

**OHJELMOINNIN PERUSTEIDEN OPPIMISEN ONGELMIA  
VIRTUAALISESSA OPPIMISYMPÄRISTÖSSÄ**

**Sirpa Torvinen**

24.9.2001

Joensuun yliopisto

Tietojenkäsittelytiede

Pro gradu -tutkielma

## **Tiivistelmä**

Ohjelmoinnin perusteiden oppimisen problematiikka näkyy erityisesti etäopetuksessa. Joensuun yliopiston lukuvuonna 2000-2001 Pohjois-Karjalan maakunnan lukiolaisille tarjoamassa tietojenkäsittelytieteen virtuaaliapprobaturissa erityisesti ohjelmointikurssien aikana opiskelijoille tuli vaikeuksia opinnoissaan. Keskeyttäneiden opiskelijoiden parissa laadittu tutkimus osoittaa, että hankalimpina asioina koettiin taulukoiden ja metodien käyttäminen sekä sovelmien ja animaatioiden tekeminen. Tutkimuksessa ilmeni, että ohjelmoinnin perusrakenteiden opiskeluun kaivataan enemmän tukea ja ohjausta. Tekemällä oppimisen hyöty osoittautui tenttivastausten nojalla ilmeiseksi. Tutkimus toteutettiin puolistrukturoituna kyselytutkimuksena kevätlukukauden 2001 aikana. Lisäksi palautettujen harjoitustehtävien ja tenttivastausten perusteella nähdään, että ohjelmoinnin perusrakenteiden oppiminen on keskeyttäneiden opiskelijoiden kohdalla jäänyt puutteelliseksi.

## *Esipuhe*

Tutkielmani syntymiseen ja aihevalintaan vaikutti kiinnostukseni virtuaaliapprobatur-opiskelijoiden menestymistä kohtaan. Olin yhtenä ryhmän jäsenenä tekemässä lukuvuoden 2000-2001 kurssimateriaaleja, mutta en valitettavasti pystynyt olemaan itse lukuvuoden aikana toiminnassa mukana seuraamassa kuinka opiskelijat selviävät virtuaalisessa oppimisympäristössä ja kuinka lukiolainen kykenee selviämään yliopistotason opinnoista omien lukio-opintojensa ohessa.

Haluan esittää tässä kiitokseni professori Erkki Sutiselle hänen tarkkanäköisyydestään ja taidostaan nähdä kiinnostukseni virtuaaliapprobaturia sekä sen kehittämistä kohtaan. Hänen ehdotuksestaan lähti ajatus tämän tutkimuksen tekemisestä keskeyttäneiden opiskelijoiden parissa. Olen äärettömän kiitollinen professori Sutiselle hänen toisaalta lempeästä kärsivällisyydestään pro gradu –tutkielmani ja siihen liittyvien aikataulun venymisten suhteen ja toisaalta oikea-aikaisesta eteenpäin työntämisestä ja kiirehtimisestä työni loppuvaiheessa.

Haluan esittää myös kiitokset assistentti Jarkko Suhoselle, jolta olen saanut paljon arvokasta tietoa virtuaaliapprobaturin käytännön toteutuksesta sekä siitä kuinka lukuvuosi 2000-2001 järjestelyiltään onnistui. Joensuun yliopiston tietojenkäsittelytieteen laitoksen kanslisti Marja-Liisa Makkonen auttoi ystävällisesti kyselykaavakkeiden kopioinnissa ja postituksessa sekä koulujen tutoropettajat huolehtivat kyselykaavakkeiden jakamisesta opiskelijoille ja muistuttelivat opiskelijoita lomakkeiden täyttämisestä. Te jokainen olette osaltanne auttaneet minua työssäni, lämmin kiitos siitä teille kaikille.

# SISÄLLYSLUETTELO

<b>1. JOHDANTO.....</b>	<b>1</b>
<b>2. OHJELMOINNIN OPPIMISEN VAIKEUKSIA.....</b>	<b>4</b>
2.1 JAVAN OPISKELUSSA KOHDATTUJA ONGELMIA .....	5
2.2 OHJELMOINNIN OPPIMISTA HELPOTTAVIA MENETELMIÄ.....	8
<b>3. VIRTUAALINEN OPPIMISYMPÄRISTÖ.....</b>	<b>12</b>
3.1 KÄSITTEITÄ .....	12
3.2 KOHTI VIRTUAALISTA OPPIMISYMPÄRISTÖÄ.....	13
3.3 VIRTUAALISEN OPPIMISYMPÄRISTÖN SUUNNITTELU .....	14
3.4 ASYNKRONISET JA SYNKRONISET VUOROVAIKUTUSVÄLINEET .....	17
3.5 OPISKELIJANA VIRTUAALISESSA OPPIMISYMPÄRISTÖSSÄ .....	20
<b>4. JOENSUUN YLIOPISTO: TIETOJENKÄSITTELYTIETEEN VIRTUAALIAPPROBATUR .....</b>	<b>27</b>
4.1 VIRTUAALIAPPROBATURIN SISÄLTÖ.....	28
4.2 SOSIAALISTEN KONTAKTIEN MAHDOLLISUUS.....	29
4.3 PAINETUN OPPIMATERIAALIN LINKITTÄMINEN VERKKOMATERIAALIIN .....	31
4.4 OPISKELU INTERNETIN VÄLITYKSELLÄ.....	32
4.5 OPPIMISYMPÄRISTÖN HELPPOKÄYTTÖISYYS .....	33
<b>5. TUTKIMUSASETELMA, TUTKIMUSONGELMAT JA TUTKIMUSMENETELMÄT...35</b>	
5.1 TUTKIMUSASETELMA .....	35
5.2 TUTKIMUSONGELMAT .....	35
5.3 TUTKIMUSMENETELMÄ.....	36
<b>6. TULOSTEN ESITTELY .....</b>	<b>38</b>
6.1 TAUSTATIETOJA .....	38
6.2 KESKEYTTÄNEIDEN LUKUMÄÄRÄ .....	39
6.3 KESKEYTTÄMISEN SYYT .....	43
6.4 VAIKEINA KOETUT OHJELMOINNIN OSA-ALUEET .....	45
6.5 HARJOITUSTEHTÄVIEN TEKEMINEN.....	47
6.6 TENTTIIN OSALLISTUMINEN .....	51
6.7 TUKITOIMIEN TARPEELLISUUS.....	54
6.8 JATKOKYSELYT .....	57
<b>7. JOHTOPÄÄTÖKSET.....</b>	<b>59</b>
<b>8. YHTEENVETO.....</b>	<b>69</b>
<b>LÄHTEET .....</b>	<b>72</b>
<b>Liite 1:</b> Itseopiskeluopas	
<b>Liite 2:</b> Kyselykaavake	
<b>Liite 3:</b> Ohjelmointi 1 –kurssin tentti 11.12.2000	

# 1. JOHDANTO

Syyslukukauden 2000 alusta alkaen Joensuun yliopiston tietojenkäsittelytieteen laitoksen tarjoama 15 opintoviikon laajuinen virtuaaliapprobatur-kokonaisuus otettiin Pohjois-Karjalan maakunnan lukioissa varsin optimistisin ajatuksin ja suurin toivein vastaan. Syksyn mittaan oppilaita alkoi jäädä kuitenkin hiljalleen pois ja siitä syntyikin halu lähteä tutkimaan miksi näin on käynyt. Mitkä syyt ovat johtaneet siihen, että oppilaat päättivät keskeyttää opintonsa? Aloittelevalla tietojenkäsittelytieteen opiskelijalla on usein vaikeuksia ymmärtää ohjelmoinnin ideaa ja aivan perustehtävienkin konstruoiminen mielessä on hankalaa. Virtuaaliapprobaturissa opiskellaan ohjelmointia osana approbatur-kokonaisuutta. Kuinka lukio-opiskelijoilta onnistuu jo normaali lähiopetuksessakin vaikeana koettu ohjelmoinnin opiskelu verkon kautta itsenäisesti?

Näistä kysymyksistä nousi esille tutkimukseni aihe ja tutkimuksellani lähdin etsimään vastausta seuraavaan kahteen kysymykseen. Ensinnäkin mitä asioita ohjelmoinnin perusteiden oppimisessa koetaan hankalana ja voitaisiinko virtuaalisessa ympäristössä oppimista jollakin tavoin tukea, jotta opiskelijat selviytyisivät ohjelmoinnin perusteiden oppimisesta paremmin? Toiseksi mitkä syyt johtivat keskeyttämiseen lukuvuonna 2000-2001?

Tietojenkäsittelytieteen opiskelua aloittelevien opiskelijoiden kohtaamat ohjelmoinnin oppimisen vaikeudet ovat yleisesti tunnettuja. Ben-Bassat Levy ym (2001) ovat tutkineet ohjelmoinnin oppimisen ongelmia ja kehittäneet oppimisen helpottamiseksi Jeliot 2000:n, jolla voidaan visualisoida algoritmeja. Animoinnin avulla oppilaiden on helpompaa ymmärtää algoritmien ja ohjelmien toimintaa. Turner ja Zachary (2001) sekä Ellis (1999) ovat tutkineet oppilaan kognitiivista työskentelytapaa ohjelmoinnin opiskelussa ja tarkastelevat tietojenkäsittelytieteen perusteita opiskelevien abstraktin ajattelutaidon omaksumisessa kohtamia ongelmia. Roberts (2001) on kehittänyt ohjelmoinnin perusteita opiskelevien käyttöön MiniJavan, jonka käytöllä pyritään helpottamaan Javan käytössä hankalina koettuja asioita. Tässä tutkimuksessa tarkastellaan Javan opiskelun vaikeutta lähinnä Robertsin mukaan. Roberts (2001) käy läpi Javan heikkouksia aloittelevan ohjelmoijan kannalta ja pohtii kuinka tietyt Javan ominaisuudet kuten luokkakirjaston määrä tai poikkeusten hallinta saattavat hankaloittaa itse ohjelmoinnin perusteiden oppimista. Erityisesti keskivertaiset tai sitä heikommat opiskelijat joutuvat uhraamaan paljon resursseja jo pelkästään käytetyn ohjelmointikielen rakenteiden ymmärtämiseen, mikä osaltaan voi johtaa turhautumiseen ja dh-

jelmoinnin opiskelusta luopumiseen jo ennen kuin opiskelija edes ehtii saada ohjelmoinnista selkeää kuvaa.

Perinteisestä luokkaopetuksesta virtuaaliseen oppimisympäristöön siirtymistä ja etäopetusmenetelmin toteutettujen kurssien suunnittelua ovat tutkineet muun muassa Cordani ja Tucker (1998) sekä Chute ym (1997). Oppimisympäristössä tarvittavien välineiden ja pedagogisten tukitoimien merkittävyyttä korostaa Adorni ym (1998) kehittämässään ja testaamassaan virtuaalisen oppimisympäristön mallissa. Tiffin ja Rajasingham (1995) ja Wolz ym (1997) puolestaan esittelevät verkkopohjaisissa oppimisympäristöissä käytettävissä olevia synkronisia ja asynkronisia viestintävälineitä.

Virtuaalisiin oppimisympäristöihin on siirrytty varsin innostuneesti ja lähes jokaisella oppilaitoksella kouluasteesta riippumatta alkaa jo olla vähintäänkin kehitteillä olevaa verkkopohjaista kurssitarjontaa. Dhanarajan (2001), Klemm (2001) ja Farrell (1999) esittävät kritiikkiä liian innokkaasta virtuaalisten kurssien toteutuksesta sekä korostavat virtuaalisen kurssin suunnittelun merkitystä. Muutosta ei tulisi tehdä vain muutoksen vuoksi vaan virtuaaliseen oppimisympäristöön siirtymisellä tulisi olla selkeä tavoite ja syy, miksi kurssit toteutetaan virtuaalisesti.

Opiskelijan rooli muuttuu Mäki-Komsin (1999) mukaan virtuaalisessa oppimisympäristössä aktiivisemmaksi ja vastuullisemmaksi kuin mitä se perinteisessä luokkahuoneessa on. Traditionaalisen luokkahuoneopetuksen sekä teknologiaympäristöä hyödyntävässä oppimisympäristössä tapahtuvan opetuksen välillä olevia eroja on tarkastellut Pollard ja Pollard (1993). Cornell ja Martin (1997) tarkastelevat opiskelijan motivaatioon ja keskeyttämispäätöksen syntymiseen vaikuttavia tekijöitä. Lisäksi Meisalo ym (2000) tarkastelevat opiskelijan sisäistä ja ulkoista motivaatiota sekä keinoja, joilla niihin voitaisiin vaikuttaa. Motivaatio vaikuttaa sekä opintojen menestyksekkääseen suorittamiseen että keskeyttämispäätöksen syntymiseen.

Joensuun yliopiston tietojenkäsittelytieteen virtuaaliapprobatur tarjoaa opiskelijoille verkkopohjaisen oppimisympäristön, jossa pääosin työskennellään Internetin välityksellä. Haataja ym (2001) esittelevät approbaturin toteutuksesta kehitetyn Candle-mallin ja sen neljä perusrakennuspilaria: *1) sosiaalisten kontaktien mahdollisuus* verkon läpi toisiin opiskelijoihin sekä ohjaajaan yliopistolla ja tutoropettajaan

koululla, 2) *painetun oppimateriaalin linkittäminen aktivoivaan WWW-materiaaliin*, 3) *lähes täysin Internetin välityksellä tapahtuva opiskelu* sekä 4) *oppimisympäristön pitäminen käyttäjäystävällisenä, selkeänä ja yksinkertaisena*. Mallin ideana on viitoittaa opiskelijoille oppimisen polku verkopohjaisessa ympäristössä ”sähköisten kynttilöiden” (Candle) avulla askel askeleelta. Opiskelu tapahtuu pääosin Internetin välityksellä ja opintoja ohjataan verkossa olevan oppimisympäristön kautta.

Virtuaaliapprobatur toteutettiin ensimmäisen kerran lukuvuonna 2000-2001, jolloin 89 pohjoiskarjalalaista nuorta lukiolaista aloitti opintonsa ympäri maakuntaa omista lukioistaan käsin. Syksyn 2000 aikana 37 % aloittaneista opiskelijoista (33 opiskelijaa) keskeytti virtuaaliopintonsa ja näistä 63 % lopetti Ohjelmointi 1 -kurssin aikana (21 opiskelijaa). Opintomenestykseen liittyvät vaikeudet näyttivät pääosin alkavan ohjelmoinnin perusteiden opiskelun aikana. Tutkimus toteutettiin puolistrukturoituna kyselytutkimuksena ja kohdistettiin keskeyttäneisiin opiskelijoihin. Jatkokysely toteutettiin sähköpostitse jatkotutkimuksiin lupautuneille opiskelijoille. Tutkimuksessa tarkasteltiin lisäksi keskeyttäneiden opiskelijoiden tenttivastauksia ohjelmoinnin perusteiden oppimisen arvioimiseksi sekä kaikkien Ohjelmointi 1 -kurssiin osallistuneiden opiskelijoiden harjoitustehtävien tekemistä ja niiden vaikutusta opinto- ja tenttimenestykseen. Tutkimuksella etsittiin ohjelmoinnin perusteiden oppimisessa vaikeina koettuja asioita sekä keinoja, joilla opiskelijoita voitaisiin auttaa virtuaalisessa oppimisympäristössä. Mielenkiintoista oli myös selvittää syyt, miksi opiskelijat jättivät lukuvuonna 2000-2001 opintonsa kesken.

## 2. OHJELMOINNIN OPPIMISEN VAIKEUKSIA

Tässä luvussa tarkastellaan aluksi asioita, joita tietojenkäsittelytieteen aloittelevat opiskelijat tuntevat opintojensa alkuvaiheessa hankalina. Sen jälkeen tarkastellaan ohjelmointikielenä suosituksi nousseen Javan opiskelussa kohdattuja vaikeuksia ja lopuksi tutustutaan keinoihin, joilla ohjelmoinnin perusteiden oppimista on pyritty tukemaan.

Aloittelevat ohjelmoinnin opiskelijat, *noviisit*, kohtaavat usein suuria vaikeuksia opintojensa alussa ohjelmoinnin perusteita opiskellessaan. Yhtenä syynä pidetään sitä, että opiskelijat eivät vielä opintojensa aloitusvaiheessa osaa luoda riittävän selkeää mielikuvaa siitä, kuinka tietokone toimii (Ben-Bassat Levy et al, 2001). Opiskelijoilla voi olla suuria vaikeuksia esimerkiksi jonkin pienen ohjelman tai yksinkertaisen algoritmin suorituksen hahmottamisessa. Erityisesti opiskelijat, joilta puuttuu kyky ajatella abstraktisti, kokevat ohjelmoimisen ainoastaan jäsentymättömänä joukkona erilaisia lauseita ja poikkeuksia ymmärtämättä toiminnallista kokonaisuutta, jonka ne muodostavat. Näiden opiskelijoiden osalta ohjelman kompleksisuus kasvaa Turnerin ja Zacharyn (2001) mukaan eksponentiaalisesti ohjelman kokoon nähden ja siksi opiskelijoiden on vaikea soveltaa lukemaansa tietoa käytännön ohjelmoimiseen ja oppia itse ohjelmoimaan. Toisaalta opiskelijat, jotka ovat omaksuneet abstraktin tavan ajatella näyttävät selviytyvän paljon paremmin ohjelmoinnin opiskelussa. Tällaiset opiskelijat ymmärtävät ohjelmat funktioiden ja luokkien jäsentyneenä joukkona, jossa funktioilla on tietty tehtävä ja luokat edustavat tiettyjä asioita ja kykenevät selviytymään paljon laajemmistakin ohjelmista (Turner & Zachary, 2001).

Ohjelmoinnin perusteita opiskelevien on opittava muodostamaan ensiksi ongelman ratkaiseva algoritmi ja käännettävä se kurssilla käytettävällä ohjelmointikielillä syntaktisesti oikein toimivaksi suorituskelpoiseksi ohjelmaksi. Kukin opiskelija oppii omalla persoonallisella tavallaan ja oppimisympäristön tulisi tarjota välineitä, joilla voidaan tukea jokaisen opiskelijan yksilöllistä oppimistyyliä (Ellis et al, 1999). Perinteisesti opiskelijoilla on ollut mahdollisuus debuggereiden avulla käydä ohjelmat läpi askel askeleelta ja toiminto toiminnolta. Nämä apuvälineet toimivat ohjelmointikielen rakenteiden tasolla ja opiskelija voi nähdä askeleiden etenemisestä, kuinka esimerkiksi silmukka suoritetaan. Sen sijaan kääntäjät eivät pysty mallintamaan kognitiivista prosessia, jota opiskelija käy läpi tutkiessaan ohjelmakoodin suoritusta. Opiskelijan



tulisi pystyä hahmottamaan mielessään, kuinka esimerkiksi lajittelualgoritmit tai sijoitukset toteutetaan käytännössä; mitä tapahtuu muuttujille ohjelman suorituksen edetessä tai miten lajiteltavat tiedot järjestetään? Monelle opiskelijalle tällainen miellekartan muodostaminen ohjelmakoodista on vaikeaa ja tätä on pyritty helpottamaan muun muassa algoritmien visualisoinnin ja animaatioiden avulla (Ben-Bassat Levy et al, 2001).

## ***2.1 Javan opiskelussa kohdattuja ongelmia***

Ohjelmoinnin perusteiden opettamiseen on perinteisesti käytetty kouluissa ohjelmointikielenä Pascalia. Graafisten käyttöliittymien ja WWW:n käytön yleistyessä on yhä enemmän tunnettu painetta siirtyä jo ohjelmoinnin perusteiden opiskelun yhteydessä käyttämään oliopohjaista ohjelmointikieltä. Vuonna 1995 Sun Microsystemsin julkaisema Java on valtaamassa voimakkaasti alaa ohjelmoinnin perusteiden opettamiskielenä. Java on ohjelmointikielenä melko yksinkertainen kieli, mutta sen oppiminen on kuitenkin koettu noviisien keskuudessa hankalaksi. Lisäksi Javan käyttämisessä ohjelmoinnin perusteiden opetuskielenä piilee riski, että lähdetäänkin opettamaan itse Java-ohjelmointia sen sijaan, että opetettaisiin ohjelmointia Javaa käyttäen (Hong, 1998). Ohjelmoinnin perusteet tulisi pystyä opettamaan käytetäänpä sitten opetuskielenä mitä ohjelmointikieltä tahansa. Opetuksessa käytetyn ohjelmointikielen ominaispiirteet tulisi pystyä häivyttämään taustalle, jotta opiskelija pystyisi näkemään ohjelmoinnin perusrakenteet ja tunnistamaan niiden käytön.

### *Luokkakirjastojen laajuus*

Javan käyttämistä opetuskielenä hankaloittaa siihen liittyvien luokkakirjastojen valtava määrä. Uusi ohjelmoija hämmentyy ja tuntee epävarmuutta lukuisten pakettien, luokkien ja metodien vuoksi. Noviisin on vaikea oppia hahmottamaan ja käyttämään näin laajaa järjestelmää, vaan ohjelmointikielestä jää hyvin paljon mystisyyden peittoon (Roberts, 2001). Myös Hong (1998) toteaa, että opetuksessa kohdattiin jonkin verran ongelmia, kun Javaa ryhdyttiin käyttämään ohjelmoinnin perusteiden opetuskielenä. Osa näistä ongelmista johtui kielen syntaksista ja Java-välineistä. Hongin (1998) mukaan suurin osa kohda-

tuista vaikeuksista johtui kuitenkin opiskelijoiden puutteellisesta kyvystä suunnitella ja toteuttaa ohjelmia, ei niinkään ohjelmointikielen omaksumisesta.

### *Java-oppikirjojen ja Java-versioiden väliset eroavuudet*

Javaa on kehitetty jatkuvasti eteenpäin ja sen työskentelyalusta on muuttunut hyvinkin merkittävästi eri versioiden aikana. Tämä on osaltaan vaikeuttanut Java-oppikirjojen tekemistä: osa oppikirjoista on jopa jouduttu vetämään tuotannosta pois, kun versioiden välillä olevat eroavuudet ovat olleet niin merkittäviä. Oppikirjojen tekijät ovat tästä syystä tinkineet Java-ohjeistuksen laadullisuudesta, mikä lopulta kaikkein eniten hankaloittaa vain Javalla ohjelmoimaan opiskelevien oppimista (Roberts, 2001). Java-oppikirjat on usein laadittu ohjelmoinnin opettamisen kannalta virheellisestä näkökulmasta: ohjelmointiin johdattelevia oppikirjoja Javasta ei löydy, vaan lähes kaikki Java-oppikirjat opettavat kuinka Javalla ohjelmoidaan eivätkä suinkaan opeta sitä, kuinka ohjelmia tehdään käyttämällä Javaa. Esimerkiksi Javan AWT-kirjaston (*Abstract Windowing Toolkit*) käyttäminen on nimenomaan Javan erityisominaisuus, joka joudutaan opiskelemaan, jotta ohjelmointikurssilla ylipäänsä pystyttäisiin työskentelemään. Ohjelmoinnin perusteiden opetuksessa päätarkoituksena olisi kuitenkin oppia ohjelmoimaan käyttämällä Javaa ohjelmointikielenä, eikä suinkaan opiskella itse Javaa ja sen erityispiirteitä (Hong, 1998).

### *Käsitteiden runsaus*

Aloittelevalla ohjelmoinnin opiskelijalla on jo pelkästään ohjelmoinnin perusteita opiskeltaessa koko joukko käsitteitä opittavana kuten esimerkiksi muuttujat, vakiot, peräkkäisyys, valintalauseet, toistolauseet, taulukot ja niin edelleen. Kun ohjelmointikielenä käytetään Javaa, tämä tuo lisäksi koko joukon uusia käsitteitä opeteltavaksi. Esimerkiksi heti ohjelmoinnin alussa törmätään käsitteisiin kuten metodi, muuttujat, luokka, palautustyyppi, metodin nimi, parametrit ja taulukko jokaisessa sovelluksessa tarvittavan main-metodin yhteydessä (Hong, 1998). Alkuvaiheessa opiskelija oppiikin vain kirjoittamaan jokaiseen sovellukseen rivin `public static void main(String[] args)` ymmärtämättä syvemmin, mitä kukin näistä käsitteistä itse asiassa tarkoittaa tai minkä vuoksi nämä on kirjoitettava ohjelmakoodiin. Ohjelmoinnin perusteiden opettamisen alkuvaiheessa ohjelmointikieleen liittyvät erityispiirteet on pidettävä

mahdollisimman yksinkertaisina opiskelijoille, jotta itse ohjelmoinnin perusrakenteet pystyttäisiin oppimaan. Tämän vuoksi on alkuvaiheessa syytä jättää selittämättä hyvinkin perusteellista laatua olevia olio-ohjelmoinnin käsitteitä kuten esimerkiksi olioiden periytyminen tai poikkeukset ja niiden hallinta (Hong, 1998). Toisaalta selittämättömät käsitteet, joita tulee esille, saattavat myös aiheuttaa opiskelijassa hämmennystä ja epävarmuuden tunnetta. Suuri osa esiin nousevista käsitteistä jää alkuvaiheessa hämärän sumuverhon taakse.

### *Standardisyytteiden, lähinnä syötetiedon lukemisen, vaikeus*

Standardisyytteiden lukemisen vaikeus on Javassa erityisen hankalaa. Noviisin on suorastaan mahdotonta ohjelmoida itse syöttölauseiden lukemiseen tarvittavaa metodia Javalla, mistä syystä monet Java-kirjojen kirjoittajat ovat päätyneet tekemään omia metodeita, joita tarjotaan oppikirjojen mukana. Roberts (2001) mainitsee esimerkkinä muun muassa Cornell ja Horstmanin (1996) syötevirran lukemista varten tekemän metodin, joka sisältää 30 riviä ohjelmakoodia.

### *Poikkeusten hallinta*

Javan oppimisen hankaluutena koetaan yleisestikin poikkeusten hallinnan ymmärtäminen. Muutamaa yleistä ajonaikaista poikkeusta lukuunottamatta Java vaatii, että jokainen metodi huolehtii poikkeuksista omassa koodissaan tai varustautuu poikkeusten mahdollisuuteen throws-lauseella. Tämä johtaa siihen, että aivan yksinkertaisissakin Java-ohjelmissa on käytettävä poikkeuksia ja jotkut Java-ohjelmoijat opastavatkin opiskelijoita kirjoittamaan kaikkiin metodeihin Throws Exception-lauseen. Tämä omalta osaltaan on omiaan johdattelemaan opiskelijoita huonoon ohjelmointityyliin (Roberts, 2001).

### *Vapaasti saatavilla olevat Java-koodit*

Hong (1998) toteaa, että opiskelijoiden saatavilla olevien Java-koodien runsaus on toisaalta hyöty mutta toisaalta myös haitta. Vapaasti Internetistä saatavilla olevat Java-koodit auttavat oppimaan Javaa, mutta toisaalta myös laittoman kopioinnin ja plagioinnin vaara on aina olemassa. Opiskelijoilta saattaa hämärtyä käsitys siitä, mitä voi ottaa mukaan omaan koodiin ja mikä on tekijänoikeuden nojalla suojattua koodia.

Lisäksi valmiina olevia Java-koodeja ei välttämättä ole ohjelmoitu käyttäen hyvää ohjelmointitapaa, joten ne voivat ohjata opiskelijaa väärään tai epätarkkaan ohjelmointityyliin. Ongelmallisena on koettu myös Java-uutisryhmien käyttäminen: opiskelijat saattavat jopa lähettää ohjelmointitehtävänsä sinne ja kysyä, kuinka tällainen ongelma ratkaistaisiin. Opettajan tulisikin ennen kurssin alkua kertoa kuinka Internetin välityksellä vapaasti saatavia lähteitä saa käyttää (Hong, 1998).

## ***2.2 Ohjelmoinnin oppimista helpottavia menetelmiä***

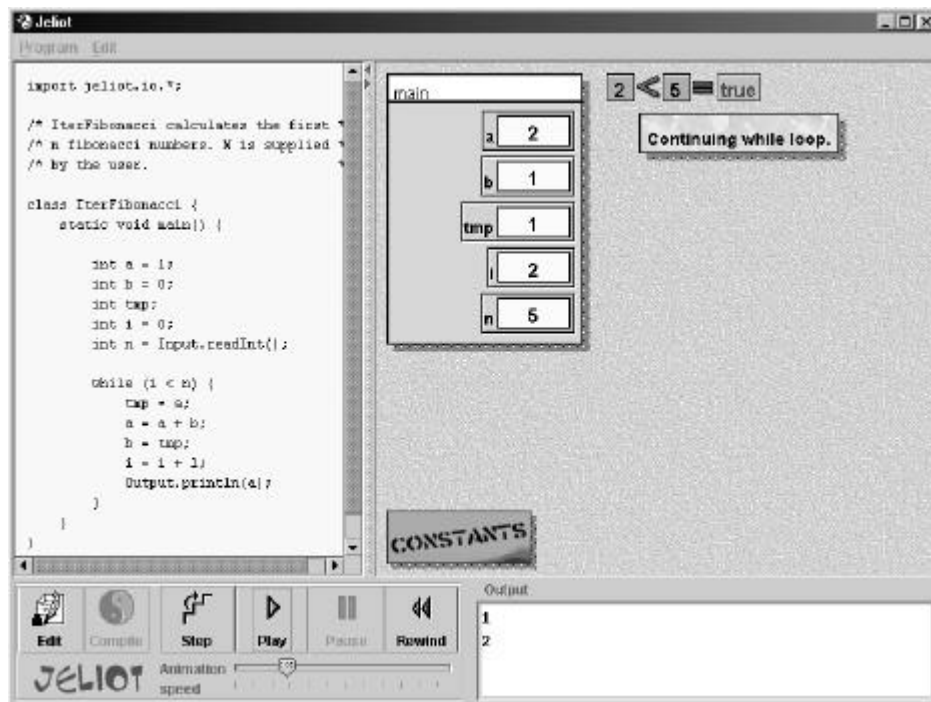
Tietotekniikan opetuksessa opettajat ovat pyrkineet käyttämään erilaisia menetelmiä, joiden avulla abstraktit käsitteet ja toiminnot saataisiin konkreettisemmiksi ja samalla opiskelijoille helpommiksi omaksua. Tietojenkäsittelytieteen alalla varsin merkittävänä tutkimuskohteena pyritään löytämään keinoja, joilla voitaisiin tukea opiskelijaa algoritmien ja ohjelmoinnin oppimisessa (Kashihara et al, 1999).

### *Animaatioiden käyttäminen*

Animaatiota on käytetty jo 1980-luvun alusta alkaen algoritmien toimintaa visualisoivana välineenä helpottamaan sellaisten opiskelijoiden oppimista, joilla on vaikeuksia luoda kuvaa abstrakteista tietorakenteista. Ronald Baeckerin *Sorting Out Sorting* –animaatio käynnisti kauan odotetun vallankumouksen tietojenkäsittelytieteen koulutuksen alalla 1981 (Anderson & Naps, 2001). Algoritmien visualisoinnilla on todettu olevan selvästi myönteistä vaikutusta oppimiseen. Opiskelijoiden on todettu algoritmien visu-

alisointia seurattessaan ymmärtävän paremmin, kuinka esimerkiksi taulukon arvoja vaihdetaan taulukkoa järjestettäessä jonkin lajittelualgoritmin mukaan. Kun opiskelija joutuu miettimään, kuinka algoritmi toimii erilaisilla syötteillä, hän oppii helpommin ymmärtämään miten algoritmi käytännössä toimii ja miksi se toimii juuri niin (Anderson & Naps, 2001).

Ben-Bassat Levy ym (2001) ovat kehittäneet ohjelmointia aloittelevien opiskelijoiden käyttöön Jeliot2000-ohjelman. Päättarkoituksena on tarjota noviiseille väline, jonka avulla he voivat animoida ohjelmia ohjelmointikielen rakenteiden tasolla. Jeliot2000:n avulla voidaan muodostaa animaatioita muuttujien arvoista, vakioiden käyttämisestä, poikkeusten käsittelystä, sijoituksista ja aliohjelmien suorituksesta. Animaatiossa tärkeää on pystyä havainnollistamaan, kuinka tieto kulkee (input-output) ohjelman suorituksen aikana sekä kuinka muuttujien arvot muuttuvat ja ohjelman suoritus etenee sijoitus-, valinta- tai toistolauseita suoritettaessa. Kuvassa 1 nähdään kuinka Jeliot2000:n visualisoi Fibonaccin luvut tulostavaa ohjelmaa.



