

Ohjelmointi ja ohjelmoinnillinen ajattelu yläkoulun matematiikan opetuksessa

Pro gradu -tutkielma
Turun yliopisto
Tietotekniikan laitos
Tietojenkäsittelytiede
huhtikuu 2023
Ann-Victoria Vuorinen

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu
Turnitin OriginalityCheck -järjestelmällä.

TURUN YLIOPISTO
Tietotekniikan laitos, Teknillinen tiedekunta

ANN-VICTORIA VUORINEN: Ohjelmointi ja ohjelmoinnillinen ajattelu yläkoulun matematiikan opetuksessa

Pro gradu -tutkielma, 52 s. + 1 liites
Tietojenkäsittelytiede, opettajan erikoistumisala
huhtikuu 2023

Ohjelmointi ja ohjelmoinnillinen ajattelu liitettiin perusopetuksen opetussuunnitelman perusteisiin vuonna 2014. Valtaosa yläkoulun ohjelmoinnin opetuksesta tapahtuu matematiikan oppitunneilla. Tämän tutkimuksen tarkoituksena on tutkia kuinka paljon, ja millä menetelmillä ohjelmointia ja ohjelmoinnillista ajattelua opetetaan matematiikan oppitunneilla yläkoulussa. Lisäksi selvitetään, mitä resursseja koulut tarjoavat opetuksen tukemiseen.

Tutkimus toteutettiin määrällisenä tutkimuksena, mutta mukaan otettiin myös pieni laadullinen osio. Tutkimuksen aineisto kerättiin sähköisen kyselylomakkeen avulla, joka suunnattiin yläkoulun matematiikan opettajille. Aineisto kerättiin vuoden 2022 lopussa ja kyselyyn vastasi kaiken kaikkiaan 48 opettajaa.

Tutkimuksen tulokset paljastivat, että ohjelmointia ja ohjelmoinnillista ajattelua opetetaan matematiikan oppitunneilla kattavasti ja opettajien mukaan riittävä määrä. Tuloksista selvisi, että opetuksessa painotetaan yksilöllistä oppimista ja tekemällä oppimista sekä hyödynnetään laajasti eri oppimateriaaleja sekä ohjelmointisivustoja ja -ympäristöjä. Koulujen tarjoamia resursseja ja laitteita pidetään noin puolissa kouluista riittävinä. Kuitenkin huoli laitteiden toimimattomuudesta sekä opettajien omasta osaamisesta vaikuttavat heikentävästi ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen.

Tämän tutkimuksen pohjalta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen tarvitaan selkeyttä sekä konkreettisia neuvoja ja malleja opetuksen käytäntöihin. Opettajia on tuettava ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa. Muutoksia tulee tehdä opetussuunnitelmaan sekä opettajankoulutukseen ja samalla rahoitusta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen kehittämiseen ja tukemiseen tulee lisätä.

Asiasanat: ohjelmointi, ohjelmoinnillinen ajattelu, yläkoulu, ohjelmoinnin opetus

UNIVERSITY OF TURKU
Department of Computing, Faculty of Technology

ANN-VICTORIA VUORINEN: Programming and computational thinking in middle school mathematics education

Master of Science Thesis, 52 p. + 1 app.p.
Computer Science, Education Track
March 2023

Programming and computational thinking were included in the national core curriculum for basic education in 2014. The majority of middle school programming is taught in mathematics lessons. The purpose of this study is to investigate how much and with which methods of programming and computational thinking are taught in middle school mathematics lessons. It also explores the resources that schools provide to support this teaching.

The study was conducted as a quantitative study, but a small qualitative component was also included. The data for the study was collected through an electronic questionnaire addressed to middle school mathematics teachers. The data was collected at the end of 2022 and a total of 48 teachers responded to the questionnaire.

The results of the study revealed that programming and computational thinking are taught comprehensively in mathematics lessons and, according to the teachers, in a sufficient amount. The results showed that teaching emphasises individual learning and learning by doing and makes extensive use of different teaching materials and programming websites and environments. The resources and equipment provided by schools are considered adequate in about half of the schools. However, concerns about equipment malfunctioning and teachers' skills have a negative impact on the teaching of programming and computational thinking.

Based on this study, there is a need for clarity for the teaching of programming and computational thinking, as well as concrete advice and models for teaching practices. Teachers need to be supported in teaching programming and computational thinking. Changes need to be made to the curriculum and teacher training while funding for the development and support of programming and computational thinking education needs to be increased.

Keywords: programming, computational thinking, middle school, teaching programming

Sisällys

| | | |
|----------|---|-----------|
| 1 | Johdanto | 1 |
| 2 | Ohjelmointi ja ohjelmoinnillinen ajattelu perusopetuksessa | 4 |
| 2.1 | Ohjelmoinnillinen ajattelu..... | 4 |
| 2.2 | Ohjelmointi osaksi perusopetusta | 6 |
| 2.3 | Ohjelmointi perusopetuksen opetussuunnitelman perusteissa | 7 |
| 3 | Ohjelmoinnin opetuksen menetelmät ja työvälineet | 10 |
| 3.1 | Erilaisia työskentelytapoja | 10 |
| 3.2 | Tietokoneettomuus | 11 |
| 3.3 | Pelit ja pelillisuus | 12 |
| 3.4 | Oppikirjat | 12 |
| 3.5 | Ohjelmointiympäristöt ja -sivustot..... | 12 |
| 3.5.1 | Scratch..... | 13 |
| 3.5.2 | VILLE | 14 |
| 3.5.3 | Code.org ja Koodaustunti | 15 |
| 3.5.4 | W3Schools ja Codecademy | 16 |
| 3.6 | Ohjelmitava elektroniikka ja robotit..... | 16 |
| 3.6.1 | micro:bit | 16 |
| 3.6.2 | Arduino ja Raspberry Pi..... | 17 |
| 3.6.3 | Lego Mindstorms | 17 |
| 4 | Aikaisempia tutkimuksia | 19 |
| 4.1 | Ohjelmoinnin ja ohjelmoinnillisen ajattelun tila Euroopan kouluissa vuonna 2016–2021..... | 19 |
| 4.2 | Opettajien ICT-taidot ja asenteet vuonna 2019 | 20 |
| 4.3 | Suomalaisten oppilaiden ja opettajien ohjelmointiosaaminen vuonna 2018 ... | 21 |
| 4.4 | Suomalaisten koulujen digitalisaatio vuosina 2017–2019..... | 22 |
| 5 | Tutkimusmenetelmän ja aineiston esittely | 25 |
| 5.1 | Tutkimuskysymykset..... | 25 |
| 5.2 | Tutkimusmenetelmän kuvaus | 25 |
| 5.3 | Aineiston keruu, analysointi ja kuvaus..... | 26 |

| | | |
|--------------|---|-----------|
| 5.3.1 | Kyselylomake..... | 26 |
| 5.3.2 | Aineiston analysointi..... | 28 |
| 5.3.3 | Tutkimuksen vastanneiden tausta..... | 29 |
| 6 | Tulokset | 33 |
| 6.1 | Opetuksen määrä..... | 33 |
| 6.2 | Opetusmenetelmät ja työskentelyn muodot..... | 36 |
| 6.2.1 | Opetettavat sisällöt | 38 |
| 6.2.2 | Oppimateriaalit ja -ympäristöt..... | 40 |
| 6.2.3 | Tyypillinen ohjelmointiin keskittyvä oppitunti | 41 |
| 6.3 | Opetuksen tukena olevat laitteet ja resurssit..... | 42 |
| 7 | Johtopäätökset ja pohdinta..... | 46 |
| 7.1 | Tulosten tarkastelu | 46 |
| 7.2 | Tutkimuksen rajoitukset ja luotettavuus | 50 |
| 7.3 | Jatkotutkimusehdotukset..... | 51 |
| 8 | Yhteenveto..... | 52 |
| | Lähteet | 53 |
| | Liitteet | |
| | Liite 1. Kyselylomake | |

1 Johdanto

Käytämme päivittäin ohjelmoituja laitteita. Kuljetamme jatkuvasti mukamme älypuhelimme; tuota pientä tietokonetta. Kotonamme robotti-imuri hoitaa kodin siivouksen automaattisesti päivämme aikana, ja työhakemuskin onnistuu jo tekoälyn kirjoittamana. Alati kehittyvä digitaalinen maailma on jo nyt vaikuttanut työelämään ja tulee muuttamaan tulevaisuuden työntekoa. Tämä on eräs useista syistä, miksi ohjelmointi ja ohjelmoinnillinen ajattelu lisättiin osaksi vuonna 2014 julkaistun perusopetuksen opetussuunnitelman perusteisiin.

Ohjelmoinnilliselle ajattelulle on olemassa monta erilaista määritelmää. Useita määritelmiä kuitenkin yhdistää ajatus, jonka mukaan ohjelmoinnillista ajattelua pidetään taitona, jonka avulla voidaan ratkaista monimutkaisia ongelmia (ks. Aho 2012, 832; Wing 2006, 33). Ohjelmoinnillinen ajattelu sisältää muun muassa kyvyn pilkkoa ongelma pienempiin hallittaviin osiin, jotta ongelma voidaan ratkaista esimerkiksi tietokoneen avulla. Ohjelmoinnillisen ajattelun voidaan myös nähdä sisältävän taidon yleistää saatu ratkaisu muotoon, jotta sitä voidaan hyödyntää uuden ongelman ratkaisemisessa. Ohjelmoinnillinen ajattelu on ennen kaikkea taito, jota tarvitaan ohjelmoimissa.

Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tarkoituksena ei ole tehdä jokaisesta oppilaasta ohjelmoijaa. Opetuksen keskiössä on, että jokaiselle oppilaalle annetaan mahdollisuus ymmärtää, mitä eri mahdollisuuksia ohjelmoinnilla on; miten esimerkiksi ympärillä olevat ohjelmoitavat laitteet toimivat ja miten ohjelmointi näyttäytyy meidän jokaisen arkielämässä? Tämän vuoksi erityisesti ohjelmoinnillista ajattelua tulee perusopetuksen opetussuunnitelman perusteiden mukaan opettaa osana kaikkea opetusta ja oppiaineita.

Varsinainen ohjelmoinnin opetus keskittyy pitkälti matematiikan ja käsitöiden oppiaineiden alle. Opetussuunnitelmaan nojautuen ohjelmoinnin opetus lähtee liikkeelle alakoulusta, jossa ohjelmointiin tutustutaan pitkälti leikkien ja pelien kautta. Alakoulun viimeisillä luokilla voidaan opetuksessa hyödyntää jo erilaisia lohkopohjaisia ohjelmointiympäristöjä, joiden käyttöön siirrytään viimeistään yläkoulussa. Lohkopohjaisten ohjelmointiympäristöjen lisäksi yläkoulussa tutustutaan myös tekstipohjaisiin ohjelmointiympäristöihin sekä eri ohjelmointikieliin.

Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetus käynnistettiin yläkouluissa syksystä 2016 alkaen, kun uusi opetussuunnitelma pantiin käytäntöön. Vaikka opetussuunnitelmaan pohjautuen ohjelmointia ja ohjelmoinnillista ajattelua on ehditty kouluissa opettamaan tähän mennessä jo yli kuusi vuotta, kokee moni opettaja sen edelleen vaikeana asiana opettaa (ks. Boccoli ym. 2022; Fagerlund 2021). Lisäksi eri tutkimukset (ks. Tanhua-Piironen, Kaarakainen, S.-S., Kaarakainen, M.-T. & Viteli 2020; Wu ym. 2019) ovat osoittaneet opettajien ohjelmointiosaamisen olevan heikkoa, mikä on lisännyt yhteiskunnallista keskustelua opettajien valmiuksista opettaa ohjelmointia (Mertala, Palsa & Slotte Dufta 2020, 21). Vaikka materiaalia sekä ohjeita ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen löytyy runsaasti, on kuitenkin edelleen epäselvää, miten opetus on kouluissa järjestetty. Erityisesti siitä, miten ohjelmointia ja ohjelmoinnillista ajattelua opetetaan matematiikan oppiaineessa, ei löydy aikaisempaa tutkimusta. Tästä syntyi kiinnostukseni tälle tutkimukselle, joka keskittyy tutkimaan, mitä kouluissa ja matematiikan oppitunneilla tehdään ohjelmoinnin ja ohjelmoinnillisen ajattelun osaamisen edistämiseksi.

Tämän tutkimuksen tarkoituksena on selvittää, kuinka paljon ohjelmointia ja ohjelmoinnillista ajattelua opetetaan yläkoulun matematiikan oppitunneilla. Lisäksi tarkoituksena on tutkia, millä menetelmillä ja työskentelyn muodoilla ohjelmointia ja ohjelmoinnillista ajattelua opetetaan, mitä ohjelmoinnin sisältöjä opetetaan ja mitä materiaaleja sekä oppimisympäristöjä käytetään ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tukena. Viimeisenä tavoitteena on selvittää, mitkä ovat koulujen tarjoamat resurssit ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tueksi ja miten ne vaikuttavat opetukseen. Tutkimus on toteutettu määrällisenä tutkimuksena, jossa on mukana myös pieni laadullinen osa. Aineisto kerättiin sähköisen kyselylomakkeen avulla, johon vastasi joukko yläkoulun matematiikan opettajia. Rajasin tutkimuksen käsittelemään vain yläkoulun matematiikan oppitunteja lukuvuonna 2021–2022.

Tutkielman toisessa luvussa avaan tarkemmin käsitettä ohjelmoinnillinen ajattelu, sekä kerron lisää syitä sille, miksi ohjelmointi ja ohjelmoinnillinen ajattelu on lisätty perusopetuksen opetussuunnitelman perusteisiin. Lisäksi tarkennan, miten ohjelmointi ja ohjelmoinnillinen ajattelu esiintyvät tuoreimmassa perusopetuksen opetussuunnitelman perusteissa. Luvussa kolme kerron erilaisista menetelmistä ja työskentelyn muodoista, joiden avulla voidaan opettaa ohjelmointia ja ohjelmoinnillista ajattelua. Esittelen myös erilaisia ohjelmoinnin opetuksessa käytettäviä välineitä ja ohjelmointisivustoja. Aikaisempia

tutkimuksia ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen liittyen käsittelen luvussa neljä, jonka jälkeen kerron muun muassani käyttämästäni tutkimusmenetelmästä sekä aineiston analysoimisesta luvussa viisi. Luku kuusi sisältää tutkimuksen tulokset, joiden johtopäätökset sekä pohdinta sijoittuu tutkielman lukuun seitsemän. Tutkielman viimeinen luku sisältää tutkimuksen yhteenvedon.

2 Ohjelmointi ja ohjelmoinnillinen ajattelu perusopetuksessa

Ohjelmointi mielletään yksinkertaisimmillaan ohjeiden eli käskyjen antamisena tietokoneelle. Ohjelmoinnillisella ajattelulla taas viitataan laajempaan käsitteeseen kuin mitä pelkkä ohjelmointi pitää sisällään. Tässä luvussa tarkastellaan, mitä ohjelmoinnillisella ajattelulla tarkoitetaan ja minkälaisia erilaisia määritelmiä käsitteelle on muodostunut. Tämän lisäksi luvussa avataan, mitkä syyt johtivat ohjelmoinnin ja ohjelmoinnillisen ajattelun lisäämisen osaksi perusopetuksen opetussuunnitelman perusteita. Luvussa tarkastellaan myös, miten ohjelmointi ja ohjelmoinnillinen ajattelu esiintyvät tuoreimmassa opetussuunnitelmassa.

2.1 Ohjelmoinnillinen ajattelu

Mietitään kakun leipomista. Kakun leipominen tapahtuu noudattamalla reseptiä eli kakun leipomiselle laadittuja toimintaohjeita. Jotta lopputuloksena saadaan maukas kakku, on ohjeiden oltava tarkkoja ja yksiselitteisiä. Ohjeita laadittaessa on esimerkiksi huomioitava kaikki ne tilanteet, jotka voivat johtaa väärään lopputulokseen. On myös huomioitava, että ohjeita noudattava henkilö voi olla ensimmäistä kertaa leipomassa. Täsmällisten ohjeiden laadinnassa tarvitaan siis liuta erilaisia ajattelun taitoja, kuten kykyä tunnistaa ja pilkkoa ongelma pienempiin osiin. Näitä ajattelun taitoja kuvataan termillä ohjelmoinnillinen ajattelu, jota voidaan harjoittaa muun muassa ohjelmoinnilla.

Ohjelmoinnillisen ajattelun teki tunnetuksi yhdysvaltalainen tietojenkäsittelytieteen professori Jeanette M. Wing. Hänen vuonna 2006 julkaisemansa artikkelin mukaan ohjelmoinnillinen ajattelu sisältää ongelmanratkaisua sekä järjestelmien suunnittelua. Wing (2006, 33) kuvaa ohjelmoinnillisen ajattelun olevan rekursiivista ajattelua; kykyä pilkkoa ongelma pienempiin käsiteltäviin osiin. Myös taito tunnistaa ongelmaan liittyvät eri osat sekä kyky ajatella eri abstraktiotasoilla kuuluvat olennaisena osana ohjelmoinnillisen ajatteluun. (Wing 2006, 33.)

Myöhemmin myös muut ovat maininneet ohjelmoinnillisen ajattelun olevan ajatusprosessi, jonka avulla voidaan ratkoa monimutkaisia ongelmia. Muun muassa Barr, Harrison ja Conery (2011, 21) kuvaavat ohjelmoinnillisen ajattelun sisältävän ongelmien muotoilemisen siten, että ongelman ratkaisemiseen voi käyttää tietokonetta ja muita työkaluja. Myös Aho (2012, 832) mainitsee määritelmässään ongelmien muotoilemisen. Muotoilu edellyttää ongelman pilkkomista pienempiin osiin sekä taitoa tarkastella ongelmaa ja sen osia

eri abstraktiotasoilta (Barr ym. 2011, 21; Bocconi ym. 2022, 27–28; Grover & Pea 2013, 39–40). Tämä taas vaatii tietojen loogista järjestelemistä sekä analysointia (Barr ym. 2011, 21).

Ongelmaan löydetty ratkaisut kuvataan Ahon (2012, 832) mukaan laskennallisina vaiheina ja algoritmeina. Myös muut liittävät ohjelmoinnillisen ajattelun määritelmässään algoritmien muodostamisen keskeiseksi ohjelmoinnillisen ajattelun taidoksi (Bocconi ym. 2022, 27; Grover & Pea 2013, 39; Leino ym. 2019, 11). Ratkaisut halutaan lisäksi automatisoida ohjelmoinnillisen ajattelun avulla (Barr ym. 2011, 21). Muun muassa Bocconi ym. (2022, 27) sekä Barr ym. (2011, 21) kuvaavat ohjelmoinnilliseen ajatteluun liittyvän ratkaisujen arvioimisen, yleistämisen sekä niiden siirtämisen monenlaisiin uusiin ongelmiin. Kuten Barr ym. (2011, 21) tiivistävät, ohjelmoinnillisen ajattelun tavoitteena on tunnistaa mahdolliset ratkaisut, analysoida niiden toimivuutta sekä lopuksi toteuttaa ne saavuttaen tehokkain ja vaikuttavin vaiheiden ja resurssien yhdistelmä.

Ohjelmoinnillisen ajattelu on Wingin (2006, 35) mukaan ennen kaikkea tapa, jolla ihmiset ajattelevat. Sen tarkoituksena ei ole saada ihmisiä ajattelemaan kuin tietokoneet ja kuten Wing (2006, 35) korostaakin, ohjelmoinnillinen ajattelu on paljon enemmän kuin tietokoneelle ohjelmointia. Pohjimmiltaan ohjelmoinnillisessa ajattelussa on kyse tietojenkäsittelytieteen mahdollisuuksien ymmärtämisestä (Fagerlund 2021, 20, 120; Grover & Pea 2013, 39; Leino ym. 2019, 11). Fagerlund (2021, 20) kuvaa väitöskirjassaan, miten ”ohjelmoinnillisen ajattelun opetuksen päämääränä on ruokkia ymmärrystä siitä, mitä tietojenkäsittelyllä voidaan tehdä, miten tietokoneet käsittelevät tietoa sekä miten tietojenkäsittelyn erilaisia työkaluja, malleja ja ideoita voidaan käyttää ratkaisemaan erilaisia oikean elämän ongelmia.” Wing (2006, 35) mainitsikin jo vuonna 2006 ohjelmoinnillisen ajattelun olevan perustaito kaikille, jotta voi toimia nyky-yhteiskunnassamme.

Eri määritelmistä huolimatta ohjelmoinnilliselta ajattelulta puuttuu edelleen ydinkuvaus, eikä sille ole löytynyt yhteistä kaikkia tieteenaloilla käytettävää termiä (Bocconi ym. 2022, 24; Fagerlund 2021, 31; Grover & Pea 2013, 20). Ydinkuvauksen puuttumisen ongelmana on, että termi tarkoittaa eri asioita eri ihmisille ja voi siten olla esteenä tieteelliselle tutkimukselle (Aho 2012, 832; Fagerlund 2021, 31). Kirjallisuudessa muun muassa käsitteet algoritmien tai ohjelmallinen ajattelu rinnastetaan usein ohjelmoinnilliseen ajatteluun. Epäselvyyksien välttämiseksi käytän tässä tutkielmassa termiä ohjelmoinnillinen ajattelu viitaten myös sen synonyymeihin.

2.2 Ohjelmointi osaksi perusopetusta

Suomessa opetus pohjautuu opetussuunnitelmiin – perusopetus perusopetuksen opetussuunnitelman perusteisiin. Perusopetuksen opetussuunnitelman perusteet on perusopetuslain ja -asetusten sekä valtioneuvoston asetusten pohjalta laadittu asiakirja, jonka tehtävänä on muun muassa tukea ja ohjata opetuksen järjestämistä sekä koulutyötä (Opetushallitus 2014, 9). Perusopetuksen opetussuunnitelman perusteiden pohjalta opetuksen järjestäjällä, usein kunnilla, on vastuu paikallisen opetussuunnitelman laadinnasta (Opetushallitus 2014, 9). Varsinaisen suunnan yhteiselle koulutyölle antaa paikallinen opetussuunnitelma, sisältäen muun muassa eri oppiaineiden tarkennetut painotukset sekä tuntijaon (Opetushallitus 2014, 12–13). Viitataan jatkossa perusopetuksen opetussuunnitelman perusteisiin lyhyemmin sanalla opetussuunnitelma.

