

OSAll: Käytännön rutiinit

- Projektin hallinta
- Ohjelmiston vaatimusten määrittely
- **Ohjelmiston suunnittelu**
- Ohjelmointi
- Testaus ja laadunvarmistus
- Ylläpito

Ohjelmiston suunnittelu

- Ohjelmiston suunnittelun tavoite
- Suunnitteluperiaatteet
 - modulaarisuus, kiinteys, kytkentä, ...
 - arkkitehtuurimallit, patternit, frameworkit
- Suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
 - olioperustaiset
- Suunnitteludokumentti, katselmus
- Erityisteemat: käyttöliittymät, suorituskyky

Ohjelmistosuunnittelun tavoite

- Ensimmäinen kehitystyövaihe
- ohjelmiston sisäinen malli
 - vaatimusmäärittely antaa ulkoisen mallin
 - suhteessa ohjelmointikieleen yms.
- Konkreettiset suunnittelun osat
 - tietosuunnitelma - arkkitehtuurisuunn.
 - toimintasuunn. - liittymäsuunnitelma

Suunnittelun osa-alueet

- Arkkitehtuurisuunnittelu - architecture d.
 - korkean tason elementit & suhteet
- Toimintosuunnittelu - procedural design
 - arkkitehtuurisuunnitelman prosessit tarkentuvat komponenttien toimintokuvauksiksi
- Tietosuunnittelu - data design
 - käsitekaavioista ja tietohakemistosta tietorakenteiksi
- Liittymäsuunnittelu - interface design
 - ulkoisten liittymien toiminta, käyttöliittymä

Ohjelmistosuunnittelun tasot

- Yleissuunnittelu -product/system/preliminary d.
 - järjestelmän kokonaisrakenne
 - yhteiset tietorakenteet/varastot
 - jako alijärjestelmiin, kommunikointi
- Yksityiskohtainen suunnittelu -
detailed/program/module design
 - moduulien sisäiset rakenteet
 - algoritmit, tietorakenteet/varastot

Suunnittelun ongelmia

- Ohjelmistosuunnitelman sopivien hierarkioiden ja abstraktiotasojen löytäminen
 - tiedon ja toimintojen abstraktiot
- Moduulien löytäminen
 - suunnittelun tuloksen tarjottava yksinkertaiset liittymät moduulien välillä (ja ympäristöön), itsenäiset moduulit ilman sivuvaikutuksia

Suunnittelun nyrkkisääntöjä

- Ole laajakatseinen, mieti vaihtoehtoisia lähestymistapoja
- Uudelleenkäytä tunnettuja ratkaisuja
- käytä välitason abstraktioita:
 - ohjaa suunnitelman tarkastelija riittävän pienissä askeleista reaalimaailman vaatimuksista ohjelmiston sisältämiin algoritmeihin
 - välitasoista syntyy kontrollimoduuleita

Suunnittelun nyrkkisääntöjä ...

- Varaudu muutokseen (dokumentoidusti)
- Varaudu poikkeustilanteisiin
 - poikkeus normaalista toimintotapauksesta on normaali osa ohjelmistoa!
- Älä ohjelmoi, muista abstraktiotaso
- Huolehdi laadun suunnittelusta
- Sido suunnitelman kohdat vaatimukseen
- Iteroi!

Suunnittelun nyrkkisääntöjä ...

- Mikä olikaan suunnittelun %osuus projektin resurssikulutuksesta?
- Suunnitteluvirheen korjauskustannus?
- Tavoite: MITÄ ... ja MITEN?

Ohjelmiston suunnittelu

- Ohjelmiston suunnittelun tavoite
- Suunnitteluperiaatteet
 - modulaarisuus, kiinteys, kytkentä, ...
 - arkkitehtuurimallit, patternit, frameworkit
- Suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
 - olioperustaiset
- Suunnitteludokumentti, katselmus
- Erityisteemat

Suunnitteluperiaatteet

- Suunnitelman toivottavat ominaisuudet
 - modulaarisuus
 - kiinteys (cohesion)
 - kytkentä (coupling)
 - tiedon kätkeyminen (information hiding)
 - ylläpidettävyys (maintainability)

Modulaarisuus

- Yksityiskohtaisen suunnittelun tasolla
 - ohjelmisto on jaettu erillisiin nimettyihin (usein erikseen käännettäviin) osiin
 - tietorakenteet yleensä moduulien yksityisiä, käyttö operaatioiden kautta
 - julkiset operaatiot liittymänä moduuliin
 - vrt objektit
 - koko (30LOC?) vs liittymien lukumäärä (3?)

Modulaarisuus

- Yleissuunnittelun tasolla
 - osajärjestelmä
 - megamoduuli
 - toteuttaa yhden selkeän tehtäväkokonaisuuden vaatimusmäärittelyn kannalta katsottuna
 - rajapinnat, ohjelmamoduulit suhteessa osajärjestelmään

Moduulihierarkkiat

- Moduulihierarkkia - ohjston rakennekuvaus
 - kutsurakenne puumallina
 - syvyys: maksimipolku juuresta lehteen
 - leveys: sisarusten maksimilukumäärä
 - sisäänhaarautuminen (fan in): kutsujien lukumäärä
 - uloshaarautuminen (fan out): kutsuttavien lukumäärä

Moduulihierarkkiat

- Pyrkimyksiä
 - kehäkytkennät -> vaikeuksia
 - puurakenteen muoto elliptinen
 - tasomallin käyttö
 - osajärjestelmien erityistehtävät omina moduuleinaan
 - kaikkien osajärjestelmien yhteisesti tarvitsemat perustehtävät alempana “abstraktina koneena”, alustana (platform), infrastruktuurina

Kiinteys (cohesion)

- Moduuli on kiinteä jos kaikki sen osa palvelevat yhtä ainoaa osatehtävää
 - moduuli ei ole kiinteä jos se sisältää toisiinsa heikosti suhteessa olevia osia
- lisää moduulin ylläpidettävyyttä
- lisää mahdollisuuksia uudelleenkäyttöön

Kiinteyden asteita

- Satunnainen kokoelma
- Samantyyppisiä tehtäviä (esim tulostus)
- Samaan aikaan/etenemisjärjestyksessä tarvittavia osia
- Yhteisiä tietorakenteita käyttäviä osia (objektit)
- Moduulin osat tuottaja-kuluttaja-suhteessa
- Toimintakokonaisuuteen perustuva

