

REVIEW-VORLAGE

REVIEW DER KLASSE [Klassenname]

Auto der Klasse: [Auto]

Gutachter: [Gutachter]

Revision Beginn: [Revision am Anfang des Reviews]

Abgeschlossen: Ja [] Nein [x]

Revision Ende: [Revision am Ende des Reviews]

REVIEW-PHASE

1. Variablen, Attribute, Methoden und Klassen

- Es gibt Attribute, Methoden oder Klassen, die nicht ausreichend sprechende Namen besitzen (dazu zählt auch die sinnvolle und korrekte Verwendung der Camel-Case-Schreibweise). [jeweils entsprechend ergänzen und mit Hilfe von REVIEW_Autorname und Nummer (1), (2), ... im Quellcode darauf verweisen]
- Es gibt nicht-lokale Variablen, die besser lokale Variablen wären.
- Es gibt statische Variablen oder Methoden, die besser nicht statisch sein sollten.
- Es gibt nicht-statische Variablen oder Methoden, die besser statisch sein sollten.
- Es existieren überflüssige Attribute oder Methoden (z.B. Getter- und Setter-Methoden)
- Es gibt Attribute, Klassen oder Methoden, dessen Sichtbarkeit anders sein sollte.
- Es gibt Methoden oder Attribute, die möglicherweise besser in der Super-Klasse aufgehoben sind.

2. Modularitätsprüfungen

- Es existiert Code, der mehrere Male wiederholt wird und deshalb besser in eine Methode ausgelagert werden würde.
- Es existieren Klassen oder Methoden, dessen Funktionalität durch Klassen oder Methoden der Java-Bibliothek ersetzt werden kann.
- Variablen- bzw. Parameter- oder Rückgabe-Typen sind nicht so allgemein wie möglich und so speziell wie nötig gewählt (z.B. Verwendung der konkreten Implementierung "ArrayList" statt des Interfaces "List").

3. Kontrollflussprüfungen

- Es gibt Schleifen, die unter Umständen nicht terminieren.
- Es gibt unnötig tiefe Block-Verschachtelungen.
- Es gibt if-Anweisungen, die besser ein Switch-Statement wären.

4. Laufzeitprüfungen

- Es gibt unnötig häufige Neuberechnungen.

Es gibt ungünstig gewählte Datenstrukturen.

Es gibt Datenstrukturen, die mit mehreren Schleifen hintereinander unnötig häufig durchlaufen und somit zu einer Schleife zusammengefasst werden können.

5. Dokumentation des Quellcodes

Die betrachtete Klasse wurde nicht ausreichend/verständlich dokumentiert. Dazu gehört eine grobe Beschreibung, evtl. Besonderheiten der Implementierung sowie der Name des Autors und falls anwendbar, eine Einordnung im mathematischen Modell.

Es gibt nicht-triviale Methoden, die nicht ausreichend dokumentiert sind. Dazu gehört eine Beschreibung der Methode, Definition der Ein- und Ausgabeparameter sowie deren Definitionsbereichs-Einschränkungen. Mögliche geworfene Exceptions werden erläutert, erwähnte Klassen/Methoden entsprechend verlinkt.

Es gibt nicht-triviale Code-Segmente, die nicht ausreichend dokumentiert sind.

Es waren Rechtschreib-/Tippfehler vorhanden (diese werden sofort vom Gutachter korrigiert und vom Autor im Revision-Compare-Editor zur Kenntnis genommen).

6. Weitere Kommentare des Gutachters

....