



OHJELMOINNIN PERUSTEIDEN KEHITTÄMINEN
TIETOJENKÄSITTELYN PERUSTUTKINNOSSA

Jouni Aho

**Kehittämishankeraportti
Toukokuu 2008**



**JYVÄSKYLÄN
AMMATTIKORKEAKOULU**
Ammatillinen opettajakorkeakoulu

Tekijä(t) Jouni Aho	Julkaisun laji Kehittämishankeraportti	
	Sivumäärä 24	Julkaisun kieli suomi
	Luottamuksellisuus <input type="checkbox"/> Salainen _____ saakka	
Työn nimi Ohjelmoinnin perusteiden kehittäminen tietojenkäsittelyn perustutkinnossa		
Koulutusohjelma Jyväskylän ammattikorkeakoulu, Ammatillinen opettajakorkeakoulu, Ammatillinen opettajankoulutus		
Työn ohjaaja(t) Kinnunen, Heli		
Toimeksiantaja(t)		
<p>Tiivistelmä</p> <p>Kehittämishankkeen tavoitteena on kehittää ohjelmoinnin perusteiden opetusta tietojenkäsittelyn perustutkinnossa. Hankkeessa määrittelen ohjelmoinnissa käsiteltävät aihealueet sellaisella tarkkuudella, että niitä voidaan suoraan soveltaa opetukseen. Valitut aihealueet ovat sellaisia, että niitä voidaan soveltaa myös käytettäessä jotain muuta kuin tässä hankkeessa valittua ohjelmointikieltä.</p> <p>Hankkeessa käsitellään ohjelmoinnin integroitumista muihin aihealuetta ympäröiviin oppiaineisiin. Kehittämishankkeeseen kuuluu myös käytettävien opetusvälineiden, kuten ohjelmistojen valinta. Tavoitteena ohjelmistojen valinnassa oli niiden helppo saatavuus ja maksuttomuus, jolloin opiskelijoilla on hyvät edellytykset suorittaa kurssilla annettavia tehtäviä myös muualla kuin koulussa.</p> <p>Kehittämishanke ottaa kantaa myös opiskelijoiden motivointiin sekä kurssin arviointiin. Motivoinnin välineenä käytetään harjoitustehtävien suoraa vaikutusta kurssista saatavaan arvosanaan. Arviointimenetelmästä olen saanut opiskelijoiltani hyvää palautetta.</p> <p>Pedagogisena lähtökohdانا oli käyttää eri oppimiskäsityksiä soveltuvin osin ja sekoittaa niihin myös omien kokemusten tuomia ajatuksia ja ideoita. Mielestäni onnistuin tässä tavoitteideni mukaisesti.</p> <p>Tässä kehittämishankkeessa kohderyhmänä on ollut Jyväskylän aikuisopiston tietojenkäsittelyn perustutkintoa opiskelevat oppilasryhmät.</p>		
Avainsanat (asiasanat) Ohjelmointi, tietojenkäsittely, tutkintovaatimukset, php		
Muut tiedot		

Author(s) Jouni Aho	Type of Publication Development project report	
	Pages 24	Language Finnish
Confidential <input type="checkbox"/> Until _____		
Title Develop the basics of programming in the Vocational Qualification of Business Information Technology		
Degree Programme Jyväskylä University of Applied Sciences, Teacher education College, Vocational teacher education		
Tutor(s) Kinnunen, Heli		
Assigned by		
Abstract <p>The objective of this development project is to develop the teaching of the basics of programming in the Vocational Qualification of Business Information Technology. In the development project all the areas to be covered are defined with such accuracy that the results can be directly applied to teaching. The chosen topics can also be taught by using another programming language than the one used in this project.</p> <p>The development project discusses the integration of programming to the other related subjects that are covered in the curriculum. The project also includes the selection of teaching methods such as applications. The objective in selecting applications is to make them as accessible and affordable that students will have good possibilities to do their tasks outside the school as well.</p> <p>The development project also discusses the motivation of students and the evaluation of the course. As a motivating factor, the course tasks are directly related to the grade of the course. I have received good feedback about the evaluation method from the students.</p> <p>The pedagogic starting point was to use different conceptions of learning in suitable proportions and also mix in some ideas I have evolved through my own experience as a teacher. In my opinion I was able to meet the objectives.</p> <p>The target group of this development project has been the students who are completing the Vocational Qualification in Business Information Technology at the Jyväskylä Institute of Adult Education.</p>		
Keywords Computer programming, data processing, degree requirements, php		
Miscellaneous		

Sisällys

1	Kehittämistyön lähtökohdat ja tavoitteet.....	5
2	Ohjelmoinnin opetuksen pedagogisia lähtökohtia	6
2.1	Behavioristinen oppimiskäsitys	7
2.2	Humanistinen oppimiskäsitys	8
2.3	Kognitiivinen oppimiskäsitys	9
2.4	Konstruktivinen oppimiskäsitys	11
3	Ohjelmoinnin opintokokonaisuus	12
3.1	Www-tuottaminen.....	12
3.2	Ohjelmoinnin perusteet	13
3.3	Tietokantojen jatkokurssi.....	14
3.4	Tietoverkkosovellusten ohjelmointi	14
4	Ohjelmointikieli, PHP.....	15
5	Ohjelmoinnin perusteiden sisältö	16
6	Välineet.....	20
7	Arviointi	21
8	Opettajan rooli kehittämistyön tulokset	23
	Lähteet.....	24

1 Kehittämistyön lähtökohdat ja tavoitteet

Jyväskylän aikuisopistossa on viimevuosien aikana opetettu ohjelmoinnin perusteita Tietojenkäsittelyn perustutkintoa suorittaville aikuisopiskelijoille kokeilemalla eri ohjelmointikieliä, opetusmenetelmiä ja jopa eri opettajia. Hankkeen tarkoituksena oli selkiyttää tämän laajahkon kokonaisuuden toteuttamistapa ja menetelmät sekä kehittää siitä yhtenäinen ja mielekäs kokonaisuus sekä opettajalle että opiskelijalle. Tarve kehittämistyölle on siis todellinen ja hyvinkin akuutti. Tarve on tullut esiin niin esimiesteni kuin kollegoideni suunnalta.

Ohjelmoinnin opetuksesta ovat vastanneet Jyväskylän aikuisopistossa aiemmin opettajat, jotka eivät valitettavasti ole enää oppilaitoksen palveluksessa. Ohjelmoinnin perusteiden opetuksen olen ottanut itselleni vastaan uutena mielenkiintoisena haasteena.

Kouluttajan työni Jyväskylän aikuisopistossa on ollut erittäin monipuolista sekä vaihtelevaa. Olen ollut jo aiemmin jollain tavoin mukana (lähinnä tutorin ominaisuudessa) ohjelmoinnin perusteiden opettamisessa, joten minulla on jo melko kokonaisvaltainen näkemys kurssin tavoitteista ja eri opetusmenetelmien käyttömahdollisuuksista. Olen ollut mukana toteuttamassa opetusta sekä perinteisin menetelmin, että virtuaalisen oppimisympäristön kautta.

