

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Marcus Vessel

**Lisamaterjali loomine kursuse „Objektorienteeritud
programmeerimine“ jaoks**

Bakalaureusetöö (9 EAP)

Juhendaja:
Marina Lepp, PhD

Tartu 2025

Lisamaterjali loomine kursuse „Objektorienteeritud programmeerimine“ jaoks

Lühikokkuvõte:

Tänapäeval on programmeerimine laialt levinud paljudes valdkondades. Ühte populaarsemat programmeerimiskeelt Java on õpetatud aastaid Tartu Ülikooli kursusel „Objektorienteeritud programmeerimine“. Aja jooksul on Java muutunud ja edasi arenenud ning enamasti pole neid uusi võimalusi kajastatud õppematerjalides. Seetõttu koostati bakalaureusetöös kursusele „Objektorienteeritud programmeerimine“ lisamaterjale, mis tutvustavad uusi programmeerimiskeelega Java seotud teemasid eesti keeles. Loodud materjalid võeti esmakordselt kasutusele 2024/2025. õppeaasta kevadel. Lisamaterjalide hindamiseks koostati küsimustik, millele vastas 211 kursusel osalejat. Üliõpilaste tagasisidest võib järeldada, et lisamaterjalid on arusaadavad ja kasulikud.

Võtmesõnad: lisamaterjal, objektorienteeritud programmeerimine, Java

CERCS: P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

Creating Supplementary Material for the Course “Object-Oriented Programming”

Abstract:

Nowadays programming is widely used in many fields. One of the most popular programming languages Java has been taught at the University of Tartu in the course “Object-Oriented Programming” for years. Over time Java has changed and evolved, but many of those new opportunities have not been reflected in the teaching materials. Therefore, as part of this bachelor's thesis, supplementary materials were created for the course “Object-Oriented Programming” that introduce new topics related to the Java programming language in Estonian. The materials were first introduced in the spring semester of the 2024/2025 academic year. To evaluate the supplementary materials, a questionnaire was prepared and answered by 211 course participants. From the students' feedback, it can be concluded that the supplementary materials are understandable and useful.

Keywords: supplementary material, object-oriented programming, Java

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

Sisukord

Sissejuhatus.....	4
1. Lisamaterjalide olemus	5
1.1 Lisamaterjalide otstarve	5
1.2 Lisamaterjalide vormid	6
2. Kursus „Objektorienteeritud programmeerimine“	8
2.1 Kursuse ülesehitus	8
2.2 Kursuse sisu	9
2.3 Lisamaterjalide vajadus	10
3. Lisamaterjalide koostamise protsess	11
3.1 Lisamaterjalide valitud teemad	11
3.2 Lisamaterjalide jaotus	12
3.3 Lisamaterjalide loomine.....	13
3.4 Enesekontrolli küsimuste loomine.....	15
4. Üliõpilaste tagasiside	17
4.1 Küsitluse koostamine ja tagasiside kogumine	17
4.2 Tagasiside analüüs	17
Kokkuvõte.....	22
Viidatud kirjandus.....	23
Lisad.....	28
I. Koostatud lisamaterjalid	28
II. Koostatud küsimustik kursusel osalejatele tagasiside kogumiseks	29
III. Litsents	35

Sissejuhatus

Tänapäeva ühiskonnas on programmeerimine laialt levinud paljudes valdkonnades, nagu energeetika, pangandus ja tervishoid (Dutta & Mathur, 2011). Programmeerimine ühendab inimesi ja tehnoloogilisi vahendeid ning selle õppimine arendab oskusi, mida saab kasutada ka väljaspool tehnoloogilist maailma. Üks populaarsemaid programmeerimiskeeli on Java, mida saab õppida Tartu Ülikooli kursusel „Objektorienteeritud programmeerimine“.