Ohjelmointi ja ohjelmoinnillinen ajattelu tulivat osaksi perusopetusta, kun tuorein opetussuunnitelma julkaistiin vuonna 2014. Samalla uutena lisäyksenä opetussuunnitelmaan lisättiin laaja-alaiset osaamisen kokonaisuudet, joiden tehtävänä on tarjota tietoa ja taitoja yli oppiainerajojen (Opetushallitus 2014, 19–20). Kokonaisuudet eivät siis ole omia oppiaineitaan, vaan niiden sisältämiä sisältöjä tulisi opettaa joustavasti yhdessä muiden aineiden opetusta. Varsinainen uuden opetussuunnitelman mukainen opetus aloitettiin oppilaitoksissa porrastetusti: 1.–6. luokilla opetus aloitettiin vuonna 2016, 7. luokalla vuonna 2017, 8. luokalla vuonna 2018 ja 9. luokalla vuonna 2019 (Opetushallitus 2023).

Syitä ohjelmoinnin ja ohjelmoinnillisen ajattelun lisäämiselle on perusteltu monesta eri näkökulmasta. Mertala ym. (2020, 23) kirjoittavat ohjelmoinnin lisäämisellä olevan taloudellista hyötyä. Teknologian kehitys on jo muuttanut työelämän tarpeita sekä työntekoa (Barr ym. 2011, 22; Leino ym. 2019, 5). Samalla osaavien ohjelmoijien kysyntä työmarkkinoilla on kasvanut (Leino ym. 2019, 5). Ohjelmoinnin ja ohjelmoinnillisen ajattelun lisäämisellä saadaan pitkällä aikavälillä kasvatettua osaavan työvoiman määrää, jolla voidaan paikata olemassa olevaa työvoimapulaa (Mertala ym. 2020, 23). Mertala ym. (2020, 23) kuvastavat ohjelmoinnin opetuksen olevan välttämätöntä Suomen tulevaisuuden ja kilpailukyvyn kannalta. Lisäämisellä on lisäksi havaittavissa myös välitöntä taloudellista hyötyä kuten teknologiayritysten ohjelmoinnin opetukseen käytettävien välineiden ja erilaisten täydennyskoulutuspalveluiden myymisen valtioille sekä kunnille (Mertala ym. 2020, 23).

Ohjelmoinnin ja ohjelmoinnillisen ajattelun sisällyttämisenä opetussuunnitelmaan ei ole tarkoitus tehdä kaikista oppilaista ohjelmoijia, vaan sen avulla halutaan tarjota mahdollisuus innostua ohjelmoinnista (Mykkänen & Liukas 2014, 8). Mertala ym. (2020, 22) puhuvat siitä, miten julkisessa keskustelussa ajatus ohjelmoinnista esitetään usein kansalaistaitona. Toimijuus digitalisoituvassa yhteiskunnassa vaatii taitoja, joita aikaisemmat sukupolvet eivät ole tarvinneet (Mertala ym. 2020, 22). Fagerlund (2022a) tuo ilmi miten ”ohjelmointia ja ohjelmoinnillista ajattelua on syytä opiskella koulussa sen vuoksi, että elämme ohjelmoidussa maailmassa.” Koska ohjelmoinnillinen ajattelu ja ohjelmointi ulottuu yhä laajemmin useille eri aloille, on tärkeää, että oppilaat ymmärtävät tietojenkäsittelyn tuomia mahdollisuuksia ja sovelluskohteita (Fagerlund 2021, 20; Francisco ym. 2017, 852; Leino ym. 2019, 5). Kuten Francisco ym. (2017, 852) painottavat, tulisi opiskelijoiden oppia ohjelmoinnillisen ajattelun keskeiset taidot, sillä ne voivat olla hyödyllisiä minkä tahansa uran kannalta. Grover ja Pea (2013, 40) tuovat lisäksi esiin, että ne, joilla on ohjelmoinnillista osaamista, ovat yhteiskunnan näkökulmasta paremmassa asemassa kuin ne, joilla osaamista ei ole. Joissain tapauksessa ohjelmoinnin kiinnostuksen herättäminen jo peruskoulussa voi myöhemmin johtaa jatkokouluttautumiseen tai työskentelyyn ammattiohjelmoijana (Fagerlund 2021, 20).

Perusteluita lisäykselle on haettu myös muualta maailmalta. Ohjelmointia, robotiikkaa, tietotekniikkaa ja tietojenkäsittelyä on jo sisällytetty muiden maiden opetussuunnitelmiin (Fagerlund 2022a). Yhä useampi maa haluaa lisäksi edistää ohjelmoinnillisen ajattelun taitojen oppimista (Bocconi ym. 2022, 4).

2.3 Ohjelmointi perusopetuksen opetussuunnitelman perusteissa

Ohjelmointi esiintyy tuoreimmassa vuoden 2014 opetussuunnitelmassa vahvimmin matematiikan sekä käsitöiden oppiaineiden alla. Lisäksi viitteitä ohjelmoinnillisesta ajattelusta löytyy neljästä laaja-alaisen osaamisen kokonaisuudesta: ajattelu ja oppimaan oppiminen (L1), monilukutaito (L4), tieto- ja viestintäteknologinen osaaminen (L5) sekä (L6) työelämätaidot ja yrittäjäyys (Opetushallitus 2014, 20-24; Mertala ym. 2020, 25). Esimerkiksi vuosiluokkien 7–9 tieto- ja viestintäteknologisen osaamisen kokonaisuuden sisällöissä mainitaan että ”ohjelmointia harjoitellaan osana eri oppiaineiden opintoja” (Opetushallitus 2014, 284).

Matematiikassa ohjelmoinnin ja ohjelmoinnillisen ajattelun opetus aloitetaan jo ensimmäisellä luokalla. Opetussuunnitelman alakoulun matematiikan opetuksen tavoitteena on muun muassa harjaannuttaa ja innostaa oppilasta laatimaan vaihteellaisia toimintaohjeita. Kolmannella luokalla tutustutaan omien ohjelmien suunnitteluun ja toteuttamiseen hyödyntäen graafisia ohjelmointiympäristöjä. Alakoulun ohjelmoinnin opetuksen tavoitteet tiivistyvät kuudennen luokan päättöarvioinnissa, jossa oppilaan hyvän ohjelmointiosaamisen kuvaus on, että ”oppilas osaa ohjelmoida toimivan ohjelman graafisessa ohjelmointiympäristössä.” (Opetushallitus 2014, 129, 235, 239.)

Siirryttäessä yläkouluun opetussuunnitelman painotus siirtyy yhä enemmän kohti oikeita ohjelmointikieliä. Matematiikan opetuksen 7.–9. vuosiluokkien eräänä tavoitteena on ohjata oppilasta soveltamaan matematiikkaa sekä ohjelmointia ongelmien ratkaisemiseen kehittämällä algoritmista ajatteluaan. Tämän lisäksi yläkoulussa harjoitellaan hyviä ohjelmointikäytäntöjä. Peruskoulun päättöarvioinnissa eräänä kriteerinä matematiikan hyvälle osaamiselle (arvosana 8) oppimäärän päättyessä on, että ”oppilas osaa soveltaa algoritmisen ajattelun periaatteita ja osaa ohjelmoida yksinkertaisia ohjelmia”. (Opetushallitus 2014, 375, 379)

Opetussuunnitelmassa ei oteta kantaa siihen, kuinka paljon juuri ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta tulisi oppilaitoksissa järjestää. Matematiikan oppiaineen opetusmäärät pohjautuvat perusopetusasetukseen, jonka mukaan yläkoulussa matematiikan opetukseen käytettävä vähimmäisvuosiviikkotuntimäärä on 7.–9. vuosiluokilla yhteensä 11 oppituntia viikossa (Valtioneuvoston asetus perusopetuslaissa tarkoitetun opetuksen valtakunnallisista tavoitteista ja perusopetuksen tuntijaosta 793/2018). Oppilaitosten tulee paikallisissa opetussuunnitelmissa määrittellä, miten oppitunnit jaetaan eri vuosiluokkien kesken (Opetushallitus 2014, 12). Esimerkiksi Turun kaupungin laatiman paikallisen opetussuunnitelman mukaan matematiikkaa opetetaan Turun kaupungin oppilaitoksissa 7. luokalla kolme ja 8. sekä 9. luokalla neljä oppituntia viikossa (Turun kaupunki 2016, 11). Opetusviikkoja yhden lukuvuoden aikana on yhteensä 38, eli yhteensä matematiikkaa opetetaan vuosiluokkien 7–9 aikana vähintään 418 oppituntia (Valtioneuvoston asetus perusopetuslaissa tarkoitetun opetuksen valtakunnallisista tavoitteista ja perusopetuksen tuntijaosta 793/2018). Keskimäärin matematiikan opetusta tulisi siis järjestää jokaisella yläkoulun vuosiluokalla vähintään 139 oppituntia yhden lukuvuoden aikana.

Tuntimäärien lisäksi opetussuunnitelmassa ei myöskään ota kantaa siihen, miten ohjelmointia tai ohjelmoinnillista ajattelua tulisi opettaa. Suomessa vastuu opetuksesta on opettajilla. He ovat muun muassa vastuussa siitä, että opetus on opetussuunnitelman mukaista, mikä edellyttää opettajien syventymistä paikallisen opetussuunnitelman sisältämiin tavoitteisiin sekä sisältöihin. (Opetushallitus 2014, 34.) Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen vaikuttaa siis vahvasti se, miten paikalliset opetussuunnitelmat on rakennettu yleisen perusopetuksen opetussuunnitelman perusteiden pohjalta. Lisäksi opetukseen vaikuttaa myös koulun tarjoamat resurssit kuten ohjelmistot ja laitteistot sekä opettajien omat valmiudet opettaa ohjelmointia. Tästä johtuen, kuten Tanhua-Piironen ym. (2020, 80) mainitsevat, ei ohjelmoinnin opetus ole vielä saanut vakiintuneita muotoja kouluissa. Opettajilla on vapaus päättää, miten he opettavat ohjelmointia, sekä mitä materiaaleja he valitsevat käytettävän (Tanhua-Piironen ym. 2020, 40). Koska kouluilla ei ole yhtenäistä linjaa ohjelmoinnin opetukselle, opetus oppilaitosten välillä vaihtelee merkittävästi (Leino ym. 2019, 10–11).

3 Ohjelmoinnin opetuksen menetelmät ja työvälineet

Opetuksen tukena olevien menetelmien ja työvälineiden valintaan vaikuttaa opetettava asia. Suomessa opettaja on vastuussa valitsemistaan työskentelytavoista ja oppitunneilla käytettävistä työvälineistä. Tässä luvussa esitellään erilaisia menetelmiä sekä välineitä, joilla voidaan tukea juuri ohjelmoinnin ja ohjelmoinnillisen ajattelun oppimista. Luvussa pureudutaan tarkastelemaan, minkälaisia oppimateriaaleja ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tueksi on olemassa. Lisäksi esitellään eri ohjelmointiympäristöjä sekä ohjelmoitavaa robotiikkaa.

3.1 Erilaisia työskentelytapoja

Työtapojen valintaan vaikuttaa opetukselle ja oppimiselle asetetut tavoitteet sekä oppilaiden tarpeet ja kiinnostuksen kohteet. Monipuolisilla työtavoilla ja arviointimenetelmillä voidaan antaa oppilaille mahdollisuus osoittaa osaamistaan eri tavoilla. (Opetushallitus 2014, 30.) Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa opetusmenetelmien on lisäksi tuettava oppimista niin, että kenestä tahansa voi tulla taitava ohjelmoija (Francisco ym. 2017, 837). Monipuoliset työtavat tuovat iloa ja onnistumisen tunteita oppimiseen (Opetushallitus 2014, 30).

Perusasteella ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa käytettäviä työskentelymuotoja ovat muun muassa leikkisä oppiminen, tekemällä oppiminen, virheistä oppiminen sekä työskentely pienryhmissä. Opiskelijat tutustuvat ohjelmoinnin peruskäsitteisiin leikkisillä aktiviteeteilla, robottien ja visuaalisten ohjelmointiympäristöjen avulla. Ohjelmoimalla robotteja tai rakentamalla omia ohjelmia oppilaat siirtyvät passiivisista teknologian kuluttajista omien digitaalisten laitteiden luojiksi. Opiskelijoiden kannustaminen omien ohjelmien luomiseen onkin Bocconi ym. (2022, 6) mukaan tehokas ohjelmoinnillisen ajattelun kehittämiseen käytetty pedagoginen lähestymistapa. (Bocconi ym. 2022, 6, 64.)

Toinen tehokas lähestymistapa ohjelmoinnillisen ajattelun kehittämiseen on Bocconi ym. (2022, 6) tutkimuksen mukaan työskentely tosielämän ongelmien parissa. Se voi tapahtua pareittain tai ryhmissä, jolloin jokaisella oppilaalla ei tarvitse olla omaa tietokonetta (Mykkänen & Liukas 2014, 71). Työskentely voi muistuttaa projektityöskentelyä ja kuten Tanhua-Piironen ym. (2020, 80) tuovat tutkimuksessaan esiin, ohjelmoinnin ja ohjelmoinnillisen ajattelun opetus onkin usein projektiluontoista. Projektimaisen työskentelyn

lisäksi opetuksessa voidaan hyödyntää koulun ulkopuolisia tahoja kuten yrityksiä tai järjestöjä (Tanhua-Piironen ym. 2020, 80).

Muita työskentelytapoja ohjelmoinnin sekä ohjelmoinnillisen ajattelun opetuksessa on muun muassa käänteinen opetus (en. flipped classroom). Käänteisen opetuksen mallissa perinteinen opetus tapahtuu koulun ulkopuolella eli esimerkiksi videoiden tai interaktiivisten sisältöjen avulla, joihin oppilaat tutustuvat ennen oppituntia. Oppitunneilla syvennyttään kotona läpikäytyyn sisältöön ja laajennetaan oppilaiden osaamista. Opetus on siis nimensä mukaisesti käännetty pääläelleen. Samalla myös opettajan rooli käänteisessä opetuksessa muuttuu tiedon levittäjästä enemmän tiedon välittäjäksi. (Nyssölä & Kumpulainen 2020, 19.)

Edellä mainittujen työskentelymuotojen lisäksi opettaja voi ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa hyödyntää yhteisopettajuutta. Yhteisopettajuus voi olla esimerkiksi yhteisten oppimateriaalien ja tuntisuunnitelmien laatimista tai oppiaineiden opetusvastuun jakamista (Tanhua-Piironen ym. 2020, 61). Yhteisopettajuudella voidaan erityisesti jakaa yksittäisten opettajien harteille kasautuvia haasteita ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen liittyen. Vaikka yhteisopettajuuden toteutuksen muodot vaihtelevat koulujen välillä, on tavan koettu lisäävän avoimuutta ja yhteisöllisyyttä (Tanhua-Piironen ym. 2020, 61).

Kuten Mykkänen ja Liukas (2014, 9) kuvastavat ohjelmoinnin opettaminen peruskoulussa on pitkälti ohjelmoinnillisen ajattelun perusteiden oppimista sekä tekemistä ja harjoittelua. Tutkimusten mukaan paras tapa opettaa ohjelmoinnillista ajattelua on ohjelmoida (Francisco ym. 2017, 836; Leino ym. 2019, 10) ja kuten Mykkänen ja Liukas (2014, 80) tuovat esiin, ei ohjelmointia voi oppia muuten kuin vain itse tekemällä. Ohjelmoinnin opiskelussa ei kuitenkaan vaadita, että oppilas oppii itse ohjelmoimaan (Fagerlund 2022a).

3.2 Tietokoneettomuus

Ohjelmoinnin mielletään usein tapahtuvan tietokoneella. Ohjelmointia ja erityisesti ohjelmoinnillista ajattelua voidaan kuitenkin opettaa ilman tieto- ja viestintäteknologisia laitteita esimerkiksi erilaisten pelien ja leikkien avulla (Leino ym. 2020, 10; Mykkänen & Liukas 2014, 71). Eräs esimerkki tästä on Juha Paanasen kehittämä Robogem lautapeli, jossa oppilas ohjaa nappulaansa pelikortteja muistuttavilla käskykortteilla (Robogem 2023). Toisena tietokoneettomana tapana oppia ohjelmoinnillisen ajattelun ja ohjelmoinnin

peruskäsitteitä on tutustua aiheeseen kirjallisuuden kautta. Esimerkiksi Linda Liukkaan Hello Ruby -kirjasarjassa tutustutaan tarinamuotoisesti ohjelmoinnin ja ohjelmoinnillisen maailmaan (Hello Ruby 2023).

3.3 Pelit ja pelillisuus

Pelit toimivat oivana välineenä opettaa ohjelmointia sekä ohjelmoinnillista ajattelua. Lindbergin, Laineen ja Haarasen (2018, 1991) mukaan markkinoille tuodaan vuosittain useita uusia ohjelmointiin keskittyneitä pelejä. Kaikissa peleissä korostuu se, että pelissä on määritelty oppimistavoitteet, minkä voidaan nähdä olevan edellytys pelin käytölle kouluopetuksessa (Lindberg ym. 2018, 1981). Osa markkinoilla olevista ohjelmointipeleistä, kuten CodeMonkey tai MinecraftEdu onkin suunniteltu tukemaan juuri kouluopetusta (Lindberg ym. 2018, 1991). Varsinaisten ohjelmointipelien lisäksi monet oppimisympäristöt, kuten ViLLE, hyödyntävät pelillisyyttä (ks. Kaila & Kurvinen 2018).

3.4 Oppikirjat

Ohjelmointiin ja ohjelmoinnilliseen ajatteluun keskittyvää teoriaa sekä tehtäviä esiintyy uuden opetussuunnitelman myötä myös oppikirjoissa. Esimerkiksi Sanoma Pron yläkoulun matematiikan oppikirjasarja Kuutiossa ohjelmointia käsitellään jokaisella vuosiluokalla yhden luvun verran (Sanoma Pro 2023). Myös Otavan yläkoulun Ääretön oppikirjasarjassa ohjelmointitehtäviä on sijoitettuna jokaisen vuosiluokan kirjaan muodostaen 3–5 oppitunnin kokonaisuuksia (Otava 2023). Kuitenkaan kolmannen suuren oppikirjakustantamo Editan yläkoulun Säde oppikirjasarja ei sisällyluettelon perusteella käsitellä lainkaan ohjelmointia. Sen sijaan kustantamo tarjoaa ohjelmoinnin lisämateriaalia sähköisenä versiona. (Edita 2023) Editan lisäksi moni muu kustantamo tarjoaa ohjelmointiin ja ohjelmoinnilliseen ajatteluun keskittyneitä painettuja tai sähköisenä versiona olevia lisämateriaaleja (ks. Sanoma Pro 2023). Yhteistä eri kustantamoiden tuottamissa materiaaleissa on, että niiden sisältämät ohjelmointitehtävät ovat pääosin tarkoitettu suoritettavaksi joissain ohjelmointiympäristössä.

3.5 Ohjelmointiympäristöt ja -sivustot

Eräs tunnetuin keino opettaa ohjelmointia ja ohjelmoinnillista ajattelua on käyttää jotakin ohjelmointiympäristöä tai -sivustoa, joka tarjoaa valmiita ohjeita sekä harjoituksia ohjelmoinnin opetuksen tukemiseen. Erilaiset ohjelmointisivustot ja -ympäristöt voidaan

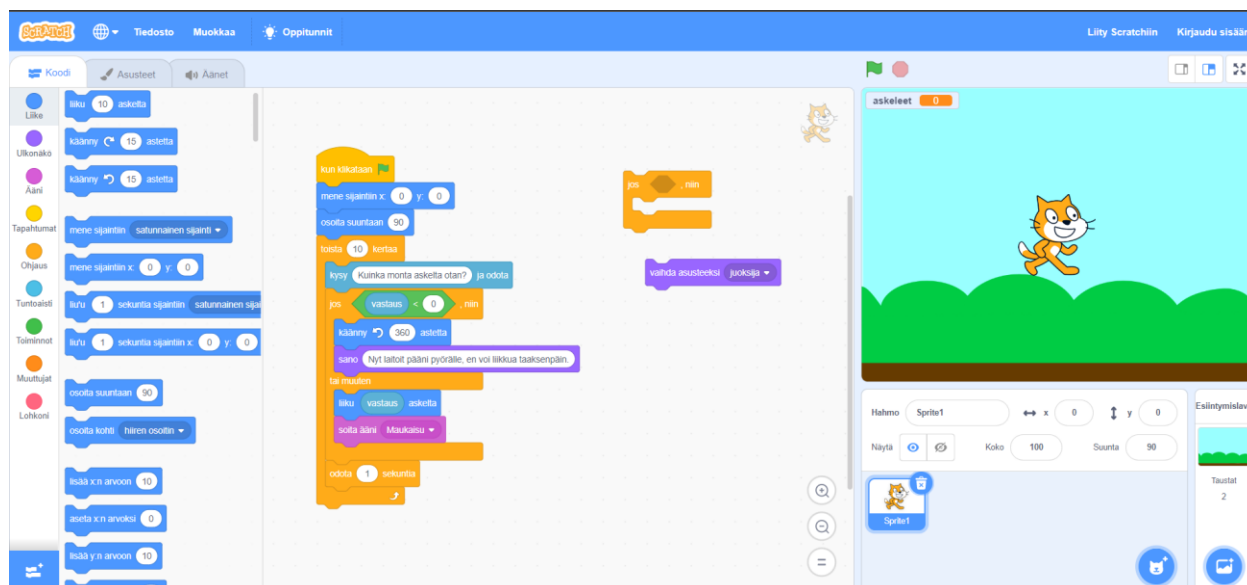
jakaa kahteen eri ryhmään sen pohjalta perustuvatko ne lohkopohjaisien vai tekstipohjaisien ohjelmointikielten käyttöön (Lindberg ym. 2018, 1983). Lohkopohjaisten sivustojen ja -ympäristöjen etuna on niiden helppokäyttöisyys, minkä vuoksi ne sopivat parhaiten aloittelevalle ohjelmoijalle (Grover & Pea 2013, 49; Lindberg ym. 2018, 1983). Kun koodi on esitetty valmiina lohkoina estää se ohjelmointikielten sisältämien kielioppisääntöjen (syntaksi) kanssa kamppailua ja samalla helpottaa ohjelmointisisältöjen oppimista (Lindberg ym. 2018, 1983). Kuten Mykkänen ja Liukas (2014, 90) mainitsevat on lohkopohjaisissa ohjelmointiympäristöissä helpompi välttää virheitä, jotka syövät aloittelijan motivaatiota.