**Kuva 1 Esimerkki Jeliot2000:n käytöstä ohjelman visualisoimiseen**

Animaation käyttämisestä ohjelmoinnin perusteiden opetuksessa saatava hyöty noviisien keskuudessa ei kuitenkaan ole ollut itsestäänselvyys, vaan noviiseilla on ollut yllättäen vaikeuksia sijoittaa animaation graafisia elementtejä algoritmeihin (Stasko et al, 1993). Usein animaatiota käytetään ainoastaan jonkin yksittäisen esimerkin havainnollistamiseen, mutta jotta opiskelijat todella hyötyisivät tästä uudesta työvälineestä, sitä tulisi käyttää pitkäkestoisesti läpi koko kurssin integroiden sen käyttö luokkahuonetilanteeseen ja harjoituksiin yhdeksi perustyövälineeksi (Ben-Bassat Levy et al, 2001). Noviisit tarvitsevat kuitenkin apua, jotta oppivat ymmärtämään, kuinka visualisointi mallintaa itse ohjelman suoritusta. Tämän vuoksi oppimisympäristössä tulisi olla jonkin verran toimivia esimerkkejä, joita suorittamalla noviisit näkevät, kuinka algoritmia voidaan käyttää ja kuinka algoritmille annetuista syötetiedoista syntyy tulosteet (Khuri, 2001). Animaatio on usein toteutettu niin, että toisessa ikkunassa näkyy ohjelmakoodi ja toisessa itse animaatio. Opiskelija pystyy näin seuraamaan animaation edetessä missä kohden ohjelmakoodia ollaan kulloinkin menossa. Lahjakkaat opiskelijat osaavat hahmottaa jo algoritmeja tai ohjelmakoodia lukiessaan, kuinka toimintojen suoritus etenee, mutta keskivertaiset tai heikommat opiskelijat tarvitsevat konkreettisen mallin ymmärtääkseen, kuinka ohjelma toimii.

Ben-Bassat Levyn ym (2001) mukaan on tärkeää huomioida, että ohjelman jokainen piirre tulee visualisoitua animaatioissa niin, että opiskelijan ei tarvitse jäädä miettimään, mistä esimerkiksi jokin vakioarvo tulee esille. Animaation tulee lisäksi olla jatkuvaa siten, että ohjelma etenee toiminnosta toiseen täydellisesti ilman toimintakatkoksia. Opettajan tulee huolehtia siitä, että hän tulkitsee animaation edetessä ohjelman suoritusta opiskelijoille. Ei riitä, että tarjotaan opiskelijoille työvälineet, vaan on myös opetettava heidät käyttämään niitä ja varmistettava opettajan antamalla animaation selityksellä, että opiskelijat todella ymmärtävät mitä itse animaatioissa tapahtuu. Opiskelijat osaavat opettajan selityksen avulla paremmin yhdistää ohjelmakoodin ja animaatioissa tapahtuvat toiminnot toisiinsa.

### *Fill-in-Blank –ohjelmointi*

Kashihara ym (1999) ovat kehittäneet *Fill-in-Blank* –ohjelmointitavan, jolla on pyritty helpottamaan ohjelmoinnin perusteita opiskelevien tapaa ymmärtää algoritmeja. Menetelmässä annetaan opiskelijalle ohjelmakoodia, jossa osa koodista on korvattu tyhjällä ruudulla (blank). Opiskelijalle annetaan ohjeet, mitä ohjelman tulee tehdä. Opiskelija joutuu miettimään, kuinka tieto ohjelmassa kulkee sekä kuinka

hänen tulee koodia täydentää, jotta ohjelmasta tulisi jälleen toimiva. Menetelmän uskotaan auttavan opiskelijan kognitiivista prosessia ja helpottavan algoritmin ymmärtämistä. Menetelmän hyötynä on, että opiskelija joutuu miettimään ohjelman toimintaa laajemminkin ja ymmärtämään mitkä toiminnot vaikuttavat mihinkin.

### *Tutkiva oppiminen*

Yhtenä kehitettynä ohjelmoinnin kitehtävien tekemismuotona on Lischner (2001) kokeillut tutkivaa oppimista (explorations). Menetelmässä opiskelijalle annetaan pieni osa ohjelmakoodia, jonka toimintaa hänen tulee osata ennakoida ennen ohjelman suorittamista. Tämän jälkeen opiskelija suorittaa ohjelman ja seuraa, oliko hän osannut ennakoida ohjelman käyttäytymistä oikein. Ohjelman suorittamisen jälkeen opiskelija saa vielä seurantakysymyksiä, joilla tarkistetaan oliko opiskelijan etukäteen tekemä ohjelman käyttäytymistä koskeva ennuste oikea ja onko opiskelija ymmärtänyt ohjelman toimintatavan. Jos opiskelijan ennustama tulos ei toteudukaan, oppilaan on ymmärrettävä ja osattava selittää, mistä ohjelman suorituksen ja ennusteen välinen ero johtuu.

### 3. VIRTUAALINEN OPPIMISYMPÄRISTÖ

Tässä luvussa tutustutaan aluksi käsitteisiin, joilla virtuaalista opetusta ja oppimista kuvataan sekä tarkastellaan mitkä tekijät ovat olleet vaikuttamassa virtuaaliseen oppimisympäristöön siirtymiseen. Sen jälkeen kohdassa 3.3 katsotaan mitä seikkoja virtuaalisen oppimisympäristön suunnittelussa tulisi ottaa huomioon ja kohdassa 3.4 tutustutaan tarkemmin oppimisympäristössä tarjolla oleviin asynkronisiin ja synkronisiin vuorovaikutusvälineisiin. Lopuksi tarkastellaan kuinka opiskelijan rooli oppijana muuttuu virtuaaliseen oppimisympäristöön siirryttäessä.

#### 3.1 Käsitteitä

Koululaitoksen tehtävänä on kautta aikojen ollut antaa oppilailleen valmiudet selvitä niistä vaatimuksista, joita yhteiskunta jäsenilleen asettaa. Yhteiskunnan kehittyminen tietoyhteiskunnaksi on muuttanut kansalaisten peruskoulutustarpeita ja koululaitoksen tulee vastata tähän haasteeseen. Viime vuosien hyvin kiivastahtinen koulutuksen kehittäminen kohti avointa ja joustavaa oppimisympäristöä on johtanut siihen, että alalla vallitsee varsin runsas käsitteiden viidakko. *Virtuaali*-sanaa on käytetty ympäri maailmaa varsin laajasti ja osin arvostelukyvottomästikin. Sitä on käytetty yleisesti synonyyminä ainakin seuraaville termeille: *avoin ja etäopetus* (open and distance learning), *jaettu opetus* (distributed learning), *verkkotyöskentelyyn pohjautuva opetus* (network learning), *verkkopohjainen opetus* (web-based learning) ja *tietokoneavusteinen opetus* (computer-aided learning) (Farrell, 1999).

Collis (1998) käyttää käsitettä *etäopiskelu* (tele-learning) kuvaamaan opiskelua, jossa on käytetty hyväksi viestintätekniikkaa (telematics). Määrittely ei ota millään tavoin kantaa maantieteelliseen etäisyyteen opiskelijan ja opettajan tai opiskelijan ja tiedon välillä. Collisin (1998) mukaan ei ole relevanttia itse oppimisen kannalta edes tarkastella sitä, kuinka kaukana toisistaan maantieteellisesti ollaan. Tärkeämpää on kiinnittää huomiota siihen, että etäopetuksella pystytään tarjoamaan parempaa opetusta kuin perinteisillä menetelmillä. Etäopiskelun tulisi olla tehokkaampaa, rikastuttavampaa tai joustavampaa opetusta kuin mitä perinteinen luokahuoneopetus on, jotta sitä ylipäänsä olisi mielekästä lähteä tarjoamaan.



Käsitteiden viidakosta löytyy silti yhteisiä piirteitä, joiden nojalla voidaan luokitella mitä on *virtuaalinen opiskelu*: virtuaalisessa opetuksessa ainakin osan aikaa opiskelija ja opettaja ovat fyysisesti eri paikoissa ja erilaisten teknisten medioiden, kuten esimerkiksi kirjoittimen, äänilaitteiden, videoiden tai tietokoneiden, käyttö vuorovaikutusvälineinä *oppilaan ja opettajan, oppilaan ja toisten oppilaiden tai oppilaan ja kurssimateriaalin* välillä on merkittävää. Virtuaalisen opetuksen luonteeseen kuuluu olennaisesti, että opiskelijalla on käytettävissään erilaisia viestintäväyliä, joiden avulla hän voi joko osallistua keskusteluihin tai aloittaa keskustelun itse. Tunnuspiirteisiin kuuluu lisäksi, että opiskelijalla on itsellään oltava vastuu omasta oppimisprosessistaan. (Lawhead et al, 1997)

### **3.2 Kohti virtuaalista oppimisympäristöä**

Yhteiskuntamme kehittyessä tietoyhteiskunnaksi on opiskelun luonne muuttunut kohti avoimempaa ja joustavampaa oppimista. Yliopistojen ja korkeakoulujen on kyettävä tarjoamaan uuden opiskelijasukupolven odotusten mukaista opetusta. Nopeiden Internet-yhteyksien välityksellä ulkomaisten opiskelijoiden lukumäärät kursseilla ovat kasvaneet ja kotimaisetkaan opiskelijat eivät välttämättä halua tulla fyysisesti paikalle yliopistoihin opiskelemaan, vaan opiskelevat mieluummin verkkopohjaisten ympäristöjen kautta (Chute et al, 1997). Jopa yliopistoalueella asuvat opiskelijat toisinaan suosivat verkkopohjaisia kursseja, koska he näin pystyvät ottamaan osaa sellaisillekin kursseille, joille ei muutoin mahtuisi mukaan esimerkiksi opetustilan asettaman fyysisen rajoitteen vuoksi. Toisaalta kurssien päällekkäisyyksistä syntyneet aikaongelmat on helppo ratkaista ottamalla toiseen kurssiin osaa verkon välityksellä (Klemm, 2001).

Virtuaalisen oppimisympäristön suunnittelun keskeisenä ajatuksena on pidettävä käsitystä oppimisesta. Oppimis- ja tiedonkäsitys on muuttunut vahvasti 1980-luvun loppupuolelta *behavioristisesta* käsityksestä kohti *konstruktivistista* oppimiskäsitystä (Tella, 1994). Behavioristinen oppimiskäsitys on hyvin opettajakeskeinen ja tarkastelee oppimista lähinnä ulkoisena tiedonsiirtona opettajalta opiskelijalle. Konstruktivistinen oppimiskäsitys puolestaan nähdään *opiskelijakeskeisenä* ja opiskelijan rooli tiedon oppimisessa on merkittävä. Suora tiedonsiirto opettajalta opiskelijalle on mahdotonta, joten opettaminen on nähtävä epäsuorana vaikuttamisena opiskelijoiden oppimiseen (Lehtinen, 1997). Konstruktivistisen oppimiskäsityksen mukaan oppiminen on opiskelijan omaa aktiivista ajatuksellista toimintaa ja informaation yksilöllistä uudelleen rakentamista vanhojen kokemusten, tietojen ja taitojen pohjalta (Korpi

yksilöllistä uudelleen rakentamista vanhojen kokemusten, tietojen ja taitojen pohjalta (Korpi et al, 2000). Oppimiseen vaikuttavia tekijöitä ovat opettajan puheen ja ohjauksen lisäksi myös kaikki muutkin oppimistilanteessa vaikuttavat tekijät kuten tiedonlähteet, työvälineet sekä opiskelijan toiminnan organisointi. Tämän vuoksi oppimisen tukemista suunniteltaessa on huomio kiinnitettävä koko siihen ympäristöön, joka kehystää opetusta ja oppimista (Lehtinen, 1997).

### **3.3 Virtuaalisen oppimisympäristön suunnittelu**

Farrell (1999) toteaa, että virtuaaliopintoja suunnitellaan hyvin innostuneesti kaikilla koulutusasteilla, mutta välttämättä ei pysähdytä miettimään *mitä*, *miksi* ja *kuinka* virtuaaliympäristöt tulisi luoda. Tietotekniikan opetuksen siirtäminen virtuaaliseen oppimisympäristöön on toisaalta varsin luontevaa alan luonteen vuoksi, mutta samalla hyvin haasteellista opiskelijoiden erilaisten oppimistyylien vuoksi. Oppimisympäristön tulisi tarjota monipuolisesti oppimismahdollisuuksia ja opiskelijalla itsellään tulisi olla valta ohjata omaa oppimistaan käyttämällä niitä välineitä, jotka tuntuvat parhaiten edistävän juuri hänen oppimistaan. Meisalon ym (2000) mukaan monipuolisia oppimismahdollisuuksia tarjoava ympäristö antaa opiskelijalle mahdollisuuden tietoverkkojen hyödyntämiseen siten, että opiskelija pääsee etsimään oppimisensa tukemiseen tarvitsemaansa tietoa myös paikallisen oppimisympäristön ulkopuolelta esimerkiksi hakukoneiden tai sähköpostin välityksellä. Toisaalta opiskelijan käyttöön on tarjottava yleiskäyttöisiä työvälineohjelmia kuten tekstinkäsittely- tai taulukkolaskentaohjelma, joiden avulla hän voi suoriutua opiskeluun liittyvistä tehtävistä.

Hyvässä ja monipuolisessa oppimisympäristössä on tarjolla opiskelijan ajatteluprosessin tukemiseen erilaisia kognitiivisia työkaluja, esimerkiksi käsitekarttaa käyttämällä opiskelija voi liittää yksittäiset käsitteet mielekkääksi kokonaisuudeksi. Toisaalta jos oppimisympäristöstä löytyy erityyppisiä opetusohjelmia, niin opiskelija voi käyttää omien henkilökohtaisten tarpeidensa mukaan sitä opetusohjelmaa mitä kulloinkin tuntee tarvitsevänsä. Digitaalisen oppimateriaalin käyttö on rikastuttava lisä verkkomateriaaliin. Opiskelijan käytössä voi olla joko paikallisesti CD-ROM -levyltä tai verkon välityksellä tutkittavissa tai poimittavissa olevia digitaalisia aineistoja. Erilaisten simulaatioiden käyttö puolestaan tarjoaa reaali- ja virtuaalimaailmojen yhdistymisen sekä auttaa usein abstraktien asioiden omaksumisessa. (Meisalo et al, 2000)

Opiskelijoilla on erilaisia tapoja oppia. Kun yksi opiskelija haluaa käyttää keksivää oppimistapaa ja miettiä, keksiä ja oivaltaa asioita, toinen puolestaan rakentaa oppimisensa ulkoa oppimisen varaan (Kuusinen & Korkiakangas, 1995). Tietotekniikan opiskelussa osa opiskelijoista oppii omatoimisesti kokeilemalla ohjelmien toimintaa tai tutkimalla valmiita ohjelmakoodeja. Sen sijaan osa opiskelijoista tarvitsee aina ensin esimerkin tai mallin, jonka jälkeen osaavat sitten lähteä soveltamaan tietoa itsenäisesti. Virtuaalisessa oppimisympäristössä erityisesti jälkimmäistä opiskelutapaa käyttävä opiskelijaryhmä on haasteellinen kurssin suunnittelijalle. Kurssia suunniteltaessa on mietittävä, kuinka opiskelijaa voisi tukea hänen opiskeluprosessissaan ja kuinka oppimisympäristöön pystyttäisiin sijoittamaan riittävästi motivoivia ja oppimista tukevia esimerkkejä. Verkkopohjaisen materiaalin suunnittelussa näyttääkin vaikeinta olevan arvioida, mikä auttaisi opiskelijoita heidän joutuessaan opinnoissaan umpikujaan sekä kuinka järjestelmä pystyisi heitä tässä tilanteessa tukemaan (Watanabe et al, 1999).

#### *Kurssimateriaalin suunnittelu*

Virtuaaliset kurssit ovat suuresta menestyksestään huolimatta tuottaneet myös pettymyksiä, kuten esimerkiksi kurssimateriaalien laadun suhteen (Dhanarajan, 2001). Opettajien ja kouluhallinnon tulisi ennen kurssin toteutusta suunnitella verkkopohjaisen toteutuksen oppisisällöt huolellisemmin. Lisäksi Dhanarajan (2001) toteaa, että virtuaalista oppimisympäristöä varten on tarpeen kehittää kokonaan uusia arviointityövälineitä, joilla voidaan mitata oppimissaavutuksia. Kurssin järjestämistä virtuaalisesti on harkittava huolellisesti eikä kurssia tulisikaan järjestää ainoastaan trendikkyuden tai muotisuuntauksen mukana. Jokaisen virtuaalikurssin suunnittelijan tulisi ennen kurssin toteuttamispäätöstä selvittää itselleen miksi hän on järjestämässä verkkopohjaista kurssia sekä onko kurssista saatava hyöty ylipäänsä koko ylimääräisen työn arvoinen (Klemm, 2001). Muutosta ei tulisi tehdä ainoastaan muutoksen vuoksi, vaan sillä on pyrittävä siihen, että materiaalit, opetus ja oppiminen muutoksen myötä tulevat laadukkaammiksi (Collis, 1998).

Klemmin (2001) mukaan materiaalin laatiminen ja vieminen virtuaaliseen oppimisympäristöön on aikaa vievää ja raskasta työtä, joten sitä tulisi miettiä huolella. Materiaalia olisi hyvä olla tekemässä eri alojen

asiantuntijoista koottu ryhmä, jossa osa toisi pedagogisen näkemyksen ja osa taas ympäristön toteutuksen teknisen asiantuntijuuden. Osa materiaalista olisi hyvä jakaa muun kuin Web-sivun kautta esimerkiksi sähköpostijakeluna. Virtuaalisen kurssin suunnittelussa materiaalin tuottajan on huomioitava, että opettaja ei ole enää pelkästään tiedonsiirtäjä vaan tärkeimpänä tavoitteena on *auttaa opiskelijoita löytämään, sulattamaan ja omaksumaan tietoa* sekä *soveltamaan* sitä käytännössä. Kurssimateriaalin suunnittelussa on siksi syytä kiinnittää erityistä huomiota siihen, että materiaalissa esitetään kaikki opiskelijan tarvitsema tieto selkeästi, jotta opiskelija löytää asiat helposti. Painopiste virtuaalisen kurssin suunnittelussa tulee olla itse asiasisällössä ja sen selkeässä ilmaisussa sekä oppimistapahtuman tukemisessa. Kurssimateriaalista tulee olla linkityksiä mahdollisiin muihin lisämateriaaleihin kuten esimerkiksi elektronisiin kirjastoihin. Lisäksi käytettävät video- tai äänileikkeet sekä digitaaliset materiaalit ovat omiaan tukemaan oppimisprosessia.

### *Tukitoimet*

Interaktiivista materiaalia opiskellessaan opiskelija saattaa joutua opiskelussaan umpikujatilanteeseen joko tiedostetusti tai tiedostamattaan ja järjestelmän tulee pystyä tarjoamaan opiskelijoille yhteydenottomahdollisuuksia joko tutoriin tai toisiin opiskelijoihin kaikissa tilanteissa. Tähän tarkoitukseen oppimisympäristöön tulee järjestää useita erilaisia välineitä ja väyliä: sähköposti, keskusteluforum ja chat-huone ovat vuorovaikutusvälineistä käytetyimmät. Opiskelijoita kannattaa ohjata ja rohkaista keskusteluihin: avoimet keskustelut ja tiimityöskentely ovat Klemmin (2001) mukaan parhaiten auttaneet oppimaan, mutta ilman tietoista keskusteluun ohjaamista opiskelijat eivät välttämättä ole kovinkaan hyvin lähteneet vuorovaikutukseen toistensa kanssa. Opettajan tai ohjaajan on syytä olla mukana rohkaisemassa oppimisympäristön vuorovaikutusvälineiden käyttöön.

Tietotekniikan ja erityisesti ohjelmoinnin opiskelussa ryhmätyö ja yhteistoiminnallinen oppiminen ovat usein hedelmällisimpiä oppimismuotoja. Kun kurssin toteutus viedään virtuaaliseen oppimisympäristöön, tulisi perinteisessä luokahuoneopetuksessa olevat oppimista tukevat menetelmät siirtää myös tähän uuteen ympäristöön (Cordani & Tucker, 1998). Virtuaalista oppimisympäristöä rakennettaessa käyttöliittymä tulisi suunnitella niin, että opiskelijat ja tutorit voivat mahdollisimman pitkälti käyttää jo ennestään tutuiksi tulleita välineitä niin kurssimateriaalin valmistelussa kuin kommunikoinnissakin. Luokkahuonetilan-

teessa ohjelmoinnin opetus jakaantuu usein kahteen osaan: luentoihin ja harjoitustehtävien tarkistamistilaisuuksiin. Monet opiskelijat oppivat nimenomaan harjoitusten tarkistamistilaisuuksissa käydyissä keskusteluissa kuinka tietty algoritmi tai ohjelma käytännössä toimii. Kun ohjelmoinnin opiskelu siirretään virtuaaliseen ympäristöön, oppimisympäristön tulisi tarjota opiskelijalle mahdollisuuksia niin julkisesti kuin yksityisestikin tapahtuvaan neuvon ja opastuksen kysymiseen (Cordani & Tucker, 1998). Valitettavasti oppimisympäristöistä usein puuttuvat tehokkaat ryhmätyövälineet, joita voisi käyttää hyväksi esimerkiksi ohjelmoinnin harjoituksista keskustelemisessä (Meisalo et al, 2001).

### *Oppimisen arvioinnin välineet*

Opiskelijoille tarjottavia arvioinnin ja itsearvioinnin välineitä ei valitettavasti useinkaan pidetä kovin merkittävinä, mutta virtuaalisessa oppimisympäristössä oppilaan oma reflektioiva oppimistapa on perusedellytys opintojen menestyksekkäälle suorittamiselle (Mäki-Komsi, 1999). Tämän vuoksi oppimisympäristössä tulisi olla riittävästi välineitä oman osaamisen arvioimiseen. Esimerkiksi WebCT-oppimisympäristö tarjoaa itsearviointivälineinä tietokilpailutyypin testin, oikein/väärin -väittämiä sekä lyhyitä valintatehtäviä (Landon, 2001). Adomi (1998) mainitsee oppimisen arvioinnin välineinä lisäksi essee-kysymykset sekä tilastolliset analysointivälineet. Ohjelmoinnin opiskelussa erilaiset visualisointivälineet, kuten esimerkiksi Jeliot tai Excel, voisivat toimia paitsi opetuksen tukivälineinä myös oppimisen itsearviointivälineinä. Opiskelija voisi testata tekemäänsä algoritmia esimerkiksi Jeliotilla ja arvioida animaation etenemisestä omaa onnistumistaan (Ben-Bassat Levy et al, 2001).

### ***3.4 Asynkroniset ja synkroniset vuorovaikutusvälineet***

Virtuaalisessa oppimisympäristössä oppimistapahtuman onnistuminen edellyttää toimivia vuorovaikutusväilyä. Verkostoituminen on tuonut mukanaan merkittäviä kehityssuuntauksia kuten esimerkiksi analogisesta tiedonsiirrosta siirtymisen digitaaliseen tiedonsiirtoon, joka mahdollistaa videon, äänen ja datan samanaikaisen lähetyksen. Televerkkojen välityksellä tapahtuvassa opetuksessa on tarjolla erilaisia teknisiä välineitä, jotka jaetaan *synkronisiin* ja *asynkronisiin* sen mukaan ovatko kommunikoinnin osapuolet samanaikaisesti läsnä tiedonsiirtotapahtumassa. Taulukossa 1 tutustutaan Wolzin ym (1997) mukaiseen jaotteluun synkronisista ja asynkronisista vuorovaikutusvälineistä. Välineet on jaettu kahden dimension,

ajan ja paikan, suhteen. Ajan dimensiossa tarkastellaan tapahtuuko vuorovaikutus samaan aikaan (synkronisesti) vai eri aikaan (asynkronisesti). Paikan dimensiossa tarkastellaan sitä, ovatko vuorovaikutukseen osallistuvat henkilöt vuorovaikutuksen tapahtuessa fyysisesti samassa paikassa.

**Taulukko 1 Synkroniset ja asynkroniset vuorovaikutusvälineet (Wolz ym (1997) mukaan)**

	<b>Samaan aikaan tapahtuva (synkroninen)</b>	<b>Eri aikaan tapahtuva (asynkroninen)</b>
<b>sama paikka</b>	<p>Ihmisten välinen:</p> <ul style="list-style-type: none"> <li>ohjaus</li> <li>vapaamuotoiset keskustelut</li> </ul> <p>Ryhmä:</p> <ul style="list-style-type: none"> <li>luento/seminaari</li> <li>laboratorio-työskentely</li> <li>ryhmä/projektityöt</li> <li>interaktiivinen TV</li> <li>vapaamuotoiset keskustelut</li> </ul>	<p>Ihmisten välinen:</p> <ul style="list-style-type: none"> <li>kirjasto</li> <li>interaktiivinen video</li> <li>CD-ROM</li> </ul> <p>Ryhmä:</p> <ul style="list-style-type: none"> <li>TV (lähetys / sähköet)</li> </ul>
<b>eri paikka</b>	<p>Ihmisten välinen:</p> <ul style="list-style-type: none"> <li>puhelin</li> <li>internet-puhelinjärjestelmä (esim. NetTalk, Net2Phone)</li> <li>Keskusteluyhteys (reaaliaikainen interaktiivinen teksti toisen käyttäjän kanssa)</li> </ul> <p>Ryhmä:</p> <ul style="list-style-type: none"> <li>telekonferenssi</li> <li>audiokonferenssi</li> <li>videokonferenssi</li> <li>Chat (järjestelmä, jota käytetään reaaliaikaiseen keskusteluun; joissakin mukana myös multimedia)</li> <li>MUs (variaatiot: MUD, MUCK, MUSH, MUSE, MOO - reaaliaikainen interaktiivinen järjestelmä, usein teksti-muotoinen, käytetään sosiaalisiin roolipeleihin, peleihin)</li> </ul>	<p>Ihmisten välinen:</p> <ul style="list-style-type: none"> <li>painettu materiaali (kirja, kirje, muistio, fax jne.)</li> <li>video-/ääni nauha</li> <li>vastauslaite / ääniposti</li> <li>sähköposti (mahdollistaa käyttäjän lähettää viestejä toinen toisilleen)</li> <li>tietokoneohjelma</li> </ul> <p>Ryhmä:</p> <ul style="list-style-type: none"> <li>LISTSERV (postituslistaohjelmat ryhmän kommunikointiin)</li> <li>USENET (asynkroninen tekstimuotoinen keskustelu useista erillisiin uutisryhmiin jaetuista aiheista)</li> <li>ryhmävälineistöt (yhteistoiminnalliset järjestelmät, jotka sisältävät jaetut kirjastot, keskustelut, tekstit, graafiset alueet jne.)</li> <li>WWW (maailmanlaajuinen http-palvelimien verkko, joka mahdollistaa tekstin, grafiikan, äänitiedostojen jne.)</li> </ul>

Synkronisesta vuorovaikutuksesta puhutaan silloin kun ohjaaja, aiheen asiantuntijat ja/tai opiskelijat työskentelevät samanaikaisesti sekä näkevät ja kuulevat toisensa. On huomattavaa, että oppimistilanteeseen osallistujien ei välttämättä tarvitse kuitenkaan olla fyysisesti samassa paikassa kuten taulukosta 1 käy ilmi. Esimerkiksi puhelin- tai chat-keskustelut ovat synkronisia vuorovaikutustilanteita,

joissa vuorovaikutukseen osallistujat voivat olla fyysisesti hyvinkin kaukana toisistaan. Olennaista on, että vuorovaikutushetkellä kaikki vuorovaikutukseen osallistuvat ovat samanaikaisesti läsnä.

Synkronisia tiedonvälitysjärjestelmiä ovat *audiokonferenssit*, *audiograafiset konferenssit* sekä *videokonferenssit*. Audiokonferenssissa eri paikoissa olevat osallistujat pystyvät kuulemaan toistensa äänet, mutta eivät voi nähdä toisiaan. Käytännössä toteutus yleensä tapahtuu puhelimen ja konferenssipuhelujen välityksellä. Audiograafinen konferenssi on muutoin vastaava kuin audiokonferenssi, mutta käytössä on toinen puhelinlinja kuvan lähettämistä varten. Kuva voidaan lähettää esimerkiksi tietokoneen välityksellä vastaanottajan näyttöpäätteelle ja kuka tahansa audiograafiseen konferenssiin osallistuja voi lähettää kuvaa tai piirtää virtuaaliselle liitutaalulle (whiteboard) kaaviokuvia, joita toiset osallistujat voivat katsoa. Videokonferenssissa kaikilla osallistujilla on videokamerat ja monitorit niin, että osallistujat näkevät ja kuulevat toinen toisensa. Videon välityksellä on mahdollista myös näyttää kuvia ja kaavioita esityksen tueksi. (Tiffin & Rajasingham, 1995)

Asynkronisessa kommunikoinnissa vuorovaikutus ei ole reaaliaikaista, vaan se on sidoksissa opiskelijan henkilökohtaiseen aikatauluun. Asynkronisia vuorovaikutusvälineitä ovat esimerkiksi sähköposti, multimedia- ja online-tietokannat, virtuaalikirjastot, sähköiset ilmoitustaulut, keskustelufoorumit ja Internet (Chute et al, 1997). Sähköposti on erityisen tärkeä väline opiskelijan ja opettajan, asiantuntijoiden tai toisten opiskelijoiden väliseen keskusteluun virtuaalisessa oppimisympäristössä. Opiskelija voi lähettää opettajalle tai asiantuntijalle kysymyksiä opiskelutilanteessa kohtaamista ongelmista ja tämä puolestaan vastaa kysymyksiin oman aikataulunsa mukaisesti. Opiskelijan ei tarvitse saada opettajaan henkilökohtaista reaaliaikaista yhteyttä, vaan mahdollisuus tuensaantiin on ajasta ja paikasta riippumatonta.

Sähköisen ilmoitustaulun käyttäminen on virtuaalisessa oppimisympäristössä lähinnä informaation lähettämistä: opettaja voi tiedottaa kurssiin liittyvät yleiset järjestelyt sekä koko opiskelijakuntaa koskevat ilmoitukset kuten esimerkiksi tenttien tai harjoitusten aikataulut. Tiffinin ja Rajasinghamin (1995) mukaan jokaisella kiinnostuneella tulisi olla mahdollisuus lukea sähköistä ilmoitustaulua sekä lisätä sinne omia viestejään. Käytännössä oppimisympäristöissä yleensä rajataan esimerkiksi keskustelufoorumi yleiseen käyttöön, jonne kuka tahansa voi laittaa kommentteja kurssijärjestelyistä tai –

materiaaleista. Sähköinen ilmoitustaulu säästetään vain viralliseen tiedottamiseen, jolloin esimerkiksi ainoastaan tutorilla, opettajalla tai teknisellä tuella olisi oikeus laittaa ilmoitustaululle ilmoituksia.

Toimivaa verkkopohjaista oppimisympäristöä suunniteltaessa ohjaajien ja palvelun tarjoajien tulisi pyrkiä löytämään oikea yhdistelmä synkronisia ja asynkronisia välineitä, jotta oppimisympäristöstä tulisi mahdollisimman rikas ja monipuolinen (Chute et al, 1997). Virtuaalisessa oppimisympäristössä oppiminen tapahtuu yleensä pääosin itsenäisesti verkkomateriaaleja hyväksi käyttäen, joten opiskelijan oppimisprosessin tukemisessa korostuvat erityisesti oppimisympäristössä tarjolla olevien asynkronisten välineiden käyttömahdollisuudet.

