // 
680:      }
681:    });
682:  }
683:
684:  /**
685:   * 
686:   */
687:  protected void setNextTransition (Transition next)
688:  {
689:    selectionGate.setNextTransition (next);
690:  }
691:
692:  /**
693:   * 
694:   */
695:  protected void doAction (SaleProcess p, SalesPoint sp)
696:  {
697:    // 
698:  }
699:
700:  /**
701:   * 
702:   */
703:  protected void removeButtons (FormSheet fs)
704:  {
705:    fs.removeAllButtons ();
706:  }
707:
708:  /**
709:   * 
710:   */
711:  protected void addButtons (FormSheet fs)
712:  {
713:    fs.addButton ("Ok", 100, new sale.Action ())
714:    {
715:      public void doAction (SaleProcess p, SalesPoint sp)
716:      {
717:        // 
718:      }
719:    });
720:  }
721:
722:  /**
723:   * 
724:   */
725:  protected void createFormSheetContent (FormSheet fs)
726:  {
727:    fs.removeAllButtons ();
728:  }
729:
730:  /**
731:   * 
732:   */
733:  protected void addContentCreator (FormSheetContentCreator creator)
734:  {
735:    fs.addContentCreator (creator);
736:  }
737:
738:  /**
739:   * 
740:   */
741:  protected void setNextTransition (Transition next)
742:  {
743:    selectionGate.setNextTransition (next);
744:  }
745:
746:  /**
747:   * 
748:   */
749:  protected void doAction (SaleProcess p, SalesPoint sp)
750:  {
751:    // 
752:  }
753:
754:  /**
755:   * 
756:   */
757:  protected void removeAllButtons (FormSheet fs)
758:  {
759:    fs.removeAllButtons ();
760:  }
761:
762:  /**
763:   * 
764:   */
765:  protected void addButton (FormSheet fs)
766:  {
767:    fs.addButton ("Ok", 100, new sale.Action ())
768:    {
769:      public void doAction (SaleProcess p, SalesPoint sp)
770:      {
771:        // 
772:      }
773:    });
774:  }
775:
776:  /**
777:   * 
778:   */
779:  protected void setNextTransition (Transition next)
780:  {
781:    selectionGate.setNextTransition (next);
782:  }
783:
784:  /**
785:   * 
786:   */
787:  protected void doAction (SaleProcess p, SalesPoint sp)
788:  {
789:    // 
790:  }
791:
792:  /**
793:   * 
794:   */
795:  protected void removeButtons (FormSheet fs)
796:  {
797:    fs.removeAllButtons ();
798:  }
799:
800:  /**
801:   * 
802:   */
803:  protected void addButtons (FormSheet fs)
804:  {
805:    fs.addButton ("Ok", 100, new sale.Action ())
806:    {
807:      public void doAction (SaleProcess p, SalesPoint sp)
808:      {
809:        // 
810:      }
811:    });
812:  }
813:
814:  /**
815:   * 
816:   */
817:  protected void createFormSheetContent (FormSheet fs)
818:  {
819:    fs.removeAllButtons ();
820:  }
821:
822:  /**
823:   * 
824:   */
825:  protected void addContentCreator (FormSheetContentCreator creator)
826:  {
827:    fs.addContentCreator (creator);
828:  }
829:
830:  /**
831:   * 
832:   */
833:  protected void setNextTransition (Transition next)
834:  {
835:    selectionGate.setNextTransition (next);
836:  }
837:
838:  /**
839:   * 
840:   */
841:  protected void doAction (SaleProcess p, SalesPoint sp)
842:  {
843:    // 
844:  }
845:
846:  /**
847:   * 
848:   */
849:  protected void removeAllButtons (FormSheet fs)
850:  {
851:    fs.removeAllButtons ();
852:  }
853:
854:  /**
855:   * 
856:   */
857:  protected void addButton (FormSheet fs)
858:  {
859:    fs.addButton ("Ok", 100, new sale.Action ())
860:    {
861:      public void doAction (SaleProcess p, SalesPoint sp)
862:      {
863:        // 
864:      }
865:    });
866:  }
867:
868:  /**
869:   * 
870:   */
871:  protected void setNextTransition (Transition next)
872:  {
873:    selectionGate.setNextTransition (next);
874:  }
875:
876:  /**
877:   * 
878:   */
879:  protected void doAction (SaleProcess p, SalesPoint sp)
880:  {
881:    // 
882:  }
883:
884:  /**
885:   * 
886:   */
887:  protected void removeButtons (FormSheet fs)
888:  {
889:    fs.removeAllButtons ();
890:  }
891:
892:  /**
893:   * 
894:   */
895:  protected void addButtons (FormSheet fs)
896:  {
897:    fs.addButton ("Ok", 100, new sale.Action ())
898:    {
899:      public void doAction (SaleProcess p, SalesPoint sp)
900:      {
901:        // 
902:      }
903:    });
904:  }
905:
906:  /**
907:   * 
908:   */
909:  protected void createFormSheetContent (FormSheet fs)
910:  {
911:    fs.removeAllButtons ();
912:  }
913:
914:  /**
915:   * 
916:   */
917:  protected void addContentCreator (FormSheetContentCreator creator)
918:  {
919:    fs.addContentCreator (creator);
920:  }
921:
922:  /**
923:   * 
924:   */
925:  protected void setNextTransition (Transition next)
926:  {
927:    selectionGate.setNextTransition (next);
928:  }
929:
930:  /**
931:   * 
932:   */
933:  protected void doAction (SaleProcess p, SalesPoint sp)
934:  {
935:    // 
936:  }
937:
938:  /**
939:   * 
940:   */
941:  protected void removeAllButtons (FormSheet fs)
942:  {
943:    fs.removeAllButtons ();
944:  }
945:
946:  /**
947:   * 
948:   */
949:  protected void addButton (FormSheet fs)
950:  {
951:    fs.addButton ("Ok", 100, new sale.Action ())
952:    {
953:      public void doAction (SaleProcess p, SalesPoint sp)
954:      {
955:        // 
956:      }
957:    });
958:  }
959:
960:  /**
961:   * 
962:   */
963:  protected void setNextTransition (Transition next)
964:  {
965:    selectionGate.setNextTransition (next);
966:  }
967:
968:  /**
969:   * 
970:   */
971:  protected void doAction (SaleProcess p, SalesPoint sp)
972:  {
973:    // 
974:  }
975:
976:  /**
977:   * 
978:   */
979:  protected void removeButtons (FormSheet fs)
980:  {
981:    fs.removeAllButtons ();
982:  }
983:
984:  /**
985:   * 
986:   */
987:  protected void addButtons (FormSheet fs)
988:  {
989:    fs.addButton ("Ok", 100, new sale.Action ())
990:    {
991:      public void doAction (SaleProcess p, SalesPoint sp)
992:      {
993:        // 
994:      }
995:    });
996:  }
997:
998:  /**
999:   * 
1000:  */
1001:  protected void createFormSheetContent (FormSheet fs)
1002:  {
1003:    fs.removeAllButtons ();
1004:  }
1005:
1006:  /**
1007:   * 
1008:   */
1009:  protected void addContentCreator (FormSheetContentCreator creator)
1010:  {
1011:    fs.addContentCreator (creator);
1012:  }
1013:
1014:  /**
1015:   * 
1016:   */
1017:  protected void setNextTransition (Transition next)
1018:  {
1019:    selectionGate.setNextTransition (next);
1020:  }
1021:
1022:  /**
1023:   * 
1024:   */
1025:  protected void doAction (SaleProcess p, SalesPoint sp)
1026:  {
1027:    // 
1028:  }
1029:
1030:  /**
1031:   * 
1032:   */
1033:  protected void removeAllButtons (FormSheet fs)
1034:  {
1035:    fs.removeAllButtons ();
1036:  }
1037:
1038:  /**
1039:   * 
1040:   */
1041:  protected void addButton (FormSheet fs)
1042:  {
1043:    fs.addButton ("Ok", 100, new sale.Action ())
1044:    {
1045:      public void doAction (SaleProcess p, SalesPoint sp)
1046:      {
1047:        // 
1048:      }
1049:    });
1050:  }
1051:
1052:  /**
1053:   * 
1054:   */
1055:  protected void setNextTransition (Transition next)
1056:  {
1057:    selectionGate.setNextTransition (next);
1058:  }
1059:
1060:  /**
1061:   * 
1062:   */
1063:  protected void doAction (SaleProcess p, SalesPoint sp)
1064:  {
1065:    // 
1066:  }
1067:
1068:  /**
1069:   * 
1070:   */
1071:  protected void removeButtons (FormSheet fs)
1072:  {
1073:    fs.removeAllButtons ();
1074:  }
1075:
1076:  /**
1077:   * 
1078:   */
1079:  protected void addButtons (FormSheet fs)
1080:  {
1081:    fs.addButton ("Ok", 100, new sale.Action ())
1082:    {
1083:      public void doAction (SaleProcess p, SalesPoint sp)
1084:      {
1085:        // 
1086:      }
1087:    });
1088:  }
1089:
1090:  /**
1091:   * 
1092:   */
1093:  protected void createFormSheetContent (FormSheet fs)
1094:  {
1095:    fs.removeAllButtons ();
1096:  }
1097:
1098:  /**
1099:   * 
1100:   */
1101:  protected void addContentCreator (FormSheetContentCreator creator)
1102:  {
1103:    fs.addContentCreator (creator);
1104:  }
1105:
1106:  /**
1107:   * 
1108:   */
1109:  protected void setNextTransition (Transition next)
1110:  {
1111:    selectionGate.setNextTransition (next);
1112:  }
1113:
1114:  /**
1115:   * 
1116:   */
1117:  protected void doAction (SaleProcess p, SalesPoint sp)
1118:  {
1119:    // 
1120:  }
1121:
1122:  /**
1123:   * 
1124:   */
1125:  protected void removeAllButtons (FormSheet fs)
1126:  {
1127:    fs.removeAllButtons ();
1128:  }
1129:
1130:  /**
1131:   * 
1132:   */
1133:  protected void addButton (FormSheet fs)
1134:  {
1135:    fs.addButton ("Ok", 100, new sale.Action ())
1136:    {
1137:      public void doAction (SaleProcess p, SalesPoint sp)
1138:      {
1139:        // 
1140:      }
1141:    });
1142:  }
1143:
1144:  /**
1145:   * 
1146:   */
1147:  protected void setNextTransition (Transition next)
1148:  {
1149:    selectionGate.setNextTransition (next);
1150:  }
1151:
1152:  /**
1153:   * 
1154:   */
1155:  protected void doAction (SaleProcess p, SalesPoint sp)
1156:  {
1157:    // 
1158:  }
1159:
1160:  /**
1161:   * 
1162:   */
1163:  protected void removeButtons (FormSheet fs)
1164:  {
1165:    fs.removeAllButtons ();
1166:  }
1167:
1168:  /**
1169:   * 
1170:   */
1171:  protected void addButtons (FormSheet fs)
1172:  {
1173:    fs.addButton ("Ok", 100, new sale.Action ())
1174:    {
1175:      public void doAction (SaleProcess p, SalesPoint sp)
1176:      {
1177:        // 
1178:      }
1179:    });
1180:  }
1181:
1182:  /**
1183:   * 
1184:   */
1185:  protected void createFormSheetContent (FormSheet fs)
1186:  {
1187:    fs.removeAllButtons ();
1188:  }
1189:
1190:  /**
1191:   * 
1192:   */
1193:  protected void addContentCreator (FormSheetContentCreator creator)
1194:  {
1195:    fs.addContentCreator (creator);
1196:  }
1197:
1198:  /**
1199:   * 
1200:   */
1201:  protected void setNextTransition (Transition next)
1202:  {
1203:    selectionGate.setNextTransition (next);
1204:  }
1205:
1206:  /**
1207:   * 
1208:   */
1209:  protected void doAction (SaleProcess p, SalesPoint sp)
1210:  {
1211:    // 
1212:  }
1213:
1214:  /**
1215:   * 
1216:   */
1217:  protected void removeButtons (FormSheet fs)
1218:  {
1219:    fs.removeAllButtons ();
1220:  }
1221:
1222:  /**
1223:   * 
1224:   */
1225:  protected void addButtons (FormSheet fs)
1226:  {
1227:    fs.addButton ("Ok", 100, new sale.Action ())
1228:    {
1229:      public void doAction (SaleProcess p, SalesPoint sp)
1230:      {
1231:        // 
1232:      }
1233:    });
1234:  }
1235:
1236:  /**
1237:   * 
1238:   */
1239:  protected void createFormSheetContent (FormSheet fs)
1240:  {
1241:    fs.removeAllButtons ();
1242:  }
1243:
1244:  /**
1245:   * 
1246:   */
1247:  protected void addContentCreator (FormSheetContentCreator creator)
1248:  {
1249:    fs.addContentCreator (creator);
1250:  }
1251:
1252:  /**
1253:   * 
1254:   */
1255:  protected void setNextTransition (Transition next)
1256:  {
1257:    selectionGate.setNextTransition (next);
1258:  }
1259:
1260:  /**
1261:   * 
1262:   */
1263:  protected void doAction (SaleProcess p, SalesPoint sp)
1264:  {
1265:    // 
1266:  }
1267:
1268:  /**
1269:   * 
1270:   */
1271:  protected void removeButtons (FormSheet fs)
1272:  {
1273:    fs.removeAllButtons ();
1274:  }
1275:
1276:  /**
1277:   * 
1278:   */
1279:  protected void addButtons (FormSheet fs)
1280:  {
1281:    fs.addButton ("Ok", 100, new sale.Action ())
1282:    {
1283:      public void doAction (SaleProcess p, SalesPoint sp)
1284:      {
1285:        // 
1286:      }
1287:    });
1288:  }
1289:
1290:  /**
1291:   * 
1292:   */
1293:  protected void createFormSheetContent (FormSheet fs)
1294:  {
1295:    fs.removeAllButtons ();
1296:  }
1297:
1298:  /**
1299:   * 
1300:   */
1301:  protected void addContentCreator (FormSheetContentCreator creator)
1302:  {
1303:    fs.addContentCreator (creator);
1304:  }
1305:
1306:  /**
1307:   * 
1308:   */
1309:  protected void setNextTransition (Transition next)
1310:  {
1311:    selectionGate.setNextTransition (next);
1312:  }
1313:
1314:  /**
1315:   * 
1316:   */
1317:  protected void doAction (SaleProcess p, SalesPoint sp)
1318:  {
1319:    // 
1320:  }
1321:
1322:  /**
1323:   * 
1324:   */
1325:  protected void removeButtons (FormSheet fs)
1326:  {
1327:    fs.removeAllButtons ();
1328:  }
1329:
1330:  /**
1331:   * 
1332:   */
1333:  protected void addButtons (FormSheet fs)
1334:  {
1335:    fs.addButton ("Ok", 100, new sale.Action ())
1336:    {
1337:      public void doAction (SaleProcess p, SalesPoint sp)
1338:      {
1339:        // 
1340:      }
1341:
```

GiveBackProcess.java

