

Ohjelmiston suunnittelu

- Suunnittelun tavoite
- Suunnitteluperiaatteet
- Perinteiset suunnittelumenetelmät
 - toimintokeskeiset
 - tietorakennepohjaiset
- Suunnitteludokumentti, katselmus
- Erityisteemat: käyttöliittymät, suorituskyky
- Olioperustaiset menetelmät

Olioperustaiset menetelmät

- Olioperustaisuus ohjelmistotuotannossa
- Olioiden kuvaaminen - peruskäsitteitä
 - olio, perintä, monimuotoisuus,
- Oliomallintaminen
 - menetelmien kirjolle yhteistäkin
 - staattinen
 - dynaaminen

Olioperustaisuus ohjelmistotuotannossa

- Analyysissä ja suunnittelussa sama peruskäsitteistö
 - oliot ja luokat; yhteyksissä abstraktiotasero
- oliomallien tehtävänä simuloida sovellusalueen ilmiöitä ja tapahtumia
 - ilmentävät luokat ja oliot yhdessä toimivia mutta itsenäisiä

Olioperustaisuus

- muutokset hallittavampia kuin perinteisessä moduulijakoon johtavassa top-down suunnittelussa?
 - Karkea kuvaus -> tarkennus alijärjestelmiksi -> ohjelmakomponentit/moduulit
 - Karkea objektimalli -> yksityiskohtainen objektimalli -> objekti

Olioperustaisuus

- Suunnittelu pohjautuu voimakkaasti tiedon kätkemisen periaatteelle
 - järjestelmä nähdään joukkona itsenäisiä olioita jotka vuorovaikutuksessa
 - oliolla tila johon voidaan koskea ainoastaan sovituilla operaatioilla
 - oliot kommunikoivat sanomien avulla

Olioiden kuvaaminen

- Kaksi osaa:
 - rajapinta: sanomat ja niihin liittyvät operaatiot
 - toteutuskuvaus: sisäinen tietorakenne, operaatioiden algoritmit
- käyttäjä tarvitsee vain protokollan kuvauksen; toteuttaja toteutuskuvauksen
 - tukee uuskäyttöä

Oliokäsitteitä

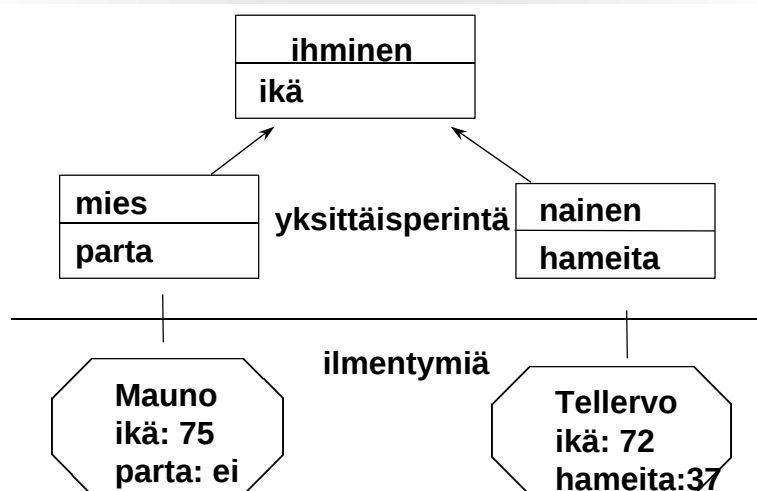
- Olio:
 - identiteetti
 - luokka/tyyppi (yhteistä samankaltaisille olioille)
 - attribuutit
 - operaatiot
 - elinkaari

Oliokäsitteitä - perintä

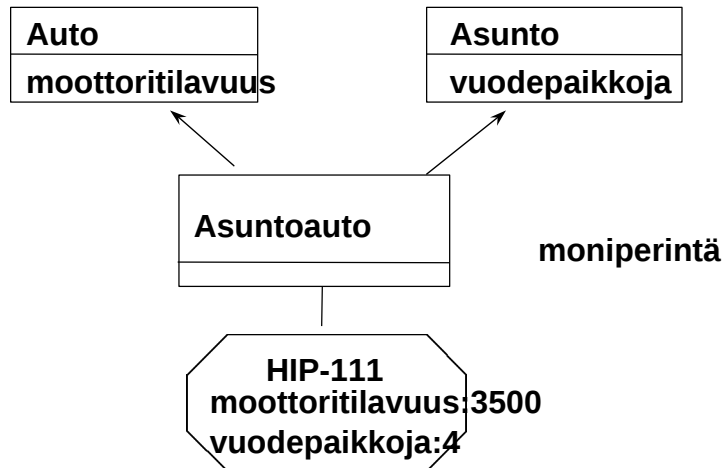
- Perintä (inheritance)
 - muotoa voidaan jakaa luokkien väliseen hierarkiaan perustuen
 - aliluokat (subclass) perivät muodon ylliluokiltaan (superclass)
 - vastaanottovirkailijakin on ihmisen muotoinen

