

1 Anhang zu den Übungsaufgaben mit YAWL

1.1 YAWL-Aufgabe1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<specificationSet xmlns="http://www.citi.qut.edu.au/yawl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="Beta 4"
xsi:schemaLocation="http://www.citi.qut.edu.au/yawl d:/yawl/schema/YAWL_SchemaBeta4.xsd">
  <specification uri="Aufgabe1.yawl">
    <metaData />
    <schema xmlns="http://www.w3.org/2001/XMLSchema" />
    <decomposition id="Aufgabe1" isRootNet="true" xsi:type="NetFactsType">
      <processControlElements>
        <inputCondition id="0_InputCondition">
          <flowsInto>
            <nextElementRef id="5_Registrieren" />
          </flowsInto>
        </inputCondition>
        <task id="5_Registrieren">
          <flowsInto>
            <nextElementRef id="4_Mietwagen" />
            <predicate>true()</predicate>
            <isDefaultFlow />
          </flowsInto>
          <flowsInto>
            <nextElementRef id="6_Hotel" />
            <predicate>true()</predicate>
          </flowsInto>
          <flowsInto>
            <nextElementRef id="2_Flug" />
            <predicate>true()</predicate>
          </flowsInto>
          <join code="xor" />
          <split code="or" />
          <decomposesTo id="Registrieren" />
        </task>
        <task id="4_Mietwagen">
          <flowsInto>
            <nextElementRef id="3_Bezahlen" />
          </flowsInto>
          <join code="xor" />
          <split code="and" />
          <decomposesTo id="Mietwagen" />
        </task>
        <task id="2_Flug">
          <flowsInto>
            <nextElementRef id="3_Bezahlen" />
          </flowsInto>
          <join code="xor" />
          <split code="and" />
          <decomposesTo id="Flug" />
        </task>
        <task id="6_Hotel">
          <flowsInto>
            <nextElementRef id="3_Bezahlen" />
          </flowsInto>
          <join code="xor" />
          <split code="and" />
          <decomposesTo id="Hotel" />
        </task>
        <task id="3_Bezahlen">
          <flowsInto>
            <nextElementRef id="1_OutputCondition" />
          </flowsInto>
          <join code="or" />
          <split code="and" />
          <decomposesTo id="Bezahlen" />
        </task>
        <outputCondition id="1_OutputCondition" />
      </processControlElements>
    </decomposition>
    <decomposition id="Hotel" xsi:type="WebServiceGatewayFactsType" />
    <decomposition id="Registrieren" xsi:type="WebServiceGatewayFactsType" />
    <decomposition id="Bezahlen" xsi:type="WebServiceGatewayFactsType" />
    <decomposition id="Mietwagen" xsi:type="WebServiceGatewayFactsType" />
  </specification>
</specificationSet>
```

```

    <decomposition id="Flug" xsi:type="WebServiceGatewayFactsType" />
  </specification>
</specificationSet>

```

1.2 YAWL-Aufgabe2.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<specificationSet xmlns="http://www.citi.qut.edu.au/yawl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="Beta 4"
xsi:schemaLocation="http://www.citi.qut.edu.au/yawl d:/yawl/schema/YAWL_SchemaBeta4.xsd">
  <specification uri="Aufgabe2.yawl">
    <metaData />
    <schema xmlns="http://www.w3.org/2001/XMLSchema" />
    <decomposition id="Aufgabe2" isRootNet="true" xsi:type="NetFactsType">
      <inputParam>
        <name>Hotelgewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
      </inputParam>
      <inputParam>
        <name>Fluggewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
      </inputParam>
      <inputParam>
        <name>Mietwagengewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
      </inputParam>
      <outputParam>
        <name>Hotelgewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
      </outputParam>
      <outputParam>
        <name>Fluggewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
      </outputParam>
      <outputParam>
        <name>Mietwagengewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
      </outputParam>
      <processControlElements>
        <inputCondition id="0_InputCondition">
          <flowsInto>
            <nextElementRef id="6_Registrieren" />
          </flowsInto>
        </inputCondition>
        <task id="6_Registrieren">
          <flowsInto>
            <nextElementRef id="3_Flug" />
            <predicate>true()</predicate>
          </flowsInto>
          <flowsInto>
            <nextElementRef id="5_Mietwagen" />
            <predicate>false()</predicate>
            <isDefaultFlow />
          </flowsInto>
          <flowsInto>
            <nextElementRef id="2_Hotel" />
            <predicate>false()</predicate>
          </flowsInto>
          <join code="xor" />
          <split code="or" />
          <startingMappings>
            <mapping>
              <expression query="&lt;Hotelgewuenscht&gt;{/Aufgabe2/Hotelgewuenscht/text()}&lt;/Hotelgewuenscht&gt;" />
              <mapsTo>Hotelgewuenscht</mapsTo>
            </mapping>
            <mapping>
              <expression query="&lt;Fluggewuenscht&gt;{/Aufgabe2/Fluggewuenscht/text()}&lt;/Fluggewuenscht&gt;" />
              <mapsTo>Fluggewuenscht</mapsTo>
            </mapping>
            <mapping>