```

127:         });
128:         // "Cancel"-Button einbauen
129:         fs.addButton ("Cancel", 101, new sale.Action()
130:         {
131:             public void doAction (SaleProcess p, SalesPoint sp)
132:             {
133:                 // Transition zum Rollback-Gate als naechste Transition setzen
134:                 selectionGate.setNextTransition(
135:                     GateChangeTransition.CHANGE_TO_ROLLBACK_GATE);
136:             }
137:         });
138:     }
139: }
140: }
141: // erstelltes FormSheet am zu betretenden Gate setzen
142: selectionGate.setFormSheet(tfs);
143: // als naechstes zu betretendes Gate zurueckgeben
144: return selectionGate;
145: }
146: }
147: }
148: }
149: // Gate zum Ausgeben des Restgeldes anlegen
150: giveTestGate = new UIGate(null, null);
151: // Parameter zum Aufsummieren des Datenkorbes festlegen
152: DataBaseBasketCondition dbc =
153:     DataBaseConditionImpl.ALL_STOCK_ITEMS;
154: toGetMoneyTransition = new Transition()
155: {
156:     public Gate perform(SaleProcess pOwner, User usr)
157:     {
158:         // Transition zum Gate, an dem das Restgeld gegeben wird
159:         DataBaseConditionImpl.allStock =
160:             DataBaseConditionImpl.ALL_STOCK_ITEMS;
161:         int sellValue = 0;
162:         int rentValue = 0;
163:         // Videobestand des Automaten
164:         CountingStockImpl videoStock =
165:             (CountingStockImpl)Shop.getTheShop ().getStock ("Video-countingstock");
166:         // aktuelle Zeit ermitteln
167:         Object date = Shop.getTheShop ().getTimer ().getTime ();
168:         // rueckzuhlenden Betrag ermitteln
169:         Iterator i = db.iterator (dbc);
170:         while (i.hasNext ())
171:             // spezielle Kassette ermitteln
172:             StoringStockItemBEntry cassetteItem =
173:                 (StoringStockItemBEntry)i.next ();
174:             CassetteStoringStockItem cassette =
175:                 (CassetteStoringStockItem)cassetteItem.getValue ();
176:             // Verkaufspreis bestimmen
177:             try {
178:                 sellValue =
179:                     ((NumberValue)((QuotValue)(videocatalog.get(
180:                         cassette.getName (), null, false).getValue ()).getBid (
181:                             .intValue ());
182:             }
183:             catch (VetoException ve) {}
184:             ((NumberValue)((QuotValue)(videocatalog.get(
185:                         cassette.getName (), null, false).getValue ()).getBid (
186:                             .intValue ());
187:             }
188:             catch (VetoException ve) {}
189:         }
190:         // mind. einen Tag ausgleichen?
191:         if (((Long) cassette.getDate ().intValue ()) > 0)
192:             - ((Long) cassette.getDate ().intValue ());
193:         // Verleihgebühr bestimmten
194:         rentValue = (((Long) date).intValue ())
195:             - ((Long) cassette.getDate ().intValue ());
196:             * rentPrice;
197:         }
198:         // Leihgebuehr kleiner als Verkaufspreis?
199:         if (rentValue < sellValue)
200:             // rueckzuhlenden Betrag erhöhen
201:             toPayBackValue = toPayBackValue + (sellValue - rentValue);
202:         }
203:         // Video in Automatenbestand zuruecklegen
204:         videotStock.add(cassette.getName (), 1, null);
205:         // Vorgang in Logdatei eintragen
206:         try {
207:             Log.getGlobalLog().log(new GiveBackLoggable(
208:                 cassetteItem.getSecondaryKey (),
209:                 customer.getCustomerID (), date));
210:         }
211:         catch (LogNoOutputStreamException lnoee) {}
212:         catch (IOException ioe) {}
213:     }
214:     // Leihgebuehr grösser als Verkaufspreis
215:     else {
216:         JOptionPane.showMessageDialog(null,
217:             cassette.getName () + " is Your's!");
218:     }
219:     // keinen Tag ausgleichen (trotzdem 3,00 DM Leihgebuehr)
220:     if (((Long) date).intValue () == 0)
221:         - ((Long) cassette.getDate ().intValue ());
222:     }
223:     // rueckzuhlenden Betrag erhöhen
224:     toPayBackValue = toPayBackValue + (sellValue - rentPrice);
225:     }
226:     }
227:     // Vorgang in Logdatei eintragen
228:     try {
229:         Log.getGlobalLog().log(new GiveBackLoggable(
230:             cassetteItem.getSecondaryKey (),
231:             customer.getCustomerID (), date));
232:     }
233:     catch (LogNoOutputStreamException lnoee) {}
234:     catch (IOException ioe) {}
235:     videotStock.add(cassette.getName (), 1, null);
236:     }
237:     // Vorgang in Logdatei eintragen
238:     try {
239:         Log.getGlobalLog().log(new GiveBackLoggable(
240:             cassetteItem.getSecondaryKey (),
241:             customer.getCustomerID (), date));
242:     }
243:     catch (LogNoOutputStreamException lnoee) {}
244:     catch (IOException ioe) {}
245:     videotStock.add(cassette.getName (), 1, null);
246:     }
247:     // ueberprüfen, ob der Kunde noch Videos im Bestand hat
248:     if (customer.getStoringStock ().size (null) == 0)
249:         }
250:         // ueberprüfen, ob der Kunde noch Videos im Bestand hat
251:         if (customer.getStoringStock ().size (null) == 0)
252:             }

```

GiveBackProcess.java

```

253:     VideoMachine.removeCustomer(customer);
254: 
255:     // rückzugebenden Betrag vom Geldbestand abziehen
256:     try {
257:         if (toPayBackValue > 0 ) {
258:             ((CountingStock)Shop.getTheShop()).getStock(
259:                 "coin slot").remove("1-Pfennig-Stück",
260:                                     toPayBackValue,
261:                                     powner.getBasket());
262:         }
263: 
264:         catch (vetoException ve) {}
265: 
266:         // am Gate darzustellendes FormSheet
267:         MsgForm mf = new MsgForm(
268:             "Give Rest",
269:             "You get " +
270:             myCurrency.toString(new IntegerValue(toPayBackValue)) +
271:             " back");
272: 
273:         // ContentCreator zur Neubeliegung des "OK"-Buttons hinzufügen
274:         mf.addContentCreator(new FormSheetContentCreator()
275:         {
276:             public void createFormSheetContent(FormSheet fs)
277:             {
278:                 // neue Aktion setzen
279:                 fs.getButton(FormSheet.BTNID_OK).setAction(new Action()
280:                     {
281:                         public void doAction(SaleProcess p, SalesPoint sp)
282:                         {
283:                             // zum Commit-Gate fuhrende Transition als
284:                             // naechste Transition setzen
285:                             giveRestGate.setNextTransition(
286:                                 GateChangeTransition.CHANGE_TO_COMMIT_GATE);
287:                         }
288:                     });
289:                 }
290:             );
291: 
292:             // erstelltes FormSheet am zu betretenden Gate setzen
293:             giveRestGate.setFormSheet(mf);
294: 
295:             // als naechstes zu betretendes Gate zurueckgeben
296:             return giveRestGate;
297:         });
298: 
299:         // FormSheetContentCreator am Kundennummer-Eingabe-Gate anmelden
300:         tif.addContentCreator(new FormSheetContentCreator()
301:         {
302:             protected void createFormSheetContent(FormSheet fs)
303:             {
304:                 // alle vorhandenen Buttons entfernen
305:                 fs.removeAllButtons();
306:             }
307: 
308:             // neuen "OK"-Button einbauen
309:             fs.addButton ("Ok", 100, new sale.Action()
310:             {
311:                 public void doAction (SaleProcess p, SalesPoint sp)
312:                 {
313:                     // Eingabe ueberpruefen
314:                     String customerId = tif.getText();
315:                     boolean isNotAnInteger = true;
316: 
317:                     try {
318:                         new Integer (customerId);
319:                         isNotAnInteger = false;
320:                     } catch (NumberFormatException nfe) {}
321: 
322:                     // Eingabe korrekt -> selectionGate
323:                     if (!isNotAnInteger && (new Integer(customerID)).intValue() > 0 )
324: 
325:                     // existiert der Kunde?
326:                     try {
327:                         boolean customerExists = false;
328: 
329:                         Set customerSet = VideoMachine.getAllCustomer();
330:                         Iterator i = customerSet.iterator();
331:                         while (i.hasNext() && !customerExists)
332:                         {
333:                             Customer myCustomer = (Customer)i.next();
334:                             if (myCustomer.getCustomerID().equals(customerID))
335:                             {
336:                                 customer = myCustomer;
337:                                 customerExists = true;
338: 
339:                                 if (customerExists)
340:                                     capabilityGate.setNextTransition(
341:                                         toSelectionTransition);
342: 
343:                                 else
344:                                     JOptionPane.showMessageDialog(null,
345:                                         "This CustomerID doesn't exist!");
346: 
347:                                     capabilityGate.setNextTransition(
348:                                         new GateChangeTransition(capabilityGate));
349: 
350:                                 }
351: 
352:                                 catch (NullPointerException ne)
353:                                     JOptionPane.showMessageDialog(null,
354:                                         "This CustomerID doesn't exist!");
355: 
356:                                 capabilityGate.setNextTransition(capabilityGate);
357: 
358:                             }
359: 
360:                         // Eingabe falsch -> Kunden informieren und
361:                         // Eingabe wiederholen
362:                         else
363:                             JOptionPane.showMessageDialog(null,
364:                                         "CustomerID must be a positive number!");
365: 
366:                         capabilityGate.setNextTransition(capabilityGate);
367: 
368:                     }
369:                 }
370: 
371:             // "Cancel"-Button einbauen
372:             fs.addButton ("Cancel", 101, new sale.Action()
373:             {
374:                 public void doAction (SaleProcess p, SalesPoint sp)
375:                 {
376:                     // Transition zum Rollback-Gate als naechste Transition setzen
377:                     capabilityGate.setNextTransition(
378:                         GateChangeTransition.CHANGE_TO_ROLLBACK_GATE);
379:                 }
380:             }
381:         }
382:     }
383: 
```

GiveBackProcess.java

```

379:         }
380:     );
381: }
382: }
383: // FormSheet am entsprechenden Gate setzen
384: capabilityGate.setFormSheet(tif);
385: capabilityGate.setLogEntry();
386: }
387: }
388: /**
389: * Beinhaltet den Logeintrag, der beim schlie&szlig;en des SalesPoints
390: * aufgerufen wird.
391: */
392: /**
393: protected void logSalesPointClosed()
394: {
395: }
396: /**
397: // public methods /////////////////////////////////
398: /**
399: /**
400: /**
401: * Gibt das Startgate des Prozesses zur&uuml;ck.
402: */
403: public Gate getInitialGate()
404: {
405: // Automat aufbauen
406: setupMachine();
407: /**
408: // ...und Capability-Gate als Startgate zurueckgeben
409: return capabilityGate;
410: }
411: /**
412: /**
413: * &Uuml;bergibt das Log-Gate. Hier das Stop-Gate, da beim Beenden des
414: * Prozesses kein Log-Eintrag geschrieben werden soll.
415: */
416: public Gate getLogGate()
417: {
418: return getStopGate();
419: }
420: */
421: }
422: }
423: /**
424: * Definiert Die Schnittstelle <CODE>Loggable</CODE>.
425: */
426: /*
427: class GiveBackLoggable implements Loggable
428: {
429: /**
430: // attributes ///////////////////////////////
431: /**
432: String name;
433: String customerId;
434: Object date;
435: /*
436: /**
437: // constructor ///////////////////////////////
438: /**
439: * Der Konstruktor legt ein neues Objekt der Klasse <CODE>MyLoggable</CODE>
440: * an, wobei der Konstruktor von <CODE>LogEntry</CODE> aufgerufen wird,
441: * wichtig f&uuml;r die Erstellung des Log-Eintrags.
442: * und die &uuml;bergebenen Variablen zugewiesen werden. Diese sind
443: * wichtig f&uuml;r die Erstellung des Log-Eintrags.
444: */
445: public GiveBackLoggable(String name,
446:                           String customerId, Object date)
447: {
448: super();
449: this.name = name;
450: this.customerID = customerId;
451: this.date = date;
452: }
453: /**
454: public GiveBackLoggable(StoringStockItemDBEntry cassetteItem,
455:                         Customer customer, Object date)
456: {
457: super();
458: name = cassetteItem.getSecondaryKey();
459: /**
460: name = cassetteItem.getCustomerID();
461: this.customerID = customerId;
462: this.date = date;
463: }
464: /**
465: // public methods ///////////////////////////////
466: /**
467: * Holt sich den Log-Eintrag, der aus den &uuml;bergebenen Daten
468: * zusammengestellt wird. Dazu wird die extra implementierte
469: * Klasse <CODE>MyLogEntry</CODE> genutzt.
470: */
471: /**
472: public LogEntry getLogData()
473: {
474: return new GiveBackLogEntry(name, customerId, date);
475: }
476: /**
477: /**
478: /**
479: /**
480: /**
481: * &Uuml;berschreibt die Klasse <CODE>LogEntry</CODE> um einen
482: * selbstdefinierten Log-Eintrag zu erm&ouml;glichen.
483: */
484: /**
485: class GiveBackLogEntry extends LogEntry
486: {
487: /**
488: /**
489: /**
490: /**
491: /**
492: /**
493: /**
494: /**
495: /**
496: /**
497: * Der Konstruktor legt ein neues Objekt der Klasse <CODE>MyLogEntry</CODE>
498: * an, wobei der Konstruktor von <CODE>LogEntry</CODE> aufgerufen wird,
499: * und die &uuml;bergebenen Variablen zugewiesen werden. Diese sind
500: * wichtig f&uuml;r die Erstellung des Log-Eintrags.
501: */
502: public GiveBackLogEntry(String name, String customerId, Object date)
503: {
504: super();
}

```

```
505:     this.name = name;
506:     this.customerID = customerID;
507:     this.date = date;
508: }
509:
510: /**
511:  * public methods ///////////////////////////////////////////////////
512: /**
513:  * Spezifiziert das Aussehen des Log-Eintrags.
514:  */
515: public String toString()
516: {
517:     return name +
518:         " gave back by customer " + customerID +
519:         " (ID) at turn " + date;
520: }
521: }
522: }
```

OfferTED.java

```
1: import data.*;
2: import data.swing.*;
3: import sale.*;
4:
5:
6:
7: /**
8:  * Ein TableEntryDescriptor zur Darstellung des Videoangebotes.
9: */
10: public class OfferTED extends DefaultMoneyBagItemTED
11: {
12:
13:     // attributes ///////////////////////////////
14:
15:     private boolean withCount; // Spalte fuer Anzahl mit anzeigen?
16:
17:
18:     // constructor ///////////////////////////////
19:
20:     /**
21:      * Konstruktor. Erzeugt einen neuen Objekt vom Typ OfferTED.
22:      */
23:     public OfferTED(boolean withCount)
24:     {
25:         super(CurrencyShop.getTheShop().getCatalog("DM"));
26:         this.withCount = withCount;
27:     }
28:
29:     // public methods ///////////////////////////////
30:
31:     /**
32:      * Gibt die Spaltenanzahl der Tabelle zur&uuml;ck.
33:      */
34:     public int getColumnCount()
35:     {
36:         return withCount?3:2;
37:     }
38:
39:
40:     /**
41:      * Gibt die &uml;berschrift einer Spalte zur&uuml;ck.
42:      */
43:     public String getColumnName(int nIndex)
44:     {
45:         return (new String[] {"Name", "Price", "Available"})[nIndex];
46:     }
47:
48:     /**
49:      * Gibt den Wert einer Tabellenzelle zur&uuml;ck.
50:      */
51:     public Object getValueAt(Object oRecord, int nIndex)
52:     {
53:         // wenn Preis angefordert...
54:         if (nIndex == 1)
55:             // die videocassette ermitteln
56:             VideoCassette videotape = (VideoCassette)(CountingStockTableModel.Record)oRecord.getDescriptor();
57:
58:             // den Verkaufspreis ermitteln und zurueckgeben
59:             return ((QuotValue)videocassette.getValue()).getBid();
60:
61:         else return super.getValueAt(oRecord, nIndex);
62:
63:     }
```

Office.java