- aliluokat erikoistavat (specialise) yleisiä ylliluokkia
 - erikoistettu aliluokka voi lisätä perittyyn muotoon omia ominaisuuksia
 - aliluokka voi uudelleenmääritellä (syrjäyttää) (override) perimänsä ominaisuuden
- ominaisuudet attribuutteja (tietorakenteita) tai operaatioita

Oliokäsitteitä - perintä



Oliokäsitteitä - perintä



Oliokäsitteitä - perintä

- perintä on merkittävin ero olio-ohjelmoinnin ja muiden ohjelmointiparadigmojen välillä.
- Perinnän päätavoitteet:
 - käsitteellinen mallintaminen
 - yleistäminen/erikoistaminen
 - pääpaino ohjelmistokehityksen määrittely- ja analyysivaiheessa
 - koodin uudelleenkäyttö
 - attribuuttien ja operaatioiden periminen
 - kasvattava ohjelmistokehitys
 - pääpaino suunnittelu- ja toteutusvaiheessa

Oliokäsitteitä

- Monimuotoisuus (polymorphism)
 - nimeen kytketyn toiminnan erilaisuus eri (sukulais)luokissa
 - olio suorittaa nimetyn operaation oman luokkansa mukaisesti (=> olio voi kuulua vain yhteen luokkarakenteen lehtiluokkaan, eli luokkaan, jolla ei ole aliluokkia)
 - sihteeri.maksaa = {antaa rahaa tarjoilijalle}
 - johtaja.maksaa = {pyytää, että sihteeri.maksaa}

- nimettyä operaatiota pyytävän olion ei tarvitse tietää mihin luokkaan operaation suorittaja todella kuuluu
 - tarjoilija voi pyytää maksua, joko sihteeriltä tai johtajalta (ja saa aina rahat)
- myöhäinen sidonta, operaation toteutustapa ratkeaa vasta operaatiota pyydettyäessä
 - laskun maksutapa selviää vasta kun lasku esitetään

Oliomallinnusmenetelmiä

- Booch

- luokat & oliot; semantiikka; yhteydet; tarkenna; toteuta

- Coad & Yordon

- luokat, hierarkkia, osa-alueet, attribuutit, palvelut

- Jacobsson

- käyttötapaukset, luokat & yhteydet

- Rumbaugh

- luokat, yhteydet, hierarkkia, käyttöskenaariot, viestiyhteykskaaviot, tietovirrat

Oliomallintaminen

- Mallinnettaessa järjestelmää olioilla erotetaan yleensä kaksi puolta:

- staattinen

- millaisia luokkia - tietosisältö, yhteydet, rakenteet
- mitä palveluja tarjolla

- dynaaminen

- miten oliot toimivat yhteistyössä
- millainen on olion elinkaari

Oliomallintaminen -1

- Staattinen malli
 - ehdokasluokkien tunnistaminen; tekstianalyysiesim
 - ehdokasluokasta luokaksi;
 - suunnittelun eteneminen
 - käyttötapauslähtöinen suunnittelu
- MVC-arkkitehtuuri: objektiohjelmistoarkkitehtuuri

-
- Eri menetelmät tarjoavat erilaisia malleja sille, mitä oliokeskeisen määrittely/suunnitteluvaiheen yhteydessä pitäisi tehdä ja miten.
 - Seuraava “geneerisen esimerkin” asemassa

Olioiden/luokkien tunnistaminen

- jäsennettävä sovellusalue,
 - tunnistettava toimijat ja heidän tehtävänsä,
 - selvitettävä mitä tietoa toimijat tarvitsevat kyetäkseen toimimaan

Olioiden/luokkien tunnistaminen - tekstin analysointi

- Laadi ongelmaa kuvaava teksti
- Analysoi kuvausta, etsi lauseista luokkaehdokkaita, alleviivaa ne
- Luokkaehdokkaita ongelman kannalta olennaiset substantiivit
- Tunnista ja karsi synonyymit
- Valitse luokat

- Määritä attribuutit

- etsi tekstistä ilmiöitä, jotka kuvailevat valittuja olioita

- Määritä yhteydet

- etsi tekstistä kuvauksia yhteyksistä, rakenteita, joissa oliot ovat mukana

- Määritä palvelut

- etsi tekstistä tekemiseen ja toimintaan liittyviä ilmauksia, kohdista ne luokkiin

Olioiden/luokkien tunnistaminen - tekstin analysointi

- Luokkina tulevat kyseeseen, esim.

