

TARTU ÜLIKOOL

Arvutiteaduse instituut

Matemaatika- ja informaatikaõpetaja õppekava

**Kaisa Liina Keps**

**Õpetajamaterjal programmeerimise  
õpetamiseks II ja III kooliastmes**

**Magistritöö (15 EAP)**

Juhendaja: Reelika Suviste

Tartu 2025

# Õpetajamaterjal programmeerimise õpetamiseks II ja III kooliastmes

## Lühikokkuvõte:

Programmeerimise õppimine on oluline õpilaste probleemilahendamise oskuse ja raalmõtlemise arenguks, kuid õpetajad ei soovi sageli programmeerimist õpetada materjalide puuduse tõttu. Magistritöö raames loodi II ja III kooliastme õpilastele suunatud digitaalsele programmeerimisõpikule *progema.ee* õpetajamaterjalid, et aidata kaasa programmeerimise õpetamisele põhikoolis ja vähendada õpetajate töökoormust. Materjali loomisel tugineti ADDIE mudelile, mille alusel analüüsiti programmeerimise õpetamise ja õppimise õppemeetodeid, kavandati tundide ülesehitus ja tunnikava struktuur ning loodi vajalikud lisamaterjalid. Materjalide loomist toetasid erinevad veebikeskkonnad ning link õpetajamaterjalide lisati ka digitaalsesse programmeerimisõpikusse. Õpetajate ja üliõpilaste tagasiside alusel tehti materjalides täiendusi, hinnati loodud materjalide kvaliteeti ning kaardistati piirangud ja edasiarenduse võimalused.

## Võtmesõnad:

Programmeerimine, teine ja kolmas kooliaste, veebirakendus, õpetajamaterjal, ADDIE

CERCS: S270 Pedagoogika ja didaktika, P175 Informaatika, süsteemiteooria

## Materials for Teaching Programming at the Second and Third School Level

### Abstract:

Learning programming is essential for developing students' problem-solving and frame thinking skills, but teachers are often reluctant to teach programming because of a lack of materials. This master's thesis aimed to create materials for teachers to contribute to the teaching of programming and to reduce the workload of teachers. The materials were aimed for the students of the second and third school level and developed using the ADDIE model, which was used to analyze teaching and learning methods for programming, to design the lesson structure and to create the necessary materials. The creation of the materials was supported by different online environments and a link to the teacher materials was included in the web application *progema.ee*. Based on feedback from teachers and students, improvements were made to the materials, the quality of the materials was assessed, and limitations and opportunities for further development were identified.

### Keywords:

Programming, second and third school level, web application, teaching material, ADDIE

**CERCS:** S270 Pedagogy and didactics, P175 Informatics, systems theory

## Sisukord

Sissejuhatus.....	6
1. Programmeerimise õpetamine .....	7
1.1 Õppemeetodid programmeerimise õpetamiseks .....	8
1.1.1 Paarisprogrammeerimine .....	8
1.1.2 Vastastikune õppimine.....	9
1.1.3 Ümberpööratud klassiruum.....	10
1.1.4 Mängupõhine õpe.....	11
1.2 Programmeerimise õppimine ja AI.....	12
1.3 Motivatsioon programmeerimise õppimiseks.....	12
1.4 Õppeühiku mudel ja õppetöö planeerimine .....	13
2. Õpetajamaterjalide loomine .....	14
2.1 Olemasolevad õpetajamaterjalid programmeerimise õpetamiseks.....	14
2.2 ADDIE mudel.....	14
2.3 Valim.....	15
2.4 Töö protseduur .....	15
2.4.1 Analüüsimine .....	15
2.4.2 Kavandamine .....	16
2.4.3 Materjali väljatöötamine .....	18
2.4.4 Tagasisideküsimumstiku koostamine .....	19
2.4.5 Tagasiside kogumine .....	20
2.4.6 Täienduste ja paranduste tegemine .....	20
3. Tulemused ja arutelu.....	21
3.1 Andmetötluse meetodid .....	21
3.2 Teemade kohta kogutud tagasiside .....	21
3.2.1 Sissejuhatus.....	21

3.2.2	Algoritm.....	23
3.2.3	Muutujad ja avaldised.....	24
3.2.4	Tingimuslaused.....	25
3.2.5	Korduslaused.....	26
3.2.6	Funktsioonid.....	27
3.2.7	Failid.....	28
3.2.8	Veahaldus.....	29
3.2.9	Projekti tutvustus ja AI.....	30
3.2.10	Järjendid.....	31
3.2.11	Sõnastikud.....	32
3.2.12	Hulgad.....	33
3.2.13	Mitmemõõtmelised andmestruktuurid.....	34
3.2.14	Projekt.....	35
3.2.15	Mängud.....	36
4.	Piirangud ja edasiarendusvõimalused.....	38
	Kokkuvõte.....	39
	Viidatud kirjandus.....	40
	Lisad.....	50
	I. Tagasisideküsimustik.....	50
	II. Ekspertide soovitusel ja täienduste ettepanekud.....	52
	III. Litsents.....	53

## Sissejuhatus

Programmeerimine on õppeaine, mis on muutumas üha populaarsemaks ning üle kolmandiku õpilastest soovib oma tuleviku siduda informaatikaga [1], kuid põhikooli riiklik õppekava katab programmeerimise teemat ainult informaatika valikkursuse raames [2]. Mitmed koolid programmeerimist ei õpeta, kuid rohkem kui pooled koolid osalevad I ja II kooliastme õpilastega programmides ProgeTiiger ja SmartLabs [3], mis viitab programmeerimise populaarsusele. Samuti leiavad mitmed spetsialistid, et peamiste programmeerimise kontseptsioonidega tutvumine enne ülikooli on eduvõti tulevikuks [4].

Programmeerimise õpetamisel on põhilisteks probleemideks õpetajate puudulikud teadmised ning õppematerjalide, hindamismudelite, erialaste koolituste ja õpetamismeetodite nappus [5]. Sageli õpetatakse klassis programmeerimist loenguformaadis, kasutades eelnevalt kirjutatud koodi [6], kuid paberil olevate või staatiliste materjalide kasutamine pole tõhus dünaamiliste programmeerimise probleemide lahendamise õpetamisel [7]. Mõned õpilased ei suuda näha programmide dünaamilisust, kui nende selgitamiseks on kasutatud pabermaterjale [8].

Programmeerimine arendab loomingu- ja kriitilist mõtlemist ning probleemi lahendamise oskust [9], mistõttu programmeerimise õpetamine on kasulik õpilase arengule. Seetõttu oleks vaja programmeerimise õpetamist koolis toetada erinevate eestikeelsete materjalidega nii õpilastele kui ka õpetajatele. Eestikeelse materjali vajalikkust näitab ka teadmine, et Tartu Ülikooli programmeerimise õpiku [10] kasutamine kasvas, kui sellele lisati õpetajamaterjal.

Selleks, et rohkem õpetajad oleksid valmis programmeerimist õpetama ja et vähendada alustavate õpetajate töökoormust, on vaja luua rohkem tugimaterjale, mis tutvustaks erinevaid meetodilisi võimalusi programmeerimise õpetamisel. Sellest tulenevalt seati magistritöö eesmärgiks luua õpetajamaterjale, mis hõlmaks erinevaid meetodeid ja ülesandeid, eestikeelsele programmeerimise digiõpikule (<https://progema.ee/>). Eesmärgi saavutamise toetamiseks püstitati kaks uurimisküsimust:

1. Millised õppemeetodid on tõhusad programmeerimise õppimisel ja õpetamisel?
2. Kuidas hindavad loodud materjale eksperdid?

Töö on jagatud nelja peatükki. Esimeses peatükis antakse ülevaade programmeerimise õpetamisest ja õppemeetoditest. Teises peatükis kirjeldatakse töö protseduuri läbi ADDIE mudeli. Kolmandas peatükis saab tutvuda tulemuste ja aruteluga ning neljandas peatükis lugeda piirangute ja edasiarenduste võimaluste kohta. Lisades on võimalik tutvuda tagasisideküsimustiku ja ekspertide antud soovitusetega.

## 1. Programmeerimise õpetamine

Tehnoloogiaajastul täidab programmeerimine suurt rolli inimeste igapäevaelus. Kaudselt puutuvad programmeerimisega kokku kõik, kes tehnoloogilisi vahendeid ja erinevaid programme kasutavad, kuid paljude jaoks on programmeerimine töö. Programmeerimise õppimine sobilike õppekavade järgi arendab aga raalmõtlemist [11], mis on oluline oskus ka neile, kes oma tulevikku informaatikaga siduda ei plaani.

Programmeerimine on probleemilahendamise protsess, mille käigus kasutaja annab arvutile käsklusi kirjutatud koodi abil [12]. Raalmõtlemine on üldine probleemilahendamise oskus koos kommunikatsiooni, kriitilise mõtlemise ja loovusega [13–15]. Raalmõtlemine on ka matemaatika alustala [16], mistõttu programmeerimise õpetamine võiks aidata kaasa ka matemaatika-alaste teadmiste ja oskuste arendamisele. Eestis on põhikooli lõpetamiseks vaja sooritada eksam eesti keeles, ühes õppeaines omal valikul ja matemaatikas [17]. Kraufmann ja Stenseth leidsid, et programmeerimist õppinud õpilased argumenteerivad rohkem süvitsi ning mitteõppijad pinnapealsemalt [18], mis võib anda programmeerimist õppivatele õpilastele eelise ka matemaatikatunnis ning tagada paremad tulemused eksamil.

Programmeerimise õppimine ei ole sageli lihtne [19,20] ning programmeerimiseks loodud keeli on mitmeid. 2020. aastal olid enim kasutatavad programmeerimiskeeled JavaScript, HTML/CSS, SQL, Python, Java, Bash/Shell/Powershell, C#, PHP, TypeScript, C++ ja C, kuid keelte kasutamise populaarsus erineb ka kuu lõikes, mistõttu kõige kasutatumat keelt üheselt välja tuua ei saa [21]. Programmeerimise õppimine toob aga õppijale kestva kasu, sest isegi siis, kui aastate pärast on õppimisel kasutatud keel vananenud, jääb õppijale programmeerimise käigus arenenud probleemilahendamise oskus [22].

Selleks, et soodustada programmeerimise õppimist, peaks õppimine toimuma emakeeles ja emakeelsete materjalidega. II kooliastme õpilastel on programmeerimise õppimist takistavateks teguriteks sageli vanemate kehtestatud arvutikasutamise piirangud ning puudulik inglise keele oskus, mis pole piisav iseseisvaks õppimiseks [23].

Väikestel riikidel, mille ressursid on piiratud, on avalikud materjalid olulisel kohal ning neid peaks vaatlema kui võimalust hariduse kvaliteeti tõsta ning muuta see kättesaadavamaks kõigi jaoks [24]. Eestikeelsete materjalide väljatöötamist on juhtinud Tartu Ülikooli arvutiteaduste instituudi töörühm, kes on välja andnud gümnaasiumi programmeerimise õpiku [10], mis toetab gümnaasiumi programmeerimise valikkursuse [25] õppimist ja õpetamist kasutades programmeerimiskeelt Python.

Eestis ei õpeta paljud koolid programmeerimist, küll aga kaasatakse I ja II kooliastme õpilastele programmeerimisega tutvumiseks programme nagu SmartLabs ja ProgeTiiger [3]. ProgeTiigri programm toetab informaatika, sealhulgas programmeerimise õppematerjalide arendamist, juhendajate ja õpetajate koolitamist ning populariseerib programmeerimist, kaasates 2018. aastal programmi 87% Eesti põhikoolidest ja 60% lasteaedadest [26]. ProgeTiigri kogumikust leiab materjale näiteks robootika õpetamiseks, informaatika digiõpiku I ja II kooliastmele, informaatika valikkursused III kooliastmele ning erinevatele programmeerimiskeskondadele suunatud vahendeid [27].

Programmeerimise õppimine edendab loovust, koostööd, suhtlemist ja sisuloomet [28], mistõttu peaks programmeerimise rolli ka üldhariduskoolides suurendama. OECD soovituslike põhimõtete hulgas õppekavade arendamiseks leiab ka, et aastal 2030 peaks õppekava toetama raalmõtlemine ja programmeerimise arendamist [29] ning ka õpetajad näevad programmeerimises võimast tööriista, mis aitab loimida õppeaineid, arendab probleemilahendamise oskust ning tõstab õpilaste kaasahaaratust [30]. Eestikeelsed programmeerimisõpikud on sageli mõeldud gümnaasiumitele või ülikoolidele ning õpetajamaterjale sisaldavad neist vähesed, kuid emakeeles õppimine on oluline maksimaalse kasuteguri tagamiseks. Programmeerimise õppimise käigus omandatavad oskused, nagu probleemilahendamine, kommunikatsioon, koostöö ja kriitiline mõtlemine on niinimetatud 21. sajandi oskused, mille nõutus ja tähtsus üha kasvab [31]. Põhikooli eesmärk on toetada õpilasi aktiivseks ja teadlikuks kodanikuks kasvamisel ning arendada tulevikuks vajalikke oskusi, et noored suudaksid end täisväärtuslikult teostada ka tööl [29], mistõttu on eelnimetatud oskuste arendamine oluline juba põhikoolis.