Lohkopohjaiset ympäristöt ja sivustot ovat kuitenkin vähemmän tehokkaita verrattuna tekstipohjaisiin ympäristöihin (Lindberg ym. 2018, 1983). Vaikka ne opettavat tärkeitä ohjelmoinnin ajatusrakenteita ja taitoja, ei niiden avulla pysty luomaan kehittyneempiä ohjelmia (Mykkänen & Liukas 2014, 90). Monimutkaisempien ohjelmien rakentaminen onnistuu tekstipohjaisilla ohjelmointiympäristöillä, jotka käyttävät korkean tason ohjelmointikieliä kuten Pythonia tai Javaa (Grover & Pea 2013, 40-41) Tämän takia ne saattavat sopia paremmin edistyneiden opiskelijoiden käyttöön kuten Lindberg ym. (2018, 1983) mainitsevat. Siirtymisessä oikeisiin ohjelmointikieliin on kuitenkin tärkeää edetä pienin askelin tarjoten oppilaille samalla tunne, että hän edistyy (Mykkänen & Liukas 2014, 100).

3.5.1 Scratch

Scratch on Massachusetts Institute of Technologyn (MIT) Yhdysvalloissa kehittämä ilmainen selaimella käytettävä lohkopohjainen ohjelmointiympäristö (Fagerlund 2022b, 122; Mykkänen & Liukas 2014, 91). Sen sisältämää Scratch ohjelmointikieltä pidetään yhtenä tunnetuimpana lohkopohjaisena ohjelmointikielenä.

Scratch toimii alkeellisen ohjelmoinnin sekä ongelmanratkaisutaitojen opettamiseen (Mykkänen & Liukas 2014, 91). Sen avulla oppilaat voivat ohjelmoiden luoda omia animaatioita, pelejä tai interaktiivista taidetta (Francisco ym. 2017, 846). Ohjelmakoodin kirjoittaminen taas tapahtuu lohkopohjaisen ohjelmointiympäristön tapaan valmiita koodipalikoita raahaamalla (Mykkänen & Liukas 2014, 91). Koodin rakentamiseen sivusto tarjoaa yksinkertaisen käyttöliittymän (KUVA 1), joka tukee esimerkiksi vain tiettyjen koodilohkojen kiinnittymisen toisiinsa (Lindberg ym. 2018 1983). Scratchista on selainversion lisäksi olemassa tablettiversio ScratchJR, joka mahdollistaa ohjelmakoodin kirjoittamisen mobiilisti. (Mykkänen & Liukas 2014, 91).



KUVA 1 Kuvaleike Scratch sivuston työskentelytilasta (otettu 14.3.2023)

3.5.2 ViLLE

ViLLE on sähköinen selaimella käytettävä oppimisjärjestelmä, joka on kehitetty Turun yliopistossa (Kaila & Kurvinen 2018, 6). ViLLE sisältää valmiita ohjelmoinnin oppimiskokonaisuuksia alakoulusta lukioon asti. Esimerkiksi yläkoulun käyttöön ViLLE tarjoaa valmista Python ohjelmointikurssia (KUVA 2). Kurssilla käydään läpi ohjelmoinnin peruskäsitteitä kuten muuttuja, ehtolause, silmukka ja lista (Kaila, Kaarto & Laine 2021, 2). Itse kurssimateriaali koostuu opettajan oppaasta, erilaisista tehtävistä ja niiden malliratkaisuista, luentokalvoista sekä kokeesta (Kaila ym. 2021, 2–3). Kurssin tehtävistä valtaosa tehdään suoraan ViLLE-järjestelmän editoriin, joka sisältää tuen monille ohjelmointikielille (Kaila & Kurvinen 2018, 106). Kurssi kuitenkin sisältää myös harjoitustehtäviä, jotka on tarkoitus tehdä ViLLEn ulkopuolisessa Pythonin koodieditorissa, joko pareittain tai pienissä ryhmissä (Kaila ym. 2021, 2). Opettaja voi vapaasti muokata kurssia omaan käyttöön sopivaksi esimerkiksi lisäämällä omia tai muiden opettajien tekemiä tehtäviä (Kaila ym. 2021, 1).

Oppilaiden tehdessä tehtäviä tallentaa ViLLE jatkuvasti tietoa muun muassa oppilaiden suoriutumisesta. Tätä oppimisdataa opettajat voivat halutessaan hyödyntää esimerkiksi kohdistamalla luokkahuoneessa tukea oppilaille, joilla on haasteita (Kaila & Kurvinen 2018, 7). Oppimisdatan avulla opettaja voi myös tarkastella koko ryhmän suoriutumista ja tarvittaessa muokata oppitunteja (Kaila & Kurvinen 2018, 7). ViLLEn yhtenä

vahvuutena on osan tehtävien automaattinen tarkastaminen, jolla voidaan helpottaa opettajien arviointityötä (Kaila & Kurvinen 2018, 7). Tehtäviä voidaan ratkaista yksin, pareittain tai ryhmissä ja ohjelmointi onnistuu niin lohko- kuin tekstipohjaisia ohjelmointikieliä käyttäen (Kaila & Kurvinen 2018, 106).



KUVA 2 Kuvaleike VILLEN ohjelmoinnin yläkoululle tarkoitetun peruskurssin tehtävästä (otettu 14.3.2023)

3.5.3 Code.org ja Koodaustunti

Code.org on voittoa tavoittelematon internetsivusto, jonka perustamisen tavoitteena oli muun muassa lisätä tietojenkäsittelytiedettä kouluihin sekä kasvattaa nuorten naisten ja vähemmistöjen mahdollisuutta oppia tietojenkäsittelytiedettä (Code.org 2022; Code.org 2023; Francisco 2017, 849). Sivusto sisältää runsaasti ohjelmointiin liittyviä materiaaleja kuten tehtäviä, videoita ja luentoja niin peruskouluun sekä lukio-opetukseen (Code.org 2022). Ohjelmointitehtävissä voidaan käyttää niin lohkopohjaista kuin tekstipohjaista ohjelmointia, minkä lisäksi sivusto tarjoaa harjoituksia, jotka on tarkoitettu tehtäväksi ilman tietokonetta. Yksi suosituista Code.orgin järjestämistä tempauksista on vuosittain järjestetty maailmanlaajuinen Hour of Code -kampanja (suom. koodaustunti) peruskouluun ja lukioon. Kampanjan tarkoitus on innostaa oppilaita ohjelmoimaan tarjoamalla valmiita 45 minuutin mittaisia oppitunteja ohjelmointiin liittyen (Code.org 2022). Opetustuntien avulla oppilaat pääsevät tutustumaan ohjelmointiin muun muassa erilaisten tehtävien, pelien ja leikkien avulla.

3.5.4 W3Schools ja Codecademy

Muita ohjelmointiympäristöjä on muun muassa W3Schools ja Codecademy. Molemmat sivustot ilmaisia, englanninkielisiä ja keskittyvät tekstipohjaisilla ohjelmointikielillä ohjelmoimiseen. Ne tarjoavat interaktiivisia selaimella suoritettavia ohjelmointiharjoituksia, joista W3Schools on keskittynyt tutoriaalipohjaisiin ja Codecademy kurssimuotoisiin harjoituksiin (Codecademy 2022; W3Schools 2022). Harjoituksia on mahdollista tehdä monella eri ohjelmointikielillä kuten HTML, JavaScript, Python ja C#.

Ohjelmointiharjoitusten lisäksi Codecademy tarjoaa mahdollisuuden tutustua tietojenkäsittelyyn ja ohjelmointiin eri aihealueiden kuten verkkosivujen suunnittelun tai koneoppimisen kautta. Sivusto sisältääkin paljon aloittelijalle tarkoitettuja tutoriaaleja, minkä tarkoituksena on johdatella oppilas aiheen pariin. Muun muassa tämän vuoksi Codecademy sopiikin parhaiten itseohjautuvalle oppilaalle (Mykkänen & Liukas 2014, 106).

3.6 Ohjelmitava elektroniikka ja robotit

Oppilaita voidaan innostaa ohjelmoinnin ja ohjelmoinnillisen ajattelun maailmaan myös elektroniikan ja robottien avulla. Ohjelmoimalla elektroniikkaa tai robotteja oppilaat pääsevät tuottamaan ja näkemään, miten heidän luomansa ohjelmat käyttäytyvät fyysisessä maailmassa. Ohjelmien avulla voidaan saada esimerkiksi kytkentälevyllä valot vilkkumaan tai robotti liikkumaan eteenpäin. Tämä auttaa hahmottamaan sekä laajentamaan ohjelmoinnin merkitystä oikeassa maailmassa. Lisäksi Francisco ym. (2017, 847) tuovat esiin, että robottien käyttö vähentää pelkoa käyttää tietokoneita.

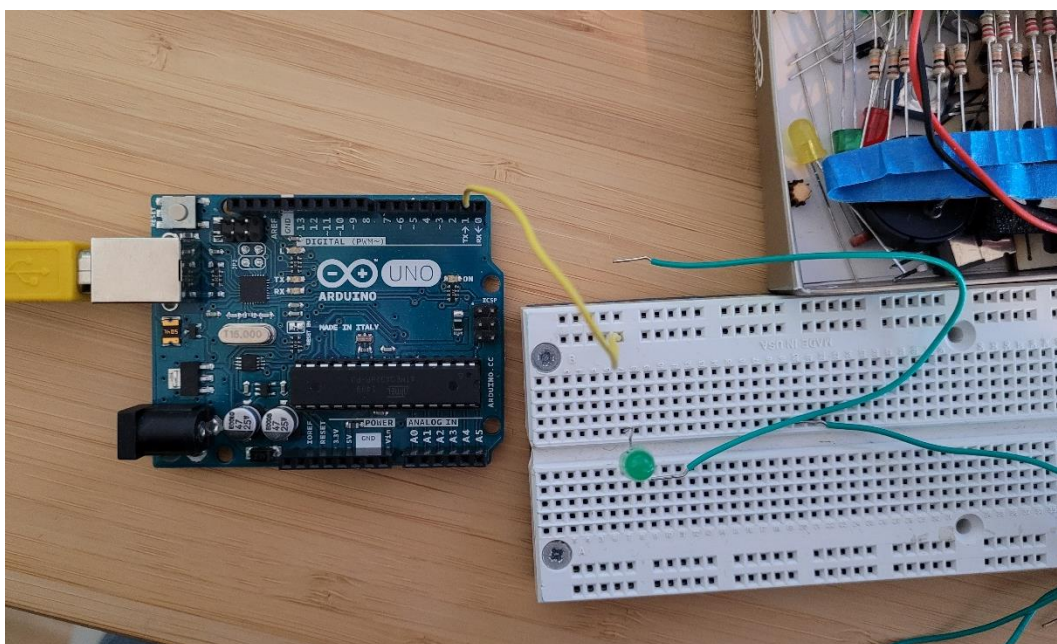
3.6.1 micro:bit

BBC micro:bit on pieni kämmenen kokoinen elektroninen alusta, jota ohjelmoimalla voidaan luoda esimerkiksi erilaisia ääniä tai ohjelmia. Se sisältää muun muassa ohjelmitavia valoja, nappuloita ja erilaisia sensoreita. Uusin versio sisältää edellä mainittujen osien lisäksi myös äänisensoreita, joiden avulla voidaan esimerkiksi mitata äänenvoimakkuutta. (micro:bit 2021.) Micro:bitiä ohjelmoidaan ohjelmakoodin avulla, jonka kirjoittamiseen löytyy erilaisia vaihtoehtoja. Helpoin tapa aloittaa työskentely on käyttää Microsoftin MakeCode editoria, jossa Scratchin tavoin ohjelmointi tapahtuu valmiita ohjelmalohkoja liikuttelemalla. Lisäksi micro:bit internetsivusto tarjoaa valmiita projekteja ja ideoita omien ohjelmien tekoon. Koodia voi kirjoittaa myös oikeilla ohjelmointikielillä käyttäen esimerkiksi micro:bitin

tarjoamaa Python-editoria. Micro:bit on lisäksi yhteensopiva monen muun ohjelmointiympäristön kanssa, jotka mahdollistavat koodin kirjoittamisen myös muilla ohjelmointikielillä. (micro:bit 2022.)

3.6.2 Arduino ja Raspberry Pi

Omia ohjelmia voidaan luoda myös Arduinon tai Raspberry Pi:n avulla. Arduino (KUVA 3) on avoimen lähdekoodin elektroninen alusta, jota voidaan ohjelmoida tietokoneen kautta käyttäen Arduinon omaa ohjelmointikieltä (Arduino 2018). Arduinon kytkentälevyille käyttäjä voi antaa jonkin syötteen ja jonka levy muuttaa sen ulostuloksi. Esimerkiksi jos tiettyä nappia painetaan, levyyn kiinnitetty led-valo syttyy päälle. Raspberry Pi on taas itsessään pieni tietokone, johon voidaan liittää erilaisia komponentteja kuten näytön ja näppäimistön (Mehackit 2022). Sitä ohjelmoidaan pääosin Pythonilla. Molemmissa laitteissa on useita portteja, joihin eri komponentteja voidaan yhdistää. Laitteista Arduino tarjoaa valmiita opetuskäyttöön tarkoitettuja paketteja, jotka sisältävät elektronisen alustan lisäksi myös tarvittavia lisäosia sekä harjoituksia alustan käyttöön (Arduino 2022).



KUVA 3 Arduino ja kytkentälevy

3.6.3 Lego Mindstorms

Lego Mindstorms on tanskalaisen The LEGO Groupin kehittänyt legosarja, jonka avulla voidaan rakentaa erilaisia ohjelmoitavia robotteja. Sarja sisältää laajan valikoiman palikoita,

antureita ja moottoreita. Rakennetun robotin ohjelmointi onnistuu erikseen asennettavassa ohjelmointiympäristössä, jossa ohjelmointi tapahtuu lohkopohjaista ohjelmointikieltä käyttäen. Robotin voi esimerkiksi ohjelmoida reagoimaan antureidensa kautta saatuihin ärsykkeisiin. Francisco ym. (2017, 847) kuvaavat Lego Mindstorms sarjan etuna olevan vuorovaikutteisuus rakennettujen robottien ja oppilaiden välillä. Oppilaat voivat muun muassa tarkkailla robottien liikkeitä ja reaktioita. Tarkkailun ansioista ohjelmakoodissa olevat virheet huomataan helpommin, mikä taas kannustaa oppilaita virheidenkorjaamiseen. (Francisco ym. 2017, 847.)

4 Aikaisempia tutkimuksia

Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta on tutkittu jo pidemmän aikaa. Varsinaisia tutkimuksia, jotka keskittyvät ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tutkimiseen Suomessa on tuoreimman opetussuunnitelman voimaantulon jälkeen tuotettu kuitenkin varsin vähän. Tässä luvussa esitellään neljä tämän tutkimuksen kannalta merkityksellistä tutkimusta, jotka antavat osviittaa tämän hetken ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tilasta niin Suomen kuin muidenkin maiden kouluissa. Tutkimukset käsittelevät muun muassa opettajien valmiuksia ja koulujen resursseja sekä oppilaiden ja opettajien ohjelmointitaitoja.

4.1 Ohjelmoinnin ja ohjelmoinnillisen ajattelun tila Euroopan kouluissa vuonna 2016–2021

European Schoolnet on 34 eurooppalaisen opetusministeriön voittoa tavoittelematon verkosto, jonka tavoitteena on tuoda innovaatioita opetukseen ja oppimiseen (European Schoolnet 2022). Heidän tuorein julkaisunsa vuodelta 2022 jatkaa vuonna 2016 tehtyä tutkimusta muun muassa ohjelmoinnin opetuksesta, antaen päivitetyn yleiskatsauksen ohjelmoinnin ja ohjelmoinnillisen ajattelun tilasta Euroopan maiden peruskouluissa. Tutkimus toteutettiin vuosina 2016–2021 ja siinä oli kaiken kaikkiaan mukana 29 Euroopan maata sekä Singapore. Tutkimus koostui erilaisista tutkimusmenetelmistä kuten kirjallisuuskatsauksesta, kyselytutkimuksesta sekä tapaustutkimuksesta. (Bocconi ym. 2022, 4, 18–22.)

Tutkimuksessa mukana olleista 29 Euroopan maasta yhteensä 25 maassa tietojenkäsittelytieteen peruskäsitteitä oli liitetty osaksi maiden opetussuunnitelmaa. Tietojenkäsittelytieteen peruskäsitteillä viitattiin tutkimuksessa muun muassa abstraktioon, algoritmiseen ajatteluun, automaatioon, hajottamiseen ja yleistämiseen sekä virheidenkorjaukseen. Tapaustutkimukseen osallistuneista 9 maasta (Englanti, Kroatia, Liettua, Norja, Puola, Ranska, Ruotsi, Slovakia ja Suomi) viidessä maassa tietojenkäsittelyä opetettiin omana oppiaineenaan ja neljässä maassa mukaan lukien Suomessa tietojenkäsittelyn perustaitoja käsiteltiin muissa oppiaineissa, pääasiassa matematiikassa. Tietojenkäsittelyn taitojen opettamiseen oli tutkimukseen osallistuvien maiden kesken varattu keskimäärin yksi oppitunti viikossa. Suomessa opetukselle varattu määrä oli keskimääräistä arvoa pienempi – alle yksi oppitunti viikossa. Haasteita ohjelmoinnin opetukseen loi koulutettujen opettajien puute sekä ohjelmoinnin ”kilpaileminen” opetussuunnitelmien sisältämien muiden

painopisteiden kanssa. Vähiten haasteita tuottivat opetusmenetelmät tai resurssien puutos. (Bocconi ym. 2022, 15, 31, 39, 59, 63.)

Ohjelmoinnin yhteydessä esiintyviä käsitteitä olivat tutkimuksen mukaan muun muassa muuttuja, silmukka, ohjelman luominen sekä virheenkorjaus. Yläkoulussa käsiteltiin lisäksi monimutkaisempia käsitteitä kuten tietorakenne ja totuusarvot. Näihin tietojenkäsittelytieteen peruskäsitteisiin tutustuttiin esimerkiksi ohjelmitavien robottien ja lohkopohjaisten ohjelmointiympäristöjen avulla. Opetusmenetelminä käytettiin muun muassa projektityöskentelyä, pelejä, yhteistoiminnallista oppimista, pariohjelmointia sekä yksilöllistä oppimista. Opetuksessa keskityttiin ongelmanratkaisuun ja loogisen ajattelun taitoihin. Näiden lisäksi opettajat korostivat opetuksessaan virheenkorjauksen merkitystä virheiden kautta oppimiseen. Opetussuunnitelmien ytimessä oli, että opiskelijat työskentelevät tosielämän ongelmien parissa ja että heitä kannustetaan luomaan jotain omaa, kuten ohjelmia, sovelluksia tai animaatioita. Oppimisympäristöinä tutkimuksessa nousivat esiin muun muassa Python, Logo ja Arduino sekä Scratch, Micro:bit, ViLLE ja Debras task. (Bocconi ym. 2022, 59, 64, 72.)

4.2 Opettajien ICT-taidot ja asenteet vuonna 2019

Wun ym. (2019, 21) teettämässä tutkimuksessa vertailtiin opettajan asenteita ICT-taitoja ja koodaustaitoja kohtaan viiden eri maan välillä. Suomen lisäksi tutkimuksessa oli mukana Manner-Kiina, Singapore, Taiwan ja Etelä-Korea. Tutkimuksella haluttiin selvittää, mitä mieltä opettajat ovat tieto- ja viestintätekniikan käytöstä kouluissa, mitkä ovat opettajien valmiudet opettaa ohjelmointia sekä minkälaisia käsityksiä opettajilla oli ohjelmointitaitojen opettamisesta. Tutkimus toteutettiin kyselylomakkeen avulla vuonna 2019. (Wu ym. 2019, 21, 24.)

Yksi merkittävimmistä tutkimuksessa nousseista havainnoista oli tutkimukseen osallistuneiden maiden opettajien heikko ohjelmointiosaaminen. Tutkimuksen mukaan tekstipohjaisten kielten kuten Pythonin ohjelmointiosaaminen oli Suomessa melko alhainen. Samalla suomalaisten opettajien käsitykset ja valmiudet tieto- ja viestintätekniikkataidoistaan olivat heikommat verrattuna muihin tutkimukseen osallistuviin maihin. Myös tieto- ja viestintätekniikan käyttöön luokahuoneissa suhtauduttiin kielteisimmin Suomessa. Tästä huolimatta visuaalisten ohjelmointiympäristöjen, kuten Scratchin käyttötaidot olivat Suomessa parhaimmat. Kuitenkin kun tutkittiin kaikkien tutkimuksessa olleiden maiden

ohjelmointitaitoja, saatiin selville, että suurin osa opettajista ei osannut ohjelmoida millään ohjelmointikielellä. Moni suomalainen opettaja vastasi ohjelmoinnin olleen heille täysin tuntematonta. Tutkimuksen mukaan opettajilta ei voida olettaa laadukasta ohjelmointitaitojen opettamista, mikäli heidän omat taitonsa ovat puutteelliset. (Wu ym. 2019, 31–32.)

Kaikkien alueiden opettajat olivat yhtä mieltä siitä, että ohjelmointia opitaan parhaiten kirjoittamalla ohjelmakoodia, visuaalisilla ohjelmointiympäristöillä, rakentamalla robotteja ja osallistumalla koulun ulkopuolisiin kerhoihin. Alle 45-vuotiaat opettajat olivat vahvasti sitä mieltä, että ohjelmointia tulisi opettaa kouluissa opettajan opastuksella. He olivat myös vahvimmin sitä mieltä, että ohjelmointia tulisi opettaa kirjojen ja ohjelmointiin tarkoitettujen sivustojen avulla. Tutkimuksen tulokset viittaavat lisäksi siihen, että suomalaiset opettajat suosivat aktiivisia oppimismenetelmiä, kuten ohjelmakoodin kirjoittamista ohjelmointiympäristössä tai rakentamalla robotteja. Opetusta tapahtui tutkimuksen mukaan myös epävirallisissa oppimisympäristöissä, kuten erilaisissa ohjelmointikerhoissa. (Wu ym. 2019, 32–33.)

Tutkimuksessa mukana olleet opettajat olivat vahvasti sitä mieltä, että ohjelmointitaitoja tarvitaan myös niille oppilaille, jotka eivät pyri ammattiohjelmoijiksi. Opettajat uskovat yleisestä ohjelmointiosaamisesta olevan hyötyä kaikille oppilaille suuntautuisikään tieto- ja viestintäteknikan aloille. (Wu ym. 2019, 31–32.)

