Ohjelmointi 1 / syksy 2007
Ensimmäinen luento

Paavo Nieminen

nieminen@jyu.fi

Tietotekniikan laitos

Informaatioteknologian tiedekunta

Jyväskylän yliopisto

Luennon suorittava ”Java-ohjelma”

```
/**
 * Leikkimielinen "ohjelma", joka toimii "luentokalvoina"
 * Ohjelmointi 1 -kurssin ensimmäisellä luennolla.
 *
 * @author Paavo Nieminen
 * @version 0.1
 */
public class EkaLuentoOhjelma{
    public static void main(String[] args){
        YliopistoKurssi ohj1 = new OhjelmointiYksiKurssi();
        ohj1.toteutaLuento(1);
        // FIXME: Ei vielä toteuta luentoa numero 2:
        // ohj1.toteutaLuento(2);
    }
}
```

Tietty kurssi ”Javalla mallinnettuna”

```
/**
 * Tämä on johdantokurssi ohjelmoinnista.
 */
public class OhjelmointiYksiKurssi extends YliopistoKurssi
        implements VerkkoKurssi, LukioonSoveltuva {

    /** Oletuksena syksyn 2007 kurssikerran kokoonpano */
    public OhjelmointiYksiKurssi(){ ... }
    public void toteutaLuento(int luennonNumero){ ... }
    public void taukoMinuutteja(double m){ ... }
    public void arvostele(){ ... }
    ...
}
```

HUOM: Kolme pistettä ”...” ohjelmalistauksessa tarkoittaa usein, että ”tässä kohtaa on oikeasti paljon muutakin, mutta juuri nyt ei olla tarkemmin kiinnostuneita juuri siitä”.

Abstrakti yliopistokurssi ("yläkäsite" Ohjelmointi 1:lle)

```
/** Kaikille yliopistokursseille yhteiset ominaisuudet. */
public abstract class YliopistoKurssi{
    public static final int MAX_ARVOSANA = 5;
    public static final int MIN_ARVOSANA = 1;

    protected double opintopistemaara;

    protected Henkilo luennoitsija;
    protected Henkilo[] tuntiopettajat;
    protected Henkilo[] opiskelijat;
    protected double[] opiskelijoidenTenttipisteet;

    public abstract void arvostele();
    public abstract void toteutaLuento(int monesko);
}
```

HUOM: Entä SUOR/HYL -arvostelu? Onko mahdollista??

Suorittaminen alkakoon!

```
1  /**
2   * Leikkimielinen "ohjelma", joka toimii "luentokalvoina"
3   * Ohjelmointi 1:n ensimmäisellä luennolla. ÄLÄ MIETI liikaa!
4   * Pidä tätä ihan tavallisena kalvoesityksenä, joka ''näyttää
5   * vähän kummalta'' ...
6   * @author Paavo Nieminen
7   * @version 0.1
8   */
9  public class EkaLuentoOhjelma{
10     public static void main(String[] args){
11         YliopistoKurssi ohj1 = new OhjelmointiYksiKurssi();
12         ohj1.toteutaLuento(1);
13     }
14 }
```

HUOM: Rivinumerot eivät kuulu ohjelmaan, mutta ne helpottavat selittämistä.

Kurssiyksilö muodostuu suoritta-malla tarvittava alustus!

```
1  public class OhjelmointiYksiKurssi extends YliopistoKurssi
2      implements VerkkoKurssi, LukioonSoveltuva {
3
4      /** Oletuksena muodostetaan syksyn 2007 kurssin kokoonpano */
5      public OhjelmointiYksiKurssi(){
6          luennoitsija = Jyu.henkkuntaNimella("Paavo Nieminen");
7
8          tuntiopettajat = new Henkilo[4];
9          tuntiopettajat[0] = Jyu.henkkuntaNimella("Jarmo Ernvall");
10         tuntiopettajat[1] = Jyu.opiskelijaNimella("Joel Lehtonen");
11         tuntiopettajat[2] = Jyu.opiskelijaNimella("Juho Tamminen");
12         tuntiopettajat[3] = Jyu.opiskelijaNimella("Irene Venäläine
13
14         opiskelijat = Jyu.ketkaKurssilla("Ohjelmointi 1 syksy 2007
15         opintopistemaara = 6.0;
16     }
```

Suorittaminen jatkuu... Tapahtuu luento!

