Tampereen yliopisto
Tampere University

TIE-02408 Programming 3: Techniques

28.8.

Implementation of a large program

Fig: KSI Photography
(CC BY-NC-ND 2.0)

Fig: Team Lupapiste (Solita Facebook publication)

Fig: Kanban board

A large program?

Tampereen yliopisto
Tampere University

What is difficult in implementing a (large) program?

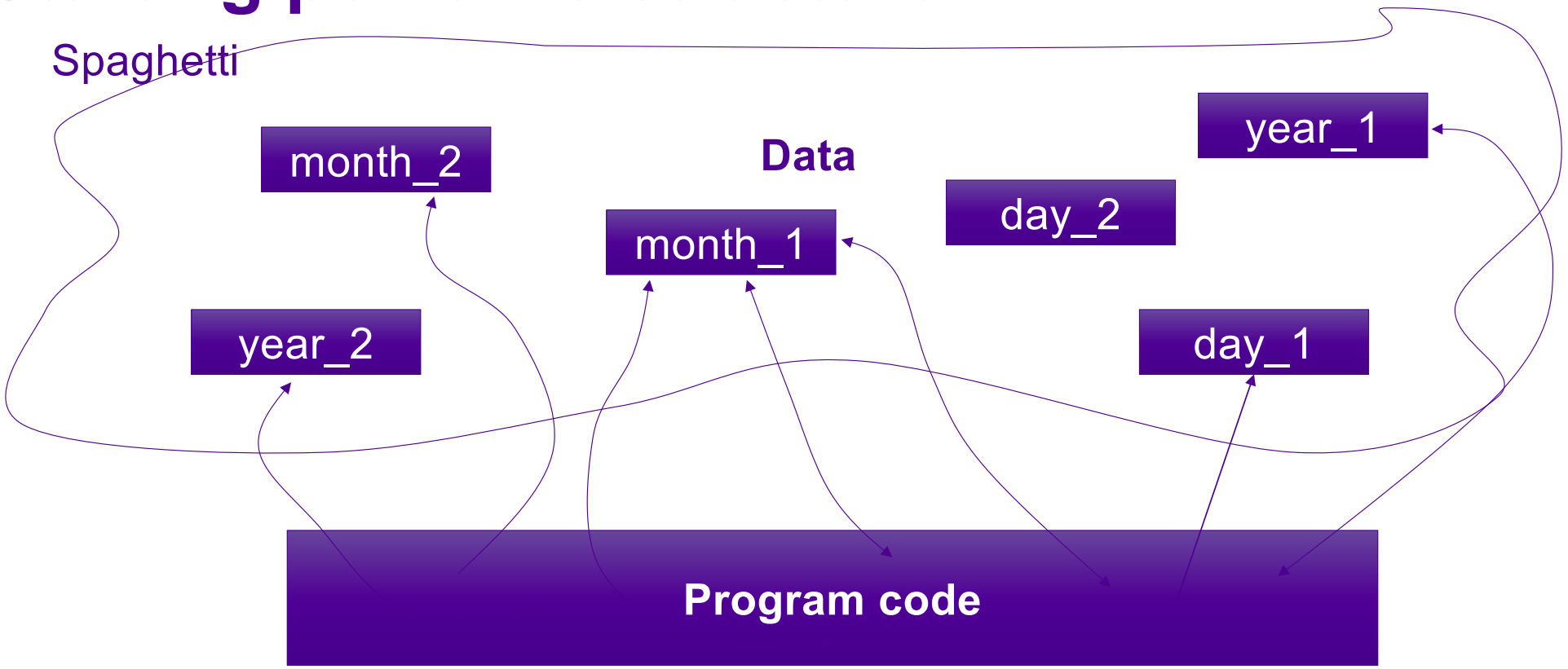
"as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem." - Edsger W. Dijkstra

