

**Luokanopettajaopiskelijoiden käsityksiä ohjelmoinnin
opettamisesta ja itsestään ohjelmoinnin opettajana**

Annina Turtiainen

Kasvatustieteen pro gradu -tutkielma
Monografiamuotoinen
Kevätlukukausi 2023
Opettajankoulutuslaitos
Jyväskylän yliopisto

TIIVISTELMÄ

Turtiainen, Annina. 2023. Luokanopettajaopiskelijoiden käsityksiä ohjelmoinnin opettamisesta ja itsestään ohjelmoinnin opettajana. Kasvatustieteen pro gradu -tutkielma. Jyväskylän yliopisto. Opettajankoulutuslaitos. 71 sivua.

Tutkimukseni tavoitteena oli saada käsitys siitä millaiseksi luokanopettajaopiskelijat kokevat oman osaamisensa ohjelmoinnin opettajina sekä siitä mitä luokanopettajaopiskelijat ajattelevat ohjelmoinnin opetuksen olevan peruskoulun vuosiluokilla 1–6 ja miten he tulevat sitä opetuksessaan toteuttamaan. Ohjelmointiosaaminen on ohjelmoinnillista ajattelua, tutkivaa työskentelyä ja ohjelmoitujen ympäristöjen tuntemista sekä niissä toimimista.

Tutkimus toteutettiin kvalitatiivisella tutkimusotteella. Tutkimusaineisto kerättiin 65 luokanopettajaopiskelijalta kirjoitelmina Webropol-lomakkeella. Aineistonkeruu aloitettiin toukokuussa 2022 ja se jatkui tammikuuhun 2023. Aineiston analyysimenetelmänä käytettiin aineistolähtöistä sisällönanalyysia.

Tutkimuksen tulosten mukaan suurin osa tutkittavista koki ohjelmoinnin itselleen vieraaksi aiheeksi ja näin ollen myös se opettamisen haasteeksi itselleen. Laitteilla ja sovelluksilla nähtiin olevan merkitystä ohjelmoinnin opetuksessa, mutta ohjelmointia voidaan opettaa myös ilman laitteita. Luokanopettajaopiskelijat nostivat ohjelmoinnin tavoitteeksi oppilaiden ohjelmointitaitojen kehittämisen sekä ohjelmoinnin ymmärtämisen osana arkipäiväisiä toimintoja.

Vaikka luokanopettajaopiskelijat kokevat oman osaamisensa vajaaksi ja ohjelmoinnin aihealueena vieraaksi, he ovat kuitenkin halukkaita kehittämään omaan osaamistaan sekä opettamaan ohjelmointia ja ohjelmoinnillista ajattelua kouluissa. Ohjelmointiosaamista pidetään tärkeänä taitona oppilaiden tulevaisuuden kannalta.

Asiasanat: luokanopettajaopiskelija, ohjelmointi, ohjelmoinnillinen ajattelu

SISÄLTÖ

TIIVISTELMÄ.....	2
SISÄLTÖ	3
1 JOHDANTO.....	5
2 OHJELMOINTI JA OHJELMOINNILLINEN AJATTELU KOULUSSA... 8	
2.1 Ohjelmoinnillinen ajattelu ja ohjelmointi yhdistyvät ohjelmoinnillisessa osaamisessa.....	8
2.2 Ohjelmoinnin opetus perusopetuksen vuosiluokilla 1-6	10
2.3 Opettajan minäpystyvyys.....	13
2.4 Opettajaopiskelijat ja ohjelmointi	15
2.5 Opettajat ja ohjelmointi	18
2.6 Opettajien käsitykset itsestään ohjelmoinnin opettajana	21
2.7 Opettajien täydennyskoulutus.....	23
3 TUTKIMUSKYSYMYKSET.....	26
4 TUTKIMUKSEN TOTEUTTAMINEN	27
4.1 Tutkimusote.....	27
4.2 Tutkittavat ja aineiston keruu	28
4.3 Aineiston analyysi	29
4.4 Eettiset ratkaisut.....	33
5 LUOKANOPETTAJAOPISKELIJOIDEN KUVAUKSET	
OSAAMISESTAAN OPETTAA OHJELMOINTIA VUOSILUOKILLA 1-6 . 35	
5.1. Luokanopettajaopiskelijoiden kuvaamat valmiudet ohjelmoinnin opettamiseen.....	35
5.2 Opettajan ominaisuudet ohjelmoinnin opettajana.....	39
5.3 Tavoitteellinen ohjelmoinnin opetus vuosiluokilla 1-6	41

6	OHJELMOINNIN OPETUS VUOSILUOKILLA 1-6.....	43
6.1	Laitteiden merkitys ohjelmoinnin opetuksessa.....	43
6.2	Ohjelmointitaitojen kehitys	45
6.3	Ohjelmointi arjessa	46
7	POHDINTA.....	48
7.1	Tulosten tarkastelu ja johtopäätökset.....	48
7.2	Tutkimuksen arviointi.....	51
7.3	Jatkotutkimusaiheet.....	53
	LÄHTEET	55
	LIITTEET.....	64

1 JOHDANTO

Suomessa ohjelmointi on osa perusopetuksen opetussuunnitelman laaja-alaisen osaamisen ”Tieto- ja viestintäteknologinen osaaminen”-osaamiskokonaisuutta ja näin ollen osa kaikkia perusopetuksessa opetettavia oppiaineita. Ohjelmointi ei siis ole oma oppiaineensa, vaan sitä harjoitellaan osana eri oppiaineita. Tämä mahdollistaa ohjelmoinnin harjoittamisen monipuolisesti osana eri oppiaineita, mutta samaan aikaan ongelmallista on se, että ohjelmoinnin opettaminen jää helpposti pintapuoliseksi (Mertala ym., 2020).

Ohjelmoinnin tultua osaksi perusopetuksen opetussuunnitelmaa syksyllä 2016, opettajat kokivat, että vaatimus ohjelmoinnin opettamisesta osana uutta opetussuunnitelmaa asetettiin heille ilman riittävää tietoa, tukea ja koulutusta (Korhonen ym., 2022). Tutkimuksissa on havaittu, että opettajilla on puutteita ohjelmoinnin sisältötietämyksessä ja ohjelmoinnillisen ajattelun käsitteen ymmärtämisessä sekä selkeässä näkemyksessä siitä, mitä ohjelmointi koulussa on (Pörn ym., 2021; Kite & Park, 2022; Morze ym., 2022). Opettajien saama vähäinen tai olematon koulutus ohjelmoinnin opettamiseen on johtanut siihen, että opettajat kokevat siitä epävarmuutta. Näin ollen ohjelmoinnillista ajattelua tukevat toimintatavat ja sisällöt sisältyvät vain pienen ohjelmoinnista innostuneen opettajajoukon opetuksen painotuksiin ja opetussisältöihin (Korhonen ym., 2022; Leino ym., 2019). Koska vain osa oppilaista saa opetusta, joka hyödyntää pedagogisesti järkevällä tavalla teknologiaa opetuksessa, luo se epätasa-arvoa oppilaiden välille (Leino ym., 2019). Opettajilla ei myöskään ole riittävästi aikaa eikä välineitä ohjelmoinnillisen osaamisen opettamiseen (Pollak & Ebner, 2019; Tanhua-Piiroinen ym., 2020; Wu ym., 2020).

Ohjelmoinnin ollessa osa opetussuunnitelmaa, ja näin ollen sen opetuksen ollessa velvoittavaa opettajien puutteellinen osaaminen ja puutteelliset resurssit aiheuttavat huolta oppilaiden ohjelmoinnillisen oppimisen kannalta. Ohjelmoinnin osaaminen sekä ohjelmoinnillisen ajattelun ymmärtäminen ovat yhä enemmän tulevaisuudessa tarvittavia taitoja, mikä korostaa niiden tärkeää roolia perusopetuksessa.

Ohjelmoinnillisen ajattelun opettaminen ja oppiminen osana eri oppiaineita ovat edelleen melko uusia ja vähän tunnettuja aihealueita, ja näin ollen ohjelmoinnin opettamisen valmiuksien opettaminen opettajaopiskelijoille on haaste myös opettajakoulutukselle (Leino ym., 2019). Lloydin ja Chandran (2020) mukaan on tärkeää, että opettajaopiskelijat kehittyvät kokemuksellisuuden kautta opettajakoulutuksessa ennemmin kuin, että opettajaopiskelijat saisivat koulutuksen aikana täyden valmiuden opettaa ohjelmointia ja ohjelmoinnillista ajattelua. Näin ollen opettajakoulutuksessa on tärkeää luoda opiskelijoille hyvä pohja ohjelmoinnin opetukselle, eikä tehdä opiskelijoista valmiita ohjelmoinnin opettajia. Hyvän perusymmärryksen saatuaan opiskelijat parantavat ymmärrystään ohjelmoinnista käytännön työssä luokassa soveltamisen ja opettamisen kautta saatuun itse kokemuksiin koulutuksen aikana (Lloyd & Chandra, 2020).

Tutkimuksen mukaan opettajaopiskelijat, joilla on parempi käsitys omasta digiosaamisestaan, osoittavat parempaa osaamista myös ohjelmoinnillisessa ajattelussa (Tondeur ym., 2018; Esteve-Mon ym., 2020). Myös opiskelijoiden asenteilla ohjelmoinnillista ajattelua kohtaan on havaittu olevan vaikutus opettajaopiskelijoiden ohjelmoinnillisen ajattelun taitojen kannalta. Opettajaopiskelijoiden positiivinen asenne ohjelmoinnillista ajattelua kohtaan ennustaa hyviä ohjelmoinnillisen ajattelun taitoja (Cutumisu ym., 2022). Opettajien opettajakoulutuksen aikaisilla asenteilla tieto- ja viestintäteknologiaa kohtaan on puolestaan havaittu olevan merkittävä vaikutus opettajien osaamiseen kehittää oppilaidensa tieto- ja viestintäteknikan käyttöä (Tondeur ym., 2018). Myös opettajaopiskelijoiden minäpystyvyyden tunteilla on vaikutusta siihen, uskovatko he olevansa kykeneviä opettamaan ohjelmointia koulussa. Opettajien on koettava minäpystyvyyttä tietokoneiden ja tekniikan käytön suhteen, jotta tekniikan ja tietokoneiden käyttö opetuksessa lisääntyisi (Zee & Koomen, 2016; Alibakhshi ym., 2020).

Opettajien minäpystyvyyden tunne vaikuttaa myös opettajien opetukseen sitoutumiseen (Skaalvik & Skaalvik, 2019). Näin ollen, kun opettajaopiskelijat saavat pystyvyyden tunteita opintojensa aikana ohjelmoinnin opettamisesta, vai-

kuttaa se heidän sitoutumiseensa ohjelmoinnin opettamiseen koulussa. Opettajien minäpystyvyyden tunteella on myös positiivinen vaikutus muun muassa oppilaiden motivaatioon, oppimistuloksiin sekä oppilaiden saamaan tukeen (Zee & Koomen, 2016; Alibakhshi ym., 2020; Perera & John, 2020). Minäpystyvyyden tunnetta voidaan parantaa täydennyskoulutuksen avulla. Täydennyskoulutus antaa opettajille itseluottamusta ja valmiuksia ohjelmoinnillisen ajattelun opettamiseen sekä rikastuttaa opettajien asenteita ja näkemyksiä ohjelmoinnista (Mason & Rich, 2019; Pörn ym., 2021).

Aiempi tutkimus on painottunut opettajien ohjelmoinnillisen osaamisen tutkimiseen, mutta harvemmin tutkimuksen kohteena ovat olleet opettajaopiskelijat. Tämä tutkimus on tärkeä, koska se tuo uusia tutkimustuloksia nimenomaan opettajaopiskelijoiden näkökulmasta. Tässä tutkimuksessa aihetta lähestytään tutkimalla luokanopettajaopiskelijoiden ajatuksia ja kuvauksia itsestään ohjelmoinnin opettajina sekä tutkimalla sitä, mitä luokanopettajaopiskelijat ajattelevat ohjelmoinnin opetuksen olevan vuosiluokilla 1–6. Tutkimuksen tavoitteena on saada aiempaa tarkempi käsitys siitä millaiseksi luokanopettajaopiskelijat kokevat oman osaamisensa ohjelmoinnin opettajina. Toisena tavoitteena on selvittää mitä luokanopettajaopiskelijat ajattelevat ohjelmoinnin opetuksen olevan vuosiluokilla 1–6 ja miten he tulevat sitä opetuksessaan toteuttamaan.

2 OHJELMOINTI JA OHJELMOINNILLINEN AJATTELU KOULUSSA

2.1 Ohjelmoinnillinen ajattelu ja ohjelmointi yhdistyvät ohjelmoinnillisessa osaamisessa

Ohjelmoinnillisen osaamisen tekijöitä ovat kyky ohjelmoinnilliseen ajatteluun sekä ohjelmointitaidot. Wing (2006, s. 33) on määritellyt, että ”ohjelmoinnillinen ajattelu sisältää ongelmien ratkaisemisen, järjestelmien suunnittelun ja ihmisten käyttäytymisen ymmärtämisen hyödyntämällä tietojenkäsittelytieteen peruskäsitteitä”. Fraillon ym. (2019, s. 27) puolestaan määrittelevät ohjelmoinnillisen ajattelun yksilön kykynä tunnistaa tosielämään pohjautuvien ongelmien ohjelmoinnillisia piirteitä sekä arvioida ja kehittää niihin algoritmisia ratkaisuja, jotka voidaan toteuttaa tietokonetta apuna käyttäen. Fagerlundin (2022) mukaan ohjelmoinnillinen ajattelu voidaan nähdä monipuolisena tieto- ja taitokokonaisuutena, jota voidaan harjoitella tutustumalla ohjelmoituun maailmaan ja ohjelmomalla koulussa eri tavoin ja säännöllisesti. Ohjelmoinnillisen ajattelun ymmärtämisen ominaisuus edellyttää sen opettamista määrätietoisesti sisällyttämällä ohjelmoinnillisen ajattelun peruseriaatteita opetussuunnitelmaan ja osaksi opettajien osaamista (Fagerlund, 2022). Wingin (2006) mukaan ohjelmoinnillisen ajattelun, aivan kuten kirjoittamisen, lukemisen ja matematiikan, tulisi olla jokaisen lapsen analyttinen kyky. Flórezin ym. (2017) mukaan ohjelmoinnin opettaminen on paras tapa opettaa ohjelmoinnillista ajattelua.

Fagerlundin ym. (2022) mukaan opettajien on tärkeää ymmärtää ohjelmoinnillinen ajattelu laajemmin kuin pelkkänä ohjelmointina. Wing (2006) on esittänyt, että ohjelmoinnillinen ajattelu edustaa kaikille tärkeää perustaitoa, jota ihmiset olisivat halukkaita oppimaan ja käyttämään. Morzen ym. (2022) mukaan ohjelmoinnillinen ajattelu rakentuu koodauksen, digitaalisen lukutaidon kehittämisen sekä ongelmanratkaisu- ja päätöksentekokykyjen perustalle. Ohjelmoinnillinen ajattelu ei siis ole vain kykyä osata tietokoneohjelmointia vaan Boomin ym. (2022) mukaan ohjelmoinnillinen ajattelu voidaan jakaa ohjelmoinnilliseen

ongelmanratkaisuun, esimerkiksi tietokoneohjelmointiin sekä erilaisiin ei-ohjelmointiongelmiin, kuten tien löytämiseen sokkelon läpi tai tanssin askelsarjojen opettelemiseen. Román-Gonzálezin ym. (2017) mukaan ohjelmoinnillinen ajattelu on ohjelmoinnin taustalla oleva ongelmanratkaisun kognitiivinen prosessi, joka mahdollistaa ohjelmoinnin osaamisen.

Ohjelmoinnillinen osaaminen ei viittaa ainoastaan ohjelmointitaitoihin vaan myös ymmärrykseen siitä, miten ohjelmoidut laitteet toimivat, mitä kaikkea ohjelmoimalla voidaan tehdä ja miten ohjelmoinnillisia menetelmiä voidaan käyttää eri tilanteissa erilaisten ongelmien ratkaisussa (Leino ym., 2019). Lyen ja Kohin (2014) mukaan ohjelmoinnillinen ajattelu onkin hyödyllinen ja jokapäiväisessä elämässä sovellettavissa oleva taito. Fagerlundin (2021) mukaan jokaisen oppilaan olisi tärkeää oppia ymmärtämään, mitä ohjelmoinnilla voidaan tehdä, miten ohjelmoidut asiat toimivat ja miten ohjelmointia voidaan soveltaa käytännössä oikean elämän ongelmien ratkaisussa. Ohjelmoinnin oppimisen ydinajatuksena on tutustua sekä ohjelmoituun ympäristöön että ohjelmoinnin yhteistoinnilliseen ja luovaan tekemiseen erilaisten leikinomaisten ja motivoivien harjoitusten ja ohjelmointitöiden kautta (Fagerlund, 2021). Ohjelmoinnissa ongelmien ratkaisemisessa loogisen ajattelun, luovuuden, kekseliäisyyden ja yhteistyötaitojen lisäksi on tärkeää oppia huomaamaan, että tietynlaisia ratkaisukäytännöitä voidaan käyttää uudelleen eri tilanteissa samalla tavalla tai hieman eri lailla (Leino ym., 2019).

Flórezin ym. (2017) mukaan ohjelmointia ei tule nähdä vain tieto- ja viestintäteknikkaan liittyvänä ihmisen ja koneen välisenä viestintävälineenä, vaan välineenä, jolla kehitetään ohjelmoinnillisen ajatteluun ja tietojenkäsittelytieteeseen liittyviä käsitteitä ja taitoja (esim. algoritminen ajattelu ja ongelmanratkaisu). Näin ollen ohjelmoinnillisen ajattelun taitojen ja ohjelmoinnin perusteiden harjoittelemisessa ei tule keskittyä pelkästään tietyn ohjelmointikielen yksityiskohtiin tai koodin kirjoittamiseen. Myös Lyen ja Kohin (2014) mukaan ohjelmointi on enemmän kuin pelkkää koodausta, sillä se altistaa ohjelmoinnilliselle ajattelulle, joka sisältää ongelmanratkaisun käyttämällä tietojenkäsittelytieteen käsit-

teitä, kuten abstraktio ja hajottaminen. Myös Fagerlundin ym. (2022) tutkimukseen osallistuneet opettajat pitivätkin tärkeämpänä ymmärtää ohjelmoinnin periaatteita kuin sitä, että oppisi ohjelmoimaan tietokonekieliä. Uudet lukutaidot (2022) määrittelee, että ohjelmoinnilliseen osaamiseen kuuluu ohjelmoinnillinen ajattelu, tutkiva työskentely ja tuottaminen sekä ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen. Näin ollen ohjelmoinnilliseen osaamiseen sisältyy myös monipuoliset ajattelun taidot ja ymmärrys ohjelmoidusta maailmasta sekä siitä, mitä kaikkea ohjelmoimalla voidaan tehdä.

Aiemmissä tutkimuksissa tutkijat ovat käyttäneet ohjelmoinnin lisäksi myös käsitettä koodaus. Zhang ja Nouri (2019) määrittelevät koodauksen tarkoittavan tietokoneohjelmointikoodin kirjoittamista. Heidän mukaansa koodaus, ohjelmointi ja ohjelmoinnillinen ajattelu nähdään kuitenkin usein synonyymeina ja niiden väliset rajat ovat hämärtyneet. Koodiaapisen (2023) mukaan koodaus on puhekielinen termi ohjelmoinnille. Englannin kielessä ohjelmoinnista puhutaan termillä "coding", joten Suomessakin on alettu puhua koodauksesta ohjelmointi käsitteen rinnalla (Koodiaapinen, 2023). Perusopetuksen opetussuunnitelmassa kuitenkin käytetään termiä ohjelmointi (Opetushallitus, 2014).

