Backtracking: Das Damenproblem

Implementieren sie mittels Backtracking einen Algorithmus, der acht Damen auf einem Schachbrett so aufgestellt, 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 8×8 Feld.

```
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] == 1)
21         return false;
22       if (zeile - i > -1 && spalte - i > -1 && spielBrett[zeile - i][spalte
         ↪  - i] == 1)
23         return false;
24       if (zeile + i < n && spalte - i > -1 && spielBrett[zeile + i][spalte -
         ↪  i] == 1)
25         return false;
26       if (zeile - i > -1 && spalte + i < n && spielBrett[zeile - i][spalte +
         ↪  i] == 1)
27         return false;
28     }
29     return true;
30   }
31
32   public static boolean löse(int zeile) {
33     if (zeile == n) {
34       return true;
35     }
36
37     for (int i = 0; i < n; i++) {
38       if (istGültig(zeile, i) == true) {
39         spielBrett[zeile][i] = DAME;
40
41         if (löse(zeile + 1) == true) {
42           return true;
43         }
44         spielBrett[zeile][i] = LEER;
45       }
46     }
47
48     return false;
49   }
```

Code-Beispiel auf Github ansehen:
`src/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Damenproblem.java`

Additum

```java
public class Damenproblem {
  static int n = 8;
  static int[][] spielBrett = new int[n][n];
  static int DAME = 1;
  static int LEER = 0;

  public static boolean istGültig(int zeile, int spalte) {
    for (int i = 0; i < n; i++) {
      for (int j = 0; j < n; j++) {
        if (spielBrett[i][j] == 1) {
          if (i == zeile || j == spalte) {
            return false;
          }
        }
      }
    }
    for (int i = 0; i < n; i++) {
      if (zeile + i < n && spalte + i < n && spielBrett[zeile + i][spalte +
      ↪  i] == 1)
        return false;
      if (zeile - i > -1 && spalte - i > -1 && spielBrett[zeile - i][spalte
      ↪  - i] == 1)
        return false;
      if (zeile + i < n && spalte - i > -1 && spielBrett[zeile + i][spalte -
      ↪  i] == 1)
        return false;
      if (zeile - i > -1 && spalte + i < n && spielBrett[zeile - i][spalte +
      ↪  i] == 1)
        return false;
    }
    return true;
  }

  public static boolean löse(int zeile) {
    if (zeile == n) {
      return true;
    }

    for (int i = 0; i < n; i++) {
      if (istGültig(zeile, i) == true) {
        spielBrett[zeile][i] = DAME;

        if (löse(zeile + 1) == true) {
          return true;
        }
        spielBrett[zeile][i] = LEER;
      }
    }

    return false;
  }

  public static void zeigeSpielBrett() {
    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-Arguemnte
72     */
73    public static void main(String[] args) {
74      fülleFeld();
75      löse(0);
76      zeigeSpielBrett();
77      new Ausgabe(n, spielBrett);
78    }
79  }
```

Code-Beispiel auf Github ansehen:
src/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Damenproblem.java

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

```
33      if (felderAnz % 2 != 0) {
34        for (int i = 0; i < feld.length; i++) {
35          for (int j = 0; j < feld[i].length; j++) {
36            g.setColor(switchColor());
37            if (feld[i][j] == 1) {
38              // Dame
39              g.setColor(Color.ORANGE);
40              g.fillRect(xPos, yPos, feldGroese, feldGroese);
41            } else {
42              // Normalfeld
43              g.fillRect(xPos, yPos, feldGroese, feldGroese);
44            }
45            xPos += feldGroese;
46          }
47          xPos = fensterRand;
48          yPos += feldGroese;
49        }
50      } else {
51        for (int i = 0; i < feld.length; i++) {
52          for (int j = 0; j < feld[i].length; j++) {
53            if (feld[i][j] == 1) {
54              // Dame
55              g.setColor(Color.ORANGE);
56              g.fillRect(xPos, yPos, feldGroese, feldGroese);
57            } else {
58              // Normalfeld
59              g.fillRect(xPos, yPos, feldGroese, feldGroese);
60            }
61
62            xPos += feldGroese;
63            g.setColor(switchColor());
64          }
65          g.setColor(switchColor());
66          xPos = fensterRand;
67          yPos += feldGroese;
68        }
69      }
70    }
71
72    private Color switchColor() {
73      if (feldColor == Color.DARK_GRAY) {
74        feldColor = Color.GRAY;
75        return Color.GRAY;
76      } else if (feldColor == Color.GRAY) {
77        feldColor = Color.DARK_GRAY;
78        return Color.DARK_GRAY;
79      } else {
80        System.err.println("Fehler switchColor!");
81        return null;
82      }
83    }
84
85  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Ausgabe.java`