

**”Koodaaminen ei oo sitä tietokonenippieliä pelkästään”:
Luokanopettajien käsityksiä ja kokemuksia ohjelmoinnin
opettamisesta
Hanna Kurkinen**

Kasvatustieteen pro gradu -tutkielma
Syyslukukausi 2018
Opettajankoulutuslaitos
Jyväskylän yliopisto

TIIVISTELMÄ

Kurkinen, Hanna. 2018. "Koodaaminen ei oo sitä tietokonenippeliä pelkäämään": Luokanopettajien käsityksiä ja kokemuksia ohjelmoinnin opettamisesta. Kasvatustieteen pro gradu -tutkielma. Jyväskylän yliopisto. Opettajan-koulutuslaitos. 86 sivua.

Tutkielman tarkoituksena oli selvittää, millaisia käsityksiä ja kokemuksia luokanopettajilla on ohjelmoinnin opettamisesta. Tutkimuksen aineisto koostui kuuden luokanopettajan haastatteluista. Litteroidun haastatteluaineiston analyysissä käytettiin laadullista sisällönanalyysiä.

Tulokset osoittivat, että opettajien käsitysten mukaan ohjelmointi on toimintaohjeiden antamista ja siihen sisältyy ohjelmoiminen sekä tieto- ja viestintäteknologisilla laitteilla että ilman niitä. Opettajat myös kuvailivat ohjelmoinnin opetuksen harjaannuttavan monenlaisia taitoja. Opettajilla oli vaihtelevia kokemuksia ohjelmoinnin opettamisesta. Ne, jotka opettivat sitä, kokivat sen pääosin myönteisenä ja hyödyllisenä, mutta myös haasteellisenä. Haastateltavien mukaan oppilaat ovat suhtautuneet ohjelmoinnin opetukseen pääosin myönteisesti. Opettajat näkivät tärkeänä ohjelmoinnin opetuksessa käyttämiensä opetusmenetelmien ja -välineiden toiminnallisuuden ja monipuolisuuden ja kuvailivat ohjelmoinnin integrointimahdollisuuksia eri oppiaineisiin laajoiksi. Ohjelmoinnin opetus kuitenkin edellyttää koululta ja opettajilta resursseja ja valmiuksia, ja erityisesti riittämätön aika ja välineistö sekä omien valmiuksien puutteet koettiin opetuksen haasteina. Moni koki tarvitsevansa tukea ja täydennyskoulutusta ohjelmoinnin opetuksen toteuttamiseen. Opettajien vastauksissa korostui, että vaikka ohjelmoinnin opetus nähdään hyödyllisenä ja sitä toteutetaan monenlaisin opetusmenetelmin, tukea ja lisäkoulutusta tarvitaan ohjelmoinnin opetuksen helpottamiseksi.

Hakusanat: ohjelmoinnin opetus, ohjelmointi, luokanopettaja, käsitykset, kokemukset

SISÄLTÖ

1 JOHDANTO	5
2 OHJELMOINNISTA JA OHJELMOINNIN OPETUKSESTA	8
2.1 Ohjelmoinnin peruskäsitteitä	8
2.2 Ohjelmointitaitojen ja ohjelmoinnin ulottuvuudet	9
2.3 Ohjelmoinnin opetus alakoulussa	14
2.3.1 Ohjelmoinnin opetuksen lyhyt historia	14
2.3.2 Opetusmateriaaleja ja resursseja ohjelmoinnin opetukseen	16
3 OPETTAJIEN KÄSITYKSIÄ JA KOKEMUKSIA TVT:N OPETUSKÄYTÖSTÄ JA OHJELMOINNIN OPETTAMISESTA	19
4 TUTKIMUSTEHTÄVÄ	23
5 TUTKIMUKSEN TOTEUTTAMINEN	24
5.1 Metodologiset valinnat.....	24
5.2 Tutkittavat ja tutkimuksen eteneminen.....	25
5.3 Aineiston keruu.....	27
5.4 Aineiston analyysi.....	28
5.5 Luotettavuus ja eettiset ratkaisut.....	34
5.5.1 Eettiset ratkaisut	34
5.5.2 Luotettavuustekijät	36
6 TULOKSET	39
6.1 Opettajien käsitykset ohjelmoinnin opettamisesta	39
6.1.1 Ohjelmoinnin muodot ja ohjelmoinnin opetuksen asema	39
6.1.2 Ohjelmoinnin rooli monien taitojen opettajana	41
6.2 Opettajien kokemukset ohjelmoinnin opettamisesta.....	42
6.2.1 Ohjelmoinnin opetuksen edellytykset	42

6.2.2	Ohjelmoinnin opetuksen toteuttaminen.....	43
6.2.3	Opettajien yleisiä tuntemuksia ohjelmoinnin opettamisesta.....	48
6.2.4	Ohjelmointi useimpien oppilaiden innostajana.....	50
7	POHDINTA.....	52
7.1	Tulosten yhteenveto.....	52
7.1.1	Käsitykset ohjelmoinnin opetuksesta.....	52
7.1.2	Kokemukset ohjelmoinnin opetuksesta.....	53
7.2	Tulosten merkitysten arviointia.....	61
7.3	Tutkimuksen luotettavuus ja jatkotutkimusmahdollisuuksia.....	65
	LÄHTEET.....	70
	LIITTEET.....	82

1 JOHDANTO

Ohjelmoinnin opetus tuli osaksi perusopetusta uusien valtakunnallisten perusopetuksen opetussuunnitelman perusteiden (OPS 2014) käyttöönoton myötä syksyllä 2016. Samalla vastuuta ohjelmoinnin opetuksesta siirrettiin alakouluun luokanopettajien harteille, ja tämä tarkoitti monelle luokanopettajalle hyppyä kohti tuntematonta. Mediassa on uutisoitu näkyvästi koodauksen tulosta perusopetukseen (esim. Remes 2017), ja ihmetelty, pitääkö tämän päivän opettajan olla jonkinlainen koodivelho (ks. Lievonen 2014). Peruskoulussa ohjelmointia ei kuitenkaan opeteta siksi, että kaikista oppilaista tulisi koodareita, ja etenkin pelkkä teknisen ohjelmointitaidon oppiminen ei ole ohjelmoinnin opetuksen keskiössä. Tämän opinnäytetyön lähtökohtana on kiinnostus selvittää, millaisia kokemuksia opettajilla on ohjelmoinnin opetuksen sisällyttämisestä opetussuunnitelmaan ja sen haasteista opetuksessa.

Tutkimuksen keskeinen käsite on ohjelmointi. Työssä käytetään ohjelmointia (engl. *programming*) ja koodausta (engl. *coding*) samaa tarkoittavina käsitteinä viittaamaan sekä ohjelmointitoimintaan tieto- ja viestintäteknologisia (tvt) laitteita hyödyntäen että ohjelmoinnilliseen toimintaan ilman tv-laitteita. Olennainen käsite on myös ohjelmointiajattelu (engl. *computational thinking*, ks. Papert 1980; Wing 2006), joka nähdään perustavanlaatuisena ja kaikille ihmisille tarpeellisena vaiheittaisena kognitiivisena ajatteluprosessina ja ongelmanratkaisutaitona (ks. García-Peñalvo & Mendes 2018; Wing 2006). Käsitteet ohjelmointiajattelu, tietokoneajattelu ja ohjelmoinnillinen, algoritminen ja laskennallinen ajattelu nähdään tässä työssä samanarvoisina. Käsitteiden päällekkäisyys johtuu siitä, että yhtä tiettyä vakiintunutta suomenkielistä käsitettä ei ole muodostunut, vaan on useita suomennoksia, joita on käytetty eri yhteyksissä.

Nykypäivän digitalisoituvassa maailmassa ohjelmointitaidon merkitystä korostetaan koulutuksessa aiempaa enemmän. Yhteiskunnassa tarvitaan yhä monipuolisempia taitoja ja edellytetään niin kutsuttujen tulevaisuuden taitojen (engl. *21st century skills*) osaamista (Multisilta, Niemi & Lavonen 2014), kuten tieto- ja viestintätaitojen (tvt-taidot) ja ongelmanratkaisutaitojen hallintaa (Bink-

ley ym. 2012; EU 2006/962/EC). Myös ohjelmointitaito ja ohjelmointiajattelu nähdään tarpeellisina kansalaistaitoina (Balanskat & Engelhardt 2015; Wing 2006). Ohjelmointitaito voi auttaa ymmärtämään digitalisoituvaa yhteiskuntaa (Balanskat & Engelhardt 2015), ja siihen sisältyviä ongelmanratkaisutaitoja ja ohjelmointiajattelua tarvitaan monissa arjen tilanteissa (Wing 2006) sekä työelämässä. Ohjelmointitaidon kehittyminen on myös yhteydessä useiden taitojen, esimerkiksi loogisen ajattelun ja ongelmanratkaisutaitojen, kehittymiseen (Balanskat & Engelhardt 2015). Lisäksi it-alan ammattilaisista on pulaa (Karkimo 2018; Räisänen 2017; ks. myös Balanskat & Engelhardt 2015), mutta innostus it-alan opiskeluun on yleisesti vähentynyt, ja erityisesti naisten määrä alalla on vähäinen (Brown ym. 2013). Tätä saattavat selittää tietämyksen ja kokemuksen puute sekä ennakkoluulot alasta (ks. Armoni 2011). Ohjelmoinnin opetuksen tuleminen perusopetuksen opetussuunnitelmaan ensimmäiseltä luokalta lähtien voi saada oppilaat, myös tytöt, kiinnostumaan koodauksesta ja it-alasta.

Suomessa, kuten muuallakin maailmassa, on pyritty vastaamaan ohjelmointiosaamisen tarpeeseen sisällyttämällä ohjelmoinnin opetusta perusopetukseen. Perusopetuksen opetussuunnitelman (OPS 2014) mukaan ohjelmoinnin opetuksen tavoitteena on muun muassa mahdollistaa oppilaille kokemuksia ohjelmoinnista ja innostaa oppilaita ohjelmoimaan, toteuttaa ohjelmointia visuaalisessa ohjelmointiympäristössä ja robotiikan avulla sekä harjaannuttaa ohjelmointiajattelua. Ohjelmoinnin ajatellaan opettavan ensisijaisesti ajattelun taitoja teknisen koodausosaamisen sijaan. Ohjelmointi mainitaan opetussuunnitelmassa (OPS 2014) yhteensä vain kymmenellä sivulla, eikä se korostu vahvasti. Sen sijaan esimerkiksi toiseen uuteen käsitteeseen, kestävään kehitykseen synonyymeineen, viitataan opetussuunnitelmassa kaiken kaikkiaan yli 80:llä sivulla. Lisäksi, vaikka ohjelmoinnin opetuksen perusopetukseen tulemiseen liittyvästä median suuresta huomiosta (esim. Konttinen 2015; Lievonen 2014; Vatanen 2016) voisi päätellä toisin, ei ohjelmoinnin opetus oikeastaan ole uusi ilmiö koulumaailmassa (Paananen 2005; Saarikoski 2006).

Monissa maissa ohjelmoinnin opetus peruskoulussa on aloitettu jo ennen Suomea (ks. Balanskat & Engelhardt 2015; García-Peñalvo ym. 2016). Balanskat ja Engelhardtin (2015) raportti osoittaa, että Euroopassa vuoteen 2015 men-

nessä jo 15 maata, muun muassa Viro ja Iso-Britannia, oli sisällyttänyt ohjelmoinnin eri kouluasteiden opetussuunnitelmiin, osa jo alakoulusta alkaen. Joissain maissa ohjelmointi on oma oppiaineensa (ks. Berry 2013), kun taas Suomessa ohjelmointi sisältyy opetussuunnitelmassa laaja-alaiseen osaamiseen sekä matematiikan ja käsityön opetukseen (OPS 2014). Yhdysvalloissa puolestaan ohjelmointi ei kuulu valtakunnallisiin opetussuunnitelmiin, mutta erilaiset järjestöt kuten Code.org ovat rakentaneet omia opetussuunnitelmiaan, joita koulut voivat halutessaan hyödyntää (ks. Code.org 2018; Singer 2017).

Ohjelmoinnin opetusta alakoulussa on tärkeää tutkia, koska aihe on ajan-kohtainen ja tuore koulumaailmassa. Vaikka ohjelmoinnin opetuksesta ja yleisesti tieto- ja viestintäteknologian käytöstä opetuksessa on tehty kansainvälisesti paljon tutkimusta, alakoulukontekstissa tapahtuva opettajien käsityksiin ja kokemuksiin keskittyvä tutkimus on vähäistä. Tutkimus on pitkälti rajoittunut oppilaiden näkökulmaan ja ylempien kouluasteiden kontekstiin. Varsinkin kotimaista tutkimusta aiheesta on tehty vain pienessä määrin, ja kansainvälisessä tutkimuksessa painottuvat yksittäiset koulukokeilut. Ohjelmointia ja ohjelmoinnin opetusta on tarkasteltu viimeaikaisessa kotimaisessa tutkimuksessa opinnäytetöissä ja väitöskirjoissa oppilaiden (ks. Hämeenaho 2016; Lakanen 2016) ja opettajien näkökulmasta (ks. Karvonen & Laukka 2016; Makkonen & Pyykönen 2018), ja aiheesta on tuotettu myös selvityksiä (ks. Kaarakainen ym. 2017). Kansainvälistä tutkimusta on tehty muun muassa erilaisista ohjelmointikokeiluista varhaiskasvatuksessa (ks. Bers, Flannery, Kazakoff & Sullivan 2014; Fessakis, Gouli & Mavroudi 2013), alakoulussa (ks. Chen ym. 2017; Zaharija, Mladenović & Boljat 2013) sekä ylempillä kouluasteilla (ks. Denner, Werner & Ortiz 2012; Kalelioğlu 2015). Lisäksi tutkimusta on tehty eri ohjelmointiympäristöjen käytöstä ohjelmoinnin opetuksessa (esim. Graczyńska 2010; Wilson & Moffat 2010).

Tämän tutkimuksen tarkoituksena oli selvittää, miten luokanopettajat käsittävät ja kokevat ohjelmoinnin opetuksen alakoulussa. On tärkeää tehdä katsoa siihen, millaista ohjelmoinnin opetus on alakoulun luokanopettajan näkökulmasta ja miten ohjelmoinnin opetusta toteutetaan alakoulussa.

2 OHJELMOINNISTA JA OHJELMOINNIN OPE- TUKSESTA

2.1 Ohjelmoinnin peruskäsitteitä

Ohjelmointi voidaan Hyvösen, Lappalaisen ja Lakasen (2013) mukaan määritellä vaiheittaisten toimintaohjeiden laatimiseksi ja antamiseksi. Usein toimintaohjeita annetaan tietokoneelle tai vastaavalle teknologiselle laitteelle jonkin toimenpiteen suorittamista varten (Balanskat & Engelhardt 2015). Tietokone voidaan Lehtosen ja Kyllösen (2010) mukaan määritellä laitteeksi, johon voidaan tallentaa tietoa ja ohjelmoida toimintoja, ja joka pystyy käsittelemään näitä ja olemaan vuorovaikutuksessa käyttäjänsä kanssa. Ohjelmointia tarvitaan esimerkiksi internetissä julkaistavien verkkosivustojen luomiseen ja muokkaamiseen (ks. Liukas & Mykkänen 2014) sekä digitaalisten pelien tekemiseen (Harvainen, Meriläinen & Tossavainen 2013).

Ohjelmointia voidaan kuitenkin ymmärtää laajemmin. Ohjelmointi yhdistetään usein vain tietokoneen äärellä tapahtuvaan mekaaniseen ohjelmointikielellä kirjoittamiseen, vaikka ohjelmoinnin tapaista toimintaa esiintyy monissa päivittäisissä arkipäivän tilanteissa (ks. Hyvönen ym. 2013). Esimerkiksi mikroaaltouunin käyttäminen, kakun leipominen reseptin avulla tai jopa ajo-ohjeiden laatiminen ovat laajasti nähden ohjelmointia, sillä ne muodostuvat toimintaohjesarjoista. Ohjelmointi on myös ongelmien ratkaisemista (Fessakis, Gouli & Mavroudi 2013; Kaijanaho 2010). Ohjelmoinnilla tai koodauksella voidaan myös viitata aktiviteetteihin, joiden avulla lapset oppivat käyttämään tiettyjä ohjelmia ja ohjelmoimaan digitaalisilla laitteilla (Balanskat & Engelhardt 2015).

Tietokoneohjelmointi on kommunikointia tietokoneelle sellaisella kielellä, mitä kone ja ihminen pystyvät molemmat tulkitsemaan (Papert 1980). Ihminen ohjeistaa tietokoneen toteuttamaan erilaisia tehtäviä kirjoittamalla ohjelmia eli käskyluetteloita. Ohjelma muodostuu useista yksittäisistä toimenpiteistä, jotka muodostavat toimenpidesarjoja (Saarenpää & Takkinen 1998), sisältäen ohjeet,

miten toimia eri tilanteissa (Saarenpää & Takkinen 1998). Tällaista toimintaohjeluetteloja, joka on sarja käskyjä, kutsutaan algoritmiksi (Hyvönen ym. 2013; Paananen 2005). Käskyt, eli ohjelmointikielellä kirjoitetut koodit, saavat tietokoneen toimimaan (Balanskat & Engelhardt 2015).

Ohjelmat eli käskyluettelot on kirjoitettu ohjelmointikielillä (Saarenpää & Takkinen 1998). Ne toimivat välineinä tai käyttöliittyminä ihmisen ja tietokoneen välillä (Harsu 2005). Toimintaohjeet kirjoitetaan sekä ihmisen että koneen ymmärtämässä muodossa (Louden 2003; Paananen 2005), jolloin ihminen pystyy ohjelmointikielen avulla välittämään tietokoneelle, mitä sen tulee tehdä (Harsu 2005; Kaijanaho 2010). Ensimmäiset ohjelmointikielet olivat Kaijanahon (2010) mukaan konekieliä, ja niillä kirjoitetut ohjelmat bittijonoja. Konekielet edustavat ohjelmointikielten ensimmäistä sukupolvea, ja symboliset konekielet toista sukupolvea. Varhaisimmat kolmannen sukupolven korkean tason ohjelmointikielet ovat syntyneet 1950-luvulla, ja ensimmäiset modernit ohjelmointikielet, kuten Pascal ja C, syntyivät puolestaan 1970-luvulla. Nykyään paljon käytettyjä tekstipohjaisia ohjelmointikieliä ovat muun muassa C, C++, C#, Pascal, Java ja Python (ks. Lappalainen & Viitanen 2012).

Nykyään on lisäksi paljon helppokäyttöisiä, ohjelmoinnin opiskeluun suunnattuja ohjelmointikieliä. Ne ovat usein graafisia eli visuaalisia ohjelmointikieliä ja -ympäristöjä, jotka koostuvat valmiista koodin palasista, koodilohkoista. Tällaisten ohjelmointiympäristöjen etuna on se, että ne mahdollistavat koodauksen opettelun ilman varsinaisen ohjelmointikoodin kirjoittamista itse (Lye & Koh 2014). Tunnetuimpia erityisesti lapsille ja yleisesti ohjelmoinnin alkeiden opetukseen suunnitelluista graafisista ohjelmointikielistä ja -ympäristöistä on *Scratch* (Burke 2012; Flannery ym. 2013). Muita helppokäyttöisiä ohjelmointikieliä ovat *Alice* (Graczyńska 2010; Werner, Campe & Denner 2012) sekä *Logo*, *ToonTalk* ja *Stagecast Creator* (Fessakis ym. 2013).

2.2 Ohjelmointitaitojen ja ohjelmoinnin ulottuvuudet

Ohjelmointia on tarkasteltu monista eri näkökulmista, ja tässä tutkimuksessa se nähdään seuraavien ulottuvuuksien kautta: tekninen ohjelmointitaito, ohjel-

mointi lukutaitona, ohjelmointi ajattelun taitona sekä ohjelmointi itseilmaisun välineenä. Ohjelmointitoiminnan käsitetään edellyttävän, ja toisaalta myös opettavan, niin teknistä ohjelmointitaitoa, ohjelmointilukutaitoa kuin ohjelmointiajattelun taitoa, sekä lisäksi mahdollistavan itseilmaisua ja luovuutta.

Neljään ulottuvuuteen jaottelun pohjana on hyödynnetty aiemmissa tutkimuksissa tehtyjä jaotteluja ja käytettyjä käsitteitä (ks. Kaila 2018; Lye & Koh 2014; diSessa 2000; Vee 2013). Kaila (2018) esimerkiksi jakaa ohjelmointitaidot kahtia tekniseen osaamiseen ja algoritmisiin ongelmanratkaisutaitoihin. Lye ja Koh (2014) taas jäsentävät ohjelmointia sen harjaannuttamien taitojen näkökulmasta kolmijakoisesti käsitteiden, käytänteiden ja näkökulmien kautta. He nostavat esiin ohjelmoinnin keskeisten käsitteiden ja ajattelukäytänteiden ymmärtämisen kehittymisen, joka lisää mahdollisuuksia itseilmaisuuksiin ja luovuuteen suhteessa muihin ja teknologiseen ympäristöön. Myös käsitettä ohjelmointilukutaito (diSessa 2000; Vee 2013) on käytetty näkökulmana ohjelmointiosaamiseen. Näiden näkökulmien avulla pystytään kuvaamaan, miten ohjelmointi edellyttää enemmän kuin vain yhtä yksittäistä taitoa; osaamista vaaditaan joutuessa eri taitoja (Vahldick, Mendes ja Marcelino 2014).

Tekninen ohjelmointitaito. Perinteisen näkökulman mukaan ohjelmointi tarkoittaa ja kehittää ennen kaikkea teknistä ohjelmointitaitoa. Ohjelmointia tehdään usein tietokoneella ja eri ohjelmointikielillä koodia kirjoittamalla (Harsu 2005; Kaijanaho 2010), jolloin tavoitteena on ohjeistaa tietokone toteuttamaan tietty tehtävä (Balanskat & Engelhardt 2015). Ohjelmoidessa harjaannutetaan kykyä ymmärtää ohjelmointikäsitteitä ja ohjelmoinnin logiikkaa (Sáez-López, Román-González & Vázquez-Cano 2016). Ohjelmointia opiskeltaessa tutustutaan keskeisiin ohjelmointikäsitteisiin ja periaatteisiin eli ohjelmoinnin teknisiin yksityiskohtiin kuten muuttujiin ja toistorakenteisiin (Brennan & Resnick 2012; Lye & Koh 2014) sekä peräkkäisyyteen ja ehdollisuuteen (Sola 2002). Ohjelmoinnin avulla voidaan oppia myös matemaattisia käsitteitä ja taitoja, kuten laskutaitoa, lukujen vertailua ja vastaavuuden ymmärtämistä (Fessakis ym. 2013) sekä etäisyyden ja ajan mittaamista (ks. Flannery ym. 2013). Ohjelmointikäsitteitä ja -periaatteita voi oppia ymmärtämään myös ilman formaalia opetus-

ta, itse ohjelmointiympäristöjä käyttämällä ja kokeilemalla (Maloney, Peppler, Kafai, Resnick & Rusk 2008).

Ohjelmointilukutaito. Digitaalisissa ympäristöissä tarvitaan uudenlaisia lukutaitoja. Goughin ja Tunmerin (1986) mukaan peruslukutaito tarkoittaa teknisen lukemisen ja tekstin ymmärtämisen taitoja. Tekniseen lukutaitoon sisältyy sanojen tunnistamisen taito sekä kirjain-äänne-vastaavuuden hallinta, ja tekstin ymmärtäminen pitää sisällään luetun ymmärtämisen ja tekstin tulkinnan taidot. García-Peñalvo (2018) esittää, että digitaalisessa maailmassa tulee oppia kielellisen lukutaidon lisäksi myös digitaalista lukutaitoa, joka pitää sisällään joukon taitoja, joita tarvitaan lukemiseen ja kirjoittamiseen digitaalisissa ympäristöissä. Voidaan puhua myös digitaalisesta sujuvuudesta tai osaamisesta (ks. Balanskat & Engelhardt 2015; Hockly 2012; Resnick ym. 2009). Resnickin ja kumppaneiden (2009) mukaan digitaalinen lukutaito edellyttää taitoa suunnitella, luoda, keksiä ja yhdistellä. Beshaw (2011) sisällyttää digitaalisiin lukutaitoihin kulttuurisen, kognitiivisen, konstruktivisen, viestinnällisen, yhteisöllisen, luovan ja kriittisen osaamisen, kuten myös luottamuksen omiin kykyihin teknologian käyttäjänä. Bawden (2008) huomauttaa, että digitaaliset lukutaidot muuttuvat ajan kuluessa ja teknologian kehittyessä, sillä teknologia muuttaa kommunikointitapoja.

Digitaalisen lukutaidon osa-alueiksi on mainittu sekä ohjelmointilukutaito (engl. *computational literacy*) että koodinlukutaito (engl. *code literacy*) (ks. diSessa 2000; Vee 2013). Niihin sisältyy koodauksen perusajatuksen ymmärtäminen (Hockly 2012) sekä kyky kirjoittaa ja lukea koodia (Vee 2013). Ohjelmointiosaaminen voidaankin nähdä lukutaitona, jonka keskiössä ovat ohjelmointikielillä lukemisen ja kirjoittamisen osaaminen, ja jonka avulla pystyy ilmaisemaan itseään ja ajatuksiaan kirjoittamalla ja lukiessaan tulkitsemaan muiden tekstejä (Vee 2013). Vee (2013) esittää, että tietokoneohjelmoinnissa ihminen pystyy tekstilukutaidon tavoin ilmaisemaan ja tulkitsemaan ajatuksia koodin avulla. Hänen mukaansa ohjelmointilukutaito sisältää tekstilukutaidon ominaisuuksia, kuten tekstin kirjoittamisen ja lukemisen elementit, mutta tekstilukutaidosta poiketen siinä rakennetaan rakenteita ennalta määritellyistä palasista. Vee näkee ohjelmointilukutaidon laajana taitokokonaisuutena, jonka avulla osa-

taan palastella kokonaisuuksia pienempiin vaiheisiin ja ilmaista nämä vaiheet ohjelmointikielellä. Ohjelmointisovellusten, kuten Scratchin, on havaittu harjaannuttavan koodinkirjoittamistaitoa ja koodinlukutaitoa, kun koodia rakennetaan lohkoista ja rakennettua koodia luetaan (Flannery ym. 2013).

Ohjelmointi ajattelun taitona. Ohjelmointikielen lukeminen ja kirjoittaminen ovat varsinaista tietokoneohjelmointia (Balanskat & Engelhardt 2015), mutta ohjelmointiajattelu on sen taustalla oleva kognitiivinen ongelmanratkaisuprosessi, joka mahdollistaa sen (García-Peñalvo & Mendes 2018). Ohjelmointitaito voidaankin nähdä myös ajattelun taitona. Wingin (2006) mukaan ohjelmointiajattelu on jokaisen tarvitsemaa perusosaamista. Ohjelmointiajattelu kuvaa ihmisen tai koneen toimintaa, kun tämä määrittelee ongelmaa ja kehittää ratkaisua. Se on ratkaisun selvittämiseen tähtäävää järkeilyä, joka pitää sisällään ongelmanratkaisua, suunnittelua ja käyttäytymisen ymmärtämistä. Ohjelmointiajattelu yhdistetään usein koodaukseen ja tietokoneohjelmointiin, mutta se tarkoittaa laajempaa kokonaisuutta. Ohjelmointiajattelua hyödynnetään myös monenlaisissa arkisissa toiminnoissa, joita ei yleensä varsinaisesti ajatella koodaustehtävinä (Wing 2006). Ohjelmointiajattelu onkin olennainen taito oppia, koska sitä tarvitaan jokaisen arjessa (ks. Lye & Koh 2014).

Ohjelmoinnin opetus tukee ohjelmointiajattelun kehittymistä (Lye & Koh 2014; Resnick ym. 2009; Sáez-López ym. 2016). Ohjelmoinnin opetus onkin ennen kaikkea ajattelun taitojen opettamista (Liukas & Mykkänen 2014). Resnickin ja kumppaneiden (2009) mukaan ohjelmoiminen harjaannuttaa ohjelmointiajattelua, ongelmanratkaisuntaitoja ja suunnittelustrategioita, jotka kantavat arjen muillekin alueille kuin vain tietotekniikkaan. Barrin ja Stephensonin (2011) mukaan ohjelmointiajatteluun sisältyy muun muassa analysoiminen, ongelmien purkaminen pienempiin osiin, luokittelu ja soveltaminen. Ohjelmointiajattelu on osaamista, jossa pystyy löytämään rakentavia ratkaisuja ja soveltamaan niitä ongelmiin (Jacobsen 2015). García-Peñalvo (2016b) korostaa, että pelkkää koodaustaitoa oleellisempaa on opettaa ongelmien ratkaisemista ohjelmointiajattelun avulla, kuten esimerkiksi asioiden palastelua pienempiin osiin ja ohjelmoinnillisen ajattelun soveltamista.