```

```

        <expression query="&lt;Mietwagengewuenscht&gt;{/Aufgabe2/Mietwagengewuenscht/text()}&lt;Mietwagengewuenscht&gt;" />
        <mapsTo>Mietwagengewuenscht</mapsTo>
    </mapping>
</startingMappings>
<completedMappings>
    <mapping>
        <expression query="&lt;Fluggewuenscht&gt;{/Registrieren/Fluggewuenscht/text()}&lt;Fluggewuenscht&gt;" />
        <mapsTo>Fluggewuenscht</mapsTo>
    </mapping>
    <mapping>
        <expression query="&lt;Hotelgewuenscht&gt;{/Registrieren/Hotelgewuenscht/text()}&lt;Hotelgewuenscht&gt;" />
        <mapsTo>Hotelgewuenscht</mapsTo>
    </mapping>
    <mapping>
        <expression query="&lt;Mietwagengewuenscht&gt;{/Registrieren/Mietwagengewuenscht/text()}&lt;Mietwagengewuenscht&gt;" />
/>
    <mapsTo>Mietwagengewuenscht</mapsTo>
</mapping>
</completedMappings>
<decomposesTo id="Registrieren" />
</task>
<task id="2_Hotel">
    <flowsInto>
        <nextElementRef id="4_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Hotel" />
</task>
<task id="5_Mietwagen">
    <flowsInto>
        <nextElementRef id="4_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Mietwagen" />
</task>
<task id="3_Flug">
    <flowsInto>
        <nextElementRef id="4_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Flug" />
</task>
<task id="4_Bezahlen">
    <flowsInto>
        <nextElementRef id="1_OutputCondition" />
    </flowsInto>
    <join code="or" />
    <split code="and" />
    <decomposesTo id="Bezahlen" />
</task>
<outputCondition id="1_OutputCondition" />
</processControlElements>
</decomposition>
<decomposition id="Hotel" xsi:type="WebServiceGatewayFactsType" />
<decomposition id="Registrieren" xsi:type="WebServiceGatewayFactsType">
    <inputParam>
        <name>Hotelgewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    </inputParam>
    <inputParam>
        <name>Fluggewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    </inputParam>
    <inputParam>
        <name>Mietwagengewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    </inputParam>
    <outputParam>
        <name>Hotelgewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>

```

```

</outputParam>
<outputParam>
  <name>Kundenname</name>
  <type>string</type>
  <namespace>http://www.w3.org/2001/XMLSchema</namespace>
</outputParam>
<outputParam>
  <name>Fluggewuenscht</name>
  <type>boolean</type>
  <namespace>http://www.w3.org/2001/XMLSchema</namespace>
</outputParam>
<outputParam>
  <name>Mietwagengewuenscht</name>
  <type>boolean</type>
  <namespace>http://www.w3.org/2001/XMLSchema</namespace>
</outputParam>
</decomposition>
<decomposition id="Bezahlen" xsi:type="WebServiceGatewayFactsType" />
<decomposition id="Mietwagen" xsi:type="WebServiceGatewayFactsType" />
<decomposition id="Flug" xsi:type="WebServiceGatewayFactsType" />
</specification>
</specificationSet>

```

1.3 YAWL-Aufgabe3.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<specificationSet xmlns="http://www.citi.qut.edu.au/yawl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="Beta 4"
xsi:schemaLocation="http://www.citi.qut.edu.au/yawl d:/yawl/schema/YAWL_SchemaBeta4.xsd">
  <specification uri="Aufgabe3.yawl">
    <metaData />
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="Flugbuchung">
        <sequence>
          <element name="von" type="string" />
          <element name="nach" type="string" />
          <element name="AnzahlTickets" type="double" />
          <element name="Flugklasse" type="string" />
        </sequence>
      </complexType>
      <complexType name="Hotelbuchung">
        <sequence>
          <element name="Hotelname" type="string" />
          <element name="AnzahlBetten" type="double" />
          <element name="Fruehstueck" type="boolean" />
        </sequence>
      </complexType>
      <complexType name="Mietwagenbuchung">
        <sequence>
          <element name="Typbezeichnung" type="string" />
        </sequence>
      </complexType>
      <complexType name="Kunde">
        <sequence>
          <element name="Name" type="string" />
          <element name="Vorname" type="string" />
          <element name="Geburtsdatum" type="date" />
        </sequence>
      </complexType>
    </schema>
    <decomposition id="Aufgabe3" isRootNet="true" xsi:type="NetFactsType">
      <localVariable>
        <name>Fluggewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
        <initialValue>true</initialValue>
      </localVariable>
      <localVariable>
        <name>Hotelgewuenscht</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
        <initialValue>true</initialValue>
      </localVariable>
      <localVariable>
        <name>Mietwagengewuenscht</name>
        <type>boolean</type>

```

