

Clean Code: Kommentare

„The proper use of comments is to compensate for our failure to express ourself in code. Note that I used the word *failure*. I meant it. Comments are always failures. We must have them because we cannot always figure out how to express ourselves without them, but their use is not a cause for celebration.”

“So when you find yourself in a position where you need to write a comment, think it through and see whether there isn’t some way to turn the tables and express yourself in code. Every time you express yourself in code, you should pat yourself on the back. Every time you write a comment, you should grimace and feel the failure of your ability of expression.“



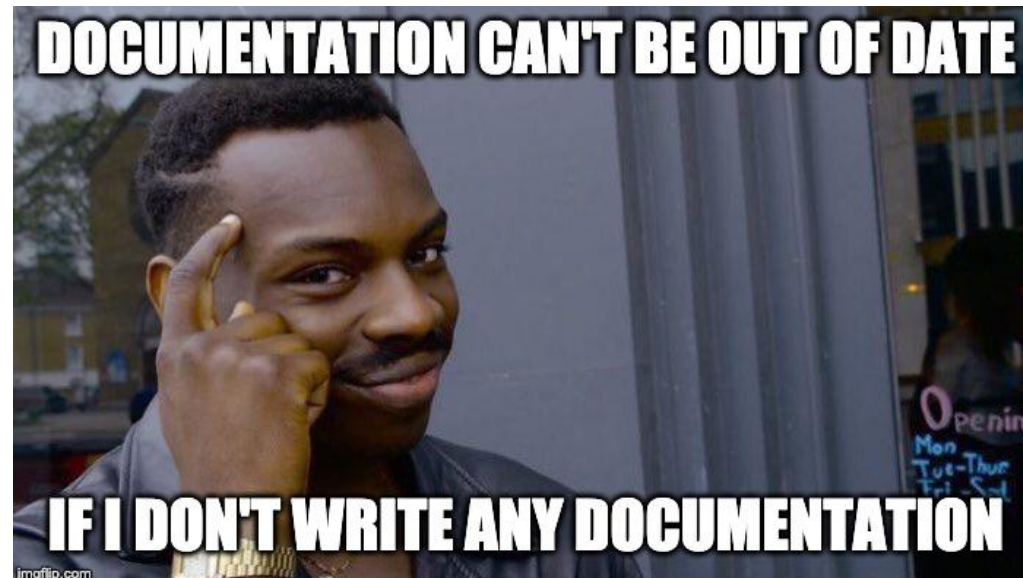
Comments lie. Not always, and not intentionally, but too often.



The older a comment is, and the farther away it is from the code it describes, the more likely it is to be just plain wrong.



The reason is simple. Programmers can't realistically maintain them.



- We write a module and we know it is confusing and disorganized. We know it's a mess. So we say to ourselves, "Ooh, I'd better comment that!"

No! You'd better clean it!

- Clear and expressive code with few comments is far superior to cluttered and complex code with lots of comments.
- Rather than spend your time writing the comments that explain the mess you've made, spend it cleaning that mess.

```
//  
// Dear Programmer:  
//  
// When I wrote this code, only God and I knew how it worked.  
// Now, only God knows!  
// So if you are trying to optimize this routine and fail,  
// (very likely) please increase the following counter  
// as a warning to the next developer:  
//  
// total_hours_lost_here = 218
```

```
// Check to see if the employee is eligible for full benefits  
if ((employee.flags & HOURLY_FLAG) && (employee.age > 65)) {
```

