

# Quellcodes

```
1 public class DoLink {
2     public static void main (CommAreaHolder cah) {
3         Task task = Task.getTask ();
4         PrintWriter out = task.out;
5         try {
6             // ## (1)
7             out.println();
8             out.println("This is program :" + task.getProgramName());
9             // ## (2)
10            String myMsg = "the answer to life the universe and
11            everything";
12            byte[] commArea = myMsg.getBytes();
13            out.println("My question :" + myMsg);
14            // ## (3)
15            Program prog = new Program();
16            prog.setName("CALLME");
17            prog.link(commArea);
18            // ## (4)
19            out.println();
20            String rply = new String(commArea);
21            out.println("And the answer is: " + rply);
22            // ## (5)
23        } catch (Exception e) {
24            e.printStackTrace(out);
25        }
26    }
27 }
28 }
29 }
30 }
31 }
```

Listing 1: Aufrufendes Programm mit LINK

```

1 public class DoXctl {
2     public static void main(CommAreaHolder cah) {
3         Task task = Task.getTask();
4         PrintWriter out = task.out;

6         try {
7             // ## (1)
8             out.println();
9             out.println(" This is program :" + task.getProgramName ());

11                // ## (2)
12                String myMsg = "the answer to life the universe and
everything";
13                byte[] commArea = myMsg.getBytes();
14                out.println("My question :" + myMsg);

16                // ## (3)
17                Program prog = new Program();
18                prog.setName("CALLME");
19                prog.xctl(commArea);

21                // ## (4)
22                out.println();
23                String rply = new String(commArea);
24                out.println("And the answer is: " + rply);

26                // ## (5)
27                } catch (Exception e) {
28                    e.printStackTrace(out);
29                }
30        }
31 }

```

Listing 2: Aufrufendes Programm mit XCTL

```
1 public class CallMe {
2     public static void main (CommAreaHolder cah) {
3         PrintWriter out = Task.getTask().out;
4         out.println();
5         out.println();
6         out.println();
7         // ## (1)
8         out.println("This is program CALLME");
9         // ## (2)
10        String theAnswer = "42";
11        cah.value = theAnswer.getBytes();
12    }
13 }
```

Listing 3: Aufrufendes Programm mit XCTL

```

1 public class NeedEuro {
2     public static void main (CommAreaHolder cah) {
3         try {
4             // ## (1)
5             Channel ch = Task.getTask().createChannel("DollarToEuro");
6             // ## (2)
7             Container dollarCon = ch.createContainer("DOLLAR");
8             // ## (3)
9             String dollarStr = "100";
10            dollarCon.put(dollarStr);
11            // ## (4)
12            Program moneyConverter = new Program ();
13            moneyConverter.setName("MONCONV");
14            // ## (5)
15            moneyConverter.link(ch);
17            PrintWriter out = Task.getTask().out;
18            out.println();
19            out.println("Give " + dollarStr + " Dollars");
20            // ## (6)
21            String euroStr = new String(dollarCon.get());
22            out.println("Get " + euroStr + " Euros");
24        } catch ( Exception e) {
25            e.printStackTrace(out);
26        }
27    }
28 }

```

Listing 4: Anfrage mit Channel

```

1 public class NeedDollar {
2     public static void main ( CommAreaHolder cah ) {
3         try {
4             // ## (1)
5             Channel ch = Task.getTask().createChannel("EuroToDollar");
6             // ## (2)
7             Container euroCon = ch.createContainer("EURO");
8             // ## (3)
9             String euroStr = "100";
10            euroCon.put(euroStr);
11            // ## (4)
12            Program moneyConverter = new Program ();
13            moneyConverter.setName("MONCONV");
14            // ## (5)
15            moneyConverter.link(ch);
16            PrintWriter out = Task.getTask().out;
17            out.println();
18            out.println("Give " + euroStr + " Euros");
19            // ## (6)
20            String dollarStr = new String(euroCon.get());
21            out.println("Get " + dollarStr + " Dollars");
22        } catch (Exception e) {
23            e.printStackTrace(out);
24        }
25    }
26 }

```

Listing 5: Anfrage mit Channel 2

```

1 public class MoneyConverter {
2     public static void main(CommAreaHolder cah) {
3         PrintWriter out = Task.getTask().out;
4         // ## (1)
5         Channel ch = Task.getTask().getCurrentChannel();
6         // ## (2)
7         if ( ch != null ) {
8             String chName = ch.getName().trim();
9             try {
10                // ## (3)
11                double resValue ;
12                if ( chName.equals("EuroToDollar")) {
13                    // ## (3-a)
14                    Container euroRec ;
15                    euroRec = ch. getContainer("EURO");
16                    double reqVal = Double.valueOf(new
String(euroRec.get()));
17                    resValue = euroToDollar(reqVal);
18                    euroRec.put(Double.toString(resValue));
19                } else if (chName.equals("DollarToEuro")) {
20                    // ## (3-b)
21                    Container dollarRec = ch. getContainer("DOLLAR");
22                    double reqVal = Double.valueOf(new
String(dollarRec.get()));
23                    resValue = dollarToEuro(reqVal);
24                    dollarRec.put(Double.toString(resValue));
25                }
26            } catch (Exception e) {
27                e.printStackTrace(out);
28            }
29        }
30        // ## (4-a)
31        private static double euroToDollar(double euro) {
32            return (euro * 1.30);
33        }
34        // ## (4-b)
35        private static double dollarToEuro(double euro) {
36            return (euro / 1.30);
37        }
38    }
39 }

```

Listing 6: Java Channel Multiplexer

```
1 Task t = Task.getTask();
2 ContainerIterator ci = t.containerIterator();
3 while (ci.hasNext()) {
4     Container custData = ci.next();
5     // Verarbeitung Container Inhalt
6 }
```

**Listing 7: Beispiel für einen ContainerIterator**