Muistetaan suoritusjärjestys: ensin piti muodostaa kurssi (1. rivi) Sen jälkeen voi pyytää kurssin tarjoamaa palvelua, luennon numero 1 toteuttamista:

```
YliopistoKurssi ohj1 = new OhjelmointiYksiKurssi();
ohj1.toteutaLuento(1);
```

...ja OhjelmointiYksiKurssi sitten toteuttaa luennon ...

```
17      public void toteutaLuento(int luennonNumero){
18          if (luennonNumero == 1){
19              luennoitsija.esitteleItsesi();
20              luennoitsija.otaOpetettavaksi(opiskelijat);
21              luennoitsija.virittaydyTunnelmaan("Ohjelmointi 1, s07"
22
23              luennoitsija.kerroAiheesta("Kurssin kuvaus");
24              luennoitsija.kerroAiheesta("Suoritusvaatimukset");
25              luennoitsija.kerroAiheesta("Arvostelu");
26              luennoitsija.kerroAiheesta("Aikataulu");
```

(jatkoa edellisestä.)

```
27              luennoitsija.kerroAiheesta("Demot");
28              luennoitsija.kerroAiheesta("Harjoitustyö");
29              luennoitsija.kerroAiheesta("Oikotiet / tenttiminen");
30
31              for (Henkilo tuntiope : tuntiopettajat){
32                  tuntiope.esitteleItsesi();
33              }
34
35              luennoitsija.vastaaKysymyksiin();
36              taukoMinuutteja(15.0);
```

(jatkoa edellisistä.)

```
38                luennoitsija.kerroAiheesta("Mitä on ohjelmistotyö");
39                luennoitsija.kerroAiheesta("Mitä on ohjelmointi");
40                luennoitsija.kerroAiheesta("Tämän kurssin rooli");
41                luennoitsija.kerroAiheesta("Opittavat käsitteet");
42                luennoitsija.kerroAiheesta("Yleiskuva kurssin luennois
43                luennoitsija.kerroAiheesta("Toimintaympäristömme");
44                luennoitsija.kerroAiheesta("Tämän viikon kotitehtävät"
45                luennoitsija.vastaaKysymyksiin();
46                luennoitsija.kiitaJaPoistu();
47            }
48        }
```

HUOM: Vaikka ”Java-ohjelmana” ensimmäinen luento päättyisi näin, todellisessa maail-

massa jatketaan vielä yhteenvedolla, jos suinkin jää aikaa.

Ohjelma on päättynyt.

”Sovellusohjelmassa” oli kaksi riviä, jotka on nyt suoritettu:

```
1  /**
2   * Leikkimielinen "ohjelma", joka toimii "luentokalvoina"
3   * Ohjelmointi 1:n ensimmäisellä luennolla. ÄLÄ MIETI liikaa!
4   * Pidä tätä ihan tavallisena kalvoesityksenä, joka ''näyttää
5   * vähän kummalta'' ...
6   * @author Paavo Nieminen
7   * @version 0.1
8   */
9  public class EkaLuentoOhjelma{
10     public static void main(String[] args){
11         YliopistoKurssi ohj1 = new OhjelmointiYksiKurssi();
12         ohj1.toteutaLuento(1);
13     }
14 }
```

Tehdään yhteenveto ja huomioita.

- Vaikka tavoite oli puhtaasti olla kalvosarja, kaikki Java-ohjelmointikielellä kirjoitettu toimii ohjelmana, joka voidaan suorittaa tietokoneella. ($\rightarrow$ näytetäänpä esimerkki videoscreenillä)
- Ohjelma on yksi tapa ilmaista asioita täsmällisesti.
- Eritoten ohjelmalla voi kuvata asioiden tapahtumista jossain järjestyksessä joidenkin tapahtumien (tai kuten meilläpäin sanotaan, *syötteiden*) ilmetessä.

Näkökulma ohjelmointiin: mallintaminen

- Ohjelma *voi olla malli reaalimaailmasta* eli ns. jonkinlainen *simulaattori*. On vaikea edes keksiä ohjelmaa, joka ei olisi malli jostakin joko todellisesta tai epätodellisesta sydeemistä, esim:
  - pelit ovat malleja virtuaalimaailmoista
  - pankkijärjestelmät ovat malleja tileistä, asiakkaista, rahavirroista ym.
  - musiikkisoftat ovat malleja nuoteista, instrumenteista, reaalimaailman ääni-ilmiöistä ym.
- Silloin ohjelma on useimmiten epätäydellinen ja/tai epätarkka verrattuna todellisuuteen.
- *Mallit useimmiten ovat! Siltä ei voi välttyä!*

Ohjelmointikielen rooli

- Ohjelmointikielellä ilmaiseminen on vain yksi tapa esittää malli.
- Se on *tarkin esitysmuoto ohjelmalle* — ja siinäpä sen vaikeus onkin . . . Jotta yksityiskohtainen ohjelma on mahdollista kirjoittaa tai ymmärtää, täytyy sitä ensin pohtia jollain *yksinkertaisemmilla tavoilla*! Suoraan ohjelmointikieleen ei ole mahdollista hypätä (paitsi triviaaleissa ohjelmissa, joista ei ole kunnon hyötyä kenellekään. . .)!
- Ohjelmointikieli on välttämätön, koska tieto(kone)järjestelmä ymmärtää vain sellaista.
- Silti ohjelmointi on paljon enemmän kuin kieli! Sanoisin, että ohjelmointi on eräänlainen mallintamisen taito.

