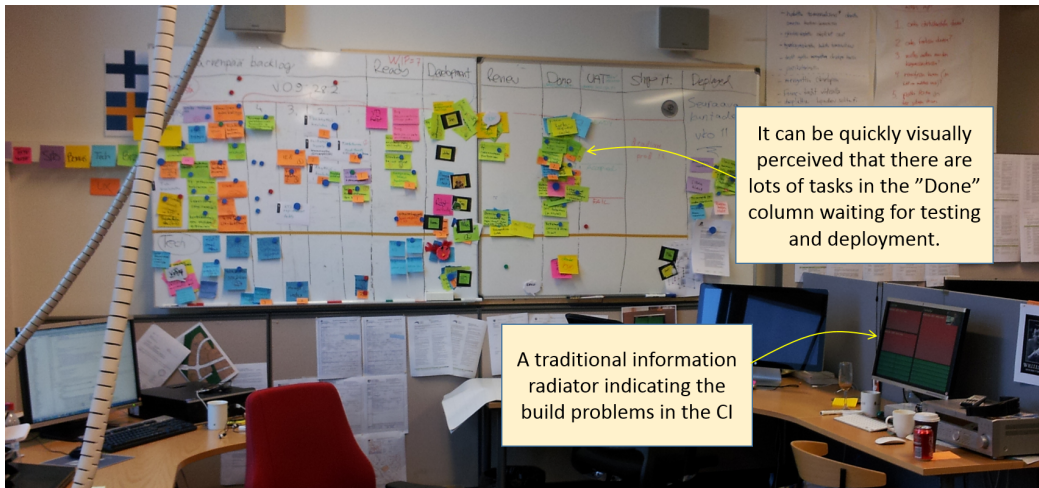


TIE-02402 Ohjelmointi 3: tekniikat

28.8.2020

Suuren ohjelman toteuttaminen



Kuva: Kanban-taulu



Kuva: Lupapiste-tiimi (Solitan Facebook-julkaisu)



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

Suuri ohjelma?



Mikä (ison) ohjelman tekemisessä on vaikeaa?