```

1: import sale.*;
2: import sale.events.*;
3: import sale.stdfoms.*;
4: import data.events.*;
5: import data.oimpl.*;
6: import log.*;
7: import log.stdfoms.*;
8:
9: import java.io.*;
10:
11: /**
12:  * Diese Klasse implementiert das B&uuml;ro. Hier kann der Manager das
13:  * Logfile einsehen und Videos nachkaufen oder aus dem Bestand nehmen.
14:  */
15: /*
16:  public class Office extends SalesPoint
17:  {
18:    // attributes ///////////////////////////////
19:    // Managerpassword
20:    private static String password;
21:    /**
22:     * constructor ///////////////////////////////
23:    /**
24:    /**
25:    /**
26:    /**
27:    * Konstruktor erzeugt ein neues Objekt der Klasse <CODE>Office</CODE>.
28:    */
29:    public Office(String name)
30:    {
31:      super(name);
32:      // neuen StockChangeListener am Geldbestand des Automaten anmelden
33:      (MoneyBeginImpl)Shop.getTheShop().getStock
34:      ("coin slot").addStockChangeListener(new StockChangeAdapter());
35:      /**
36:       * StockItem wurde hinzugefugt
37:       */
38:      public void commitAddStockItems(StockChangeEvent sce)
39:      {
40:        // hat sich der Geldbestand geaendert, FormSheet neu darstellen
41:        if (getCurrentProcess() == null && hasUseableDisplay(null))
42:          try {
43:            setFormSheet(null, getDefaultFormSheet());
44:          }
45:          catch (InterruptedException iexc) {
46:            System.err.println("Update interrupted");
47:          }
48:      }
49:      /**
50:       * StockItem wurde geloescht
51:       */
52:      public void commitRemoveStockItems(StockChangeEvent sce)
53:      {
54:        // hat sich der Geldbestand geaendert, FormSheet neu darstellen
55:        if (getCurrentProcess() == null && hasUseableDisplay(null))
56:          try {
57:            setFormSheet(null, getDefaultFormSheet());
58:          }
59:          catch (InterruptedException iexc) {
60:            System.err.println("Update interrupted");
61:          }
62:      }
63:    }
64:    /**
65:     * neuen TimerListener am Timer des Automaten anmelden
66:     */
67:    Shop.getTheShop().getTimer().addTimerListener(new TimerAdapter()
68:    {
69:      /**
70:       * wurde Zeit weitergeschalten, FormSheet neu darstellen
71:       */
72:       try {
73:         setFormSheet(null, getDefaultFormSheet());
74:       }
75:       catch (InterruptedException ie) {}
76:     }
77:   );
78: }
79:
80: /**
81:  * Gibt das Standard-FormSheet für das B&uuml;ro zurück.
82:  */
83: /**
84:  * Gibt das Standard-FormSheet für das B&uuml;ro zur&uuml;ck.
85:  */
86: public FormSheet getDefaultFormSheet()
87: {
88:   return new FormSheet("Office", new DefaultOfficeFormCreator(this), false);
89: }
90: /**
91:  * Gibt das Standard-MenuSheet zurück.
92:  */
93: /**
94:  * Public MenuSheet getDefaultMenuSheet()
95:  {
96:    /**
97:     * neues Menue anlegen, wird Wartungsmenu
98:     */
99:    MenuSheet msSubMenu = new MenuSheet("Maintenance");
100:   /**
101:    * neuen Menupunkt zum Weiterschalten der Zeit anlegen
102:    */
103:   msSubMenu.add(new MenuSheetItem("Advance time", new sale.Action()
104:   {
105:     /**
106:      * Zeit weiterschalten
107:      */
108:      Shop.getTheShop().getTimer().goAhead();
109:    }
110:   );
111:   /**
112:    * public void doAction(SaleProcess p, SalesPoint sp)
113:    */
114:   try {
115:     /**
116:      * LogTableForm erzeugen
117:      */
118:      FileInputStream fis = new FileInputStream("machine.log");
119:      LogInputStream lis = new LogInputStream(fis);
120:      LogTableForm ltf = new LogTableForm("View log file", lis);
121:      /**
122:       * "Cancel"-Button entfernen
123:       */
124:      ltf.removeButton(FormSheet.BTNID_CANCEL);
125:   }
126: }

```

Office.java

```
127: // falls Datei nicht existiert
128: catch (FileNotFoundException fnfexc) {
129:     try {
130:         setFormSheet(null, new MsgForm("Error", "Log file not found."));
131:     }
132:     catch (InterruptedException iexc) {}
133: }
134:
135: // falls Fehler beim Lesen der Datei
136: catch (IOException iexc) {
137:     try {
138:         setFormSheet(null, new MsgForm("Error",
139:                                         "Log file corrupt. It might be empty."));
140:     }
141:     catch (InterruptedException iexc) {}
142: }
143:
144: // falls Setzen des FormSheets unterbrochen
145: catch (InterruptedException iexc) {
146:     try {
147:         setFormSheet (null, new MsgForm ("Error", iexc.toString()));
148:     }
149:     catch (InterruptedException iexc2) {}
150: }
151: }
152: );
153:
154:
155: // neuen Menuepunkt zum Einsehen und Editieren des Bestandes
156: msSubMenu.add(new MenuItemItem ("Edit Video-Stock", new sale.Action()
157: {
158:     public void doAction(SaleProcess p, SalesPoint sp)
159:     {
160:         // Code zum Ausfuehren des Prozesses
161:         sp.runProcess(new SeeVideoStockProcess());
162:     }
163: });
164:
165: // neues Menue anlegen, wird Menueleiste
166: MenuSheet msMenu = new MenuSheet("office menu");
167:
168: // Wartungsmeine in Menueleiste des Bueros einbauen
169: msMenu.add(msSubMenu);
170:
171: // Menueleiste zurueckgeben
172: return msMenu;
173: }
174:
175:
176:
177: /**
178: * Setzt das Managerpasswort.
179: */
180: public static void setPassword(String password)
181: {
182:     Office.password = password;
183: }
184:
185:
186: /**
187: * Testet ob das ueumlbergabeene Passwort das Managerpasswort ist.
188: */
189: public static boolean testPassword(String password)
```

```

1: import sale.*;
2: import sale.stdforms.*;
3: import data.*;
4: import data.oobjimpl.*;
5: import data.stdforms.*;
6: import users.*;
7: import log.*;
8:
9: import java.util.*;
10: import java.text.*;
11: import java.lang.*;
12: import java.io.*;
13:
14: import javax.swing.JTextField;
15: import javax.swing.JPanel;
16: import javax.swing.JOptionPane;
17: import javax.swing.BoxLayout;
18: import javax.swing.JLabel;
19:
20: /**
21:  * Verkaufs- bzw. Verleihprozess, der von einem registrierten Kunden
22:  * am Automaten durchgefuumlt werden kann.
23:  */
24: public class RentProcess extends SaleProcess
25: {
26:
27: /**
28: // attributes
29: /**
30: // Gates
31: protected UIGate capabilityGate;
32: protected UIGate selectionGate;
33: protected UIGate rentGate;
34: protected UIGate decisionGate;
35: protected Gate decisionGate;
36: protected UIGate getChangeGate;
37:
38: // Transitions
39: protected Transition toSelectionTransition;
40: protected Transition toPayingTransition;
41: protected Transition toDecisionTransition;
42: protected Transition toGetChangeTransition;
43:
44: // verwendete Waehrung
45: protected Currency myCurrency;
46:
47: // zu zahlender Betrag
48: private IntegerValue topayValue;
49:
50: // gezahlter Betrag
51: private IntegerValue paidValue;
52:
53: // Bewertung der Relation zwischen zu zahlendem und gezahltem Betrag
54: private int payAssessment;
55:
56: // Kunde, der ausleihen moechte
57: private Customer customer;
58:
59: /**
60: // constructor
61: /**
62: * Erzeugt ein neues Objekt der Klasse RentProcess.
63:
64: /*
65: public RentProcess()
66: {
67: super ("Rent-Process");
68: }
69:
70: // protected methods
71:
72: /**
73: * Baut die Oberfluumle der Verleihvorgang auf.
74: */
75: protected void setupMachine()
76: {
77: // verwendete Wahlurne ermitteln
78: myCurrency = (Currency) Shop.getTheShop().getCatalog("DM");
79:
80: // Gate zur Kundennummer-Abfrage anzeigen
81: capabilityGate = new UIGate(null, null);
82:
83: // Formsheet fuer Eingabe der Kundennummer
84: final TextInputForm tif = new TextInputForm("Customer-ID",
85: "Customer-ID",
86: " ");
87:
88: // Auswahl-Gate anlegen
89: selectionGate = new UIGate(null, null);
90:
91: // Gate zum Ausleihen/Bezahlen anlegen
92: rentGate = new UIGate(null, null);
93:
94: // Gate zum Ausgeben des Wechselgeldes anlegen
95: getChangeGate = new UIGate(null, null);
96:
97: // zu verwendenden Datenkorbo ermitteln
98: final DataBasket db = getBasket();
99:
100: // zu verwendenden Bestand ermitteln
101: final CountingStockImpl cs =
102: (CountingStockImpl) Shop.getTheShop().getStock("Video-CountingStock");
103:
104: // Transition zum Selection Gate
105: toselectionTransition = new Transition()
106: {
107: public Gate perform(SaleProcess power, User user)
108: {
109: // Quell-CountingStock
110: // am Gate darzustellendes FormSheet
111: TwoTableFormSheet ttfs = TwoTableFormSheet.create(
112: "Make your selection", // Titel des FormSheets
113: cs,
114: db, // Ziel-DataBasket
115: selectionGate, // Gate, an dem das FormSheet darzustellen ist
116: null, // Comparator fuer Quelltabelle
117: null, // Comparator fuer Ziel-Tabelle
118: false, // Zeilen mit Anzahl == 0 in der Quelltabelle anzeigen?
119: new OfferTED(true), // TableEntryDescriptor fur Quelltabelle
120: null, // TableEntryDescriptor fur Zieltabelle
121: null, // Transferstrategie
122: );
123:
124: // FormSheetContentCreator am Auswahl-Gate anmelden
125: ttfs.addContentCreator(new FormSheetContentCreator()
126: {

```

```

127:     protected void createFormSheetContent (FormSheet fs)
128:     {
129:         // alle vorhandenen Buttons entfernen
130:         fs.removeAllButtons ();
131:         // neuen "OK"-Button einbauen
132:         fs.addButton ("Ok", 100, new sale.Action()
133:         {
134:             public void doAction (SaleProcess p, SalesPoint sp)
135:             {
136:                 // Transition zum Bezahlen als naechste Transition setzen
137:                 selectionGate.setNextTransition(toPayingTransition);
138:             }
139:         });
140:         // "Cancel"-Button einbauen
141:         fs.addButton ("Cancel", 101, new sale.Action()
142:         {
143:             // Transition zum Rollback-Gate als naechste Transition setzen
144:             selectionGate.setNextTransition(
145:                 public void doAction (SaleProcess p, SalesPoint sp)
146:                 {
147:                     // Transition zum Rollback-Gate als naechste Transition setzen
148:                     selectionGate.setNextTransition(
149:                         GateChangeTransition.CHANGE_TO_ROLLBACK_GATE);
150:                     }
151:                 );
152:             }
153:         });
154:         // erstelltes FormSheet am zu betretenden Gate setzen
155:         selectionGate.setFormSheet(tfs);
156:         // als naechstes zu betretendes Gate zurueckgeben
157:         return selectionDate;
158:     }
159:     return selectionDate;
160: }
161: }
162: // Transition zum Wechselgeld-Gate
163: toGetChangeTransition = new Transition()
164: {
165:     public Gate perform(SaleProcess pOwner, User usr)
166:     {
167:         // am Gate darzustellendes FormSheet
168:         MsgForm mf = new MsgForm(
169:             "Get Change",
170:             "You get " +
171:                 myCurrency.toString((IntegerValue)paidValue.subtract(toPayValue)) +
172:                 " Change.");
173:     }
174:     // ContentCreator zur Neubeliegung des "OK"-Buttons hinzufuegen
175:     mf.addContentCreator(new FormSheetContentCreator()
176:     {
177:         public void createFormSheetContent (FormSheet fs)
178:         {
179:             // neue Aktion setzen
180:             fs.getButton(FormSheet_BTNID_OK).setAction(new Action()
181:             {
182:                 public void doAction(SaleProcess p, SalesPoint sp)
183:                 {
184:                     // zum Commit-Gate fuhrende Transition als
185:                     // naechste Transition setzen
186:                     getChangeGate.setNextTransition(
187:                         GateChangeTransition.CHANGE_TO_COMMIT_GATE);
188:                 }
189:             });
190:         }
191:     });
192: }
193: // erstelltes FormSheet am zu betretenden Gate setzen
194: getChangeGate.setFormSheet(mf);
195: // als naechstes zu betretendes Gate zurueckgeben
196: return getChangeGate();
197: // Gate anlegen, das entscheidet, ob Wechselgeld gegeben, zum Bezahlen
198: // oder zum Commit-Gate gegangen wird
199: decisionGate = new Gate()
200: {
201:     public Transition getNextTransition(SaleProcess pOwner, User usr)
202:     throws InterruptedException
203:     {
204:         switch(payAssessment) {
205:             // zuwenig bezahlt - Message anzeigen und zurueck zum Bezahlen
206:             case -1:
207:                 FormSheet mf = new MsgForm("Error",
208:                     "You have to pay more!",
209:                     false);
210:                 pOwner.getContext().popUpFormSheet(pOwner, mf);
211:             return new GateChangeTransition(rentGate);
212:         }
213:         // genau richtig bezahlt - zum Commit-Gate
214:         case 0:
215:             return GateChangeTransition.CHANGE_TO_COMMIT_GATE;
216:         // zuviel bezahlt - zur Wechselgeldausgabe
217:         case 1:
218:             return toGetChangeTransition;
219:         case 2:
220:             // falscher Wert in payAssessment - Fehlermeldung und Abbruch
221:             return GateChangeTransition.CHANGE_TO_QUIT_GATE;
222:         case 3:
223:             // zuviel bezahlt - zur Wechselgeldausgabe
224:             return toGetChangeTransition;
225:         case 4:
226:             // falscher Wert in payAssessment - Fehlernachricht und Abbruch
227:             default:
228:                 FormSheet mf2 = new MsgForm("Error",
229:                     "Internal error at Decision Gate. " +
230:                     "Will quit process. " ,
231:                     false);
232:                 pOwner.getContext().popUpFormSheet(pOwner, mf2);
233:                 return GateChangeTransition.CHANGE_TO_QUIT_GATE;
234:             }
235:         }
236:     }
237:     // Transition zum Entscheidungsgate
238:     toDecisionTransition = new Transition()
239:     {
240:         public Gate perform (SaleProcess powner, User usr)
241:         {
242:             // ausreichend Geld gegeben?
243:             if (paidValue.compareTo (toPayValue) >= 0) {
244:                 // zu zahlenden Betrag im Safe des Restaurants ablegen, wenn
245:                 // ueberhaupt etwas zu bezahlen ist
246:                 if (toPayValue.getvalue()>0)
247:                     ((CountingStock)Shop.getTheShop()).getStock(
248:                         "coin slot").add("1-Pfennig-Stück",
249:                             toPayValue.getvalue().intValue());
250:             }
251:         }
252:     }

```

RentProcess.java

```

316: // Genauso richtig gezahlt? - payAssessment entsprechend belegen
317: if (paidValue.compareTo(toPayValue) == 0)
318:     final Object date = Shop.getTheShop().getTimer().getTime();
319: else
320:     payAssessment = 1;
321: }
322: while (i.hasNext()) {
323:     final CountingStockItemDBEntry cassetteItem =
324:         (CountingStockItemDBEntry)i.next();
325:     int number = cassetteItem.count();
326:     for ( ; number > 0; number --) {
327:         // Vorgang in Logdatei eintragen
328:         try {
329:             Log.getGlobalLog().log(new MyLoggable(
330:                 cassetteItem.getSecondaryKey(),
331:                 customer.getCustomerID(), date));
332:         }
333:         catch (LogNoOutputStreamException lnoe) {}
334:         catch (IOException ioe) {}
335:         // Kassetten in Bestände uebernehmen
336:         customer.addVideoCassette(new CassetteStorageStockItem(
337:             cassetteItem.getSecondaryKey(), date));
338:     }
339: }
340: }
341: }
342: // ... und naechste Transition setzen
343: rentGate.setNextTransition(toDecisionTransition);
344: }
345: catch(ParseException pexc) {
346:     // eingegebener Text konnte nicht verarbeitet werden
347:     // Meldung erzeugen und ausgeben
348:     MsgForm mf = new MsgForm("Error",
349:         "The specified amount does " +
350:         "not have an appropriate format.");
351: }
352: try {
353:     p.getContext().popupFormSheet(p, mf);
354: }
355: catch(InterruptedException iexc) {
356: }
357: }
358: }
359: }
360: }
361: // "Back"-Button einbauen
362: fs.addButton("Back", 101, new sale.Action());
363: }
364: public void doAction(SaleProcess p, SalesPoint sp)
365: {
366:     // naechste Transition setzen, fuehrt ohne weitere Aktionen
367:     // zum Auswahl-Gate
368:     rentGate.setNextTransition(new GateChangeTransition(selectionGate));
369: }
370: }
371: // "Cancel"-Button einbauen
372: fs.addButton("Cancel", 102, new sale.Action());
373: }
374: // "OK"-Button einbauen
375: fs.removeAllButtons();
376: public void doAction(SaleProcess p, SalesPoint sp)
377: {
378:     // eingegabenen Text verarbeiten
379:     paidValue =

```