```

<namespace>http://www.w3.org/2001/XMLSchema</namespace>
<initialValue>true</initialValue>
</localVariable>
<processControlElements>
  <inputCondition id="0_InputCondition">
    <flowsInto>
      <nextElementRef id="5_Registrieren" />
    </flowsInto>
  </inputCondition>
  <task id="5_Registrieren">
    <flowsInto>
      <nextElementRef id="2_Mietwagen" />
      <predicate>/Aufgabe3/Mietwagengewuenscht/text()='true'</predicate>
      <isDefaultFlow />
    </flowsInto>
    <flowsInto>
      <nextElementRef id="6_Flug" />
      <predicate>/Aufgabe3/Fluggewuenscht/text()='true'</predicate>
    </flowsInto>
    <flowsInto>
      <nextElementRef id="4_Hotel" />
      <predicate>/Aufgabe3/Hotelgewuenscht/text()='true'</predicate>
    </flowsInto>
    <join code="xor" />
    <split code="or" />
    <completedMappings>
      <mapping>
        <expression query="&lt;Fluggewuenscht&gt;{/Registrieren/Flug/text()}&lt;Fluggewuenscht&gt;"/>
        <mapsTo>Fluggewuenscht</mapsTo>
      </mapping>
      <mapping>
        <expression query="&lt;Mietwagengewuenscht&gt;{/Registrieren/Mietwagen/text()}&lt;Mietwagengewuenscht&gt;"/>
        <mapsTo>Mietwagengewuenscht</mapsTo>
      </mapping>
      <mapping>
        <expression query="&lt;Hotelgewuenscht&gt;{/Registrieren/Hotel/text()}&lt;Hotelgewuenscht&gt;"/>
        <mapsTo>Hotelgewuenscht</mapsTo>
      </mapping>
    </completedMappings>
    <decomposesTo id="Registrieren" />
  </task>
  <task id="6_Flug">
    <flowsInto>
      <nextElementRef id="3_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Flug" />
  </task>
  <task id="4_Hotel">
    <flowsInto>
      <nextElementRef id="3_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Hotel" />
  </task>
  <task id="2_Mietwagen">
    <flowsInto>
      <nextElementRef id="3_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Mietwagen" />
  </task>
  <task id="3_Bezahlen">
    <flowsInto>
      <nextElementRef id="1_OutputCondition" />
    </flowsInto>
    <join code="or" />
    <split code="and" />
    <decomposesTo id="Bezahlen" />
  </task>
  <outputCondition id="1_OutputCondition" />
</processControlElements>
</decomposition>
<decomposition id="Hotel" xsi:type="WebServiceGatewayFactsType">

```

```

<outputParam>
  <name>Buchung</name>
  <type>Hotelbuchung</type>
  <namespace>http://www.w3.org/2001/XMLSchema</namespace>
</outputParam>
</decomposition>
<decomposition id="Registrieren" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Hotel</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Mietwagen</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Flug</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="Bezahlen" xsi:type="WebServiceGatewayFactsType" />
<decomposition id="Mietwagen" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Buchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="Flug" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Buchung</name>
    <type>Flugbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
</specification>
</specificationSet>

```

1.4 YAWL-Aufgabe4.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<specificationSet xmlns="http://www.citi.qut.edu.au/yawl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="Beta 4"
xsi:schemaLocation="http://www.citi.qut.edu.au/yawl d:/yawl/schema/YAWL_SchemaBeta4.xsd">
  <specification uri="Aufgabe4.yawl">
    <metaData />
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="Flugbuchung">
        <sequence>
          <element name="von" type="string" />
          <element name="nach" type="string" />
          <element name="AnzahlTickets" type="double" />
          <element name="Flugklasse" type="string" />
        </sequence>
      </complexType>
      <complexType name="Hotelbuchung">
        <sequence>
          <element name="Hotelname" type="string" />
          <element name="AnzahlBetten" type="double" />
          <element name="Fruehstueck" type="boolean" />
        </sequence>
      </complexType>
      <complexType name="Mietwagenbuchung">
        <sequence>
          <element name="Typbezeichnung" type="string" />
        </sequence>
      </complexType>
      <complexType name="Kunde">