### ***3.5 Opiskelijana virtuaalisessa oppimisympäristössä***

Vuosien myötä oppimisen ja opettamisen käsitykset ovat muuttuneet ja uusien virtuaalisten oppimisympäristöjen käyttöönoton yhteydessä syntyy tarpeita kokonaan uusiin tehtäväkenttiin. Virtuaalisessa oppimisympäristössä mukana toimivat ainakin seuraavat henkilöt (Korpi et al, 2000):

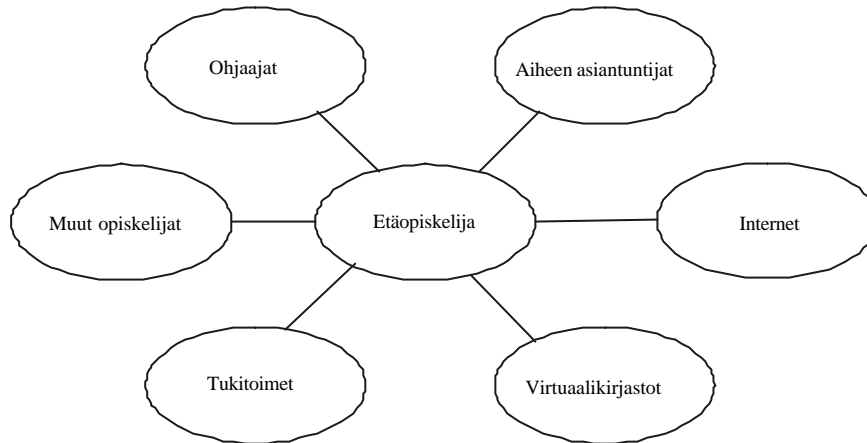
- *oppilas* (opiskelija, oppija),
- *opettaja* (ohjaaja, tutor),
- *mentor* (toinen opiskelija, joka tukee opiskelutovereitaan opiskelussaan),
- *kurssin sisältöalueen asiantuntija*,
- *tekninen tuki*,
- *oppimateriaalin suunnittelija*,
- *oppimateriaalin tuottaja*,
- *projektipäällikkö* koko prosessin ohjaajana,
- *kurssisihteeri* oppilashallinnon työntekijänä sekä
- koulutusorganisaation *rehtori tai johtaja*.

Virtuaalisessa oppimisympäristössä opiskelijan rooli korostuu ja ympäristöä kuvataan yleensä oppilaskeskeisenä (Chute et al, 1997). Kuvassa 2 tarkastellaan mitä kaikkia tekijöitä verkkopohjaisessa oppimisympäristössä on yksittäisen etäopiskelijan opiskelun tukena. Opiskelija on



verkon välityksellä yhteydessä toisiin opiskelijoihin, ohjaajiin, asiantuntijoihin sekä tarvittaessa myös tekniseen tukeen. Lisätietoja hän voi hakea Internetistä tai virtuaalisista kirjastoista.

Verkkopohjaisessa oppimisympäristössä avun saaminen riippuu omasta aktiivisuudesta: jos opiskelija kohtaa omassa työskentelyssään ongelmia, hänen on itsenäisesti kyettävä etsimään siihen apua ohjaajilta, asiantuntijoilta tai muilta kurssin opiskelijoilta.



**Kuva 2 Opiskelija verkkopohjaisessa oppimisympäristössä (Chute ym (1997) mukaan)**

Verkkopohjaista oppimisympäristöä ovat kuvanneet lisäksi muun muassa Multisilta (1997), Meisalo ym (2000) ja Adorni ym (1998). Taulukkoon 2 on koottu näistä malleista keskeiset tunnuspiirteet yhdessä kuvassa 2 esitetyn Chute ym:n (1997) mallin kanssa.

**Taulukko 2 Verkkopohjaisen oppimisympäristön keskeiset piirteet**

<b>Malli</b>	<b>Keskeinen piirre</b>	<b>Rakennetekijät</b>
Adomi ym (1998)	<i>Käyttöliittymäkeskeinen:</i> Huomio kiinnitetään käyttöliittymään ja oppilaalle tarjottaviin työvälineisiin sekä niiden helppokäyttöisyyteen ja käytettävyyteen opintojen tukemisessa.	<ol style="list-style-type: none"> <li>1. Oppimateriaali toteutettuna hypermediaa hyväksi käyttäen</li> <li>2. Vuorovaikutusvälineet kuten sähköposti, ilmoitustaulut jne.</li> <li>3. Välineet arviointia ja itsearviointia varten</li> <li>4. Harjoitusten suorittamiseen tarvittavat ohjelmistot ja muut välineet esimerkiksi simulaatiot</li> </ol>
Chute ym (1997)	<i>Oppilaskeskeinen:</i> Huomio kiinnittyy opiskelijan vastuullisuuteen omista opinnoistaan. Lisätiedon hankinta ja tuen saanti opiskelijan omasta aktiivisuudesta kiinni.	<ol style="list-style-type: none"> <li>1. Internet</li> <li>2. Opittavan aiheen asiantuntijat</li> <li>3. Ohjaajat</li> <li>4. Muut opiskelijat</li> <li>5. Tukitoimet</li> <li>6. Virtuaalikirjastot</li> </ol>
Meisalo ym (2000)	<i>Informaatiokeskeinen:</i> Avoin torimalli, jossa opiskelija etsii tietoa poimien sitä tiedon torilla olevilta kojuilta. Opiskelija nähdään aktiivisena tiedon kerääjänä ja tietokone on työväline, jonka avulla opiskelija voi työstää kokoamaansa tietoa.	<ol style="list-style-type: none"> <li>1. Internet</li> <li>2. Opiskelijat</li> <li>3. Työvälineohjelmistot</li> <li>4. Ryhmätyötilat</li> </ol>
Multisilta (1997)	<i>Verkkokeskeinen:</i> Huomio kiinnittyy verkon kautta tapahtuvaan tiedon siirtoon. Opiskelija yhteydessä toisiin opiskelijoihin, tietokantoihin, hallinnon rutineihin ym. verkon välityksellä.	<ol style="list-style-type: none"> <li>1. Opiskelijat</li> <li>2. Opettaja</li> <li>3. Oppimateriaalin tuottaja</li> <li>4. Tietokannat</li> <li>5. Asiantuntijat</li> </ol>

### *Opiskelijan vastuullisuus*

Perinteisessä luokkahuonetyöskentelyssä hyvä opettaja kykenee näkemään opiskelijoiden kasvoilta onko opetettava asia ymmärretty. Opettaja "lukee" tilannetta tarkkaillen opiskelijoiden keskittyneisyyttä, ilmeitä, eleitä ja tarvittaessa toistaa asian vielä uudelleen tai antaa lisäesimerkkejä varmistaakseen, että uusi opittava asia on todella ymmärretty. Virtuaalisessa ympäristössä oppimisesta vastuun kantaminen jää opiskelijan itsensä harteille. Opiskelijalta edellytetään Mäki-Komsin (1999) mukaan tietotaidollisten valmiuksien ohella omatoimisuutta, itsenäisyyttä ja luovuutta sekä kykyä reflektiiviseen toimintaan. Omaa oppimista on kyettävä arvioimaan ja omaa opiskelutapaa tarvittaessa muuttamaan. Virtuaalisessa oppimisympäristössä oppimista auttaa, jos opiskelija on tietoinen siitä, mitkä tiedot tai taidot ovat

hyödyllisiä tai hyödyttömiä jonkin tehtävän suorittamisen tai asioiden ymmärtämisen kannalta. Näiden *metakognitiivisten taitojen* oppiminen on koko oppimisprosessin kannalta merkittävä taito (Kuusinen & Korkiakangas, 1995).

**Taulukko 3 Perinteisen ja teknologiaa hyödyntävän opiskeluympäristön tyypillisiä piirteitä (Tella, 1994)**

<b>TRADITIONAALINEN LUOKKAHUONEOPETUS</b>	<b>TEKNOLOGIAA HYÖDYNTÄVÄ OPISKELUYMPÄRISTÖ</b>
Vastuu oppimisesta opettajalla	Vastuu oppimisesta oppijalla
Opettaja sisällön asiantuntija	Opettaja oppimisen asiantuntija
Opettaminen instruktiivisena prosessina	Opettaminen konstruktivisena prosessina
Passiiviset oppijat	Aktiiviset oppijat
Opettaja asioiden esittäjänä ja tiedonvälittäjänä	Opettaja oppimistapahtuman helpottajana ja tiedon järjestelijänä
Oppijalla käytettävissään vain oppikirjoja ja muuta painettua, vanhentunutta aineistoa	Oppijalla käytettävissään uuden tekniikan avulla valtavia määriä informaatiota
Luokkahuoneen eristyneisyys	Opiskelu ympäristö ulottuu luokan seinien ulkopuolelle
Oppija informaation vastaanottaja	Oppija luovana ongelmanratkaisijana ja informaation käyttäjänä
Painotus yksilöllisissä projekteissa ja saavutuksissa	Painotus yhteistoiminnallisissa ja ryhmäprojektorientoituneissa toiminnoissa
Opettaja hoitamassa hallinnollisia tehtäviä valtaosan päivää	Tietotekniikka helpottamassa opettajien hallintotyötä

Tella (1994) kuvaa traditionaalisen luokkahuoneopetuksen ja nykyaikaista teknologiaa hyödyntävän oppimisympäristön eroavuuksia Pollard ja Pollardin (1993) mukaan taulukossa 3. Merkittävimpänä eroavuutena nähdään opiskelijan roolin muuttuminen aktiivisemmaksi sekä opiskelijan vastuullisuus

omista opinnoista ja omasta oppimisesta on kasvanut siirryttäessä perinteisestä luokkahuoneopetuksesta teknologiaa hyödyntävään oppimisympäristöön. Perinteisessä luokkahuoneopetuksessa opettaja opettaa uudet asiat erilaisia havaintovälineitä hyväksi käyttäen suljetussa luokkahuonetilanteessa. Tiedon siirtäminen ja antaminen on opettajan käsissä. Tieto- ja viestintäteknologiaa hyödyntävissä oppimisympäristöissä opiskelijoilta edellytetään valmiuksia itsenäiseen tiedon hankintaan ja prosessointiin (Mäki-Komsi, 1999). Menestyksekkään oppimisen edellytyksenä on, että opiskelija erottaa oleellisen asian ja osaa muodostaa uusia tiedonrakenteita vanhojen tietorakennelmien pohjalta.

Tietotekniikan perusteita virtuaalisessa ympäristössä opiskellessaan opiskelijan tulisi osata hakea tukea riittävän ajoissa, jos ongelmia tulee vastaan. Mikäli opiskelijalle jää puutteellinen tai osin virheellinenkin käsitys esimerkiksi ohjelmoinnin perusteista, hänelle tulee hyvin todennäköisesti myöhemmin opintojen edetessä ongelmia, sillä ohjelmointi rakentuu aiemmin opittujen rakenteiden päälle. Lee (2000) toteaa, että opiskelijan luonteenpiirteet ja henkilökohtaiset ominaisuudet kuten motivaatio, itsearviointi ja aiemmat oppimiskokemukset yhdessä oppimisympäristön kanssa vaikuttavat siihen, kuinka hän oivaltaa ja ymmärtää opetettavan asian. Ymmärrettyään uuden asian, opiskelija mukauttaa sen osaksi vanhaa jo aiemmin opittua asiaa. Uusi asia siis rakentuu aiemmin opitun päälle. Virtuaalisessa oppimisympäristössä näitä oivaltavia oppimistilanteita tulisi pyrkiä järjestämään opiskelijalle esimerkiksi tarjoamalla hieman helpompien esimerkkien ja tehtävien avulla onnistumisen elämyksiä oppimistilanteeseen. Onnistumisen elämykset puolestaan vaikuttavat myönteisesti opiskelijan motivaatioon.

## *Motivaatio*

Opiskelijan opiskelumotivaation ylläpitäminen on tärkeää sillä verkkopohjaisessa opiskelussa jopa 30-50 % kurssien aloittaneista jättäytyy pois ennen kurssin päättymistä (Cornell & Martin, 1997).

Motivoitumiseen vaikuttavat:

- *tapa, jolla oppilaat ja opettajat suhtautuvat teknologiapohjaisen oppimisympäristöön työvälineenä,*
- *aiempi kokemus,*
- *ennakkoluulot teknologiaa kohtaan,*
- *opiskeltavan asian asiayhteys,*
- *kurssin interaktiivisuuden aste,*
- *järjestelmän käytön hankaluudet,*
- *järjestelmän helppo saatavuus sekä*
- *oppilaiden ja opettajien kyvykkyys tai kyvyttömyys kommunikointiin (Cornell & Martin, 1997).*

Opiskelumotivaatioon vaikuttaa sekä ulkoinen että sisäinen motivaatio. Ulkoisia motivointikeinoja ovat esimerkiksi arvosana, rangaistuksen pelko, stipendi tai hyväpalkkainen työpaikka. Sisäisestä motivoinnista Meisalo ym (2000) käyttävät hyvin osuvaa nimeä *oppimisen halu*, joka auttaa opiskelijaa sitoutumaan opiskeluun verkkopohjaisessa oppimisympäristössä. Opiskelijan sisäistä motivaatiota tulisi pyrkiä ylläpitämään läpi koko kurssin esimerkiksi tarjoamalla onnistumisen elämyksiä tai käyttämällä muita kannustimia. Oppiminen on järjestettävä merkityksellisen oppimisen ja tarkoituksenmukaisten tavoitteiden ympärille, jotka omalta osaltaan ovat auttamassa opiskelijan motivoitumisessa (Korpi et al, 2000).

Cornellin ja Martinin (1997) mukaan onnistuneen kurssisuorituksen saavuttamiseksi vaaditaan kolme asiaa:

1. *opiskelijalla tulee olla halu suorittaa kurssi loppuun saakka.*

2. *liian varhainen työn vastaanottaminen saattaa vaikuttaa opintojen keskeyttämiss päätöksen syntymiseen ja*
3. *opiskelijat, jotka ovat jo aiemmin suorittaneet jonkin etäkurssin loppuun, suorittavat herkemmin myös uuden etäkurssin loppuun saakka.*

Opiskelun tukitoimet tulisi suunnata niille opiskelijoille, joilla motivaatio ei ole kovin hyvä. Kun opiskelija on virtuaalisessa oppimisympäristössä itse vastuussa opinnoistaan ja niiden etenemisestä, hänellä on oltava mahdollisuus saada tukea oikea-aikaisesti juuri silloin kun hän opinnoissaan sitä tarvitsee esimerkiksi sähköpostin välityksellä (Korpi et al, 2000). Oppimistilanne tulisi järjestää niin, että se tukee yhteisöllisyyttä, jaettua asiantuntijuutta sekä mahdollisuutta päästä sisään opiskelijoiden muodostamaan keskustelemaan yhteisöön. Oppimisympäristön tulee olla niin turvallinen, että opiskelija tarvittaessa pystyy pyytämään yksityisesti tukea ja saamaan henkilökohtaista palautetta suorituksestaan.

#### 4. JOENSUUN YLIOPISTO: TIETOJENKÄSITTELYTIETEEN VIRTUAALIAPPROBATUR

Joensuun yliopiston tietojenkäsittelytieteen laitos on järjestänyt vuoden 2000 syyslukukaudelta alkaen Pohjois-Karjalan maakunnan lukiolaisille mahdollisuuden suorittaa pääosin verkkotyöskentelyyn pohjautuen 15 opintoviikon laajuisen tietojenkäsittelytieteen approbatur –opintokokonaisuuden. Lukiolaisille tarjottua opintokokonaisuutta voidaan varsin hyvällä syyllä edellä luvussa 3 esitettyihin määritelmiin nojautuen kutsua *virtuaaliseksi opintokokonaisuudeksi*. Joensuun virtuaaliapprobatur-projekti on osa Suomen virtuaaliyliopistohanketta ja liittyy opetusministeriön käynnistämään valtakunnalliseen koulutuksen ja tutkimuksen tietostrategia 2000-2004 -hankkeeseen. Hankkeen tarkoituksena on jatkaa Suomen kehittämistä kohti tiedon ja osaamisen yhteiskuntaa. Tavoitteena on antaa tietoyhteiskuntavalmiudet kaikille, vahvistaa opetushenkilöstön tietoyhteiskuntaosaamista sekä lisätä tieto- ja sisältöteollisuuden ammattilaisten osaamista. Pyrkimyksenä on lisäksi vakiinnuttaa verkko-opiskelu, tutkimustiedon ja oppimateriaalin elektroninen julkaiseminen, jäsentäminen ja jakelu sekä tietoyhteiskunnan rakenteiden vahvistaminen. (Opetusministeriö, 2001)

Joensuun yliopiston tietojenkäsittelytieteen laitos markkinoi virtuaaliapprobaturia ensimmäisen kerran Pohjois-Karjalan maakunnan lukiolaisille toukokuussa 2000 lähetetyllä esitteellä sekä elokuussa 2000 lähetetyllä julisteilla. Maakunnan kuudestatoista lukiosta kaiken kaikkiaan kahdestatoista ilmoittautui mukaan yhteensä 89 opiskelijaa. Lukiolaiset työskentelevät omalta kotipaikkakunnaltaan käsin käyttäen hyväksi Joensuun yliopiston tietojenkäsittelytieteen laitoksen tekemää verkkomateriaalia sekä kursseihin liittyvää kurssikirjallisuutta. Virtuaalisen opintokokonaisuuden järjestämisestä Joensuussa on käytetty nimeä *Candle-malli* (Haataja et al, 2001), jossa opiskelijoille pyritään näyttämään oppimisen polku verkkopohjaisessa opiskeluympäristössä ”sähköisten kynttilöiden” (Candle) avulla. Opiskelu tapahtuu pääosin Internetin välityksellä ja opintoja ohjataan verkossa olevan oppimisympäristön kautta. Oppimisympäristön suunnittelu rakentuu seuraavan neljän peruspilarin varaan:

1. *opiskelijalla on verkkopohjaisesta työskentelytavasta huolimatta mahdollisuus sosiaalisiin kontakteihin,*
2. *painettu oppimateriaali linkitetty aktivoivaan WWW-materiaaliin,*
3. *opetuksen järjestäminen lähes täysin Internetin välityksellä sekä*

#### 4. oppimisympäristön säilyttäminen käyttäjäystävällisenä ja selkeänä.

Candle-mallissa lisäksi oletetaan, että opiskelijalla on koululla tutoropettajan tuki sekä käytössään kursseihin kuuluvat oppikirjat.

Tutustutaan seuraavaksi tarkemmin virtuaaliapprobaturin sisältöön ja toteutukseen.

#### 4.1 Virtuaaliapprobaturin sisältö

Virtuaaliapprobatur on 15 opintoviikon laajuinen opintokokonaisuus, jonka suorittamisesta opiskelijat saavat arvosanasuorituksen. Lukiolaiset suorittavat approbatur-kurssit lähes kokonaan verkkotyöskentelyyn pohjautuen. Tutustutaan seuraavassa virtuaaliapprobatur -opintokokonaisuuteen ja sen toteutusaikatauluun (Kuva 3).

Kurssin nimi	2000					2001											
	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Virtuaaliapprobaturiin ilmoittautuminen sekä aloitustilaisuus	■																
Tietokone, käyttöjärjestelmä ja tietoverkko		■	■														
Tietokoneen, työvälineohjelmistojen ja verkkopalvelujen käyttö				■	■												
Tietotekniikka ja yhteiskunta																■	■
Algoritmien suunnittelu													■	■	■		
Tietokoneen rakenne ja käyttöjärjestelmä						■	■	■	■								
Tietojenkäsittelytieteen tutkimusaloja													■	■	■		
Ohjelmointi, osa 1		■	■	■	■												
Ohjelmointi, osa 2						■	■	■	■	■							
Perusopintojen harjoitustyö											■	■	■	■	■		
Virtuaaliapprobaturin päätöstilaisuus																	■

**Kuva 3 Virtuaaliapprobaturin toteutusaikataulu lukuvuonna 2000-2001**

Opintojen aloitustilaisuudessa opiskelijoille jaettiin itseopiskeluopas (liite 1), jossa esiteltiin käytettävä oppimisympäristö, annettiin yleisiä ohjeita etäopiskelua varten sekä esiteltiin approbatur-kurssien sisällöt ja vaatimukset lukuvuonna 2000-2001. Kurssien laajuudet on esitelty taulukossa 4.



**Taulukko 4 Virtuaaliapprobatur-kurssien laajuudet lukuvuonna 2000-2001**

<b>Kurssin nimi</b>	<b>Opintoviikkoja</b>
3 ov Tietotekniikka Tietokone, käyttöjärjestelmä ja tietoverkko Tietokoneen, työvälineohjelmistojen ja verkkopalvelujen käyttö Tietotekniikka ja yhteiskunta	1 ov 1 ov 1 ov
5 ov Johdatus tietojenkäsittelytieteeseen Algoritmien suunnittelu Tietokoneen rakenne ja käyttöjärjestelmä Tietojenkäsittelytieteen tutkimusaloja	1 ov 2 ov 2 ov
5 ov Ohjelmointi Ohjelmointi, osa 1 Ohjelmointi, osa 2	2 ov 3 ov
2 ov Perusopintojen harjoitustyö	2 ov
<b>Yhteensä</b>	<b>15 ov</b>

#### *4.2 Sosiaalisten kontaktien mahdollisuus*

Vaikka opiskelijat opiskelevat verkkopohjaisessa ympäristössä pääosin itsenäisesti, heille on tarjolla useita väyliä sosiaalisten kontaktien muodostamiseen sekä tuen saantiin. Autenttisiin opiskelutarpeisiin tukea annetaan harjoitusten, opiskeluvälineiden sekä palautteiden kautta (Haataja et al, 2001).

#### *Tutoropettajat*

Opiskelijoiden tukena kouluilla ja yhteyslinkkinä yliopiston ja opiskelijoiden välillä toimii tutoropettaja, joka auttaa alkuvaiheessa opiskelijoita WebCT-oppimisympäristöön tutustumisessa ja ohjaa heitä itsenäiseen työskentelyyn. Osalla lukioista tutoropettaja pitää virtuaaliapprobatur-opiskelijoiden kanssa säännöllisiä viikkotapaamisia. Näissä tapaamisissa tutoropettaja ja opiskelijat käyvät läpi verkkopohjaisessa työskentelyssä esiin tulleita ongelmatilanteita sekä pohtivat mahdollisia muitakin opiskeluun liittyviä ongelmakysymyksiä. Tutoropettaja valvoo kouluilla pidettävät tentit sekä arvioi näyttökokeet. Tutoropettajilta ei edellytetä erityistä tietotekniikan osaamista.

### *Ohjaajat yliopistolla*

Yliopistolla toimivien ohjaajien tehtävänä on hoitaa kurssin ylläpidolliset tehtävät opiskelijatunnusten jakamisesta alkaen. Ohjaajat huolehtivat, että kurssimateriaalit ja harjoitustehtävät ovat oikeaan aikaan saatavilla sekä tarkastavat tehtävät. Tehtävien tarkastamisen yhteydessä ohjaajat kommentoivat tehtävän suoritusta. Mikäli opiskelijalla on ollut jokin selkeä vaikeus tehtävän suorittamisessa, sen ratkaisemiseksi annetaan pientä opastusta sekä kehoitetaan tutustumaan malliratkaisuihin. Ohjaajat tiedottavat yleisistä asioista lähinnä keskustelufoorumia käyttäen ja vastaavat keskustelufoorumien tai sähköpostin kautta opiskelijoiden esittämiin kysymyksiin. Tehtäviin kuuluu lisäksi kaikkien tenttien, myös lukioilla pidettyjen, tarkastaminen sekä yliopistolla pidettyjen tenttitilaisuuksien valvominen. Opiskelijoilla on lisäksi mahdollisuus kysyä ohjaajilta henkilökohtaisesti neuvoa sähköpostitse tai puhelimitse.

### *Muut virtuaaliapprobatur opiskelijat*

Toisten opiskelijoiden antama tuki tulee virtuaalisessa oppimisympäristössä perinteistä kontaktiopetusta merkittävämmäksi, koska opiskelijat ovat pääosin yksin opiskelutilanteessa. Joensuussa oppimisympäristönä käytetyn WebCT:n neljä chat-huonetta ja keskustelufoorum tarjoavat opiskelijoille kanavat keskusteluihin. Parhaimmillaan nämä voisivat toimia merkittävänä opiskelun tukivälineinä. Sen lisäksi opiskelijoilla on mahdollisuus perinteiseen tapaan kokoontua omalla koululla ryhmiin ja keskustella opiskeluun liittyvistä ongelmakohdista jo ennestään tuttujen opiskelutovereiden kanssa.

### 4.3 Painetun oppimateriaalin linkittäminen verkkomateriaaliin

Opiskelumateriaali koostuu lähes kaikkien virtuaaliapprobatur-kurssien kohdalla verkkomateriaalista ja oppikirjasta. Ohjelmointi 1 –kurssilla oppikirjana käytetään kirjaa Wikla, A.: Ohjelmoinnin perusteet Javalla, OtaData, 1999. Oppilaat käyvät teorian läpi oppikirjasta ja verkkomateriaalista ja tekevät sen jälkeen viikoittaiset harjoitustehtävät. Palautusaikaa harjoitustehtäviin lukuvuonna 2000-2001 oli viikosta kahteen viikkoon. Teoriasivustoilta on viittaukset oppikirjan sivuihin ja tehtäviin, joihin opiskelijoiden tuli kunkin teoriaviikon aikana tutustua. Verkkomateriaalissa on teorian lisäksi lisäesimerkkejä opetettavasta asiasta. Verkkomateriaali toimii samalla opiskelijoille opiskeluoppaana, sillä se rytmittää opinnot viikoittaisiin opintokokonaisuuksiin.

OHJELMOINNIN PERUSTEET (osa 1)

Viikko 5 - Valintalauseet ja loogiset operaattorit - Loogiset operaattorit

Loogiset operaattorit:

Loogisten operaattorien operandit voivat olla vain todenarvoisia lausekkeita tai vertailuja.

Operaatio	Merkitys	Arvot
ja	ehdollinen "ja"	true, jos molemmat operandit ovat arvoitaan true
	ehdollinen "tai"	true, jos jompikumpi operandi on true
^	poissulkeva "tai" (XOR)	true, kun toinen operandi on true ja toinen false
!	negatio "ei"	muuttaa true arvon falseksi ja false arvon trueksi

alla olevalla animaatiolla voit kokeilla eri loogisten operaattorien toimintaa, kun operandit saavat arvoja true tai false.

**AND-operaatio &&**

If (ehto1 == true) && (ehto2 == true)  
Lause on tosi kun molemmat ehdot täyttyvät

OFF && ON = False

Kuva 4 Esimerkki Ohjelmointi 1 -kurssin materiaalista

Opiskelijat saavat tekemistään harjoitustehtävistä palautetta sekä pystyvät seuraamaan omaa tehtyjen tehtävien kertymää WebCT:ssä. Kurssimateriaaliin ilmestyy malliratkaisut kaikista harjoitustehtävistä sen jälkeen, kun kyseisen viikon harjoitustehtävät on tarkastettu. Malliratkaisuuina käytetään opiskelijoiden tekemiä ratkaisuja tai yliopistolta ohjaajan tekemää esimerkkiratkaisua. Visualisointikeinoina on käytetty muun muassa Flash-animaatioita (Kuva 4) ohjelmoinnin perusrakenteiden toiminnan mallintamiseen. Candle-mallissa korostuu tekemällä oppimisen merkitys ja harjoitusten tekemisen tärkeyttä on korostettu muun muassa tehdyistä harjoituksista saatavilla bonuspisteillä.

#### 4.4 Opiskelu Internetin välityksellä

Opiskelu on järjestetty lähes kokonaan Internetin välityksellä. Kurssien suorittamistapoja on kolme: tentti yliopistolla, tentti tai näyttökoe lukiolla sekä itsenäisesti tehtävä työ, joka palautetaan joko sähköpostin tai WebCT:n kautta. Taulukkoon 5 on koottu virtuaaliapprobatur-kurssit jaoteltuna niiden suorittamistavan mukaan.

**Taulukko 5 Virtuaaliapprobatur-kurssien suoritustapa lukuvuonna 2000-2001**

Lukiolla suoritettavat tentit	Yliopistolla suoritettavat tentit	Internetin välityksellä palautettavat työt
Tietokone, käyttöjärjestelmä ja tietoverkko	Ohjelmointi, osa 1	Tietojenkäsittelytieteen tutkimusaloja (tehtävänanto kesäkoulussa yliopistolla)
Tietokoneen, työvälinohjelmistojen ja verkkopalvelujen käyttö (näyttökoe)	Ohjelmointi, osa 2	Tietotekniikka ja etiikka
Algoritmien suunnittelu (tehtävänanto ja alkuluennot kesäkoulussa yliopistolla)		Perusopintojen harjoitustyö (tehtävänanto ohjelmointi 2 –tentin jälkeen yliopistolla)
Tietokoneen rakenne ja käyttöjärjestelmä		

Harjoitustyötä ja näyttökokeella suoritettavaa kurssia lukuunottamatta kaikkiin muihin kursseihin liittyy myös verkkomateriaalia.

#### 4.5 Oppimisympäristön helppokäyttöisyys

Candle-mallin luomisessa on pidetty tärkeänä rakentaa oppimisympäristön käyttöliittymä mahdollisimman yksinkertaiseksi ja opiskelijaystävälliseksi (Haataja et al, 2001). Oppimisympäristön valinta suoritettiin keväällä 2000, jolloin virtuaaliapprobatur-työryhmä tutki tutkimustuloksia eri oppimisympäristöistä ja niiden käytöstä sekä testasi, mikä tuntuisi sopivimmalta virtuaaliapprobaturin toteutuslueksi Joensuussa. Oppimisympäristöksi valittiin WebCT. Päätöksessä ratkaisevana tekijänä oli, että WebCT:n kautta toteutettuna oppimisympäristö on opiskelijoiden saatavilla vaivattomasti ja ilman opiskelijoille aiheuttavia kohtuuttoman suuria kustannuksia. Opiskelijat tarvitsevat ainoastaan tietokoneen ja toimivan Internet-yhteyden sekä Web-selaimen päästäkseen oppimisympäristöön. Kuvassa 5 on esimerkki virtuaaliapprobaturin yksittäisen kurssin pääsivusta.



**Kuva 5** Virtuaaliapprobaturin yksittäisen kurssin pääsivu

WebCT-ympäristö jakaantuu neljään toisistaan erottuvaan osioon:

1. *oppimateriaaliin,*
2. *kurssin hallintatyökaluihin,*
3. *vuorovaikutusvälineisiin sekä*
4. *oppimista tukeviin työvälineisiin.*

Yksittäisen kurssin eri osioihin pääsee kyseisen kurssin pääsivun kautta ja kaikkien kurssien pääsivuille on lisätty yleiset vuorovaikutukseen ja kurssin hallintaan liittyvät toiminnot. Jokaisen kurssin pääsivulta löytyy linkit keskustelufoorumille, ilmoitustaululle ja sähköpostiin. Tutustutaan seuraavaksi Joensuussa käytössä olleen WebCT-oppimisympäristön tarjoamiin työskentelyvälineisiin Landon ym:n (2001) laatiman jaottelun mukaan taulukossa 6.