Malli on ”näkymä” ohjelmaan.

Ohjelmointikieli on siis tarkin mahdollinen malli. Mitä yksinkertaisempaa/yleispätevämpää on? Vain pari esimerkkiä:

- Algoritmimallit (esitys esim. ”pseudokoodina”)
- Oliomallit (luokittelu, luodut yksilöt)
- Tapahtumien järjestyksen ja osallistujien kuvaukset
- Käyttöliittymäkuvaukset (esim. ”screenshotit”)
- Järjestelmämallit (mukana ihmiset, laitteet, tietovarastot, kommunikaatio, . . .)

HUOM: Olennaista on osata mallin asteittainen tarkentaminen: Ensin karkea malli, sitten

hienojakoisempi ja hienojakoisempi . . . lopulta ohjelmointikieltä . . . mutta vasta lopulta!

Näyttäisinkö esimerkkejä mallinnuksesta?

- Vaikeus: Pitäisi kuulemma välttää ”perinteistä yliopistovirhettä”, että asioiden esitys menee yli hilseen . . . Tarkoitus on oppia ”vain ohjelmoimaan”, joten kannattaako edes näyttää mitään muuta?
- Luennoitsijan näkemys 1: abstraktit esitysmuodot on nimenomaan kehitetty yksinkertaistamaan mm. ohjelmien ymmärtämistä! Niitä katsomalla pitäisi kenen tahansa vastaantulijan tajuta. . . ainakin jotain . . .
  Mm. siksi ne ovat paljolti kuvallisia ja jättävät kertomatta paljon yksityiskohtia.

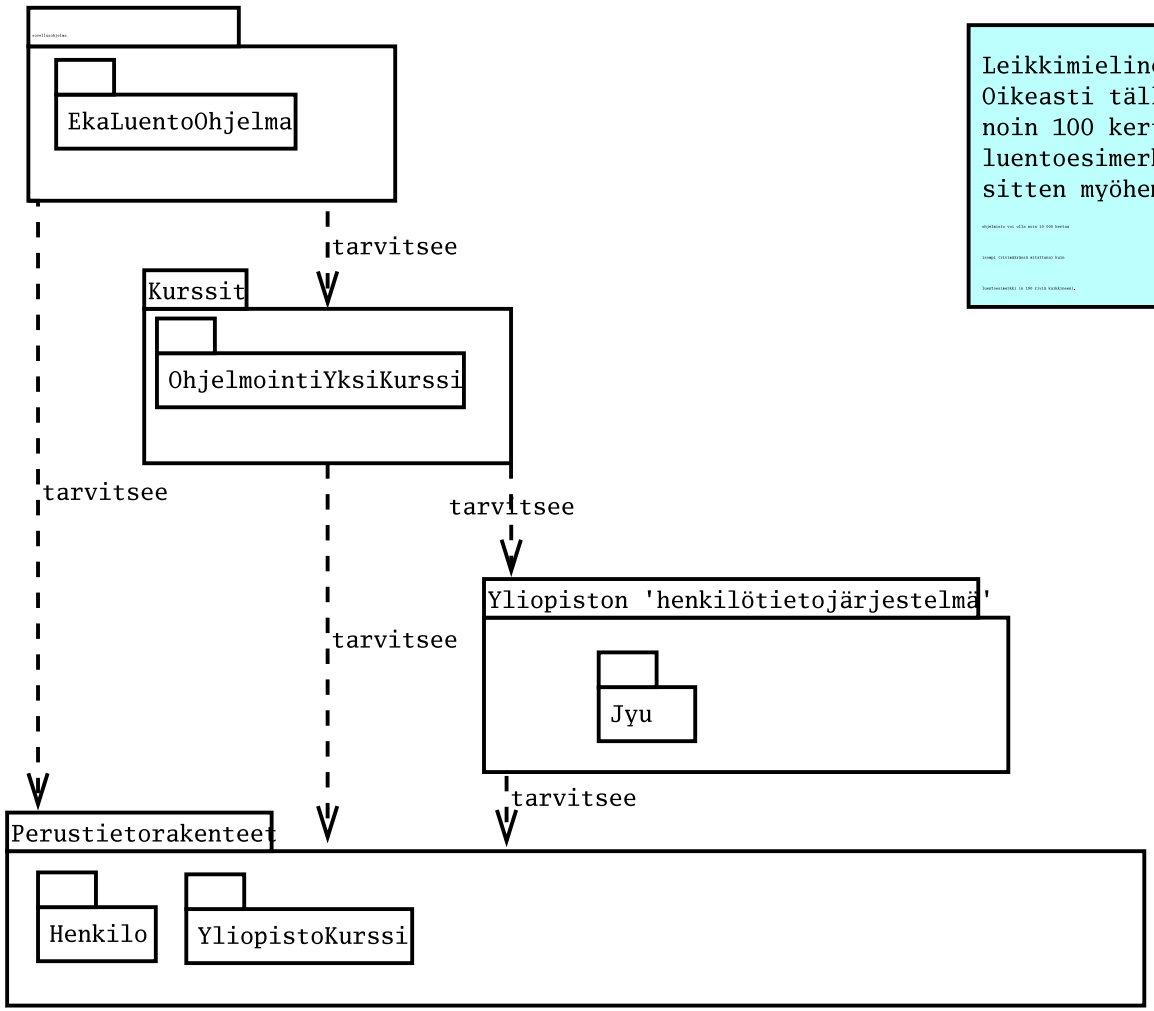
Näyttäisinkö sittenkään esimerkkejä?