Ohjelmointi harjoittaa myös muita kognitiivisia taitoja. Ohjelmointia opiskeltaessa harjaannutetaan esimerkiksi loogista päättelyä (Balanskat & Engelhardt 2015), avaruudellista hahmotuskykyä (Flannery ym. 2013; Harlow ym. 2015) sekä kykyä visualisoida ja käsitteellistää (Liukas & Mykkänen 2014). Brennanin ja Resnickin (2012) mukaan ohjelmoinnin avulla harjaannutaan myös vaihteellisessa toiminnassa, testaamisessa, virheiden korjaamisessa, ideoiden uudelleenkäytössä ja yhdistelemisessä. Oppilas oppii ohjelmoidessaan myös arvioimaan, ennustamaan, yhdistelemään ja erottelemaan (Flannery ym. 2013). Brennanin ja Resnickin (2012) mukaan ohjelmointiajatteluun liittyy myös kyseenalaistaminen, joka viittaa siihen, että teknologian käyttäjä ei ota teknologiaa itsestäänselvyytenä vaan esittää kysymyksiä ja jäsentää sen kautta ohjelmointimaailmaa.

Ohjelmointiajattelun taitojen kehitystä voidaan tukea esimerkiksi käyttämällä asianmukaisia ohjelmointiympäristöä. Sáez-López ja kumppanit (2016) ovat todenneet visuaalisten ohjelmointikielten, kuten Scratchin, käytön tukevan oppilaiden ohjelmointitaitojen kehittymistä ja tietokonekäytäntöjen ymmärtämistä. Visuaalisten ohjelmointiympäristöjen on havaittu kehittävän myös ongelmanratkaisutaitoja (Esteves, Fonseca, Morgado & Martins 2011) ja mahdollistavan välittömän palautteen saamisen, minkä on esitetty tukevan ohjelmointiajattelun kehittymistä (Esteves ym. 2011; Fessakis ym. 2013). Oppilas saa ohjelmoidessaan tietokoneelta välitöntä palautetta saadessaan ohjelman toimimaan, ja esimerkiksi visuaalisessa ohjelmointiympäristössä näkee heti, toimiiko ohjelma halutulla tavalla (Esteves ym. 2011). Lye ja Koh (2014) esittävät, että ohjelmointiajattelua voidaan tukea myös luomalla ongelmanratkaisukeskeinen oppimisympäristö, jossa ratkotaan aitoja ongelmia, prosessoidaan aitoa informaatiota ja reflektoidaan ja jossa tuetaan oppilaita sopivasti. Ohjelmointiajattelua voidaan kehittää myös hyödyntämällä opetuksessa pelillisyyttä (Kaila 2018) ja opetuspelejä (Kazimoglu, Kiernan, Bacon & Mackinnon 2012).

Ohjelmointi itseilmaisun välineenä. Erilaiset ohjelmointiympäristöt mahdollistavat luovuuden (Balanskat & Engelhardt 2015; Smith, Sutcliffe & Sandvik 2014) ja tarjoavat mahdollisuuksia itseilmaisuuksiin (Kahn, Sendova, Sacristán & Noss 2011) sekä esimerkiksi luovaan tarinankerrontaan (Burke 2012).

Resnickin ja kumppaneiden (2009) mukaan koodaustaito laajentaa luomisen ja itsensä ilmaisun mahdollisuuksia tietokoneella. Brennan ja Resnick (2012) esittävät, että ohjelmointi voidaan nähdä pelkän ohjelmien käyttämisen sijaan myös suunnittelun, luomisen ja itseilmaisun välineenä. Tietotekniikka luo Resnickin (2007) mukaan lisää mahdollisuuksia luovan ajattelun oppimiseen. Ohjelmointiympäristöjen avulla voi esimerkiksi luoda animaatioita, pelejä tai virtuaalimaailmoja (Kelleher, Pausch & Kiesler 2007). Scratch Jr.:n kaltaiset sovellukset harjaannuttavat oppijaa muun muassa tarinankerronnassa, luovassa ajattelussa ja itseilmaisussa (Flannery ym. 2013). Scratchissä oppilas voi personoida projekteja esimerkiksi lisäämällä omia kuvia (Resnick ym. 2009). Resnick kollegoineen (2009) korostaa, että luovuuden ilmaisua ja luovan ajattelun oppimista on tärkeää mahdollistaa kouluopetuksessa antamalla oppilaille tilaisuuksia toimia keksimällä, luomalla, leikkimällä, jakamalla ja refleктоimalla.

Ohjelmoinnin opetuksen on havaittu tukevan oppilaiden sosiaalisia ja yhteistyötaitoja sekä omaa ilmaisua yhdessä muiden kanssa (Fessakis ym. 2013). Brennan ja Resnick (2012) ovat korostaneet ohjelmoinnin roolia yhteisöllisenä jakamisen välineenä, sillä sen avulla voi luoda sekä toisille että toisten kanssa. Luovuuteen ja oppimiseen liittyykin Brennanin ja Resnickin mukaan vahvasti sosiaalisuus: muita voidaan viihdyttää, osallistaa tai opettaa tai heille voidaan jakaa ideoita.

2.3 Ohjelmoinnin opetus alakoulussa

2.3.1 Ohjelmoinnin opetuksen lyhyt historia

Teknologiasta ja koodauksesta puhutaan usein hyvin futuristiseen sävyyn, vaikka niiden hyödyntäminen kouluopetuksessa ei ole uusi asia. Ohjelmoinnillisia piirteitä voidaan nähdä olevan luonnostaan monenlaisissa koulun rutineissa ja työskentelytavoissa. Koska ohjelmointi on yksinkertaisimmillaan vaiheittaisten toimintaohjeiden laatimista ja antamista (ks. Hyvönen ym. 2013), ohjelmoinnin omaisia piirteitä on nähty jopa esimerkiksi tilanteissa, joissa opettaja antaa oppitunnilla työskentelyohjeita oppilaille.

Tieto- ja viestintäteknologian opetuskäyttö ja ohjelmoinnin opetus eivät myöskään ole uusia ilmiöitä koulumaailmassa. Paanasen (2005) mukaan ensimmäinen nykyaikainen tietokone kehitettiin 1950-luvulla, ja samalla vuosikymmenellä hankittiin Suomeenkin ensimmäiset tietokoneet. Tietokoneet levisivät 1980-luvulla laajamittaisesti sekä koteihin (Järvinen 1999) että oppilaitoksiin (Paananen 2005). Huomioitavaa on, että tietoteknisiä laitteita, kuten tietokoneita, ei ole alun perin luotu opetusteknologiaksi (Papert 1980), vaan tietokoneita alettiin luonnehtia opetusteknologiaksi vasta sen jälkeen kun ne tuotiin kouluihin (Pirhonen 2010). Opetussovelluksina markkinoiduista teknologisista sovelluksista monet onkin Pirhosen (2010) mukaan luotu ilman suoraa kytkentää pedagogiseen osaamiseen.

Tietokoneet ja ohjelmointi tekivät tuloaan oppilaitoksiin jo 1980-luvun alussa (Paananen 2005). Varhaisen atk-opetuksen edelläkävijöitä olivat koulu-kokeilut ja atk-kerhot 1960- ja 1970-luvuilla, kunnes 1980-luvun lopulla atk-opetus tuli pakolliseksi koulujen opetussuunnitelmiin (Saarikoski 2006). Saarikosken (2006) mukaan tietotekniikkaa opetettiin aluksi vain lukiossa, mutta siitä tuli peruskoulun yläkoulussa opetettava valinnainen oppiaine vuonna 1984. Virallisesti tietotekniikan opetus valinnaisena aineena (atk) yläkoulussa alkoi vuosien 1987-1988 aikana. Atk-opetus sisälsi tietokoneen hallinnan perusteita, tietoverkkojen käyttöä, ohjelmointia sekä tietokoneavusteista opetusta eli tietokoneohjelmien hyödyntämistä opetuksessa. Opetuksen tavoitteena oli rohkaista oppilaita loogiseen ajatteluun omia ohjelmia tekemällä sekä atk-opetuksen integrointi eli yhdistäminen osaksi muitakin oppiaineita. Saarikosken (2006) mukaan atk-opetuksen ongelmakohdiksi muodostuivat tuolloin tietokoneiden puute, ohjelmien ja opetukseen käytetyn ajan vähyys sekä riittämätön opettajakoulutus ja ohjelmoinnin asema opetuksessa vahvasti kytköksissä matematiikan opetukseen. Atk-opetus sai kritiikkiä sisällöllisten ongelmien lisäksi myös siitä, että alakoulu jätettiin kokonaan opetuksen ulkopuolelle. Alakoulussa atk-opetus lähti liikkeelle pienimuotoisesti kokeilumuotoisena kerhotoimintana vuonna 1984.

2.3.2 Opetusmateriaaleja ja resursseja ohjelmoinnin opetukseen

Ohjelmoinnin opetuksessa voidaan hyödyntää monenlaisia resursseja. García-Peñalvon ja Mendesin (2018) mielestä ohjelmoinnin opetus tarjoaa monipuolisia mahdollisuuksia esimerkiksi koodaukseen, robotteihin, mobiililaitteiden käyttämiseen ja oppimisen pelillistämiseen. Ohjelmoinnin opetuksessa voidaan Kallelioglundin (2015) mukaan voida opettaa oppilaille monin eri keinoin tietotekniikkaa, ohjelmointia ja robottien ohjelmointia sekä tukea oppilaiden kiinnostusta ohjelmointia kohtaan ja ohjelmointiajattelun, kriittisen ja luovan ajattelun sekä ongelmanratkaisutaitojen kehitystä. Vaikka ohjelmointia pidetään yleisesti vaikeana, ja sen on ajateltu olevan erityisesti lapsille haastava taito oppia sen abstraktiuden vuoksi, ohjelmointia voi oppia jo varhaisella iällä, kunhan ohjelmointivälineet ovat lapselle iän ja kehitystason puolesta sopivia (Sola 2002).

Lapsille ja alakouluikäisille suunnattua ohjelmointiin liittyvää kirjallisuutta, jossa opastetaan ohjelmointiajattelun alkeisiin, on ilmestynyt jo jonkin verran. Kotimaisia teoksia ovat esimerkiksi Hello Ruby – Maailman paras koodisatukirja (Liukas 2015) ja Dibitassut – Matka ohjelmoinnin maailmaan (Hiltunen, Hiltunen & Hiltunen 2015) jatko-osineen. Koululaisille suunnatuissa ohjelmointioppaissa ja -kirjoissa opetetaan konkreettisesti ohjelmoinnin perusteita eri ohjelmointikielillä vaihtelevissa ohjelmointiympäristöissä. Niissä käydään läpi muun muassa ehtolauseita, toistorakenteita, funktioita ja muuttujia (esim. Vorderman ym. 2017; Woodcock 2015). Kirjoissa useimmin esiintyvä ohjelmointikieli on Scratch (esim. Christian 2016; Vorderman ym. 2017; Wainwright 2017; Woodcock 2016). Opettajille suunnattua tietoutta ohjelmoinnin opettamisesta löytyy useilta verkkosivustoilta, joille on koottu vinkkejä, tuntisuunnitelmia ja oppaita ohjelmoinnin opetukseen (esim. CS Unplugged; Koodi2016; Koodiaapinen; Koodikirja; ScratchEd), sekä kirjallisuudesta, josta esimerkkinä Reseptit OPSin käyttöön (Luostarinen & Peltomaa 2016).

Ohjelmointia voi opettaa sekä ilman tvt-laitteita että niitä hyödyntäen. Ohjelmoinnin opetuksessa ilman tieto- ja viestintäteknologiaa voidaan hyödyntää erilaisia leikkejä, kuten esimerkiksi robotti-leikkiä (esim. Harold the Robot) tai ei-teknologisia tuotteita, kuten loogisia paloja tai lautapelejä (esim. Robogem).

Tvt:n osalta voidaan käyttää ohjelmoitavia robotteja (esim. Bee-Bot, Blue-bot, Jimu, Lego Mindstorms), mobiililaitteille tai tietokoneille saatavissa olevia sovelluksia (esim. Bomberbot, Lightbot, Scratch Jr., Kodu) tai verkkosivustojen ohjelmia (esim. Code Academy, Code.org, Scratch). Joillain sovelluksilla voi pelata ohjelmoinnillista peliä (esim. Lightbot), mutta osaa voi käyttää myöskin omien pelien luomiseen (esim. Gamestar Mechanic, Scratch).

Ohjelmointia opiskeltaessa tvt-laitteita käyttäen ohjelmointi tapahtuu usein symboleilla tai koodilohkoilla yksinkertaisissa visuaalisissa ohjelmointiympäristöissä, ja tästä edetään tekstimuotoisen koodin kirjoittamiseen oppilaan koodaustaitojen kehittyessä (Vahldick ym. 2014). Visuaalisten ohjelmointiympäristöjen graafisuuden ansiosta ohjelmointi ei edellytä lapselta liian haastavien ohjelmoinnin taustalla olevien abstraktien prosessien ymmärtämistä, eikä symboleilla ohjelmoinnissa edellytetä tekstimuotoisen koodin luku- ja kirjoitustaitoa (Fessakis ym. 2013; Vahldick ym. 2014). Vaikka visuaaliset ohjelmointiympäristöt on havaittu ohjelmoinnin opetuksessa hyviksi ja useiden eri taitojen kehitystä tukeviksi (Esteves ym. 2011; Fessakis ym. 2013), myös tekstipohjaisia ohjelmointiympäristöjä pidetään ohjelmoinnin opetukseen sopivina (García-Peñalvo ym. 2016; Vahldick ym. 2014).

Sovelluksia ja materiaaleja ohjelmoinnin opetukseen löytyy runsaasti, ja osa on parempia kuin toiset. Scratch on yksi yleisimpiä ohjelmoinnin opetuksen alkutaipaleella käytetyistä ohjelmointiympäristöistä (ks. Sáez-López ym. 2016; Wilson & Moffat 2010). Scratchin lisäksi myös Code.org-verkkosivuston tarjoamat resurssit ovat suosittuja ja yleisesti käytettyjä ohjelmoinnin opetuksessa (ks. Israel, Pearson, Tapia, Wherfel & Reese 2015). Tällaiset ympäristöt tekevät ohjelmoinnin oppimisesta lapsille helppoa, koska ohjelmointikieli on yksinkertaistettua (Bruckman 1997), ohjelmia luodaan graafisia lohkoja järjestelemällä ja toiminnot tapahtuvat animoidussa ympäristössä (Maloney ym. 2004). Huomioitavaa on kuitenkin, että ohjelmointisovellusten ja -ympäristöjen suuri puute yleisesti on se, että ne eivät mukaudu käyttäjän kehitykseen. Esimerkiksi lohkoilla ohjelmointia pidetään riittämättömänä käyttäjän taitojen kehittyessä pidemmälle (Vahldick ym. 2014).

Tutkimusta ohjelmoinnin opetukseen käytettävistä sovelluksista on jonkin verran (esim. Graczyńska 2010; Sáez-López ym. 2016; Werner ym. 2012; Wilson & Moffat 2010), mutta tutkimukseen perustuvia koonteja tai suosituksia resursseista tai opettajien materiaaleista ei ole juuri tarjolla. Poikkeuksena García-Peñalvo kumppaneineen (2016) on koonnut TACCLE 3 -projektin tutkimukseen pohjaavan listan parhaista sovelluksista ja roboteista ohjelmoinnin opetukseen. Heidän mukaansa parhaita sovelluksia ohjelmoinnin opetukseen ovat muun muassa Code Combat, Puzzlets, Scratch Jr. Ja SpaceChem Mobile, ja robotiikkasarjoista parhaimpia ovat muun muassa Bee-Bot, Lego Mindstorms, Ozobots ja Robbo. Kriteereinä näille arvioinneille olivat muun muassa resurssien laatu, pedagogiset mahdollisuudet sekä TACCLE-luokittelun eli algoritmien, logiikan käytön, asioiden kontrolloimisen, luomisen ja debuggauksen huomioonottaminen (García-Peñalvo 2016a). Vahldick ja kumppanit (2014) ovat puolestaan koonneet kirjallisuuden pohjalta listauksen 40 ohjelmoinnin opetukseen soveltuvasta pelistä. He rajasivat mukaan vain sellaiset pelit, jotka selvästi liittyvät ohjelmoinnin opiskeluun, kuten esimerkiksi Cube-Game, Daisy the Dino ja Wu's Castle. He eivät kuitenkaan esitä suosituksia parhaimmista peleistä ja kritisoivat listan pelejä siitä, että kaikki eivät ole todellisuudessa saatavilla kokeilu varten. Paikallinen saatavuus voikin olla ongelmana, kun kyseessä on kansainvälinen koonti sovelluksista.

Koska useampia vastaavia tai laajempia koonteja ohjelmointiin erityisesti alakoulukontekstissa soveltuvista sovelluksista ei ole tehty, olen lisäksi itse koonnut yhteen esimerkkejä käyttökelpoisista ja saatavilla olevista sovelluksista ja verkkosivustoista kotimaiseen alakoulun ohjelmoinnin opetukseen (ks. liite 3). Tekemässäni koonnissa sovellukset ja verkkosivustot on jaoteltu kolmeen tasoon. Luokitelluista ohjelmista osassa ohjelmointi tapahtuu symboleita siirtelämällä ja osassa ohjelmoidaan valmiita koodilohkoja liikuttelemalla tai kirjoittamalla tekstimuotoista koodia. Lisää vastaavia resursseja löytyy muista lähteistä (esim. Common Sense Education 2018; Liukas & Mykkänen 2014).

3 OPETTAJIEN KÄSITYKSIÄ JA KOKEMUKSIA TVT:N OPETUSKÄYTÖSTÄ JA OHJELMOINNIN OPETTAMISESTA

Opettajien käsityksiin ja kokemuksiin ohjelmoinnin opettamisesta keskittyvää tutkimusta on toistaiseksi vähän, mutta tutkimusta tieto- ja viestintäteknologian opetuskäytöstä ja ohjelmoinnista yleisesti on paljon. Opettajien käsityksiä ja kokemuksia on kuitenkin sivuttu joissain tutkimuksissa ja selvityksissä.

Uusi opetussuunnitelma (OPS 2014) velvoittaa uudella tavalla toteuttamaan ohjelmoinnin opetusta. Opettajien valmiuksissa kuten myönteisissä asenteissa opetusteknologian käyttöä kohtaan (vrt. Albirini 2006; Hietikko, Ilves & Salo 2016) ja puutteellisissa ohjelmointitaidoissa (vrt. Alimisis ym. 2007; Kaarakainen ym. 2017) ei kuitenkaan mielenkiintoisesti näytä tapahtuneen merkittävää muutosta viimeisen vuosikymmenen aikana, vaikka opettajien tvt-osaaminen onkin yleisesti lisääntynyt (Hietikko ym. 2016). Kiinnostavaa on myös, että tämän päivän tvt:n opetuskäytön ja ohjelmoinnin opetuksen ongelmakohdat muistuttavat 1980-luvun ensiaskelia tvt:n (tuolloin atk:n nimellä toteutetussa) opetuksessa suomalaiskouluissa. Opettajat kuvasivat tuolloin, lukio-opetuksen kontekstissa, opetuksen haasteiden liittyvän muun muassa seuraaviin tekijöihin: tietokoneiden vähäisyys, ajan puute sekä puutteellinen opettajankoulutus (Saarikoski 2006). Ajan puute (ks. Şahin-Kızıl 2011), opettajien heikko osaaminen sekä käytettävissä olevien tvt-laitteiden vähäinen määrä (ks. Israel ym. 2015) ovat nousseet esille myös viimeaikaisten tvt:n opetuskäyttöä ja ohjelmoinnin opetusta koskevien tutkimusten tuloksissa.

Tutkimushavainnot tvt:n opetuskäytöstä. Opettajilla on todettu olevan yleisesti varsin myönteinen suhtautuminen opetusteknologiaa kohtaan (ks. Albirini 2006; Cavas, Cavas, Karaoglan & Kisla 2009; Hietikko ym. 2016). Opettajien tvt-osaamisen on tutkimuksissa havaittu ennustavan myönteistä suhtautumista opetusteknologian käyttöä kohtaan (ks. Albirini 2006; Cavas ym. 2009; Sherer, Tondeur, Siddiq & Baran 2018). Tutkimustulokset opettajien tvt-

osaamisesta ovat kuitenkin ristiriitaisia. Osa tutkimustuloksista viittaa siihen, että opettajien tv-taidot ovat varsin hyvät (ks. Kaarakainen ym. 2017; Karvonen & Laukka 2016; Taussi 2017), kun taas osassa opettajien tv-osaamisen on havaittu olevan puutteellista (ks. Hietikko ym. 2016).

Hietikko on kollegoineen (2016) raportoinut Opettajien Ammattijärjestön (OAJ) selvitystyötä koulutuksen digitalisaation tilasta ja keskeisimmistä digiloikkaa edistävästä ja estävistä tekijöistä. Kaikki kouluasteet kattavaan kyselyyn vastasi 1515 henkilöä, jotka toimivat lastentarhanopettajina ja päiväkotien johtajina, perusopetuksen opettajina ja rehtoreina, lukion opettajina ja rehtoreina, sekä ammatillisen oppilaitoksen, ammattikorkeakoulun ja yliopiston opettajina. Selvitys osoitti, että opettajilla on myönteinen asenne opetusteknologiaa kohtaan, mutta esiin tuli riittävän täydennyskoulutuksen tarkeys. Suuri osa opettajista kuvaili tv-valmiuksiaan heikoiksi tai kohtalaisiksi. Opettajia ja oppilaitosten johtajia huolestuttivat muun muassa oppilaitosten erilaiset varustetasot ja osaamisen ja koulutuksen puute. Lisäksi selvisi, että tv:n yleistymisen myötä opettajien opetukseen käyttämä aika oli pysynyt samana tai hieman kasvanut, mutta viestintään, opetuksen valmisteluun ja oppimateriaalin kokoamiseen käytetty aika oli lisääntynyt.

Tutkimushavainnot ohjelmoinnin opetuksesta. Opettajien asenne ohjelmointia kohtaan on aiemmissa tutkimuksissa havaittu neutraaliksi tai myönteiseksi (ks. Karvonen & Laukka 2016). Toisaalta on ilmennyt, että opettajat eivät ole innostuneita oppimaan lisää ohjelmoinnista, mikäli eivät hallitse jo valmiiksi ohjelmointitaitoa (Saari, Blanchfield & Hopkins 2016). Lisäksi monet opettajat kokevat, että heillä on riittämättömät taidot ja työvälineet tv:n käyttämiseen opetuksessa (Hietikko ym. 2016), ja vain harva opettaja kokee osaavansa ohjelmoida jossakin graafisessa ohjelmointiympäristössä tai jollakin lausekielellä (Kaarakainen ym. 2017). Osaamattomuus ja tietämättömyys aiheesta voi estää teknologian käyttöönottoa osaksi opetusta sekä tunnistamasta esimerkiksi robotiikan hyötyjä opetuksessa (ks. Alimsis ym. 2007; Matarić, Koenig & Feil-Seifer 2007).

Karvosen ja Laukan (2016) opinnäytetyössä tutkittiin verkkokyselyn avulla 317 opettajan ohjelmointiosaamista, näkemyksiä ohjelmoinnin asemasta pe-

rusopetuksessa sekä asenteita ohjelmointia kohtaan. Kysely toteutettiin verkkosivuston koodiaapinen.fi ohjelmointikurssin osallistujille. Tutkimukseen osallistuneista lähes kaikki olivat peruskoulussa opettavia opettajia, ja yli puolet alakoulussa toimivia opettajia. Noin joka kymmenes vastaajista arvioi pystyvänsä opettamaan jotain ohjelmointikieliä Scratch, Scratch Jr., Code.org tai HTML/CSS, mutta vain muutamat raportoivat osaavansa opettaa jollain muulla ohjelmointikielellä, kuten Hopscotchilla, Logolla, Pascalilla tai TurtleRoylla. Suurin osa opettajista kuvaili uskovansa pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia, jos käytettävissä on riittävästi aikaa tai tukea. Lisäksi opettajien käsitys oli se, että ohjelmoinnin opetus peruskoulussa on tarpeellista. Opettajien asenne ohjelmointia kohtaan havaittiin neutraaliksi tai lievästi myönteiseksi.

Taussi (2017) puolestaan tutki opinnäytetyössään ohjelmoinnin opetusta perusopetuksen ensimmäisellä luokalla. Tutkimuksen tarkoituksena oli selvittää, millä tavoin ohjelmointia opetetaan 1.-luokkalaisille sekä millaista on opettajien oma tvt-osaaminen ja millaisia kokemuksia opettajilla on ohjelmoinnin opettamisesta. Tutkittavat olivat ensimmäisen luokan opettajia, ja aineistona oli yksi haastattelu ja 29 verkkokyselyn vastausta. Tutkimustulokset osoittivat, että alkuopetuksessa ohjelmoinnin opetuksessa hyödynnettiin paperilla tehtäviä harjoituksia, Lego- ja Bee-Bot-robotteja sekä kapteeni käskee -leikkiä. Lisäksi käytettiin sovelluksia kuten Scratch Jr., Kodable ja Bomberbot, verkkosivustoja kuten Code.org sekä Hello Ruby ja Dibitassut -ohjelmointikirjoja, Robogem-lautapeliä ja Multilink-palikoita. Mielenkiintoista Taussin tuloksissa on, että vain 10 prosenttia tutkimukseen osallistuneista opettajista käytti tietoteknisiä sovelluksia ohjelmoinnin opettamiseen. Lisäksi tutkimustulokset osoittivat, että opettajat arvioivat omat tvt-taitonsa hyviksi tai erinomaisiksi. Puolet opettajista kertoi saaneensa perehdytystä ohjelmoinnin opettamiseen työnantajalta tai oli perehtynyt aiheeseen omalla ajalla, mutta suurin osa kuvaili ohjelmoinnin opettamista silti haasteelliseksi.

Kaarakainen ja kumppanit (2017) ovat esitelleet Digiajan peruskoulu -hankkeen väliraportissaan keskeisiä tuloksia perusopetuksen digitalisaation tilasta. Aineisto koottiin verkkokyselyllä, johon vastasi peruskoulun rehtoreita

(n=217), opettajia (n=1990) ja oppilaita (n=25838). Lisäksi 2162 opettajaa ja 5455 9.-luokkalaista vastasi ICT-taitotestiin. Ohjelmoinnin osalta tutkimuksessa selvisi, että opettajien ja oppilaiden ohjelmointiosaaminen on varsin heikkoa. Tutkimukseen osallistuneista opettajista 90 % kuvasi, ettei osaa ohjelmoida millään lausekielellä, kuten Python- tai Java-kielellä, kun taas noin 70 % kuvasi, ettei hallitse sujuvasti mitään graafista ohjelmointiympäristöä (esim. Scratch tai Kodu). Useampi siis hallitsi graafiset ohjelmointiympäristöt kuin tekstipohjaiset ohjelmointikielet. Alakoulun opettajat osasivat käyttää yläkoulun opettajia paremmin graafisia ohjelmointiympäristöjä, mutta yläkoulun opettajat hallitsivat paremmin lausekielisen ohjelmoinnin. Vain viidennes vastaajista oli ylipäätään kokeillut ohjelmointia opetuksessa. Yli puolet opettajista arvioi kuitenkin omat tv-taitonsa perustasoisiksi ja noin viidennes kehittyneiksi. Vähemmän työvuosia opettaneilla opettajilla oli paremmat tv-taidot kuin pidempään opettaneilla.