- ulkoiset oliot

- tahot, jotka tuottavat tai saavat tietoa järjestelmältä
- toiset järjestelmät, ihmiset, laitteet
- tietosisältöön kuuluvat asiat
- raportit, näytöt, kirjeet, signaalit

- **tapahtumat**, joihin järjestelmän pitäisi reagoida
- **roolit**, jotka edustavat erilaisia tapoja toimia järjestelmän kanssa
- **organisaatioyksiköt**
- **paikat**, jotka luovat ympäristön ongelmalle
- **rakenteet**
- **Toiminnot eivät ole luokkia**

Esimerkki

- SafeHome software enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system and interacts with the homeowner through a key pad and function keys contained in the SafeHome control panel .

- During installation, the SafeHome control panel is used to 'program' and configure the system. Each sensor is assigned a number and type, a master password is programmed for alarming and disarming the system, and telephone numbers are input for dialing when a sensor event occurs.

Esimerkki

- When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during the system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location, reporting the nature of the event that has been detected. The number will be redialed every 20 seconds until telephone connection is obtained.

- All interaction with safeHome is managed by a user interaction subsystem that reads input provided through the key pad and function keys, and displays prompting messages and system status on the LDC display. Keyboard interaction takes the following form ...

Esimerkki - ehdokkaita

homeowner	ulkoinen olio / rooli
sensors	ulkoinen olio
key pad	ulkoinen olio
function keys	ulkoinen olio
control panel	ulkoinen olio

installation	tapahtuma
type	attribuutti
number	attribuutti
telephone numbers	asia
audible alarm	ulkoinen olio
security system	organisaatioyksikkö / asia
system status	attribuutti
interaction subsystem	organisaatioyksikkö / asia
master password	asia
sensor event	tapahtuma

Esimerkki - ehdokkaita lisää

nature of the event	attribuutti
<u>configuration activities</u>	toiminta
telephone connection	ulkoinen olio
prompting messages	asia
LDC display	ulkoinen olio
monitoring service	organisaatioyksikkö
location	paikka

Ehdokasluokasta luokaksi - kriteerejä (Coad & Yourdon)

- **kaikkien pitäisi toteutua**
- (1) muistamistarve
 - tiedon säilyttäminen luokan olioista on välttämätöntä järjestelmän toiminnan kannalta
 - sisältää muistettavaa tietoa

- (2) palvelujen tarjonta
 - luokan olioiden täytyy tarjota palveluita, luokan attribuuttien käsittelyyn
- (3) moniattribuuttisuus
 - määrittelyvaiheessa yksiattribuuttiset oliot harvoin ovat oleellisia ongelman kannalta

Ehdokasluokasta luokaksi - kriteerejä (Coad & Yourdon)

- (4) attribuutien jaettavuus
 - luokkaan on liitettävissä joukko sen kaikille esiintymille soveltuvia attribuutteja
- (5) palvelujen jaettavuus
 - luokkaan on liitettävissä joukko palveluita, jotka luokan kaikki esiintymät kykenevät suorittamaan
- (6) olennaisuus ongelman kannalta

Esim. ehdokkaiden analysointi

homeowner	ulkoinen olio / rooli	-1, -2, +6
sensors	ulkoinen olio	+ kaikki
key pad	ulkoinen olio	-6
function keys	ulkoinen olio	-6
control panel	ulkoinen olio	+kaikki
installation	tapahtuma	-1, -2

type	attribuutti	
number	attribuutti	
telephone numbers	asia	-3
audible alarm	ulkoinen olio	-1, + muut
security system	organisaatioyksikkö / asia	+ kaikki
system status	attribuutti	
interaction subsystem	organisaatioyksikkö / asia	-6
master password	asia	-3
sensor event	tapahtuma	+ kaikki

Esim. ehdokkaiden analysointi

nature of the event	attribuutti	
<u>configuration activities</u>	toiminta	
telephone connection	ulkoinen olio	-1,-2,-3,
prompting messages	asia	-1,-6
LDC display	ulkoinen olio	-6
monitoring service	organisaatioyksikkö	-1, -2
location	paikka	-1,-2

Esim. ehdokkaiden analysointi

● Attribuutteja:

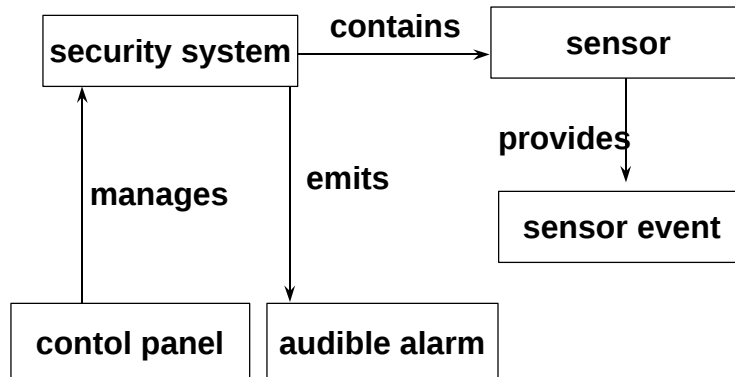
- sensor
 - type
 - number
 - tuning data
- control panel
 - function keys
 - LDC display
 - messages
- sensor event
 - nature of event
 - telephone numbers

Esim. ehdokkaiden analysointi

- audible alarm
 - alarm specs.
 - Delay
 - state
- security system
 - telephone numbers
 - system status
 - location
 - delay time
 - master password
 - no of tries

Esim. ehdokkaiden analysointi

● Yhteydet



Esim. Ehdokkaiden analysointi

● Palvelut

– sensor

- turnOn
- turnOff
- adjust

– control panel

- get Input
- show State
- show Message

- audible alarm

- Emit Alarm
- adjust delay

Esim. ehdokkaiden analysointi

- palvelut jatkuu
 - security system
 - Turn on
 - Turn off
 - Register location
 - Adjust delay
 - Identify
 - Process event
 - sensor event
 - Set connection

Suunnittelun eteneminen

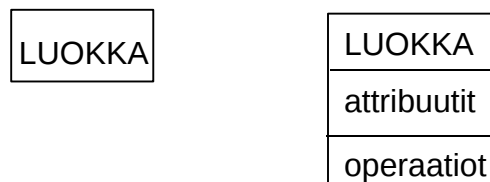
- tieto- tai toimintokeskeisiä painotuksia
 - tietokeskeisen oliosuunnittelun lähtökohtana on tiedon piilottaminen:
 - olioluokkia löydetään tietoabstraktioiden kautta
 - toimintokeskeistä painotusta edustaa ns. vastuuseen perustuva oliosuunnittelu
 - olioiden määrittelyperustana on olion vastuu jostakin (yleensä toiminnallisesta) kokonaisuudesta ja
 - olion tietosisältö määräytyy vastuun mukaisesti
- johtavat eril. olio- ja luokkarakenteeseen

Missä oltiin?

- Mallinnettaessa järjestelmää olioilla erotetaan yleensä kaksi puolta:
 - staattinen
 - millaisia luokkia - tietosisältö, yhteydet, rakenteet
 - mitä palveluja tarjolla
 - dynaaminen
 - miten oliot toimivat yhteistyössä
 - millainen on olion elinkaari

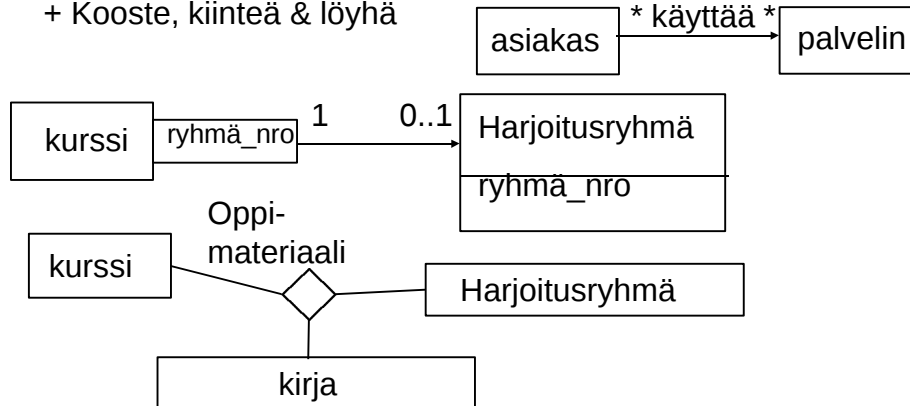
Staattinen malli

- Esim. UML-luokkakaaviot



● Ja luokkien väliset yhteydet

+ Kooste, kiinteä & löyhä



Suunnittelun eteneminen

- Järj. toiminnan kuvausta tarkennetaan, pyrkimyksenä löytää olioiden yhteistyöketju:
 - Liikkeelle lähdetään käyttäjän toiminnan vastaanottavista olioista ja niiden luokista.
 - Liittymäoliolle määritellään kttäjäoperaation vastaan-ottava(t) palvelu(t).
 - Toiminta tarkennetaan toimenpiteiksi ja ehdoiksi, joiden vallitessa toimenpide suoritetaan.