Kursus „Objektorienteeritud programmeerimine“ on toimunud üle kümne aasta, mille jooksul on Java muutunud ja edasi arenenud. Praegused õppematerjalid ei kajasta kõiki uute Java versioonide võimalusi, mis aitaksid paremini mõista programmeerimist. Nende kohta saab uurida paljudest enamasti inglise keelsetelt veebilehtedelt. Laialdase valiku tõttu võib olla algeliste programmeerimise teadmistega üliõpilasel keeruline valida asjakohast materjali. Samuti ei ole tehisintellekt alati usaldusväärne allikas, kuna mudelite, nagu ChatGPT vastustes võib esineda hallutsinatsioonide mõjul erinevaid vigu (Ho jt, 2024). Lisaks toimub kursus eesti keeles ja seetõttu võiks eksisteerida nende kohta materjale eesti keeles.

Käesoleva bakalaureusetöö eesmärk on luua Tartu Ülikooli kursusele „Objektorienteeritud programmeerimine“ lisamaterjale, mis tutvustavad uusi programmeerimiskeele Java seotud teemasid eesti keeles. Bakalaureusetöös valmivate materjalide kõige olulisem ülesanne on anda uusi teadmisi arvestades juba õpetatavate teemadega.

Bakalaureusetöö koosneb neljast sisulisest peatükist. Esimeses osas tutvustatakse lisamaterjalide erinevaid eesmärke ja võimalikke vorme. Teine peatükk annab ülevaate kursuse „Objektorienteeritud programmeerimine“ ülesehitusest, praeguste õppematerjalide sisust ja põhjendab lisamaterjalide vajadust. Kolmandas peatükis kirjeldatakse lisamaterjalide teemade valimise ja loomise protsessi ning teadmiste kinnistamiseks enesekontrollitestide loomist. Viimases peatükis käsitletakse üliõpilastele lisamaterjalide hindamiseks küsimustiku koostamist ja analüüsitakse saadud tagasisidet. Töö lõpus olevates lisades on lingid lisamaterjalidele ja küsimustik.

1. Lisamaterjalide olemus

See peatükk annab ülevaate mitmetest lisamaterjalide eesmärkidest ja nende vormidest.

1.1 Lisamaterjalide otstarve

Lisamaterjalide sisu ja eesmärgid määratakse vastavalt põhimaterjalile. Pop jt (2015) kirjutasid, et teadusartikli lisamaterjalide üks otstarvetest on anda lugejale rohkem informatsiooni, mis on kooskõlas tekstis olevate teemadega. Nende sõnul on lisamaterjalid tihti detailsemad ja mõnikord isegi mahukamad kui terve teadustekst ning seetõttu saab esitada töö sisu täpselt ja lühidalt. Samas suure mahukuse tõttu võib olla lugejal keeruline leida teda huvitavat osa. Samuti märgivad autorid, et mõned teadusajakirjad piiravad artikli maksimaalset lehekülgede arvu, mistõttu on lisade teine eesmärk hoiustada materjale, mis ei mahu peamisesse teksti. Sarnaselt on ka kursuse „Objektorienteeritud programmeerimine“ põhimaterjalide hulk piiratud, et õppurid ei ületaks ettenähtud õppemahtu. Seetõttu on kursusel vaja just vabatahtlikke lisamaterjale, mitte täiendatud kodutöid või loenguid.

Abimaterjalidel on veel potentsiaalseid eesmärke. Hačatrjana jt (2023) koostasid kooliõpilastele lisamaterjale üldoskuste edendamiseks. Täpsemalt oli eesmärk arendada probleemilahendusoskust, mis hõlmab probleemi mõistmist, sellele lahenduse otsimist ja teostamist ning tulemuse analüüsimist. Ka programmeerimist saab vaadelda kui probleemi, mille lahendamiseks tuleb seda mõista, disainida programmi ning kirjutada ja siluda koodi (Fedorenko jt, 2019). Lisaks taheti edendada õpilaste enesereguleeritud õppimise oskusi, nagu eesmärkide seadmist, ajaplaneerimist, abi küsimist ja enesehindamist. Programmeerimisel aitab see oskus efektiivsemalt programme luua ja olla motiveeritud (Ramírez-Echeverry jt, 2025).