## **1.1 Õppemeetodid programmeerimise õpetamiseks**

Nüüdisaegse õpikäsituse järgi on iga õppija kaasamine planeerimisse ja hindamisse ning üldpädevuste arendamine tähtis [29]. Õppijakeskne lähenemine, mis on tihedalt seotud õppetöö mängustamise, koostöö, projektipõhise õppe, tagasisidestamise ja õppimisele hinnangu andmisega [32], toetab nüüdisaegse õpikäsituse põhitõdesid. Erinevate meetodite kasutamine on oluline, et parandada õpilaste õppetulemusi, kaasata neid pidevalt õppeprotsessi ning tõsta õpilaste arvamust ja suhtumist programmeerimisse [33].

### **1.1.1 Paarisprogrammeerimine**

Paarisprogrammeerimine on koostöine lähenemine programmeerimise õppimisel, mille käigus kaks inimest töötavad ühe arvutiga ühise ülesande lahendamise eesmärgil, kus üks paariline

kirjutab koodi ning teine vaatleb, juhendab ja jälgib tööprotsessi [34]. Paarisprogrammeerimise kasumlikkust on uurinud mitmed teadlased aastakümnete jooksul ning on selgunud, et paarisprogrammeerimine arendab koostööd ning laiendab teadmisi programmeerimise vallas ning suurendab kaasatust [35–39]. Samuti edendab paarisprogrammeerimine koostööd ja kommunikatsiooni [40]. Meetodit on pigem uuritud üliõpilaste ja täiskasvanute peal, kuid Weijit näitasid, et paarisprogrammeerimise kasu laieneb ka põhikooli õpilastele, arendades muuhulgas nende raalmõtlemist ja enesetõhusust [41].

Õpetajana on võimalik paarisprogrammeerimise jaoks ise paarid moodustada. Paarides, kus ühel paarilisel on rohkem teadmisi programmeerimise valdkonnas kui teisel, on programmeerimine kasulik mõlemale paarilisele, kuid suurem tõus on nõrgema õpilase õppetulemustes, kui panna ta paari tugevama programmeerijaga [37]. Võrdväärsete partnerite korral ei toimu küll nii selget paarilise pealt õppimist, küll aga avaldub koostöine probleemilahendamine, mille käigus mõlemad partnerid panustavad ning vastutavad võrdselt [38]. Põhikooli riikliku õppekava [17] alusel peab kool kujundama õpilaste pädevusi nii, et õpilane oskab teha koostööd teiste inimestega erinevates situatsioonides ning aktsepteerida ja arvestada inimeste erinevusi teistega suhtlemisel. Seetõttu on ka paarisprogrammeerimise puhul oluline teha koostööd erinevate inimestega.

Paarisprogrammeerimise kasutamine programmeerimise õpetamisel loob eeliseid ka õpetajale. Koos töötades lahendavad õpilased esinevad probleemid üldiselt iseseisvalt ja ei vaja õpetaja abi ning samuti on töid, mida on vaja üle vaadata ja hinnata poole vähem [42].

Kuigi paarisprogrammeerimine on üldjoontes tõhus koostöö ja kommunikatsiooni arendamiseks, on meetodil ka puuduseid. Chong ja Hurlbutt leidsid, et kaasatus oli püsivam programmeerijal, kes kontrollis klaviatuuri ja hiirt, kuid mõlema paarilise kaasatuse parandamiseks võib kasutada kahte klaviatuuri [39]. Samuti tuleb meeles pidada, et umbes 5% õpilastest, kes sageli on ka kõige paremad programmeerijad, ei saa paarisprogrammeerimisest kasu ning eelistavad töötada üksi, et kaaslane ei pidurdaks tempot [42]. Seetõttu on oluline jätta tundidesse ka aega iseseisvaks ülesannete lahendamiseks.

### **1.1.2 Vastastikune õppimine**

Vastastikune õpe on õppijakeskne õppemeetod, mis soodustab õpilaste kaasatust [43]. Programmeerimise õppes kasutatakse sageli klikkereid, et õppijad saaksid vastata valikvastustega küsimustele, läbi mille on nad aktiivsed osalejad õppeprotsessis [44]. Tavaliselt vastavad õppijad esmalt küsimusele individuaalselt, pärast mida arutavad nad

küsimuse üle väikses rühmas, peavad konsensuseni ja vastama grupina samale küsimusele uuesti [45]. Klikkeritega vastamine tagab vastaja anonüümsuse ning on teada, et peaaegu kolmveerand vastajatest leiab, et anonüümsus paneb nad end tundma enesekindlamalt ja vastama seeläbi ausamalt [46].

Lisaks klikkeritele on võimalik õpilasi kaasata kasutades erinevaid mobiilirakendusi. Samuti nagu vastamisel, aitab kaasa ka see, kui õpilased saavad küsimusi esitada anonüümselt ning teineteisele anonüümselt vastata [47]. Anonüümsust hindavad kõrgelt nii tudengid kui ka põhikooli ja keskkooli õpilased. Erinevatest tehnoloogilistest vahenditest peetakse lihtsaks ja mugavaks näiteks rakendust Plickers [48], mille kasutamine sarnaneb klikkeritele. Vastastikust õpet saab läbi viia ka ilma tehnoloogiliste abivahenditeta, kasutades paberit ja kirjutusvahendit, kuid ajakulu on suurem.

### **1.1.3 Überpööratud klassiruum**

Überpööratud klassiruum on õppijakeskne õppemeetod, mis koosneb kõige levinuma definitsiooni järgi kahest osast: kodus uue teema õppimisest ja tunnis teema kinnistamisest ehk ülesannete lahendamisest [49]. Meetod toetab probleemipõhist õpet, kus ühte õiget vastust sageli pole, vaid loeb lahenduskäik [50], nagu programmeerimisel, kus tulemus võib olla sama aga kirjutatud kood erinev. Õppimine überpööratud klassiruumi rakendades on tõhus, kui meetod on läbimõeldult planeeritud ja ellu viidud ning samuti suurendab see õppijate kaasatust klassiruumis [51]. Überpööratud klassiruumi kasutamine tõstis õppijate suhtumist programmeerimisse, suurendas õpetaja-õppija vahelist kommunikatsiooni ning parandas õppetulemusi [33] ja enesetõhusust programmeerimise suhtes [52].

Iseseisvaks õppimiseks kodus nimetatakse üheks eelistatuimaks vahendiks videoid, kuid kaasata võib ka erinevaid veebilehti, faile ja dokumente [53]. Õpilased ise peavad oluliseks võimalust õppematerjalide poole tagasi pöörduda, kui selleks vajadus tekib [50]. Probleemiks võib kujuneda videote kvaliteet ning mittemotiveeritud õpilaste, kellel puudub huvi programmeerimise vastu, kaasamine [54]. Samuti peavad õpetajad videote loomist ajamahukaks, kuid leiavad, et überpööratud klassiruumi kasutamisel jääb klassiruumis veedetud aeg õppijate individuaalsele juhendamisele ning ei kulu uue teema selgitamiseks [55].

Erinevate allikate põhjal saab järeldada, et õppimiseks kõige tõhusamad on kuni 10-minutilised videod [56,57]. Alla 5-minutiliste videote tõhusus on suurem kui üle 10-minutilistel videotel, sest õpilaste tähelepanu püsib videol ja kaasahaaratus on seega suurem [57]. Optimaalsed on

ka 6–10 minuti pikkused videod, mis on piisavalt lühikesed tähelepanu hoidmiseks, kuid piisavalt pikad, et edastada vajalik informatsioon [56].

Ümberpööratud klassiruumi meetod kogub hariduses populaarsust. Meetod on tõhusam kui paljud teised õppemeetodid, kui rakendada seda vanemate õppijate õpetamisel põhikooli lõpuklassidest kõrgkoolini [58]. Kuigi videod on levinud ja populaarne viis programmeerimise iseseisvaks õppimiseks, siis tuleks võimalusel kaasata ka interaktiivseid vahendeid [59].

#### **1.1.4 Mängupõhine õpe**

Mängupõhine õpe on programmeerimise õppimisel mõistetak kahte moodi: läbi hariduslike mängude õppimine ja ise mängu luues õppimine. Mängupõhine õpe on meetod uute kontseptsioonide ja oskuste õppimiseks läbi mängude [60]. Didchev ja Didcheva leidsid, et mängupõhine õpe, mis on sageli ka üks koostõise õppe vorme, on kõige efektiivsem meetod õpilastevahelise koostöö soodustamiseks [61].

Programmeerimise õppe soodustamiseks on loodud mitmeid digitaalseid mänge, nagu on loonud Scratch ja Code.org [4], kuid loomingulisust nõudvate lauamängude mängimine on positiivses korrelatsioonis mängija loovuse ja originaalsusega [62]. Lisaks sellele, et lauamängude abiga saavad mängijad uusi teadmisi, arendab lauamängude mängimine ka õpilaste suhtlemisoskust [63]. Kuna programmeerimine nõuab sageli uutele probleemidele loovat lähenemist ning koostööd teiste inimestega, tasub tundidesse kaasata ka mittedigitaalseid mänge, mille mängimisel on olulisel kohal teistega suhtlemine.

Programmeerimise õppimiseks kasutatakse sageli ka ise mängude loomist. Seralidou ja Douligeris leidsid, et mängude loomine plokkide-põhises programmeerimiskeskonnas laiendab õppijate programmeerimise teadmisi ja oskusi [64], kuid üleminek plokkide-põhiselt keskkonnalt tekstipõhisele, nagu Python, osutub sageli keeruliseks [65]. Ühtset arusaama, kas mängude loomine Pythonis on algajale jõukohane, ei ole. Samuti on suurem osa uuringuid viidud läbi üliõpilaste peal. Simian ja Vulpeanu leidsid, et Python on sobilik keel algajale mänguloomise põhimõtetega tutvumiseks [66], millele võivad kaasa aidata ka veebis leiduvad materjalid mänguloomise kohta. Pythoni teemadest kõige keerulisemaks on osutunud algajatele objektorienteeritud programmeerimine ja algoritmid [67], mis on aga mängu loomise põhilised komponendid [66]. Seetõttu soovivad Szydłowska jt jätta need kaks teemat algajatele suunatud õppeprotsessist kõrvale [67].

Robotid mängustavad programmeerimise õpet. Uuringud näitavad, et robotite kasutamine programmeerimise õppes tõstab õpilaste rahulolu, huvi ja enesetõhusust programmeerimises

ning suurendab võimalust, et õppija teeb karjäärivaliku informaatika kasuks [68]. Robotikat ja programmeerimist on võimalik lõimida projektipõhises õppes, mis tõstab motivatsiooni ning arendab 21. sajandi oskusi [69]. Robotite kaasamise muudab problemaatiliseks vahendite puudus ja robotite maksumus, mistõttu kõik koolid pole suutelised neid endale lubama. Samuti tuleb arvestada, et robotite kaasamine on kõige tõhusam lasteaia õppes ning kõige vähem kasu saavad gümnaasiumiõpilased, kusjuures positiivne mõju tüdrukute õpitulemustele on suurem kui poistele [70].

## **1.2 Programmeerimise õppimine ja AI**

Tänapäeval mängib üha suuremat rolli nii õppimisel kui ka õpetamisel tehisintellekt (ingl *artificial intelligence* ehk AI). Õpetajatest 45% peavad oma teadmisi tehisintellekti kohapealt piiratuks ning 35% tavapäraseks [71], mistõttu tehisintellekti kaasamine haridusse vajaks õpetajate koolitamist tehisintellekti kasutamise, võimaluste ja piirangute seisukohast.

Yilmaz ja Yilmaz leidsid, et keelemudelite kasutamine edendab õpilaste enesetõhusust ja motivatsiooni, on efektiivne tööriist raalmõtlemise arendamiseks ning tehisintellekti tuleks rakendada programmeerimise õpetamisel [72]. Üks AI kasutamise võimalusi on selle kasutamine paarisprogrammeerimisel, kuid hetkel on tehtud veel vähe uuringuid, et teha konkreetseid otsuseid, kumb on tõhusam – AI ja inimese paarisprogrammeerimine või kahe inimese vaheline paarisprogrammeerimine [73].