Ohjelmointi ja ohjelmoinnin oppiminen edellyttävät ohjelmoinnillista ajattelua. Ohjelmoinnin harjoittelu ja oppiminen puolestaan tukevat ohjelmoinnillisen ajattelun kehittymistä. Pelkän ohjelmointitaidon oppimista tärkeämpää on ohjelmoinnillisen ajattelun oppiminen, jota opitaan parhaiten käytännössä tekemällä, eli ohjelmoimalla. Kun ohjelmoinnillista ajattelua harjoitellaan ongelmanratkaisun keinoin, se tukee muitakin oppimisen osa-alueita kuin pelkästään ohjelmoinnin oppimista.

2.2 Ohjelmoinnin opetus perusopetuksen vuosiluokilla 1–6

Suomen opetussuunnitelmassa ohjelmointi on kytketty laaja-alaisen "Tieto- ja viestintäteknologinen osaaminen" (L5) -osaamiskokonaisuuden kautta pakolliseksi osaksi kaikkia oppiaineita (Leino ym., 2019). Laaja-alaisen osaamisen tavoitteissa ohjelmointi mainitaan suoraan vain tieto- ja viestintäteknologisen

osaamisen tavoitteissa, mutta ohjelmoinnin oppimiseen liittyviä tavoitteita sisältyy myös ”oppimaan oppiminen ja ajattelun taidot”, ”yrittäjäyys ja työelämäntaidot” sekä ”monilukutaidon” osa-alueisiin. (Opetushallitus, 2014). Laaja-alaisen osaamisen tavoitteiden lisäksi ohjelmointi liitetään perusopetuksen opetussuunnitelmassa matematiikan oppiaineeseen ensimmäiseltä luokalta alkaen sekä käsityön oppiaineeseen kolmannelta luokalta alkaen (Opetushallitus, 2014, s.129, 271). Vuosiluokilla 1–2 tutustutaan ohjelmoinnin alkeisiin laatimalla ja kokeilemalla vaiheittaisia toimintaohjeita sekä mahdollistamalla oppilaille ikäkaudelle sopivia kokemuksia ohjelmoinnista (Opetushallitus, 2014, s. 101, 129). Vuosiluokilla 3–6 oppilaat harjoittelevat suunnittelemaan ja toteuttamaan ohjelmia graafisessa ohjelmointiympäristössä sekä saavat kokemuksia siitä, miten ihmisen tekemät ratkaisut vaikuttavat teknologian toimintaan (Opetushallitus, 2014, s. 157, 235). Vuosiluokkien 3–6 käsityön sisältöihin kuuluu ohjelmoimalla aikaan saatuja toimintojen, kuten esimerkiksi automaation ja robotiikan harjoittelu (Opetushallitus, 2014, s. 271).

Suomessa ohjelmointi ei siis ole oma itsenäinen oppiaineensa, vaan niin sanottu läpäisyaine, jota harjoitellaan muiden sisältöjen ohella osana jokaista oppiainetta (Opetushallitus, 2019, s. 284; Mertala ym., 2020; Wu ym., 2020). Mertalan ym. (2020) mukaan läpäisyaineet tarjoavat mahdollisuuden moninaiseen ja monipuoliseen ilmiöiden tarkasteluun, mutta niiden käsittely jää helposti pintapuoliseksi tai olemattomaksi, koska ne eivät ole varsinaisesti kenenkään vastuulla. Tämän lisäksi Suomessa opetussuunnitelma antaa opettajille mahdollisuudet lähestyä ohjelmoinnin opetusta suhteellisen vapaasti. Lindberg ym. (2019) ovat tutkineet eri maiden opetussuunnitelmia ohjelmoinnin opetuksen suhteen. Heidän mukaansa Suomen opetussuunnitelma on vähäsanaisin ja vähiten kuvaileva. Opetussuunnitelmassa todetaan yksinkertaisesti korkean tason tavoitteet ja tarkkojen vaatimusten luettelemisen sijaan opetussuunnitelmassa käytetään lyhyitä lauseita. Opettajille opetussuunnitelmassa annettu vapaus lähestyä ohjelmointia suhteellisen vapaasti ilman tarkkaa kuvausta siitä, mitä tulisi opettaa ei välttämättä riitä ohjaamaan opettajia tarpeeksi ohjelmoinnin opettamiseen.

Ohjelmoinnin tavoitteeksi opetussuunnitelmassa on asetettu ohjelmoinnillisen ajattelun opettaminen, mutta tarkka tavoite ohjelmoinnillisen ajattelun opetukselle on kuitenkin epäselvä; mitä opettaa ja miten arvioida ohjelmoinnillista ajattelua (Hsu ym., 2018; Fagerlund ym., 2021). Pörnin ym. (2021) mukaan tarvitaan myös lisää koulutusta, jotta matemaattisen sisällön ja ohjelmoinnin välinen yhteys saadaan paremmin esille alakoulun opettajille. Perusopetuksen opetussuunnitelman lisäksi, opettajien tueksi on Opetushallituksen ja Kansallisen audiovisuaalisen instituutin Uudet lukutaidot (2022) -kehittämishjelmassa julkaistu perusopetuksen käyttöön digitaalisen osaamisen kuvaukset, jotka tukevat perusopetuksen opetussuunnitelman perusteiden (Opetushallitus, 2014) linjausten toimeenpanoa.

Vaikka Suomen opetussuunnitelmassa olisi määritelty tarkemmin mitä ohjelmoinnista tulisi opettaa, se ei välttämättä parantaisi oppilaiden ohjelmointiosaamista. Zhangin ja Nourin (2019) mukaan vaikka opetussuunnitelmissa kerrotaan tarkasti ohjelmoinnin opetuksen sisällöt, mutta opetuksen toteutusta ei ole tutkittu käytännössä, on mahdollista, että opetussuunnitelmien ”mitä opettaa” ei välttämättä vastaa sitä, mitä oppilaat voivat todella oppia. Pelkästään opetussuunnitelman tarkat kuvaukset ohjelmoinnin opetuksesta eivät siis takaa oppilaiden oppimista. Näin ollen Suomessa opettajien vapaampi lähestymistapa voi olla yhtä tehokas oppilaiden ohjelmoinnin oppimisen kannalta kuin tarkempi kuvaus ohjelmoinnin opetuksesta opetussuunnitelmassa.

Lasten koodinlukutaitoa voidaan kehittää sisällyttämällä ohjelmointia koulutusjärjestelmään ja opetussuunnitelmaan koulunkäynnin varhaisessa vaiheessa, kuten Suomessa on tehty (Pörn ym., 2021). The International Society for Technology in Educationin (2020) mukaan ohjelmoinnillisen ajattelun ja sen soveltamisen tulee olla oppiainerajat ylittäviä taitoja, niin kuin Suomen opetussuunnitelmassakin määritellään. Flórezin ym. (2017) mukaan ohjelmointia ja ohjelmoinnillista ajattelua tulisi puolestaan opiskella erillisenä oppiaineena peruskoulussa oppilaiden kognitiivisen kehityksen käynnistämiseksi varhaisessa iässä.

Popat ja Starkey (2019) ovat tarkastelleet ohjelmointia koulussa oppivien oppilaiden koulutuloksia. Tutkimuksensa tuloksena he esittivät, että oppilaiden oppiessa ohjelmoimaan voidaan ohjelmoinnin kautta saavuttaa myös oppimistuloksia matemaattisessa ongelmanratkaisussa, kriittisessä ajattelussa, sosiaalisissa taidoissa ja itseohjautumisessa sekä yleisimmissä akateemisissa taidoissa. Jiang ja Li (2021) puolestaan havaitsivat tutkimuksessaan oppilaiden luovuuden, yhteistyötaitojen sekä kriittisen ajattelun taitojen kehittyvän ohjelmointiin käytettävän Scratch-ohjelman koulukäytön myötä. Näin ollen ohjelmoinnin ja ohjelmointillisen ajattelun opetus koulussa tukee myös oppilaiden kognitiivisten taitojen kehitystä ohjelmointitaitojen oppimisen lisäksi.

2.3 Opettajan minäpystyvyys

Minäpystyvyydellä tarkoitetaan ihmisen arviota omista kyvyistä organisoida ja toteuttaa toimintatapoja, joita vaaditaan tietyn tyyppisten suoritusten saavuttamiseksi (Bandura, 2002). Minäpystyvyys on ihmisen motivaation, hyvinvoinnin ja saavutusten perusta (Bandura, 2006). Jos ihmiset eivät usko voivansa saavuttaa toivottuja vaikutuksia toimillaan, heillä ei ole ärsykettä toimia sinnikkäästi vaikeuksien edessä (Bandura, 2006). Opetuksen alueella minäpystyvyydellä tarkoitetaan opettajien itsearviota kyvystä suoriutua opetukseen liittyvistä tehtävistä, joita kasvatustavoitteiden saavuttaminen edellyttää (Granziera & Perera, 2019). McLennanin ym. (2017) mukaan minäpystyvyyden muotoutumiseen vaikuttaa persoonallisuus ja oppimiskokemukset sekä kiinnostuksen kohteet. Pereran ja Johnin (2020) mukaan minäpystyvyyden tunteet vaikuttavat yksilön mahdollisuuksien ja esteiden havaitsemiseen ympäristössä sekä päätöksiin, joita ihmiset tekevät ponnistelujen kohdentamiseksi ja tavoitteeseen liittyvien toimintojen jatkamiseksi. Minäpystyvyyden tunteet vaikuttavat myös ihmisten pyrkiänsä sekä heidän kykyynsä motivoida itseään (Skaalvik & Skaalvik, 2019).

Opettajien kokemalla minäpystyvyydellä on pedagogisia, oppijaan liittyviä sekä psykologisia seurauksia. Opettajien minäpystyvyydellä on positiivisia yh-

teyksiä oppilaiden akateemiseen sopeutumiseen, motivaatioon ja oppimistuloksiin sekä opetuksen laatuun liittyviin käytäntöihin, kuten opetuksen hallintaan ja oppilaiden tukeen (Zee & Koomen, 2016; Alibakhshi ym., 2020; Perera & John, 2020). Tämän lisäksi opettajien minäpystyvyyden tunteet vaikuttavat opettajien psykologisen hyvinvoinnin taustalla oleviin tekijöihin, kuten työtyytyväisyyteen, työhön sitoutumiseen ja alhaisempaan uupumuksen kokemiseen. Opettajien kokemilla minäpystyvyyden tunteilla on vaikutusta myös oppilaiden oppimistuloksiin. Perera ja John (2020) havaitsivat tutkimuksessaan, että opettajien minäpystyvyyden tunteet opetuksessa ovat yhteydessä oppilaiden oppimistuloksiin. Luokissa, joissa opettajilla on vahvempi minäpystyvyyden tunne opettamastaan oppiaineesta, ovat oppilaiden oppimistulokset parempia. Opettajien minäpystyvyyden tunteet opetuksessa ovat myös positiivisesti yhteydessä opettajien työtyytyväisyyteen (Perera & John, 2020).

Alhaisella minäpystyvyyden tunteella puolestaan on negatiivinen vaikutus opettajien työhyvinvointiin. Zeen ja Koomenin (2016) mukaan opettajat, joilla on alhainen minäpystyvyyden tunne, kokevat enemmän uupumuksen sekä alhaisemman tyytyväisyyden ja työhön sitoutumisen tunteita. Minäpystyvyyden tunne vaikuttaa näin ollen kokonaisvaltaisesti opettajan työskentelyyn, eikä se heijastu pelkästään opetuksellisiin seikkoihin. Siksi jo opettajankoulutuksessa olisi tärkeää vahvistaa opiskelijoiden minäpystyvyyden tunteita sekä mahdollistaa oppimiskokemuksia, joiden avulla minäpystyvyyden tunteita voidaan saavuttaa. McLennanin ym. (2017) mukaan opettajaopiskelijoiden kokemat minäpystyvyyden tunteet ennustavat heidän optimistista suhtautumistaan ja sitoutumistaan omaan työuraan opettajana. Tämä havainto on tärkeä opettajankoulutukselle, koska minäpystyvyyden tunteisiin voidaan vaikuttaa opintojen aikana ja niitä voidaan parantaa oppimiskokemusten kautta.

Opettajan opetukseen liittyvät pystyvyyden tunteet vaikuttavat opettajan sitoutumiseen opetustilanteessa sekä uusien opetusmenetelmien käyttöönottoon (Skaalvik & Skaalvik, 2019). Zeen ja Koomenin (2016) mukaan huolimatta uusien opetuskäytäntöjen hyödystä oppilaiden kehitykselle, kaikki opettajat eivät tunne kykenevänsä toteuttamaan ja käyttämään tällaisia käytäntöjä luokassa. Heidän

mukaansa opettajien, joilla on korkea yleinen minäpystyvyys, on osoitettu pitävän uusien opetusmenetelmien käyttöönottoa tärkeämpänä ja yhdenmukaisempana omien käytäntöjensä kanssa. Näin ollen myös uusien opetusmenetelmien laaja-alaisemman käyttöönoton edistämiseksi olisi tärkeää tukea opettajien minäpystyvyyden tunteita. Jotta esimerkiksi tekniikan ja tietokoneiden käyttö opetuksessa lisääntyisi, on opettajien koettava minäpystyvyyttä tietokoneiden ja tekniikan käytön suhteen (Zee & Koomen, 2016; Alibakhshi ym., 2020). Opettajien riittävä koulutus tekniikan ja tietokoneiden käyttöön osana opetusta sekä minäpystyvyyden tunteiden saaminen jo opettajaopintojen aikana on tärkeää.

2.4 Opettajaopiskelijat ja ohjelmointi

Teknologian tuodessa muutoksia koulutukseen opettajankoulutuslaitosten tulisi kehittää opiskelijoille kannustava ympäristö, joka tarjoaa mahdollisuuksia kokeilla uusia käytäntöjä teknologian integroimiseksi osaksi opetus- ja oppimisprosesseja (Tondeur ym., 2018). Opettajankoulutukseen on myös integroitava ohjelmoinnillisen ajattelun opetusta, joka valmentaa opettajaopiskelijoita sen opettamiseen koulussa (Angeli & Giannakos, 2020). Tästä huolimatta Leinon ym. (2019) mukaan suurin osa ICILS-2018 tutkimukseen osallistuneista opettajista oli hankkinut teknologiataitonsa muualta kuin opettajankoulutuksesta ja vain 39 %:lla opettajista opettajankoulutus oli sisältänyt yleisiä tieto- ja viestintäteknologian käytön opintoja. Leinon ym. (2019) mukaan myös tieto- ja viestintäteknologian käytön harjoittelu opetustarkoitukseen oli sisällytynyt vain 25 %:lla opettajista opettajankoulutuksen opintoihin. Näin ollen teknologinen osaaminen voi olla hyvin vaihtelevaa eri opettajien välillä.

Myös opiskelijoiden lähtökohdat ja osaaminen ennen opettajakoulutusta voivat olla hyvin vaihtelevia. Osalla opiskelijoista saattaa olla vain vähän tai ei lainkaan taustatietoa tietojenkäsittelystä ja useimmilla opiskelijoilla ei myöskään ole kokemusta digitekniikan käytöstä luokassa omilta kouluajoilta (Lloyd & Chandra, 2020). Tämän takia opiskelijat voivat olla tieto- ja viestintäteknologiaa

käyttäessään ja sen käyttöä opettaessaan tekemisissä heille tuntemattoman pedagogiikan kanssa. Leinon ym. (2019) mukaan opettajankoulutuksen haasteena on, että ohjelmoinnillisen ajattelun opettaminen ja oppiminen perusopetuksessa osana eri oppiaineita on edelleen melko uusi ja vähän tunnettu aihealue.

Kun ohjelmoinnillinen ajattelu on yhä enemmän integroitu opetussuunnitelmiin, on ohjelmoinnillista ajattelua opettavia opettajankoulutusohjelmia suunniteltaessa tärkeää ottaa huomioon sekä ohjelmoinnillisen ajattelun kognitiiviset että ei-kognitiiviset näkökulmat (Román-González ym., 2017; Cutumisu ym., 2022). Tämän lisäksi Mouzan ym. (2017) mukaan opettajankoulutuksen suunnittelussa tulisi myös huomioida, että koulutus auttaisi opiskelijoita kehittämään teknologista sisältötietoa suhteessa ohjelmoinnilliseen ajatteluun ja sen käytäntöihin. Mouzan ym. (2017) mukaan tässä tavoitteessa ei kuitenkaan koulutuksesta huolimatta ole onnistuttu, vaan osa opiskelijoista ei kykene opettajaopintojensa suorittamisen jälkeen suunnittelemaan oppitunteja, jotka integroisivat mielekkäästi ohjelmoinnillisen ajattelun käsitteitä ja työkaluja pedagogiikkaan. Näin ollen opettajankoulutuksessa tulisi sisältötiedon lisäksi opettaa myös pedagogiikkaa, jonka avulla ohjelmointia opettaan kouluissa. Lloyd ja Chandra (2020) pitävät ohjelmoinnin ja ohjelmoinnillisen ajattelun opettamista tärkeänä opettajien koulutuksessa myös siksi, että koulut odottavat valmistuneiden opiskelijoiden tuovan yliopistossa oppimaansa uutta tietoa kouluihin.

Opettajaopiskelijoiden valmistaminen ohjelmoinnillisen ajattelun opettamiseen on tärkeää kaikissa opettajankoulutuksen vaiheissa (Morze ym., 2022). Heidän mukaansa opettajien koulutuksessa tulisi mahdollistaa digivälineiden käyttöön liittyviä koulutuksia sekä valinnaisia kursseja, jotka auttaisivat lisäämään opettajaopiskelijoiden digiosaamista. Opettajakoulutuksen opetusteknologiakurssit kehittävät opettajaopiskelijoiden tietämystä ohjelmoinnillisen ajattelun käsitteistä, työkaluista ja käytännöistä (Mouza ym., 2017). Näin ollen opiskelijoiden on tärkeää saada opintojen aikana käytännön kokemusta ohjelmoinnista ja sen opettamisesta. Käytännön kokemuksen kautta opiskelijat oppivat ja vahvistavat osaamistaan parhaiten. Mitä enemmän opettajaopiskelijat koulutuk-

sensa aikana näkevät opettajakouluttajat roolimalleina, pohtivat teknologian roolia opintojen aikana, käyttävät teknologiaa tarkoituksellisesti, tekevät yhteistyötä vertaisten kanssa, sekä saavat jatkuvasti palautetta, sitä parempi on opiskelijoiden kyky käyttää tieto- ja viestintäteknikkaa oppimisprosesseissa ja vahvistaa opetuskäytäntöjään (Tondeur ym., 2018). Lloyd ja Chandra (2020) pitävät tärkeänä, että opettajaopiskelijat kehittyvät kokemuksellisuuden kautta sen sijaan, että he saavuttaisivat täyden valmiuden opettaa ohjelmointia ja ohjelmoinnillista ajattelua. Heidän mukaansa opiskelijat parantavat ymmärrystään ohjelmoinnista käytännön työssä ja luokassa soveltamisen ja opettamisen kautta, saatuaan itse kokemuksia koulutuksen aikana. Opettajaopintojen aikaisen koulutuksen ja opettajan työkokemuksen lisäksi täydennyskoulutus opettajan työn aikana lisää ja vahvistaa opettajien osaamista.

Opettajakoulutuksessa saadun koulutuksen lisäksi opiskelijoiden käsitys omasta digiosaamisesta ennustaa heidän ohjelmoinnillisen ajattelun osaamistaan. Opettajaopiskelijat, joilla on parempi käsitys omasta digiosaamisestaan, osoittavat parempaa osaamista myös ohjelmoinnillisessa ajattelussa (Tondeur ym., 2018; Esteve-Mon ym., 2020). Vastaavasti alhaisemmat pisteet ohjelmoinnillisessa ajattelussa ovat yhteydessä opettajaopiskelijoiden huonompaan käsitykseen omasta digiosaamisesta (Esteve-Mon ym., 2020). Opettajien itsearviointi ennustaa myös teknologian käytön määrää opetuksessa sekä heidän oppilaidensa ohjelmoinnillisen ajattelun edistämistä. Lorenzin ym. (2022) mukaan opettajien itsearvio teknologian pedagogisesta ja sisältöosaamisesta sekä tieto- ja viestintäteknologian käyttämisestä opetuksessa ennustaa teknologian käyttötiheyttä ja oppilaiden ohjelmoinnillisen ajattelun edistämistä.