```

379:     // naechste Transition setzen, fuehrt zum Rollback-Gate
380:     rentGate.setNextTransition(   // Eingabe wiederholen
381:         GateChangeTransition.CHANGE_TO_ROLLBACK_GATE);
382:     });
383:   });
384: }
385: });
386: });
387: // FormSheet am entsprechenden Gate setzen
388: rentGate.setFormSheet(tif);
389: // ...und naechstes zu betretendes Gate zurueckgeben
390: // ...und naechstes zu betretendes Gate zurueckgeben
391: return rentGate;
392: );
393: );
394: );
395: // FormSheetContentCreator am Kundennummer-Eingabe-Gate anmelden
396: if(f.addContentCreator(new FormSheetContentCreator()
397: {
398:     protected void createFormSheetContent (FormSheet fs)
399:     {
400:         // alle vorhandenen Buttons entfernen
401:         fs.removeAllButtons();
402:         // neuen "OK"-Button einbauen
403:         fs.addButton ("Ok", 100, new sale.Action()
404:             {
405:                 public void doAction (SaleProcess p, SalesPoint sp)
406:                 {
407:                     public void doAction (SaleProcess p, SalesPoint sp)
408:                     {
409:                         // Eingabe ueberpruefen
410:                         String customerId = tif.getText();
411:                         boolean isNotAnInteger = true;
412:                         try {
413:                             new Integer(customerId);
414:                             isNotAnInteger = false;
415:                         }
416:                         catch(NumberFormatException nfe) {
417:                             isNotAnInteger = true;
418:                         }
419:                         // Eingabe korrekt -> selectionGate
420:                         if (!isNotAnInteger && (new Integer(customerID)).intValue() > 0) {
421:                             // nein, dann neu anlegen und der Liste reg. Kunden hinzufuegen
422:                             VideoMachine.addCustomer(customer);
423:                             // existiert der Kunde?
424:                             try {
425:                                 customer = new Customer(customerID);
426:                             }
427:                             capabilityGate.setNextTransition(toSelectionTransition);
428:                             // ja, dann diesen holen
429:                             catch (DuplicateKeyException dke) {
430:                                 customer = VideoMachine.getCustomerByID(customerID);
431:                             }
432:                             capabilityGate.setNextTransition(capabilityGate);
433:                             catch (DuplicateKeyException dke) {
434:                                 customer = VideoMachine.getCustomerByID(customerID);
435:                             }
436:                             * Definiert Die Schnittstelle <Code>Loggable</CODE>.
437:                             */
438:                             capabilityGate.setNextTransition(capabilityGate);
439:                             */
440:                             // Eingabe falsch -> Kunden informieren und
441:                             // attributes
442:                         }
443:                         else {
444:                             JOptionPane.showMessageDialog(null,
445:                                 "Customer ID must be a positive Number!");
446:                             capabilityGate.setNextTransition(capabilityGate);
447:                         }
448:                         */
449:                     }
450:                 }
451:             }
452:             /*
453:             fs.addButton ("Cancel", 101, new sale.Action()
454:                 {
455:                     public void doAction (SaleProcess p, SalesPoint sp)
456:                     {
457:                         // Transition zum Rollback-Gate als naechste Transition setzen
458:                         capabilityGate.setNextTransition(
459:                             GateChargeTransition.CHANGE_TO_ROLLBACK_GATE);
460:                         */
461:                     }
462:                 }
463:             );
464:             /*
465:             capabilityGate.setFormSheet(tif);
466:             */
467:         }
468:     }
469: );
470: /**
471:  */
472: /**
473:  */
474: /**
475:  */
476: /**
477:  */
478: /**
479:  */
480: /**
481:  */
482: /**
483:  */
484: /**
485:  */
486: /**
487:  */
488: /**
489:  */
490: /**
491:  */
492: /**
493:  */
494: /**
495:  */
496: /**
497:  */
498: /**
499:  */
500: /**
501:  */
502: /**
503:  */
504: /**
505:  */
506: /**
507:  */
508: /**
509:  */
510: /**
511:  */
512: /**
513:  */
514: /**
515:  */
516: /**
517:  */
518: /**
519:  */
520: /**
521:  */
522: /**
523:  */
524: /**
525:  */
526: /**
527:  */
528: /**
529:  */
530: /**
531:  */
532: /**
533:  */
534: /**
535:  */
536: /**
537:  */
538: /**
539:  */
540: /**
541:  */
542: /**
543:  */
544: /**
545:  */
546: /**
547:  */
548: /**
549:  */
550: /**
551:  */
552: /**
553:  */
554: /**
555:  */
556: /**
557:  */
558: /**
559:  */
560: /**
561:  */
562: /**
563:  */
564: /**
565:  */
566: /**
567:  */
568: /**
569:  */
570: /**
571:  */
572: /**
573:  */
574: /**
575:  */
576: /**
577:  */
578: /**
579:  */
580: /**
581:  */
582: /**
583:  */
584: /**
585:  */
586: /**
587:  */
588: /**
589:  */
590: /**
591:  */
592: /**
593:  */
594: /**
595:  */
596: /**
597:  */
598: /**
599:  */
599: /**
600:  */
601: /**
602:  */
603: /**
604:  */
605: /**
606:  */
607: /**
608:  */
609: /**
609:  */
610: /**
611:  */
612: /**
613:  */
614: /**
615:  */
616: /**
617:  */
618: /**
619:  */
620: /**
621:  */
622: /**
623:  */
624: /**
625:  */
626: /**
627:  */
628: /**
629:  */
630: /**
631:  */
632: /**
633:  */
634: /**
635:  */
636: /**
637:  */
638: /**
639:  */
640: /**
641:  */
642: /**
643:  */
644: /**
645:  */
646: /**
647:  */
648: /**
649:  */
650: /**
651:  */
652: /**
653:  */
654: /**
655:  */
656: /**
657:  */
658: /**
659:  */
660: /**
661:  */
662: /**
663:  */
664: /**
665:  */
666: /**
667:  */
668: /**
669:  */
670: /**
671:  */
672: /**
673:  */
674: /**
675:  */
676: /**
677:  */
678: /**
679:  */
680: /**
681:  */
682: /**
683:  */
684: /**
685:  */
686: /**
687:  */
688: /**
689:  */
689: /**
690:  */
691: /**
692:  */
693: /**
694:  */
695: /**
696:  */
697: /**
698:  */
699: /**
699:  */
700: /**
701:  */
702: /**
703:  */
704: /**
705:  */
706: /**
707:  */
708: /**
709:  */
709: /**
710:  */
711: /**
712:  */
713: /**
714:  */
715: /**
716:  */
717: /**
718:  */
719: /**
719:  */
720: /**
721:  */
722: /**
723:  */
724: /**
725:  */
726: /**
727:  */
728: /**
729:  */
729: /**
730:  */
731: /**
732:  */
733: /**
734:  */
735: /**
736:  */
737: /**
738:  */
739: /**
739:  */
740: /**
741:  */
742: /**
743:  */
744: /**
745:  */
746: /**
747:  */
747: /**
748:  */
749: /**
749:  */
750: /**
751:  */
752: /**
753:  */
754: /**
755:  */
756: /**
757:  */
758: /**
759:  */
759: /**
760:  */
761: /**
762:  */
763: /**
764:  */
765: /**
766:  */
767: /**
768:  */
769: /**
769:  */
770: /**
771:  */
772: /**
773:  */
774: /**
775:  */
776: /**
777:  */
778: /**
779:  */
779: /**
780:  */
781: /**
782:  */
783: /**
784:  */
785: /**
786:  */
787: /**
787:  */
788: /**
789:  */
789: /**
790:  */
791: /**
792:  */
793: /**
794:  */
795: /**
796:  */
797: /**
797:  */
798: /**
799:  */
799: /**
800:  */
801: /**
802:  */
803: /**
804:  */
805: /**
806:  */
807: /**
808:  */
809: /**
809:  */
810: /**
811:  */
812: /**
813:  */
814: /**
815:  */
815: /**
816:  */
817: /**
818:  */
819: /**
819:  */
820: /**
821:  */
822: /**
823:  */
824: /**
825:  */
826: /**
827:  */
827: /**
828:  */
829: /**
829:  */
830: /**
831:  */
832: /**
833:  */
834: /**
835:  */
835: /**
836:  */
837: /**
837:  */
838: /**
839:  */
839: /**
840:  */
841: /**
841:  */
842: /**
843:  */
844: /**
844:  */
845: /**
846:  */
846: /**
847:  */
847: /**
848:  */
848: /**
849:  */
849: /**
850:  */
851: /**
851:  */
852: /**
853:  */
853: /**
854:  */
855: /**
855:  */
856: /**
857:  */
857: /**
858:  */
859: /**
859:  */
860: /**
861:  */
861: /**
862:  */
862: /**
863:  */
864: /**
864:  */
865: /**
866:  */
866: /**
867:  */
868: /**
868:  */
869: /**
869:  */
870: /**
871:  */
871: /**
872:  */
872: /**
873:  */
873: /**
874:  */
874: /**
875:  */
875: /**
876:  */
876: /**
877:  */
877: /**
878:  */
878: /**
879:  */
879: /**
880:  */
880: /**
881:  */
881: /**
882:  */
882: /**
883:  */
883: /**
884:  */
884: /**
885:  */
885: /**
886:  */
886: /**
887:  */
887: /**
888:  */
888: /**
889:  */
889: /**
889:  */
890: /**
890:  */
891: /**
891:  */
892: /**
892:  */
893: /**
893:  */
894: /**
894:  */
895: /**
895:  */
896: /**
896:  */
897: /**
897:  */
898: /**
898:  */
899: /**
899:  */
900: /**
900:  */
901: /**
901:  */
902: /**
902:  */
903: /**
903:  */
904: /**
904:  */
905: /**
905:  */
906: /**
906:  */
907: /**
907:  */
908: /**
908:  */
909: /**
909:  */
910: /**
910:  */
911: /**
911:  */
912: /**
912:  */
913: /**
913:  */
914: /**
914:  */
915: /**
915:  */
916: /**
916:  */
917: /**
917:  */
918: /**
918:  */
919: /**
919:  */
920: /**
920:  */
921: /**
921:  */
922: /**
922:  */
923: /**
923:  */
924: /**
924:  */
925: /**
925:  */
926: /**
926:  */
927: /**
927:  */
928: /**
928:  */
929: /**
929:  */
930: /**
930:  */
931: /**
931:  */
932: /**
932:  */
933: /**
933:  */
934: /**
934:  */
935: /**
935:  */
936: /**
936:  */
937: /**
937:  */
938: /**
938:  */
939: /**
939:  */
940: /**
940:  */
941: /**
941:  */
942: /**
942:  */
943: /**
943:  */
944: /**
944:  */
945: /**
945:  */
946: /**
946:  */
947: /**
947:  */
948: /**
948:  */
949: /**
949:  */
950: /**
950:  */
951: /**
951:  */
952: /**
952:  */
953: /**
953:  */
954: /**
954:  */
955: /**
955:  */
956: /**
956:  */
957: /**
957:  */
958: /**
958:  */
959: /**
959:  */
960: /**
960:  */
961: /**
961:  */
962: /**
962:  */
963: /**
963:  */
964: /**
964:  */
965: /**
965:  */
966: /**
966:  */
967: /**
967:  */
968: /**
968:  */
969: /**
969:  */
970: /**
970:  */
971: /**
971:  */
972: /**
972:  */
973: /**
973:  */
974: /**
974:  */
975: /**
975:  */
976: /**
976:  */
977: /**
977:  */
978: /**
978:  */
979: /**
979:  */
980: /**
980:  */
981: /**
981:  */
982: /**
982:  */
983: /**
983:  */
984: /**
984:  */
985: /**
985:  */
986: /**
986:  */
987: /**
987:  */
988: /**
988:  */
989: /**
989:  */
990: /**
990:  */
991: /**
991:  */
992: /**
992:  */
993: /**
993:  */
994: /**
994:  */
995: /**
995:  */
996: /**
996:  */
997: /**
997:  */
998: /**
998:  */
999: /**
999:  */
1000: /**
1000:  */
1001: /**
1001:  */
1002: /**
1002:  */
1003: /**
1003:  */
1004: /**
1004:  */
1005: /**
1005:  */
1006: /**
1006:  */
1007: /**
1007:  */
1008: /**
1008:  */
1009: /**
1009:  */
1010: /**
1010:  */
1011: /**
1011:  */
1012: /**
1012:  */
1013: /**
1013:  */
1014: /**
1014:  */
1015: /**
1015:  */
1016: /**
1016:  */
1017: /**
1017:  */
1018: /**
1018:  */
1019: /**
1019:  */
1020: /**
1020:  */
1021: /**
1021:  */
1022: /**
1022:  */
1023: /**
1023:  */
1024: /**
1024:  */
1025: /**
1025:  */
1026: /**
1026:  */
1027: /**
1027:  */
1028: /**
1028:  */
1029: /**
1029:  */
1030: /**
1030:  */
1031: /**
1031:  */
1032: /**
1032:  */
1033: /**
1033:  */
1034: /**
1034:  */
1035: /**
1035:  */
1036: /**
1036:  */
1037: /**
1037:  */
1038: /**
1038:  */
1039: /**
1039:  */
1040: /**
1040:  */
1041: /**
1041:  */
1042: /**
1042:  */
1043: /**
1043:  */
1044: /**
1044:  */
1045: /**
1045:  */
1046: /**
1046:  */
1047: /**
1047:  */
1048: /**
1048:  */
1049: /**
1049:  */
1050: /**
1050:  */
1051: /**
1051:  */
1052: /**
1052:  */
1053: /**
1053:  */
1054: /**
1054:  */
1055: /**
1055:  */
1056: /**
1056:  */
1057: /**
1057:  */
1058: /**
1058:  */
1059: /**
1059:  */
1060: /**
1060:  */
1061: /**
1061:  */
1062: /**
1062:  */
1063: /**
1063:  */
1064: /**
1064:  */
1065: /**
1065:  */
1066: /**
1066:  */
1067: /**
1067:  */
1068: /**
1068:  */
1069: /**
1069:  */
1070: /**
1070:  */
1071: /**
1071:  */
1072: /**
1072:  */
1073: /**
1073:  */
1074: /**
1074:  */
1075: /**
1075:  */
1076: /**
1076:  */
1077: /**
1077:  */
1078: /**
1078:  */
1079: /**
1079:  */
1080: /**
1080:  */
1081: /**
1081:  */
1082: /**
1082:  */
1083: /**
1083:  */
1084: /**
1084:  */
1085: /**
1085:  */
1086: /**
1086:  */
1087: /**
1087:  */
1088: /**
1088:  */
1089: /**
1089:  */
1090: /**
1090:  */
1091: /**
1091:  */
1092: /**
1092:  */
1093: /**
1093:  */
1094: /**
1094:  */
1095: /**
1095:  */
1096: /**
1096:  */
1097: /**
1097:  */
1098: /**
1098:  */
1099: /**
1099:  */
1100: /**
1100:  */
1101: /**
1101:  */
1102: /**
1102:  */
1103: /**
1103:  */
1104: /**
1104:  */
1105: /**
1105:  */
1106: /**
1106:  */
1107: /**
1107:  */
1108: /**
1108:  */
1109: /**
1109:  */
1110: /**
1110:  */
1111: /**
1111:  */
1112: /**
1112:  */
1113: /**
1113:  */
1114: /**
1114:  */
1115: /**
1115:  */
1116: /**
1116:  */
1117: /**
1117:  */
1118: /**
1118:  */
1119: /**
1119:  */
1120: /**
1120:  */
1121: /**
1121:  */
1122: /**
1122:  */
1123: /**
1123:  */
1124: /**
1124:  */
1125: /**
1125:  */
1126: /**
1126:  */
1127: /**
1127:  */
1128: /**
1128:  */
1129: /**
1129:  */
1130: /**
1130:  */
1131: /**
1131:  */
1132: /**
1132:  */
1133: /**
1133:  */
1134: /**
1134:  */
1135: /**
1135:  */
1136: /**
1136:  */
1137: /**
1137:  */
1138: /**
1138:  */
1139: /**
1139:  */
1140: /**
1140:  */
1141: /**
1141:  */
1142: /**
1142:  */
1143: /**
1143:  */
1144: /**
1144:  */
1145: /**
1145:  */
1146: /**
1146:  */
1147: /**
1147:  */
1148: /**
1148:  */
1149: /**
1149:  */
1150: /**
1150:  */
1151: /**
1151:  */
1152: /**
1152:  */
1153: /**
1153:  */
1154: /**
1154:  */
1155: /**
1155:  */
1156: /**
1156:  */
1157: /**
1157:  */
1158: /**
1158:  */
1159: /**
1159:  */
1160: /**
1160:  */
1161: /**
1161:  */
1162: /**
1162:  */
1163: /**
1163:  */
1164: /**
1164:  */
1165: /**
1165:  */
1166: /**
1166:  */
1167: /**
1167:  */
1168: /**
1168:  */
1169: /**
1169:  */
1170: /**
1170:  */
1171: /**
1171:  */
1172: /**
1172:  */
1173: /**
1173:  */
1174: /**
1174:  */
1175: /**
1175:  */
1176: /**
1176:  */
1177: /**
1177:  */
1178: /**
1178:  */
1179: /**
1179:  */
1180: /**
1180:  */
1181: /**
1181:  */
1182: /**
1182:  */
1183: /**
1183:  */
1184: /**
1184:  */
1185: /**
1185:  */
1186: /**
1186:  */
1187: /**
1187:  */
1188: /**
1188:  */
1189: /**
1189:  */
1190: /**
1190:  */
1191: /**
1191:  */
1192: /**
1192:  */
1193: /**
1193:  */
1194: /**
1194:  */
1195: /**
1195:  */
1196: /**
1196:  */
1197: /**
1197:  */
1198: /**
1198:  */
1199: /**
1199:  */
1200: /**
1200:  */
1201: /**
1201:  */
1202: /**
1202:  */
1203: /**
1203:  */
1204: /**
1204:  */
1205: /**
1205:  */
1206: /**
1206:  */
1207: /**
1207:  */
1208: /**
1208:  */
1209: /**
1209:  */
1210: /**
1210:  */
1211: /**
1211:  */
1212: /**
1212:  */
1213: /**
1213:  */
1214: /**
1214:  */
1215: /**
1215:  */
1216: /**
1216:  */
1217: /**
1217:  */
1218: /**
1218:  */
1219: /**
1219:  */
1220: /**
1220:  */
1221: /**
1221:  */
1222: /**
1222:  */
1223: /**
1223:  */
1224: /**
1224:  */
1225: /**
1225:  */
1226: /**
1226:  */
1227: /**
1227:  */
1228: /**
1228:  */
1229: /**
1229:  */
1230: /**
1230:  */
1231: /**
1231:  */
1232: /**
1232:  */
1233: /**
1233:  */
1234: /**
1234:  */
1235: /**
1235:  */
1236: /**
1236:  */
1237: /**
1237:  */
1238: /**
1238:  */
1239: /**
1239:  */
1240: /**
1240:  */
1241: /**
1241:  */
1242: /**
1242:  */
1243: /**
1243:  */
1244: /**
1244:  */
1245: /**
1245:  */
1246: /**
1246:  */
1247: /**
1247:  */
1248: /**
1248:  */
1249: /**
1249:  */
1250: /**
1250:  */
1251: /**
1251:  */
1252: /**
1252:  */
1253: /**
1253:  */
1254: /**
1254:  */
1255: /**
1255:  */
1256: /**
1256:  */
1257: /**
1257:  */
1258: /**
1258:  */
1259: /**
1259:  */
1260: /**
1260:  */
1261: /**
1261:  */
1262: /**
1262:  */
1263: /**
1263:  */
1264: /**
1264:  */
1265: /**
1265:  */
1266: /**
1266:  */
1267: /**
1267:  */
1268: /**
1268:  */
1269: /**
1269:  */
1270: /**
1270:  */
1271: /**
1271:  */
1272: /**
1272:  */
1273: /**
1273:  */
1274: /**
1274:  */
1275: /**
1275:  */
1276: /**
1276:  */
1277: /**
1277:  */
1278: /**
1278:  */
1279: /**
1279:  */
1280: /**
1280:  */
1281: /**
1281:  */
1282: /**
1282:  */
1283: /**
1283:  */
1284: /**
1284:  */
1285: /**
1285:  */
1286: /**
1286:  */
1287: /**
1287:  */
1288: /**
1288:  */
1289: /**
1289:  */
1290: /**
1290:  */
1291: /**
1291:  */
1292: /**
1292:  */
1293: /**
1293:  */
1294: /**
1294:  */
1295: /**
1295:  */
1296: /**
1296:  */
1297: /**
1297:  */
1298: /**
1298:  */
1299: /**
1299:  */
1300: /**
1300:  */
1301: /**
1301:  */
1302: /**
1302:  */
1303: /**
1303:  */
1304: /**
1304:  */
1305: /**
1305:  */
1306: /**
1306:  */
1307: /**
1307:  */
1308: /**
1308:  */
1309: /**
1309:  */
1310: /**
1310:  */
1311: /**
1311:  */
1312: /**
1312:  */
1313: /**
1313:  */
1314: /**
1314:  */
1315: /**
1315:  */
1316: /**
1316:  */
1317: /**
1317:  */
1318: /**
1318:  */
1319: /**
1319:  */
1320: /**
1320:  */
1321: /**
1321:  */
1322: /**
1322:  */
1323: /**
1323:  */
1324: /**
1324:  */
1325: /**
1325:  */
1326: /**
1326:  */
1327: /**
1327:  */
1328: /**
1328:  */
1329: /**
1329:  */
1330: /**
1330:  */
1331: /**
1331:  */
1332: /**
1332:  */
1333: /**
1333:  */
1334: /**
1334:  */
1335: /**
1335:  */
1336: /**
1336:  */
1337: /**
1337:  */
1338: /**
1338:  */
1339: /**
1339:  */
1340: /**
1340:  */
1341: /**
1341:  */
1342: /**
1342:  */
1343: /**
1343:  */
1344: /**
1344:  */
1345: /**
1345:  */
1346: /**
1346:  */
1347: /**
1347:  */
1348: /**
1348:  */
1349: /**
1349:  */
1350: /**
1350:  */
1351: /**
1351:  */
1352: /**
1352:  */
1353: /**
1353:  */
1354: /**
1354:  */
1355: /**
1355:  */
1356: /**
1356:  */
1357: /**
1357:  */
1358: /**
1358:  */
1359: /**
1359:  */
1360: /**
1360:  */
1361: /**
1361:  */
1362: /**
1362:  */
1363: /**
1363:  */
1364: /**
1364:  */
1365: /**
1365:  */
1366: /**
1366:  */
1367: /**
1367:  */
1368: /**
1368:  */
1369: /**
1369:  */
1370: /**
1370:  */
1371: /**
1371:  */
1372: /**
1372:  */
1373: /**
1373:  */
1374: /**
1374:  */
1375: /**
1375:  */
1376: /**
1376:  */
1377: /**
1377:  */
1378: /**
1378:  */
1379: /**
1379:  */
1380: /**
1380:  */
1381: /**
1381:  */
1382: /**
1382:  */
1383: /**
1383:  */
1384: /**
1384:  */
1385: /**
1385:  */
1386: /**
1386:  */
1387: /**
1387:  */
1388: /**
1388:  */
1389: /**
1389:  */
1390: /**
1390:  */
1391: /**
1391:  */
1392: /**
1392:  */
1393: /**
1393:  */
1394: /**
1394:  */
1395: /**
1395:  */
1396: /**
1396:  */
1397: /**
1397:  */
1398: /**
1398:  */
1399: /**
1399:  */
1400: /**
1400:  */
1401: /**
1401:  */
1402: /**
1402:  */
1403: /**
1403:  */
1404: /**
1404:  */
1405: /**
1405:  */
1406: /**
1406:  */
1407: /**
1407:  */
1408: /**
1408:  */
1409: /**
1409:  */
1410: /**
1410:  */
1411: /**
1411:  */
1412: /**
1412:  */
1413: /**
1413:  */
1414: /**
1414:  */
1415: /**
1415:  */
1416: /**
1416:  */
1417: /**
1417:  */
1418: /**
1418:  */
1419: /**
1419:  */
1420: /**
1420:  */
1421: /**
1421:  */
1422: /**
1422:  */
1423: /**
1423:  */
1424: /**
1424:  */
1425: /**
1425:  */
1426: /**
1426:  */
1427: /**
1427:  */
1428: /**
1428:  */
1429: /**
1429:  */
1430: /**
1430:  */
1431: /**
1431:  */
1432: /**
1432:  */
1433: /**
1433:  */
1434: /**
1434:  */
1435: /**
1435:  */
1436: /**
1436:  */
1437: /**
1437:  */
1438: /**
1438:  */
1439: /**
1439:  */
1440: /**
1440:  */
1441: /**
1441:  */
1442: /**
1442:  */
1443: /**
1443:  */
1444: /**
1444:  */
1445: /**
1445:  */
1446: /**
1446:  */
1447: /**
1447:  */
1448: /**
1448:  */
1449: /**
1449:  */
1450: /**
1450:  */
1451: /**
1451:  */
1452: /**
1452:  */
1453: /**
1453:  */
1454: /**
1454:  */
1455: /**
1455:  */
1456: /**
1456:  */
1457: /**
1457:  */
1458: /**
1458:  */
1459: /**
1459:  */
1460: /**
1460:  */
1461: /**
1461:  */
1462: /**
1462:  */
1463: /**
1463:  */
1464: /**
1464:  */
1465: /**
1465:  */
1466: /**
1466:  */
1467: /**
1467:  */
1468: /**
1468:  */
1469: /**
1469:  */
1470: /**
1470:  */
1471: /**
1471:  */
1472: /**
1472:  */
1473: /**
1473:  */
1474: /**
14
```