"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

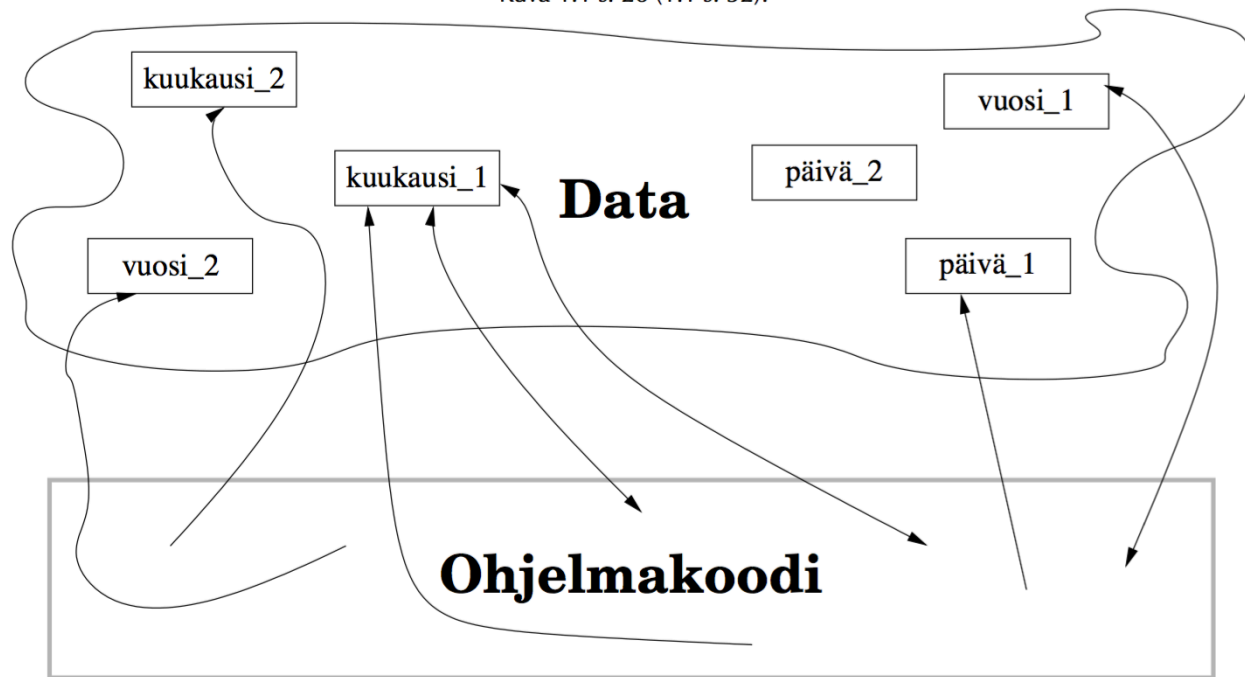
Jakaminen osiin

Kokonaisuuksien hallinta: ongelman jakaminen yhden ihmisen hallittaviin paloihin ja yksinkertaistaminen abstrahoimalla

- Tietorakenteet
- Moduulit
- Oliot
- Komponentit

Alkupiste: ei rakennetta Spagetti

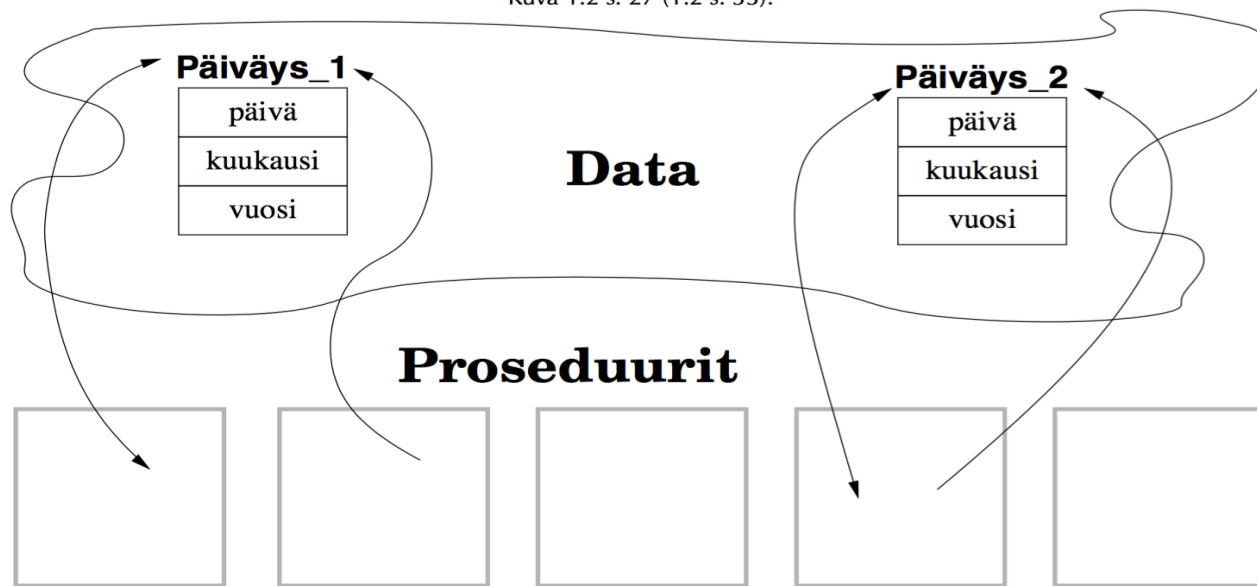
Kuva 1.1 s. 26 (1.1 s. 32):