Myös opettajaopiskelijoiden asenteilla ohjelmoinnillista ajattelua kohtaan on havaittu olevan merkitystä ohjelmoinnillisen ajattelun osaamisen. Opettajaopiskelijoiden positiivinen asenne ohjelmoinnillista ajattelua kohtaan ennustaa hyviä ohjelmoinnillisen ajattelun taitoja (Cutumisu ym., 2022). Puolestaan asenteet teknologian- ja koodinlukutaitoa sekä ongelmanratkaisua kohtaan eivät kuitenkaan ennusta ohjelmoinnillisen ajattelun taitoja, vaikka nämä tekijät tyypillisesti liitetään ohjelmointitaitoihin (Cutumisu ym., 2022). Tondeurin ym. (2018)

mukaan opettajien opettajankoulutuksen aikaisilla asenteilla tieto- ja viestintäteknologiaa kohtaan on merkittävä vaikutus opettajien osaamiseen kehittää oppilaiden tieto- ja viestintäteknikan käyttöä.

2.5 Opettajat ja ohjelmointi

Opettajien käsitykset ohjelmoinnista ovat hyvin erilaisia, vaihdellen alkeellisesta askel askeleelta-ajattelusta kehittyneempään päättelyyn, joka liittyy keskeisiin ohjelmoinnillisen ajattelun näkökohtiin ja muihin oppimistuloksiin (Pörn ym., 2021). Pörnin ym. tutkimuksen (2021) mukaan suurin osa opettajista liitti ohjelmoinnin lähinnä ohjeiden kirjoittamiseen, antamiseen ja seuraamiseen. Leinon ym. (2019) mukaan ohjelmoinnilliseen ajatteluun liittyvistä taidoista suomalaiset opettajat painottivat puolestaan tietojen etsimistä eri tavoin sekä monimutkaisten prosessien pilkkomista pienempiin osiin.

Opettajat katsovat ohjelmoinnin kehittävän loogista ajattelua sekä toimivan työkaluna ongelmanratkaisussa (Pörn ym., 2021; Kite & Park, 2022). Lisäksi opettajat liittivät ohjelmoinnin myös matematiikkaan sekä algoritmien käyttöön, mutta eivät juurikaan maininneet sitä, mitä ohjelmoinnillinen ajattelu sen määritelmän mukaan on (Kite & Park, 2022). Myös osa Pörnin ym. (2021) tutkimukseen osallistuneista opettajista ilmaisi puutteita ohjelmoinnin sisältötietämyksessään sekä selkeässä näkemyksessään siitä, mitä ohjelmointi koulussa on. Vaikka opettajilla on myös väärinymmärrystä ohjelmoinnillisen ajattelun käsitteestä, siitä huolimatta opettajilla on kiinnostusta oppia lähestymistapoja, jotka voisivat auttaa ohjelmoinnillisen ajattelun opettamisessa (Pörn ym., 2021; Morze ym., 2022).

Opettajien erilaisten sisällöllisten näkemysten lisäksi opettajien suhtautuminen ohjelmointiin on vaihtelevaa. Pörnin ym. (2021) mukaan opettajat suhtautuivat ohjelmointiin koulussa ristiriitaisin tuntein. Opettajilla, jotka suhtautuivat negatiivisiin tuntein ohjelmointiin ja jotka kokivat ohjelmoinnin opettamisen valmiutensa heikommaksi, oli myös suppeampi tai kielteisempi näkemys koulussa ohjelmoinnista. Useat kielteisiin tuntein ohjelmointiin suhteutuvat opettajat myös

kyseenalaistivat ohjelmoinnin sisällyttämisen opetussuunnitelmaan sekä sen tarkoituksen ja mahdollisen hyödyn (Pörn ym., 2021). Suurimmalla osalla Korhosen ym. (2022) tutkimukseen osallistuneista opettajista kuitenkin oli positiivinen asenne ohjelmointia kohtaan. Myös ICILS 2018-tutkimukseen osallistuneet opettajat suhtautuivat pääosin myönteisesti tieto- ja viestintäteknologian käyttöön opetuksessa, vaikka vain osa tutkimukseen osallistuneista opettajista itse hyödyntää tieto- ja viestintäteknologiaa opetuksessaan ja kokee hallitsevansa sen käytön (Leino ym., 2019).

Wun ym. (2020) tutkimukseen osallistuneiden muun maalaisten opettajien käsitykset tieto- ja viestintäteknologian hyödyllisyydestä opetuksessa olivat myönteisempiä kuin suomalaisten opettajien käsitykset keskimäärin. Korhosen ym. (2022) mukaan opettajien asenteet ohjelmointia kohtaan ovat yhteydessä heidän tunteisiinsa. Ilon ja uteliaisuuden tunteilla on merkittävimmät yhteydet positiivisiin asenteisiin, ja ahdistuksen ja hämmennyksen tunteilla puolestaan on yhteys sekä negatiivisiin että positiivisiin asenteisiin. Tärkeimmät tekijät positiiviseen suhtautumiseen ohjelmoinnin opetusta kohtaan, motivaatioon opettaa ohjelmointia sekä hyviksi koettuihin valmiuksiin, ovat riittävä alan tuntemus sekä positiiviset ohjelmointikokemukset (Yukselturk & Altiok, 2017; Pörn ym., 2021; Fagerlund ym., 2022). Opettajien luontainen ohjelmointimotivaatio edistää ohjelmoinnillisen ajattelun opetuksen painotusta niin kontekstuaalisempiin ohjelmoinnillisen ajattelun taitoihin, kuin myös taitoihin, jotka ovat eivät suoraan kontekstualisoidu ohjelmointiin (Fagerlund ym., 2022).

Opettajien ohjelmoinnin ja ohjelmoinnillisen ajattelun opettamiseen suhtautumisen lisäksi opettajien osaamiseen vaikuttaa Wun ym. (2020) mukaan se, jos opettajilta puuttuu virallinen ja riittävä koulutus siitä, miten ohjelmoinnillisen ajattelun ja koodaustaitojen harjoittelua voidaan toteuttaa koulutyössä. Yadavin ym. (2017) mukaan haaste ohjelmoinnin opetukselle voi olla se, ettei opetuskokemusta omaavilla opettajilla ole sisältötietoa ohjelmoinnin opetuksesta ja puolestaan opettajilla, joilla on kokemusta ohjelmoinnista ei välttämättä ole opetustaustaa tehokkaiden ohjelmointi oppituntien pitämiseen. Tanhua-Piironen ym. (2020) mukaan opettajat tarvitsevat ohjelmoinnin opetukseen valmiita

materiaaleja sekä selkeästi määritellyt sisällöt ja lisäksi ohjelmoinnin opetusvas-
tuun määrittelyä tulisi selkeyttää.

Sen lisäksi, että opettajilta puuttuu tarkka ja virallinen ohje siitä, miten oh-
jelmoinnillista ajattelua ja ohjelmointitaitoja koulussa tulisi opettaa, opettajilla on
puutteita myös omassa ohjelmointiosaamisessaan sekä taidoissa, miten ohjel-
mointia ja ohjelmoinnillista ajattelua tulisi opettaa. Digiajan peruskoulu II-hank-
keen selvityksen mukaan suomalaisten opettajien ohjelmoinnin osaaminen on
melko vähäistä; kaksi kolmasosaa opettajista jäi vaille pisteitä ICT-taitotestin al-
keisohjelmoinnin tehtävissä (Tanhua-Piironen ym., 2020). Opettajien puutteelli-
nen ohjelmointiosaaminen ja ajan puute ovat isoimpia tekijöitä, joiden takia opet-
tajat eivät opeta ohjelmointia enemmän (Pollak & Ebner, 2019; Tanhua-Piironen
ym., 2020).

Ajanpuutteen ja opettajien melko vähäisen ohjelmoinnin osaamisen on ha-
vaittu aiheuttavan epätasa-arvoa ohjelmoinnin opetukseen ja oppimiseen kou-
luissa. Korhosen ym. (2022) sekä Leinon ym. (2019) mukaan ohjelmoinnillista
ajattelua tukevat toimintatavat ja sisällöt sisältyivät vain pienen ohjelmoinnista
innostuneen opettajajoukon opetuksen painotuksiin ja opetussisältöihin. Nämä
tekijät epätasa-arvoistavat oppilaita, koska vain osa oppilaista saa opetusta, joka
hyödyntää pedagogisesti järkevällä tavalla teknologiaa (Leino ym., 2019). Myös
koulujen erilaiset resurssit työskentelyvälineiden, materiaalien ja ohjelmien han-
kintaan ja käyttöön lisäävät eriarvoisuutta koulujen välillä (Pörn ym., 2021; Kor-
honen ym., 2022).

Vaikka opettajilla on haasteita resurssien ja ohjelmoinnillisen osaamisen
kanssa, pitävät opettajat ohjelmointitaitoa tärkeänä tulevaisuuden taitona. Kor-
hosen ym. (2022) mukaan opettajat pitivät ohjelmoinnin opetusta hyvänä asiana,
koska se haastaa ja motivoi opettajia ja oppilaita oppimaan uusia taitoja, joita yh-
teiskunnassa tarvitaan nyt ja tulevaisuudessa. Opettajat ajattelevat ohjelmoinnin
olevan hyödyllinen taito oppilaille myös tulevassa työelämässä (Pörn ym., 2021;
Kite & Park, 2022). Suomalaiset opettajat uskovat, että ohjelmointitaidosta on
hyötyä, vaikka oppilaat eivät tulisikaan työskentelemään tieto- ja viestintäteknolo-
gian alalla (Wu ym., 2020). Suurin osa suomalaisista opettajista piti niin ikään

tärkeänä sitä, että koodaustaitoja opetetaan myös niille, jotka eivät pyri ammattiohjelmoijiksi. Opettajat voivat myös pitää ohjelmointia koulutuksellisesti tärkeänä sisältönä, vaikka he eivät ole välttämättä kovin kiinnostuneita siitä henkilökohtaisesti. Suurin osa Fagerlundin ym. (2022) tutkimukseen osallistuneista opettajista ei voinut kuvitella ohjelmoivansa vapaa-ajallaan, vaikka yli puolet piti ohjelmointia mielenkiintoisena ja lähes puolet opettajista opetti tai olisi halunnut opettaa ohjelmointia mielellään.

Vaikka suomalaiset opettajat kokevat epävarmuutta omasta ohjelmointiosaamisestaan sekä kyvystään opettaa ohjelmointia osana opetustaan, suomalaiset oppilaat osaavat ohjelmointia ja ohjelmoinnillista ajattelua. Suomalaisten nuorten ohjelmoinnillisen ajattelun osaaminen on kansainvälisesti mitattuna hyvällä tasolla (Leino ym., 2019). ICILS 2018 -tutkimuksessa suomalaiset oppilaat sijoittuivat ohjelmoinnillisessa ajattelussa kolmanneksi yhdeksän maan joukossa, ja neljännes oppilaista ylsi erinomaiselle tasolle ohjelmoinnillisessa ajattelussa (Leino ym., 2019). Teknologian integrointi ja ohjelmoinnillisen osaamisen opettaminen oppilaille on tärkeää, mutta Leinon ym. (2019) mukaan teknologian integroinnin tulee kuitenkin tapahtua pedagogisista lähtökohdista eli teknologian käytön pitää lähteä opetusta tukevasta tarpeesta tai opittavista taidoista. Näin ollen teknologian käyttö opetuksessa ei ole itsetarkoitus, vaan opetuksessa käytetään niitä teknologioita ja sovelluksia, jotka opetettavan asian opetukseen parhaiten soveltuvat.

2.6 Opettajien käsitykset itsestään ohjelmoinnin opettajana

Opettajien vähäinen koulutus ja kokemus ohjelmoinnin opettamisesta sekä opetussuunnitelman suppea kuvaus ohjelmoinnin opetuksesta vaikuttavat opettajien käsityksiin itsestä ohjelmoinnin opettajana. Misfeldin ym. (2019) mukaan opettajat ovat kuitenkin kiinnostuneita työskentelemään ohjelmoinnin parissa, vaikka he eivät koe olevansa hyvin valmistautuneita tähän tehtävään. Ohjelmoinnin tultua osaksi opetussuunnitelmaa opettajien valmiudet opettaa ohjelmointia olivat vaihtelevat. Korhosen ym. (2022) mukaan opettajat kokivat, että

vaatimus ohjelmoinnin opettamisesta osana uutta opetussuunnitelmaa asetettiin heille ilman riittävää tietoa, tukea ja koulutusta. Opettajat myös pitivät ohjelmoinnin opetusta monimutkaisena, koska he kokivat epävarmuutta ja puutteellista osaamista siihen liittyvien sisältöjen sekä työkalujen ja ohjelmien suhteen. Myös Wun ym. (2020) vertaillessa eri maiden opettajien valmiuksia opettaa koodaustaitoja, useat suomalaiset opettajat kokivat, että koodaaminen on heille täysin tuntematon aihe.

Opettajien luottamus omiin ohjelmoinnin opetuksen taitoihin vaikuttaa heidän ohjelmoinnin opetukseensa sekä yhteistyöhön muiden opettajien kanssa. Leinon ym. (2019) mukaan opettajien ohjelmoinnilliseen ajatteluun liittyvien sisältöjen painotus, asenteet teknologiaa kohtaan ja tieto- ja viestintäteknologian käytön yleisyys olivat yhteydessä siihen, luottivatko opettajat omiin tieto- ja viestintäteknologian käyttötaitoihinsa ja oliko opettajilla teknologiaan liittyvää yhteistyötä muiden opettajien kanssa. Opettajien luottamuksella omiin taitoihin on yhteys myös opettajien käsitykseen teknologian hyödyistä opetuksessa. Leinon ym. (2019) mukaan opettajien hyvä luottamus omiin taitoihinsa ja teknologian aktiivinen käyttö opetuksessa ovat yhteydessä opettajien myönteisiin käsityksiin teknologian hyödyistä opetuksessa.

Myös opettajien erilaiset ominaisuudet vaikuttavat opettajien käsitykseen omasta digiosaamisestaan. Hämäläisen ym. (2021) mukaan ikä, sukupuoli sekä tieto- ja viestintäteknologian käytön yleisyys työn ulkopuolella selittävät opettajien uskoa hallita työssään tarvittavat tietotekniset taidot. Todennäköisimmin nuoret, miehet ja aktiivisesti tieto- ja viestintäteknologiaa käyttävät opettajat uskoivat omaavansa opettajan työssään tarvittavat tietotekniset taidot. Ikä vaikuttaa digitaitojen osaamiseen opettajilla, jotka ovat työskennelleet opettajina ennen kuin digivälineet yleistyvät kouluissa. Hämäläisen ym. (2021) mukaan vanhemmilla opettajilla on heikommat digitaaliset taidot ja he kokivat, ettei heillä ole työssä tarvittavia taitoja. Mitä vanhempi opettaja oli, sitä huonommin he kokivat pystyvänsä tukemaan oppilaiden oppimista tieto- ja viestintäteknologian avulla, mutta sitäkin tärkeämpänä he pitivät tieto- ja viestintäteknologian avulla opettamiseen panostamista. Sitä vastoin opettajien nuorempi ikä puolestaan vaikutti

opettajien osaamiseen positiivisesti. Suurin osa nuorimman ikäryhmän vastaajista uskoi olevansa hyvin tai erittäin hyvin valmistautuneita käyttämään digitekniologiaa opetuksessa (Hämäläinen ym., 2021).

Opettajien käsityksellä itsestään ohjelmoinnin opettajana ja suhtautumisella ohjelmoinnin opetukseen on vaikutusta oppilaiden kiinnostuksen heräämiseen ohjelmointia kohtaan. Opettajien tietämys, uskomukset ja motivaatio ovat merkittävässä osassa oppilaiden ohjelmoinnillisen ajattelun herättämisessä ja kiinnostuksen kasvattamisessa ohjelmointia ja teknologiaa kohtaan (Pörn ym., 2021). Opettajien täydennyskoulutuksella ja oppilaiden oppimisen havaitsemisella on puolestaan positiivinen vaikutus opettajien omaan osaamiseen ja itsetuottamukseen. Richin ym. (2021) mukaan opettajien täydennyskoulutus ja oppilaiden menestymisen näkeminen koodauksessa lisäsivät opettajien luottamusta koodauksen ja ohjelmoinnillisen ajattelun opettamiseen. Täydennyskoulutus vaikuttaa myös opettajien tietämykseen ja itsetuottamukseen ohjelmoinnillisen ajattelun toteuttamisesta osana kaikkea opetusta (Bal ym., 2022). Balin ym. (2022) mukaan opettajat integroivat ohjelmoinnillista ajattelua luokassa jo ennen osallistumista täydennyskoulutukseen, mutta he eivät olleet tietoisia tästä integraatiosta ennen täydennyskoulutuksen suorittamista. Näin ollen opettajat saattavat tiedostamattaan opettaa ohjelmoinnillista ajattelua luokissaan ja olla pätevämpiä ohjelmoinnin opettajia kuin he itse ajattelevat.

2.7 Opettajien täydennyskoulutus

Opettajien täydennyskoulutuksella on vaikutusta opettajien ohjelmoinnilliseen osaamiseen, käsityksiin itsestä ohjelmoinnin opettajana sekä oppilaiden kiinnostuksen herättämiseen ohjelmointia kohtaan. Opettajille on tärkeää tarjota mahdollisuuksia täydennyskoulutuksiin, jotka mahdollistavat ohjelmointikokemuksia ja tukevat opettajien luottamusta omiin pedagogisiin taitoihin ohjelmoinnin opettamisen suhteen (Flórez ym., 2017; Sentance & Csizmadia, 2017; Mason & Rich, 2019; Pörn ym., 2021). Heidän mukaansa täydennyskoulutuksella pysty-

tään hankkimaan tarvittavaa kokemusta, teknisiä taitoja, luottamusta ja ymmärrystä sopivista pedagogiikoista, jotta opetussuunnitelmaa voidaan toteuttaa menestyksekkäästi. Tehokkaan täydennyskoulutuksen avulla opettajat vahvistavat ohjelmoinnillisen ajattelun opettamiseen liittyviä pätevyyden ja itseluottamuksen tunteita (Mason & Rich, 2019). Täydennyskoulutus ja oman osaamisen kehittäminen vaikuttavat myös positiivisesti opettajien valmiuksiin ja asenteisiin sekä rikastuttavat opettajien näkemyksiä ohjelmoinnista (Mason & Rich, 2019; Pörn ym., 2021).

Ohjelmointiosaamisen ja siitä seuraavan ammatillisen itseluottamuksen lisäksi opettajat saavat täydennyskoulutuksesta osaamista ohjelmoinnin opetuksen toteuttamisesta koulussa. Täydennyskoulutuksessa opettajat oppivat, miten ohjelmoinnillisen ajattelun opetusta suunnitellaan, miten ohjelmoinnillista ajattelua opetetaan ja arvioidaan sekä miten teknologiaa hyödynnetään ohjelmoinnin käsitteiden opettamisessa (Angeli & Giannakos, 2020). Täydennyskoulutuksen aikana opettajat myös kehittävät ymmärrystään ohjelmoinnillisen ajattelun käsitteistä ja käytännöistä sekä ohjelmoinnin sisältöosaamisesta ja teknologian sisältötiedosta (Kong ym., 2020; Rich ym., 2021).

Ohjelmoinnin sisältötiedon oppimisen lisäksi opettajien on tärkeää oppia myös pedagogiikkaa, jonka avulla ohjelmointia ja ohjelmoinnillista ajattelua opetetaan. Meneksen (2015) sekä Masonin ja Richin (2019) mukaan opettajien täydennyskoulutuksen tulisikin painottua enemmän pedagogisen sisältötiedon opettamiseen pelkän sisältötiedon sijaan. Kongin ym. (2020) mukaan täydennyskoulutus kehittää opettajien pedagogista ymmärrystä ympäristön hyödyntämisestä ohjelmoinnin opetuksessa ja ohjelmoinnillisen ajattelun kehittämisessä. Täydennyskoulutuksen avulla opettajat myös oivaltavat, miten ohjelmoinnillinen ajattelu liittyy pedagogiikkaan ja miten ohjelmoinnillinen ajattelu voisi tulla esille heidän luokassaan (Bal ym., 2022). Täydennyskoulutuksen myötä myös opettajien kyky ymmärtää ohjelmoinnillista ajattelua laajempaan asiana kuin pelkkänä laitteiden ohjelmointina paranee. Täydennyskoulutuksessa opettajien tietoisuus ja ymmärrys ohjelmoinnillisesta ajattelusta jokapäiväisessä elämässä sekä opetuksessa lisääntyy (Bal ym., 2022).

