TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Kristiina Keps

Veebirakendus programmeerimise õppimiseks II ja III kooliastmes

Magistritöö (30 EAP)

Juhendaja: Marina Lepp

Tartu 2023

Veebirakendus programmeerimise õppimiseks II ja III kooliastmes

Lühikokkuvõte:

Põhikoolis programmeerimise õppimine aitaks suunata rohkem noori infotehnoloogia sektorisse ja oleks kasulik ka õpilastele endile. Õpilaste seas on huvi programmeerimise õppimise vastu olemas, kuid praegu puuduvad sobivad eestikeelsed õppematerjalid sellele vanuserühmale. Magistritöö raames loodi II ja III kooliastme õpilastele suunatud digitaalne programmeerimisõpik, mis võimaldab programmeerimist iseseisvalt õppida. Õppematerjali loomisel tugineti ADDIE mudelile, mille esimestes etappides analüüsiti vajadust uue õppematerjali järele ning uuriti programmeerimise õpetamist ja digitaalse õppematerjali loomist. Kavandamise etapis määrati õpiväljundid ja õpiku struktuur ning kolmandas etapis loodi õpik. Õpiku loomiseks kasutati veebiraamistikku Angular, mis võimaldas paindlikkust erinevate interaktiivsete komponentide koostamisel. Loodud õpik on leitav aadressilt *progema.ee*. Viimaks hindasid valminud õpikut õpetajad ja õpilased, kelle tagasiside põhjal viidi sisse parandusi, anti hinnang loodud õppematerjalile ja kaardistati edasiarendusvõimalused.

Võtmesõnad:

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

CERCS: S270 Pedagoogika ja didaktika, P175 Informaatika, süsteemiteooria

Web Application for Studying Programming at the Second and Third School Level

Abstract:

Teaching programming in basic schools would assist in leading more young people toward the information technology sector and benefit the students as well. The pupils are interested in learning programming, but currently, there are no suitable study materials in Estonian for this age group. This master's thesis aimed to create a digital programming study material for the students of the second and third school level that allows them to study programming independently. The study materials were developed using the ADDIE model. During the first stage of the model, the need for a new study material was analyzed and some research was carried out about teaching

programming and creating study materials. In the design phase, the learning outcomes were defined along with the structure of the study material. The third phase focused on developing the study material, using the Angular framework, which allowed flexibility in designing interactive components. The study material is available on *progema.ee*. Finally, teachers and students gave their opinion of the study material, based on which the material was improved and assessed. The feedback was also used to map out possible further plans for the study material.

Keywords:

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

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

Sisukord

Sissejuhatus

Eesti Töötukassa on aastaid viinud läbi tööturu uuringuid ja hinnanud tööjõu nõudlust eri valdkondades. Iga aasta on nende uuringute põhjal olnud suur puudus tarkvaraarendajatest ja prognooside kohaselt on tarkvaraarenduses tööjõu puudus ka lähiaastatel [1]. Et rohkem noori leiaks tee infotehnoloogia sektorisse (IT-sektorisse), on vaja neile seda valdkonda tutvustada ja lasta neil programmeerimist omal käel kogeda. Programmeerimise õppimine on kasulik ka õpilastele endile, arendades õpilastes nii loomingulist kui kriitilist mõtlemist ja probleemide lahendamise oskust [2].

Riiklik õppekava ei näe ette põhikoolis programmeerimise õpetamist [3], samas õpilased ise oleksid huvitatud programmeerimise õppimisest [4]. Praegu on informaatika õpetamine Eesti üldhariduskoolides ebaühtlane ja vaid 1% Eesti üldhariduskoolidest viitab oma õppekavas programmeerimisele või robootikale [5]. IT-huviringe pakuvad kaks kolmandikku Eesti üldhariduskoolidest, millest omakorda vaid 38% keskenduvad programmeerimisele [5]. Seega puudub paljudel õpilastel võimalus koolis või huviringis programmeerimist õppida, kuigi huvi selle vastu on olemas. Iseseisvat õppimist ingliskeelsete õppematerjalidega võib noores eas takistada veel vähene keeleoskus [6] ja suurem osa eestikeelseid õppematerjale on suunatud üliõpilastele või gümnasistidele.

Probleemi aitaks lahendada uus eestikeelne digitaalne õppematerjal, mis oleks suunatud noortele ja mille abil neil oleks võimalik iseseisvalt programmeerimist õppida. Sellest tulenevalt seati magistritöö eesmärgiks II ja III kooliastmele digitaalse programmeerimisõpiku loomine. Püstitati ka kaks uurimisküsimust, mis toetavad selle eesmärgi saavutamist:

1) Kuidas luua põhikoolile sobiv digitaalne programmeerimisõpik?
2) Kuidas hindavad loodud õpikut õpetajad ja õpilased?

Töö on jagatud viide peatükki. Esimeses peatükis antakse ülevaade programmeerimise õpetamisest. Teine peatükk räägib olemasolevatest programmeerimise õppematerjalidest. Kolmas peatükk keskendub digitaalse õppematerjali loomisele. Eelviimases peatükis kirjeldatakse magistritöö tulemusena valminud veebirakenduse loomist ja viimases peatükis tuuakse välja töö piirangud ja edasiarendamise võimalused.

1. Programmeerimise õpetamine

Juba rohkem kui 20 aastat tagasi täheldati, kui oluline on programmeerimisega tutvumine infotehnoloogilises maailmas. Seda mitte ainult selle eriala inimestele, vaid kõigile, kuna programmeerimine arendab abstraktset mõtlemisoskust [7]. Programmeerimise õppimine õpetab ka probleemide lahendamist ja süsteemide kavandamist, sest programmeerides tuleb luua plaan probleemi lahendamiseks, see plaan täpseteks käskudeks teisendada ning lõpuks hinnata tulemust [8]. Programmeerimine on väärtuslik oskus, kuid algajana võib selle õppimine olla keeruline [9].

Vujošević-Janičić ja Tošić kirjutavad, et programmeerimise õppimise lihtsustamiseks on vajalik valida sobiv programmeerimiskeel, mis laseks keskenduda programmeerimise põhitõdedele, mitte keele süntaksile [10]. Näiteks C, C++ ja Java on küll väga populaarsed IT-tööstuses, kuid süntaksilt liiga keerulised esimeseks programmeerimiskeeleks. Stajano arvates sobib kõige paremini esimeseks programmeerimiskeeleks Python, tänu selle keele abstraktsusele, lihtsusele ja mitmekesisusele [11]. Pythoni kasutamine aitab hoida õppijate motivatsiooni ja vähendada kursuse katkestajate arvu [12]. Noortele sobivad alustamiseks paremini visuaalsed programmeerimiskeeled, nagu Scratch, kuid keerulisemate teemadeni jõudes tuleb jätkata mõne tekstipõhise programmeerimiskeelega, milleks sobib noortele samuti hästi Python [13].

Programmeerimise õpetamine peab sisaldama nii süntaksi, kontseptsioonide ja konstruktsioonide õpetamist, kui ka reeglite ja algoritmide koostamist. Kusjuures süntaksist arusaamine on tihtipeale lihtsam kui algoritmide planeerimine [14]. Algoritmilist mõtlemist aitab arendada samm-sammult lahenduskäikude nägemine ja selgituste saamine [15]. See muudab õpilastele arusaadavamaks programmeerimise ajal toimuvad mõtteprotsessid.

Süntaksist ja erinevatest konstruktsioonidest arusaamise testimiseks sobib kasutada enesekontrolliküsimusi, kus näiteks antakse ette koodijupp ja palutakse ennustada programmi töö tulemust [16]. Programmeerimise suurtes kursustes ehk MOOC-ides (ingl *massive open online course*) on samuti kasutatud vabatahtlikke enesekontrolliküsimusi ning 96,1% kursusel õppijatest neid küsimusi õppimisel ka kasutas ja 95,5% kasutajatest pidas neid ka kasulikuks [17].

Programmeerimist tuleks hakata õpetama juba noores eas ja järjest enam hakatakse seda ka eri riikides põhikoolides õpetama [13]. Juba põhikoolis ja gümnaasiumis programmeerimist õppides

on lootust, et peale lõpetamist minnakse edasi informaatikat õppima ja selle haridusega inimestest on tööturul suur puudus [13, 18]. Teisalt on noortele väärtuslikud programmeerimisega arendatav probleemilahendusoskus, analüütiline mõtlemine ja loovus [18].

Dagienė jt uurisid informaatika õpetamist põhikoolis eri riikides ning leidsid, et tihti jääb takistuseks õpetajate vähene valmisolek ja põhjalike teadmiste puudumine informaatika valdkonnas [19]. Kõige madalamaks hindasid õpetajad just enda teadmisi programmeerimise ja algoritmide õpetamisel.

1.1 Programmeerimise õpetamine Eestis

Eesti üldhariduskoolid lähtuvad oma tegevuses riiklikust õppekavast, kus 2011. aasta seisuga on informaatika valikaine, mida koolid pole kohustatud õpetama [20]. Põhikooli informaatika valikõppeaine keskendub tekstitöötlusele, info otsimisele, failide haldamisele, andmetöötlusele ning diagrammide ja esitluste koostamisele, kuid ei näe ette programmeerimise või informaatika kui arvutiteaduse õpetamist [3].

Rakendusliku Antropoloogia Keskuse (RAK) 2018. aastal läbiviidud uuringust „Tuleviku tegija teekond *startup* ökosüsteemi“ selgus, et 1.–6. klassi laste jaoks on oluline, et õppimine oleks loominguline ja mänguline [6]. Lihtsalt juhendit järgides hakkab neil igav. Selles eas takistab iseseisvat õppimist veel vähene inglise keele oskus, mis on vajalik mõistmaks ingliskeelseid programmeerimise õppematerjale ja foorumeid. Nii noorte IT-huviliste kui ka iduettevõtjate sõnul omandatakse suur osa oskusi iseseisvalt õppides, sest kooli arvutitunnid ei anna vajalikke oskusi ning jäävad tihtipeale igavaks.

Aasta hiljem TransferWise’i tellimusel valminud uuring „IT oskuste arendamine Eesti koolides“ kinnitab samuti, et enamasti on arvutitundide sisuks teksti- ja tabelitöötlus ning infootsing, samas kui õpilaste endi sõnul oleksid nad kõige rohkem huvitatud programmeerimise õppimisest [4]. Umbes 57% uuringus osalenud 9. ja 12. klassi õpilastest olid programmeerimist juba õppinud. Kõige enam õpiti programmeerimist koolis, kuid teisel kohal oli iseseisev õppimine ning seejärel huviringid ja kursused väljaspool kooli. Samas vaid 2,6% õpilastest vastas, et nad on loonud keerukamaid programme, 17,3% vastas, et nad on loonud lihtsamaid programme, 20,9% on natuke programmeerimist proovinud ning ülejäänud 16,2%, kes olid programmeerimist õppinud, ei tea

programmeerimisest eriti midagi või omavad vaid ettekujutust programmeerimisest. Sarnaselt RAK-i uuringule, tõid ka siin õpilased välja, et mängulisus ja ise katsetamine on olulised IT-huvi algatamises.

Veltmann uuris oma bakalaureusetöös informaatika õpetamist teises ja kolmandas kooliastmes Tartumaal [21]. Koolides läbiviidud intervjuudest selgus, et õpilastel on huvi programmeerimise vastu ja nad on huvitatud enamast kui Scratchi abil piltprogrammeerimisest. Paraku on II ja III kooliastmele programmeerimise õpetamiseks sobivaid materjale vähe ning ülikooli korraldatud MOOC-id jäävad õpilastele liiga keeruliseks.

Informaatikaõpetajad ise tunnevad kõige rohkem koolitusvajadust praktiliste oskuste ja tehnoloogiahariduse-alaste teadmiste valdkonnas [22]. 2021. aasta alguses informaatikaõpetajate seas läbi viidud uuringus selgus, et üle poole vastanutest pidas vajalikuks täiendavaid praktiliste oskuste koolitusi, sealhulgas programmeerimise koolitusi. Samas on umbes kolmandiku arvates koolitustel osalemine keeruline töögraafiku ja ajapuuduse tõttu. Lisaks selgus uuringust, et informaatikaõpetajad tunnevad puudust õppematerjalidest.

2017. aastal valminud Praxise uuring „IKT-haridus: digioskuste õpetamine, hoiakud ja võimalused üldhariduskoolis ja lasteaias“ kinnitab samuti, et digioskuste õpetamine erinevates üldhariduskoolides on ebaühtlane [5]. Umbes pooltes koolides toimub digioskuste õpetamine eraldi õppeainena ja pooltes lõimitult teistesse õppeainetesse. Samuti on ebaühtlane erinevate õpetatavate oskuste osakaal. Kõige rohkem keskendutakse infootsingule ja teksti vormindamisele, tehnoloogiaharidusega seotud tegevusi, nagu robootika või programmeerimine, õpetatakse väga vähesel määral. Vaid 1% üldhariduskoolidest on oma ainekavas viidanud programmeerimisele või robootikale.

Praxise uuringust selgus ka, et IT-huviringe pakuvad vaid kaks kolmandikku üldhariduskoolidest, eelkõige suuremad koolid. Pakutavatest huviringidest 38% keskenduvad programmeerimisele [5]. Seega, suurel osal üldhariduskoolide õpilastest puudub võimalus programmeerimist koolis või huviringis õppida.

2. Veebimaterjalid programmeerimise õppimiseks

Kuna koolis ja huviringis on paljudel õpilastel piiratud võimalused programmeerimist õppida, siis jääb nende jaoks variandiks kasutada mõnda digitaalset õppematerjali iseseisvaks õppimiseks. Selles peatükis on uuritud tasuta saadaolevaid digitaalseid eesti- ja ingliskeelseid õppematerjale ning kas need võiksid sobida põhikooli õpilastele.

2.1 Eestikeelsed õppematerjalid

Suure osa eestikeelseid õppematerjale on välja arendanud Eesti ülikoolid ja need on peamiselt koolides või mõnel ülikooli kursusel kasutamiseks. Põhikoolis informaatika õpetamiseks on loodud digiõpikud, gümnaasiumi jaoks programmeerimise valikkursused ning ülikooli kursuste jaoks programmeerimisõpikud. Lisaks on vabalt leitavad ka osad Tartu Ülikooli korraldatud MOOC-ide materjalidest.

2.1.1 Tartu Ülikooli „Programmeerimise õpik“

Tartu Ülikooli arvutiteaduse instituudis kirjutatud programmeerimisõpik annab hea ülevaate programmeerimise põhitõdedest ja sobib ka iseõppijatele [23]. See on aga suunatud eelkõige informaatika bakalaureuseõppe üliõpilastele, seega põhikooli õpilastele võivad siin toodud kirjeldused jääda keeruliseks ja materjal ei pruugi neid köita. Samuti pole selles õpikus interaktiivseid enesekontrolliküsimusi.

Teemadest kaetakse siin õpikus [23] avaldised, lihtlaused, tingimus- ja korduslaused, funktsioonid, andmestruktuurid (järjend, mitmemõõtmeline järjend, hulk, sõnastik) ning rekursioon. Samuti on õpikus juhendeid mitmete moodulite kasutamiseks, näiteks kilpkonnagraafika, EasyGui, Pygame ja Matplotlib, ning lisamaterjali ka programmidevahelise suhtluse ja veebiprogrammide kohta.