Tietorakenteet: tiedon kerääminen yhteen

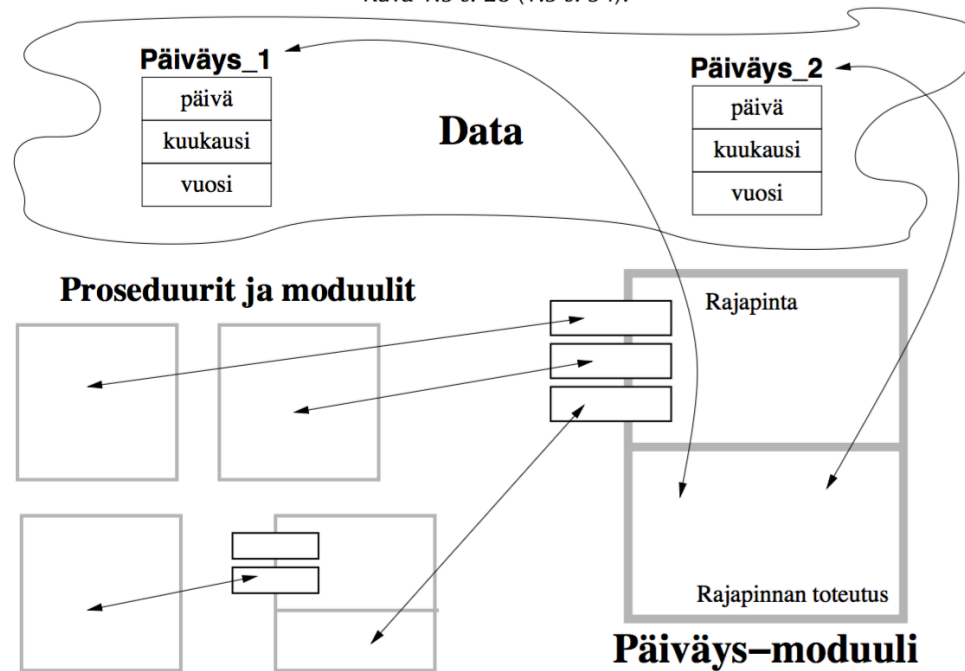
Tietorakenteet

Kuva 1.2 s. 27 (1.2 s. 33):



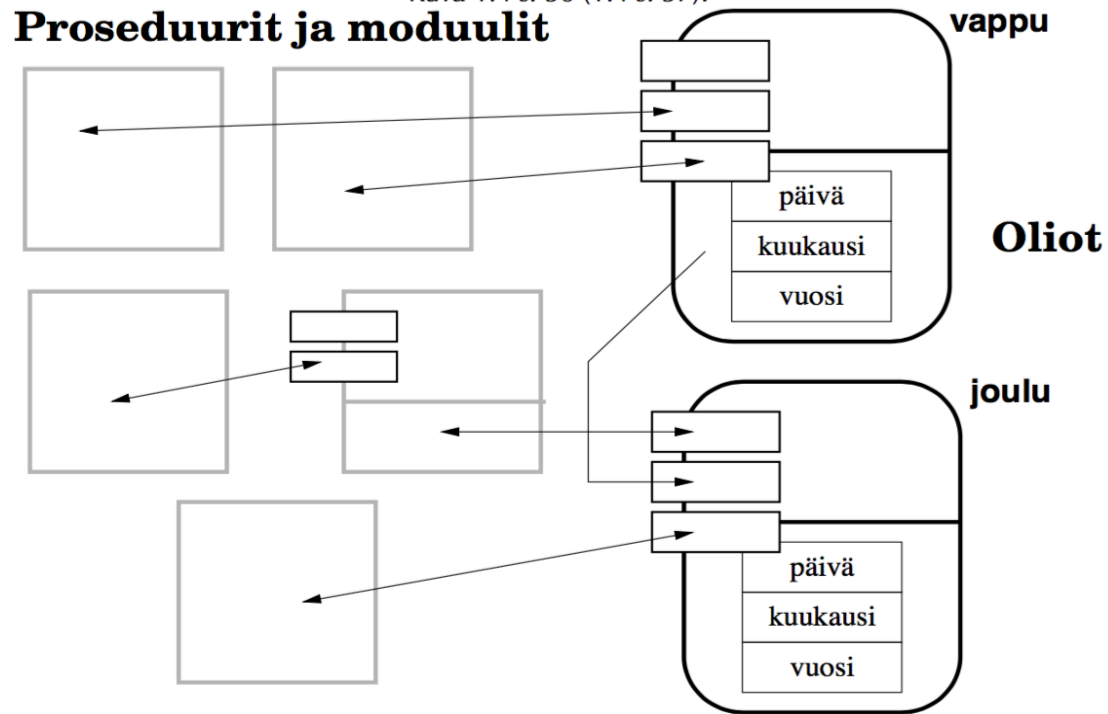
Moduulit: myös koodi yhteen

Kuva 1.3 s. 28 (1.3 s. 34):