- Luennoitsijan näkemys 2: Ohjelmointitaito on välttämätön näiden kuvien *piirtämiselle* ja ns. *”täydelliselle” ymmärtämiselle*, mutta ei niiden *katsomiselle*.
- Luennoitsijan oletus: Näiden katselu tukee ja helpottaa (mutta ei yksinään tee mahdolliseksi!) ohjelmoinnin oppimista
- Fakta: Näiden mallien syvällinen merkitys ja se, miten näitä kannattaa tehdä, on ehdottomasti jatkokurssien asia (mm. Oliokeskeinen tietojärjestelmien kehittäminen, Johdatus ohjelmistotekniikkaan, . . .).

Taidanpa siis näyttää vähän kuvia...

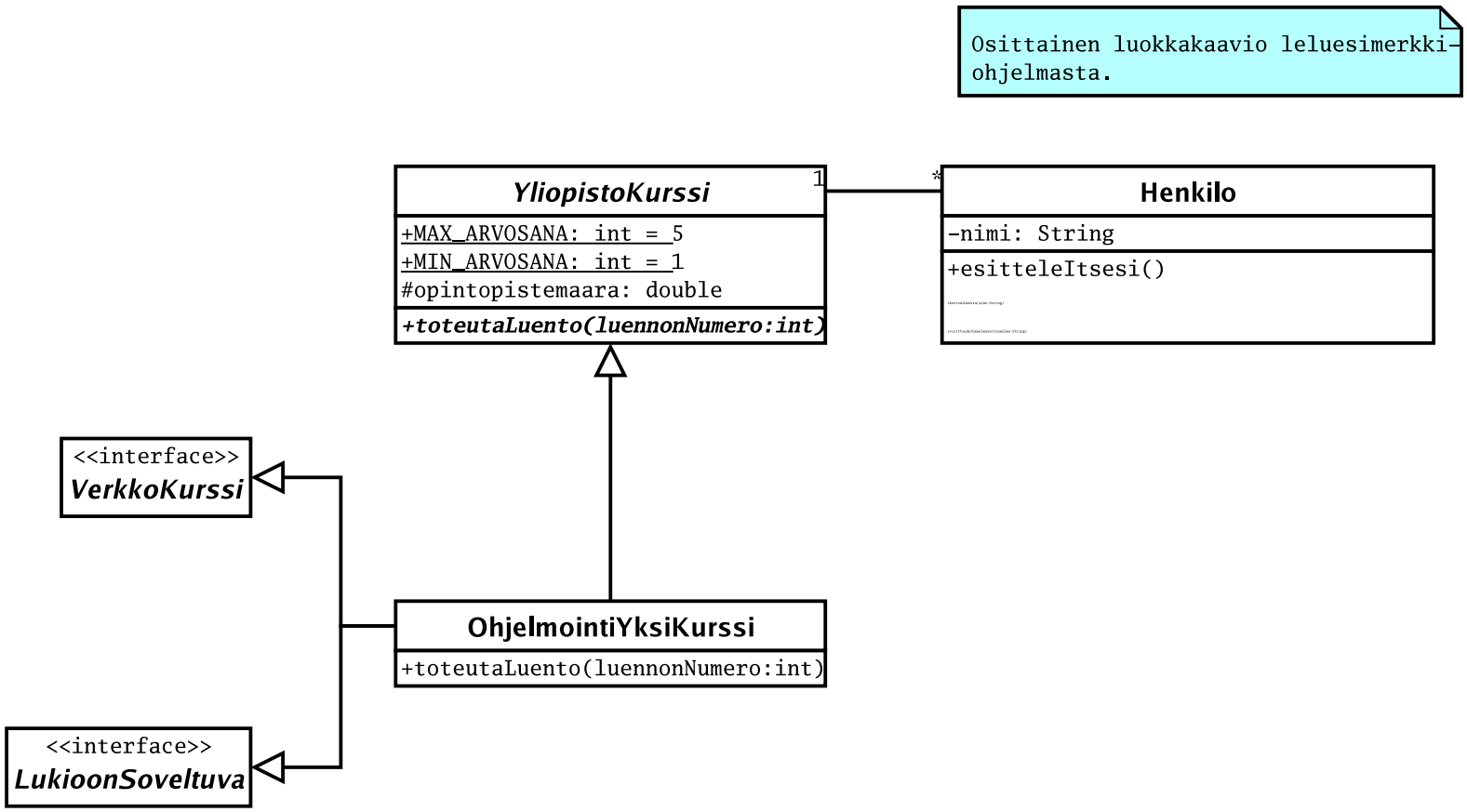
- Aika paljon IT-opinnoista tulee pyörimään näiden parissa, joten mielelläni otan ne mukaan heti aluksi!