Kytcentä (coupling)

- Kuvaa vuorovaikutuksia moduulien välillä
- Heikosti kytkettyjä moduuleja helpompi ylläpitää
 - muutokset eivät vaikuta muihin moduuleihin

Kytkenän asteita

- Ei vuorovaikutusta
- Käsiteltävä tieto välitetään yksittäisinä parametreina /rakenteisina parametreina
- Ohjaustiedon välittäminen parametreina
- Yhteiskäyttöiset laitteet
- Yhteiskäyttöinen tietorakenne
- Sisäisten rakenteiden hyväksikäyttö

Tiedon kätkeminen

- Vain liittymät paljastavat tietoa
 - syötteet, tulosteet, julkiset operaatiot
 - data vs informaatio
- toteutus, testaus ja ylläpito paikallista
- suojaus tehostuu

Moduulien riippumattomuus

- Mittarina
 - moduulin sisäinen yhteenkuuluvuus = kiinteys (cohesion)
 - moduulien välisen kytkentäasteen (coupling) pienuus

Ylläpidettävyys

- Modulaarisuus, kiinteys, heikko kytkentä
- Tiedon piilottaminen
- Suunnitteludokumentin laatu
 - yleis- ja yksityiskohtaisen suunnittelun välisen suhteen selkeys
 - selkeä suhde vaatimusmäärittelyihin
- Muutoksiin varautumisen aste
- Suunnitelman laadukkuus

Ohjelmiston suunnittelu

- Ohjelmiston suunnittelun tavoite
- Suunnitteluperiaatteet
 - modulaarisuus, kiinteys, kytkentä, ...
 - arkkitehtuurimallit, patternit, frameworkit
- Suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
 - olioperustaiset
- Suunnitteludokumentti, katselmus
- Erityisteemat

Suunnitteluperiaatteet

- Edelliset lähinnä toiminto/tietosuunn.
- Millä keinoin suunnitelman toivottuihin ominaisuuksiin (modularisuus, eheys,...) voidaan päästä?
- Suhde arkkitehtuurisuunnitelmaan?

Suunnitteluperiaatteet

- Ohjelmistoarkkitehtuurit

- M Shaw, D Garlan: Software Architecture - Perspectives on emerging discipline, Prentice Hall, 1996.

- Arkkitehtuuriratkaisujen uudelleenkäyttö, patternien käyttö, frameworkien käyttö

- ero abstraktiotasossa jolla käytetään

- Gamma, Helm, Johnson, Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software.

Ohjelmistoarkkitehtuurimalleja

- Rakennemalli, työnjako ohjelmiston sisällä

- vrt yleissuunnitelma, yhteyskaavio, systeemisuunnitelma jossa näkyy jako eri toimijatyypeille

- Komponenttien välinen vuorovaikutus muodostaa arkkitehtuurin perustan

- Jako muodostaa näkökulman, jonka mukaan tietyt palvelut ja tiedot kuuluvat samaan kokonaisuuteen

Ohjelmistoarkkitehtuurimalleja

- putket ja suodattimet
- oliot ja palvelut
- tapahtumapohjainen / reaktiivinen
- kerrosarkkitehtuuri
- jaettu tietovarasto
- tulkki
- rinnakkaiset prosessit
- pää-/aliohjelma
- tilakone
- asiakas-palvelija

putket ja suodattimet -malli

- tietovuolähestymistapa
- suodattimet toisistaan riippumattomia tiedonmuokkaajia
- suodattimen osattava tulkita saapuvaa tietovirtaa (jaettu käsitys rakenteesta)
- suodattimen ei tarvitse tietää keneltä tieto tulee tai minne se menee
- rinnakkaisuutta tai peräkkäisyyttä
- esim. unix-putket

Arkkitehtuurimalleja

● oliot ja palvelut malli

- olio tarjoaa palveluja
- palvelun pyytäjän tunnettava tarjoaja ja palvelurajapinta (kutsu- tai viestirakenne)
- tarjoajien ei tarvitse tuntea pyytäjiä (asiakkaita)
- tieto kätkeyty olion sisään

tapahtumapohjainen, epäsuoran käsittelyn malli

- palveluja ei kutsuta suoraan
- komponentti voi tuottaa tapahtumia (event), joihin toiset komponentit reagoivat
- tapahtumaan voi reagoida yksi tai useampi komponentti
- reagoija voi rekisteröityä vastaanottamaan tapahtumia
- tai tapahtumat hoidetaan yleislähetystenä ja sille herkistynyt reagoi
- tapahtuman aiheuttajan ei tarvitse tietää reagoijia, ei välttämättä edes reagoiko joku
- esim. windows -tapahtumat, tietokantatriggerit

Kerrosarkkitehtuuri

- kukin kerros tarjoaa palveluja seuraavaksi ylemmän tason kerrokselle
 - palvelujen abstraktiotaso kasvaa kerros kerrokselta
 - esim. tietoliikenneprotokollat



Kerrosarkkitehtuuri ...

- Ylemmän kerroksen palvelut tuotetaan alemman kerroksen palveluiden avulla
 - eri käsitteistö, ylempänä voimakkaammat välineet
- Suljetussa kerrosarkkitehtuurissa vain edellisen tason palvelut käytettävissä
 - ylläpidettävyys
- avoimessa kaikkien alempien tasojen palvelut käytettävissä
 - joustavuus
- esim käyttöliittymäkerros, OSI-malli

Arkkitehtuurimalleja

- **tulkkiarkkitehtuuri**
 - virtuaalikone
 - ohjauskieli, jota tulkitaan
 - tila ja tilan esitys
- **rinnakkaiset prosessit**
 - prosessien välinen kommunikointi
- **pää-/aliohjelma**
 - pääohjelmassa kontrollisilmukka
 - aliohjelmissa erikoispalvelut

Arkkitehtuurimalleja

- **Kommunikoinnin toteutustapa on arkkitehtuuritason ratkaisu:**
 - aliohjelmakutsu
 - viestinvälitys
 - yhteiset tiedostot tai muistialueet

“Patterns”