```

505:     String name;
506:     String customerID;
507:     Object date;
508:
509:
510:
511:    // constructor ///////////////////////////////////////////////////////////////////
512:
513:    /**
514:     * Der Konstruktor legt ein neues Objekt der Klasse <CODE>MyLoggable</CODE>
515:     * an, wobei der Konstruktor von <CODE>Logable</CODE> aufgerufen wird,
516:     * und die &uuml;bergebenen Variablen zugewiesen werden. Diese sind
517:     * wichtig f&uuml;r die Erstellung des Log-Eintrags.
518:
519:    public MyLoggable(String name,
520:                      String customerID, Object date)
521:    {
522:        super();
523:        this.name = name;
524:        this.customerID = customerID;
525:        this.date = date;
526:
527:
528:        // public methods ///////////////////////////////////////////////////////////////////
529:
530:
531:        /**
532:         * Holt sich den Log-Eintrag, der aus den &uuml;bergebenen Daten
533:         * zusammengestellt wird. Dazu wird die extra implementierte
534:         * Klasse <CODE>MyLogEntry</CODE> genutzt.
535:
536:        public LogEntry getLogData()
537:        {
538:            return new MyLogEntry(name, customerID, date);
539:        }
540:    }
541:
542:
543:
544:    /**
545:     * &Uuml;berschreibt die Klasse <CODE>LogEntry</CODE> um einen
546:     * selbstdefinierten Log-Eintrag zu erm&ouml;glichen.
547:    */
548:    class MyLogEntry extends LogEntry
549:    {
550:
551:        // attributes ///////////////////////////////////////////////////////////////////
552:
553:        String name;
554:        String customerID;
555:        Object date;
556:
557:        // constructor ///////////////////////////////////////////////////////////////////
558:
559:        /**
560:         * Der Konstruktor legt ein neues Objekt der Klasse <CODE>MyLogEntry</CODE>
561:         * an, wobei der Konstruktor von <CODE>LogEntry</CODE> aufgerufen wird,
562:         * und die &uuml;bergebenen Variablen zugewiesen werden. Diese sind
563:         * wichtig f&uuml;r die Erstellung des Log-Eintrags.
564:
565:        public MyLogEntry(String name, String customerID, Object date)
566:        {
567:

```

1

SeeVideoStockProcess.java

```

1: import sale.*;
2: import data.*;
3: import data.oobjimpl.*;
4: import data.srdforms.*;
5: import data.swing.*;
6: import data.events.*;
7: import users.*;
8: 
9: import java.util.*;
10:
11: import javax.swing.*;
12:
13: /**
14:  * 
15:  * Prozess zur Bestandsanzeige und zum Editieren des Bestandes.
16:  */
17: public class SeeVideoStockProcess extends SaleProcess
18: {
19:     // attributes ///////////////////////////////
20: 
21:     // Gates
22:     protected UIGate selectionGate;
23: 
24:     // Transitions
25:     protected Transition newVideoTransition;
26:     protected Transition removeVideoTransition;
27:     protected Transition removeVideoCassette;
28: 
29:     // constructor ///////////////////////////////
30: 
31: /**
32:  * Erzeugt ein neues Objekt der Klasse <CODE>SeeVideoStockProcess</CODE>.
33:  */
34: 
35:     public SeeVideoStockProcess()
36:     {
37:         super ("SeeVideoProcess");
38:     }
39: 
40:     // protected methods ///////////////////////////////
41: 
42: /**
43:  * Baut die Oberfl&uuml;che f&uuml;r den Verleihvorgang auf.
44:  */
45:     protected void setupMachine()
46:     {
47:         // Auswahl-Gate anlegen
48:         selectionGate = new UIGate(null, null);
49: 
50:         // darzustellendes Formsheet erstellen
51:         final SingleTableFormSheet stfs =
52:             SingleTableFormSheet.create("See videos in stock and edit Process",
53:                                         (CountingStockImpl)Shop.getTheShop().getStock("Video-Countingstock"),
54:                                         selectionGate,
55:                                         getBasket(),
56:                                         true,
57:                                         new EditableVideoStockTED((CountingStockImpl)
58:                                         Shop.getTheShop()).getStock("Video-Countingstock"),
59:                                         (CountingStockImpl)Shop.getTheShop().getStock("Video-Countingstock"));
60: 
61:         // Transition zum Anlegen eines neuen Videos
62:         newVideoTransition = new Transition()
63:         {
64:             public Gate perform(SaleProcess owner, User usr)
65:             {
66:                 // Anlegen eines neuen Videos
67:                 // Eingabefelder erzeugen
68:                 JTextField jTextField1 = new JTextField();
69:                 JTextField jTextField2 = new JTextField();
70:                 JTextField jTextField3 = new JTextField();
71:                 JTextField jTextField4 = new JTextField();
72: 
73:                 // neues JPanel anlegen und vertikales Boxlayout setzen
74:                 JPanel jPanel1 = new JPanel();
75:                 jPanel1.setLayout(new BoxLayout(jPanel1,
76:                     BoxLayout.Y_AXIS));
77: 
78:                 // Label und Passwortfelder einfuegen
79:                 jTextField1.add(new JLabel("Name"));
80:                 jTextField1.add(jTextField1);
81:                 jTextField1.add(new JLabel("Buy (in Pf)"));
82:                 jTextField1.add(jTextField2);
83:                 jTextField1.add(new JLabel("Sell (in Pf)"));
84:                 jTextField1.add(jTextField3);
85:                 jTextField1.add(new JLabel("Amount"));
86:                 jTextField1.add(jTextField4);
87: 
88:                 // Dialog darstellen
89:                 JOptionPane.showMessageDialog (null,
90:                     "New Video-Cassette",
91:                     JOptionPane.QUESTION_MESSAGE);
92: 
93:                 // New Video-Message
94: 
95:                 // Video-Katalog holen
96:                 Catalog videoCatalog =
97:                     Shop.getTheShop().getCatalog("Video-Catalog");
98: 
99:                 // Video-Bestand holen
100:                CountingStock videoCountingStock =
101:                    (CountingStock)Shop.getTheShop().getStock("Video-Countingstock");
102: 
103:                // Name des Videos
104:                String name = jTextField1.getText();
105: 
106:                // Verkaufspreis
107:                int buy = 0;
108:                try {
109:                    buy = (new Integer(jTextField2.getText())).intValue();
110:                } catch (NumberFormatException ne) {}
111: 
112:                // Einkaufspreis
113:                int sell = 0;
114:                try {
115:                    sell = (new Integer(jTextField3.getText())).intValue();
116:                } catch (NumberFormatException ne) {}
117: 
118:                // Anzahl
119:                int amount = 0;
120:                try {
121:                    amount = (new Integer(jTextField4.getText())).intValue();
122:                } catch (NumberFormatException ne) {}
123: 
124:            }
125:        }
126:    }

```

SeeVideoStockProcess.java