Loomuliku keele ja programmeerimiskeele kasutamisel esineb mitmeid sarnasusi, näiteks mõlemal on kindel süntaks, mis määrab vastavalt lause ehitust või programmeerimisel meetodi ja klassi struktuuri (Fedorenko jt, 2019). Seega võib programmeerimise mõistmist edendada loomuliku keele õpetamise võtted ja tähelepanekud. Dodd jt (2015) toovad esile, et võõrkeele edukaks õppimiseks on kriitilise tähtsusega motivatsioon ning kõrgema huviga õpilased saavutavad paremaid õpitulemusi. Samas üks kõige tavalisematest tööriistadest keele õppimiseks on õpikud, mis võivad olla igavad ja monotoonsed. Motivatsiooni parandamiseks

saab kasutada huvitavaid lisamaterjale, mis peaksid muuhulgas toetama õppekava ja õpiväljundeid ning vähendada õpetaja tunni ettevalmistuseks kuluvat aega (Karki, 2018).

Siiski on keele õppimiseks oluline lugemine ning sellest oskusest võib sõltuda õppurite programmeerimiskeele õppimise ja kasutamise edukus (Endres jt, 2021). Lyonni jt (2023) sõnul tõmbavad õpilaste tähelepanu ning arendavad kirjutamist ja sõnavara just jutustavad tekstid (näiteks romaanid, näidendid ja koomiksid). Nad märgivad, et veel paremad on sellised jutud, mille juured asetsevad kohalikus kultuuris, aga analüüsi tulemusel oli selliseid tekste õppematerjalis vähe, mistõttu koostati lisamaterjalina raamat kohalikest rahvajuttudest. Siinkohal mainiksin, et kultuuriruumid ja nendes olevate relevantsete juttude arvud on erinevad ja seetõttu pole alati võimalik selliseid materjale luua. Sellegipoolest esineb kursusel „Objektorienteeritud programmeerimine“ (Tartu Ülikool, 2025) temaatilisi koodinäiteid ja ülesandeid (näiteks ID-kaardist, kolmnurgast ning isikutest) mitmeid kordi.

1.2 Lisamaterjalide vormid

Lisamaterjalide tüüpe saab kategoriseerida erinevate omaduste alusel. Üheks jaotuseks saab olla see, kas materjal on paberil (näiteks õpikud ja töövihikud) või digitaalne (näiteks veebilehed, videod ja puuteekraaniga tahvlid) (Karki, 2018). Delgado jt (2018) analüüsisid mõlema mõju teksti arusaamisele ja leidsid, et ajapiirang võib vähendada elektroonilise materjali mõistmist. Lisaks leidub digitaalses keskkonnas keskendumist segavaid tegureid, nagu teistes akendes lahti olevad programmid (Z. Liu, 2005). Samas on digitaalseid vahendeid kasutades lihtne erinevatele materjalidele ligi pääseda ja muuta enda lugemiskogemust meeldivamaks, näiteks kohandades teksti suurust (Pae, 2020). Kursusel „Objektorienteeritud programmeerimine“ on kõik materjalid kättesaadavad digitaalselt. Samas on mõned soovitatud materjalid, nagu David J. Ecki (2022) loodud raamat „Introduction to Programming Using Java“, olemas paberkujul.

Programmeerimise õppematerjalid võivad sisaldada erinevaid osasid. Tallinna Ülikooli gümnasistidele suunatud programmeerimise algkursuse materjalis esineb mõistete selgitamist, illustreerivaid pilte, tabeleid, hulganisti näiteid ja tihti on näidisprogrammi kood esitatud veel pseudokoodi või ühtse modelleerimiskeele kujul (Mironova jt, 2017). Loodus- ja täppiseaduste valdkonna ainetes kasutatakse ka interaktiivseid teste (Sharmin jt, 2025; Veedla, 2022).