2.1.2 Digiõpikud I ja II kooliastmele

Nii esimesele kui ka teisele kooliastmele on koostatud digiõpikud, mis on suunatud nii õpetajatele kui õpilastele tunnis kasutamiseks [24]. Mõlema kooliastme teemade seas on ka programmeerimist, esimese kooliastme puhul tutvustatakse programmeerimist BeeBot robotite,

Daisy the Dinosaur ja BitByBit rakenduste abil. BeeBot robotit saab programmeerida selle seljal asuvate nuppudega. Võimalikud käsud on edasi-tagasi liikumine, pööramine ja hetkeks pausi tegemine. Daisy the Dinosaur on ingliskeelne rakendus, mis on mõeldud iPadile. Seal saab tegevuste klotse programmeerimise aknasse üksteise alla lohistades programmeerida dinosaurust erinevaid tegevusi tegema. BitByBit mängus tuleb linnukesi liigutada mööda mängulauda oma pesasse, kasutades selleks erinevaid liikumise käsklusi. Lisaks õpetatakse digiõpikus ka mängu loomist Kodu Game Lab abil ning tutvustatakse hariduslikku robotit Edison ja tema programmeerimiseks mõeldud Edware programmi.

Teisele kooliastmele õpetatakse digiõpikus [24] programmeerimist graafilise programmeerimiskeele Scratch abil. Käsitletavate teemade hulgas on algoritm, programm, muutuja, juhuslikkus, tsükkel, tingimuslause, plokkskeem ja testimine. Lisaks on digiõpikus teemasid, mis seovad teisi aineid programmeerimisega, näiteks muusika loomine, arvutamine suvaliste arvudega, planeetide joonistamine ja infosiltide loomine ning küsimustike koostamine. Teisele kooliastmele mõeldud materjalist ei puudu ka robootika ning siin kasutatakse mBot robotit koos mBlock tarkvaraga.

Digiõpiku uuendatud versioonides jäävad teemad üldjoontes samaks [25, 26]. I kooliastme õpikus kasutatakse erinevaid õpikusse integreeritud mänge ja ka mängu Run Marco!. Lisaks tutvustatakse natuke BeeBoti ja Scratchi. II kooliastme õpikus on põhiliseks töövahendiks jätkuvalt Scratch. Robootika osas kasutatakse LEGO Education WeDo 2.0 ja LEGO MINDSTORMS EV3.

2.1.3 Kursus „Programmeerimisest maalähedaselt“

„Programmeerimisest maalähedaselt“ on Tartu Ülikooli korraldatud MOOC, mille õppematerjalid on avalikud [27]. Erinevalt programmeerimise õpikust on selle kursuse materjalid suunatud kõigile huvilistele erinevas vanuses ning seetõttu kirjutatud lihtsamalt ja rohkete näidetega. Samuti on lisatud materjali enesekontrolliküsimusi. Teemadest käsitletakse sellel kursusel algoritme, muutujaid, andmetüüpe, valiklauseid, sisendit ja väljundit, tsükleid, regulaaravaldisi, funktsioone, andmevahetust ja kilpkonnagraafikat. Lisaks on materjalis palju silmaringiteemasid, mis tutvustavad programmeerimise ja informaatika kasutusalasid. Kursus on mõeldud vaid programmeerimist tutvustama ja ei lähe süvitsi keerulisematesse teemadesse, mistõttu võib jääda programmeerimisest huvituvate õpilaste jaoks liiga pealiskaudseks.

2.1.4 Informaatika valikkursus gümnaasiumile „Programmeerimine“

Tartu Ülikooli arvutiteaduse instituudi töörühm on koostanud materjalid gümnaasiumi programmeerimise valikkursusele [28]. Materjal sisaldab ka erinevaid enesekontrolliküsimusi, kus on nii plokkide järjestamist kui ka valikvastuseid. Sellel kursusel käsitletavad teemad on üldjoontes samad, mis programmeerimisest maalähedaselt kursusel, kuid lisatud on veel järjendite teema ning põhjalikum graafika peatükk.

TalTech pakub samuti gümnaasiumiõpilastele mõeldud programmeerimise algkursust [29]. See kursus toimub TalTechi Moodle’i keskkonnas ja kursuse materjalid ei ole kursusele registreerimata avalikud.

2.1.5 TalTechi Pythoni õppematerjalid

TalTech on koondanud kokku üliõpilastele mõeldud Pythoni õppematerjalid, mis hõlmavad laia valikut teemasid [30]. Kaetakse ära põhiteadmised, tingimuslaused, tsüklid, andmestruktuurid, funktsioonid, andmevahetus ja veebist lugemine, objektorienteeritud programmeerimine ning ka tarkvaraarendus. Mõnes peatükis on ka programmeerimisharjutused, kus saab lahenduse otse lehel olevas koodiredaktoris kirjutada ja seda käivitada. Kuigi õppematerjal ise on eestikeelne, kasutatakse koodinäidetes vaid inglise keelt.

2.1.6 Videokursus „Programmeerimine Pythonis“

Nelja Pärnu kooli ühisprojekti „Pärnu koolide digipööre oskusainetes“ [31] raames loodi videokursus põhikooliõpilastele Pythoni õppimiseks [32]. Kursus on vabalt kättesaadav YouTube’i keskkonnas ja koosneb 16 osast, mis on 10–20 min pikad. Kursusel õpetatakse kilpkonnagraafikat, muutujaid, lihtsamaid andmetüüpe, sisendi ja väljundi kasutamist, tingimuslauseid, Thonny silumistööriista, tsükleid, funktsioone, järjendeid ning luuakse teegi Pygame abil lihtne mäng. See kursus annab põhiteemadest hea ülevaate, kuid sellele lisaks võiks olla ka kirjalikud materjalid, milles on veel põhjalikumad seletused, rohkem näiteid ja ka enesekontrolliküsimused. Tartu Ülikooli korraldatud „Programmeerimisest maalähedaselt“ MOOC-is kasutati samuti videoid ning osalejate tagasisidest selgus, et videod üksi ei ole igale õppijale piisavad, vaid heaks lisavahendiks tekstilise materjali kõrval [33].

2.2 Ingliskeelsed õppematerjalid

Ingliskeelseid õppematerjale programmeerimise õppimiseks on palju. YouTube'is on erinevaid videokursusi, nutiseadmetele on erinevaid algoritmilist mõtlemist arendavaid mänge ning on ka erinevaid programmeerimist õpetavaid keskkondi, millest tuntuimad on Code.org, Codecademy ja Codelearn.io.

2.2.1 Code.org

Code.org on üks tuntumaid ingliskeelseid veebilehekülgi programmeerimise õppimiseks lastele [34]. Tegemist on mittetulundusliku projektiga, mille eesmärk on muuta informaatika õppimine kõigile noortele kättesaadavaks. Lehekülg on suunatud igas vanuses lastele ning seal on õppematerjale ka lasteaia tasemele. Programmeerimise algkursusel, mis on suunatud vanustele 9–18, räägitakse tegevuste järjestusest, *sprite*'idest (kahemõõtmelised kujutised), sündmustest, tsüklitest, tingimuslausetest, funktsioonidest ja muutujatest. Õpetamiseks kasutatakse piltprogrammeerimist. Lisaks algkursusele on leheküljel veel erinevaid mooduleid, näiteks veebimoodul õpetab HTML-i (ingl *HyperText Markup Language*) ja CSS-i (ingl *Cascading Style Sheets*) abil lihtsat koduleheküle looma ning rakenduste moodul õpetab JavaScripti kasutades kilpkonnagraafikat. Samuti on õpilastele kättesaadavad erinevad videod, mis selgitavad üldisemaid informaatikaga seotud teemasid, näiteks kuidas arvutid, internet ning tehisintellekt töötavad. Selle keskkonna õppematerjalid on tasuta kättesaadavad, kuid lehele on vaja luua konto.

2.2.2 Codecademy

Codecademy lehel on palju õppematerjale nii programmeerimiskeele kui valdkonna kaupa [35]. Õppida saab 15 erinevat programmeerimiskeelt ning valdkondi on alates veebi- ja mobiilirakenduste loomisest kuni masinõppe, andmeteaduse ja küberturvalisuseni. Õppematerjalid on suunatud pigem vanematele õpilastele ja täiskasvanutele ning paljud lehel leiduvatest kursustest on kättesaadavad vaid lisatasu eest. Pythoni algkursus on Codecademy leheküljel tasuta ning annab ülevaate andmetüüpidest, tingimuslausetest, funktsioonidest, järjenditest ja sõnastikest ning tsüklitest. Igas õppepeatükis on veebileht jaotatud visuaalselt kolmeks vertikaalseks plokiks: vasakul on lühike tööjuhend, keskel koodiredaktor ning paremal konsool. Et liikuda järgmise peatüki juurde, tuleb ülesanne edukalt lahendada.

2.2.3 Codelearn.io

Codelearn.io lehel on sarnaselt Codecademy lehele palju erinevaid kursusi, mis õpetavad programmeerimist erinevates keeltes, nagu Python, Java, C, C++, SQL ja JavaScript, kuid ka võrgutehnoloogiat, algoritme ning pilveteenuseid [36]. Pythoni kursuse ülesehitus on väga sarnane Codecademy lehele, kus vasakul ekraanipoolel on juhend ning paremal koodiredaktor ja konsool. Ka kaetud teemade nimekiri on sama, mis Codecademy kursusel. Codelearn.io lehel on kursused tasuta, kuid kursuste nägemiseks ning läbimiseks tuleb endale kasutaja teha ning huvipakkuvale kursusele registreerida.

Saadaval on palju erinevaid õppematerjale, kuid eestikeelsetest materjalidest on puudus põhjaliku programmeerimisõpiku järele, mis oleks piisavalt lihtne põhikooli õpilastele, kataks ära kõik tähtsamad programmeerimisteemad ja tutvustaks ka keerulisemaid teemasid nagu andmestruktuurid ja objektorienteeritud programmeerimine. Järgmises peatükis uuritakse kvaliteetse digitaalse õppematerjali loomist, et luua vajadustele vastav nõuetekohane õppematerjal.

3. Digitaalse õppematerjali loomine

Hariduse Infotehnoloogia Sihtasutuse (HITSA) eestvedamisel koostati 2015. aastal digitaalse õppematerjali loomise soovitused [37]. Digitaalse õppematerjali all mõeldakse digitaalselt kasutatavaid ja levitatavaid õppeotstarbelisi materjale, mis sisaldavad teksti, graafikat ja multimeediat. Kvaliteetne õppematerjal peab olema õppimist toetav, kvaliteetne, motiveeriv, kohandatav, interaktiivne, autoriõigusi järgiv, kasutajasõbralik, tehniliselt korrektne, ühilduv ja leitav.

Õppimise toetamiseks peavad õppematerjalil olema kindlad õpieesmärgid ja -tulemused, mis sõnastatakse õppijakeskselt, ning materjal peab nendele vastama [37]. Et luua motiveeriv ja eakohane õppematerjal, tuleb silmas pidada sihtrühma ning arvestada õppija eelteadmiste, oskuste ja võimalustega. Õppematerjalis esitatavad väited ja andmed peavad olema korrektsed ning õppesisu ja -tegevuste mahu vahel tuleb leida tasakaal, et materjal oleks ülevaatlik, kuid ei hajutaks tähelepanu. Digitaalses õppematerjalis kasutatavad interaktiivsed tegevused aitavad õppijal materjali paremini omandada, kuid interaktiivseid elemente luues tuleb silmas pidada, et need oleksid kasutajasõbralikud ja üheselt mõistetava tagasisidega. Kvaliteetse õppematerjali juures on oluline ka vormistus, et materjal oleks kergesti loetav, liigendatud ning visuaalselt haarav. Tähtis on ka jälgida, et digitaalne õppematerjal oleks töökorras ning kättesaadav erinevates seadmetes ja veebilehitsejates.

3.1 Mudelid

Õppematerjali väljatöötamiseks saab kasutada mudeleid, millest üks levinumaid on ADDIE (ingl *analyse, design, development, implementation, evaluation*) [37]. ADDIE mudel koosneb analüüsi, kavandamise, väljatöötamise, kasutamise ja hinnangu andmise etappidest. Mudeleid saab kasutada ka õppematerjali testimiseks ja sellele hinnangu andmiseks.

ADDIE mudeli analüüsi etapis kaardistatakse õpiväljundid ja eesmärgid ning võrreldakse neid õpilase teadmiste ja oskustega [38]. Kavandamise etapis pannakse paika kasutatavad vahendid, meetodid ja tegevuskava. Kolmas etapp keskendub materjalide loomisele, testimisele ning parandamisele. Peale seda etappi on materjalid avalikustamiseks ja kasutamiseks valmis. Nii

materjalide väljatöötamise kui kasutamise ajal hinnatakse nende vastavust seatud õpiväljunditele ja vajadusel tehakse korrektuure.

Lisaks õppematerjalide väljatöötamisele on võimalik erinevaid mudeleid kasutada ka õppematerjali ülesehituse jaoks. Üheks võimalikuks mudeliks on SCATE (ingl *scope, content, activity, thinking, extra*), mille viis osa on sissejuhatus, õppematerjalid, kinnistamine, arutlemine ja lisamaterjalid [37]. Sissejuhatuse osas antakse ülevaade õpiväljunditest ja käsitletavatest teemadest [39]. Õppematerjalide osas antakse edasi sisu ning sellele järgneb kordamisküsimuste ja enesekontrollidega kinnitamise osa ning aruteluküsimustega arutlemise osa. Viimase sammuna antakse viiteid lisamaterjalile.

Valminud õppematerjali testimine ning sellele hinnangu andmine on oluline osa õppematerjali väljatöötamisest, et tagada kvaliteet [37]. Esimese sammuna saab materjali koostaja ära teha digitaalse õppematerjali tehnilise testimise, mille käigus kontrollitakse, kas kõik materjali osad toimivad levinumate veebilehitsejatega ja erinevate seadmetega. Teine samm on sisuline testimine, millega saavad aidata kolleegid ning mille käigus kontrollitakse, kas õppematerjal täidab oma eesmärke. Selles etapis on sobilik kasutada mõnda kvaliteedi hindamismudelit, nagu LORI (ingl *Learning Object Review Instrument*) [40, 41].

LORI mudelis on 9 komponenti, mida hinnatakse 5-palli skaalal [41]. Hinnatakse sisu kvaliteeti, vastavust õpieesmärkidega, õpikus antavat tagasisidet ja sisu kohandumist, õpilaste motivatsiooni materjali kasutada, esitlust ja kujundust, kasutatavust, ligipääsetavust, taaskasutatavust ja standarditele vastavust. Täpne nimekiri komponentide ja selgitustega on toodud lisas 1.

