

CLEAN CODE - CHEAT SHEET

General	Environment	Understandability	Objects
<ul style="list-style-type: none"> • Keep it simple, stupid (KISS) • Don't repeat yourself (DRY) • Standard Konventionen folgen • Keine Redundanzen • Keinen ungenutzten Code • Principle of least surprise folgen • Frühzeitige Optimierungen vermeiden • Die Pfadfinder-Regel beachten (Code besser hinterlassen als vorgefunden) 	<ul style="list-style-type: none"> • Versionskontrollsysteme verwenden • Buildmanagement Tool verwenden 	<ul style="list-style-type: none"> • Konsistent sein • Erklärende Variablen nutzen • Negative Bedingungen (wie „not positive(x)“) vermeiden 	<ul style="list-style-type: none"> • So klein wie möglich halten • Sollten nur eine Sache machen • Kleine Anzahl von Instanzvariablen verwenden
Design <ul style="list-style-type: none"> • Nicht willkürlich sein • Polymorphismus statt if/else, switch-Anweisungen bevorzugen • Multi-threaded Code nicht mit „normalem“ Code mischen • Code soll „kommunizieren, einfach und flexibel sein“ (Beck) 	Structure	Naming	Comments
	<ul style="list-style-type: none"> • Einheitliche Strukturierung • Zeilen kurz halten • Einheitliche Einrückungen • Lokale Deklaration von Variablen (vor erster Nutzung und geringer Umfang) • Refraction patterns verwenden • Bedingungen kapseln (Methoden statt komplexe Strukturen) • Information Hiding Prinzip folgen 	<ul style="list-style-type: none"> • Präzise, eindeutige & einheitliche Benennungen verwenden • Aussprechbare Namen verwenden • Magic numbers vermeiden 	<ul style="list-style-type: none"> • Keine überflüssigen Kommentare (Kommentare auf Klassen- und Package-Ebene verwenden) • Explain yourself in code (Code selbsterklärend halten) • Kommentare zum: informieren, erklären oder warnen des Lesers • Ungenutzten Code nicht auskommentieren sondern löschen • Eine Kommentiersprache nutzen (z.B. JavaDoc)
	Exception handling	Functions	Dependencies
	<ul style="list-style-type: none"> • Exceptions so spezifisch wie möglich sammeln • Nicht „Null“ zurückgeben 	<ul style="list-style-type: none"> • So klein wie möglich halten • Beschreibende Namen verwenden • Funktionen sollten nur eine Sache machen • Abhängige Funktionen sollten nahe bei einander liegen • Functions weiter unten platzieren 	<ul style="list-style-type: none"> • „Nebenwirkungen“ vermeiden, sicherstellen, dass Änderungen lokale Konsequenzen haben