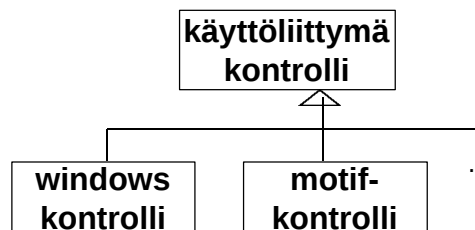
- Suunnittelumalli on systemaattinen tapa ratkaista tietty usein esiintyvä ongelma toimivaksi havaitulla perinteisellä tekniikalla
 - yksityiskohtaisen suunnittelutason malli
 - voi toteuttaa erilaisilla ohjelmontkikielillä

Suunnittelumallit - rakenne (Design Patterns)

- tarkoitus eli ratkaistava ongelma
- sovelluskohteet
- rakenne, mistä komponenteista malli muodostuu ja miten komponentit liittyvät toisiinsa
- mallin edut ja haitat
- sovellusesimerkkejä
- toteutusvaihtoehtoja (koodirunkoja)

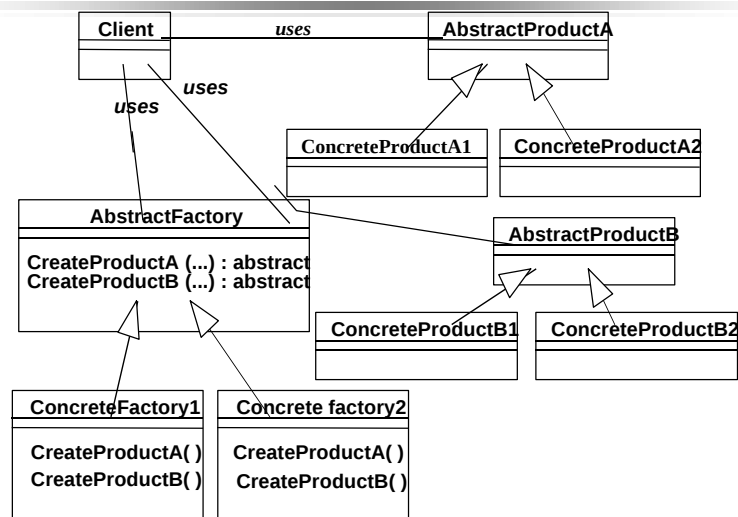
Esimerkki: abstract factory

- Tarkoitus: malli esittää tavan luoda tietyn luokan jälkeläisluokkien ilmentymiä tuntematta näitä luokkia
- Esimerkki: yleiskäyttöinen käyttöliittymäkirjasto



kontrollit erilaisia eri ympäristöissä kirjaston pitäisi olla yleinen, miten voidaan yleisellä tasolla luoda ilmentymiä ympäristökohtaisille luokille

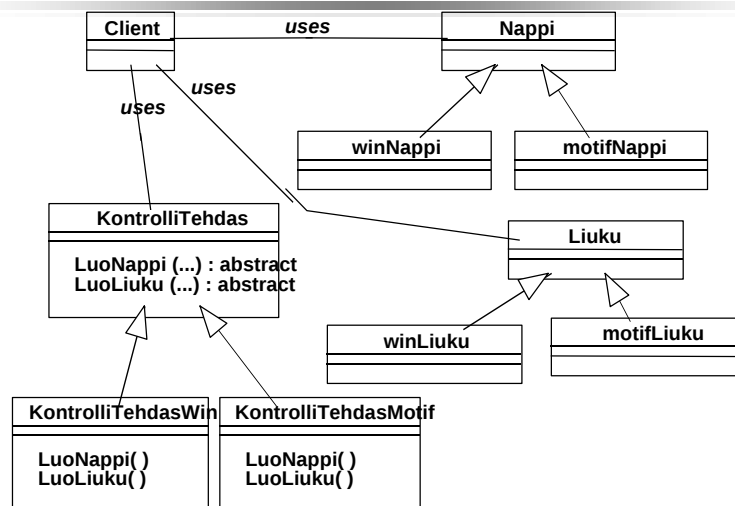
Abstract Factory



Abstract Factory

- Client (siis hyödyntävä ohjelma) tuntee abstraktin tehtaan ja abstraktit tuotteet ja kutsuu näiden tarjoamia palveluja. Abstraktin tehtaan palvelukutsu ohjautuu konkreettisen tehtaan vastaavaan luontipalveluun, joka siis luo ilmentymän kyseisen tyyppiselle tuotteelle
- esim windows-kontrollitehtaan luoNappi-palvelu luo ilmentymän winNappi-luokkaan.

Abstract Factory sovelle



Abstract Factory

- Mallilla eristetään konkreettiset tehtaat asiakkaasta
- Tuoteperhe voidaan helposti vaihtaa toiseksi
- Asiakkaan tunnettava tuotetyypit, joten uusia tuotetyyppejä ei voida lisätä pelkästään aliluokkia lisäämällä -> muutettava asiakkaan ja abstraktin tehtaan koodia
 - esim. kilistimen lisäys käyttöliittymäkontrolleihin

Sovelluskehykset (application frameworks)

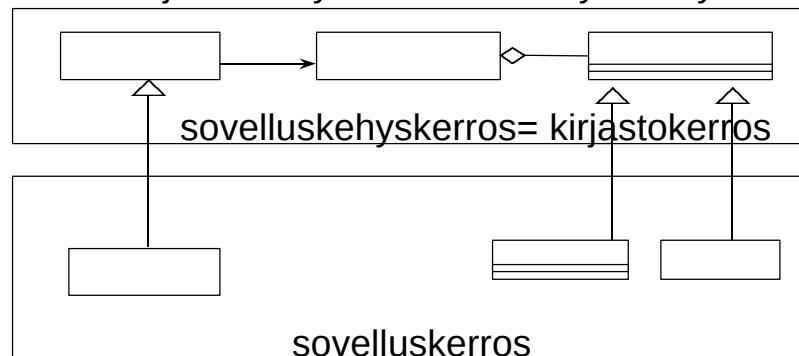
- uudelleenkäytettäväksi tarkoitettua kokoelmaa yhdessä toimivia yleensä abstrakteja luokkia, jotka tarjoavat perusrakenteen ja -toiminnallisuuden tietyn tyyppiselle sovellukselle
 - käyttöliittymäkehykset
 - yleiset sovelluskehykset (MacApp, Borland OWL,...)
 - sovellusaluekohtaiset kehykset,