Kehittämishankkeen tavoitteena oli kehittää selkeä uusi toimintasuunnitelma ohjelmoinnin perusteiden opettamiseen, sekä nivoa se tiiviiksi paketiksi muiden aiheeseen liittyvien kurssien kanssa. Tavoitteena on siis tarjota opiskelijoille looginen eri aihealueet yhteen nitova opintokokonaisuus, jonka etenemistä tukevat aiemmat opinnot.

Ohjelmoinnin opettaminen ja oppiminen on haastava tehtävä varsinkin kun kohderyhmänä on hyvin hajanainen joukko aikuisopiskelijoita. Opiskelijoiden taustat ovat hyvin erilaisia. Osa on atk:ta harrastaneita nuoria miehiä tai naisia, joille tämän kaltaisen uuden asian oppiminen ei ole ongelma. Joukossa on kuitenkin aina myös henkilöitä, joille tietokoneen toiminta ja käyttö

ylipäättänsä ei ole niin itsestään selvää, saati tuttua. Opettajan tulisi kuitenkin huomioida nämä kaikki erilaiset oppijatyypit ja sopeuttaa opetus vastaamaan kaikkien kykyjä ja tarpeita. Tärkeimmäksi tavoitteeksi muodostuukin kaikkien erityyppisten oppijoiden motivoiminen ohjelmoinnin opiskeluun.

Yhtenä hankkeen tavoitteena oli saada koulutuskokonaisuudesta mahdollisimman hyvin valtakunnallisia opetussuunnitelman perusteita ja oppilaitoskohtaista toteuttamissuunnitelmaa vastaava, jolloin koulutuskokonaisuus olisi valtakunnallisestikin mahdollisimman vertailukelpoinen muiden tutkinnon toteuttajien kanssa.

Ohjelmoinnin opettaminen toteutetaan PHP-kielellä, joten siihen nivoutuu vahvasti ja luonnollisesti pohjatietona vaadittava www-tuottamisen kurssi, sekä kollegani toteuttama SQL-kurssi. Osalle opiskelijoista, suuntautumisvaihtoehdosta riippuen, tulee myös Vuorovaikutteisen verkkopalvelun toteuttamisen kurssi, jonka lähtötietoina edellytetään ohjelmoinnin perusteiden sekä SQL -kurssin suorittamista. Näistä osa-alueista opiskelijalle muodostuu selkeä kokonaisuus, jonka antamaa oppia voi hyödyntää kokonaisvaltaisesti erilaisten verkkosovellusten toteuttamisessa. Ei siis voida puhua yhden aineen opettamisesta tai oppimisesta vaan kokonaisuudesta, jossa yhdistyvät eri aihealueista opitut tiedot ja taidot.

Hankkeen aihealueista ja sitä ympäröivistä aiheista tällä hetkellä minä itse opetan www-palveluiden sekä ohjelmoinnin perusteiden osuudet.

Tietokantapalveluiden opetuksesta vastaa kouluttaja Rauno Kujala ja Vuorovaikutteisen verkkopalvelun kouluttajana on toiminut Ari Rantala.

2 Ohjelmoinnin opetuksen pedagogisia lähtökohtia

Oppimiskäsitykset ovat muuttuneet ja kehittyneet vuosien saatossa. Hankkeen aikana perehdyin erilaisiin oppimiskäsityksiin opintopiirimme aiheesta tekemän raportin pohjalta.

Opetustilanteessa tulee harvoin ajateltua oman opetuksen teoreettisia lähtökohtia, ennemminkin opetus soljuu ”maaston mukaan”. Toisin sanoen,

olen omassa opetuksessani pyrkinyt aina käyttämään kyseiselle ryhmälle tai opiskelijalle parhaiten sopivaa menetelmää. Menetelmät saattavat olla hyvin erilaisia eri tapauksissa.

Kysyttäessä mitä opetusmenetelmiä käytän ohjelmoinnin opetuksessa, vastaan mielelläni, että kaikesta vähän kaikkea. Käytetyt opetusmenetelmät voivat myös vaihdella paljon eri ryhmien välillä. Mielestäni ei ole tarkoituksen mukaista päättää, että nyt toimitaan tämän ja tämän oppimiskäsityksen aatteiden mukaan vaan on parempi valita tilanteeseen sopiva opetusmenetelmä oppimiskäsityksestä riippumatta. Käytäntö osoittaa parhaiten, kuinka opetuksessa on toimittu ja kuinka oppi on mennyt perille. Käytännön kokemusten ja aiemmin opittujen asioiden kautta uuden oppiminen on monesti paljon tuloksellisempaa, kuin tiedon systemaattinen pönttääminen tai ulkoa opettelu.

2.1 Behavioristinen oppimiskäsitys

1900-luvun alussa kehitettiin didaktiseen tiedonsiirron malliin perustuva behavioristinen oppimiskäsitys. Behavioristinen oppimiskäsityksen mukaan oppimista tapahtuu, kun tietoa siirretään opettajalta oppijalle. Oppija nähdään passiivisena tiedon vastaanottajana. Oppimista kontrolloidaan käyttämällä apuna palkkioita ja rangaistuksia (keppi ja porkkana). Oppimisen tutkimus perustuukin luonnontieteelliseen menetelmään, oletuksena että ihmisillä ja eläimillä oppiminen perusmuodoissaan ilmenee samanlaisena.

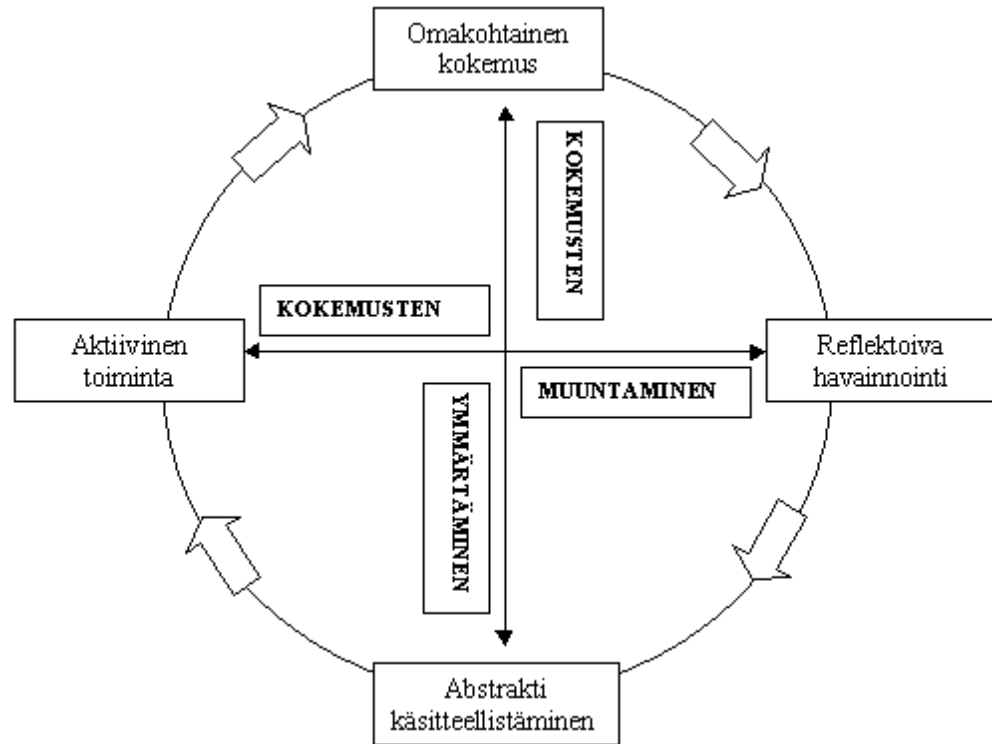
Behavioristisen oppimiskäsityksen ajatuksena on jakaa opetettava asia loogisiin osatekijöihinsä, ja sitten opettaa asiat yksi kerrallaan. Opiskelija nähdään tyhjänä saavina, johon opettaja pyrkii vähitellen kaatamaan tietoa. Opetettavaa ohjataan välittömällä palautteella, ja hän osoittaa oppineensa asian suorittamalla joitain täsmällisesti määrättyjä tehtäviä. Oppimisen tapahtuminen havaitaan käyttäytymisen muutoksena opetettavan oppiessa suorittamaan annettua tehtävää jollain etukäteen määrättyllä opettajan määrittelemällä "oikealla" tavalla. Kun opetettava osoittaa omaksuneensa yhden asian, siirrytään suunnitelmassa eteenpäin, askel kerrallaan. Tämän