Seni on uuringud keskendunud rohkem elukutseliste programmeerijate AI kasutusele, millel on nii eeliseid kui ka puuduseid, kuid suuremad väljakutsed esinevad AI kasutamisel neil inimestel, kes programmeerimist alles õpivad [74]. On oht, et õpilased hakkavad liigselt toetuma tehisintellektile ning väärtarvitavad keelemudeleid [75]. Samuti on väljatoodud potentsiaalsed murekohad, millena nähakse õppijate-vahelise suhtlemise vähenemist klassiruumis, AI täpsust ja korrektsust ning sotsiaalsete oskuste mandumist [76].

Tehisintellekti tõhusaks kaasamiseks õppeprotsessis, on vaja õpilastele õpetada, kuidas oma nägemus ja mõtted kirja panna, et AI suudaks genereerida kõige puhtama ja parema koodi, mis kataks kasutaja vajadused [72]. Õpetajad saavad samuti tehisintellekti õppetööd planeerides kasutada, kuna see on tõhus vahend ülesannete diferentseerimiseks [75].

## **1.3 Motivatsioon programmeerimise õppimiseks**

Paar aastat tagasi oli IT-sektoris suur tööjõu puudus, mis on aga tänaseks tasakaalustunud [77]. Tasakaalu säilitamiseks on endiselt vaja koolitada tarkvara arendajaid ja programmeerijaid,

kuna töökohti luuakse valdkonnas aina juurde ning üha automatiseerivas ühiskonnas on informaatikutel suur roll.

Tuleb arvestada, et kõik noored ei soovi oma tulevikku programmeerimisega siduda, kuid programmeerimise käigus omandatavad oskused on olulised nii tööturul kui ka edasises elus [31]. Seetõttu tuleks selgitustööd teha läbi üldpädevuste omandamise, et motiveerida õpilasi programmeerimist õppima.

#### **1.4 Õppeühiku mudel ja õppetöö planeerimine**

Õppeühiku planeerimiseks on välja töötatud erinevaid mudeleid, mis on tõhusad uue materjali õppimiseks ja kinnistamiseks. Paraku puudub ülevaade, millised õppeühiku mudelid sobivad enim programmeerimise tunni planeerimiseks ja läbiviimiseks. Õppetöö planeerimine aitab tõsta õpetamise kvaliteeti ning lihtsustab õpitulemuste saavutamist [78].

Käesoleva töö raames kasutatakse tunnikavade loomisel Gagné õppeühiku mudelit, mis koosneb üheksast etapist: tähelepanu haaramine, õppija õpieesmärkidest teavitamine, varem õpitu meeldetuletamine, uue materjali esitamine, õppimise suunamine, õpitu kontrollimine, tagasisidestamine, õppimisele hinnangu andmine, õpitu kinnistamine ja üldistamine [79].

## 2. Õpetajamaterjalide loomine

Vajadus programmeerimise õpetajamaterjalide järele on aktuaalne. Programmeerimise edukaks õpetamiseks uuriti eelnevalt erinevaid meetodeid ja õppetühiku etappe, millele õpetajamaterjalide loomisel toetuti.

### 2.1 Olemasolevad õpetajamaterjalid programmeerimise õpetamiseks

Tartu Ülikooli arvutiteaduste instituudi töörühm on koostanud programmeerimise õpiku gümnaasiumile, mille juurde kuulub ka õpetajamaterjal [10]. Õpik on ülesehitatud toetudes programmeerimiskeelele Python. Ülemineku sujuvamaks muutmiseks ja materjalide sidumiseks on ka käesoleva töö raames loodud õpetajamaterjal mõeldud programmeerimiskeele Pythoni õpetamiseks.

Tunnikavasid leiab ka Tartu Ülikooli arvutiteaduste instituudi kodulehelt teemaplokis „Programmeerimine lastele ja noortele“ [80], kuid loodud tunnikavad on mõeldud erinevate programmidega tutvumiseks ning ei moodusta ühtset tervikut programmeerimiskeele Python õppimiseks.

### 2.2 ADDIE mudel

Üheks levinuimaks mudeliks õppematerjalide väljatöötamisel on ADDIE mudel, mis võtab arvesse erinevaid aspekte ja vaatenurki ning aitab seetõttu kaasa mitmekülgse tõhusa õppematerjali loomisele [81]. ADDIE mudeli etappideks on analüüsimine, kavandamine, arendamine, kasutamine ja hinnangu andmine, millest viimase abil on võimalik luua usaldusväärseid materjale, mis katavad kasutaja vajadused [82].

Õpetajamaterjal loodi põhikooli õpikule, mis on mõeldud eelkõige iseseisvaks õppimiseks, kuid milles sisalduvad enesekontrolli küsimusi ja ülesandeid saab rakendada ka koolitunnis. Gümnaasiumile on loodud programmeerimisõpik koos õpetajamaterjalidega [25], mis on mõeldud programmeerimiskeele Python õppimiseks ja õpetamiseks. Materjalide sidumiseks sooviti luua tugimaterjale just Pythoni õpetamiseks, et pedagoogidel oleks rohkem valikuid ja võimalusi, mis lihtsustaks tööplaneerimist ja toetaks materjalide kohandamist.

Käesoleva töö eesmärgiks oli luua õpetajamaterjalid, mis sisaldaks erinevaid meetodeid ja ülesandeid. Selleks, et materjalid oleksid kasutatavad suuremas osas koolides, loobuti robotite kaasamisest, kuna kõikides koolides roboteid ei ole.

## **2.3 Valim**

Loodud õpetajamaterjali hindamiseks ja kasutamiseks kaasati arendusuuringusse informaatikaõpetajad. Sihipärase valimi kasutamise eesmärgiks oli koguda sügavat ja laiapinnalist tagasisidet oma ala spetsialistidelt, et saada võimalikult asjakohast ja kasulikku tagasisidet [83].

Väljatöötatud materjali eesmärgist ehk alustavate õpetajate toetamisest lähtuvalt kaasati valimisse ka õpetajad, kes ei ole varem programmeerimist õpetanud, et saada ka nende hinnang loodud materjalide kvaliteedile ja eesmärgipärasusele. Vastajaid oli sõltuvalt teemast 2–5, kelle seas oli nii õpetajaid, üliõpilasi kui ka õpetajana töötavaid üliõpilasi. Kokku anti tagasiside 48 korral. Samuti leidis valimis programmeerimise õpetamise kogemusega eksperte, kui ka neid, kes pole varem programmeerimist õpetanud.

## **2.4 Töö protseduur**

### **2.4.1 Analüüsimine**

Materjali väljatöötamisel kasutati ADDIE mudelit, mille esimeseks etapiks on analüüsimine. Selles etapis mõtestatakse, mida õpilased õppima hakkavad, sõnastatakse probleem ja põhjendatakse vajadus materjalide järele ning pannakse paika üldised õpitulemused [81]. Uurimisprobleem ja eestikeelsete õpetajamaterjalide vajadus on sõnastatud peatükis 1.

Informaatika on valikõppeaine, mille õppe sisu koosneb üldistatult raalmõtlemisest, mis aitab õpilasel lahendada elulisi ülesandeid kasutades algoritme, tuvastades mustreid ning jaotades probleemi osadeks, ning disainmõtlemisest, mis on kasutajakeskne, loov ja koostöine eluliste ülesannete lahendamise viis [2]. Programmeerimine on üks võimalik teema informaatikaõppes, mille õppimise tulemusel õpilane mõistab ja kasutab teadlikult põhilisi mõisteid; analüüsib etteantud programmi ja ennustab selle töö tulemust; koostab programmi etteantud kirjelduse alusel; kirjeldab algoritmide ning programmide kasutamise lisandväärtust erinevates eluvaldkondades; koostab lihtsamaid avaldise ja algoritme, mida on võimalik kasutada reaalses juhtprogrammis; selgitab rakenduse töö testimise vajadust ja olemust ning parandab tekkinud vead; koostab lihtsama ülesande täitmiseks valmisdetailidest mehaanilise seadme ja selle juhtprogrammi [2]. Taotlevatest õpitulemustest toetab materjal kuue esimese õpitulemuse saavutamist ning neist seitsmes on käesoleva materjali raames soovituslik, kuna roboteid kõikides koolides ei ole.

Õpetajamaterjalide loomisel toetuti veebipõhise programmeerimisõpiku <https://progema.ee/> õppeteemadele ning olemasolevatele materjalidele.

#### **2.4.2 Kavandamine**

Teise etapina kavandati õpetajamaterjali ülesehitus. Kavandamise käigus mõtestati lahti järgmised aspektid: mõõdetavad ja kontrollitavad õpitulemused, sobivad õppemeetodid õpitulemuste saavutamiseks ning lisamaterjalid ja keskkonnad, mis toetavad õppeprotsessi [81].

Informaatikat on soovitatav õpetada igas kooliastmes vähemalt 35 tundi [2], mistõttu plaaniti tunnikavad koostada 35 tunni tarbeks. Samuti on olulisel kohal informaatikas rühmatööna projekti loomine, mis eeldab tegevuste planeerimist, panustamist meeskonnaliikmena ning lõpetuseks projekti aruande esitamist või kaitsmist. Sellest tulenevalt sooviti kursuse keskel tutvustada õpilastele juba projekti ning anda neile võimalus varakult rühmatööga alustada.

Õpitulemuste sõnastamisel toetuti põhikooli riiklikus õppekavas välja toodud valikõppeaine informaatika õpitulemustele teemas „Programmeerimine“ [2]. Samuti toetasid õpitulemuste seadmist programmeerimisõpikus käsitletud teemad.

Teemad, mille õpetamist õpetajamaterjal katab, on järgmised: sissejuhatus, algoritmid, muutujad ja avaldised, tingimuslause, korduslause, funktsioonid, failid, veahaldus, järjendid, sõnastikud, hulgad, ennikud, mitmemõõtmelised struktuurid, projekt ja tehisintellekti kasutamine programmeerimises. Iga teema läbimiseks (v.a projekt) kavandati 1–3 õppetundi, et anda õpilastele võimalus nii materjaliga tutvumiseks kui ka ülesannete lahendamiseks, mis toetaks seeläbi teadmiste ja oskuste kinnistamist (vt tabel 1).

Tabel 1. Tundide jaotus

Teema	Tunnid	Teema	Tunnid
Sissejuhatus	1.	Projekti tutvustus ja AI	15.
Algoritm	2.	Järjendid	16.–17.
Muutujad ja avaldised	3.–4.	Sõnastikud	18.–19.
Tingimuslauseid	5.–6.	Hulgad	20.–21.
Korduslauseid	7.–8.	Mitmemõõtmelised	22.
Funktsioonid	9.–10.	Projekti loomine	23.–30.
Failist lugemine	11.	Projekti esitluse loomine	31.
Faili kirjutamine	12.	Projektide kaitsmised	32.–33.
Failid	13.	Mängud	34.–35.
Veahaldus	14.		

Tunnikavade loomisel tugineti Gagné õppeühikumudelil, mille peamised etapid on järgmised: tähelepanu haaramine, õppija õpiesmärkidest teavitamine, varem õpitu meeldetuletamine, uue materjali esitamine, õppimise suunamine, õpitu kontrollimine, tagasisidestamine, õppimisele hinnangu andmine, õpitu kinnistamine ja üldistamine [79]. Iga tund algab meeldetuletusega möödunud tunnist ning võimalusel kasutatakse tähelepanu haaramiseks mängu, mis aitavad ka õpitud meelde tuletada. Tunni algusfaasis tutvustatakse õpilastele tunni eesmärgi ning üheskoos tutvutakse uue osaga, millele üldiselt järgneb ülesannete lahendamine. Õpitu kinnistamiseks ja kontrollimiseks kaasati tundidesse samuti erinevaid mängu ja harjutusi, mis annaks võimaluse õpilastel ka oma õppimist hinnata. Tagasisidestamine toimub tundides jooksvalt ning sõltuvalt kooli eripärast on õpetajal võimalik valida, kuidas ja mil määral ta tunnis tehtavaid ülesandeid hindab. Õpitud kinnistatakse jooksvalt pea igas tunnis, sest ülesanded lähevad järk-järgult keerulisemaks ning nõuavad üha enam eelnevalt õpitud teadmiste ja oskuste rakendamist. Gagné õppeühikumudeli kohaselt peab kõik etapid läbima ühe õppeühiku jooksul, mille antud töö kontekstis määrab ühe teema õppimiseks kuluv aeg.

Õppemeetoditest kasutati tundide planeerimisel näiteks paarisprogrammeerimist, vastastikust õpet, ümberpööratud klassiruumi, mängupõhist õpet ja iseseisvat ülesannete lahendamist. Detailsem ülevaade õppemeetoditest on välja toodud peatükis 1.