Sovelluskehyykset

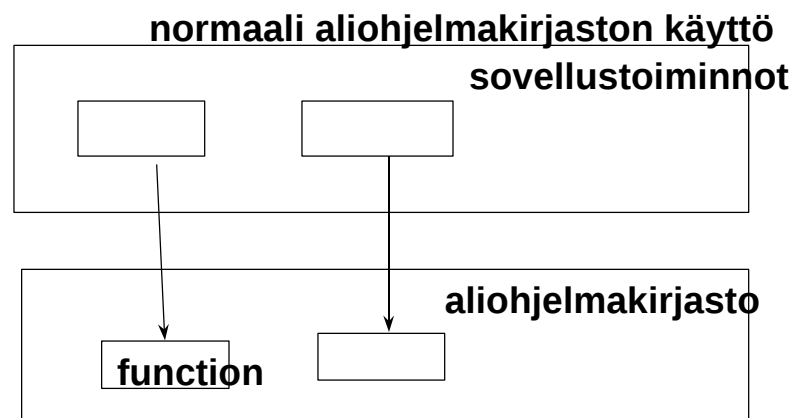
- runko, johon esim. perintää ja syrjäyttämistä hyväksikäyttäen voidaan liittää sovelluskohtaista toiminnallisuutta.
- Runkoluokat kutsuvat abstraktien luokkien operaatioita, jotka sitten konkreettisten jälkeläisluokkien oliot toteuttavat
- hollywood-periaate: don't call us we call you,
 - kirjastosta kutsutaan sovelluskohtaisia rutiineja

Sovelluskehyykset

kirjaston käyttö sovelluskehystä käytettäessä



Sovelluskehyykset



- Sovelluskehys toteuttaa joitakin (useita) suunnittelumalleja ja tarjoaa niihin perustuvan yleisrakenteen
- sovellusalueelle (application domain) tyypillisiä erityispalveluja
- esim CASE-kehys

CASE-kehys

- Kehyskohtaisesti
 - Kaavioikkunoidene hallinta
 - kaavioelementtien luonti, poisto, siirtely
 - kaavioelementtien yhteyksien käsittely
 - kaavion paperitulostus
- sovelluskohtaisen aliluokan tehtäviä
 - kaavioelementin piirtäminen
 - uusien kaavioelementtien määrittely

Patterns vs Frameworks

- Design patterns more abstract than frameworks
- desing patterns smaller architectural elements than frameworks
- design patterns less specialized than frameworks

Ohjelmiston suunnittelu

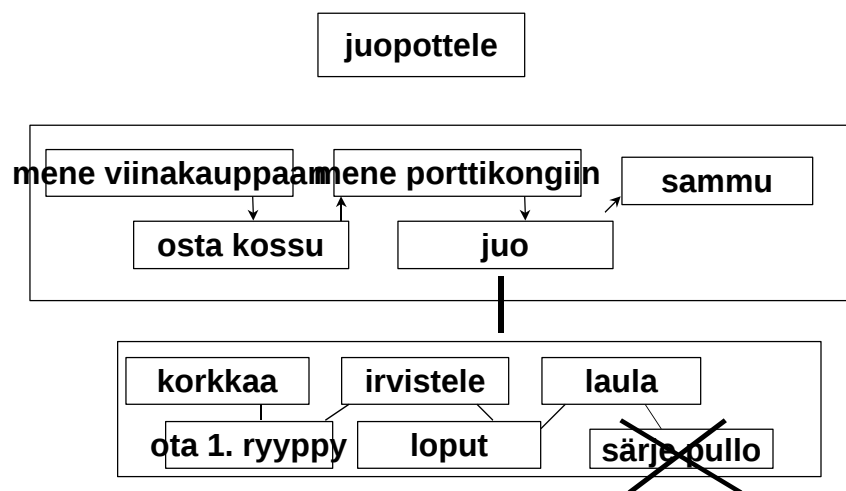
- Ohjelmiston suunnittelun tavoite
- Suunnitteluperiaatteet
 - modulaarisuus, kiinteys, kytkentä, ...
 - arkkitehtuurimallit, patternit, frameworkit
- Suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
 - olioperustaiset
- Suunnitteludokumentti, katselmus
- Erityisteemat

Suunnitteluprosessi

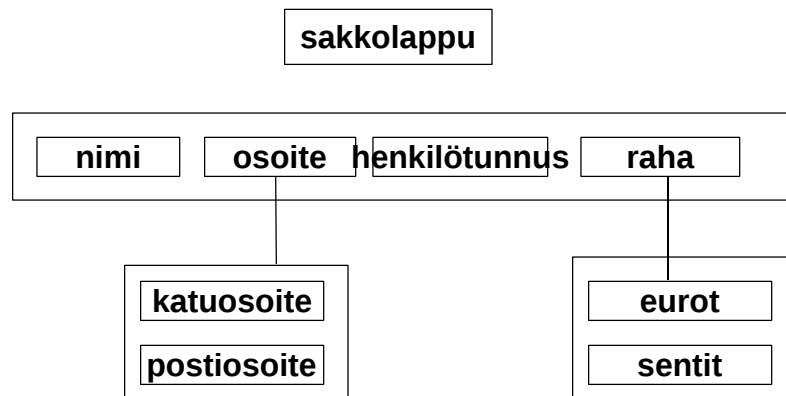
- Asteittain tarkentava suunnittelu
 - selvitä ongelman olennaiset osat
 - vaatimusanalyysi, älä lykkää selvitystehtäviä
 - etsi ratkaisumahdollisuus(k)s(ia)
 - simuloi keskeiset tilanteet, myös poikkeukset
 - valitse yksinkertaisin riittävä/kattava
 - kuvaa hierarkisesti tarkentaen
 - muodosta vastaava rakennekaavio

- Kuvaa kunkin tarkennustason osaset
 - toiminnot
 - tieto
- tiukka top-down -suunnittelu ei yksin riitä
- iteraatio!

