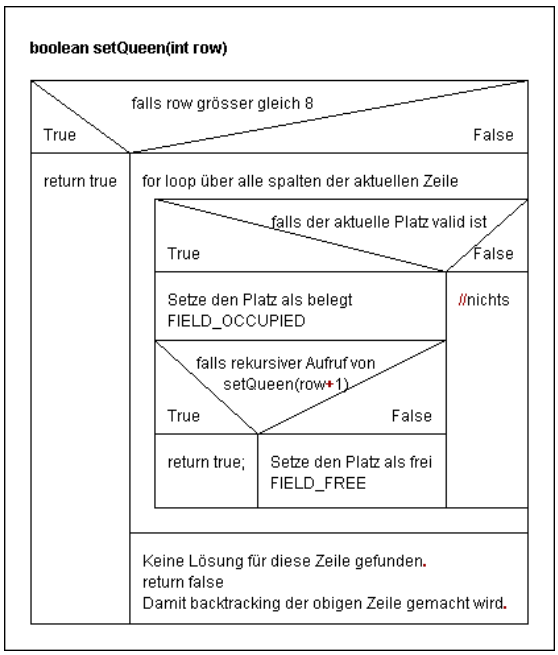
Damen-Problem

Lösung bis Schritt 4 (alles)

Der Unterschied zum Schritt 3 ist das Umsetzen des in der Aufgabe beschriebenen Struktogrammes für die Methode setQueen()




```java
package ch.bbw.pr.dame;

/**
 * Dame Applicatioin
 *
 * @author Peter Rutschmann
 * @version 25.01.2018
 */
public class Application {

    public static void main(String[] args) {
        int size = 8;
        DameProblem solver = new DameProblem(size);

        System.out.println("Damen Problem");
        System.out.println();

        //Start mit Zeile 0
        if (solver.setQueen(0))
        {
            //Printout des Spielfeldes
            for (int i = 0; i < size; i++)
            {
                for (int j = 0; j < size; j++)
                {
                    if (solver.getBoard()[i][j] == 1)
                    {
                        System.out.print("Q ");
                    }
                    else
                    {
                        System.out.print("* ");
                    }
                }
                System.out.println();
            }
        }
    }
}
```

```java
package ch.bbw.pr.dame;

/**
 * Dame Data-Class
 *
 * @author Peter Rutschmann
 * @version 25.01.2018
 */
public class DameProblem {
    private static final int FIELD_FREE = 0;
    private static final int FIELD_OCCUPIED = 1;

    private int size;
    private int[][] board;

    public int[][] getBoard() {
        return board;
    }

    public DameProblem(int size) {
        super();
        this.size = size;
        this.board = new int[size][size];
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                board[i][j] = FIELD_FREE;
            }
        }
    }

    public boolean setQueen(int row)
    {
        if (row >= size)
        {
            return true;
        }
        for (int column = 0; column < size; column++)
        {
            if (isValid(row, column))
            {
                board[row][column] = FIELD_OCCUPIED;
                if (setQueen(row + 1))
                {
                    return true;
                }
                else
                {
                    board[row][column] = FIELD_FREE;
                }
            }
        }
        return false;
    }
```

```
54
55      private boolean isValid(int r, int c) {
56          int i, j;
57          /* Suche in der gleichen Spalte oberhalb
58           * ob es eine Dame hat
59           */
60          for (i = 0; i < r; i++)
61          {
62              if (board[i][c] == FIELD_OCCUPIED)
63              {
64                  return false;
65              }
66          }
67          //Suche Diagonal nach links oben
68          i = r - 1;
69          j = c - 1;
70          while ((i >= 0) && (j >= 0))
71          {
72              if (board[i--][j--] == FIELD_OCCUPIED)
73              {
74                  return false;
75              }
76          }
77          //Suche nach rechts oben
78          i = r - 1;
79          j = c + 1;
80          while ((i >= 0) && (j < size))
81          {
82              if (board[i--][j++] == FIELD_OCCUPIED)
83              {
84                  return false;
85              }
86          }
87          return true;
88      }
89  }
```