

Lesbares Java

Steffen Gemkow

Zur Person

- Steffen Gemkow
 - Geschäftsführer
 - Java seit 1996
 - Agile Coach, Scrum-Master
 - JUnitDoclet, TheWayOut
 - Der Pragmatische Programmierer



Wartbarkeit

- Funktion von
 - **Lesbarkeit**
 - Testbarkeit
 - Architektur
 - Entwickler
 - ...

Paradox

- Computer „versteht“ Quelltext nicht
 - Schuld des Programmierers
- Leser versteht Quelltext nicht
 - Schuld Lesers?

Lesbarkeit, was ist das?

- Gruppierung (Leerzeichen und -zeilen)
- Vertrauen (in Bezeichner und Algorithmen)
- Konsistente Namenswelt
- Konstante „Flughöhe“

Gestaltpsychologie

- Prägnanz
- Nähe
- Ähnlichkeit
- Kontinuität
- Geschlossenheit
- gemeinsame Bewegung
- fortgesetzt durchgehende Linie
- gemeinsame Region
- Gleichzeitigkeit
- verbundene Elemente

Bjarne sagt

Don't be clever!

Bjarne Stroustrup (C++)



Quelle: Wikipedia

Java 2012

(geboren 1995)

- Eine Programmiersprache wird erwachsen
 - Unschuld verloren
 - viel probiert, nicht alles war gut
 - bereits ein paar Narben
 - Leistungsfähig und auf dem Zenit

Symptome

- Lange Namen Weil Man Nicht Ueber Kurze Namen Nachdenken Will
- Generics-Missbrauch
- Voreilige „Optimierung“
 - lange Methoden
 - Variablen inline definiert

Hier

```
for (int i=0; i<SIZE; i++) {  
    VeryLongClassName evenLongerVariableName = getSomeValue();  
}
```

Nochmal: **Wo** passiert etwas?

```
int main() {  
    int x = 0;  
    while (x < 10) {  
        x = x + 1;  
    }  
}
```

Hier

```
VeryLongClassName someValue;  
  
for (int i=0; i<SIZE; i++) {  
    someValue = getSomeValue();  
}
```

Was war das denn?

- Ein Bild sagt mehr...
- Deklarationen herausziehen
- Kurze Bezeichner

Oder auch so

```
protected class DependenciesPageRole playedBy DependenciesPage {  
  
    callin String[] getRequiredSectionLabels() {  
        List<String> labels = Arrays.asList(base.getRequiredSectionLabels());  
        List<String> newLabels = new ArrayList<String>(labels);  
        newLabels.add("Remove versions");  
        return (String[]) newLabels.toArray(new String[newLabels.size()]);  
    }  
  
    // ...  
}
```

bzw. so

```
protected class DependenciesPageRole playedBy DependenciesPage {  
  
    callin String[] getRequiredSectionLabels() {  
        String[]    result;  
        List<String> labels;  
        List<String> newLabels;  
  
        labels      = Arrays.asList(base.getRequiredSectionLabels());  
        newLabels   = new ArrayList<String>(labels);  
  
        newLabels.add("Remove versions");  
  
        result      = (String[]) newLabels.toArray(new String[newLabels.size()]);  
  
        return result;  
    }  
  
    // ...  
}
```