```

127:    // ueberpruefen, ob Eingabe korrekt          190:   // db.commit() ;
128:    if (sell > 0 && buy > 0 && amount > 0 && !name.equals("") ) {      191:   catch (VetoException ve) {
129:      // in Katalog uebernehmen           192:     catch (VetoException ve) {
try {                                         193:       ve.printStackTrace();
130:       videoCatalog.add(new VideoCassette(name, new QuoteValue
131:         (new IntegerValue (buy),
132:          new IntegerValue (sell))), null);
133:     }
134:   }
135:   // in Bestand uebernehmen           194:   JOptionPane.showMessageDialog(null,
136:     videoCountingStock.add(name, amount, null);           195:     "The selected item can't be removed!" );
137:   }
138:   // Element existiert bereits        196:   }
139:   catch (DuplicateKeyException dke) {           197:   }
140:     JOptionPane.showMessageDialog(null,           198:   // ausgeliehen Videos vorhanden
141:       "Element exists");
142:   }
143:   return selectionGate;
144: }
145: }
146: // Eingabe fehlerhaft           199:   // ausgeliehen Videos vorhanden
147: else {                           200:   else {
148:   JOptionPane.showMessageDialog(null,           201:     JOptionPane.showMessageDialog(null,
149:     "The given input was not correct!");
150:     "There are still rented videos!");
151:   }
152:   // Auswahl-Gate zurueckgeben
153:   // Auswahl-Gate zurueckgeben
154:   return selectionDate;
155: }
156: }
157: // Transition zum Loeschchen anlegen
158: // removeVideoTransition = new Transition()
159: removeVideoTransition = new Transition();
160: public Gate perform(SaleProcess pOwner, User usr)
161: {
162:   // markierte Zeile holen
163:   Object record = stfs.getSelectedRecord();
164:   VideoCassette videoCassette =
165:     (VideoCassette)(data.swing.CountingStockTableModel.Record)
166:     record).getDescriptor();
167:   // Anzahl der entliehenen Videos bestimmen
168:   int rented = 0;
169:   try {
170:     Iterator i = VideoMachine.getAllCustomer().iterator();
171:     while (i.hasNext()) {
172:       if (Customer i.next().getStorageStock().countItems(
173:         ((Customer i.next()).getStorageStock(), null),
174:         videoCassette.getName(), null));
175:       rented = rented +
176:         ((Customer i.next()).getStorageStock().countItems(
177:           videoCassette.getName(), null));
178:     }
179:   catch (NullPointerException npe) {
180:   }
181:   // ueberpruefen, ob gewaehltes Video ausgeliehen
182:   if (rented <= 0) {
183:     // loeschen des gewaehlten Eintrags
184:     try {
185:       // DataBasket db = new DataBasketImpl();
186:       Shop.getTheShop().getCatalog("Video-catalog").remove(
187:         videoCassette, null);
188:     }
189:   }
190:   // FormSheetContentCreator am Auswahl-Gate anmelden
191:   stfs.addContentCreator(new FormSheetContentCreator()
192:   {
193:     protected void createFormSheetContent(FormSheet fs)
194:     {
195:       // alle vorhandenen Buttons entfernen
196:       fs.removeAllButtons();
197:     }
198:   });
199:   // FormSheetContentCreator am Auswahl-Gate anmelden
200:   stfs.addContentCreator(new FormSheetContentCreator()
201:   {
202:     protected void createFormSheetContent(FormSheet fs)
203:     {
204:       // alle vorhandenen Buttons entfernen
205:       fs.removeAllButtons();
206:     }
207:   });
208:   // FormSheetContentCreator am Auswahl-Gate anmelden
209:   stfs.addContentCreator(new FormSheetContentCreator()
210:   {
211:     protected void createFormSheetContent(FormSheet fs)
212:     {
213:       // alle vorhandenen Buttons entfernen
214:       fs.removeAllButtons();
215:     }
216:   });
217:   // neuen "New"-Button einbauen
218:   fs.addButton ("New", 100, new sale.Action()
219:   {
220:     public void doAction (SaleProcess p, SalesPoint sp)
221:     {
222:       // Transition zum Erstellen eines neuen Videos als
223:       // naechste Transition setzen
224:       selectionDate.setNextTransition(newVideoTransition);
225:     }
226:   });
227: }
228: // neuen "Remove"-Button einbauen
229: fs.addButton ("Remove", 101, new sale.Action()
230: {
231:   public void doAction (SaleProcess p, SalesPoint sp)
232:   {
233:     // Transition zum Loeschen eines Videos als
234:     // naechste Transition setzen
235:     selectionDate.setNextTransition(removeVideoTransition);
236:   }
237: }
238: }
239: // neuen "Ok"-Button einbauen
240: fs.addButton ("Ok", 102, new sale.Action()
241: {
242:   public void doAction (SaleProcess p, SalesPoint sp)
243:   {
244:     // Transition zum Commit-Gate als naechste Transition setzen
245:     selectionDate.setNextTransition(GateChangeTransition.CHANGE_TO_COMMIT_GATE);
246:   }
247: }
248: }
249: }
250: // neuen "Cancel"-Button einbauen
251: fs.addButton ("Cancel", 103, new sale.Action()
252: 
```

SeeVideoStockProcess.java

```
253: {  
254:     public void doAction (SaleProcess p, SalesPoint sp)  
255:     {  
256:         // Transition zum Rollback-Gate als naechste Transition setzen  
257:         selectionGate.setNextTransition(  
258:             GateChangeTransition.CHANGE_TO_ROLLBACK_GATE);  
259:     }  
260: }  
261:  
262: }  
263: }  
264: }  
265:  
266: ///////////////////////////////////////////////////////////////////  
267:  
268: /**  
269: * Gibt das Startgate des Prozesses zur&uuml;ck.  
270: */  
271: public Gate getInitialGate()  
272: {  
273:     // Automat aufbauen  
274:     setupMachine();  
275:  
276:     // ... und Auswahl-Gate als Startgate zurueckgeben  
277:     return selectionGate;  
278: }  
279:  
280: /**  
281: * &Uuml;beribt das Log-Gate. Hier das Stop-Gate, da beim Beenden des  
282: * Prozesses kein Log-Eintrag geschrieben werden soll.  
283: */  
284: public Gate getLogGate()  
285: {  
286:     return getStopGate();  
287: }  
288:  
289:  
290: }
```

```
1: import data.ooimpl.*;
2: import data.*;
3:
4:
5: public class VideoCassette extends CatalogItemImpl
6: {
7:
8:     // constructor ///////////////////////////////
9:
10:    /**
11:     * Der Konstruktor reicht die an ihn &uuml;bergebenen Parameter an
12:     * den Konstruktor der Klasse <CODE>CatalogItemImpl</CODE> weiter.
13:     */
14:    public VideoCassette (String name, QuoteValue value)
15:    {
16:        super (name, value);
17:    }
18:
19:    // public methods ///////////////////////////////
20:
21:    /**
22:     * Setzt den Wert der speziellen Videokassette.
23:     */
24:    public void setValue (QuoteValue value)
25:    {
26:        super.setValue (value);
27:    }
28:
29:
30:    // protected methods ///////////////////////////////
31:
32:    /**
33:     * Erzeugt eine Kopie des Objektes der Klasse <CODE>VideoCassette</CODE>
34:     * mit gleichem Namen und gleichem Wert.
35:     */
36:    protected CatalogItemImpl getShallowClone()
37:    {
38:        return new VideoCassette (new String (getName()),
39:                               (QuoteValue)getValue().clone());
40:    }
41:
42:
43: }
```