Sectioning a program

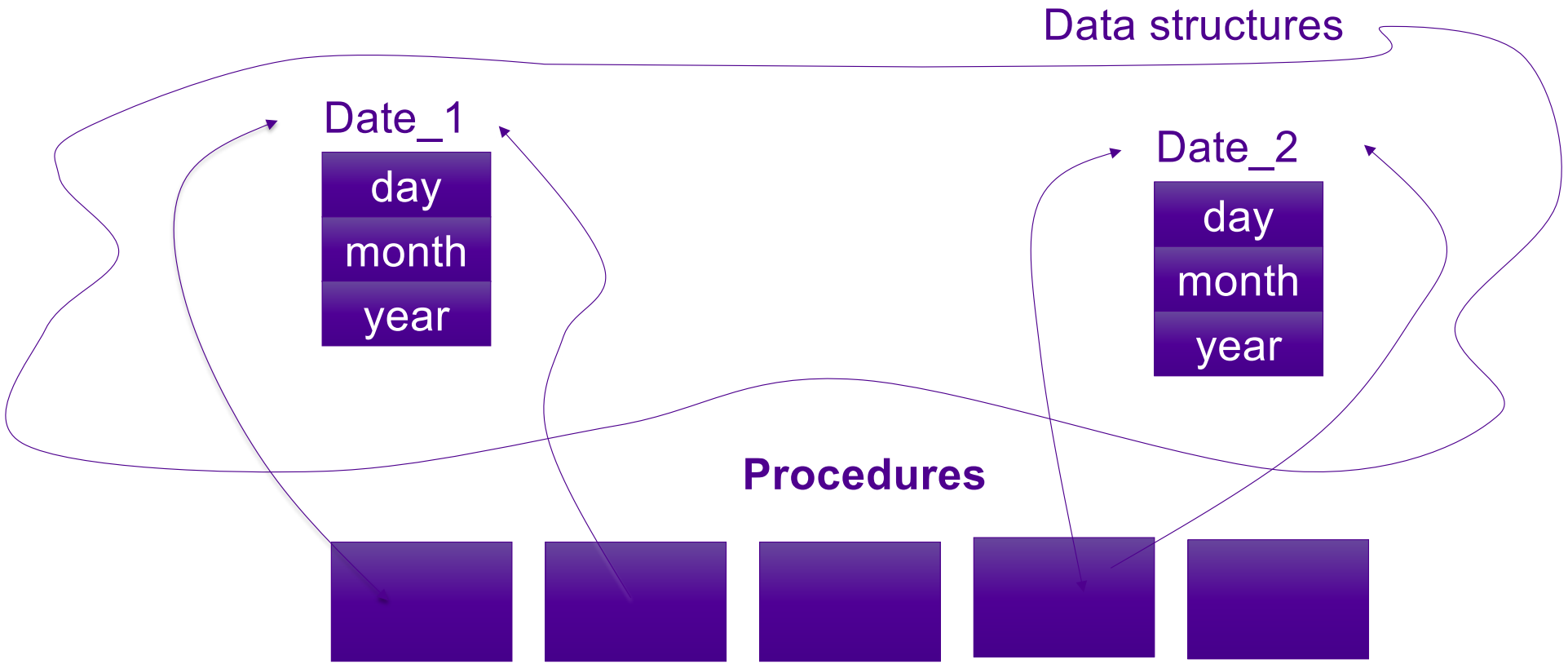
Handling the whole: dividing the problem into parts that can be managed by a single person and simplifying it by abstraction