Meetodite rakendamist toetavad erinevad keskkonnad, nagu Plickers, Mentimeter, Blooket, Kahoot, JeopardyLabs, mis on veebipõhised ja annavad õpilastele soovi korral kohest tagasisidet. Videote loomisel kasutati Thonny, Microsoft PowerPointi ja ClipCharmi ning lõpp-produkt lisati Youtube'i. Lisaülesannete ja plakatite loomisel olid abiks Canva, Google'i esitlused ja dokumendid ning Microsoft Word.

### 2.4.3 Materjali väljatöötamine

Õpetajamaterjalide väljatöötamist alustati tunnikava põhja loomisest. Tunnikavad otsustati kirja panna tabelina. Tund jagati kolmeks osaks: ettevalmistus, põhiosa ja lõpetav osa. Kavas kirjeldati tunni tegevusi, hinnati tegevusele kuluvat aega ning lisati vajaminevad materjalid (vt joonis 1).

4. TUND (MUUTUJAD JA AVALDISED)			
Tunni osad	Tegevusele kuluv aeg	Tegevuse kirjeldus	Materjalid
I Ettevalmistus	5 min	Üheskoos tuletatakse meelde põhilised andmetüübid ja käsk <i>input</i> . Tunnis lahendatakse erinevaid ülesandeid andmetüüpide ja muutujatega.  Eelmises tunnis tehti lihtsustatud programm tervitamise kohta, nüüd arendatakse seda programmi edasi.	
II Põhiosa	30 min	Harjutuste "Tervitus" ja "Võileivatorit" lahendamine iseseisvalt õpikust. Kiirematele lisatülesanded.	<a href="#">Muutujad ja avaldised</a> <a href="#">Lisatülesanded</a>
III Lõpetav osa	10 min	Õpetaja jagab õpilastele Quizzize linki ja palub vastata ilma materjalideta. Nii õpilased kui ka õpetaja saavad ülevaate, mis vajab veel harjutamist ning mis on selge.  Quizzize kasutamiseks peab õpetajal olema konto.	<a href="#">Quizziz</a>

Joonis 1. Tunnikava

Materjalide loomisel arvestati, et teemade juurde kuuluvad tunnikavad kataksid kõik Gagné mudeli õppeühiku etapid. Materjali ei lisatud ülesannete lahendusi ega õigeid vastuseid, kuna materjalid on kättesaadavad ka õpilastele.

Iga teema juurde prooviti luua ka mõni mäng, mis aitaks teema põhitõdesid kinnistada ja kontrollida. Mängude mängimisel ei kasutata Pythonit, vaid õpilased peavad koodi vaadates aru saama, mis on programmi väljundiks. Mängude abil on võimalik toetada õpitulemuse „õpilane analüüsib etteantud programmi ja ennustab selle töö tulemust“ saavutamist. Õpetajamaterjali on lisatud nii veebipõhiseid kui ka printitavaid mängu, mida on võimalik kasutada ka aasta lõpus kordavates tundides.

Enne ülesannete loomist vaadati, kui palju harjutusülesandeid on programmeerimisõpikus antud ning hinnati, kui palju lisaülesandeid on vaja luua. Ülesannete koostamisel prooviti

arvestada põhikooli õpilaste võimalike huvidega, et tõsta õpilaste kaasatust. Diferentseerimiseks loodi ülesandeid rohkem ning lisati mitmele ülesandele ka edasiarenduse võimalusi, et pakkuda aeglasematele lahendajatele jõukohast pingutust ja edasijõudnutele väljakutseid (vt joonis 2).

### **Klassivanem**

Koosta programm, mis küsib kasutajalt hääletajate arvu ning laseb sisestada nii palju kordi klassivanema kandidaate. Programm jätab meelde, kui palju kellegi poolt hääletatud on ning väljastab pärast hääletuse lõppu tulemused.

*Edasijõudnutele.*

- Programm väljastab hääletuse tulemuste põhjal klassivanema nime. Kui tulemus on viigiline, annab programm õpetajale ette enim hääli saanud õpilaste nimed ning palub anda oma hääl.

Joonis 2. Ülesanne koos edasiarendusega

Tunnikavade juures anti metoodilisi soovitusi tunni läbiviimiseks, näiteks paarisprogrammeerimise rakendamiseks ja eelmises tunnis puudujate kaasamiseks. Samuti toodi näitlikustamiseks välja erinevaid võimalusi, et õpetaja saaks sobiva näite valida lähtuvalt õpilaste eelteadmistest ja tasemest.

Sõnastike teema juurde loodi materjal, mis toetab ümberpööratud klassiruumi rakendamist. Esmalt loodi teema juurde lühikesed eestikeelsed õppevideod, et uue teemaga tutvumisel oleks õpilastel lisaks õpikule võimalus kasutada ka alternatiivset varianti. Videote loomisel võeti arvesse nende pikkust, et vältida õpilaste tähelepanu hajumist [56,57].

Loodud õpetajamaterjalid on leitavad programmeerimisõpiku juures. Materjalide leidmise lihtsustamiseks lisati õpiku menüüribale jaotis „Õpetajale“, millele klõpsates avaneb lühike tutvustus ja kus asub link õpetajamaterjalidele.

#### **2.4.4 Tagasisideküsimustiku koostamine**

Tagasisideküsimustiku koostamisel toetuti LORI mudelile, mis koosneb üheksast komponendist, mida hinnatakse 5-palli skaalal: sisu kvaliteet, vastavus õpieesmärkidega, tagasiside ja sisu kohandamine, motivatsioon, kujundus, kasutatavus, ligipääsetavus, taaskasutatavus, standarditele vastavus [84].

Küsimustik koosnes kolmest osast, millest esimeses koguti taustainfot programmeerimise õpetamise ning vastaja eriala kohta. Teises osas esitati lähtuvalt LORI mudelist kaheksa väidet, mida kasutaja hindas Likerti skaalal (1 – ei nõustu üldse, 2 – ei nõustu, 3 – ei oska öelda, 4 – nõustun, 5 – nõustun täielikult) [85]. Iga väite juures oli vastajal võimalik oma vastust täiendada. Kolmandas osas küsiti avatud küsimusi, et koguda detailsemat informatsiooni soovitude, täienduste ja puudujääkide kohta.

Tagasisideküsimustik loodi iga teema kohta, et kasutaja saaks läbi töötada ühe teemaploki ning kohe küsimustikule vastata. Küsimustikke oli kokku 15. Eraldi küsimustikud andsid ekspertidele võimaluse töötada materjal läbi soovitud järjekorras ning jätta ajapuudusel ka vastamine pooleli.

Küsimustiku alguses kirjeldati, miks tagasisidet kogutakse ning kuidas kasutatakse. Kõik hindajad said jääda anonüümseks. Tagasisidevormi näidis on välja toodud lisas 1.

#### **2.4.5 Tagasiside kogumine**

Tagasiside kogumiseks kasutati keskkonda Google Forms ning jagati materjale informaatikaõpetajate meililistiga. Materjalide kasutamiseks ja hindamiseks oli soovijatel aega kolm nädalat. Esimest korda saadeti informatsioon õpetajatele 8. aprillil ning tagasiside andmise viimane päev oli 27. aprill. Igale küsimustikule vastas 2–5 eksperti.

#### **2.4.6 Täienduste ja paranduste tegemine**

Viimase etapina kohandati materjale lähtuvalt ekspertide tagasisidest. Täienduste ja muudatuste kohta koguti informatsiooni vastajate vabavastustest. Suurem osa soovitudest aktsepteeriti, kuid esines üks erand, mida ressursi puudumise ja vastajate erimeelsuse tõttu ellu ei viidud. Ekspertide soovitudega saab tutvuda lisas 2.

### **3. Tulemused ja arutelu**

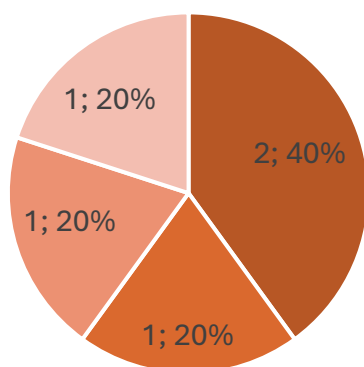
#### **3.1 Andmetöötluse meetodid**

Saadud tagasisidet analüüsiti Microsoft Excelis kasutades nii kvalitatiivse kui ka kvantitatiivse andmeanalüüsi meetodeid. Likerti skaalal hinnatud väiteid analüüsiti kirjeldava statistika abil ning välja toodi hinnangute mediaan (Me). Avatud küsimuste vastuseid ja väidete kommentaaride analüüsimiseks kodeeriti lõigud ning kategoriseeriti koodid, mille abil vabavastuseid analüüsiti. Analüüsi tulemusel leiti vastus küsimusele, kuidas hindavad õpetajad loodud materjale. Samuti tehti kogutud tagasiside põhjal täiendusi ja muudatusi õpetajamaterjalides.

#### **3.2 Teemade kohta kogutud tagasiside**

##### **3.2.1 Sissejuhatus**

Sissejuhatuses kohti andis tagasisidet viis eksperti, kellest kaks olid õpetajad, kaks üliõpilased ning üks õpetajana töötav üliõpilane. Vastanute seas leidis erineva programmeerimise õpetamise taustaga inimesi, kellest kolmel oli eelnev kokkupuude programmeerimise õpetamisega või soov programmeerimist õpetada (vt joonis 3).



- Ma pole kunagi programmeerimist õpetanud ega soovi seda teha ka tulevikus
- Soovin tulevikus programmeerimist õpetada
- Õpetan programmeerimist
- Olen varem programmeerimist õpetanud

Joonis 3. Sissejuhatusetele tagasisidet andnud ekspertide programmeerimise õpetamise kogemus LORI-mudelile toetuvaid väiteid hinnati küllaltki sarnaselt (vt tabel 2).

Tabel 2. Sissejuhatusete väidete hinnangud

Väide	1	2	3	4	5	Me
Õpetajamaterjal on mitmekesine	0	0	0	1	4	5
Materjal toetab õpieesmärkide saavutamist	0	0	0	2	3	5
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	1	4	5
Materjal on arusaadavalt struktureeritud	0	0	0	1	4	5
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	4	5
Materjal on lihtsasti kättesaadav	0	0	0	1	4	5
Materjali saab korduvalt kasutada	0	0	0	0	5	5
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	5	5

Sissejuhatava tunni materjale hindasid eksperdid kõrgelt. Neli vastajat tõi välja, et kõige väärtuslikum on inimbingo mäng, mille abil õpilaste programmeerimise oskust ja kogemust hinnata, kuid puudust tunti faili redigeerimise võimalusest. Üks ekspert leidis, et materjal on väga sisukas ning seetõttu ei pruugi see mahtuda 45 minuti sisse.

Soovi korral oli vastajatel võimalik oma mõtted küsimustiku lõpus kirja panna. Seda võimalust kasutasid kolm vastajat, kellest kaks jagasid metoodilisi soovitusi: „Idee - kas samasugust telge ei võiks teha ka seal aasta lõpu poole, kui tunnid on kõik läbitud. Õpetaja teeb nt aasta alguses olevast teljest pilti ja siis aasta lõpus saavad õpilased võrrelda mõlemat telge, kas neil on toimunud areng“ ning „Põhiosas võiks küsimuste üle alustuseks arutleda väikestes rühmades ja siis klassiüleselt, et õpilased oleksid rohkem kaasatud. Lõpus võiks olla ka midagi kinnistavat lisaks tagasisidele“. Kolmas vastaja lisas: „Tundub mõnus interaktiivne ja õpilaste jaoks loodetavasti positiivne esmane kokkupuude programmeerimisega!“

### 3.2.2 Algoritm

Algoritmide teemale andis tagasisidet kaks üliõpilast, üks õpetaja ja üks õpetajana töötav üliõpilane. Kahel vastajatest puudus soov tulevikus programmeerimist õpetada, kuid üks ekspert oli varem juba programmeerimist õpetanud ja teine soovis seda teha tulevikus. Kõigi esitatud väidetega nõustuti (vt tabel 3) ning materjali juures ei tuntud millestki puudust.