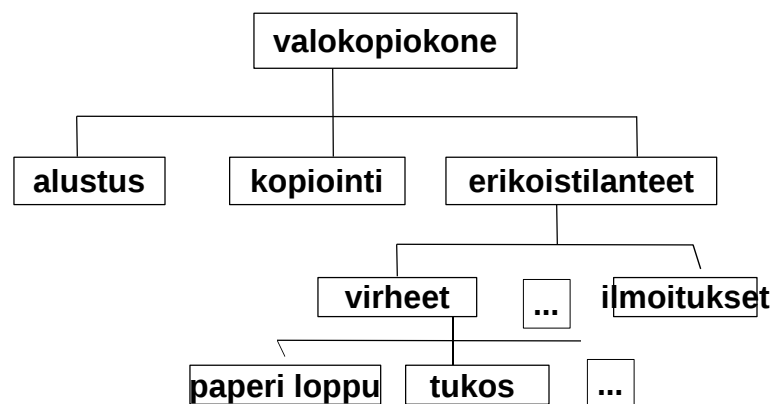
Esimerkki / toiminta



Esimerkki / tieto

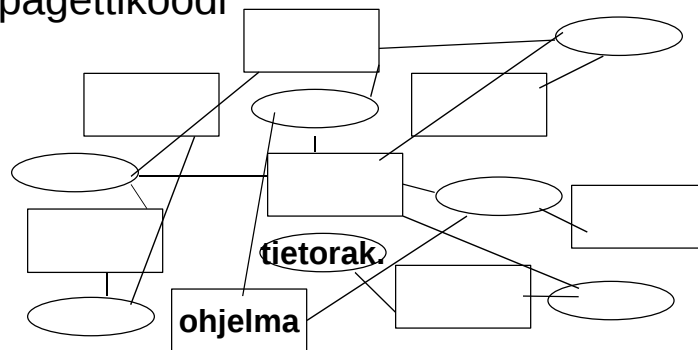


jako/tarkennushierarkia



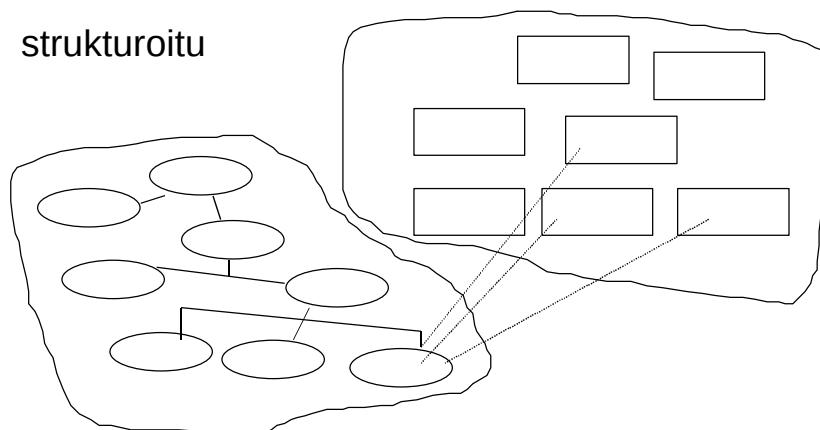
ohjelmarakenteita

- spagettikoodi



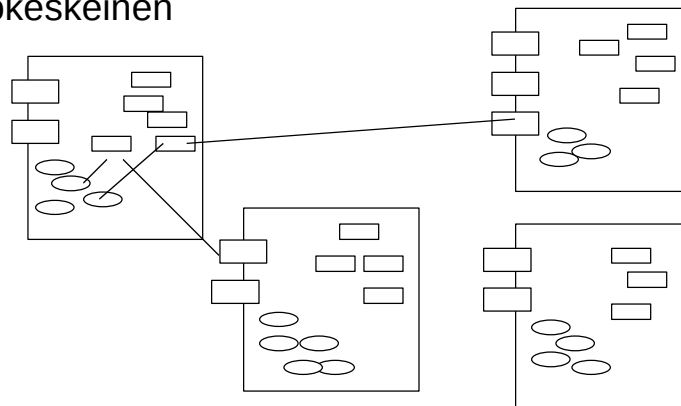
ohjelmarakenteita

strukturoitu



ohjelmarakenteita

oliokeskeinen



Suunnitteluprosessi

- Erityisteemoja
 - suunnitelman laadun arviointi
 - ohjelmiston laadun suunnittelu
 - käyttöliittymät
 - suorituskyky

Suunnittelumenetelmät

- Toimintopohjainen
- Tietopohjainen
- Oliopohjainen

Toimintopohjaiset menetelmät

- Lähtökohtana toimintopohjainen vaatimusanalyysi:
 - tietovuokaaviot
 - toimintojen hierarkkinen jäsentäminen
- Suunnitteluvaiheessa
 - jäsennetään ratkaisu samalta pohjalta
 - strukturoitu suunnittelu ylhäältä alaspäin (top-down) (Constantine, Yourdon, Myers)

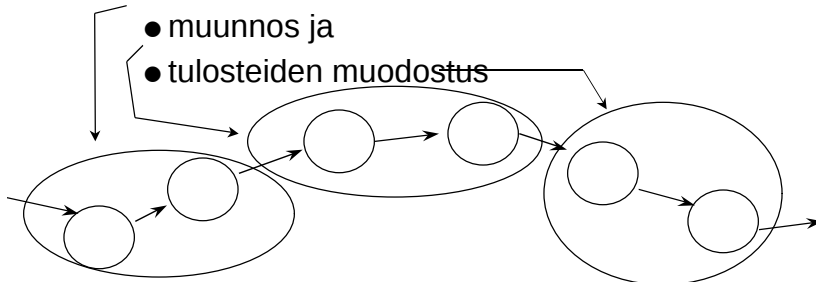
Toimintopohjaiset menetelmät

- Toiminnan luonne - kaksi päätyyppiä:
 - muunnosjärjestelmä (*transform flow*):

■ tieto muunnetaan syöttötiedosta tulostiedoksi

■ osat :

- syöttötietojen käsittely,
- muunnos ja
- tulosteiden muodostus



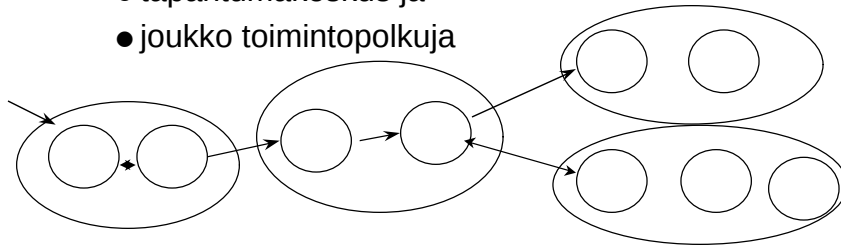
Toimintopohjaiset menetelmät

– tapahtumankäsittelyjärjestelmä (*transaction flow*):