Füüsiliselt käeshoitavad lisamaterjalid ei koosne ainult paberil olevast tekstist. Shin jt (2021) toovad esile, et tavainimestele on värvide abil objektide kirjeldamine enesestmõistetav, aga sama ei kehti nägemispuuetega isikutele. Nad saavad õppida värviteooriat läbi abstraktsete kirjelduste ja Braille'i kirjas loodud värviraamatute. Autorite sõnul polnud see piisav, mistõttu löid nad UV-prinditud lisamaterjale, mille kompamisel on tunda erinevaid sümboleid ja tekstuure, mis seostuvad erinevate värvidega. Ka programmeerimist saab esitada füüsilisel kujul. Näiteks Baweja (2020) tegi klotse, mis sümboliseerivad erinevaid programmeerimise operatsioone ja mida ühendades saab väljendada mitmeid algoritme. Samuti kasutatakse koodi pärismaailma toomiseks programmeeritavaid roboteid, nagu Lego MindStorms (Powers jt, 2006).

2. Kursus „Objektorienteeritud programmeerimine“

See peatükk keskendub kursuse „Objektorienteeritud programmeerimine“ ülesehitusele ja praeguste õppematerjalide sisule. Viimane alapeatükk kirjeldab kursusele lisamaterjale loomise vajadust.

2.1 Kursuse ülesehitus

Kursuse „Objektorienteeritud programmeerimine“ eesmärk on anda põhiteadmisi objektorienteeritud programmeerimisest ning arendada programmide kirjutamise oskust ja koostöö tegemist (Tartu Ülikooli õppeinfosüsteem, s.a.). Kursusel kasutatakse läbivalt programmeerimiskeelt Java (Tartu Ülikool, 2025). Õppeaine läbinud üliõpilane peaks teadma kursusel käsitletavaid mõisteid, nagu klass, isend, pärimine, polümorfism ja abstraktsioon (Tartu Ülikool, 2025). Lisaks, kuidas kasutada levinumaid andmestruktuure, sündmusi, erindeid ja lõimi.

Kursus on mõeldud bakalaureuseõppe õppuritele, kes on läbinud eeldusaine „Programmeerimine“ või „Programmeerimise alused II“ (Tartu Ülikooli õppeinfosüsteem, s.a.). Mõlemad neist on programmeerimise algkursused, milles kasutatakse programmeerimiskeelt Python (Leping jt, 2009; Tartu Ülikool, 2024b, 2024c). Kursuse „Objektorienteeritud programmeerimine“ läbimine on muuhulgas kohustuslik mitmele arvutimängudega seotud ainele, nagu „Arvutimängude loomine ja disain“ ja „Mängumootorid“ ning samuti kursustele, kus kasutatakse Javat või teisi programmeerimiskeeli, nagu „Algoritmid ja andmestruktuurid“ ja „Tarkvaratehnika“ (Tartu Ülikooli õppeinfosüsteem, s.a.).

Kursuse hindamine on eristav skaalal A kuni F, kus A on kõige parem tulemus (Tartu Ülikooli õppeinfosüsteem, s.a.). Kursuselt on võimalik kokku saada maksimaalselt 108 punkti, millest kuus¹ on lisapunktid. Selle läbimiseks peab koguma vähemalt 51 punkti, kusjuures hinde A saab üle 90-ne punktiga. Lõpphinnet kujundavad mitmed tegurid. Loengutele järgnevatest testidest on võimalik koguda maksimaalselt 12 punkti. Kodutööde lahendamine ja praktikumides osalemine annab kokku samuti 12 punkti. Igast kontrolltööst saab teenida kuni

¹ Tartu ülikooli õppeinfosüsteemi järgi saab lisaülesannetest kokku kuni viis punkti. Märkimata on kursuse keskel toimuv vaheküsitlus, mille täitmisel saab veel ühe lisapunkti.

16 punkti. Kursusel toimub kaks viiepunktilist rühmatööd ning teisele järgneb ka kolmepunktiline esitlus. Kõige rohkem punkte annab eksam, mis on väärt 33 punkti.

Kursus kestab kokku 16 nädalat ja selle maht on kuus ainepunkti ehk 156 tundi, millest suurema osa moodustab iseseisev töö (Tartu Ülikooli õppeinfosüsteem, s.a.). Ülejäänud moodustavad arvutipraktikumid, kus lahendatakse vastava nädala harjutusi ning salvestatud videod, kus tutvustatakse erinevaid teemasid (Tartu Ülikool, 2025). Praktikumid ei toimu kontrolltööde või rühmatöö esitlusega samal nädalal (Tartu Ülikool, 2025). Samuti ei ole loenguid kontrolltööde nädalatel.