Tabel 3. Algoritmi väidete hinnangud

Väide	1	2	3	4	5	Me
Õpetajamaterjal on mitmekesine	0	0	0	1	3	5
Materjal toetab õpieesmärkide saavutamist	0	0	0	2	2	4,5
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	1	3	5
Materjal on arusaadavalt struktureeritud	0	0	0	0	4	5
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	2	2	4,5
Materjal on lihtsasti kättesaadav	0	0	0	0	4	5
Materjali saab korduvalt kasutada	0	0	0	0	4	5
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	4	5

Sarnaselt sissejuhatavale teemale selgus, et materjal on mahukas ning selle läbitöötamise kiirus sõltub õpilaste motiveeritusest ja eelnevatest oskustest. Väärtuslike materjalidena toodi välja joonistamise töölehed ja video, mille puhul nimetati kaht kasutamist mõjutavat tegurit – video pikkus ja keel.

Optimaalset video pikkust on uurinud mitmed autorid ning on leitud, et õppeprotsessis on mõistlik kasutada alla 10-minutilisi videoid [56,57]. Üks vastaja leidis, et tunnikavasse lisatud video on pisut pikk, kuid annab hea arusaamise, mida arvuti tegelikult teeb. Teine vastaja lisas, et õpetaja saab reguleerida, kui pikalt videot näitab ning teha vajadusel pause, et nähtu üle koos õpilastega arutada ja vajadusel tõlkida. Keeleoskus on oluline uue materjali omandamisel ning puudulik keeleoskus piirab õppeprotsessi [23], mistõttu vastaja mure võõrkeelse video pärast on aktuaalne. Enne video kasutamist peab õpetaja veenduma, milline on õppijate keeleoskus ning vajadusel olema valmis tõlkima.

Tunni lõpuosa kohta toodi välja järgmist: „[---] Mulle väga meeldib ka plokkskeem tunni lõpu kohta!“ ning „Lõpus võiksid õpilased olla rohkem kaasatud. Plokkskeemi kuvamise asemel võiksid nad selle näiteks mõnes veebikeskkonnas ise kokku panna või on plokkskeem välja prinditud/lõigatud ja nad panevad selle paari peale kokku.“

### **3.2.3 Muutujad ja avaldised**

Muutujate ja avaldise teemale andsid tagasisidet üks õpetaja, üks üliõpilane ja üks õpetajana töötav üliõpilane. Programmeerimise õpetamise soovi ega kogemust ei olnud ühel vastajal. Kõige kasulikumate materjalidena toodi välja lisatöölehed, Quizziz ning seosed matemaatikaga. Soovitusena lisati, et kõikide kasutajakontot vajavate platvormide juures võiks konto vajadus olla kirjas.

Ekspertide hinnangul oli materjal taas sisutihe, kuid selle kohandamine võimalik (vt tabel 4). Ühe võimalusena toodi välja järgmist: „Mul meeldib 3. tunni alustamine, kus õpilased peavad ise andmetüübid leidma. Ma oma tunnis ilmselt paluksin leida ainult 3 kategooriat ja jätaksin tõeväärtused välja, kuna see on kohati selline kõige raskemini hoomatavam õpilastele. Mulle väga meeldib ka näide muutujate järjekorra vahetamisega. Samuti meeldisid mulle väga 4. tunni lisaülesanded, need tunduvad täpselt paraja raskusastmega. Quizziz on ka väga hea, kindlasti kasutaksin seda!“

Tabel 4. Muutujate ja avaldiste väidete hinnangud

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	0	3	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	1	2	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	<b>5</b>

### 3.2.4 Tingimuslaused

Tingimuslausete teemale andsid tagasisidet õpetajad, kes programmeerimist õpetanud ei ole ega soovi seda teha, üliõpilane, kes on varem programmeerimist õpetanud ja õpetajana töötav üliõpilane, kes soovib tulevikus programmeerimist õpetada. Puudujääke materjalis ei tuvastatud ning kaks eksperti tõid välja, et ülesannete loomine ja seejärel kaaslaste koostatud ülesannete lahendamine võiks õpilaste motivatsiooni tõsta. Samuti tõi üks vastaja välja, et Blooketi kasutamine võiks tuua tundi elevust.

Mängude kasutamine tunnis on efektiivne meetod õpilaste koostöö tagamiseks [61] ning koostöö on üks enim hinnatud oskus tänapäeva tööturul [31]. Seetõttu hindavad õpetajad kõrgelt õppemeetodeid, mis nõuavad koostööd ja pidevat suhtlust.

LORI-mudeli põhjal koostatud väidetega olid vastajad üldiselt nõus, mistõttu võib eeldada, et loodud materjal oli piisavalt läbimõeldud ja antud teema omandamiseks sobilik (vt tabel 5).

Tabel 5. Väidete hinnangud tingimuslausete teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	1	0	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	0	3	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	0	3	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	<b>5</b>

### 3.2.5 Korduslaused

Küsimustikule vastas kolm eksperti, kelle hinnangul olid kõige kasulikumad antud teema õppimiseks mõeldud lisäülesanded ja idee rakendada tunnis paarisprogrammeerimist. Üldise tähelepanekuna toodi välja, et õpieesmärgid võiksid olla kirjas ka iga teema juures eraldi.

Kuigi ülesandeid peeti elulisteks, hinnati skeptiliselt nende sobivust ja huvipakkuvust noorematele õpilastele: „Mulle tundub, et nooremate õpilaste jaoks ei pruugi tõenäosuse ja intresside teemad piisavalt põnevad olla, kuigi tegemist on jällegi väga eluliste ja kasulike ülesannetega, mis aitavad ka matemaatikatunni teadmisi rakendada.“ Samas leiti, et sellest hoolimata on materjale võimalik lähtuvalt vajadustest kohandada ning materjal toetab õpilase motivatsiooni programmeerimise õppimisel (vt tabel 6).

Tabel 6. Väidete hinnangud korduslausete teemale

Väide	1	2	3	4	5	Me
Õpetajamaterjal on mitmekesine	0	0	1	1	1	4
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	5
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	2	1	4
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	5
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	2	5
Materjal on lihtsasti kättesaadav	0	0	0	0	3	5
Materjali saab korduvalt kasutada	0	0	0	0	3	5
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	5

### 3.2.6 Funktsioonid

Funktsioonide teema kohta andsid tagasisidet kaks õpetajat, kellest üks õpetab programmeerimist ja teine ei ole õpetanud ega soovi seda teha ka tulevikus, ning üliõpilane, kes on varem programmeerimist õpetanud. Ekspertide hinnangud väidetele olid sarnased (vt tabel 7), kuid vabavastustes esines lahkhelisid. Üks vastaja kirjutas: „Palju asjalikke mõtteid, kuidas funktsioonide teemat seletada. Eriti kasulikud on ka lisaülesanded.“ Teine vastaja vastandus ning lisas, et tundis puudust materjalist õpetajale: „Materjali õpetajale, kuidas lihtsamalt seletada õpilastele funktsioonide olemust ja mõtet.“

Programmeerimise õpetamise muudab keeruliseks asjakohaste materjalide nappus [5], mistõttu vastajad tõid välja, et eriti kasulikuks peavad nad lisaülesandeid, mille koostamine on ajamahukas: „Lisaülesanded. Kiirematele õpilastele lisaülesannete andmine tahab alati põhjalikumat ette valmistamist; hea, et need siin olemas on.“ Kasulike materjalide loendisse kuulusid veel õpiku peatükk ja Plickers. Üks ekspert tõi veel välja, et funktsioonide teema on keeruline ja ajamahukas, mistõttu võib selle teema läbimiseks kuluda rohkem kui kaks tundi.

Tabel 7. Väidete hinnangud funktsioonide teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	1	0	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	1	0	2	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	1	2	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	<b>5</b>

### 3.2.7 Failid

Failide teemale andsid tagasisidet üks õpetaja, üks üliõpilane ja üks õpetajana töötav üliõpilane. Programmeerimise õpetamise soovi ega kogemust ei olnud ühel vastajal. Vastajate hinnangul oli materjali piisavalt, kuid iseseisvat tööd palju. Kõige kasulikumaks peeti lisaülesandeid ja detailseid tunnikirjeldusi. Üks vastaja lisis: „Mulle meeldib väga mõtte jagada õpilasi paardesse memoriini põhjal, see paneb neil juba mõtte käima. Eriti meeldib mulle ka salakirja ülesanne, ma arvan, et sellist ülesannet on õpilastel väga põnev lahendada.“ Hariduslike lauamängude mängimine arendab lisaks erialastele teadmistele ka õpilaste suhtlemisoskust [63], mis on tänapäeva tööturul nõutud oskus [31] ning seetõttu hindavad õpetajad loodud lauamänge kõrgelt.

Esitatud väidetele antud hinnangud olid pigem üksmeelsed (vt tabel 8). Väitele „Õpetajamaterjal on mitmekesine“ vastas „Ei oska öelda“ üks õpetaja, kes täiendas oma hinnangut kommentaariga „Materjali on piisavalt ja iseseisvat tööd palju.“ Vastanud ei tundnud puudust ühestki aspektist ning täiendusettepanekuid selle teema kohta ei esinenud.

Tabel 8. Väidete hinnangud failide teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	1	1	1	<b>4</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	2	1	<b>4</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	2	1	<b>4</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	<b>5</b>

### 3.2.8 Veahaldus

Veahalduse teema juures vastanud kolme eksperdi hinnangud olid üldjoontes üksmeelsed (vt tabel 9). Vastaja, kes ei osanud öelda, kas materjali on võimalik kohandada, tõi välja, et veahaldusele keskenduv tund tugineb suuresti õpikule, mistõttu ei ole ta kindel, kui palju annab õpiku peatükki vastavalt vajadusele diferentseerida.

Vastustest selgus, et arusaam teema keerukusest sõltub programmeerimisõpetamise kogemusest. Programmeerimise õpetamise kogemusega õpetaja leidis, et tegu on vahelduseks lihtsa ja arusaadava teemaga, kuid varem programmeerimist õpetanud üliõpilane oponeeris: „Mu kogemus näitab, et veahaldus on üks keerulisemaid teemasid õpilaste jaoks ja võib neid natuke segadusse ajada. Samas ma arvan, et sellest ei saa ka rääkimata jätta, sest see on oluline teema, eriti vigade lugemise osas ja et nad saaks paremini aru erinevatest andmetüüpidest ja kuidas neid kasutada.“

Kõige kasulikumana toodi välja selgitus, miks teema vajalik on ning võimalus tutvuda, millised variandid vigade haldamiseks koodis on. Täiendusena soovitas üks ekspert lisada materjalidesse lause, et paljudel juhtudel saame juba tingimuslausetega kontrollida sisendite sobivust.

Tabel 9. Väidete hinnangud veahalduse teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	2	1	<b>4</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	3	0	<b>4</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	1	0	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	<b>5</b>

### 3.2.9 Projekti tutvustus ja AI

Projekti ja AI tutvustamiseks loodud tunni materjale hindas kolm vastajat. Mediaanhinnangud olid vahemikus 4–5 (vt tabel 10). Toodi välja, et AI kasutamine on aktuaalne ning kogunud õpetajatel on kokkupuude õpilastega, kes kirjutavad koodi tehisintellekt kasutades, ise midagi mõistmata. Teema juures pidasid vastajad kõige kasulikumaks küsimusi, mida õpilastelt küsida, ning tehisintellekt laialdasest kasutamisest tulenevat olulisust. On oluline õpetada õpilasi tehisintellekti kasutama, et tulemused vastaksid kasutaja vajadustele [72], mida tõdesid ka käesoleva teema materjale tagasisidestanud eksperdid.

Täiendustena soovisid vastajad teema juures näha, mis on projekti üldine eesmärk ning milline on optimaalne rühmasuurus. Samuti soovitati tunnikonspektis rõhutada, et projekti idee ei ole hinnata kui ägeda ja suure asja õpilased valmis saavad, vaid hoopis näidata, kuidas omandatud teadmisi osatakse rakendada.

Tabel 10. Väidete hinnangud projekti tutvustuse ja AI teemale

Väide	1	2	3	4	5	Me
Õpetajamaterjal on mitmekesine	0	0	0	2	1	4
Materjal toetab õpieesmärkide saavutamist	0	0	1	1	1	4
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	1	0	2	5
Materjal on arusaadavalt struktureeritud	0	0	1	1	1	4
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	1	0	2	5
Materjal on lihtsasti kättesaadav	0	0	0	1	2	5
Materjali saab korduvalt kasutada	0	0	0	1	2	5
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	1	2	5

### 3.2.10 Järjendid

Järjendite teemale koostatud materjalidele andsid hinnangu kaks õpetajat ja üks üliõpilane, kes on varem programmeerimist õpetanud. Vastajad ei tundnud materjali juures puudust millestki ning kõige kasulikumaks nimetati kaardimängu ja videoid. Ümberpööratud klassiruumi rakendamisel on üheks tõhusaimaks õppevahendiks lühikesed õppevideod [53], mille kvaliteet jätab aga sageli soovida [54]. See võib olla ka põhjus, miks videoid hinnati kõrgelt. Samuti tõstab videote väärtust keel, mida videotest kasutatakse. Emakeelsed videod sobivad paremini ümberpööratud klassiruumi rakendamiseks, sest puudulik keelesokus piirab sageli iseseisvat õppimist [23].

