

Ein Beispiel: Das Damenproblem

Acht Damen sollen auf einem Schachbrett so aufgestellt werden, dass keine zwei Damen einander gemäß ihren in den Schachregeln definierten Zugmöglichkeiten schlagen können. Für Damen heißt dies konkret: Es dürfen keine zwei Damen auf derselben Reihe, Linie oder Diagonale stehen.

Es gibt 92 mögliche Lösungen für das 8x8 Feld – aber wie findet man diese?

```
3 public class Damenproblem {
4     static int n = 8;
5     static int[][] spielBrett = new int[n][n];
6     static int DAME = 1;
7     static int LEER = 0;
8
9     public static boolean istGültig(int zeile, int spalte) {
10        for (int i = 0; i < n; i++) {
11            for (int j = 0; j < n; j++) {
12                if (spielBrett[i][j] == 1) {
13                    if (i == zeile || j == spalte) {
14                        return false;
15                    }
16                }
17            }
18        }
19        for (int i = 0; i < n; i++) {
20            if (zeile + i < n && spalte + i < n && spielBrett[zeile + i][spalte + i] ==
21                ↪ 1)
22                return false;
23            if (zeile - i > -1 && spalte - i > -1 && spielBrett[zeile - i][spalte - i]
24                ↪ == 1)
25                return false;
26            if (zeile + i < n && spalte - i > -1 && spielBrett[zeile + i][spalte - i]
27                ↪ == 1)
28                return false;
29            if (zeile - i > -1 && spalte + i < n && spielBrett[zeile - i][spalte + i]
30                ↪ == 1)
31                return false;
32        }
33        return true;
34    }
35
36    public static boolean löse(int zeile) {
37        if (zeile == n) {
38            return true;
39        }
40
41        for (int i = 0; i < n; i++) {
42            if (istGültig(zeile, i) == true) {
43                spielBrett[zeile][i] = DAME;
44
45                if (löse(zeile + 1) == true) {
46                    return true;
47                }
48                spielBrett[zeile][i] = LEER;
49            }
50        }
51        return false;
52    }
53
54    public static void zeigeSpielBrett() {
```

```

52     for (int i = 0; i < spielBrett.length; i++) {
53         for (int j = 0; j < spielBrett[i].length; j++) {
54             System.out.print(spielBrett[i][j] + " ");
55         }
56         System.out.println();
57     }
58 }
59
60 public static void fülleFeld() {
61     for (int i = 0; i < spielBrett.length; i++) {
62         for (int j = 0; j < spielBrett[i].length; j++) {
63             spielBrett[i][j] = 0;
64         }
65     }
66 }
67
68 /**
69  * Das Fenster wird irgendwie nicht in Sway WM angezeigt.
70  *
71  * @param args Kommandozeilen-Argumente
72  */
73 public static void main(String[] args) {
74     fülleFeld();
75     löse(0);
76     zeigeSpielBrett();
77     new Ausgabe(n, spielBrett);
78 }
79
80 }

3 import javax.swing.*;
4
5 import java.awt.Color;
6 import java.awt.Graphics;
7
8 @SuppressWarnings("serial")
9 public class Ausgabe extends JPanel {
10     JFrame f;
11     int[][] feld;
12     int feldGroesse;
13     final int fensterRand = 50;
14     final int fensterGroesse = 500;
15     Color feldColor = Color.DARK_GRAY;
16     int felderAnz;
17
18     public Ausgabe(int felderAnz, int[][] feld) {
19         feldGroesse = fensterGroesse / felderAnz;
20         this.felderAnz = felderAnz;
21         this.feld = feld;
22         f = new JFrame("Damenproblem");
23         f.setSize(fensterGroesse + 2 * fensterRand, fensterGroesse + 2 *
24             ↪ fensterRand);
25         f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26         f.setVisible(true);
27         f.add(this);
28     }
29
30     @Override
31     public void paint(Graphics g) {
32         int xPos = fensterRand;
33         int yPos = fensterRand;
34         if (felderAnz % 2 != 0) {

```

```

35     for (int i = 0; i < feld.length; i++) {
36         for (int j = 0; j < feld[i].length; j++) {
37             g.setColor(switchColor());
38             if (feld[i][j] == 1) {
39                 // Dame
40                 g.setColor(Color.ORANGE);
41                 g.fillRect(xPos, yPos, feldGroese, feldGroese);
42             } else {
43                 // Normalfeld
44                 g.fillRect(xPos, yPos, feldGroese, feldGroese);
45             }
46             xPos += feldGroese;
47         }
48         xPos = fensterRand;
49         yPos += feldGroese;
50     }
51 } else {
52     for (int i = 0; i < feld.length; i++) {
53         for (int j = 0; j < feld[i].length; j++) {
54             if (feld[i][j] == 1) {
55                 // Dame
56                 g.setColor(Color.ORANGE);
57                 g.fillRect(xPos, yPos, feldGroese, feldGroese);
58             } else {
59                 // Normalfeld
60                 g.fillRect(xPos, yPos, feldGroese, feldGroese);
61             }
62
63             xPos += feldGroese;
64             g.setColor(switchColor());
65         }
66         g.setColor(switchColor());
67         xPos = fensterRand;
68         yPos += feldGroese;
69     }
70 }
71 }
72
73 private Color switchColor() {
74     if (feldColor == Color.DARK_GRAY) {
75         feldColor = Color.GRAY;
76         return Color.GRAY;
77     } else if (feldColor == Color.GRAY) {
78         feldColor = Color.DARK_GRAY;
79         return Color.DARK_GRAY;
80     } else {
81         System.err.println("Fehler switchColor!");
82         return null;
83     }
84 }
85
86 }

```