```

1: import sale.*;
2: import log.*;
3: import users.*;
4: import data.*;
5: import data.coimpl.*;
6: import java.io.*;
7: import java.util.*;
8: import java.lang.*;
9: import javax.swing.*;
10: import javax.swing.*;
11: import javax.swing.*;
12: /**
13:  * Diese Klasse ist der "Shop" der Anwendung. Sie stellt das Grundger&uuml;st
14:  * f&uuml;r die Anwendung dar.
15:  */
16: public class VideoMachine extends Shop
17: {
18:     public VideoMachine()
19:     {
20:         /**
21:          * attributes
22:          * Liste aller registrierten Kunden
23:          * private static Set<Customer> customerSet = new HashSet();
24:          * constructor
25:          */
26:         /**
27:          * Konstruktor. Erzeugt ein neues Objekt vom Typ VideoAutomat.
28:          */
29:         public VideoMachine()
30:         {
31:             super();
32:             /**
33:              * Name und Preis fuer die Standard-Katalogeintrage
34:              * String[] videos = {"Video 01", "Video 02", "Video 03", "Video 04",
35:              * "Video 05", "Video 06", "Video 07", "Video 08",
36:              * "Video 09", "Video 10"};
37:             int[] buy = {5000, 5000, 5000, 4000, 4000, 3000, 3000};
38:             int[] sell = {4000, 4000, 4000, 3500, 3500, 3500, 3000};
39:             /**
40:                * Eintragung in den Catalog und den Stock
41:                * Catalog videoCatalog = new CatalogImpl("Video-catalog");
42:                * addCatalog(videoCatalog);
43:                * videoCatalog.add(new VideoCassette(videos[i], new QuoteValue
44:                * (new IntegerValue(buy[i]), new IntegerValue(sell[i])), null));
45:                * for (int i = 0; i < videos.length; i++)
46:                * videoCatalog.add(new VideoCassette(videos[i], new QuoteValue
47:                * (new IntegerValue(buy[i]), new IntegerValue(sell[i])), null));
48:            }
49:            /**
50:              * CountingStock cs = new CountingStockImpl("Video-Countingstock",
51:              * (CatalogImpl)videoCatalog);
52:              * addStock(cs);
53:              */
54:              /**
55:                * Erstellen eines Bestands von je 5 Videos (sollte spaeter
56:                * vom Manager uebernommen werden)
57:                * Iterator<Cassettes> cassettes = videoCatalog.keySet(null).iterator();
58:                * while (cassettes.hasNext())
59:                * cs.add((String)cassettes.next(), 5, null);
60:            }
61:            /**
62:              * Anlegen einer Waehrung und Erstellen eines dazugehoerigen
63:              * Geldbestandes
64:              */
65:              /**
66:                * MoneyBag coinSlot = new MoneyBagImpl("coin slot",
67:                * (CurrencyImpl)getCatalog("DM"));
68:                * coinSlot.add("1-Pfennig-St&uuml;ck", 100000, null);
69:                * addStock(coinSlot);
70:            }
71:            /**
72:              * public methods
73:            */
74:            /**
75:              * Die Main-Methode startet die Anwendung.
76:              */
77:              /**
78:                * public static void main (String[] args)
79:                {
80:                    // neuen Videoautomaten erzeugen
81:                    VideoMachine vidMachine = new VideoMachine();
82:                    serTheShop(vidMachine);
83:                    /**
84:                      * Log undefinieren...
85:                      * Log.setLogCreator(new MyLogCreator());
86:                      */
87:                      /**
88:                        * Datei zuweisen, in die das Log geschrieben werden soll
89:                        */
90:                        try {
91:                            catch (IOException e) {
92:                                System.err.println("Unable to create log file.");
93:                            }
94:                            /**
95:                              * Verleih bzw. Kasse anlegen
96:                              * Counter c = new Counter("Video Rental");
97:                              * c.attach(new DataBasketImpl());
98:                              * c.attach(new User("SalespointUser"));
99:                              * vidMachine.addSalesPoint(c);
100:                             /**
101:                               * Anlegen des Manager-Passwords
102:                               * JPanel jPasswordField1 = new JPasswordField();
103:                               * JPanel jPasswordField2 = new JPasswordField();
104:                               */
105:                               /**
106:                                 * neues JPanel anlegen und vertikales Boxlayout setzen
107:                                 * JPanel jPasswordFieldRequest.setLayout(new BoxLayout(jPasswordField1,
108:                                 * jPasswordFieldRequest, new JPanel()));
109:                                 */
110:                                 /**
111:                                   * Label und Passwortfelder einfügen
112:                                   * JLabel jLabel = new JLabel("Enter the manager's password:");
113:                                   * jPasswordFieldRequest.add(jLabel);
114:                                   * jPasswordFieldRequest.add(new JLabel("Re-enter the manager's password:"));
115:                                   */
116:                                   /**
117:                                     * Passwortabfrage wiederholen bis Uebereinstimmung zwischen den Worten
118:                                     */
119:                                     do {
120:                                         /**
121:                                           * ManagerpasswordDialog (null,
122:                                           * "Manager",
123:                                           * JOptionPane.showMessageDialog(null,
124:                                           * jPasswordFieldRequest,
125:                                           * "Manager");
126:                                           */
127:                                         }
128:                                         while (!new String(jPasswordField1.getPassword()).equals(
129:                                         new String(jPasswordField2.getPassword())));
130:                                         */
131:                                         /**
132:                                           * Managerpasswort setzen
133:                                           */
134:                                         */
135:                                         */
136:                                         */
137:                                         */
138:                                         */
139:                                         */
140:                                         */
141:                                         */
142:                                         */
143:                                         */
144:                                         */
145:                                         */
146:                                         */
147:                                         */
148:                                         */
149:                                         */
150:                                         */
151:                                         */
152:                                         */
153:                                         */
154:                                         */
155:                                         */
156:                                         */
157:                                         */
158:                                         */
159:                                         */
160:                                         */
161:                                         */
162:                                         */
163:                                         */
164:                                         */
165:                                         */
166:                                         */
167:                                         */
168:                                         */
169:                                         */
170:                                         */
171:                                         */
172:                                         */
173:                                         */
174:                                         */
175:                                         */
176:                                         */
177:                                         */
178:                                         */
179:                                         */
180:                                         */
181:                                         */
182:                                         */
183:                                         */
184:                                         */
185:                                         */
186:                                         */
187:                                         */
188:                                         */
189:                                         */
190:                                         */
191:                                         */
192:                                         */
193:                                         */
194:                                         */
195:                                         */
196:                                         */
197:                                         */
198:                                         */
199:                                         */
200:                                         */
201:                                         */
202:                                         */
203:                                         */
204:                                         */
205:                                         */
206:                                         */
207:                                         */
208:                                         */
209:                                         */
210:                                         */
211:                                         */
212:                                         */
213:                                         */
214:                                         */
215:                                         */
216:                                         */
217:                                         */
218:                                         */
219:                                         */
220:                                         */
221:                                         */
222:                                         */
223:                                         */
224:                                         */
225:                                         */
226:                                         */
227:                                         */
228:                                         */
229:                                         */
230:                                         */
231:                                         */
232:                                         */
233:                                         */
234:                                         */
235:                                         */
236:                                         */
237:                                         */
238:                                         */
239:                                         */
240:                                         */
241:                                         */
242:                                         */
243:                                         */
244:                                         */
245:                                         */
246:                                         */
247:                                         */
248:                                         */
249:                                         */
250:                                         */
251:                                         */
252:                                         */
253:                                         */
254:                                         */
255:                                         */
256:                                         */
257:                                         */
258:                                         */
259:                                         */
260:                                         */
261:                                         */
262:                                         */
263:                                         */
264:                                         */
265:                                         */
266:                                         */
267:                                         */
268:                                         */
269:                                         */
270:                                         */
271:                                         */
272:                                         */
273:                                         */
274:                                         */
275:                                         */
276:                                         */
277:                                         */
278:                                         */
279:                                         */
280:                                         */
281:                                         */
282:                                         */
283:                                         */
284:                                         */
285:                                         */
286:                                         */
287:                                         */
288:                                         */
289:                                         */
290:                                         */
291:                                         */
292:                                         */
293:                                         */
294:                                         */
295:                                         */
296:                                         */
297:                                         */
298:                                         */
299:                                         */
200:                                         */
201:                                         */
202:                                         */
203:                                         */
204:                                         */
205:                                         */
206:                                         */
207:                                         */
208:                                         */
209:                                         */
210:                                         */
211:                                         */
212:                                         */
213:                                         */
214:                                         */
215:                                         */
216:                                         */
217:                                         */
218:                                         */
219:                                         */
220:                                         */
221:                                         */
222:                                         */
223:                                         */
224:                                         */
225:                                         */
226:                                         */
227:                                         */
228:                                         */
229:                                         */
230:                                         */
231:                                         */
232:                                         */
233:                                         */
234:                                         */
235:                                         */
236:                                         */
237:                                         */
238:                                         */
239:                                         */
240:                                         */
241:                                         */
242:                                         */
243:                                         */
244:                                         */
245:                                         */
246:                                         */
247:                                         */
248:                                         */
249:                                         */
250:                                         */
251:                                         */
252:                                         */
253:                                         */
254:                                         */
255:                                         */
256:                                         */
257:                                         */
258:                                         */
259:                                         */
260:                                         */
261:                                         */
262:                                         */
263:                                         */
264:                                         */
265:                                         */
266:                                         */
267:                                         */
268:                                         */
269:                                         */
270:                                         */
271:                                         */
272:                                         */
273:                                         */
274:                                         */
275:                                         */
276:                                         */
277:                                         */
278:                                         */
279:                                         */
280:                                         */
281:                                         */
282:                                         */
283:                                         */
284:                                         */
285:                                         */
286:                                         */
287:                                         */
288:                                         */
289:                                         */
290:                                         */
291:                                         */
292:                                         */
293:                                         */
294:                                         */
295:                                         */
296:                                         */
297:                                         */
298:                                         */
299:                                         */
200:                                         */
201:                                         */
202:                                         */
203:                                         */
204:                                         */
205:                                         */
206:                                         */
207:                                         */
208:                                         */
209:                                         */
2010:                                         */
2011:                                         */
2012:                                         */
2013:                                         */
2014:                                         */
2015:                                         */
2016:                                         */
2017:                                         */
2018:                                         */
2019:                                         */
2020:                                         */
2021:                                         */
2022:                                         */
2023:                                         */
2024:                                         */
2025:                                         */
2026:                                         */
2027:                                         */
2028:                                         */
2029:                                         */
2030:                                         */
2031:                                         */
2032:                                         */
2033:                                         */
2034:                                         */
2035:                                         */
2036:                                         */
2037:                                         */
2038:                                         */
2039:                                         */
2040:                                         */
2041:                                         */
2042:                                         */
2043:                                         */
2044:                                         */
2045:                                         */
2046:                                         */
2047:                                         */
2048:                                         */
2049:                                         */
2050:                                         */
2051:                                         */
2052:                                         */
2053:                                         */
2054:                                         */
2055:                                         */
2056:                                         */
2057:                                         */
2058:                                         */
2059:                                         */
2060:                                         */
2061:                                         */
2062:                                         */
2063:                                         */
2064:                                         */
2065:                                         */
2066:                                         */
2067:                                         */
2068:                                         */
2069:                                         */
2070:                                         */
2071:                                         */
2072:                                         */
2073:                                         */
2074:                                         */
2075:                                         */
2076:                                         */
2077:                                         */
2078:                                         */
2079:                                         */
2080:                                         */
2081:                                         */
2082:                                         */
2083:                                         */
2084:                                         */
2085:                                         */
2086:                                         */
2087:                                         */
2088:                                         */
2089:                                         */
2090:                                         */
2091:                                         */
2092:                                         */
2093:                                         */
2094:                                         */
2095:                                         */
2096:                                         */
2097:                                         */
2098:                                         */
2099:                                         */
2010:                                         */
2011:                                         */
2012:                                         */
2013:                                         */
2014:                                         */
2015:                                         */
2016:                                         */
2017:                                         */
2018:                                         */
2019:                                         */
2020:                                         */
2021:                                         */
2022:                                         */
2023:                                         */
2024:                                         */
2025:                                         */
2026:                                         */
2027:                                         */
2028:                                         */
2029:                                         */
20210:                                         */
20211:                                         */
20212:                                         */
20213:                                         */
20214:                                         */
20215:                                         */
20216:                                         */
20217:                                         */
20218:                                         */
20219:                                         */
20220:                                         */
20221:                                         */
20222:                                         */
20223:                                         */
20224:                                         */
20225:                                         */
20226:                                         */
20227:                                         */
20228:                                         */
20229:                                         */
20230:                                         */
20231:                                         */
20232:                                         */
20233:                                         */
20234:                                         */
20235:                                         */
20236:                                         */
20237:                                         */
20238:                                         */
20239:                                         */
20240:                                         */
20241:                                         */
20242:                                         */
20243:                                         */
20244:                                         */
20245:                                         */
20246:                                         */
20247:                                         */
20248:                                         */
20249:                                         */
20250:                                         */
20251:                                         */
20252:                                         */
20253:                                         */
20254:                                         */
20255:                                         */
20256:                                         */
20257:                                         */
20258:                                         */
20259:                                         */
20260:                                         */
20261:                                         */
20262:                                         */
20263:                                         */
20264:                                         */
20265:                                         */
20266:                                         */
20267:                                         */
20268:                                         */
20269:                                         */
20270:                                         */
20271:                                         */
20272:                                         */
20273:                                         */
20274:                                         */
20275:                                         */
20276:                                         */
20277:                                         */
20278:                                         */
20279:                                         */
20280:                                         */
20281:                                         */
20282:                                         */
20283:                                         */
20284:                                         */
20285:                                         */
20286:                                         */
20287:                                         */
20288:                                         */
20289:                                         */
20290:                                         */
20291:                                         */
20292:                                         */
20293:                                         */
20294:                                         */
20295:                                         */
20296:                                         */
20297:                                         */
20298:                                         */
20299:                                         */
20300:                                         */
20301:                                         */
20302:                                         */
20303:                                         */
20304:                                         */
20305:                                         */
20306:                                         */
20307:                                         */
20308:                                         */
20309:                                         */
20310:                                         */
20311:                                         */
20312:                                         */
20313:                                         */
20314:                                         */
20315:                                         */
20316:                                         */
20317:                                         */
20318:                                         */
20319:                                         */
20320:                                         */
20321:                                         */
20322:                                         */
20323:                                         */
20324:                                         */
20325:                                         */
20326:                                         */
20327:                                         */
20328:                                         */
20329:                                         */
20330:                                         */
20331:                                         */
20332:                                         */
20333:                                         */
20334:                                         */
20335:                                         */
20336:                                         */
20337:                                         */
20338:                                         */
20339:                                         */
20340:                                         */
20341:                                         */
20342:                                         */
20343:                                         */
20344:                                         */
20345:                                         */
20346:                                         */
20347:                                         */
20348:                                         */
20349:                                         */
20350:                                         */
20351:                                         */
20352:                                         */
20353:                                         */
20354:                                         */
20355:                                         */
20356:                                         */
20357:                                         */
20358:                                         */
20359:                                         */
20360:                                         */
20361:                                         */
20362:                                         */
20363:                                         */
20364:                                         */
20365:                                         */
20366:                                         */
20367:                                         */
20368:                                         */
20369:                                         */
20370:                                         */
20371:                                         */
20372:                                         */
20373:                                         */
20374:                                         */
20375:                                         */
20376:                                         */
20377:                                         */
20378:                                         */
20379:                                         */
20380:                                         */
20381:                                         */
20382:                                         */
20383:                                         */
20384:                                         */
20385:                                         */
20386:                                         */
20387:                                         */
20388:                                         */
20389:                                         */
20390:                                         */
20391:                                         */
20392:                                         */
20393:                                         */
20394:                                         */
20395:                                         */
20396:                                         */
20397:                                         */
20398:                                         */
20399:                                         */
20400:                                         */
20401:                                         */
20402:                                         */
20403:                                         */
20404:                                         */
20405:                                         */
20406:                                         */
20407:                                         */
20408:                                         */
20409:                                         */
20410:                                         */
20411:                                         */
20412:                                         */
20413:                                         */
20414:                                         */
20415:                                         */
20416:                                         */
20417:                                         */
20418:                                         */
20419:                                         */
20420:                                         */
20421:                                         */
20422:                                         */
20423:                                         */
20424:                                         */
20425:                                         */
20426:                                         */
20427:                                         */
20428:                                         */
20429:                                         */
20430:                                         */
20431:                                         */
20432:                                         */
20433:                                         */
20434:                                         */
20435:                                         */
20436:                                         */
20437:                                         */
20438:                                         */
20439:                                         */
20440:                                         */
20441:                                         */
20442:                                         */
20443:                                         */
20444:                                         */
20445:                                         */
20446:                                         */
20447:                                         */
20448:                                         */
20449:                                         */
20450:                                         */
20451:                                         */
20452:                                         */
20453:                                         */
20454:                                         */
20455:                                         */
20456:                                         */
20457:                                         */
20458:                                         */
20459:                                         */
20460:                                         */
20461:                                         */
20462:                                         */
20463:                                         */
20464:                                         */
20465:                                         */
20466:                                         */
20467:                                         */
20468:                                         */
20469:                                         */
20470:                                         */
20471:                                         */
20472:                                         */
20473:                                         */
20474:                                         */
20475:                                         */
20476:                                         */
20477:                                         */
20478:                                         */
20479:                                         */
20480:                                         */
20481:                                         */
20482:                                         */
20483:                                         */
20484:                                         */
20485:                                         */
20486:                                         */
20487:                                         */
20488:                                         */
20489:                                         */
20490:                                         */
20491:                                         */
20492:                                         */
20493:                                         */
20494:                                         */
20495:                                         */
20496:                                         */
20497:                                         */
20498:                                         */
20499:                                         */
20500:                                         */
20501:                                         */
20502:                                         */
20503:                                         */
20504:                                         */
20505:                                         */
20506:                                         */
20507:                                         */
20508:                                         */
20509:                                         */
20510:                                         */
20511:                                         */
20512:                                         */
20513:                                         */
20514:                                         */
20515:                                         */
20516:                                         */
20517:                                         */
20518:                                         */
20519:                                         */
20520:                                         */
20521:                                         */
20522:                                         */
20523:                                         */
20524:                                         */
20525:                                         */
20526:                                         */
20527:                                         */
20528:                                         */
20529:                                         */
20530:                                         */
20531:                                         */
20532:                                         */
20533:                                         */
20534:                                         */
20535:                                         */
20536:                                         */
20537:                                         */
20538:                                         */
20539:                                         */
20540:                                         */
20541:                                         */
20542:                                         */
20543:                                         */
20544:                                         */
20545:                                         */
20546:                                         */
20547:                                         */
20548:                                         */
20549:                                         */
20550:                                         */
20551:                                         */
20552:                                         */
20553:                                         */
20554:                                         */
20555:                                         */
20556:                                         */
20557:                                         */
20558:                                         */
20559:                                         */
20560:                                         */
20561:                                         */
20562:                                         */
20563:                                         */
20564:                                         */
20565:                                         */
20566:                                         */
20567:                                         */
20568:                                         */
20569:                                         */
20570:                                         */
20571:                                         */
20572:                                         */
20573:                                         */
20574:                                         */
20575:                                         */
20576:                                         */
20577:                                         */
20578:                                         */
20579:                                         */
20580:                                         */
20581:                                         */
20582:                                         */
20583:                                         */
20584:                                         */
20585:                                         */
20586:                                         */
20587:                                         */
20588:                                         */
20589:                                         */
20590:                                         */
20591:                                         */
20592:                                         */
20593:                                         */
20594:                                         */
20595:                                         */
20596:                                         */
20597:                                         */
20598:                                         */
20599:                                         */
20600:                                         */
20601:                                         */
20602:                                         */
20603:                                         */
20604:                                         */
20605:                                         */
20606:                                         */
20607:                                         */
20608:                                         */
20609:                                         */
20610:                                         */
20611:                                         */
20612:                                         */
20613:                                         */
20614:                                         */
20615:                                         */
20616:                                         */
20617:                                         */
20618:                                         */
20619:                                         */
20620:                                         */
20621:                                         */
20622:                                         */
20623:                                         */
20624:                                         */
20625:                                         */
20626:                                         */
20627:                                         */
20628:                                         */
20629:                                         */
20630:                                         */
20631:                                         */
20632:                                         */
20633:                                         */
20634:                                         */
20635:                                         */
20636:                                         */
20637:                                         */
20638:                                         */
20639:                                         */
20640:                                         */
20641:                                         */
20642:                                         */
20643:                                         */
20644:                                         */
20645:                                         */
20646:                                         */
20647:                                         */
20648:                                         */
20649:                                         */
20650:                                         */
20651:                                         */
20652:                                         */
20653:                                         */
20654:                                         */
20655:                                         */
20656:                                         */
20657:                                         */
20658:                                         */
20659:                                         */
20660:                                         */
20661:                                         */
20662:                                         */
20663:                                         */
20664:                                         */
20665:                                         */
20666:                                         */
20667:                                         */
20668:                                         */
20669:                                         */
20670:                                         */
20671:                                         */
20672:                                         */
20673:                                         */
20674:                                         */
20675:                                         */
20676:                                         */
20677:                                         */
20678:                                         */
20679:                                         */
20680:                                         */
20681:                                         */
20682:                                         */
20683:                                         */
20684:                                         */
20685:                                         */
20686:                                         */
20687:                                         */
20688:                                         */
20689:                                         */
20690:                                         */
20691:                                         */
20692:                                         */
20693:                                         */
20694:                                         */
20695:                                         */
20696:                                         */
20697:                                         */
20698:                                         */
20699:                                         */
20700:                                         */
20701:                                         */
20702:                                         */
20703:                                         */
20704:                                         */
20705:                                         */
20706:                                         */
20707:                                         */
20708:                                         */
20709:                                         */
20710:                                         */
20711:                                         */
20712:                                         */
20713:                                         */
20714:                                         */
20715:                                         */
20716:                                         */
20717:                                         */
20718:                                         */
20719:                                         */
20720:                                         */
20721:                                         */
20722:                                         */
20723:                                         */
20724:                                         */
20725:                                         */
20726:                                         */
20727:                                         */
20728:                                         */
20729:                                         */
20730:                                         */
20731:                                         */
20732:                                         */
20733:                                         */
20734:                                         */
20735:                                         */
20736:                                         */
20737:                                         */
20738:                                         */
20739:                                         */
20740:                                         */
20741:                                         */
20742:                                         */
20743:                                         */
20744:                                         */
20745:                                         */
20746:                                         */
20747:                                         */
20748:                                         */
20749:                                         */
20750:                                         */
20751:                                         */
20752:                                         */
20753:                                         */
20754:                                         */
20755:                                         */
20756:                                         */
20757:                                         */
20758:                                         */
20759:                                         */
20760:                                         */
20761:                                         */
20762:                                         */
20763:                                         */
20764:                                         */
20765:                                         */
20766:                                         */
20767:                                         */
20768:                                         */
20769:                                         */
20770:                                         */
20771:                                         */
20772:                                         */
20773:                                         */
20774:                                         */
20775:                                         */
20776:                                         */
20777:                                         */
20778:                                         */
20779:                                         */
20780:                                         */
20781:                                         */
20782:                                         */
20783:                                         */
20784:                                         */
20785:                                         */
20786:                                         */
20787:                                         */
20788:                                         */
20789:                                         */
20790:                                         */
20791:                                         */
20792:                                         */
20793:                                         */
20794:                                         */
20795:                                         */
20796:                                         */
20797:                                         */
20798:                                         */
20799:                                         */
20800:                                         */
20801:                                         */
20802:                                         */
20803:                                         */
20804:                                         */
20805:                                         */
20806:                                         */
20807:                                         */
20808:                                         */
20809:                                         */
20810:                                         */
20811:                                         */
20812:                                         */
20813:                                         */
20814:                                         */
20815:                                         */
20816:                                         */
20817:                                         */
20818:                                         */
20819:                                         */
20820:                                         */
20821:                                         */
20822:                                         */
20823:                                         */
20824:                                         */
20825:                                         */
20826:                                         */
20827:                                         */
20828:                                         */
20829:                                         */
20830:                                         */
20831:                                         */
20832:                                         */
20833:                                         */
20834:                                         */
20835:                                         */
20836:                                         */
20837:                                         */
20838:                                         */
20839:                                         */
20840:                                         */
20841:                                         */
20842:                                         */
20843:                                         */
20844:                                         */
20845:                                         */
20846:                                         */
20847:                                         */
20848:                                         */
20849:                                         */
20850:                                         */
20851:                                         */
20852:                                         */
20853:                                         */
20854:                                         */
20855:                                         */
20856:                                         */
20857:                                         */
20858:                                         */
20859:                                         */
20860:                                         */
20861:                                         */
20862:                                         */
20863:                                         */
20864:                                         */
20865:                                         */
20866:                                         */
20867:                                         */
20868:                                         */
20869:                                         */
20870:                                         */
20871:                                         */
20872:                                         */
20873:                                         */
20874:                                         */
20875:                                         */
20876:                                         */
20877:                                         */
20878:                                         */
20879:                                         */
20880:                                         */
20881:                                         */
20882:                                         */
20883:                                         */
20884:                                         */
20885:                                         */
20886:                                         */
20887:                                         */
20888:                                         */
20889:                                         */
20890:                                         */
20891:                                         */
20892:                                         */
20893:                                         */
20894:                                         */
20895:                                         */
20896:                                         */
20897:                                         */
20898:                                         */
20899:                                         */
20900:                                         */
20901:                                         */
20902:                                         */
20903:                                         */
20904:                                         */
20905:                                         */
20906:                                         */
20907:                                         */
20908:                                         */
20909:                                         */
20910:                                         */
20911:                                         */
20912:                                         */
20913:                                         */
20914:                                         */
20915:                                         */
20916:                                         */
20917:                                         */
20918:                                         */
20919:                                         */
20920:                                         */
20921:                                         */
20922:                                         */
20923:                                         */
20924:                                         */
20925:                                         */
20926:                                         */
20927:                                         */
20928:                                         */
20929:                                         */
20930:                                         */
20931:                                         */
20932:                                         */
20933:                                         */
20934:                                         */
20935:                                         */
20936:                                         */
20937:                                         */
20938:                                         */
20939:                                         */
20940:                                         */
20941:                                         */
20942:                                         */
20943:                                         */
20944:                                         */
20945:                                         */
20946:                                         */
20947:                                         */
20948:                                         */
20949:                                         */
20950:                                         */
20951:                                         */
20952:                                         */
20953:                                         */
20954:                                         */
20955:                                         */
20956:                                         */
20957:                                         */
20958:                                         */
20959:                                         */
20960:                                         */
20961:                                         */
20962:                                         */
20963:                                         */
20964:                                         */
20965:                                         */
20966:                                         */
20967:                                         */
20968:                                         */
20969:                                         */
20970:                                         */
20971:                                         */
20972:                                         */
20973:                                         */
20974:                                         */
20975:                                         */
20976:                                         */
20977:                                         */
20978:                                         */
20979:                                         */
20980:                                         */
20981:                                         */
20982:                                         */
20983:                                         */
20984:                                         */
20985:                                         */
20986:                                         */
20987:                                         */
20988:                                         */
20989:                                         */
20990:                                         */
20991:                                         */
20992:                                         */
20993:                                         */
20994:                                         */
20995:                                         */
20996:                                         */
20997:                                         */
20998:                                         */
20999:                                         */
20100:                                         */
20101:                                         */
20102:                                         */
20103:                                         */
20104:                                         */
20105:                                         */
20106:                                         */
20107:                                         */
20108:                                         */
20109:                                         */
20110:                                         */
20111:                                         */
20112:                                         */
20113:                                         */
20114:                                         */
20115:                                         */
20116:                                         */
20117:                                         */
20118:                                         */
20119:                                         */
20120:                                         */
20121:                                         */
20122:                                         */
20123:                                         */
20124:                                         */
20125:                                         */
20126:                                         */
20127:                                         */
20128:                                         */
20129:                                         */
20130:                                         */
20131:                                         */
20132:                                         */
20133:                                         */
20134:                                         */
20135:                                         */
20136:                                         */
20137:                                         */
20138:                                         */
20139:                                         */
20140:                                         */
20141:                                         */
20142:                                         */
20143:                                         */
20144:                                         */
20145:                                         */
20146:                                         */
20147:                                         */
20148:                                         */
20149:                                         */
20150:                                         */
20151:                                         */
20152:                                         */
20153:                                         */
20154:                                         */
20155:                                         */
20156:                                         */
20157:                                         */
20158:                                         */
20159:                                         */
20160:                                         */
20161:                                         */
20162:                                         */
20163:                                         */
20164:                                         */
20165:                                         */
20166:                                         */
20167:                                         */
20168:                                         */
20169:                                         */
20170:                                         */
20171:                                         */
20172:                                         */
20173:                                         */
20174:                                         */
20175:                                         */
20176:                                         */
20177:                                         */
20178:                                         */
20179:                                         */
20180:                                         */
20181:                                         */
20182:                                         */
20183:                                         */
20184:                                         */
20185:                                         */
20186:                                         */
20187:                                         */
20188:                                         */
20189:                                         */
20190:                                         */
20191:                                         */
20192:                                         */
20193:                                         */
20194:                                         */
20195:                                         */
20196:                                         */
20197:                                         */
20198:                                         */
20199:                                         */
20200:                                         */
20201:                                         */
20202:                                         */
20203:                                         */
20204:                                         */
20205:                                         */
20206:                                         */
20207:                                         */
20208:                                         */
20209:                                         */
20210:                                         */
20211:                                         */
20212:                                         */
20213:                                         */
20214:                                         */
20215:                                         */
20216:                                         */
20217:                                         */
20218:                                         */
20219:                                         */
20220:                                         */
20221:                                         */
20222:                                         */
20223:                                         */
20224:                                         */
20225:                                         */
20226:                                         */
20227:                                         */
20228:                                         */
20229:                                         */
20230:                                         */
20231:                                         */
20232:                                         */
20233:                                         */
20234:                                         */
20235:                                         */
20236:                                         */
20237:                                         */
20238:                                         */
20239:                                         */
20240:                                         */
20241:                                         */
20242:                                         */
20243:                                         */
20244:                                         */
20245:                                         */
20246:                                         */
20247:                                         */
20248:                                         */
20249:                                         */
20250:                                         */
20251:                                         */
20252:                                         */
20253:                                         */
20254:                                         */
20255:                                         */
20256:                                         */
20257:                                         */
20258:                                         */
20259:                                         */
20260:                                         */
20261:                                         */
20262:                                         */
20263:                                         */
20264:                                         */
20265:                                         */
20266:                                         */
20267:                                         */
20268:                                         */
20269:                                         */
20270:                                         */
20271:                                         */
20272:                                         */
20273:                                         */
20274:                                         */
20275:                                         */
20276:                                         */
20277:                                         */
20278:                                         */
20279:                                         */
20280:                                         */
20281:                                         */
20282:                                         */
20283:                                         */
20284:                                         */
20285:                                         */
20286:                                         */
20287:                                         */
20288:                                         */
20289:                                         */
20290:                                         */
20291:                                         */
20292:                                         */
20293:                                         */
20294:                                         */
20295:                                         */
20296:                                         */
20297:                                         */
20298:                                         */
20299:                                         */
20300:                                         */
20301:                                         */
20302:                                         */
20303:                                         */
20304:                                         */
20305:                                         */
20306:                                         */
20307:                                         */
20308:                                         */
20309:                                         */
20310:                                         */
20311:                                         */
20312:                                         */
20313:                                         */
20314:                                         */
20315:                                         */
20316:                                         */
20317:                                         */
20318:                                         */
20319:                                         */
20320:                                         */
20321:                                         */
20322:                                         */
20323:                                         */
20324:                                         */
20325:                                         */
20326:                                         */
20327:                                         */
20328:                                         */
20329:                                         */
20330:                                         */
20331:                                         */
20332:                                         */
20333:                                         */
20334:                                         */
20335:                                         */
20336:                                         */
20337:                                         */
20338:                                         */
20339:                                         */
20340:                                         */
20341:                                         */
20342:                                         */
20343:                                         */
20344:                                         */
20345:                                         */
20346:                                         */
20347:                                         */
20348:                                         */
20349:                                         */
20350:                                         */
20351:                                         */
20352:                                         */
20353:                                         */
20354:                                         */
20355:                                         */
20356:                                         */
20357:                                         */
20358:                                         */

```

VideoMachine.java