Esim: Moduulijakokaavio

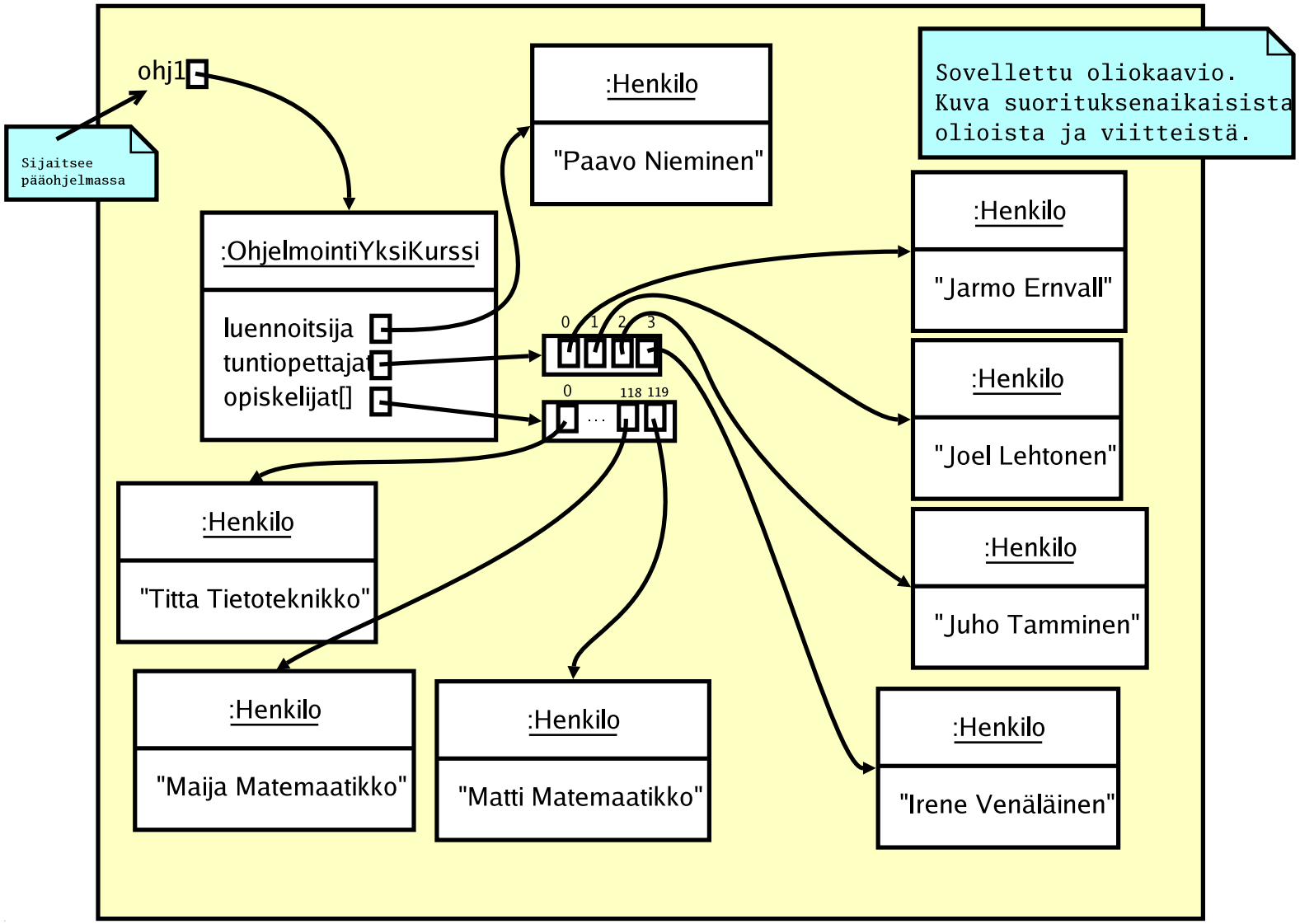


Leikkimielinen 'arkkitehtuurikuva'.
Oikeasti tällaisessa on mieltä vasta
noin 100 kertaa isommassa ohjelmassa kuin
luentoesimerkki ... Eli nämä tulevat eteen
sitten myöhemmin; oikeissa järjestelmissä