2.2 Kursuse sisu

Kursuse jooksul läbitakse 12 kodutööd, mis üldjuhul koosnevad erinevate teemade tutvustamisest, koodinäidetest, enesekontrolli testidest ja iseseisvatest ülesannetest (Tartu Ülikool, 2025). Alates 2024. aasta kevadest kehtib uus nõue, mille kohaselt peab esitama iga kodutöö lahendusega logifaili (Tartu Ülikool, 2024a). See ei kehti esimesele kodutööle, kus on vaja enda arvutisse paigaldada programmeerimiskeel Java (Tartu Ülikool, 2025).

Hinnatavad kodutööd algavad teisest nädalast, kus käsitletakse aritmeetilisi tehteid, ümardamist, suvalise arvu genereerimist, meetodite defineerimist, massiive ning nende läbimist (Tartu Ülikool, 2025). Kolmandal nädalal keskendutakse klassi erinevatesse osadesse (isendiväli, konstruktor, `get`- ja `set`-meetodid, meetod `toString`, piiritlejad) ning võrreldakse algtüüpi ja viittüüpi. Sellele järgneb kodutöö, mis on pühendatud sõnade ja tekstifailide töötlemisele ning andmestruktuuri list kasutamisele. Viiendal nädalal käsitletakse liidest, objektide võrdlemist ja vähesel määral iteraatorit. Nädal enne esimest kontrolltööd tutvustatakse pärilust, meetodite ülekatmist, polümorfismi, dünaamilist seostamist ja abstraktseid klasse.

Peale esimest kontrolltööd läbitakse veel kuus peatükki (Tartu Ülikool, 2025). Neist esimesed kaks keskenduvad graafilistele programmidele, mida luuakse JavaFX² teegi ja Maven³ arendusvahendiga. Üheksandal nädalal käsitletakse failisüsteemides navigeerimist, sisend- ja väljundvooge ning nende puhverdamist. Järgmisena tutvustatakse veahaldust, erindeid, püüniseid ja erindiklasside loomist. Üheteistkümnes nädal tutvustab levinumaid

² JavaFX. <https://openjfx.io/>

³ Maven. <https://maven.apache.org/>

andmestruktuure (massiiv, list, kujutus, hulk jt), geneerilisi tüüpe ja meetodeid ning piiratud tüübiparameetreid. Viimane nädal õpitakse lõime kasutamise põhimõtteid.

2.3 Lisamaterjalide vajadus

Kursus „Objektorienteeritud programmeerimine“ (LTAT.03.003) on toimunud üle kümne aasta⁴ ja selle aja jooksul on ilmunud uusi Java versioone, mis täiendavad võimalusi koodi kirjutamiseks (Oracle Corporation, s.a.-a). Näiteks võtmesõna `record`, mida tutvustati 14. versioonis või `switch`-lause, mis on saadaval olnud aastast 2019. Neid muudatusi pole kajastatud praeguses õppematerjalis. Lisaks pole käsitletud kõiki teemasid, mis laiendavad silmaringi ning aitavad paremini aru saada Javast ja programmeerimisest laiemalt. Näiteks regulaaravaldised (Oracle Corporation, s.a.-b), mida saab kasutada tekstitöötlusel.

Sügavamate teadmiste omandamiseks eksisteerib internetis palju lehekülgi, nagu GeeksforGeeks (s.a.) või W3Schools (s.a.) ja seetõttu võib olla algeliste programmeerimise teadmistega üliõpilasel keeruline valida sobilikku materjali. Samuti on paljud nendest inglise keeles, aga kursus toimub eesti keeles, mistõttu peaksid olema leitavad vastavad eestikeelsed materjalid. Lisaks võib veebilehel viidatav lehekülg mittekättesaadavaks muutuda. Liu jt (2022) uurimusest selgus, et programmeerimisega seotud küsimuste ja vastuste platvormil Stack Overflow⁵ ei tööta 14,2% viidatud linkidest. Selliste probleemide vältimiseks on kindlam luua kursusele oma õppematerjale ja mitte toetuda välistele allikatele.