Ekspertid nõustusid kõikide õppematerjali kvaliteeti hindavate väidetega (vt tabel 11). Rahulolu käesoleva teema jaoks loodud materjalidega kirjeldasid ka ekspertide vastused: „Mulle meeldib sissejuhatus esimesse tundi, kus õpilased peavad ise pakkuma milline andmestruktuur võiks olla järjend ja mis võiksid olla selle omadused. Lisaks meeldib mulle väga kaardimängu idee, selle käigus peavad õpilased tutvuma järjendite operatsioonidega, aga samas on lisatud mängulisus ja põnevus. Seda kasutan kindlasti! Lisaülesanded on ka kõik põnevad ja nendes peab rakendama erinevaid järjendite operatsioone“ ja „Taaskord kõik need lisamaterjalid, mis on tunnikava juurde koostatud on super!“

Tabel 11. Väidete hinnangud järjendite teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	1	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	0	3	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	0	3	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	1	2	<b>5</b>

### 3.2.11 Sõnastikud

Sõnastike teemale andis tagasisidet kolm eksperti, kelle hinnangud materjalile ei erinenud sõltuvalt eelnevast programmeerimise õpetamise kogemusest (vt tabel 12). Materjal oli vastajate hinnangul sisukas ning kõik eksperdid nimetasid kõige kasulikuma materjalina lisaülesandeid. Lisaülesannetele lisas hindajate sõnul väärtust diferentseerimise võimalus. Kaks vastajat tõid välja ka mängu Blooket, mida peeti teema kontrollimiseks heaks abivahendiks.

Tabel 12. Väidete hinnangud sõnastike teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	1	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	2	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	1	2	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	1	2	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	3	<b>5</b>

### 3.2.12 Hulgad

Hulkasid sisaldavaid tunnimaterjale hindasid kaks õpetajat ja üks programmeerimise õpetamise kogemusega üliõpilane. Väidete hinnangud jäid vahemikku 4–5, mis tähendab, et vastajad nõustusid või nõustusid täielikult kõikide esitatud väidetega (vt tabel 13).

Ekspertide hinnangul oli materjal sisutihe ja mahukas, mistõttu on õpetajal võimalik teha valikuid vastavalt klassile. Murekohana toodi välja, et kuigi õpilaste endi koostatud ülesannete lahendamine on üldiselt hea idee, võib see õpetaja jaoks olla keeruline, kuna õppijad tegelevad erinevate ülesannetega ja õpetaja peab pidevalt erinevatesse ülesannetesse süvenema.

Tabel 13. Väidete hinnangud hulkade teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	1	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	0	3	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	1	2	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	3	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	1	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	3	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	1	2	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	1	2	<b>5</b>

### 3.2.13 Mitmemõõtmelised andmestruktuurid

Mitmemõõtmeliste andmestruktuuride teemat tagasisidestas kolm eksperti, kelle hinnangud esitatud väidetele olid sarnased (vt tabel 14). Ükski vastaja ei nimetanud materjalides puudujääke, küll aga mainis üks vastanud õpetaja, et materjal on palju erinevaid käsklusi, mille meeldejätmisega tekib probleem õpilasel, kel puudub huvi.

Kõik vastajad tõid esile Kuldvillaku mängu, mis võtab kokku kogu andmestruktuuride teema. Programmeerimist õpetanud üliõpilane täiendas oma hinnanguid: „Kuldvillak on suurepärane viis võtta kokku kogu andmestruktuuride teema. Ma arvan, et ma keskenduksingi põhiliselt sellele ja vaid põgusalt tutvustaks mitmemõõtmelisi andmestruktuure, et õpilased oleksid teadlikud, et neid on võimalik moodustada.“

Tabel 14. Väidete hinnangud mitmemõõtmeliste andmestruktuuride teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	1	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	2	1	<b>4</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	1	2	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	1	2	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	2	1	<b>4</b>
Materjal on lihtsasti kättesaadav	0	0	0	1	2	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	1	2	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	1	2	<b>5</b>

### 3.2.14 Projekt

Projekti teemale loodud tunnikavasid ja materjale hindas kolm õpetajat ja üks programmeerimist õpetanud üliõpilane. Väidete hinnangud olid vahemikus 3–5 (vt tabel 15). Üks vastaja täiendas oma hinnanguid kommentaariga „Kel on huvi, saab projektiga kenasti hakkama. Kel huvi vähene võib hätta jääda ja ka lisamaterjal ei pruugi aidata“.

Kõige kasulikumaks pidasid vastajad ideid, kuidas esitlemise protsessi läbi viia ning projekti esitlust. Üks õpetaja lisas, et tundides projektide arendamine on hea mõte, kuid ajakulu on keeruline hinnata ning reaalne ajakulu selgub alles siis, kui alustada tunnikavade järgi tundide läbiviimist. Kõige suuremaks murekohaks pidasid vastajad õpilaste võimalikku huvipuudust, mille korral ei ole ka lisamaterjalist väga palju kasu.

Tabel 15. Väidete hinnangud projekti teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	1	1	2	<b>4,5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	3	<b>5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	1	1	2	<b>4,5</b>
Materjal on arusaadavalt struktureeritud	0	0	1	1	2	<b>4,5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	1	0	3	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	4	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	1	3	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	2	2	<b>4,5</b>

### 3.2.15 Mängud

Kokkuvõtvale mängude teemale vastas kaks eksperti, kelle hinnangul oli materjali kvaliteet hea (vt tabel 16). Mõlemad vastajad tõid välja, et kursuse on hea lõpetada mänguliselt. Samuti toodi välja, et mängu saab kasutada ka teiste tundide juures ja anda näiteks kiirematele lahendajatele. Lisaks olemasolevatele viidetele soovitas üks vastaja täiendada tunnikava ka edasiõppimise võimalustega: „[---] Võib-olla võiks lisaks pakkuda välja ka edasiõppimise võimalusi, näiteks Tartu Ülikooli objektorienteeritud programmeerimise kursus või ülikoolide korraldatud veebikursused ja MOOCid, siis on õpilastel, kes soovivad programmeerimisega edasi tegelda, mingi siht olemas.“

Tabel 16. Väidete hinnangud mängude teemale

<b>Väide</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Me</b>
Õpetajamaterjal on mitmekesine	0	0	0	0	2	<b>5</b>
Materjal toetab õpieesmärkide saavutamist	0	0	0	1	1	<b>4,5</b>
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel	0	0	0	0	2	<b>5</b>
Materjal on arusaadavalt struktureeritud	0	0	0	0	2	<b>5</b>
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest	0	0	0	0	2	<b>5</b>
Materjal on lihtsasti kättesaadav	0	0	0	0	2	<b>5</b>
Materjali saab korduvalt kasutada	0	0	0	0	2	<b>5</b>
Õpetajamaterjal toetab programmeerimise õpetamist tunnis	0	0	0	0	2	<b>5</b>

#### **4. Piirangud ja edasiarendusvõimalused**

Õpetajamaterjali kvaliteedi hindamiseks kasutatud valimit võib pidada töö üheks piiranguks, sest väike valim ei pruugi olla piisav pädeva hinnangu saamiseks. Kitsaskohaks võib pidada ka asjaolu, et materjale polnud võimalik klassiruumis katsetada ega loodud tunnikavade alusel tunde läbi viia.

Tulevikus võiks õpetajatel olla võimalus luua materjalidele ligipääsuks ja haldamiseks oma konto. Hetkel ei lisatud materjalidesse ülesannete lahendusi, kuna kõik materjalid on vabalt kättesaadavad ka õpilastele. Õpetajamaterjal peaks olema piiratud, et oleks võimalik kaasata ka lahendusi ja muid materjale, mis ei tohi olla õpilastele nähtavad. Samuti tasub kaasata ülesannetele automaatkontrollid, et hõlbustada õpetajate tööd veel rohkem.

Tunnipikkused on koolides väga erinevad. Edasiarenduse käigus on võimalik tunnikavade juurde lisada soovitusi tunni planeerimiseks vastavalt õppetunni kestvusele. Samuti tuleks rohkem tähelepanu pöörata ülesannete diferentseerimisele. Ajalisest piirangust tulenevalt loodi käesolev lisamaterjal eeldusel, et õpilased alles alustavad programmeerimisega. Klassiruumis võib aga sageli leida õpilasi, kes on juba varem programmeerimisega kokku puutunud ning kellele tavapärased ülesanded pole piisavalt väljakutsuvad. Edasijõudnute edukaks kaasamiseks on vaja anda õpetajatele veel rohkem soovitusi ja nõuandeid, kuidas läheneda erineva tasemega õppijatele.

## Kokkuvõte

Magistritöö eesmärgiks oli luua õpetajamaterjal programmeerimisõpikule *progema.ee*. Õpetajamaterjale katsetasid üliõpilased ja õpetajad, kelle tagasiside põhjal loodud materjale täiendati ning kohandati. Töö teoreetilises ülevaates analüüsiti varasemaid töid ning uuriti, millised õppemeetodid on tõhusad programmeerimise õppimisel ja õpetamisel. Samuti valiti välja mudelid, mille abil materjale luua ja hinnata.

Materjalide loomisel kasutati ADDIE mudelit, mis oli aluseks töö planeerimisel. Esimeses etapis toimus vajaduste analüüsimine ja teoreetilise tausta uurimine, millele järgnes kavandamine. Kolmandas etapis loodi paika pandud struktuuri alusel õpiku juurde sobivad tunnikavad ja lisamaterjalid, pärast mida koostati tagasisideküsimustik ning ekspertidel oli võimalik loodud materjale katsetada ja tagasisidestada. Kogutud tagasiside põhjal hinnati loodud materjalide kvaliteeti ning tehti täiendusi ja parandusi.

Õpetajate ja üliõpilaste tagasiside põhjal vastati teisele uurimisküsimusele – kuidas hindavad loodud õpetajamaterjale eksperdid. Küsimustikele vastas 2–5 eksperti ning tulemustest selgus, et õppematerjalidega jäädi rahule. Ekspertide soovitude abil täiendati loodud materjale või toodi ettepanekud välja edasiarenduse võimalustena.

## Viidatud kirjandus

- [1] Kori K., Beldman P., Tõnisson E., Luik P., Suviste R., Siiman L., Pedaste M. IT oskuste arendamine Eesti koolides.  
<https://wise.com/documents/IT%20oskuste%20arendamine%20Eesti%20koolides.pdf>  
(04.12.2024).
- [2] Vabariigi Valitsuse 6. jaanuari 2011. a määrus nr 1 „Põhikooli riiklik õppekava“ Lisa 10.  
[https://www.riigiteataja.ee/akti/isa/1100/8202/4002/18m\\_pohi\\_lisa10.pdf#](https://www.riigiteataja.ee/akti/isa/1100/8202/4002/18m_pohi_lisa10.pdf#) (16.03.2025).
- [3] Dagienė V, Laanpere M, Borzovs J. Computing Education Research in Baltic Countries. Past, Present and Future of Computing Education Research. A Global Perspective., Cham: Springer International Publishing, 2023, p. 279–310. [https://doi.org/10.1007/978-3-031-25336-2\\_14](https://doi.org/10.1007/978-3-031-25336-2_14) (02.01.2025).
- [4] Demirkiran M. C., Tansu Hocanin F. An investigation on primary school students' dispositions towards programming with game-based learning. *Education and Information Technologies*, 2021, Vol. 26, p. 3871–3892. <https://doi.org/10.1007/s10639-021-10430-5> (23.01.2025).
- [5] Abesadze S., Nozadze D. Make 21st Century Education: The Importance of Teaching Programming in Schools. *International Journal of Learning and Teaching*, 2020, Vol. 6, No. 3, p. 158–163. <https://doi.org/10.18178/ijlt.6.3.158-163> (04.12.2024).
- [6] Lin P.-H., Chen S.-Y. Design and Evaluation of a Deep Learning Recommendation Based Augmented Reality System for Teaching Programming and Computational Thinking. *IEEE Access*, 2020, Vol. 8, p. 45689–45699. <https://doi.org/10.1109/ACCESS.2020.2977679> (04.12.2024).
- [7] Zhang X, Zhang C, Stafford TF, Zhang P. Teaching Introductory Programming to IS Students: The Impact of Teaching Approaches on Learning Performance. *Journal of Information Systems Education*, 2013, Vol. 24(2), p. 147–156.  
<https://jise.org/volume24/n2/JISEv24n2p147.pdf> (04.12.2024).
- [8] Gomes A., Mendes A. J. Learning to program - difficulties and solutions. *International Conference on Engineering Education*, 2007, Vol. 7.  
[https://www.researchgate.net/publication/228328491\\_Learning\\_to\\_program\\_-\\_difficulties\\_and\\_solutions](https://www.researchgate.net/publication/228328491_Learning_to_program_-_difficulties_and_solutions) (04.12.2024).