```

```

<sequence>
  <element name="Name" type="string" />
  <element name="Vorname" type="string" />
  <element name="Geburtsdatum" type="date" />
</sequence>
</complexType>
</schema>
<decomposition id="Aufgabe4" isRootNet="true" xsi:type="NetFactsType">
  <localVariable>
    <name>Fluggewuenscht</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue>true</initialValue>
  </localVariable>
  <localVariable>
    <name>Hotelgewuenscht</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue>true</initialValue>
  </localVariable>
  <localVariable>
    <name>Mietwagengewuenscht</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue>true</initialValue>
  </localVariable>
  <processControlElements>
    <inputCondition id="0_InputCondition">
      <flowsInto>
        <nextElementRef id="2_Registrieren" />
      </flowsInto>
    </inputCondition>
    <task id="2_Registrieren">
      <flowsInto>
        <nextElementRef id="3_Flug" />
        <predicate>/Aufgabe4/Fluggewuenscht/text()='true'</predicate>
      </flowsInto>
      <flowsInto>
        <nextElementRef id="6_SubNetMietwagen" />
        <predicate>/Aufgabe4/Mietwagengewuenscht/text()='true'</predicate>
        <isDefaultFlow />
      </flowsInto>
      <flowsInto>
        <nextElementRef id="4_Hotel" />
        <predicate>/Aufgabe4/Hotelgewuenscht/text()='true'</predicate>
      </flowsInto>
      <join code="xor" />
      <split code="or" />
      <completedMappings>
        <mapping>
          <expression query="&lt;Fluggewuenscht&gt;{/Registrieren/Flug/text()}&lt;Fluggewuenscht&gt;" />
          <mapsTo>Fluggewuenscht</mapsTo>
        </mapping>
        <mapping>
          <expression query="&lt;Mietwagengewuenscht&gt;{/Registrieren/Mietwagen/text()}&lt;Mietwagengewuenscht&gt;" />
          <mapsTo>Mietwagengewuenscht</mapsTo>
        </mapping>
        <mapping>
          <expression query="&lt;Hotelgewuenscht&gt;{/Registrieren/Hotel/text()}&lt;Hotelgewuenscht&gt;" />
          <mapsTo>Hotelgewuenscht</mapsTo>
        </mapping>
      </completedMappings>
      <decomposesTo id="Registrieren" />
    </task>
    <task id="3_Flug">
      <flowsInto>
        <nextElementRef id="5_Bezahlen" />
      </flowsInto>
      <join code="xor" />
      <split code="and" />
      <decomposesTo id="Flug" />
    </task>
    <task id="6_SubNetMietwagen">
      <flowsInto>
        <nextElementRef id="5_Bezahlen" />
      </flowsInto>
      <join code="xor" />
    </task>
  </processControlElements>
</decomposition>

```

```

    <split code="and" />
    <decomposesTo id="SubNetMietwagen" />
  </task>
  <task id="4_Hotel">
    <flowsInto>
      <nextElementRef id="5_Bezahlen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <decomposesTo id="Hotel" />
  </task>
  <task id="5_Bezahlen">
    <flowsInto>
      <nextElementRef id="1_OutputCondition" />
    </flowsInto>
    <join code="or" />
    <split code="and" />
    <decomposesTo id="Bezahlen" />
  </task>
  <outputCondition id="1_OutputCondition" />
</processControlElements>
</decomposition>
<decomposition id="Hotel" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Buchung</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="SubNetMietwagen" xsi:type="NetFactsType">
  <localVariable>
    <name>FuehrerscheinGueltig</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue />
  </localVariable>
  <processControlElements>
    <inputCondition id="13_InputCondition">
      <flowsInto>
        <nextElementRef id="15_FuehrerscheinPruefen" />
      </flowsInto>
    </inputCondition>
    <task id="15_FuehrerscheinPruefen">
      <flowsInto>
        <nextElementRef id="16_Mietwagen" />
        <predicate ordering="0">/SubNetMietwagen/FuehrerscheinGueltig/text()='true'</predicate>
      </flowsInto>
      <flowsInto>
        <nextElementRef id="14_OutputCondition" />
        <isDefaultFlow />
      </flowsInto>
      <join code="xor" />
      <split code="xor" />
      <completedMappings>
        <mapping>
          <expression
query="&lt;FuehrerscheinGueltig&gt;{/FuehrerscheinPruefen/FuehrerscheinGueltig/text()}&lt;/FuehrerscheinGueltig&gt;" />
          <mapsTo>FuehrerscheinGueltig</mapsTo>
        </mapping>
      </completedMappings>
      <decomposesTo id="FuehrerscheinPruefen" />
    </task>
    <task id="16_Mietwagen">
      <flowsInto>
        <nextElementRef id="14_OutputCondition" />
      </flowsInto>
      <join code="xor" />
      <split code="and" />
      <decomposesTo id="Mietwagen" />
    </task>
    <outputCondition id="14_OutputCondition" />
  </processControlElements>
</decomposition>
<decomposition id="Registrieren" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Hotel</name>
    <type>boolean</type>

```

```

    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Mietwagen</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Flug</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="Bezahlen" xsi:type="WebServiceGatewayFactsType" />
<decomposition id="Mietwagen" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Buchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="FuehrerscheinPruefen" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>FuehrerscheinGueltig</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="Flug" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Buchung</name>
    <type>Flugbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
</specification>
</specificationSet>