Kursusele pakutava lisaõppematerjalina pole ka tehisintellekt parim informatsiooniallikas, kuna laialt kasutuses olevad mudelid võivad nii-öelda hallutsineerida ehk väljastada informatsiooni, mis on vale (Maleki jt, 2024). Sun jt (2024) ChatGPT dialoogide analüüsist selgus, et vastustes esineb peamiselt fakti-, loogika- ja arutlusvigu ning arvudega seotud ebatäpsusi. Seetõttu pole tehisintellekt usaldusväärne allikas, kust saada informatsiooni. See ei tähenda, et kursusel ei tohiks üldse kasutada tehisaru abivahendeid. Courses'i keskkonnas (Tartu Ülikool, 2025) asuvas teise nädala kodutöös on esile toodud, et tehisintellekti võib kasutada kodu- ja rühmatöodes juhul, kui üliõpilasel jääb probleemi lahendamisega hätta. Siiski on hoiatatud, et sellised abivahendid pole perfektsed ning on keelatud kontrolltööde ja eksami ajal.

⁴ Enne 2017/18. õppeaastat oli kursuse kood MTAT.03.130.

⁵ Stack Overflow. <https://stackoverflow.com>

3. Lisamaterjalide koostamise protsess

Selles peatükis antakse ülevaade lisamaterjalide teemade valikust ja õppematerjalide koostamise protsessist. Lisaks kirjeldatakse peatükkide lõpus olevate enesekontrolli testide loomist.

3.1 Lisamaterjalide valitud teemad

Lisamaterjalide teemade valimine koosneb kahest etapist. Esimesena peab looma loetelu potentsiaalsetest teemadest, mis on tihedalt seotud kursuse materjalidega või on täiesti uued, aga üldteadmisenä kasulikud. Teisena peab jaotama need 12-le nädalale. Selleks on vajalik analüüsida kursuse igat nädalat ja seostada neid koostatud nimekirja teemadega.

Lisamaterjalide kõige olulisem ülesanne on anda uusi teadmisi, kuid on tähtis arvestada ka juba õpetatavate teemadega. Kursuse materjalid on enamasti loodud Java 8 ajal ning järgnevate versioonide uuendusi ja täiendusi pole kajastatud. Java dokumentatsioonis leidub sellistest teemadest loetelu, mis on jaotatud versioonide järgi (Oracle Corporation, s.a.-a). Selles on nimetatud uusi konseptsioone, nagu var-tüüpi muutuja. Samas leidub ka kursusel käsitletavate teemade edasiarendusi, näiteks võtmesõna `sealed` ja lülitidirektiivi teine kuju. Javas eksisteerivad alates versioonist 21 peale tavaliste lõimede ka virtuaalsed lõimed, aga seda nimekirjas ei leidu.

Java dokumentatsioon pole ainus teemade allikas. Kursus „Objektorienteeritud programmeerimine“ on mõeldud algajatele ja seetõttu pole eeldusi, et üliõpilane oskaks programmeerimiskeelt Java. Samuti oletatakse, et kursusel kasutatav arenduskeskkond IntelliJ IDEA on tundmatu, mistõttu eksisteerib Courses keskkonnas IntelliJ ülesseadmise ja kasutamise juhend. Selles jääb mainimata, et IntelliJ's on suur hulk klahvikombinatsioone, mis kiirendavad koodi kirjutamist. Klahvikombinatsioone peaks tutvustama võimalikult vara, et nende kasutamine muutuks harjumuseks. See aeg saabub teisel nädal, mil üliõpilased peavad hakkama arenduskeskkonda kasutama. Sarnane mõttekäik kehtib ka seitsmenda õppenädala teemale ehitusinstrument (ingl *build tool*), mis on vajalik graafiliste programmide loomiseks. Alates 2024/2025. õppeaasta kevadsemestrist soovitatakse kursusel kasutada ehitusinstrumenti Mavenit. Varasemalt on kasutuses olnud Gradle ja lisaks eksisteerib veel Ant.