**Taulukko 6 WebCT-oppimisympäristön tarjoamat ja käytetyt välineet (Landon et al, 2001)**

Opiskelun välineet	Tukivälineet
<p>Web-selaimen kautta käytettävissä</p> <ul style="list-style-type: none"> <li>• saatavuus</li> <li>• kirjanmerkki-toiminto: mahdollistaa istunnon jatkamista siitä kohden mihin istunto on viimeksi lopetettu</li> <li>• multimedian sisällyttäminen materiaaliin</li> </ul> <p>Asynkroniset välineet:</p> <ul style="list-style-type: none"> <li>• sähköposti</li> <li>• BBS-tiedostojen vaihto</li> <li>• uutisryhmät</li> </ul> <p>Synkroniset välineet:</p> <ul style="list-style-type: none"> <li>• chat</li> <li>• liitutaulu (whiteboard)</li> <li>• sovellusten jakaminen: ryhmä opiskelijoita pystyy jakamaan tiedostoja ja muokkaamaan niitä yhdessä</li> <li>• virtuaalimaailma (virtuaalinen kokoushuone jne)</li> <li>• videokonferenssi</li> </ul> <p>Opiskelijan välineet</p> <ul style="list-style-type: none"> <li>• itsearviointi: tietokilpailut, testit ja arvioinnit</li> <li>• istuntojen rekisteröinti: opiskelijan yhteydenottoja tai aihekohtaisesti tietyn sivun käyttöä koskien</li> <li>• sanahaku: etsintätyökalu materiaalista tai keskusteluista tietyn kriteerin mukaan</li> <li>• kotisivu: oppilaalla mahdollisuus tehdä itsestään esittelysivu tai linkittää omat Internetissä olevat sivut WebCT:n kotisivulle</li> <li>• muistiinpanovälineet, joilla voi merkitä tietyn teoriasivun ja tulostaa sen</li> </ul>	<p>Kurssivälineet</p> <ul style="list-style-type: none"> <li>• kurssin luomisen työkalut</li> <li>• kurssin järjestämistyökalut</li> <li>• kurssin mukauttamisvälineet (omien toiveiden mukaiseksi)</li> <li>• kurssin tarkkailuvälineet: opiskelijälähtöisesti (yhteydenotot) tai aihepähtöisesti</li> </ul> <p>Luennot</p> <ul style="list-style-type: none"> <li>• ohjattu suunnittelu (sisältää malleja erilaisten sivustojen rakentamiseen)</li> <li>• esitysmuodot: sallii kaikki Web-pohjaisessa ympäristössä toimivat esitysmuodot (multimedia, kuvat, äänet, html, pdf ja muut dokumentit)</li> <li>• testausvälineet: tietokilpailut, testit ja arvioinnit</li> </ul> <p>Tiedon käsittely</p> <ul style="list-style-type: none"> <li>• automaattinen monivalintatehtävien, lyhyiden vastausten, laskutoimitusten ja yhdistämistehtävien tarkistaminen</li> <li>• tiedostojen järjestäminen käyttäjänimen mukaan</li> <li>• analysointivälineet: tilastollisten perusanalyysien suorittaminen</li> </ul> <p>Hallinto</p> <ul style="list-style-type: none"> <li>• asentaminen paikallisesti</li> <li>• oikeudet: opiskelijan, oppimisen avustajan, ohjaajan/suunnittelijan tai WebCT:n pääkäyttäjän oikeudet</li> <li>• rekisteröinti</li> <li>• salasanasuojaus</li> </ul> <p>Help-toiminnot</p> <ul style="list-style-type: none"> <li>• opiskelijan tukeminen on-line help keskustelufoorumissa ja sähköpostissa</li> <li>• ohjaajan tukeminen: täysi on-line tuki aihepiireittäin, ensi kerran käyttäjälle ohjeet, WebCT:n käyttäjien postituslista</li> </ul>

## **5. TUTKIMUSASETELMA, TUTKIMUSONGELMAT JA TUTKIMUSMENETELMÄT**

Tässä luvussa käydään aluksi läpi tutkimuksen syntyyn vaikuttaneet tekijät, sen jälkeen tutustutaan tutkimusongelmiin ja tutkimuksen tavoitteisiin sekä lopuksi kohdassa 5.3 käydään läpi millaisia menetelmiä käyttäen tutkimus on toteutettu.

### ***5.1 Tutkimusasetelma***

Tutkimukseni peruslähtökohtana on ohjelmoinnin oppimisen ongelmat virtuaalisessa opiskeluympäristössä. Kun siirrytään opettamaan ohjelmointia virtuaalisessa oppimisympäristössä, opiskelijan rooli muuttuu vastuullisemmaksi ja toisaalta siirtyminen uuteen oppimisympäristöön tuo opetuksen järjestämiselle omat haasteensa. Pelkästään perinteisen kurssin siirtäminen verkkopohjaiseen ympäristöön ei riitä virtuaalisen kurssin toteutukseksi (Cordani & Tucker, 1998).

### ***5.2 Tutkimusongelmat***

Tutkimuksen pääongelmana on löytää asiat, joita ohjelmoinnin perusteiden oppimisessa koetaan vaikeaksi. Tutkimuksen tavoitteena on löytää keinoja, joilla opetusta virtuaalisessa oppimisympäristössä voitaisiin kehittää niin, että opiskelijat selviytyisivät ohjelmoinnin perusteiden oppimisessa paremmin tulevana lukuvuosina. Saatujen palautteiden avulla pyritään kehittämään Ohjelmointi 1 –kurssia edelleen tulevia lukuvuosia varten niin, että vaikeina koettuihin ongelmakohtiin osattaisiin varautua jo kurssin suunnitteluvaiheessa.

Alaongelmana tutkimuksella on selvittää syyt, jotka johtivat opintojen keskeyttämiseen lukuvuonna 2000-2001. Tulevana lukuvuosina pyritään kurssisuunnittelulla ja kurssien toteutuksella vaikuttamaan siihen, että tyypillisimmät keskeyttämissyyt pystyttäisiin välttämään.

### *5.3 Tutkimusmenetelmä*

Tutkimukseni on empiirinen tapaustutkimus, joka on toteutettu keskeyttäneiden opiskelijoiden parissa puolistrukturoituna kyselytutkimuksena ja sähköpostin kautta toteutettuna jatkokyselynä. Toisaalta toimintatutkimuksessa Suojasen (2001) mukaan voidaan pitää keskeisenä sitä, että tutkimuksen tuloksilla pyritään kehittämään tutkimuksen kohdetta ja tutkimukseen osallistuneet henkilöt ovat mukana kehittämistyössä. Käyttämästäni tutkimusstrategiasta löytyy useita toimintatutkimuksen tunnuspiirteitä, joten tutkimusstrategiaa voidaan pitää varsin perustellusti toimintatutkimuksena. Tutkimuksen tarkoituksena ei ole ollut ainoastaan vaikeina koettujen asioiden tai opintojen keskeyttämiseen johtaneiden syiden objektiivinen tutkiminen vaan ohjelmointikurssin kehittäminen ja virtuaalisen oppimisympäristön toimintojen muuttaminen entistä paremmin oppimista tukevaksi. Tutkimukseen osallistuneet opiskelijat olivat antamiensa palautteiden kautta vaikuttamassa siihen, kuinka ohjelmointikurssia muutettiin seuraavaa lukuvuotta varten, joten toimintatutkimuksen tunnuspiirteet täyttyvät näiltä osin.

Tutkimuksessani mukana ovat Pohjois-Karjalan maakunnan lukioista seuraava lukiot: Joensuun lyseon lukio, Joensuun normaalikoulun lukio, Juuan lukio, Kiteen lukio, Kontiolahden lukio, Lieksan lukio, Liperin lukio, Niinivaaran lukio, Pyhäselän lukio sekä Outokummun lukio. Lisäksi Tohmajärven lukiosta oli opiskelijoita, jotka opiskelivat Kiteen lukion tutoropettajan ohjauksessa. Huhtikuun 2001 alussa anoin Joensuun kaupungilta tutkimuslupaa koskien Joensuun lyseon lukiota ja Niinivaaran lukiota. Maakunnan muiden lukioiden osalta soitin asianomaisen koulun rehtorille ja pyysin lupaa saada lähettää keskeyttäneille virtuaaliapprobatur-opiskelijoille kyselykaavakkeet. Koulujen rehtoreiden suhtautuminen tutkimukseen oli erittäin myönteistä ja kannustavaa.

Kyselykaavakkeiden jakaminen opiskelijoille tapahtui tutoropettajien välityksellä. Liitteenä 2 on opiskelijoille lähetetty kyselykaavake saatekirjeineen. Kyselykaavakkeita postitettiin yhteensä kymmenelle lukiolle (Tohmajärven lukio Kiteen lukion kautta). Kullekin koululle lähetettiin lomakkeita sen mukaan, kuinka WebCT:n tilastoinnin sekä yliopiston henkilökunnan näkemyksen mukaan näyttäisi olevan keskeyttäneitä (Suhonen, 2001). Lisäksi soitin kaikille tutoropettajille ja kävin heidän kanssaan läpi kunkin koulun keskeyttäneiden opiskelijoiden lukumäärät. Tutoropettajien kanssa sovittiin



puhelimitse kyselykaavakkeiden jakamisesta ja kokoamisesta koululla. Kysely kohdistettiin keskeyttäneisiin, sillä tutkimuksen päätarkoituksena oli löytää keinoja nimenomaan vaikeina koettujen ohjelmoinnin osa-alueiden opettamisen kehittämiseen.

Varsinaisen kyselyn lisäksi tein jatkokyselyn, jolla halusin selvittää erityisesti mitä asioita olisi pitänyt toteuttaa toisin ja missä puolestaan oli onnistuttu. Halukkuutta jatkokyselyyn kysyin kyselykaavakkeen viimeisessä kysymyksessä. Ennen sähköpostikyselyn lähettämistä kävin läpi kaikki kouluilta palautuneet kyselykaavakkeet ja kokosin vastauksista tuntemuksia yleisellä tasolla. Sähköpostikyselyssä keskityin pääosin löytämään yleisesti hankalina koettuihin asioihin konkreettisia parannusehdotuksia.

Kyselyn lisäksi tutkin WebCT-oppimisympäristön rekisteröimiä tietoja opiskelijoiden palauttamista harjoitustehtävistä. Luin läpi keskustelufoorumilla opiskelijoiden käymät keskustelut, jotka osittain sisälsivät tutkimukseen kuuluvia asioita ja toisaalta antoivat minulle hyvän yleiskuvan siitä, kuinka tyytyväisiä opiskelijat olivat yleensäkin virtuaaliapprobatur-kurssien järjestelyihin. Tutustuin lisäksi opiskelijoille jaettuun itseopiskeluoppaaseen sekä tutoreille jaettuun tutoroppaaseen ja kävin läpi ohjelmoinnin varsinaisen tentin tenttivastaukset.

Joensuun yliopiston assistentti Jarkko Suhoselta sain hyvin paljon arvokasta tietoa liittyen käytännön järjestelyihin ja niiden onnistumiseen opetuksen järjestäjän näkökulmasta katsottuna. Assistentti Suhonen on ollut alusta alkaen mukana luomassa virtuaaliapprobaturia ja hänen vastuullaan on virtuaaliapprobaturin käytännön järjestelyt. Tarkoitukseni oli myös tutkia WebCT:n rekisteröimiä tietoja siitä, kuinka opiskelijat olivat käyttäneet hyväkseen teoriasivustoja. WebCT kokoaa taustarekisteriä, jonne kirjautuu muun muassa yhteydenottojen lukumäärä sekä kuinka paljon mitäkin sivua on käytetty. Saamani tiedon mukaan (Suhonen, 2001) kyseinen automaattinen rekisteröinti ei kuitenkaan valitettavasti ole osoittautunut kovin luotettavaksi, joten en katsonut mielekkääksi lähteä analysoimaan ja arvioimaan opiskelijoiden käyttäytymistä näiden rekisteritietojen pohjalta.

## 6. TULOSTEN ESITTELY

Tässä luvussa käydään läpi tutkimuksen tulokset. Aluksi analysoidaan kyselykaavakkeesta saatuja vastauksia kohdissa 6.1-6.4. Tämän jälkeen tarkastellaan opiskelijoiden harjoitustehtävien tekemistä sekä tenttituloksia kohdissa 6.5-6.6. Lopuksi analysoidaan kohdissa 6.7-6.8 millaisia tukitoimia keskeyttäneet opiskelijat kaipaivat kyselyn ja jatkokyselyn perusteella.

### 6.1 Taustatietoja

Kyselyyn vastasi 25 opiskelijaa, joista 12 ensimmäisen, 11 toisen ja kaksi kolmannen vuoden lukio-opiskelijaa. Poikia vastaajista oli 17, tyttöjä 7 ja yksi vastasi nimettömänä. Taustatietoina kysyin opiskelijoiden aiempaa tietoteknistä taitoa sekä ohjelmointikokemusta. Suurin osa vastaajista hallitsi tietotekniikan perustaidot kuten tekstinkäsittelyn ja kotisivujen tekemisen html-kielellä. Tekstinkäsittelyä ei ole erityisesti taidokseen maininnut kuin kaksi vastaajaa, mutta uskon tekstinkäsittelytaidon kuitenkin olevan hallussa huomattavasti useammalla, sillä kysymyksellä haettiin vastausta lähinnä aiempaan ohjelmointikokemukseen. Sen sijaan niillä opiskelijoilla, jotka mainitsivat taidoiksi esimerkiksi tekstinkäsittelyn, ei ollut ohjelmoinnista minkäänlaista aiempaa kokemusta.

Ohjelmointikokemusta löytyi lähinnä seuraavista kielistä: html, Pascal, Turbo-Pascal, Visual Basic, GW-Basic ja Q-Basic. Ohjelmointikielten osaaminen nähdään taulukosta 7. Yhteensä viidellätoista vastaajista oli ainakin jonkinlaista ohjelmointikokemusta jo ennen virtuaaliapprobatur-opintojen aloittamista, neljällä oli kokemusta olio-ohjelmoinnista Visual Basicillä tai C++:lla. Ainoastaan yhdellä vastaajista oli kokemusta kahdesta ohjelmointikielestä (Perl ja C++).

**Taulukko 7 Kokemus ohjelmointikielistä**

Ohjelmointikieli	Lkm
C++	3
Perl	1
Pascal	9
Basic	3
Yhteensä	16

Vastauksista oli kuitenkin pääteltävissä, että useimmilla opiskelijoilla kyseessä on vain yhden tai kahden kurssin mittainen kokemus joko yläasteelta tai lukiosta, joten varsinaisesta aiemmasta ohjelmointikokemuksesta ei juurikaan voida keskeyttäneiden osalta puhua. Kymmenellä kyselyyn vastanneesta ei ollut minkäänlaista ohjelmointikokemusta ennen virtuaaliapprobatur-opintojen aloittamista.

Kyselytutkimuksessani kysymyksestä 2 (Arvioi seuraavia väittämiä käyttäen asteikkoa 1-5, Oppimisympäristön tekninen hallinta) osoittautui, että suurin osa vastanneista (56 %) koki WebCT:n helppokäyttöiseksi ja rakenteeltaan selkeäksi. Oppimisympäristöä koskevissa kysymyksissä ainoastaan harjoitustehtävien palauttaminen koettiin työläänä ja osalla opiskelijoista oli vaikeuksia saada Java-kääntäjä ja -tulkki asennettua. Mitään muuta selkeää teknistä ongelmakohtaa ei vastauksista noussut esille.

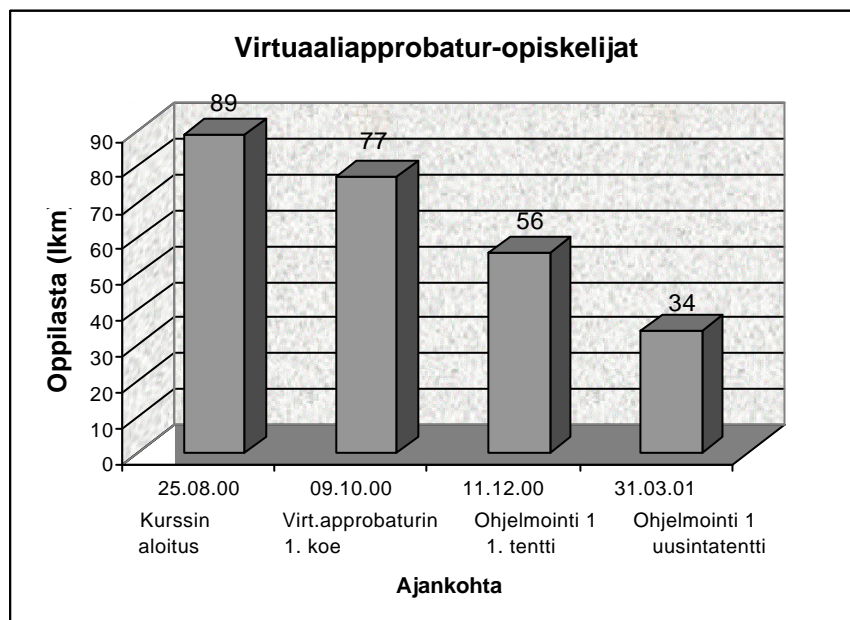
Vastaavasti myös kysymyksessä 1 (Arvioi seuraavia väittämiä käyttäen asteikkoa 1-5, Ohjelmointi 1 – kurssimateriaali ja sen käytettävyys) ei noussut esille mitään selkeitä puutteita vaan vastausten keskiarvot sijoittuivat varsin neutraalisti 2,5 – 2,8 välille. Avoimissa kysymyksissä 8 ja 9 kaivattiin lisäesimerkkejä verkkomateriaaliin.

## ***6.2 Keskeyttäneiden lukumäärä***

Keskeyttäneiden määrän arvioiminen näytti alkuun olevan varsin ongelmallista. Virtuaaliapprobaturissa ilmoittaudutaan ainoastaan opintojen alussa koko 15 opintoviikon laajuisen kokonaisuuden suorittamiseen ja yksittäiselle kurssille ei erikseen tarvitse ilmoittautua. Tämän johdosta määrittelin kriteerit, joiden mukaan tarkastelen keskeyttäneitä. Tässä tutkimuksessa opiskelija katsotaan keskeyttäneeksi, mikäli hän ei ole palauttanut kursseihin liittyviä harjoitustehtäviä pitkään aikaan eikä ole myöskään osallistunut aprobatur-opintojen suorittamiseen osallistumalla tenttiin tai palauttamalla harjoitustöitä. Opiskelija tulkitaan lukuvuonna 2000-2001 keskeyttäneeksi vaikka hänellä olisi tarkoituksena osallistua seuraavana lukuvuonna uuden ryhmän mukana aprobatur-opintoihin.

Keskeyttäneiden lukumäärää lähdin arvioimaan muutaman selkeän tarkasteluajankohdan mukaan. Tarkasteluajankohdiksi otin virtuaaliapprobaturin aloitustilaisuutta edeltävän päivän 25.8.2000, johon mennessä opiskelijoiden tuli ilmoittautua. Toisena tarkastelukohtana on ensimmäinen tenttipäivä 9.10.2000 (Tietokone, käyttöjärjestelmä ja tietoverkot –kurssin tentti) sekä kolmantena Ohjelmointi 1 –kurssin varsinainen tenttipäivä 11.12.2000. Lisäksi halusin ottaa tarkasteluun mukaan kevätlukukaudelta yhden kontrollipäivän 31.3.2001, jotta näkisin onko keskeyttämisen suunta säilynyt myös kevään ohjelmointikurssin aikana samana kuin syksyn ohjelmointikursseilla.

Tarkastellaan seuraavaksi mukana olevien opiskelijoiden lukumääriä tarkasteluajankohdittain (Kuva 6). Virtuaaliopintoja aloitettaessa mukaan ilmoittautui kaiken kaikkiaan 89 opiskelijaa. Lukumäärä on koottu yliopiston palvelimelle tarvittavista käyttäjätunnusten hakulomakkeista. Ensimmäiseen virtuaaliapprobatur-tenttiin 9.10.2000 osallistui 77 opiskelijaa. Ohjelmointi 1 - kurssin varsinaiseen tenttiin 11.12.2000 osallistui 54 opiskelijaa sekä tammikuun uusinnassa lisäksi kaksi sellaista opiskelijaa, jotka eivät joulukuun tenttiin osallistuneet, joten tarkastelupäivän 11.12.2000 mukana olevien määräksi sain näin 56 opiskelijaa.

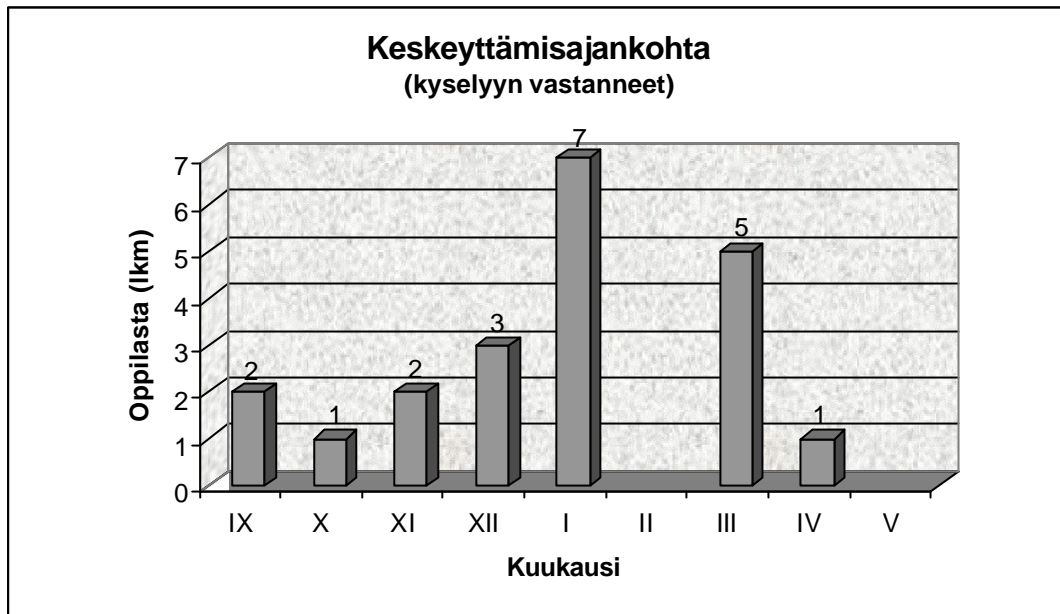


**Kuva 6 Virtuaaliapprobatur-opiskelijat lukuvuonna 2000-2001**

Kevätpuolella tarkasteluajankohdaksi otin päivämäärän 31.3.2001. Viimeisen tarkasteluajankohdan halusin mahdollisimman lähelle kyselyn lähettämistä, jotta keskeyttäneistä mahdollisimman moni olisi mukana tutkimuksessa. Kevään tarkasteluajankohtana mukana olevien opiskelijoiden lukumäärä on ainoa, joka perustuu arvioon toisaalta assistentti Jarkko Suhosen näkemykseen keskeyttäneiden lukumäärästä (Suhonen, 2001) ja toisaalta oman tutkimukseeni harjoitusten palauttamistilastojen ja tutoropettajien kanssa käymieni keskustelujen pohjalta. Kuten kuvasta 6 näkyy mukana olevien opiskelijoiden määrä on tasaisesti pudonnut, kuitenkin niin, että pääosa keskeyttämisistä sijoittuu Ohjelmointi 1 –kurssin ajalle syyskuusta 2000 – joulukuulle 2000.

Aloittaneesta 89 opiskelijasta kymmenen ei osallistunut yhteenkään tenttiin eikä myöskään palauttanut ainoitakaan harjoituksia ja kaksi opiskelijaa teki aluksi muutamalla viikolla harjoitustehtäviä, mutta eivät kuitenkaan lopulta osallistuneet yhteenkään tenttiin. Ensimmäisen virtuaaliapprobatur-tentin jälkeen opintonsa keskeytti 21 opiskelijaa ja Ohjelmointi 1 -kurssin tentin jälkeen opinnot on keskeyttänyt 22 opiskelijaa. Ohjelmointi 1 -kurssi alkoi yhtä aikaa Tietokone, käyttöjärjestelmä ja tietoverkot –kurssin kanssa syyskuun 2000 alusta ja jatkui joulukuulle 2000 saakka. Kevään seurantapäivään 31.3.2001 memessä on opintonsa keskeyttänyt yhteensä 55 opiskelijaa eli 62 % aloittaneista.

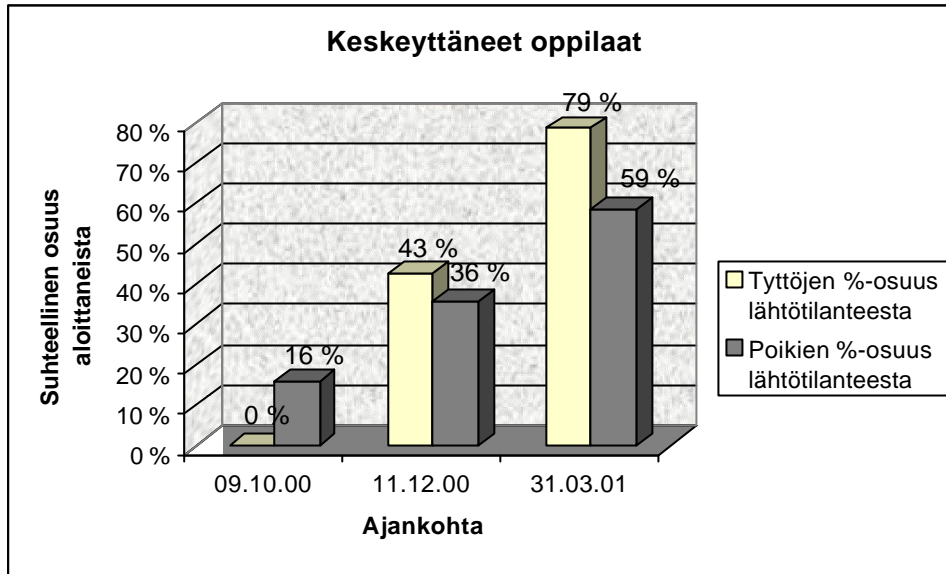
Kyselyyn vastanneiden opiskelijoiden keskeyttämisajankohdat näkyvät kuvassa 7. Yli puolet vastanneista (60 %) keskeytti viimeistään Ohjelmointi 1 -kurssin uusintatentin jälkeen tammikuussa 2001 ja 24 % kyselyyn vastanneista keskeytti kevätlukukaudella Ohjelmointi 2 -kurssin aikana. Kyselyyn vastanneista 25 opiskelijasta neljä ei vastannut tähän kysymykseen. Suurin osa vastaajista on keskeyttänyt Ohjelmointi 1 –kurssin aikana tai viimeistään siinä vaiheessa kun ei saanut Ohjelmointi 1 -kurssin uusintatenttiä läpi.



**Kuva 7 Kyselyyn vastanneiden keskeyttämisaikakohtat**

Ennen ensimmäistä tenttiä kaksitoista poikaa jättäytyi pois opinnoista, sen sijaan kaikki tytöt suorittivat vähintään Tietokone, käyttöjärjestelmä ja tietoverkot -kurssin. Aloittaneita tyttöjä on lukumäärältään huomattavasti vähemmän kuin aloittaneita poikia, joten tarkastellaan seuraavaksi keskeyttäneiden tyttöjen ja poikien osuutta suhteessa aloittaneiden tyttöjen ja poikien lukumääriin (Kuva 8).

Ensimmäisen tentin jälkeen tytöistä kukaan ei ole vielä keskeyttänyt kun taas pojista 16 % on keskeyttänyt jo tuossa vaiheessa. Joulukuussa Ohjelmointi 1 -kurssin tentin jälkeen tytöistä 43 % on keskeyttänyt ja poikienkin osuus lähtötilanteesta on kasvanut 36 %:iin. Tässä vaiheessa keskeyttäneiden tyttöjen osuus on ollut yllättävän voimakasta. Kaikki tytöt suorittivat ensimmäisen kurssin, joka koskee tietotekniikan perusteita. Opetuksen muuttuessa ohjelmointipainotteiseksi on tyttöjen keskeyttämisherkkyys kasvanut.

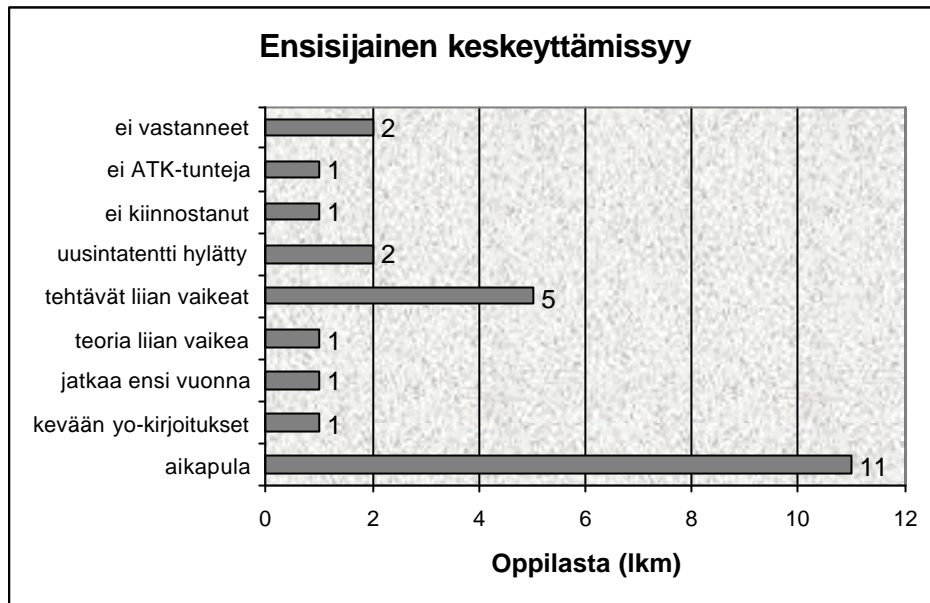


**Kuva 8 Keskeyttäneiden tyttöjen ja poikien suhteellinen osuus lähtötilanteesta**

Kevätlukukaudelle siirryttäessä keskeyttäneiden tyttöjen osuus näyttää kasvavan yhä vielä poikia voimakkaammin sillä tytöistä jo 79 % ja pojista 59 % on luopunut virtuaaliapprobatur-opinnoista. Kevätlukukaudella erityisesti Ohjelmointi 2 –kurssi on tuottanut ongelmia suoriutumisen suhteen.

### **6.3 Keskeyttämisen syyt**

Kyselykaavakkeessa kysyin opintojen pääasiallista keskeyttämissyitä kaavakkeen loppupuolella kysymyksellä 7. Tähän kysymykseen vastasi 23 opiskelijaa ja kaksi jätti vastaamatta. Opiskelijoilla oli mahdollisuus numeroida useampiakin keskeyttämissyitä numeroimalla järjestykseen 1 = tärkein syy, 2 = toiseksi tärkein jne. Katsotaan seuraavassa ensin tärkeimmäksi keskeyttämissyyksi mainitut syyt (kuva 9) ja sen jälkeen toiseksi tärkeimmäksi mainitut syyt (kuva 10). Opiskelijoille annettiin joukko valmiita vastausvaihtoehtoja sekä avoin kohta, jossa opiskelija voi tarvittaessa nimetä muunkin syyn. Kuvasta 9 näkyy annetuista vastausvaihtoehdoista ne, joihin tuli vähintään yksi vastaus.

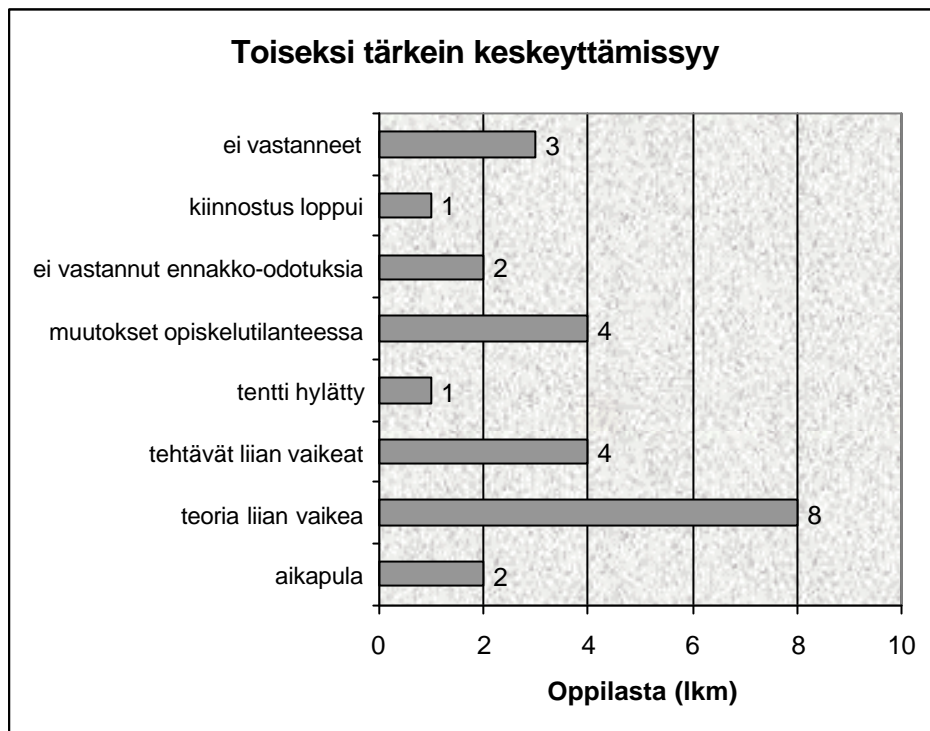


**Kuva 9 Pääsyy keskeyttämiselle (n=25)**

Kyselyyn vastanneista 11 (44 %) nimeää aikapulan pääsyyksi keskeyttämiselleen. Viisi vastaajaa (20 %) koki tehtävät liian vaikeina ja kaksi vastaajaa (8 %) keskeytti, koska ei päässyt Ohjelmointi 1 -kurssin uusintatenttiä läpi. Seuraavat keskeyttämissyyt olivat yksittäisiä: osallistuminen kevään yo-kirjoituksiin, aikoo jatkaa ensi lukuvuonna, teorian vaikeus, kurssi ei kiinnostanut sekä opiskelijalla ei ollut Internet-yhteyttä eikä ATK-tunteja (Kuva 9).