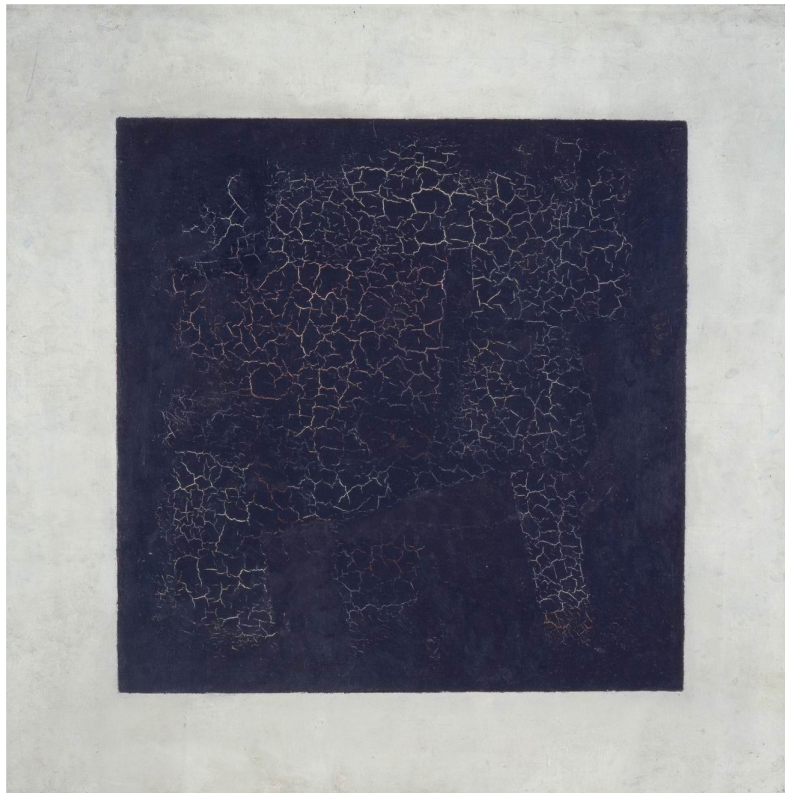


Oliot: myös rajapinnat yhteen

Kuva 1.4 s. 30 (1.4 s. 37):



Abstraktio ja tiedon kätkentä



Kuva: Kazimir Malevich: Black Square

Lokaalisuusperiaate

Lokaalisuuden säilyttäminen: vahvasti kytkeytyneiden moduulien paketoiminen uuden pelkistetymmän rajapinnan taakse

- pyrkii minimoimaan ohjelmakomponenttien välisiä yhteyksiä
- kompleksisuuden hallinta
- n moduulia \rightarrow vähintään $n-1$ riippuvuutta

Kapselointi

Kuinka monella tavalla voi päivämäärän sisäisen toteutuksen tehdä?



Kuva: Dafne Cholet (CC BY 2.0)