Suunnittelun eteneminen

- Suorittajalla on oltava riittävästi tietoa palvelun suorittamiseksi.
 - Lisätään tarvittaessa luokalle uusia ominaisuuksia tai palvelulle parametreja.
 - varmistetaan, että palvelua käyttävät oliot ja palvelut kykenevät ne toimittamaan.
 - Luokkaan voidaan myös liittää avustajia, joille voidaan delegoida osa toiminnasta.

Suunnittelun eteneminen

- Tunteminen voi perustua:
 - avustava olio yleisesti tunnettu (globaali)
 - avustaja ilmoitetaan tarvitsijalle palvelua pyydettäessä (parametri)
 - oliosta on tietorakenteiden avulla luotu yhteys avustavaan olioon.
- Olorakenteet ja yhteistyö suunnitellaan samanaikaisesti

MVC-arkkitehtuuri

- Model-view-controller -arkkitehtuurimalli
- eristetään tietosisältö (model) sen esittämisestä (view) ja siihen kohdistuvan käsittelyn hallinnasta (control)
 - sisältöoliot
 - näkymäoliot
 - ohjausolio

- Sisältöoliot
 - pitävät sisällään kaikki esitettävät tiedot
 - tietoja muuttavat operaatiot
 - käyttöliittymästä riippumattomia
- esim sisältöolio kello:
 - kellonaika, ajan kysely, ajan asetus

- Näkymäoliot
 - esittävät niihin liitetyn sisältöolion tilaa
 - käyttävät sisältöolion operaatioita tilatiedon hankkimiseen
- esim kellolla kaksi näkymäoliota
 - analogianäyttö
 - digitaalinäyttö

- Ohjausolio
 - miten käyttäjä voi vaikuttaa sisältöolioiden tietoihin
 - “input”, jos näkymäolio tarjosi “output”-puolen sisältöolioihin
- Esim. Kellonajan asettaminen

MVC - arkkitehtuurimalli

- Sisältöolioon voi liittyä monta näkymäoliota ja ohjausoliota
 - rekisteröinti
- onko ohjausolio kytketty suoraan sisältöolioon vain näkymäolion kautta?

Käyttötapauslähtöinen suunnittelu

(Laine: Olioiden maihinnousu, luku 6.4, Suomen Atk-kustannus, 1997)

- Käyttötapauslähtöisessä suunnittelussa suunnittelu etenee käyttötapaus kerrallaan.
- Suunnittele kokonaisvaltaisesti käyttöliittymän perusratkaisu
 - esim ikkunat, painikkeet, valikot, yms.

Käyttötapaus

1. Selvitä käyttäjät ja laitteet jotka käyttävät suunniteltavaa järjestelmää
2. Selvitä käyttäjien roolit. Samalla käyttäjällä voi olla monta roolia.
3. Ryhmitellään rooliluokiksi -> aktori
4. Selvitetään kunkin aktorin vaatimat päätehtävät
5. Tehtävien vaatimat tiedot

-
6. Mitä tietoa aktorit tarjoavat järjestelmälle?
Ulkoisen ympstön muutokset?
 7. Mitä tietoa aktori tarvitsee järjestelmältä?
 8. Mistä poikkeustilanteista aktori tarvitsee tietoa?

Käyttötapauslähtöinen suunnittelu

- Käy läpi järjestelmän käyttötapaukset
 - laadi käyttötapauskohtainen näkemys järjestelmän oliorakenteesta (vain ne luokat, jotka osallistuvat toimintaan)
 - Käyttötapauskohtaisessa näkymässä kuvataan vain ne osat oliomallista, jotka ovat välttämättömiä käyttötapauksen kannalta.
 - hahmottele käyttötapauksen läpivientiin liittyvä olioysteisyö