```

127:     Office.setPassword(new String(jPasswordField.getPassword()));
128: 
129:     // Titel setzen und starten
130:     vidMachine.setShopFrameTitle("Videooverleihautomat *** HOMECINEMA *** 24h");
131:     vidMachine.start();
132: 
133:     vidMachine.getShopFrame().setSize(640, 480);
134:     vidMachine.getShopFrame().validate();
135: }
136: 
137: /**
138: * Erzeugt die Menuleiste des Hauptfenters unseres Automatens.
139: */
140: public MenuSheet createShopMenuSheet()
141: {
142:     // urspruengliche Menuleiste erzeugen und abspeichern
143:     MenuSheet msMenuBar = super.createShopMenuSheet();
144: 
145:     // 'Shop' -Menue heraus suchen lassen
146:     MenuSheet msShopMenu = (MenuSheet)msMenuBar.getTaggedItem(SHOP_MENU_TAG, true);
147:     msShopMenu.getShopMenuBar().getTaggedItem(SHOP_MENU_TAG, true);
148: 
149:     // nach unten zu schliebende Menupunkte zunaechst aus dem
150:     // Menue loeschen, aber in lokalen Variablen speichern
151:     MenuItem msLoadItem = (MenuItem)msShopMenu.remove(LOAD_TAG);
152:     MenuItem msSaveItem = (MenuItem)msShopMenu.remove(SAVE_TAG);
153:     MenuItemSeparator msSeparator = (MenuItemSeparator)msShopMenu.remove(SEPARATOR_TWO_TAG);
154:     MenuItem msQuitItem = (MenuItem)msShopMenu.remove(SEPARATOR_ONE_TAG);
155: 
156:     msMenuBar.removeItem(msLoadItem);
157:     msMenuBar.removeItem(msSaveItem);
158:     msMenuBar.removeItem(msSeparator);
159:     msMenuBar.removeItem(msQuitItem);
160: 
161:     new sale.Action()
162:     {
163:         public void doAction (SaleProcess p, SalesPoint sp)
164:         {
165:             List l = getSalesPoints();
166:             for (int i = 0; i < l.size(); i++)
167:                 if (((SalesPoint)l.get(i)).getName() == "Office")
168:                     javax.swing.JOptionPane.showMessagedialog(null,
169:                         "There is an already an Office",
170:                         "Error",
171:                         javax.swing.JOptionPane.ERROR_MESSAGE);
172:         }
173: 
174:         // neues JPanel anlegen und vertikales BoxLayout setzen
175:         JPanel jPasswordField = new JPanel();
176:         jPasswordField.setLayout(new BoxLayout(BoxLayout.Y_AXIS));
177: 
178:         // Passwortfeld erzeugen
179:         jPasswordField.setLayout(new GridLayout());
180:         jPasswordField.add(jPasswordField);
181: 
182:         // Label und Passwortfelder einfuegen
183:         jPasswordField.add(new JLabel("Enter the manager's password:"));
184:         jPasswordField.add(jPasswordField);
185: 
186:         // Passwortabfrage
187:         JOptionPane.showMessagedialog(null,
188:             jPasswordField,
189:             "Manager",
190:             JOptionPane.QUESTION_MESSAGE);
191: 
192:         // ist eingegebenes Passwort korrekt?
193:         if (Office.testPassword(new String(jPasswordField.getPassword())))
194:             /**
195:             * ja -> Buero anlegen
196:             * new Office("Office");
197:             * o.attach (new DataBaseBasketImpl());
198:             * o.attach (new User ("Manager"));
199:             * Shop.getTheShop().addSalesPoint (o);
200:         */
201: 
202:         // nein -> Info an Nutzer
203:         else
204:             JOptionPane.showMessagedialog(null,
205:                 "Access denied!");
206:         }
207:     );
208: 
209:     // Menueeintrage in der gewunschten Reihenfolge wieder einfuegen
210:     msShopMenu.add(msManagerItem);
211:     msShopMenu.add(msSeparator);
212:     msShopMenu.add(msLoadItem);
213:     msShopMenu.add(msSaveItem);
214:     msShopMenu.add(msSeparator);
215:     msShopMenu.add(msQuitItem);
216: 
217:     // Menuleiste zurueck geben
218:     return msMenuBar;
219: }
220: 
221: /**
222: * Beendet das Programm ohne den Stand abzuspeichern.
223: */
224: public void quit()
225: {
226:     public void quit()
227:     {
228:         if (Shop.getTheShop().shutdown (false))
229:             System.exit (0);
230:     }
231: }
232: /**
233: * F&uuml;igt der Liste registrierter Kunden einen neuen hinzu.
234: */
235: public static void addCustomer(Customer customer)
236: {
237:     customerSet.add(customer);
238: }
239: 
240: /**
241: * L&ouml;scht den angegebenen Kunden aus der Liste registrierter
242: * Kunden heraus.
243: */
244: public static void removeCustomer(Customer customer)
245: {
246:     customerSet.remove(customer);
247: }
248: 
249: /**
250: * Liefert eine Liste aller registrierter Kunden.
251: */
252: */

```

```
253: public static Set getAllCustomer()
254: {
255:     return customerSet.isEmpty() ? null : customerSet;
256: }
257:
258:
259: /**
260:  * Liefert das Kundenobject zur &uuml;bergebenen Kundennummer.
261:  */
262: public static Customer getCustomerByID(String customerID)
263: {
264:     Iterator i = customerSet.iterator();
265:     while (i.hasNext()) {
266:         Customer customer = (Customer)i.next();
267:         if (customer.getCustomerID().equals(customerID))
268:             return customer;
269:     }
270:     return null;
271: }
272: }
```

Table of Contents

1 CassetteStoringStockItem.java	1 pages	40 lines	01/03/20 11:28:17
2 Counter.java	1 pages	32 lines	01/03/20 11:28:17
3 Customer.java	1 pages	74 lines	01/03/20 11:28:17
4 DMCellEditor.java	1 pages	90 lines	01/03/20 11:28:17
5 DefaultCounterFormCreator.java	1 pages	71 lines	01/03/20 11:28:17
6 DefaultOfficeFormCreator.java	1 pages	76 lines	01/04/09 14:46:36
7 EditableVideoStockRED.java	2 pages	204 lines	01/03/20 11:28:17
8 GiveBackProcess.java	5 pages	522 lines	01/03/20 11:28:17
9 OfferRED.java	1 pages	64 lines	01/03/20 11:28:17
10 Office.java	2 pages	194 lines	01/03/20 11:28:17
11 RentProcess.java	5 pages	586 lines	01/04/01 19:00:37
12 SeeVideoStockProcess.java	3 pages	290 lines	01/03/20 11:28:17
13 Videocassette.java	1 pages	43 lines	01/03/20 11:28:17
14 VideoMachine.java	3 pages	272 lines	01/03/20 11:28:17