Kui tehniline ja sisuline testimine on tehtud, on viimase sammuna kasulik viia läbi eakohasuse ja arusaadavuse testimine, milleks kaasatakse sihtrühm [37]. Õppijatelt tagasiside saamiseks sobib kasutada LOES-S (ingl *Learning Object Evaluation Scale For Students*) hindamismudelit, millega hinnatakse kolme kategooriat: õppimine, kvaliteet ja kaasahaaravus [42]. Kokku on mudelis 12 küsimust, mida hinnatakse 5-astmesel Likerti skaalal ning lisaks kaks vabas vormis küsimust, kus õpilastel palutakse kirjutada, mis neile meeldis ja mis ei meeldinud õppematerjali juures. Nimekiri küsimustest on toodud lisas 1.

4. Veebirakenduse loomine programmeerimise õppimiseks

Eelnevalt sai kindlaks tehtud vajadus uue õppematerjali järele ja uuritud kvaliteetse õppematerjali loomist. Tuginedes ADDIE mudelile loodi eestikeelne õppematerjal, mis on suunatud teisele ja kolmandale kooliastmele ning õpetab piltprogrammeerimise asemel tekstilist programmeerimist.

4.1 Analüüsimine

ADDIE mudeli esimese etapi eesmärgiks on kaardistada õppematerjali õpiväljundid ja õppija oskused ja teadmised. Selles etapis toimus ka olemasolevate õppematerjalide analüüs (peatükk 2) ning uue õppematerjali vajaduse kindlakstegemine (peatükk 1).

Põhikooli riiklik õppekava [20] näeb II ja III kooliastmes pädevustena ette arvuti ja interneti kasutamist suhtlus- ja info otsimise vahendina ning oskust vormindada arvutis tekste. Programmeerimise õpetamist põhikoolis riiklik õppekava ette ei näe ja tihtipeale seda ka ei õpetata [5, 21]. Seega võib eeldada, et programmeerimise osas keskmisel põhikooliõpilasel teadmised puuduvad ja programmeerimise õpetamist tuleb alustada põhitõdedest. Küll aga võib eeldada erinevate matemaatiliste teadmiste olemasolu, nagu erinevate tehete tundmine, ümardamine, murdudega arvutamine, protsendi leidmine, ühikute teisendamine, kujundite tundmine ja pindalade arvutamine [43].

Programmeerimisõpiku õpiväljundite seadmisel toetuti olemasolevatele õppematerjalidele (peatükk 2). Õpiku läbi töötanud õpilane peaks omama teadmisi programmeerimiskeelte, algoritmide, andmetüüpide, muutujate, tingimus- ja korduslausete, funktsioonide, failide, veahalduse, erinevate andmestruktuuride ja objektorienteeritud programmeerimise kohta. Täielik õpiväljundite loetelu on toodud lisas 2.

4.2 Kavandamine

Kavandamise etapis määrati kindlaks õppematerjali peatükid, peatükkide struktuur ja kujundus. Loodav õppematerjal on mõeldud teise ja kolmanda kooliastme õpilastele iseseisvaks õppimiseks, seega on oluline, et materjal oleks interaktiivne, annaks õpilastele automaatselt tagasisidet, oleks lihtsa sõnastusega ja visuaalselt atraktiivne.

4.2.1 Peatükid

Õpiväljundite põhjal koostati nimekiri õpiku peatükkidest, mis jagavad õpetatavad teemad väiksematesse osadesse, ja pandi need loogilisse järjekorda. Õpik jagati neljaks kategooriaks koos peatükkidega. Esimesed kolm kategooriat hõlmavad programmeerimise alusteadmisi, andmestruktuure ja objektorienteeritud programmeerimise põhitõdesid. Neljas kategooria on projektid. Peatükkides navigeerimise lihtsustamiseks ja parema ülevaate andmiseks liigendati peatükid veel alapeatükkideks.

Esimene kategooria on põhiteadmised, milles alustatakse arvutite ja programmeerimiskeelte tutvustusega, räägitakse algoritmide koostamisest, muutujate ja avaldiste loomisest, tingimus- ja korduslausetest, funktsioonidest, failidest lugemisest ja kirjutamisest ning lõpuks ka veahaldusest. Teine teema on andmestruktuuridest ja seal antakse ülevaade neljast põhilisest andmestruktuurist: järjendid, sõnastikud, hulgad ja ennikud. Lisaks räägitakse mitmemõõtmeliste andmestruktuuride koostamisest ja kasutamisest. Kolmas kategooria on objektorienteeritud programmeerimine, kus on kaks alapeatükki: klassid ja isendid ning pärilus. Viimane teema keskendub kõige õpitu rakendamisele projektides ja seal on peatükid mängu, veebirakenduse ja töölauarakenduse loomise kohta. Täielik õpiku sisukord on toodud lisas 3.

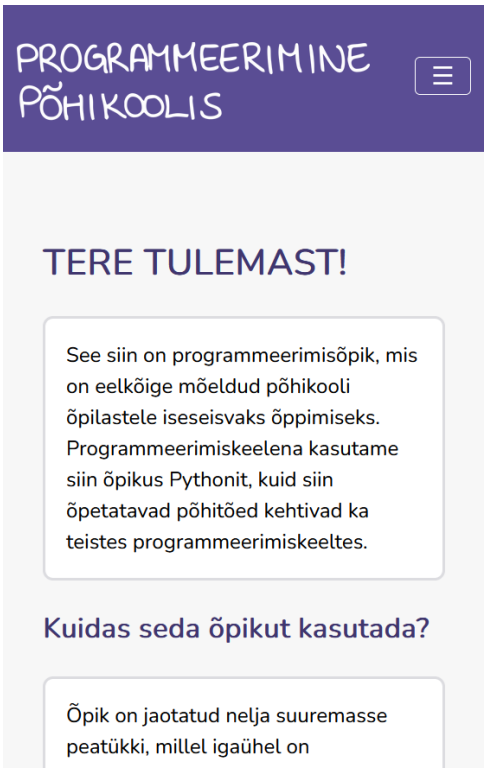
Otsustati ka üldine peatüki struktuur. SCATE mudeli põhjal võiks õppematerjal koosneda sissejuhatusest, õppematerjalidest, kinnistamisest, arutlemisest ja lisamaterjalidest [39]. Magistritöö raames plaaniti luua igale peatükile sissejuhatuse, õppematerjalide ja kinnistamise osa. Arutlemisküsimuste ja lisamaterjalide osa otsustati jätta hilisemaks tööks. Iga peatükk algab õpiväljunditega, mis annavad täpse ülevaate, mis teadmisi selle peatüki raames omandatakse. Sellele järgneb lühike sissejuhatav osa ja siis juba sisuline õppematerjali osa. Materjali kinnistamiseks loodi enesekontrolliküsimused, mis aitavad õpilastel kontrollida oma arusaamist materjalist [16, 17]. Enesekontrolliküsimused otsustati põimida õppematerjaliga, et õpilane saaks juba materjali läbitöötamise ajal interaktiivselt tagasisidet. Algoritmilise mõtlemise treenimiseks lisati programmeerimisharjutuste juurde vihjed, mis lasevad õpilasel samm-sammult näha ülesande lahendamise protsessi [15].

4.2.2 Kujundus

Rakenduse kujunduse loomisel oli oluline silmas pidada, et see oleks kohanduv nii eri suuruses arvutiekraanidele kui ka tahvlitele ja nutitelefonidele. Laiema ekraaniga vaates otsustas autor luua menüü lehe vasakusse äärde, et kasutajal oleks alati teemade valik silme ees ja ta saaks ka menüüs olevate linkide abil kiiremini rakenduses ringi liikuda. Väiksema ekraaniga pakitakse menüü kokku ja liigutatakse ekraani ülemisse äärde, kus on näha vaid õpiku pealkiri ja kolme kriipsuga nn hamburgeri nuppu, millele vajutades avaneb täispikk menüü. Ka kõik õpiku sisulised komponendid ja visuaalid peavad muutma oma suurust vastavalt ekraani suurusele, et igal hetkel oleks sisu hästi paigutatud ja loetav.

Värvivalik on veebirakenduse kujundamisel oluline [44]. Värvirikkad erksad toonid tõmbavad tähelepanu ja sobivad hästi linkide ja nuppude jaoks. Menüü jaoks sobib kasutada natuke tuhmimat tumedat värvi, mis jätab professionaalsema mulje ja suunab kasutajaid liikuma. Parimad põhivärvid veebilehtedele on sinakad [45]. Veebirakenduse põhivärvi valimisel võeti aluseks Kuo jt uuringus [45] teisele kohale jäänud sinakas-lillakas toon #6A5ACD. Seda muudeti veel natuke tumedamaks ja tuhmimaks, et saavutada parem kontrast menüü teksti valge värviga.

Joonis 1. Veebirakenduse avaleht töölauavaates



Joonis 2. Veebirakenduse avaleht mobiilivaates

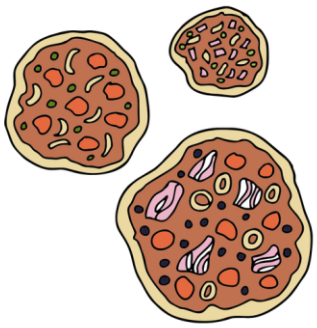
Vähene kontrastsus väsitab silmi ja võib tekitada peavalu [46], seetõttu on oluline valida veebilehele taust ja tekstivärv, mida oleks lihtne lugeda. Taustavärviks on kindel valik valge [44], mis valiti ka õppematerjalis põhiliseks taustavärviks. Tekst jäeti musta värvi, sest seda on heledalt taustalt lihtne lugeda. Kuna linkide jaoks soovitati värvilisi erksaid toone [44], siis valiti veebirakenduses linkidele roosakas värv, mis paistab teksti seest hästi silma. Joonisel 1 on veebirakenduse avalehe näide koos menüüga töölauavaates ja joonisel 2 mobiilivaates.

Tähtsamad veebilehe osad peavad kasutajale silma paistma [46]. Seetõttu otsustati võtta kasutusele eraldi värvid koodiplokkide, enesekontrolliküsimuste, ülesannete vihjete ja õpiväljundite jaoks. Koodiplokid peavad eristuma tavatekstist, kuid ei pea olema nii silmapaistvad kui näiteks enesekontrolliküsimused, seetõttu valiti neile helehall taust (joonis 3). Ülesannete vihjetele valiti helelilla taust, mis sobitus põhivärviks valitud sinakas-lilla värviga (joonis 4). Enesekontrolliküsimustele ja iga peatüki alguses olevatele õpiväljunditele valiti helesinine taust, mis tõmbab tähelepanu, kuid samas on piisavalt hele, et mitte häirida silmi ja teksti lugemist (joonised 5 ja 6).

Täiendame oma programmi nüüd nii, et laseme funktsioonil arvutada kõigi kolme pitsasuuruse jaoks vajalikud taignakogused, salvestame funktsiooni tagastatud väärtused muutujatesse ning hiljem liidame need kokku, et anda kasutajale teada, palju tal kolme eri suuruses pitsa jaoks tainast kokku läheb.

```
def pitsatainas(läbimõõt):
    pindala = (läbimõõt / 2)**2 * 3.14
    taigna_kogus = round(pindala * 2.3, 1)
    return  taigna_kogus

kogus_20 = pitsatainas(20)
kogus_28 = pitsatainas(28)
kogus_38 = pitsatainas(38)
kokku = kogus_20 + kogus_28 + kogus_38
print("Kokku kulub 20 cm, 28 cm ja 38 cm pitsa peale " + str(kokku) + " g tainast.")
```



Joonis 3. Koodiplokk

Nüüd saame uut argumenti funktsiooni sees kasutada. Selleks, et ühe pitsa taignakoguse asemel tagastada etteantud arvu pitsade jaoks vajalik taignakogus, tuleb üksiku pitsa kogus korrutada argumendiks antud pitsade arvuga.

← Eelmine vihje → Järgmine vihje

Joonis 4. Ülesannete vihjed

Mida selles peatükis õpime?

1. Mis on funktsioonid ja milleks neid kasutada?
2. Kuidas funktsioone kirjutada ja kasutada?
3. Kuidas funktsioonist väärtust tagastada?

Joonis 5. Peatüki õpiväljundid

Enesekontroll. Mida väljastab järgmine koodijupp?

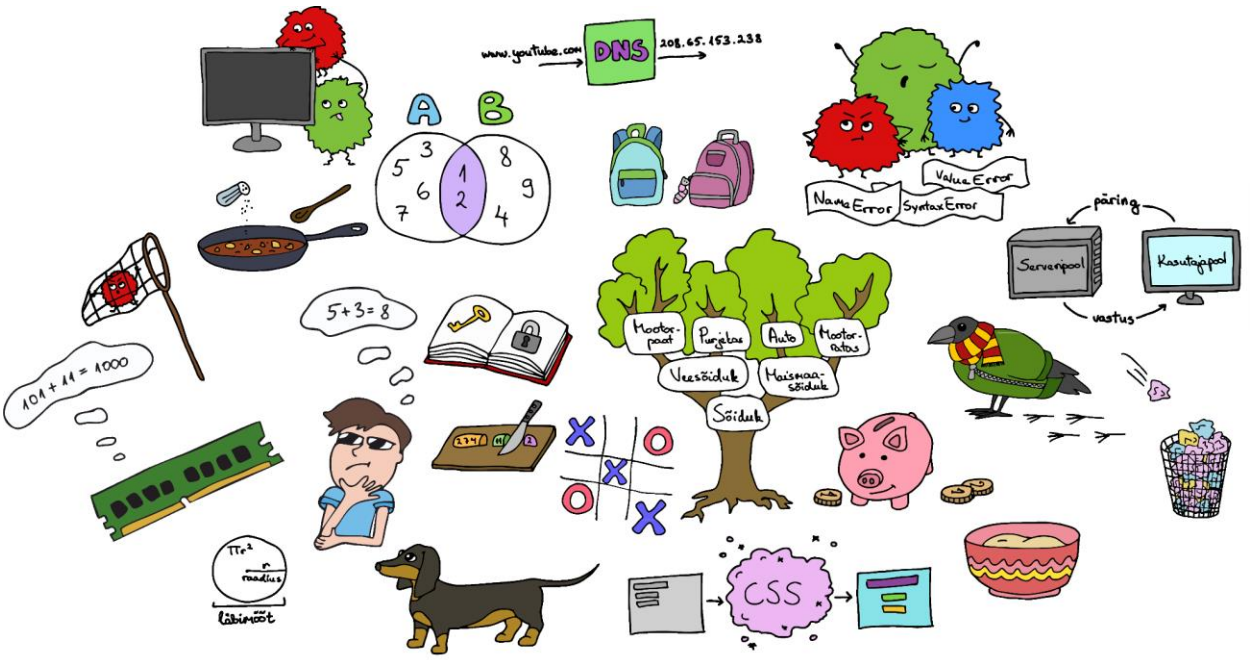
```
def lahuta(esimene, teine):
    print(teine - esimene)

lahuta(3, 7)
```

- ○ 3
- ○ -4
- ○ Mitte midagi
- ○ 10
- ○ 4

Kontrolli

Joonis 6. Enesekontrolliküsimus



Joonis 7. Illustratsioonid

Kõikidele õpiku sisupeatükkidele lisati ka autori joonistatud illustratsioonid, millest mõned näited on toodud joonisel 7. Illustratsioonid aitavad mõnda keerulisemat ideed seletada ning muudavad õpiku atraktiivsemaks ja lõbusamaks.