- [9] Wong G. K.-W., Cheung H.-Y. Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Enviroments*, 2020, Vol. 28(4), p. 438–450. <https://doi.org/10.1080/10494820.2018.1534245> (04.12.2024).
- [10] Tõnisson E., Palts T., Säde M., Tõnisson K., Suviste R., Meier H. Programmeerimine. Tartu Ülikool. <https://web.htk.tlu.ee/digitalu/programmeerimine/> (21.01.2025).
- [11] Relkin E., de Ruiter L. E., Bers M. U. Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 2021, Vol. 169. <https://doi.org/10.1016/j.compedu.2021.104222> (21.01.2025).
- [12] Moraiti I., Fotoglou A., Drigas A. Coding with Block Programming Languages in Educational Robotics and Mobiles, Improve Problem Solving, Creativity & Critical Thinking Skills. *International Journal of Interactive Mobile Technologies*, 2022, Vol. 16(20), p. 59–78. <https://doi.org/10.3991/ijim.v16i20.34247> (21.02.2025).
- [13] Bocconi S., Chiocciariello A., Dettori G., Ferrari A., Engelhardt K., Kamylyis P., Punie Y. Exploring the Field of Computational Thinking as a 21st Century Skill. *EDULEARN16 Proceedings*, 2016, p. 4725–4733. <https://doi.org/10.21125/edulearn.2016.2136> (21.01.2025).
- [14] Barr D., Harrison J., Conery L. Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology*, 2011, Vol. 38(6), p. 20–23. <https://files.eric.ed.gov/fulltext/EJ918910.pdf> (21.01.2025).
- [15] Voogt J., Erstad O., Dede C., Mishra P. Challenges to learning and schooling in the digital networked world of the 21st century. *Journal of Computer Assisted Learning*, 2013, Vol. 29(5), p. 403–413. <https://doi.org/10.1111/jcal.12029> (21.01.2025).
- [16] Angraini L. M., Yolanda F., Muhammad I. Augmented Reality: The Improvement of Computational Thinking Based on Students' Initial Mathematical Ability. *International Journal of Instruction*, 2023, Vol. 16(3), p. 1033–1054. <https://repository.uir.ac.id/23065/> (21.01.2025).
- [17] Põhikooli riiklik õppekava, Riigi Teataja. <https://www.riigiteataja.ee/akt/129082014020> (21.01.2025).
- [18] Kaufmann O. T., Stenseth B. Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 2021, Vol. 52(7), p. 1029–1048. <https://doi.org/10.1080/0020739X.2020.1736349> (21.01.2025).

- [19] Fitzgerald S., Simon B., Thomas L. Strategies that students use to trace code: an analysis based in grounded theory. Proceedings of the first international workshop on Computing education research, 2005, p. 69–80. <https://doi.org/10.1145/1089786.1089793> (21.01.2025).
- [20] Holvikivi J. Conditions for Successful Learning of Programming Skills. Key Competencies in the Knowledge Society, 2010, p. 155–164. [https://doi.org/10.1007/978-3-642-15378-5\\_15](https://doi.org/10.1007/978-3-642-15378-5_15) (21.01.2025).
- [21] Sobral S. R. The Old Question: Which Programming Language Should We Choose to Teach to Program? Advances in Digital Science, 2021, p. 351–364. [https://doi.org/10.1007/978-3-030-71782-7\\_31](https://doi.org/10.1007/978-3-030-71782-7_31) (21.01.2025).
- [22] Rich P. J., Larsen R. A., Mason S. L. Measuring teacher beliefs about coding and computational thinking. Journal of Research on Technology in Education, 2021, Vol. 53, p. 296–316. <https://doi.org/10.1080/15391523.2020.1771232> (02.01.2025).
- [23] Koppel K., Tammsaar H., Solnik S., Jaanits J., Kaljuvee E., Mäekivi D. Tuleviku tegija teekond startup ökosüsteemi. Rakendusliku Antropoloogia Keskus, Tartu, 2018. <https://media.voog.com/0000/0037/5345/files/Raport%2015.11.18.pdf> (21.01.2025).
- [24] Põldoja H., Laanpere M. Open Educational Resources in Estonia. Current State of Open Educational Resources in the “Belt and Road” Countries, 2020, p. 35–47. [https://doi.org/10.1007/978-981-15-3040-1\\_3](https://doi.org/10.1007/978-981-15-3040-1_3) (21.01.2025).
- [25] Gümnaasiumi informaatika uute valikkursuste ja õppeprotsesside kirjeldused. <https://oppekava.ee/wp-content/uploads/2021/08/Gumnaasiumi-informaatika-kursuste-ja-oppeprotsesside-kirjeldused.pdf> (21.01.2025).
- [26] Aru-Chabilan H. Tiger Leap for digital turn in the Estonian education. Educational Media International, 2020, Vol. 57, p. 61–72. <https://doi.org/10.1080/09523987.2020.1744858> (22.01.2025).
- [27] ProgeTiigri kogumik. <https://progetiiger.ee/> (22.01.2025).
- [28] Macrides E., Miliou O., Angeli C. Programming in early childhood education: A systematic review. International Journal of Child-Computer Interaction, 2022, Vol. 32, p. 100396. <https://doi.org/10.1016/j.ijcci.2021.100396> (21.01.2025).
- [29] Henno I., Ivanova K., Jürimäe M., Kasesalk M., Lamesoo K., Pihu M., Tagamets E., Vanari K. Eesti põhikooli riiklik õppekava rahvusvahelises võrdluses, 2021.

[https://www.hm.ee/sites/default/files/documents/2022-07/oeed\\_sekundaaranaluu\\_jaan\\_2022.pdf](https://www.hm.ee/sites/default/files/documents/2022-07/oeed_sekundaaranaluu_jaan_2022.pdf) (21.01.2025).

[30] Kilhamn C., Bråting K., Rolandsson L. Teachers' arguments for including programming in mathematics education. *Svensk förening för MatematikDidaktisk Forskning*, 2021, p. 169–176. <https://uu.diva-portal.org/smash/get/diva2:1541426/FULLTEXT01.pdf> (21.01.2025).

[31] Thornhill-Miller B., Camarda A., Mercier M., Burkhardt J.-M., Morisseau T., Bourgeois-Bougrine S., Vinchon F., El Hayek S., Augereau-Landais M., Mourey F., Feybesse C., Sundquist D., Lubart T. Creativity, Critical Thinking, Communication, and Collaboration: Assessment, Certification, and Promotion of 21st Century Skills for the Future of Work and Education. *Journal of Intelligence*, 2023, Vol. 11(3), p. 54. <https://doi.org/10.3390/jintelligence11030054> (22.01.2025).

[32] Tang K. H. D. Student-centered Approach in Teaching and Learning: What Does It Really Mean? *Acta Pedagogica Asiana*, 2023, Vol. 2(2), p. 72–83. <https://doi.org/10.53623/apga.v2i2.218> (01.05.2025).

[33] Taşpolat A., Özdamlı F., Soykan E. Programming Language Training With the Flipped Classroom Model. *Sage Open*, 2021, Vol. 11. <https://doi.org/10.1177/21582440211021403> (23.01.2025).

[34] Kandemir C. M., Kalelioğlu F., Gülbahar Y. Pedagogy of teaching introductory text-based programming in terms of computational thinking concepts and practices. *Computer Applications in Engineering Education*, 2021, Vol. 29, p. 29–45. <https://doi.org/10.1002/cae.22374> (22.01.2025).

[35] Campe S., Denner J., Green E., Torres D. Pair programming in middle school: variations in interactions and behaviors. *Computer Science Education*, 2020, Vol. 30, p. 22–46. <https://doi.org/10.1080/08993408.2019.1648119> (22.01.2025).

[36] Denner J., Werner L., Campe S., Ortiz E. Pair Programming: Under What Conditions Is It Advantageous for Middle School Students? *Journal of Research on Technology in Education*, 2014, Vol. 46, p. 277–96. <https://doi.org/10.1080/15391523.2014.888272> (22.01.2025).

[37] Maguire P., Maguire R., Hyland P., Marshall P. Enhancing collaborative learning using pair programming: Who benefits? *The All Ireland Journal of Teaching and Learning in*

*Higher Education*, 2014, Vol. 6(2). <https://ojs.aishe.org/index.php/aishe-j/article/view/141/251> (22.01.2025).

[38] Van Toll III T., Lee R., Ahlswede T. Evaluating the Usefulness of Pair Programming in a Classroom Setting. *6th IEEE/ACIS International Conference on Computer and Information Science*, 2007, p. 302–308. <https://doi.org/10.1109/ICIS.2007.96> (22.01.2025).

[39] Chong J., Hurlbutt T. The Social Dynamics of Pair Programming. *29th International Conference on Software Engineering*, 2007, p. 354–363. <https://doi.org/10.1109/ICSE.2007.87> (22.01.2025).

[40] Bjursten E.-L., Nilsson T., Gumaelius L. Computer programming in primary schools: Swedish Technology Teachers' pedagogical strategies. *International Journal of Technology and Design Education*, 2023, Vol. 33, p. 1345–1368. <https://doi.org/10.1007/s10798-022-09786-7> (23.01.2025).

[41] Wei X., Lin L., Meng N., Tan W., Kong S.-C., Kinshuk. The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers & Education*, 2021, Vol. 160. <https://doi.org/10.1016/j.compedu.2020.104023> (22.01.2025).

[42] Williams L., McCrickard D. S., Layman L., Hussein K. Eleven Guidelines for Implementing Pair Programming in the Classroom. *Agile 2008 Conference*, 2008, p. 445–452. <https://doi.org/10.1109/Agile.2008.12> (22.01.2025).

[43] Michinov N., Morice J., Ferrières V. A step further in Peer Instruction: Using the Stepladder technique to improve learning. *Computers & Education*, 2015, Vol. 91, p. 1–13. <https://doi.org/10.1016/j.compedu.2015.09.007> (23.01.2025).

[44] Simon B., Kohanfars M., Lee J., Tamayo K., Cutts Q. Experience report: peer instruction in introductory computing. *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, p. 341–345. <https://doi.org/10.1145/1734263.1734381> (23.01.2025).

[45] Porter L., Bailey Lee C., Simon B., Cutts Q., Zingaro D. Experience report: a multi-classroom report on the value of peer instruction. *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, 2011, p. 138–142. <https://doi.org/10.1145/1999747.1999788> (23.01.2025).

- [46] Rivadeneira J., Inga E. Interactive Peer Instruction Method Applied to Classroom Environments Considering a Learning Engineering Approach to Innovate the Teaching–Learning Process. *Education Sciences*, 2023, Vol. 13, p. 301. <https://doi.org/10.3390/educsci13030301> (23.01.2025).
- [47] Tan E., Small A., Lewis P. Have a question? Just ask it: using an anonymous mobile discussion platform for student engagement and peer interaction to support large group teaching. *Research in Learning Technology*, 2020, Vol. 28. <https://doi.org/10.25304/rlt.v28.2323> (23.01.2025).
- [48] Mshayisa V. V. Students’ perceptions of Plickers and crossword puzzles in undergraduate studies. *Journal of Food Science Education*, 2020, Vol. 19, p. 49–58. <https://doi.org/10.1111/1541-4329.12179> (23.01.2025).
- [49] Eppard J., Rochdi A. A Framework for Flipped Learning. International Association for the Development of the Information Society, 2017. <https://eric.ed.gov/?id=ED579204> (23.01.2025).
- [50] Tawfik A. A., Lilly C. Using a Flipped Classroom Approach to Support Problem-Based Learning. *Technology, Knowledge and Learning*, 2015, Vol. 20, p. 299–315. <https://doi.org/10.1007/s10758-015-9262-8> (23.01.2025).
- [51] Awidi I. T., Paynter M. The impact of a flipped classroom approach on student learning experience. *Computers & Education*, 2019, Vol. 128, p. 269–83. <https://doi.org/10.1016/j.compedu.2018.09.013> (23.01.2025).
- [52] Özyurt H., Özyurt Ö. Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy. *Computer Applications in Engineering Education*, 2018, Vol. 26, p. 2036–2046. <https://doi.org/10.1002/cae.21973> (23.01.2025).
- [53] Ozdamli F., Asiksoy G. Flipped Classroom Approach. *World Journal on Educational Technology: Current Issues*, 2016 Vol. 8, p. 98–105. <https://eric.ed.gov/?id=EJ1141886> (23.01.2025).
- [54] Sobral S. R. Flipped Classrooms for Introductory Computer Programming Courses. *International Journal of Information and Education Technology*, 2021, Vol. 11, p. 178–183. <https://doi.org/10.18178/ijiet.2021.11.4.1508> (23.01.2025).

