

Example 1 (ChatGPT-4o): migration test no. 1

Konvertierung 1

```
BeispielProgramm.java X
chatgpt4o > BeispielProgramm.java > ...
1 public class BeispielProgramm {
2
3     // Working-Storage Section
4     private static int counter = 42;
5
6     private static class MonatsNamen {
7         String jan = "Januar";
8         String feb = "Februar";
9         String mar = "Maerz";
10        String apr = "April";
11        String mai = "Mai";
12        String jun = "Juni";
13        String jul = "Juli";
14        String aug = "August";
15        String sep = "September";
16        String okt = "Oktober";
17        String nov = "November";
18        String dez = "Dezember";
19    }
20
21    private static MonatsNamen monatsNamen = new MonatsNamen();
22    private static String[] monate = {
23        monatsNamen.jan, monatsNamen.feb, monatsNamen.mar, monatsNamen.apr,
24        monatsNamen.mai, monatsNamen.jun, monatsNamen.jul, monatsNamen.aug,
25        monatsNamen.sep, monatsNamen.okt, monatsNamen.nov, monatsNamen.dez
26    };
27
28    public static void main(String[] args) {
29        // Procedure Division
30
31        // DISPLAY #1
32        System.out.println(counter);
33
34        // MOVE 84 to counter
35        counter = 84;
36
37        // DISPLAY #2
38        System.out.println(counter);
39
40        // DISPLAY #3
41        System.out.println(monate[8]);
42
43        // MOVE "-----" TO monate(9)
44        monate[8] = "-----";
45
46        // DISPLAY #4
47        System.out.println(monate[8]);
48
49        // MOVE "Monat 01Monat 02" TO my-struct
50        monatsNamen.jan = "Monat 01";
51        monatsNamen.feb = "Monat 02";
52        // Update the monate array to reflect changes in monatsNamen
53        monate[0] = monatsNamen.jan;
54        monate[1] = monatsNamen.feb;
55
56        // DISPLAY #5
57        System.out.println(monatsNamen.feb);
58
59        // DISPLAY #6
60        System.out.println(monate[1]);
61
62        // DISPLAY #7
63        System.out.println(monate[2]);
64    }
65
66 }
```

Falsches Ergebnis:

```
42
84
September
-----
Monat 02
Monat 02
Maerz
```

Firstly, it stands out that the **int** type has been selected for the `counter` variable. In COBOL, this is a PIC 9, i. e. an unsigned single-digit integer with the value range $\{x: 0 \leq x \leq 9\}$. The Java `int` type, on the other hand, has a value range of $\{x: -2.147.483.648 \leq x \leq 2.147.483.647\}$. Accordingly, after initialisation with 42, the variable in Java contains the value 42, whereas in COBOL it only contains 2. The same applies to the assignment of 84 further down.

The type for the month names (PIC X(8) in COBOL) is also incorrectly selected with **String** in Java. In COBOL, the length of the month name is always exactly 8 characters, whereas in Java it can be shorter or (e. g. in the case of *September*) longer. As a general rule, native Java types should not be used for migrated data elements, as COBOL data has different properties than Java data. Replicating the COBOL types with Java classes is the better choice here.

In the Java programme, the `my-struct` and `monats-liste` overlay is achieved by separate assignments to `monate[0]` and `monate[1]`. However, this does not take into account that the MOVE "Monat 01Monat 02" TO my-struct COBOL command changes the entire structure. The values of the `mar` to `dez` fields are filled with spaces. This does not happen in Java, which is why the last output line is **Maerz** and not empty, as is the case with the COBOL programme.

Example 1 (ChatGPT-4o): migration test no. 2

Konvertierung 2

```
BeispielProgramm.java X
chatgpt4o > BeispielProgramm.java > ...
1 public class BeispielProgramm {
2
3     // Counter variable
4     private static int counter = 42;
5
6     // Monatsnamen structure
7     private static class MonatsNamen {
8         String jan = "Januar";
9         String feb = "Februar";
10        String mar = "Maerz";
11        String apr = "April";
12        String mai = "Mai";
13        String jun = "Juni";
14        String jul = "Juli";
15        String aug = "August";
16        String sep = "September";
17        String okt = "Oktober";
18        String nov = "November";
19        String dez = "Dezember";
20    }
21
22    // Monatsliste array
23    private static String[] monate = {
24        "Januar", "Februar", "Maerz", "April", "Mai", "Juni",
25        "Juli", "August", "September", "Oktober", "November", "Dezember"
26    };
27
28    public static void main(String[] args) {
29        // DISPLAY #1
30        System.out.println(counter);
31
32        // MOVE 84 to counter
33        counter = 84;
34
35        // DISPLAY #2
36        System.out.println(counter);
37
38        // DISPLAY #3
39        System.out.println(monate[8]); // COBOL is 1-based, Java is 0-based, so 9th element is at index 8
40
41        // MOVE "-----" TO monate(9)
42        monate[8] = "-----";
43
44        // DISPLAY #4
45        System.out.println(monate[8]);
46
47        // MOVE "Monat 01Monat 02" TO my-struct
48        MonatsNamen myStruct = new MonatsNamen();
49        myStruct.jan = "Monat 01";
50        myStruct.feb = "Monat 02";
51
52        // DISPLAY #5
53        System.out.println(myStruct.feb);
54
55        // DISPLAY #6
56        System.out.println(monate[1]); // 2nd element is at index 1
57
58        // DISPLAY #7
59        System.out.println(monate[2]); // 3rd element is at index 2
60    }
61 }
62
```

As far as the types and missing fill up of spaces for the Maerz to Dezember fields are concerned, the same mistakes are being made here as in the first translation.

This time, however, the my-struct and monats-liste overlay is not taken into account at all. Therefore, in addition to the last output Maerz, the penultimate output is also incorrect.

Falsches Ergebnis:

```
42
84
September
-----
Monat 02
Februar
Maerz
```