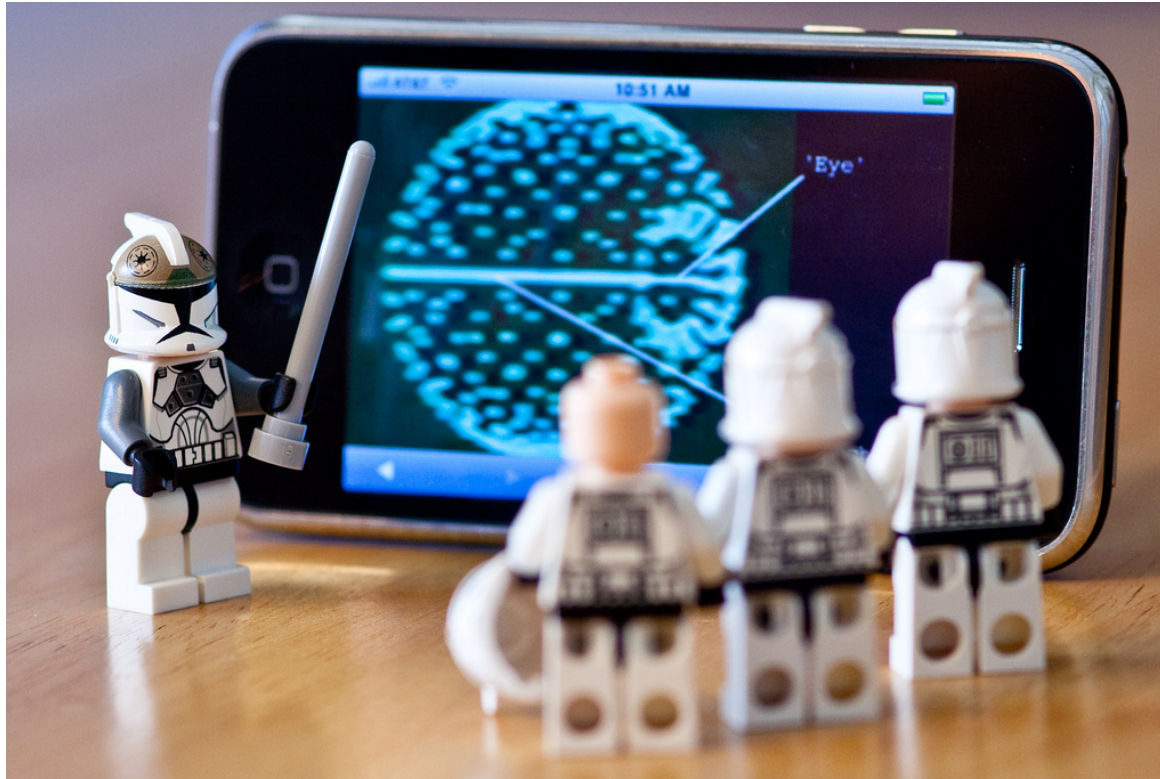
Hyvä rajapinta?