Kong ja Lai (2023) ovat tutkineet täydennyskoulutuksen vaikutusta opettajien ohjelmoinnillisen sisältötiedon parantumiseen ja opettajien väliseen yhteistyöhön ohjelmoinnillisen ajattelun opetuksessa sekä sitä, miten nämä voivat vaikuttaa oppilaiden oppimistuloksiin. Heidän mukaansa täydennyskoulutukseen osallistuneiden opettajien sisältöosaaminen ja yhteistyöhön perustuva sitoutuminen paranivat merkittävästi. Opettajien sisältöosaaminen, yhteistyöhön sitoutuminen sekä opetuskokemus olivat merkittäviä oppilaiden saavutusten ennustajia. Kongin ja Lain (2023) mukaan opettajien täydennyskoulutus on merkittävämpi tekijä ohjelmoinnillisen ajattelun kehittämiseksi kuin opettajien tietotekninen tausta. Näin ollen oman tietoteknisen taustan ei pitäisi olla este ohjelmoinnin opettamiselle koulussa, vaan hyvällä täydennyskoulutuksella opettajista voitulla hyviä ohjelmoinnin opettajia aiemmasta ohjelmointi- ja tietoteknisestä taustasta huolimatta. Myös täydennyskoulutuksen kestolla on merkitystä opettajien oppimisen kannalta. Meneksen (2015) mukaan täydennyskoulutukset kestävät usein vain viikon, ja se on liian lyhyt aika, jotta opettajat voisivat oppia ja harjoitella uusia asioita tarpeeksi ja rauhassa. Näin ollen täydennyskoulutusten tulisi olla pidempikestoisia ja laadukkaita, jotta opettajat voisivat todella oppia.

3 TUTKIMUSKYSYMYKSET

Tutkimuksen tehtävänä on selvittää, millaisiksi ohjelmoinnin opettajiksi luokanopettajaopiskelijat kokevat itsensä sekä millaisia käsityksiä luokanopettajaopiskelijoilla on siitä, mitä ohjelmoinnin opetus on perusopetuksen vuosiluokilla 1–6 ja miten ohjelmoinnin opetusta toteutetaan. Tutkimuksen tavoitteena on myös selvittää millaiseksi luokanopettajaopiskelijat kokevat valmiutensa opettaa ohjelmointia perusopetuksen vuosiluokilla 1–6 sekä saada käsitys siitä, mitä opettajaopiskelijat ajattelevat ohjelmoinnin olevan perusopetuksen vuosiluokilla 1–6 ja miten he tulevat sitä toteuttamaan.

Tutkimuskysymykseni ovat:

1. Millaisiksi ohjelmoinnin opettajiksi luokanopettajaopiskelijat kuvailevat itseään?
2. Mitä opettajaopiskelijat ajattelevat ohjelmoinnin opetuksen olevan perusopetuksen vuosiluokilla 1–6 ja miten he tulevat toteuttamaan ohjelmoinnin opetusta?

4 TUTKIMUKSEN TOTEUTTAMINEN

4.1 Tutkimusote

Tutkimuksen tarkoituksena on selvittää, millaisiksi ohjelmoinnin opettajiksi luokanopettajaopiskelijat kuvaavat itseään ja mitä luokanopettajaopiskelijat ajattelevat ohjelmoinnin ja ohjelmoinnin opetuksen olevan vuosiluokilla 1–6. Tutkimuskysymyksiä on lähestytty laadullisen tutkimusotteen mukaisesti, ja tutkimuksen tavoitteena on selvittää opiskelijoiden henkilökohtaisia näkemyksiä ja ajatuksia. Laadullisella tutkimuksella pyritään kuvaamaan jotakin ilmiötä ja näin ollen tutkimukseen osallistuvilla tulee olla kokemusta tutkittavasta asiasta (Tuomi & Sarajärvi, 2018). Tutkittavien valinta ei voi näin ollen olla satunnaista vaan sen pitää olla harkittua ja tarkoitukseen sopivaa. Valitsin tutkittaviksi luokanopettajaopiskelijoita, koska ohjelmoinnin opetus tulee olemaan osa heidän työtään, ja näin ollen heillä on jonkinlaisia kokemuksia ja ajatuksia ohjelmoinnista ja sen opettamisesta.

Laadullinen tutkimus ei päädy yleistettävään tietoon vaan tuottaa ymmärrystä lisäävää ja hyödyllistä tietoa tutkittavasta ilmiöstä sekä kuvastaa tutkittavien käsityksiä ja kokemusmaailmaa mahdollisimman hyvin (Aaltio & Puusa, 2020). Myöskään tässä tutkimuksessa ei pyritä yleistettävään tietoon vaan ymmärryksen lisääntymiseen. Tutkimukseni pohjautuu sosiaaliseen konstruktioon, jossa ”tiedon, todellisuuden ja sen rakenteiden ja ilmiöiden nähdään muodostuvan sosiaalisessa ja kielellisessä vuorovaikutuksessa” (Jyväskylän yliopisto, 2015). Sosiaalisen konstruktioonin viitekehyksestä lähtevässä tutkimuksessa tavoitteena ei ole yhden totuuden tavoittelu (Saaranen-Kauppinen & Puusniekka, 2009). Sosiaalisessa konstruktioonissa ihmisten nähdään tuottavan kielen avulla sen sosiaalisen maailman, jonka sisällä he elävät (Puusa & Juuti, 2020).

Konstruktivistisen ontologian mukaan ei ole yhtä totuutta, joka olisi kaikille sama, vaan todellisuus rakentuu ihmisille eri tavoin riippuen ajasta, käsityksistä

ja aiemmasta elämänkokemuksesta (Heikkinen ym., 2005). Tutkimukseni ontologia perustuu konstruktivismiin, koska jokaiselle tutkittavalle todellisuus rakentuu eri tavoin, eikä ole olemassa yhtä totuutta, joka olisi kaikille sama. Epistemologia puolestaan viittaa siihen, millainen suhde tutkijalla on tutkittavien kanssa ja mitä yleensä on mahdollista tietää (Puusa & Juuti, 2020). Konstruktivistisen epistemologian mukaan tutkija on osa tutkimaansa todellisuutta, eikä todellisuuden ulkopuolinen tarkkailija (Heikkinen ym., 2005). Puusan ja Juutin (2020) mukaan ilmiötä on mahdollista ymmärtää vain osallistujan näkökulmasta, ikään kuin ilmiöiden sisältä päin. Tutkimukseni epistemologia perustuu konstruktivistiseen epistemologiaan, jossa olen tutkijana osa tutkimaani todellisuutta.

Suoritettuani Digitaalisten teknologioiden pedagoginen soveltaminen-opintojakson osana omia luokanopettajaopintoja, havahduin itsekin siihen, etten edelleenkään tiennyt tarpeeksi siitä, mitä ohjelmoinnin opetus on vuosiluokilla 1-6. Tämä herätti mielenkiintoni tutkia luokanopettajaopiskelijoiden käsityksiä itsestä ohjelmoinnin opettajina sekä sitä mitä he ajattelevat ohjelmoinnin opetuksen olevan vuosiluokilla 1-6 ennen kuin he menevät työskentelemään kouluihin.

4.2 Tutkittavat ja aineiston keruu

Keräsin tutkimusaineiston luokanopettajaopiskelijoilta kirjoitelmatehtävällä Webropol-lomakkeella, jossa oli apukysymyksiä liittyen ohjelmoinnin opettamiseen (liite 1). Kirjoitelmapyynnöllä hankitun aineiston avulla voidaan tarkastella ihmisten kokemuksia ja ajatuksia (Saaranen-Kauppinen & Puusniekka, 2009). Heidän mukaansa tutkittavien kirjoittamat tekstit sopivat tutkimuksiin, joissa ollaan kiinnostuneita tavoittelemaan tutkittavan kokemusmaailmaa tutkittavan näkökulmasta. Valitsin aineistonkeruumenetelmäksi kirjoitelmatehtävän myös siksi, että halusin saada määrällisesti kattavan aineiston. Haastattelemalla en olisi saanut näin paljon vastauksia kuin nyt sain. Kirjoitelmatehtävän etuna on myös sen, että tutkimukseen osallistujat pystyivät valitsemaan vapaasti itse, milloin he kirjoittivat kirjoitelman sekä käyttämään vastaamiseen itselleen sopivasti

aikaa. Kirjoitelmaa kirjoittaessaan tutkittavilla ei ole myöskään tutkijan läsnäolon mahdollisesti aiheuttamaa painetta ja halua miellyttää vastauksillaan läsnä olevaa tutkijaa, kuten haastattelutilanteessa voi olla. Rajasin kutsun tutkimukseen osallistumisesta vain luokanopettajaopiskelijoille, jotta saisin tietoa nimenomaan luokanopettajaopiskelijoiden valmiuksista opettaa ohjelmointia.

Aineistonkeruu alkoi toukokuussa 2022 ja päättyi tammikuussa 2023. Ensin yritin kerätä aineistoa lähestymällä luokanopettajaopiskelijoita sähköpostilla, mutta en juurikaan saanut vastauksia. Tämän jälkeen tein lyhyen YouTube-videon, jossa kerroin tutkimuksestani ja esitin tutkimukseen osallistumispyynnön. Tämä video näytettiin digitaalisia teknologioita käsittelevällä opintojaksolla. Kuulan (2006) mukaan tutkittavat ovat myötämielisempiä osallistumaan tutkimukseen, kun heitä lähestytän suoralla kontaktilla kirjallisen viestin sijasta. Videon esittämisen jälkeen sain kerättyä suurimman osan aineistostani.

Sain yhteensä 65 vastausta ja tekstiaineistoa kertyi yhdeksän sivua. Kirjoitelmat olivat keskimäärin 58 sanan mittaisia. Pisimmän kirjoitelman sanamäärä oli 212 sanaa ja lyhimmän 22 sanaa. Osa aineiston keruusta tapahtui digitaalisia teknologioita käsittelevällä opintojaksolla ja osa opintojakson jälkeen. Tulosluvussa on lainauksia tutkittavien kirjoitelmista, joissa käytän tunnustekoodia V, jolla tarkoitetaan vastaajaa sekä vastauksen järjestyslukua, esimerkiksi V23 tarkoittaa 23. vastaajaa.

4.3 Aineiston analyysi

Analysoin aineiston aineistolähtöisen sisällönanalyysin menetelmin. Päädyin aineistolähtöiseen sisällönanalyysiin, koska minulla ei ollut tutkimukseni pohjana valmista tutkimusta ohjaavaa teoriaa. Aineistolähtöisessä analyysissä aikaisemalla tiedolla, havainnoilla ja teorioilla ei pitäisi olla mitään tekemistä analyysin toteutumisen ja lopputuloksen kanssa (Tuomi & Sarajärvi, 2018). Graneheimin ja Lundmanin (2004) mukaan tutkijan tulkintaan kuitenkin vaikuttaa väistämättä tutkijan henkilöhistoria. Aineistolähtöisen tutkimuksen toteutuksen haasteena on myös se, ettei ole olemassa objektiivisia, ”puhtaita” havaintoja, vaan muun

muassa jo käytetyt käsitteet, tutkimusasetelma ja -menetelmät ovat tutkijan asettamia ja vaikuttavat aina tuloksiin (Tuomi & Sarajärvi, 2018). Laadullisessa sisällönanalyysissä tulkinta on tasapainoilua ja Granheimin ja Lundmanin (2004) mukaan tutkijan on mahdotonta ja ei-toivottavaa olla lisäämättä tiettyä näkökulmaa tutkittaviin ilmiöihin, mutta toisaalta tutkijan on myös ”annettava tekstin puhua”. Tutkimusaineistoani analysoidessa olen tiedostanut, että oma henkilöhistoriani voi vaikuttaa tutkimusaineiston tulkintaan, mutta olen pyrkinyt analysoimaan aineiston niin, ettei se vaikuttaisi analyysiin. Aaltion ja Puusan (2020) mukaan aineiston analyysin kuvauksessa tulee pyrkiä yksityiskohtaisuuteen ja perusteluihin. Omassa aineiston analyysin kuvauksessa pyrin kuvaamaan tekemiäni aineiston analyysia yksityiskohtaisesti ja perustelemaan tekemiäni valintoja.

Tuomen ja Sarajärven (2018) mukaan tutkijan tulee rajata tarkkaan ilmiö, jota tutkitaan ja siitä on kerrottava kaikki mitä irti saadaan. Heidän mukaansa sisällönanalyysi pohjautuu päättelyyn ja tulkintaan, jossa edetään empiirisestä aineistosta kohti käsitteellisempää näkemystä tutkittavista ilmiöistä. Laadullinen sisällönanalyysi on systemaattinen ja joustava analyysimenetelmä, mutta se edellyttää tutkijalta tarkkaa keskittymistä valittuihin merkitysnäkökohtiin, eli näkökohtiin, jotka liittyvät tutkimuskysymykseen (Schreier, 2014). Näin tutkija myös karsii aineistosta pois tutkittavan asian kannalta epäolennaiset maininnat. Laadullinen sisällönanalyysi myös tiivistää aineistoa pakottamalla tutkijan luokittelemaan jokaisen koodausyksikön vain yhteen alakategoriaan (Schreier, 2014). Laadulliselle sisällönanalyysille on myös ominaista, että se korostaa kategorioiden välisiä eroja ja yhtäläisyyksiä (Graneheim & Lundman, 2004). Aineiston analyysin aikana aineistoa tiivistetään ja tulkitaan samalla kun käydään vuoropuhelua teorian, empirian ja oman ajattelun kanssa (Saaranen-Kauppinen & Puusniekka, 2009).

Aloitin tutkimusaineiston analysointivaiheen lukemalla aineiston läpi useita kertoja, jolloin sain käsityksen kokonaisuudesta. Analysoin ensin tutkittavien kuvauksia ja ajatuksia itsestä ohjelmoinnin opettajina ja tämän jälkeen samalla menetelmällä tutkittavien kuvauksia ja ajatuksia ohjelmoinnin opetuksesta

vuosiluokilla 1–6 (liitteet 2. & 3.). Sisällönanalyysin ensimmäinen vaihe on alkuperäisdatan redusointi, eli pelkistäminen siten, että aineistosta karsitaan tutkimukselle epäoleellinen pois (Tuomi & Sarajärvi, 2018). Rajasin aineistosta tutkimukseni kannalta oleelliset maininnat muun tekstin joukosta ja otin kaikki nämä maininnat mukaan analyysiini. Ensin etsin aineistosta ilmaisuja, jotka viittasivat tutkittavien kuvauksiin ja ajatuksiin itsestä ohjelmoinnin opettajana ja alleviivasin ne. Tämän jälkeen alleviivasin aineistosta ilmaisuja, jotka viittasivat tutkittavien kuvauksiin ja ajatuksiin siitä, mitä ohjelmoinnin opetus on koulussa ja miten he sitä tulevat toteuttamaan. Alleviivausten jälkeen kirjoitin kaikki tutkimuskysymyksiini vastaavat alkuperäisilmaukset allekkain Word-dokumenttiin.

Analyysin seuraava vaihe on klusterointi, eli ryhmittely, jossa aineistosta kerätyt alkuperäisilmaukset käydään läpi ja aineistosta etsitään yhtäläisyyksiä ja eroavaisuuksia (Tuomi & Sarajärvi, 2018). Klusterointi oli myös tekemäni analyysin seuraava vaihe. Luin läpi listaamiani alkuperäisilmauksia ja tiivistin niistä pelkistettyjä ilmauksia. Tämän jälkeen aloin etsimään niistä yhtäläisyyksiä ja eroja ja yhdistin samaa kuvaavat pelkistetyt ilmaukset samaan alaluokkaan ja nimesin luokat niitä kuvaavilla nimillä. Schreierin (2014) mukaan luokkien nimien tulee antaa tiivis kuvaus siitä, mihin luokka viittaa. Luokkien muodostamisessa on tärkeää, että ne ovat tyhjentäviä ja toisensa poissulkevia, eli mitään tarkoitukseen liittyvää tietoa ei pitäisi sulkea pois sopivan luokan puuttumisen vuoksi eivätkä tiedot saisi sijoittua kahden luokan väliin tai kuulua useampaan kuin yhteen luokkaan (Schreier, 2014; Krippendorff, 2019). Luokittelussa aineisto tiivistyy, koska yksittäiset maininnat sisällytetään yleisempiin käsitteisiin (Tuomi & Sarajärvi, 2018).

Luokittelua jatketaan muodostamalla yläluokkia alaluokkia yhdistelemällä ja yläluokkia yhdistelemällä muodostetaan pääluokkia (Tuomi & Sarajärvi, 2018). Jatkoin analyysiä yhdistämällä samansisältöisiä alaluokkia yhteen ja muodostamalla niistä yläluokkia. Lopuksi yhdistin muodostamani yläluokat pääluokiksi ja molempiin tutkimuskysymyksiin muodostui kolme pääluokkaa. Ensimmäiseen tutkimuskysymyksen analyysistä muodostui pääluokat omien valmiuksien kuvaaminen (taulukko 1), opettajan ominaisuudet sekä tavoitteellinen

ohjelmoinnin opetus. Toiseen tutkimuskysymykseen muodostui pääluokat laitteiden merkitys, ohjelmointitaitojen kehitys ja ohjelmointitaitojen hyödynnettävyys.

Taulukko 1

Esimerkki aineiston luokittelusta ensimmäisen tutkimuskysymyksen kohdalla

Pelkistetty ilmaus	Alaluokka	Yläluokka	Pääluokka
Tavanomaista parempi osaaminen Osaa opettaa ohjelmoinnillista ajattelua Osaa hyödyntää erilaisia laitteita	Keskivertoa paremmat taidot	Hyvät valmiudet	Omien valmiuksien kuvaaminen
Melko paljon tietoa perusteista Kohtuullinen osaaminen	Osaa perusteet Aloittelija	Kohtalaiset valmiudet	
En ollenkaan tiedä, miten opettaa Oma taitotaso on huono Todella epävarma	Heikko osaaminen Ei osaa opettaa ohjelmointia Epävarmuus koko käsitteestä	Heikot valmiudet	

Aineistolähtöisyyden takia ei voida etukäteen määrittää, minkä tasoisia ja mitä luokkia aineistosta voidaan muodostaa (Tuomi & Sarajärvi, 2018). Oma aineistonanalyysini päättyi pääluokkiin, eikä yhdistäviä luokkia syntynyt. Aineiston klusterointia voi seurata vielä analyysin seuraava vaihe, aineiston abstrahointi eli käsitteellistäminen, jossa edetään alkuperäisdatan käyttämistä ilmaisuisista teoreettisiin käsitteisiin ja johtopäätöksiin (Tuomi & Sarajärvi, 2018). Abstrahointia jatketaan yhdistelemällä luokituksia niin kauan kuin se on aineiston sisällön näkökulmasta mahdollista. Omassa analyysissäni päädyttyäni kolmeen pääluokkaan kummankin tutkimuskysymyksen kohdalla, ei aineiston abstrahointia ollut mielekästä enää jatkaa.

4.4 Eettiset ratkaisut

Tutkimusprosessista on pyrittävä tekemään ”läpinäkyvä”, jotta tutkimuksen eettisten kysymysten arvioiminen mahdollistuu (Saaranen-Kauppinen & Puusniekka, 2009). Olen pyrkinyt koko ajan ”läpinäkyvyyteen” raportoidessani tutkimuksen toteutusta. Olen kuvannut aineistonkeruuprosessia sekä tekemääni aineistonanalyysiä tarkasti läpi tutkimuksen raportoinnin. Saaranen-Kauppisen ja Puusniekan (2009) mukaan tutkimusprosessin eettiset näkökulmat voidaan jakaa kolmeen luokkaan. Ensimmäisenä tulee pohtia tutkimusaiheen eettistä oikeutusta eli miksi juuri tämän ilmiön tutkiminen on perusteltua. Tässä tutkimuksessa tutkittavan ilmiön tutkiminen on perusteltua, koska aiempaa tutkimusta luokanopettajaopiskelijoiden ohjelmoinnillisesta osaamisesta ei juurikaan ole. Näin ollen tämä tutkimus tuo uutta tietoa ja ymmärrystä asiasta.