4.3 Suomalaisten oppilaiden ja opettajien ohjelmointiosaaminen vuonna 2018

Kansainvälinen nuorten monilukutaidon ja ohjelmoinnillisen ajattelun tutkimus (ICILS, International Computer and Information Literacy Study) arvioi 8. luokkalaisten oppilaiden monilukutaitoa sekä ohjelmoinnillisen ajattelun taitoja. Tuorein tutkimus toteutettiin vuonna 2018, missä myös Suomi oli ensimmäistä kertaa mukana. Vuoden 2018 tutkimus oli samalla ensimmäinen kansainvälinen tutkimus, jossa arvioitiin nuorten ohjelmoinnillisen ajattelun taitoja. (Leino ym. 2019, 5.) Tutkimuksessa ohjelmoinnillinen ajattelu määriteltiin ”yksilön kykyä tunnistaa tosielämään perustuvien ongelmien ohjelmoinnillisia piirteitä sekä arvioida ja kehittää ongelmiin algoritmisia ratkaisuja, jotka voidaan toteuttaa tietokonetta hyödyntäen” (Leino ym. 2019, 11).

Ohjelmoinnillisen ajattelun arviointiin osallistui tutkimukseen osallistuvasta 12 maasta ja kahdesta alueesta vain kahdeksan maata sekä yksi erillinen alue. Tutkimuksen

aineisto kerättiin oppilaskokeen muodossa, johon osallistui yhteensä 2 546 14–15 -vuotiasta suomalaista oppilasta. Kokeessa käytettiin lohkopohjaista ohjelmointiympäristöä, joten oppilailta ei edellytetty tehtävän ratkaisemiseksi ohjelmointikielien osaamista tai koodin kirjoittamistaitoa. Tutkimustuloksissa on otettava lisäksi huomioon, että aineisto kerättiin keväällä 2018, jolloin 8. luokalla opetus pohjautui vielä vanhaan opetussuunnitelmaan. Vasta syksyllä 2018 siirryttiin noudattamaan uutta opetussuunnitelmaa. (Leino ym. 2019, 6, 11, 13, 32.)

Suomi sijoittui tutkimuksen ohjelmoinnillisen ajattelun arvioinnissa kolmanneksi ja oli ainut arviointiin osallistunut maa, jossa tyttöjen pistemäärän keskiarvo oli suurempi kuin poikien. Tutkimuksessa oli määritelty myös kolme suoritustasoa kuvaamaan ohjelmoinnillisen ajattelun taitotasoa: välttävä suoritus, hyvä suoritus sekä erinomainen suoritus. Erinomaisen tason saavutti 20 % suomalaisista oppilaista. Lisäksi vähintään hyvään tasoon ylsi 72 % suomalaisista oppilaista, mikä ylitti kansainvälisen keskiarvon. Välttävälle tasolle sijoittui 28 % suomalaisista oppilaista. (Leino ym. 2019, 19–20). Tuloksia tutkittiin myös alueellisesti, jota varten Suomen aineisto jaettiin viiteen alueeseen. Alueellisesti korkein pistemäärä saavutettiin pääkaupunkiseudulla ja heikoin taas Itä-Suomessa. Alueellisia eroja selitettiin muun muassa alueiden sosioekonomisella taustalla. (Leino ym. 2019, 19–20, 22.)

Tutkimuksessa tutkittiin lisäksi koulujen tieto- ja viestintäteknologisia resursseja, jonka mukaan Suomen tieto- ja viestintäteknologiset resurssit olivat hyvät kansainvälisesti verrattuna. Lähes kaikissa suomalaisissa kouluissa oli käytössä keskeiset teknologiset laitteet ja sovellukset. Noin 66 %:lla kouluista oli lisäksi mahdollisuus robottien käyttöön. Määrällisesti laitteita kouluissa oli keskimäärin yksi tietokone tai tabletti kolmea (1:3) oppilasta kohtaan. Kuitenkin vaihtelu koulujen välillä oli suurta, mistä kertoo esimerkiksi se, että heikoimmin varustelluissa kouluissa yhtä laitetta kohtaan oli jopa 25 (1:25) oppilasta. (Leino ym. 2019, 42.)

4.4 Suomalaisen koulujen digitalisaatio vuosina 2017–2019

Opetus- ja kulttuuriministeriön teettämässä vuonna 2020 julkaistussa tutkimuksessa selvitettiin, miten koulujen digitalisaatio oli edennyt vuosina 2017–2019. Tutkimus oli jatkoa vuoden 2016–2018 Digiajan peruskoulu -hankkeelle ja sen aineistona toimi Kansallisen koulutuksen arviointikeskuksen (Karvi) keräämä aineisto, joka koostui rehtoreiden ja opettajien sekä toisen, viidennen ja kahdeksannen luokan oppilaiden itsearviointikyselyiden

vastauksista sekä opettajien ja yhdeksäsluokkalaisten digitaitoja mittaavan testin tuloksista. (Tanhua-Piironen ym. 2020, 9.)

Tutkimuksesta kävi ilmi, että rehtorit ja koulujen johtajat vastaavat koulujen digitaalisten toimintaympäristöjen resursseista kuten opettajien täydennyskoulutuksista. Digitalisaatiota toteutettiin maltillisesti pedagogiset lähtökohdat edellä, mikä tarkoittaa sitä, että uusia teknologioita hyödynnettiin silloin, kuin sen koettiin olevan tarkoituksenmukaista. Kuitenkin opettajilla on vapaus valita mitä tarjotuista materiaaleista he käyttävät opetuksessaan. Tutkimuksen mukaan 60 %:lla opettajista oli oma henkilökohtainen kannettava tietokone ja hieman yli 50 %:lla oma tabletti. Oppilaille tabletteja oli keskimäärin yksi neljää (1:4) oppilasta kohden ja kannettavia tietokoneita keskimäärin yksi seitsemää (1:7) oppilasta kohden. Lisäksi pöytäkoneita oli keskimäärin yksi reilua 12 (1:12,5) oppilasta kohden. Oppimateriaaleista perinteisiä oppimateriaaleja kuten oppikirjoja ja vihkoja käytettiin opetuksessa digitaalisia resursseja enemmän. Tämä saattoi johtua siitä, että digimateriaaleja pidettiin kalliina verrattuna niiden antamaan opetuksen lisäarvoon. Lisäksi tutkimuksesta selvisi, että opettajat suosivat opetuksessaan valmiita materiaaleja. (Tanhua-Piironen ym. 2020, 40–44, 53.)

Tutkimuksesta kävi ilmi, että opettajien työ oli uuden opetussuunnitelman ja digitalisaation myötä muuttunut monipuolisemmaksi mutta myös haastavammaksi. Kun tietoa on jatkuvasti saatavilla, vie se opettajilta enemmän aikaa hyödyllisen tiedon löytämiseen ja tarkistamiseen. Yhteisopettajuus oli vuosien aikana lisääntynyt ja sen on koettu lisäävän avoimuutta ja yhteisöllisyyttä opetuksessa. Toteutustavat vaihtelivat niin yhteisten oppimateriaalien laatimisella kuin opetusvastuun jakamisena. (Tanhua-Piironen ym. 2020, 61.)

Tutkimuksen sisältämällä ICT-taitotestillä mitattiin muun muassa opettajien ja oppilaiden ohjelmointiosaamista. Opettajien alkeisohjelmointiosaaminen oli testin tulosten mukaan heikkoa. Jopa kaksi kolmesta (2/3) opettajista jäi ICT-taitotestin alkeisohjelmointiosuudessa ilman pisteitä. Erityisen huolestuttavaa tutkimustuloksista oli että, yhdeksäsluokkalaisten alkeisohjelmointiosaaminen koettiin edelleen heikoksi. ICT-taitotestin alkeisohjelmointitehtävistä peräti 89 % ei saanut testistä yhtään pistettä, vaikka tehtävät vaativat lähinnä päättelykykyä, annettujen ohjeiden perusteella selkokielisten kommentojen antamista ja yksinkertaisten laskutoimitusten ratkaisua. (Tanhua-Piironen ym. 2020, 78–79.)

Tutkimuksesta selvisi, ettei ohjelmoinnin opetus ollut saanut kouluissa vakiintuneita muotoja. Yläkoulussa ohjelmointia opetettiin pääosin matematiikan ja käsitöiden oppiaineissa, joissa ohjelmoinnin nähtiin jäävän usein erilliseksi osaksi, eikä täten oppimista koettu kovin mielekkääksi. Lisäksi kouluissa saattoi toimia esimerkiksi digiaiheisia kerhoja, joissa voitiin ohjelmoida. Kerhot eivät kuitenkaan olleet oppilaille pakollisia, jolloin niihin osallistui usein vain aiheesta kiinnostuneita oppilaita. Kaiken kaikkiaan tutkimukseen haastatelluista kymmenestä koulusta neljässä ei ohjelmoinnin opetus toteutunut millään tavalla opettajien puutteellisen osaamisen takia. Kahdessa koulussa opettajat eivät edes tienneet, mitä opetussuunnitelmassa mainittiin ohjelmoinnista. Opettamattomuutta perusteliin myös ajan puutteella. (Tanhua-Piironen ym. 2020, 80–81.)

5 Tutkimusmenetelmän ja aineiston esittely

Tässä luvussa keskitytään kuvaamaan tutkimuksen menetelmiä sekä kulkua. Luvun alussa esitellään tutkimuskysymykset sekä kerrotaan lyhyesti käytetyistä tutkimus- ja aineistonkeruumenetelmistä. Luvussa lisäksi avataan tarkemmin, miten kyselylomakkeen kautta saatua aineistoa analysoitiin ja muokattiin. Lopussa avataan tarkemmin tutkimukseen vastanneiden opettajien taustoja.

5.1 Tutkimuskysymykset

Tämän tutkimuksen tarkoituksena on selvittää, kuinka paljon ohjelmointia sekä ohjelmoinnillista ajattelua opetetaan peruskoulun matematiikan oppitunneilla. Tutkimuksessa selvitetään, millä menetelmillä ja välineillä ohjelmointia ja ohjelmoinnillista ajattelua opetetaan. Lisäksi tavoitteena on saada tietoa siitä, mitkä ovat opettajien käytössä olevat resurssit ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen sekä miten nämä resurssit vaikuttavat opetukseen. Tutkimuskysymykset ovat:

1. Kuinka paljon yläkoulun matematiikan oppitunneilla opetetaan ohjelmointia ja ohjelmoinnillista ajattelua?
2. Millä menetelmillä ja työskentelyn muodoilla ohjelmointia ja ohjelmoinnillista ajattelua opetetaan yläkoulun matematiikan oppitunneilla?
 - a. Mitä ohjelmoinnin sisältöjä opetetaan?
 - b. Mitä ohjelmointiin tarkoitettuja materiaaleja ja oppimisympäristöjä käytetään opetuksen tukena?
3. Mitkä ovat opettajien käytössä olevat resurssit ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tueksi ja miten ne vaikuttavat opetukseen?

5.2 Tutkimusmenetelmän kuvaus

Tutkimus toteutettiin kvantitatiivisena eli määrällisenä kyselylomaketutkimuksena, jonka pienessä osassa hyödynnettiin myös laadullisia menetelmiä. Määrällinen tutkimus perustuu päätelmien tekoon hyödyntämällä numeroita ja tilastoja. Hirsijärven, Remeksen ja Sajavaaran (2009, 140) mukaan määrällisessä tutkimuksessa keskeisessä roolissa ovat muun muassa

aikaisempien tutkimusten johtopäätökset sekä teoriat. Myös aineistonkeruumenetelmän on sovelluttava numeeriseen mittaamiseen. Laadullisia menetelmiä käyttäen taas saadaan tutkittua kohdetta kokonaisvaltaisemmin. (Hirsijärvi ym. 2009, 137, 140, 161.) Lisäksi laadullisen sisällönanalyysin avulla, kuten aineiston kvantifioinnilla, aineistoa voidaan tarkastella määrällisten menetelmien avulla (Tuomi & Sarajärvi 2012, 120).

Kyselylomakkeen käyttö on Vallin (2010, 103) mukaan yksi perinteisimmistä tavoista kerätä tutkimuksen aineisto. Sen avulla voidaan kerätä laaja tutkimusaineisto, minkä Hirsijärvi ym. (2009, 195) näkevät yhtenä kyselytutkimuksen eduista. Kyselyn tyypillisiin piirteisiin kuuluu tiedon kerääminen standardoidussa muodossa joukolta ihmisiä. Olennainen osa kyselytutkimusta on lisäksi perusjoukon sekä niistä poimitun otoksen määrittely. Mitä suurempi otos on, sitä tarkemmin tutkimuksesta saadut tulokset vastaavat koko perusjoukkoa. (Hirsijärvi ym. 2009, 134, 180, 195.) Tässä tutkimuksessa perusjoukon muodostivat kaikki yläkoulun matematiikan opettajat, ja otos koostui kyselyyn vastanneista yläkoulun matematiikan opettajista.

5.3 Aineiston keruu, analysointi ja kuvaus

5.3.1 Kyselylomake

Tutkimuksen aineisto kerättiin Webropol -ohjelmistolla luodulla sähköisellä kyselylomakkeella (liite 1). Kysely koostui yhteensä 19 kysymyksestä, jotka keskittyivät käsittelemään matematiikan oppitunteja lukuvuonna 2021–2022. Kysymyksistä kahdeksan oli strukturoituja monivalintakysymyksiä, joiden vastausvaihtoehdoissa oli hyödynnetty aikaisempia tutkimuksia sekä aineistoja. Seitsemän avoimen kysymysten tarkoituksena oli saada esiin uusia näkökulmia, joita tutkija ei ole osannut etukäteen ajatella (Hirsijärvi ym. 2009, 199). Ennen kyselyn lähettämistä suoritettiin kyselylomakkeelle esitestaus. Muutamalta ennakkoon valikoidulta eri henkilöiltä kysyttiin palautetta kyselylomakkeen sisällöstä sekä ymmärrettävyydestä. Lopullisen kyselylomakkeen linkki jaettiin vuoden 2022 marras-joulukuussa neljään eri Facebook-ryhmään, johon kuului yläkoulun matematiikan opettajia. Tämä lisäksi linkki julkaistiin Matemaattisten Aineiden Opettajien Liitto MAOL ry:n uutiskirjeessä joulukuussa 2022. Kaiken kaikkiaan kyselyyn vastasi 48 yläkoulun matematiikan opettajaa 31 eri kunnasta.

Kyselylomake jakautui neljään osaan, joista ensimmäinen osa käsitteli vastaajan eli opettajan taustoja. Taustakysymyksien tarkoituksena oli kerätä vastanneista tietoa, jota voitaisiin myöhemmin käyttää hyväksi tulosten tarkastelussa, esimerkiksi tulosten keskinäisessä vertailussa. Kysymyksillä pyrittiin selvittämään, missä kunnassa opettaja työskentelee, sekä hänen työskentelemänsä oppilaitoksen oppilasmäärä. Tämän lisäksi opettajalta kysyttiin hänen työkokemustaan vuosina, jotta saatuja tuloksia voitiin jatkossa verrata muun muassa ohjelmoinnin opetuksen määrään.

Loput taustakysymykset keskittyivät selvittämään opettajien käsityksiä termeistä ohjelmointi sekä ohjelmoinnillinen ajattelu. Koska ohjelmointia ja ohjelmoinnillista ajattelua on kouluissa opetettu vasta muutamia vuosia, on erityisen tärkeää saada tietää miten opettajat itse määrittelevät kyseiset käsitteet. Opettajien käsitys vaikuttaa merkittävästi heidän opetustyöhönsä. Tässä tutkimuksessa kysymyksillä halutaan saada varmuus, että opettajat tietävät, mitä käsitteillä ohjelmointi sekä ohjelmoinnillinen ajattelu tarkoitetaan. Lisäksi kysymyksillä pyritään parantamaan tutkimuksen luotettavuutta.

Toinen osio käsitteli ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen määrää. Osion ensimmäisessä kysymyksessä esitettiin väittämä ”Olen opettanut ohjelmointia viime lukuvuonna 2021–2022”, jonka tarkoituksena oli selvittää ohjelmointia opettavien opettajien lukumäärä. Ohjelmointia opettaneilta opettajilta pyydettiin lisäksi arvioita heidän ohjelmointiin ja ohjelmoinnilliseen ajatteluun keskittyneiden oppituntien lukumäärästä. Heitä, jotka vastasivat kieltävästi ohjelmoinnin opetusta koskevaan kysymykseen, pyydettiin kertomaan suurimmat syyt, miksi he eivät olleet opettaneet ohjelmointia. Toisen osion viimeiseen kysymykseen vastasivat kaikki kyselyyn osallistuneet ja sen avulla haluttiin selvittää opettajien arvioita riittävästä ohjelmoinnin opetukseen käytettävästä oppituntimäärästä yhden lukuvuoden aikana yhtä opetusryhmää kohden.

Ohjelmoinnin sisältöjä ja opetusmenetelmiä käsiteltiin kyselylomakkeen osiossa kolme, joka sisälsi yhteensä kuusi kysymystä. Osion sisältämiin kysymyksiin vastasivat vain he, jotka olivat aikaisemmin kyselyssä vastanneet opettaneensa ohjelmointia. Osion kysymyksillä pyrittiin selvittämään, mitä työskentelymuotoja tai opetusmenetelmiä opettaja olivat käyttäneet ohjelmoinnin opetuksessa. Avoimella kysymyksellä haluttiin lisäksi saada tarkempaa tietoa siitä, ovatko käytetyt menetelmät olleet opettajan mielestä toimivia. Sisältöjä käsittelevien kysymysten tarkoituksena oli selvittää, mitä ohjelmointikieliä ja ohjelmoinnin

sisältöjä opettajat ovat opettaneet. Lisäksi haluttiin saada selville, mitä oppimateriaaleja, ohjelmointiympäristöjä tai -sivustoja opettajat ovat käyttäneet ohjelmoinnin opetuksessa. Osion viimeisenä kysymyksenä vastaajat saivat avoimesti kertoa, minkälainen on heidän tyypillinen ohjelmointiin keskittyvä oppitunti. Kysymyksellä haluttiin saada laadullista tietoa ohjelmoinnin opetuksesta ja oppitunneista.

Kyselylomakkeen viimeinen osio käsitteli koulujen resursseja. Kysymyksillä haluttiin selvittää, mitä laitteita ja resursseja vastaajan koululla oli tarjota ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tueksi. Lisäksi haluttiin saada selville ovatko käytettävät resurssit opettajien mielestä riittäviä. Kyselylomakkeen viimeisenä kysymyksenä kysyttiin, miten he kokevat olemassa olevien resurssien vaikuttavan ohjelmoinnin opetukseen. Myös tällä kysymyksellä haluttiin saada laadullista tietoa ohjelmoinnin opetuksen resursseista.

5.3.2 Aineiston analysointi

Kerätyn aineiston analyysi koostuu Hirsijärven ym. (2009, 221–222) mukaan kolmesta vaiheesta; tietojen tarkistamisesta, tietojen täydentämisestä sekä aineiston järjestämisestä. Tutkimuksen aineisto tarkastettiin mahdollisten virheiden ja puuttuvien tietojen varalta. Muutoksia aineistoon tehtiin muun muassa nostamalla strukturoitujen monivalintakysymysten ”Jotain muuta, mitä?” vastausvaihtoehdon vastauksissa usein esiintyviä teemoja omaksi luokaksi. Toinen muutos tehtiin lisäämällä ohjelmointia opettamattomien opettajien opettettujen tuntien määräksi 0 oppituntia. Lisäys mahdollisti vertailun opettettujen tuntien ja riittävän oppituntimäärän arvioiden välille. Muuten kyselyn aineisto saatiin Webropol -ohjelmistolla muokattua suoraan taulukoituun muotoon, joka mahdollisti tulosten analysoimisen prosenttiosuuksia vertailemalla. Lisäksi kyselyyn vastanneiden opettajien riittävästä ohjelmoinnin opetuksen oppituntimäärän arvioista laskettiin keskiarvo sekä keskihajonta.

Kyselylomakkeen sisältämät avointen kysymysten vastaukset analysoitiin käyttäen laadullista sisällönanalyysia kahdella eri tavalla. Vastanneiden ajatuksia käsitteistä ohjelmointi ja ohjelmoinnillinen ajattelu analysoitiin muodostamalla ensin käsitteisiin liittyviä luokkia aikaisempiin määritelmiin pohjautuen. Esimerkiksi käsitteelle ohjelmoinnillinen ajattelu muodostui yhteensä 11 luokkaa, kuten ”ajattelun taito” tai ”ongelman pilkkominen osiin”. Tämän jälkeen aineisto kvantifioitiin muodostettujen luokkien perusteella.

Kvantifioimalla saatiin tietoa siitä, kuinka monta kertaa sama luokka esiintyi vastanneiden kuvauksissa, esimerkiksi kuinka monessa vastauksessa määriteltiin ohjelmoinnillisen ajattelun sisältävän ongelman pilkkomista osiin. Näin saatiin tietoa siitä, miten hyvin opettajien vastauksen vastasivat käsitteiden ohjelmointi ja ohjelmoinnillinen ajattelu tutkimuslähteisiin perustuvia määritelmiä. Samankaltaista analysointia käytettiin tyypillisen ohjelmointiin keskittyvän oppitunnin kuvauksissa, jossa teoriapohjaisen luokittelun sijaan luokat muodostuivat vastauksissa usein esiintyvien piirteiden perusteella.

Käytettyjen opetusmenetelmien toimivuutta sekä olemassa olevien laitteiden ja resurssien vaikuttavuutta ohjelmoinnin opetukseen käsittelevien kysymysten vastauksia analysoitiin pelkistämällä jokainen alkuperäinen vastaus sekä ryhmittelemällä ja yhdistämällä pelkistetyt vastaukset omiin kategorioihinsa. Esimerkiksi erään opettajan vastaus ”Kaikilla oppilailla kone käytössä, mikä helpottaa opetusta” yksinkertaistettiin muotoon ”Kaikilla oppilailla on oma tietokone”, joka liitettiin kuuluvaksi kategoriaan ”Laitteita on riittävästi.” Vastauksen oli mahdollista kuulua samanaikaisesti moneen eri kategoriaan. Lisäksi laitteiden ja resurssien vaikuttavuutta käsittelevän kysymyksen ryhmittelyssä hyödynnettiin vastaajan vastausta kysymystä edeltäneeseen kysymykseen, joka käsitteli opettajien näkemystä laitteiden ja resurssien riittävydestä. Näin saatiin esimerkiksi tarkempaa tietoa siitä, miten he, jotka kertoivat aikaisemman kysymyksen perusteella laitteiden ja resurssien olevan riittäviä, kokivat niiden vaikuttavan ohjelmoinnin opetukseen.

