Clean Code Cheatsheet

Fundamentals

Delete useless code/comments ...... its really not needed
Precise naming .... someMethod() does what?
Avoid magic numbers .... constants are better
Prefer polymorphism .... not conditionals
side effects of functions .... avoid them
fields shall define state .... no trashcans
exceptions ... catch specific, react meaningful, fail fast
refactor patterns ..... fowler and IDE are friends

Coding Source

Local declaration ..... kids shouldn't go far away
Explanatory Vars .... easier to follow small steps
Nesting .... deep nested - more specific
Separate Multi-threading code .... from "normal"

Code Conditionals

Encapsulate Conditionals this.shouldBeDeleted(Object o)
Avoid Negative Conditionals .. they are harder to read
Encapsulate Boundary Conditions .... don't hide

Coding principles

Dont repeat yourself .... Don't repeat yourself
Keep it simple, stupid .... others read it too
Code Communication, Simplicity, Flexibility Selbsterklärend für Dritte
Avoid early Optimizations .... it's complicated
Single Level of Abstraction ..... calling vs. doing
Keep configurable Data at High Levels you won't search
Over-Configurability .... defaults will help
Hidden Temporal Coupling Rule .... no fixed call order
Dont be arbitrary .... maintain guidelines

Code Quality

Use constraints .... assert often
Check your conventions FindBugs,PMD and style
Unit Testing .... Test, Test, Test! Always!
Code Coverage .... 80 % is a good goal
Test first ..... better quality nad power of CI/CD

Avoid

Too many arguments for functions Needless complexity

IMPORTANT

"Leave the campground cleaner than you found it!"

Links and information

`http://martinfowler.com/refactoring/catalog/`
`https://www.planetgeek.ch/wp-content/uploads/2014/11/Clean-Code-V2.4.pdf`

Denis Renning, BHT 914556
`http://devtty.de/`