- Data structures
- Modules
- Objects/classes
- Components

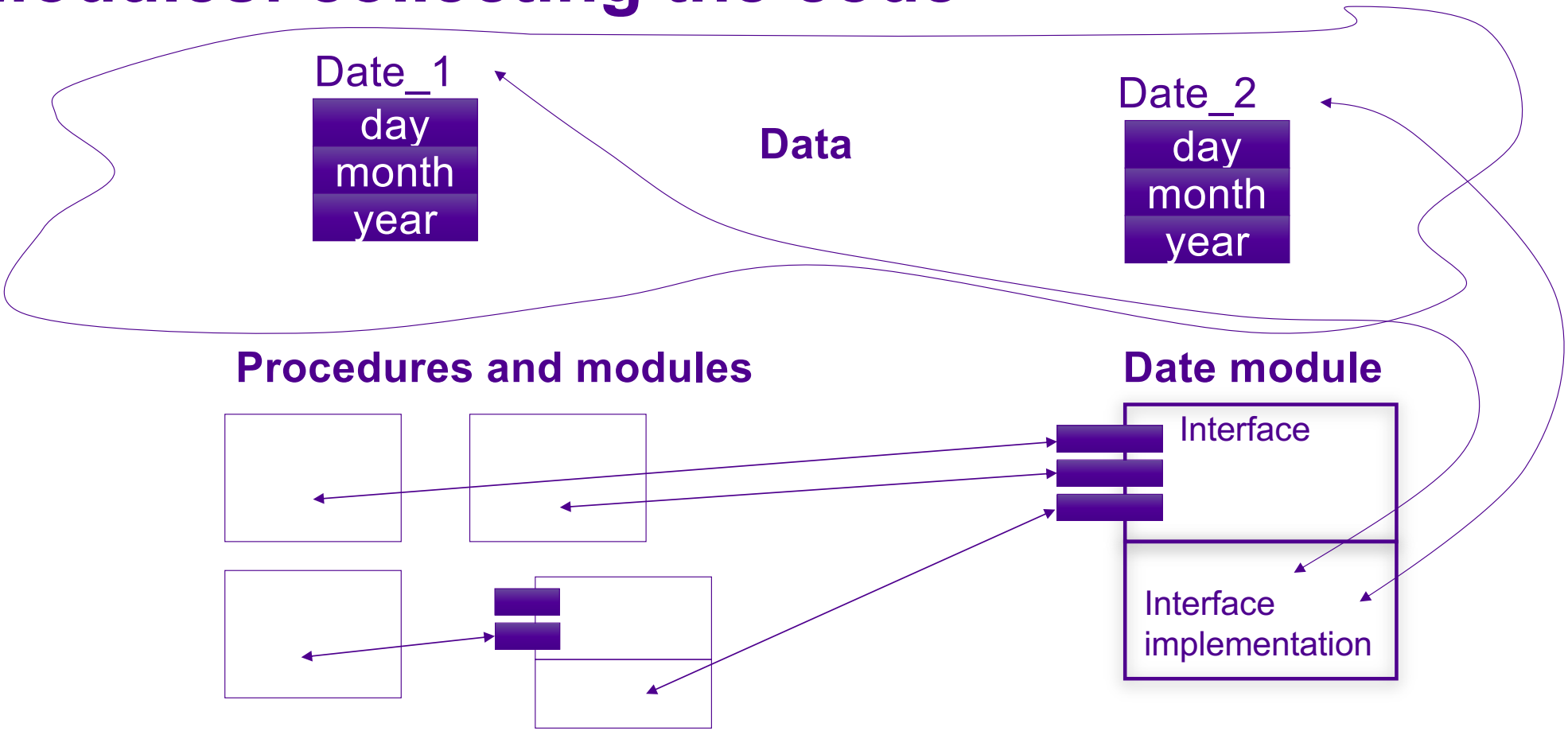
Starting point: no structure



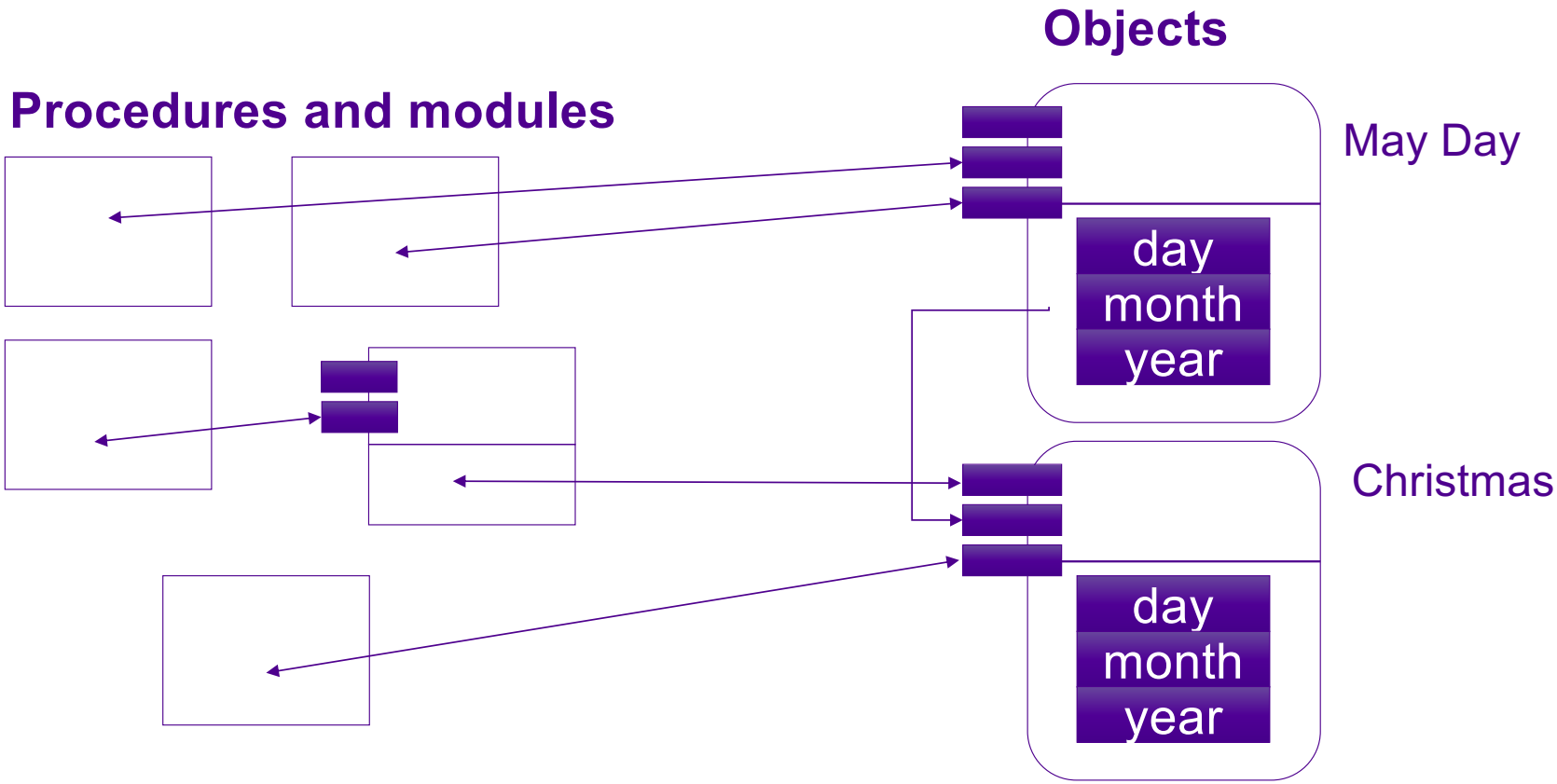
Data structures: collecting the data



Modules: collecting the code



Objects: collecting the interface



Abstraction and information hiding

Kuva: Kazimir Malevich: Black Square

Locality principle

Preserving locality: packaging strongly connected modules behind a new reduced interface

- to minimize the connections between components
- to manage complexity
- n modules → at least n-1 dependences

Tampereen yliopisto
Tampere University

Encapsulation

How many ways are there to implement a date?

Kuva: Dafne Cholet (CC BY 2.0)

Tampereen yliopisto
Tampere University

A good interface?

- **Complete**
- Beautiful
- Cute

Fig: clement127 (CC BY-NC-ND 2.0)

Tampereen yliopisto
Tampere University

Rational design process and how and why to fake it – David Parnas ja Paul Clements