Toiseksi tärkeimpänä keskeyttämissyyinä kahdeksan vastaajaa (32 %) kertoo teorian olleen liian vaikeaa, neljä vastaajaa (16 %) pitää tehtäviä liian vaikeina ja samoin neljälle vastaajalle tuli opiskelutilanteeseen muutoksia niin, että päättikin keskeyttää virtuaaliapprobatur-opinnot (Kuva 10). Edelleen vastaajista kaksi (8 %) nimeää aikapulan toiseksi tärkeimmäksi keskeyttämissyyksi. Samoin kaksi vastaajaa koki, että Ohjelmointi 1 -kurssi ei vastannut heidän ennakko-odotuksiaan.





**Kuva 10 Toiseksi tärkein keskeyttämissyy (n=25)**

Kyselyyn vastanneista syyslukukaudella keskeyttäneitä oli 71 % ja kevätlukukaudella keskeyttäneitä 29 %. Vastauksissa syyslukukaudella Ohjelmointi 1 -kurssin aikana keskeyttäneiden ja kevätlukukaudella Ohjelmointi 2 -kurssin aikana keskeyttäneiden osalta ei esille noussut merkittävää eroa, sillä molemmilla ryhmillä pääasiallisesti keskeyttämissyyksi nousi aikapula.

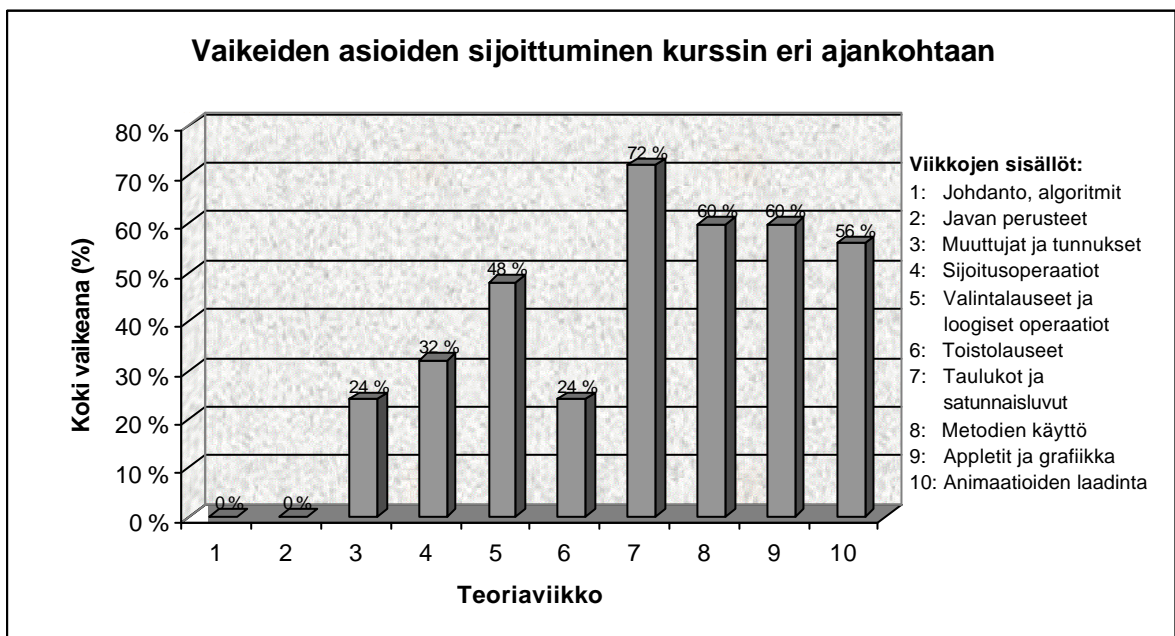
#### **6.4 Vaikeina koetut ohjelmoinnin osa-alueet**

Kysymyksellä 4 (Arvioi seuraavien ohjelmoinnin osa-alueiden vaikeutta) halusin kartoittaa ohjelmoinnin eri osa-alueiden hankaluutta opiskelijan kannalta. Opiskelijat arvioivat asteikolla 1-5 (1= helppo, 2 = melko helppo, 3 = en osaa sanoa, 4 = melko vaikea, 5 = vaikea) seuraavia ohjelmoinnin osa-alueita: muuttujien ja tunnusten käyttö, sijoitusoperaatioiden sekä syöttö- ja tulostuslauseiden käyttö, toistolauseiden käyttö, valintalauseiden ja loogisten operaatioiden käyttö, taulukoiden ja satunnaislukujen käyttö, metodien käyttö, sovelmien (applettien) ja grafiikan käyttö sekä animaatioiden käyttö.

**Taulukko 8 Vaikeana tai melko vaikeana koetut ohjelmoinnin osa-alueet (n=25)**

Vaikeina tai melko vaikeina koetut osa-alueet	%
Taulukot ja satunnaisluvut	72 %
Metodien käyttö	60 %
Sovelmien ja grafiikan käyttö	60 %
Animaatioiden laadinta	56 %
Valintalauseet ja loogiset operaatiot	48 %
Sijoitus-, syöttö- ja tulostuslauseet	32 %
Toistolauseet	24 %
Muuttujat ja tunnukset	24 %

Yllä nähdään taulukosta 8, mitkä ohjelmoinnin osa-alueet koettiin vaikeaksi tai melko vaikeaksi (kysymyksen vastausvaihtoehdot 4 ja 5). Vastaajista 72 % koki taulukoiden ja satunnaislukujen käytön hankalana. Sovelmien ja grafiikan sekä metodien käytön koki 60 % vastaajista vaikeina. Yli puolet (56 %) vastaajista piti animaatioiden tekemistä vaikeana ja lähes puolet (48 %) koki valintalauseet ja loogiset operaatiot hankalina oppia. Myöskään näissä vastauksissa ei syyslukukaudella keskeyttäneiden ja kevätlukukaudella keskeyttäneiden osalta ollut juurikaan eroa.



**Kuva 11 Vaikeina koettujen asioiden sijoittuminen kurssin eri ajankohtaan (n=25)**

Kuvasta 11 nähdään, että vaikeina koetut asiat ovat sijoittuneet kurssiaikataulussa kurssin puolesta välistä kurssin loppuun ja alussa on ollut useita helppoja teoriaviikkoja. Viikosta 7 lähtien ei ole enää ollut yhtään helpompaa viikkoa, vaan kaikki kurssin loppupuolella käsitellyt asiat on koettu vaikeina asioina.

Mielenkiintoista on havaita, että opiskelijat ovat kokeneet asioiden tasaisesti vaikeutuvan kurssin edetessä, mutta yllättäen viikolla 6 esitellyt toistolauseet onkin koettu helppoina. Toistolauseiden yhteydessä yleensä käytetään jo aiemmin kurssin aikana tutuksi tulleita valintalauseita ja boolean-lausekkeita, mikä osaltaan selittää sitä, että opiskelijoiden oli helpompi omaksua uutena asiana esitelty toistorakenne. Toisaalta myöhemmin tenttitulosten tarkastelun yhteydessä kohdassa 6.6 havaitaan, että toistolauseita ei kuitenkaan osattu käyttää tentissä oikein. Opiskelijoiden on mitä ilmeisemmin ollut helppo ymmärtää toistolauseiden toimintaperiaate ja kokevat ne siksi helppoina, mutta näyttää siltä, että käytännössä toistolauseet ovat olleet opiskelijoille kuitenkin vaikeampia kuin mitä kuvan 11 perusteella voisi ymmärtää.

### 6.5 Harjoitustehtävien tekeminen

Joensuun yliopiston matemaattis-luonnontieteellisessä tiedekunnassa noudatetaan harjoitustehtävien tekemisessä ns. *1/3-sääntöä*, jonka mukaan harjoitustehtävistä tulee olla vähintään 1/3 suoritettuna, jotta kurssi voidaan hyväksyä suoritetuksi. Tätä sääntöä noudatetaan myös virtuaaliapprobaturissa. Tarkastellaan seuraavaksi, kuinka opiskelijat palauttivat harjoitustehtäviä Ohjelmointi 1 -kurssin aikana (taulukko 9). Harjoitustehtäviä oli kymmenelle viikolle, kaiken kaikkiaan 50 tehtävää, jotka jakautuivat niin, että kullakin viikolla oli 4-7 tehtävää. Opiskelijoiden tuli suorittaa tehtävistä vähintään 17.

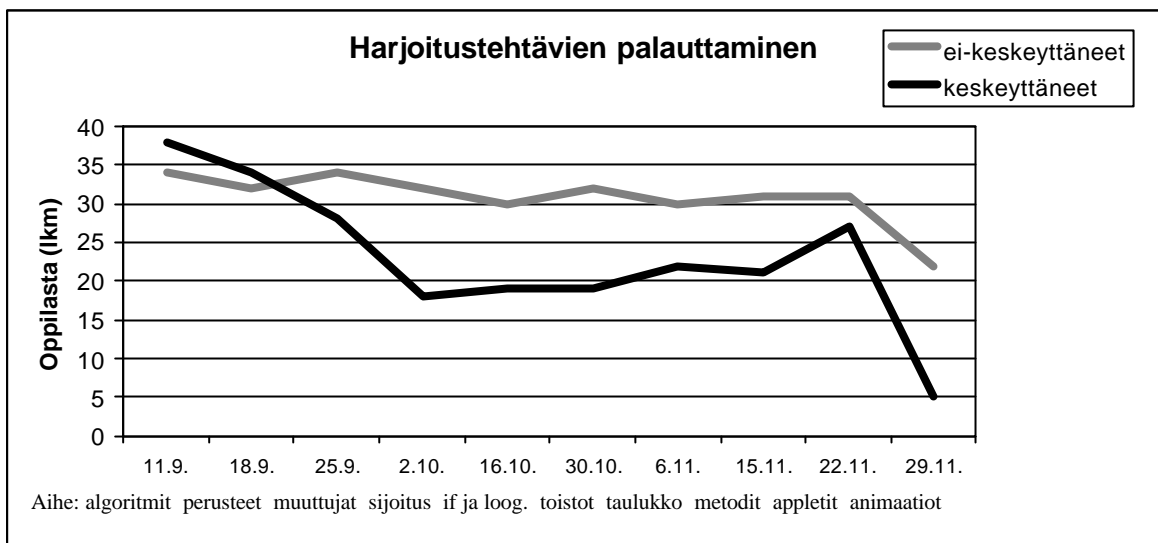
**Taulukko 9 Harjoitustehtävien palauttaminen**

Harjoitustehtävät	kaikki	keskeyttäneet	ei-keskeyttäneet
Jätti palauttamatta korkeintaan kahdet harjoitukset	47 %	16 %	88 %
Lopetti kun 1/3 täyttyi	32 %	47 %	12 %
Ei saanut 1/3 täyteen	22 %	38 %	0 %

Ensimmäiselle riville taulukossa 9 on koottu ne opiskelijat, jotka palauttivat lähes jokaisella viikolla harjoitustehtävät. Nämä opiskelijat eivät siis välttämättä ole tehneet jokaisella viikolla kaikkia harjoitustehtäviä, mutta tekivät kuitenkin vähintään kahdeksalla viikolla ainakin osan tehtävistä. Kaikista opiskelijoista 47 % ja keskeyttäneistä 16 % palautti vähintään kahdeksalla viikolla harjoitustehtävät. Tästä voidaan päätellä, että alle puolet ohjelmointikurssin aloittaneista opiskelijoista oli aktiivisesti mukana läpi koko kurssin. Jos tarkastellaan tilannetta ei-keskeyttäneiden osalta, niin peräti 88 % heistä palautti vähintään kahdeksalla viikolla harjoitustehtävät.

Keskeyttäneistä lähes puolet (47 %) ja ei-keskeyttäneistä 12 % lopetti harjoitustehtävien palauttamisen siinä vaiheessa kun vaadittu 1/3 tehtävistä tuli täyteen. Ei-keskeyttäneistä kaikki opiskelijat saivat tehtyä vähintään 1/3 tehtävistä, keskeyttäneistä puolestaan 38 % ei saanut vähimmäismäärää eli vähintään 17 tehtävää suorit etuksi.

Kun verrataan, kuinka keskeyttäneet ja ei-keskeyttäneet opiskelijat palauttivat harjoitustehtäviä ohjelmointi 1 -kurssin aikana tarkasteltuna tehtyjen tehtävien lukumääriä (kuva 12), niin ei-keskeyttäneiden opiskelijoiden harjoitustehtävien palauttaminen näyttää olevan tasaista läpi koko kurssin ajan. Ainoastaan kurssin lopussa tulee pieni väsyminen. Keskeyttäneiden osalta sen sijaan löytyy kolme selkeää muutosta: alussa opiskelijat ovat aloittaneet innokkaasti tehdä tehtäviä, mutta heti ensimmäisten harjoitustehtävien jälkeen innostus on lähtenyt laskemaan hyvinkin voimakkaasti. Muutamaa viikkoa ennen kurssin loppumista keskeyttäneetkin opiskelijat ovat jälleen palauttaneet hieman aktiivisemmin harjoitustehtäviä. Kahdella viimeisellä harjoituskerralla jälleen tulee selvä pudotus palautettujen tehtävien määrässä.



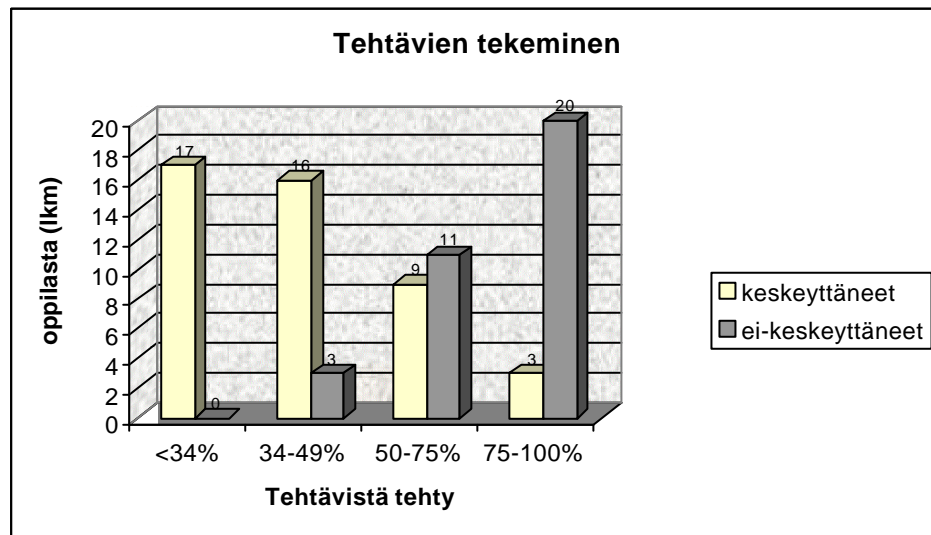
**Kuva 12 Harjoitustehtävien palauttaminen keskeyttäneiden ja ei-keskeyttäneiden osalta**

Edellä kuvassa 12 esitettyjen tietojen pohjalta voidaan arvioida opiskelijoiden viikoittaista työskentelytapaa kurssin suorittamisessa. Ei-keskeyttäneiden ja keskeyttäneiden opiskelijoiden tehtävien palautuksissa on selkeitä eroavuuksia. Ei-keskeyttäneet ovat tehneet tehtäviä tasaisesti läpi kurssin, ainoastaan lopussa tulee pieni pudotus palautettujen tehtävien lukumäärässä. Näillä opiskelijoilla oli tuossa vaiheessa jo vaaditut 1/3 tehtäviä suoritettuna, joten koulun koeviikon alkaessa ja ohjelmointitentin lähestyessä he pystyivät vähentämään tehtävien tekemistä. Näyttäisi siltä, että keskeyttäneet opiskelijat eivät puolestaan ole aikapulan, tehtävien vaikeuden tai oman kiinnostuneisuuden vuoksi saaneet harjoitustehtäviä suoritettua viikoittain. Heti kurssin alussa keskeyttäneetkin opiskelijat palauttivat ensimmäiset harjoitustehtävät hyvin.

Etukäteen oletin, että keskeyttäneiden harjoitustehtävien palautusta kuvaava käyrä korreloisi vaikeina koettuja asioita (ks. Kuva 11 sivulla 46), mutta näin ei näyttäisi olevan vaan nimenomaan alussa helppojen viikkojen aikana on harjoitustehtäviä palautettu yllättäen vähemmän ja kurssin keskivaiheella vaikeina koettujen asioiden aikana palauttaminen on tasaantunut ja pikemminkin hieman parantunut. Yksi syy voi olla, että opiskelijat kokivat alkupuolen algoritmit tehtävät ja Javan perusteita koskevat tehtävät niin helppoina, etteivät tunteneet tarvetta näiden tehtävien tekemiseen vaan tiiviin aikataulun vuoksi keskittyivät muihin opintoihin sen sijaan. Asioiden vaikeutuessa jouduttiin yllättäen sitten tilanteeseen, että tehtävät olivat vaikeampia ja toisaalta tehtävien tekemiseen tarvittavaa rutiinia ei alkupuolen tehtävien

aikana ollut ehtinyt muodostua. Tämä on saattanut johtaa siihen, että opiskelijat putosivat tasaisesta opiskelurytmistä pois ja pääsy uudelleen harjoitustehtävien rytmiin olikin vaikeaa. Toisaalta loppupuolella metodien ja sovelmien kohdalla oleva pieni piristymisen antaisi ymmärtää, että opiskelijat ovat yrittäneet saada vaaditun 1/3 tehtävistä suoritettua, jotta pääsisivät osallistumaan tenttiin. Kuvassa 12 näkyvä viimeinen pudotus korreloi kyllä vaikeana koetun animaation kanssa. Viimeisten (29.11.) palautettujen harjoitustehtävien määrää selittää myös se, että sellaiset opiskelijat, jotka tiesivät etteivät kuitenkaan saa kolmasosaa tehtävistä suoritettua, luovuttivat.

Kuvassa 13 tarkastellaan tehtävien tekemistä jaettuina neljään tarkasteluryhmään: ensimmäisen ryhmän muodostavat ne opiskelijat, jotka tekivät alle kolmasosan tehtävistä ja joilla ei siis tämän vuoksi ollut tenttioikeutta Ohjelmointi 1 -kurssin tenttiin. Toisena ryhmänä ovat vähintään kolmasosan mutta alle puolet tehtävistä tehneet ja kolmantena vähintään puolet, mutta alle ¾ tehtävistä tehneet sekä viimeisenä ryhmänä vähintään ¾ kaikista tehtävistä tehneet opiskelijat. Myös tässä tarkastellaan tilannetta keskeyttäneiden ja ei-keskeyttäneiden opiskelijoiden kohdalla erikseen.



**Kuva 13 Harjoitustehtävien tekeminen (n=79)**

Kuten kuvasta 13 nähdään, keskeyttäneiden opiskelijoiden osalta harjoitustehtävien tekemisen painopiste on selvästi vähemmän tehtyjen tehtävien puolella: suurin osa keskeyttäneistä sai alle 50 %

tehtävistä tehtyä. Osaltaan tätä ilmiötä kasvattaa se, että keskeyttämispäätökseen kypsyneet opiskelijat luonnollisesti myös lopettivat jossakin vaiheessa harjoitustehtävien tekemisen ja se kasvattaa alle kolmasosan tehneiden määrää. Toisaalta mitään selkeää rajaa harjoitustehtävien tekemisen lopettamiseen ei löytynyt, vaan opiskelijat ovat saattaneet jättää usealla viikolla tehtävät tekemättä ja sitten jälleen palauttavatkin tehtäviä. Ei-keskeyttäneistä opiskelijoista suurin osa on tehnyt vähintään  $\frac{3}{4}$  tehtävistä ja jopa 91 % teki vähintään puolet tehtävistä (50-100 %). Keskeyttäneistä 73 % sai tehtyä korkeintaan puolet (<49 %) harjoitustehtävistä. Tekemällä oppiminen on Candle-mallissa peruslähtökohta ja seuraavassa luvussa tenttivastausten analysoinnin yhteydessä nähdään kuinka selvästi tehtävien tekemisestä on ollut hyötyä hyvän tenttimenestyksen saavuttamiseksi.

## 6.6 Tettiin osallistuminen

Ohjelmointi 1 –kurssin tenttiin osallistui kaiken kaikkiaan 56 opiskelijaa: 54 opiskelijaa varsinaisessa tentissä joulukuussa 2000 ja kaksi opiskelijaa uusintatentissä tammikuussa 2001. Seuraavassa tarkastellaan tilannetta kaikkien keskeyttäneiden osalta taulukossa 10.

**Taulukko 10 Keskeyttäneiden osallistuminen Ohjelmointi 1 -kurssin tentteihin**

Menestyminen Ohjelmointi 1 -kurssin tenteissä (n=56)		
	Osallistui	Hylättiin
Varsinainen tentti	20	11
Uusintatentti	12	6
<b>Kävi ainoastaan uusintatentissä</b>		
Joulukuussa hylätty, uusintatentissä suoritus		2
Joulukuussa hylätty, uusintatentissä suoritus		4
Uusintatentillä korotti joulukuussa hyväksytyä suoritusta		1
Ei osallistunut kumpaankaan tenttiin		33

Keskeyttäneistä opiskelijoista tenttiin osallistui 20 opiskelijaa, joista 11 hylättiin. Kuusi opiskelijaa sai tammikuun uusintatentissä hyväksytyyn suorituksen. Kaksi tammikuun uusintatenttiin osallistunutta opiskelijaa ei osallistunut ollenkaan varsinaiseen tenttiin joulukuussa ja yksi opiskelija korotti joulukuussa hyväksyttynä saamaansa arvosanaa. Näin joulukuussa hylätyistä 11 opiskelijasta kymmenen yritti tammikuussa uudelleen ja näistä kymmenestä viisi hylättiin vielä uusimassakin.

Taulukossa 11 on esitelty Ohjelmointi 1 –kurssin tenttitulokset joulukuun tentin osalta. Taulukossa on esitelty tenttikysymykset, maksimipisteet sekä vastausten keskiarvot niin keskeyttäneiden kuin ei-keskeyttäneidenkin osalta. Viimeiseen sarakkeeseen (Suhteellinen ero %) on laskettu keskeyttäneiden tenttivastausten pistemäärien suhteellinen osuus ei-keskeyttäneiden pistemääristä. Ohjelmointi 1 -kurssin joulukuussa 2000 pidetty tentti on liitteenä 3.

Kun tarkastellaan keskeyttäneiden opiskelijoiden tenttivastauksia verrattuna ei-keskeyttäneiden tenttivastauksiin, niin samat asiat näyttävät nousevan esille vaikeina asioina kuin kyselytutkimuksen kysymyksessä 4, jossa opiskelijat arvioivat ohjelmoinnin eri osa-alueiden vaikeutta. Tentissä oli neljä kysymystä, joista jokaisesta sai korkeintaan kuusi pistettä. Ensimmäinen kysymys koski if-lauseen merkityksen selvittämistä. Kyselytutkimuksen vastauksista ilmeni, että lähes puolet vastaajista (48 %) koki valintalauseiden käytön vaikeana tai melko vaikeana. Tenttivastauksista osa keskeyttäneiden opiskelijoiden vastauksista oli varsin hyviä viiden tai viiden ja puolen pisteen vastauksia, mutta keskivertaisiakin kolmen pisteen vastauksia löytyi seitsemän kappaletta. Ensimmäisen tenttivastauksen vastausten keskiarvo on suhteellisen hyvä 4,1 maksimin ollessa kuusi pistettä (taulukko 11). Ei-keskeyttäneistä kaikki olivat saaneet vähintään 4 pistettä ja keskiarvoksi tulikin 5.

**Taulukko 11 Ohjelmointi 1 -kurssin varsinaisen tentin tulokset (n = 54)**

Kysymys	Maksimi pisteet	Keskiarvo ei-keskeyttäneet	Keskiarvo Keskeyttäneet	Suhteellinen ero %
1: if-lause	6	5,0	4,1	81 %
2: toisto- ja valintalause	6	4,7	2,9	62 %
3: taulukko ja satunnaisluvut	6	4,1	1,9	46 %
4: sovelman tekeminen	6	3,7	0,5	12 %
Yhteensä	24	12,0	9,3	78 %

Toisessa kysymyksessä opiskelijoiden tuli laatia Java-sovellus. Ohjelman tulee kysyä 10 lukua, laskea niiden summa sekä ilmoittaa luvuista pienin. Keskeyttäneistä tenttiin osallistuneista 13 ei osannut käyttää hyväksi toistolauseita lukujen kysymisessä, muutamasta vastauksesta puuttui muuttujien esittelyt ja yksi vastaaja ymmärsi, että numerot tulisi kysyä silmukassa, mutta ei ollut osannut sitä toteuttaa. Neljä vastaajaa käytti toistolauseita enemmän tai vähemmän onnistuneesti ja kolmella toistolause oli rakennettu



joko täysin oikein tai ainakin lähes oikein. Toisen kysymyksen pisteet jäivät melko alhaisiksi keskeyttäneiden osalta: 2,9 pistettä kuudesta pisteestä. Ohjelmoinnin osa-alueista opiskelijat kokivat toistolauseet suhteellisen helppoina kuten Kuva 11 sivulla 46 osoittaa. Tenttitulos viittaa kuitenkin siihen, että toistolause on ollut vaikeampi kuin opiskelijat ovat itse arvioineet.

Kolmannen ja neljännen kysymyksen vastausten pistemäärät vahvistavat kyselyssä esille tulleita tuloksia. Vastaajista 72 % koki taulukoiden ja satunnaislukujen käytön hankalana ja tentissä tehtävässä 3 tuli käyttää hyväksi näitä molempia. Viisi keskeyttänyttä ei ollut edes yrittänyt tehdä tehtävää ja loppujenkin osalta vastaukset olivat varsin vaatimattomia. Kolmannen tehtävän vastausten keskiarvoksi muodostuu 1,9. Neljännessä tehtävässä opiskelijoita pyydettiin tekemään sovelma, joka tulostaa annetun kuvion. Tämä tehtävä osoittautui kaikkein vaikeimmaksi kuten vastausten pistemäärän keskiarvostakin (0,5) voi päätellä. Kahdeksan opiskelijaa jätti kokonaan vastaamatta ja seitsemän yritti vastata jotakin, mutta sai silti 0 pistettä.

Jos tarkastellaan tenttituloksia taulukossa 11 myös ei-keskeyttäneiden osalta, niin näyttäisi heidän keskuudessaan olevan sovelmien tekeminen kaikkein vaikeinta. Sen sijaan ei-keskeyttäneiden parissa ohjelmoinnin perusrakenteet kuten valinta- ja toistolauseet on omaksuttu hyvin. Mielenkiintoista on nähdä taulukosta 11 kuinka tenttivastausten suhteellinen ero korreloi asioiden vaikeutta. Helpompien kysymysten osalta ei vastauksissa keskeyttäneiden ja ei-keskeyttäneiden välillä näyttäisi juurikaan olevan eroa. Suhteellinen ero kuitenkin kasvaa sitä mukaa kuin tehtävät hankaloituvat ja keskeyttäneet opiskelijat suoriutuvat vaikeammista tehtävistä ei-keskeyttäneitä selvästi heikommin. Tästä voidaan päätellä, että mitä vaikeampi ohjelmoinnin aihe on kyseessä, sen suuremmat ovat erot hyvin menestyvien ja keskitasoisesti tai sitä heikommin menestyvien opiskelijoiden välillä.

## *6.7 Tukitoimien tarpeellisuus*

Virtuaaliapprobatur-opiskelijoille jaettiin itseopiskeluopas, jossa esitellään käytössä oleva oppimisympäristö sekä annetaan ohjeita verkkopohjaista opiskelua varten. Oppimisympäristön käyttöä esiteltiin myös virtuaaliapprobaturin aloitustilaisuudessa Joensuun yliopistolla sekä kouluilla toimivien tutoropettajien tuli alussa olla opiskelijoiden tukena. Itseopiskeluoppaaseen tutustuminen jäi opiskelijoiden itsensä vastuulle. Oppaassa on kerrottu kurssien sisällöt ja aikataulut sekä annettu ohjeita kurssien suorittamiseksi. Tehtävien tekemisestä on kerrottu 1/3 suoritusvaatimus sekä esitelty kuinka tehdyistä harjoitustehtävistä kertyy bonuspisteitä.

Tutoropettajat toimivat kouluilla opiskelijoiden tukena ja erityisesti työskentelyn aloitusvaiheessa auttoivat opiskelijoita tutustumaan WebCT-oppimisympäristöön sekä Java-kääntäjän käyttöön. Lisäksi opiskelijoilla oli mahdollisuus ottaa yhteyttä joko sähköpostitse tai puhelimitse yliopistolla toimivaan ohjaajaan ongelmia kohdatessaan. WebCT:n keskustelupalsta toimi myös sekä opiskelijoiden että yliopiston ohjaajan taholta yleisenä informaation jakeluväylänä. Keskustelupalstalla kuitenkin keskustelut olivat melko vähäisiä ja lähinnä verkkomateriaaleissa eikä muissa järjestelyissä tai tehtävänannossa ilmenneitä puutteita tai virheellisyyksiä koskevia.

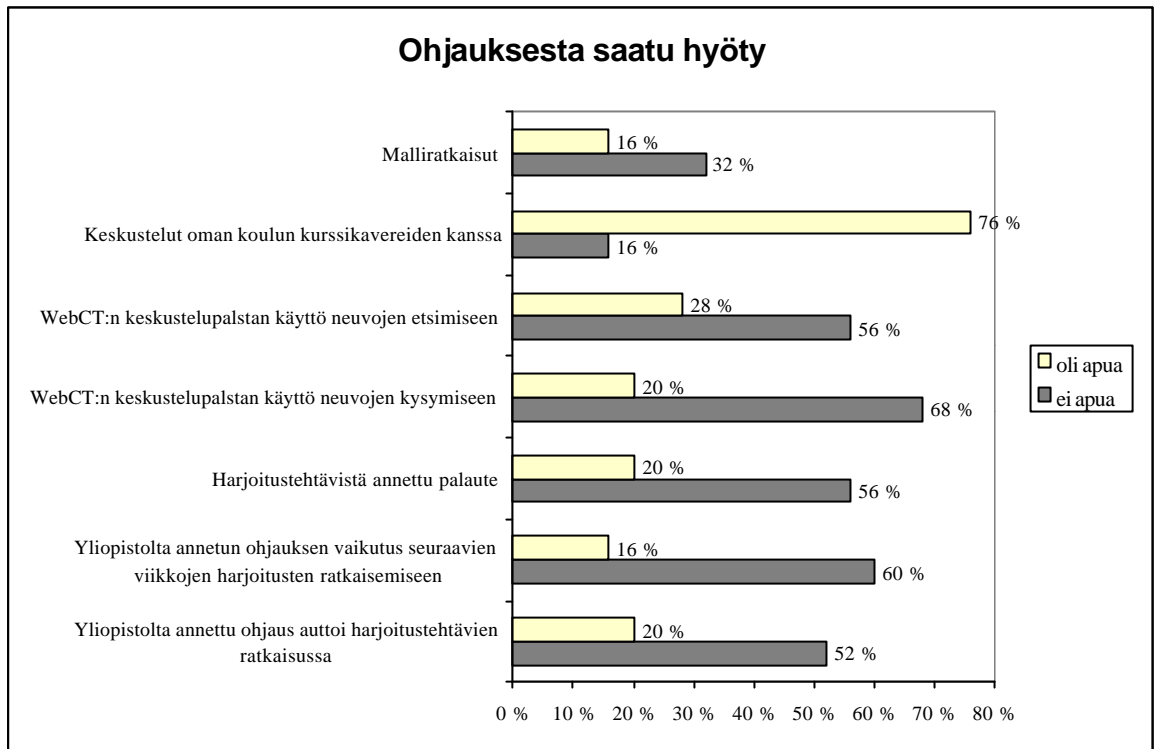
Kysymyksellä 5 (Arvioi olisiko seuraavista opetusmuodoista hyötyä ohjelmoinnin eri osa-alueiden oppimisessa) pyrin kartoittamaan erilaisten opetuksellisten tukitoimien tarpeellisuutta ohjelmoinnin eri osa-alueilla. Opiskelijoiden tuli arvioida olisiko luennosta, videoneuvotteluyhteydestä tai WebCT:ssä toteutetusta ohjatusta keskustelusta hyötyä ohjelmoinnin eri osa-alueiden oppimisessa. Kyselyn tulokset on esitetty taulukossa 12. Opiskelijoista suurin osa (53 %) kaipasi ohjelmointikurssin tueksi luentoja. Mielenkiintoista on, että helpoiksi koettuihin asioihin, kuten esimerkiksi muuttujien ja tunnusten käytön opettamiseen, on kaivattu lähiopetusjaksoja luennon muodossa yllättävänkin runsaasti (44 %).