Israel kollegoineen (2015) tarkasteli tapaustutkimuksessaan, miten erityiskoulussa työskentelevät opettajat, joiden tv-taidot ovat vähäiset, integroivat tieto- ja viestintäteknikkaa eri opetustilanteissa. Osallistujina oli seitsemän luokanopettajaa ja kaksi hallinnossa toimivaa henkilöä, ja aineisto koostui tutkittavien haastatteluista sekä opetustilanteiden havainnoinneista. Tutkimuksessa selvisi opettajien kokemuksiin liittyen, että vaikka opettajilla oli pääosin myönteinen suhtautuminen ohjelmointia kohtaan, he suhtautuivat aluksi varauksella tv:n integrointiin opetukseen. Toisaalta, kun heille tarjottiin asiantuntijoiden tukea, heidän itseluottamuksensa ja kokemuksensa lisääntyivät, ja heidän asenteensa olivat myönteisempiä integroinnin suhteen. Opettajat kuvailivat tv:n integroimisen suurimmiksi haasteiksi tv-laitteiden vähäisyyden, tv-osaamisen puutteen ja oppilaiden taitoerot näiden eritasoisista taustoista johtuen. Keskeistä integroinnin onnistumisen kannalta oli opettajien ammatillinen kehittyminen, ohjauksen saaminen ja tv-taitojen kehittyminen.

4 TUTKIMUSTEHTÄVÄ

Tutkimuksen tarkoituksena oli selvittää, millaisia käsityksiä ja kokemuksia luokanopettajilla on ohjelmoinnin opettamisesta. Luokanopettajien käsityksiä ja kokemuksia ohjelmoinnista oli tärkeää selvittää, koska opetussuunnitelma (OPS 2014) toi ohjelmoinnin osaksi luokanopettajien toteuttamaa alakouluopetusta syksystä 2016 alkaen. Kiinnostuksen kohteena on, millaista ohjelmoinnin opetus on opettajan näkökulmasta ja millä tavalla ohjelmoinnin opetusta toteutetaan alakoulussa.

5 TUTKIMUKSEN TOTEUTTAMINEN

5.1 Metodologiset valinnat

Tämä tutkimus on laadullinen tutkimus, jonka tavoitteena on kuvata tutkittavaa ilmiötä ja tehdä siitä tulkintoja (Patton 2015). Lähtökohtana oli laadullisen tutkimuksen tavoin todellisen elämän kokonaisvaltainen kuvaaminen (Hirsjärvi, Remes & Sajavaara 2016; Tuomi & Sarajärvi 2018). Tutkimuksen kohteena olivat opettajien käsitykset ja kokemukset. Tavoitteena oli, että tutkimukseen osallistujien ääni kuuluisi, ja heidän ajatuksensa ja kokemuksensa pääsisivät esiin mahdollisimman autenttisesti (ks. Hirsjärvi ym. 2016). Laadullinen tutkimus soveltui tähän tutkimukseen hyvin, koska tutkimuksen ensisijaisena tarkoituksena ei ollut tuottaa laaja-alaisesti yleistettävää tietoa aiheesta (ks. Hirsjärvi ym. 2016), vaan kiinnostuksen kohteena olivat rajatun kohderyhmän yksilöiden käsitykset ja kokemukset ohjelmoinnin opettamisesta. Tutkimus keskittyi pieneen otokseen perustuvien haastattelujen perinpohjaiseen analysointiin. Litteroitu haastatteluaineisto analysoitiin sisällönanalyysin menetelmällä.

Tämän tutkimuksen lähestymistavassa on piirteitä sekä fenomenologis-hermeneuttisesta että fenomenografisesta tutkimusotteesta. Tutkimuksessa haluttiin selvittää, millaisia käsityksiä opettajilla on ohjelmoinnin opettamisesta ja miten he ovat ohjelmoinnin opettamisen kokeneet. On luonnollista tutkia samassa yhteydessä sekä käsityksiä että kokemuksia, sillä niiden voidaan ajatella kulkevan käsi kädessä. Kokemukset liittyvät tiiviisti käsityksiin, koska yksilön kokemukset muotoutuvat niille annettavien merkitysten mukaan (Laine 2010), ja toisaalta yksilön kokemustausta muovaa tämän mielikuvia ilmiöistä (Metsämuuronen 2009).

Fenomenologis-hermeneuttisessa tieteenfilosofisessa suuntauksessa keskeisessä roolissa ovat yksilöiden kokemukset ja näiden kokemusten merkitykset, joiden ajatellaan rakentuvan vuorovaikutuksessa ympäristön kanssa (Laine 2010). Fenomenologinen tutkimus tutkii kokemuksia, jotka rakentuvat merkitysten mukaan (Laine 2010). Nämä voidaan käsittää ihmisen kokemuksellisenä

suhteena ympäröivään maailmaan ja yksilön omaan todellisuuteen. Kiinnostuksen kohteena olevia ilmiöitä pyritään tarkastelemaan ja tulkitsemaan (Metsämuuronen 2009). Hermeneuttisessa tutkimuksessa keskeistä on sekä tutkittavan kyky ilmaista omia kokemuksiaan ja käsityksiään että tutkijan kyky ymmärtää ja tulkita niitä. Tässä tutkimuksessa kiinnostuksen kohteena olivat tutkimukseen osallistujien kokemukset ja heidän niille antamat merkitykset. Tutkija pyrki ymmärtämään haastateltavien ilmausten merkityksiä ja tekemään ilmausten perusteella tulkintoja opettajien kokemuksista ohjelmoinnin opetuksen toteuttamisesta. Kokemusten ajateltiin tässä yhteydessä kattavan opettajien tuntemukset, havainnot, kiinnostukset ja toiminnan sekä toiminnan syyt (ks. Laine 2010). Tarkastelun kohteena olivat myös opettajien arjen toiminnastaan tekemät kuvaukset sekä heidän muodostamansa mielipiteet ja arvioinnit.

Fenomenografiassa keskeisenä tutkimuskohteena ovat Hirsjärven ja Hurmeen (2008) mukaan yksilön käsitykset. Tällöin tavoitteena on tarkastella ihmisten erilaisia käsityksiä tutkittavasta ilmiöstä (Metsämuuronen 2009). Yksilöiden käsitykset asioista voivat vaihdella muun muassa heidän taustastaan, iästään ja sukupuolestaan riippuen, ja käsitykset voivat dynaamisuutensa vuoksi muuttua (Metsämuuronen 2009). Tässä tutkimuksessa tarkastelun kohteena olivat opettajien kokemusten lisäksi myös heidän moninaiset käsityksensä ohjelmoinnin opettamisesta. Haastateltavien ilmausten joukosta etsittiin erilaisten käsitysten kirjoa tutkittavasta aiheesta. Käsitysten ajateltiin tässä tutkimuksessa pitävän sisällään myös opettajien uskomukset, sillä niitä on vaikeaa erottaa toisistaan tutkittavien ilmaisuissa siitä, miten he käsittävät jonkin asian.

5.2 Tutkittavat ja tutkimuksen eteneminen

Tutkittavat. Tutkimuksen aineisto koostui kuuden luokanopettajan yksilöhaastatteluista (5 naista ja 1 mies). Tutkimukseen osallistujat olivat koulutukseltaan luokanopettajia ja työskentelivät tutkimushetkellä luokanopettajina neljässä eri koulussa neljällä paikkakunnalla. Haastateltavista kolme oli Päijät-Hämeestä ja kolme Keski-Suomesta. Haastateltavien keski-ikä oli 35 vuotta (24–44 vuotta) ja keskimääräinen työkokemus 11 vuotta (1–20 vuotta). Opettajista kolme oli toi-

minut opettajana yli 15 vuotta, yksi noin 6 vuotta ja kaksi alle 2 vuotta. Kolme heistä opetti 3. luokkaa, yksi 4. luokkaa ja yksi 5. luokkaa, ja yksi opetti luokkia 1–3. Haastateltavien opettamilla luokka-asteilla ei kuitenkaan ollut tämän tutkimuksen kannalta merkitystä, sillä opetussuunnitelma (OPS 2014) sisällyttää ohjelmoinnin opetusta jokaiselle perusopetuksen vuosiluokalle.

Neljällä haastateltavalla oli kertomansa mukaan kokemusta ohjelmoinnin opettamisesta, kun taas kahdella ei ollut ollenkaan kokemusta. Ohjelmointia opettaneista opettajista kolme kertoi opettavansa oman luokkansa lisäksi koulussaan myös valinnaisia tai pakollisia tvt- tai koodauskursseja. Opettajat arvioivat opettavansa ohjelmointia omalle luokalleen satunnaisesti jaksoissa noin kerran kuussa. Ohjelmoinnin valinnaisainetta opettavat opettajat opettivat sitä säännöllisesti viikoittain tai jaksoissa. Opettajien ohjelmointikokemus ei ollut edellytys tutkimukseen osallistumiselle, mutta tutkittavien heterogeeniset kokemukset ohjelmoinnin opettamisesta olivat toivottuja.

Tutkimuksen eteneminen. Tutkimusprosessi käynnistyi alkuvuodesta 2017 tutkimuksen aiheen rajauksen varmistuttua. Haastattelurunko valmistui keväällä 2017, ja sitä edelsivät alustavat tutkimuskysymykset, jotka ohjasivat haastattelukysymysten muotoutumista. Ennen haastattelujen toteuttamista olin tutustunut tutkimusaiheeseeni perehtymällä aihetta käsitteleviin lähteisiin. Haastattelukysymykset muovautuivat osittain kirjallisuudesta aiheesta saadun esiyymmärryksen pohjalta. Haastattelukysymysten pilotointi toteutui kahdella pilottihaastattelulla toukokuussa ja marraskuussa 2017 ennen varsinaisten haastattelujen toteuttamista.

Haastateltavien etsiminen aloitettiin paikkakuntien valinnalla. Tutkimukseen osallistuvien paikkakuntien ja koulujen valinnassa painottui niiden sijainti sopivan etäisyyden päässä mahdollistamaan paikan päällä tehtävät haastattelut. Otin yhteyttä yhteensä viiteen paikkakuntaan ja yhdeksään kouluun Keski-Suomessa ja Päijät-Hämeessä. Suullisten tutkimuslupien saamisen jälkeen valittujen paikkakuntien koulutoimenjohtajilta ja koulujen rehtoreilta koulujen luokanopettajille esitettiin haastattelupyynnö. Opettajia lähestyttiin pääsääntöisesti sähköpostitse. Yhteydenotoissa opettajille kuvailtiin tutkimuksen aihe, tarkoitus ja aineistonkeruumenetelmät. Haastattelupyynnöjä jouduttiin lähettämään

suurelle määrälle opettajia sen vuoksi, että suurin osa jätti vastaamatta haastattelukutsuun. Tutkimukseen osallistujiksi valikoituivat ne luokanopettajat, jotka vastasivat myöntävästi haastattelukutsuun.

5.3 Aineiston keruu

Aineistonkeruu toteutettiin tammi-toukokuun 2018 aikana. Aineistonkeruu tapahtui yksilöhaastatteluin, mikä on Hirsjärven ja Hurmeen (2008) mukaan tavallisin tapa suorittaa aineistonkeruuhaastattelu. Haastattelumenetelmänä oli puolistrukturoitu teemahaastattelu.

Tässä tutkimuksessa aineistonkeruumenetelmänä käytetty haastattelu on yksi kvalitatiivisen tutkimuksen yleisimmistä aineistonkeruumenetelmistä (Tuomi & Sarajärvi 2018). Menetelmäksi valikoitui haastattelu, koska tutkimuksessa haluttiin saada tietoa tutkittavien autenttisista ajatuksista ja toiminnasta, mikä onnistuu erityisen hyvin juuri haastattelun keinoin (ks. Tuomi & Sarajärvi 2018). Hirsjärven ja Hurmeen (2008) mukaan haastattelulla pyritään tavoittamaan haastateltavan käsityksiä, ajatuksia, kokemuksia ja tunteita. Haastateltava pukee kokemuksensa sanoiksi ja tutkija pyrkii tekemään tulkintoja haastateltavan ilmausten perusteella (Laine 2010). Vaikka haastattelu onkin aikaa vievä aineistonkeruumenetelmä (Tuomi & Sarajärvi 2018), se sopi aineistonkeruumenetelmäksi tähän tutkimukseen, koska sen avulla haastateltavien ääni pääsee kuuluviin, sen joustavuus auttaa selventämään ja syventämään haastateltavien antamia vastauksia, ja haastattelukysymysten esittämisjärjestys voi vaihdella keskustelun etenemisen mukaan (ks. Hirsjärvi & Hurme 2008).

Haastattelumenetelmäksi valittiin puolistrukturoitu teemahaastattelu, koska tavoitteena oli saada haastateltavien opettajien ääni kuuluviin mahdollisimman hyvin (ks. Hirsjärvi & Hurme 2008). Puolistrukturoidussa haastattelussa kaikille haastateltaville esitetään samat kysymykset, mutta haastateltavalla on tilaa vastata niihin omin sanoin (Eskola & Suoranta 2014). Teemahaastattelussa keskustelu keskittyy tiettyjen teemojen ympärille (Hirsjärvi & Hurme 2008). Haastattelujen pohjana oli aikaisemmin luotu haastattelurunko, jossa oli pääkysymyksiä eri teemoista sekä tarkentavia alakysymyksiä (ks. liite 2). Haas-

tattelukysymykset suunniteltiin pääosin avoimiksi kysymyksiksi, joiden etu on Pattonin (2015) mukaan se, että ne eivät rajoita vastaamista, kun niihin vastataan omin sanoin.

Haastattelut toteutettiin haastateltavien työpaikoilla koulujen tiloissa, joko haastateltavien omissa luokkahuoneissa tai koulun neuvottelutilassa, koska paikan haluttiin olevan rauhallinen ja haastateltavalle turvallinen (ks. Hirsjärvi & Hurme 2008). Haastattelutilanteessa opettajilta pyydettiin kirjallinen suostumus haastatteluun, ja tutkijana annoin heille kirjallisen lupauksen siitä, että aineistoa käsitellään luottamuksellisesti (ks. liite 1). Haastattelut nauhoitettiin.

5.4 Aineiston analyysi

Aineiston analyysi ja tulosten raportointi tapahtuivat elo-lokakuussa 2018. Analyysissä käytettiin aineistolähtöistä sisällönanalyysiä, jolla pyrittiin tavoittamaan tutkittavien käsityksiä ja kokemuksia ja niille annettuja merkityksiä. Sisällönanalyysin menetelmällä aineisto ensin pelkistettiin ja luokiteltiin. Tämän jälkeen aineistosta muodostettiin teemoja, joiden avulla pyrittiin vastaamaan tutkimuskysymyksiin.

Tutkimuksen aineiston analyysissä käytettiin aineistolähtöistä sisällönanalyysiä, koska haluttiin muodostaa aineistosta esiin nousevien teemojen avulla ja tulkintoja tekemällä kokonaiskuva tarkasteltavasta ilmiöstä sekä yhdistää löydökset aiempaan tutkimustietoon (ks. Tuomi & Sarajärvi 2018). Hirsjärven ja kumppaneiden (2016) mukaan laadullisessa tutkimuksessa voidaan tällaisen analyysin avulla tarkastella aineistoa yksityiskohtaisesti ja monipuolisesti ilman hypoteeseja. Sisällönanalyysi soveltuu sekä käsityksiä että kokemuksia ilmaisevien ilmaisujen luokitteluun. Tuomen ja Sarajärven (2018) mukaan fenomenologis-hermeneuttisessa tutkimuksessa hyödynnetään usein aineistolähtöistä sisällönanalyysiä, jolla pyritään samaan teoreettista ymmärrystä ilmiöstä. Sisällönanalyysi soveltuu sekä käsityksiä että kokemuksia ilmaisevien ilmaisujen luokitteluun. Analyysissä ilmaisuja luokitellaan ja tutkija pyrkii selittämään ilmausten merkityksiä kokoamalla niitä yhteen (Metsämuuronen 2009).

Sisällönanalyysi analyysimenetelmänä. Tuomi ja Sarajärvi (2018) esittävät, että laadullisessa sisällönanalyysissä aineisto pilkotaan ensin osiin, käsitteellistetään ja lopuksi järjestetään uudeksi loogiseksi kokonaisuudeksi. Analyysi perustuu päättelyyn ja tulkintojen tekemiseen. Tavoitteena on saada tiivistetty kuvaus ilmiöstä ja kytkeä tutkimustulokset ilmiön laajempaan kontekstiin sekä muihin aihetta koskeviin tutkimustuloksiin. Alasuutari (2011) lisää, että ilmiöstä pyritään muodostamaan kokonaiskuva selittämällä tuloksia aineistosta löydettyjen johtolankojen pohjalta. Tuloksia myös tarkastellaan esimerkkinä muustakin kuin vain tutkitusta yksittäistapauksesta.

Tässä tutkimuksessa aineiston analysoinnissa edettiin pienemmistä yksiköistä suurempia kokonaisuuksia kohti pyrkien saamaan kokonaiskuva käsiteltävästä ilmiöstä induktiivisen logiikan mukaisesti (ks. Patton 2015; Tuomi & Sarajärvi 2018). Tutkimusaineiston tarkastelusta edettiin kohti teoreettista ymmärrystä eli yksittäisistä havainnoista kohti yleisiä teemoja. Hirsjärvi ja kumppanit (2016) tarkentavat, että koska induktiivisessa analyysissä tutkija pyrkii paljastamaan odottamattomia tuloksia aineistosta, aikaisemmat teoriat ja tiedot eivät johda analyysiä, ja analyysiyksiköitä ei päätetä etukäteen. Sen sijaan aineistoa pyritään tarkastelemaan mahdollisimman yksityiskohtaisesti ja monipuolisesti. Se oli myös tämän tutkimuksen tavoite, sillä tarkoituksena oli selvittää opettajien käsityksiä ja kokemuksia ohjelmoinnin opettamisesta, ei pyrkiä muodostamaan yleistettävää kuvausta ohjelmoinnin opettamisesta alakoulussa. Lisäksi analyysissä lähestymistapana oli Krippendorffin (2013) esittelemä ongelmalähtöinen lähestymistapa.

Aineiston litterointi. Sisällönanalyysissä tarkastelun kohteena on tekstimuotoinen aineisto (Tuomi & Sarajärvi 2018), joka on tässä tapauksessa litteroitu haastatteluaineisto. Ennen aineiston analyysiä puhemuotoinen haastatteluaineisto litteroitiin eli kirjoitettiin auki tietokoneella tekstinkäsittelyohjelmalla. Haastattelujen kesto oli keskimäärin 37 minuuttia (27 min – 55 min). Yhteensä nauhoitettua haastatteluaineistoa kertyi noin 3 tuntia ja 39 minuuttia. Litteroidun tekstiaineiston koko oli yhteensä 41 sivua (fontti Times New Roman, 12 pt, riviväli 1). Litterointivaiheessa kukin haastateltava nimettiin peitenimellä (Emma, Heidi, Marianna, Minna, Teemu ja Terhi).

Koska kielenkäyttö ei ole analyysin kohteena tässä tutkimuksessa, karkea litterointitapa oli riittävä (ks. Saaranen-Kauppinen & Puusniekka 2006). Sana-tarkka litterointi tehtiin vain haastateltavan puheen sisällöstä, koska se oli analyysin kohteena. Pois litteraatista jätettiin kuitenkin joitain toistuvia täytesanoja. Litteroimatta jätettiin lisäksi haastattelujen sellaiset osat, jotka eivät olleet tässä tutkimuksessa tarkastelun kohteena, kuten haastattelun aloitukseen ja lopetukseen liittyvä puhe. Haastattelijan puheenvuoroista kirjattiin ylös vain esitetyt kysymykset tai oleelliset tarkennukset.

Analyysiprosessi. Ennen varsinaisen analyysin aloittamista litteroituun aineistoon tutustuttiin lukemalla se huolella läpi. Tämän jälkeen aineistoa analysoitiin sisällönanalyysin kolmen vaiheen mukaisesti aineistoa pelkistämällä ja ryhmittelemällä sekä luomalla kokoavia teoreettisia käsitteitä (ks. Tuomi & Sarajärvi 2018).

Analyysin ensimmäisessä vaiheessa tutkimuksen aineisto *pelkistettiin* eli jaettiin tutkimuksen kannalta merkityksellisiin ilmauksiin eli merkitysyksiköihin. Tuomen ja Sarajärven (2018) mukaan aineiston pelkistämisen eli redusointivaiheessa aineisto tiivistetään ja pilkotaan pienempiin osiin siten, että jäljelle jäävät vain tutkimuksen kannalta olennaiset osat. Analysoitaviksi tämän tutkimuksen haastatteluaineistosta valittiin tutkimuskysymysten kannalta olennaiset kohdat. Redusointi tehtiin värikoodaamalla aineisto. Koodaamisen avulla aineistoa pystytään jaottelemaan ja järjestelemään (Tracy 2013). Aineistosta koodattiin aluksi eri värein ilmauksia, jotka liittyivät tutkimuskysymyksiin. Analyysiyksikköinä olivat haastatteluaineiston haastateltavien puheenvuorojen virkkeet tai useasta virkkeestä koostuvat ajatuskokonaisuudet (ks. Tuomi & Sarajärvi 2018). Tämän jälkeen aineistossa esiintyneistä alkuperäisilmauksista muodostettiin pelkistettyjä ilmauksia, mistä esitellään esimerkki taulukossa 1.

TAULUKKO 1. Esimerkki aineiston redusoinnista

Alkuperäisilmaukset	Pelkistetyt ilmaukset
– – on ollu mahtava sit huomata, meil ku on ollu esimerkiks valinnaisaineena ohjelmointi, niin aivan intopenttinä muutama sellanen oppilas, jotka on muuten sellasia passiivisia ja negatiivisesti suhtautuvia koulunkäyntiin – –	Oppilaiden myönteiset kokemukset
No on todellakin [eroa oppilaiden välillä]. Niit on siis todellakin yö ja päivä. Niinku mä sanoin, on niitä valtavan edistyneitä, tuntuu et ne tekee ihan uskomattomia projekteja sillä koneella. Sit on niitä, jotka ei tosiaan uskalla kokeilla yhtään mitään. – –	Oppilaiden taitoerot
No meil on 2 [Lego-robottia]. Ne on tuolla laatikossa. Kaks robottia on. Me pärjätään niillä periaatteessa. Mut et se tilanne ei oo sama koko kunnassa. – –	Laitteistoresurssit

Aineiston pelkistämisen jälkeen analyysin toisessa vaiheessa aineisto *klusteroitiin* muodostaen luokkia teemoittelun avulla. Tuomen ja Sarajärven (2018) mukaan klusteroinnissa eli ryhmittelyssä aineistosta etsitään ja ryhmitellään yhteen samankaltaisuuksia kuvaavia ilmauksia. Samaa asiaa kuvaavat ilmaukset yhdistellään alaluokiksi tai alateemoiksi, jotka nimetään luokan sisältöä kuvaavalla käsitteellä. Samalla aineisto tiivistyy. Tässä tapauksessa samoilla väreillä merkityjä ilmauksia ryhmiteltiin yhteen muodostamaan teemoja. Aineiston klusteroinnissa käytettiin apuna haastattelukysymyksiä. Kaikki analyysiyksiköt siirrettiin aluksi erilliseen tiedostoon ja ryhmiteltiin haastattelukysymysten alle. Tämän jälkeen yhdisteltiin samankaltaisia ilmauksia ja luotiin aineiston sisältöä kuvaavat alateemat. Tästä esimerkkinä on taulukko 2.

TAULUKKO 2. Esimerkki aineiston klusteroinnista

Pelkistetyt ilmaukset	Alateema
Laitteistoresurssit Suunnittelu-aika Ajan puute	Välineet ja aika resurssina
Pelit Leikit Toiminnallisuus	Ei-teknologisten opetusmenetelmien toiminnallisuus

Aineiston klusteroinnin jälkeen analyysin kolmannessa vaiheessa aineisto *abstrahoitii*. Tuomen ja Sarajärven (2018) mukaan abstrahoinnissa eli käsitteellistämässä tutkija erottelee aineistosta tutkimuksen kannalta olennaisen tiedon ja muodostaa sen perusteella teoreettisia käsitteitä ja tekee johtopäätöksiä. Alateemojen muodostamisen jälkeen yhdistelemällä samaan teemaan liittyviä alateemoja luotiin kaksi pääteemaa, jotka vastaavat tämän tutkimuksen tutkimuskysymyksiin. Esimerkki tästä on taulukossa 3.

TAULUKKO 3. Esimerkki aineiston abstrahoinnista

Alateema	Pääteema
Ohjelmointi toimintaohjeina Ohjelmointi välineellisenä taitona Ohjelmointi arjen taitojen tukijana	OPETTAJIEN KÄSITYKSET OHJELMOINNIN OPETTAMISESTA
Välineet ja aika resurssina Asenteet ja taidot resurssina Ohjelmoinnin opetuksen haasteellisuus	OPETTAJIEN KOKEMUKSET OHJELMOINNIN OPETTAMISESTA

Analyysiprosessin aikana muodostetut pelkistetyt ilmaukset sekä prosessin tuloksena luodut teemat ovat kokonaisuudessaan nähtävissä taulukossa 4.

TAULUKKO 4. Aineiston analyysissä tunnistettuja teemoja

Pääteema	Alateema	Pelkistetyt ilmaukset
1. OPETTAJIEN KÄSITYKSET OHJELMOINNIN OPETTAMISESTA	Ohjelmointi kahdentyyppisenä toimintana	ohjelmointi tv-t-laitteilla, ohjelmointi ilman tv-t-laitteita
	Ohjelmointi toimintaohjeina	toimintaohjeiden antaminen, ohjeiden noudattaminen
	Ohjelmointi opetussuunnitelmallisena uudelleen nimeämisenä	uudelleen nimeäminen, ohjelmointi ei uusi asia, ohjelmointia koulun arjessa
	Ohjelmoinnin opetuksen merkitys	ohjelmoinnin opetuksen tärkeys, tarpeellisuus arkielämässä ja työelämässä, ei prioriteetti
	Ohjelmoinnin opetuksen keskeneräisyys	keskeneräisyys, epäselvyys, täsmennysten tarve
	Ohjelmointi välineellisenä taitona	eri taitojen kehittyminen ohjelmoinnin avulla, matemaattiset taidot, tekninen ohjelmointitaito, arjen taidot
	Ohjelmointi matemaattis-teknologisten taitojen tukijana	tietokoneen logiikka, tekninen ohjelmointitaito, matemaattiset taidot, looginen päättely, ongelmanratkaisu, palastelu
Ohjelmointi arjen taitojen tukijana	arjen taidot, kriittisyys, luetun ymmärtäminen, järjestelmällisyys, hahmottaminen, ohjeen noudattaminen, keskittyminen, sinnikkyys, syy-seuraus-suhde	
2. OPETTAJIEN KOKEMUKSET OHJELMOINNIN OPETTAMISESTA	Välineet ja aika resursseina	laitteistoresurssit, raha, aika-resurssit, suunnittelu-aika, opetuksen käytettävä aika, ajan puute
	Asenteet ja taidot resursseina	opettajien myönteinen asenne, puutteelliset ohjelmointitaidot
	Ei-teknologisten opetusmenetelmien toiminnallisuus	pelit, leikit, toiminnallisuus, opetusmateriaalit, havainnollistaminen
	Teknologisten opetusmenetelmien monipuolisuus	sovellukset, sivustot, robotit, saatavuus, helppokäyttöisyys, eriyttäminen, edistymisen seuranta, soveltuvuus integrointiin, tuttuus
	Ohjelmoinnin integroinnin laajat mahdollisuudet	integrointi eri oppiaineisiin, lisäarvon tuominen opetukseen, integroinnin helppous ja monet mahdollisuudet, ajan puute

(jatkuu)

TAULUKKO 4. Aineiston analyysissä tunnistettuja teemoja (jatkuu)

Pääteema	Alateema	Pelkistetyt ilmaukset
2. OPETTAJIEN KOKEMUKSET OH- JELMOINNIN OPETTAMISESTA	Ohjelmoinnin opettami- sen herättämät tunteet	opettajien puutteellinen osaami- nen, ohjelmointia opetettu ai- emminkin, opetuksen mielek- kyys, neutraalit tunteet, opetu- ksen haasteellisuus, jännitys
	Ohjelmoinnin opetuksen haasteellisuus	opetuksen haasteellisuus, aika- resurssit, laitteistoresurssit, lait- teiden toimimattomuus, oppi- laiden taitoerot, eriyttäminen
	Tuen tarve ohjelmoinnin opetuksessa	tuen tarve, täydennyskoulutuk- sen puute, oma perehtyminen, kollegiaalinen tuki, koulutukset, lisäkouluttautumisen välttämät- tömyys
	Ohjelmointi useimpien oppilaiden innostajana	oppilaiden myönteiset koke- mukset, oppilaiden kielteiset kokemukset, oppilaiden suku- puolella ei yhteyttä kokemuk- siin, oppilaiden taitoerot, erot kiinnostuksessa, erot laitteiden- käyttökokemuksessa

Lopuksi teemojen luomisen ja tulosten kirjaamisen jälkeen aineiston pohjalta lähdettiin tekemään tulkintoja ja johtopäätöksiä. Tämä tarkoittaa Pattonin (2015) mukaan aineiston kuvailutasoa syvemmälle menemistä, löydösten tulkitsemista ja merkitysten ja selitysten antamista löydöksille, johtopäätösten vetämisestä, päätelmien tekemistä sekä merkitysten pohtimista. Laine (2010) kuvailee, että analyysin lopuksi tutkija pyrkii luomaan kokonaiskuvan tutkittavasta ilmiöstä ja tekemään synteisiä eli yhteenvedoa merkityksistä ja niiden välisistä suhteista. Samalla pyritään Eskolan ja Suorannan (2014) mukaan näkemään ilmiöön liittyviä syvempiä merkityksiä.