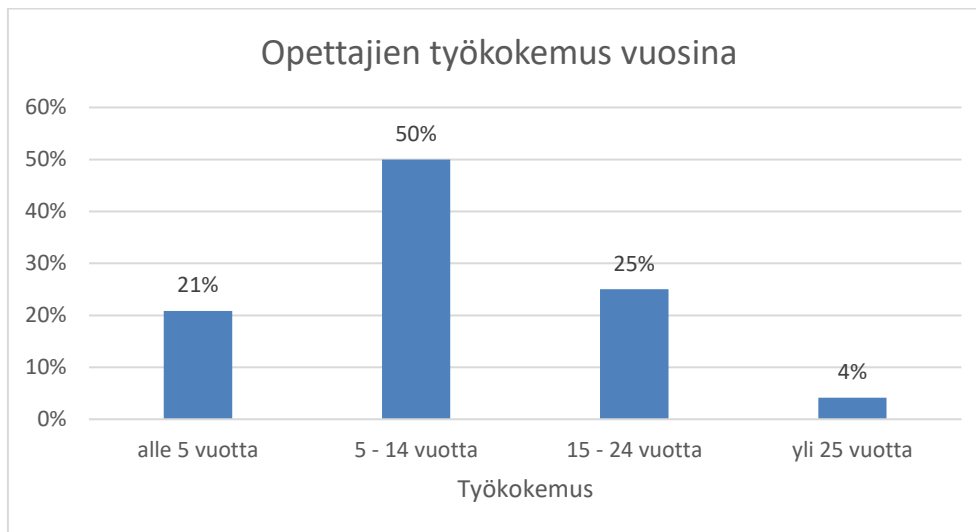
5.3.3 Tutkimuksen vastanneiden tausta

Tutkimukseen vastasi yhteensä 48 opettajaa 31 eri kunnasta. Kuten taulukko 1 osoittaa 42 % vastanneista työskenteli Uudellamaalla. Muiden maakuntien kohdalla opettajat jakautuivat tasaisemmin.

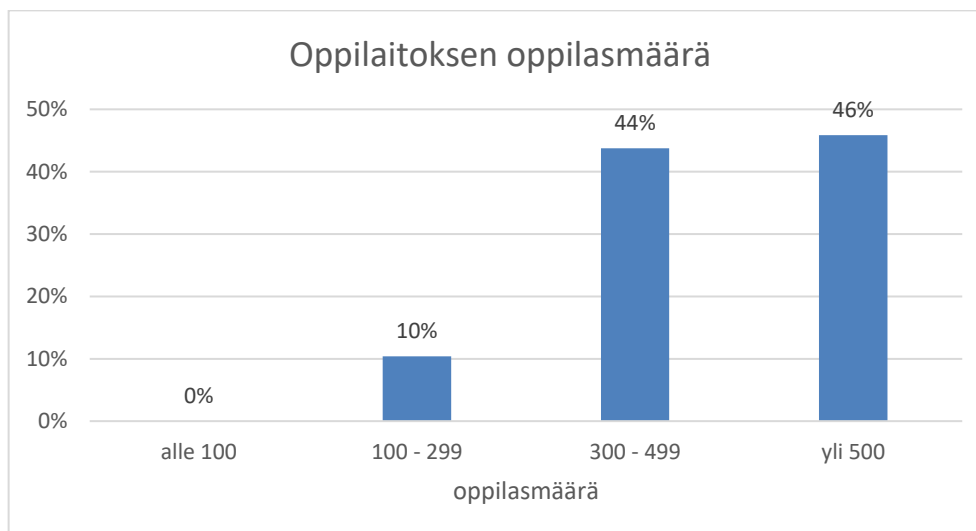
TAULUKKO 1 Vastaajien jakautuminen maakunnittain (n=48)

| Maakunta | Vastaajien lukumäärä | Vastaajien osuus prosentteina |
|-------------------|----------------------|-------------------------------|
| Uusimaa | 20 | 42 % |
| Varsinais-Suomi | 5 | 10 % |
| Pohjois-Pohjanmaa | 3 | 6 % |
| Satakunta | 3 | 6 % |
| Keski-Suomi | 3 | 6 % |
| Pirkanmaa | 2 | 4 % |
| Kymenlaakso | 2 | 4 % |
| Etelä-Savo | 2 | 4 % |
| Keski-Pohjanmaa | 2 | 4 % |
| Kanta-Häme | 1 | 2 % |
| Pohjois-Savo | 1 | 2 % |
| Pohjois-Karjala | 1 | 2 % |
| Etelä-Pohjanmaa | 1 | 2 % |
| Päijät-Häme | 0 | 0 % |
| Etelä-Karjala | 0 | 0 % |
| Pohjanmaa | 0 | 0 % |
| Kainuu | 0 | 0 % |
| Lappi | 0 | 0 % |
| Ahvenanmaa | 0 | 0 % |

Kuviosta 1 nähdään, että puolet vastanneista (50 %) oli työskennellyt opettajana 5–14 vuotta ja neljäsosa (25 %) 15–24 vuotta. Reilu viidennes (21 %) vastanneista oli työskennellyt alle viisi vuotta ja vain 4 % yli 25 vuotta. Lähes jokainen kyselyyn vastanneista opettajista työskenteli kouluissa, joissa oppilasmäärät olivat suuria (KUVIO 2). Hieman alle puolet (46 %) vastanneista työskenteli yli 500 oppilaan oppilaitoksessa ja 44 % vastanneista työskenteli oppilaitoksessa, jossa on 300–499 oppilasta. Vastanneista vain 10 % kertoi työskentelevänsä 100–299 oppilaan oppilaitoksessa.



KUVIO 1 Opettajien työkokemus vuosina (n=48)



KUVIO 2 Oppilaitosten oppilasmäärä (n=48)

Tulosten tarkastelusta maakuntakohtaisesti luovuttiin, sillä Uudellamaalla työskentelevien osuus oli merkittävän suuri verrattuna muihin maakuntiin. Lisäksi kyselyn vastausten tarkastelusta suhteessa oppilaitosten kokoon luovuttiin oppilaitosten oppilasmäärien ollessa suuria. Kuitenkin kyselyn vastausten tarkastelua suhteessa opettajan työkokemukseen voitiin pitää perusteltuna, sillä vastanneiden työkokemusten välillä oli havaittavissa selkeitä eroja.

Kyselyyn vastanneista lähes jokainen vastasi tietävänsä, mitä käsitteillä ohjelmointi ja ohjelmoinnillinen ajattelu tarkoitetaan. Yli puolet (52 %) vastaajista kertoi olevansa täysin samaa mieltä väitteen ”tiedän mitä ohjelmointi ja ohjelmoinnillinen ajattelu tarkoittaa”

kanssa. 48 % vastanneista vastasi olevansa ”osittain samaa mieltä” ja vain yksi vastanneista (2 %) vastasi olevansa ”osittain eri mieltä” väitteen kanssa.

Opettajat saivat lisäksi omin sanoin kertoa, mitä käsitteet ohjelmointi ja ohjelmoinnillinen ajattelu heidän mielestään tarkoittavat. Vastaukset luokiteltiin aikaisempien tutkimusten ja kirjallisuuden pohjalta muodostettuihin kategorioihin. Lähes puolet (44 %) vastanneista mainitsi ohjelmoinnillisen ajattelun sisältävän ratkaisujen muodostamista, esimerkiksi algoritmien suunnittelua. Reilu kolmannes (35 %) koki käsitteen viittaavan ajattelun taitoon. Useissa vastauksissa (35 %) nousi lisäksi esiin ohjelmoinnillisen ajattelun sisältävän ongelman pilkkomista osiin ja myös moni (27 %) vastasi käsitteen sisältävän kaavojen tai toistojen tunnistamista. Useissa vastauksissa, joissa opettajan vastaus oli tulkittu sisältävän jotain muuta (19 %) kuin ohjelmoinnillisen ajattelun määritelmän mukaisia sisältöjä, katsottiin sisältävän ohjelmoinnin käsitteeseen liittyviä tyypillisiä elementtejä.

Käsitteen ohjelmointi määrittämisessä useissa vastauksessa mainittiin sen olevan ohjeiden tai käskyjen kirjoittamista (79 %). Yksittäisissä vastauksissa tuotiin ilmi, että ohjelmointiin kuuluu myös ohjelmien suunnittelua (6 %) sekä testausta (4 %). Vastaukset, joiden tulkittiin sisältävän jotain muuta (25 %) kuin ohjeiden kirjoittamista tai ohjelmien testausta tai suunnittelua sisälsivät kuitenkin ohjelmointiin rinnastettavia elementtejä, mutta niiden ei voitu suoraan nähdä kuuluvan mihinkään valmiista luokista. Useat vastaukset sisälsivät esimerkiksi maininnan jostain kohteesta, jossa voidaan hyödyntää ohjelmointia.

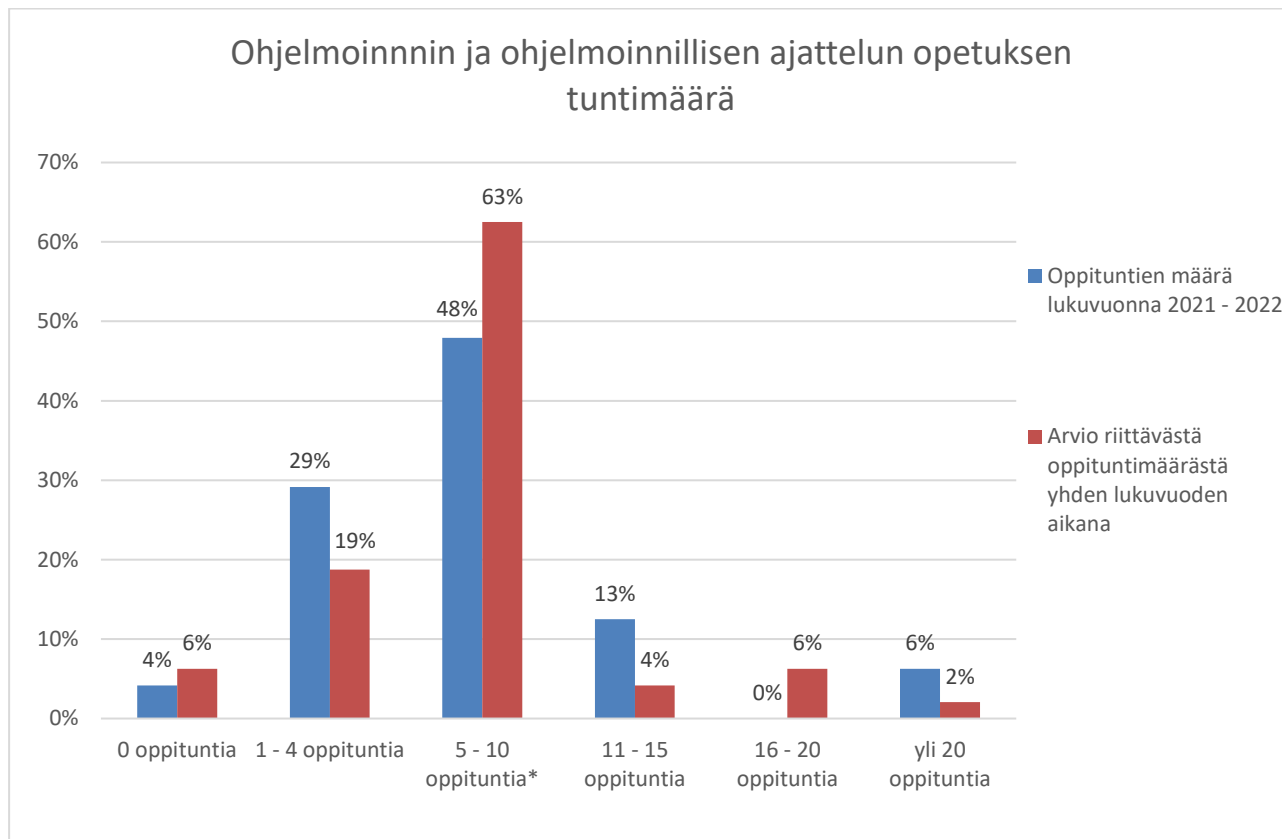
6 Tulokset

Tässä luvussa esitellään kyselytutkimuksen tulokset tutkimuskysymyksiin nojautuen järjestyksessä. Tulokset käsittelevät lukuvuoden 2021–2022 matematiikan oppitunteja ja niiden läpikäymisessä on otettava huomioon, että termillä ohjelmointi viitataan lisäksi ohjelmoinnillisen ajattelun käsitteeseen.

6.1 Opetuksen määrä

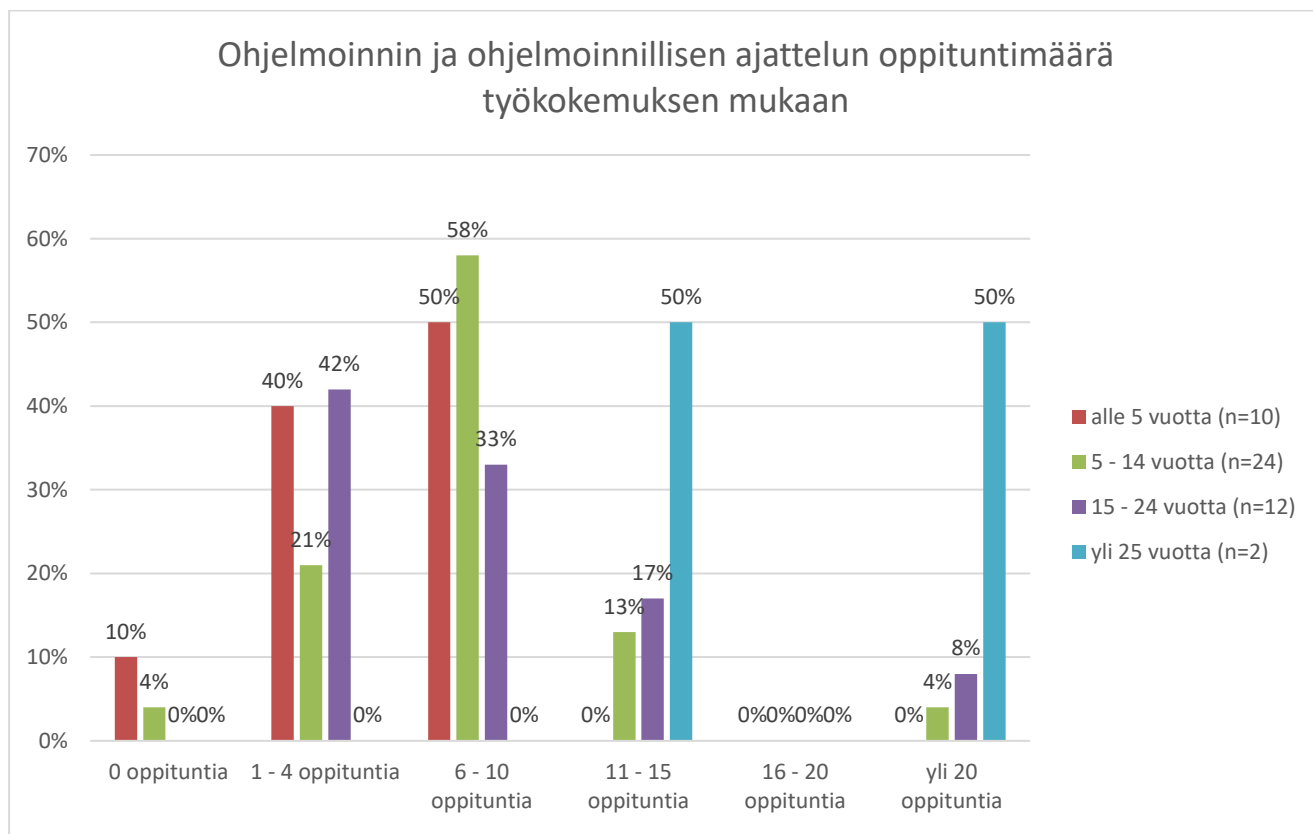
Kaikista 48 kyselyyn vastanneista opettajista 96 % oli opettanut ohjelmointia. Vain kaksi (4 %) vastanneista kertoi, etteivät he olleet opettaneet ohjelmointia. Syitä opetuksen puuttumiselle perusteltiin muun muassa ajanpuutteena sekä sillä, että sopivia materiaaleja oli hankala löytää oppilaiden vaihtelevan taitotason takia. Ohjelmointia opettaneista opettajista noin puolet (48 %) kertoivat opettaneen ohjelmointia keskimäärin 6–10 oppituntia yhtä opetusryhmää kohtaan. 29 % ohjelmointia opettaneista olivat opettaneet 1–4 oppituntia, 13 % 11–15 oppituntia ja loput 6 % kertoi opettaneensa ohjelmointia yli 20 oppituntia yhtä opetusryhmää kohden.

Kyselyyn vastanneilta kysyttiin arviota ohjelmoinnin opetukseen käytettävästä oppituntimäärästä yhtä opetusryhmää kohden yhden lukuvuoden aikana. Vastausten keskiarvoksi saatiin 7,9 oppituntia, keskihajonnan ollessa 6,4 oppituntia. Kuviosta 3 nähdään, että suurin osa (63 %) vastanneista arvioi 5–10 oppituntia olevan riittävä määrä ohjelmoinnin opetusta yhden lukuvuoden aikana. Neljäsosa (25 %) vastasi alle viiden oppitunnin riittävän ja kolme (6 %) opettajaa oli sitä mieltä, ettei ohjelmointia tarvitsisi opettaa ollenkaan.



KUVIO 3 Opettajien ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen käyttämä tuntimäärä sekä arvio riittävästä tuntimäärästä yhtä opetusryhmää kohden. (n=48) (* sisältää kysymykseen " Kuinka monta oppituntia olet opettanut ohjelmointia viime lukuvuonna 2021–2022?" 6–10 oppituntia vastanneet)

Ohjelmoinnin opetuksen tuntimääriä verrattiin lisäksi opettajien työkokemukseen (KUVIO 4). Alle viisi vuotta opettajina toimineista puolet (50 %) kertoi opettaneensa ohjelmointia 6–10 oppituntia ja toinen puolet (50 %) alle viisi oppituntia. Pidempään, 5–14 vuotta opettajina toimineista vastaajista hieman yli puolet (58 %) kertoi opettaneensa ohjelmointia 6–10 oppituntia ja 15–24 vuotta työkokemusta kerryttäneistä opettajista 42 % vastasi opettaneensa alle viisi oppituntia. Viimeisimpänä ryhmänä yli 25 vuotta opettajina toimineista opettajista toinen (50 %) kertoi opettaneensa 11–15 oppituntia ja toinen (50 %) yli 20 oppituntia.



KUVIO 4 Opetettujen ohjelmoinnin ja ohjelmoinnillisen ajattelun oppituntien määrä opettajan työkokemuksen mukaan yhtä opetusryhmää kohtaan lukuvuonna 2021–2022 (n=48)

Myös opettajien arvioita riittävästä oppituntimäärästä ohjelmoinnin opetukselle verrattiin heidän työkokemuksensa määrään (TAULUKKO 2). Alle viisi vuotta työkokemusta omaavien opettajien arvio riittävän ohjelmoinnin opetukseen käytettävän tuntimäärän keskiarvoksi saatiin 6,3 oppituntia (keskihajonta 3,6) yhtä opetusryhmää kohden yhden lukuvuoden aikana. Heillä, joilla työkokemusta oli 5–14 vuotta vastaavaksi keskiarvoksi saatiin 7,1 oppituntia (keskihajonta 5,3) ja 15–24 vuotta työkokemusta omaavilla opettajilla 10,3 oppituntia (keskihajonta 9,7). Molemmat yli 25 vuotta opettajina toimineet opettajat arvioivat ohjelmoinnin opetukseen käytettävän riittävän tuntimäärän olevan 10 oppituntia (keskihajonta 0,0) yhtä opetusryhmää kohden yhden lukuvuoden aikana.

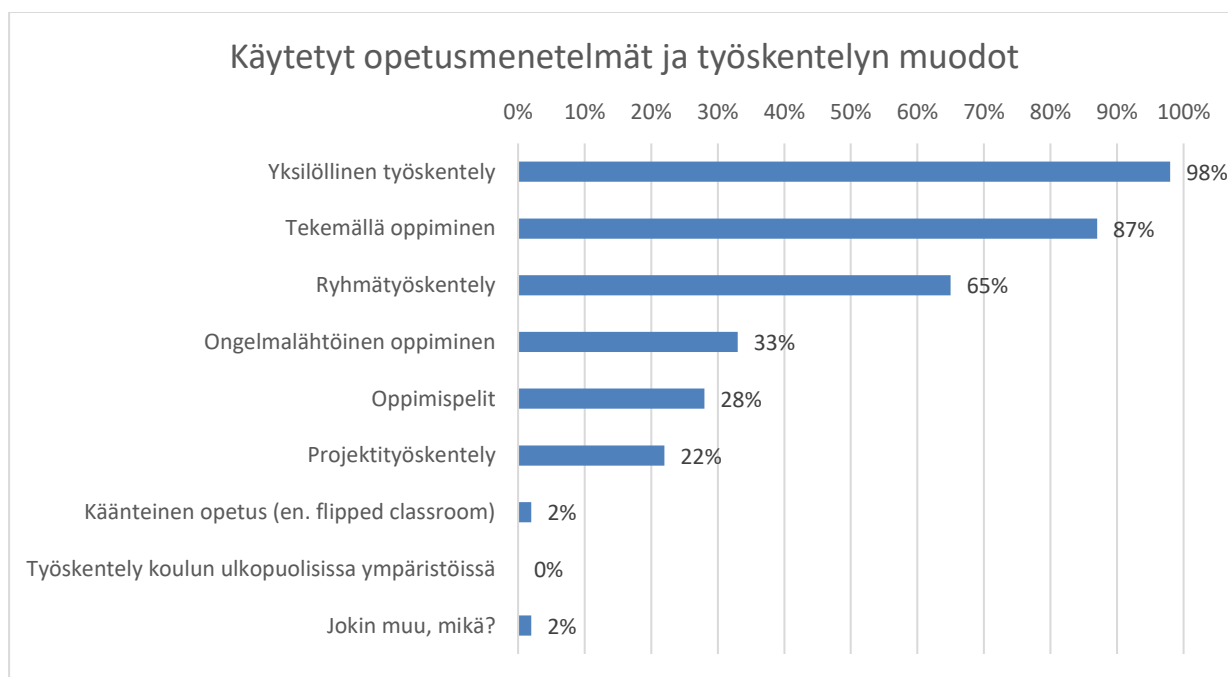
TAULUKKO 2 Opettajien arviot riittävästä ohjelmoinnin ja ohjelmoinnillisen ajattelun oppituntimäärästä opettajan työkokemuksen mukaan yhtä opetusryhmää kohden yhden lukuvuoden aikana (n=48)

| Opettajan työkokemus | n | Oppituntien keskiarvo (t) | Keskihajonta |
|----------------------|----|---------------------------|--------------|
| alle 5 vuotta | 10 | 6,3 | 3,6 |
| 5 - 14 vuotta | 24 | 7,1 | 5,3 |
| 15 - 24 vuotta | 12 | 10,3 | 9,7 |
| yli 25 vuotta | 2 | 10,0 | 0,0 |

6.2 Opetusmenetelmät ja työskentelyn muodot

Tämän osion kysymyksiin vastasivat vain he, jotka vastasivat opettaneensa ohjelmointia lukuvuonna 2021–2022 (n=46).

