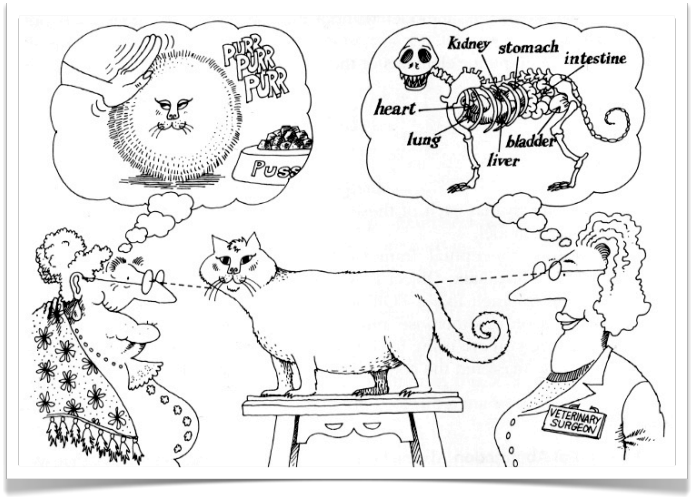
Principles

of object-oriented design

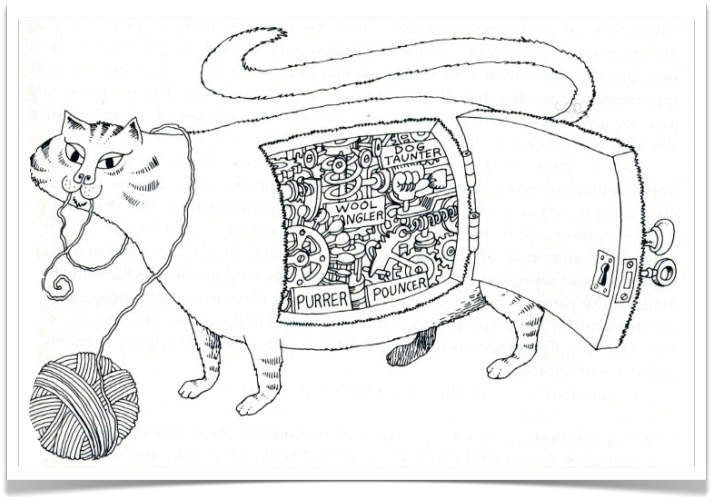
- Abstraction – Hide details
- Encapsulation – Keep changes local
- Modularity – Control information flow
  High cohesion • weak coupling • talk only to friends
- Hierarchy – Order abstractions
  Classes open for extensions, closed for changes • Subclasses that do not require more or deliver less • depend only on abstractions

Goal: *Maintainability* and *Reusability*

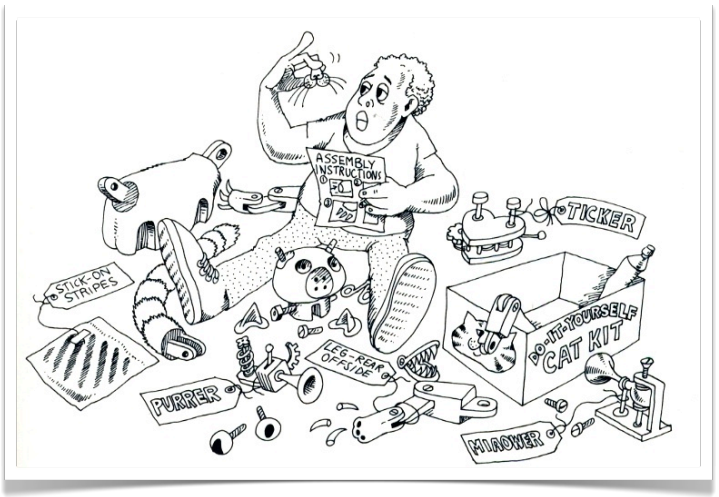
Abstraction



Encapsulation



Modularity



Principles of Modularity

- High cohesion – Modules should contain functions that logically belong together
- Weak coupling –Changes to modules should not affect other modules
- Law of Demeter – talk only to friends

Call your Friends

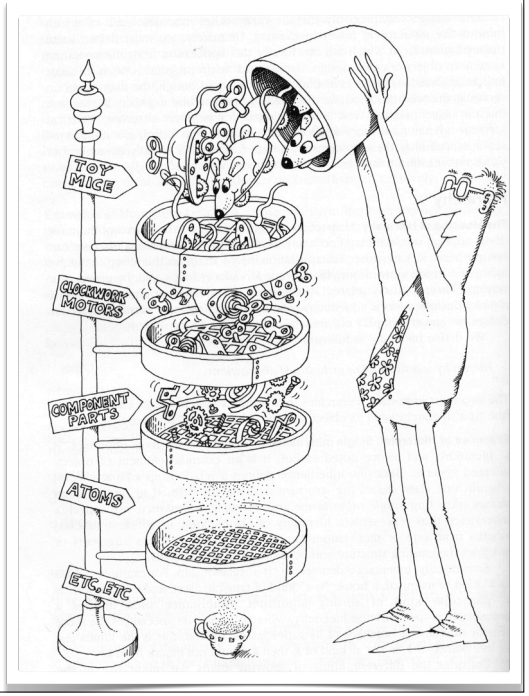
A method M of an object O should only call methods of

1. O itself
2. M's parameters
3. any objects created in M
4. O's direct component objects

"single dot rule"

Hierarchy

"Hierarchy is a ranking or ordering of abstractions."



Hierarchy principles

- Open/Close principle – Classes should be open for extensions
- Liskov principle – Subclasses should not require more, and not deliver less
- Dependency principle – Classes should only depend on abstractions