4.3 Väljatöötamine

Õpiku väljatöötamine oli kõige mahukam etapp. Iga õpiku peatükk sisaldab selgitavat teksti, näiteid, enesekontrolliteste ja harjutusi, mis on kõik loodud pidades silmas teist ja kolmandat kooliastet, nende teadmisi ja neile huvipakkuvaid teemasid.

4.3.1 Tehnoloogiad ja vahendid

Autor otsustas veebirakenduse koodi ise kirjutada, et võimaldada paindlikkust interaktiivsete komponentide osas. Tuginedes oma kogemustele ja oskustele, otsustas autor veebirakenduste raamistikku Angular [47] kasuks, mis võimaldab luua dünaamilisi üheleheküljelisi veebilehti. Kasutusele võeti rakenduse alustamise hetkel kõige uuem saadaolev Angulari versioon 14. Kuna õppematerjaliks mõeldud veebirakendus on oma arhitektuurilt lihtne ning ei nõua andmete

töötlemist ega hoiustamist, otsustati piirduda vaid Angulariga loodud kasutajapoolse (ingl *front-end*) rakendusega ning mitte kasutada andmebaasi ega serveripoolset (ingl *back-end*) rakendust.

Lisaks otsustati veebirakenduse lihtsamaks kujundamiseks võtta kasutusele Angularile mõeldud Bootstrapi tööriistakomplekt ng-bootstrap [48], mis võimaldab läbi CSS-klasside ja komponentide lihtsa vaevaga kohalduvaid kujundusi luua. Ikoonide jaoks lisati teek angular-fontawesome [49] ja koodinäidete kujundamiseks teek ngx-highlightjs [50].

Versioonihalduseks võeti kasutusele versioonihaldussüsteem Git ja koodi hoiustamiseks loodi koodihoidla autori GitHubi lehel, mis on ka avalikult leitav [51]. Veebirakenduse juurutamiseks (ingl *deploy*) ja hoiustamiseks võeti kasutusele Heroku platvorm [52], mis teeb rakenduse juurutamise protsessi lihtsaks. Heroku konto saab siduda koodihoidlaga ja seadistada iga uuenduse peale automaatselt uut versiooni ehitama ja juurutama. Vastavalt projekti suurusele ja vajadustele pakub Heroku erineva hinnaga teenuseid. Kuna tegemist on väiksemahulise ja tehnoloogiliselt lihtsa rakendusega, siis sobis programmeerimisõpiku jaoks pakett Basic.

Heroku loob igale rakendusele ka domeeninime lõpuga *herokuapp.com*. Selle domeeni kaudu on rakendus küll igalt poolt kasutajatele kättesaadav, kuid domeeninimi pole meeldejääv. Lihtsama ja tabavama domeeninime jaoks registreeris autor uue domeeni nimega *progema.ee*. Domeeni registreerimiseks kasutati Veebimajutus.ee teenust [53].

Heroku võimaldab läbi CNAME või ALIAS domeeninimede süsteemi (DNS) kirjete suunata ka teisi domeene Heroku rakenduste vastu. Klassikaline lahendus juurdomeeni suunamiseks on kasutada A-kirjeid, millele tuleb anda sihtpunktiks staatiline IP-aadress. Pilvelahenduste puhul, nagu Heroku, pole aga staatilised IP-d võimalikud. Veebimajutus.ee DNS ei lubanud aga dünaamilisi CNAME või ALIAS kirjeid juurdomeenile, mistõttu tuli vahetada ka domeeninimeserverit. Autor valis uueks teenusepakkujaks DNSimple'i [54], mis võimaldas juurdomeeni Heroku vastu suunamist läbi ALIAS-kirje.

Lisaks veebirakendusega seotud tehnoloogiatele oli vaja õpilaste huvi suurendamiseks ja õpiku atraktiivsemaks muutmiseks lisada õpikusse ka illustratsioonid. Algsete illustratsioonide loomiseks kasutas autor tahvelarvutit ja hiljem vektorgraafikaprogrammi CorelDRAW [55].

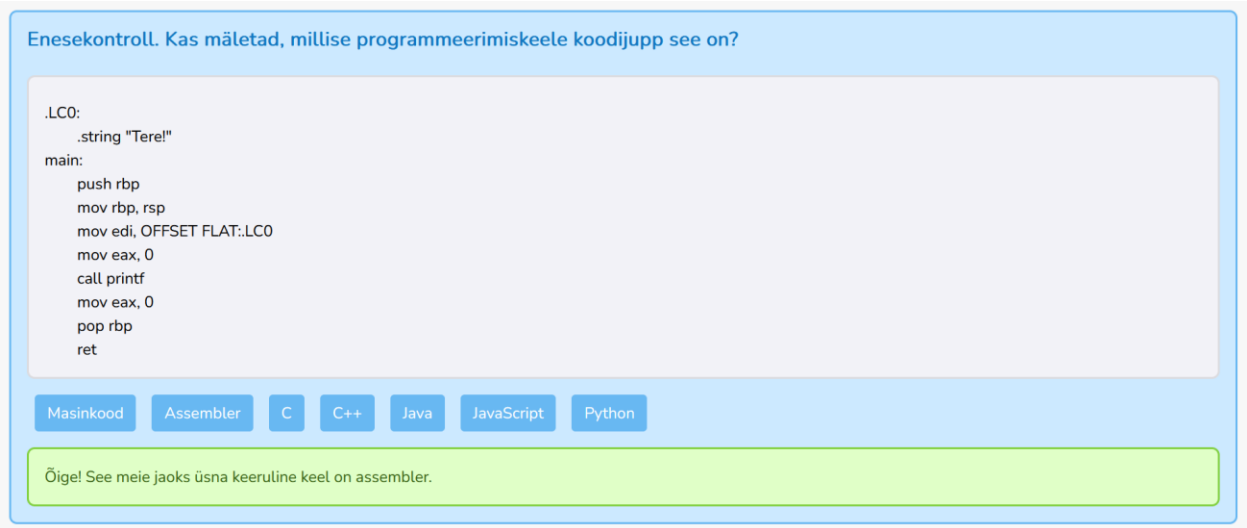
Angulari rakendused koosnevad erinevatest väiksematest alamüksustest ehk komponentidest, mis kirjeldavad ära mingi kasutajaliidese osa kujunduse ja käitumise. Loodud veebirakenduses on iga õpiku peatükk omaette komponent koos aadressiga. Peatükile vastavale aadressile liikudes kuvatakse õpiku sisuosas vastav peatükk.

Ka menüü on eraldi komponent, mis luuakse veebirakenduse avamise hetkel ja mis jääb püsima ka erinevate lehekülgede vahel liikudes. Samuti on erinevatesse komponentidesse paigutatud kõik korduvkasutatavad rakenduse osad, näiteks enesekontrollitestid ja ülesannete vihjed.

4.3.2 Enesekontrollid

Enesekontrolliteste loodi kuut erinevat sorti. Nende hulgas olid valikvastustega testid, nii ühe kui mitme õige vastusega, lünka sobiva variandi valimine, kategoriseerimine ja järjestamine.

Esimene valikvastustega testidest on nuppudega test. Selles testis on valikuvariandid kuvatud nuppudena ja tagasisidet kuvatakse kohe mõnele nupule vajutades (joonis 8).

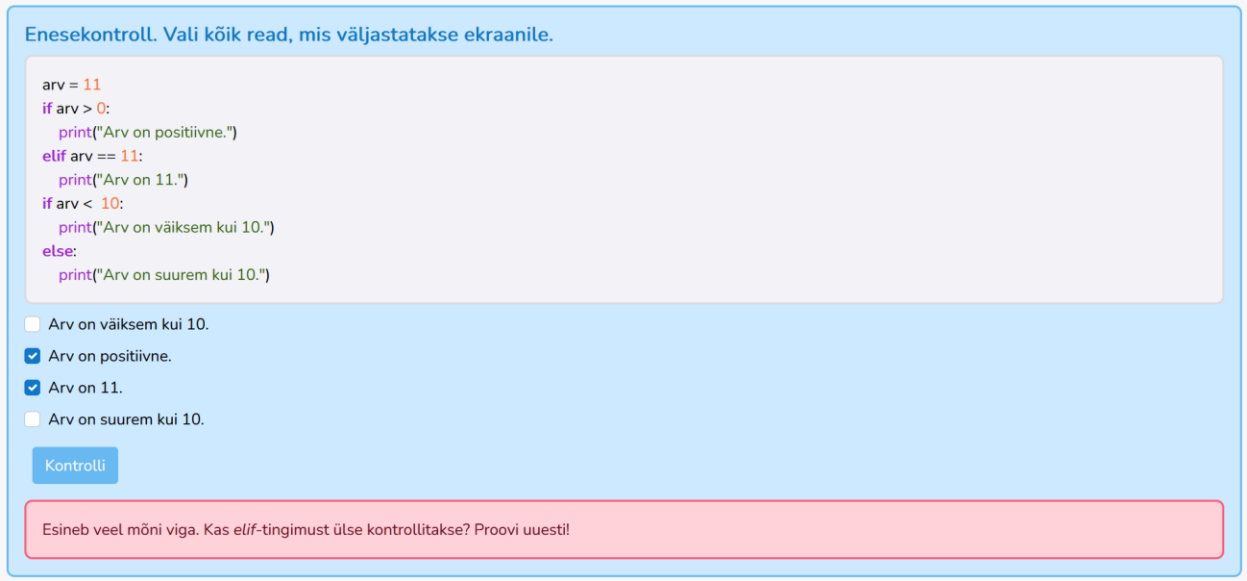


Joonis 8. Nuppudega enesekontrollitest

Nuppudega testile sarnane on ka raadionuppudega test, kus saab samuti valida vaid ühe vastusevariandi. Selle testi puhul ei kuvata tagasisidet koheselt, vaid alles nuppu „Kontrolli“ vajutades. Sarnaselt nuppudega testile antakse iga vale vastusevariandi korral vihje, mis võiks õpilast suunata õige vastuse poole (joonis 9).

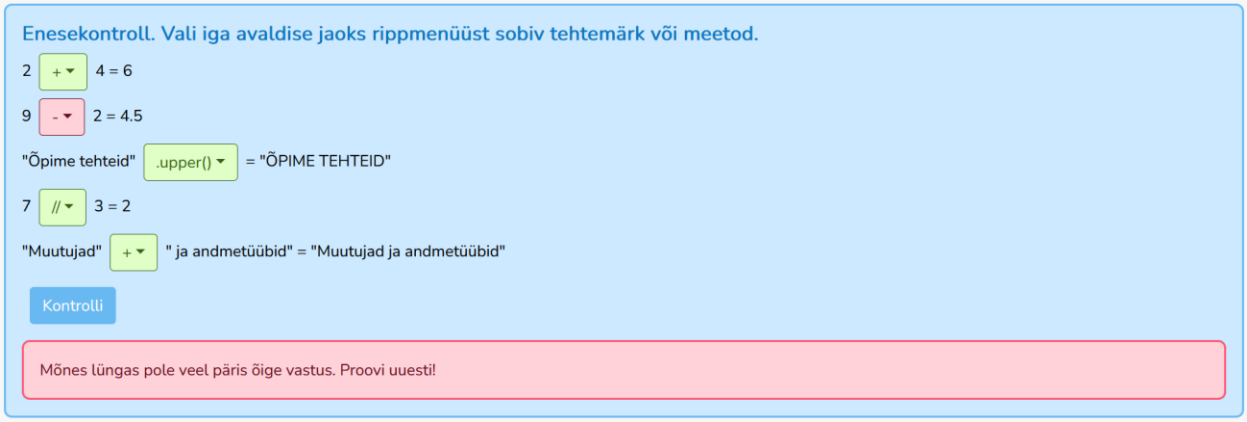
Joonis 9. Raadionuppudega enesekontrollitest

Märkeruutudega testis saab valida mitu vastusevarianti. Sarnaselt eelmistele testidele, kuvatakse vihjeid, kui mõni märkeruut on valesti märgitud (joonis 10).



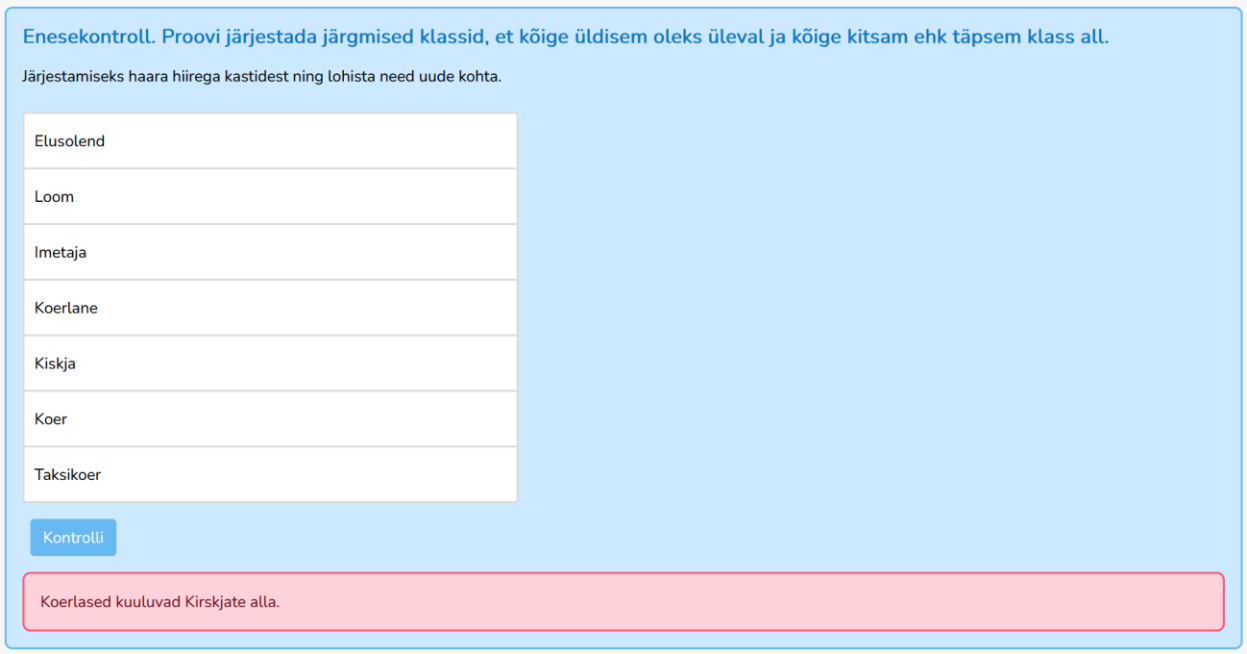
Joonis 10. Märkeruutudega enesekontrollitest

Lünka sobiva variandi valimise testis saab õpilane rippmenüüst erinevate vastusevariantide vahel valida. Alguses on kõik lüngad valge taustaga, nuppu „Kontrolli“ vajutades värvuvad lüngad kas roheliseks või punaseks vastavalt sellele, kas valik oli õige või mitte. Seejärel saab lünkades valikut muuta ja uuesti kontrollida (joonis 11).



