Damen-Problem

Lösung bis Schritt 3 mit isValid noch ohne Backtracking

```
Damen Problem

Queen isValid: 0 0:true
Queen isValid: 1 0:false
Queen isValid: 2 0:false
Queen isValid: 3 0:false
Queen isValid: 4 0:false
Queen isValid: 5 0:false
Queen isValid: 6 0:false
Queen isValid: 7 0:false
Q * * * * * * *
Q * * * * * * *
Q * * * * * * *
Q * * * * * * *
Q * * * * * * *
Q * * * * * * *
Q * * * * * * *
Q * * * * * * *
```

isValid wird nur mal aufgerufen und das Resultat wird auf die Konsole ausgegeben.
Noch wird kein Backtracking gemacht!!
➔ Das macht es einfacher, die Methode isValid zu prüfen

```
52    public boolean setQueen(int row)
53    {
54        if (row >= size)
55        {
56            return true;
57        }
58        for (int column = 0; column < size; column++)
59        {
60            board[row][column] = FIELD_OCCUPIED;
61            System.out.println("Queen isValid: " + row + " " + column + ":" +  isValid(row, column));
62
63            if (setQueen(row + 1))
64            {
65                return true;
66            }
67        }
68        return false;
69    }
```

```
95      private boolean isValid(int r, int c) {
96          int i, j;
97          /* Suche in der gleichen Spalte oberhalb
98           * ob es eine Dame hat
99           */
100         for (i = 0; i < r; i++)
101         {
102             if (board[i][c] == FIELD_OCCUPIED)
103             {
104                 return false;
105             }
106         }
107         //Suche Diagonal nach links oben
108         i = r - 1;
109         j = c - 1;
110         while ((i >= 0) && (j >= 0))
111         {
112             if (board[i--][j--] == FIELD_OCCUPIED)
113             {
114                 return false;
115             }
116         }
117         //Suche nach rechts oben
118         i = r - 1;
119         j = c + 1;
120         while ((i >= 0) && (j < size))
121         {
122             if (board[i--][j++] == FIELD_OCCUPIED)
123             {
124                 return false;
125             }
126         }
127         return true;
128     }
129 }
```