Opettajilta kysyttiin monivalintakysymyksellä heidän käyttämistään opetusmenetelmistä ja työskentelyn muodoista. Vastanneista lähes kaikki (98 %) kertoivat käyttäneensä ohjelmoinnin opetuksessaan yksilöllistä työskentelyä (KUVIO 5). Muita suosittuja työskentelyn muotoja olivat tekemällä oppiminen (87 %) sekä ryhmätyöskentely (65 %). Vähemmän käytetyksi työskentelyn muodoksi ilmeni ongelmalähtöinen oppiminen, jota kertoi käyttäneensä kolmasosa (33 %) vastanneista. Myös oppimispelit (28 %) ja projektityöskentely (22 %) jäivät vähemmän käytetyiksi työskentelyn muodoiksi.



KUVIO 5 Opettajien eri opetusmenetelmien ja työskentelyn muotojen käyttö ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa

Avoimella kysymyksellä haluttiin saada lisätietoa opetusmenetelmien toimivuudesta. Hieman yli puolista (54 %) vastauksista kävi ilmi, että opettajien käyttämät opetusmenetelmät miellettiin toimivina, kun taas 28 % vastanneista eivät kokeneet käyttämiään opetusmenetelmiä toimivina. Lisäksi 15 % vastaajista nähtiin kuuluvan ryhmään, jossa opetusmenetelmät koettiin osittain toimivina. Kaikista vastauksista yksi (2 %) oli tyhjä.

Heistä, jotka kokivat opetusmenetelmien olleen toimivia, noin kolmasosa (32 %) oli sitä mieltä, että ohjelmointia oppii parhaiten itse tekemällä ja kokeilemalla.

”Kyllä. Jokaisen täytyy laittaa itsekin 'kädet saveen' eli kirjoittaa itse koodia. Pelkästään vierestä seuraamalla ei opi. Tulkiohjelma antaa hyvät puitteet oppimiseen kokeilemalla yrityksen ja erehdyksen kautta”

Kuudessa (24 %) vastauksessa vastanneet toivat esiin, että heidän käyttämiensä opetusmenetelmien avulla oppilaat pääsevät etenemään oman tahdin ja taitotason mukaisesti. Myös opetuksen eriyttäminen koettiin onnistuneen.

”Ovat, erityisesti oppilaiden yksilölliset oppimispolut tarjoaa mahdollisuuden jokaisen toimia omalla tasollaan ja omien mielenkiintojen piirissä”

”Eriyttäminen onnistuu, kaikki ovat saaneet jotakin aikaan ja opetussuunnitelman tavoitteisiin on päästy.”

Lisäksi kolmessa vastauksessa (12 %) kävi ilmi, että käytetyt opetusmenetelmät herättivät oppilaissa onnistumisen sekä innostumisen tunteita ja ne motivoivat oppilaita ohjelmoimaan. Vastaajat kokivat oppilaiden oppivan.

He, jotka kokivat käyttämiensä opetusmenetelmien olleen toimimattomia, syitä toimimattomuudelle perusteltiin lähes puolissa (46 %) vastauksissa sillä, etteivät ne saa oppilaita motivoitumaan tai keskittymään oleelliseen.

” Ohjeiden noudattaminen kirjasta on monelle haastavaa. Tehtäviin keskittyminen tuottaa myös haasteita oppilaiden puolelta. Nopea turhautuminen ja luovuttaminen yleistä”

”Ei välttämättä, koska oppilaita on ollut tosi vaikea saada innostumaan ohjelmoinnista. Ne, jotka osaa, ovat innostuneita. Muut eivät.”

Toisena suurena syynä opetuksen toimimattomuudelle oli, ettei opettajat olleet löytäneet toimivaa tapaa opettaa ohjelmointia. Perustelu esiintyi lähes puolien (46 %) opettajien vastauksista. Toimivan opetusmenetelmän löytäminen koettiin hankalaksi oppilaiden heikon osaamisen vuoksi. Myös oppilaiden osaamisen välillä olevien suurten erojen koettiin hankaloittavan koko ryhmälle toimivan opetustavan löytämistä. Eräässä vastauksessa tuotiin lisäksi esiin, miten toimivan opetusmenetelmän käyttö vaatisi opettajan oman osaamisen kehittämistä.

”Mieleistäni oppilaiden lähtötaso ihan pelkässä tietokoneen käyttämisessä on niin alhainen, etten koe pystyväni käyttämään muunlaisia opetusmenetelmiä kuin

esimerkistä oppimista ja sen soveltamista. Jotkut oppivat hyvin ja nopeasti, mutta muutamat eivät koko aikana saa kiinni edes perusasioista.”

”Eivät ole parhaita mahdollisia. Meillä on paljon heikkoja oppilaita, emmekä ole löytäneet sopivaa opetustapaa.”

Kolmessa vastauksessa (23 %) tuotiin lisäksi esiin, ettei kouluilla ole tarvittavia resursseja tai olemassa olevat resurssit kuten oppimateriaalit koettiin puutteellisiksi.

Yli puolet (57 %) opettajista, jotka kertoivat opetusmenetelmien olleen osittain toimivia, kertoivat opetusmenetelmän toimivuuden riippuvan opiskelijoiden motivaatiosta tai taitotasosta.

”Osalle toimivia, mutta osalle ei. Motivaatio ratkaisee.”

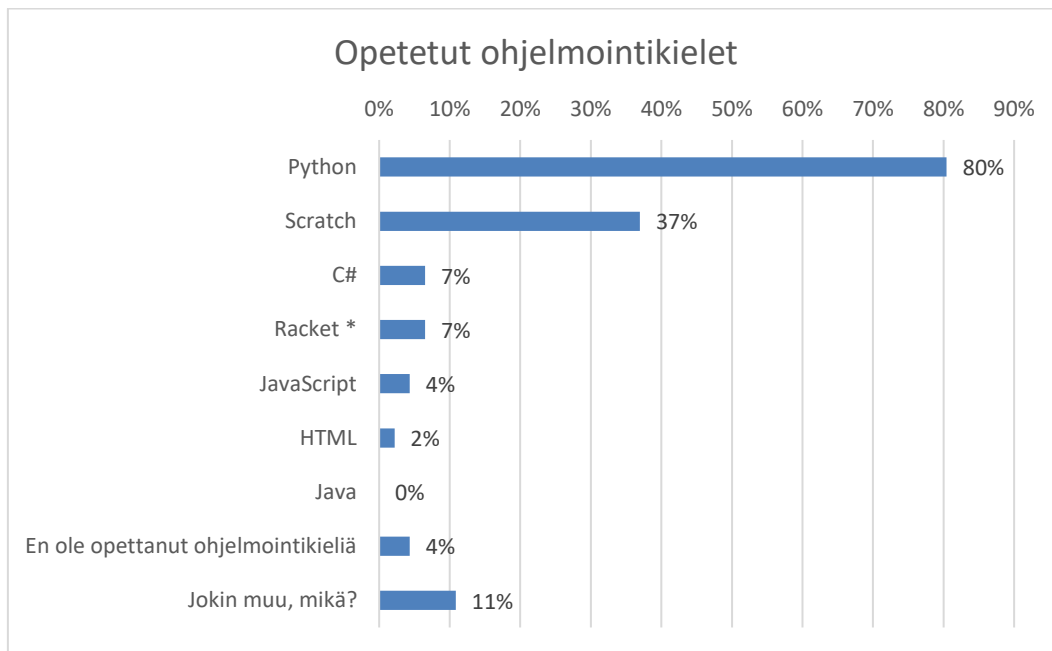
”Osalle oppilaita, jotka kykenevät omatoimiseen ajatteluun ja tiedon soveltamiseen, sekä hyväksymään pettymyksiä ja silti jatkamaan yrittämistä ja testaamista.”

Kahdessa (29 %) vastauksessa tuotiin lisäksi esiin, että vaikka opetus on toimivaa, voisi se olla tehokkaampaa toista menetelmää tai oppimateriaalia käyttäen.

”Ovat toimivia sinänsä, että oppiminen tapahtuu paljolti esimerkin avulla ja ongelmanratkaisun kautta. Tehokkaampaakin se varmaan voisi olla kyllä. Esimerkit eivät ole tällä hetkellä ihan parhaat mahdolliset.”

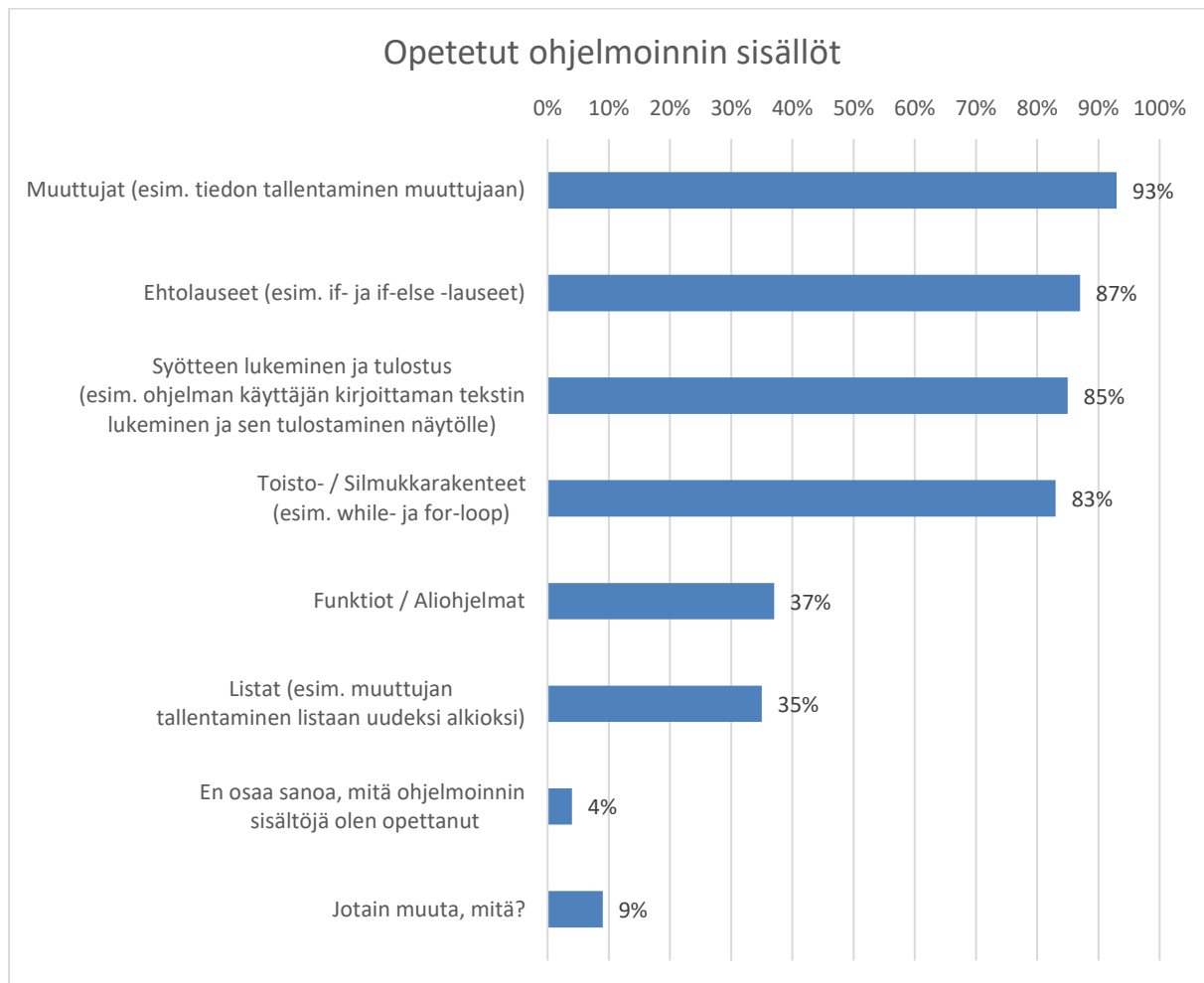
6.2.1 Opetettavat sisällöt

Suurin osa (80 %) opettajista oli opettanut Python-ohjelmointikieltä (KUVIO 6). Toiseksi suosituin ohjelmointikieli oli Scratch, jota oli opettanut reilu kolmannes (37 %) vastanneista. Selkeästi vähiten opetettiin ohjelmointikieliä C# (7 %), JavaScript (4 %), HTML (2 %) sekä Racket-ohjelmointikieltä (7 %), joka nousi esiin ”Jokin muu, mikä?” vastauksista. Lisäksi 4 % vastanneista kertoi, ettei ole opettanut ohjelmointikieliä ja 11 % vastanneista valitsi vastausvaihtoehdoksi ”Jokin muu, mikä?”.



KUVIO 6 Opettajien opettamat eri ohjelmointikiel (* poimittu ”Jokin muu, mikä?” vastauksista)

Kun kysyttiin, mitä ohjelmoinnin sisältöjä vastanneet ovat opettaneet, osoittautui eniten opetetuksi sisällöksi muuttujat (93 %). Ehtolauseet (87 %), syötteen lukeminen ja tulostus (85 %) sekä toisto- / silmukkarakenteet (83 %) kuuluivat myös sisältöihin, joita reilusti suurin osa opettajista kertoi opettaneen (KUVIO 7). Vähemmälle jäi funktioiden (37 %) sekä listojen (35 %) opetus, joita kumpaakin kertoi opettavansa vain reilu kolmasosa vastaajista.

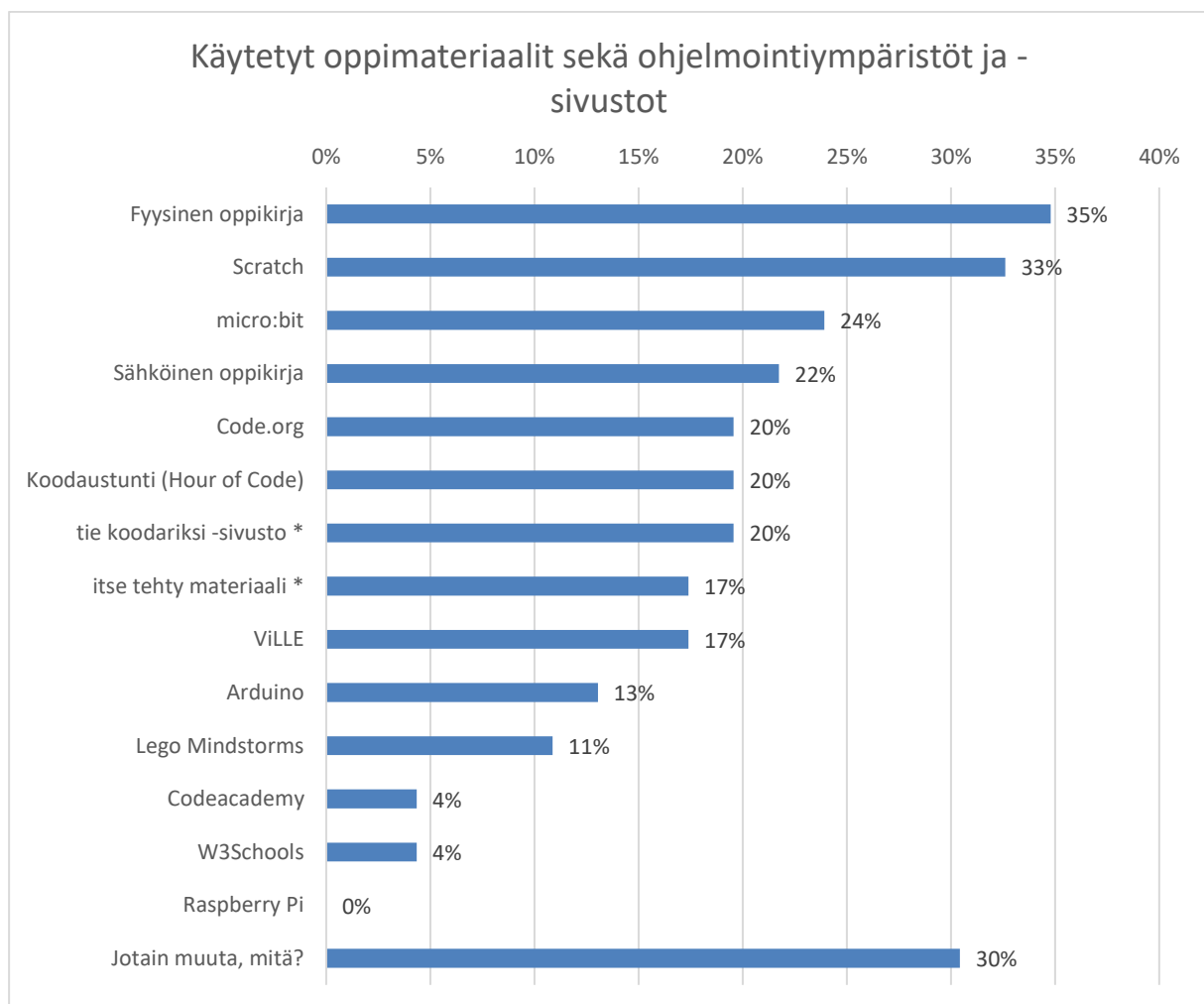


KUVIO 7 Opettajien opettamat eri ohjelmoinnin sisällöt

6.2.2 Oppimateriaalit ja -ympäristöt

Kuten kuvioista 8 nähdään, suosituimmaksi oppimateriaaliksi nousi kyselyn mukaan fyysinen oppikirja, jota käytti 35 % vastanneista. Sähköistä oppikirjaa taas käytti vain reilu viidennes (22 %) vastanneista. Ohjelmointiympäristöistä käytetyin oli Scratch, jota kertoi käyttäneensä 33 % opettajista. Code.org -sivustoa kertoi käyttäneensä joka viides (20 %) samoin kuin koodaustunti-sivustoa (20 %). Myös ”Jotain muuta, mitä?” vastauksista ilmeni useampaan otteeseen esiin tie koodariksi -sivusto, jota 20 % vastanneista kertoi käyttävänsä. Edellä mainittuihin sivustoihin verrattuna hieman vähemmälle käytölle jäi ViLLE-oppimisympäristö, jota kertoi käyttäneensä 17 % opettajista. Vähiten listatuista sähköisistä ohjelmointiympäristöistä opettajat kertoivat käyttäneensä Codecademy sekä W3Schools -sivustoja, joita molempia käytti vain 4 % vastanneista.

Ohjelmoitavasta elektroniikasta ja roboteista suosituin oli micro:bit, jota kertoi käyttäneensä noin joka neljäs opettaja (24 %). Arduinoja käytti 13 % vastanneista ja hieman harvempi opettajista (11 %) kertoi käyttäneensä opetuksessa Lego Mindstorms -robotteja. ”Jotain muuta, mitä?” vastauksissa ilmeni useaan otteeseen itse tehdyt materiaalit, joita lopulta noin joka viides opettajista (17 %) kertoi käyttävänsä. Poistamalla ”Jotain muuta, mitä?” vastauksista tie kodariksi -sivustoon ja itse tehtyyn materiaaliin viittaavat vastaukset, jäljelle jäi 14 (30 %) ”Jotain muuta, mitä?” vastausvaihtoehdon valinnutta opettajaa. Heidän vastauksissansa esiintyi yksittäisiä oppimateriaaleja ja ohjelmointiympäristöjä.



KUVIO 8 Opettajien käyttämät eri oppimateriaalit sekä ohjelmointiympäristöt ja -sivustot (* poimittu ”Jotain muuta, mitä?” vastauksista)

6.2.3 Tyypillinen ohjelmointiin keskittyvä oppitunti

Tarkempaa tietoa ohjelmointiin keskittyvien oppituntien kulusta haluttiin saada pyytämällä vastaajia kuvailemaan heille tyypillistä ohjelmointiin keskittyvää oppituntia. Lähes jokaisessa

avoimen kysymyksen vastauksessa (91 %) ilmeni, että tyypillisellä ohjelmointiin keskittyvällä oppitunnilla oppilaat tekivät tehtäviä joko itsenäisesti tai ryhmissä. Oppilaat saivat esimerkiksi valita itsenäisesti oman taitotasonsa mukaan tehtäviä. Puolissa vastauksissa (50 %) kerrottiin oppitunnin alkaneen opettajajohtoisesti opettajan selostuksella tai esityksellä tunnilla käsiteltävästä aiheesta. Esityksessä opettaja esimerkiksi johdatteli oppilaat päivän aiheeseen erilaisten esimerkkien avulla.

”Sama perinteinen rakenne kuin muissakin matikan tunneissa: ope näyttää aluksi teoriaa ja esimerkkejä, sen jälkeen oppilaat tekevät tehtäviä.”

”Oppilaille jaettu materiaali, jossa kolmen tasoisia tehtäviä: a-kohdassa aina valmis koodi, jonka kirjoittavat mallin mukaan; b-kohdassa vähän edellisen koodin muokkaamista ja c-kohta soveltava tehtävä. Oppilaat etenevät omaan tahtiin yksin tai parin kanssa ja päättävät itse, tekevätkö vain a-kohdat, a+b-kohdat vai kaikki kolme.”

Joka neljännessä (26 %) vastauksista mainittiin opettajan avustavan oppilaita tehtävien teossa. Opettaja esimerkiksi kierteli luokassa antaen neuvoja ja apua niitä tarvitseville oppilaille. Eräässä tapauksessa opettaja hyödynsi myös edistyneimpiä oppilaita auttamaan heikompia ryhmäläisiä tehtävien teossa. Tämän lisäksi kahdessa vastauksessa (4 %) mainittiin tunnilla olevan mukana toinen opettaja tai ohjaaja.