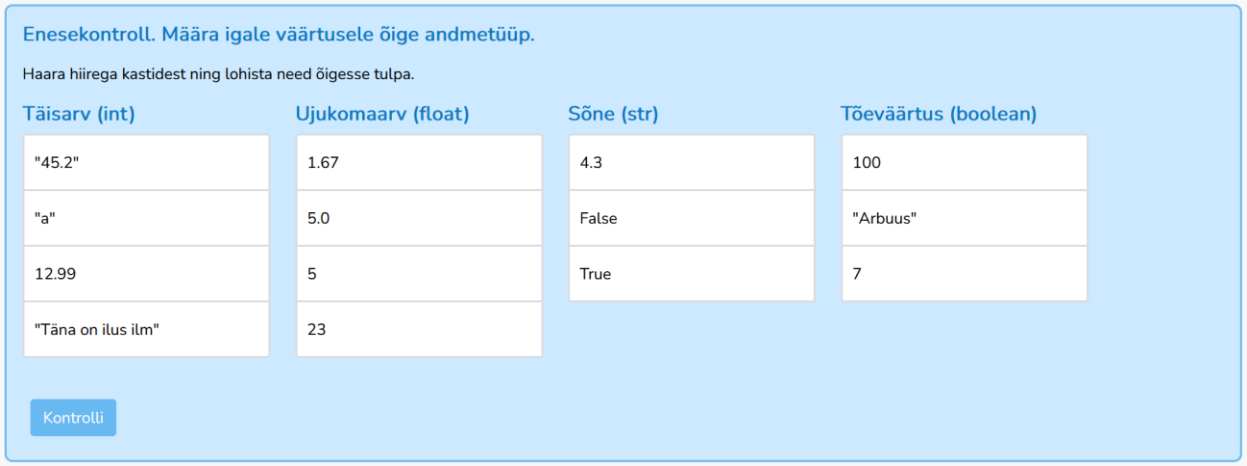
Joonis 11. Lünkadega enesekontrollitest

Viies testiliik on järjestamine. Selles testis saavad õpilased kaste lohistades nad õigesse järjekorda viia ja oma tulemust kontrollida (joonis 12).



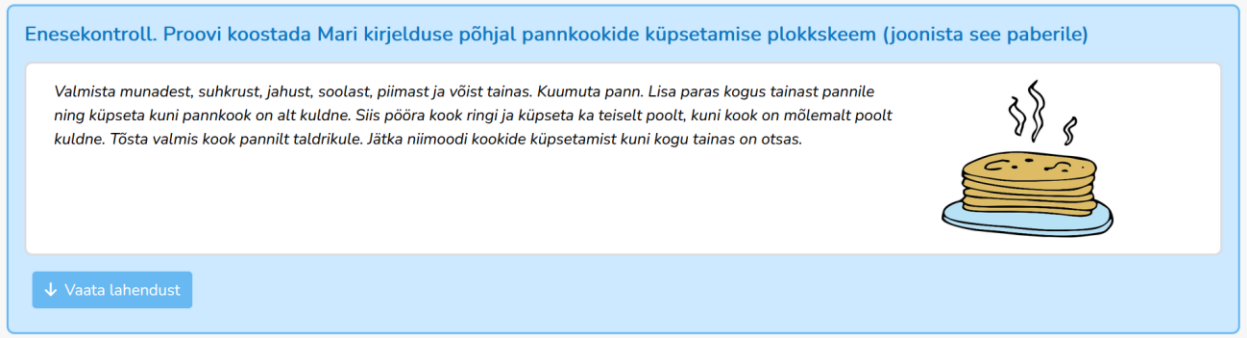
Joonis 12. Järjestamisega enesekontrollitest

Sarnasel lohistamisel põhineb ka kategoriseerimise test. Selles testis on antud mitu tulpa erinevate kategooriate jaoks ja kõik kastid tuleb õigesse tulpa viia. Alguses on kastid kategooriatesse juhuslikult jagatud (joonis 13).



Joonis 13. Kategoriseerimisega enesekontrollitest

Lisaks eeltoodud kuuele testile on õpikus kasutusel ka paberile joonistamise test, mida küll automaatselt kontrollida ei saa, kuid kus õpilane saab soovi korral endale näidislahendust kuvada Selliseid teste kasutati peamiselt algoritmide peatükis ja näide ühest sellisest testis on toodud joonisel 14.

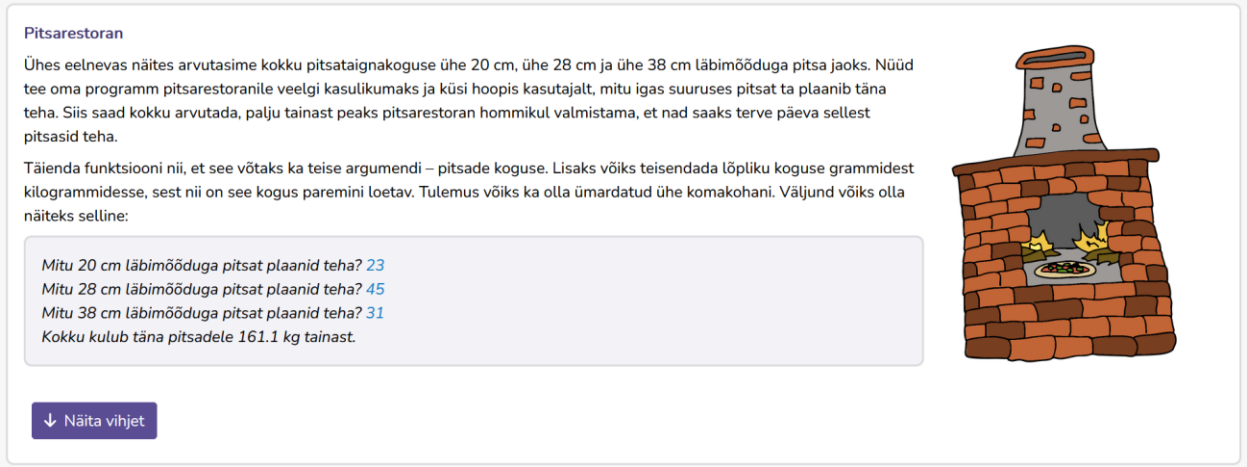


Joonis 14. Paberile joonistamisega enesekontrollitest

Enesekontrollitestid aitavad õpilastel uutest kontseptsioonidest ja süntaksist aru saada. Vale vastuse korral antavad vihjed ei anna kohe õiget vastust ette, vaid suunavad õpilast edasi mõtlema ja annavad märku, miks valitud vastus ei sobi.

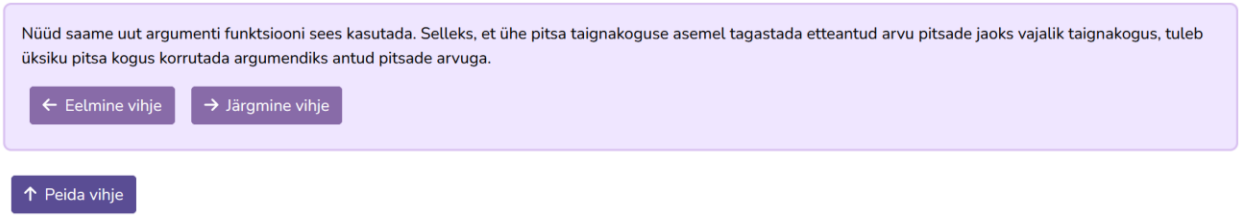
4.3.3 Ülesannete vihjed

Alates kolmandast peatükist on õpilastele lisatud ka programmeerimisülesanded. Programmeerimisülesanne koosneb ülesande kirjeldusest, väljundi näidisest ja lisaks ka vihjetest. Vaikimisi on vihjed peidetud ja näha on vaid nupp „Näita vihjet“ (joonis 15).

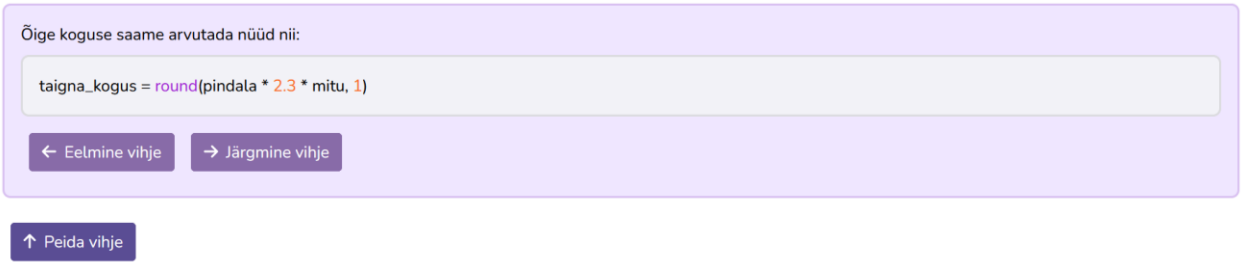


Joonis 15. Ülesanne koos „Näita vihjet“ nupuga

Nupule vajutades avatakse vihjed ja „Näita vihjet“ nupp muutub „Peida vihje“ nupuks. Nuppudega saab vihjete vahel ka edasi ja tagasi liikuda. Vihjetes hakatakse andma ideid ja näiteid ülesande lahendamiseks. Alustatakse täitsa algusest ja lõpetatakse tervikliku näidislahendusega. Vihje võib olla lihtsalt sõnaline (joonis 16) või sisaldada ka koodijuppi (joonis 17). Kõigepealt üritatakse sõnaliselt selgitada, mis peaks olema järgmine samm, ja alles seejärel kuvatakse koodinäide. See annab õpilasele võimaluse proovida kõigepealt ise vastav koodijupp kirja panna, nägemata kohe lahendust.



Joonis 16. Avatud tekstiline vihje



Joonis 17. Avatud vihje koos koodinäitega

Vihjete abil saavad õpilased samm-sammult ülesande lahendamisele kaasa mõelda ja iga sammu kohta selgitusi lugeda. Lõpus leiduv näidislahendus on kasulik ka neile, kes saavad ülesande iseseisvalt lahendatud, kuid soovivad seda võrrelda mõne teise lahendusega. Samuti võib sealt saada ideid oma koodi parendamiseks.

4.4 Kasutamine ja hinnangu andmine

ADDIE mudeli põhjal on õppematerjali väljatöötamise viimased etapid kasutamine ja hinnangu andmine. Kasutamise etapis tegi autor esimese sammuna ära õppematerjali tehnilise testimise, katsetades rakenduse toimimist erinevate brauserite ja seadmetega.

Brauseritest kasutati testimiseks Mozilla Firefoxi (versioon 112.0), Google Chrome'i (versioon 112.0.5615.121), Microsoft Edge'i (versioon 112.0.1722.48) ja Safarit (versioon 15.5). Füüsilistest seadmetest kasutati kahte sülearvutit, kahte nutitelefoni ja kahte tahvelarvutit. Sülearvutiteks olid Lenovo IdeaPad 5 Pro Windows 11 operatsioonisüsteemiga ja Apple MacBook Pro macOS Monterey versioon 12.4 operatsioonisüsteemiga. Nutitelefonidest kasutati Samsung Galaxy A52s operatsioonisüsteemiga Android 13 ja iPhone 13 Pro Max operatsioonisüsteemiga iOS 16.4.1 ning tahvelarvutitest Samsung Galaxy Tab S6 Lite operatsioonisüsteemiga Android 12 ja Apple iPad operatsioonisüsteemiga iOS 16.3.1.

Sisulise testimise jaoks küsiti tagasisidet põhikooli õpetajatelt, kes annavad informaatika ainet või juhendavad programmeerimisringe. Õpetajatelt tagasiside küsimiseks võeti põhjaks LORI mudel [41], mida kohandati õpiku jaoks. Näiteks jäeti välja küsimused ligipääsetavuse ja standarditele vastavuse jaoks. Ligipääsetavust sai autor juba ise testida ja kuna tegemist on veebirakendusega, siis on materjal kättesaadav kõigile, kellel on ligipääs internetile ja veebilehitsejale. Standarditest kontrollis autor veebirakenduse vastavust erinevatele W3C standarditele. Näiteks kasutati W3C HTML-i valideerijat [56] ning WCAG 2.1 AA standardile vastavat värvide kontrastsuse valideerijat [57].

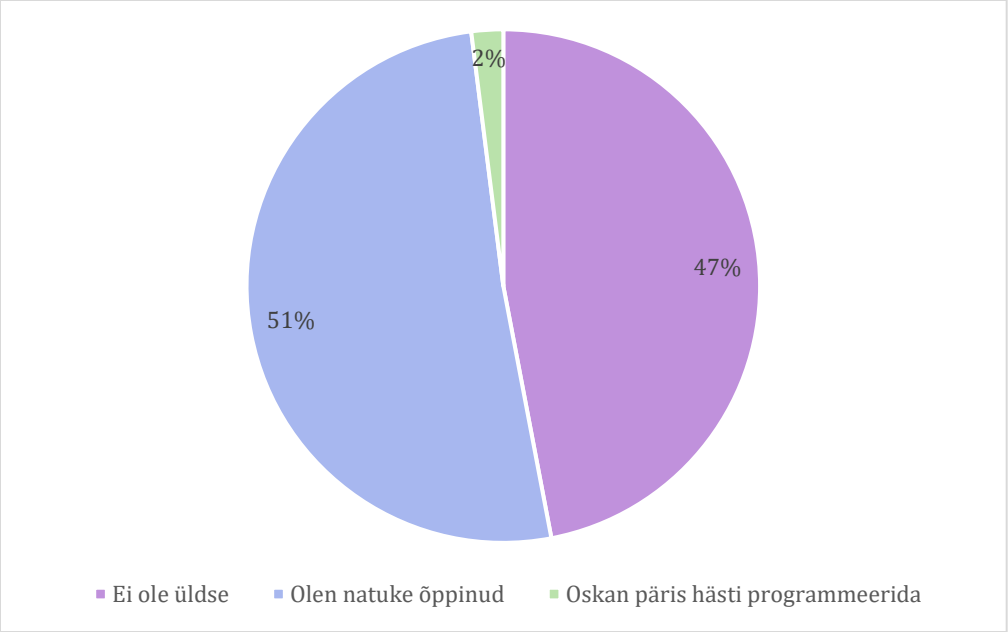
Õpetajate tagasisideküsitlus koosnes neljateistkümnest hinnangu andmise küsimusest, kus skaalal ühest viieni tuli hinnata, mis määral õpetaja nõustub etteantud väidetega. Üks tähistas üldse mitte nõustumist ja viis täielikult nõustumist. Hinnati teemade valikut ja järjestust, teksti lihtsust ja korrektsust, õpiku kujundust, kasutamismugavust, enesekontrolliteste, vihjeid ning õpiku

raskusastet. Lisaks uuriti, kas õpetajad kasutaks seda õpikut oma tunnis või huviringis, kas nad soovitaks seda õpikut teistele õpetajatele, ning mis kooliastet nad õpetavad. Küsitluse lõpus oli kaks vabas vormis küsimust puuduolevate teemade ja üldise tagasiside jaoks.