```

1.5 YAWL-Aufgabe5.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<specificationSet xmlns="http://www.citi.qut.edu.au/yawl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="Beta 4"
xsi:schemaLocation="http://www.citi.qut.edu.au/yawl d:/yawl/schema/YAWL_SchemaBeta4.xsd">
  <specification uri="Aufgabe5.ywl">
    <metaData />
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="Flugbuchung">
        <sequence>
          <element name="von" type="string" />
          <element name="nach" type="string" />
          <element name="AnzahlTickets" type="double" />
          <element name="Flugklasse" type="string" />
        </sequence>
      </complexType>
      <complexType name="Hotelbuchung">
        <sequence>
          <element maxOccurs="unbounded" name="Zimmer">
            <complexType>
              <sequence>
                <element name="AnzahlBetten" type="double" />
                <element name="Fruehstueck" type="boolean" />
              </sequence>
            </complexType>
          </element>
        </sequence>
      </complexType>
      <complexType name="Mietwagenbuchung">
        <sequence>
          <element name="Typbezeichnung" type="string" />
        </sequence>
      </complexType>
    </schema>
  </specification>
</specificationSet>

```

```

</complexType>
<complexType name="Kunde">
  <sequence>
    <element name="Name" type="string" />
    <element name="Vorname" type="string" />
    <element name="Geburtsdatum" type="date" />
  </sequence>
</complexType>
<complexType name="Zimmer">
  <sequence>
    <element name="AnzahlBetten" type="double" />
    <element name="Fruehstueck" type="boolean" />
  </sequence>
</complexType>
</schema>
<decomposition id="Aufgabe4" isRootNet="true" xsi:type="NetFactType">
  <inputParam>
    <name>Hotelb</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <inputParam>
    <name>Flugh</name>
    <type>Flugbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <inputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <inputParam>
    <name>Mietwagenb</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <outputParam>
    <name>Hotelb</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Flugh</name>
    <type>Flugbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Mietwagenb</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <localVariable>
    <name>Fluggewuenscht</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue>true</initialValue>
  </localVariable>
  <localVariable>
    <name>Hotelgewuenscht</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue>true</initialValue>
  </localVariable>
  <localVariable>
    <name>Mietwagengewuenscht</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    <initialValue>true</initialValue>
  </localVariable>
  <processControlElements>
    <inputCondition id="0_InputCondition">
      <flowsInto>

```

```

    <nextElementRef id="2_Registrieren" />
  </flowsInto>
</inputCondition>
<task id="2_Registrieren">
  <flowsInto>
    <nextElementRef id="6_SubNetHotel" />
    <predicate>/Aufgabe4/Hotelgewuenscht/text()='true'</predicate>
    <isDefaultFlow />
  </flowsInto>
  <flowsInto>
    <nextElementRef id="5_SubNetMietwagen" />
    <predicate>/Aufgabe4/Mietwagengewuenscht/text()='true'</predicate>
  </flowsInto>
  <flowsInto>
    <nextElementRef id="4_Flug" />
    <predicate>/Aufgabe4/Fluggewuenscht/text()='true'</predicate>
  </flowsInto>
  <join code="xor" />
  <split code="or" />
  <completedMappings>
    <mapping>
      <expression query="&lt;Fluggewuenscht&gt;{/Registrieren/Flug/text()}&lt;/Fluggewuenscht&gt;" />
      <mapsTo>Fluggewuenscht</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Mietwagengewuenscht&gt;{/Registrieren/Mietwagen/text()}&lt;/Mietwagengewuenscht&gt;" />
      <mapsTo>Mietwagengewuenscht</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Hotelgewuenscht&gt;{/Registrieren/Hotel/text()}&lt;/Hotelgewuenscht&gt;" />
      <mapsTo>Hotelgewuenscht</mapsTo>
    </mapping>
  </completedMappings>
  <decomposesTo id="Registrieren" />
</task>
<task id="5_SubNetMietwagen">
  <flowsInto>
    <nextElementRef id="3_Bezahlen" />
  </flowsInto>
  <join code="xor" />
  <split code="and" />
  <startingMappings>
    <mapping>
      <expression query="&lt;Mietwagenbuchung&gt;{/Aufgabe4/Mietwagenb/*}&lt;/Mietwagenbuchung&gt;" />
      <mapsTo>Mietwagenbuchung</mapsTo>
    </mapping>
  </startingMappings>
  <completedMappings>
    <mapping>
      <expression query="&lt;Mietwagenb&gt;{/SubNetMietwagen/Mietwagenbuchung/*}&lt;/Mietwagenb&gt;" />
      <mapsTo>Mietwagenb</mapsTo>
    </mapping>
  </completedMappings>
  <decomposesTo id="SubNetMietwagen" />
</task>
<task id="6_SubNetHotel">
  <flowsInto>
    <nextElementRef id="3_Bezahlen" />
  </flowsInto>
  <join code="xor" />
  <split code="and" />
  <startingMappings>
    <mapping>
      <expression query="&lt;Hotelbuchung&gt;{/Aufgabe4/Hotelb/*}&lt;/Hotelbuchung&gt;" />
      <mapsTo>Hotelbuchung</mapsTo>
    </mapping>
  </startingMappings>
  <completedMappings>
    <mapping>
      <expression query="&lt;Hotelb&gt;{/SubNetHotel/Hotelbuchung/*}&lt;/Hotelb&gt;" />
      <mapsTo>Hotelb</mapsTo>
    </mapping>
  </completedMappings>
  <decomposesTo id="SubNetHotel" />
</task>
<task id="4_Flug">
  <flowsInto>

```