- [55] Enfield J. Looking at the Impact of the Flipped Classroom Model of Instruction on Undergraduate Multimedia Students at CSUN. *TechTrends*, 2013, Vol. 57, p. 14–27. <https://doi.org/10.1007/s11528-013-0698-1> (23.01.2025).
- [56] Manasrah A., Masoud M., Jaradat Y. Short Videos, or Long Videos? A Study on the Ideal Video Length in Online Learning. *2021 International Conference on Information Technology*, 2021, p. 366–370. <https://doi.org/10.1109/ICIT52682.2021.9491115> (10.04.2025).
- [57] Yu Z., Gao M. Effects of Video Length on a Flipped English Classroom. *SAGE Open*, 2022, Vol. 12. <https://doi.org/10.1177/21582440211068474> (10.04.2025).
- [58] Galindo-Dominguez H. Flipped Classroom in the Educational System: Trend or Effective Pedagogical Model Compared to Other Methodologies? *Educational Technology & Society*, 2021, Vol. 24, p. 44–60. <https://www.jstor.org/stable/27032855> (23.01.2025).
- [59] Hendrik H., Hamzah A. Flipped Classroom in Programming Course: A Systematic Literature Review. *International Journal of Emerging Technologies in Learning*, 2021, Vol. 16, p. 220–236. <https://www.learntechlib.org/p/218943/> (23.01.2025).
- [60] Grace L. *Doing Things with Games : Social Impact Through Play*. Boca Raton: CRC Press. 2019. <https://doi.org/10.1201/9780429429880> (23.01.2025).
- [61] Dichev C., Dicheva D. Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International Journal of Educational Technology in Higher Education*, 2017, Vol. 14. <https://doi.org/10.1186/s41239-017-0042-5> (23.01.2025).
- [62] Mercier M., Lubart T. Board games enhance creativity: Evidence from two studies. *Psychology of Aesthetics, Creativity, and the Arts*, 2023. <https://doi.org/10.1037/aca0000547> (23.01.2025).
- [63] Wong C. H. T., Yunus M. M. Board Games in Improving Pupils’ Speaking Skills: A Systematic Review. *Sustainability*, 2021, Vol. 13. <https://doi.org/10.3390/su13168772> (23.01.2025).
- [64] Seralidou E., Douligeris C. Learning programming by creating games through the use of structured activities in secondary education in Greece. *Education and Information Technologies*, 2021, Vol. 26, p. 859–898. <https://doi.org/10.1007/s10639-020-10255-8> (24.01.2025).

- [65] Espinal A., Vieira C., Guerrero-Bequis V. Student ability and difficulties with transfer from a block-based programming language into other programming languages: a case study in Colombia. *Computer Science Education*, 2023, Vol. 33, p. 567–599.  
<https://doi.org/10.1080/08993408.2022.2079867> (24.01.2025).
- [66] Simian D., Vulpeanu A. Using Python in developing video games. *Bulletin of the Transilvania University of Brasov. Series III: Mathematics and Computer Science*, 2022, p. 203–214. <https://doi.org/10.31926/but.mif.2022.2.64.2.16> (24.01.2025).
- [67] Szydłowska J., Miernik F., Ignasiak M. S., Swacha J. Python Programming Topics That Pose a Challenge for Students. *Third International Computer Programming Education Conference*, 2022, Vol. 102 p. 7:1–7:9. <https://doi.org/10.4230/OASiCs.ICPEC.2022.7> (24.01.2025).
- [68] Yilmaz Ince E., Koc M. The consequences of robotics programming education on computational thinking skills: An intervention of the Young Engineer’s Workshop (YEW). *Computer Applications in Engineering Education*, 2021, Vol. 29, p. 191–208.  
<https://doi.org/10.1002/cae.22321> (24.01.2025).
- [69] Valls Pou A., Canaleta X., Fonseca D. Computational Thinking and Educational Robotics Integrated into Project-Based Learning. *Sensors*, 2022, Vol. 22.  
<https://doi.org/10.3390/s22103746> (24.01.2025).
- [70] Zhang Y., Zhu Y. Effects of educational robotics on the creativity and problem-solving skills of K-12 students: a meta-analysis. *Educational Studies*, 2024, Vol. 50, p. 1539–1557.  
<https://doi.org/10.1080/03055698.2022.2107873> (24.01.2025).
- [71] Temel Y., Coşkun B. A Review on Artificial Intelligence Applications in Education in Some Countries and Türkiye. *International Technology and Education Journal*, 2024, Vol. 8(1), p. 9–23. <https://files.eric.ed.gov/fulltext/EJ1445417.pdf> (22.01.2025).
- [72] Yilmaz R., Karaoglan Yilmaz F. G. The effect of generative artificial intelligence (AI)-based tool use on students’ computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, 2023, Vol. 4.  
<https://doi.org/10.1016/j.caeai.2023.100147> (24.01.2025).
- [73] Ma Q., Wu T., Koedinger K. Is AI the better programming partner? Human-Human Pair Programming vs. Human-AI pAIr Programming, 2023.  
<https://doi.org/10.48550/arXiv.2306.05153> (24.01.2025).

- [74] Sarkar A., Gordon A. D., Negreanu C., Poelitz C., Ragavan S. S., Zorn B. What is it like to program with artificial intelligence? 2022. <https://doi.org/10.48550/arXiv.2208.06213> (24.01.2025).
- [75] Denny P., Prather J., Becker B. A., Finnie-Ansley J., Hellas A., Leinonen J., Luxton-Reilly ., Reeves B. N., Santos E. A., Sarsa S. Computing Education in the Era of Generative AI. *Commun. ACM*, 2024, Vol. 67, p. 56–67. <https://doi.org/10.1145/3624720> (24.01.2025).
- [76] Al-Tkhayneh K. M., Alghazo E. M., Tahat D. The Advantages and Disadvantages of Using Artificial Intelligence in Education. *Journal of Educational and Social Research*, 2023, Vol. 13(4), p. 105–117. <https://doi.org/10.36941/jesr-2023-0094> (24.01.2025).
- [77] Postrid. Töötukassa. <https://www.tootukassa.ee/et> (24.01.2025).
- [78] Iqbal Md. H., Siddiqie S. A., Mazid Md. A. Rethinking theories of lesson plan for effective teaching and learning. *Social Sciences & Humanities Open*, 2021, Vol. 4. <https://doi.org/10.1016/j.ssaho.2021.100172> (04.03.2025).
- [79] Curry J. H., Johnson S., Peacock R. Robert Gagné and the Systematic Design of Instruction. [https://edtechbooks.org/id/robert\\_gagn\\_and\\_systematic\\_design](https://edtechbooks.org/id/robert_gagn_and_systematic_design) (24.01.2025).
- [80] Tartu Ülikooli kursused. Programmeerimine lastele ja noortele. <https://courses.cs.ut.ee/t/kids/Main/Tunnikavad> (22.01.2025).
- [81] Adeoye M. A., Wirawan K. A. S. I., Pradnyani M. S. S, Septiarini N. I. Revolutionizing Education: Unleashing the Power of the ADDIE Model for Effective Teaching and Learning. *Jurnal Pendidikan Indonesia*, 2024, Vol. 13, p. 202–9. <https://doi.org/10.23887/jpiundiksha.v13i1.68624> (04.03.2025).
- [82] Surdyanto A., Kurniawan W. Developing critical reading module using integrated learning content and language approach. *Studies in English Language and Education*, 2020, Vol. 7, p. 154–169. <https://doi.org/10.24815/siele.v7i1.15098> (04.03.2025).
- [83] Palinkas L. A., Horwitz S. M., Green C. A., Wisdom J. P., Duan N., Hoagwood K. Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Administration and Policy in Mental Health and Mental Health Services Research*, 2015, Vol. 42(5), p. 533–544. <https://doi.org/10.1007/s10488-013-0528-y> (16.03.2025).

[84] Nesbit J. C., Li J. Web-Based Tools for Learning Object Evaluation. 2004.

[https://www.researchgate.net/publication/252814007\\_Web-Based\\_Tools\\_for\\_Learning\\_Object\\_Evaluation](https://www.researchgate.net/publication/252814007_Web-Based_Tools_for_Learning_Object_Evaluation) (20.03.2025).

[85] South L., Saffo D., Vitek O., Dunne C., Borkin M. A. Effective Use of Likert Scales in Visualization Evaluations: A Systematic Review. *Computer Graphics Forum*, 2022, Vol. 41(3), p. 43–55. <https://doi.org/10.1111/cgf.14521> (20.03.2025).

## Lisad

### I. Tagasisideküsimustik

1. Kas töötad õpetajana või oled tudeng?

- Töötan õpetajana
- Olen tudeng
- Olen õpetajana töötav tudeng

2. Milline on Sinu programmeerimise õpetamise kogemus?

- Olen varem programmeerimist õpetanud
- Õpetan programmeerimist
- Soovin tulevikus programmeerimist õpetada
- Ma pole kunagi programmeerimist õpetanud ega soovi seda teha ka tulevikus

3. Palun hinda, mil määral nõustud järgmiste väidetega.

	1 – ei nõustu üldse	2 – ei nõustu	3 – ei oska öelda	4 – nõustun	5 – nõustun täielikult
Õpetajamaterjal on mitmekesine					
Materjal toetab õpieesmärkide saavutamist					
Materjal toetab õpilase motivatsiooni programmeerimise õppimisel					
Materjal on arusaadavalt struktureeritud					
Materjali on võimalik kohandada ja diferentseerida lähtuvalt vajadustest					

Materjal on lihtsasti kättesaadav					
Materjali saab korduvalt kasutada					
Õpetajamaterjal toetab programmeerimise õpetamist tunnis					

4. Soovi korral täienda väidete hinnanguid.

5. Kas tundsid materjali juures millestki puudust?

- Ei
- Jah

6. Kui vastasid eelmisele küsimusele jah, siis palun täienda, millest puudust tundsid.

7. Millised materjalid olid kõige kasulikumad?

8. Soovi korral saad oma lisamõtted kirja panna siin.

## II. Ekspertide soovitused ja täienduste ettepanekud

Teema	Soovitus	Kas võeti arvesse?
Sissejuhatus	Tõsta märges Thonny keele kontrollimise kohta materjalis ettepoole.	jah
	Luu juurde inimbingo mängulaud 12 väljaga.	jah
	Lisada materjalidesse bingolaua fail, mida on võimalik redigeerida.	jah
	Täiendada materjale oskuste teljega kursuse lõpus.	jah
Algoritm	Täiendada tunni lõppu metodoloogiliste soovitustega, näiteks võib õpetaja lõigata plokk skeemi tükkideks ja õpilased peavad selle ise kokku panema.	jah
Muutujad ja avaldised	Lisada tundide juurde, mille kavas on Blooket, et õpetajal peab mängu kasutamiseks olema konto.	jah
Funktsioonid	Luu materjal, kuidas lihtsamalt õpilastele seletada funktsioonide olemust ja loomust.	ei
Veahaldus	Täiendada tundi sõnumiga, et paljudel juhtudel saame sisendi sobivust kontrollida tingimuslausega.	jah
Projekti tutvustus ja AI	Panna kirja optimaalne rühma suurus ja projekti üldine eesmärk.	jah
Sõnastikud	Täiendada keskmise temperatuuri väljastamise lisaülesannet nii, et kuu nimi on failis samal real kui vastav temperatuur ja kasutaja peab sisestama kuu nimetuse varasema numbri asemel. Naisjalgpallurite ülesandes kirjutada klubi nimi ja naiste arv samale reale, näiteks semikooloniga eraldatult.	jah
Hulgad	Mainida õpilastele, et õpikus on peatükk ka neljanda andmestruktuuri (ennikute) kohta, mille kohta soovijad saavad iseseisvalt uurida.	jah
Projekt	Lisada õpetaja esitluse lõppu slaid „Aitäh kuulamast!“, et õpilastel oleks silme eest hea tava, kuidas esitlus lõpetatakse.	jah
Mängud	Pakkuda välja ka edasiõppimise võimalusi.	jah

### **III. Litsents**

#### **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Kaisa Liina Keps,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Õpetajamaterjal programmeerimise õpetamiseks II ja III kooliastmes“, mille juhendaja on Reelika Suviste, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Kaisa Liina Keps*

**19.05.2025**