Tagasisidet küsiti ka sihtgrupilt ehk põhikooli õpilastelt, et saada parem ülevaade õpiku eakohasusest ja arusaadavusest ning õpilaste motiveeritusest seda õpikut kasutada. Õpilaste tagasisideküsitluse põhjaks võeti LOES-S mudel [42], mida samuti kohandati konkreetse materjali jaoks.

Ka õpilaste tagasisideküsitluses oli põhilisel kohal väidetega nõustumise kohta hinnangute andmine. Kasutatud skaala oli sama, mis õpetajate tagasiside puhul. Sarnaselt õpetajatele pidid õpilased hindama õpiku teemadevalikut, teksti selgust ja lihtsust, õpiku kujundust, enesekontrolliküsimusi, vihjeid ja raskusastet. Lisaks uuriti õpilastelt, mis klassis nad käivad, kas ja kuidas nad on varem programmeerimist õppinud, millistest teemadest tundsid nad puudust, millised ülesannete vihjed olid nende arvates eriti kasulikud ja millistest vihjetest tundsid nad puudust. Oli ka võimalus vabas vormis tagasisidet anda.

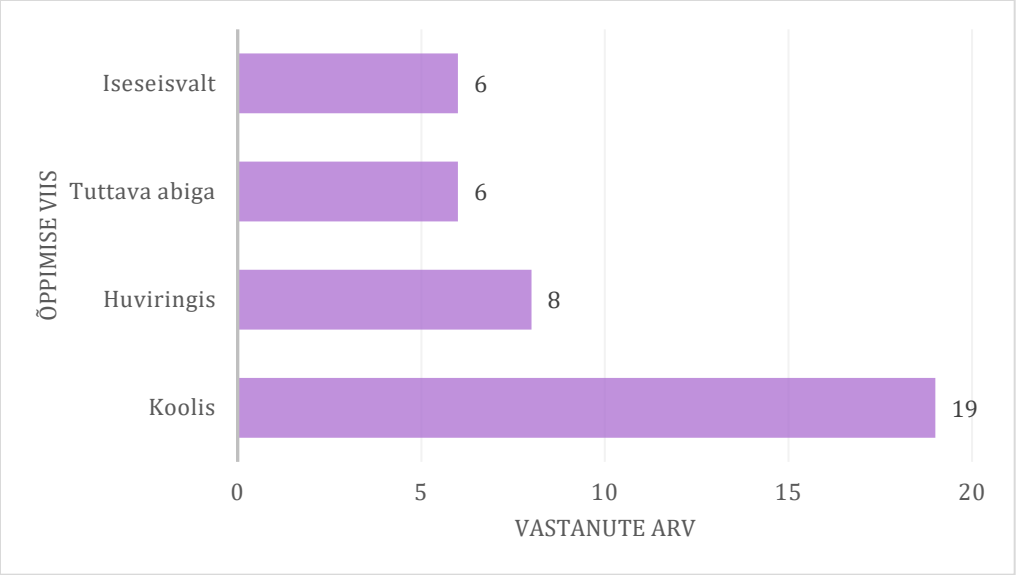
Tagasisidet koguti ühe kuu jooksul, ajavahemikus 15. märts 2023 kuni 15. aprill 2023. Õpetajate puhul kasutati mugavusvalimit ja küsitlus saadeti seitsmele õpetajale, kellel on kogemust põhikoolis programmeerimise õpetamisega. Mõned õpetajad palusid ka oma õpilastel õpikuga tutvuda ja õpilastele mõeldud küsimustikku täita. Lisaks saadeti õpilastele mõeldud küsitlus ka mõnele autori tuttavale vastava kooliastme õpilasele. Kokku paluti tagasisidet 50 õpilaselt. Küsitlusele vastas neli õpetajat ja 45 õpilast. Kaks õpetajatest juhendasid klasse nii teises ja kolmandas kooliastmes kui ka gümnaasiumis. Üks õpetaja õpetas vaid gümnaasiumis ja üks vaid teises kooliastmes. Õpilaste seas teisest kooliastmest vastajaid ei olnud, 84% õpilastest olid kolmandast kooliastmest ja 16% gümnaasiumist.



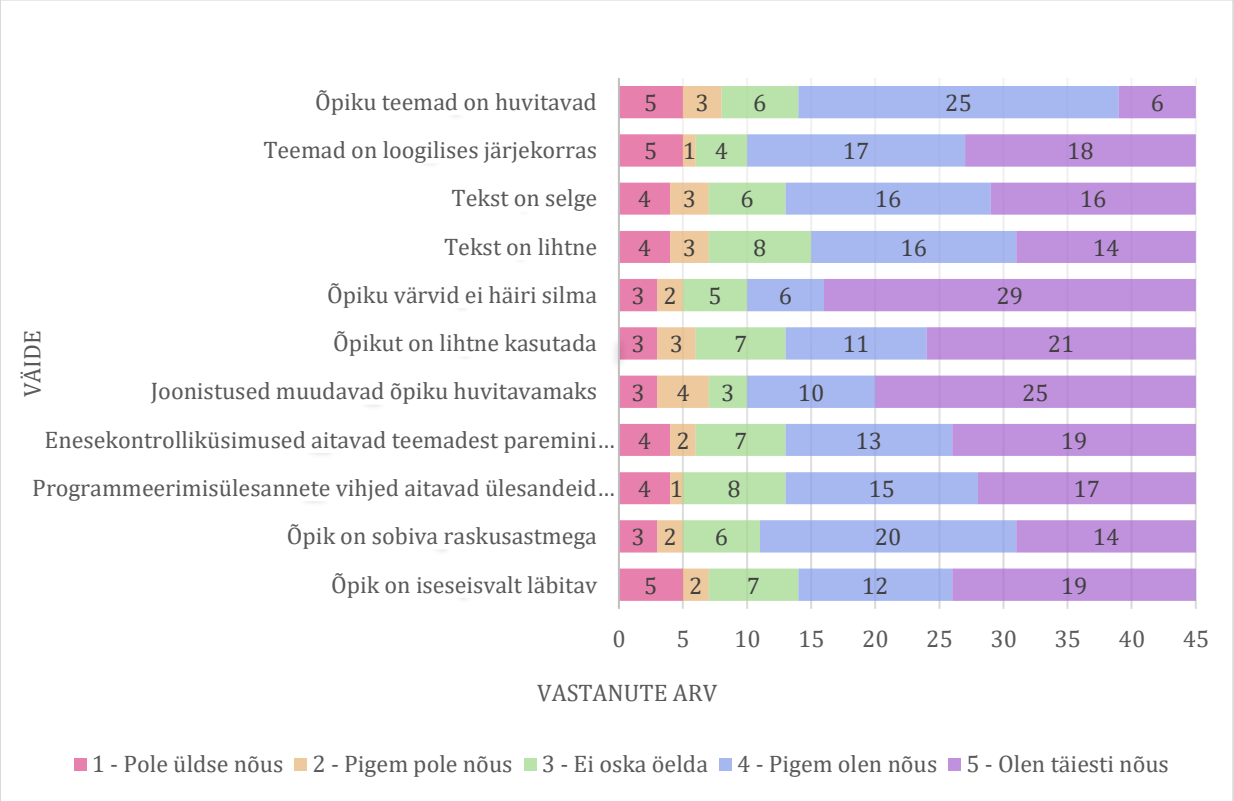
Joonis 18. Õpilaste varasem kogemus programmeerimisega

Küsitlusele vastanud õpilaste seas oli vaid üks õpilane, kes oskas enda hinnangul hästi programmeerida, ülejäänud vastanud jagunesid umbes pooleks, 23 vastanutest olid natuke programmeerimist õppinud ja 21 polnud üldse programmeerimisega kokku puutunud (joonis 18). Nendelt õpilastelt, kes olid programmeerimisega kokku puutunud, uuriti ka, kuidas nad on programmeerimist õppinud. Variandid olid: koolis, huviringis, tuttava abiga ja iseseisvalt ning õpilane sai valida kõik variandid, mis tema puhul kehtisid. Suur osa varasemast programmeerimiskogemusest oli omandatud koolis (joonis 19), kusjuures 14 vastajat olid omandanud varasemat kogemust ainult koolis, viis vastajat ainult huviringis, neli vaid tuttava abiga ja kaks täiesti iseseisvalt.

Õpilaste hinnangul oli üle kõigi väidete keskmine tulemus viiepalliskaalal 3,9, mis tähendab, et väidetega oldi pigem nõus. Kõige madalama tulemuse sai väide „Õpiku teemad on huvitavad“ ja kõige kõrgema tulemuse väide „Õpiku värvid ei häiri silma“ (tabel 1). Mediaaniks kujunes pea kõikidel väidetel 4, välja arvatud väited värvivaliku ja joonistuste kohta, mille mediaaniks kujunes 5. Täpsem hinnangute jagunemine on toodud joonisel 20.



Joonis 19. Varasema programmeerimiskogemuse omandamise viis



Joonis 20. Õpilaste hinnangute jaotumine

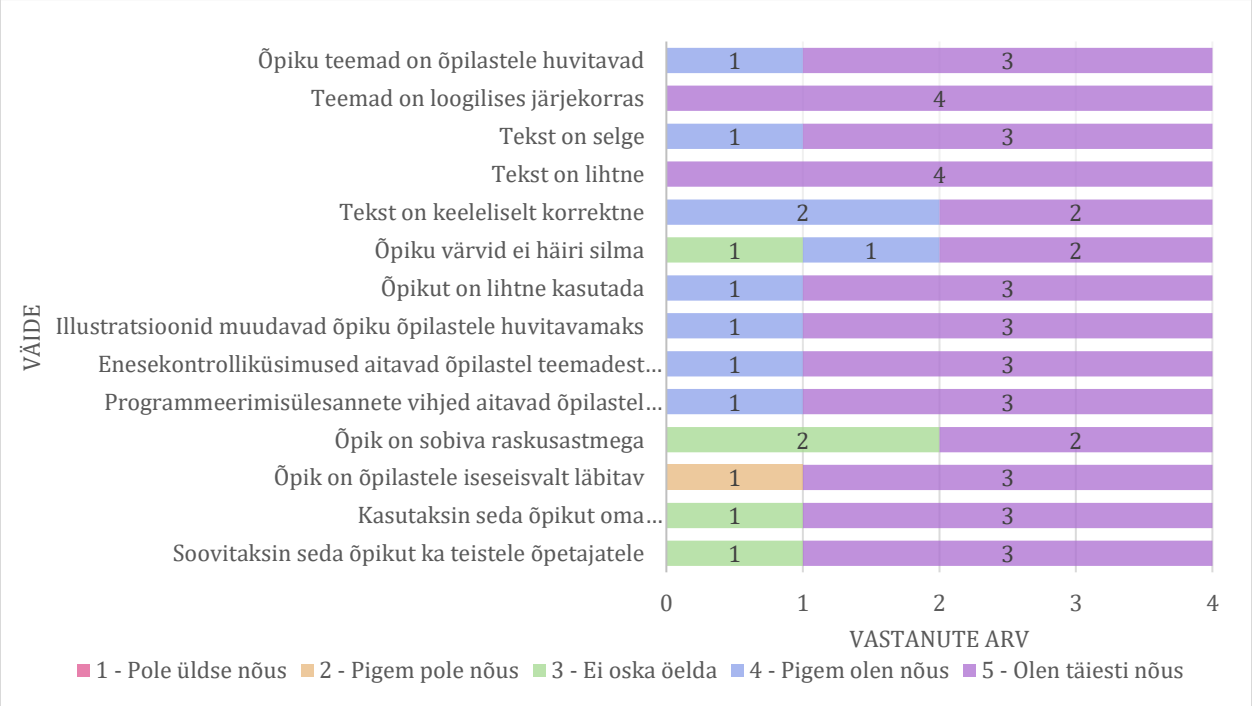
Tabel 1. Õpilaste hinnangud

Väide	**Aritmeetiline keskmine**	**Standardhälve**	**Mediaan**	***Min***	***Max***
Õpiku teemad on huvitavad	3,53	1,16	4	1	5
Teemad on loogilises järjekorras	3,93	1,27	4	1	5
Tekst on selge	3,82	1,25	4	1	5
Tekst on lihtne	3,73	1,23	4	1	5
Õpiku värvid ei häiri silma	4,24	1,23	5	1	5
Õpikut on lihtne kasutada	3,98	1,23	4	1	5
Joonistused muudavad õpiku huvitavamaks	4,11	1,27	5	1	5
Enesekontrolliküsimused aitavad teemadest paremini aru saada	3,91	1,26	4	1	5
Programmeerimisülesannete vihjed aitavad ülesandeid lahendada	3,89	1,21	4	1	5
Õpik on sobiva raskusastmega	3,89	1,11	4	1	5
Õpik on iseseisvalt läbitav	3,84	1,33	4	1	5

Õpilased said vabas vormis kirjutada, millistest teemadest nad puudust tundsid. Suurem osa õpilastest ei kirjutanud küsimuse vastuseks midagi või ütlesid, et ei tundnud ühestki teemast puudust. Küll aga tõid üksikud õpilased välja rekursiooni ja teegi teemad, mida pole veel õpikusse lisatud. Samuti uuriti õpilastelt, millised ülesannete vihjed olid nende jaoks kasulikud ja millistest vihjetest tundsid nad puudust. Mõned õpilased kirjutasid, et neil polnud vihjeid vaja, kuna said ise ülesannetega hakkama ja seega polnud vihjed neile otseselt kasulikud. 12 õpilast kirjutasid, et

nende jaoks olid kõik vihjed kasulikud ja üks õpilane tõi veel eraldi välja funktsioonide peatüki ülesannete vihjed, kuid ei lisanud kas need olid tema arvates eriti kasulikud või puudulikud. Lisaks mainis üks õpilane, et pärast funktsioonide peatükki läksid vihjed tema arvates keeruliseks. Vabas vormis kommentaaridena tõid õpilased ka välja mõned ettepanekud, näiteks muuta avalehel kõik peatükkide nimed hüperlinkideks, mis viivad vastava peatükini, lisada arvutikomponentide juurde ingliskeelsed nimetused, sest need on noorte seas rohkem tuntud ja lisada rakendusele ka tume režiim (ingl *dark mode*).

Õpetajate antud hinnangud olid õpilaste omadest kõrgemad. Kõige kõrgemalt hinnati väiteid „Teemad on loogilises järjekorras“ ja „Tekst on lihtne“, millega olid kõik õpetajad täielikult nõus (tabel 2). Kõige kehvema keskmise hinnangu, milleks oli 4, sai väide „Õpik on sobiva raskusastmega“, see tähendab siiski, et keskmiselt õpetajad on pigem nõus selle väitega. Õpiku iseseisvalt läbitavuse kohta arvas üks II kooliastme õpetaja, et õpik pigem pole õpilastele iseseisvalt läbitav, kuid ülejäänud õpetajad arvasid siiski, et on (joonis 21). Seega II kooliastmele võib õpik jääda natuke keeruliseks, kuid III kooliastmele peaks raskusaste olema paras. Ka õpetajatelt uuriti, millised teemad võiks õpikus veel olla, ja üks õpetaja tõi sarnaselt õpilastele välja rekursiooni teema.