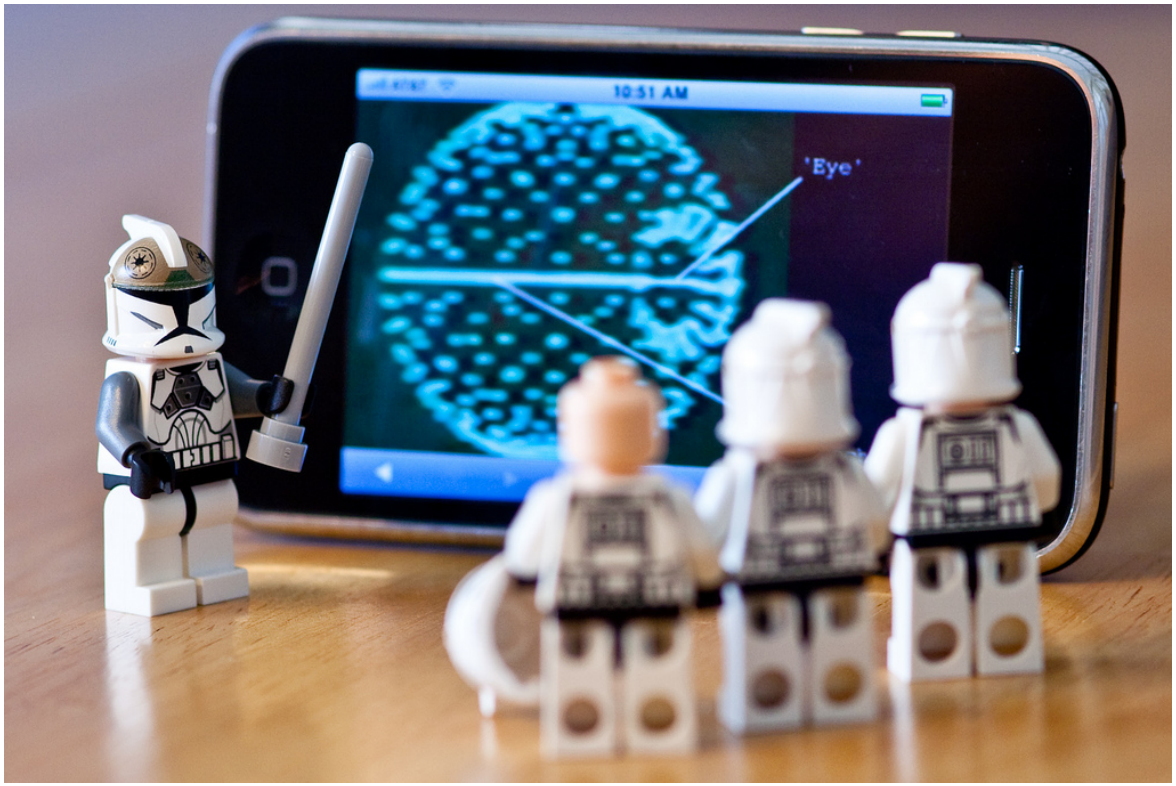
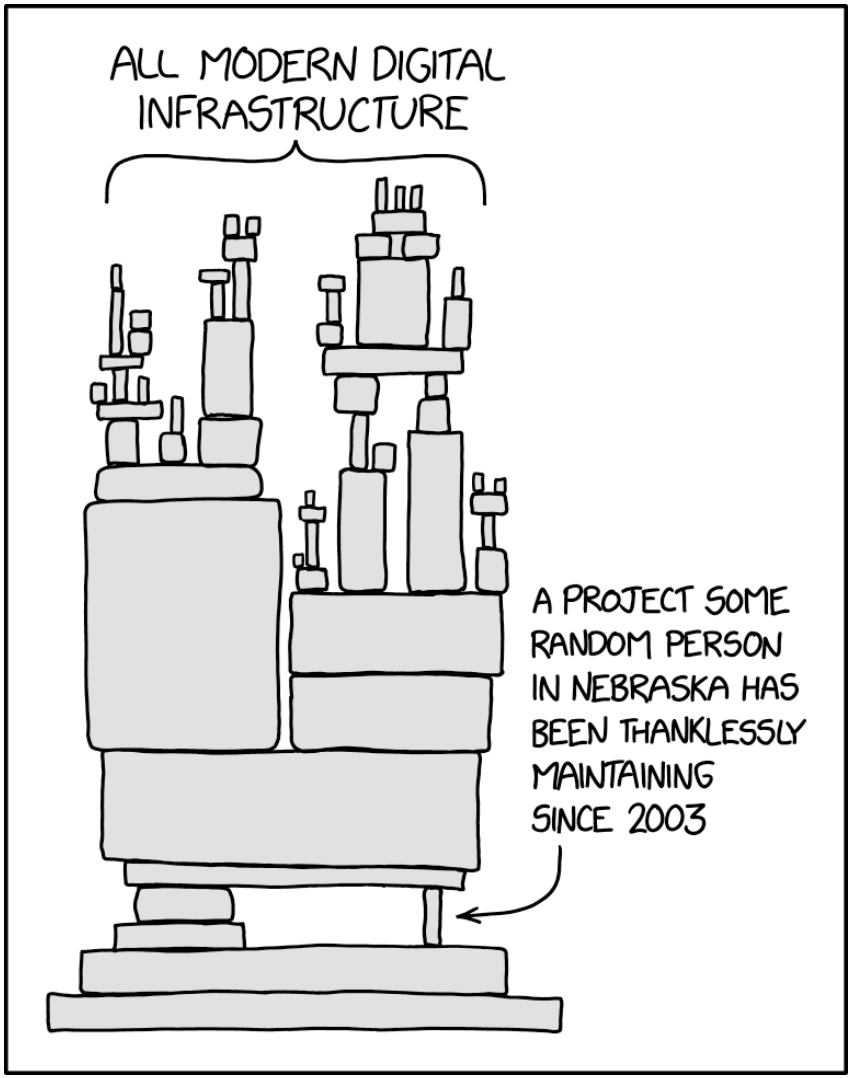


Fig: Nick Olejniczak(CC BY-NC 2.0)

Programming techniques



Kuva: XKCD (CC BY-NC 2.5)