Saaranen-Kauppisen ja Puusniekan (2009) mukaan toinen pohdittava asia tutkimuksen eettisessä arvioinnissa on tutkimusmenetelmät, eli saadaanko tavoiteltava tieto suunnitelluilla aineistonkeruumenetelmillä. Pohdin tätä kysymystä suunnitellessani aineistonkeruumenetelmää ja päädyin siihen tulokseen, että keräämällä aineiston Webropol-kyselylomakkeella saan tavoittelemani tiedon mahdollisimman monelta luokanopettajaopiskelijalta ja anonyymisti. Kolmas tutkimusprosessin eettinen näkökulma liittyy Saaranen-Kauppisen ja Puusniekan (2009) mukaan tutkimusaineiston analyysiin ja raportointiin. Heidän mukaansa tutkijan on analyysivaiheessa toimittava siten, ettei tutkittavat ole tunnistettavissa ja tutkittavien on pystyttävä luottamaan siihen, että heidän anonymiteettinsä säilyy. Tässä tutkimuksessa tutkittavilta ei kerätty mitään taustatietoja ja jokainen tutkittava vastasi kirjoitelmatehtävään anonyymisti. Myöskään minä tutkijana en tiedä kuka kirjoitelmapyyntöni on vastannut. Nämä seikat lisäävät tutkittavien anonymiteettiä.

Tutkittavalle tulee antaa tarpeeksi tietoa tutkimuksesta, esimerkiksi perustiedot tutkimuksesta ja tutkimuksen toteuttajasta sekä kerättyjen tietojen käyttötarkoituksesta (Kuula, 2006). Tutkittaville tulee myös kertoa tutkimuksen tarkoituksesta ja mahdollisista vaikutuksista (Saaranen-Kauppinen & Puusniekka, 2009). Tutkielmani aineistonkeruupyynnön yhteydessä oli tietosuojailmoitus,

jonka tutkittavat lukivat ennen tutkimukseeni osallistumista (liite 4.). Tietosuojailmoituksessa kuvasin tutkielmani tarkoitusta sekä kerroin aineistonkeruusta. Tietosuojailmoituksessa tutkittaville myös kerrottiin, että tutkimukseen osallistuminen on vapaaehtoista ja tutkimuksen osallistumisen voi halutessaan keskeyttää. Tutkittavien oli myös annettava suostumuksensa tutkimukseen osallistumiseen ennen kuin he pääsivät vastaamaan kirjoitelmatehtävään.

Aineistoa säilytettiin Jyväskylän yliopiston tarjoamalla omalla henkilökohtaisella U-aseamalla, jota suositellaan käytettäväksi tutkimusaineiston säilytykseen (Jyväskylän yliopisto, 2023). Tutkimuksen valmistuttua tutkimusaineisto hävitetään asianmukaisesti ja Webropoliin tallennetut tiedostot poistetaan.

5 LUOKANOPETTAJAOPISKELIJOIDEN KUVAKSET OSAAMISESTAAN OPETTAA OHJELMOINTIA VUOSILUOKILLA 1-6

Luokanopettajaopiskelijat kuvasivat kirjoitelmissaan omia valmiuksiaan ohjelmoinnin opettamiseen oman ohjelmoinnillisen osaamisen sekä ohjelmoinnin opettamisen näkökulmista. Kirjoitelmissa kuvattiin myös, millaisia ominaisuuksia vastaajilla on ohjelmoinnin opettajina sekä millaisia tavoitteita he asettavat ohjelmoinnin opetukselle.

5.1. Luokanopettajaopiskelijoiden kuvaamat valmiudet ohjelmoinnin opettamiseen

Suurin osa tutkittavista (50 vastaajaa) koki oman osaamisensa ohjelmoinnin opettajana vielä huonoksi tai vajaaksi sen suhteen, mitä ohjelmoinnista tulisi opettaa ja miten ohjelmointia opetetaan. Vastaajat kuvasivat myös oman ohjelmoinnillisen osaamisensa olevan heikkoa, ja tämän vaikuttavan heidän kykynsä opettaa ohjelmointia oppilaille. Vastaajat ilmaisivat ymmärryksensä ohjelmoinnista olevan heikkoa ja kokivat ohjelmoinnin olevan heille vieras asia, "Ohjelmoinnin opettajana olen huono, koska en osaa itsekä ohjelmoida, saatika sitten opettaa sitä muille" (V53) ja "En ymmärrä itsekään ohjelmoinnista juuri mitään, sillä minulla on siitä hyvin vähän kokemuksia" (V39). Osassa vastauksia kävi ilmi, että vastaajat eivät ymmärrä ohjelmoinnista juuri mitään, "... en itse asiaa osaa ollenkaan enkä ymmärrä sen ideaa" (V11). Tämän lisäksi vastaajat kuvasivat myös, miten he kokevat tarvitsevansa lisää oppia ja käytännön kokemusta ohjelmoinnista, ennen kuin he ajattelevat olevansa valmiita opettamaan ohjelmointia koulussa vuosiluokilla 1-6, "Tarvisin vielä paljon opetusta ja aiheeseen perehdytystä, jotta lähtisin opettamaan aihetta koulussa" (V23).

Kun vastaajat kokivat oman ohjelmoinnillisen osaamisen olevan vielä heikkoa, osa vastaajista ilmaisi olevansa epämukavuusalueella ohjelmoinnin opetuksessa. Vastaajat kuvasivat olevansa opettajina hyvin epävarmoja, koska he eivät itsekään ymmärrä ohjelmointia, "Olen ohjelmoinnin opettajana hyvin epävarma ja kädetön. En ymmärrä itsekään ohjelmoinnista juuri mitään..." (V39). Vastaajat kokivat epävarmuutta myös siksi, että he eivät usko tietävänsä ohjelmoinnista tarpeeksi, "Tunnen olevani aiheessa aika lailla epämukavuusalueellani, syystä että en mielestäni oikein tiedä asiasta paljoakaan" (V5). Vastaajat kuvasivat itseään ohjelmoinnin opettajina epävarmoiksi ja kokivat itsensä epäpäteviksi opettajiksi, koska heillä ei ole tarpeeksi tietoa ohjelmoinnista ja sen opettamisesta, "Olen ohjelmoinnin opettajana todella epävarmana" (V24) ja "Minulla ei ole hirveästi tietoa siitä, joten koen olevani epäpätevä siihen hommaan" (V27). Epävarmuuden tunteiden lisäksi vastaajat ilmaisivat kokevansa ohjelmoinnin opettamisen jopa pelottavana ja kauhistusta aiheuttavana asiana, koska ohjelmointi koettiin itselle haastavaksi ja vielä melko tuntemattomaksi asiaksi, "Ohjelmointi tässä kohtaa tuntuu vielä melko vieraalta ja tästä syystä myös sen opettaminen aika pelottavaltakin" (V33) ja "Ohjelmointi on minulle täysin vierasta, ei ole mitään eväitä sen opettamiseen, sen opettaminen jopa kauhistuttaa" (V52).

Kun ohjelmoinnin opettaminen koetaan itselle vieraaksi asiaksi opettaa, ajatellaan ohjelmoinnin opettamiseen valmistautumiseen tarvittavan paljon aikaa, "ohjelmoinnin opettaminen on itselle todella vierasta ja se vaatii paljon valmistautumista myös itseltä" (V24) ja "Joutuisin perehtymään asiaan paljon ennen opettamista" (V16). Aikaa tarvitaan opetuksen suunnittelun lisäksi siihen, että ensin itse opetellaan asia mitä opetetaan. Vastaajat pitivät tärkeänä, että he itse perehtyvät ensin ohjelmointiin ja hallitsevat sen edes jollakin tasolla ennen kuin he voivat opettaa sitä oppilaille, "Ohjelmoinnin opettamisessa tärkeää on perehtyä itse ensin aiheeseen ja ainakin suht hyvällä tasolla hallita opettamansa asia" (V29).

Osalle vastaajista oli vielä haastavaa edes pohtia, millaiseksi ohjelmoinnin opettajaksi he itseään kuvailisivat, koska he eivät juurikaan tieneet mitä ohjelmoinnin opetus koulussa voisi olla, "Ohjelmoinnin opettajana näen itseni aika

tiedottomana ja taidottomana tällä hetkellä. Opettamisesta tulevaisuudessa en oikeastaan osaa sanoa sillä ainoa mitä tiedän on biibotit ja jotain varmaan matikan kirjoissa ohjelmoinnista on” (V25). Osalle taas oli haastavaa pohtia, miten he kuvailisivat itseään ohjelmoinnin opettajana, koska heillä ei ole kokemusta ohjelmoinnista tai sen opettamisesta, ”Itseä ohjelmoinnin opettajana on haastavaa pohtia, sillä kokemusta siitä ei ole, ja ohjelmointikin on minulle vielä vierasta” (V38) ja ”En tiedä ohjelmoinnista juuri mitään, joten on vaikea sanoa millainen olisin sen opettajana” (V55). Osa vastaajista kuvaili omaa osaamistaan vielä niin heikoksi, etteivät he osanneet ajatella itseään ohjelmoinnin opettajina, ”En oikeastaan osaa kuvitella itseäni ohjelmoinnin opettajana, koska osaaminen tuntuu olevan niin heikolla tasolla” (V36).

Vastaajat ajattelivat ohjelmoinnin laajaksi kokonaisuudeksi, joka pitäisi hallita, jotta he pystyisivät opettamaan ohjelmointia, ”Ohjelmoinnin opettaminen on laaja käsite, jota en siis vielä hallitse kuin pienen osan. Opettamiseen ei riitä, että hallitsee muutaman pelin, koko ohjelmoinnin idea ja laajuus olisi käsitettävä, ymmärrettävä laitteiden toimintoja, käskyjä, arjessa mutta myös vaikeammassa teknologiassakin” (V6). Vastauksista tuli myös ilmi, että ensimmäinen ajatus ohjelmoinnista on usein se, että ohjelmointi on monimutkaista ja jotain mitä vain ammattikoodarit osaavat. Toisaalta vastauksista nousi esiin, että kun vastaajat alkoivat pohtimaan tarkemmin mitä ohjelmointi on, ohjelmointi nähtiin yksinkertaisempänä ja arkipäiväisempänä kuin ensin ajateltuna, ”Ohjelmointi kuulostaa itsestä aika pelottavalta, sillä aluksi mieleen tulee erilaisten sivustojen ja pelien ohjelmointi ym., mutta kun miettii yksinkertaisemmin, ymmärrän kuinka yksinkertaisia ja arkisiakin asioita ohjelmointi voi olla” (V62).

Omaa osaamistaan ohjelmoinnin opettajana huonoksi kuvaavien vastaajien lisäksi 13 vastaajaa kuvasi omaa ohjelmointiosaamistaan ja kykyjään opettaa ohjelmointia vuosiluokilla 1–6 kohtalaiseksi. Vastaajat kuitenkin kokivat, etteivät he vielä osaa opettaa ohjelmointia kovin hyvin, ”En ole itse opiskellut ohjelmointia yliopiston digikurssien lisäksi, joten koen vielä, etten osaa opettaa ohjelmointia hirveän hyvin” (V22). He uskoivat osaavansa opettaa ohjelmoinnin perusteita ja näin ollen pystyvänsä opettamaan ohjelmointia koulussa. Perusteiden lisäksi

he eivät kuitenkaan osaa tarkempaa teoriaa ohjelmoinnin opetuksen taustalla kuten esimerkiksi selittää, miksi ohjelmoitaessa tehdään tiettyjä asioita tietyllä tavalla, ”Ohjelmointi on itselleni vähän tuttua. Olen saanut koulussa perusteet ohjelmoinnin opettamiseen ja osaan opettaa ne, mutten osaa esimerkiksi sen tarkemmin selittää, miksi tiettyjä asioita tehdään ohjelmoinnissa” (V37).

Monissa kirjoitelmissa ilmeni, että vastaajat ajattelevat oman osaamisensa olevan sillä tasolla, että he pystyisivät opettamaan ohjelmointia vain alkuopetuksessa, ”Yksinkertaiset harjoitteet onnistuvat esimerkiksi alkuopetuksen luokille, mutta pidemmälle en koe uskaltavani mennä” (V61). Vastaajat kuvasivat myös omien ohjelmointitaitojen riittävän vain opetussuunnitelman velvoittamaan opetukseen, ”Aion tällä hetkellä opettaa ohjelmointia vain vaadittavan määrän” (V24). Haasteeksi koettiin myös se, etteivät omat taidot riitä tarvittaessa opetuksen eriyttämiseen, ”Omat taidot eivät riitä esimerkiksi ylöspäin eriyttävään opettamiseen” (V24). Vastaajat kokivat, että he saisivat varmuutta opettaa ohjelmointia, jos he saisivat opettaa ohjelmointia yhdessä toisen opettajan kanssa. Tällöin he saisivat tukea toisesta opettajasta, ”...koska en kuitenkaan ole mielestäni todella taitava, opettaisin mielusti yhteistyössä jonkun toisen osaavan opettajan kanssa” (V20).

Kaikista tutkimukseen osallistuneista (65) vastaajista vain kaksi kertoi omaavansa hyvät taidot opettaa ohjelmointia vuosiluokilla 1–6, ”Uskon omaavani keskivertoa paremmat taidot ohjelmoinnin opettamiseen” (V58). Vastaajat kertoivat osaavansa ohjelmointikieliä, joista voi olla apua ohjelmoinnillisen ajattelun ymmärtämisessä ja ohjelmoinnin opettamisessa, ”Osaan jo ennestään joitakin ohjelmointikieliä (esim. JavaScript ja PHP), joista on apua ohjelmoinnillisen ajattelun ymmärtämisessä” (V58). Hyvien taitojen karttumista on edesauttanut kiinnostus ohjelmoinnin oppimista ja opettamista kohtaan sekä opinnot yliopistossa, ”Olen opiskellut hieman ohjelmointia yliopistossa, joten koen olevani hieman tavanomaista opettajaa vahvempi ohjelmoinnin opettajana” (V10). Useissa vastauksissa nousi esille, että vastaajien ainoat ohjelmointikokemukset ovat tulleet yliopisto-opintojen aikana.

5.2 Opettajan ominaisuudet ohjelmoinnin opettajana

Tärkeinä ominaisuuksina ohjelmoinnin opettajalle nähtiin myönteinen ja innostunut asenne oman ohjelmoinnillisen osaamisen kehittämiseksi sekä se, että opettaja huomioi oppilaiden tarpeita opettaessaan ohjelmointia. Vastaajat pitivät ohjelmoinnillista osaamista tärkeänä taitona, jota he haluavat oppia paremmin, "Ohjelmoinnin opettamisen näen tärkeänä koulussa ja haluaisin itse saada siitä lisää tietoa" ja "...olen aiheesta kiinnostunut ja innostunut oppimaan siitä lisää" (V57). Oman ohjelmoinnillisen osaamisen kehittämisen lisäksi vastaajien kehitysmuutokset tulivat ilmi myös haluna oppia opettamaan ohjelmointia, "Aion opettaa alakoulussa ainakin ohjelmoinnin perusteet, ja toivon, että opin opettamaan sitä myös syvemmin" (V37) ja "Toivon, että opin itse ohjelmoimaan ja saan vinkkejä siihen, miten ohjelmointia voi opettaa alakoulussa" (V13).

Opiskelijoiden kehitysmuutoksista kertoi myös vastaajien rohkeus kokeilla uusia laitteita ja opetella käyttämään niitä, "Minulla on melko pieni kynnyksellä erilaisia uusiakin laitteita, ja hyödynnänkin mielelläni myös erilaisia laitteita opetuksessa" (V2). Vastaajat pitivät tärkeänä, että opettajat kehittävät omaa osaamistaan sen suhteen, miten ohjelmointia voi soveltaa opetuksessa esimerkiksi osana eri oppiaineita ja muuallakin kuin omassa luokassa "...olisi hyvä tiedostaa opetuksessa mihin ohjelmointia voisi soveltaa opetuksessa ja koulumaailmassa" (V23) ja "Olen kuitenkin kiinnostunut oppimaan ja hyödyntämään ohjelmointia niin tv-tunneilla kuin myös muiden aineiden oppitunneilla" (V28). Kehitysmuutoksia osoitti myös vastaajien halu oppia itse lisää ohjelmoinnista sekä oppia opettamaan ohjelmointia ennen kuin he kokevat olevansa valmiita opettamaan sitä oppilaille, "Ohjelmoinnin opettamisessa tärkeää on perehtyä itse ensin aiheeseen ja ainakin suht. hyvällä tasolla hallita opettamansa asia" (V29).

Toinen opettajan ominaisuus, joka nousi vastauksista esille, oli oppilaita huomioiva opettaja. Vastaajat pitivät tärkeänä, että ohjelmoinnin opetus ja ohjeiden anto on selkeää ja oppilaille ymmärrettävää sekä sitä, että aluksi harjoiteltavat asiat ovat tarpeeksi yksinkertaista, "Samoin kuin muitassakin oppiaineissa, lähtisin ohjelmoinninkin kanssa yksinkertaisista ja helposti ymmärrettävistä asi-

oista liikkeelle” (V38). Vastaajien mukaan on tärkeää, että opettaja huomioi oppilaiden taitotasot sekä sen, että ohjelmointia harjoitellaan oppilaille mielekkäillä harjoitteilla, tarvittaessa eriyttäen opetusta oppilaiden taitotason mukaan, ”Uskon, että ohjelmoinnin opettamisessa on tärkeää muistaa selkeiden ohjeiden ja arviointikriteereiden lisäksi myös oppilaiden eri tasot ja eriyttäminen” (V43). Eriyttävien tehtävien antamisen nähtiin myös tukevan oppilaiden osallisuutta, ”Pyrin tukemaan oppilaan osallisuutta antamalla heille vaihtoehtoisia tapoja tehtävien tekemiseen” (V9). Vastauksista nousi esille, että luokanopettaja-opiskelijat pitävät tärkeänä osana ohjelmoinnin opetusta huomioida oppilaiden kokemusmaailma, ”Ohjelmoinnin opetus on tärkeää tuoda lähelle oppilaiden kokemus maailmaan, ihan kuin kaiken muunkin opettamisessa” (V2).

Kuvatessaan opettajalle tärkeitä ominaisuuksia ohjelmoinnin opettajana, vastaajat pitivät hyvänä asiana, että opettaja on kärsivällinen ja kannustava oppilaita kohtaan sekä tukee oppilaiden luovuutta. Opettajan on hyvä luoda luokkaan miellyttävä ja kokeiluun kannustavan ilmapiiri, jossa opitaan lapsilähtöisesti ja jossa myös epäonnistumiset sallitaan, ”Ohjelmoinnin opettelussa mielestäni tärkeää korostaa sitä, ettei siinä heti tarvitse onnistua. Asiaa harjoitellaan siinä missä muitakin asioita” (V25). Vastaajat näkivät merkittäväksi sen, että oppilaille annetaan tarpeeksi aikaa uusien asioiden oppimiseen ja harjoitteluun, ”Opettajana olen kärsivällinen ja ohjaava, annan aikaa opetella uutta” (V6).

Oppilaiden aktiivista roolia pidettiin tärkeänä osana ohjelmoinnin oppimista. Opettajan tulisi mahdollistaa oppilaille konkreettisia ohjelmointi harjoituksia, joissa oppilaat pääsevät harjoittelemaan itse ohjelmointia, ”Ohjelmoinnissa tärkeää on mahdollistaa oppilaille ohjelmoinnin kokemuksia, opettaa alkeet ja päästä erilaisilla tavoilla kokeilemaan ohjelmointia” (V7). Opettajan on myös muistettava varmistaa, että oppilaat osaavat edellisen asian ennen kuin mennään eteenpäin seuraavaan asiaan, ”Tärkeää on, että varmistetaan kaikkien osaaminen jokaisella ohjelmoinnin tasolla, jotta ei tiputa kyydistä ja ohjelmoinnin eri vaiheet tulevat tutuksi” (V37).