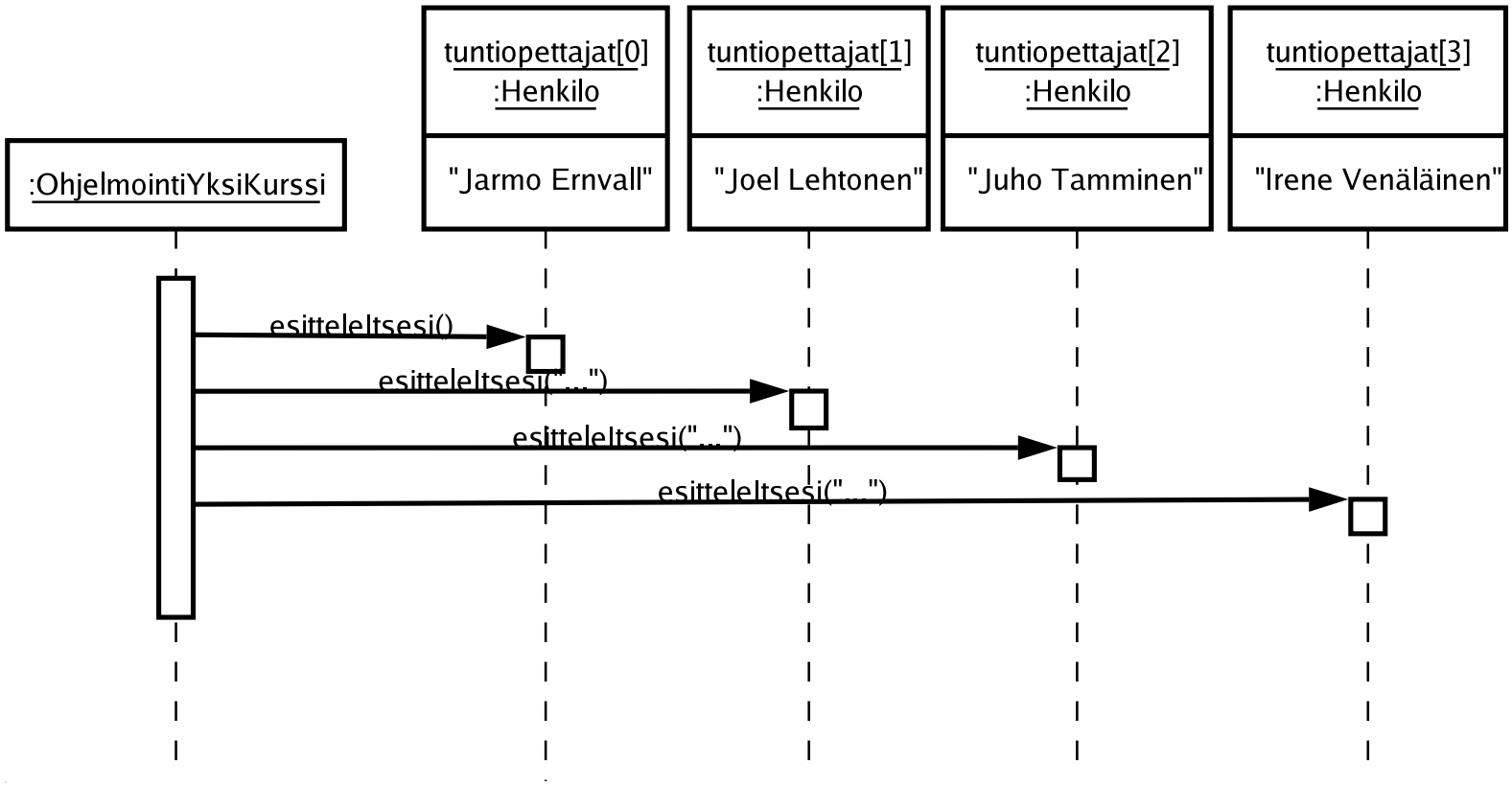
Esim: Luokkakaavio



Esim: Oliokaavio

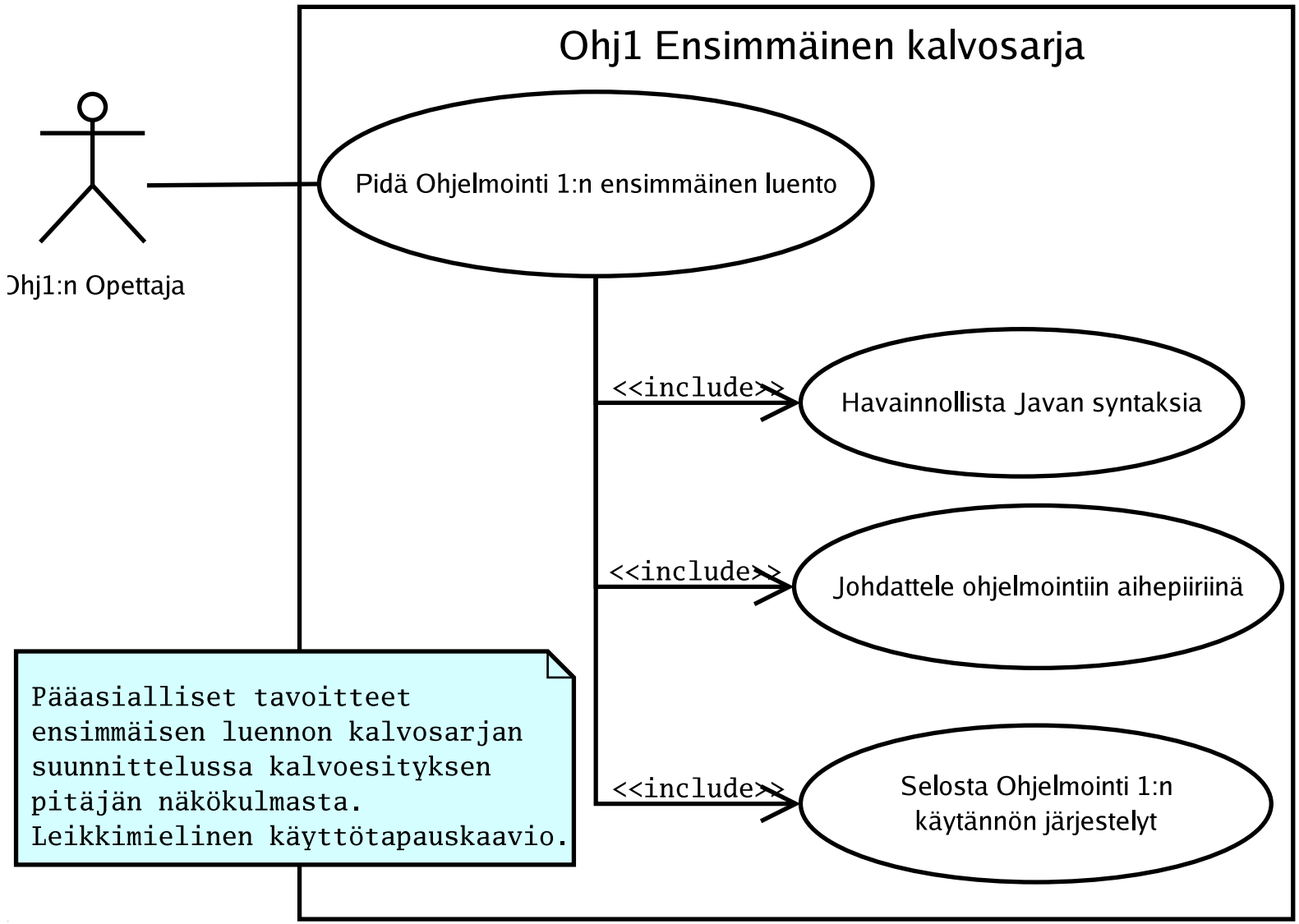


Esim: Sekvenssikaavio



Tämä on varmaan viimeinen sekvenssikaavio Ohjelmointi 1 -kurssilla, koska emme pääse olioiden välisten toimenpiteiden suunnitteluun vielä juurikaan käsiksi . . .

Esim: Käyttötapauskaavio



Esim: Pseudokoodi – mitä se on

- Tarkoituksella ei puhtaasti ohjelmointikieltä
- Kuitenkin rakenteeltaan vastaavaa + riittävän täsmällistä
- Voi kirjoittaa lähes siten kuin jotakin tuntemaansa ohjelmointikieltä
- Tavoite 1: kuka tahansa ohjelmointitaitoinen ymmärtää, vaikkei osaisikaan nimenomaan samoja ohjelmointikieliä kuin kirjoittaja
- Tavoite 2: ohjelmointitaitoinen osaa toteuttaa itse tuntemallaan kielellä

Esim: Pseudokoodi – mihin soveltuu

- Pseudokoodi on hyvä kommunikointikeino ohjelmoinnillisten ideoiden siirtoon. Se on myös suhteellisen täsmällinen tapa kirjoittaa algoritmi.
- Suositeltava vaihe algoritmin suunnittelussa ennen ohjelmointia
- Esimerkki: Kirjoita pseudokoodina, miten koordinoidaan, että jokainen opettaja vastaa kaikkien opiskelijoiden kaikkiin kysymyksiin.
- Seuraavalla kalvolla pseudokoodissa on sukkelasti sekaisin ainakin ohjelmointikieliä Python, Java, C, Pascal sekä epämääräistä suomen kieltä; se on siis tosi ’pseudoa’ koodia.) . . .