■ tiedon käsittelyn ydin on tapahtumakeskus, josta käsittely jakautuu moneksi erilliseksi toimintopoluksi

■ osat:

- syöttötietojen vastaanotto,
- tapahtumakeskus ja
- joukko toimintopolkuja



Muunnos tietovuokaaviosta ohjelmarakenteeksi

- Erotetaan vuokaavion pääosat:
 - syöttö - muunnoskeskus - toiminta/tulos
- Muodostetaan näistä pääosista ylimmän tason rakennekaavio
- Tarkennetaan kutakin alipuuta, kunnes kaikki vuokaavion solmut on sijoitettu

- Rakennekaavio:

- muunnosjärjestelmä:

- rakennekaavion juureksi valvontasolmu, sen kolmeksi alipuuksi:

- syöttö, muunnos ja tulostus

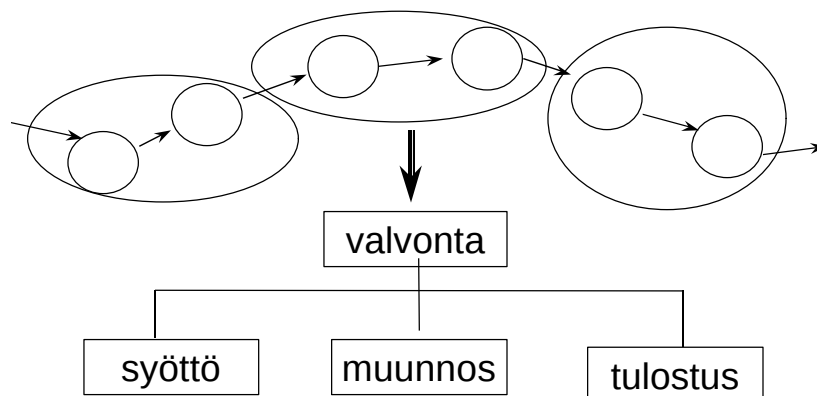
- tapahtumankäsittelyjärjestelmä:

- rakennekaavion juureksi tapahtumakeskus,

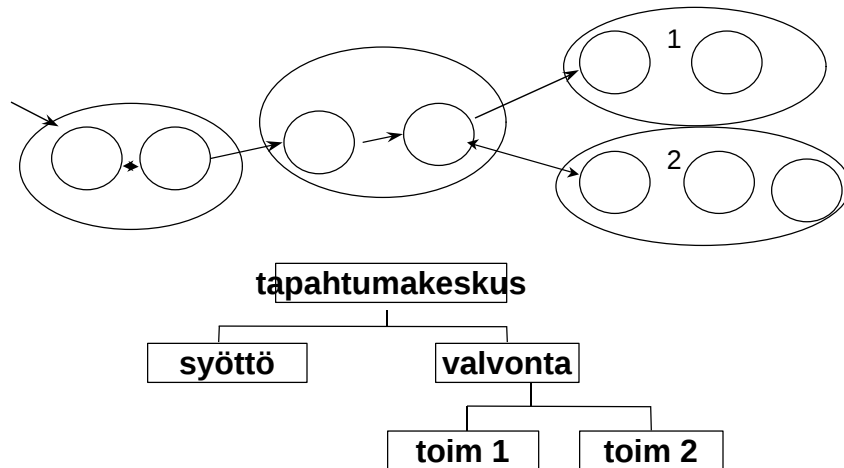
- sen toiseksi alipuuksi syötteiden vastaanotto

- ja toiseksi alipuuksi valvontasolmu, jonka alipuina ovat toimenpidepolut

Muunnos tietovuokaaviosta ohjelmarakenteeksi



Muunnos tietovuokaaviosta ohjelmarakenteeksi



- Horizontal partitioning
- vertical partitioning
- suhde työnjakoon projektissa
- suhde ylläpidettävyyteen

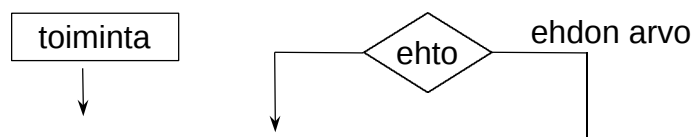
Tiedot ja toimintopohjainen suunnittelu

- Tietovuokuvauksista johdetaan myös kunkin moduulin tietosisällöt, jotka tarkennetaan hierarkkisesti edeten

Yksityiskohtainen suunnittelu - algoritmit

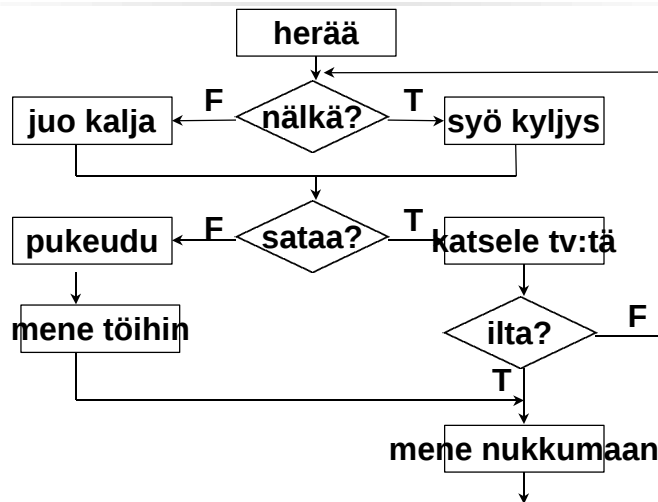
- Algoritmeille useita kuvaustekniikkoja, esim:
 - perinteinen kulkukaavio (flow chart)
 - pseudokoodi
 - rakenteinen kulkukaavio (Nassi-Sneidermann)
 - päätöstaulu