5.3 Tavoitteellinen ohjelmoinnin opetus vuosiluokilla 1–6

Vastaajien mukaan ohjelmoinnin opetuksen tulee olla tavoitteellista. Ohjelmointitaito nähdään tärkeänä taitona sekä nyt että tulevaisuudessa. Luokanopettaja-opiskelijat nostivat esille ohjelmoinnin merkityksen peruskoulun jälkeisessä elämässä sekä pitivät tärkeänä ohjelmoinnin opetuksen jatkumoa läpi vuosiluokkien 1–6. Ohjelmoinnin opetusta pidetään tärkeänä oppilaiden tulevaisuuden koulutuksen ja tulevaisuudessa työllistymisen kannalta, ”Ohjelmoinnin opettaminen tulee olemaan tärkeä osa tulevaisuuden koulutusta, sillä maailma digitalisoituu entisestään” (V14). Ohjelmointiosaamista pidetään taitona, jota oppilaat tulevat tarvitsemaan tulevaisuudessa koko ajan yhä enemmän teknologisoituvassa maailmassa ja useissa ammateissa, ”Nykypäivänä tarvitaan monessa eri ammatissa ohjelmointitaitoja” (V24).

Ohjelmointiosaaminen nähtiin tulevaisuudentaitona, josta oppilaiden on hyvä saada kokemusta jo peruskoulussa, ”Ohjelmointi on tulevaisuuden taito, jota on hyvä opetella jo koulumaailmassa” (V11) ja ”Oppilaiden on tärkeää saada ohjelmoinnin kokemuksia jo koulussa, koska työpaikat koodauksen parissa ovat lisääntyneet” (V7). Vastaajat pitivät tärkeänä, että ohjelmoinnin opetus aloitetaan jo alkuopetuksessa, jolloin luodaan perusta ohjelmoinnilliselle ajattelulle, ”Ohjelmoinnin opettaminen on mielestäni tärkeää aloittaa mahdollisimman varhain” (V7) ja ”Näen että pohja ohjelmoinnilliselle ajattelulle luodaan jo varhain ja myöhemmin lähdetään harjoittelemaan monimutkaisempia asioita” (V3). Jotta oppilaille olisi mahdollisuus saada vuosiluokkien 1–6 aikana hyvä ohjelmoinnillinen osaaminen, on ohjelmointia opetettava säännöllisesti läpi kaikkien vuosiluokkien, ”Aion opettaa ohjelmointia säännöllisesti eri luokka-asteilla, jottei perusteet unohdu liiaksi” (V20).

Useimmat vastaajat liittivät ohjelmoinnin opetuksen osaksi matematiikan opetusta. Osa vastaajista kuitenkin yhdisti ohjelmoinnin opetuksen myös osaksi muiden oppiaineiden opetusta, ”Ohjelmointia on hyvin luontaista integroida esimerkiksi matematiikkaan, mutta myös muihin oppiaineisiin kuten esim. liikuntaan tai suomen kieleen ja kirjallisuuteen” (V2). Ohjelmoinnin opettaminen osana

eri oppiaineita ja koulussa opetettavia kokonaisuuksia nähtiin tärkeäksi, "Ohjelmoinnillisen ajattelua tulee soveltaa oppiaineiden sisällä, ei vain matematiikassa erillisillä ohjelmointitunneilla" (V10) ja "Sitä saa hyvin integroitua eri oppiaineisiin esim. käsityöhön (vohvelikankaan kirjonta, silmukat...) tai vaikka liikuntaan (ohjeet miten toimitaan temppuradalla)" (V44). Oppiaineiden lisäksi vastaajat liittävät ohjelmointiosaamisen tietotekniseen osaamiseen sekä tekniikan tuntemukseen, "...se opettaa hyviä tietoteknisiä taitoja oppilaille" (V53) ja "Näen sen myös äärimmäisen tärkeänä tulevaisuuden taitona ja kriittisenä osana tietoteknistä osaamista" (V3). Vastauksissa nousi esiin myös tietoturvan merkitys ohjelmoinnin opetuksessa, "Tietoturva ja oikeaoppinen käyttö on tärkeää huomioida" (V30).

6 OHJELMOINNIN OPETUS VUOSILUOKILLA 1-6

Kirjoitelmissa nousi esille laitteiden merkitys ohjelmoinnin opetuksessa. Ohjelmointia voidaan opettaa laitteiden avulla tai ilman laitteita. Toiseksi keskeiseksi asiaksi ohjelmoinnin opetuksessa nähtiin ohjelmointitaitojen kehitys. Ohjelmoinnin opetuksen tavoitteeksi nähtiin ohjelmoinnillisen ajattelun kehittyminen sekä ohjelmoinnillisten käsitteiden ja ohjelmointikielten oppiminen. Kolmas keskeiseksi noussut asia oli ohjelmointitaitojen hyödynnettävyys. Ohjelmointia tulisi integroida osaksi eri oppiaineita sekä lisätä oppilaiden ymmärrystä siitä, miten ohjelmointi on osa arkipäivää ja näkyy arjessa.

6.1 Laitteiden merkitys ohjelmoinnin opetuksessa

Suuri osa vastaajista ajatteli, että laitteilla on merkitystä ohjelmoinnin opettamisessa, ”Ohjelmointia voidaan opettaa erilaisin sovelluksin ja laittein, kuten beeboteilla” (V49) ja ”liikuvien robottien ohjaamista koodauksen avulla padin kautta” (V59). Bee-bot robotit mainittiin useissa vastauksissa hyvänä apuna ohjelmoinnin opetuksessa, varsinkin silloin kun ohjelmointia aletaan harjoittelemaan. Vastaajat pitivät tärkeänä, että ohjelmoinnin opetuksessa tutustutaan monipuolisesti eri ohjelmointialustoihin, välineisiin ja sovelluksiin, ”Peruskoulun edetessä on myös hyvä harjoitella erilaisten ohjelmointisovelluksien käyttämistä” (V35). Vastaajat mainitsivat myös yleisesti, että ohjelmoinnin opetuksessa voidaan käyttää apuna laitteita, sovelluksia ja tietokonepelejä, mutta muita laitteita ja sovelluksia ei juurikaan nimetty kuin Bee-botit. Yksittäisen maininnan saivat Micro:bitit, Scratch Jr, code.org sekä Lego-robotit. Laitteilla nähtiin olevan merkitystä myös, jos ohjelmoinnin opetuksessa edetään koodien kirjoittamiseen ja ohjelmointikielten harjoitteluun.

Ohjelmointia voidaan harjoitella myös ilman digitaalisia laitteita toiminnallisten harjoitusten ja kirjallisten ohjelmointitehtävien avulla, ”Toiminnallisilla harjoituksilla koodausta voi myös harjoitella, aina ei tarvita tietokonetta tai muita

laitteita" (V7). Erilaiset toiminnalliset harjoitukset saivat useita mainintoja kirjoitelmissa. Toiminnalliset harjoitteet nähtiin ensimmäisiä harjoituksina, joiden kautta ohjelmointiin ja ohjelmoinnin ideaan aletaan tutustumaan, "Jos olen ymmärtänyt oikein, ohjelmoinnin opettaminen voi lähteä juuri käskyjen antamisesta ja sen mukaan toimimisesta. Sitten mukaan voi tuoda yksinkertaisia sovelluksia ja Beebotteja" (V22). Esimerkkinä toiminnallisesta harjoitteesta mainittiin kaverin ohjelmointi, "Ohjelmointi voi olla myös esim luokkakaverinnohjeistusta paikasta a paikkaan b antaen ohjeita. esim. käänny vasemmalle 45 astetta ja kävele eteenpäin 5 askelta" (V49). Vastauksissa mainittiin myös lautapelien ja matematiikankirjan koodaustehtävien hyödyntäminen ohjelmoinnin opetuksessa.

Yleisimmin ohjelmoinnin opetus liitettiin osaksi matematiikan opetusta, mutta toiminnallisia ohjelmointiharjoituksia voidaan tehdä myös esimerkiksi liikunnan oppitunneilla, "Esimerkiksi jaetaan lapset pareihin ja vuorotellen toinen ohjelmoi vaikkapa tempuradan läpi toisen "käskyillä" (V19) ja "Olen myös käyttänyt alakoulussa ohjelmoinnin harjoittelua ulko-tunnilla, jossa esim. käpy tarkoittaa nuoli alas, kivi = nuoli ylös jne." (V7). Vastaajien mukaan ohjelmointia voidaan opettaa ja harjoitella myös osana siirtymiä, "Oppilaiden kanssa tätä voi esimerkiksi harjoitella siten, että toinen parista antaa toiselle tarkat ohjeet, miten edetä luokasta koulun ruokalaan" (V33) tai hyödyntää esimerkiksi tehtävien ohjeiden antamisessa, "ohjelmoinnin ajatusta voidaan hyödyntää vaikka luokan sääntöjen tai tehtävän ohjeiden esittämisessä" (V49).

Vastaajat kuvasivat, miten ensin harjoitellaan ohjelmointia sanallisesti, esimerkiksi antamalla kaverille toimintaohjeita ja tällaisten toiminnallisten harjoitusten jälkeen siirtyään harjoittelemaan ohjelmointia laitteiden ja sovellusten avulla, "Alkuvaiheessa ohjelmointia voi harjoitella antamalla tarkkoja ja yksityiskohtaisia toimintaohjeita, jonka mukaan esim. pari toimii. Tällaisten harjoitusten jälkeen voidaan käyttää esim. laitteita ja sovelluksia, jotka on suunniteltu ohjelmoinnin opettamiseen" (V5).

6.2 Ohjelmointitaitojen kehitys

Ohjelmoinnin opetuksen tavoitteeksi ilmaistiin ohjelmoinnillisen ajattelun ja ohjelmointiosaamisen kehittyminen. Ohjelmoinnillinen ajattelu nähtiin perustaksi myöhemmälle ohjelmointiosaamiselle, "Näen että pohja ohjelmoinnilliselle ajattelulle luodaan jo varhain ja myöhemmin lähdetään harjoittelemaan monimutkaisempia asioita. Kuitenkin täytyy olla ymmärrys ohjelmoinnillisesta ajattelusta ennenkuin pystyy tehdä hankalampia asioita" (V3). Ohjelmoinnin ja ohjelmoinnin opetuksen ajatellaan kehittävän oppilaiden loogista ajattelua ja loogisten yhteyksien löytämistä, syys-seuraussuhteiden ymmärtämistä, ongelmanratkaisutaitoja sekä ylipäättänsä ajatteluntaitoja, "Ohjelmoinnin opettaminen on mielestäni syys-seuraussuhteiden opettelua. Siinä opetellaan, kuinka tietyt asiat vaikuttavat tiettyihin asioihin" (V37) ja " Ohjelmointi myös kehittää ajattelun taitoja, loogista päättelyä ja ongelmanratkaisutaitoja" (V7).

Pelkän ohjelmoinnillisen ajattelun oppimisen lisäksi vastaajat pitivät tärkeänä, että oppilaat ymmärtäisivät, mitä ohjelmointi konkreettisesti tarkoittaa ja miten se näkyy arjessa, "...käytännöllistä opetusta, jotta oppilas ymmärtää, mitä oikeasti tapahtuu kun hän ohjelmoi ja miten ohjelmointia voi hyödyntää "oikeassa" elämässä" (V22). Vastaajien mukaan ohjelmoinnin opetuksessa tulisi opettaa vähintään ohjelmoinnin perusasiat, jotta jokainen oppilas saa käsityksen mitä ohjelmointi tarkoittaa. Ohjelmoinnin perusasioiden opettamisen jälkeen voidaan edetä haastavampiin harjoituksiin, "Ohjelmoinnin opettamisessa tärkeää on, että opetetaan ohjelmoinnin perusidea ja sen jälkeen edetään haastavampiin juttuihin. Aion opettaa ohjelmointia koulussa lähtien liikkeelle perusasioista ja perustaidoista hyödyntäen konkretiaa... Siitä sitten pikkuhiljaa siirrytään ohjelmointiin teknologian avustuksella" (V60).

Ohjelmoinnin opetuksessa pidettiin tärkeänä ohjelmoinnillisen ajattelun harjoittelemista varsinaisen ohjelmoinnin avulla, "Ohjelmoinnin ajattelun ymmärtäminen ja ajattelun vieminen käytännön ohjelmointiin on tärkeää" (V41). Muutama vastaaja ajattelin ohjelmoinnin opetuksen koulussa sisältävän ohjelmoinnillisen ajattelun lisäksi myös varsinaisen koodaamisen opettamista oppilaille, "Ohjelmoinnin opettaminen on ohjelmoinnillisen ajattelun ja koodaamisen

opettelu.” (V21). Ohjelmoinnin opetus nähtiin myös tietynlaisen kielen opettamisena, ”Opettajana toisin juurikin sen periaatteen, että ohjelmointi on tietynlaista kielen opettelu” (V32).

Osa vastaajista kuitenkin ajatteli, ettei koulun ohjelmoinnin opetuksessa tarvitse opetella varsinaisesti ohjelmointia vaan keskittyä pelkästään ohjelmoinnillisen ajattelun ymmärtämiseen, ”Ohjelmoinnin opettaminen on mielestäni enemmänkin ohjelmoinnillisen ajattelun opettamista kuin itse varsinaista ohjelmointia...Varsinaiseen koodaamiseen ei ole syytä mennä, vaan on parempaa käyttää sovelluksia, joissa koodi on valmiina” (V35). Vaikka ohjelmoinnin opetukseen ei nähty välttämättä sisältyvän ohjelmointikielten opettamista, nähtiin kuitenkin antavan valmiuksia niiden opiskeluun tulevaisuudessa, ”Ohjelmoinnin opettaminen ei niinkään ole yksittäisen ohjelmointikielen opettamista, vaan sellaisten työkalujen antamista, että oppilaat kehittävät omaa ohjelmoinnillista ajattelua siihen suuntaan, että ohjelmointikielten opiskelu mahdollistuu” (V58).

6.3 Ohjelmointi arjessa

Vastaajat pitivät olennaisena ohjelmoinnin opetuksessa sitä, että oppilaat ymmärtäisivät ohjelmoinnin olevan osa jokapäiväistä elämää, eikä arjesta irrallinen asia, jota opiskellaan vain satunnaisesti koulussa, ”Opettamisessa on tärkeää ettei aihe jää erilliseksi leijumaan ilmaan vaan oppilaille on saatava luotsattua ymmärrys miten ohjelmointi kuuluu arkeen ja omaan elämään” (V6). Vastaajat pitivät tärkeänä, että oppilaat ymmärtäisivät, miten ohjelmointi näkyy koulun ulkopuolella esimerkiksi eri laitteiden toiminnassa, ”...kytkeä opetusta arjen tilanteisiin ikätasoisesti ja koettaa tuottaa lapsille käsityksen siitä, että se on jokapäiväistä lähellämme olevaa, tottakai pelien ja robotiikan avulla, mutta myös toinen toisillemme koodauskieltä käyttäen ohjausta antaen.”(V6).

Merkittäväksi asiaksi nousi myös se, että opettaja opettaa oppilaille, ettei ohjelmoinnin ajateltaisi olevan vain vaikeaa koodauskieltä ja koodaamista vaan oppilaille koko ajan yhä tärkeämpi taito tulevaisuudessa, ”Mielestäni tärkeää on

myös saada oppilaat ymmärtämään, että käytämme ohjelmointia elämässämme päivittäin. Eli ohjelmointi ei ole vain vaikeaa koodauskieltä” (V7). Myös ohjelmointiin liittyvien negatiivisten mielikuvien häivyttäminen nähtiin tärkeäksi osaksi opetusta, ”Ohjelmoinnin opettamisessa tärkeää on sen leiman ja stresotypian [stereotypian] vähentäminen. Se ei ole vain koodaamista tms vaan hyödyllinen arjen taito” (V16). Vastaajat pitivät merkittävänä sitä, että ohjelmoinnin opetus tuodaan lähelle oppilaiden kokemusmaailmaa, samoin kuin muukin opetus sekä, että oppilaille tuodaan ilmi, että ohjelmointitaitojen osaaminen on tärkeää myös koulun ulkopuolella, ”Uskon, että ohjelmoinnin taidot ovat hyödyllisiä, joita voi käyttää myös muissa yhteyksissä esim. koulun ulkopuolella” (V57).

7 POHDINTA

7.1 Tulosten tarkastelu ja johtopäätökset

Tutkimuksen tavoitteena oli selvittää luokanopettajaopiskelijoiden kokemuksia ja ajatuksia itsestään ohjelmoinnin opettajina sekä sitä, millaiseksi luokanopettajaopiskelijat kokevat oman osaamisensa ohjelmoinnin opettajina. Tutkimuksen tulosten mukaan suurin osa tutkittavista koki ohjelmoinnin itselleen vieraaksi aiheeksi ja näin ollen myös se opettamisen haasteeksi itselleen. Pienempi osa vastaajista kuvasi omaa osaamistaan kohtalaiseksi ja vain pari vastaajaa kuvasi omaavansa hyvät taidot opettaa ohjelmointia. Tutkimuksen toisena tavoitteena oli selvittää, mitä luokanopettajaopiskelijat ajattelevat ohjelmoinnin opetuksen olevan vuosiluokilla 1–6 ja miten he tulevat sitä opetuksessaan toteuttamaan. Tuloksissa nousi esille, että laitteilla ja sovelluksilla nähtiin olevan merkitystä ohjelmoinnin opetuksessa, mutta ohjelmointia voidaan opettaa myös ilman laitteita. Ohjelmoinnin opetuksen tavoitteeksi esitettiin oppilaiden ohjelmointitaitojen kehittäminen sekä ohjelmoinnin ymmärtäminen osana arkipäiväisiä toimintoja.

Useat tutkimukseni tulokset vahvistavat aiempia tutkimustuloksia. Kuten aiemmissa tutkimuksissa on havaittu, opettajilla on puutteita ohjelmoinnin sisältötietämyksessä ja ohjelmoinnillisen ajattelun käsitteen ymmärtämisessä sekä selkeässä näkemyksessä siitä, mitä ohjelmointi koulussa on (Pörn ym., 2021; Kite & Park, 2022; Morze ym., 2022). Korhosen ym. (2022) mukaan opettajat pitivät ohjelmoinnin opetusta monimutkaisena, koska opettajat kokivat epävarmuutta ja puutteellista osaamista siihen liittyvien sisältöjen sekä työkalujen ja ohjelmien suhteen. Tutkimukseni tulokset osoittavat samanlaisia havaintoja: suuri osa tutkittavista ilmaisi puutteita omassa ohjelmoinnillisessa osaamisessa ja jopa ymmärryksessä siitä, mitä ohjelmointi on. Näin ollen he kokevat myös ohjelmoinnin opettamisen haasteelliseksi, kun oma ymmärryskin on vielä puutteellista. Tutkimuksessani nousi esiin myös useampia vastauksia, joiden mukaan ohjelmointi ja koodaus ovat tutkittaville täysin tuntematon aihe. Samanlainen havainto on tehty myös aiemmassa tutkimuksessa. Wun ym., (2020) vertaillessa eri maiden

opettajien valmiuksia opettaa koodaustaitoja useat suomalaiset opettajat kokivat, että koodaaminen on heille täysin tuntematon aihe.

Tässä tutkimuksessa tuli ilmi, kuten aiemmissakin tutkimuksissa (Pörn ym., 2021; Morze ym., 2022), että opettajat ovat kiinnostuneita oppimaan lähestymistapoja, jotka voisivat auttaa ohjelmoinnillisen ajattelun toteuttamisessa, vaikka osalla opettajista on väärinymmärrystä ohjelmoinnillisen ajattelun käsitteestä. Vaikka tutkimuksessani tuli ilmi, että opettajaopiskelijoilla on puutteita ohjelmoinnillisessa osaamisessa, suhtautuvat luokanopettajaopiskelijat myönteisesti ohjelmoinnin opettamiseen ja oman ohjelmoinnillisen osaamisen kehittämiseen. Tämä havainto tukee aiempaa tutkimusta, jonka mukaan opettajat suhtautuivat pääosin myönteisesti tieto- ja viestintäteknologian käyttöön opetuksessa, vaikka vain osa tutkimukseen osallistuneista opettajista itse hyödyntää tieto- ja viestintäteknologiaa ja kokee hallitsevansa sen käytön (Leino ym., 2019).