```

    <nextElementRef id="3_Bezahlen" />
  </flowsInto>
  <join code="xor" />
  <split code="and" />
  <completedMappings>
    <mapping>
      <expression query="&lt;Flug&gt;{/Flug/Buchung/*}&lt;/Flug&gt;" />
      <mapsTo>Flugb</mapsTo>
    </mapping>
  </completedMappings>
  <decomposesTo id="Flug" />
</task>
<task id="3_Bezahlen">
  <flowsInto>
    <nextElementRef id="1_OutputCondition" />
  </flowsInto>
  <join code="or" />
  <split code="and" />
  <startingMappings>
    <mapping>
      <expression query="&lt;Mietwagenbuchung&gt;{/Aufgabe4/Mietwagenb/*}&lt;/Mietwagenbuchung&gt;" />
      <mapsTo>Mietwagenbuchung</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Flugbuchung&gt;{/Aufgabe4/Flugb/*}&lt;/Flugbuchung&gt;" />
      <mapsTo>Flugbuchung</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Kunde&gt;{/Aufgabe4/Kunde/*}&lt;/Kunde&gt;" />
      <mapsTo>Kunde</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Hotelbuchung&gt;{/Aufgabe4/Hotelb/*}&lt;/Hotelbuchung&gt;" />
      <mapsTo>Hotelbuchung</mapsTo>
    </mapping>
  </startingMappings>
  <completedMappings>
    <mapping>
      <expression query="&lt;Hotelb&gt;{/Bezahlen/Hotelbuchung/*}&lt;/Hotelb&gt;" />
      <mapsTo>Hotelb</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Flugb&gt;{/Bezahlen/Flugbuchung/*}&lt;/Flugb&gt;" />
      <mapsTo>Flugb</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Kunde&gt;{/Bezahlen/Kunde/*}&lt;/Kunde&gt;" />
      <mapsTo>Kunde</mapsTo>
    </mapping>
    <mapping>
      <expression query="&lt;Mietwagenb&gt;{/Bezahlen/Mietwagenbuchung/*}&lt;/Mietwagenb&gt;" />
      <mapsTo>Mietwagenb</mapsTo>
    </mapping>
  </completedMappings>
  <decomposesTo id="Bezahlen" />
</task>
<outputCondition id="1_OutputCondition" />
</processControlElements>
</decomposition>
<decomposition id="SubNetMietwagen" xsi:type="NetFactsType">
  <inputParam>
    <name>Mietwagenbuchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <outputParam>
    <name>Mietwagenbuchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <localVariable>
    <name>FuehrerscheinGueltig</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </localVariable>
  <initialValue>&amp;lt;Mietwagenbuchung&gt;&amp;lt;Typbezeichnung&gt;&amp;lt;Typbezeichnung&gt;&amp;lt;Mietwa
genbuchung&gt;&lt;/initialValue>
  </inputParam>
  <outputParam>
    <name>Mietwagenbuchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <localVariable>
    <name>FuehrerscheinGueltig</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </localVariable>

```

```

    <initialValue />
  </localVariable>
</processControlElements>
  <inputCondition id="18_InputCondition">
    <flowsInto>
      <nextElementRef id="20_FuehrerscheinPruefen" />
    </flowsInto>
  </inputCondition>
  <task id="20_FuehrerscheinPruefen">
    <flowsInto>
      <nextElementRef id="19_OutputCondition" />
      <isDefaultFlow />
    </flowsInto>
    <flowsInto>
      <nextElementRef id="21_Mietwagen" />
      <predicate ordering="0">/SubNetMietwagen/FuehrerscheinGueltig/text()='true'</predicate>
    </flowsInto>
    <join code="xor" />
    <split code="xor" />
    <completedMappings>
      <mapping>
        <expression
query="&lt;FuehrerscheinGueltig&gt;{/FuehrerscheinPruefen/FuehrerscheinGueltig/text()}&lt;/FuehrerscheinGueltig&gt;" />
        <mapsTo>FuehrerscheinGueltig</mapsTo>
      </mapping>
    </completedMappings>
    <decomposesTo id="FuehrerscheinPruefen" />
  </task>
  <task id="21_Mietwagen">
    <flowsInto>
      <nextElementRef id="19_OutputCondition" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <completedMappings>
      <mapping>
        <expression query="&lt;Mietwagenbuchung&gt;{/Mietwagen/Buchung/*}&lt;/Mietwagenbuchung&gt;" />
        <mapsTo>Mietwagenbuchung</mapsTo>
      </mapping>
    </completedMappings>
    <decomposesTo id="Mietwagen" />
  </task>
  <outputCondition id="19_OutputCondition" />
</processControlElements>
</decomposition>
<decomposition id="ZimmerPruefen" xsi:type="WebServiceGatewayFactsType">
  <inputParam>
    <name>Zimmer</name>
    <type>Zimmer</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <outputParam>
    <name>Zimmer</name>
    <type>Zimmer</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="Registrieren" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Hotel</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Mietwagen</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Flug</name>
    <type>boolean</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>

```