Käyttötapauksen läpivientimallit

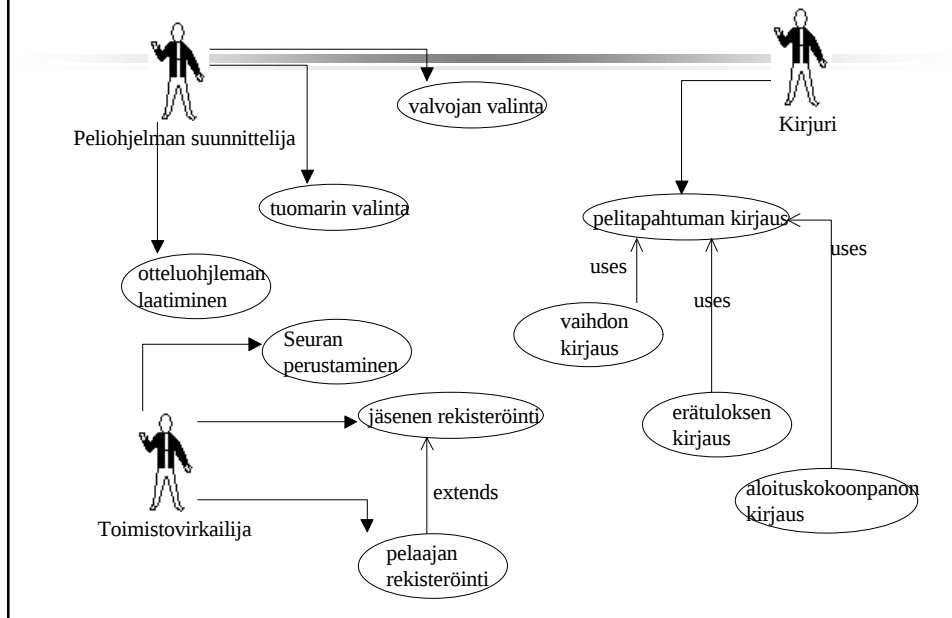
- käyttötapauksen läpivientimalli taulukkona

käyttäjä-tehtävä	suoritustapa	järjestelmän toiminta
tehtävä asia	kuvaus tavasta, jolla tehtävä tehdään	karkeahko kuvaus siitä, miten järjestelmä reagoi käyttäjän toimiin
...		

Käyttötapauksen läpivientimallit

Käyttäjä-tehtävä	Suoritustapa	Järjestelmän toiminta
Valitse luokka (a)	1. Vie kaavioikkunassa hiirikursori luokkasymbolin päälle 2. Näpäytä hiiren vasenta näppäintä	Järjestelmä tutkii oliko näpäytyssä kohdassa luokkasymboli. Jos oli, niin vanha valintajoukko tyhjennetään ja valittua symbolia vastaava luokka lisätään valintajoukkoon. Valintajoukkoon kuuluvat näytetään ruudulla valittuina

UML - use cases

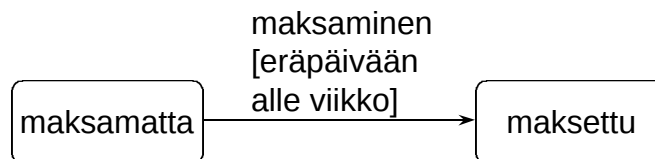


Oliomallintaminen - 2

- Dynaaminen malli
 - Olioiden käyttäytymisen kuvaus
 - Olioiden elinkaarien kuvaukset
 - Yhteistyökaaviot
 - CRC-mallitus (Class-Responsibility-Collaborator Modeling) apuna
 - UML-kaaviot

UML - life cycle

- Siirtymään voidaan liittää ehto, joka säätelee siirtymismahdollisuutta.

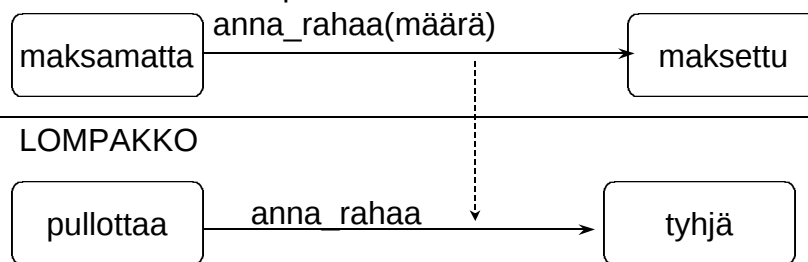


Laskua ei saa maksaa liian aikaisin.