”Meillä on teema, jota käydään läpi tehtävien kautta. Aluksi selitän, mikä on hommassa ideana ja käydään yhdessä teoriaa läpi. Tämän jälkeen alkavat tehdä tehtäviä omaan tahtiinsa. Mä neuvon niitä yksilöllisesti, jotka ei pääse eteenpäin ja jos avuntarvitsijoita on monta, nakitan pidemmällä olevia neuvomaan.”

”Seiskoille on pidetty Lego-robotitunti yhdessä digitutorin kanssa. ... ”

Neljässä vastauksessa (9 %) tyypillistä ohjelmointiin keskittyvää oppituntia oli kuvailtu jollain muulla tavalla. Esimerkiksi yhdestä vastauksesta kävi ilmi, että ohjelmointia on opetettu jaksoissa. Tyhjiä vastauksia oli yksi (2 %).

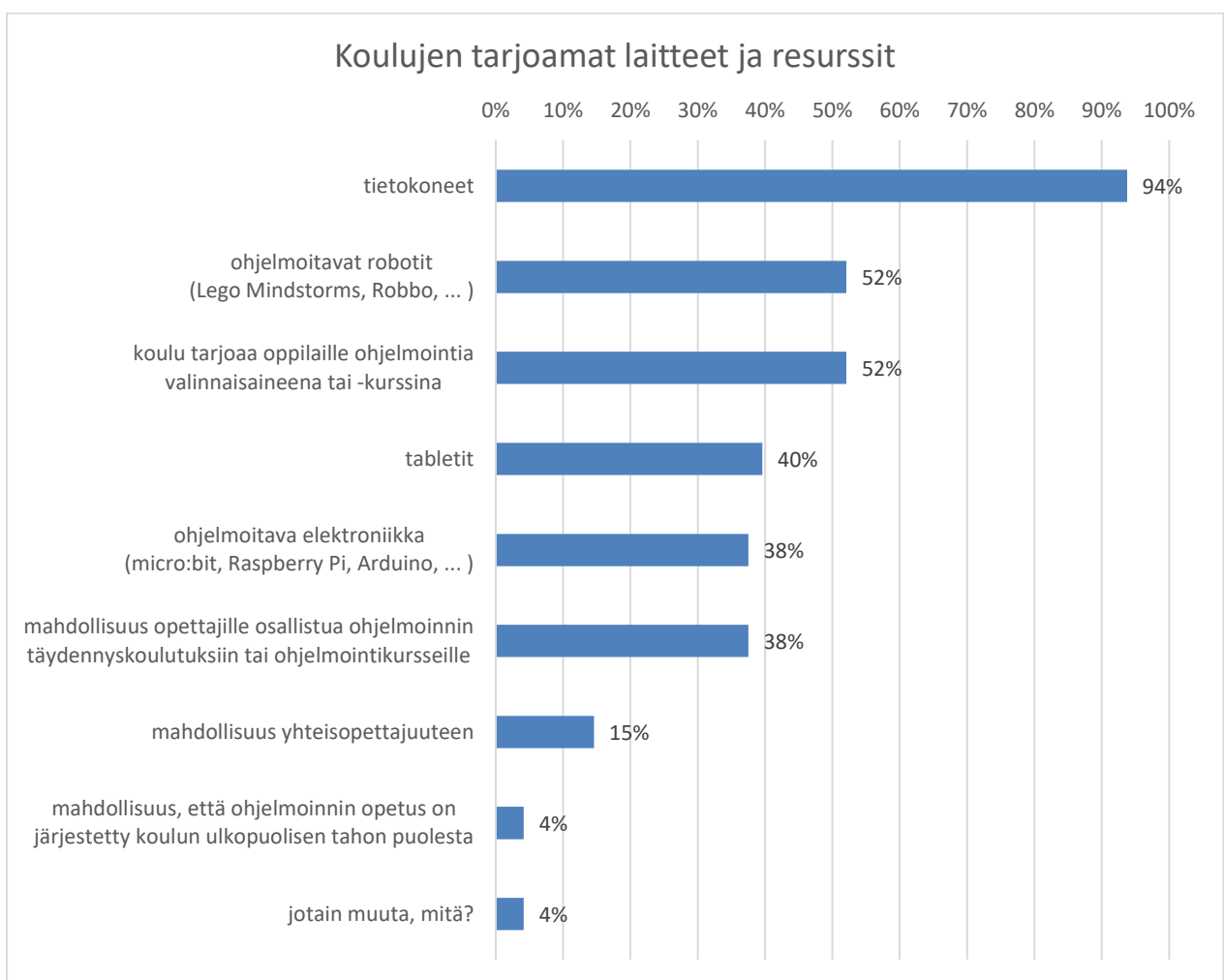
6.3 Opetuksen tukena olevat laitteet ja resurssit

Kyselylomakkeen viimeisen osion kysymyksiin vastasivat kaikki kyselylomakkeeseen vastanneet (n=48).

Koulujen tarjoamia resursseja ohjelmoinnin opetuksen tueksi kysyttiin monivalintakysymyksellä (KUVIO 9). Lähes jokainen opettaja (94 %) kertoi koulunsa tarjoavan tietokoneita ohjelmoinnin opetusta varten. Tabletteja tarjottiin kouluissa tietokoneita

harvemmin (40 %). Hieman reilu puolet (52 %) vastanneista kertoi koulujen tarjoavan ohjelmoitavia robotteja, kun taas ohjelmoitavaa elektroniikkaa tarjottiin 38 %:sta kouluista.

Hieman yli puolet (52 %) vastanneista kertoi koulunsa tarjoavan ohjelmointia valinnaisaineena tai -kurssina. Myös koulujen mahdollistavan opettajien mahdollisuutta osallistua ohjelmoinnin täydennyskoulutukseen tai ohjelmointikurssille tarjottiin reilusti yli kolmannesosasta kouluista (38 %). Vastanneista vain 15 % kertoi koulujen tarjoavan mahdollisuutta yhteisopettajuuteen ja vain muutamassa (4 %) koulussa tarjottiin mahdollisuus siihen, että ulkopuolinen taho järjestäisi ohjelmoinnin opetuksen.



KUVIO 9 Koulujen tarjoamat eri laitteet ja resurssit ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tueksi

Avoimella kysymyksellä vastanneilta saatiin lisäksi lisätietoa olemassa olevien laitteiden ja resurssien riittävydestä ja vaikuttavuudesta ohjelmoinnin opetukseen. Yli puolet (52 %) kaikista vastanneista piti koulun tarjoamia laitteita ja resursseja riittävinä ohjelmoinnin opetukseen. Heistä lähes puolet (44 %) kertoi, että laitteita oli riittävästi saatavilla.

Esimerkiksi jokaisella oppilaalla oli käytössään oma tietokone. Joka viidennestä (20 %) vastauksesta ilmeni, että käytössä olevat laitteet ja ohjelmistot tukivat ohjelmoinnin opetusta. Sama määrä (20 %) opettajia koki pystyvänsä toteuttamaan asioita omien toiveiden mukaisesti.

”Helppo pitää ohjelmointitunti millä vain oppitunnilla, kun oppilailla on omat läppärit”

” Näillä laitteilla pystyy opettamaan ne asiat ja aihealueet mitä nyt on sovittu opetettavaksi.”

”tarjontaa on enemmän kuin ehdin käyttää!”

Vaikka laitteet ja resurssit miellettiin riittäviksi, esiintyi vastauksissa kuitenkin huolia niiden toimimattomuudesta. Neljässä (16 %) vastauksessa käytössä olevia laitteita kuvailtiin puutteellisiksi ja samoin laitteiden saatavuudessa koettiin olevan hankaluuksia (16 %). Muutama opettaja kuitenkin painotti, ettei anna osittain puutteellisten tai heikkojen resurssien haitata opetusta. Lisäksi kaksi opettajaa (8 %) kertoi tarvitsevänsä enemmän koulutusta laitteiden käytöstä.

”Tietokoneita vain yksi luokallinen. Tarkkaa vuorottelua eri luokkien kanssa. Tabletit surkeita koodaamiseen.”

”Oppilailla on käytössä Chromebookit, se rajoittaa käytettäviä ohjelmointiympäristöjä hieman.”

Kaikista kyselyyn vastanneista 35 % oli sitä mieltä, että koulun tarjoamat laitteet ja resurssit eivät olleet riittäviä. Heistä lähes puolet (47 %) koki olemassa olevien laitteiden olevan vanhoja tai puutteellisia. Esimerkiksi olemassa oleville tietokoneille ei voinut asentaa haluamiaan ohjelmia opetusta varten. Lisäksi kolmanneksessa (18 %) vastauksista opettajat kertoivat olemassa olevan oppimateriaalin olevan puutteellista.

”Oppilailla päätelaitteina tabletit, jotka eivät ole monilta osin riittävät tai ainakaan kovin käytännölliset sellaisessa ohjelmoinnissa, jossa kirjoitetaan koodia ja ajetaan tehtyjä ohjelmapätkiä.”

Myös opettajien oma osaaminen rajoitti opetusta. 41 % vastanneista koki, ettei heidän oma osaamisensa ollut riittävä ohjelmoinnin opetukseen ja he tarvitsisivat tukea tai koulutusta ohjelmoinnin opetuksesta.

”Olen pätevä matematiikan opettaja, mutten ole opiskellut päivääkään ohjelmointia. Jos minun tulee opettaa sitä enemmänkin, niin olisi saatava siihen koulutusta (koulutuspäivä tmv.; en halua käyttää tähän omaa vapaa-aikaa)”

Reilu kymmenes (13 %) kaikista vastanneista ei osannut sanoa olivatko koulun tarjoamat laitteet ja resurssit riittäviä ohjelmoinnin opetukseen. Heistä kaksi (33 %) ei tarkkaan tiennyt koulun resursseja. Muutoin avoimen kysymyksen vastauksista ilmeni samoja teemoja kuin edellä; ei riittävästi laitteita (17 %), käytössä olevat laitteet puutteellisia (17 %) ja opettaja tarvitsisi lisää koulutusta ohjelmoinnin opetuksesta (17 %).

7 Johtopäätökset ja pohdinta

7.1 Tulosten tarkastelu

Tämän tutkimuksen tarkoituksena oli selvittää, kuinka paljon ja millä menetelmillä ohjelmointia ja ohjelmoinnillista ajattelua opetetaan matematiikan oppitunneilla yläkoulussa. Lisäksi tutkittiin, mitä ohjelmoinnin sisältöjä kouluissa opetettiin ja mitä ohjelmointiin tarkoitettuja materiaaleja ja oppimisympäristöjä käytettiin opetuksessa. Tutkimuksessa selvitettiin myös, mitä resursseja koulut tarjoavat ohjelmoinnin opetukseen sekä miten nämä resurssit vaikuttavat ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen.

Tutkimuksen tulokset osoittavat, että ohjelmointia ja ohjelmoinnillista ajattelua opetetaan peruskoulun yläluokkien matematiikan oppitunneilla kattavasti. Lähes jokainen tutkimukseen osallistuneista opettajista kertoi opettaneensa ohjelmointia tai ohjelmoinnillista ajattelua lukuvuonna 2021–2022. Tuloksesta voidaan päätellä matematiikan opettajien noudattavan voimassa olevaa opetussuunnitelmaa, jonka mukaan ohjelmointia ja ohjelmoinnillista ajattelua tulisi opettaa osana matematiikan opetusta. Lisäksi kun tulosta verrataan aikaisempaan Tanhua-Piironen ym. (2020) teettämään tutkimukseen, missä kymmenestä tutkimukseen osallistuneesta koulusta neljässä ohjelmoinnin opetus ei toteutunut millään tavalla, viestii tulos positiivisesta kehityksestä.

Kehitystä on havaittavissa myös opettajien käsityksistä käsitteistä ohjelmointi ja ohjelmoinnillinen ajattelu. Vuonna 2017–2019 (Tanhua-Piironen ym. 2020) useampi opettaja kertoi ohjelmoinnin olevan heille täysin tuntematon aihealue. Nyt taustakysymysten vastausten perusteella voitiin sanoa matematiikan opettajien tietävän, mitä käsitteillä ohjelmointi ja ohjelmoinnillinen ajattelu tarkoitetaan. Useissa vastauksissa esiintyi aikaisempien tutkimusten ja teorioiden pohjalta nostettuja ohjelmoinnin tai ohjelmoinnillisen ajattelun käsitteisiin liittyviä teemoja. Lisäksi kysymysten vastauksista näkyi, että suurin osa opettajista osasi erottaa ohjelmoinnin ja ohjelmoinnillisen ajattelun määritelmät toisistaan.

Noin joka toinen opettaja kertoi opettaneensa ohjelmointia ja ohjelmoinnillista ajattelua 5–10 oppituntia ja joka kolmas 1–4 oppituntia yhtä opetusryhmää kohden lukuvuonna 2021–2022. On vaikea sanoa, ovatko oppituntimäärät riittäviä ohjelmoinnin ja ohjelmoinnillisen ajattelun sisältöjen opetukseen, sillä opetussuunnitelmassa ei oteta kantaa, kuinka paljon ohjelmointia ja ohjelmoinnillista ajattelua tulisi opettaa. Opetustuntimääriä

voidaan kuitenkin verrata matematiikan opetukselle annettuun opetuksen vähimmäismäärään. Kun vuosiluokilla 7–9 matematiikkaa tulee opettaa keskimäärin vähintään 139 oppituntia yhden lukuvuoden aikana vastaa 1–4 oppituntia noin 1–3 % koko lukuvuoden matematiikan oppitunneista. Samalla 5–10 oppituntia vastaa 3–7 % lukuvuoden kaikista matematiikan oppitunneista. Näin tarkasteltuna 1–4 oppituntia ohjelmointia yhden lukuvuoden aikana vaikuttaa vähäiseltä määrältä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta. Toisaalta opettajien arviot ohjelmointiin ja ohjelmoinnilliseen ajatteluun käytettävästä tuntimäärästä yhtä opetusryhmää kohden vastasivat heidän opettamiaan tuntimääriä. Samalla huomioiden, että ohjelmointia oli lähes jokaisessa koulussa opetettu vähintään yhden oppitunnin verran, voidaan opettajien pedagogiseen osaamiseen ja arviointikykyyn luottaen pitää opetettuja tuntimääriä riittävinä.

Ohjelmoinnin ja ohjelmoinnillisen ajattelun oppituntimäärien tarkastelussa suhteessa opettajan työkokemukseen ilmeni, ettei yksikään alle viisi vuotta työelämässä ollut opettaja ollut opettanut ohjelmointia yli kymmentä oppituntia lukuvuonna 2021–2022. Lisäksi kun tarkasteltiin opettajien arvioita ohjelmointiin ja ohjelmoinnillisen ajatteluun käytettävästä tuntimäärästä huomattiin, että mitä enemmän opettajalla oli työkokemusta, sitä suuremmaksi hän arvioi opetuksen käyttävän tuntimäärän olevan. Kuitenkin myös keskihajonnan nähtiin kasvavan samassa suhteessa. Tuloksia voidaan selittää muun muassa sillä, ettei opettajankoulutuksessa tuoda esiin, miten ohjelmointia ja ohjelmoinnillista ajattelua voitaisiin kouluissa opettaa. Koska matematiikan opetus on jo pitkään noudattanut samankaltaista kaavamaisuutta, on pidempään opettajan työtä tehneille opettajille todennäköisemmin muodostunut jo rutiineja sekä käytäntöjä opetukselle. Samaan aikaan vähemmän työkokemusta kerryttänyt opettaja voi olla vasta rakentamassa omaa opettajaminäänsä. Tällöin keskittyminen ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen saattaa jäädä muiden opetussuunnitelman matematiikan sisältöjen varjoon. Tästä kertoi muun muassa Bocconi ym. (2022) tutkimuksessaan nostaten esiin ohjelmoinnin ”kilpailemisen” opetussuunnitelmien muiden painopisteiden kanssa olevan eräs haaste ohjelmoinnin opetukselle. Pidempään työtä tehneet opettajat ovat myös voineet osallistua ohjelmoinnin ja ohjelmoinnillisen ajattelun täydennyskoulutuksiin.

Toinen selittävä tekijä sille, miksi pidempään opettajana toimineet arvioivat ohjelmoinnin opetuksen tuntimäärän suuremmaksi, liittyy opettajan käsityksiin ohjelmoinnista ja ohjelmoinnillisesta ajattelusta. Pidempään työelämässä saattavat nähdä

ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen irrallisena osana matematiikan opetusta. Samalla taas vähemmän aikaa työelämässä olleet, ja ensimmäisenä noudatettavana opetussuunnitelmana laaja-alaisen osaamisen kokonaisuuksien sisältämään opetussuunnitelmaan tutustuneet opettajat voivat nähdä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen sijoittuvan laajemmin myös koulun muuhun opetukseen. Näin ollen arviot käytetystä sekä riittävästä määrästä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta voivat olla pienempiä kuin todellisuudessa opetukseen käytetty tuntimäärä.

Tutkimuksen tulosten mukaan ohjelmoinnin ja ohjelmoinnillisen ajattelun opiskelussa painottuu yksilöllinen työskentely sekä tekemällä oppiminen. Osa opettajista koki, että ohjelmointia ja ohjelmoinnillista ajattelua opittiin parhaiten itse tekemällä ja kokeilemalla. Luonteeltaan oppitunnit noudattivat matematiikan opetukselle tyypillistä kaavaa, jossa tunnin alussa opettaja johdattelee tunnin aiheeseen, minkä jälkeen oppilaat siirtyvät tekemään tehtäviä joko yksin tai yhdessä. Tulokset tukevat Wun ym. (2019) tutkimuksen päätelmiä, jonka mukaan ohjelmointia opitaan parhaiten kirjoittamalla koodia. Lisäksi tuloksissa on yhtenäisyyttä toiseen Wun ym. (2019) tutkimuksen tulokseen, jonka mukaan suomalaiset opettajat suosivat aktiivisia opetusmenetelmiä. Kaiken kaikkiaan tämän tutkimuksen tulokset osoittavat, että opettajat kokivat pääosin heidän käyttämänsä opetusmenetelmät toimivina. Kuitenkin jopa joka neljäs opettaja kertoi olleensa tyytymätön käyttämiinsä opetusmenetelmiin. Suurimmaksi syyksi opetusmenetelmien toimimattomuudelle oli, ettei niiden avulla saa oppilaita motivoitumaan tai keskittymään ohjelmoinnin oppimiseen. Usean opettajan mielestä oli hankala löytää toimiva opetusmenetelmä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen.

Tutkimuksen perusteella Python oli selkeästi opetetuin ohjelmointikieli. Myös Scratchia opetettiin noin joka kolmannessa kouluissa. Verrattuna aikaisempiin tutkimuksiin, tulos on yllättävä, sillä esimerkiksi Wu ym. (2019) tutkimuksen mukaan opettajien ohjelmointiosaaminen koettiin alhaiseksi erityisesti Pythonilla osalta. Vaikka opetussuunnitelmassa ei sanota tarkasti, mitä ohjelmoinnin sisältöjä kouluissa tulisi opettaa, ohjelmointiin liittyviä sisältöalueita opetettiin kouluissa kattavasti. Yli 80 % kouluista opetettiin muuttujiin, ehtolauseisiin, syötteen lukemiseen ja tulostukseen tai toisto- / silmukkarakenteisiin liittyviä asioita. Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa keskityttiin selvästi perusteiden ja alkeiden oppimiseen, josta kertoo muun muassa se, että listoihin tai funktioihin liittyviä sisältöjä opetettiin huomattavasti vähemmän. Kun opetettuja

sisältöjä peilataan ohjelmoinnin ja ohjelmoinnillisen ajattelun oppituntimääriin, on selvää, ettei vähäinen tuntimäärä riitä haastavampien sisältöjen kuten listojen ja funktioiden opettamiseen.

Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen tukena käytettiin laajasti eri oppimateriaaleja ja ohjelmointiympäristöjä tai -sivustoja, eikä mikään yksittäinen materiaali tai sivusto erottunut muista. Perinteisten oppimateriaalien kuten oppikirjojen käyttö oli suosittua. Tämän tutkimuksen perusteella moni opettaja kertoi myös käyttäneensä itse tehtyä materiaalia opetuksessaan, vaikka aikaisempi tutkimus (ks. Tanhua-Piironen ym. 2020) osoittaa, että yleisesti ottaen opettajat suosivat opetuksessaan valmiita materiaaleja. Kuitenkin tutkimuksen aikaisempi tulos osoitti, ettei ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukselle ole löydetty motivoivia opetusmenetelmiä. Tämän vuoksi opettajat ovat saattaneet turvautua itse tehtyyn materiaaliin.

Ohjelmointisivustoista ja -ympäristöistä suosituimmaksi osoittautui Scratch, mutta myös code.org, koodaustunti, tie kodariksi -sivusto ja ViLLE olivat monen opettajan käytössä. Erilaisten sivustojen käytön mahdollisti se, että tutkimukseen osallistuneista kouluista kolmea koulua lukuun ottamatta oli käytössä tietokoneet. Opetuksessa käytettiin myös ohjelmitavaa elektroniikkaa kuten micro:bitteja ja Arduinoja, joita tarjottiin joka kolmannessa kouluista. Robottien tarjonta oli edelleen varsin suurta. Leinon ym. (2019) vuoden 2018 tutkimuksen mukaan 66 % kouluista mahdollistivat robottien käytön opetuksessa ja edelleen tämän tutkimuksen perusteella robotteja tarjotaan yli puolissa kouluista. Ohjelmitavista roboteista suosituin oli Lego Mindstorms. Syy siihen, miksi digitaalisia oppimateriaaleja ja -ympäristöjä käytettiin runsaasti, saattaa johtua ohjelmoinnin luonteesta, sillä ohjelmointia pidetään usein ohjeiden kirjoittamisena tietokoneelle. Toisena syynä voidaan nähdä koulutuksen digitalisaation murros, joka vaikuttanut vahvasti myös perusopetukseen. Ei siis ole lainkaan yllättävää, että ohjelmoinnin opetuksessa käytetään runsaasti erilaisia digitaalisia ympäristöjä.