Potentsiaalseid teemasid leidub ka teistest materjalidest, sest mõnikord selgitatakse kursuse teemasid piiratud mahu tõttu lühidalt. Näiteks viienda nädala kodutöö keskendub muuhulgas

liidesele. Selles on esile toodud liides `Iterable`, mis võimaldab luua indeksita objekte läbivat tsüklit. Materjalis pole kirjeldatud sama liidese meetodit, mis tagastab liidese `Iterator`. Seda saab kasutada väga paljudel andmestruktuuridel elementide läbimiseks ja sellel eksisteerib alamliides `ListIterator`, mis pakub rohkem võimalusi listidele. Seitsmenda nädala praktikumi kolmanda ülesande etteantud koodis kasutatakse võtmesõna `instanceof`. Seda pole mitte kuskil lahti seletatud, aga samas on see objekti tüübi tuvastamiseks tähtis. Kaheksandal nädalal õpitakse, kuidas kasutaja ja graafiline programm suhtlevad läbi sündmuste. Nende kirjutamiseks saab kasutada lambda-avaldisi, aga nendest on toodud vaid üks näide. Lisaks, kodutöö lõpus on vajalik analüüsida graafilise mängu `trips-traps-trull` koodi, mis sisaldab materjalis käsitlemata väärtustikku (ingl *enum*). 11. nädal käsitleb levinumaid andmestruktuure. Kodutöös on nii hulgal kui ka kujutusel näitena toodud üks klass ja loengus on veel paari klassi mainitud, aga nende erinevate meetodite töösse pole süvitsi mindud.

Mõned mitte käsitletavat teemad ei ole eksklusiivsed ainult objektorienteeritud keeltele. Kursuse eeldusained, mis toimuvad Pythonis, käsitlevad rekursiooni ja vähemalt algeliselt regulaaravaldisi. Javas on neil omad nüansid, millega peaks arvestama. Lisaks eksisteerib mõlemas keeles võimalusi koodi automaatseks testimiseks, aga seda ei tutvustata kummaski eeldusaines.

3.2 Lisamaterjalide jaotus

Peale potentsiaalsete lisamaterjalide teemade leidmist on vajalik neid paigutada erinevatele nädalatele. Kasulikud klahvikombinatsioonid, iteraator, võtmesõna `instanceof`, ehitusinstrumendid, lambda-avaldis, väärtustik, andmestruktuurid on eelnevas alapeatükis juba seotud kindlate nädalatega. Teised teemad ei ole veel jaotatud.

Kursuse algul on võimalike teemade valik piiratud, sest Java tutvustamine algab nullist. Esimene nädal ei ole kodutööd ja seetõttu materjali, millega teemasid siduda. Teisel nädalal peaksid üliõpilased olema erinevat tüüpi muutujatega kursis. Seetõttu saab lisamaterjalis tutvustada `var`-tüüpi muutujat, mille tüüp määratakse kompileerides. Kolmanda nädala lõpuks on õpitud, kuidas luua ja kasutada meetodeid ning massiive. Sellest piisab, et tutvustada rekursiooni. Neljandal nädalal on fookuses teksti hoidvad sõned. Need on väga tihedalt seotud kursusel mittekäsitlevate regulaaravaldistega, mille abil leitakse tekstist näiteks kindlaid sümboleid. Viies nädal on esimene kord, kui vaadatakse erilisi klassi tüüpe ja lisamaterjalides saaks tutvustada võtmesõna `record` abil loodavaid klasse. Kuuendal nädalal õpitakse klasside

pärilust ja sellest tulenevalt saab lisamaterjalides käsitleda võtmesõna `sealed`, mis aitab hallata pärilust.

