

# Backtracking: Das Damenproblem

*(Damenproblem)*

**Stichwörter:** Backtracking, Implementierung in Java

---

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  $8 \times 8$  Feld.

```
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;  
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Damenproblem.java](https://github.com/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Damenproblem.java)

## Additum

```
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++) {
        for (int j = 0; j < spielBrett[i].length; j++) {
            System.out.print(spielBrett[i][j] + " ");
        }
        System.out.println();
    }
}

public static void fülleFeld() {
    for (int i = 0; i < spielBrett.length; i++) {
        for (int j = 0; j < spielBrett[i].length; j++) {
            spielBrett[i][j] = 0;
        }
    }
}

/**
 * Das Fenster wird irgendwie nicht in Sway WM angezeigt.
 *
 * @param args Kommandozeilen-Argumente
 */
public static void main(String[] args) {
    fülleFeld();
    löse(0);
    zeigeSpielBrett();
    new Ausgabe(n, spielBrett);
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Damenproblem.java](https://github.com/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Damenproblem.java)

```
import javax.swing.*;

import java.awt.Color;
import java.awt.Graphics;

public class Ausgabe extends JPanel {
    JFrame f;
    int[][] feld;
    int feldGroesse;
    final int fensterRand = 50;
    final int fensterGroesse = 500;
    Color feldColor = Color.DARK_GRAY;
    int felderAnz;

    public Ausgabe(int felderAnz, int[][] feld) {
        feldGroesse = fensterGroesse / felderAnz;
        this.felderAnz = felderAnz;
        this.feld = feld;
    }
}
```

```
f = new JFrame("Damenproblem");
f.setSize(fensterGroesse + 2 * fensterRand, fensterGroesse + 2 * fensterRand);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
f.add(this);
}

@Override
public void paint(Graphics g) {
    int xPos = fensterRand;
    int yPos = fensterRand;

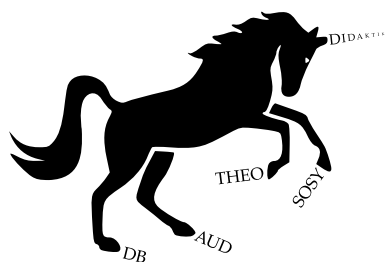
    if (felderAnz % 2 != 0) {
        for (int i = 0; i < feld.length; i++) {
            for (int j = 0; j < feld[i].length; j++) {
                g.setColor(switchColor());
                if (feld[i][j] == 1) {
                    // Dame
                    g.setColor(Color.ORANGE);
                    g.fillRect(xPos, yPos, feldGroesse, feldGroesse);
                } else {
                    // Normalfeld
                    g.fillRect(xPos, yPos, feldGroesse, feldGroesse);
                }
                xPos += feldGroesse;
            }
            xPos = fensterRand;
            yPos += feldGroesse;
        }
    } else {
        for (int i = 0; i < feld.length; i++) {
            for (int j = 0; j < feld[i].length; j++) {
                if (feld[i][j] == 1) {
                    // Dame
                    g.setColor(Color.ORANGE);
                    g.fillRect(xPos, yPos, feldGroesse, feldGroesse);
                } else {
                    // Normalfeld
                    g.fillRect(xPos, yPos, feldGroesse, feldGroesse);
                }

                xPos += feldGroesse;
                g.setColor(switchColor());
            }
            g.setColor(switchColor());
            xPos = fensterRand;
            yPos += feldGroesse;
        }
    }
}

private Color switchColor() {
    if (feldColor == Color.DARK_GRAY) {
        feldColor = Color.GRAY;
    }
}
```

```
        return Color.GRAY;
    } else if (feldColor == Color.GRAY) {
        feldColor = Color.DARK_GRAY;
        return Color.DARK_GRAY;
    } else {
        System.err.println("Fehler switchColor!");
        return null;
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Ausgabe.java](https://github.com/bschlangaul/aufgaben/blob/main/java/org/bschlangaul/aufgaben/aud/muster/backtracking/damenproblem/Ausgabe.java)



## Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: [https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/30\\_AUD/60\\_Algorithmenmuster/50\\_Backtracking/Aufgabe\\_Damenproblem.tex](https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/30_AUD/60_Algorithmenmuster/50_Backtracking/Aufgabe_Damenproblem.tex)