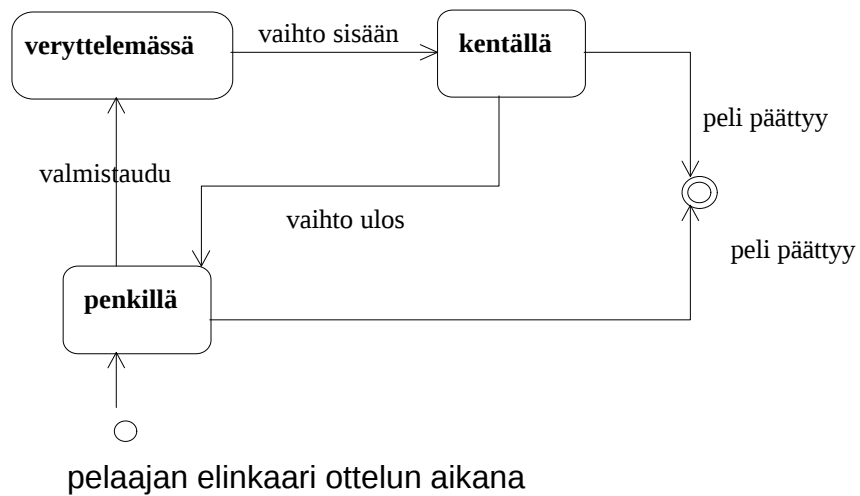
UML - life cycle

- Siirtymään voidaan liittää myös toimintaa tai viestin lähetys toiselle oliolle.

LASKU: maksaminen
[eräpäivään alle viikko]
/ lompakko.



UML - elinkaari



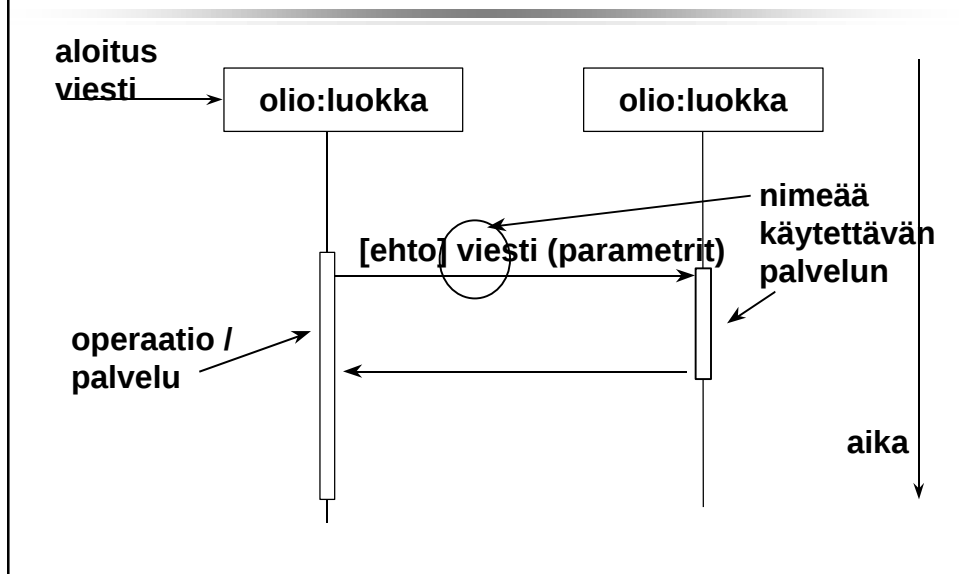
UML - yhteistyö

- **Sekvenssikaavio** (sequence diagram)
 - operaatioiden tapahtumajärjestys / aika!
 - toimintaan liittyvien viestien kulkua.
 - OMT: event trace, OOSE: interaction diagram, Mainstream objects: transaction sequence, Olioiden maihinnousu: yhteistoimintakaavio
- **Yhteistyökaavio** (collaboration diagram)
 - olioiden väliset kytkennät

