

## Java-Implementierung von FindeLoesung:

```
// Backtracking Algorithmus zum Finden eines Weges in einem Labyrinth

final static int LEER = 0, SACKGASSE = -1; // Feldinformation
final static int[] STEPX = { 0, 1, 0,-1 }; // STEPX,-Y: Schritte
final static int[] STEPY = { -1, 0, 1, 0 }; // in alle Richtungen

boolean FindeLoesung(int index, Lsg loesung, int aktX, int aktY) {
    // index = aktuelle Schrittzahl
    // loesung = Referenz auf bisherige Teil-Lösung
    // aktX, aktY = aktuelle Feldposition im Labyrinth
    int schritt = -1;

    // while(es gibt es noch neue Teil-Lösungsschritte)
    while (schritt != LINKS) {
        // Wähle einen neuen Teil-Lösungsschritt schritt;
        schritt++; // Weg nach 1. oben, 2. rechts, 3. unten
                  // und 4. links zu erweitern versuchen.

        int neuX = aktX + STEPX[schritt];
        int neuY = aktY + STEPY[schritt];

        // Tests, ob schritt gültig ist

        boolean ok = true;
        // Test, ob schritt innerhalb Brett bleibt
        if (neuX < 0 || neuX >= K) ok = false;
        if (neuY < 0 || neuY >= L) ok = false;

        // Test, ob schritt durch Wand führt (sofern innerhalb)
        if (ok && hatWand(aktX,aktY,neuX,neuY,schritt)) ok = false;

        // Test, ob schritt auf ein bereits besuchtes Feld führt
        if (ok && loesung.feld[neuX][neuY] != LEER) ok = false;

        // if (schritt ist gültig)
        if (ok) {
            // Erweitere loesung um schritt
            // Markiere neues Feld mit aktueller Schrittzahl
            loesung.feld[neuX][neuY] = index;

            // if (loesung noch nicht vollständig)
            if (!ausgangGefunden(neuX, neuY) {
                // rekursiver Aufruf von FindeLoesung
                if (FindeLoesung(index+1, loesung, neuX, neuY)) {
                    // Lösung gefunden
                    return true;
                } else {
                    // Wir sind in einer Sackgasse:
                    // Mache schritt rückgängig: Backtracking
                    loesung.feld[neuX][neuY] = SACKGASSE;
                }
            } else return true; // Lösung gefunden -> fertig
        }
    }
    return false; // keine Lösung gefunden
}
// Bei true als Rückgabewert steht die Lösung in loesung
```