5.5 Luotettavuus ja eettiset ratkaisut

5.5.1 Eettiset ratkaisut

Laadullisessa tutkimuksessa tutkimuksen eettisyyskysymykset liittyvät Tuomen ja Sarajärven (2018) mukaan tutkimusaiheen valintaan sekä itse tutkimustoimintaan, kuten tutkittavien informoimiseen, aineiston keräämiseen ja analyysi-

sin menetelmien luotettavuuteen, anonymiteettisuojaan ja tulosten esittämistapaan. Lisäksi he esittävät, että tutkimuksen uskottavuus edellyttää hyvien tieteellisten käytäntöjen noudattamista ja tutkijalta eettisiä ratkaisuja. Eskola ja Suoranta (2014) tarkentavat, että uskottavuus viittaa siihen, vastaavatko tutkijan tekemät tulkinnat tutkittavien käsityksiä.

Tutkimuksen tekoa varten pyydettiin tarvittavat luvat ja tutkittavia informoitiin tutkimuksesta riittävästi. Ennen tutkimuksen aloittamista pyydettiin tutkimuslupa paikkakuntien koulutoimenjohtajilta ja koulujen rehtoreilta. Tutkittaville ilmoitettiin jo kirjallisessa haastattelupyynnössä, että tutkimukselle on saatu tutkimuslupa, ja että tutkimuksessa ei tulla mainitsemaan tutkittavien tai koulujen nimiä. Lisäksi tutkittaville kerrottiin, mikä tutkimuksen tarkoitus on, miten tutkimus toteutetaan ja miten aineistoa käsitellään sekä kenellä on pääsy käsiksi aineistoon, ja tutkijana annoin lupaukseni käsitellä aineistoa luottamuksellisesti ja tutkimuseettisesti. Tutkimukseen osallistuvilta opettajilta kerättiin myös kirjallinen suostumus tutkimukseen osallistumiseen. Haastateltavien annettiin valita työpaikaltaan mieluisa paikka haastattelujen toteuttamiselle, ja heiltä pyydettiin suostumus haastattelujen nauhoittamiseen.

Tutkimuksen teossa ja tietojen käsittelyssä ja säilyttämisessä keskeistä on luottamuksellisuus ja anonymiteettisuoja (ks. Eskola & Suoranta 2014). Tutkimukseen osallistuneiden tunnistettavuuden estäminen kuuluu tutkimuseettisiin normeihin (Kuula 2015), joten haastateltavien nimettömyys eli anonymiteetti taataankin tässä tutkimuksessa (ks. Eskola & Suoranta 2014; Tuomi & Sarajarvi 2018). Aineistosta tulee poistaa tutkittavien henkilötiedot mahdollisimman varhaisessa vaiheessa (Kuula 2015). Tämän tutkimuksen aineistosta on poistettu jo litterointivaiheessa haastateltavien arkaluontoiset henkilötiedot, kuten henkilöiden, koulujen ja paikkakuntien nimet. Säilytetty on ainoastaan tutkimuksen kannalta oleelliset taustatiedot haastateltavista, kuten tieto iästä, sukupuolesta, työkokemuksen määrästä sekä koulutuksesta ja erikoistumisaineista. Tutkimukseen osallistuvien henkilöllisyyden suojaamiseksi heidän nimensä korvattiin peitenimillä (Emma, Heidi, Marianna, Minna, Teemu, Terhi). Se on Kuulan (2015) mukaan parempi vaihtoehto kuin erisnimien korvaaminen esimerkiksi kirjainmerkillä, sillä nimillä ihmisistä puhuminen tekee aineistosta

ymmärrettävämmän ja koherenttimman. Myöskään tutkittavien kouluja tai paikkakuntia ei esitellä tässä tutkimuksessa nimillä.

Koska tutkijalla on vastuu säilyttää aineisto ja haastattelumuistiinpanot huolellisesti (Patton 2015), tämän tutkimuksen haastatteluaineisto säilytettiin tutkijan omalla tietokoneella salasanojen takana ja haastattelumuistiinpanot lukkojen takana. Aineisto ja muistiinpanot hävitetään asianmukaisesti tutkimuksen valmistumisen jälkeen, eikä niitä luovuteta ulkopuolisille missään vaiheessa. Näin tutkimukseen osallistuneille tehdyt lupaukset aineiston käsittelystä ja anonymiteetista säilyvät.

5.5.2 Luotettavuustekijät

Tutkimuksen luotettavuutta arvioidessa kiinnitetään huomiota muun muassa siihen, miksi tutkimuksen aihe on valittu ja mikä on tutkimuksen tarkoitus, miten aineiston keruu on tapahtunut, millä perusteella tutkimuksen osallistujat on valittu ja miten tutkimus on analysoitu ja raportoitu (Tuomi & Sarajärvi 2018). Tutkimuksessa tavoitellaan objektiivisuutta, mutta subjektiivisuus on silti aina läsnä. Tuomi ja Sarajärvi (2018) nostavat esiin, että tutkijan ollessa tutkimusasetelman luoja ja aineiston tulkitsija, tutkijan persoona vaikuttaa väistämättä jollain tasolla hänen tekemiinsä tulkintoihin. Tässä tutkimuksessa objektiivisuus näkyy siinä, että tutkijana pyrin tarkastelemaan aineistoa puolueettomasti ilman, että omat arvoni tai asenteeni näkyvät tutkimustuloksissa ja tulkinnoissa (ks. Tracy 2013). Eskola ja Suoranta (2014) esittävät, että objektiivisuus syntyy subjektiivisuuden tiedostamisesta. Tutkijan ennakko-oletusten huomioiminen lisää heidän mukaansa tutkimuksen varmuutta. Vaikka minulla oli tutkijana joitain ennakkokäsityksiä aiheesta, olin niistä tietoinen ja pyrin objektiivisyyteen tutkimuksen kaikissa vaiheissa.

Lisäksi laadullinen tutkimus on Eskolan ja Suorannan (2014) mukaan hypoteesitonta eli tutkijalla ei tule olla lukkoon lyötyjä ja tutkimusta rajoittavia ennakko-oletuksia tutkimuksen kohteesta tai tuloksista. Tutkijana minulla oli joitain ennakko-odotuksia tutkimuksen tuloksista aikaisempien kokemusteni ja havaintojeni perusteella, mitä ei pystytä välttämään (ks. Eskola & Suoranta 2014), mutta ne eivät rajanneet tutkimuksen toteuttamista.

Alasuutari (2011) esittää, että tutkimustulosten validiteetti ja reliabiliteetti määräytyvät sen perusteella, miten koherentteja aineistosta tehdyt selitykset ovat, miten monet johtolangat aineistossa puhuvat niiden puolesta ja miten relevantteja ne ovat selitettäessä muutakin kuin tutkittavaa aineistoa. Validiteetti kuvaa Metsämuurosen (2009) mukaan tutkimuksen luotettavuutta ja yleistettävyyttä. Tämä tutkimus ei laadullisen luonteensa ja pienen otoksensa vuoksi pyri yleistykseen. Tuomen ja Sarajärven (2018) mukaan validiteetissa on myös kyse siitä, onko tutkimuksessa tutkittu sitä, mitä on tarkoitus tutkia. Tutkimusta voidaan pitää luotettavana, koska tutkimus kohdistui siihen, mihin oli tarkoitus. Tutkimuksen validiteettia määrittävät myös tutkijan tekemät tulkinnat. Olennaista on, vastaavatko tehdyt tulkinnat ja johtopäätökset aineistoa (Eskola & Suoranta 2014), ja tutkijan tuleekin pystyä perustelemaan tekemänsä tulkinnat (Tuomi & Sarajärvi 2018). Tässä tutkimuksessa lukijalle on lisätty nähtäväksi olennaisimpia aineistokatkelmia. Aineistosta tehdyt tulkinnat perustellaan aineistoviittauksilla eli sitaateilla. Näin on pyritty varmistamaan aineiston ja tulkintojen yhdenmukaisuus.

Reliabiliteetti tarkoittaa tutkijan ja tutkimusmenetelmien pysyvyyttä ja johdonmukaisuutta (Tracy 2013) ja se viittaa tutkimustulosten toistettavuuteen (Tuomi & Sarajärvi 2018). Tämä tutkimus ei ole toistettavissa täysin samankaltaisena eri tutkimuskontekstissa, mikä määrittää tutkimuksen reliabiliteettia. Toisesta kohteesta olisi saatettu saada erilaista tietoa, ja eri tutkija olisi saattanut analysoida sen eri tavoin ja tehden eri asioita painottavia johtopäätöksiä. Kuitenkin tutkimus on toteutettu johdonmukaisesti.

Tulee myös huomioida tutkimuksen vahvistettavuus. Vahvistettavuus viittaa siihen, että tutkijan tekemät tulkinnat saavat vahvistusta muista vastaavaa ilmiötä käsitelleistä tutkimuksista (Eskola & Suoranta 2014). Tässä tutkimuksessa vahvistettavuutta luo se, että pohdinnassa keskustellaan saaduista tutkimustuloksista toisten samaa ilmiötä tarkastelleiden tutkimusten kanssa ja luodaan yhteyksiä näiden välille kirjallisuuden avulla.

Lisäksi on hyvä tarkastella tutkimuksessa käytettyyn aineistonkeruumenetelmään, haastatteluun, liittyviä rajoitteita. Haastattelun käyttäminen aineistonkeruumenetelmänä mahdollisti muun muassa haastattelussa saatujen vasta-

usten tarkentamisen, jolloin saatiin syvällisempiä vastauksia. Toisaalta haastateltavat saattoivat antaa sosiaalisesti hyväksyttäviä vastauksia aitojen ajatustensa ja kokemustensa esiin tuomisen sijaan (ks. Hirsjärvi ym. 2016). Haastattelukysymyksissä esiintyi jonkin verran toistoa, mutta toisaalta tällöin vastaukset syvenivät. Eskola ja Suorantakin (2014) esittävätkin, että saman asian kysyminen eri tavoin varmistaa vastausten tarkkuuden. Lisäksi testaamalla haastattelukysymykset pilottihaastatteluilla ennen varsinaisia haastatteluja pystyttiin tarkastelemaan kysymysten sopivuutta tutkimusongelman selvittämiseen ja testaamaan tutkimuksen validiteettia. Tulosten luotettavuutta saattaa kuitenkin heikentää se, että kaksi haastateltavaa ei kokenut voivansa vastata yhteen haastattelukysymykseen ohjelmoinnin opettamisen kokemuksista puutteellisen ohjelmoinninopetuskokemuksen vuoksi.

Luotettavuutta lisääväksi tekijäksi voidaan ajatella myös se, että tutkija ja tutkittavat eivät tunne toisiaan entuudestaan (ks. Eskola & Suoranta 2014). Tutkimukseen osallistuvat opettajat valikoituivat satunnaisella otannalla sen mukaan, ketkä vastasivat myöntävästi haastattelupyyntöön, mikä lisää tulosten luotettavuutta (ks. Patton 2015). Riittävän osallistujamäärän löytäminen oli haastavaa, joten jouduin pyytämään erästä minulle entuudestaan tuttua henkilöä osallistumaan tutkimukseen. Se, että tunsin yhden tutkimukseen osallistujista, on saattanut vaikuttaa tuloksiin. Lisäksi pilottihaastattelut tehtiin kahdelle tutkijalle entuudestaan tutulle luokanopettajalle yhteydenottamisen helppouden vuoksi, vaikka esihaastatteluissa yleisesti kehoitetaan käyttämään muita kuin tuttuja (ks. Eskola & Suoranta 2014).

Lopuksi tulee huomioida tarkastelun kohteena olevan aiheen vaikutus tuloksiin. Tutkimuksen aihe on saattanut rajata tutkimukseen osallistujia, sillä siitä on voinut saada sen käsityksen, että tutkimukseen osallistuminen edellyttäisi tietämystä ohjelmoinnista. Tutkimusaihe sekä haastattelukysymysten asetelu saattoivat myös osin ohjata vastaamaan ohjelmointimyönteisesti. Tuloksia voidaan kuitenkin tässä suhteessa pitää luotettavina, sillä opettajien ilmaisuissa esiintyi sekä myönteistä että myös jokseenkin kielteisempää suhtautumista ohjelmoinnin opetusta kohtaan.

6 TULOKSET

6.1 Opettajien käsitykset ohjelmoinnin opettamisesta

6.1.1 Ohjelmoinnin muodot ja ohjelmoinnin opetuksen asema

Ohjelmointi kahdentyyppisenä toimintana. Opettajat kuvailivat ohjelmoinnin voivan olla kahdentyyppistä toimintaa. Toisaalta ohjelmointi voi tarkoittaa ohjelmoimista tv-t-laitteella, toisaalta esimerkiksi ohjelmoinnillisia aktiviteetteja tai leikkejä, joissa ei käytetä tv-t-laitteita. Opettajat luonnehtivat, että myös ohjelmoinnin opetukseen sisältyy heidän näkemystensä mukaan ohjelmointi laitteilla ja ilman niitä. Luokassa voidaan esimerkiksi ohjelmoida robottia tai käyttää sovelluksia esimerkiksi tablet-laitteilla, mutta toisaalta ohjelmoinnillisia piirteitä ilmenee myös monenlaisessa leikkisessä ja pelillisessä toiminnassa.

Ohjelmointi toimintaohjeina. Opettajien käsitysten mukaan ohjelmointi on toimintaohjeiden tai käskyjen antamista ja noudattamista. Haastateltavat toivat esiin, että toimintaohjeita voidaan antaa tietokoneelle, robotille tai ihmiselle. Toisaalta eräs haastateltavista, jolla ei omien sanojensa mukaan ollut kokemusta ohjelmoinnin opetuksesta, kuvaili mielikuvansa koodaamisesta olevan pelkkää ”tietokoneella näpräämistä”. Opettajien vastauksista käy ilmi, että osalla heistä on selkeä käsitys siitä, mitä ohjelmointi on, mutta osalla ei.

Ohjelmoinnista tulee ekana mieleen et se on jotain hirveen hienoo tietokoneen dataa, että tehään jotain vaikeita, mutkikkaita kaavoja ja laskuja ja lauseita ja... tiiätkö, tulee heti mieleen se, ku oli niitä vanhoja niitä tietokoneita et oli musta tausta ja siin menee valkosella hyppii niitä rivejä, ihan ihmeellistä salakieltä. Ni ensimmäinen mielikuva on se.
(Marianna)

Ohjelmointi opetussuunnitelmallisena uudelleen nimeämisenä. Opettajilla oli käsitys, että opetussuunnitelmassa on nimetty jotkut asiat uudelleen, muun muassa ohjelmointi. Ohjelmointia on opettajien mielestä sisältynyt kouluopetukseen aiemminkin, ja osa ajatteli ohjelmointia olevan kouluopetuksessa aina jossain muodossa. Tämä kertoo siitä, että ohjelmointi ei ole uusi asia koulun arjessa; se on vain vihdoin nimetty opetussuunnitelmaan omana käsitteenään. Opettajat luonnehtivat, ettei uusi opetussuunnitelma muuttanut opetusta mer-

kittävästi. Ohjelmoinnin opetuksen sisällyttäminen opetussuunnitelmaan ei välttämättä muutakaan opetusta merkittävästi, jos ohjelmoinnin tapaista toimintaa on tehty koulussa aiemminkin.

– – Sitä nyt on aikaisemminkin tavallaan ollu sitä ohjelmointia, mut nyt se on ohjelmoinnin nimellä ja sitten on ihan osa-alue matematiikkaankin sisällytetty. – – (Heidi)

Ohjelmoinnin opetuksen merkitys. Suurin osa haastateltavista piti ohjelmoinnin opetusta alakoulussa tärkeänä. Heistä osan mukaan ohjelmointiosaamista tarvitaan, koska nykyisessä tietoyhteiskunnassa tarvitaan tällaista osaamista, ja se on hyödyllistä myös arkielämän ja työllistymisen kannalta. On myös tärkeää ymmärtää teknologisten laitteiden toimintaa. Lisäksi ohjelmointiin keskeisesti liittyvän ongelmanratkaisutaidon ajateltiin olevan keskeinen asia elämässä. Esitettiin myös, että jo pelkkä ohjelmoinnin sisältyminen opetussuunnitelmaan itsessään tekee siitä tärkeän.

Toisaalta tuotiin myös esiin, että ohjelmoinnin opettaminen ei ole alakoulussa prioriteetti, vaan ensisijaista on muiden perustaitojen opettaminen. Ohjelmointitaitoa tärkeämpänä nähtiin esimerkiksi keskittymisen ja vuorovaikutustaitojen kehittäminen.

Me eletään tietoyhteiskunnassa, joten on järkevää, että oppilaatkin oppivat ymmärtämään, miten koneet toimivat. Ihan kahvinkeitinkin tarvitsee sen, että siellä laitetaan nappi päälle, mut siellä tapahtuu jotain, et se kahvi rupee keittymään. Eli ymmärretään sitä, että ohjelmoinnin perusasia on se, lyhyet ja selkeät käskyt, joita ei voi tulkita väärin. Se on jotenkin myös täällä koulussa. Sen opettaminen on helppoa ja tärkeää, koska sitä tarvitaan kaikessa. Ei tarvii aina aatella, että pitää käyttää tietokonetta siihen ohjelmointiin vaan se voi olla, että ohjelmoidaan itse itsensä toimimaan jollain sovitulla tavalla. (Minna)

Kauheeta sanoo näin, mutta mun mielestä [ohjelmoinnin opettaminen] ei [ole tärkeää]. Koska on niin paljon tärkeämpiäkin asioita, mikä nykylapsilla on niinkun... sanotaanko, ei nyt voi sanoa pielessä, mutta missä ne kaipaa treeniä. Ihan tällaiset perustaidot, niinkun keskittyminen ja tämmönen sitoutuminen siihen omaan työhön ja ryhmä...kaveritaidot, tällaiset. Niin mun mielestä se koodaus on aika vähäarvoista verrattuna tämmösiin elämisen taitoihin. Toki sitten, jos mä tietäisin, mikä se pointti sillä koodaamisella oikeesti on, et mä sen itse ymmärtäisin niin ehkä mä ajattelisin toisin. Mutta koska mä en itse ymmärrä, niin mä en sitä arvotakaan kauheen korkeelle. (Marianna)

Ohjelmoinnin opetuksen keskeneräisyys. Opettajilla oli käsitys, että ohjelmoinnin opetuksen sisällyttäminen alakouluopetukseen on vielä ”keskeneräinen” asia. Heidän vastauksistaan ilmeni, että koska ohjelmoinnin opetus on kuitenkin vielä niin tuore ilmiö alakoulussa, ovat opetuksen tavoitteiden määrittely ja koulujen puitteiden valmistelu, kuten esimerkiksi laitteiden hankinta

kouluille, vielä kesken opettajien näkökulmasta. Eräs haastateltavista ilmaisi, että ohjelmoinnin opetus on hänelle vielä varsin epäselvä asia. Hän korosti, että ohjelmoinnin määritelmää tulisi täsmentää ja ohjelmoinnin opetuksen tarkoitusta avata tarkemmin, sillä ne eivät ole aina selviä opettajille ja oppilaille. Koodaushan ei ole pelkkää "tietokonenippeliä", vaan paljon muutakin. Opettajien vastauksista voi päätellä, että ohjelmoinnin opetus ei ole näyttäytynyt opettajille vielä täysin selvänä asiana, vaan he kaipaivat jonkinlaista lisäohjeistusta tai -koulutusta opetuksen toteuttamisesta. Vastaukset viestivät siitä, että sekä opettajille että oppilaille kaivattaisiin lisää tietoutta aiheesta.

Mut kyl mä edelleen oon sitä mieltä, että se pitää tarkemmin avata, mikä on sen [ohjelmoinnin opetuksen pointti]... Ja täsmentää sitä, että se koodaaminen ei oo sitä tietokonenippeliä pelkästään. Ehkä se vaatii sitä, että opettajat ja lapsetkin ymmärtää sen, että mitä kaikkea se koodaaminen voi olla. Ja että siinä nähdään se joku järki. Eihän kukaan halua opetella eikä opettaa mitään, missä ei itse näe olevan mitään järkeä. (Marianna)

6.1.2 Ohjelmoinnin rooli monien taitojen opettajana

Ohjelmointi välineellisenä taitona. Lähes kaikilla haastateltavilla oli käsitys, että ohjelmoinnin opetus tukee varsinaisen teknisen ohjelmointiosaamisen lisäksi myös muiden taitojen kehittymistä. Teknisen ohjelmointitaidon oppiminen ei myöskään ollut opettajien käsitysten mukaan ohjelmoinnin opetuksen oleellisin tavoite, vaan se nähtiin hyvänä lisänä muiden taitojen oppimisen päälle.

Ohjelmointi matemaattisteknologisten taitojen tukijana. Opettajien näkemysten mukaan oppilaat voivat oppia ohjelmoinnin myötä teknisen ohjelmointitaidon lisäksi muun muassa ymmärrystä ohjelmoinnin perusajatuksista, tietokoneen logiikasta sekä ihmisen ja koneen välisestä erosta. Vastauksissa painottui myös erilaisten matemaattisten taitojen kehittyminen. Ohjelmoinnin nähtiin opettavan matemaattista ajattelua, hahmottamista, loogista päättelyä ja ongelmanratkaisutaitoa sekä mahdollistavan esimerkiksi matematiikan tehtävien tai erilaisten toimintojen palastelun pienempiin osiin.

Ohjelmointi arjen taitojen tukijana. Ohjelmoinnista nähtiin olevan hyötyä myös arjen oleellisten perustaitojen kehittymiselle. Sen kuvailtiin harjaannuttavan esimerkiksi luetun ymmärtämistä, järjestelmällisyyttä, ohjeen noudat-

tamista, keskittymistä, sinnikkyyttä, kriittisyyttä sekä syy-seuraus-suhteen ymmärrystä.

– – He [oppilaat] oppii ymmärtämään tämän tietokoneen sitä logiikkaa. Sit he oppii järjestelmällisyyttä, täs ei voida oikoa. Kynän kanssa voidaan vielä oikasta jotenkin, mut täs se ei toimi, jos sä yrität oikaista. Sitte mä sanoisin, että sinnikkyyttä. Näähän on sellasia taitoja, ei siis suoraan... Totta kai ne on oppinu ohjelmoimaan, totta kai se on tietysti se ihan perus. Mut sitte muita asioita, niitäkin on. (Emma)

Juurikin sitä ongelmanratkaisua [oppii]. Vähän ajattelemaan eri näkökulmista, miten sitä vois ratkaista tai onko eri ratkaisuvaihtoehtoja. Kyllä siinä varmaan voi oppia monia muitakin taitoja. Keskittymistä ja pitkäjänteisyyttä. Ja tällaisia taitoja mitä nyt muutenkin harjoitellaan erilaisissa jutuissa koulussa. (Heidi)

6.2 Opettajien kokemukset ohjelmoinnin opettamisesta

6.2.1 Ohjelmoinnin opetuksen edellytykset

Välineet ja aika resursseina. Opettajien vastauksista ilmeni, että ohjelmoinnin opetus edellyttää erilaisia resursseja ja valmiuksia sekä koululta että opettajalta. Opettajien mukaan ohjelmoinnin opetus edellyttää ensinnäkin koululta laitteisto-resursseja eli tarpeeksi välineistöä. Osa opettajista kertoi, että koululta löytyy varsin hyvin välineitä ohjelmoinnin opetukseen, joskin osan mielestä laitteita ei ole riittävästi. Raha nähtiin ratkaisevana tekijänä koulujen mahdollisuuksissa hankkia lisää välineitä ohjelmoinnin opetukseen. Eräs haastateltava toi esiin, että laitteiden hankinta kouluille ei tapahdu hetkessä, eikä heidänkään koulussaan ”seittemää robottia kerralla osteta”. Tuotiin myös esiin, että on tärkeää, että laitteet toimivat. Eräs haastateltava nosti esiin myös koulun fyysisten tilojen merkityksen: liian pieni luokkatila vaikeuttaa opetusta.

– – No toki sitte niitä välineitä. Taas esimerkiks ne Bee-Botit. Jos niitä ei oo, ni mistäs niitä sit keksit. Koska mun mielestä siitä menee vähän se pointti, jos siinä ei ois mitään ohjelmitavaa juttua, että käyt siirtämässä vaikka jonkun nallen istumaan siihen, niin ei siinä näkään oo sitte mitään järkeä. – – Mutta aikaa, vaivaa, rahaa. Kaikki vaatii aina rahaa. Välineitä. (Marianna)

Lisäksi ohjelmoinnin opetus vaatii opettajien mukaan aikaresursseja. Opettaja tarvitsee aikaa sekä opetuksen suunnitteluun että ohjelmoinnin opettamiseen. Etukäteissuunnittelu on tärkeää, jotta ohjelmoinnin opetus olisi sujuvaa eikä jäisi irralliseksi puuhasteluksi. Haastateltavien vastauksissa korostui ajan puute, joka rajoitti erityisesti opettajien omaehtoista perehtymistä ohjelmointiin, tosin joistain vastauksista voi tulkita, että aikaa myös ohjelmoinnin opetukselle

koulun lukujärjestykseen kaivattaisiin lisää. Tuotiin esiin, että jos aikaa olisi enemmän, ohjelmointia voitaisiin opettaa syvällisemmin. Eräs haastateltava ilmaisi, että tulisi olla aikaa opettaa ohjelmointia pidemmissä jaksoissa, jottei opetus jää pelkäksi ”närppimiseksi”. Ohjelmoinnin opetukseen ei kuitenkaan voida käyttää liikaa aikaa, sillä on paljon muitakin oleellisia opetettavia asioita, ja ohjelmoinnin opetus ei ole ”ykkösprioriteetti” alakoulussa.

– – No siihen ei ole aikaa kyllä, siihen omaehtoiseen perehtymiseen. Mut opettamiseen on. Itehan me suunnitellaan kuitenkin se. Se on aina se, et mikä nähdään tärkeimpänä. (Emma)

Aikaresurssit. Riippuu miten sitä ohjelmointia opettaa. Yleensä laitteet vie enemmän aikaa kuin että leikitään ja tehdään, otetaan kirja esille tai rakennetaan palikoista jotain. Riippuu, miten sen suunnittelee totta kai. Mut ehkä se suunnittelun aikaresurssi, sitä se vaatii. Et mitä sujuvampaa, sitä enemmän menee aikaa siihen suunnitteluun opettajalla. (Heidi)

Asenteet ja taidot resursseina. Opettajat kuvailivat ohjelmoinnin opetuksen edellyttävän opettajalta ohjelmointiosaamista sekä myönteistä asennetta. Opettajilla näyttää olevan myönteinen asenne ohjelmoinnin opettamista kohtaan, mutta osalla on riittämättömät taidot opetuksen toteuttamiseen. Suurin osa haastateltavista painotti myönteistä asennettaan ohjelmoinnin opettamista kohtaan. Vaikka ohjelmoinnin alkeiden tai ”palikkakoodamisen” opettamiseen alakoulussa nähtiin riittävän ohjelmoinnin perusosaaminen, vain osa haastateltavista kuvaili omaavansa valmiuksia opettaa ohjelmointia. Hekin luonnehtivat omaavansa riittävät taidot vain tämän hetken tarpeisiin ohjelmoinnin alkeiden opettamiseen, ja tulevaisuus huolestutti, sillä tulevaisuudessa heidän taitonsa eivät välttämättä enää riitä ohjelmoinnin opetuksen sisältöjen muuttuessa haastavammiksi. Vaikka joidenkin opettajien ohjelmointitaidot olivat hyvin heikot, kaikki kuvasivat yleisiä tv-taitojaan hyväksi tai vähintäänkin välttäviksi.