koulukunnan edustajia ovat olleet mm. E.L. Thorndike, I. Pavlov, J.B. Watson ja B.F. Skinner.

2.2 Humanistinen oppimiskäsitys

Kokemuksellinen oppiminen on monipuolisesti koskettava ja aktivoiva toiminnallinen prosessi, jossa hyödynnetään eri aistikanavia, tunteita, elämyksiä, mielikuvia ja mielikuvitusta. Oppiminen on jatkuvaa tiedon syventämistä ja ymmärtämistä sekä kokemusten muuttumista ja laajentumista. Humanistiseen psykologiaan pohjautuvan pedagogisen toiminnan mallin mukaan toiminta tapahtuu vaiheittaisessa prosessissa, jossa ensin toimitaan ja koetaan. Tämän jälkeen arvioidaan kokemuksia ja lopuksi käsitteellistetään näin saadut uudet kokemukset, jotta niiden pohjalta pystyttäisiin luomaan uusi teoria, joka lopuksi testataan tai kokeillaan. Humanistisessa oppimiskäsityksessä painotetaan oppimisen sisäistä kontrollia.

Oppimisen katsotaan perustuvan hyvin pitkälti oppijan yksilöllisiin kokemuksiin ja opetuksen tehtävänä on tukea oppijan kasvua ja itseohjautuvuutta. Humanistisen oppimiskäsityksen ehkä suosituin teoretikko Kolb erottaa neljä ”oppimisen orientaatiota”. Kun nämä orientaatiot yhdistetään toisiinsa, tuloksena on kokemuksellisen oppimisen sykli: kokemukset – reflektointi – käsitteellistäminen – aktiivinen kokeilu.



Kuvio 1. Kokemuksellisen oppimisen malli (Kolb 1984, 42)

Kokemukset merkitsevät konkreettista kokemista, jossa korostuvat henkilökohtaiset kokemukset ja tunteet. Reflektoinnissa keskitytään kokemusten ja tilanteiden monipuoliseen havainnointiin. Käsitteellistämässä on ominaista systemaattinen ajattelu ja ongelmanratkaisu. Aktiivisessa kokeilussa korostuu käytännön toiminta ja ihmisiin tai tilanteisiin vaikuttaminen.

2.3 Kognitiivinen oppimiskäsitys

Kognitiivinen oppimiskäsitys kehittyi behaviorismin rinnalle 1950-luvulla. Katsottiin, että oppiminen voisi perustua yksilön havaintoihin, muistikuviiin, kieleen sekä ajattelun tuottamiin kokemuksiin.

Kognitiivisen oppimiskäsityksen syntyä voidaan omalta osaltaan pitää jonkinlaisena muutosehdotuksena behavioristisille näkemyksille oppimisesta. Oppimista pidetään tämän oppimiskäsityksen mukaan yhtenä ihmisen toimintaprosessina. Oppimisen ja opetuksen toimintaprosessien katsotaan

olevan saman prosessikokonaisuuden eri vaiheita. (Rauste - von Wright 1998, 104.)

Kognitiivisessa oppimiskäsityksessä ei ole määritelty kaikkia koskevaa yhteistä oppimisteoriaa, vaan oppi perustuu ennemminkin yksilön kokemuksiin erilaisista oppimistilanteista, joten oppijan oma toiminta ja opittavien asioiden kokeminen käsitetään keskeisiksi asioiksi oppimisessa.

Kognitiiviseen oppimiskäsitykseen kuuluu olennaisesti myös ajatus tiedon jatkuvasta muuttumisesta, toisin sanoen ei oleteta, että opittu asia on ja pysyy samanlaisena tilanteiden muuttuessa ja ennen kaikkea erilaisten oppijoiden kohdalla.

Kognitiivinen oppimiskäsitys on jättänyt jälkensä myös myöhemmin kehitettyihin oppimiskäsityksiin, esimerkiksi konstruktivismi pohjautuu joiltain osin kognitiiviseen oppimiskäsitykseen.

Keskeistä kognitiivisessa oppimisessa on myös oppijoiden metakognitiivisten taitojen kehittäminen. Metakognitiiviset taidot tarkoittavat oppimisen kriittistä arviointia ja arvioinnin pohjalta oman oppimisen ja oppimistapojen kehittämistä. Oppimisen kannalta mielestäni on ensiarvoisen tärkeää nimenomaan ymmärtää asiat, joita ei vielä hallitse ja näin ollen keskittää kaikki oppimisenergia olennaisiin aiheisiin ja asioihin. Este uusien asioiden oppimiselle usein onkin se, ettei tiedetä mitä kaikkea voisikaan vielä oppia.

Kognitiiviseen oppimiskäsitykseen kuuluu myös se, että oppija ei ole ainoastaan tietoa vastaanottava vaan myös tietoa käsittelevä ja jo opittua asiaa omien kokemustensa pohjalta prosessoiva henkilö.

Kognitiivisen oppimiskäsityksen mukaan ihmisen oma aktiivinen mentaalinen toiminta (tiedostettu tai tiedostamaton) oppimistapahtuman aikana on oppimisen kannalta oleellista. Oppimistyyliksi voidaan kutsua strategiaa, jota ihminen käyttää opeteltavan informaation hankintaan, käsittelyyn ja liittämiseen aiemmin opittuun tietoon. Koska kognitiivinen oppimiskäsitys pyrkii löytämään ihmisen oppimisprosessia kuvaavia yleisiä piirteitä, voidaan sen käsitteitä soveltaa kaikenlaisissa oppimisissa.