- *Täydellinen*
- Kaunis
- Söpö



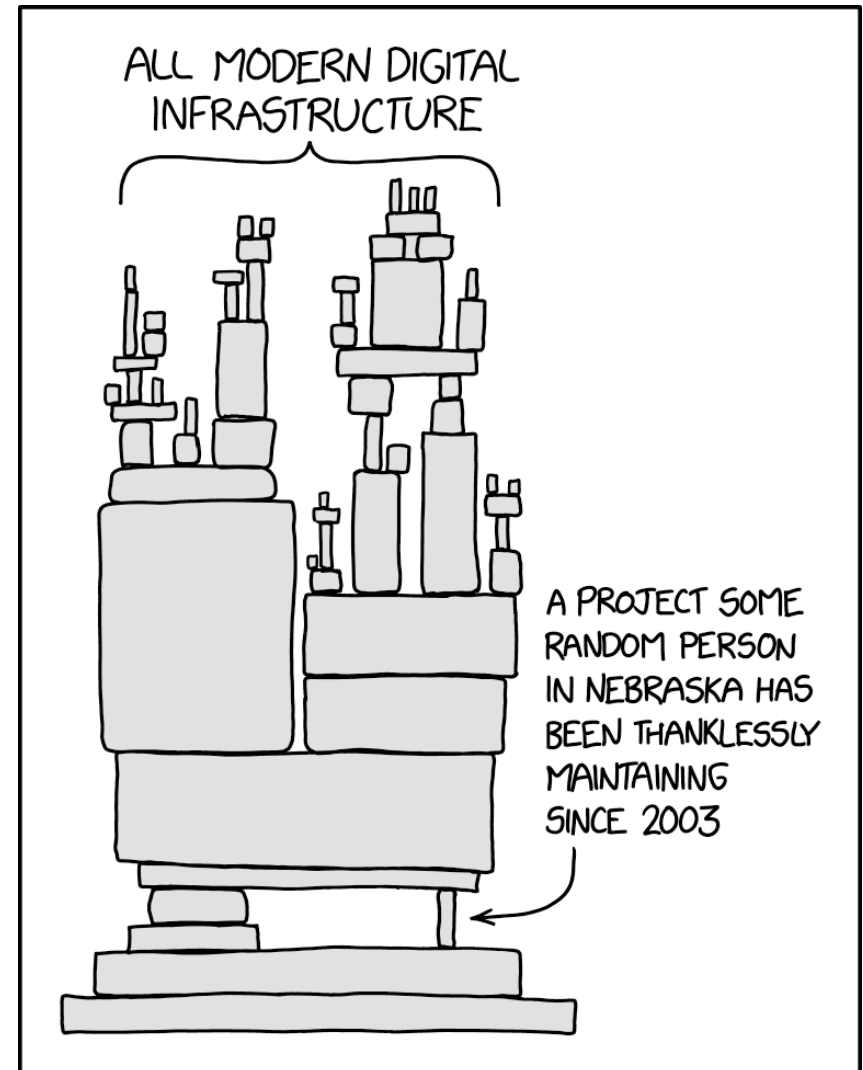
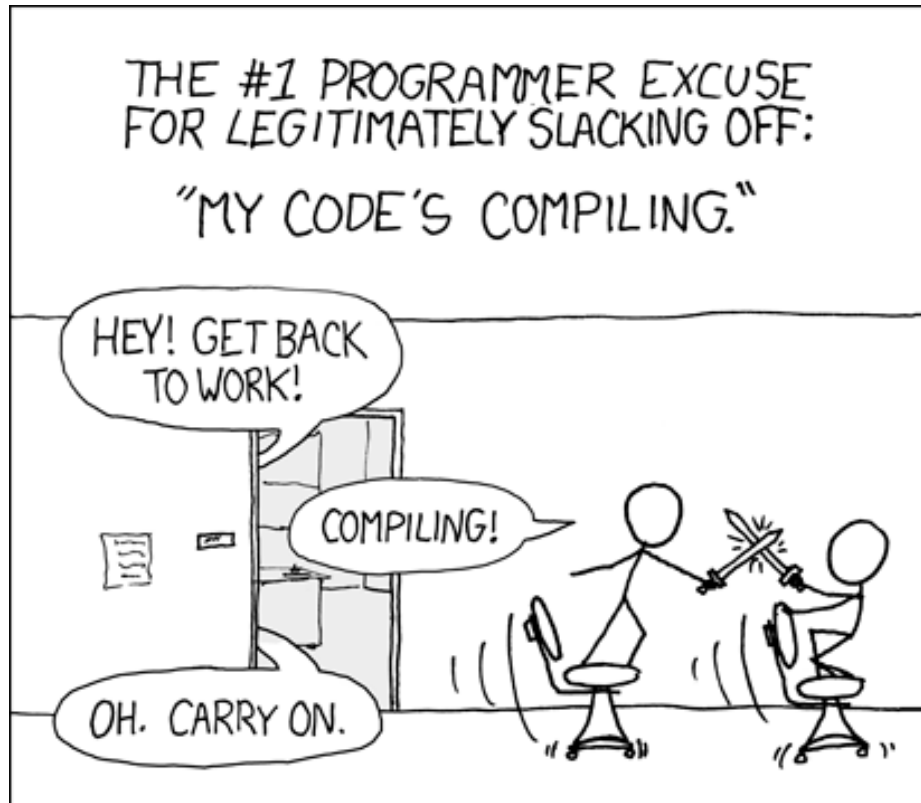
Kuva: clement127 (CC BY-NC-ND 2.0)

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



Kuva: Nick Olejniczak(CC BY-NC 2.0)

Ohjelmointi 3: tekniikat



Kuvat: XKCD (CC BY-NC 2.5)