

## Java-Implementierung von FindeLoesung

### zum n Damen Problem:

```
// Konstante: dame[Spalte] = Zeile = UNDEF, wenn die Spalte keine Dame hat
final int UNDEF = -1;

boolean FindeLoesung(int spalte, DameLsg loesung) {
    // spalte = aktuelle Schrittzahl und zugleich aktuelle Spalte
    // loesung = Referenz auf bisherige Teil-Lösung
    int zeile = -1;

    // while(es gibt es noch neue Teil-Lösungsschritte)
    while (zeile != n-1) {
        // Wähle einen neuen Teil-Lösungsschritt schritt;
        zeile++; // Zeilen von oben nach unten der Reihe nach ausprobieren

        // if (schritt ist gültig),
        // d.h. Dame (zeile, spalte) wird nicht bedroht
        if (loesung.z[zeile] && loesung.d1[zeile+spalte] &&
            loesung.d2[zeile-spalte+n-1]) {
            // Erweitere loesung um schritt
            // Platziere Dame auf (zeile, spalte)
            loesung.dame[spalte] = zeile;
            loesung.z[zeile] = false;
            loesung.d1[zeile + spalte] = false;
            loesung.d2[zeile - spalte + n-1] = false;

            // if (loesung noch nicht vollständig)
            if (spalte != n-1) {
                // rekursiver Aufruf von FindeLoesung
                if (FindeLoesung(spalte+1, loesung)) {
                    // Lösung gefunden
                    return true;
                } else {
                    // Wir sind in einer Sackgasse:
                    // Mache schritt rückgängig: Backtracking
                    loesung.dame[spalte] = UNDEF;
                    loesung.z[zeile] = true;
                    loesung.d1[zeile + spalte] = true;
                    loesung.d2[zeile - spalte + n-1] = true;
                }
            } else return true; // Lösung gefunden -> fertig
        }
    }
    return false; // keine Lösung gefunden
}
```