Joonis 21. Õpetajate hinnangute jaotumine

Tabel 2. Õpetajate hinnangud

Väide	**Aritmeetiline keskmine**	**Standardhälve**	**Mediaan**	***Min***	***Max***
Õpiku teemad on õpilastele huvitavad	4,75	0,5	5	4	5
Teemad on loogilises järjekorras	5	0	5	5	5
Tekst on selge	4,75	0,5	5	4	5
Tekst on lihtne	5	0	5	5	5
Tekst on keeleliselt korrektne	4,5	0,58	4,5	4	5
Õpiku värvid ei häiri silma	4,25	0,96	4,5	3	5
Õpikut on lihtne kasutada	4,75	0,5	5	4	5
Illustratsioonid muudavad õpiku õpilastele huvitavamaks	4,75	0,5	5	4	5
Enesekontrolliküsimused aitavad õpilastel teemadest paremini aru saada	4,75	0,5	5	4	5
Programmeerimisülesannete vihjed aitavad õpilastel ülesandeid lahendada	4,75	0,5	5	4	5
Õpik on sobiva raskusastmega	4	1,15	4	3	5
Õpik on õpilastele iseseisvalt läbitav	4,25	1,5	5	2	5
Kasutaksin seda õpikut oma informaatikatunnis/huviringis	4,5	1	5	3	5
Soovitaksin seda õpikut ka teistele õpetajatele	4,5	1	5	3	5

Vabas vormis tagasisides mainiti, et enesekontrolliküsimused on selged ja meeldib just see, et ei öelda kohe vastust ette, vaid antakse vihje. Samuti meeldisid õpetajatele sammhaaval antavad ülesannete vihjed. Andmetüüpide ja -struktuuride tähtsamate operatsioonide kuvamist tabelis kiideti samuti, öeldes, et nii on informatsioon kompaktne ja kiiresti leitav. Võrreldes gümnaasiumi programmeerimisõpikuga toodi eelisena välja just illustratsioonid, mis aitavad loetut visualiseerida. Üks õpetajatest arvas, et see on üks parimaid teemakohaseid eestikeelseid õpikuid, mis on lihtsas lapsele arusaadavas keeles kirjutatud, kuid väljendas kahtlust, kas õpilased objektorienteeritud programmeerimise teemast korralikult aru saavad. Kiideti ka projektide peatükki, kuna õpilased tahavadki teha just selliseid n-ö päris asju nagu mängud ja rakendused. Lisaks pakuti välja sõnastuse parandusi mõningate peatükkide juurde ja mainiti, et oodati võimalust enesekontrolliküsimust uuesti genereerida rohkematel küsimustel, hetkel sai see võimalus lisatud vaid ühele enesekontrolliküsimusele.

Lisaks tagasisideküsitlustele jälgiti õpiku kasutamist tagasiside andmise perioodil ka Google Analytics [58] tööriista abil. Selle aja jooksul külastas veebirakendust 123 erinevat kasutajat, kellest enamik jõudis lehele otse aadressi või lingi kaudu, kuid mõningatel juhtudel jõuti veebilehele ka läbi mõne otsingumootori otsingu. Kõige populaarsem lehekülg oli avaleht, mis on ka loogiline tulemus, sellele järgnesid „Sissejuhatus“, „Algoritm“ ning „Muutujad ja avaldised“. Õpiku viimaseid peatükke vaadati vähem kui esimesi.

Õpetajate ja õpilaste tagasiside põhjal viidi sisse parandusi sõnastuse osas, lisati arvutiosade teema juurde ingliskeelsed nimetused ning muudeti avalehel sisukorras peatükkide nimed hüperlinkideks. Tagasisidet kasutati ka õppematerjalile hinnangu andmiseks ja edasiarendusvõimaluste kaardistamiseks, mis on toodud järgmises peatükis.

5. Piirangud ja edasiarendusvõimalused

Loodud õppematerjalile hinnangu andmiseks kasutatud õpilaste ja õpetajate valimit võib lugeda üheks töö piiranguks. Õpetajate puhul oli vastajaid neli, mis võib olla liiga vähe pädeva hinnangu saamiseks. Siiski tasub täheldada, et õpetajad andsid ka palju vabas vormis tagasisidet, mis on kasulik hinnangute mõtestamisel. Õpilaste puhul oli valimi suurus piisav, kuid on kaheldav, kas kõik õpilased ka õpikusse süvenesid ja küsitlusele vastamist tõsiselt võtsid. Näiteks on vastajate seas mitmeid õpilasi, kes väidetega ei nõustunud üldse või pigem ei nõustunud, kuid vabas vormis tagasisides kirjutasid, et kõik oli hästi ja midagi kriitikana välja ei toonud. Üleüldiselt oli ka vabas vormis ettepanekuid ja kommentaare vähe, mistõttu on raske aru saada täpsematest probleemsetest kohtadest õpilaste vaatenurgast.

Õpilaste hinnangute põhjal võib öelda, et üldiselt olid õpilased õpikuga rahul. Kõige madalama keskmise hinnangu (3,53) saanud väite „Õpiku teemad on huvitavad“ põhjal võib oletada, et ilmselt võiks õpiku teksti uuesti üle vaadata ja proovida õpilastele huvitavamaks teha. Samas ei andnud ükski õpilane täpsemalt teada, mis tema jaoks õpiku teemade juures igavaks jäi. Vastajad ei olnud ka tingimata programmeerimisest huvitatud noored, kelle jaoks see õpik eelkõige loodud on. Samuti võiks õpiku teksti veelgi lihtsamaks muuta, sest väide „Tekst on lihtne“ oli ka üks madalama hinnanguga väidetest (hinnang 3,73). Õpetajad hindasid samuti kõige madalamalt väidet „Õpik on sobiva raskusastmega“, mis kinnitab seda, et õpikut võiks veel lihtsustada. Samas oli õpetajate antud keskmine hinnang neli, mis tähendab, et nad pigem nõustusid selle väitega.

Vabas vormis tagasiside põhjal selgus, et õpetajad ja õpilased sooviksid näha õpikus veel kahte teemat: rekursioon ja teegid. Vastavad peatükid võiks tulevikus edasiarenduste käigus õpikusse lisada. Õpiku peatükkidesse võiks lisada ka arutlemisküsimusi ja viiteid lisamaterjalidele. Samuti võiks täiendada enesekontrolliküsimuste kogu, et lisada rohkematele enesekontrolliküsimustele võimalus küsimust uuesti genereerida. Hetkel jäi see võimalus vaid programmeerimiskeelte tundmise küsimuse juurde, kuna seal oli lihtne kasutada kõiki õpikus toodud programmeerimiskeelte näiteid. Ülejäänud enesekontrolliküsimuste juures nõuab see täiesti uute ülesannete väljamõtlemist. Õpilaste ettepanekutest jäi veel sisse viimata tumeda ja heleda režiimi vahetamise võimalus, mis jääb samuti üheks võimalikuks edasiarenduseks.

Autori enda tehnilisest testimisest jäid veel silma kaks probleemi, mida ei ole jõutud veel lahendada. Esiteks ei keri brauserid Mozilla Firefox ja Safari vahepeal lehekülge täiesti üles, kui peatükki vahetada. Brauseritega Google Chrome ja Microsoft Edge seda probleemi ei tuvastatud, seega tegemist on brauserispetsiifilise veaga, mida autoril pole veel õnnestunud parandada. See muudab õpiku kasutamist natuke ebamugavamaks, kuid ei takista seda. Teine probleem on seotud mobiilivaatega. Navigeerides mõnele lehele, kus on lai pilt või tabel, mis ei mahu vaatesse ära, venitatakse kogu lehekülg laiemaks ja ka nn hamburgerimenüü liigub paremasse äärde. Nutitelefon jätab sellisel juhul osa leheküljest vaatest välja ja laiade tabelite, piltide ja hamburgerimenüü nägemiseks tuleb kas ise lehekülge paremale kerida või kogu lehekülje suumi vähendada. See võib tekitada mobiilivaates kasutajale segadust ja tuleb autoril ära lahendada: probleemsed pildid tuleks teha mobiilivaate jaoks väiksemaks ja tabelitele lisada eraldi kerimisriba.

Üldiselt võib loodud õpikuga rahule jääda, seda kinnitavad ka õpilaste ja õpetajate hinnangud ja tagasiside. Autor jäi samuti õpiku loomise protsessiga rahule. Väga kasulik oli erinevate mudelite rakendamine, mis andis kindlad juhised erinevates etappides töötamiseks.

Magistritöö raames loodud õpik on Creative Commons Attribution 4.0 International License [59] litsentsiga, mis tähendab, et kõik võivad loodud õpikut kasutada ja jagada. Peale eeltoodud õpiku tehniliste probleemide lahendamist plaanib autor õpikut ka e-koolikotis [60] jagada, mis aitab õpetajatel ja õpilastel üle Eesti loodud õppematerjali leida.

Kokkuvõte

Magistritöö eesmärgiks oli luua digitaalne programmeerimisõpik II ja III kooliastmele programmeerimise iseseisvaks õppimiseks ja loodud õppematerjali tagasiside põhjal analüüsida ja täiendada. Esimese uurimisküsimusega sooviti välja selgitada, kuidas luua noortele programmeerimist tutvustav õppematerjal. Selleks analüüsiti magistritöö teoreetilises osas varasemaid töid ja loodud õppematerjale ning valiti välja mudelid, millele õppematerjali koostamisel tugineda.

Õpiku loomisel kasutati ADDIE mudelit [38], mis aitas seada eesmärke ning planeerida tööd. Esimeses etapis toimus analüüs ja töö teoreetilise osa kirjutamine. Teises etapis pandi paika õpiku struktuur, õpiväljundid ja käsitletavad teemad. Kolmandas ehk väljatöötamise etapis loodi digitaalne õppematerjal koos enesekontrollitestide ja programmeerimisülesannetega. Loodud õppematerjalis on 18 erinevat peatüki, milles kaetakse ära programmeerimise põhiteadmised, objektorienteeritud programmeerimine ning lisatud on ka projektid mängu ja rakenduste loomiseks. Tehnoloogiatest kasutati õpiku loomiseks veebiraamistikku Angular, mille abil oli mugav luua erinevaid interaktiivseid komponente. Digitaalne programmeerimisõpik on kõigile kättesaadav aadressil *progema.ee*. Viimases etapis küsiti õpetajatelt ja õpilastelt õpikule tagasisidet, mille põhjal tehti parandusi ja hinnati õppematerjali.

Õpetajate ja õpilaste tagasiside põhjal saadi vastus teisele uurimisküsimusele – kuidas hindavad loodud õppematerjali õpetajad ja õpilased. Küsimustikule vastas neli õpetajat ja 45 õpilast ja tulemustest selgus, et õppematerjali hinnati üldiselt positiivselt. Tehti ka ettepanekuid parandusteks ja lisadeks, mis viidi õppematerjali sisse või toodi välja võimalike edasiarendustena.

Viidatud kirjandus

[1] Eesti Töötukassa. Tööjõuvajaduse baromeeter. https://www.tootukassa.ee/baromeeter/poster (07.02.2022)

[2] Wong G. K.-W., Cheung H.-Y. Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming, *Interact. Learn. Environ.*, vol. 28, no. 4, pp. 438–450, 2020. doi: 10.1080/10494820.2018.1534245

[3] Vabariigi Valitsuse 6. jaanuari 2011. a määrus nr 1 „Põhikooli riiklik õppekava" Lisa 10. https://www.riigiteataja.ee/aktilisa/1230/4202/1010/1m%20lisa10.pdf# (29.01.2022)

[4] 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 (29.01.2022)

[5] Leppik C., Haaristo H.-S., Mägi E. IKT-haridus: digioskuste õpetamine, hoiakud ja võimalused üldhariduskoolis ja lasteaias, Poliitikauuringute Keskusele Praxis, Tallinn, 2017. https://www.praxis.ee/wp-content/uploads/2016/08/IKT-hariduse-uuring_aruanne_mai2017.pdf (07.04.2023)

[6] 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 (29.01.2022)

[7] National Research Council, Computer Science and Telecommunications Board, Committee on Information Technology Literacy, *Being Fluent with Information Technology*. Washington, D.C.: National Academies Press, 1999, p. 48. doi: 10.17226/6482

[8] Chao P.-Y. Exploring students' computational practice, design and performance of problem-solving through a visual programming environment, *Comput. Educ.*, vol. 95, pp. 202–215, 2016. doi: 10.1016/j.compedu.2016.01.010

[9] Crews T., Butterfield J. Using technology to bring abstract concepts into focus: A programming case study, *J. Comput. High. Educ.*, vol. 13, no. 2, pp. 25–50, 2002. doi: 10.1007/BF02940964

[10] Vujosevic Janicic M., Tošić D. The role of programming paradigms in the first programming courses, *Teach. Math.*, vol. 11, pp. 63–83, 2008. https://eudml.org/doc/257050 (09.04.2023)

[11] Stajano F. Python in Education: Raising a Generation of Native Speakers. https://www.cl.cam.ac.uk/~fms27/papers/2000-Stajano-native.pdf (09.04.2023)

[12] Pabon O. S., Villegas L. E. M. Fostering Motivation and Improving Student Performance in an Introductory Programming Course: An Integrated Teaching Approach, *Rev. EIA*, vol. 16, p. 65, 2019. doi: 10.24050/reia.v16i31.1230

[13] Noone M., Mooney A. Visual and textual programming languages: a systematic review of the literature, *J. Comput. Educ.*, vol. 5, no. 2, pp. 149–174, 2018. doi: 10.1007/s40692-018-0101-5

[14] Lin Y.-T., Yeh M. K.-C., Tan S.-R. Teaching Programming by Revealing Thinking Process: Watching Experts' Live Coding Videos With Reflection Annotations, *IEEE Trans. Educ.*, vol. 65, no. 4, pp. 617–627, 2022. doi: 10.1109/TE.2022.3155884

[15] Gaspar A., Langevin S. Restoring „coding with intention" in introductory programming courses, in *Proceedings of the 8th ACM SIGITE conference on Information technology education*, Destin Florida USA: ACM, 2007, pp. 91–98. doi: 10.1145/1324302.1324323

[16] Brusilovsky P., Sosnovsky S. Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK, *ACM J. Educ. Resour. Comput.*, vol. 5, 2005. doi: 10.1145/1163405.1163411

[17] Lepp M., Luik P., Palts T., Papli K., Suviste R., Säde M., Hollo K., Vaherpuu V., Tõnisson E. Self- and Automated Assessment in Programming MOOCs, in *Technology Enhanced Assessment*, D. Joosten-ten Brinke and M. Laanpere, Eds., in Communications in Computer and Information Science. Cham: Springer International Publishing, 2017, pp. 72–85. doi: 10.1007/978-3-319-57744-9_7

[18] Fessakis G., Komis V., Dimitracopoulou A., Prantsoudi S. Overview of the Computer Programming Learning Environments for primary education, *Rev. Sci. Math. ICT Educ.*, vol. 13, no. 1, Art. no. 1, 2019. doi: 10.26220/rev.3140

[19] Dagienė V., Jevsikova T., Stupurienė G. Introducing Informatics in Primary Education: Curriculum and Teachers' Perspectives, in *Informatics in Schools. New Ideas in School Informatics*, in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 83–94. doi: 10.1007/978-3-030-33759-9_7