**Taulukko 12 Tukitoimien tarpeellisuus ohjelmoinnin eri osa-alueilla (n=25)**

Ohjelmoinnin osa-alue	luento	video- neuvottelu	ohjattu keskustelu	keskiarvo
Muuttujien ja tunnusten käyttö	44 %	<b>20 %</b>	24 %	29 %
Sijoitus- sekä syöttö- ja tulostuslauseet	52 %	12 %	20 %	28 %
Toistolauseiden käyttö	40 %	16 %	24 %	27 %
Valintalauseet ja loogiset operaatiot	<b>56 %</b>	<b>8 %</b>	20 %	28 %
Taulukoiden ja satunnaislukujen käyttö	52 %	16 %	<b>40 %</b>	<b>36 %</b>
Metodien käyttö	56 %	<b>20 %</b>	28 %	35 %
Sovelmien ja grafiikan käyttö	56 %	16 %	24 %	32 %
Animaatioiden laadinta	<b>64 %</b>	<b>20 %</b>	24 %	<b>36 %</b>
Keskiarvo	<b>53 %</b>	16 %	26 %	

Uudet opettamismuodot kuten videoneuvotteluyhteys tai verkon kautta käyty ohjattu keskustelu eivät näyttäisi saavan kovinkaan innostunutta vastaanottoa, sillä pääosin alle 24 % vastaajista koki kummastakaan näistä olevan hyötyä ohjelmoinnin oppimisessa. Vaikeana koettujen taulukoiden osalta ohjatun keskustelun osuus kasvaa kuitenkin 40 %:iin. Valintalauseiden opiskelun tukemiseen peräti 56 % vastaajista uskoo luennosta olevan apua, mutta yllättäen vain 8 % uskoo videoneuvotteluyhteyden kautta olevan luennon auttavan. Tämä on sinällään mielenkiintoinen ilmiö: molemmissahan on käytössä sama opetusmuoto, ainoastaan toinen on kontaktiopetuksena annettua ja toinen sähköisen viestinnän kautta välitettyä luento-opetusta. Yhtenä selittävänä tekijänä näin suureen eroon näkisin opiskelijoiden ennakko-asenteet ja varovaisuuden uutta opetusmuotoa, videoneuvotteluyhteyden kautta olevaa luentoa, kohtaan.

Yleisesti tukitoimena virtuaaliopinnoille luentoa halusi 53 % vastaajista, videoneuvottelun kautta toteutettuja luentoja 16 % ja ohjattua keskustelua WebCT:ssä 26 %. Jos tarkastellaan, kuinka opiskelijat kaipasivat ohjelmoinnin eri osa-alueille ohjausta, niin taulukoiden ja satunnaislukujen käyttöön sekä animaatioiden tekemiseen kaipasi 36 % vastaajista tukea joko luentona, videoneuvotteluna tai ohjattuna keskusteluna. Metodien käyttöön kaipasi 35 % ja sovelmien tekemiseen 32 % vastaajista tukitoimintoja verkkomateriaalin ja kurssikirjallisuuden lisäksi.



**Kuva 14 Ohjauksesta saatu hyöty (n = 25)**

Kysymyksellä 3 (Ohjauksen saaminen ongelmatilanteissa) selvittelin eri ohjausmuotojen hyödyllisyyttä. Tulokset on esitelty kuvassa 14. Kaikkein eniten opiskelijat kokivat olevan hyötyä keskusteluista omalla koululla olevien kurssikavereiden kanssa (76 % vastaajista). Sen sijaan verkkoympäristön tarjoamien tukivälineiden, kuten WebCT:n keskustelupalstan, kautta saatu hyöty on jäänyt varsin vaatimattomaksi sillä ainoastaan 20 % vastaajista sai apua keskustelupalstalle lähettämänsä kysymyksen myötä ja 28 % seurasi WebCT:n keskustelupalstaa aktiivisesti löytääkseen sieltä ratkaisua omiin ongelma-kohtiin.

Opiskelijat eivät ole vielä tottuneet uuteen työskentelytapaan ja osa opiskelijoista ei todennäköisesti ole edes osannut ajatellakaan, että keskustelufoorumia voisi käyttää myös harjoitustehtävissä tai teorian opiskelussa kohdattujen ongelmatilanteiden selvittämiseen. Verkko-oppimisessa yhteistoiminnalliseen oppimiseen rohkaiseminen on ensi arvoisen tärkeää ja kuvan 14 tulosten perusteella lukio-ikäisiä tulisi tietoisesti ohjata käyttämään oppimisympäristössä olevia välineitä itsenäiseen ohjauksen ja tuen hakemiseen. Malliratkaisuista saatu hyöty on jäänyt yllättävän pieneksi keskeyttäneiden opiskelijoiden

osalta, sillä ainoastaan 16 % vastanneista arvioi hyötynensä niistä. Tämä on ongelmallista, sillä mikäli opiskelija ei edes harjoitusten jälkeen annettujen malliratkaisujen jälkeen ymmärrä annettua tehtävää, häneltä jää hyvin todennäköisesti itse opittava asiakin omaksumatta. Toisaalta on oletettavaa, että opiskelijoilla ei ole ollut riittävästi aikaa enää palata edellisen viikon harjoitustehtäviin ja niiden malliratkaisuihin, mikä osaltaan vaikuttaa tulokseen. Kaiken kaikkiaan voidaan kuitenkin päätellä, että opiskelijat eivät ole hyötynet verkon kautta saadusta ohjauksesta, sen sijaan omalla koululla opiskelijoiden kesken käydyistä keskusteluista on koettu olevan selkeästi hyötyä oppimisprosessissa.

## ***6.8 Jatkokyselyt***

Jatkokyselyt toteutin sähköpostikyselynä, koska uskoin näin saavani vastaukset kaikilta jatkokyselyyn tai haastatteluun lupautuneilta kuudelta opiskelijalta. Neljältä sain lopulta vastaukset. Ennen jatkokyselyä kävin läpi kouluilta palautuneet kyselykaavakkeet ja poimin sieltä tärkeimmiksi kokemani jatkokysymykset. Jatkokyselyssä oli neljä kysymystä, joista kolme ensimmäistä koskivat

- *teoriasivustojen kehittämistä,*
- *käytetyn ohjelmointikielen (Javan) omaksumista ja*
- *ohjauksen lisäämistä.*

Neljäntenä kysymyksenä oli avoin kysymys, jossa pyysin opiskelijoita arvioimaan, mikä kuluneen lukuvuoden järjestelyistä auttoi oppimista ja mistä taas ei oikein ollut hyötyä.

Jatkokyselyn tulokset eivät merkittävästi antaneet lisäinformaatiota varsinaiseen kyselyyn nähden, mutta vahvistivat kyllä muutamaa jo kyselyssä esille nousutta asiaa. Ensimmäkin jatkokyselyistä osasta vastauksista kuvastui valtavan suuri pettymys siitä, että opinnot keskeytyivät. Varsinaisessa kyselyssäkin avoimien kysymysten 8-9 (Mitä muuttaisitte ohjelmointikurssista? Ottaisitte jotakin pois tai haluaisitte jotakin lisää? sekä Olisiko sinulla muita toiveita kurssimateriaalin tai käytännön teknisen toteutuksen parantamiseksi?) osalta pystyi ”lukemaan tunteita”, mutta erityisesti jatkokyselyssä tunteiden purkaukset korostuivat.

Toinen asia, mikä vahvistui jatkokyselyssä, oli pettymys kurssin teoriaosuuteen niin kurssikirjan kuin WWW-materiaalinkin osalta. Teoriasivustojen kehittämisestä yksi jatkokyselyyn vastanneista toivoi, että WWW-materiaalista voisi teoriaa vähentää ja lisätä sivustoille mieluummin esimerkkejä ja animaatioita. Yksi vastaaja taas tyrmäsi sekä käytetyn Java-oppikirjan että WWW-materiaalin täysin ja kaksi vastaajaa koki, että WWW-materiaalissa teoria oli esitetty selkeämmin kuin oppikirjassa ja materiaalia tulisi vain kehittää vielä eteenpäin. Esimerkkien ja animaatioiden sekä helppojen harjaantumistehtävien lisäämistä pitivät kaikki vastaajat tarpeellisina, mutta yksi vastaajista epäili löytyykö aikaa välttämättä vapaaehtoisten harjaantumistehtävien tekemiseen. Erään vastaajan antama vastaus kertoo paljon: ”Tehtäviä tukevat esimerkit ovat aina hyväksi ja lisäävät onnistumisen iloa.” Esimerkkien ja helppojen harjaantumistehtävien määrää on siis syytä lisätä opiskelijan motivaation ja onnistumisen elämysten vuoksi.

Ohjauksesta omalla koululla uskoi kolme vastaajaa olevan hyötyä, mutta sen sijaan WebCT:n keskusteluhuoneen toimivuutta ohjaavissa keskusteluissa epäiltiin. Javaa ohjelmointikielenä kaikki neljä vastaajaa pitivät vaikeana tai ainakin haasteellisena. Kurssikirjaa kolme vastaajaa sanoi sekavaksi ja erityisesti yksi vastaajista kaipasi esimerkkien selityksiä ja luentoja oppimisen tueksi. Myös kyselylomakkeen avoimissa kysymyksissä välittyi sama viesti kurssikirjallisuuden sekä WWW-materiaalin suhteen: kaivattiin lisää selventäviä esimerkkejä WWW-materiaaliin.

## 7. JOHTOPÄÄTÖKSET

Kyselylomakkeita postitettiin yhteensä 55 kappaletta ja niistä palautui 25 kappaletta. Ohjelmointi 1 – kurssin jälkeen keskeyttäneitä opiskelijoita oli yhteensä 33, joista 10 oli sellaisia, jotka eivät osallistuneet yhteenkään tenttiin eivätkä palauttaneet yksiäkään harjoitustehtäviä. Palautuneiden lomakkeiden määrää voidaan mielestäni pitää tutkimuksen kannalta varsin hyvänä, sillä palauttamatta jääneisiin kyselykaavakkeisiin löytyy luonnollisia selityksiä: niistä kymmenestä opiskelijasta, jotka eivät osallistuneet yhteenkään tenttiin kahdeksan ei myöskään vastannut kyselyyn. Toinen selkeä ryhmä vastaamattomia olivat lukion kolmasluokkalaiset. Kyselykaavakkeet postitettiin huhtikuun alussa, jolloin abiturientit ehtivät jo lähteä kouluilta pois. Toisaalta on muistettava, että abiturienttien osalta oli tiedossa jo opintoja aloitettaessa, etteivät he ehdi suorittaa koko 15 opintoviikon kokonaisuutta ja he lähtivätkin suorittamaan ainoastaan osaa approbaturkurseista. Palautuneista lomakkeista aikavälillä syyskuu 2000 – tammikuu 2001 keskeyttäneitä oli 15 eli 71 % kyselykaavakkeisiin vastanneista keskeytti Ohjelmointi 1 –kurssin aikana tai heti sen jälkeen tammikuussa, joten vastaukset antavat mielestäni varsin luotettavaa tietoa ohjelmoinnin perusteiden oppimisen vaikeuteen virtuaalisessa oppimisympäristössä (Kuva 7 sivulla 42).

*Tutkimusasetelman ongelmat: keskeyttäneiden lukumäärän määrittäminen ja tutkimuksen henkilökohtaisuus*

Tutkimukseni ongelmallisimmaksi kohdaksi näytti alkuun nousevan niinkin yksinkertaiselta tuntuva asia kuin kuka on keskeyttänyt. Onko sellainen opiskelija keskeyttänyt, joka oli virtuaaliapprobatur-opintojen aloitustilaisuudessa mukana, mutta ei sen jälkeen osallistunut millään tavoin virtuaaliapprobatur-opintoihin? Voidaanko tällaiset opiskelijat ottaa mukaan tutkimukseen ja opiskelijatilastoihin tarkasteluajankohdittain? Entä opiskelijat, jotka ovat suorittaneet kurseja, mutta jotka eivät WebCT:n tilastoinnin mukaan ole enää pitkään aikaan palauttaneet harjoitustehtäviä tai osallistuneet tentteihin. Kun tutoropettaja tarjosi tällaiselle opiskelijalle täytettäväksi tutkimukseen liittyvää kyselykaavaketta, opiskelija kieltäytyikin vastaamasta koska ei mielestään ole keskeyttänyt opintojaan. Tämän vuoksi määrittelin, että opiskelija katsotaan keskeyttäneeksi, mikäli hän ei ole enää pitkään aikaan osallistunut virtuaaliapprobatur-työskentelyyn palauttamalla harjoitustehtäviä tai osallistumalla tentteihin. Opiskelijalla

itsellään saattaa olla tarkoituksena jatkaa opintoja seuraavana lukuvuonna, mutta tässä tutkimuksessa hänet silti tulkitaan keskeyttäneeksi lukuvuonna 2000-2001.

Tarkasteluajankohdat valitsin niin, että pystyn jollakin tavoin varmuudella näkemään montako opiskelijaa missäkin vaiheessa oli mukana. Opiskelijoiden ei tarvitse ilmoittautua jokaiselle kurssille erikseen, mikä omalta osaltaan vaikeuttaa keskeyttäneiden opiskelijoiden tilastointia. Kokoamiani keskeyttäneiden määriä voidaan kuitenkin pitää varsin luotettavina, sillä ainoastaan 31.3.2001 päivän tieto perustuu opiskelijoiden opiskelukäyttäytymiseen liittyvään arvioon. Kevään tarkastelupäivään 31.3.2001 mennessä keskeytti 62 % aloittaneista, mikä on hieman korkeampi kuin yleensä verkkopohjaisissa kursseissa Cornellin ja Martinin (1997) mukaan. He toteavat, että verkkopohjaisen kurssin keskeyttää keskimäärin 30-50 % aloittaneista opiskelijoista. Keskeyttämispäätöksen syntyyn vaikuttavat opiskelijoiden ja opettajien teknologiapohjaisen oppimisympäristön hyväksyminen työväliseinä, aiempi kokemus, ennakkoluulot teknologiaa kohtaan, opiskeltavan asian asiayhteys, kurssin interaktiivisuuden aste, järjestelmän käytön hankaluudet, järjestelmän helppo saatavuus sekä opiskelijoiden ja opettajien kyvykkyys tai kyvyttömyys kommunikointiin (Cornell & Martin, 1997). Joensuussa niin opiskelijoille ja opettajille kuin opintojen järjestäjälle yliopistollekin virtuaaliapprobatur oli ensimmäinen näin laajamittainen verkkopohjainen toimintamalli: työskentelytavat ja järjestelmän käyttäminen olivat uusia opeteltavia asioita ja tämä on varmasti osaltaan vaikuttanut keskeyttämisherkkyyteen.

Toinen vaikeus tutkimuksessani on toimintatutkimukselle tyypillisesti tutkimuksen henkilökohtaisuus. Olin itse mukana tekemässä virtuaaliapprobaturia lukuvuodeksi 2000-2001 ja päävastuualueenani oli Ohjelmointi 1 - kurssin tekeminen. Kun kyselyyn vastaajana oli kurssin keskeyttänyt opiskelija ja muotona kirjallinen kyselytutkimus sekä sähköpostitse toteutettu jatkokysely, niin osassa vastauksista purkautui tunteita vahvastikin. Pystyin kuitenkin ottamaan etäisyyttä tilanteeseen ymmärtämällä, että vastaukset olisivat voineet olla toisenlaisia, jos kysely olisi ulotettu kaikille opiskelijoille tai jos kysely olisi toteutettu haastatteleamalla. Joissakin vastauksissa tuntui, että keskeyttäneen opiskelijan pettymys ja paha olo purkautui kyselyn myötä hyvinkin ankarana kritiikkinä kaikkia virtuaaliapprobaturissa jollakin tavoin mukana olevia kohtaan teoriasivustojen tekijää, kurssikirjan kirjoittajaa, yliopistolla toimivia ohjaajia ja koululla olevaa tutoropettajaa myöten. Tuohtuminen ja kaikkien virtuaaliapprobaturissa jollakin tavoin mukana olevien syytely kertovat suuresta pettymyksestä, jonka opiskelija koki siinä vaiheessa kun



havaitti, ettei pystykään omaksumaan opittavaa asiaa niin hyvin kuin etukäteen oletettiin. Toisaalta kyselyyn vastanneista opiskelijoista osalta tuli myös kiitoksia koko virtuaaliapprobatur-ryhmälle ja erityisesti maininta ”yksikään virtuaaliapprobaturin parissa viettämäni tunti ei ollut hukkaan heitetty” antaa uskoa jatkaa työtä kurssien kehittämisessä eteenpäin. Epäonnistuneen suorituksen jälkeen aina osa reagoinnista tapahtuu tunnetasolla. Kuusinen (1995) toteaaakin, että opintosuorituksessaan epäonnistunut opiskelija tuntee toisia kohtaan suurta suuttumusta ja pyrkii itseään suojellakseen tulkitsemaan omien epäonnistumisten syyt ulkoisiksi.

#### *Opintojen keskeyttämissyyt: aikapula ja teoriapainotteisuus*

Ensisijaiseksi opintojen keskeyttämissyyksi nousi aikapula. Lähes puolella vastaajista (44 %) tuli ongelmia löytää riittävästi aikaa yliopisto-opintoihin lukio-opintojen ohella. Yliopisto-opinnoissa peruslaskusääntönä on, että yksi opintoviikko vastaa noin 40 työtuntia. Ohjelmoinnin perusteiden kurssi on kahden opintoviikon laajuinen kokonaisuus, joten opiskelijalta tulisi syyslukukauden aikana lukio-opintojen ohessa löytyä noin 80 tuntia aikaa ohjelmoinnin perusteiden itseopiskeluun sekä harjoitusten tekemiseen. Vaikka opiskelijoiden on kyettävä itsenäiseen työskentelyyn ja otettava vastuu omien opintojensa etenemisestä, niin silti tulisi opiskelijoita ohjata kurssivalinnoissa ja *ohjauksellista interventiota* opettajan taholta tarvitaan. Lukioikäisen on ollut vaikea arvioida itse virtuaaliopintoihin tarvittavaa aikaa ja ajanpuutteen vuoksi keskeyttämisspätöksen syntyminen toi mukanaan syviäkin pettymyksiä osalle opiskelijoista.

Moni opiskelija (20 %) koki lisäksi, että tehtävät olivat liian vaikeita. Sekä keskustelufoorumissa että kyselykaavakkeen avointen kysymysten palautteissa mainittiin myös, että tehtäviä oli liikaa. Näihin tuloksiin perustuen olemme muuttaneet lukuvuodeksi 2001-2002 harjoitustehtävien määrää pienemmäksi antamalla harjoitustehtävien tekemiseen lisää aikaa ja yhdistämällä joitakin osin kahden viikon teorit yksisiin harjoituksiin: esimerkiksi harjoitustehtävät 1 sisältää viikkojen 1 ja 2 teorit. Vaikeina koettujen asioiden kohdalla kuten esimerkiksi valintalauseet, taulukot ja metodit, on kullekin teoriaviikolle omat harjoitukset. Toisaalta päivitettyyn WWW-materiaaliin on lisätty opittavaan asiaan liittyviä esimerkkejä sekä helppoja harjaantumistehtäviä. Harjaantumistehtävät ovat sellaisia, joista ei saa bonuspisteitä ja joita ei lasketa mukaan tehtyjen harjoitusten lukumääriin, mutta joita läpikäymällä opiskelija pääsee heti

teorian opiskeluaan harjoittelemaan opittavaa asiaa käytännössä. Harjaantumistehtävät ovat lisäksi niin yksinkertaisia, että myös heikompien tai keskivertaisten opiskelijoiden tulisi niistä selviytyä. Ratkaisevaksi lopulta jää opiskelijan oma aktiivisuus ja vastuullisuus omista opinnoistaan kuten Mäki-Komsi (1999) toteaa.

*Vaikeina koetut osa-alueet: taulukoiden ja satunnaislukujen käyttö, metodien käyttö, sovelmien sekä animaatioiden laatiminen*

Ohjelmoinnin eri osa-alueista vaikeina koettiin odotetusti taulukoiden ja satunnaislukujen sekä metodien käyttö. Sen sijaan yllätyksellisenä pidin sitä, että myös sovelmien ja grafiikan käytön koki peräti 60 % vastaajista vaikeiksi asioiksi. Valintalauseiden ja loogisten operaatioiden käytön lähes puolet vastaajista (48 %) koki hankalaksi, jota pidin myös yllättävän suurena määränä.

Toisaalta kun tarkastellaan vaikeiden asioiden sijoittumista kurssin eri ajankohtiin (Kuva 11 sivulla 46), niin vaikeana tai melko vaikeana koetut asiat ovat kasautuneet kurssin loppupuolelle ja viikosta 7 alkaen ei ole ollut enää yhtään helpompaa teoriaviikkoa. Parannustoimenpiteinä Ohjelmointi 1 -kurssin aikataulua on muutettu niin, että alkupuolen helpompina koetut asiat muuttujien ja tunnusten käyttö, sijoitus-, syöttö- ja tulostuslauseiden sekä toistolauseiden käyttö käydään läpi tiiviimmässä aikataulussa kuin lukuvuonna 2000-2001. Sen sijaan taulukoiden ja satunnaislukujen käyttöön, sovelmien tekemiseen sekä metodien käyttöön on varattu enemmän aikaa. Muun muassa valintalauseiden ja taulukoiden läpikäyminen on jaettu useammalle viikolle. Lukuvuonna 2001-2002 animaatioiden laadinta jätetään ensimmäiseltä kurssilta kokonaan pois ja opetetaan vasta kevään ohjelmointikurssin aikana. Ohjelmointi 1 -kurssiin puolestaan lisätään näppäinten ohjelmointi, jolla pyritään motivoimaan opiskelijoita graafiseen ohjelmointiin. Näppäinohjelmoinnin avulla pystytään harjaannuttamaan vielä loppukurssistakin valintalauseiden ja loogisten operaatioiden käyttämistä 16-19 - vuotiaille mielekkäämmällä ja innostavammalla tavalla graafista ohjelmointiympäristöä hyväksi käyttäen. Toisaalta näin pystytään myös linkittämään alkukurssin teorit loppukurssin sovelmien tekemiseen. Lisäksi kurssille on varattu viimeinen viikko kertausta varten, jolloin ennen tenttiä opiskelijoilla olisi yksi kokonainen viikko aikaa palautella mieliin alkukurssin asioita kertaustehtävien muodossa eikä uutta opittavaa asiaa enää tulisi.

*Harjoitustehtävien tekeminen: keskivertaiset tai sitä heikommat opiskelijat eivät juurikaan hyötäneet harjoitustehtävistä*

Harjoitustehtävien tekemisessä keskeyttäneiden ja ei-keskeyttäneiden opiskelijoiden toimintatavoissa on selkeää eroa. Paradoksaalista on, että juuri ne opiskelijat, jotka tarvitsisivat harjaantumista, näyttävät tekevän ainoastaan sen mikä on pakollista. Opiskelijat kokevat varsin usein annetut tehtävät vain työksi, joka tulee suorittaa eivätkä ymmärrä harjoitusten opetuksellista tarkoitusta (Carpone et al, 2000). Keskeyttäneiden opiskelijoiden harjoitustehtävien palauttamisessa oli nähtävissä kolme selkeää muutosta: aivan opintojen alussa keskeyttäneet opiskelijat olivat innokkaasti mukana ja palauttivat tunnollisesti ensimmäisen viikon tehtävät. Sen jälkeen alkoi kuitenkin innostus vaimentua: ensimmäiset viikot koettiin asioina helpoiksi, joten olisi odottanut, että myös harjoitustehtävistä olisi ollut helppo selviytyä. Yksi mahdollinen selitys tälle ilmiölle on, että opiskelijat ovat kokeneet tehtävät alussa liiankin helppona ja ajattelivat, ettei niitä tarvitse suorittaa ja priorisoivat ajankäyttönsä alkusyksystä muualle kuin helpolta vaikuttavan ohjelmoinnin opiskeluun. Lokakuussa ensimmäisen virtuaaliapprobatur-tentin jälkeen (9.10.) harjoitustehtävien tekeminen Ohjelmointi 1 –kurssilla tasaantuu. Osa opiskelijoista lopetti aktiivisen työskentelyn ensimmäisen tentin jälkeen, joten ensimmäisen tentin tuloksella on ollut vaikutusta opintojen lopettamispäätökseen. Toisaalta Ohjelmointi 1 –kurssin teoria on edennyt tuossa vaiheessa jo valintalauseisiin ja loogisiin operaatioihin, joten on todennäköistä, että keskivertaisilla opiskelijoilla alkoivat tässä vaiheessa vaikeudet, varsinkin kun pohjalta puuttui tehtävien tekemiseen tarvittava rutiini.

Keskeyttäneiden osalta harjoitustehtävien palauttaminen jatkuu suhteellisen tasaisena, kunnes marraskuussa tulee pientä piristymistä metodien käytön ja sovelmien tekemisen yhteydessä. Osalla opiskelijoista on selvästi ollut tavoitteena saada 1/3 tehtävistä tehtyä, jotta saisi kurssin suoritettua. Marraskuun bpulla kuitenkin tulee jälleen selvä pudotus, johon näkisin kaksi syytä. Ensimmäkin ne opiskelijat, jotka arvioivat etteivät kuitenkaan saa suoritettua vähintään 1/3 kaikista tehtävistä, lopettivat tuossa vaiheessa yrittämisen. Toisaalta loppukurssin asiat (animaatiot) koettiin vaikeina asioina opiskella ja osalle opiskelijoista harjoitustehtävät ovat olleet liian hankalia, eivätkä he yksinkertaisesti saaneet harjoitustehtäviä suoritetuiksi.

Ensimmäisen lukuvuoden perusteella motivoivaksi porokanaksi tarkoitettu bonuspistejärjestelmä näyttää toimivan hyvien opiskelijoiden keskuudessa hyvin: menestyneet opiskelijat tekevät myös harjoitustehtäviä tunnollisesti, hyötyvät bonuspistejärjestelmästä ja menestyvät tentissä hyvin. Sen sijaan juuri ne opiskelijat, jotka tarvitsisivat harjaantumista, kannustusta ja ohjausta, eivät hyödy bonuspisteistä juurikaan. Itseopiskeluoppaassa sekä teoriasivuilla ja opintojen aloitustilaisuudessa kannattaisi korostaa vieläkin selkeämmin esimerkkien, harjaantumistehtävien, viikkoharjoitusten sekä malliratkaisujen läpikäymisen merkitystä oman opiskelun ja oppimisprosessin tukena. Leen (2000) mainitsevat opiskelijan henkilökohtaiset ominaisuudet, kuten motivaatio ja itsearviointi, ovat merkittäviä näkökohtia oppimistulosten tarkastelussa virtuaalisessa oppimisympäristössä. Lukuvuoden 2001-2002 materiaaliapäivityksen yhteydessä opiskelua on pyritty tukemaan lisäämällä harjoitustehtäviin sellaisia tehtäviä, jotka on integroitu teoriasivustojen harjaantumistehtäviin ja esimerkkeihin. Näin pyritään ohjaamaan opiskelijoita viimeistään harjoitustehtävien tekovaiheessa lukemaan teoriasivujen esimerkit ja harjaantumistehtävät läpi sekä arvioimaan näiden avulla omaa osaamistaan. Toisaalta teoriasivuilla olevien helppojen harjaantumistehtävien tarkoituksena on tarjota keskitasoisille ja sitä heikommille opiskelijoille onnistumisen tunnetta ja kasvattaa näin motivaatiota opintoihin ja varsinaisten harjoitusten tekemiseen. Tavoitteena on synnyttää oppimisen halu.

Virtuaalisessa oppimisympäristössä opiskelijoiden toinen toisillensa antaman tuen merkitys korostuu ja oppimisympäristön tulisikin pystyä tarjoamaan tehokkaammat välineet ryhmätyöskentelyyn esimerkiksi ohjelmoinnin harjoitustehtävien tekemiseen liittyvissä kysymyksissä (Meisalo et al, 2001). Tulevina lukuvuosina opiskelijoita on syytä tietoisesti rohkaista käyttämään oppimisympäristön tarjomia väyliä kuten esimerkiksi keskustelufoorumia tai Chat-huoneita harjoitustehtäviin liittyviin ryhmäkeskusteluihin. Valitettavasti WebCT:ssä ei ole valmiina välinettä, jotka tukisivat yhteistoiminnallista oppimista. Varsinkin heikommat opiskelijat saattavat pelätä leimautuvansa lähettäessään yleiselle keskustelufoorumille kysymyksiä viikon harjoitustehtävien ratkaisemiseen liittyen. Lukuvuoden 2000-2001 kokemus osoittaa, että keskustelufoorumia käyttivät lähinnä hyvin menestyneet opiskelijat. Oppimisympäristön tulisi tarjota välineitä, jotka takaavat mahdollisuuden esimerkiksi muutaman opiskelijan ryhmäkeskusteluihin. Ryhmäkeskusteluissa oppilaalla tulisi olla niin turvallinen tunne, että voi kysyä aivan mitä tahansa pelkäämättä tulevansa leimatuksi tyhmäksi tai esittävänsä liian yksinkertaisia kysymyksiä.

*Tenttivastaukset: erityisesti toistolauseiden ja satunnaislukujen käyttö sekä sovelmien tekeminen jäänyt omaksumatta*

Keskeyttäneiden opiskelijoiden tenttivastausten läpikäynti vahvisti kyselytutkimuksessa esille nousseita vaikeina tai melko vaikeina koettuja ohjelmoinnin osa-alueita. Jos Ohjelmointi 1 –kurssin varsinaista tenttiä 11.12.2000 tarkastellaan keskeyttäneiden opiskelijoiden hankalina kokemien asioiden valossa, niin tentin kaikki kysymykset ovat sisältäneet vaikeina koettuja asioita. Ensimmäinen tenttitehtävä on yleensä tarkoitettu helpommaksi aloitustehtäväksi, josta jokainen pystyy ilman suurempia ponnisteluja selviytymään. Valintalauseiden käytön koki keskeyttäneistä lähes puolet (48 %) vaikeana tai melko vaikeana. Tenttivastauksissa oli erittäin hyviä lähes täyden pisteen vastauksia, mutta myös keskinkertaisiakin löytyi seitsemän kappaletta. Toisessa tenttikysymyksessä tarvittava toistolause on jäänyt keskeyttäneiden opiskelijoiden osalta selkeästi omaksumatta, vaikka opiskelijat eivät kokeneetkaan toistolauseita kovin hankalina oppia. Ohjelmointi 1 -materiaalipäivityksen yhteydessä valintalauseiden teoriaosuus on jaettu kahdelle viikolle ja annettu näin enemmän aikaa omaksua ja harjoitella sitä. Lisäksi toistolauseisiin ja toistolauseiden ja valintalauseiden yhteiskäyttöön on lisätty sekä esimerkkejä että harjaantumistehtäviä. Mikäli opiskelija ei pysty näitä ohjelmoinnin perusrakenteita omaksumaan osaksi rutiininomaista ohjelmointi-osaamistaan, hänelle tulee suuria vaikeuksia ohjelmoinnin opiskelun edetessä. Näiden perusteiden oppimiseen kannattaa siis panostaa.

