

CCD Cheat Sheet

1. Klassen sollten kleiner als 100 Zeilen sein
2. KISS
Komplexität soweit wie möglich verringern
3. Konfigurationen stets auf hohen Ebenen platzieren
4. Beschreibende/eindeutige Namen verwenden
5. Namen beschreiben das Level der Abstraktion
6. Funktionen/Methoden sind danach benannt, was sie tun
7. Funktionen haben keine Nebeneffekte
8. Unvermeidbare Nebeneffekte müssen sich im Funktionsnamen widerspiegeln
9. Je größer der Scope, desto länger der Name
10. Keine Typ oder Scope Informationen im Namen
11. Methoden erledigen exakt eine Aufgabe
12. Nutzlose Kommentare und auskommentierter Code wird entfernt
13. "Magische Zahlen" vermeiden
Zahlen für Vergleiche/ IF-Bedingungen/... vorher einer Variablen (mit beschreibendem Namen) zuweisen.
14. Polymorphie anstelle von if/else oder switch/case verwenden (falls möglich)
15. Komplexere IF Bedingungen (z.B. mit &&) sollten durch eine beschreibende Funktion ersetzt werden
16. Negative Bedingungen vermeiden
17. Dont Repeat Yourself
Copy und Paste durch Kapselung und Abstraktion vermeiden
18. Single Level of Abstraction
doing und calling Code wird nicht innerhalb einer Funktion vermischt
19. You Aint Gonna Need It
Funktionalität, die nicht dem Kernziel dient wird entfernt

Der Beispiel Code aus TicTacToe.py ist Clean Code weil:

1. Alle Klassen haben weniger als 100 Zeilen
2. Es wurden Variablen- und Funktionsnamen verwendet, die beschreibend sind (z.B. makeMove, activePlayer, board,...)
3. Die Funktionen sind danach benannt, was sie tun und erfüllen zudem stets nur eine Aufgabe.
4. Keine nutzlosen Kommentare oder aaskommentierter Code
5. Auf Typ oder Scopes in Namen wurde verzichtet
6. Polymorphie anstelle von if/else verwendet (sign_to_printable())
Anstelle von Klassen wurde ein Object verwendet, um zu verhindern, dass die Komplexität steigt. Daher ist der Begriff Polymorphie hier eigentlich nicht richtig und soll nur auf die entsprechende Regel hinweisen.
7. If-Bedingung wurde durch Funktion ersetzt (in makeMove())
8. Negative Bedingungen wurden vermieden, sofern es ging. In Zeile 74 war dies nicht möglich, da sonst die Boolean in der Funktion getauscht werden müssten. Dies würde dazu führen, dass die Funktion an sich schwerer zu verstehen ist. Deshalb wurde darauf verzichtet. Ebenso in 85.
9. Single Level of Abstraction wurde innerhalb der Klassen und Funktionen eingehalten. Lediglich innerhalb der While Schleife (Zeile 74) wurde dies nicht eingehalten. Eine Einhaltung würde den Code deutlich komplexer machen und diverse weitere Funktionen wären nötig. Dadurch wäre das Prinzip KISS verletzt.
10. Es wurde nur Funktionalität hinzugefügt die auch aktiv verwendet wird und dem Kernziel, dem Tic Tag Toe spielen, beiträgt.
11. Die Funktionen print_board und checkWin Widersprechen dem Prinzip "Dont Repeat Yourself". Jedoch waren andere Implementierungsversuche deutlich aufwendiger und komplexer und hätten gegen das Prinzip KISS verstoßen.
12. Magische Zahlen wurden vermieden, indem Beispielsweise die Symbole der Spieler (1 und 2) durch Variablennamen (symbol_x) "entzaubert" wurden.
Die magischen Zahlen in den Funktionen print_board und checkWin wurden beibehalten, da eine Zuweisung zu einer Variablen deutlich aufwendiger wäre und der Informationsgehalt nicht sonderlich steigen würde.