Perinteinen kulkukaavio



nuoli kuvaa kontrollin etenemistä

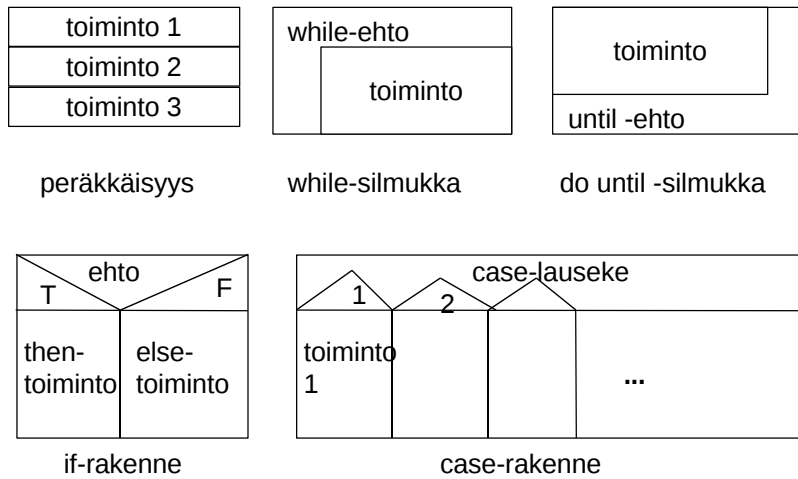
Kulkukaavio: johtaja Turhapuron päivä



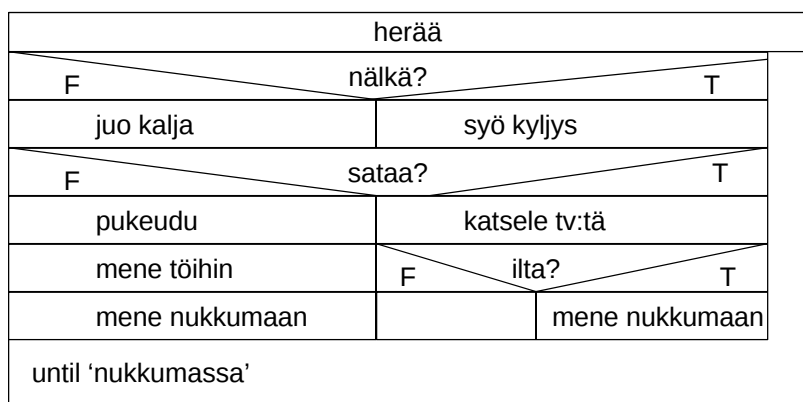
pseudokoodi: johtaja Turhapuron päivä

```
begin
  herää;
  repeat until 'mene nukkumaan' happens:
    if nälkä then
      syö kyljys
    else
      juo kalja;
    if sataa then begin
      katsele tv:tä;
      if ilta then mene nukkumaan;
    end
  else begin
    pukeudu;
    mene töihin;
    mene nukkumaan;
  end;
end;
```

rakenteiset kulkukaaviot



rakenteinen kulkukaavio: johtaja Turhapuron päivä



Päätöstaulut

Sääntö		n	n+1	
Ehto i		T	F	
Toiminta I Toiminta I+1		x	x	

ehdon arvo säännössä

säännön tunnus

merkintä siitä, että toiminta suoritetaan kun sääntö voimassa

Päätöstauluesimerkki

Sääntö	1	2	3	4	5	
Sairaana	T	F	F	F	F	
Sataa	-	T	T	F	F	
Viikonloppu	-	T	F	T	F	
Töihin					x	
Sänkyyn	x					
Kylään				x		
TV auki	x	x	x			

ei merkitystä

kaikki arvoyhdistelmät otettava mukaan (muutoin-vaihtoehtoa voi käyttää)

Algoritmin valinta

- Algoritmin soveltuvuus:
 - virheettömyys
 - voimassaoloalue
 - tehokkuus
 - totetutusympäristöstä riippuvat rajoitteet

Suunnittelumenetelmät

- Toimintopohjainen
- Tietopohjainen
- Oliopohjainen

Tietorakennepohjaiset menetelmät

- Käsiteltävän tiedon rakenne määrää ohjelmiston rakenteen:
 - peräkkäisyys, ehdollisuus, toisto
 - hierarkkisuus
- Lähtökohtana tulosteiden, syötteiden, tietokannan, kyselyn, tapahtuman rakenne (esim. Jacksonin JSP, Warnier & Orr)

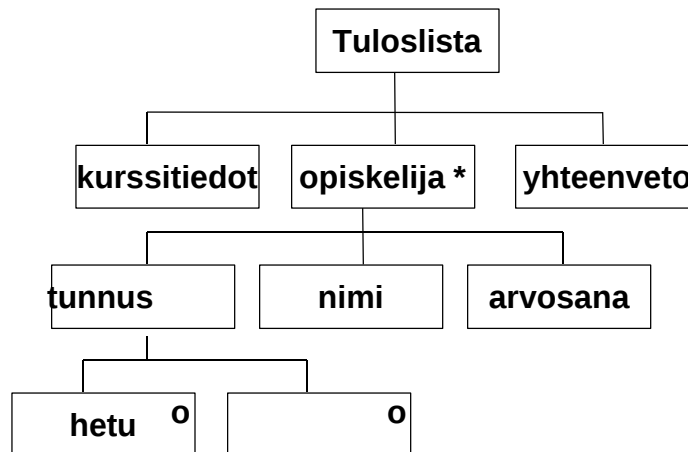
- Osatehtävät:
 - tietorakenteiden kuvaaminen perusprimitiivien avulla:
 - peräkkäisyys, vaihtoehtoisuus, toisto
 - tietorakenteen muuntaminen kontrollihierarkiaksi
 - ohjelmistohierarkian tarkentaminen
 - ohjelmiston toiminnallinen kuvaus

Tietorakennepohjaiset menetelmät

- Tyypillisiä sovellusalueita:
 - kaupallis-hallinnolliset sovellukset
 - tietokoneavusteinen suunnittelu ja valmistus
 - (CAD, CAM, CAE)
 - sovellukset, joiden käsittelemällä tiedolla on
 - hyvin määritelty ja hierarkkinen rakenne

- Esim.: JSP - Jackson Structured Programming (-75)
 - Tietorakenteesta kuvattavissa
 - peräkkäisyys,
 - vaihtoehtoisuus ja
 - toisto
 - Tietorakenteet kuvataan hierakisena kaaviona, joka voidaan esittää myös tekstimuodossa.