Edelleen tentissä tehtävät 3 ja 4 osoittivat, että kyselyn antamat suuntaviivat olivat varsin oikeita. Kyselyssä peräti 72 % vastaajista koki taulukoiden ja satunnaislukujen käytön hankalana, tentissä taulukkoa ja satunnaisluvun käyttöä vaativasta tehtävästä (tehtävä 3) keskeyttäneiden vastausten pistemäärän keskiarvo jäi varsin vaatimattomaksi (1,90). Sovelmien käytön koki 60 % keskeyttäneistä vaikeina ja myös tentissä keskeyttäneiden vastausten pistemäärän keskiarvo (0,45) vahvistaa, että sovelmien tekemiseen ei ole ennätetty harjaantua riittävästi. Materiaalipäivityksen myötä aikaistettiin sovelmiin siirtymistä viikolle 8 ja metodien opettaminen tehdään sovelmia käyttäen.

Tenttitulos voisi sovelmien osalta olla parempi, mikäli koe tehtäisiin tietokoneilla, jolloin opiskelija pystyisi kokeilemaan sovelmien toimintaa ja tarvittaessa korjaamaan sitä. Käytännössä kuitenkin lähes 60 opiskelijan tenttiä varten tarvittavan koneluokan tai koneluokkien järjestäminen ja tentin valvonta on

lähes mahdotonta jo teknisistä syistä. Toisaalta onko koneen käyttäminen kokeessa kuitenkin ainoa autuaaksi tekevä ratkaisu? Ohjelmointiongelmia ei käytännön elämässäkään ratkaista suoraan koneen ääressä istumalla. Siitä huolimatta osa opiskelijoista hahmottaa ratkaistavan ongelman paremmin, kun pystyy tekemään koodin ja samalla kokeilemaan sen toimintaa.

*Oppimisen tukeminen: lukioikäiset kaipaavat opintojen tueksi luentoa tai muuta tukea*

Virtuaalisessa oppimisympäristössä opiskelijan on kyettävä itsenäiseen työskentelyyn ja otettava vastuu omista opinnoista. Tässä tutkimuksessa seitsemän vastaajaa kaipasi luentoa jokaisen opetettavan asian tueksi mikä kertoo siitä, että nämä opiskelijat eivät olleet vielä kypsiä itsenäiseen verkkopohjaiseen työskentelyyn. Yleisestikin keskeyttäneiden vastauksista kävi ilmi, että opiskelijat kaipasivat opintoihin tukea ja ohjausta ja kokivat, että ohjelmoinnin opiskelu itsenäisesti verkon välityksellä on aivan liian vaikeaa. Lukuvuodelle 2001-2002 päivitetystä Ohjelmointi 1 -materiaalissa on pyritty tukemaan opiskelua esimerkein ja helpoin harjaantumistehtävin. Lopullinen onnistuminen jää kuitenkin opiskelijan oman vastuullisuuden ja aktiivisuuden varaan. Opiskelija on virtuaalisessa oppimisympäristössä avainasemassa kuten Chute ym (1997) kaaviokuvassaan (Kuva 2 sivulla 21) esittää. Opiskelijan ympärille on syytä rakentaa riittävän hyvä tukiverkosto, jotta opiskelija ei koe jäävänsä yksin ja ilman tukea oppimistilanteeseen. Pelkkä itseopiskeluoppaan jakaminen ei riitä evääksi virtuaaliopintojen aloitukselle lukioikäiselle opiskelijalle, vaan opiskelijaa on syytä hetken matkaa auttaa ja tukea opintojen menestyksekkään ja mielekkään aloituksen turvaamiseksi. Jos heti alussa koetaan epäonnistumista, niin se voi heijastua koko approbaturin suorittamiseen motivaation laskuna.

Perinteisesti suomalaisessa yliopistokulttuurissa ei juurikaan holhota opiskelijoita heidän opintojensa tai oppimisen suhteen. Virtuaaliapprobaturissa on kuitenkin opiskelijoiden ikäjakaumana pääosin 16-19-vuotiaat lukio-opiskelijat, joista osalle voi olla hyvinkin suuri muutos jo opetella opiskelemaan lukio-opintoja peruskoulun jälkeen puhumattakaan yliopistotason opiskelusta. Kuten Lehtinen (1997) toteaa, oppimisen tukemista suunniteltaessa on huomio kiinnitettävä koko siihen ympäristöön, joka kehystää opetusta ja oppimista. Suunnittelussa olisi otettava huomioon, että osa kurssilaisista on vasta 16-vuotiaita eivätkä välttämättä kovin valmiita itsenäiseen ja vastuulliseen, omatoimiseen opiskeluun. Opintojen järjestäjän, yliopiston, taholta on itsestään selvää, että lukiolaisille tarjotun approbaturin täytyy olla

vaativuustasoltaan lähiopetuksessa olevan tietojenkäsittelytieteen approbaturin tasoinen, joten kurssisisältöjä ei voida helpottaa. Kuinka lukiolaisilla sitten olisi mahdollisuus selvittää approbatur-opinnoista kunnialla läpi? Ohjauksesta saadun hyödyn analysoinnin perusteella (Kuva 14 sivulla 56) on syytä panostaa opiskelijoiden tukemiseen ja ohjaamiseen siitäkkin huolimatta, että suomalainen yliopistokulttuuri ei ole toimintaperiaatteiltaan opiskelijoita holhoava. Opintojen aloitusvaiheessa syksyllä kannattaisi niin yliopiston kuin koululla tutoropettajankin taholta panostaa siihen, että aluksi valvotaan jonkun aikaa, että opiskelijat todella ottavat vastuun omista opinnoistaan ja tekevät muutakin kuin vain sen, mikä on pakollista. Tätä opiskelijoiden oman aktiivisuuden korostamista olisi syytä tuoda esille kaikissa mahdollisissa tilanteissa: itseopiskeluoppaassa, aloitustilaisuudessa sekä koululla tutoropettajan järjestämissä tilaisuuksissa. Esimerkit ja harjaantumistehtävät on suunnattu nimenomaan keskitason tai sitä heikommille opiskelijoille, jotka valitettavasti varsin usein tekevät vain sen, mikä on pakko. Hyvät opiskelijat taas kiinnostuvat ja haluavat yrittää ratkaista kaikki mahdolliset tehtävät, niin pakolliset kuin vapaavalintaisetkin, ja juuri he pärjäisivät ilman lisäharjoituksiakin!

Lukuvuonna 2000-2001 tarjottiin virtuaaliapprobatur-opintomahdollisuutta kaikille Pohjois-Karjalan maakunnan lukiolaisille. Osalla lukioista tutoropettaja esitteli opiskelumahdollisuutta ainoastaan niille opiskelijoille, jotka hänen näkemyksensä mukaan voisivat menestyä yliopistotason opinnoissa. Osalta lukioista taas tulivat mukaan kaikki asiasta kiinnostuneet opiskelijat riippumatta siitä oliko heillä riittävästi tietoteknistä taustaa. Keskeyttäneiden opiskelijoiden joukossa varsinainen aiempi ohjelmointitaito näytti olevan varsin vähäistä, mikä osaltaan varmasti selittää sitä, että Ohjelmointi 1 -kurssin asiat koettiin vaikeina. Toisaalta kun kyselyyn vastanneista opiskelijoista lähes puolet halusi jokaisen ohjelmoinnin osan alueen opettamisen tueksi luennot yliopistolta, niin herää kysymys olivatko opiskelijat itse asiassa kovinkaan valmiita opiskelemaan virtuaalisesti? Toisaalta oppimisympäristössä on tarjolla erilaisia väyliä, joita opiskelijat eivät välttämättä osanneet vielä käyttää hyväkseen. Esimerkiksi opiskelijoita tulisi rohkaista yhteistoiminnalliseen työskentelytapaan niin oman koulun approbatur-opiskelijoiden kuin verkon läpi muiden lukioiden opiskelijoiden kanssa. Keskustelufoorumia voisi käyttää ongelmakohtien selvittämisen ja avun pyytämisen väylänä, jolloin opiskelijat pystyisivät auttamaan toinen toistaan ohjelmoinnin perusteiden oppimisessa. WebCT:n keskusteluhuoneisiin voisi sopia esimerkiksi viikoittaisen kokoontumisajan, jolloin keskusteluhuoneeseen saapuvat pystyisivät keskustelemaan kyseisen viikon teoriasta ja harjoitustehtävistä ja niiden ratkaisusta. Ohjelmoinnin perusteiden

opiskelussa ryhmätyö usein antaa paljon ryhmän eri jäsenten tuodessa ongelman ratkaisuun oman näkemyksensä. Virtuaalisessa ympäristössä tämä yhteistoiminnallinen työskentelytapa tapahtuisi vain verkon välityksellä.

Erilaiset algoritmien visualisointivälineiden kuten esimerkiksi Jeliotin, BlueJ:n tai Excelin visualisointiominaisuuksien käyttöä voisi lisätä varsinkin vaikeina koettujen asioiden opettamisessa. Animoinnissa tärkeää on kuitenkin varmistua, että opiskelija todella ymmärtää kuinka graafinen esitys mallintaa algoritmin toimintaa (Ben-Bassat Levy et al, 2001). Virtuaalisessa ympäristössä tukitoimien käytön opettelussa on tutoropettajan rooli merkittävä. Aluksi koulujen tutoropettajat tulisi kouluttaa käyttämään esimerkiksi Jeliotia ja sen jälkeen pyytää heitä ohjaamaan opiskelijoita tämän visualisointivälineen hyödyntämiseen osana opiskeluprosessia.



## 8. YHTEENVETO

Tutkimus toteutettiin lukuvuonna 2000-2001 virtuaaliapprobaturin keskeyttäneiden opiskelijoiden keskuudessa kyselytutkimuksena ja sähköpostitse toteutettuna jatkokyselynä kuudelle jatkokyselyyn tai haastatteluun lupautuneelle opiskelijalle. Tutkimuksessa käytiin läpi keskeyttäneiden opiskelijoiden tenttivastaukset sekä Ohjelmointi 1 -kurssin osalta kaikkien kurssilaisten harjoitustehtävien palautukset. Jatkokysely olisi ollut järkevintä tehdä esimerkiksi haastattelemalla, vaikkakin haastateltavat sijoittuivatkin ympäri Pohjois-Karjalaa ja maantieteellisesti kauaksikin toisistaan. Haastattelun avulla olisi voinut todennäköisesti paremmin johdatella opiskelijaa kertomaan ongelmallisista kohdista sekä kuinka ohjelmoinnin oppimista virtuaalisessa ympäristössä voitaisiin tukea.

Tutkimuksessa ilmeni, että keskeisimmäksi keskeyttämissyyksi nousivat aikapula ja teoriapainotteisuus. Molempiin näihin tekijöihin on pyritty tuomaan helpotusta lukuvuoden 2001-2002 materiaalipäivitysten myötä. Ohjelmoinnin perusteiden opiskelussa koettiin hankalaksi taulukoiden ja satunnaislukujen käyttö, metodien käyttö, sovelmien ja grafiikan käyttö sekä animaatioiden tekeminen. Lähes puolet koki myös valintalauseiden käytön vaikeaksi. Keskeyttäneet opiskelijat olivat saaneet tehtyä harjoitustehtäviä huomattavasti vähemmän kuin ei-keskeyttäneet. Malliratkaisujen, esimerkkien ja harjaantumistehtävien läpikäymistä on syytä korostaa opiskelijoille kaikissa mahdollisissa tilanteissa muistaen kuitenkin, että kyse on alle 19-vuotiaista nuorista lukiolaisista. Jokaisena lukuvuonna aloittavien opiskelijoiden joukkoon mahtuu varmasti myös sellaisia opiskelijoita, jotka tarvitsisivat vielä holhoavaa ohjausta ja tukea. Vastuullisuuden kasvaminen vie aikaa ja alussa voisi olla hyväkin, jos esimerkiksi tutoropettaja voisi olla opiskelijan tukena ikään kuin saattelemassa häntä itsenäiseen työskentelyyn.

Millaisin toimin opetusta virtuaalisessa oppimisympäristössä tulisi kehittää, jotta opiskelijat selviytyisivät ohjelmoinnin perusteiden oppimisesta paremmin? Yli puolet opiskelijoista haluaisi luentoa verkkopohjaisen opiskelun tueksi lähes kaikkiin ohjelmoinnin osa-alueisiin. Ohjattua keskustelua ja videoneuvottelua kohtaan oltiin hieman varauksellisia. Ainoastaan taulukoiden ja satunnaislukujen käytön osalta oppimisympäristössä tapahtuva ohjattu keskustelu näytti kiinnostavan opiskelijoita. Tukitoimina lukuvuodelle 2001-2002 on lisätty esimerkkejä ja harjaantumistehtäviä sekä tehty aikataulusta sellainen,

että opiskelijoilla pitäisi olla rauhassa aikaa tehdä tehtäviä. Lisäksi harjoitustehtävistä pyritään antamaan perusteellisempaa palautetta. Java-kääntäjän ja -tulkin asentaminen on otettu osaksi ensimmäisen viikon teoriaa, koska kääntäjän moitteeton toiminta on kurssin jatkon kannalta perusedellytys.

Tutkimuksen aikana heräsi useita kysymyksiä, joita olisi mielenkiintoista tutkia. Ensinnäkin olisi mielenkiintoista verrata poikkeavatko virtuaaliapprobaturin menestyksekkäästi loppuun saakka suorittaneiden opiskelijoiden näkemykset ohjelmoinnin perusteiden oppimisessa hankaliksi koetuista asioista opintonsa keskeyttäneiden opiskelijoiden näkemyksistä? Millaisia tukitoimia menestyneet opiskelijat kaipaavat ohjelmoinnin perusteiden opiskeluun virtuaalisessa oppimisympäristössä?

Toisaalta olisi mielenkiintoista tutkia vertailuryhmänä syksyllä 2001 alkavaa virtuaaliapprobaturiryhmää ja näiden opiskelijoiden selviytymistä ohjelmoinnin perusteiden oppimisessa. Lukuvuodelle 2001-2002 on tehty muutoksia kurssimateriaalien päivitysten sekä aikataulujen suhteen tämän tutkimuksen ja opiskelijoiden antamien palautteiden perusteella. On mielenkiintoista tutkia, onko nyt tehdyistä muutoksista todella hyötyä opiskelijoille ja auttavatko ne keskitasoista opiskelijaa selviytymään opinnoissaan paremmin.

Kolmanneksi kiinnostavaksi kysymykseksi nousi minkä verran keskeyttämiseen vaikutti uusi virtuaalinen opiskelutapa. Ehkäpä tutkimuksen voisi toteuttaa niin, että varsinaisista lähiopetuksessa olevista yliopisto-opiskelijoista oleva vertailuryhmä tekisi saman materiaalin läpi kuin virtuaaliapprobatur-opiskelijat mutta kuitenkin niin, että vaikeina koettujen asioiden kohdalla vertailuryhmä saisi lisäksi lähiopetusta luentojen tai videoneuvotteluyhteyksien kautta ja toinen ryhmä opiskelisi täysin, tai ainakin lähes täysin, ilman lähiopetusjaksoja. Löytyisikö oppimistuloksissa eroavaisuuksia ja saataisiinko eri opiskelutapojen vaikutukset näin esille?

Yhtenä kehittämishankkeena kannattaa miettiä ohjauksellista interventiota. Millainen indikaattori voisi kertoa opettajalle missä vaiheessa opiskelijan olisi parempi keskeyttää opintonsa ja missä vaiheessa puolestaan kannattaa edelleen kannustaa oppilasta jatkamaan, vaikka hän olisi kokenut toistuvinkin epäonnistumisia opinnoissaan? Virtuaalinen oppimisympäristö on haasteellinen ympäristö opintojen järjestämiseen ja opiskelijan itsenäisestä roolista huolimatta opetukselliseen interventioon on tarvetta.

Lukuvuonna 2000-2001 approbatur-opiskelijat pääosin olivat kuitenkin tyytyväisiä kurssijärjestelyihin, joten kehittämistyötä virtuaaliapprobaturin parissa kannattaa jatkaa.

## LÄHTEET

Adorni, G., Barbieri, M. S., Bianchi, D., Calabrese, E. & Sugliano, A.M.: How to distribute learning facilities by means of a network: some issues and a case study. Teoksessa Verdejo, F. & Davies, G. (toim). *The Virtual Campus - Trends for higher education and training*. Chapman & Hall, Lontoo, 1998, ss. 211-224.

Anderson, J. M. & Naps, T. L.: A Context for the Assessment of Algorithm Visualization Systems as Pedagogical Tools. *Proceedings of the First Program Visualization Workshop*, July 7-8, Porvoo, Finland, 2001, ss. 121-130.

Ben-Bassat Levy, R., Ben-Ari, M. & Uronen, P.A.: An Extended Experiment with Jeliot 2000. *Proceedings of the First Program Visualization Workshop*. Yliopistopaino, Joensuu, 2001, ss. 131-140.

Carpone, A., Hurst, J., Mitchell, I. & Gunstone, D.: Principles for Designing Programming Exercises to Minimise Poor Learning Behaviours in Students. *Proceedings of the on Australasian computing education conference*, December 4-6, Melbourne, Australia, 2000, ss. 26-33.

Chute, A.G., Sayers, P.K. & Gardner, R.P.: Networked Learning Environments. Teoksessa Cyr, T.E. (toim): *Teaching and Learning at a Distance: What It Takes to Effectively Design, Deliver, and Evaluate Programs*. *New directions for teaching and learning*. Number 71. Jossey-Bass Publishers, San Francisco, 1997, ss. 75-83.

Collis, B.: New Wine and Old Bottles? Tele-Learning, Telematics and the University of Twente. Teoksessa Verdejo, F. & Davies, G. (toim). *The Virtual Campus - Trends for higher education and training*. Chapman & Hall, Lontoo, 1998, ss. 1-17.

Cordani, J.R. & Tucker, R.J.: Tools for Higher Education Distance Teaching. *Proceedings of the 26<sup>th</sup> SIGUCCS Conference on User Services*, October 25-28, Bloomington, USA, 1998, ss. 71-76.

Cornell, G. & Horstmann, C.: *Core Java*. SunSoft Press, Mountain View, CA, 1996.

Cornell, R. & Martin, B.L.: The Role of Motivation in Web-Based Instruction. Teoksessa Khan, B.H. (toim). *Web-Based Instruction*. Educational Technology Publications, New Jersey, USA, 1997, ss. 93-106.

Dhanarajan, G.: *Distance Education: promise, performance and potential*. Open Learning. The Journal of Open and Distance Learning, **16**(1): 61-68, 2001.

Ellis, A., Lowder, J., Robinson, J., Hagan, D., Doube, W., Tucker, S., Sheard, J. & Carbone, A.: A collaborative strategy for developing shared Java teaching resources to support first year programming. *Proceedings of the 4<sup>th</sup> annual SIGCSE/SIGCUE on Innovation and technology in Computer Science education*, June 27-30, Krakow, Poland, 1999, ss. 84-87.

Farrell, G. Introduction. Tutkimuksessa Farrell, G. et al. *The Development of Virtual Education: A global perspective*. Department for International Development, Lontoo, 1999. Internet WWW-sivu, URL: [http://www.col.org/virtualed/chapter1\\_intro.pdf](http://www.col.org/virtualed/chapter1_intro.pdf) (3.5.2001).

Haataja, A., Kontkanen, S., Suhonen, J. & Sutinen, E.: Teaching University-Level Computer Science to High School Students over the Web. *World Conference on Educational Multimedia, Hypermedia & Telecommunication*. June 25-30, Tampere, Finland, 2001.

Hong, J.: *The Use of Java as an Introductory Programming Language*. ACM Crossroads, 4.4. The ACM's First Electronic Publication, 1998. Internet WWW-sivu, URL: <http://www.acm.org/crossroads/xrds4-4/introjava.html> (29.8.2001).

Kashihara, A., Terai, A. & Toyoda J.: Making Fill-in-Blank Program Problems for Learning Algorithm. Teoksessa Cumming, G. (toim). *Advanced Research in Computers and Communications in Education*, IOS Press, Amsterdam, 1999, ss. 776-783.

Khuri, S.: Designing Effective Algorithm Visualizations. *Proceedings of the First Program Visualization Workshop*, July 7-8, Porvoo, Finland, 2001, ss. 1-12.

Klemm, W.: *Creating Online Courses: A Step-by-Step Guide*. The Technology Source, May/June 2001. Internet WWW-sivu, URL: <http://horizon.unc.edu/TS/default.asp?show=article&id=861>, (11.5.2001).

Korpi, M., Niemi, P., Ovaskainen, T., Siekkinen, P. & Junttila, V.: *Virtuaalinen oppimisympäristö koulutusta järjestävän organisaation työvälineenä*. Jyväskylän yliopisto. Tietotekniikan tutkimusinstituutti. Tietotekniikan tutkimusinstituutin julkaisuja 7/2000. HetiMonex Oy, Jyväskylä, 2000.

Kuusinen, K.: Motivaatio. Teoksessa Kuusinen, J. (toim): *Kasvatuspsykologia*. WSOY. Porvoo, 1995, ss. 191-224.

Kuusinen, J. & Korkiakangas, M. Oppiminen. Teoksessa Kuusinen, J. (toim): *Kasvatuspsykologia*. WSOY. Porvoo, 1995, ss. 23-68.

Landon, B., Bruce, R. & Harby, A.: *Features/Tools and Tech Info for WebCT*. Online educational applications. A web tool for comparative analysis. Centre for Curriculum, Transfer & Technology. Internet WWW-sivu, URL: [http://www.c2t2.ca/landonline/side\\_one.asp?appRow=10&appName=WebCT](http://www.c2t2.ca/landonline/side_one.asp?appRow=10&appName=WebCT) (18.9.2001).

Lawhead, P.B., Albert, E., Bland, C.G., Carswell, L., Cizmar, D., DeWitt, J., Dumitru, M., Fahraeus, E.R. & Scott, K.: The Web and distance learning: what is appropriate and what is not. Report of ITiCSE'97 Working Group on the Web and Distance Learning. SIGCSE/SIGCUE: *Integrating Technology into Computer Science Education ITiCSE'97*, Uppsala, Sweden, 1997, ss. 27-37.

Lee, M-G. (2000): Profiling students' adaptation styles in Web-based learning. *Computers & Education*. **36**(1):121-132, 2001.

Lehtinen, E. 1997. Tietoyhteiskunnan haasteet ja mahdollisuudet oppimiselle. Teoksessa Lehtinen, E. (toim). *Verkkopedagogiikka*. Oy Edita Ab, Helsinki, 1997, ss. 12-40.

Lischner, R.: Explorations: Structured Labs for First-Time Programmers. *The Proceedings of the Thirty Second SIGCSE Technical Symposium on Computer Science Education*, February 21-25, Charlotte, North Carolina, 2001.

Meisalo, V., Sutinen, E. & Tarhio, J.: *Modernit oppimisympäristöt - Tietotekniikan käyttö opetuksen ja oppimisen tukena*. Tietosanoma, Helsinki, 2000.

Meisalo, V., Suhonen, J., Sutinen, E. & Torvinen, S.: *Formative Evaluation Scheme for a Web-based Course Design*. (Käsikirjoitus julkaistavaksi) 2001.

Multsilta, J.: Miltä näyttää WWW-maailma oppimisympäristönä. Teoksessa Lehtinen, E. (toim). *Verkkopedagogiikka*. Oy Edita Ab, Helsinki, 1997, ss. 101-111.

Mäki-Komsi, S.: *Opettamisen ja oppimisen muodot muuttuvat, muuttuuko oppimiskulttuuri - heijastuksia opetuksen kehittämisprojektista OpinNetista*. 1999. Opetushallituksen julkaisuja. Internet WWW-sivu, URL: <http://www.edu.fi/julkaisut/opinnet1.pdf> (13.5.2001).

Opetusministeriö: *Koulutuksen ja tutkimuksen tietostrategia 2000-2004*. Internet WWW-sivu, URL: <http://www.minedu.fi/tietostrategia/tietostrategia.html> (23.8.2001).

Pollard, C. J. & Pollard, R. R. (1993). Restructuring the Teacher/Student Relationship Through Technology. Teoksessa Estes, N. & Thomas, M. (Toim.) *Rethinking the Roles of Technology in Education*. The Tenth International Conference on Technology and Education. March 21-24, 1993. Massa-

chusetts Institute of Technology, Cambridge, Massachusetts. Volume 1. Austin, TX: The University of Texas at Austin, College of Education, ss. 10-12.

Roberts, E.: An Overview of MiniJava. *The Proceedings of the Thirty Second SIGCE Technical Symposium on Computer Science Education*. February 21-25, Charlotte, North Carolina, 2001, ss.1-5.

Stasko, J., Badre, A. & Lewis, C.: Do algorithm animations assist learning: An empirical study and analysis. *The Proceedings of the INERCHI '93 Conference on Human Factors in Computing Systems*. Amsterdam, 1993, ss. 61 -66.

Suhonen, J., keskustelut virtuaaliapprobaturin toteutuksesta lukuvuonna 2000-2001, Joensuun yliopisto, 2001.

Suojanen, U.: Toimintatutkimus ammatillisen kehittymisen välineenä. Metodix. Menetelmät. Internet WWW-sivu. URL:

<http://www.metodix.com/showres.dll/fi/metodit/methods/metodiartikkelit/toimintatutkimus/> (18.9.2001).

Tella, S. *Uusi tieto- ja viestintäteknikka avoimen oppimisympäristön kehittäjänä. Osa 1.* Helsingin yliopiston opettajankoulutuslaitoksen tutkimuksia 124. Yliopistopaino, Helsinki, 1994.

Tiffin, J. & Rajasingham, L.: *In Search of the Virtual Class. Education in an information society.* Routledge, London, 1995.

Turner, J.A. & Zachary, J.L.: Javiva: A Tool for Visualizing and Validating Student-Written Java Programs. *The Proceedings of the Thirty Second SIGCSE Technical Symposium on Computer Science Education*, February 21-25, Charlotte, North Carolina, 2001, ss. 45-49.



Watanabe, S., Nakabayashi, T., Satoh, H., Jiang, T. & Oda, T.: Web-based Educational System: Monitoring and Assisting Learners. Teoksessa Cumming, G. (toim) *Advanced Research in Computers and Communications in Education*. IOS Press, Amsterdam, 1999.

Wolz, U., Palme, J., Anderson, P., Chen, Z., Dunne, J., Karlsson, G., Laribi, A., Männikkö, S., Spielvogel, R. & Walker, H.: Computer-mediated communication in collaborative educational settings. Report of ITiCSE'97 Working Group on CMC in Collaborative Educational Settings. SIGCSE/SIGCUE: *Integrating Technology into Computer Science Education ITiCSE'97*, Uppsala, Sweden, 1997, ss. 51-68.

# **Opiskeluopas**

**Joensuun yliopiston  
Tietojenkäsittelytieteen laitoksen lukio-opetus**

**Tietojenkäsittelytieteen approbatur-opintokokonaisuuden virtuaaliopetus  
2000 - 2001**

# SISÄLLYSLUETTELO

<b>JOHDANTO</b>	<b>3</b>
<b>OPPIMISYMPÄRISTÖ</b>	<b>4</b>
<b>OPISKELU</b>	<b>6</b>
<b>KURSSIEN SISÄLTÖ</b>	<b>7</b>
<b>KURSSIEN AIKATAULU</b>	<b>9</b>
<b>TEHTÄVIEN TEKEMINEN</b>	<b>10</b>
<b>OHJAUS JA NEUVONTA</b>	<b>11</b>

## JOHDANTO

Tämä opas on tarkoitettu tietojenkäsittelytieteen virtuaaliopetukseen osallistuville oppilaille. Oppaan selvitetään mitä virtuaaliopiskelu on ja mitä kurssien suorittaminen etäopiskeluna merkitsee oppilaan kannalta katsottuna.

Ensimmäiseksi oppaassa esitellään lyhyesti käytettävä oppimisympäristö, sen välineet ja työskentely oppimisympäristössä. Sen jälkeen käydään läpi kurssisisällöt sekä kurssien suorittamiseen ja tehtävien palauttamista koskevat ohjeet.

## OPPIMISYMPÄRISTÖ

Virtuaaliapprobatur on rakennettu **WebCT-ohjelmistoon** pohjautuvaan ympäristöön.. Tässä oppaassa on vain lyhyt esittely oppimisympäristöstä, koska tarkemmat ohjeet löydät ympäristössä olevasta opaskirjasta. Virtuaaliapprobaturiin osallistuvat oppilaat saavat yksilölliset käyttäjätunnukset, joiden avulla he pääsevät kirjautumaan kulloinkin meneillään olevalle kurssille. Käyttäjätunnukseksi saat virtuaaliapprobatur-opintojen tutor-opetajaltasi.

Ensimmäiset kurssit ovat näkyvillä **ke 30.8.2000 klo 12:00** osoitteessa:

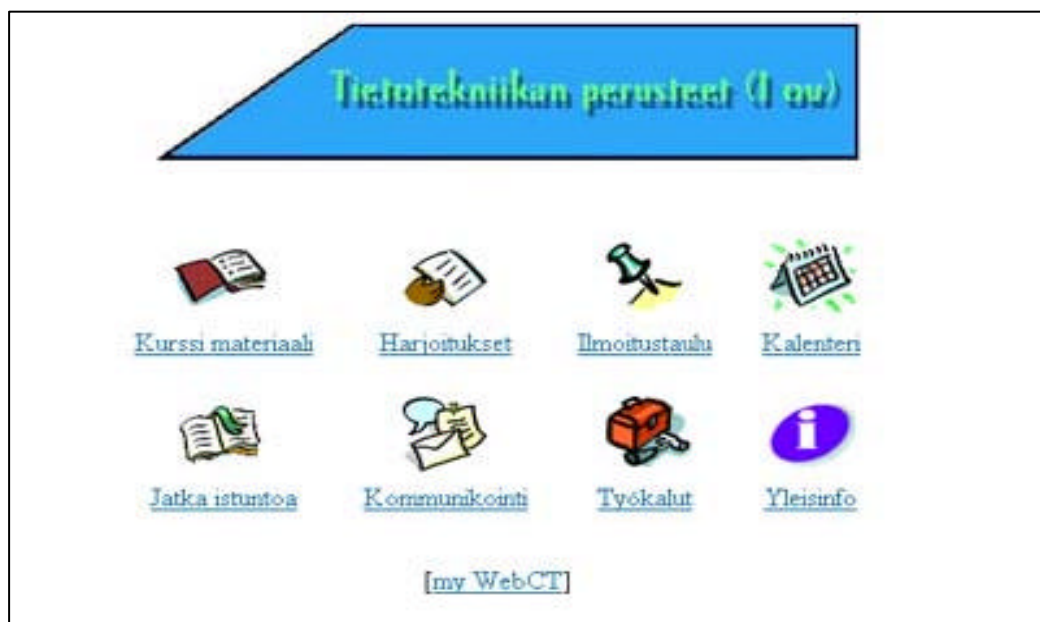
<http://cs.joensuu.fi:8900/>

Kurssien aikataulut löytyvät [www-sivulta](http://www.joensuu.fi):

<http://cs.joensuu.fi/lukioyt/virtual.htm#Approbatur>

Oppimisympäristössä käytössäsi ovat mm.