– – Mä uskon, et muutaman vuoden päästä jokaisen niinku perustaso alkaa olemaan niin hyvä, et sit on pakko koulussaki huomioida se, että ei niinku ihan näillä alkeilla tai perusteilla tarvii enää pelailla niin paljon. Voidaan heti siirtyä monimutkaisiin, tai ei monimutkaisiin mut enemmänkin siihen et ollaan sen ohjelmointikielen kans tekemisissä ja luodaan selkeesti jotain, enemmän. (Teemu)

6.2.2 Ohjelmoinnin opetuksen toteuttaminen

Opettajat kertoivat opettavansa ohjelmointia sekä tv-laitteilla kuten tietokoneilla ja tablet-laitteilla erilaisten sovellusten ja verkkosivustojen avulla sekä ilman

laitteita pelien ja leikkien avulla. Opettajat luettelivat useita esimerkkejä, miten ovat opettaneet ohjelmointia (ks. taulukko 5). Haastateltavien vastaukset kertovat siitä, että ohjelmoinnin opetuksessa on mahdollista hyödyntää lukuisia eri työtapoja ja opetusmateriaaleja, ja ohjelmointia voidaan opettaa sekä ei-teknologisin että teknologisin opetusmenetelmin.

TAULUKKO 5. Haastateltujen opettajien ohjelmoinnin opetuksessa käyttämät opetusmenetelmät

Ei-teknologiset opetusmenetelmät	Koodauspläjäys-kortit
	Koordinaatistotehtävät
	Leikit
	Matematiikan oppikirjan tehtävät
	Pelit
	Pyyhittävät koodauspuuhat -puuhakirja
	Robogem-lautapeli
	Salakirjoitus (ohjelmointikielen luominen)
	Toimintaohjeiden antaminen toiselle
Teknologiset opetusmenetelmät	Code.org
	Code.org - Koodaustunti
	Lego Mindstorms -robotit
	Scratch

Ei-teknologisten opetusmenetelmien toiminnallisuus. Haastateltavat kuvailivat toiminnallisuuden olevan tärkeää ohjelmoinnin alkeiden opetuksessa. Ohjelmointia voidaan heidän mukaansa opettaa monin tavoin ilman teknologiaa. Opettajat hyödynsivät opetuksessa sekä valmiita materiaaleja että itse kehitettyjä aktiviteetteja. Vastaukset kertovat opettajien innovatiivisuudesta ja panostamisesta opetuksen monipuolisuuteen ja innostavuuteen, toisaalta myös mahdollisesti valmiiden materiaalien riittämättömyydestä.

Haastateltavien vastauksista käy ilmi, että he lähtevät liikkeelle ohjelmoinnin opetuksessa aluksi toiminnallisuuden ja leikinomaisuuden kautta, ja tvt tuodaan mukaan vasta myöhemmin. Useimmin opettajat näyttävät hyödyn-

tävän itse laadittuja harjoituksia ja aktiviteetteja; valmiiden materiaalien, kuten lautapeliin tai puuhakorttien, käyttäminen oli vähäisempää. Opettajat kertovat käyttävänsä opetuksessa esimerkiksi harjoitusta, jossa luokkatoveri tai opettaja ohjelmoidaan liikkumaan toimintakäskyjä antamalla suullisesti, kirjoittaen tai piirtäen. Opetuksessa käytetään myös aktiviteetteja, joissa liikutaan koordinaatistossa ohjeiden mukaan, kehitetään oma ohjelmointikieli kirjoittaen tai piirtäen salakieltä, tai luodaan oma peli. Samaten on myös kokeiltu toiminnallisesti ja pelillisesti ruudukossa tapahtuvaa laivanupotus-peliä, miinakentän ylitystä tai timanttien etsimistä labyrintistä. Lisäksi ohjelmoinnin opetuksessa hyödynnetään muun muassa lautapelejä, kortteja ja puuhakirjoja, ja ohjelmointitehtäviä tehdään myös matematiikan oppikirjasta.

Opettajat perustelivat tiettyjen työtapojen valintaa siten, että niiden avulla pystyy lisäämään opetukseen toiminnallisuutta ja havainnollistamista. Tuotiin esiin, että toiminnallisuus esimerkiksi luokkatoverin ohjelmoimisessa liikkumaan auttaa ymmärtämään ohjelmointia, ja itse liikkuen opitaan saaden ”koko kehon muistijälkiä” opittavista asioista.

– – me lähetään liikenteeseen aina, ohjelmoidaan ensin opettajaa. Ohjelmoidaan opettajaa ja sen jälkeen ohjelmoidaan toisiamme ja usein käytän sellasta ruudukkoharjoitusta, jossa on erilaisia aarteita, mitä metsästetään ja parin pitää tehdä sitte se koodi, et mihin se kaveri liikkuu näitten korttien avulla. Tääl on näitä tällasia [suunta-]kortteja. Se on semmonen perusharjoitus, että mistä lähetään liikkeelle. Sit mä käytän Robogemia. Meil on Koodauspläjäys-kortit, niitä käytetään. Ja sitte viel semmonen yks kirjanen, se on tuol pöydällä. Pyyhittävät koodauspuuhat. Tavallaan se, et on ensin paljon materiaalia, ennen ku mennään koneelle. – – (Emma)

Toiminnallisuus on mun mielestä kaikista tärkein siihen alkuun. Silloin ne oikeesti lapset hoksaa sen ja se lähtee ihan siitä ensimmäisestä harjoituksesta. Mä yleensä lähen siitä, et mä istun tossa jakkaralla ja lasten pitää ohjelmoida mut pesemään kädet. Sit ku mähän toimin todella kuin robotti, et mä teen juuri niinkun ne käskee ja jos se komento on semmonen, et mä en ymmärrä, ei oo tarpeeks yksinkertanen ni sit menee error-tilaan. Se on hauskaa. (Emma)

Koen, että lapset oppivat tekemällä ja kokemalla ja sen takia haluan laittaa ne liikkumaan. Tulee koko kehon muistijälkiä asioista, ei vain kynä ja paperi. (Minna)

Teknologisten opetusmenetelmien monipuolisuus. Opettajien kertoman mukaan alun toiminnallisten aktiviteettien jälkeen siirrytään digitaalisten välineiden pariin oppimaan ohjelmointia. Digitaalisista opetusvälineistä vastauksissa painottuivat Scratch-sovellus ja Code.org -verkkosivusto sekä sen alisivusto Koodaustunti. Opetuksessa hyödynnettiin myös Lego Mindstorms -robotteja.

Opettajat korostivat teknologisten opetusvälineiden ja resurssien etuna niiden monipuolisia ominaisuuksia.

Haastateltavat puhuivat käyttämiensä ohjelmien puolesta muun muassa niiden eriyttämismahdollisuuksien vuoksi. Code.org-sivuston tehtäviä esimerkiksi kuvailtiin hyväksi, sillä niistä löytyy eritasoisille oppilaille sopivia tehtäviä. Tärkeänä kuvailtiin onnistumisen elämyksien tuottaminen kaiktasoisille oppijoille. Code.orgin hyvänä ominaisuutena pidettiin myös sitä, että opettaja pystyy seuraamaan oppilaiden edistymistä.

Lisäksi sovellusten valinnassa kiinnitettiin huomiota niiden soveltuvuuteen muihin oppiaineisiin integrointiin. Sovellusten ja sivustojen käyttöä perusteltiin myös niiden suomenkielisyydellä ja sillä, että ne ovat käyttövalmiita ohjelmia.

– – Ku lähin ite tutkimaan sitä Koodaustunti-sivua esimerkiksi, siel oli tosi selkeesti, että lähti helpommasta ja meni kohti vaikeempaa ja oikeesti eriyttää. Helpot on helppoja ja niissä pärjää kaikki, kun sen hoksaa. Mut haastavat rupee olee nii haastavia, et oikeesti ne terävätki joutuu miettimään tosi paljon, et ne menee läpi. Et siihen tuli eriyttäminen vahvasti. Kaikki saa onnistumisen elämyksiä. Koodaustunnissa oli myös kiva se, että siellä oli vaikka Frozenista ja Angry Birdsista, tämmösiä tuttuja lapsille ne pelien hahmot, ni ne innostu siitä menemään kenttiä läpi. Sitte Scratchi taas, se oli semmonen ohjelma, mikä näissä koulujen padeissa on, ja sitten taas siellä pääsee luomaan omia hahmoja, paikkoja, tarinoita. Just siihen äidinkieleen integrointiin se on semmonen kiva animaationtuottamisväline. (Minna)

Se on tosi hyvä, että on tehty tällaisia valmiita ja suomenkielisiä ohjelmointiin ohjaavia pelejä. Ne on tullu vastaan ja oon kokenu, että ne on ollu kiinnostavia ja ihan hyviä. Pikkuhiljaahan se kehittyy tässä, että miten sitä sitten enemmänkin opettaa. (Heidi)

Opettajat kuvailivat ohjelmoinnin opetuksessa käyttämiensä ohjelmien myös lisäävän itseilmaisun mahdollisuuksia ja olevan yleisesti oppilaita innostavia. Scratchin puolesta puhuttiin esimerkiksi sen rakentamismahdollisuuden ansiosta; oppilas pääsee itse luomaan hahmoja ja taustakuvia projekteihin. Code.orgin Koodaustuntia taas kehuttiin oppilaille tuttujien ja heitä innostavien tehtävien ja alustojen (esim. Angry Birds, Frozen, Minecraft) ansiosta.

Toisinaan ohjelmat näyttivät opettajien vastausten perusteella valikoituvan käyttöön niiden joko niiden tuttuuden ja helppokäyttöisyyden ansiosta tai sattumalta. Tiettyjen ohjelmien käyttöä perusteltiin sillä, että opettaja osaa itse käyttää niitä ja siten myös neuvoa oppilaita niiden käytössä. Opettajat kertoivat käyttävänsä esimerkiksi Scratchiä ja Koodaustunti-sivustoa myös siksi, että ne tulivat vastaan, kun he alun perin lähtivät etsimään sopivia sovelluksia ohjel-

moinnin opetukseen. Lisäksi tuotiin esiin, että Lego-robotit on valittu joidenkin koulujen opetusvälineiksi kunnan taholta.

Varmaan ihan sen takia [käytän näitä], ku niist [Scratchistä ja Lego-roboteista] mä osaan neuvoo. Et jos mä ottasin jonku uuden, mä joutuisin ensin opettelemaan itse. Mä haluan kuitenkin jonku verran osata itse ennen ku mä... Et nää on ollu ne, mihin mä oon tutustunu. Nää Legot on valittu meidän kunnassa... Mut Scratchi ei oo mitenkää ollu, et se hyvin vois olla joku muukin ohjelmoinnin kieli, mut visuaalisen kielen kautta tällä hetkellä. Meillä yläkoulussa tulee sitten vasta kielellinen ohjelmointi. (Emma)

Ohjelmoinnin integroinnin laajat mahdollisuudet. Opettajat kuvasivat vastauksissaan ohjelmoinnin muuhun opetukseen sisällyttämisen laajoja mahdollisuuksia. Ohjelmoinnin kuvailtiin sopivan sisällytettäväksi oikeastaan mihin tahansa oppiaineeseen. Haastateltavat kertoivat integroivansa ohjelmoinnin opetusta paljon eri oppiaineisiin, ja ohjelmoinnin sisällyttäminen osaksi muuta opetusta näyttäytyi opettajien vastausten perusteella varsin helppona. Ohjelmoinnin integroinnissa muuhun opetukseen keskeisintä näyttää olevan integroinnin tarkoituksenmukaisuus ja lisäarvon tuominen opetukseen.

Ohjelmointia integroitiin eniten matematiikan ja äidinkielen opetukseen, mutta osa opettajista sisällytti sitä myös liikuntaan, ympäristöoppiin, kuvataiteeseen, yhteiskuntaoppiin sekä musiikkiin. Haastateltavat kertoivat yhdistävänsä ohjelmointia esimerkiksi liikuntaan liikunnallisen pelin luomisessa ja yhteiskuntaoppiin vaikuttaja ja vaikeuttaja -leikissä. Äidinkielessä ohjelmointi voi tarkoittaa sanallistamista ja ymmärtämistä, ja lisäksi voidaan luoda omaa salakirjoitusta tai kirjoittaa ja animoida oma tarina. Kuvataiteessa on kuvitettu ohjelmointikäskyjä, ja musiikissa käytetty GarageBandiä, jonka käytön kuvaillaan olevan myös ohjelmointia. Ainoastaan yksi haastateltava kertoi pitävänsä ohjelmoinnin opetuksen erillään muista oppiaineista.

Ohjelmoinnissakin se, et toisi muuhun asiaan [jotain lisäarvoa]. Et se ei ois päälle liimattu asia. Matikan kirjassahan siitä on jotain. Mut ne on viimeinen kappale kirjassa ja siihen ei yleensä keritä. (Minna)

Omaan luokkaan, liikunta on hyvä. – – No sä voit luoda liikkuvan pelin. Mä oon ottanu sitä noiden kans kyllä. Mä oon ottanu myös yhteiskuntaoppia. Yhteiskuntaopissa vaikuttaja ja vaikeuttaja -leikkihän on hirveen hyvä. Vaikeuttaja antaa käskyjä ja vaikeuttaa etenemistä, kun vaikuttaja taas edistää etenemistä. Toimii. Sitte matikkaan hyvin. Siinä luodaan vaikka pelejä. Erilaisilla geometrisillä muodoilla voi antaa käskyjä, tavallaan niin sitä koodikieltä pystytään luomaan. No äidinkielihän on hyvin perinteinen, koodaus, salakirjoitus aikojen alusta. En mä nyt näe, että joku oppiaine ei varsinaisesti soveltuisi siihen. Jos mä nyt luettelisin niitä, niin mä luulen, että me ollaan tehty aika monella oppiaineella. (Teemu)

– – Esimerkiksi liikunnassa meillä on ollut kanssa ohjelmointia tavallaan. Tämmönen tehtävä oli, että oli erilaisia välineitä. Oli rengas, joku hyppynaru ja joku tötsä ja niillä oli kaikilla eri tehtävä, että mitä pitää niiden kohdalla tehdä. Oppilaat teki sellasia omia ratoja ja me menttiin niitten mukaan. Vaikka tämmönen voisi olla. Ja kylhän sitä voisi musiikkiinkin varmaan yhdistää. Laittaa ohjeita nuoteista ja sitten soitetaan vaikka pianolla tai GarageBandilla. (Heidi)

Opettajien vastaukset kuitenkin paljastavat, että aikaresurssit saattavat muodostua esteeksi ohjelmoinnin sisällyttämiselle muuhun opetukseen sen monipuolisista integrointimahdollisuuksista huolimatta. Haastateltavien vastauksista ilmenee, että ohjelmointi sopii hyvin eri oppiaineiden opetukseen, mutta aikaresurssit rajoittavat sen laajaa integrointia. Vastauksissa korostuu se, että vaikka ohjelmointia voisi yhdistää moneen oppiaineeseen, aika ei riitä; ohjelmoinnin yhdistämistä eri aiheisiin ei ehditä suunnittelemaan tarpeeksi.

6.2.3 Opettajien yleisiä tunteita ohjelmoinnin opettamisesta

Ohjelmoinnin opettamisen herättämät tunteet. Opettajilla oli monenlaisia kokemuksia ohjelmoinnin opetuksen toteuttamisesta. Kokemuksia opettamisesta kuvailtiin myönteisiksi, neutraaleiksi sekä haasteellisiksi.

Ohjelmointia opettaneet opettajat kertoivat kokeneensa ohjelmoinnin opettamisen pääosin mieluisaksi. Mielekkääksi keuhuttiin esimerkiksi sitä, kun näkee sellaisten oppilaiden innostuvan koodauksesta, jotka eivät muuten yleensä innostu opetuksen sisällöistä. Osa opettajista kertoi kuitenkin alun perin kokeneensa ohjelmoinnin opetuksen aloittamisen neutraalina tai jännittävänä asiana. Esiin nostettiin alussa esiintyneitä jännityksen ja pelon tunteita liittyen omaan osaamattomuuteen tai tietämättömyyteen ohjelmoinnin saralla. Osa taas kuvaili, että ohjelmoinnin opetuksen aloittaminen ei jännittänyt, mitä perusteltiin sillä, että ohjelmointia oli tehty jo aiemminkin, ja alakoulussa ohjelmoinnin opetuksessa ”pärjää perusjutuilla”. Yllättäen kukaan ei kuitenkaan tuonut esiin selvästi erottuvia myönteisiä tunteita kuten innostusta, joskin yksi opettaja ilmaisi ohjelmoinnin olevan tervetullut lisä opetussuunnitelmaan.

Varmaan joku ’Apua! En minä tiedä siitä mitään.’ Siinä varmaan tuli ensimmäisenä se, että enhän minä tiä siitä mitään. Miten mä osaan sitä opettaa yhtään millään lailla, ku en mäköän itsekään osaa enkä tiedä siitä mitään. (Marianna)

– – Kai sitä on tavallaan mun mielestä ohjelmointia aina opetettu. Ollaanhan me opetettu lapsia ohjelmoimaan, heidät työskentelemään oikein. Ja se, että nyt otetaan lisäksi siihen ihan oikeesti tietokoneet mukaan, se ei mun mielestä oo suuri askel. Mut se varmaan on

joidenkin kohdalla semmonen peikko, et ne ajattelee, et he eivät kykene ja ymmärrä ja pysty ja pärjää. – – Tai minä oon ainakin ohjelmoinu [jo ennen uutta opetussuunnitelmaa]. Et jos mä annan lapselle toimintaohjeen, et hae kirja, istu siihen ja siihen paikalle ja ota se ja se sivu esiin, ni onhan se jo tavallaan ohjelmointia. Mä ohjelmoin sitä lasta toimimaan. (Emma)

Ohjelmoinnin opetuksen haasteellisuus. Opettajat kuvailivat ohjelmoinnin opettamista osin myös vaikeaksi. Ohjelmoinnin opetuksen ongelmakohtina opettajat nostivat esiin aikaresurssit, laitteistoresurssit ja oppilaiden taitoerot. Olennaisimpana haasteena ohjelmoinnin opetukselle haastateltavat korostivat liian vähäisiä aikaresursseja. Laitteiden käyttämisen nähtiin esimerkiksi vievän toiminnallisia ja kirjan tehtäviä enemmän aikaa. Osa opettajista haluaisi myös edetä ohjelmoinnin opetuksessa nykyistä alkeistasoa pidemmälle, mutta opetukseen käytettävissä oleva aika ei riitä syvemmälle menemiseen.

Toinen ongelmakohta olivat koulun resurssit, eli onko laitteita tarpeeksi ja toimivatko ne. Lisäksi opetuksen eriyttäminen koettiin vaikeana oppilaiden suurien taitoerojen vuoksi, osin myös johtuen opettajan omasta taitotasosta. Ohjelmoinnin opetuksen arveltiin muuttuvan entistä vaikeammaksi tulevaisuudessa edistyneempien oppilaiden kanssa. Tuotiin myös esiin, että oppilaiden motivointi saattaa joskus olla hankalaa, ja että ongelmallisia voivat olla opettajan tai oppilaiden ennakkoluulot ohjelmointia kohtaan.

Siis se on kakspiippunen juttu. Sehän on tosi mukavaa, ku sä näät, että siel innostuu sellasia lapsia, jotka ei muuten meinaa innostua. Ja sit toisaalta se, et ku jos niil on joku kysymys niillä edistyneillä ja sit menee peukalo omaan suuhun ni kylhän siin on sellasta nöyryyttä sen asian suhteen, et hei mä en osaa, mä en tiiä tähän vastausta, mietitään yhdessä. Kylhän siin joutuu niinku nöyrytymään, olemaan nöyrä. (Emma)

Nyhän tietysti, ku ollaan silleen alkuvaiheessa koko koodauksen opetuksessa siinä mielessä, että hyvin pitkälle se on ollu samansisältöistä, et sit siel on yksittäisiä oppilaita eriytetty ylös- tai alaspäin. Mut varmaan ens vuonna tilanne on jo taas toinen, ku siel on koko esimerkiksi nelos-kutosten ryhmä on sellanen, joka on jo käyny tämän, ni sit pitää löytää sieltä ne haasteet, mitä halutaan viedä eteenpäin. (Emma)

Tuen tarve ohjelmoinnin opetuksessa. Riittävän tuen ja lisäkoulutuksen tärkeys, ja toisaalta myös niiden puute ilmenee vahvasti opettajien vastauksista. Haastatelluista opettajista suurin osa kuvaili kaipaavansa tukea ohjelmoinnin opetukseen. Lisäosaamista kaivattiin ohjelmoinnin alkeiden ja ohjelmoinnin pidemmälle opettamiseen sekä ohjelmoinnin tarkoituksenmukaiseen integrointiin eri aiheisiin tai oppiaineisiin. Osa opettajista kaipasi lisäosaamista esimerkiksi robotiikan hyödyntämisestä opetuksessa, kun taas osa toi esiin, ettei koe

tarvitsevansa tukea ohjelmoinnin opetukseen ainakaan vielä toistaiseksi. Toivoimmaksi tuen muodoksi haastateltavien vastauksissa nousi mahdollisuus täydennyskoulutukseen. Lisäkouluttautumista pidettiin välttämättömänä ohjelmoinnin opetuksen jatkon kannalta. Täydennyskoulutusta ei kuitenkaan opettajien kokemusten mukaan ole tarjolla riittävästi. Toisaalta täydennyskoulutusta ohjelmoinnista ei välttämättä nähty tällä hetkellä omalla kohdalla prioriteettina.

Haastateltavat korostivat tukeutuvansa tällä hetkellä paljon internetiin tuen saamiseksi. Internetiä käytettiin omaehtoisesti ohjelmointiin perehtymiseen omalla ajalla ja ohjelmoinnin opetukseen sopivien sovellusten etsimiseen. Tukea oli saatu myös kollegoilta sekä aikaisemmista täydennyskoulutuksista.

– – Toivois ehkä että olis jonkinlaidista koulutusta enemmän koulun kautta. Että omalla ajallahan se menee aika usein. (Heidi)

– – Ja muutenhan [ilman täydennyskouluttautumista] se ois ihan mahdotonta mun mielestä opettaa, et jos vetäis ikuisesti Scratchiä. Se ois kyllä virhe. (Teemu)

6.2.4 Ohjelmointi useimpien oppilaiden innostajana

Opettajat kuvailivat oppilaiden suhtautuneen ohjelmoinnin opetukseen pääosin myönteisesti. Sekä pienemmät että isommat oppilaat ovat opettajien mukaan olleet aiheesta innostuneita ja ”tosi täpinöissään”. Ohjelmoinnin opetuksen on nähty auttaneen motivoimaan sellaisiakin oppilaita, joita koulunkäynti ei muuten olisi niin kiinnostanut. Toisaalta osan oppilaista kerrottiin suhtautuneen ohjelmoinnin opetukseen neutraalisti eli ottaneen ohjelmoinnin vain ”yhtenä asiana muiden joukossa”.

Tosi positiivisesti [oppilaat ovat suhtautuneet ohjelmointiin]. Sehän on, ne on tosi mielenkiintoisia. Ei tarvii kauheesti sysätä, et kyl ne innolla sitä tekee. Se on virtuaalimaailmassa elämistä ja liikkumista ni se ei oo tän päivän lapsille kauhee ongelma. (Teemu)

Vaikka ohjelmointi on oppilaille yleensä mielekästä, kaikki oppilaat eivät ole opettajien mukaan kokeneet sitä pelkästään myönteisesti. Esiin tuotiin, että pieni osa oppilaista on kokenut ohjelmoinnin opetuksen myös kielteisesti. Opettajien mukaan oppilaiden välillä on eroja kiinnostuksessa, taitotasossa ja laitteiden käyttöhistoriassa, ja he näkivät erityisesti taitoerojen olevan olennainen syytä joidenkin oppilaiden kielteisiin kokemuksiin ohjelmoinnista. Laitteiden-

käyttökokemuksen määrän arveltiin olevan yksi syy suuriin taitoeroihin. Ohjelmointi on opettajien mukaan saatettu kokea tylsäksi, vaikeaksi tai jopa pelottavaksi. Arveltiin, että ohjelmointi saattaa tuntua ikävältä sellaisesta oppilaasta, jolle se on vaikeaa. Osa oppilaista oli kokenut nimenomaan kokeilemisen ja virheiden tekemisen pelottavaksi, jonka eräs opettaja arveli johtuvan rohkeuden puutteesta kokeilla. Toisen opettajan mukaan oppilaiden mielestä koodaus Scratchillä oli tylsää, mutta hän uskoi, että oppilaat saattaisivat innostua erityyppisistä koodausharjoituksista, kuten robotin ohjelmoimisesta. Opettajat tuntuivat kuitenkin pitävän tavanomaisena ilmiönä koulun arjessa sitä, että oppilaiden välillä on eroja kiinnostuksessa ja osaamisessa.

Se on ihan kakspiippunen juttu. Siel on ne, jotka on superinnostuneita ja siel on ne oppilaat, joita pelottaa tavallaan sen koodin tekeminen koneelle. Jotenkin siel on myös niitä, jotka ei oo niin rohkeita kokeilemaan. Mulla siis ihan siinä valinnaisaineryhmässäkin on muutama oppilas, jotka, ne ois pitäny olla niin valmiit. Mä sanoin, et kokeile. Mut jos se menee väärin. No se ei haittaa, tehään sitte uudestaan. Tavallaan se, et ehkä me on opetettu tai tiettyjen lasten semmonen temperamentti tai luonne on semmonen, et haluis olla heti oikeessa. – – Mut se liittyy ehkä just näihin rohkeuteen toteuttaa, kokeilla. (Emma)

Haastatellut opettajat olivat yhtä mieltä siitä, että ohjelmoinnissa oppilaiden kiinnostuksessa tai taidoissa ei ole merkittävää eroa sukupuolten välillä. Tyttöjen nähtiin pitävän ohjelmoinnista siinä missä pojatkin, joskin mainittiin myös, että pojat valitsevat ohjelmoinnin koulussa valinnaisaineeksi tyttöjä useammin.

7 POHDINTA

7.1 Tulosten yhteenvetoa

Tämän tutkimuksen tarkoituksena oli selvittää, millaisia käsityksiä ja kokemuksia luokanopettajilla on ohjelmoinnin opettamisesta. Tutkimuksen aineisto koostui kuuden luokanopettajan haastatteluista, joiden analyysissä hyödynnettiin laadullista sisällönanalyysiä. Opettajien vastauksissa korostui, että ohjelmoinnin opetus tukee useiden taitojen kehittymistä ja ohjelmoinnin opetuksessa hyödynnetään monenlaisia opetusmenetelmiä, mutta tukea ja täydennyskoulusta tarvitaan ohjelmoinnin opetuksen helpottamiseksi. Ohjelmoinnin opetuksen nähdään edellyttävän valmiuksia ja resursseja sekä opettajilta että kouluilta, mutta kaikilla luokanopettajilla ei tulosten perusteella näytä olevan valmiuksia toteuttaa ohjelmoinnin opetusta.

7.1.1 Käsitykset ohjelmoinnin opetuksesta

Opettajat käsittävät ohjelmoinnin pääosin tärkeänä ja hyödyllisenä opettavana asiana, ja ainakin osalla opettajista näyttää olevan varsin selkeä käsitys siitä, mitä ohjelmointi on. Tutkimukseen osallistuneiden opettajien antamat määritelmät ovat pitkälti linjassa Hyvösen ja kumppaneiden (2013) määritelmän kanssa ohjelmoinnista toimintaohjeiden laatimisena. Lisäksi opettajat mainitsivat olennaisena ohjelmoinnissa ongelmanratkaisun, minkä osuutta ohjelmoinnissa on korostettu myös aikaisemmissa tutkimuksissa (ks. García-Peñalvo 2016b; Peltoniemi 1999; Resnick ym. 2009). Opettajat näkivät ohjelmoinnin opetuksen hyödyllisenä, yhteneväisesti muun muassa Karvosen ja Laukan (2016) tulosten kanssa. Toisaalta esiin tuotiin myös, että kaikilla opettajilla ei ole valmiuksia opettaa ohjelmointia. Osa opettajista oli opettanut ohjelmointia jo ennen uutta opetussuunnitelmaa, mutta kaikki opettajat eivät olleet kuitenkaan opettaneet ohjelmointia lainkaan. Tämä kertoo siitä, että ohjelmointi ei ole uusi asia kouluopetuksessa, mutta joillekin opettajille se on varsin uusi asia. Tulokset osoittavat, että ohjelmoinnin opetus jää luokanopettajista

vain osan vastuulle. Siinä missä osa opettajista opettaa ohjelmointia paljon, osa ei opeta sitä lainkaan.