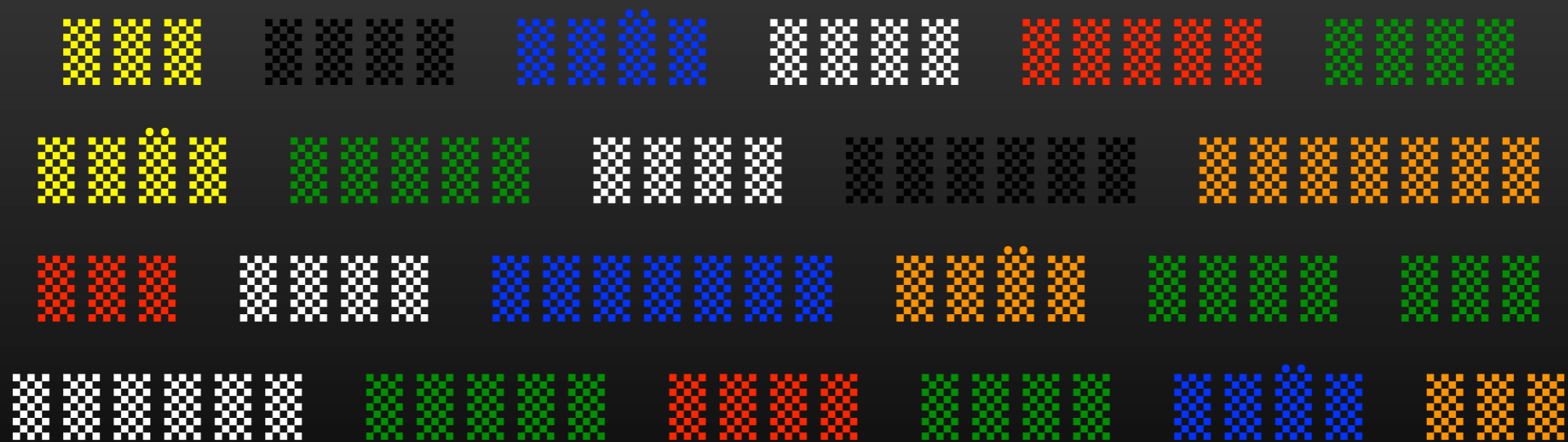
Funktionsaufbau

```
protected class DependenciesPageRole playedBy DependenciesPage {  
  
    callin String[] getRequiredSectionLabels() {  
        String[]    result;  
        List<String> labels;  
        List<String> newLabels;  
  
        labels      = Arrays.asList(base.getRequiredSectionLabels());  
        newLabels   = new ArrayList<String>(labels);  
  
        newLabels.add("Remove versions");  
  
        result      = (String[]) newLabels.toArray(new String[newLabels.size()]);  
  
        return result;  
    }  
  
    // ...  
}
```

Aber <some_excuse>

- ... weniger Zeilen sind besser.
 - **Nein.**
- ... mein Eclipse formatiert anders.
 - Dann formatiere mit **JIndent.**
- ... kleine Scopes sind besser.
 - Stimmt, also schreibe **kleine Methoden.**

Experiment



Experiment

ROT BLAU GRÜN GELB WEISS BLAU
GRÜN WEISS GELB ORANGE SCHWARZ
ROT GELB SCHWARZ GRÜN BLAU ROT
ORANGE WEISS GELB BLAU GRÜN ROT

Kommentare

```
/**  
 * The method setValue sets the value.  
 * @param value    the value to be set  
 */  
public void setValue(int value) {  
    // ...  
}
```

Kommentare

```
/**  
 * The method setValue sets the value.  
 * @param value the value to be set  
 */  
public void setValue(int value) {  
 // ...
```

Kommentare

Kommentiere das Warum,
nicht das Wie.

Vertrauen schaffen

```
if (condition) {  
    doSomething();  
}
```

Vertrauen schaffen

```
if (condition) {  
    doSomething();  
} // no else
```

Vertrauen schaffen

```
switch(index) {  
  case ONE:  
  case TWO:  
    handleFew();  
    break;  
  case THREE:  
    handleMany();  
}
```

Vertrauen schaffen

```
switch(index) {  
  case ONE: // fall through  
  case TWO:  
    handleFew();  
    break;  
  case THREE:  
    handleMany();  
    break;  
  default:  
    // just ignore other cases  
}
```


flüssig vs. überflüssig

```
if (variable>CONST) {  
    result = true;  
} else {  
    result = false;  
}  
  
return result;
```

Vereinfachen

```
if (condition) {  
  
} else {  
    doSomething();  
}
```

Flughöhe

```
void doSequenceOnAll(List list) {  
  
    // TODO Parameterpruefung  
    initAll(list);  
    calculateAll(list);  
    for(Object o:list) {  
        System.out.println(o);  
    }  
}
```

Lesen vs. Schreiben

*The ratio of the time spent reading
to the time spent writing is 10:1.*

Robert C. Martin

Was kann ich tun?

- CodeReview by Peers
- CodingStyle
- CheckStyle
 - ggf. weitergehende (3rdParty) Checks
 - IfWithElseCheck
 - AvoidInlineDeclarationCheck

Wir tun es für uns!

*Wir programmieren heute
die Legacy-Systeme von morgen.*

Quältext?

Schreibe fürs Lesen!

Danke

info@objectfab.de

www.objectfab.de