Koulun tarjoamia laitteita ja resursseja ympäröi kuitenkin suuri huoli niiden riittävydestä ja toimivuudesta. Vain reilu puolissa kouluista niitä pidettiin riittävinä ja niiden koettiin tukevan ohjelmoinnin opetusta. Laitteiden ja resurssien riittämättömyyden isoimmiksi syiksi osoittautui laitteiden puutteellisuus sekä opettajien oman heikko osaaminen. Vaikka tämän tutkimuksen pohjalta voidaan sanoa matematiikan opettajien tietävän, mitä tarkoitetaan

käsitteillä ohjelmointi ja ohjelmoinnillinen ajattelu, voi opettajien ohjelmointiosaaminen olla edelleen aikaisempien tutkimuksiin (ks. Tanhua-Piironen ym. 2020; Wu ym. 2019) nojaten heikkoa. Opettajien omien sanojensa mukaan he tarvitsisivat lisää koulutusta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksesta. Nyt vain joka kolmannes koulu tarjoaa mahdollisuutta osallistua ohjelmoinnin täydennyskoulutuksiin tai ohjelmointikursseille.

Tutkimuksen tulokset kertovat siitä, että ohjelmointia ja ohjelmoinnillista ajattelua opetetaan kouluissa opettajien mukaan riittävästi. Tutkimuksen perusteella on havaittavissa, että opetus noudattaa matematiikan opetukselle tyypillistä kaavamaisuutta, jossa oppitunti alkaa opettajajohtoisella selostuksella, minkä jälkeen oppilaat siirtyvät ratkaisemaan tehtäviä. Yksilöllinen työskentely ja tekemällä oppiminen ovat ensisijaisia tapoja ohjelmoinnin perustaitojen oppimiseen. Ohjelmointia opetetaan niin oppikirjoja kuin erilaisia sähköisiä ympäristöjä käyttäen, joita koulut tarjoavat opettajien näkemysten mukaan riittävästi. Lisäksi positiivisesta kehityksestä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen kentällä kertoo opettajien kuvailut käsitteistä ohjelmointi ja ohjelmoinnillinen ajattelu. Kuitenkin tämän tutkimuksen perusteella opettajat tarvitsevat lisää koulutusta ja selkeyttä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen. Puutteellista tai kokonaan puuttuva koulutusta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksesta ei kuitenkaan voida enää pitää esteenä opetukselle (ks. Tanhua-Piironen ym. 2020).

Itse ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen syventyessäni yhdyn Tanhua-Piironen ym. (2020, 95) näkemykseen siitä, että ohjelmoinnin opetukseen tulee vahvistaa selkeät sisällöt sekä selkeyttää ohjelmoinnin opetusvastuun määrittelyä. Ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen tarvitaan konkreettisia neuvoja sekä malleja ja tämä mahdollistetaan lisäämällä rahoitusta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen kehittämiseen. Muutoksia tulee tehdä niin opetussuunnitelmaan kuin opettajakoulutukseen, jossa ei tällä hetkellä juuri kerrota, miten ohjelmointia ja ohjelmoinnillista ajattelua voitaisiin kouluissa opettaa. Suomessa riittää motivoituneita opettajia, joten tuetaan heitä myös ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa.

7.2 Tutkimuksen rajoitukset ja luotettavuus

Eräänä kyselytutkimuksen heikkoutena on, ettei sen avulla voida varmistua vastaajan suhtautumisesta tai perehtyneisyydestä kyselylomakkeen aiheeseen (Hirsijärvi ym. 2009, 195). Tämän heikkouden vaikuttavuutta pyrittiin tässä tutkimuksessa lieventämään kysymällä,

miten vastanneet määrittelisivät käsitteet ohjelmointi ja ohjelmoinnillinen ajattelu.

Tutkimuksen luotettavuutta kuitenkin heikensi epävarmuus opettajien pätevyydestä sekä siitä vastasivatko he kyselylomakkeen kysymyksiin lukuvuoden 2021–2022 matematiikan oppituntien pohjalta. Vaikka lukuvuotta painotettiin useassa kyselylomakkeen kysymyksessä, mainittiin tutkimuksen keskittyvän juuri matematiikan oppitunteihin ainoastaan kyselyn alussa. Lisäksi kyselylomakkeen linkkiä jaettiin vain Facebook ryhmissä ja MAOL-uutiskirjeessä, mikä rajaa ne opettajat ulos tutkimuksesta, jotka eivät kuulu ryhmiin tai eivät ole MAOL ry:n jäseniä. Samalla on mahdollista, että kyselyyn vastasivat vain ne opettajat, jotka olivat opettaneet ohjelmointia, eikä täten otos välttämättä kuvaa koko matematiikan yläkoulun opettajien joukkoa.

Tutkimus käsittelee vain matematiikan oppitunteja lukuvuonna 2021-2022. Tämä rajausta ja vähäinen määrä aikaisempia tutkimuksia aiheesta hankaloittaa laadukkaiden johtopäätösten muodostamista. Ilman jatkotutkimuksia ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksesta perusopetuksessa on vaikeaa johtaa yleistettäviä päätelmiä.

7.3 Jatkotutkimusehdotukset

Tutkimus ei ota kantaa oppilaiden tai opettajien ohjelmointiosaamiseen eikä opetuksen laatuun. Jatkotutkimuksena voitaisiin tutkia, onko esimerkiksi oppilaiden tai opettajien ohjelmointiosaamisessa tapahtunut muutosta. Lisäksi voitaisiin tutkia ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen laatua sekä sitä, onko ohjelmointiosaamisella vaikutusta opetuksen laatuun. Laajempi tutkimus ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksesta perusopetuksessa toisi myös hedelmällistä tietoa siitä, miten ohjelmoinnin ja ohjelmoinnillisen ajattelun opetus on järjestetty muissa kuin matematiikan oppiaineessa. Erityisesti ohjelmoinnillisen ajattelun opetusta yläkoulussa olisi mielenkiintoista tutkia opetussuunnitelman laaja-alaisten osaamisen kannalta.

8 Yhteenveto

Ohjelmointi ja ohjelmoinnillinen ajattelu lisättiin osaksi vuoden 2014 perusopetuksen opetussuunnitelman perusteita. Konkreettisimmin lisäys näkyi matematiikan opetuksessa, johon ohjelmoinnin ja ohjelmoinnillisen ajattelun opetus vahvasti kouluissa painottuu. Vaikka ohjelmointia ja ohjelmoinnillista ajattelua on ehditty kouluissa opettaa jo useamman vuoden ajan, ei opetukselle ole vielä kukaan vakiintunut toimivia opetustapoja. Tämä oli eräs syy tälle kyselytutkimukselle, jonka tarkoituksena oli tutkia, miten ohjelmoinnin ja ohjelmoinnillisen ajattelun opetus oli yläkoulujen matematiikan oppitunneilla järjestetty. Tutkimuksessa haluttiin myös selvittää, onko opetuskentällä tapahtunut muutoksia aikaisempien tutkimusten tuloksiin, joiden mukaan ohjelmointia ja ohjelmoinnillista opetusta opetettiin kouluissa vaihtelevasti.

Tämän tutkimuksen tulokset osoittavat, että ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksessa on tapahtunut kehitystä viime vuosina. Ohjelmointia ja ohjelmoinnillista ajattelua opetettiin yläkoulujen matematiikan oppitunneilla opettajien mukaan riittävästi; yhtä opetusryhmää kohden keskimäärin 5–10 oppituntia lukuvuonna 2021–2022. Tunnit noudattivat pitkälti matematiikan opetukselle tyypillistä mallia, jossa oppitunti alkaa opettajajohtoisella ohjeistuksella, jonka jälkeen oppilaat siirtyvät tekemään tehtäviä. Ohjelmoinnin perustaitojen ja -sisältöjen opetuksessa korostuikin erityisesti tekemällä oppiminen sekä yksilöllinen työskentely. Eri oppimateriaaleja ja ohjelmointiympäristöjä hyödynnettiin oppitunneilla monipuolisesti eikä mikään yksittäinen materiaali tai ympäristö erottunut muiden joukosta. Myös koulujen tarjoamia resursseja ohjelmoinnin ja ohjelmoinnilliseen ajatteluun pidettiin varsin riittävinä. Kuitenkin huoli laitteiden toimivuudesta varjosti ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta. Lisäksi opettajat kokivat edelleen tarvitsevänsä tukea niin ohjelmoinnin ja ohjelmoinnillisen ajattelun taitojen oppimiseen kuin opetukseen.

Vaikka positiivista kehitystä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen kentällä on selvästi tapahtunut, tarvitaan opetustyöhön edelleen selkeyttä sekä konkreettisia neuvoja ja ohjeita siihen, miten opettaa ohjelmointia ja ohjelmoinnillista ajattelua. Muutoksia tarvitsee tehdä niin opetussuunnitelmaan sekä opettajankoulutukseen ja tämä mahdollisesta muun muassa lisäämällä rahoitusta ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen kehittämiseen.

Lähteet

- Aho, A. (2012). Computation and Computational Thinking. *The Computer Journal*, 55 (7), 832–835.
- Arduino (2018) What is Arduino? <https://www.arduino.cc/en/Guide/Introduction> (luettu 12.12.2022.)
- Arduino (N.d.). Arduino Education. <https://www.arduino.cc/education> (luettu 12.12.2022.)
- Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning and leading with technology*, 38, 20–23.
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V. & Stupurienė, G. (2022). Reviewing Computational Thinking in Compulsory Education. *Teoksessa Inamorato dos Santos, A., Cachia, R., Giannoutsou, N. & Punie, Y. (toim.) Publications Office of the European Union. Luxembourg.*
- Codecademy (N.d.). About Codecademy. <https://www.codecademy.com/about> (luettu 22.11.2022.)
- Code.org (2023). What is Code.org? <https://support.code.org/hc/en-us/articles/204784827-What-is-Code-org-> (luettu 31.3.2023.)
- Code.org (N.d.). About Us. <https://code.org/international/about> (luettu 22.11.2022.)
- Edita (N.d.). <https://oppiminen.edita.fi/ylakoulu/sade-oppimateriaalisarja/> (luettu 22.3.2023.)
- European Schoolnet (N.d.). Who we are <http://www.eun.org/about> (luettu 18.11.2022.)
- Fagerlund, J. (2022a). Tuoreita näkökulmia ohjelmoinnin ja ohjelmoinnillisen ajattelun opetukseen koulussa. *Dimensio-lehti*, 10.2.2022. <https://dimensiolehti.fi/tuoreita-nakokulmia-ohjelmoinnin-ja-ohjelmoinnillisen-ajattelun-opetukseen-koulussa/>. (luettu 2.11.2022.)
- Fagerlund, J. (2022b). Tietokonevallankumous ja ohjelmoinnillinen ajattelu peruskoulussa : havaintoja mikro- ja makrotasolta. *Kasvatus ja aika*, 16 (1), 121–127.
- Fagerlund, J. (2021). Teaching, Learning and Assessing Computational Thinking through Programming with Scratch in Primary Schools. Väitöskirja. Opettajankoulutuslaitos, Jyväskylän yliopisto. <https://jyx.jyu.fi/handle/123456789/78190>.
- Francisco, B., Casallas, R., Hernandez Hoyos, M., Reyes, A., Restrepo, S. & Danies, G. (2017). Changing a Generation’s Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research*. 87 (4), 834–860.

- Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42 (1), 38–43. <http://www.jstor.org/stable/23360476>.
- Hello Ruby (N.d.). <http://www.helloruby.com/fi> (luettu 3.2.2023.)
- Hirsijärvi, S., Remes, P. & Sajavaara, P. (2009). Tutki ja kirjoita. Helsinki: Tammi.
- Kaila, E., Kaarto, H. & Laine, H. (2021). Opettajan opas. Ohjelmoinnin peruskurssi yläkouluun. (luettu 22.11.22.)
- Kaila, K. & Kurvinen, E. (2018). ViLLE Opettajan kirja. (Saatavilla: <https://www.oppimisanalytiikka.fi/ville/ohjeet/>) (luettu 22.11.2022.)
- Leino, K., Rikala, J., Puhakka, E., Niilo-Rämä, M., Sirén, M., & Fagerlund, J. (2019). Digiloikasta digitaitoihin : kansainvälinen monilukutaidon ja ohjelmoinnillisen ajattelun tutkimus (ICILS 2018). <https://jyx.jyu.fi/handle/123456789/66250>.
- Lindberg, R., Laine, T. & Haaranen, L. (2018). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games: Gamifying programming education in K-12. *British Journal of Educational Technology*, 50 (3), 1979–1995.
- Mehackit (N.d.). Projekt: Raspberry Pi. https://mehackit.org/kurssit/robotiikka_tulevaisuudessa/02-internet_of_things/04-raspberry_pin_testaaminen/ (luettu 12.12.2022.)
- Mertala, P., Palsa, L. & Slotte Dufta, T. (2020). Monilukutaito koodin purkajana: Ehdotus laaja-alaiseksi ohjelmoinnin pedagogiikaksi. *Media & Viestintä*, 43 (1).
- micro:bit (2021) What is a micro:bit? <https://support.microbit.org/support/solutions/articles/19000013983-what-is-a-micro-bit-> (luettu 23.11.22.)
- micro:bit (N.d.). Let's code. <https://microbit.org/code/> (luettu 17.11.22.)
- Mykkänen, J. & Liukas, L. (2014). Koodi 2016: Ensiapua ohjelmoinnin opetukseen peruskoulussa. Helsinki.
- Nyssölä, K. & Kumpulainen, T. (2020). Perusopetuksen ja kouluverkon tulevaisuuden näkymiä. Opetushallitus. Raportit ja selvitykset 2020:25.
- Opetushallitus. (2014). Perusopetuksen opetussuunnitelman perusteet 2014. Helsinki.
- Opetushallitus (N.d.). Perusopetuksen opetussuunnitelman perusteet <https://www.oph.fi/fi/koulutus-ja-tutkinnot/perusopetuksen-opetussuunnitelman-perusteet> (luettu 2.2.2023.)

- Otava (N.d.). <https://oppimisenpalvelut.otava.fi/oppimateriaalit/luokat-7-9/aareton/> (luettu 22.3.2023.)
- Robogem (N.d.). <https://robogem.fi/> (luettu 3.2.2023.)
- Sanoma Pro (N.d.). <https://www.sanomapro.fi/sarjat/kuutio/> (luettu 22.3.2023.)
- Tanhua-Piiroinen, E., Kaarakainen, S.-S., Kaarakainen, M.-T. & Viteli, J. (2020). Digiaajan peruskoulu II. Opetus- ja kulttuuriministeriön julkaisuja 2020: 17.
- Tuomi, J. & Sarajärvi, A. (2012). Laadullinen tutkimus ja sisällönanalyysi. Helsinki: Tammi.
- Turun kaupunki (2016). Turun perusopetuksen opetussuunnitelma luvut 1–12. (Saatavilla: <https://blog.edu.turku.fi/ops2016/hyvaksytyt/paikalliset-suunnitelmat/>) (luettu 3.2.2023.)
- Valli, R. (2010). Kyselylomaketutkimus. Teoksessa Aaltola, J. & Valli, R. (toim.) Ikkunoita tutkimusmetodeihin 1. Metodien valinta ja aineistonkeruu: virikkeitä aloittelevalle tutkijalle. Jyväskylä: PS-kustannus. 103–127.
- Valtioneuvoston asetus perusopetuslaissa tarkoitetun opetuksen valtakunnallisista tavoitteista ja perusopetuksen tuntijaosta 793/2018.
<https://www.finlex.fi/fi/laki/alkup/2018/20180793>.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49 (3), 33–35.
- Wu, L., Looi, C.-K., Multisilta, J., How, M.-L., Choi, H., Hsu, T.-C. & Tuomi, P. (2019). Teacher's Perceptions and Readiness to Teach Coding Skills: A Comparative Study Between Finland, Mainland China, Singapore, Taiwan, and South Korea. *The Asia - Pacific Education Researcher*, 29 (1), 21–34.
- W3Schools (N.d.). About W3Schools. <https://www.w3schools.com/about/default.asp> (luettu 22.11.2022.)


Liitteet

Liite 1. Kyselylomake



Turun yliopisto
University of Turku

Ohjelmointi ja ohjelmoinnillinen ajattelu yläkoulun matematiikan opetuksessa

 Pakolliset kysymykset merkitty tähdellä (*)

Tämän kyselyn tarkoituksena on selvittää kuinka paljon ja millä menetelmillä sekä välineillä ohjelmointia opetetaan matematiikan oppitunneilla peruskoulun 7-9 luokilla. Lisäksi selvitetään, mitkä ovat koulujen resurssit ohjelmoinnin opetuksen tueksi. Kyselyn vastauksia hyödynnetään Turun yliopiston pro gradu -tutkielmassa. Tutkimus keskittyy erityisesti matematiikan oppitunneilla tapahtuvaan ohjelmoinnin opetukseen, joten vastaathan kysymyksiin matematiikan oppituntien pohjalta.

Vastaukset käsitellään anonymisti. Kyselyyn vastaaminen vie noin 5-15 minuuttia.

Kiitos ajastasi!

TAUSTATIEDOT

Missä kunnassa opetat? *

Kuinka monta oppilasta oppilaitoksessasi opiskelee? *

- alle 100
- 100 - 299
- 300 - 499
- yli 500

Työkokemuksesi vuosina. *

- alle 5 vuotta
- 5 - 14 vuotta
- 15 - 24 vuotta
- yli 25 vuotta

OHJELMOINTI JA OHJELMOINNILLINEN AJATTELU

Tiedän mitä käsitteet ohjelmointi ja ohjelmoinnillinen ajattelu tarkoittavat. *

- täysin samaa mieltä
- osittain samaa mieltä
- en osaa sanoa
- osittain eri mieltä
- täysin eri mieltä

Kerro omin sanoin, mitä ohjelmoinnillisella ajattelulla mielestäsi tarkoitetaan? *

Kerro omin sanoin, mitä ohjelmoinnilla mielestäsi tarkoitetaan? *

OHJELMOINNIN OPETUKSEN MÄÄRÄ

Jatkossa käsitteellä ohjelmointi viitataan myös ohjelmoinnilliseen ajatteluun.

Olen opettanut ohjelmointia viime lukuvuonna 2021-2022. *

- Kyllä
- En
- En osaa sanoa

Kuinka monta oppituntia olet opettanut ohjelmointia viime lukuvuonna 2021-2022? *

Anna arvio oppituntien lukumäärästä yhtä opetusryhmää kohden.
Yksi oppitunti vastaa 45 min opetusta.

- alle 5 oppituntia
- 6 - 10 oppituntia
- 11 - 15 oppituntia
- 16 - 20 oppituntia
- yli 20 oppituntia

Mitkä ovat suurimmat syyt sille, että et ole opettanut ohjelmointia? *

Mikä on mielestäsi riittävä ohjelmoinnin opetukseen käytettävä oppituntimäärä yhden lukuvuoden aikana yhtä opetusryhmää kohden. *

Yksi oppitunti vastaa 45 min opetusta.

Anna arviosi kokonaislukuna.

oppituntia

OHJELMOINNIN SISÄLLÖT JA OPETUSMENETELMÄT

Mitä seuraavista opetusmenetelmistä tai työskentelyn muodoista olet käyttänyt ohjelmoinnin opetuksessa viime lukuvuonna 2021-2022? *

Valitse kaikki käyttämäsi opetusmenetelmät.

- Yksilöllinen työskentely
- Ryhmätyöskentely
- Tekemällä oppiminen
- Oppimispelit
- Projektityöskentely
- Ongelmalähtöinen oppiminen
- Käänteinen opetus (en. flipped classroom)
- Työskentely koulun ulkopuolisissa ympäristöissä
- Jokin muu, mikä?

Ovatko käyttämäsi opetusmenetelmät mielestäsi toimivia? Miksi? *

Mitä seuraavista ohjelmointikielistä olet opettanut viime lukuvuonna 2021-2022? *

Valitse kaikki opettamasi ohjelmointikielet.

- Python
- Scratch
- Java
- JavaScript
- C#
- HTML
- En ole opettanut ohjelmointikieliä
- Jokin muu, mikä?

Mitä seuraavista ohjelmoinnin sisällöistä olet opettanut viime lukuvuonna 2021-2022? *

Valitse kaikki opettamasi sisällöt.

- Muuttujat (esim. tiedon tallentaminen muuttujaan)
- Syötteen lukeminen ja tulostus (esim. ohjelman käyttäjän kirjoittaman tekstin lukeminen ja sen tulostaminen näytölle)
- Toisto- / Silmukkarakenteet (esim. while- ja for-loop)
- Ehtolauseet (esim. if- ja if-else -lauseet)
- Listat (esim. muuttujan tallentaminen listaan uudeksi alkioksi)
- Funktiot / Aliohjelmat
- En osaa sanoa, mitä ohjelmoinnin sisältöjä olen opettanut
- Jotain muuta, mitä?

Mitä oppimateriaaleja, ohjelmointiympäristöjä tai -sivustoja olet käyttänyt ohjelmoinnin opetuksen tukena viime lukuvuonna 2021-2022? *

Valitse kaikki käyttämäsi materiaalit, ympäristöt ja sivustot.

- Fyysinen oppikirja
- Sähköinen oppikirja
- Scratch
- ViLLE
- Code.org
- Codeacademy
- Koodaustunti (Hour of Code)
- Lego Mindstorms
- W3Schools
- micro:bit
- Raspberry Pi
- Arduino
- Jotain muuta, mitä?

Kerro tai kuvaile tarkemmin minkälainen on tyypillinen ohjelmointiin keskittyvä oppitunti? *

KOULUJEN RESURSSIT

Mitä laitteita ja resursseja koulullasi on tarjota ohjelmoinnin opetusta varten? *

Valitse kaikki tarjottavat laitteet ja resurssit.

- tietokoneet
- tabletit
- ohjelmitava elektroniikka (micro:bit, Raspberry Pi, Arduino, ...)
- ohjelmitavat robotit (Lego Mindstorms, Robbo, ...)
- mahdollisuus, että ohjelmoinnin opetus on järjestetty koulun ulkopuolisen tahon puolesta
- mahdollisuus yhteisopettajuuteen
- mahdollisuus opettajille osallistua ohjelmoinnin täydennyskoulutuksiin tai ohjelmointikursseille
- koulu tarjoaa oppilaille ohjelmointia valinnaisaineena tai -kursseina
- jotain muuta, mitä?

Ovatko koulusi tarjoamat laitteet ja resurssit mielestäsi riittävät ohjelmoinnin opetukseen? *

- Kyllä
- Ei
- En osaa sanoa

Kerro tai kuvaile tarkemmin, miten olemassa olevat laitteet ja resurssit vaikuttavat ohjelmoinnin opetukseesi? *

“Believe you can then you will”

– Mulan