Esim: Pseudokoodi

```
Funktio vastausKaikkiinKysymyksiin:
    /* Oletetaan, että on henkilöitä sisältävät rakenteet:
       opiskelijat[] ja opettajat[] */

    for each opiskelija in opiskelijat[] do:
        kysymykset[] := puristaKysymyksetIrti(opiskelija)
        for each kysymys in kysymykset[] do:
            for each opettaja in opettajat[] do:
                kommunikoiVastaus(kysymys, opettaja, opiskelija)

/* On jätetty vielä tarkentamatta, miten kysymys puristetaan
irti opiskelijasta. Vaikeampia osatehtäviä kannattaakin miettiä
erillisinä, aina pienemmiksi osatehtäviksi pilkkoen. Samoin on
jätetty tarkentamatta, miten vastaus kommunikoidaan opiskelijalle.
Osatehtävät on kuitenkin pseudokoodissa jo oikeilla paikoillaan!*/
```

Esim: Algoritmi suomen kielellä

Pseudokoodi on jo lähellä tietokonetoteutusta. Ennen pseudokoodia, on varmasti aina hyvä miettiä tehtävää parhaiten tuntemallaan kielellä, yleensä meilläpäin suomeksi. Esim:

Tiskaaminen tarkemmin ajatellen on sitä, että 1) jos on vielä likaisia astioita, niin otan niistä seuraavan käteeni 2) harjaan sitä kunnes se ei ole likainen 3) laitan astian kuivumaan 4) siirryn seuraavaan likaiseen astiaan eli palaan kohtaan 1.

Esim: Raapustukset

Kynä ja paperi näppiksen viereen! Jos voit piirtää, kirjoittaa tai tuhrata ruutupaperiin ihan mitä tahansa, joka auttaa sinua ajattelemaan ohjelmointiongelmaasi, TEE SE! Älä tuijota kuvaruutua vaan ajattele aivoilla, käsillä, jaloilla, millä se helpoiten vaan käy! Tartu näppikseen vasta kun on jotain kirjoitettavaa ohjelmointikielellä. Silloin sinulla on jo tärkein eli ajatus.

Missä määrin Ohj1:llä mallinnetaan

- Algoritmit, algoritmien tarkentaminen asteittain ja niiden ajatteleminen pseudokoodina on kurssin tärkein (oikeastaan AINOA) asia!
- Java on työkalumme; sen osaaminen tulee pakosti ”kaupan päälle”. Java perustuu täysin olioiden hyödyntämiseen, joten olioita on ymmärrettävä.
- *Ymmärrämme* oliot *alustavasti*, mutta *emme suunnittele* niitä vielä! $\rightarrow$ Ohj2, GKO, projektityöt
- Emme ”suunnittele järjestelmiä” $\rightarrow$ kurssit mallintamisesta ja järjestelmäsuunnittelusta (Oliokeskeinen tietojärjestelmien kehittäminen jne.)
- Kuitenkin ymmärrämme jokaisen ohjelman olevan tietynlainen (ääriyksinkertainen. . .) järjestelmä

Säännöt on tehty rikottaviksi?

- Edellä sanottiin, että ensin pitää mallintaa ja sitten tehdä ohjelma . . . Siihen onkin syytä palata *aina, kun jonkun ohjelman tekeminen tuntuu vaikealta!*
- Mutta yksi ohjelmoinnin ilo on, että voit vaan kirjoitella ohjelmointikieltä, ja kokeilla mitä kivaa mikäkin tekee. Voit muokata ja vetjailla ohjelmaa mielin määrin, kunnes se ehkä toteuttaa ihan eri mallin kuin aluksi . . .
- Kunhan *JATKUVASTI YMMÄRRÄT*, *mitä* ohjelmasi tekee, *miksi* se tekee juuri niin, ja *onko se riittävän hyvä tapa* kulloiseenkin tarkoitukseesi!!

Ei tätä oikein voi selittää sanoin yhdellä kerralla. Pitää vaan lähteä tekemään, tekemään ja tekemään tätä kurssia meidän kaikkien!

Yhteenveto

- Tämä oli ”shokkialoitus” ohjelmointiin
- Hämmennys ja ihmettely on toivottu olotila ;)
- Ensi viikolla otetaan rauhallisempi lähtöruutu
- Herännee kysymys: mikä tässä kaikessa oli olennaista?

Olennaista oli

- Ohjelmointikieli on ainoa mitä tietokone ymmärtää
- Ohjelmointikieli on vain kieli kielien joukossa; joku hullu voi kuvata sillä vaikka luennon etenemisen aiheesta toiseen!
- Ohjelmointikieli on vain yksi monista perspektiiveistä nähdä tietokoneohjelma
- Jostain päästä aina aloitetaan: Ohjelmointi 1 keskittyy algoritmien tekemiseen, joka on kaiken ohjelmoinnin peruslähtökohta
- Suuri osa äsken nähdyistä asioista jätetään hautumaan tulevia kursseja varten.

Ensi viikolla nähdään!

Tehkää ahkerasti demoja ja muistakaa myös ajatella.