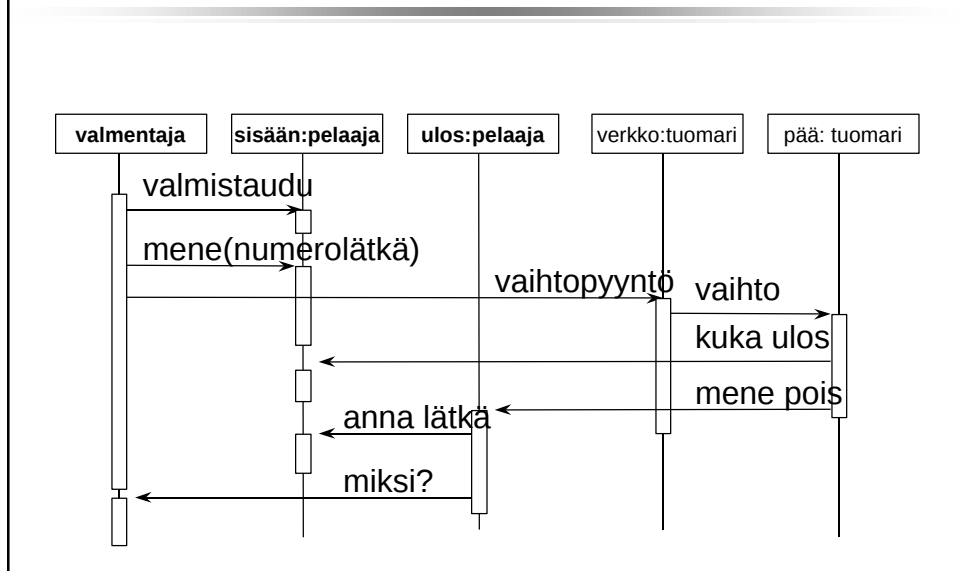
UML - yhteistyö

- **Sekvenssikaavio kuvaa:**
 - käyttötapauksen ilmentymään liittyvän olioiden yhteistyön
 - käyttötapauksen ilmentymästä käytetään esim. OMT- menetelmässä nimitystä skenaario (scenary) = **tietty asioiden kulku**
 - olion palvelun suoritukseen liittyvän olioiden yhteistyön

UML - yhteistyö



UML - yhteistyö

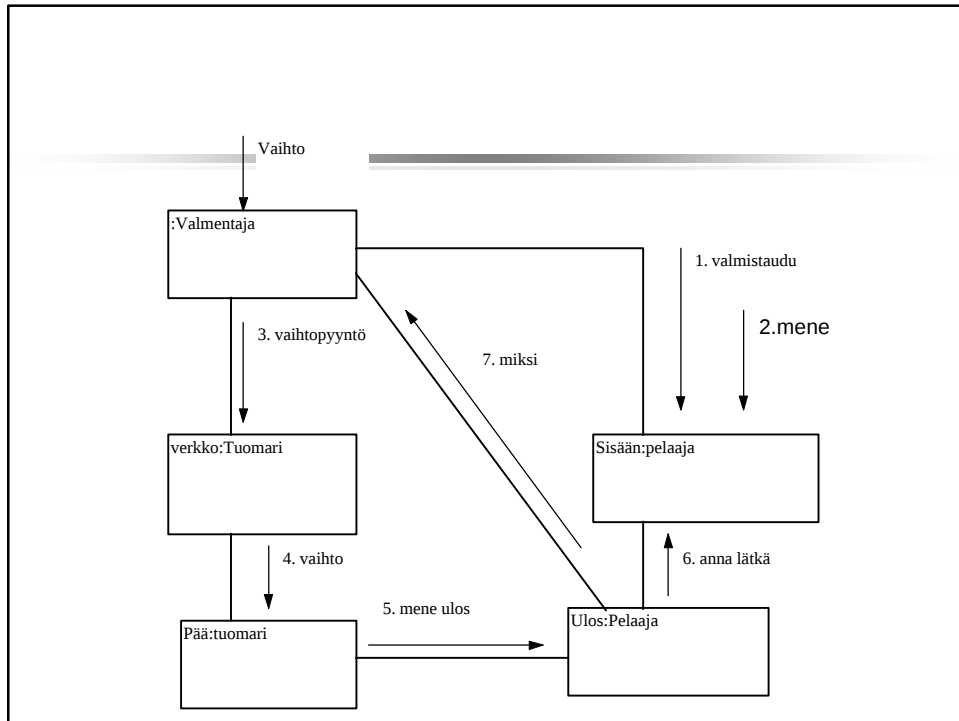


UML - yhteistyö

- valmistaudu
- mene(numerolätkä)
- vaihtopyyntö
- vaihto
- kuka ulos
- mene pois
- riisuu veryttelypuvun
- ottaa numerolätkän, kävelee kentän laidalle
- puhalttaa pilliin, antaa käsimerkin päätuomarille, tarkistaa pöytäkirjan
- katsoo sisään tulijan lätkästä kuka ulos, osoittaa ulosmenijän
- heiluttelee lätkää
- kävelee laidalle, ottaa lätkän

UML - yhteistyö

- UML:n yhteistyökaavio
 - Pohjana oliokaavio. Yhteyksiin liitetään niiden kautta kulkevat viestit. Kuvausta käytetään samaan tarkoitukseen kuin sekvenssikaaviota.
 - kaaviolla voidaan varmistaa, että oliomalli tarjoaa tarvittavat kytkennät palvelujen saamiseksi (viestien välittämiseksi)



UML - yhteistyö

- yhteydet olioiden välillä tarkoittavat sitä, että pelaajat, valmentaja ja tuomarit näkevät toisensa kentällä. Oliot ovat siis tavallaan globaalisti tunnettuja.
- **Palvelua voi pyytää vain tunnetulta oliolta.**
- Numero viestin edessä kuvaa viestin järjestyksen yhteistyöketjussa.