Opettajien vastaukset viestivät siitä, että ohjelmoinnista on hyötyä useiden eri taitojen kehittymiselle. He kuvailivat ohjelmoinnin opetuksen kehittävän monenlaista osaamista, kuten muun muassa ohjelmointitaitoa, matemaattista ajattelua ja ongelmanratkaisutaitoja. Tulokset ovat näiltä osin tiiviisti linjassa aiemman tutkimuksen kanssa. Ohjelmoinnin opetuksen on todettu aiemmissakin tutkimuksissa harjaannuttavan erityisesti matemaattisia taitoja ja ongelmanratkaisutaitoja (esim. Fessakis ym. 2013; Kalelioğlu 2015; ks. myös Niskanen 2015). Ennen kaikkea sen on korostettu harjaannuttavan ohjelmointiajattelua (esim. Wing 2006). Ohjelmoinnin on havaittu kehittävän myös kykyä ymmärtää ohjelmoinnin logiikkaa ja lisäävän sinnikkyyttä (ks. Sáez-López ym. 2016), harjaannuttavan hahmottamisen taitoja (Román-González, Pérez-González & Jiménez-Fernández 2017), sekä kehittävän sosiaalisia taitoja (Fessakis ym. 2013) ja kykyä palastella asioita pienempiin osiin (ks. Harlow ym. 2015). Lisäksi Niskanen (2015) on havainnut, että ohjelmoinnin opetuksen lisänneen oppilaiden luottamusta omiin kykyihin arkipäivän ongelmien ratkaisemisessa, kun ohjelmoinnin kautta opittu osaaminen siirtyi arkipäivän tilanteisiin.

7.1.2 Kokemukset ohjelmoinnin opetuksesta

Ohjelmoinnin opetuksen edellytykset. Opettajien puutteelliset valmiudet ohjelmoinnin opettamiseen nousivat vahvasti esiin aineistosta, mutta ohjelmoinnin edellytyksinä mainittiin lisäksi myös opettajien asenteet, koulun laitteistoresurssit ja aikaresurssit. Vaikka suomalaisopettajien digitaidot ovat kehittyneet viimeaikoina, heidän ohjelmointitaitonsa ovat edelleen valtakunnallisella tasolla heikot (Karakainen ym. 2017). Suuri osa opettajista kokee oman osaamisensa ja käytössään olevat työvälineet riittämättömiksi opetusteknologian käyttämiseen opetuksessa (Hietikko ym. 2016), ja kansainvälisesti on esitetty arvioita, että jopa kaksi kolmasosaa opettajista saattaa kokea heillä olevan riittämättömät taidot ja työvälineet ohjelmoinnin opettamiseen (ks. George 2017). Samaten tutkimukset osoittavat, että monet opettajat eivät tunnista esimerkiksi robottien hyötyä opetuskäytössä ja vaikka tunnistaisivat, heillä ei olisi valmiuksia käyt-

tää robotiikkaa opetuksessa (ks. Alimsis ym. 2007; Matarić ym. 2007), ja vain harva opettaja osaa opettaa jotain ohjelmointikieltä (Karakainen ym. 2017; Karvonen & Laukka 2016). Karakaisen ja kumppaneiden (2017) mukaan ohjelmointiosaaminen on sekä opettajilla että oppilailla varsin heikkoa, ja opettajista vain viidennes on ylipäätään kokeillut ohjelmointia oppilaiden kanssa. Lisäksi tähän tutkimukseen osallistuneiden opettajien mukaan tarpeeksi opetusmateriaaleja ja digitaalisia laitteita ei ole aina saatavilla, mikä on linjassa aiempien tulosten kanssa (esim. Israel ym. 2015; Taussi 2017).

Tässä tutkimuksessa haastateltujen opettajien tv-taidot olivat kuitenkin heidän kertomansa mukaan hyvät, mikä myötäilee joidenkin aikaisempienkin tutkimusten tuloksia (ks. Karakainen ym. 2017; Taussi 2017). Toisaalta osa tutkimuksista osoittaa päinvastaisesti, että opettajien tv-osaaminen ei ole riittävää (esim. Hietikko ym. 2016).

Tutkimukseen osallistuneet opettajat mainitsivat omaavansa myönteisen asenteen ohjelmoinnin opetusta kohtaan, mikä on linjassa joidenkin aiempien tutkimusten tulosten kanssa (ks. Israel ym. 2015; Karvonen & Laukka 2016). Toisaalta opettajien on myös havaittu suhtautuvan ohjelmoinnin opetukseen varauksella esimerkiksi puutteellisen ohjelmointiosaamisen vuoksi, ja myönteisen suhtautuminen on edellyttänyt ammatillisen tuen saamista (ks. Israel ym. 2015). Tutkimuksissa opettajien tv-osaamisen on tutkimuksissa havaittu ennustavan myönteistä suhtautumista opetusteknologian käyttöä kohtaan (ks. Albirini 2006; Cavas ym. 2009; Sherer, Tondeur, Siddiq & Baran 2018), ja sama pätee myös ohjelmointiin. Samaten on myös havaittu, että tv:n käyttäminen opetuksessa edellyttää riittävää tukea ja täydennyskoulutusta (Hietikko ym. 2016), ja että opettajien asenteita pystytään parantamaan koulutuksen avulla (van Aalderen-Smeets, Walma van der Molen & Asma 2012).

Tämän tutkimuksen tuloksista ilmennyt opettajien myönteinen suhtautuminen ohjelmoinnin opetukseen saattaa viitata siihen, että opettajat olisivat valmiita tarttumaan ohjelmoinnin opetuksen haasteeseen ennakkoluulottomasti. Toisaalta esiin tuotu jännitys siitä, ettei osaa opettaa ohjelmointia, viestii siitä, että kaikilla opettajilla ei ole tarpeeksi valmiuksia ohjelmoinnin opettamiseen, ja

jotta ohjelmointia pystyttäisiin opettamaan, heille tulisi taata riittävästi tukea ja lisäkoulutusta.

Aiempien tutkimusten (esim. Thibaut, Knipprath, Dehaene & Depaepe 2018; Xie, Talin & Sharif 2014) tulokset viittaavat siihen, että opettajien asenteilla on yhteys heidän opetusmenetelmiinsä. Ylimaz ja Bayaktar (2014) ovat havainneet, että opettajien asenteet opetusteknologiaa kohtaan ovat yhteydessä heidän opetusmenetelmiensä innovatiivisuuteen. Xien ja kumppaneiden (2014) mukaan myös opettajien osaaminen on yhteydessä käytettyihin opetusmenetelmiin. Samalla saadaan viitteitä siitä, että parantamalla opettajien asenteita ja osaamista pystytään kehittämään myös opetusta.

Vaikka opettajien iän yhteys ohjelmoinnin opettamiseen ei ollutkaan tässä tutkimuksessa tutkimuskohteena, on mielenkiintoinen huomio, että opettajien ikä ei näyttänyt olevan yhteydessä heidän suhtautumiseensa ohjelmointia kohtaan. Tutkimukseen osallistuneet opettajat olivat hyvin eri-ikäisiä, mutta vähäisiä ohjelmointikokemuksia löytyi sekä nuoremasta että vanhemmasta päästä. Opettajien iän yhteydestä heidän asenteisiinsa opetusteknologiaa kohtaan on saatu ristiriitaisia tuloksia aiemmissä tutkimuksissa. Toisaalta tulokset ovat osoittaneet iän ja asenteiden välisen yhteyden (ks. Cavas ym. 2009; Saari ym. 2016), toisaalta taas yhteyttä ei ole havaittu kaikissa tapauksissa (ks. Ylimaz ja Bayaktark 2014). Samaten Saari kollegoineen (2016) on havainnut, että vähemmän kokemusta omaavilla opettajilla oli kokeneempia opettajia myönteisempi asenne ohjelmoinnin oppimista kohtaan sekä enemmän uskoa omiin kykyihinsä oppia ohjelmointia. Kaarakainen ja kumppanit (2017) ovat havainneet, että opettajien ikä oli yhteydessä myös heidän tvt-osaamiseensa: nuoremmat pärjäsivät vanhempia paremmin. Ei siis voida ajatella, että nuoret opettajat olisivat aina kokeneempia opettajia opetusteknologia- ja ohjelmointimyönteisempiä sekä kokeneempia ohjelmoinnissa.

Ajan ja laitteiston puute opetusteknologian käytön ongelmina on ollut esillä useissa tutkimuksissa. Riittämätön aika on aiemmissä tutkimuksissa todettu yhdeksi opetusteknologian hyödyntämisen suurimmiksi ongelmakohtiksi (esim. Şahin-Kızıl 2011), vaikka ohjelmoinnin opetuksen suurimpana haasteena onkin yleensä opettajien heikko osaaminen (ks. Israel ym. 2015; Larson

2013; Tivia 2015). Tähän tutkimukseen osallistuneiden opettajien mukaan tarpeeksi opetusmateriaaleja ja digitaalisia laitteita ei ole aina saatavilla, mikä on linjassa aiempien tulosten kanssa (esim. Israel ym. 2015; Taussi 2017). Koulujen vaihtelevat välineistöresurssit voivat olla opetusta eriarvoistava tekijä. Vaikka valtakunnallisesti digitaalisissa resursseissa ja opettajien ja oppilaiden osaamisessa ei olekaan havaittu merkittäviä eroja koulujen välillä, huolta kannetaan koulujen eriarvoistumisesta kuntakohtaisen vaihtelun vuoksi (Hietikko ym. 2016). Lisäksi pulmallista voi olla se, saako teknologiaa hyödyntämällä tuotua lisäarvoa opetukseen (McCulloch, Hollebrands, Lee, Harrison & Mutlu 2018).

Ohjelmoinnin opetuksen toteuttaminen. Tutkimustuloksista ilmenee, että kaikki luokanopettajat eivät opeta ohjelmointia, vaikka opetussuunnitelma (OPS 2014) siihen velvoittaa. Tulosten perusteella on nähtävissä, että se, opettavatko opettajat ohjelmointia, riippuu pitkälti opettajien omasta osaamisesta, innostuksesta sekä resursseista, kuten koulun opetusmateriaaleista ja aika-resursseista. Aineistosta nousi myös esiin, että ohjelmointia tapahtuu koulun arjessa jatkuvasti tiedostamattakin. Samaan tapaan myös García-Peñalvo ja kumppanit (2016) ovat esittäneet, että monet opettajat käyttävät tiedostamattaan opetuksessa aktiviteetteja, jotka kehittävät koodausosaamista.

Opettajat nostivat esiin ohjelmoinnin opetuksen menetelminä muun muassa erilaiset leikit, pelit, sovellukset ja robotit. Opettajien vastauksissa näkyi opettajien kekseliäisyys ja innovatiivisuus – vain taivas tuntui olevan rajana ohjelmoinnin opetuksessa, jopa ilman digitaalisia laitteita. Opettajat kuvaukset ohjelmoinnin opetuksen etenemisestä olivat yhteneväisiä opetussuunnitelman (OPS 2014) ohjelmoinnin opetuksen tavoitteiden kanssa. Erityisesti alkuvaiheessa ohjelmointia opiskellaan leikkien ja pelaten ja opetellaan antamaan vaihtelevia toimintaohjeita. Myöhemmin opitaan antamaan tietokoneelle käskyjä graafisissa ohjelmointiympäristöissä ja tutustaan esimerkiksi robotiikkaan. Toisaalta opettajien käyttämät teknologiset opetusmateriaalit ohjelmoinnin opetukseen näyttivät kuitenkin rajoittuvan vain muutamaan ohjelmaan. Opettajilla ei näytä olevan tarpeeksi aikaa perehtyä muihin sovelluksiin ottaakseen niitä käyttöön tai tarpeeksi osaamista muiden sovellusten mahdollisuuksien ymmärtämiseen, vaan he tarvitsisivat lisäkoulutusta niistä. Vaihtoehtoisesti kyseiset

resurssit on saatettu todeta niin hyväksi, että niiden käytössä halutaan pitäytyä. Opettajien antamat perustelut opetusmateriaalien valinnoille, kuten muun muassa niiden tuttuus, olivat pitkälti yhteneväisiä Taussin (2017) tulosten kanssa.

Opettajien käyttämät Scratch ja Code.org ovat kansainvälisestikin suosittuja ja yleisesti käytettyjä resursseja ohjelmoinnin opetukseen (esim. Israel ym. 2015). Samaten opettajien mainitsema Lego Mindstorms on García-Peñalvon ja kumppaneiden (2016) mukaan yksi parhaimmista robotiikkaresursseista. Opettajien käyttämät opetusmateriaalit ovat linjassa Taussin (2017) tulosten kanssa, jotka osoittavat, että alkuopetuksessa ohjelmoinnin opetuksessa hyödynnetään paperilla tehtäviä harjoituksia, robotteja, leikkejä, sovelluksia, verkkosivustoja, ohjelmointikirjoja, lautapelejä ja palikoita. Mielenkiintoista Taussin tuloksissa on, että vain harva opettaja käytti tietoteknisiä sovelluksia ohjelmoinnin opettamiseen. Taussin tulokset yhdessä tämän tutkimuksen tulosten kanssa vahvistavat ajatusta siitä, että ohjelmointia pystyy opettamaan hyvin myös ilman tieto- ja viestintäteknologiaa, erityisesti ohjelmoinnin alkeita.

Tulokset osoittavat, että opetussuunnitelman (OPS 2014) hengen mukaisesti ohjelmointia opetetaan yli oppiainerajojen osana useiden eri oppiaineiden opetusta. Ohjelmointia integroidaan selvästi eniten matematiikkaan ja äidinkielen, mutta myös muihin oppiaineisiin. Tämä on ristiriidassa Marttalan (2017) tulosten kanssa, jotka viittaavat siihen, että vaikka peruskouluissa tv:n opetusta integroidaan usein muihin oppiaineisiin, ohjelmoinnin opetus tapahtuu pääosin erillisenä kokonaisuutena. Tämän tutkimuksen tulokset ovat kuitenkin osin linjassa kansainvälisten tutkimusten kanssa, joissa on havaittu, että ohjelmoinnin opetusta sisällytetään pääosin matematiikan ja luonnontieteiden opetukseen (esim. Israel ym. 2015). Lisäksi ohjelmoinnin on esitetty sopivan integroitavaksi myös kuvataiteen, kielten, historian ja yhteiskuntaopin opetukseen esimerkiksi projektityöskentelyn kautta (ks. Sáez-López ym. 2016). Ohjelmoinnin opetuksen voisi ajatella sopivan erityisen hyvin juuri äidinkielen ja matematiikan opetukseen, koska muun muassa lukutaito ja matemaattista ajattelu kehittyvät ohjelmointia opiskellessa (ks. Flannery ym. 2013).

Opettajien ja oppilaiden vaihtelevat kokemukset ohjelmoinnin opetuksesta. Ohjelmoinnin opettaminen koetaan opettajien vastausten perusteella mie-

lekkäänä, mutta haasteellisena, ja opetuksen ongelmakohtina näyttäytyvät ajan puute, oppilaiden taitoerot ja laitteistoresurssit. Kiinnostavaa tuloksissa on, että aineistosta esiin kummunneet ohjelmoinnin opetuksen ongelmakohdat muistuttavat 1980-luvun ensiaskelia tv:n opetuksessa, jolloin opetuksen haasteita olivat muun muassa tietokoneiden riittämättömyys, opetukseen käytettävän ajan vähyys sekä puutteet opettajankoulutuksessa (Saarikoski 2006). Tulokset viestivät siitä, että samantapaiset vaikeudet ovat edelleen ongelmana kouluopetuksessa. Lisäksi Matarić ja kumppanit (2007) esittävät, että muita esteitä ohjelmoinnin opetukselle ovat yleisesti käyttövalmiiden ja oppilaiden iälle sopivien opiskelumateriaalien ja edullisten laitteiden vähäisyys. Testattua suomenkielistä opetusmateriaalia ei esimerkiksi ole vielä toistaiseksi paljon tarjolla (Liukas & Mykkänen 2014).

Tulokseni viittaavat aiempia tutkimustuloksia (esim. Kalelioğlun 2015; Sáez-López ym. 2016; Wilson & Moffat 2010) myötäillen siihen, että opettajien lisäksi myös oppilailla on pääosin myönteinen asenne ohjelmoinnin opetusta kohtaan. Opettajat kertoivat, että ohjelmoinnin opetus motivoi oppilaita, mikä on linjassa aiempien tulosten kanssa (esim. Fessakis ym. 2013). Tutkimuksissa on tullut ilmi, että oppilaat kokevat ohjelmoinnin yleisesti innostavaksi (Fessakis ym. 2013) sekä hyödylliseksi (Sáez-López ym. 2016). Ohjelmoinnin opiskelun on nähty olevan myös yleissivistävää toimintaa ja tuovan ammatillista erityisosaamista (Hämeenaho 2016). Toisaalta opettajien mukaan osa oppilaista kokee ohjelmoinnin tylsänä tai vaikeana. Aiemmissakin tutkimuksissa on havaittu, että ongelmana saattaa olla kiinnostuksen puute tai se, että ohjelmointi koetaan vaikeaksi (ks. Ouahbi, Kaddari, Darhmaoui, Elachqar & Lahmine 2014).

Esteves ja kumppanit (2011) ovat havainneet, että oppilaiden asenne ohjelmointia kohtaan on yhteydessä heidän sinnikkyyteensä ja työskentelyyn sitoutumiseen, ja näkyy tuotosten laadussa. Durak ja Saritepeci (2018) ovat lisäksi todenneet oppilaiden matemaattisen osaamisen ja myönteisen asenteen olevan yhteydessä ohjelmointiajattelun taitoihin. Useiden tutkimusten tulokset viittaavat siihen, että lasten asenteet tieto- ja viestintäteknologiaa ja ohjelmointia kohtaan muovautuvat pitkälti heidän vanhempiensa asenteiden mukaan (ks. Hart

2010; Margolis & Fisher 2002). Koulussa voidaan kuitenkin pyrkiä lisäämään oppilaiden innostusta ohjelmointia kohtaan.

Tutkimukseen osallistuneiden opettajien mukaan ohjelmointi saattaa toisinaan olla oppilaille haastavaa. Aiemmat tutkimukset osoittavat samankaltaisesti, että ohjelmointi, ja erityisesti tarkkojen toimintakäskyjen antaminen, on joillekin oppilaille vaikeaa (esim. Fessakis ym. 2013; Kalelioğlu 2015). Fessakis kumppaneineen (2013) kuitenkin painottaa, että ongelmat hälvenevät pian ohjelmistojen tullessa lapsille tutuiksi. Lisäksi on huomioitavaa, että ohjelmoinnissa keskeistä ei ole se, tehdäänkö jokin asia oikein vai väärin, vaan se, miten eteen tulleet ongelmat voitaisiin ratkaista (Papert 1980). Ajatusmallina toimii virheiden etsiminen ja korjaaminen. Pelkkä joko oikein tai väärin tekemisen ajatusmalli saattaa Papertin (1980) mukaan estää oppilasta oppimasta.

Opettajien mukaan oppilaiden taitotasossa tai motivaatiossa ei ole ohjelmoinnin opetuksessa merkittäviä eroja sukupuolten välillä. Aiemmatkin tutkimukset ovat osoittaneet, että sukupuolen ja ohjelmointitaidon välillä ei ole havaittu yhteyttä (esim. Bruckman, Jensen & Deonte 2002; Kalelioğlun 2015; Ows-ton, Wideman, Ronda & Brown 2009; Vos, van der Meijden & Denessen 2011). Mielenkiintoista kuitenkin on, että muista poiketen Atmatzidoun ja Demetriadiksen (2016) ovat tulleet päinvastaiseen tulokseen. Heidän mukaansa tytöt tarvitsevat usein poikia enemmän harjoittelua yltääkseen samaan taitotasoon. Myös Román-González ja kumppanit (2017) ovat esittäneet, että pojilla olisi jo lähtökohtana tyttöjä paremmat ohjelmointiajattelun taidot. Toisaalta nämä havainnot ovat täysin ristiriidassa sen kanssa, että Robertson (2012) on havainnut tyttöjen suoriutuneen ohjelmoinnissa peräti poikia paremmin. Bruckman on kumppaneineen (2002) esittänyt, että taitoerot selittyvät todennäköisimmin ohjelmointikokemuksen määrällä: pojilla on usein tyttöjä enemmän kokemusta ohjelmoinnista. He havaitsivat ohjelmointitehtävissä suoriutumisen riippuvan sekä tehtävään käytetystä ajasta että aikaisemmasta ohjelmointikokemuksesta.

Opettajien kokema tuen tarve ohjelmoinnin opetuksessa. Tutkimukseen osallistuneista opettajista suurin osa kuvaili tarvitsevansa tukea ja lisäosaamista ohjelmoinnin opetukseen. Toivotuin tuen muoto oli täydennyskoulutus. Se nähdään aiemmissakin tutkimuksissa erittäin tärkeänä ja jopa ohjelmoinnin

opetuksen edellytyksenä (ks. Israel ym. 2015; Kaarakainen ym. 2017; Şahin-Kızıllı 2011). Israel ja kumppanit (2015) havaitsivat, että opettajat, joilla ei ollut aiempaa ohjelmointiosaamista, eivät olisi ilman ammatillista kehittymistä edes yrittäneetkään opettaa ohjelmointia.

Täydennyskoulutusta ei kuitenkaan ole tarpeeksi tarjolla, tai täydennyskoulutus saattaa olla riittämätöntä tai väärin suunnattua. Vaikka tutkimukset osoittavat, että opetuksen muuttamiseen tarvitaan tukea, ohjausta ja johtajuutta (ks. NRC 2000; Thibaut ym. 2018), täydennyskoulutusta ei tämän tutkimuksen tulosten mukaan ole tarpeeksi tarjolla. Mielenkiintoista on myös, että Kaarakaisen ja kumppaneiden (2017) mukaan nuoret opettajat ovat saaneet vanhempia opettajia enemmän digitäydennyskoulutusta. Täydennyskoulutusmahdollisuudet eivät näytä siis jakautuvan opettajien kesken tasapuolisesti. Lisäksi Hietikko ja kumppanit (2016) raportoivat, että opettajien tv-täydennyskoulutus saattaa olla vain muutaman tunnin mittainen, ja täydennyskoulutus ei keskity pedagogiikkaan, vaan lähinnä laitteiden tekniseen käyttöön. Digikoulutus liikkuu kuitenkin oikeaan suunaan, sillä Hietikon ja kumppaneiden havaintojen mukaan opettajankoulutuksen aikana saadaan aiempaa enemmän tv-koulutusta, vaikka se ei katakaan edelleenkään kaikkia opettajaopiskelijoita, ja suurin osa opettajista saa tv-täydennyskoulutusta kentällä työskennellessään.

Täydennyskoulutukseen osallistumisen ohella osa tutkimukseen osallistuneista opettajista oli harjaannuttanut itse omaa ohjelmointiosaamistaan esimerkiksi perehtymällä aiheeseen internetissä. Samankaltaisesti myös Taussin (2017) tulokset ovat osoittivat, että opettajista yli puolet käytti myös omaa aikaansa esimerkiksi erilaisiin sovelluksiin tai kirjallisuuteen perehtymiseen. García-Peñalvo ja kumppanit (2016) peräänkuuluttavatkin sellaisten verkkosivustojen luomista opettajien tueksi, joista löytyisi ideoita ja resursseja ohjelmoinnin opetukseen sekä ajankohtaista tutkimustietoa aiheesta. Lisäksi Kaarakaisen ja kumppaneiden (2017) tulosten mukaan opettajien tukemisessa myös vertaistuki ja tutoropettajat ovat suuressa roolissa.

7.2 Tulosten merkitysten arviointia

Tämän tutkimuksen tulokset antavat viitteitä siitä, että luokanopettajien ohjelmointiosaamisessa olisi parannettavaa, ja että ohjelmoinnin opetusta toteutetaan vaihtelevasti opettajasta ja resursseista riippuen. Tulokset viestivät vahvasti siitä, että jatkossa tulee kohdistaa tarkempaa huomiota opettajien kouluttamiseen ohjelmoinnin opetuksen mahdollistamiseksi.

Opettajien ammatillisen kehittymisen merkitys. Kaikki luokanopettajat eivät omaa hyviä taitoja ohjelmoinnissa eivätkä opeta ohjelmointia tietoisesti ja säännöllisesti. Opettajien tiedon puutteen onkin todettu olevan yhteydessä siihen, opettavatko he ohjelmointia ja jos opettavat, miten he sitä opettavat, ja myös oppilaat saattavat tietämättömyyden vuoksi luulla ohjelmointia tylsäksi tai vaikeaksi (Wilson & Moffat 2010). Myös Armoni (2011) on esittänyt, että opettajien ja oppilaiden virheelliset mielikuvat ja ennakkoluulot ovat yhteydessä heidän asenteisiinsa ohjelmoinnin oppimista ja ohjelmointialan uraa kohtaan. Israelin ja kumppaneiden (2015) tulokset vahvistavat näitä näkemyksiä, mutta he ovat kuitenkin havainneet, että opettajien saadessa ohjausta ammatilliseen kehittymiseen nämä opettajien ennakkoluulot hävisivät. Kokemuksen myötä huoli ohjelmoinnin opettamisesta siis väheni. Tästä voidaan tehdä johtopäätös, että opettajat ovat todennäköisesti valmiita ohjelmoinnin opettamiseen, mutta sen ehdoton edellytys on riittävä tuki ja täydennyskoulutus.

Tulokset viittasivat siihen, että ohjelmoinnin opettamisessa on opettajien kohdalla yksilökohtaista vaihtelua, joka selittyy ammatillisuuteen ja opettajien persooniin liittyvillä eroilla sekä koulujen resursseilla ja aikarajoitteilla. Israel kollegoineen (2015) esittää, että tulevaisuudessa tulisi arvioida, miten voitaisiin tukea opettajien ammatillista kehittymistä, jotta he pystyisivät opettamaan ohjelmointia. He painottavat, että ilman riittävää aikaa ja taloudellisia resursseja ohjelmointia ei pystytä opettamaan alakoulussa. Tulevaisuudessa tulee pohtia, miten voitaisiin tukea erityisesti sellaisten luokanopettajien ammatillista kehittymistä, joilla on vähän tai ei lainkaan kokemusta ohjelmoinnista, jotta he pystyisivät opettamaan ohjelmointia. Tukea tulisi lisäksi tarjota myös niille opetta-

jille, joilla on ohjelmoinnin perusteet hallussa, jotta he pystyisivät opettamaan ohjelmointia pidemmälle.

Opettajien kouluttamisen tarve. Opettajien kouluttaminen on tärkeää ohjelmoinnin opetuksen toteutumisen kannalta. Tulee panostaa sekä jo kentällä olevien opettajien täydennyskouluttamiseen että opettajankoulutuksen muokkaamiseen paremmin työelämän vaatimuksia vastaavaksi. Opettajat tarvitsevat tukea uuden opetussuunnitelman käyttöönotossa ja ohjelmoinnin opetuksessa. Yhtenä haasteena on riittävän täydennyskoulutuksen mahdollistaminen kentällä työskenteleville opettajille. Koulujen muutos on pitkälti opettajien varassa, mutta heitä ei saa jättää ilman tukea. Hietikko ja kumppanit (2016) muistuttavat, että opettajilla on oikeus täydennyskoulutukseen, ja se myös kuuluu heidän työhönsä. Tämän tutkimuksen tulokset antavat viitteitä siitä, että täydennyskoulutukselle on tarvetta, mutta tällä hetkellä oman ohjelmointiosaamisen kehittäminen jäänee kuitenkin pitkälti opettajien omalle vastuulle ja omalle ajalle. Lisäksi huomioitavaa on, että täydennyskoulutuksen tulisi kohdentua pedagogiikkaan, ei pelkästään laitteiden käyttöön (Hietikko ym. 2016). Ihannetilanteessa jokaisella opettajalla olisi mahdollisuus täydennyskouluttautua. Opettajien toiveita kuuntelemalla voitaisiin saada aikaan parempia täydennyskoulutuksia, jotka kohdistuisivat nimenomaan niihin teemoihin, joista opettajat kokevat tarvitsevansa lisäkoulutusta.