Oppimiskäsitys käytännössä

Tietotekniikan opiskelussa tulee usein tilanteita joissa oppijoille tarjotaan opittavaksi asioita, joiden omaksumiseksi tarvitaan jo aiemmin opetettuja asioita. Näin ollen oppija joutuu prosessoimaan yhteen aiemmin oppimiansa ja nyt opittavia asioita ja muodostamaan näiden tietojen pohjalta toimivia kokonaisuuksia ja ajattelumalleja. Kognitiivinen oppimiskäsitys siis tulee esiin usein tämänkaltaisissa oppimistilanteissa.

Esimerkkinä edelliseen nykyisin paljon käytetty opetusmenetelmä, verkko-opetus. Verkko-oppimisympäristöt soveltuvat erittäin hyvin pohjaksi edellä kuvatun kaltaisen kognitiivisia prosesseja tukevan oppimisympäristön toteuttamiseen. Tiedon esitystä voidaan esimerkiksi monipuolistaa interaktiivisen multimedian avulla. Verkko tarjoaa myös laajat mahdollisuudet kytkeä opetettava tieto hierarkisesti muuhun tietoon ja esim. tarjota linkkejä ulkoisiin syventäviin tietolähteisiin. On kuitenkin huomattava, ettei pelkkä opetusmateriaalin tuottaminen hypertekstimuotoon vielä takaa tehokkaampaa oppimiskokemusta. Erilaiset ihmiset omaksuvat tietoa eri tavoilla ja joillekin verkkoympäristöille ominainen tiedon epälineaarinen esitystapa voi jopa vaikeuttaa oppimista. Tiedon verkottamisen lisäksi on oppimisympäristön tarjottava tehokas (kognitiivista taakkaa keventävä) käyttöliittymä tiedon löytämiseen ja käsittelyyn. Käyttöliittymän tulisi siis myös osaltaan tukea ihmisen yksilöllisiä kognitiivisia prosesseja.

2.4 Konstruktiivinen oppimiskäsitys

”Joka kerran kun joku ennenaikaisesti opettaa lapselle jotakin, minkä lapsi olisi voinut itse saada selville, lasta estetään keksimästä sitä ja samalla ymmärtämästä sitä täydellisesti” (Piaget 1983, 113). ”Konstruktivismin painopiste on siinä, mitä oppija itse aktiivisesti rakentaa ja tuottaa, ei siinä, mitä hän opetuksen kautta vastaanottaa” (Puolimatka 2002, 13).

Konstruktivismin olemusta on tulkittu monin eri tavoin. Joidenkin mukaan konstruktivismi ei ole ainoastaan oppimiskäsitys, vaan kyseessä on tapa jäsentää tietoa ja hahmottaa todellisuutta. Tapio Puolimatka kirjassaan

Opetuksen teoria, toteaa, että sikäli kuin oppiminen sisältää aina jonkin uuden rakentamista vanhan tietoaiksen päälle, voidaan useita erilaisia oppimiskäsityksiä luonnehtia konstruktivistisiksi. Kognitiiviseen oppimiskäsityksen liittyy mainittua aiemmin opittujen taitojen hyödyntämistä uuden oppimisessa. Lisäksi rakentamisen tai keksimisen aste oppimisprosessissa painottuu eri tavoin eri näkemyksissä. Siksi konstruktivismiin sisältä löytyy monia oleellisesti erilaisia käsityksiä.

3 Ohjelmoinnin opintokokonaisuus

Opintokokonaisuus, johon ohjelmoinnin perusteetkin kuuluu, koostuu siis useista toisiinsa liittyvistä aihealueista. Seuraavassa kerron vähän pohjatietoa näistä aihealueista ja niiden kytkeytymisestä toisiinsa.

3.1 Wwv-tuottaminen

Wwv-tuottamisen kurssi toteutetaan xhtml -kieleen tutustumalla ja välineenä käytetään ilmaista Notepad++ nimistä koodieditoria. Verkkosivut syntyvät tällä kurssilla siis puhtaasti koodaamalla, jolloin päästään haluttuun hyvään, validiin ja rakenteellisesti oikeaoppiseen lopputulokseen. Kurssilla opetellaan kaikki normaalin verkkosivun tärkeimmät perusrakenteet ja kurssilla paneudutaan myös lomakkeiden luomiseen. Ohjelmoinnin perusteet kurssilla tarvitaan lomakkeita koodattavien pienimuotoisten sovellusten käyttöliittymänä, joten tämä kurssi on välttämätöntä käydä ennen ohjelmoinnin perusteiden suorittamista. Kurssi voitaisiin suorittaa myös käyttäen jotain täyseditoria, mutta haluan opettaa oppilaitani tekemään asiat oikein, vaikka sitten vähän hitaammin.

Oppilaitoksen toteutussuunnitelmassa kurssin tavoite on määritelty seuraavasti:

Opintojakson hyväksyttävästi suorittanut opiskelija tuntee Webin erityisluonteen viestinnän kanavana (pyrkimys laite- ja ohjelmistoriippumattomuuteen, (hyperteksti, monimediaisuuden mahdollisuus, dokumenttien rakenteisuus).

Opiskelija hallitsee konkreettisesti keskeiset staattisen Web-sivuston suunnitteluun ja toteuttamiseen liittyvät asiat. Opiskelija osaa käyttää hyväkseen vuorovaikutteisessa Web-palvelussa tarvittavia valmiita ohjelmia. Hän omaa näkemystä arvioida erilaisten Web-tekniikoiden soveltuvuutta erilaisissa käyttökohteissa. (Jyväskylän aikuisopisto 2002, 16.)

3.2 Ohjelmoinnin perusteet

Ohjelmoinnin perusteet kurssilla aikuisopiskelijat tutustuvat ohjelmoinnin perusrakenteisiin. Tavoitteena on saada käsitys ohjelmoinnista ja ohjelmointityöstä yleensä. Kurssilla tehdään yksinkertaisia perusrakenteisiin perustuvia pieniä ohjelmia.

Tietojenkäsittelyn perustutkinnon tutkinnonperusteissa todetaan seuraavaa:

Opiskelijan on osattava suunnitella ja ohjelmoida pienimuotoisia, rakenteisia ohjelmia, jotka sisältävät käyttöliittymän, laskentalogiikkaa sekä tietokannan tai tiedoston käsittelyn, jolloin hänen on tunnettava ohjelmoinnin perusrakenteet ja -käsitteistö. (Opetushallitus 2000, 49.)

Oppilaitoksen toteutussuunnitelmassa kurssin tavoite on määritelty seuraavasti:

Opiskelija ymmärtää ohjelmoinnin perusrakenteet. Hän osaa jäsentää arkipäivän ongelman algoritmiksi. Opiskelija osaa laatia pienen ohjelmointiongelman kirjallisesta kuvauksesta vuokaavion, josta hän tuottaa toimivan ohjelman valitulla ohjelmointivälineellä. (Jyväskylän aikuisopisto 2002, 14.)

