Martin Schmidt

Clean Code

Checkliste

* Sind alle Variablen mit Standardwerten initialisiert?
* Wurden nicht benutzte Variablen entfernt?
* Sind redundante Variablen verwendet?
* Sind alle Variablen nach in einem möglich geringem Gültigkeitsbereich definiert?
* Sind die verwendeten globalen Variablen notwendig?
* Ist die Funktion von Variablen am Namen erkennbar?
* Sind Variablen/Konstanten/Funktionen/Klassen/Methoden einheitlich benannt? (Bevorzugt CamelCase, gleiche Sprache)
* Sind Funktionen dokumentiert?
* Werden Fehlerursachen abgefangen und wenn möglich ausgegeben?
* Sind Funktionalitäten in kleine wiederverwendbare Teile gegliedert?
* Werden selbstgeschriebene Grundfunktionalitäten in komplexen Abläufen eingesetzt?
* Sind die Methoden einer Klasse einem homogen Funktionsumfeld gewidmet?
* Werden Code und Design weitest möglich voneinander getrennt?
* Sind nicht benötigte Programmbibliotheken eingebunden?
* Sind temporäre Kopien mit einem führendem \_ gekennzeichnet?
* Sind sinnlose, d.h. nicht der Orientierung dienende, Leerzeichen entfernt?

Die Liste ist sicher bei weitem noch nicht vollständig, aber viele Sachen macht man mittlerweile so unterbewusst richtig oder falsch, dass man gar nicht darauf achtet. Die Liste sollte und wird mit der Zeit weiter wachsen.

Mein Programm erfüllt diese Punkte soweit möglich. Das liegt natürlich auch daran, dass die Checkliste aufgrund des mir angewöhnten Styleguides entstanden ist und das Programm natürlich mit derselben Attitüde entstanden ist. In meinem Arbeitsumfeld ist es in der Regel notwendig, dass auch andere Programmierer sich in meinen Quellcode einarbeiten können, daher lege ich vor allem Wert auf eindeutige Benennung und klare Strukturierung. Platz für Optimierung sowohl style,- als auch performancetechnisch ist fast immer.