- Kurssi materiaalit:
  - ◆ Kurssin teoriasivut (tärkeimmät aiheet)
  - ◆ ”Viikottain” ilmestyvät tehtävät
- Keskustelufoorumi (ilmoitustaulu)
- Kommunikointivälineet:
  - ◆ Oppimisympäristön sisäiseen käyttöön tarkoitettu sähköposti
  - ◆ Neljä Chat-huonetta
  - ◆ WhiteBoard (elektroninen ”liitutaulu”)
  - ◆ Oma esittelysivu (kurssilaisen kotisivu)
- Kalenteri (henkilökohtainen + yleiset ilmoitukset)



**Kuva 1:** Tietokone, käyttöjärjestelmä ja tietoverkot -kurssin etusivu

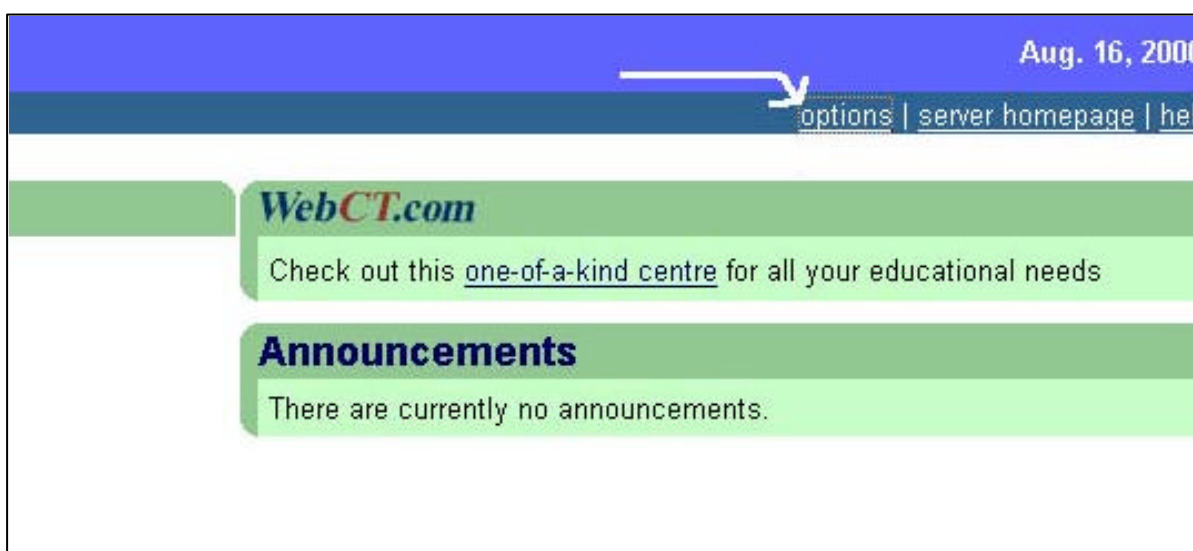
**HUOM!** Sähköposti, keskustelufoorumi ja kalenteri toimivat kurssikohtaisesti eli jokaisella kurssilla on omat ”versionsa”. Joten kaikki yhteydenpito tapahtuu aina kurssin sisäisesti, eikä ympäristössä ole tarjolla esimerkiksi sähköpostia, joka olisi kaikille kursseille yhteinen.

**Yleistä opiskelusta WebCT-ympäristössä:**

WebCT –oppimisympäristöä käytetään internetiin kytkettyjen tietokoneiden www-selaimilla. Näin ollen tarvitsen internet-yhteydellä varustetun tietokoneen ja www-selaimen (esim. Netscape Navigator, Netscape Communicator tai Microsoft Internet Explorer). Selaimen version tulee olla vähintään 3.\*, mieluummin 4.\*. Jos aiot opiskella modeemiyhteydellä, modeemin siirtonopeuden tulisi olla vähintään 14 400 bps.

**Salasanan vaihto:**

Kun otat ensimmäistä kertaa yhteyden osoitteeseen <http://ujocs.joensuu.fi:8900/>, niin valitse linkki **myWebCT**. Kirjoittaudu sisään tutoriltasi saamalla käyttäjätunnuksella ja salasanalla. Kun myWebCT -näyttö latautuu kannattaa aluksi vaihtaa tutoriltasi saamasi salasana uudeksi omaksi salasanaksi. Tämä onnistuu MyWebCT-näytön oikeasta yläkulmasta (kts. kuva alla) olevan **Options**-linkin takaa.



**Kuva 2 Options valikon sijainti**

Options-linkin takaa valitse **Change Password**, jonka avulla pystyt vaihtamaan salasanasi haluamaksesi. Salasanan vaihtaminen on tärkeää, jotta käyttämäsi salasana pysyy vain sinun tiedossasi. Jos salasanan vaihdossa tulee ongelmia ota yhteyttä tutor-opettajaasi tai yliopistolle (yhteystiedot löytyvät tämän oppaan lopusta).

## OPISKELU

### Yleistä:

Tietojenkäsittelytieteen opiskelu on kurssimaista ja perustuu teoriaan, harjoituksiin ja harjoitustöihin. Virtuaaliapprobatur opinnoissa teoria osuus koostuu kurssikirjasta ja verkosta löytyvästä materiaalista. Kursseihin liittyvät harjoitukset (tehtävät) löytyvät oppimisympäristöstä (kts. Tehtävien tekeminen) ja harjoitusten tarkoituksena on ohjata ja antaa tukea teorian omaksumiseksi. Aktiivinen oppiminen edellyttää harjoitustehtävien ja -töiden tunnollista suorittamista.

Virtuaaliapprobatur-opinnot perustuvat itseopiskeluun, joten opiskelijan kannalta tärkeintä on varata viikkotasolla riittävästi aikaa teoriaan tutustumiseen ja harjoitusten tekoon. Yleensä tietojenkäsittelytieteen opinnoissa myöhempi oppiaines perustuu aikaisemmin opittuun, joten opintojen tunnollinen suorittaminen helpottaa työskentelyä jatkossa.

### Kurssien suorittaminen:

Pääosa kursseista suoritetaan tutustumalla ensin teoriaan ja tekemällä kurssiin liittyviä tehtäviä, jotka ilmestyvät oppimisympäristöön pääsääntöisesti viikoittain. Kurssiin liittyy yleensä tentti, jonka hyväksytyt suorittaminen edellyttää, että olet tehnyt **ennalta määritellyn määrän kurssilla olevista tehtävistä** (yleensä 1/3).

Tentti laaditaan kurssikirjan teoriaosuuksiin ja verkkomateriaaliin perustuen. Kurssiin liittyvät tehtävät kannattaa käydä läpi huolellisesti, sillä niistä saattaa olla hyötyä tentissä. Verkkomateriaalista ja kurssikirjasta löytyvät [www-linkit](#) muihin materiaaleihin ovat niitä varten, jotka haluavat lisätietoa aiheesta. Täten linkkien päästä löytyvien materiaalien osaamista ei tentissä vaadita (saattavat olla kuitenkin hyödyllisiä tehtäviä tekemiseen).

Seuraavat kurssit poikkeavat edellä esitellystä perusmenettelystä: Tietokoneen, työvälineohjelmistojen ja verkkopalvelujen käyttö, Tietojenkäsittely ja yhteiskunta sekä Perusopintojen harjoitustyö. Edellä mainittujen kurssien toteutuksesta ilmoitetaan myöhemmin oppimisympäristössä.

### Kaava kurssin suorittamiseksi:

1. Tutustu teoriaan
2. Tee teoriaan liittyvät tehtävät
3. Varmista, että olet tehnyt riittävän määrän tehtäviä
4. Lue tenttiin
5. Osallistu tenttiin

<b>KURSSIEN SISÄLTÖ</b>
-------------------------

<b>Nimi:</b>	<b>Tietokone, käyttöjärjestelmä ja tietoverkko</b>
<b>Laajuus:</b>	1 ov
<b>Sisältö:</b>	Tietokonelaitteistot, tietokoneohjelmat, käyttöjärjestelmät, tietoliikenne, Internet, lähiverkot, multimedia, tietoturva, tietotekniikan ergonomia ja tietotekniikan juridiikka.

Harjoituksia yhteensä 30 kpl, joista hyväksytyjä oltava vähintään 1/3 eli 10 kpl. Tenti 9.10.2000 koululla. Kurssista annetaan arvosana tentin perusteella. Arvostelu: 1-3.

**Oppimateriaali:** Verkkomateriaali.  
Kirja: J. Paananen: Tietotekniikan peruskirja, Teknolit Oy, 1999, s.7-254, 311-368.

<b>Nimi:</b>	<b>Tietokoneen, työvälineohjelmistojen ja verkkopalvelujen käyttö</b>
<b>Laajuus:</b>	1 ov
<b>Sisältö:</b>	Tietokoneen ja yleisimpien työvälineohjelmistojen sekä verkkopalvelujen perustaitojen oppiminen. Kurssin runkona ovat tietotekniikan ajokorttitutkinnon kuuden käytännön moduulin tiedot ja taidot.

Kurssivaatimuksena on lukioissa järjestettävien näyttökokeiden hyväksyttävä suorittaminen. Arvostelu: hyväksyty / hylätty.

**Oppimateriaali:** Verkkomateriaali.  
Kirja: J. Paananen: Tietotekniikan peruskirja, Teknolit Oy, 1999, s. 369-418.

<b>Nimi:</b>	<b>Tietojenkäsittely ja yhteiskunta</b>
<b>Laajuus:</b>	1 ov
<b>Sisältö:</b>	Tietotekniikan vaikutukset yhteiskuntaan, tietoyhteiskunta, tietotekniikka organisaatioissa ja kotitalouksissa.

Kurssi suoritetaan kirjoittamalla aiheesta essee. Arvostelu: 1-3.

**Oppimateriaali:**

<b>Nimi:</b>	<b>Algoritmien suunnittelu</b>
<b>Laajuus:</b>	1ov
<b>Sisältö:</b>	Algoritmien suunnittelu. Algoritmit, ohjelmat ja ohjelmointikielät, algoritmien rakentaminen, ohjelman perusrakenteita, rekursio, tietorakenteet (listarakenne, taulukot ja puut). Arvostelu: 1-3.

**Oppimateriaali:** Verkkomateriaali.  
Luentomoniste: J. Boberg: Johdatus tietojenkäsittelyyn, Painosalama Oy, 1999, s. 7-74.



<b>Nimi:</b>	<b>Tietokoneen rakenne ja käyttöjärjestelmä</b>
<b>Laajuus:</b>	2 ov
<b>Sisältö:</b>	Tietokoneen rakenne ja toiminta. Matemaattiset ja fysikaaliset perusteet, tietokoneen komponentteja, mikro-ohjelmointi, konekieli, kommunikointi, ohjelmointikielten kääntäminen, kääntäjän toiminta, käyttöjärjestelmät, tiedostojärjestelmät.  Tentti 9.4.2001 koululla. Arvostelu: 1-3.
<b>Oppimateriaali:</b>	Verkkomateriaali. Luentomoniste: J. Boberg: Johdatus tietojenkäsittelyyn, Painosalama Oy, 1999, s. 102-177.
<b>Nimi:</b>	<b>Tietojenkäsittelytieteen tutkimusaloja</b>
<b>Laajuus:</b>	2 ov
<b>Sisältö:</b>	Tietojenkäsittelytieteen tutkimusaloja: tekoäly, laskennan teoria yms. Ongelmat, algoritmit, tehtävän algoritmien ratkeavuus, kompleksisuus ja oikeellisuus. Arvostelu: 1-3.
<b>Oppimateriaali:</b>	Verkkomateriaali. Luentomoniste: J. Boberg: Johdatus tietojenkäsittelyyn, Painosalama Oy, 1999, s. 75-101 ja 178-199.
<b>Nimi:</b>	<b>Ohjelmointi, osa 1</b>
<b>Laajuus:</b>	2 ov
<b>Sisältö:</b>	Ohjelmoinnin ajattelutapa, ohjelmoinnin perusteet ohjelmointikielenä oliokeskeinen Java. Mitä on ohjelmointi, mikä on Java, ohjelmoinnin peruskäsitteitä, ohjelmoinnin rakenteita, olio-ohjelmointia. Graafisen ohjelmoinnin perusteita.  Itseopiskelua. Harjoitukset. Tentti 11.12.2000 yliopistolla sali M3 klo 10-12. Arvostelu:1-3.
<b>Oppimateriaali:</b>	Verkkomateriaali Kirja: A. Wikla: Ohjelmoinnin perusteet Javalla, OtaData, 1999, s. 1-48, 67-78, 97-118, 233-238, 255 (Lue.java)
<b>Nimi:</b>	<b>Ohjelmointi, osa 2</b>
<b>Laajuus:</b>	3 ov
<b>Sisältö:</b>	Ohjelmien laatimisen välineet, olio-ohjelmoinnin perusteet (luokka, alaluokka, ylläluokka, rajapinnat, poikkeukset), ohjelmointitekniikka. Tapahtumien käsittely.  Itseopiskelua. Harjoitukset. Tentti 14.5.2001 yliopistolla sali M3 klo 10-12. Arvostelu: 1-3.
<b>Oppimateriaali:</b>	Verkkomateriaali Kirja: A. Wikla: Ohjelmoinnin perusteet Javalla, OtaData, 1999, s. 49-66, 80-87, 120-207, 222-224, 228-232

<b>Nimi:</b>	<b>Perusopintojen harjoitustyö</b>
<b>Laajuus:</b>	2 ov
<b>Sisältö:</b>	Ohjelman laatiminen itsenäisesti annetusta aiheesta. Harjoitustyön ohjaajana toimii kurssin vastuuhenkilö yliopistolla. Oppilaat voivat saada sähköpostineuvontaa. Harjoitustyö tehdään Java-kieltä käyttäen. Arvostelu: 1-3.

<b>KURSSIEN AIKATAULU</b>
---------------------------

Kurssit suoritetaan seuraavan aikataulun mukaisesti:

1 ov	Tietokone, käyttöjärjestelmä ja tietoverkko	30.8.2000-9.10.2000 (1. jakso)
2 ov	Ohjelmointi, osa 1	30.8.2000-11.12.2000 (jaksot 1-2)
1 ov	Tietokoneen, työvälineohjelmistojen ja verkkopalvelujen käyttö	10.10.2000-11.12.2000 (2. jakso)
2 ov	Tietokoneen rakenne ja käyttöjärjestelmä	Kevät 2001 (jaksot 3-4)
3 ov	Ohjelmointi, osa 2	Kevät 2001 (jaksot 3-5)
2 ov	Perusopintojen harjoitustyö	Kesä 2001
1 ov	Algoritmien suunnittelu	Syky 2001 (1. jakso)
2 ov	Tietojenkäsittelytieteen tutkimusaloja	Syky 2001 (jaksot 1-2)
1 ov	Tietojenkäsittely ja yhteiskunta	Syky 2001 (2. jakso)

## TEHTÄVIEN TEKEMINEN

Tehtävien tekeminen on oleellinen osa virtuaaliapprobatur-kurssien suorittamista. Tehtävien avulla voit testata opiskelemasi teorian osaamista käytännössä. Etenkin ohjelmointikursseilla tehtävien ratkomisesta (edes yrittäminen) on hyötyä asian ymmärtämisessä.

### Yleistä

Koska kurssin suorittamiseksi edellytetään, että olet tehnyt tietyn määrän harjoituksia, niin tehdyistä tehtävistä pidetään kirjaa. Harjoitusten määrä vaihtelee kurssikohtaisesti. Kurssin aikana voit seurata tekemiesi tehtävien määrää, jolloin olet koko ajan selvillä vaadittavien tehtävien määrästä. Viikoittain harjoitustehtävien määrä saattaa vaihdella. Hyväksyttävän merkinnän saa tehtävästä, josta näkee, että sitä on todella yritetty tehdä (vastausten ei tarvitse olla täysin oikein).

Voit tehdä kunkin viikon harjoitustehtävistä ne, mitkä tunnet osaavasi ja palauta ne annettuun palautuspäivämäärään mennessä oppimisympäristöön. Kurssin vastuuhenkilö yliopistolta tarkistaa tehtävät ja kommentoi niitä tarpeen mukaan. Näin sinulla on mahdollisuus saada palautetta tekemästäsi työstä.

Palautusajan jälkeen kurssin sivuilla julkaistaan harjoitustehtävien esimerkkivastaukset, joita voit tarkastella. Huomattavaa on, että joskus esimerkkivastaus on vain esimerkkivastaus, ts. vastauksesi saattaa olla ihan oikein, vaikka se onkin tehty eri tavalla. Kaikkien tehtävien vastaukset ovat esillä mahdollisimman ajoissa ennen tenttiä, jotta tehtäviä voi käyttää hyväksi tenttiin valmistautumisessa.

### Tehtävien pisteytys

Jokainen tehtävä on yhden pisteen arvoinen, jonka saa, kun kurssin vastuuhenkilö on hyväksynyt tehtävän suoritetuksi. Tehtävästä ei saa pistettä, kun joko tehtävään ei ole vastattu tai huomataan ettei tehtävää ole yritettykään ratkaista. Esimerkiksi Tietokone, käyttöjärjestelmä ja tietoverkko -kursseilla on yhteensä 30 tehtävää, joten tehtävistä saatu maksimipistemäärä on täten 30 pistettä.

Kuten aikaisemmin on mainittu tehtävistä täytyy olla hyväksyttynä tietty ennalta ilmoitettu määrä (yleensä 1/3), joten se on minimimäärä tenttiin osallistumisen varmistamiseksi. Lisäksi hyväksytyistä tehtävistä saat bonuspisteitä tenttiä varten. Bonuspisteiden määrät vaihtelevat kurssista toiseen, mutta seuraava taulukko esittää Tietokone, käyttöjärjestelmä ja tietoverkko -kurssin tehtävistä saatavat bonuspisteet:

Tehtäviä hyväksyttynä	Bonus (pistettä)
10-12	0,5
13-15	1
16-18	1,5
19-21	2
22-24	2,5
25-30	3

Oppimisympäristöstä löydät tarkemmat ohjeet sekä tenttien pisteytyksestä että edellä esitellystä pisteytysjärjestelmästä. Yleensä maksimi bonuspistemäärä on 1/10-osa tentin maksimipistemäärästä.

## **OHJAUS JA NEUVONTA**

Toivomme, että ongelmatilanteissa käännyt ensin tutor-opettajasi puoleen. Hän yrittää selvittää ongelman itse tai ottaa yhteyttä kurssien vastuuhenkilöihin. Myös oppimisympäristöstä löytyvät **WebCT:n käyttöopas** ja **FAQ-lista** saattavat antaa ratkaisun ongelmaasi.

Voit tuki ottaa yhteyttä tarvittaessa kurssien vastuuhenkilöihin:

**puhelin:** 013 – 251 2806 (maanantaisin 13-16)

**E-mail:** [appro@cs.joensuu.fi](mailto:appro@cs.joensuu.fi) (virka-aikana)

JOENSUUN YLIOPISTO  
Matemaattis-luonnontieteellinen tiedekunta  
Tietojenkäsittelytieteen laitos  
PL 111  
80101 JOENSUU

12.4.2001

«Koulun\_nimi»  
«Tutoropettaja»

### **KYSELY VIRTUAALIAPPROBATUR-OPINNOT KESKEYTTÄNEILLE OPISKELJOILLE**

Teen Joensuun yliopiston tietojenkäsittelytieteen laitokselle pro gradu -tutkielmaani liittyvää tutkimusta oppilaista, jotka ovat keskeyttäneet virtuaaliapprobaturiopintonsa kuluneen lukuvuoden aikana. Tutkimuksen tarkoituksena on löytää tekijät, jotka ovat vaikuttaneet oppilaiden keskeytyspäätökseen ja pyrkiä näin kehittämään virtuaaliapprobaturia ja erityisesti sen ohjelmointikursseja eteenpäin niin, että jatkossa yhä useampi opiskelija saisi arvosanasuorituksen. Gradussani käsitelen myös ohjelmoinnin problematiikkaa.

Oheisena on kyselylomakkeet jaettavaksi niille «Koulun\_nimi»n oppilaille, jotka ovat keskeyttäneet virtuaaliopintonsa. Koen tärkeänä, että tutkimuksessa olisivat mukana myös ne oppilaat, jotka ilmoittautuivat syksyllä opiskelijoiksi, mutta jotka eivät jostakin syystä pystyneetkään jatkamaan opintojaan pidemmälle.

Pyydän Teitä myös ystävällisesti kokoamaan kyselylomakkeet oppilailta sekä palauttamaan ne oheisessa palautuskuoressa *perjantaihin 27.4.2001 mennessä* allekirjoittaneelle.

Yhteistyöstä kiittäen

Sirpa Kontkanen  
Tutkimusamanuenssi  
Joensuun yliopisto  
Puh. 013 – 251 5272 tai 040 – 734 7175  
e-mail: skontka@cs.joensuu.fi

### **LITTEET**

Kyselylomakkeet  
Palautuskuori

## SAATE

Teen Joensuun yliopiston tietojenkäsittelytieteen laitokselle pro gradu-tutkielmaa, joka käsittelee virtuaaliapprobaturiin liittyen ohjelmoinnin problematiikkaa sekä syitä, jotka ovat johtaneet virtuaaliopintojen keskeyttämiseen. Olen pyytänyt koulunne tutoropettajaa jakamaan oheiset kyselylomakkeet niille virtuaaliapprobatur opiskelijoille, jotka ovat keskeyttäneet opintonsa joko aivan heti opintojen alussa syksyllä 2000 tai jossakin vaiheessa lukuvuoden aikana.

Pyydän Sinua ystävällisesti täyttämään oheisen kyselylomakkeen ja palauttamaan sen täytettynä tutor-opettajallesi. Antamasi palaute on ensiarvoisen tärkeää, jotta pystymme kehittämään virtuaaliapprobaturia ja sen kursseja eteenpäin. Kaikki vastaukset käsitellään luottamuksellisesti eikä nimesi tule missään vaiheessa julki.

Toivomme lisäksi, että pystyisit jatkamaan approbatur-opintoja suorittamalla tentit yliopiston yleisenä tenttipäivänä, joka järjestetään kesällä.

Yhteistyöstäsi kiittäen

Sirpa Kontkanen  
Tutkimusamanuenssi  
Joensuun yliopisto  
Tietojenkäsittelytieteen laitos  
Puh. 013 – 251 5272 tai 040 – 734 7175  
e-mail: skontka@cs.joensuu.fi

**KYSELYLOMAKE**

**Taustatietoja:**

Vastaajan nimi: \_\_\_\_\_  
Lukion nimi: \_\_\_\_\_

Olen aloittanut lukio-opinnot  v. 2000  v. 1999  v. 1998  v. 1997

Ennen virtuaaliopintojen alkamista, osasin:

- käyttää VBA-makroja
- tehdä yksinkertaisia pelejä
- tehdä Java-appletteja
- tehdä kyselykielellä kyselyjä tietokannasta
- tehdä kotisivuja html-kielellä
- en mitään edellä mainituista
- muuta, mitä? \_\_\_\_\_

Jos sinulla on aiempaa ohjelmointikokemusta, niin mistä ohjelmointikielistä?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Kysymykset:**

Arvioi seuraavia väittämiä käyttäen asteikkoa 1-5 (missä 1 = täysin eri mieltä , 2 = jokseenkin eri mieltä , 3 = en osaa sanoa, 4 = jokseenkin samaa mieltä, 5 = täysin samaa mieltä ).

**1. Ohjelmointi 1- kurssimateriaali ja sen käytettävyys**

<i>Arvioitava osa-alue</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
WebCT:n teoriasivut toimivat hyvin oppikirjan tukena	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teoriasivuilta löytyi tarvittava tieto helposti	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teoriasivut olivat rakenteiltaan johdonmukaiset ja selkeät	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Harjoitustehtäviä oli riittävästi asian oppimisen kannalta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Harjoitustehtävät kattoivat hyvin kurssin eri osa-alueet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**2. Oppimisympäristön tekninen hallinta**

<i>Arvioitava osa-alue</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Oppimisympäristöä (WebCT) oli helppo käyttää	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Oppimisympäristö (WebCT) oli rakenteeltaan selkeä	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Harjoitustehtävien palauttaminen sujui helposti	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Malliratkaisut olivat helposti saatavilla	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Java-kääntäjän ja -tulkin asentaminen sujui ongelmitta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Java-kääntäjää ja -tulkkia oli helppo käyttää	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**KYSELYLOMAKE**

**3. Ohjauksen saaminen ongelmatilanteissa**

<i>Arvioitava osa-alue</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Yliopistolta saamani ohjaus auttoi harjoitustehtävien ratkaisemisessa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Yliopistolta saamani ohjaus auttoi seuraavien viikkojen harjoitus-tehtävien ratkaisemisessa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tehdyistä harjoitustehtävistä saatu palaute auttoi ymmärtämään vaikeita kohtia	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Käytin hyväkseni WebCT:n keskustelupalstaa kysyäkseni neuvoja kurssikavereilta eri kouluilta ja sain sieltä apua	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Seurasin WebCT:n keskustelupalstaa ja löysin toisten oppilaiden tekemistä kysymyksistä ja niihin tulleista vastauksista ratkaisun myös itseäni askarruttaviin kysymyksiin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Keskustelin omalla koulullani olevien kurssikavereiden kanssa ja sain heiltä apua ongelmallisissa tehtävissä tai tilanteissa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Harjoitusten jälkeen annetut malliratkaisut auttoivat ymmärtämään omassa ratkaisussani mahdollisesti olleet ongelmat ja virheet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**4. Arvioi seuraavien ohjelmoinnin osa-alueiden vaikeutta asteikolla 1-5 (1= helppo, 2= melko helppo, 3= en osaa sanoa, 4 = melko vaikea, 5= vaikea).**

<i>Arvioitava osa-alue</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Muuttujien ja tunnusten käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sijoitusoperaatioiden sekä syöttö- ja tulostuslauseiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Toistolauseiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Valintalauseiden ja loogisten operaatioiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Taulukoiden ja satunnaislukujen käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Metodien käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applettien ja grafiikan käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Animaatioiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



**KYSELYLOMAKE**

5. Arvioi olisiko seuraavista opetusmuodoista hyötyä ohjelmoinnin eri osa-alueiden oppimisessa. Mikäli et katso olevan vastaavaa hyötyä, jätä asianomainen kohta tyhjäksi.

<i>Arvioitava osa-alue</i>	<i>Luento Joensuun yliopistolla</i>	<i>Video- neuvottelu yliopiston ja lukioiden välillä</i>	<i>Ohjattu keskustelu WebCT:ssä</i>
Muuttujien ja tunnusten käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sijoitusoperaatioiden sekä syöttö- ja tulostuslauseiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Toistolauseiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Valintalauseiden ja loogisten operaatioiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Taulukoiden ja satunnaislukujen käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Metodien käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applettien ja grafiikan käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Animaatioiden käyttö	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

6. Missä vaiheessa keskeytit approbatur-opintosi (laita rasti keskeytysajankohtaan)?

Syyskuu	Lokakuu	Marraskuu	Joulukuu	Tammikuu	Helmikuu	Maaliskuu
Virtuaaliopin- tojen aloitus Ohjelmointi1- kurssin aloitus	Tietokone, käyttöjärjestelmä ja tietoverkko -kurssin tentti		Ohjelmointi1 -kurssin tentti	Ohjelmointi2 -kurssin aloitus Ohjelmointi1 -kurssin uusintatentti		

7. Mikä oli pääsyy keskeyttämiseesi (voit myös numeroida useammankin syyn käyttäen järjestystä 1= tärkein syy, 2 = toiseksi tärkein jne):

- aikapula  
 osallistuin kevään yo-kirjoituksiin  
 aion jatkaa opintoja ensi lukuvuonna  
 teoria tuntui liian vaikealta  
 tehtävät tuntuivat liian vaikeilta  
 en läpäissyt Ohjelmointi1-kurssin varsinaista tenttiä  
 en läpäissyt Ohjelmointi1-kurssin uusintatenttiä  
 tuli muutoksia opiskelutilanteeseeni  
 kurssisisältö ei vastannut ennakko-odotuksiani  
 kurssi ei kiinnostanut  
 muu syy, mikä?

**KYSELYLOMAKE**

Liite 2 (6)  
4(4)

8. Mitä muuttaisit ohjelmointikurssista? Ottaisitko jotakin pois tai haluaisitko jotakin lisää?

---

---

---

---

---

---

---

---

---

---

9. Olisiko sinulla muita toiveita kurssimateriaalin tai käytännön teknisen toteutuksen parantamiseksi?

---

---

---

---

---

---

---

---

---

---

10. Oletko halukas myös jatkotutkimukseen asian tiimoilta (esim. haastattelu tai jatk okysely)?

Kyllä olen, sähköpostiosoitteeni:

En ole

***Kiitos Sinulle vastauksestasi!***  
*Jokainen annettu parantamisasiidea auttaa  
meitä työssämme kurssien kehittämisessä.*

Joensuun yliopisto /tietojenkäsittelytieteen laitos 11.12.2000

Tentin laatinut: Anja Kareinen / normaalikoulu

Tarkastavat: Arto Haataja, Jarkko Suhonen

### Ohjelmoinnin perusteet

1) Ystäväsi osaa vähän Java-ohjelmointia. Hän tietää, mikä on muuttuja, vakio sekä osaa käyttää syöttö- ja tulostuslauseita. Hän ei kuitenkaan ymmärrä if -lauseen merkitystä. Laadi vastaus, jossa selität hänelle if -lauseen rakenteen erilaisissa tilanteissa. Käytä esimerkkejä selittäessäsi.

6 p

2)

Tee (itsenäinen) Java-**ohjelma**, joka kysyy 10 lukua sekä tulostaa lukujen summan.

Edellä mainitun lisäksi ohjelman on tulostettava annetuista luvuista pienin.

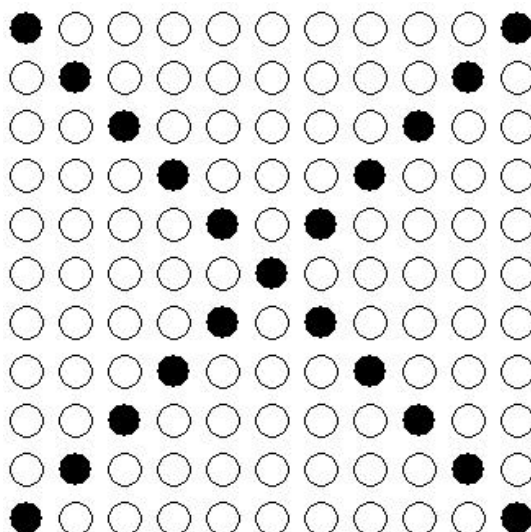
6 p

3) Tee metodi, joka **arpoo** 1000 lukua väliltä 0-9, **laskee ja myös tulostaa** kuinka monta kertaa kukin luku on arvottu. **Huom.** Reaaliluvun välillä  $0 \leq \text{luku} < 1$  saat seuraavalla käskyllä:

```
double luku1=Math.random();
```

6 p

4) Tee **appletin paint -metodi**, joka tulostaa 121 ympyrää. 11 ympyrää / rivi (11 riviä) seuraavan kuvan mukaisesti. Esitä myös muuttujien määrittelyt.



6 p

Hyvää joulua

# Luntilappu (Ohjelmointi 1)

## Itsenäinen ohjelma

```
public class Tervetuloa {  
    public static void main(String [] args){  
  
        System.out.println("Tervetuloa opiskelemaan Javaa!");  
    }  
}
```

## Grafiikkakäskyjä

```
g.drawString("teksti", 50,100);  
g.drawLine(10,20,50,70);  
g.drawRect(20,30,100,300);  
g.drawOval(50,50,200,200);  
g.setColor(Color.blue);  
g.fillRect(20,30,100,300);  
fillOval(50,50,200,300);  
drawArc(50,50,100,100,0,100)
```

### **Kuvan lisäämiseksi tarvitaan:**

```
Image kuva1;  
kuva1=getImage(getDocumentBase(),"taulu.gif");  
g.drawImage(kuva1,100,100,this);
```

## Animaation toteuttaminen

-tästä voit katsoa myös appletin, jossa ei ole animaatioita rakenteen

```
import java.awt.*;
import java.applet.*;

public class Malli extends Applet implements Runnable {

    Thread saie=null;

    public void start() {

        if (saie==null) {
            saie = new Thread(this);
            saie.start();
        }

    }

    public void paint(Graphics g){ }

    public void run() {
        while (true) {
            try {
                Thread.sleep(100);
            }
            catch (InterruptedException e){ }
        }
    }

}
```