3.3 Tietokantojen jatkokurssi

Tietokantojen jatkokurssilla kurssilla opiskelijat tutustuvat tietokantojen kyselykieleen, SQL:ään. Kurssilla käydään läpi kaikki kyselykielen peruslauseet. Opiskelija tarvitsee SQL taitoja suoriutuakseen php-jatko kurssista.

Tietojenkäsittelyn perustutkinnon tutkinnonperusteissa todetaan seuraavaa:

Opiskelijan on osattava suunnitella tietokantojen verkkokäyttö tukemaan yrityksen liiketoimintaa ja markkinointia. Hänen tulee osata yhdistää eri tekniikoita verkkosivulle siten, että hän saa aikaan toimivan kokonaisuuden. Opiskelijan on osattava hyödyntää laajoja tietokantoja jollakin tietokantaohjelmalla. Hänen tulee osata käyttää kyselykieltä tehostaessaan tietokannan käyttöä. (Opetushallitus 2000, 54.)

Oppilaitoksen toteutussuunnitelmassa kurssin tavoite on määritelty seuraavasti:

Opiskelijan on osattava hyödyntää laajoja tietokantoja. Hänen on osattava käyttää kyselykieltä tehostaakseen tietokannan käyttöä. (Jyväskylän aikuisopisto 2002, 19.)

3.4 Tietoverkkosovellusten ohjelmointi

Tietoverkkosovellusten ohjelmointi kurssilla opiskellaan vaativampia ohjelmointitaitoja ja tutustutaan PHP-kielen kehittyneempiin ominaisuuksiin. Tavoitteena on toteuttaa vuorovaikutteinen verkkosovellus, jonka toteuttamisessa tarvitaan sekä ohjelmoinnin perusteiden että SQL:n hallintaa. Kurssin laajuus on 5 opintoviikkoa

Tietojenkäsittelyn perustutkinnon tutkinnonperusteissa todetaan seuraavaa:

Opiskelijan on osattava ohjelmoida sähköinen kauppapaikka, asiointijärjestelmä tai muu verkkopalvelu. Opiskelijan on osattava toimia tähän tarkoitukseen organisoidussa projektissa, jolle on määritelty tavoitteet tehtävät, tekijät, aikataulu ja resurssit. Toteutuksessa opiskelijan on osattava suorittaa hänelle annetut ohjelmointiin, testaukseen ja dokumentointiin liittyvät tehtävät. Opiskelijan on osattava tehdä oma osuutensa, niin että se on sopusoinnussa kokonaisuuden kanssa. Hänen on osattava dokumentoida järjestelmä tarkoituksenmukaisesti ottaen huomioon tulevat käyttäjät ja ylläpitäjät sekä laatia tarvittavat ohjeet. Hänen on osattava kouluttaa ja neuvoa verkkopalvelun käyttäjiä. (Opetushallitus 2000, 54.)

Oppilaitoksen toteutussuunnitelmassa kurssin tavoite on määritelty seuraavasti:

Opintojakson hyväksyttävästi suorittanut opiskelija tuntee tietoverkkosovellusten erityisluonteen viestinnän kanavana (pyrkimys laite- ja ohjelmisto riippumattomuuteen, hyperteksti, monimediaisuuden mahdollisuus, dokumenttien rakenteisuus). Hän osaa rakentaa pienehkön erilaisille käyttäjäryhmille tarkoitetun tietokantaa käyttävän tietoverkkosovelluksen.

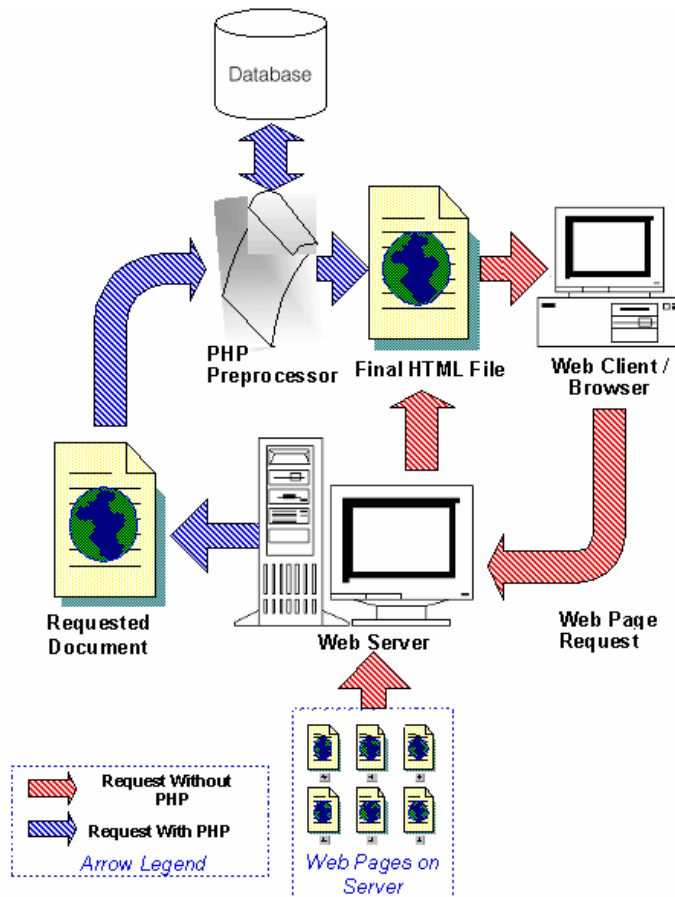
Opiskelija hallitsee konkreettisesti keskeiset staattisen Web-sivuston suunnitteluun ja toteuttamiseen liittyvät asiat. Opiskelija osaa käyttää hyväkseen vuorovaikutteisessa Web-palvelussa tarvittavia valmiita ohjelmia. Hän omaa näkemystä arvioida erilaisten Web-tekniikoiden soveltuvuutta erilaisissa käyttökohteissa. (Jyväskylän aikuisopisto, 2002, 17.)

4 Ohjelmointikieli, PHP.

Ohjelmointi itsessään on varmasti outo aihealue monelle tämän raportin lukijalle. Ohjelmointikieliä on olemassa useita erilaisia. Seuraavassa kerron hieman siitä mikä on PHP –ohjelmointikieli.

PHP (lyhenne sanoista PHP: Hypertext Preprocessor) on ohjelmointikieli, jota käytetään dynaamisten verkkopalveluiden luomiseen. PHP sai alkunsa vuonna 1995 ja on siitä kehittynyt vuosien varrella eniten käytetyksi verkko-ohjelmointikieleksi. PHP ohjelmointikielenä on hyvin pitkälle samankaltaista C/C++ ohjelmointikielien kanssa. Näin ollen C/C++ taitajille PHP:n opiskelu on suhteellisen helppoa. PHP on niin sanottu palvelinpään kieli, joka tarkoittaa sitä, että ohjelmakoodi tulkitaan www-palvelimella jolle on asennettu PHP-tulkki. Tulkki muuntaa PHP-ohjelmakoodin tuotoksen HTML/XHTML muotoon jolloin lopullinen käyttäjä ei näe sivun lähteenä ollutta PHP-koodia sivun lähdekoodissa. Tästä on muun muassa se hyöty, että käyttäjä ei pääse manipuloimaan sivun sisältöä muuttamalla ohjelmakoodia, tällaisia tapauksiahan on muiden tekniikoiden yhteydessä ilmennyt viimeaikoina. Valitsin PHP:n ohjelmointikieleksi sen vuoksi, että se parhaiten sopii tähän opintokokonaisuuteen. Pelkästään ohjelmoinnin perusteita ajatellen ohjelmointikielellä ei ole niin suurta merkitystä, koska perusasiat ovat pitkälti samanlaiset kielestä riippumatta.

