CleanCode : Smells and heuristics

List include many Martin Fowler (book "Refactoring") smell and adds many more

- **Comments**
  - C1: Inappropriate Information
  - C2: Obsolete Comment
  - C3: Redundant Comment
  - C4: Poorly Written Comment
  - C5: Commented-Out Code
- **Environment**
  - E1: Build Requires More Than One Step
  - E2: Tests Require More Than One Step
- **Functions**
  - F1: Too Many Arguments
  - F2: Output Arguments
  - F3: Flag Arguments
  - F4: Dead Function
- **Java**
  - J1: Avoid Long Import Lists by Using Wildcards
  - J2: Don't Inherit Constants
  - J3: Constants versus Enums
- **Names**
  - N1: Choose Descriptive Names
  - N2: Choose Names at the Appropriate Level of Abstraction
  - N3: Use Standard Nomenclature Where Possible
  - N4: Unambiguous Names
  - N5: Use Long Names for Long Scopes
  - N6: Avoid Encodings
  - N7: Names Should Describe Side-Effects
- **Tests**
  - T1: Insufficient Tests
  - T2: Use a Coverage Tool!
  - T3: Don't Skip Trivial Tests
  - T4: An Ignored Test Is a Question about an Ambiguity
  - T5: Test Boundary Conditions
  - T6: Exhaustively Test Near Bugs
  - T7: Patterns of Failure Are Revealing
  - T8: Test Coverage Patterns Can Be Revealing
  - T9: Tests Should Be Fast
- **General**
  - G1: Multiple Languages in One Source File
  - G2: Obvious Behavior Is Unimplemented
  - G3: Incorrect Behavior at the Boundaries
  - G4: Overridden Safeties
  - G5: Duplication
  - G6: Code at Wrong Level of Abstraction
  - G7: Base Classes Depending on Their Derivatives
  - G8: Too Much Information
  - G9: Dead Code
  - G10: Vertical Separation
  - G11: Inconsistency
  - G12: Clutter
  - G13: Artificial Coupling
  - G14: Feature Envy
  - G15: Selector Arguments
  - G16: Obscured Intent
  - G17: Misplaced Responsibility
  - G18: Inappropriate Static
  - G19: Use Explanatory Variables
  - G20: Function Names Should Say What They Do
  - G21: Understand the Algorithm
  - G22: Make Logical Dependencies Physical
  - G23: Prefer Polymorphism to If/Else or Switch/Case
  - G24: Follow Standard Conventions
  - G25: Replace Magic Numbers with Named Constants
  - G26: Be Precise
  - G27: Structure over Convention
  - G28: Encapsulate Conditionals
  - G29: Avoid Negative Conditionals
  - G30: Functions Should Do One Thing
  - G31: Hidden Temporal Couplings
  - G32: Don't Be Arbitrary
  - G33: Encapsulate Boundary Conditions
  - G34: Functions Should Descend Only One Level of Abstraction
  - G35: Keep Configurable Data at High Levels
  - G36: Avoid Transitive Navigation