Haasteena on myös kehittää opettajankoulutusta vastaamaan tämän päivän sekä tulevaisuuden työelämän tarpeita. Opettajankoulutusta tulee kehittää sellaiseksi, että jokainen koulutuksesta valmistuva luokanopettaja saisi yhtä hyvät lähtökohdat ja valmiudet lähteä opettamaan ohjelmointia. Opiskelijoiden tulisi saada yhdenmukaista opetusta ohjelmoinnista riippumatta siitä, minkä yliopiston koulutuksessa opiskelee. Ei voida olettaa, että opettajat selviytyisivät ohjelmoinnin opettamisesta, jos heitä ei kouluteta siihen. Luokanopettajankoulutuksen tulisi taata kaikille opiskelijoille riittävä osaaminen ohjelmoinnista ja tvt:sta, jotta he pystyvät työelämässä opettamaan opetussuunnitelman hengen mukaisesti.

Oppilaiden yhdenvertaisuus ohjelmoinnin opiskelussa. Opettajien lisäkoulutusta tarvitaan, koska oppilaiden yhdenvertaisuus ei näytä tällä hetkellä

täysin toteutuvan ohjelmoinnin opetuksen kohdalla suomalaisissa peruskouluissa. Suomalainen koulujärjestelmä antaa korkeasti koulutetuille opettajille vapauden toteuttaa opetussuunnitelmaa ja opetusta varsin vapaasti omalla tavallaan. Toisaalta kun kunkin opettajan henkilökohtaiset ominaisuudet pääsevät vaikuttamaan työn toteuttamiseen, opetuksen sisältö ja laatu vaihtelevat koulujen välillä ja opettajasta riippuen, eikä kaikille oppilaille pystytä välttämättä takaamaan samanlaisia mahdollisuuksia oppia (ks. Norrena 2013). Tämä tarkoittaa sitä, että yhdenmukainen ja tasapuolinen peruskouluopetus ei täysin toteudu. Kuitenkin opetussuunnitelman (OPS 2014) lisäksi myös perusopetuslaissa (628/1998, 2§) ja yhdenvertaisuuslaissa (1325/2014, 6§) määrätään perusopetuksen yhdenvertaisuudesta. Kaikilla oppilailla tulisi siis tällä perusteella olla samanlaiset opiskelumahdollisuudet, eli samanlaiset mahdollisuudet esimerkiksi kokeilla ohjelmointia ja kehittää omaa tv-t- ja ohjelmointiosaamistaan.

Tarkemman määrittelyn tarve. Tutkimuksen tulokset viittaavat siihen, että kouluissa olisi tarvetta nykyistä tarkemmalle ohjeistukselle ohjelmoinnin opetuksen toteuttamisesta, sillä kaikki luokanopettajat eivät näytä opettavan ohjelmointia, vaikka opetussuunnitelma velvoittaa siihen. Vaikka ohjelmoinnin opetuksen tavoitteet on kirjattu opetussuunnitelmaan (OPS 2014), se ei näytä tarjoavan tarpeeksi tarkkaa määritelmää siitä, mitä ohjelmoinnin opetuksen tulee pitää sisällään. Koska opetussuunnitelman raamit ovat varsin väljät, tarkemmat päätökset ja valinnat opetuksen toteuttamisen muodoista jäävät koulukohtaisiksi ja opettajien vastuulle. Ohjelmoinnin opetuksen sisältöjä ja tavoitteita voisi olla tarpeen tarkentaa koulukohtaisesti paikallisissa opetussuunnitelmissa.

Oleellista olisi kuitenkin yleisesti määritellä ohjelmoinnin käsite alakoulu-kontekstissa, sillä ei voida olettaa, että ohjelmointia opetettaisiin tasapuolisesti koulusta riippumatta ja noudatettaisiin opetussuunnitelman vaatimuksia ohjelmoinnin opetuksen suhteen, jos ei alun alkaenkaan määritellä tarkasti, mitä ohjelmoinnilla tarkoitetaan ja mitä kaikkea ohjelmoinnin opetukseen tulisi sisältyä. Kirjallisuudessakin ohjelmointi määritellään yksinkertaisimmillaan hyvinkin laveasti toimintaohjeiden antamiseksi (ks. Hyvönen ym. 2013), joka jättää paljon tulkinnan varaa.

Tulisi myös tehdä yhteinen linjaus siitä, onko jokaisen oppilaan tarkoitus oppia koodaamaan alakoulussa, ja keskitytäänkö opetuksessa opettamaan konkreettisia ohjelmointitaitoja vai laajempia kokonaisuuksia kuten ohjelmointiajattelua. Balanskat ja Engelhardt (2015) nostaisivat määriteltäväksi sen, min-kälaisia ohjelmointitaitoja jokainen oppilas tarvitsee tulevaisuuden digitaalisessa maailmassa, ja miksi koodaustaito olisi heille hyödyllinen taito. He esittävät, että tulee pohtia, miten pystyttäisiin opettamaan ohjelmointia parhaalla mahdollisella tavalla, mitä hyötyjä ja rajoitteita sen opettamisella yli oppiainerajojen on, ja miten sen opetusta pystytään arvioimaan. Tulisi myös pohtia, miten ohjelmoinnin opiskelusta voitaisiin tehdä merkityksellistä, yksinkertaista ja mielekäästä, ja miten se saataisiin integroitua osaksi perusopetusta siten, että se toimisi yhtenä työvälteenä muiden joukossa tukemassa oppimista. Oleellista olisi lisäksi pohtia, millainen merkitys ohjelmoinnin formaalilla opetuksella nähdään ongelmanratkaisutaitojen ja ohjelmointiajattelun oppimisessa, sillä tällaisia ohjelmoinnillisia taitoja voidaan harjaannuttaa monenlaisen muunkin toiminnan kautta. On myös järkevää miettiä, tarvitsevatko tämän päivän koululaiset ohjelmointiosaamista tulevaisuudessa yhtä paljon kuin tällä hetkellä kuvitellaan.

Opetusmenetelmiä ohjelmoinnin opetukseen. Näyttää siltä, että koulu-maailmassa olisi tarvetta lisätiedolle siitä, millaisilla konkreettisilla aktiviteeteilla, sovelluksilla ja ohjelmointikielillä ohjelmointia voi opettaa eri luokka-asteilla sekä miten ohjelmointia voidaan tarkoituksenmukaisesti integroida eri oppiaineisiin (ks. myös Balanskat ja Engelhardt 2015). Tarvittaisiin myös yksinkertaisia opetusmateriaaleja, jotka opettajien olisi helppo ottaa käyttöön. Fessakis ja kumppanit (2013) painottavat, että pelkkä ohjelmointiohjelmien olemassaolo ei kuitenkaan riitä koodaustaidon opettamiseen ja ohjelmoinnin opetuksen integroimiseen sujuvasti muuhun opetukseen, vaan tarvitaan tutkittuja, hyviä opetuskäytänteitä ja -materiaaleja. Lisäksi huomionarvoista on, että pelkkä tieto- ja viestintäteknologian käyttö itsessään ei paranna oppimistuloksia (Buckingham 2007; Honey, McMillan Culp & Carrigg 2000; Kaila 2018); olennaista on tv:t:n tarkoituksenmukainen käyttö osana opetusta.

On tärkeää, että opettajille on tarjolla valmista opetusmateriaalia ohjelmoinnin opetukseen, sillä se mahdollistaa ohjelmoinnin opettamisen sellaisille-

kin opettajille, joilla ei ole ohjelmointiosaamista eikä valmiuksia itsenäisesti suunnitella innovatiivista ohjelmoinnin opetusta. Opettajien onneksi internetistä löytyy useita verkkosivustoja, joihin on koottu valmiita tuntisuunnitelmia ja ideoita opetukseen (esim. Code.org; ScratchEd). Samoin joiltain sivustoilta löytyy suoraan käyttöön otettavia koodaustunteja ja -kurseja oppilaille (esim. Code.org), jotka eivät edellytä opettajalta opetuksen suurta etukäteissuunnittelua. Ongelmana on kuitenkin se, että tällaisiin resursseihin perehtyminen jää opettajien omalle vastuulle. Erilaisia resursseja opetukseen on tarjolla todella paljon, mutta sopivien materiaalien löytäminen vie aikaa ja niiden mahdollisuuksien oivaltaminen vaatii ymmärrystä ohjelmoinnista ja sen opettamisesta.

7.3 Tutkimuksen luotettavuus ja jatkotutkimusmahdollisuuksia

Tutkimuksen luotettavuus ja rajoitukset. Tutkimuksen luotettavuutta arvioitaessa tulee ottaa huomioon tutkimukseen liittyvät rajoitteet. Keskeistä on tutkimuksen yleistettävyyteen, otokseen, tutkijaan, tutkimuksen relevanssiin sekä hyvien tieteellisten käytäntöjen noudattamiseen liittyvien seikkojen arvioiminen.

Koska tässä tapauksessa kyseessä on laadullinen tutkimus, tutkimuksessa ei pyritä yleistyksiin vaan tavoitteena on ymmärtää tutkittavaa ilmiötä (ks. Hirsjärvi ym. 2016). Laadullisen tutkimuksen ei Alasuutarin (2011) mukaan tarvitse päteä muussakin kuin tutkitussa tapauksessa, ja laaja yleistettävyys ei ole keskiössä fenomenografisesti ja fenomenologisesti yksilön käsityksiä ja kokemuksia tarkasteltaessa. Tutkimuksen tavoitteena olikin yleistämisen sijaan valottaa opettajien käsityksiä ja kokemuksia ohjelmoinnin opettamisesta. Tarkoituksena ei ollut vetää yleispäteviä johtopäätöksiä ohjelmoinnin opetuksesta, vaan tutkimuksessa pyrittiin ymmärtämään, miten ohjelmoinnin opetus käsitellään ja koetaan, sekä kuvailla ohjelmoinnin opetuksen toteutumista alakoulussa. Kuitenkin tutkimuksen tulosten pitkälti vastatessa aiempia tutkimustuloksia opettajien käsityksistä ja kokemuksista ohjelmoinnin opettamisesta niitä voidaan pitää jossain määrin yleistettävinä.

Aineiston pieni koko voi olla yksi tutkimuksen luotettavuutta heikentävä tekijä, vaikka kvalitatiivisessa tutkimuksessa tutkimus kohdistuukin usein pieneen joukkoon (Eskola & Suoranta 2014). Otoksen koko saattaa heikentää tulosten yleistettävyyttä, ja otoksen pienuuden vuoksi tulokset ovat luonteeltaan kuvailevia. Toisesta tutkimuskohteesta olisi saatettu saada erilaista tietoa. Tutkimukseen osallistujien määrä on varsin pieni, koska haastateltavien löytäminen osoittautui haasteelliseksi. Erityisesti heterogeenisyyttä aineistosta etsittäessä aineiston koon tulisi Hirsjärven ja kumppaneiden (2016) mukaan olla suurempi. Toisaalta jo näin pienessä tutkimusjoukossa alkoi esiintyä saturaatiota eli samankaltaisia näkemyksiä (ks. Eskola & Suoranta 2014), joskin Hirsjärvi ja kumppanit (2016) huomauttavat, että saturaation ongelmakohtana on, että saturaation ja uusien näkökulmien havaitseminen on tutkijasta kiinni.

Laadullisen tutkimuksen luotettavuuden kannalta tutkijan rooli on merkittävä. Tässä tutkimuksessa tutkijan rooli nousee esiin fenomenologishermeneuttisen ja fenomenografisen tutkimusotteen ja aineistolähtöisyyden tuomissa tulkinnan haasteissa sekä tutkijan kokemattomuudessa. Tutkimuksen validiteettia määrittävät tutkijan tekemät tulkinnat, joihin liittyy aina kysymyksiä tiedon oikeellisuudesta (ks. Eskola & Suoranta 2014). Koska tutkijan subjektiivisuus on aina tutkimusta tehdessä läsnä ja tutkijan persoona vaikuttaa hänen tekemiinsä tulkintoihin (Tuomi & Sarajärvi 2018), on mahdollista, että toinen tutkija olisi saattanut nostaa tutkimuksen aineistosta esiin eri asioita ja tehdä erilaisia tulkintoja ja johtopäätöksiä. Lisäksi, koska tätä tutkimusta toteuttamassa oli vain yksi henkilö ja aineiston analysointi ja johtopäätösten tekeminen olivat yhden ihmisen vastuulla, ei päässyt tapahtumaan tutkijatriangulaatiota (ks. Eskola & Suoranta 2014), mikä voi heikentää tutkimuksen luotettavuutta. Jos tutkimuksen tekoon olisi osallistunut useampi tutkija, oltaisiin voitu saada monipuolisempia näkökulmia ilmiöön. Tutkijana kokemattomuuteni tutkimuksen teosta saattoi näkyä esimerkiksi haastattelutilanteissa, joissa tarvittavia tarkennuksia ei välttämättä osattu aina tehdä, sekä tutkimuksen raportoinnin mahdollisissa puutteissa tai epä johdonmukaisuuksissa. Aineistosta löydettiin kuitenkin tutkimuksen kannalta olennaisia asioita. Tutkimuksen luotettavuutta ja uskot-

tavuutta lisää olennaisesti se, että tutkimustulosten kanssa yhteneväisiä ilmiöitä on löydettävissä aiemmista tutkimuksista (ks. Moilanen & Rähä 2010).

Tulkintojen teossa oleellista tämän tutkimuksen kannalta on ollut myös tutkijana oma suhteeni tutkimukseen osallistujiin ja tutkimusaiheeseen. Sen lisäksi, että tunsin yhden haastateltavista etukäteen, tutkimusaihe itsessään oli itselleni tuttu ja itseäni henkilökohtaisesti kiinnostava. Tutkimusaiheen henkilökohtaisuus tutkijalle on merkittävä tekijä tutkimuksen luotettavuuden kannalta (Eskola & Suoranta 2014). Luokanopettajaopiskelijana olen yliopistopintojeni aikana perehtynyt ohjelmoinnin ja tietotekniikan hyödyntämiseen perusopetuksessa ja myös opettanut ohjelmointia alakoulussa opintoihini sisältyvien opetusharjoittelujen aikana. Omat kokemukseni ohjelmoinnista ja ohjelmoinnin opettamisesta ovat olleet väistämättä jollain tasolla läsnä tätä tutkimusta tehdessä ja ovat saattaneet ohjata tulkintojani. Tämä voidaan nähdä rajoitteena tulkintojen monipuolisuudelle, mutta se on toisaalta ollut mahdollisuus ymmärtää tarkastelun kohteena olevaa ilmiötä syvemmin. Tutkimusaiheen on Eskolan ja Suorannan (2014) mukaan yleisesti hyvä olla tutkijasta kiinnostava, mutta ei kuitenkaan niin läheinen, että etäisyyden ottaminen siihen ei olisi mahdollista monipuolisen tarkastelun toteuttamiseksi. Koen, että olen tutkijana onnistunut saamaan etäisyyttä aineistooni ja pystynyt tarkastelemaan sitä objektiivisesti eri näkökulmista.

Tutkimuksessa yhdeksi keskeiseksi luotettavuutta lisääväksi tekijäksi nousee tutkimuksen relevanssi ja tutkijan kyky tuottaa relevanttia tietoa. Tulee huomioida, että tutkimuksella on käytännön kannalta merkitystä (Eskola & Suoranta 2014). Tämän tutkimuksen aihe on tällä hetkellä hyvin ajankohtainen. Tutkimuksen relevanssi voidaan nähdä siinä, että se tuo esiin ohjelmoinnin opetuksen toteuttamiseen liittyviä seikkoja, jotka kuvastavat ohjelmoinnin opetuksen toteutumista suhteessa opetussuunnitelman vaatimuksiin sekä ohjelmoinnin opetuksen edellytyksiä, kuten opettajien ja koulujen valmiuksia ja resursseja sekä opetukseen tarvittavaa tukea.

Tutkimuksen teossa pyrittiin noudattamaan hyviä tieteellisiä käytäntöjä, kuten tarkkaa raportointia, rehellisyyttä ja eettisyyttä sekä tutkimusmenetelmien valinnassa että tutkimuksen toteutuksessa ja raportoinnissa, mikä lisää tut-

kimuksen luotettavuutta (ks. Tuomi & Sarajärvi 2018). Tutkimuksen tarkka raportointi ja läpinäkyvä kuvaus välittävät lukijoille kokonaisvaltaisen kuvan tutkimuksen eri vaiheista. Tutkimus on tällöin myös mahdollista toistaa. Lisäksi tutkimuksen raportoinnissa esiin on tuotu rehellisesti myös havaitut epäkohdat. Tutkimustulokset esitetään tukeutuen aineiston kuvailuun ja tehdyt tulokset perustellaan haastateltavien autenttisilla sitaateilla, jolloin lukijan on mahdollista päästä käsiksi aineistoon paremmin ja ymmärtää tutkittavien ajatuksia (ks. Patton 2015). Tutkimuksen raportoinnissa on viitattu asianmukaisesti ja laajasti aiempiin tutkimustuloksiin. Lisäksi tutkimuksen kaikissa vaiheissa on huolehdittu eettisistä kysymyksistä ja tutkittavien suojasta.

Jatkotutkimusmahdollisuuksia. Tämä tutkimus keskittyi opettajien käsityksiin ja kokemuksiin ohjelmoinnin opettamisesta. Tutkimus vahvisti pitkälti aiempia tutkimustuloksia aiheesta. Tulevaisuudessa tulee tehdä lisää tutkimusta ohjelmoinnin opetuksesta alakoulukontekstissa, jolloin aihetta voitaisiin tutkia syvällisemmin, sillä tämän tutkimuksen tulokset ovat lähinnä aihetta kartoittavia. Tutkimusten olisi hyvä kattaa suurempi tutkimusjoukko, jotta pystyttäisiin paremmin tekemään yleistyksiä. Tämän tutkimuksen tuloksia ajatellen tulevaisuudessa olisi oleellista kohdistaa kiinnostusta siihen, kuinka moni opettaja Suomessa todellisuudessa opettaa ohjelmointia opetussuunnitelman mukaisesti, ja tarkastella, millaista tukea opettajat tarvitsevat ohjelmoinnin opetuksen toteuttamiseen. Tärkeää olisi etsiä syitä siihen, miksi kaikki opettajat eivät opeta ohjelmointia, ja tarjota ratkaisua tähän ongelmaan, esimerkiksi tarjoamalla lisää mahdollisuuksia osallistua täydennyskoulutukseen. Lisäksi tarvitaan enemmän tutkimusta siitä, millälaiset opetusvälineet ja -materiaalit ovat sopivia alakoulun ohjelmoinnin opetukseen ja miten niitä pystytään innovatiivisesti ja monipuolisesti hyödyntämään, sekä miten niiden avulla ohjelmoinnin opetusta pystyisi tarkoituksenmukaisesti integroimaan eri oppiaineiden opetukseen. Olisi myös hyvä tarkastella perusteellisemmin, millaisia taitoja oppilaat oppivat ohjelmoinnin opetuksessa.

Tämän tutkimuksen tuloksia voitaisiin hyödyntää tulevaisuudessa opettajien tukemisessa ohjelmoinnin opetuksen toteuttamisessa. Tulokset voivat toimia pohjana jatkotutkimuksille aiheesta, tarjoten näkökulmaa opettajien käsi-

tyksiin ja kokemuksiin. Lisäksi niin opettajat, opettajaopiskelijat kuin täydennyskoulutuksen järjestäjätkin voivat halutessaan hyödyntää tässä tutkimuksessa esiin tuotuja ohjelmoinnin opetuksen resursseja. Jatkossa keskeistä olisi tukea luokanopettajia ohjelmoinnin opetuksessa ja huolehtia siitä, että jokaisella tulevalla sekä kentällä jo työskentelevällä luokanopettajalla olisi tarvittavat valmiudet ja mahdollisuudet opettaa ohjelmointia opetussuunnitelman velvoittavalla tavalla. Tällöin peräänkuulutetaan valmiuksia ja resursseja sekä opettajilta että kouluilta. Toivon luokanopettajien pystyvän ajattelemaan ohjelmoinnista tulevaisuudessa: ”Vaikka en olisikaan koodivelho, osaan opettaa ohjelmointia, ja tiedän, että koodaaminen on muutakin kuin pelkkää tietokonenippiä.”

LÄHTEET

- van Aalderen-Smeets, S. I., Walma van der Molen, J. H. & Asma, L. J. 2012. Primary teachers' attitudes toward science: A new theoretical framework. *Science Education* 96 (1), 158–182.
- Alasuutari, P. 2011. *Laadullinen tutkimus 2.0*. Tampere: Vastapaino.
- Albirini, A. 2006. Teachers' attitudes toward information and communication technologies: the case of Syrian EFL teachers. *Computers & Education* 47 (4), 373–398.
- Alimisis, D., Moro, M., Arlegui, J., Pina, A., Frangou, S. & Papanikolaou, K. 2007. Robotics & Constructivism in Education: the TERECOP project. Konferenssitys. 11th European Logo Conference, Bratislava, Slovakia 2007.
- Armoni, M. 2011. Computing in schools - The nature of CS in K-12 curricula: The roots of confusion. *Featured columns. ACM Inroads* 2 (4), 19–20. <http://dx.doi.org/10.1145/2038876.2038883>. Luettu 2.9.2018.
- Atmatzidou, S. & Demetriadis, S. 2016. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems* 75 (B), 661–670.
- Balanskat, A. & Engelhardt, K. 2015. Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europe. Brysseli: European Schoolnet. <http://www.eun.org/fi/resources/detail?publicationID=661>. Luettu 1.9.2018.
- Barr, V. & Stephenson, C. 2011. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *ACM Inroads* 2 (1), 48–54.
- Bawden, D. 2008. Origins and concepts of digital literacy. *Teoksessa C. Lankshear & M. Knobel (toim.) Digital Literacies: Concepts, Policies and Practices*. New York, NY: Peter Lang, 17–32.
- Berry, M. 2013. Computing in the national curriculum. A guide for primary teachers. *Computing at school*. <http://www.computingschool.org.uk/data/uploads/CASPrimaryComputing.pdf>. Luettu 10.9.2018.
- Bers, M. U., Flannery, L., Kazakoff, E. R. & Sullivan, A. 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education* 72, 145–157.

- Binkley, M., Erstad, O., Herman J., Raizen, S. Ripley, M., Miller-Ricci, M. & Rumble, M. 2012. Defining twenty-first century skills. Teoksessa P. Griffin, B. McGaw & E. Care (toim.) *Assessment and teaching of 21st century skills*. New York, NY: Springer, 17–66.
- Brennan, K. & Resnick, M. 2012. New frameworks for studying and assessing the development of computational thinking. Konferenssiesitys. 2012 Annual American Educational Research Association meeting. <https://scholar.harvard.edu/kbrennan/publications/new-frameworks-studying-and-assessing-development-computational-thinking>. Luettu 10.9.2018.
- Brown, N. C. C., Kölling, M., Crick, T., Peyton Jones, S., Hymphreys, S. & Sentance, S. 2013. Bringing computer science back into schools: lessons from the UK. Konferenssiesitys. 44th ACM Technical Symposium on Computer Science Education 2013. <http://dx.doi.org/10.1145/2445196.2445277>. Luettu 2.10.2018.
- Bruckman, A. S. 1997. *MOOSE Crossing: Construction, community, and learning in a networked virtual world for kids*. Massachusetts Institute of Technology. Department of Architecture. Program in Media Arts and Sciences. Väitöskirja.
- Bruckman, A., Jensen, C. & DeBonte, A. 2002. Gender and programming achievement in a CSCL environment. Konferenssiesitys. 2002 Conference on Computer Support for Collaborative Learning: Foundations for a CSCL community, Boulder, CO.
- Buckingham, D. 2007. *Beyond technology: Children's learning in the age of digital Culture*. Cambridge: Polity Press.
- Burke, Q. 2012. The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education* 4 (2), 121–135.
- Cavas, B., Cavas, P., Karaoglan, B. & Kislak, T. 2009. Study on science teachers' attitudes toward information and communication technologies in education. *The Turkish Online Journal of Educational Technology - TOJET* 8 (2), 20–32.
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X. & Eltoukhy, M. 2017. Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education* 109, 162–175.
- Christian, J. 2016. *Minecraft-ohjelmointikirja*. Alkuperäisjulkaisu How to code in Minecraft. Suom. V.-P. Ketola. Helsinki: Readme.fi.
- Code.org. 2018. Hour of Code. <https://hourofcode.com/uk>. Luettu 10.10.2018.

- Common Sense Education. 2018. Common Sense.
<https://www.commonsense.org/education/>. Luettu 10.10.2018.
- CS Unplugged. 2018. <https://csunplugged.org/>. Luettu 15.10.2018.
- Denner, J., Werner, L. & Ortiz, E. 2012. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education* 58 (1), 240–249.
- diSessa, A. A. 2000. *Changing minds: Computers, learning, and literacy*. Cambridge: MIT Press.
- Durak, H. Y. & Saritepeci, M. 2018. Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education* 116, 191–202.
- Eskola, J. & Suoranta, J. 2014. *Johdatus laadulliseen tutkimukseen*. 10. painos. Tampere: Vastapaino.
- Esteves, M., Fonseca, B., Morgado, L., & Martins, P. 2011. Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology* 42 (4), 624–637.
- EU 2006/962/EC. Recommendation of the European Parliament and of the Council of 18 December 2006 on key competences for lifelong learning. *Official Journal of the European Union* 49, L 394/10, 30.12.2006. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32006H0962>. Luettu 6.11.2018.
- Fessakis, G., Gouli, E. & Mavroudi, E. 2012. Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education* 63, 87–97.
- Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P. & Resnick, M. 2013. Designing ScratchJr: Support for early childhood learning through computer programming. *Konferenssiesitys. 12th International Conference on Interaction Design and Children 2013*.
<http://dx.doi.org/10.1145/2485760.2485785>. Luettu 10.10.2018.
- García-Peñalvo, F.J. 2016a. Template for TACCLE 3 resources reviewing.
<https://dx.doi.org/10.6084/m9.figshare.3545033.v1>. Luettu 4.11.2018.
- García-Peñalvo, F. J. 2016b. What computational thinking is. *Journal of Information Technology Research* 9 (3), 5–13.
- García-Peñalvo, F. J. 2018. Computational thinking. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* 13 (1), 17–19.

- García-Peñalvo, F. J. & Mendes, A. J. 2018. Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior* 80, 407–411.
- García-Peñalvo, F. J., Rees, A. M., Hughes, J., Jormanainen, I., Toivonen, T. & Vermeersch, J. 2016. A survey of resources for introducing coding into schools. *Konferenssiesitys. Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality 2016*.
<http://dx.doi.org/10.1145/3012430.3012491>. Luettu 4.10.2018.
- George, M. 2017. Two-thirds of teachers 'don't have the skills or tools to teach coding'. *Tes*, 25.10.2017. <https://www.tes.com/news/two-thirds-teachers-dont-have-skills-or-tools-teach-coding>. Luettu 15.10.2018.
- Graczyńska, E. 2010. ALICE as a tool for programming at schools. *Natural Science* 2 (2), 124–129.
- Gough, P. B. & Tunmer, W. E. 1986. Decoding, reading and reading disability. *Remedial and Special Education* 7 (1), 6–10.
- Harlow, D. B., Dwyer, H., Hansen, A. K., Hill, C., Iveland, A., Leak, A. E. & Franklin, D. M. 2015. Computer programming in elementary and middle school: Connections across content. *Teoksessa M. J. Urban & D. A. Falvo (toim.) Improving K-12 STEM education outcomes through technological integration*. Hershey, PA: IGI Global, 337–361.
- Harsu, M. 2005. *Ohjelmointikielet. Periaatteet, käsitteet, valintaperusteet*. Helsinki: Talentum.
- Harold the Robot. CS Unpluggedin verkkosivu.
<https://classic.csunplugged.org/harold-the-robot-2/>. Luettu 1.10.2018.
- Hart, M. L. 2010. Making contact with the forgotten k-12 influence: are you smarter than your 5th grader? *Konferenssiesitys. 41st ACM Technical Symposium on Computer Science Education 2010*.
<http://dx.doi.org/10.1145/1734263.1734349>. Luettu 4.10.2018.
- Harviainen, J. T., Meriläinen, M. & Tossavainen, T. 2013. *Pelikasvattajan käsikirja*. <http://www.pelikasvatus.fi/pelikasvattajankasikirja.pdf>. Luettu 1.10.2018.
- Hietikko, P., Ilves, V. & Salo, J. 2016. *Askelmerkit digiloikkaan. OAJ:n julkaisusarja 3:2016*.
<https://www.oaj.fi/cs/oaj/OAJn%20askelmerkit%20digiloikkaan>. Luettu 10.10.2018.
- Hiltunen, E., Hiltunen, E. & Hiltunen, O. 2015. *Dibitassut – Matka ohjelmoinnin maailmaan*. Helsinki: Readme.fi.