```

</outputParam>
</decomposition>
<decomposition id="Hotelzimmer_waehlen" xsi:type="WebServiceGatewayFactsType">
  <outputParam>
    <name>Hotelbuchung</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="Bezahlen" xsi:type="WebServiceGatewayFactsType">
  <inputParam>
    <name>Mietwagenbuchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <inputParam>
    <name>Flugbuchung</name>
    <type>Flugbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <inputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <inputParam>
    <name>Hotelbuchung</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <outputParam>
    <name>Mietwagenbuchung</name>
    <type>Mietwagenbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Flugbuchung</name>
    <type>Flugbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Kunde</name>
    <type>Kunde</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
  <outputParam>
    <name>Hotelbuchung</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
</decomposition>
<decomposition id="SubNetHotel" xsi:type="NetFactsType">
  <inputParam>
    <name>Hotelbuchung</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </inputParam>
  <outputParam>
    <name>Hotelbuchung</name>
    <type>Hotelbuchung</type>
    <namespace>http://www.w3.org/2001/XMLSchema</namespace>
  </outputParam>
<processControlElements>
  <inputCondition id="13_InputCondition">
    <flowsInto>
      <nextElementRef id="16_Hotelzimmer_waehlen" />
    </flowsInto>
  </inputCondition>
  <task id="16_Hotelzimmer_waehlen">
    <flowsInto>
      <nextElementRef id="15_ZimmerPruefen" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <completedMappings>
      <mapping>
        <expression query="&lt;Hotelbuchung&gt;{/Hotelzimmer_waehlen/Hotelbuchung/*}&lt;/Hotelbuchung&gt;" />
      </mapping>
    </completedMappings>
  </task>
</processControlElements>

```

```

        <mapsTo>Hotelbuchung</mapsTo>
    </mapping>
</completedMappings>
<decomposesTo id="Hotelzimmer_waehlen" />
</task>
<task id="15_ZimmerPruefen" xsi:type="MultipleInstanceExternalTaskFactsType">
    <flowsInto>
        <nextElementRef id="14_OutputCondition" />
    </flowsInto>
    <join code="xor" />
    <split code="and" />
    <completedMappings>
        <mapping>
            <expression
query="&lt;Zimmer&gt;&lt;AnzahlBetten&gt;{/ZimmerPruefen/Zimmer/AnzahlBetten/text()}&lt;/AnzahlBetten&gt;&lt;Fruehstueck&gt;{/
ZimmerPruefen/Zimmer/Fruehstueck/text()}&lt;/Fruehstueck&gt;&lt;Zimmer&gt;"/>
            <mapsTo>Hotelbuchung</mapsTo>
        </mapping>
    </completedMappings>
    <decomposesTo id="ZimmerPruefen" />
    <minimum>1</minimum>
    <maximum>10</maximum>
    <threshold>5</threshold>
    <creationMode code="static" />
    <miDataInput>
        <expression query="/SubNetHotel/Hotelbuchung" />
        <splittingExpression query="for $e in /Hotelbuchung/* return $e" />
        <formalInputParam>Zimmer</formalInputParam>
    </miDataInput>
    <miDataOutput>
        <formalOutputExpression
query="&lt;Zimmer&gt;&lt;AnzahlBetten&gt;{/ZimmerPruefen/Zimmer/AnzahlBetten/text()}&lt;/AnzahlBetten&gt;&lt;Fruehstueck&gt;{/
ZimmerPruefen/Zimmer/Fruehstueck/text()}&lt;/Fruehstueck&gt;&lt;Zimmer&gt;"/>
        <outputJoiningExpression query="&lt;Hotelbuchung&gt;{for $d in /ZimmerPruefen/Zimmer return $d}&lt;/Hotelbuchung&gt;"/>
        <resultAppliedToLocalVariable>Hotelbuchung</resultAppliedToLocalVariable>
    </miDataOutput>
    </task>
    <outputCondition id="14_OutputCondition" />
</processControlElements>
</decomposition>
<decomposition id="Mietwagen" xsi:type="WebServiceGatewayFactsType">
    <outputParam>
        <name>Buchung</name>
        <type>Mietwagenbuchung</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    </outputParam>
</decomposition>
<decomposition id="FuehrerscheinPruefen" xsi:type="WebServiceGatewayFactsType">
    <outputParam>
        <name>FuehrerscheinGueltig</name>
        <type>boolean</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    </outputParam>
</decomposition>
<decomposition id="Flug" xsi:type="WebServiceGatewayFactsType">
    <outputParam>
        <name>Buchung</name>
        <type>Flugbuchung</type>
        <namespace>http://www.w3.org/2001/XMLSchema</namespace>
    </outputParam>
</decomposition>
</specification>
</specificationSet>

```

1.6 Containerbearbeiten.java

