

SEW-Inhalte

Arrays

```
private int[][] solution;
```

```
private Button[][] board = new Button[9][9];
```

```
int[][] board = new int[9][9];
```

```
fillBoard(board);
```

```
for (int row = 0; row < 9; row++) {
```

```
    for (int col = 0; col < 9; col++) {
```

```
        ...
```

```
    }
```

```
}
```

Collections

```
List<Integer> numbers = new ArrayList<>();
```

```
for (int i = 1; i <= 9; i++) {
```

```
    numbers.add(i);
```

```
}
```

```
Collections.shuffle(numbers);
```

Enum

```
enum difficulty {
```

```
    EASY, MEDIUM, HARD
```

```
}
```

Verschachtelte Schleifen

```
for (int blockRow = 0; blockRow < 3; blockRow++) {
```

```
    for (int blockCol = 0; blockCol < 3; blockCol++) {
```

```
        GridPane block = new GridPane();
```

Nico Kliche, Henry Koran

```
...
    for (int row = 0; row < 3; row++) {
        for (int col = 0; col < 3; col++) {
            int absoluteRow = blockRow * 3 + row;
            int absoluteCol = blockCol * 3 + col;
            ...
        }
    }
}
```

Sudoku Algorithmus

```
for (int row = 0; row < 9; row++) {
    for (int col = 0; col < 9; col++) {
        if (board[row][col] == 0) {
            List<Integer> numbers = new ArrayList<>();
            ...
            for (int number : numbers) {
                if (isValid(board, row, col, number)) {
                    ...
                }
            }
            if (fillBoard(board)) {
                return true;
            }
        }
    }
}
```

Sonstiges

Style-Guide mit Checkstyle

- Durchgehend sprechende Variablennamen (selectedCell, fehlerCounter).
- Methodennamen im camelCase.
- Einrückung & Leerzeilen konform.

Einfachheit

- UI-Logik und Spiellogik sind gut getrennt.
- Komponenten wie VBox, HBox, GridPane werden gezielt verwendet.

Laufzeiteffizienz

- Sudoku-Board wird nur einmal vollständig generiert, Lösung wird gecached (solution = deepCopy(board)).
- Rekursion wird abgebrochen, sobald ein gültiges Board generiert wurde (return true beim ersten Treffer).

Zusatzfeatures

- UI-Zellen erhalten visuelles Feedback (Farbe grün/rot).
- Fehlermeldung und Endbildschirm mit Restart-Funktion.
- Schwierigkeitsgrad-Auswahl vor Spielstart.
- Highlighting der ausgewählten Zelle.