- Hirsjärvi, S. & Hurme, H. 2008. Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Helsinki: Yliopistopaino.
- Hirsjärvi, S., Remes, P. & Sajavaara, P. 2016. Tutki ja kirjoita. 21. painos. Helsinki: Tammi.
- Hockly, N. 2012. Digital literacies. *ELT Journal* 66 (1), 108–112.
- Honey, M., McMillan Culp, K. & Carrigg, F. 2000. Perspectives on technology and education research: Lessons from the past and present. *Journal of Educational Computing Research* 23 (1), 5–14.
- Hämeenaho, M. 2016. Alakoululaisten käsityksiä ohjelmoinnin opiskelusta. Oulun yliopisto. Kasvatustieteiden tiedekunta. Opettajankoulutuslaitos. Pro gradu -tutkielma.
- Hyvönen, M., Lappalainen, V. & Lakanen, A.-J. 2013. Ohjelmointi 1 : C#. 3., korjattu painos. Luentomoniste 17. Jyväskylän yliopisto. Tietotekniikan laitos. <http://urn.fi/URN:ISBN:978-951-39-4859-7>. Luettu 1.10.2018.
- Israel, M., Pearson, J. M., Tapia, T., Wherfel, Q. M. & Reese, G. 2015. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education* 82, 263–279.
- Jacobsen, H. 2015. Digital experts say coding leads to empowerment (and jobs). EURACTIV, 15.10.2015. <https://www.euractiv.com/section/social-europe-jobs/news/digital-experts-say-coding-leads-to-empowerment-and-jobs/>. Luettu 16.9.2018.
- Järvinen, A. 1999. Digitaaliset pelit ja pelikulttuurit. Teoksessa A. Järvinen & I. Mäyrä (toim.) Johdatus digitaaliseen kulttuuriin. Tampere: Vastapaino, 165–184.
- Karakainen, M.-T., Karakainen, S.-S., Tanhua-Piiroinen, E., Viteli, J., Syvänen, A. & Kivinen, A. 2017. Digiajan peruskoulu 2017 – Tilannearvio ja toimenpidesuositukset. Valtioneuvoston selvitys- ja tutkimustoiminta. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 72/2017.
- Kahn, K., Sendova, E., Sacristán, A. I. & Noss, R. 2011. Young students exploring cardinality by constructing infinite processes. *Technology, Knowledge and Learning* 16 (1), 3–34.
- Kaijanaho, A.-J. 2010. Ohjelmointikielten periaatteet. Luentomoniste 16. Jyväskylän yliopisto. Tietotekniikan laitos.
- Kaila, E. 2018. Utilizing educational technology in computer science and programming courses: Theory and practice. TUCS Dissertations 260. University of Turku. Department of Future Technologies. Väitöskirja. <http://urn.fi/URN:ISBN:978-952-12-3677-8>. Luettu 4.11.2018.

- Kalelioğlu, F. 2015. A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior* 52, 200–210.
- Karkimo, A. 2018. Nyt on masentava ennuste: ”Suomesta tulossa ohjelmistoalan halpatuotantomaa, 10 000 koodaria palkattava ulkomailta”. *Tivi*, 6.4.2018. https://www.tivi.fi/Kaikki_uutiset/nyt-on-masentava-ennuste-suomesta-tulossa-ohjelmistoalan-halpatuotantomaa-10-000-koodaria-palkattava-ulkomailta-6719049. Luettu 14.9.2018.
- Karvonen, V.-P. & Laukka, P. 2016. Suomalaisten opettajien asenteita ja valmiuksia ohjelmoinnin opetukseen. Oulun yliopisto. Kasvatustieteiden tiedekunta. Kasvatustieteen laitos. Pro gradu -tutkielma.
- Kazimoglu, C., Kiernan, M., Bacon, L. & Mackinnon, L. 2012. A serious game for developing computational thinking and learning introductory computer programming. *Procedia – Social and Behavioral Sciences* 47, 1991–1999.
- Kelleher, C., Pausch, R. & Kiesler, S. 2007. Storytelling Alice motivates middle school girls to learn computer programming. Konferenssiesitys. 2007 SIG-CHI Conference on Human Factors in Computing Systems. <http://dx.doi.org/10.1145/1240624.1240844>. Luettu 2.10.2018.
- Konttinen, M. 2015. Koodaus puskee peruskoulun opetussuunnitelmaan – ministeriö turvautuu yritysten apuun. *YLE*, 20.2.2015. <https://yle.fi/uutiset/3-7817538>. Luettu 16.9.2018.
- Koodi2016. 2014. Ensiapua ohjelmoinnin opettamiseen peruskoulussa. <http://koodi2016.fi>. Luettu 3.9.2018.
- Koodiaapinen. 2018. Opettajan opas koodaukseen koulussa. <http://koodiaapinen.fi>. Luettu 3.9.2018.
- Koodikirja. 2018. <http://www.koodikirja.fi/>. Luettu 3.9.2018.
- Krippendorff, K. 2013. Content analysis. An introduction to its methodology. 3. painos. Thousand Oaks, CA: SAGE.
- Kuula, A. 2015. Tutkimusetiikka. Aineistojen hankinta, käyttö ja säilytys. Verkkoaineisto. Tampere: Vastapaino.
- Laine, T. 2010. Miten kokemusta voidaan tutkia? Fenomenologinen näkökulma. Teoksessa J. Aaltola & R. Valli. (toim.) Ikkunoita tutkimusmetodeihin II. Näkökulmia aloittelevalle tutkijalle tutkimuksen teoreettisiin lähtökohtiin ja analyysimenetelmiin. 3., uudistettu ja täydennetty painos. Jyväskylä: PS-kustannus, 28–45.
- Lakanen, A.-J. 2016. On the impact of computer science outreach events on K-12 students. *Jyväskylä studies in computing* 236. Jyväskylän yliopisto. Tietotekniikan laitos. Väitöskirja.

- Lappalainen, V. & Viitanen, S. 2012. Ohjelmointi 2. Uudistettu painos. Luentomoniste 11. Jyväskylän yliopisto. Informaatioteknologian tiedekunta. Tietotekniikan laitos. <http://urn.fi/URN:ISBN:978-951-39-4624-1>. Luettu 5.9.2018.
- Larson, S. 2013. Schools aren't teaching kids to code; here's who is filling the gap. ReadWrite, 18.10.2013. <http://readwrite.com/2013/10/18/kids-learn-code-programming/>. Luettu 30.10.2017.
- Lehtonen, S. & Kyllönen, K. 2010. Electrasta Pongiin – Pongista Pleikkaan. Teoksessa K. Kyllönen & S. Lehtonen (toim.) Sähköistä pelikulttuuria – Electrasta Pongiin, Pongista Pleikkaan. Tampere: Mediamuseo Rupriikki, Perinneyhdistys Elektra, 4–9.
- Lievonen, H. 2014. Jokamiesluokan koodausta. Opettaja 6, 4.
- Liukas, L. 2015. Hello Ruby – Maailman paras koodisatukirja. Helsinki: Otava.
- Liukas, L. & Mykkänen, J. 2014. Koodi2016. Ensiapua ohjelmoinnin opettamiseen peruskoulussa. <http://koodi2016.fi>. Luettu 1.10.2018.
- Louden, K. C. 2003. Programming Languages. Principles and Practice. 2. painos. Pacific Grove, CA: Brooks/Cole.
- Luostarinen, A. & Peltomaa, I.-M. 2016. Reseptit OPSin käyttöön. Opettajan opas työssä onnistumiseen. Jyväskylä: PS-kustannus.
- Lye, S. Y., and 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.
- Makkonen, J. & Pyykönen, A. 2018. "Se on mun mielest taas yks tapa rikastuttaa sitä opiskelua ja oppimista, itekki oppii sit uusii juttui.": Alakoulun opettajien käsityksiä ohjelmoinnin opettamisesta. Jyväskylän yliopisto. Kasvatustieteiden tiedekunta. Opettajankoulutuslaitos. Pro gradu -tutkielma.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B. & Resnick, M. 2004. Scratch: A sneak preview. Proceedings of the 2nd International Conference on Creating, Connecting, and Collaborating through Computing. Washington, DC: IEEE, 104–109.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M. & Rusk, N. 2008. Programming by choice: urban youth learning programming with scratch. Konferenssiesitys. 39th SIGCSE Technical Symposium on Computer Science Education 2008. <http://dx.doi.org/10.1145/1352135.1352260>. Luettu 2.10.2018.
- Margolis, J. & Fisher, A. 2002. Unlocking the Clubhouse: Women in Computing. Cambridge, MA: MIT Press.

- Marttala, L. 2017. Tietotekniikan valtakunnallisten oppisisältöjen toteutuminen Keski-Suomen peruskoulujen opetuskäytänteissä. Jyväskylän yliopisto. Informaatioteknologian tiedekunta. Tietotekniikan laitos. Pro gradu -tutkielma.
- Matarić, M. J., Koenig, N. & Feil-Seifer, D. 2007. Materials for enabling hands-on robotics and STEM education. Konferenssiesitys. 2007 AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education, Palo Alto, CA.
- McCulloch, A. W., Hollebrands, K., Lee, H., Harrison, T. & Mutlu, A. 2018. Factors that influence secondary mathematics teachers' integration of technology in mathematics lessons. *Computers & Education* 123, 26–40.
- Metsämuuronen, J. 2009. Tutkimuksen tekemisen perusteet ihmistieteissä. 4. laitos, 1. painos. Helsinki: International Methelp.
- Moilanen, P. & Räihä, P. 2010. Merkitysrakenteiden tulkinta. Teoksessa J. Aaltonen & R. Valli (toim.) Ikkunoita tutkimusmetodeihin II. Näkökulmia aloittelevalle tutkijalle tutkimuksen teoreettisiin lähtökohtiin ja analyysimenetelmiin. Uudistettu ja täydennetty painos. Jyväskylä: PS-kustannus, 46–69.
- Multisilta, J., Niemi, H. & Lavonen, J. 2014. Miten suomalainen koulu valmistaa tulevaisuuteen? Teoksessa H. Niemi & J. Multisilta (toim.) Rajaton luokahuone. Jyväskylä: PS-kustannus, 286–298.
- Niskanen, P. 2015. Scratch-ohjelmoinnin vaikutus 6. luokan oppilaiden käsityksiin itsestä arkisten ongelmien ratkaisijana. Oulun yliopisto. Kasvatustieteiden tiedekunta. Opettajankoulutuslaitos. Pro gradu -tutkielma.
- Norrena, J. 2013. Opettaja tulevaisuuden taitojen edistäjänä. ”Jos haluat opettaa noita taitoja, sinun on ensin hallittava ne itse”. Jyväskylä Studies in Computing 169. Jyväskylän yliopisto. Väitöskirja.
- NRC. National Research Council. 2000. Inquiry and the national science education standards: A guide for teaching and learning. Washington, DC: National Academy Press.
- OPS 2014. Perusopetuksen opetussuunnitelman perusteet. 2014. Helsinki: Opetushallitus.
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A. & Lahmine, S. 2014. Learning basic programming concepts by creating games with Scratch programming environment. *Procedia – Social and Behavioral Sciences* 191 (2015), 1479–1482.
- Owston, R., Wideman, H., Ronda, N. S. & Brown, C. 2009. Computer game development as a literacy activity. *Computers & Education* 53 (3), 977–989.

- Paananen, J. 2005. Tietotekniikan peruskirja. 6. laitos, 1. painos. Jyväskylä: Docendo.
- Papert, S. 1980. *Mindstorms. Children, Computers, and Powerful Ideas*. New York, NY: Basic Books.
- Patton, M. Q. 2015. *Qualitative Research & Evaluation Methods*. 4. painos. Thousand Oaks, CA: SAGE.
- Perusopetuslaki 628/21.8.1998. 2§: Opetuksen tavoitteet.
<https://www.finlex.fi/fi/laki/ajantasa/1998/19980628>. Luettu 5.11.2018.
- Pirhonen, A. 2010. Do we bring up passive conformists or active constructors? Some thoughts about technological determinism and its reflections in Finnish comprehensive school. Teoksessa A. Rasinen & T. Rissanen (toim.) *In the spirit of Uno Cygnaeus – Pedagogical questions of today and tomorrow. 200th anniversary of the birthday of Uno Cygnaeus Symposium*. Jyväskylä: Opettajankoulutuslaitos. Jyväskylän yliopisto, 143–150.
- Remes, M. 2017. TIVIA News: Koodaus tuli kouluihin.
<http://www.tivia.fi/tiviassa-tapahtuu/tivia-news/koodaus-tuli-kouluhin>. Luettu 15.5.2017.
- Resnick, M. 2007. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. Konferenssiesitys. 6th ACM SIGCHI Conference on Creativity & Cognition 2007.
<http://dx.doi.org/10.1145/1254960.1254961>. Luettu 2.10.2018.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, J. & Kafai, Y. 2009. Scratch: programming for all. *Communications of the ACM* 52 (11), 60–67.
- Robertson, J. 2012. Making games in the classroom: Benefits and gender concerns. *Computers & Education* 59 (2), 385–398.
- Román-González, M., Pérez-González, J.-P. & Jiménez-Fernández, C. 2017. Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior* 72, 678–691.
- Räisänen, K. 2017. Töitä olisi heti 9 000 osaavalle koodarille, mutta tekijöitä ei löydy – ”Vaikka maksaisit mitä, se ei muuttaisi tilannetta”, sanoo ohjelmistoyrityksen teknologiajohtaja. *Helsingin Sanomat*, 17.5.2017.
<https://www.hs.fi/talous/art-2000005225501.html>. Luettu 10.9.2018.
- Saaranen-Kauppinen, A. & Puusniikka, A. 2006. *KvaliMOTV – menetelmäopetuksen tietovaranto*. Tampere: Yhteiskuntatieteellinen tietoarkisto.
<http://www.fsd.uta.fi/metelmäopetus/>. Luettu 10.9.2018.

- Saarenpää, A. & Takkinen, K. 1998. Ohjelmoinnin peruskirja. Jyväskylä: Teknolit.
- Saari, E. M., Blanchfield, P. & Hopkins, G. 2016. Computational thinking – A tool to motivate understanding in elementary school teachers. Conference paper. Switzerland: Springer, 348–364. http://dx.doi.org/10.1007/978-3-319-29585-5_20. Luettu 1.9.2018.
- Saarikoski P. 2006. Koneen ja koulun ensikohtaaminen. Suomalaisen ATK-koulutuksen varhaisvaiheet peruskoulussa ja lukiossa. Tekniikan Waiheita 3, 5–19.
- Sáez-López, J.-M., Román-González, M. & Vázquez-Cano, E. 2016. Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education* 97, 129–141.
- Şahin-Kızıl, A. 2011. EFL Teachers attitudes towards information and communication technologies (ICT). Konferenssiesitys. 5th International Computer & Instructional Technologies Symposium, Elazığ, Turkki 2011.
- ScratchEd. 2018. Verkkosivu. <http://scratched.gse.harvard.edu>. Luettu 10.10.2018.
- Sherer, R., Tondeur, J., Siddiq, F. ja Baran, E. 2018. The importance of attitudes toward technology for pre-service teachers' technological, pedagogical, and content knowledge: Comparing structural equation modeling approaches. *Computers in Human Behavior* 80, 67–80.
- Singer, N. 2017. How Silicon Valley Pushed Coding Into American Classrooms. *The New York Times*, 27.6.2017. <https://www.nytimes.com/2017/06/27/technology/education-partovi-computer-science-coding-apple-microsoft.html>. Luettu 10.10.2018.
- Smith, N., Sutcliffe, C. & Sandvik, L. 2014. Code Club: bringing programming to UK primary schools through Scratch. Konferenssiesitys. 45th ACM Technical Symposium on Computer Science Education 2014. <http://dx.doi.org/10.1145/2538862.2538919>. Luettu 2.10.2018.
- Sola, H. 2002. Ohjelmointia ilman koodausta. Teoksessa P. Hietala & S. Ovaska (toim.) Lasten käyttöliittymät. Julkaisusarja B. Tampereen yliopisto. Tietojenkäsittelytieteiden laitos. Tampere: Tampereen yliopisto, 184–194.
- Taussi, S. 2017. Ohjelmoinnin opetus ekaluokkalaisille. Haaga-Helia ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö.
- Thibaut, L., Knipprath, H., Dehaene, W. & Depaepe, F. 2018. The influence of teachers' attitudes and school context on instructional practices in integrated STEM education. *Teaching and Teacher Education* 71, 190–205.

- Tivia. 2015. Suomalaiset opettajat ohjelmoivat maailmanhistoriaa. *Lehdistötiedote*, 30.9.2015. <http://tivia.fi/lehdistotiedote/suomalaiset-opettajat-ohjelmoivat-maailmanhistoriaa>. Luettu 14.9.2018.
- Tracy, S. J. 2013. *Qualitative research methods. Collecting evidence, crafting analysis, communicating impact*. Chichester, UK: Wiley-Blackwell.
- Tuomi, J. ja Sarajärvi, A. 2018. *Laadullinen tutkimus ja sisällönanalyysi*. Uudistettu laitos. Helsinki: Tammi.
- Vahldick, A., Mendes, A. J. & Marcelino, M. J. 2014. A review of games designed to improve introductory computer programming competencies. *Konferenssiesitys*. 2014 IEEE Frontiers in Education Conference. <http://dx.doi.org/10.1109/FIE.2014.7044114>. Luettu 1.11.2018.
- Vatanen, P. 2016. Koodi tulee kouluun - pian jo pienet lapset ymmärtävät ohjelmoinnin salat. *Yle*. <https://yle.fi/uutiset/3-9081859>. Luettu 10.6.2018.
- Vee, A. 2013. Understanding computer programming as a literacy. *LiCS Literacy in Composition Studies*, 1 (2), 42-64.
- Vorderman, C., Woodcock, J., McManus, J., Steele, C. Quigley, C. & McCafferty, D. 2017. *Koululaisen ohjelmointikirja. Ohjelmointi on helppoa!* Suom. V.-P. Keitola. 3. painos. Helsinki: Readme.fi.
- Vos, N., van der Meijden, H. & Denessen, E. 2011. Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education* 56 (1), 127-137.
- Wainwright, M. 2017. *Hei me koodataan pelejä. Koululaisen ohjelmointikirja*. Suom. M. Niemi. Helsinki: Readme.fi.
- Werner, L., Campe, S. & Denner, J. 2012. Children learning computer science concepts via Alice game-programming. *Konferenssiesitys*. 43rd ACM Technical Symposium on Computer Science Education 2012. <http://dx.doi.org/10.1145/2157136.2157263>. Luettu 4.10.2018.
- Wilson, A. & Moffat, D. C. 2010. Evaluating Scratch to introduce younger schoolchildren to programming. *Konferenssiesitys*. 22nd Annual Workshop of the Psychology of Programming Interest Group 2010. <http://www.ppig.org/workshops/ppig-2010-22nd-annual-workshop>. Luettu 10.10.2018.
- Wing, J. M. 2006. Computational thinking. *Communications of the ACM* 49 (3), 33-35.
- Woodcock, J. 2015. *Kivaa koodausta. Opetellaan ohjelmointia*. Suom. M. Heino-nen. Helsinki: Sanoma Media Finland.

- Woodcock, J. 2016. Koululaisen peliohjelmointikirja. Kielenä Scratch. Vaiheittainen visuaalinen opas tietokonepelien tekemiseen. Suom. V.-P. Ketola. Helsinki: Readme.fi.
- Xie, M., Talin, R. & Sharif, S. 2014. The relationship between teachers' knowledge, attitude and belief with the implementation of inquiry-based learning in China. *International Journal of Learning, Teaching and Educational Research* 8 (1), 149–161.
- Yhdenvertaisuuslaki 1325/2014. 6§: Koulutuksen järjestäjän velvollisuus edistää yhdenvertaisuutta. <https://www.finlex.fi/fi/laki/alkup/2014/20141325>. Luettu 5.11.2018.
- Ylimaz, O. & Bayraktar, D. M. 2014. Teachers' attitudes towards the use of educational technologies and their individual innovativeness categories. *Procedia - Social and Behavioral Sciences* 116, 3458–3461.
- Zaharija, G., Mladenović, S. & Boljat, I. 2013. Introducing basic programming concepts to elementary school children. *Procedia - Social and Behavioral Sciences* 106, 1575–1584.

LIITTEET

Liite 1. Lupalomake opettajille

Hyvä _____ koulun opettaja

Olen Hanna Kurkinen ja opiskelen Jyväskylän yliopistossa luokanopettajaksi. Tällä hetkellä olen opinnoissani siinä vaiheessa, että teen pro gradu -tutkielmaani. Olen saanut xx:n kunnalta/kaupungilta kirjallisen tutkimusluvan ja koulunne johtajalta suullisen suostumuksen kerätä aineistoa tutkielmaani varten tässä koulussa. Työni käsittelee **luokanopettajien kokemuksia ohjelmoinnin opettamisesta**. Aineisto on tarkoitus kerätä haastattelemalla alakoulussa työskenteleviä luokanopettajia. Tutkimuksen raportoinnissa ei tulla käyttämään haastatteluun osallistuneiden nimiä eikä koulun tai kunnan nimeä tulla mainitsemaan missään vaiheessa. Kaikki kerätty tieto säilytetään ja käsitellään luottamuksellisesti salasanojen takana. Tietoja, joista henkilöt olisivat tunnistettavissa, ei luovuteta ulkopuolisille.

Pyydän suostumustanne osallistua tutkimukseeni. Kunnioitan myös päätöstänne olla osallistumatta tähän tutkimukseen. Jos haluatte kysyä minulta lisää tutkimuksestani, älkää epäröikö ottaa yhteyttä email: hanna.k.e.kurkinen@student.jyu.fi tai puh. 044 xxxxxxx.

Tutkijana ja haastattelijana annan lupaukseni, että käsittelen saamaani tietoa ja materiaalia luottamuksellisesti ja salassapitovelvollisuudesta kiinni pitäen.

Paikka ja päiväys:

Allekirjoitus:

Nimenselvennys:

Työni ohjaajat Jyväskylän yliopistolla:

Marja-Kristiina Lerkkanen, professori, Dos., KT, marja-kristiina.lerkkanen@jyu.fi
Anna-Maija Poikkeus, varadekaani, professori, PhD, anna-maija.poikkeus@jyu.fi

Tämä osa jää suostumuksen antajalle.

SUOSTUMUS OSALLISTUA HANNA KURKISEN PRO GRADU -TUTKIMUKSEEN AIHEESTA ALAKOULUN OPETTAJIEN KOKEMUKSIA OHJELMOINNIN OPETTAMISESTA

suostun en suostu

_____ Allekirjoitus ja nimenselvennys

Liite 2. Haastattelurunko

Hanna Kurkinen

Pro gradu -tutkielma, 2017-2018

Haastattelukysymykset

Aihe: Luokanopettajien kokemuksia ohjelmoinnin opettamisesta

Kohderyhmä: Alakoulussa työskentelevät luokanopettajat

Taustatiedot: Kerro itsestäsi (*nimi, ikä, luokka, koulutus, erikoistumisaine/-aineet, työkokemus*)

1. Mitä mielestäsi on ohjelmointi? (Millaisia käsityksiä sinulla on ohjelmoinnista?)

2. Mitä ajattelet siitä, että ohjelmointia pitää opettaa peruskoulussa?

- Mitä mieltä olet henkilökohtaisesti uudesta opetussuunnitelmasta?
- Millaisia tunteita sinussa herätti se, että uusi opetussuunnitelma toi ohjelmoinnin perusopetukseen? Miksi? (*Esim. innokkuus, jännitys, jne.*)
- Onko ohjelmoinnin opetus koulussa mielestäsi tärkeää? Miksi/miksi ei?
- Mihin ohjelmoinnin opetusta tarvitaan? Onko siitä mielestäsi jotain hyötyä/haittaa?

3. Oletko opettanut ohjelmointia? Jos olet, miten paljon ja miksi? Jos et ole, miksi et?

- Onko sinulla muuta kokemusta ohjelmoinnista tai ohjelmoinnin opettamisesta? Minkälaista? (*esim. aiempi kokemus, koulutus, harrastukset, työkokemus*)
- Olitko opettanut ohjelmointia jo ennen kuin uusi opetussuunnitelma tuli voimaan?

4. Miten opetat / voisit opettaa ohjelmointia? (*voit kuvailla esimerkiksi jotain opetustilannetta, jossa olet opettanut ohjelmointia*)

- Mitä mielestäsi sisältyy ohjelmoinnin opetukseen?
- Millä työtavoin, työvälineillä, laitteilla tai sovelluksilla olet opettanut / voidaan opettaa ohjelmointia? (*Millaisia tehtäviä, projekteja, menetelmiä?*)
- Miksi käytät ko. työtapoja? (*Mitkä työtavat olet kokenut hyödyllisiksi ja miksi?*)

5. Millaisia valmiuksia sinulla on / pitäisi olla ohjelmoinnin opettamiseen? *(esim. tiedot ja taidot, asenne, resurssit)*

- Minkälainen on oma osaamisesi tieto- ja viestintäteknikassa? Entä ohjelmoinnissa?
- Koetko, että sinulla on riittävät tiedot/taidot ohjelmoinnin opettamiseen?

6. Minkä aiheen tai oppiaineen yhteydessä opetat / voisit opettaa ohjelmointia? *(Integroitko ohjelmointia eri oppiaineisiin?)*

- Miten paljon/usein opetat ohjelmointia?

7. Jos olet opettanut ohjelmointia, millaiseksi olet kokenut ohjelmoinnin opettamisen? *(Miltä se on tuntunut? Esim. helppoa, haastavaa?)*

- Oletko kohdannut haasteita ohjelmoinnin opettamisessa? Jos olet, millaisia?
- Oletko kokenut tarvitsevasi tukea ohjelmoinnin opettamiseen? Missä asioissa?
- Mistä olet saanut tarvittaessa tukea? *(Esim. kollegoilta, työnantajalta, täydennyskoulutuksista, internetistä?)* Millaista tukea olet saanut?
- Millaisia resursseja ohjelmoinnin opettaminen vaatii *(opettajalta/koululta)?* *(Esim. aikaresurssit, opetusvälineet/-materiaalit)*

8. Miten kokemuksesi mukaan oppilaat ovat suhtautuneet / miten luulisit oppilaiden suhtautuvan ohjelmoinnin opetukseen?

- Miten oppilaat kokevat ohjelmoinnin? Miksi? *(Esim. hyödylliseksi, mielekkääksi, vaikeaksi?)*
- Mitä oppilaat ovat oppineet *(ohjelmoinnin opetuksen avulla)?*
- Onko oppilaiden välillä eroa? Jos on, millaisia eroja? *(Esim. erot motivaatiossa, kiinnostuksessa, sukupuolten välillä, taitoerot, jne.)*

9. Mitä näet ohjelmoinnin opetuksen hyödyksi muille taidoille / muulle oppimiselle? *(Onko siitä hyötyä myös muulle oppimiselle, esim. laaja-alainen osaaminen, jne.?)*

- Millaisten taitojen kehittymistä ohjelmoinnin opettaminen tukee? *(Millaisia taitoja oppilaat voivat oppia ohjelmoinnin opetuksen avulla?)*

Lopuksi: Kysymyksiä, kommentteja, lisättävää?

Liite 3. Esimerkkejä resursseista ohjelmoinnin opetukseen

Resursseja ohjelmointiin symboleilla, koodilohkoilla tai tekstimuotoista koodia kirjoittamalla

1. OHJELMOINTI SYMBOLEILLA	2. OHJELMOINTI KOODILOHKOILLA	3. OHJELMOINTI TEKSTIMUOTOISTA KOODIA KIRJOITTAMALLA
Algo Bot	Alice	Chilicorn Code Club
Bee-Bot	A.L.E.X	Codeacademy
Bomberbot	Blockly	Code Avengers
Cargo-Bot	Box Island	Code Combat
CodeSpark Academy with the Foos	Code.org Code.org - Koodaustunti Code.org - Hour of Code Code.org - Code Studio	Code Monkey
Kodu Game Lab	Daisy the Dinosaur	Code Monster
Kodable	Gamestar Mechanic	Gamemaker Studio
Lightbot: Code Hour	Hobscotch: Coding for Kids	Khan Academy
MinecraftEdu	Made with Code	KidsRuby
Move the Turtle	Scratch	Ruby Warrior
Robozzle	Swift Playgrounds	Turtle Academy
Scratch Jr.	Thunkable	Turtle JS / Turtle Roy
SpriteBox: Code Hour	Tynker Tynker - Hour of Code	Vidcode