Alla olevassa kaaviossa on kuvattu PHP -pohjaisen sivun generoitumista asiakaskoneelle.



(KUVIO 1. PHP:n toiminta. (www.onlamp.com))

5 Ohjelmoinnin perusteiden sisältö

Ohjelmoinnin perusteet – kurssille valitsin sisältöaiheiksi opetus suunnitelman ja toteuttamissuunnitelman vaatimusten perusteella seuraavien aihealueiden käsittelyn, joiden kautta tutustutaan ohjelmoinnin peruseriaatteisiin. Vaikka tässä hankkeessa käytetään ohjelmointikielenä PHP:tä, niin siitä huolimatta nämä peruseriaatteet ovat täysin samankaltaisia käytettäessä melkein mitä tahansa ohjelmointikieltä.

Ehtolauseet

Ehtolauseet ovat keskeinen osa mitä tahansa ohjelmointikieltä. Sovellukset koostuvat hyvin harvoin pelkistä peräkkäisrakenteista. Ehtolauseita tarvitaan silloin, kun ohjelmassa tulee tehdä erilaisia valintoja eri vaihtoehtojen välillä tai kun vertaillaan eri arvoja. Useimmissa ohjelmointikielissä ehtolauseet suoritetaan if-, elseif- ja else-lauseilla.

Silmukat

Silmukoita tarvitaan ohjelmissa silloin kun halutaan toistaa jotain samaa ohjelmalauseketta useamman kerran. Silmukasta on silloin se hyöty, että samaa ohjelmalauseetta ei tarvitse kirjoittaa useaan kertaan vaan yksi ja sama käydään läpi tarvittava määrä silmukan avulla. Silmukoita voi kirjoittaa ja toteuttaa useimmissa ohjelmointikielissä joko for-, while- tai do while -lauseilla.

Taulukot

Taulukot ovat ohjelmoinnissa hyödyllisiä tietovarastoja käsiteltäessä suurempaa tietomäärää. Yhteen taulukkoon voidaan tallentaa useita arvoja. Taulukot löytyvät useimmista ohjelmointikielistä ja näin ollen ne ovat luonnollinen osa ohjelmoinnin perusteita.

Aliohjelmat

Aliohjelmia käytetään silloin kun, jotain ohjelmalohkoa tarvitaan useammassa kohtaa ohjelmakoodia. Aliohjelma on siis pätkä ohjelmakoodia, joka suorittaa tietyn toimenpiteen ja pääohjelmassa kutsutaan tätä aliohjelmaa suorittamaan tarpeen tullen oma tehtävänsä. Aliohjelmien avulla ohjelmakoodista saadaan huomattavasti lyhyempi kuin ilman niitä.

Vuokaaviot

Yhtenä oleellisena osana ohjelmoinnin opetusta käytetään vuokaavioita.

Vuokaaviot kuvaavat graafisesti ohjelman toimintaa ja etenemistä.

Vuokaavion tarkoituksena on selvittää ohjelmoijalle itselleen ohjelman

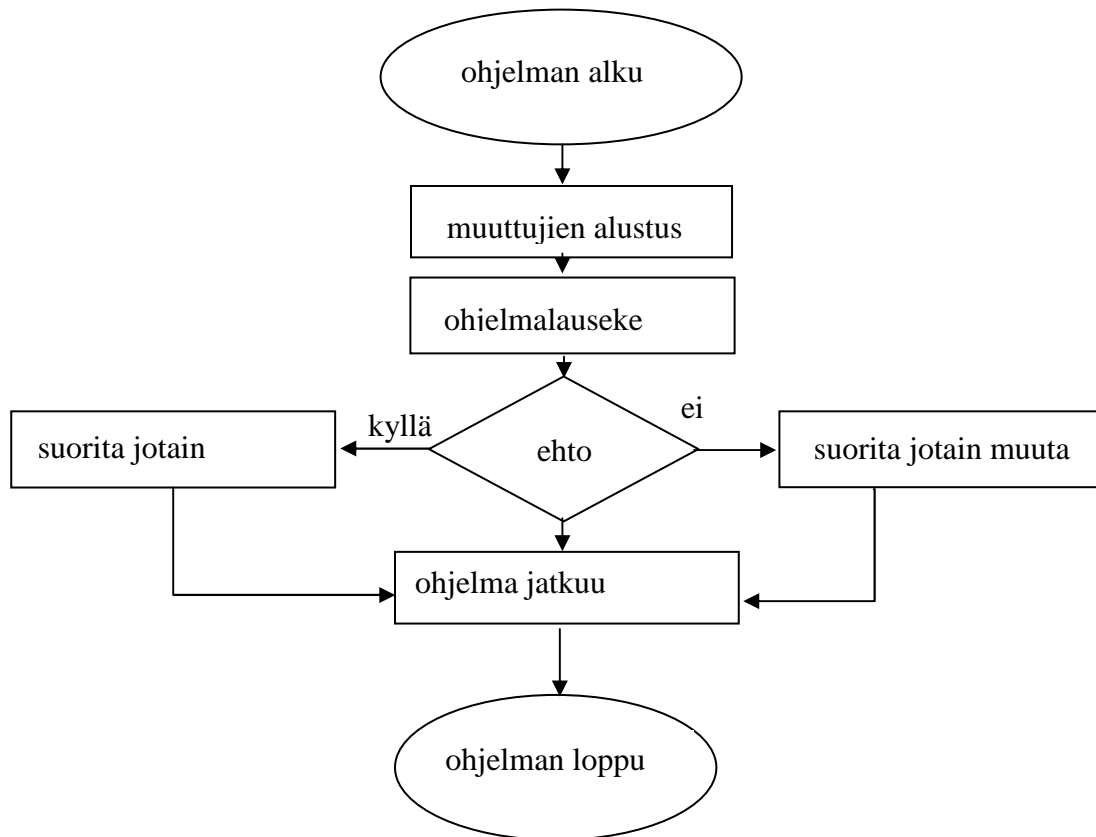
toimintaa ja samalla mahdollisesti paljastaa ohjelmassa olevia puutteita tai

virheitä. Vuokaavio tehdään annetusta ohjelmointitehtävästä ennen

varsinaisen ohjelmakoodin kirjoittamista. Vuokaavion merkitys ohjelmaidean

testaajana ei ole merkityksetön.