Tutkimukseni mukaan luokanopettajaopiskelijat pitävät ohjelmoinnin osaamista tärkeänä tulevaisuuden taitona, jota voidaan tarvita myös työelämässä. Tämä tulos tukee aiemmissa tutkimuksissa saatuja tuloksia, joiden mukaan opettajat pitivät ohjelmoinnin opetusta hyvänä asiana, koska se haastaa ja motivoi oppilaita oppimaan uusia taitoja, joita tarvitaan nyt ja tulevaisuudessa yhteiskunnassa sekä työelämässä (Pörn ym., 2021; Kite & Park, 2022; Korhonen ym., 2022). Aiemman tutkimuksen mukaan opettajat uskoivat, että ohjelmointitaidosta on hyötyä, vaikka oppilaat eivät tulisikaan työskentelemään tieto- ja viestintäteknologian alalla (Wu ym., 2020).

Tutkimukseni tulosten mukaan luokanopettajaopiskelijat tulevat hyödyntämään ohjelmoinnin opetuksessa erilaisia sovelluksia, pelejä ja laitteita sekä muita oppilaita osallistavia menetelmiä. Tätä havaintoa tukee myös Wun ym. (2020) aiempi tutkimus, jonka mukaan suomalaiset opettajat hyödyntävät opetussovelluksia ja pelejä ohjelmoinnin opetuksessa sekä suosivat opetuksessaan muitakin aktiivisia opetusmenetelmiä opettaessaan ohjelmointitaitoja. On mahdollista, että luokanopettajaopiskelijoiden oman ohjelmointiosaamisen ollessa

puutteellista ja opettajien kokiessa epävarmuutta itsestään ohjelmoinnin opettajina, he turvautuvat valmiiseen materiaaliin ja sovelluksiin, jotka ohjaavat oppilaita oppimaan ohjelmointia ja ohjelmoinnillista osaamista.

Tässä tutkimuksessa tuli ilmi, että vain pieni osa tutkittavista kokee osavansa opettaa ohjelmointia kohtalaisesti tai hyvin. Näin ollen on mahdollista, että vain osa oppilaista saa laadukasta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta. Tämä tutkimustulos tukee Leinon ym., (2019) huolta siitä, että opettajien osaamisen ollessa puutteellista vain osa oppilaista saa opetusta, joka hyödyntää pedagogisesti järkevällä tavalla teknologiaa opetuksessa, ja näin ollen tämä luo epätasa-arvoa oppilaiden välille. Haasteen ohjelmoinnin opetukselle luo myös se, että opettajat usein liittävät ohjelmoinnin ja ohjelmoinnin opetuksen vain matematiikan opetukseen, vaikka sen tulisi laaja-alaisen Tieto- ja viestintäteknologinen osaaminen (L5) -osaamiskokonaisuuden kautta olla osa kaikkia oppiaineita. Useat tutkimukseeni osallistuneet luokanopettajaopiskelijat liittivät ohjelmoinnin opetuksen osaksi matematiikan opetusta ja muutamat liittivät sen osaksi käsityön ja liikunnan oppiaineita. Tämä havainto tukee aiempaa tutkimusta (Misfeld ym., 2019), jonka mukaan matematiikan opettajat tyypillisesti kokevat yhteyden ohjelmoinnin ja matematiikan opetuksen välillä.

Tutkimukseni tulosten mukaan opettajaopiskelijat ajattelevat ohjelmoinnin opetuksen opettavan ohjelmointitaidon lisäksi loogisen ajattelun taitoja, syysseuraussuhteiden ymmärtämistä, ajatteluntaitoja sekä ongelmanratkaisutaitoja. Tämä havainto tukee aiemmissa tutkimuksissa saatua tietoa, jonka mukaan opettajat katsovat ohjelmoinnin edistävän loogisen ajattelun kehittymistä sekä toimivan työkaluna ongelmanratkaisussa (Pörn ym., 2021; Kite & Park, 2022). Leinon ym. (2019) mukaan ohjelmoitaessa ongelmien ratkaisemisen, loogisen ajattelun, luovuuden, kekseliäisyyden ja yhteistyötaitojen lisäksi opitaan huomaamaan tietynlaisia ratkaisukaavoja, joita voidaan käyttää uudelleen eri tilanteissa.

Ohjelmoinnilliseen osaamiseen kuuluu ohjelmoinnillisen ajattelun lisäksi tutkiva työskentely, ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen sekä monipuoliset ajattelun taidot (Uuden lukutaidot, 2022). Näitä taitoja oppilaat oppivat koulussa ja näiden taitojen opettaminen on osa opetusta, vaikka

opettajat eivät tietoisesti opeta ohjelmointia ja ohjelmoinnillista ajattelua. Tutkimukseni tulosten mukaan suuri osa opettajaopiskelijoista ilmaisi, ettei osaa opettaa ohjelmointia. On mahdollista, että he osaavat opettaa ja opettavat ohjelmointia osin ymmärtämättä sitä, että he opettavat sitä osana muiden taitojen opettamista ja oppimista. Myös aiempi Balin ym. (2022) tutkimus tukee tätä ajatusta. Tutkimuksen mukaan opettajat integroivat ohjelmoinnillista ajattelua luokassa jo ennen osallistumista täydennyskoulutukseen, mutta he eivät olleet tietoisia tästä integraatiosta ennen täydennyskoulutuksen suorittamista. Täydennyskoulutuksella on vaikutusta myös opettajien tietämykseen ja itseluottamukseen ohjelmoinnillisen ajattelun toteuttamisesta osana kaikkea opetusta (Bal ym., 2022). Myös tutkimukseni mukaan luokanopettajaopiskelijat olisivat halukkaita oppimaan itse lisää ohjelmoinnista ja sen opettamisesta, jotta he osaisivat opettaa sitä oppilailleen.

Tutkimukseni tulosten perusteella ohjelmointia tulisi opettaa opettajakoulutuksessa enemmän ja syvällisemmin, jotta opettajien ohjelmoinnin sisältöosaaminen paranisi ja itseluottamus ohjelmointia opettaessa lisääntyisi. Opettajaopiskelijoille olisi hyvä tarjota mahdollisuuksia täydennyskoulutukseen jo opintojen aikana, esimerkiksi yliopiston tarjoamia ohjelmointikursseja, joita voi suorittaa lähiopetuksena tai itsenäisesti verkossa. Ohjelmointiosaamisen lisäämiseksi opettajaopiskelijoille tulisi jakaa tietoa myös erilaisista yliopiston ulkopuolisista avoimista ohjelmointikursseista.

7.2 Tutkimuksen arviointi

Tutkimuksen luotettavuutta pohdittaessa on mietittävä, ovatko tutkimuksen luonne ja tutkimusaihe vaikuttaneet siihen, kuinka tutkimukseen osallistuneet ovat vastanneet (Saaranen-Kauppinen & Puusniekka, 2009). Omassa tutkimuksessani tutkittavien vastauksiin on voinut vaikuttaa se, että vastaajat ovat ajatelleet, että heidän pitäisi osata ohjelmointia ja sen opettamista, koska se kuuluu osaksi opetussuunnitelmaa. Näin ollen osa vastaajista on voinut kuvata omaa osaamistaan edes kohtalaiseksi sen sijaan, että he olisivat sanoneet sen olevan

heikkoa. Tästä ei kuitenkaan ollut vahvoja viitteitä, koska useimmat vastaajista ilmaisivat osaamisensa heikoksi.

Tutkimustulosten luotettavuuteen vaikuttaa se, kuinka hyvin luokat kattavat koko aineiston, eli mitään oleellista ei ole jätetty aineistosta pois eikä myöskään epäolennaista ole sisällytetty analysoitavaan aineistoon (Graneheim & Lundman, 2004). Tutkimuksessa muodostamiani luokat kattavat koko aineiston. Olen myös pyrkinyt kuvaamaan ja perustelemaan tarkasti tutkimukseni vaiheet tutkimuksen luotettavuuden lisäämiseksi. Tutkimuksessa tehtyjen valintojen perustelu sekä tarkka analyysin kuvaus auttavat tutkimuksen luotettavuuden arvioinnissa (Saaranen-Kauppinen & Puusniekka, 2009; Kyngäs ym., 2011). Heidän mukaansa tämä on tärkeää, jotta lukija pystyy tarkastelemaan, onko tutkimus edennyt sopivalla tavalla tutkimuksen tavoitteiden kannalta. Tutkimukseni raportoinnissa olen esittänyt esimerkkejä analyysiprosessista, jotka havainnollistavat sitä lukijalle. Salmonsin (2015) mukaan laadulliselle tutkimukselle tyypillinen suorien aineistolainauksen käyttö ilmaisee osallistujien kokemuksia alkuperäisessä muodossa. Tässä tutkimuksessa olen pyrkinyt lisäämään tulosten raportoinnin luotettavuutta käyttämällä suoria lainauksia aineistosta, jolloin minun tulkintani tutkijana ei ole vaikuttanut ilmauksiin. Tutkimuksen luotettavuutta lisää myös se, että tutkimuksessa käytetyt lähteet painottuvat kansainvälisiin refereearvioituihin tutkimusartikkeleihin.

Tutkimus on aina yhdenlainen versio tutkittavasta aiheesta, eikä se tarjoa objektiivista ja absoluuttista tietoa, vaan tutkimuksen tuloksista tehdään johtopäätöksiä, jotka ovat sidoksissa paikkaan, aikaan ja tukijaan (Saaranen-Kauppinen & Puusniekka, 2009). Tässä tutkimuksessa tarkasteltiin vain yhden yliopiston opiskelijoita, joten todennäköisesti heidän vastauksiinsa vaikuttaa kyseisessä yliopistossa vaikuttavat käytänteet. Näin ollen myöskään tämän tutkimuksen tulokset eivät ole objektiivista ja absoluuttista tietoa, vaan tuloksiin on vaikuttanut muun muassa tutkittavien käymän yliopiston opetussuunnitelma ja opiskelijoiden saama opetus. Tutkimuksen luotettavuuteen vaikuttaa myös se, että tulokset ovat yhden tutkijan tulkintaa. Olen kuitenkin saanut tutkimuksestani vertaispa-

lautetta, joka Creswellin ja Millerin (2000) mukaan lisää tutkimuksen luotettavuutta. Jos tutkimus toistettaisiin toisissa olosuhteissa tai tutkimusasetelmaa kehitettäisiin, voitaisiin nähdä, saadaanko samansuuntaisia tuloksia ja näin tutkimuksen luotettavuus paranisi (Saaranen-Kauppinen & Puusniekka, 2009).

Pääsin tutkimuksessani sen tavoitteeseen, eli sain käsityksen siitä, millaiseksi ohjelmoinnin opettajiksi luokanopettajaopiskelijat kuvaavat itseään sekä mitä he ajattelevat ohjelmoinnin opetuksen olevan vuosiluokilla 1–6. Tulokset antavat käsityksen luokanopettajaopiskelijoiden ohjelmoinnillisesta osaamisesta, mutta tuloksia ei voida yleistää suhteellisen pienen otoksen takia. Tuloksia ei voida myöskään yleistää, koska tutkittavat edustavat vain yhden yliopiston opiskelijoita. Tutkimustuloksiin vaikuttaa myös se, missä vaiheessa opintoja tutkittavat olivat tutkimukseeni osallistuessaan. Jos tätä aihetta tutkittaisiin lisää, olisi mielenkiintoista tietää, onko opintojen vaiheella merkitystä opiskelijoiden ohjelmoinnilliseen osaamiseen.

7.3 Jatkotutkimusaiheet

Lähestyn jatkotutkimusaiheita kolmesta eri näkökulmasta. Ensimmäiseksi tätä tutkimusta voitaisiin laajentaa koskemaan laajempaa joukkoa Suomen luokanopettajaopiskelijoita, jolloin saataisiin kokonaisvaltaisempi käsitys Suomen luokanopettajaopiskelijoiden ohjelmoinnillisesta osaamisesta. Tässä tutkimuksessa tutkittiin vain yhden yliopiston luokanopettajaopiskelijoita, joten saatiin käsitys vain kyseisen yliopiston luokanopettajaopiskelijoiden ohjelmoinnillisesta osaamisesta. Ottaen huomioon eri yliopistojen väliset mahdolliset erot ohjelmoinnin opetuksen painotuksissa, ei tämä tutkimus riitä kuvaamaan suomalaisten luokanopettajaopiskelijoiden ohjelmointiosaamista yleisellä tasolla.

Tässä tutkimuksessa tutkittiin luokanopettajaopiskelijoiden ajatuksia ja kuvauksia itsestään ohjelmoinnin opettajana, ennen kuin he ovat siirtyneet työelämään. Tutkimusta voisi jatkaa selvittämällä vastaavatko luokanopettajaopiskelijoiden ajatukset sitä, miten he todellisuudessa toteuttavat ohjelmoinnin opetusta kouluissa ja ovatko he ohjelmoinnin opettajina sellaisia kuin he nyt opiskelijoina

ajattelivat olevansa. Suurin osa tutkimukseen osallistuneista luokanopettajaopiskelijoista koki oman ohjelmointiosaamisensa huonoksi. Opetussuunnitelman tavoitteiden toteutumisen kannalta olisi tarpeen tutkia, vaikuttaako opettajien huonoksi koettu ohjelmointiosaaminen ja epävarmuus ohjelmoinnin opettamisesta esimerkiksi ohjelmoinnin opetuksen määrään ja jättävätkö opettajat ohjelmoinnin opetuksen toisten opettajien vastuulle tai jopa ohjelmoinnin opettamisen kokonaan pois opetuksestaan.

Kolmas näkökulma, josta tämän tutkimusaiheen tutkimista voisi jatkaa, on yliopiston rooli luokanopettajaopiskelijoiden ohjelmointiosaamisen koulutuksessa ja se, kuinka luokanopettajaopiskelijoiden ohjelmointivalmiuksia voisi parantaa. Tällöin voitaisiin selvittää, millä ohjelmoinnin opettamisen osa-alueella luokanopettajaopiskelijat kokevat suurimpia puutteita ja minkälaista koulutusta he kokevat tarvitsevänsä. Voitaisiin myös selvittää pitäisikö luokanopettajaopinnoissa olla enemmän koulutusta ohjelmoinnin opettamisesta vai kokisivatko he täydennyskoulutuksen hyödyllisemmäksi työelämään päästyään. Näiden kysymysten tutkiminen voisi osaltaan auttaa yliopistoja kehittämään ohjelmointiopetuksen valmiuksien koulutusta luokanopettajaopiskelijoille.

LÄHTEET

- Aaltio, I. & Puusa, A. (2020). Mitä laadullisen tutkimuksen arvioinnissa tulisi ottaa huomioon? Teoksessa A. Puusa & P. Juuti (toim.), *Laadullisen tutkimuksen näkökulmat ja menetelmät* (s. 177–188). Gaudeamus.
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in human behavior*, 105, artikkeli 106185.
<https://doi.org/10.1016/j.chb.2019.106185>
- Alibakhshi, G., Nikdel, F., & Labbafi, A. (2020). Exploring the consequences of teachers' self-efficacy: A case of teachers of English as a foreign language. *Asian-Pacific journal of second and foreign language education*, 5(1), 1–19. <https://doi.org/10.1186/s40862-020-00102-1>
- Bal, I. A., Alvarado–Albertorio, F., Marcelle, P., & Oaks–Garcia, C. T. (2022). Pre–service Teachers Computational Thinking (CT) and Pedagogical Growth in a Micro–credential: A Mixed Methods Study. *TechTrends*, 66(3), 468–482. <https://doi.org/10.1007/s11528-022-00732-x>
- Bandura, A. (2002). Social foundations of thought and action. Teoksessa D. Marks (toim.), *The health psychology reader* (s. 94–107). SAGE.
<https://doi.org/10.4135/9781446221129>
- Bandura, A. (2006). Adolescent Development from an Agentic Perspective. Teoksessa F. Pajares, & T. Urdan (toim.), *Self-Efficacy Beliefs of Adolescents* (s. 1–43). Greenwich, CT: Information Age Publishing.
- Boom, K., Bower, M., Siemon, J., & Arguel, A. (2022). Relationships between computational thinking and the quality of computer programs. *Education and information technologies*, 27(6), 8289–8310.
<https://doi.org/10.1007/s10639-022-10921-z>
- Cutumisu, M., Adams, C., Glanfield, F., Yuen, C., & Lu, C. (2022). Using Structural Equation Modeling to Examine the Relationship Between Preservice Teachers' Computational Thinking Attitudes and Skills. *IEEE transactions on education*, 65(2), 177–183.
<https://doi.org/10.1109/TE.2021.3105938>

- Creswell, J. W., & Miller, D. L. (2000). Determining Validity in Qualitative Inquiry. *Theory into practice*, 39(3), 124–130.
https://doi.org/10.1207/s15430421tip3903_2
- Esteve-Mon, F. M., Llopis, M. Á., & Adell-Segura, J. (2020). Digital Competence and Computational Thinking of Student Teachers. *International journal of emerging technologies in learning*, 15(2), 29–41.
<https://doi.org/10.3991/ijet.v15i02.11588>
- Fagerlund, J. (2021). *Teaching, learning and assessing computational thinking through programming with Scratch in primary schools*. [Väitöskirja, Jyväskylän yliopisto]. <https://jyx.jyu.fi/handle/123456789/78190>
- Fagerlund, J. (2022). Tietokonevallankumous ja ohjelmoinnillinen ajattelu peruskoulussa – Havaintoja mikro- ja makrotasoilta. *Kasvatus & Aika*, 16(1), 121–127.
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer applications in engineering education*, 29(1), 12–28.
<https://doi.org/10.1002/cae.22255>
- Fagerlund, J., Leino, K., Kiuru, N., & Niilo-Rämä, M. (2022). Finnish teachers' and students' programming motivation and their role in teaching and learning computational thinking. *Frontiers in Education*, 7. <https://doi.org/10.3389/educ.2022.948783>
- Flórez, F. B., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research*, 87(4), 834–860.
<https://doi.org.ezproxy.jyu.fi/10.3102/0034654317710096>
- Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2019). *IEA International Computer and Information Literacy Study 2018 Assessment Framework*. <https://doi.org/10.1007/978-3-030-19389-8>
- Graneheim, U., & Lundman, B. (2004). Qualitative content analysis in nursing research: Concepts, procedures and measures to achieve

trustworthiness. *Nurse education today*, 24(2), 105–112.

<https://doi.org/10.1016/j.nedt.2003.10.001>

Granziera, H., & Perera, H. N. (2019). Relations among teachers' self-efficacy beliefs, engagement, and work satisfaction: A social cognitive view. *Contemporary educational psychology*, 58, 75–84.

<https://doi.org/10.1016/j.cedpsych.2019.02.003>

Heikkinen, H., Huttunen, R., Niglas, K., & Tynjälä, P. (2005). Kartta kasvatustieteen maastosta. *Kasvatus*, 36(5), 340–354.

Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and education*, 126, 296–310.

<https://doi.org/10.1016/j.compedu.2018.07.004>

Hämäläinen, R., Nissinen, K., Mannonen, J., Lämsä, J., Leino, K., & Taajamo, M. (2021). Understanding teaching professionals' digital competence: What do PIAAC and TALIS reveal about technology-related skills, attitudes, and knowledge?. *Computers in Human Behavior*, 117, artikkeli

106672. <https://doi.org/10.1016/j.chb.2020.106672>

Jiang, B., & Li, Z. (2021). Effect of Scratch on computational thinking skills of Chinese primary school students. *Journal of computers in education (the official journal of the Global Chinese Society for Computers in Education)*, 8(4), 505–525. <https://doi.org/10.1007/s40692-021-00190-z>

Jyväskylän yliopisto. (2015). *Sosiaalinen konstruktioismi*.