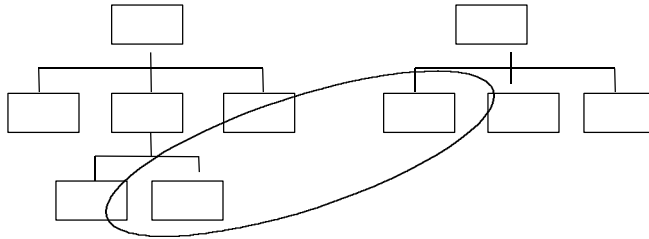
JSP-rakennekaavio



JSP-periaate

- Muodostetaan palveluun liittyviä peräkkäisrakenteita vastaavat tietorakenteet (1 tai useampia) esim. raportin rakennekuvaus, syöttötiedostojen rakennekuvaukset.
- Jos rakenteita on useampia määrätään eri rakennekuvausten rakenneosien vastaavuus kytkemällä vastinosat yhteen.
 - Kutsutaan vastionosien yhteenliittymää osapariksi (yhteenliittymässä voi olla yli 2 osaa)
 - raportin rakenneosa & tietokannan rakenneosa

JSP-periaate



JSP-periaate

- Aikaansaatu kytketty tietorakennekaavio muunnetaan ohjelmarakennekaavioksi:
 - moduuli jokaista osaparia kohden
 - moduuli jokaista paritonta kohden
- Rakenteiden yhteensovittamisessa voi syntyä rakenneristiriitoja
 - (esim raportti eri järjestyksessä kuin syöttöaineisto)
 - Menetelmä tarjoaa standardiratkaisuja näiden hoitamiseen.

Ohjelmiston suunnittelu

- Ohjelmiston suunnittelun tavoite
- Suunnitteluperiaatteet
 - modulaarisuus, kiinteys, kytkentä, ...
 - arkkitehtuurimallit, patternit, frameworkit
- Suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
 - olioperustaiset
- Suunnitteludokumentti, katselmus
- Erytisteemat: käyttöliittymät, suorituskyky

Suunnitteludokumentti

- Arkkitehtuurisuunnitelma
- Toimintasuunnitelma
- Tietosuunnitelma
- Liittymäsuunnitelma
- Yhteys vaatimukseen (ulkoinen käytös)
- Testaussuunnitelma
 - myöhemmin esillä

Suunnitteludokumentin rakenne

- 1 Johdanto
 - systeemin tavoitteet ja tarkoitus
 - esittelee sovelluksen käsitemaailman
 - normaali johdannon rakenne (TIKI!)
- Järjestelmän yleiskuvaus
 - Sovellusalue
 - Järjestelmäkuvaus ja ohjelmiston rooli

- 3. Arkkitehtuurisuunnitelma
 - ratkaisun filosofia ja ohjelmiston toimintaperiaate
 - moduulit ja niiden väliset suhteet
 - tietokanta-arkkitehtuuri
 - huomaa yleis- ja detaljisuunnitelmat molemmat syytä esittää

- Yhdistelmä toiminnan ja tiedon kuvauksista: 4. Moduulikuvaukset
 - yleiskuvaus (esim UML-menetelmällä)
 - tietorakenteet
 - toiminnot
 - erityiset tekniset suunnitteluratkaisut
 - poikkeustilanteiden käsittely
 - ratkaisuvaihtoehtoja - miksi hylättiin?
 - Ratkaisun rajoitukset -- ylläpidettävyys
- Yhteys vaatimusedokumentin vast. kohtiin!

- Käyttöliittymä
 - tyypilliset käyttöskenaariot
 - prototyyppikuvaus tai kuvia muuten piirrettynä
- Muut liittymät: KJ, muut ohjelmistot, ...

-
- Toteutuksen rajoitukset
 - noudatettavat standardit
 - ohjelmointikielet, KJ, apuohjelmat
 - kokoarviot
 - tyyliohjeet

-
- Lähdeluettelo
 - Liitteet
 - kaavioiden symbolien kuvaukset
 - käytettyjen sisäisten kielten syntaksit
 - hallintoliitteet: muutokset aik. dokumentaatioon

Ohjelmiston suunnittelu

- Ohjelmiston suunnittelun tavoite
- Suunnitteluperiaatteet
 - modulaarisuus, kiinteys, kytkentä, ...
 - arkkitehtuurimallit, patternit, frameworkit
- Suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
 - olioperustaiset
- Suunnitteludokumentti, katselmus
- Erytisteemat: käyttöliittymät, suorituskyky

Erytisteemat: käyttöliittymät

- Käyttöliittymän oltava yhtenäinen
 - look and feel standardit
 - sovellusalueen käsitteistön käyttö
 - rauhallinen ulkoasu: yksinkertainen on kaunista! Ja tehokasta!
- Annettu palaute (tulosteet, virheilmot) oltava järkevää

Käyttöliittymät: palaute

- hyvä virheilmo ei nolaa käyttäjää vaan
 - auttaa ymmärtämään virheen syyn,
 - ilmaisee vaurioiden laajuuden ja
 - toipumiseen tarvittavat toimet

- Toiminnan ohjaus
 - soveltumattomat operaatiot eivät tarjolla
 - ei toisiinsa liittymättömiä asioita yhtäaikaan tarjolla (kiinteys!)
 - opastusjärjestelmä

- Toimintojen vahvistaminen
 - voisiko peruuttaa
 - vaatisiko vahvistuksen
 - transaktion käsite
- Älä rasita käyttäjän muistia
 - lyhytkestoinen muisti noin 5 pikku juttua, silmänräpäyksen ajan, ei yli toiminnon

Erityisteemat: suorituskyky

- Otettava huomioon kaikissa elinkaaren vaiheissa
- vaatimusanalyysi: riittävän tarkat suoritusaika- ja -tilavaatimukset
 - arviointitapa tarpeille?
- Arkkitehtuurisuunnittelu:
 - epäonnistunut tietorakenteiden ja tiedostojen valinta voi tuhota suorituskyvyn
 - jonoverkkoanalyysillä mahdollista etsiä kriittiset kohdat

-
- Moduulisuunnittelussa algoritmin valinta vaatimuksia vastaavasti
 - mistä vaihtoehtoiset algoritmit?
 - Dokumentin sisällettävä aika-tila-analyysit!
 - Koodaus ja testaus: liika viilaaminen epäolennaisissa paikoissa voi tuhota ylläpidettävyyden -- optimoi pullonkauloissa