Esimerkki vuokaaviosta:



KUVIO 2. Esimerkki vuokaaviosta

Sanallinen algoritmi

Sanallisella algoritmilla tarkoitetaan ohjelmaa, jossa ei käytetä mitään ohjelmointikieltä vaan lausekkeet kuvaillaan sanoilla. Sanallisen algoritmin tarkoituksena on selkiyttää ohjelman toimintaa sekä hahmottaa tulevaa ohjelmakoodia. Sanallinen algoritmi tehdään ennen varsinaista ohjelmakoodia. Sanallisia algoritmeja käytetään opetuksessa yhtenä oppimiskeinona. Sanalliset algoritmit auttavat ymmärtämään ohjelmoinnin rakenteellisuutta sekä ohjelman loogista etenemistä.

Esimerkki sanallisesta algoritmista:

1. toista

1. lue henkilötunnus
2. tarkista henkilötunnus
 1. tarkista päivämäärä-luvun oikeellisuus
 2. tarkista kuukausiluvun oikeellisuus
 3. tarkista vuosiluvun oikeellisuus
 4. tarkista välimerkin oikeellisuus
 5. tarkista juoksevan numeron oikeellisuus
 6. laske 9-numeroinen luku ppkkvvxxx
 7. laske jakojäännös $ppkkvvxxx/31$ ja lisää 1
 8. onko taulukon tarkistenumero sama kuin annettu tarkiste y?
 on: henkilötunnus annettu oikein
 ei: henkilötunnus annettu väärin
3. tulosta tarkistuksen tulos: tunnus oikein tai tunnus väärin

kunnes kaikki henkilötunnukset käsitelty

2. algoritmi päättyy, kaikki työt tehty

(http://www.it.lut.fi/kurssit/98-99/1511/harjoitukset/harj_2/ratkaisut.html)

Puoliohjelma

Puoliohjelma on verrattavissa sanalliseen algoritmiin, mutta sen voidaan todeta olevan jo lähempänä varsinaista ohjelmakoodia kuin sanallinen algoritmi. Puoliohjelmasta on helppoa ja luonnollista siirtyä varsinaiseen esim. php-kieliseen ohjelmointiin. Puoliohjelman niin kuin myös sanallisen algoritmin tärkeimpänä hyötynä on se, että ohjelman ongelmakohtat ja ratkaisumallit tulee mietittyä valmiiksi jo tuossa vaiheessa, varsinainen ohjelmointityö näin ollen helpottuu suuresti.

esimerkki puoliohjelmasta

Jos (ehto) niin

(arvo 1) niin toiminnot 1

(arvo 2) niin toiminnot 2

(arvo 3) niin toiminnot 3

muuten toiminnot 4

Loppu jos

6 Välineet

Kurssilla käytettävät ohjelmistot on valittu opiskelijaystävällisesti. Kaikki käytössä olevat ohjelmistot on nimittäin ilmaisia Internetistä vapaasti ladattavia versioita, jolloin opiskelijoilla on mahdollisuus ladata ne myös omille kotikoneilleen. Muitakin vastaavia ohjelmistoja on saatavilla ja niiden käyttöön ei luonnollisesti ole mitään estettä, mutta tällä kertaa valitsin käytettäviksi ohjelmiksi seuraavat mielestäni parhaiten tarkoitukseen sopivat ohjelmat:

Notepad++

Notepad++ ohjelma on editoriohjelma, joka on tarkoitettu ohjelmakoodin kirjoittamiseen. Notepad++ soveltuu useiden, jopa kymmenien erilaisten ohjelmakielien kirjoittamiseen. Tätä samaa ohjelmaa käytetään www-tuottamisen, ohjelmoinnin perusteiden sekä vuorovaikutteisen verkkosovelluksen kurseilla. Valintaperusteina oli siis soveltuvuus useiden oppiaineiden työvälineeksi, ilmaisuus sekä käyttötarkoitukseen hyvä soveltuvuus. WWW-editorina olisi voitu käyttää myös monia muita ohjelmia kuten kaupallisia MS Frontpage tai Macromedian Dreamweaveria, mutta nämä ohjelmat eivät olisi soveltuneet muiden oppiaineiden käyttöön jolloin opiskelija olisi joutunut käyttämään eri ohjelmia eri aineissa. Yhden ja saman ohjelman käytöstä on varmasti kiistaton etu.

Xampp

Ohjelmoinnin perusteet suoritetaan siis käyttäen PHP – ohjelmointikieltä. Kuten aiemmin olen jo tässä raportissa todennut, on PHP palvelin puolen kieli, joka tarvitsee toimiakseen PHP tulkin. WWW-palvelimella pitää siis olla tällainen tulkki, jotta ohjelmakoodi toimisi. Käytännössä valmiit sivut siirretään palvelimelle, josta tällainen tulkki löytyy. Ohjelmointivaiheessa palvelimen käyttö on hieman työlästä, koska siinä joutuu jokaisen ohjelmamuutoksen jälkeen siirtämään uusimman version palvelimelle ennen kuin voi todeta muutoksen vaikutuksen ohjelmaan. XAMPP ohjelmapaketti on ratkaisu tähän ongelmaan. XAMPP on palvelinohjelmisto joka asennetaan paikallisesti omalle koneelle, jolloin ohjelmamuutosten testaus nopeutuu ja helpottuu kun

prosessista jää palvelimelle siirtovaihe kokonaan pois. Xampp ohjelmistossa on sekin hyvä puoli, että sitä voidaan hyödyntää muiden oppiaineiden yhteydessä. SQL – kurssia varten siitä löytyy MySql sovellus tietokantojen hallintaa ja muokkausta varten. WWW-tuottamisen kurssia sekä ohjelmoinnin perusteita varten siitä taas vastaavasti löytyy Apache www-palvelin ohjelmisto

Opix

Opix on oppilaitoksemme tietohallinnon ylläpitämä ”oikea” www-palvelin, joka on tarkoitettu oppilaskäyttöön. Käytämme tätä palvelinta Xampp ympäristön rinnalla testatessamme PHP sovelluksiamme. Tämä palvelin ja sen käyttö tulee oppilaille tutuksi jo ennen ohjelmoinnin perusteita www-tuottamisen kursseilla. Tuolla kurssilla palvelinta käytetään staattisten www-sivujen sijoituspaikkana, jolloin oppilaiden sivuja on päässyt tarkastelemaan kuka tahansa Internetin käyttäjä. Opix palvelimelta löytyy PHP koodin edellyttämä PHP – tulkki, Apache www-palvelin sekä SQL kyselyitä ja tietokantojen luontia varten MySql palvelinohjelmisto. Palvelimelta löytyy siis kaikki samat välineet kuin käyttämästämme testiympäristö XAMPP:sta, mutta näin käytettäessä oppilaat pääsevät tutustumaan oikeaan palvelintilanteeseen ja myös sivujen ylläpito ja päivitys tapahtuu todellisessa ympäristössä. Pidän tätä ”oikean elämän” ympäristön käyttöä erittäin tärkeänä, mutta koska käytännössä Xamppin käyttö on yksinkertaisempaa, niin käytämme sitä pääasiallisena ympäristönä.