<https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tieteenfilosofiset-suuntaukset/sosiaalinen-konstruktioismi>

Jyväskylän yliopisto. (2023). *Aineistonhallinnan palvelut ja työkalut*.

<https://openscience.jyu.fi/fi/tutkimusdata/palvelut-ja-tyokalut>

Kite, V., & Park, S. (2022). What's Computational Thinking?: Secondary Science Teachers' Conceptualizations of Computational Thinking (CT) and Perceived Barriers to CT Integration. *Journal of science teacher education*, 1–24. Julkaistu ennakkoon verkossa.

<https://doi.org/10.1080/1046560X.2022.2110068>

- Koodiaapinen. (2023). *Opettajan opas koodaamiseen koulussa*.
<https://koodiaapinen.fi/ukk/>
- Kong, S., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers and education*, 151, artikkeli 103872.
<https://doi.org/10.1016/j.compedu.2020.103872>
- Kong, S., & Lai, M. (2023). Effects of a teacher development program on teachers' knowledge and collaborative engagement, and students' achievement in computational thinking concepts. *British journal of educational technology*, 54(2), 489–512. <https://doi.org/10.1111/bjet.13256>
- Korhonen, T., Salo, L., Laakso, N., Seitamaa, A., Sormunen, K., Kukkonen, M., & Forsström, H. (2022). Finnish teachers as adopters of educational innovation: Perceptions of programming as a new part of the curriculum. *Computer science education*, 1–23. Julkaistu ennakkoon verkossa. <https://doi.org/10.1080/08993408.2022.2095595>
- Krippendorff, K. (2019). *Content Analysis: An Introduction to Its Methodology*.
<https://doi.org/10.4135/9781071878781>
- Kuula, A. (2006). *Tutkimusetiikka: Aineistojen hankinta, käyttö ja säilytys*.
 Vastapaino.
- Kyngäs, H., Elo, S., Pölkki, T., Kääriäinen, M., & Kanste, O. (2011).
 Sisällönanalyysi suomalaisessa hoitotieteen tutkimuksessa. *Hoitotiede*, 23(2), 138–148.
- Leino, K., Rikala, J., Puhakka, E., Niilo-Rämä, M., Siren, M., & Fagerlund, J. (2019). *Digiloikasta digitaitoihin: Kansainvälinen monilukutaidon ja ohjelmoinnillisen ajattelun tutkimus (ICILS 2018)*. Koulutuksen tutkimuslaitos.
<https://jyx.jyu.fi/bitstream/handle/123456789/66250/978-951-39-7937-9.pdf?sequence=1&isAllowed=y>
- Lindberg, R. S. N., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries

- and programming games. *British journal of educational technology*, 50(4), 1979–1995. <https://doi.org/10.1111/bjet.12685>
- Lloyd, M., & Chandra, V. (2020). Teaching coding and computational thinking in primary classrooms: Perceptions of Australian preservice teachers. *Curriculum perspectives*, 40(2), 189–201. <https://doi.org/10.1007/s41297-020-00117-1>
- Lorenz, R., Heldt, M., & Eickelmann, B. (2022). Relevance of pre-service teacher training to use ICT for the actual use in classrooms - focus on German secondary schools. *Technology, pedagogy and education*, 31(5), 563–577. <https://doi.org/10.1080/1475939X.2022.2129772>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in human behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4). <https://citejournal.org/volume-19/issue-4-19/general/preparing-elementary-school-teachers-to-teach-computing-coding-and-computational-thinking>
- McLennan, B., McIlveen, P., & Perera, H. N. (2017). Pre-service teachers' self-efficacy mediates the relationship between career adaptability and career optimism. *Teaching and teacher education*, 63, 176–185. <https://doi.org/10.1016/j.tate.2016.12.022>
- Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer science education*, 25(4), 325–350. <https://doi.org/10.1080/08993408.2015.1111645>
- Mertala, P., Palsa, L., & Slotte Dufva, T. (2020). Monilukutaito koodin purkajana: Ehdotus laaja-alaiseksi ohjelmoinnin pedagogiikaksi. *Media & viestintä*, 43(1), 21–46. <https://doi.org/10.23983/mv.91079>

- Misfeldt, M., Szabo A., & Helenius, O. (2019). Surveying teachers' conception of programming as a mathematics topic following the implementation of a new mathematics curriculum. Teoksessa U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (toim.), *Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education* (s. 2713–2720). Utrecht, the Netherlands: Freudenthal Group & Freudenthal Institute, Utrecht University and ERME. http://www.mathematik.tu-dortmund.de/~prediger/ERME/CERME11_Proceedings_2019.pdf
- Morze, N., Barna, O., & Boiko, M. (2022). The importance of computational thinking training for primary school teachers. *Educational Dimension*, 58, 22–39. <https://doi.org/10.31812/educdim.4466>
- Mouza, C., Yang, H., Pan, Y., Yilmaz Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, 33(3). <https://doi.org/10.14742/ajet.3521>
- Opetushallitus (2014). *Perusopetuksen opetussuunnitelman perusteet*. Määräykset ja ohjeet 96. Opetushallitus.
- Perera, H. N., & John, J. E. (2020). Teachers' self-efficacy beliefs for teaching math: Relations with teacher and student outcomes. *Contemporary educational psychology*, 61, artikkeli 101842. <https://doi.org/10.1016/j.cedpsych.2020.101842>
- Pollak, M., & Ebner, M. (2019). The Missing Link to Computational Thinking. *Future internet*, 11(12), 263. <https://doi.org/10.3390/fi11120263>
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers and education*, 128, 365–376. <https://doi.org/10.1016/j.compedu.2018.10.005>
- Puusa, A. & Juuti, P. (2020). Laadullisen tutkimuksen tieteenfilosofinen tausta. Teoksessa A. Puusa & P. Juuti (toim.), *Laadullisen tutkimuksen näkökulmat ja menetelmät*. (s. 25–55). Gaudeamus.

- Pörn, R., Hemmi, K., & Kallio-Kujala, P. (2021). Inspiring or confusing – a study of Finnish 1–6 teachers' relation to teaching programming. *LUMAT*, 9(1). <https://doi.org/10.31129/LUMAT.9.1.1355>
- Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers and education*, 168, artikkeli 104196. <https://doi.org/10.1016/j.compedu.2021.104196>
- Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Saaranen-Kauppinen, A., & Puusniekka, A. (2009). *Menetelmäopetuksen tietovaranto KvaliMOTV: Kvalitatiivisten menetelmien verkko-oppikirja* (Toinen vedos.). Yhteiskuntatieteellinen tietoarasto Tampereen yliopisto.
- Salmons, J. (2015). *Doing Qualitative Research Online*. SAGE Publications Ltd.
- Schreier, M. (2014). Qualitative Content Analysis. Teoksessa U. Flick (toim.), *The SAGE Handbook of Qualitative Data Analysis* (s.170–183). SAGE Publications Ltd.
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and information technologies*, 22(2), 469–495. <https://doi.org/10.1007/s10639-016-9482-0>
- Skaalvik, E., & Skaalvik, S. (2019) Teacher Self-Efficacy and Collective Teacher Efficacy: Relations with Perceived Job Resources and Job Demands, Feeling of Belonging, and Teacher Engagement. *Creative Education*, 10, 1400–1424. <https://www.scirp.org/journal/paperinformation.aspx?paperid=93609#ref10>
- Tanhua-Piironen, E., Kaarakainen, S-S., Kaarakainen, M-T., & Viteli, J. (2020). Digiajan peruskoulu II: Opetus- ja kulttuuriministeriön julkaisuja 2020: 17. https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/162236/OK_M_2020_17.pdf


- The International Society for Technology in Education. (2020). Computational thinking competencies. <https://www.iste.org/standards/computational-thinking>
- Tondeur, J., Aesaert, K., Prestridge, S., & Consuegra, E. (2018). A multilevel analysis of what matters in the training of pre-service teacher's ICT competencies. *Computers and education*, 122, 32–42. <https://doi.org/10.1016/j.compedu.2018.03.002>
- Tuomi, J., & Sarajärvi, A. (2018). *Laadullinen tutkimus ja sisällönanalyysi* (Uudistettu laitos.). Kustannusosakeyhtiö Tammi.
- Uudet lukutaidot. (2022). Ohjelmointiosaaminen. <https://uudetlukutaidot.fi/ohjelmointiosaaminen-2/>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wu, L., Looi, C., Multisilta, J., How, M., Choi, H., Hsu, T., & Tuomi, P. (2020). Teacher's Perceptions and Readiness to Teach Coding Skills: A Comparative Study Between Finland, Mainland China, Singapore, Taiwan, and South Korea. *The Asia-Pacific education researcher*, 29(1), 21–34. <https://doi.org/10.1007/s40299-019-00485-x>
- Yadav, A., Gretter, S., Hambruch, S., & Sands, P. (2017). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer science education*, 26(4), 235–254. <https://doi.org/10.1080/08993408.2016.1257418>
- Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British journal of educational technology*, 48(3), 789–801. <https://doi.org/10.1111/bjet.12453>
- Zee, M., & Koomen, H. M. Y. (2016). Teacher Self-Efficacy and Its Effects on Classroom Processes, Student Academic Adjustment, and Teacher Well-Being: A Synthesis of 40 Years of Research. *Review of educational research*, 86(4), 981–1015. <https://doi.org/10.3102/0034654315626801>

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers and education*, 141, artikkeli 103607. <https://doi.org/10.1016/j.compedu.2019.103607>

LIITTEET

Liite 1. Aineistonkeruulomake

Ohjelmointi alakoulussa

 Pakolliset kysymykset merkitty tähdellä (*)

1. Olen perehtynyt tutkimusta koskevaan tiedotteeseen (tietosuojailmoitus) ja saanut riittävästi tietoa tutkimuksesta ja sen toteuttamisesta. Hyväksyn tietojeni käytön tietosuojailmoituksessa kuvattuun tutkimukseen. Minulla on ollut riittävästi aikaa harkita tutkimukseen osallistumista. Ymmärrän, että tähän tutkimukseen osallistuminen on vapaaehtoista. Minulla on oikeus, milloin tahansa tutkimuksen aikana ja syytä ilmoittamatta keskeyttää tutkimukseen osallistuminen tai peruuttaa suostumukseni tutkimukseen. Tutkimuksen keskeyttämisestä tai suostumuksen peruuttamisesta ei aiheudu minulle kielteisiä seuraamuksia. *

Kyllä

**2. Pohdi ja kuvaile itseäsi ohjelmoinnin opettajana. Pohdi esimerkiksi:
Millaisena ohjelmoinnin opettajana näet itsesi?
Mitä ohjelmoinnin opettaminen on?
Mikä ohjelmoinnin opettamisessa on tärkeää?
Miten aiot opettaa ohjelmointia alakoulussa? ***

Lähetä

Liite 2. Ensimmäisen tutkimuskysymyksen sisällönanalyysi

Pelkistetty ilmaus	Alaluokka	Yläluokka	Päälouokka
<p>Tavanomaista parempi osaaminen</p> <p>Hyvät valmiudet</p> <p>Osaa opettaa ohjelmoinnillista ajattelua</p> <p>Osaa hyödyntää erilaisia laitteita</p> <p>Pystyy jakamaan omaa osaamistaan muille</p>	Keskivertoa paremmat taidot	Hyvät valmiudet	Omien valmiuksien kuvaaminen
<p>Melko paljon tietoa perusteista</p> <p>Osaaminen alkuopetuksen tasolla</p> <p>Kohtuullinen osaaminen</p> <p>Yhdessä toisen opettajan kanssa</p>	<p>Osaa perusteet</p> <p>Aloittelija</p>	Kohtalaiset valmiudet	
<p>En ollenkaan tiedä, miten opettaa</p> <p>En tiedä asiasta mitään</p> <p>Oma taitotaso on huono</p> <p>Vieras käsite</p> <p>Todella epävarma</p>	<p>Heikko osaaminen</p> <p>Ei osaa opettaa ohjelmointia</p> <p>Epävarmuus koko käsitteestä</p>	Heikot valmiudet	
<p>Haluaa oppia ohjelmoinnin opetusta</p> <p>Motivoitunut kehittämään omia taitojaan</p> <p>Kokeilee rohkeasti uusia laitteita</p>	<p>Myönteinen ja innostunut suhtautuminen</p> <p>Halu oppia lisää</p>	Kehitysmuutos	Opettajan ominaisuudet
<p>Oppilaiden innostaminen tutkimiseen ja kokeilemiseen</p> <p>Kärsivällisyys, kannustaminen, ohjaaminen</p> <p>Tarpeeksi yksinkertaisia ja motivoivia harjoitteita</p> <p>Erilaisten taitotasojen huomioiminen</p>	<p>Sopivat harjoitteet</p> <p>Opettajan innostava suhtautuminen opettamiseen</p>	Oppilaita huomioiva opettaja	
<p>Tärkeää tulevaisuuden koulutuksessa</p> <p>Taito, jota tarvitaan tulevaisuudessa työelämässä</p> <p>Opetus tärkeää teknologisoituvassa maailmassa</p>	<p>Tärkeää tulevaisuutta ajatellen</p> <p>Tärkeää työllistymisen kannalta</p>	Ohjelmoinnin merkitys peruskoulun jälkeisessä elämässä	Tavoitteellinen ohjelmoinnin opetus

Pelkistetty ilmaus	Alaluokka	Yläluokka	Pääluokka
Alkuopetuksesta alkaen Opetus eri luokka-asteilla Ohjelmoinnin opettaminen osana eri oppiaineita	Opettaminen läpi vuosiluok- kien 1-6 Integrointi osaksi eri oppi- aineita	Ohjelmoinnin opetuksen jat- kumo	Tavoitteellinen ohjelmoinnin opetus

Liite 3. Toisen tutkimuskysymyksen sisällönanalyysi

Pelkistetty ilmaus	Alaluokka	Yläluokka	Pääluokka
<p>Tutustuminen teknisiin alustoihin</p> <p>Sovellukset, tietokonepelit</p> <p>Bee-botit, robotit</p> <p>Toimintaohjeiden antaminen koneelle</p>	<p>Laitteiden ja sovelusten avulla opettaminen</p>	<p>Laitteilla on rooli</p>	<p>Laitteiden merkitys</p>
<p>Toimintaohjeiden antaminen kaverille</p> <p>Leikit, lautapelit</p> <p>Luonnossa ja ulkona ohjelmointi</p> <p>Matematiikankirjan ohjelmointitehtävät</p>	<p>Toiminnalliset harjoitukset</p> <p>Kirjalliset ohjelmointitehtävät</p>	<p>Laitteilla ei ole roolia</p>	
<p>Koodien kirjoittaminen</p> <p>Ohjelmointikielien harjoittelu</p>	<p>Ohjelmointikielten harjoittelu</p> <p>Oikeiden termien käyttäminen</p>	<p>Tavoitteena ohjelmoinnillisten käsitteiden ja ohjelmointikielten oppiminen</p>	<p>Ohjelmointitaitojen kehitys</p>
<p>Ohjelmoinnillisen ajattelun oppiminen</p> <p>Ajattelun opettamista, loogisten yhteyksien löytämistä</p>	<p>Luoda perusta ohjelmoinnilliselle ajattelulle</p> <p>Työkaluja kehittää ohjelmoinnillista ajattelua</p>	<p>Tavoitteena ohjelmoinnillisen ajattelun kehittyminen</p>	
<p>Ohjelmointia voi hyödyntää kaikessa opetuksessa</p> <p>Ohjelmoinnin avulla voi opettaa oppisisältöjä</p>	<p>Tiedostaa mihin ohjelmointia voisi soveltaa opetuksessa</p> <p>Ohjelmointi eri oppiaineissa</p>	<p>Integrointi osaksi eri oppiaineita</p>	<p>Ohjelmointitaitojen hyödynnettävyys</p>
<p>Kytkeminen arjentalteisiin</p> <p>Ohjelmointi on jokapäiväistä ja lähellä olevaa</p> <p>Ohjeiden tekeminen ja antaminen on ohjelmointia</p>	<p>Osa arkea päivittäin käyttämämme laitteissa</p> <p>Ohjelmointitaidot koulun ulkopuolella</p> <p>Ohjelmointi eri ammateissa</p>	<p>Ohjelmointi arjessa</p>	

Liite 4. Tutkimuksen tietosuojailmoitus



Tietosuoja-asetus (679/2016) 12-14, 30 artikla

TIETOSUOJAILMOITUS TUTKIMUKSESTA TUTKIMUKSEEN OSALLISTUVALLE

11.4.2021

Tutkimukseen osallistuminen on vapaaehtoista, eikä tutkittavan ole pakko toimittaa mitään tietoja, tutkimukseen osallistumisen voi keskeyttää.

1. TUTKIMUKSEN NIMI, LUONNE JA KESTO

Tutkimuksessa pyritään selvittämään, millaisia käsityksiä maisterivaiheen luokanopettajaopiskelijoilla on ohjelmoinnin opettamisesta alakoulussa. Tutkimus on kertatutkimus. Tutkimustulokset valmistuvat syksyllä 2022.

2. MIHIN HENKILÖTIETOJEN KÄSITTELY PERUSTUU

EU:n yleinen tietosuoja-asetus, artikla 6, kohta 1

Tutkittavan suostumus

3. TUTKIMUKSESTA VASTAAVAT TAHOT

Annina Turtiainen, annina.a.turtiainen@student.jyu.fi

Tutkimuksen ohjaaja: Ulla Valleala, ulla.valleala@jyu.fi

4. TUTKIMUKSEN TAUSTA JA TARKOITUS

Tämän tutkimuksen tavoitteena on selvittää, millaisia käsityksiä maisterivaiheen luokanopettajaopiskelijoilla on ohjelmoinnin opettamisesta alakoulussa. Tutkimukseen osallistuvat henkilöt, jotka opiskelevat luokanopettakoulutuksen maisterivaiheessa. Tutkimukseen osallistuu noin 20 tutkittavaa.

5. TUTKIMUKSEN TOTEUTTAMINEN KÄYTÄNNÖSSÄ

Tutkimukseen osallistujat saavat linkin webropol-lomakkeelle ja kirjoittavat sinne kirjoitelman. Tutkimukseen osallistuminen kestää noin 20 minuuttia.

6. TUTKIMUKSEN MAHDOLLISET HYÖDYT JA HAITAT TUTKITTAVILLE

Tutkimus tuottaa tietoa valmistuvien luokanopettajien käsityksistä ohjelmoinnin opettamisesta alakoulussa.

7. HENKILÖTIETOJEN SUOJAAMINEN

Tutkimuksessa kerättyjä tietoja ja tutkimustuloksia käsitellään luottamuksellisesti tietosuojalainsäädännön edellyttämällä tavalla. Tietojasi ei voida tunnistaa tutkimukseen liittyvistä tutkimustuloksista, selvityksistä tai julkaisuista.

Tutkimusaineistoa säilytetään Jyväskylän yliopisto tutkimusaineiston käsittelyä koskevien tietoturvakäytänteiden mukaisesti.

8. TUTKIMUSTULOKSET

Tutkimuksesta valmistuu opinnäytetyö.

9. TUTKITTAVAN OIKEUDET JA NIISTÄ POIKKEAMINEN

Tutkittavalla on oikeus peruuttaa antamansa suostumus, kun henkilötietojen käsittely perustuu suostumukseen. Jos tutkittava peruuttaa suostumuksensa, hänen tietojaan ei käytetä enää tutkimuksessa.

Tutkittavalla on oikeus tehdä valitus Tietosuojavaltuutetun toimistoon, mikäli tutkittava katsoo, että häntä koskevien henkilötietojen käsittelyssä on rikottu voimassa olevaa tietosuojalainsäädäntöä. (lue lisää: <http://www.tietosuoja.fi>).

Tutkimuksessa ei poiketa muista tietosuojalainsäädännön mukaisista tutkittavan oikeuksista.

10. REKISTERÖIDYN OIKEUKSIEN TOTEUTTAMINEN

Jos sinulla on kysyttävää rekisteröidyn oikeuksista voit olla yhteydessä tutkimuksen tekijään.