```

import com.ibm.workflow.api.*;
import java.io.*;

```

```

public class Containerbearbeiten
{

```

```

    //Zeigt alle alle Containerblätter an

```

```

public static void DisplayContainer( Container cnr ) throws FmcException
{
    // get all container leaves
    ContainerElement[] leaves = cnr.allLeaves( );

    System.out.println( "Container Values:" );

    // show all container leaves with their name, type and value
    for ( int i = 0; i < leaves.length; i++ )
    {
        System.out.print( leaves[ i ].fullName( ) );
        System.out.print( " [" + leaves[ i ].type( ) + "]: " );
        try
        {
            if ( leaves[ i ].type( ).equals( "STRING" ) )
            {
                String value = leaves[ i ].getString( );
                System.out.println( value );
            }
            else if ( leaves[ i ].type( ).equals( "LONG" ) )
            {
                int value = leaves[ i ].getLong( );
                System.out.println( value );
            }
            else if ( leaves[ i ].type( ).equals( "FLOAT" ) )
            {
                double value = leaves[ i ].getDouble( );
                System.out.println( value );
            }
            else if ( leaves[ i ].type( ).equals( "BINARY" ) )
            {
                byte[] buffer = leaves[ i ].getBuffer( );
                System.out.println( "<BINARY DATA>" );
            }
        }
        catch( FmcException xcpt )
        {
            if ( xcpt.rc == FmcException.FMC_ERROR_MEMBER_NOT_SET )
            {
                System.out.println( "<NOT SET>" );
            }
            else
            {
                System.out.println( "Error getting the data member " +
                    leaves[ i ].fullName( ) );
                System.out.println( xcpt.getMessage( ) );
            }
        }
    }
}

public static ReadWriteContainer SetContainer(ReadWriteContainer outCnr)
    throws FmcException
{
    BufferedReader ein = new BufferedReader(new InputStreamReader(System.in));
    String aString=new String();

    try {
        aString=ein.readLine();
    }
    catch (Exception e) {
        System.out.println("Eingabe fehlgeschlagen: "+e.getMessage());
        // Programm sofort beenden
        System.exit(1);
    }

    for (int k = 0; k < outCnr.leafCount(); k++)
    {
        try
        {
            if ( outCnr.leaves()[k].isEmpty() )
                continue;

            String aName = outCnr.leaves()[k].fullName();

```

```

String aType = outCnr.leaves()[k].type();

System.out.print("Bitte Wert eingeben fuer: " + aName + " Type: " + aType + " :");
ContainerElement element = outCnr.getElement(aName);

        //Einlesen der Eingabe
        try {
            aString=ein.readLine();
        }
        catch (Exception e) {
            System.out.println("Eingabe fehlgeschlagen: "+e.getMessage());
            // Programm sofort beenden
            System.exit(1);
        }
    if (aType.equals("STRING"))
    {
        outCnr.setString( aName, aString );
    }
    else if (aType.equals("LONG"))
    {
        outCnr.setLong( aName, Integer.parseInt(aString) );
    }
    else if (aType.equals("FLOAT"))
    {
        outCnr.setDouble( aName, Double.parseDouble(aString) );
    }
    else if (aType.equals("BINARY"))
    {
        byte[] aBuffer = aString.getBytes();
        outCnr.setBuffer( aName, aBuffer );
    }
}
catch( FmcException xcpt )
{
    if ( xcpt.rc == FmcException.FMC_ERROR_MEMBER_NOT_SET )
    {
        System.out.println( "<NOT SET>" );
    }
    else if ( xcpt.rc != FmcException.FMC_ERROR_MEMBER_CANNOT_BE_SET )
    {
        System.out.println( "Error getting/setting the data member " +
            outCnr.leaves()[k].fullName() );
        System.out.println( xcpt.getMessage( ) );
    }
}
}

return outCnr;
}

public static void main( String[] argv )
{

    try
    {
        Agent agent = new Agent( );

        agent.setLocator( Agent.LOC_LOCATOR );
        agent.setName( "LOCAL" );

        ExecutionAgent pea = agent.getExecutionAgent( );

        if ( pea != null )
        {
            // get the input- and output container
            ReadOnlyContainer input = pea.inContainer( );
            ReadWriteContainer output = pea.outContainer( );

            // display all input container members
            System.out.println( "Input Container:" );
            DisplayContainer( input );
                System.out.println("Weiter mit ENTER");
                System.in.read();

            // copy the data members from the input- to the output container
            SetContainer(output );

```

```
output.setLong("_RC", 0);

// return the output container to the PEA
pea.setOutContainer( output );
}
}
catch( FmcException xcpt )
{
    System.out.println( "Error in MQSeries Workflow call:" );
    xcpt.printStackTrace();
}
catch( Throwable xcpt )
{
    System.out.println( "Internal error:" );
    xcpt.printStackTrace();
}
finally
{
    try
    {
        System.out.println();
        System.out.println( "Press ENTER to end the program." );
        System.in.read();
    }
    catch( Exception xcpt )
    { }
}
}
```