7 Arviointi

Portfolio

Portfolio on henkilön omista töistään keräämä kokoelma, johon on kerätty esimerkiksi opiskelun aikana syntyneitä tuotoksia ja saavutuksia. Portfoliosta on nähtävissä tuolloin opiskelijan oppimisprosessin aikana kehitykseen vaikuttaneet asiat. Portfolio voi edetä kronologisesti jolloin siitä ilmenee selkeästi oppimisprosessin eteneminen. Portfolion voi siis luoda melkein pä mistä tahansa asiasta: opiskelusta, työstä tai vaikka harrastuksesta.

Portfoliolla tarkoitetaan siis yleensä joukkoa luonnoksia, suunnitelmia tai valmiita töitä, joita on työstetty jopa useita vuosia tai portfolio on voinut syntyä yhden kurssin aikana tehdyistä töistä. Tärkeintä portfolioissa on se, että siitä välittyy tekijänsä persoona ja hänelle tärkeät asiat.

Ohjelmointi oppiaineena on sangen vaativa useille opiskelijoille. Ongelmaksi muodostuu usein motivoituminen kurssille. Motivaation lisäämiseksi päätin ottaa tällä kurssilla käyttöön portfolio pohjaisen arvioinnin. Kurssin aikana oppilaat tekevät hyvin paljon opettajan antamia harjoituksia ja näiden harjoitusten tekeminen on ensiarvoisen tärkeää asian oppimisen kannalta. Perinteisesti laiskimmat oppilaat eivät ole motivoituneet harjoitusten tekoon. He ovat suorittaneet vain pienen osan tai ovat jättäneet ne tekemättä kokonaan, koska he ovat kokeneet vaikeaksi motivoitua tehtävään, jolla ei ole vaikutusta arvosanaan. Harjoitusten tekemättömyys on vastaavasti näkynyt myös oppimistuloksissa. Oppilaat keräävät näistä tekemistään harjoituksista omaa ohjelmoinnin portfolioa, joka vaikuttaa suoraan kurssista saatavaan arvosanaan. Tällä kurssilla syntyvä portfolio ei ehkä vastaa täysin perinteistä käsitystä portfolioista, koska siitä jää useimmilla uupumaan tekijänsä ”käden jälki”. Tässä tapauksessa portfolioista voi tulla hyvinkin samanlaisia tehtävien luonteesta johtuen. Portfolion vaikutus arvosanaan selviää seuraavasta taulukosta.

Harjoituksia tehtynä %	Korottava vaikutus kokeesta saatuun arvosanaan
90	+1
75	+0,75
50	+0,5

Mikäli opiskelija siis saa kokeesta arvosanan 4 ja on suorittanut kaikki kurssiin liittyvät harjoituksen niin arvosanaksi tällöin muodostuu 5.

Koe

Arvioinnin toisena osana on perinteinen luokkatyöskentelynä tehtävä koe. Kokeessa testataan opiskelijan saavuttamaa ohjelmointitaitoa ohjelmointitehtävän avulla. Kokeeseen liittyy myös vuokaavion luomista tai

tulkintaa ja teoreettisia kysymyksiä. Kokeen on tarkoitus toimia osaamistason ja kokonaisuuden ymmärtämisen mittarina. Koetta en esittele tämän tarkemmalla tasolla tässä raportissa, koska luonnollisesti en halua julkaista koetta kaiken kansan ja ennen kaikkea tulevien opiskelijoiden nähtäväksi.

8 Opettajan rooli kehittämistyön tulokset

Opettajan rooliksi muodostuu behavioristiselta vaikuttavan tiedon kaatamisen lisäksi hyvin pitkälti ohjaajan, kannustajan ja valmentajan rooli. Opiskelu koostuu suurimmalta osin tekemällä oppimisesta, jolloin oppija kaipaa ja tarvitsee juuri tämän tyyppistä ohjaavaa otetta opetuksessa. Vaikka opetus saattaa vaikuttaa behavioristiselta, niin sitä se ei kuitenkaan ole. Opiskelijat tekevät tehtäviä ja oppivat tekemästään. Opiskelussa tulee esille muun muassa konstruktivisen oppimiskäsityksen mukaisia asioita, kuten opiskelijan itsensä aktiivisuus oppimisprosessissa.

Kehittämishankkeeni toteutin työpaikallani Jyväskylän aikuisopistossa. Kehittämishankkeeni aiheena oli ohjelmoinnin perusteiden opetuksen kehittäminen Tietojenkäsittelyn perustutkinnossa. Tämän hankkeen toteuttaminen oli antoisa ja mielenkiintoinen prosessi, josta lopulta syntyi käyttökelpoinen koulutuskokonaisuus. Kehitettävää toki aina on lisää. Tärkeimpänä tämän hankkeen etuna pidän sitä, että itselleni moni asia aukesi, ja opin paremmin ymmärtämään tämän koulutusosion merkityksen osana opintojaksoa. Toivon, että ainoaksi hyötyjäksi en jäisi minä itse vaan että tästä hankkeesta ja sen kautta tulleista uusista ideoista hyötyisi ennen kaikkea opiskelijat ja myös työtoverini, jotka opettavat tämän opintojakson muita aiheita.

Opintojakson arviointimenetelmästä olen jo saanut positiivista palautetta. Opiskelijat mielsivät harjoitusten ”pakollisuuden” hyväksi tavaksi lisätä motivaatiota omaan opiskeluun. Opiskelijat olivat tyytyväisiä siihen, että heidän tekemänsä suuri työ harjoitusten parissa ei mene arvioinnin osalta hukkaan. Suurin osa opiskelijoista toki ymmärtää, että harjoituksia tehdään oman oppimisen vuoksi, ei siis pelkästään arvosanaa korottavana tekijänä.

Lähteet

Ammatillisen peruskoulutuksen opetussuunnitelman ja näyttötutkinnon perusteet, Tietojenkäsittelyn perustutkinto, datanomi, Opetushallitus 2000

Tietojenkäsittelyn perustutkinto, oppilaitoskohtainen opetussuunnitelma, Jyväskylän aikuisopisto 2002.

PHP koodin generoituminen. Viitattu 23.5.2008.

http://www.onlamp.com/php/2001/02/22/graphics/php_f_1.gif

Raahen tekniikan ja talouden yksikkö, ohjelmoinnin perusteet, OpenSource materiaali. Viitattu 23.5.2008. <http://www.ratol.fi/opensource/ohjelmointi/index.html>

LTY, tietotekniikan osasto, Ohjelmoinnin perusteet, harjoitustehtävät. Viitattu 23.5.2008.

http://www.it.lut.fi/kurssit/98-99/1511/harjoitukset/harj_2/ratkaisut.html

Puolimatka, T. Opetuksen teoria. 2002. Kustannusosakeyhtiö Tammi.

Piaget , J. 1983. Piagets theory teoksessa P.H. Mussen (toim.), Handbook of Child Psychology 1.

Rauste-von Wright Maijaliisa. Opettaja tienhaarassa. Konstruktivismia käytännössä. Juva 1998.

Kolb, D.A. 1984. Experiential learning. Experience as the source of learning and development. Englewood Cliffs N.J. Prentice-Hall