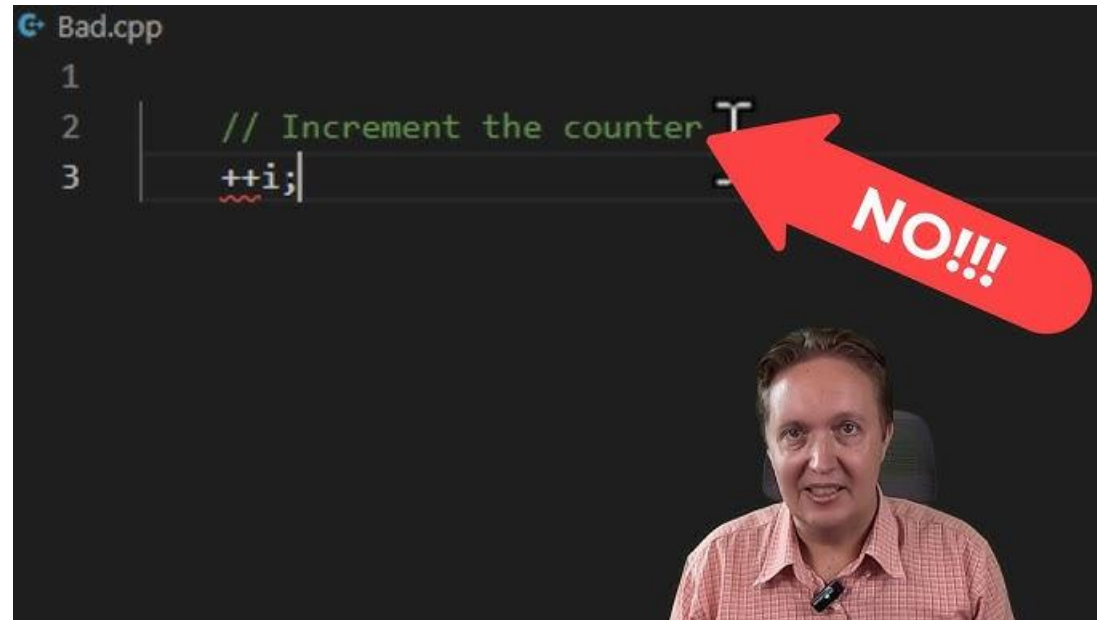
oder

```
if (employee.isEligibleForFullBenefits()) {
```

Maßnahmen, um Kommentare zu vermeiden

Self-documenting code

Auf Naming von Variablen und Klassen achten



Warum wurde etwas gemacht?

```
var uid = readByte();
if (uid == 0) {
    // Somehow the battery sends 0x00 byte at the beginning of F42 responses - we need to skip that byte
    uid = readByte();
}
```

Javadocs für Public APIs

Bei Libraries sind Javadocs wichtig

// TODO Kommentare

```
return this.timeDataQueryDebugExecutor.debugMetrics().entrySet().stream() //
    .collect(toUnmodifiableMap(//
        // TODO implement getId()
        e -> ID + "/" + e.getKey(), //
        e -> new JsonPrimitive(e.getValue())));
```

Bedeutung verstärken

```
public void ReadCsv(CsvReader reader) {
    // The trim is really important. The provider is sending spaces after csv determinator
    var username = reader.get(1).Trim();
}
```

Schlechte Kommentare

```
/**  
 * Duration in minutes  
 */  
private int duration;
```



```
private int durationInMinutes;  
oder:  
private TimeSpan duration;
```

```
/**  
 * Gets the update meta information.  
 * @return the meta information  
 */  
UpdateableMetaInfo getMetaInfo();
```



```
Updateable.UpdateableMetaInfo  
getMetaInfo();
```

```

var areSurplusConditionsMet = this.areSurplusConditionsMet(this.parent, chargers, config);
var offTime = LocalTime.parse(config.surplusFeedInOffTime());

if (chargers.isEmpty()) {
    // Is no Charger set (i.e. is this not a Commercial 40-40 "DC")
    this.setState(SurplusFeedInStateMachine.DEACTIVATED);
} else if (LocalTime.now(componentManager.getClock()).isAfter(offTime)) {
    // Passed surplusOffTime?
    this.setState(SurplusFeedInStateMachine.PASSED_OFF_TIME);
} else if (areSurplusConditionsMet) {
    // Always immediately activate surplus-feed-in if conditions are met
    this.setState(SurplusFeedInStateMachine.ACTIVATED);
} else if (this.state == SurplusFeedInStateMachine.UNDEFINED) {
    this.setState(SurplusFeedInStateMachine.DEACTIVATED);
}

```



```

var areSurplusConditionsMet = this.areSurplusConditionsMet(this.parent, chargers, config);

if (chargers.isEmpty()) {
    // Mostly the case if this is not a Commercial 40-40 "DC"
    this.setState(SurplusFeedInStateMachine.DEACTIVATED);
} else if (this.hasPassedSurplusOffTime(config, componentManager)) {
    this.setState(SurplusFeedInStateMachine.PASSED_OFF_TIME);
} else if (areSurplusConditionsMet) {
    this.setState(SurplusFeedInStateMachine.ACTIVATED);
} else if (this.state == SurplusFeedInStateMachine.UNDEFINED) {
    this.setState(SurplusFeedInStateMachine.DEACTIVATED);
}

public boolean hasPassedSurplusOffTime(Config config, ComponentManager componentManager) {
    var offTime = LocalTime.parse(config.surplusFeedInOffTime());
    return LocalTime.now(componentManager.getClock()).isAfter(offTime);
}

```


Aktuelle Statistik

- Java Source Code Lines: 402.072
- Java Comment Lines: 141.623
- => Auf jede dritte Codezeile folgt eine Kommentarzeile



Passen wir die Checkstyle Regeln an?