[20] Põhikooli riiklik õppekava, *Riigi Teataja*. https://www.riigiteataja.ee/akt/123042021010 (29.01.2022)

[21] Veltmann A. Informaatikaõpe II ja III kooliastmes Tartumaa koolide näitel, TÜ arvutiteaduse instituudi bakalaureusetöö. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=71666&year=2021 (16.04.2023)

[22] Parve K. Informaatikat õpetavate õpetajate koolitusvajadus, TÜ arvutiteaduse instituudi magistritöö. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=72544&year=2021 (20.03.2023)

[23] TÜ Arvutiteaduse instituudi programmeerimise algkursuse õpik. https://progeopik.cs.ut.ee/index.html (16.04.2023)

[24] Tartu Ülikooli digiõpikud. https://courses.cs.ut.ee/t/digiopik/ (16.04.2023)

[25] Jaaksaar K., Kirbits M., Oad M. *Digiõpik I kooliaste*. https://web.htk.tlu.ee/informaatika/opik1/ (16.04.2023)

[26] Poudel D., Oad M., Mustjatse K. *Digiõpik II kooliaste*. https://web.htk.tlu.ee/informaatika/opik2/ (16.04.2023)

[27] Tartu Ülikooli kursus Programmeerimisest maalähedaselt. https://courses.cs.ut.ee/2023/progmaa/spring (16.04.2023)

[28] Tõnisson E., Palts T., Säde M., Tõnisson K., Suviste R., Meier H. *Programmeerimine*. Tartu Ülikool. https://web.htk.tlu.ee/digitaru/programmeerimine/part/sissejuhatus/ (16.04.2023)

[29] TalTechi programmeerimise algkursus, 09.03.2019. https://taltech.ee/koostoo-koolidega/valikained (16.04.2023)

[30] TalTechi Pythoni õppematerjalid. https://pydoc.pages.taltech.ee/ (16.04.2023)

[31] Pärnu koolide digipööre oskusainetes 2021-2023, *Pärnu Kuninga Tänava Põhikool*. https://kuninga.parnu.ee/projektid/parnu-koolide-digipoore-oskusainetes-2021-2023/ (16.04.2023)

[32] Programmeerimine Pythonis, videokursus. https://www.youtube.com/playlist?list=PLAp4ww4WBdn-IesrpQf3YreSrpjp_1dxT (16.04.2023)

[33] Lepp M., Luik P., Palts T., Papli K., Suviste R., Säde M., Tõnisson E. MOOC in Programming: A Success Story, *Proc. 12th Int. Conf. E-Learn.*, pp. 138–147. https://www.proquest.com/openview/66098b63a50342a7f95ff5668b93e0fc/1?pq-origsite=gscholar&cbl=1796414 (30.04.2023)

[34] Code.org. https://studio.code.org/home (16.04.2023)

[35] Codecademy. https://www.codecademy.com/ (16.04.2023)

[36] Codelearn. https://codelearn.io/home (16.04.2023)

[37] Villems A., Aluoja L., Pilt L., Naulainen M.-M., Kusmin M., Rogalevitš V., Tokko U. Digitaalse õppematerjali loomise soovitused – Juhend digitaalse õppematerjali autorile. https://oppevara.edu.ee/kvaliteet/#mis-on-digitaalne-oppematerjal (13.02.2022)

[38] Allen W. C. Overview and Evolution of the ADDIE Training System, *Adv. Dev. Hum. Resour.*, vol. 8, no. 4, pp. 430–441, 2006. doi: 10.1177/1523422306292942

[39] Dafoulas G. A., Frumkin L. A., Mimirinis M., Murphy A. Investigating computer-supported cooperative learning: Applications for flexible learning environments, in *The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005.*, Cairo, Egypt: IEEE, 2005, pp. 83–94. doi: 10.1109/AICCSA.2005.1387142

[40] Põldoja H. Õppematerjalide koostamise protsess ja kvaliteet, *Digitaalne õppevara*, 05.02.2017. https://digioppevara.wordpress.com/lugemismaterjalid/oppematerjalide-koostamise-protsess-ja-kvaliteet/ (27.02.2022)

[41] Nesbit J., Belfer K., Leacock T. Learning Object Review Instrument (LORI), 26.01.2004. http://web.archive.org/web/20040126041853/http://elera.matchbox.surrey.sfu.ca/eLera/Home/Articles/LORI%201.5.pdf (27.02.2022)

[42] Kay R. H., Knaack L. Assessing learning, quality and engagement in learning objects: the Learning Object Evaluation Scale for Students (LOES-S), *Educ. Technol. Res. Dev.*, vol. 57, no. 2, pp. 147–168, 2009. doi: 10.1007/s11423-008-9094-5

[43] Vabariigi Valitsuse 6. jaanuari 2011. a määrus nr 1 „Põhikooli riiklik õppekava“ Lisa 3. https://www.riigiteataja.ee/aktilisa/1290/8201/4020/1m%20lisa3.pdf# (16.04.2023)

[44] Ferris K., Zhang S. A Framework for Selecting and Optimizing Color Scheme in Web Design, in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 532–541. doi: 10.1109/HICSS.2016.73

[45] Kuo L., Chang T., Lai C.-C. Affective psychology and color display of interactive website design, *Displays*, vol. 71, p. 102134, 2022. doi: 10.1016/j.displa.2021.102134

[46] Gu Z., Lou J. Data driven webpage color design, *Comput.-Aided Des.*, vol. 77, pp. 46–59, 2016. doi: 10.1016/j.cad.2016.03.001

[47] Angular. https://angular.io/ (16.04.2023)

[48] Angular Bootstrap. https://ng-bootstrap.github.io/#/home (16.04.2023)

[49] Angular Fontawesome. https://github.com/FortAwesome/angular-fontawesome (16.04.2023)
[50] Sousli M. Angular Highlight.js. https://github.com/MurhafSousli/ngx-highlightjs (16.04.2023)
[51] Keps K. Veebirakenduse koodirepositoorium. https://github.com/kristiinakeps/programmeerimine (16.04.2023)
[52] Heroku, 12.09.2023. https://www.heroku.com/ (16.04.2023)
[53] Veebimajutus. https://www.veebimajutus.ee/ (16.04.2023)
[54] DNSimple. https://dnsimple.com/ (16.04.2023)
[55] CorelDRAW. https://www.coreldraw.com/en/ (16.04.2023)
[56] The W3C Markup Validation Service. https://validator.w3.org/ (30.04.2023)
[57] Color Contrast Accessibility Validator. https://color.a11y.com/Contrast/ (30.04.2023)
[58] Google Analytics. https://analytics.google.com/analytics/web/#/ (24.04.2023)
[59] Creative Commons CC BY 4.0. https://creativecommons.org/licenses/by/4.0/ (01.05.2023)
[60] E-koolikott. https://e-koolikott.ee/ (01.05.2023)

Lisad

I. LORI ja LOES-S mudelid

LORI mudeli osad on [41]:

1) sisu kvaliteet – kas materjal on tõene, täpne ning piisava detailsusega;
2) vastavus õpieesmärkidega – kas õpieesmärgid, tegevused, hindamismeetodid ning sihtgrupi eripärad on omavahel kooskõlas;
3) tagasiside ja kohandumine – kas materjalis on õppija tegevustele vastav kohandatud sisu ja tagasiside;
4) motivatsioon – kas materjal on õppijate jaoks huvitav ja motiveeriv;
5) esitlus ja kujundus – kas visuaalse- ja audiomaterjali kujundus on õppimist toetav;
6) Interaktsiooni kasutatavus – kas materjalis on lihtne navigeerida, kas kasutajaliides on ennustatav ja abivahendid kvaliteetsed;
7) ligipääsetavus – kas materjal sobib erivajadustega ja mobiilsetele õppijatele;
8) taaskasutatavus – kas materjal sobib kasutamiseks erinevates olukordades ja erineva taustaga õpilastele;
9) standarditele vastavus – kas materjal vastab rahvusvahelistele standarditele ja spetsifikatsioonidele.

LOES-S mudeli väited ja küsimused on [42]:

Õppimine

1) Materjaliga töötamine aitas mul õppida.
2) Materjali tagasiside aitas mul õppida.
3) Graafika ja animatsioonid õppematerjalis aitasid mul õppida.
4) Õppematerjal aitas mulle uut kontseptsiooni õpetada.
5) Õppematerjal aitas mul üldiselt õppida.

Kvaliteet

6) Abi saamise tööriistad õppematerjalis olid kasulikud.

7) Juhised olid lihtsalt järgitavad.
8) Õppematerjali oli lihtne kasutada.
9) Õppematerjal oli hästi organiseeritud.

Kaasavus

10) Mulle meeldis õppematerjali temaatika.
11) Minu jaoks oli õppematerjal motiveeriv.
12) Mulle meeldiks uuesti seda õppematerjali kasutada.

Vabas vormis küsimused:

13) Kas ja mis sulle õppematerjali juures meeldis?
14) Kas ja mis sulle õppematerjali juures ei meeldinud?

II. Õpiväljundid

Õpiku läbinud õpilane:

1) teab, mis on arvuti ja millistest osadest arvuti koosneb;
2) teab, mis on programmeerimiskeeled ja milleks neid kasutatakse;
3) oskab algoritme plokkskeemide abil kujutada;
4) tunneb erinevaid andmetüüpe ja oskab neid kasutada;
5) teab, mis on muutujad ja oskab neid kasutada;
6) oskab koostada mitmeharulisi tingimuslauseid;
7) oskab koostada ja kombineerida erinevaid tõeväärtustehteid;
8) oskab koostada ja kasutada korduslauseid;
9) oskab kirjutada ja kasutada funktsioone;
10) oskab funktsioonist väärtust tagastada;
11) oskab tekstifailist andmeid lugeda ja tekstifaili andmeid kirjutada;
12) teab, milliseid vigu esineb programmeerides;
13) oskab erindeid kinni püüda ja tõstatada;
14) oskab järjendeid luua, järjendist elemente kätte saada ja muuta, järjendisse elemente lisada ja sealt elemente eemaldada;
15) oskab sõnastikke luua ja kasutada, sõnastikust elemente kätte saada ja neid sinna lisada;
16) oskab luua ja kasutada hulki, tunneb hulgaoperatsioone;
17) oskab luua ja kasutada ennikuid;
18) oskab sõltuvalt olukorrast valida ülesande lahendamiseks sobiv andmestruktuur;
19) oskab kõiki andmestruktuure tsüklis läbida;
20) teab, kuidas koostada mitmemõõtmelisi andmestruktuure ja kuidas neid kasutada;
21) teab, mis on objektorienteeritud programmeerimine, klassid ja isendid;
22) oskab luua klassi koos konstruktori ja meetoditega;
23) oskab luua klassi põhjal isendeid;
24) teab, kuidas kasutada pärilust ja luua alamklasse;
25) suudab õpitud teadmisi rakendada erinevates projektides.

III. Õpiku sisukord

Õpiku sisukord on järgmine:

1.7. Failid
    1.7.1. Failist lugemine
    1.7.2. *For*-tsükkel
    1.7.3. Faili kirjutamine
    1.7.4. Kodeering
    1.7.5. Harjutused
1.8. Veahaldus
    1.8.1. Erindite püüdmine
    1.8.2. Hea praktika
    1.8.3. Erindite tõstatamine
    1.8.4. Harjutused
2. Andmestruktuurid
2.1. Järjendid
    2.1.1. Indeksid
    2.1.2. Operatsioonid, funktsioonid, meetodid
    2.1.3. Funktsioon *range*
    2.1.4. Harjutused
2.2. Sõnastikud
    2.2.1. Operatsioonid, funktsioonid, meetodid
    2.2.2. Harjutused
2.3. Hulgad
    2.3.1. Operatsioonid, funktsioonid, meetodid
    2.3.2. Harjutused
2.4. Ennikud
    2.4.1. Operatsioonid, funktsioonid, meetodid
    2.4.2. Harjutused
2.5. Mitmemõõtmelised andmestruktuurid
    2.5.1. Järjend
    2.5.2. Kombineerimine
    2.5.3. Harjutused

3. Objektorienteeritud programmeerimine
    3.1. Klassid ja isendid
        3.1.1. Konstruktor
        3.1.2. Isendimeetodid
        3.1.3. Meetod *__str__*
        3.1.4. Isendid andmestruktuurides
        3.1.5. Harjutused
    3.2. Pärilus
        3.2.1. Funktsioon *super*
        3.2.2. Ülekatmine
        3.2.3. Harjutused
4. Projektid
    4.1. Mäng
        4.1.1. Akna loomine
        4.1.2. Tegelase loomine
        4.1.3. Liikumine
        4.1.4. Lumepallid
        4.1.5. Kokkupõrked
        4.1.6. Kingitused
        4.1.7. Aja loendamine
        4.1.8. Lõpuaken
        4.1.9. Heli
    4.2. Veebirakendus
        4.2.1. Tööpõhimõte
        4.2.2. Loomine
        4.2.3. HTML
        4.2.4. Kodutöö lisamine
        4.2.5. Kodutöö kustutamine
        4.2.6. CSS
    4.3. Töölauarakendus
        4.3.1. Rakenduse aken

4.3.2. Vidinad

4.3.3. Paigutus

4.3.4. Vorm

4.3.5. Vorm vidinaks

4.3.6. Mängija klass

4.3.7. Nupud

4.3.8. Kujundus

IV. Tagasisideküsitlused

Õpetajate tagasisideküsitluse väited:

1) Õpiku teemad on õpilastele huvitavad.
2) Teemad on loogilises järjekorras.
3) Tekst on selge.
4) Tekst on lihtne.
5) Tekst on keeleliselt korrektne.
6) Õpiku värvid ei häiri silma.
7) Õpikut on lihtne kasutada.
8) Joonistused muudavad õpiku õpilastele huvitavamaks.
9) Enesekontrolliküsimused aitavad õpilastel teemadest paremini aru saada.
10) Programmeerimisülesannete vihjed aitavad õpilastel ülesandeid lahendada.
11) Õpik on sobiva raskusastmega.
12) Õpik on õpilastele iseseisvalt läbitav.
13) Kasutaksin seda õpikut oma informaatika tunnis/huviringis.
14) Soovitaksin seda õpikut ka teistele õpetajatele.

Õpilaste tagasisideküsitluse väited:

1) Õpiku teemad on huvitavad.
2) Teemad on loogilises järjekorras.
3) Tekst on selge.
4) Tekst on lihtne.
5) Õpiku värvid ei häiri silma.
6) Õpikut on lihtne kasutada.
7) Joonistused muudavad õpiku huvitavamaks.
8) Enesekontrolliküsimused aitavad teemadest paremini aru saada.
9) Programmeerimisülesannete vihjed aitavad ülesandeid lahendada.
10) Õpik on sobiva raskusastmega.
11) Õpik on iseseisvalt läbitav.

V. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Kristiina Keps,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Veebirakendus programmeerimise õppimiseks II ja III kooliastmes“, mille juhendaja on Marina Lepp, 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 Commonsi 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.

Kristiina Keps

09.05.2023