Jätkates kursuse keskpunktist, seitsmenda nädala lisamaterjalides on kaks teemat, millest üks on võtmesõna `instanceof`. Selle kasutusvõimalused laienevad kasutades lülitidirektiivi, mida tutvustatakse lühidalt esimeses praktikumis. Kaheksanda nädala lisamaterjalides käsitletakse lambda-avaldisi, mille mõistmine on tähtis Stream API kasutamiseks. Seega sobib seda tutvustada üheksandal nädalal. Kümnendal nädalal saab tutvustada testide kirjutamist, kuna testimiseks on vaja teada ehitusinstrumentide kasutamist. Kaheksas ja üheksas nädal on juba täidetud, mis jätab testide kirjutamise esimeseks võimaluseks kümnenda nädala. Viimase nädala teemaks on lõimed ja seetõttu sobib lisamaterjalides ideaalselt tutvustada virtuaalseid lõimeid. Lõpuks on lisamaterjalide teemad jaotatud järgnevalt (lisa I):

- 1. nädal: puudub;
- 2. nädal: `var` tüüp ja kasulikud klahvikombinatsioonid IntelliJ's;
- 3. nädal: rekursioon;
- 4. nädal: regulaaravaldised;
- 5. nädal: võtmesõna `record` ja iteraator;
- 6. nädal: võtmesõna `sealed`;
- 7. nädal: lülitidirektiiv, võtmesõna `instanceof` ja ehitusinstrumendid;
- 8. nädal: väärtustik ja lambda-avaldised;
- 9. nädal: Stream API;
- 10. nädal: testide kirjutamine;
- 11. nädal: veel kogumeid;
- 12. nädal: virtuaalsed lõimed.

3.3 Lisamaterjalide loomine

Lisamaterjalide teemad on erinevad ja seetõttu pole ühegi peatüki ülesehitus identne. Siiski on nende üldine struktuur sarnane. Iga uus teema algab sissejuhatava lõiguga, milles seostatakse konkreetse nädala materjali lisamaterjali teemaga. Selleks on tihti esile toodud hüpoteetiline olukord, kus antud teema teadmine aitab probleemi lahendada. Näiteks kuuenda nädala teemaks on klasside pärimine. Sama nädala lisamaterjalides on esile toodud, et klasse saab pärida ja selle peatamiseks kasutatakse võtmesõna `final`. Sellele järgneb kirjeldus probleemist. Antud juhul on see olukord, kus teatud klassid peavad olema pärijad ja ülejäänud

mitte. Lõpuks esitatakse lahendus, milleks on lubatud klasside nimekirju võimaldav võtmesõna `sealed`.

Mitmes lisamaterjalis tutvustatakse klasse või liideseid. Igatühel neist on omad meetodid, mida on vaja lahti seletada. Erinevatel põhjustel pole alati kõiki meetodeid loetletud. Klassi või liidese meetodite arvul pole ülempiiri ja seetõttu võib neid olla kümneid, nagu klassil `Stream`. Mõned meetodid on väga üldised või ei oma suurt tähtsust klassi kasutamiseks. Näiteks väärtustiku teemas oleva klassi `Enum` meetodid `toString`, `hashCode`, `equals` eksisteerivad igal objektil ja nende järjekordne väljatoomine ei anna uusi teadmisi.

Üks olulisemaid ja esimesena valmivaid osasid lisamaterjalidest on koodinäited, mis puuduvad ainult teise nädala teemast: kasulikud klahvikombinatsioonid IntelliJ's. Veel on erandlik ehitusinstrumentide peatükk, kus kasutatakse koodi ainult programmi ülesseadmiseks. Koodinäidete peamine ülesanne on näidata, kuidas rakendada antud teemat. Üldiselt on peatüki algul lihtsamad koodinäited, millele võivad järgneda keerulisemad näited või teiste meetodite kasutamine. Koodilõikude paremaks mõistmiseks on tihti koodi üleval seda lahtiseletav tekst ning koodi all kasutamisel tekkiv väljund või tagastus.

Ühe peatüki loomise protsess algab antud teema uurimisest. Kõige usaldusväärsem allikas on Java dokumentatsioon, aga selles pole alati praktilisi koodinäiteid. Seetõttu on ka teised veebiallikad olulised, kuigi nende sisu võib olla aegunud või lünklik. Piisavate teadmiste omandamisele järgneb algsete koodinäidete loomine. Lisamaterjalide huvitavaks tegemiseks on mõnikord need temaatilised, näiteks viiendal nädalal on osa koodist tasside keskne (joonis 1). Järgmisena valmivad koodi kommenteerivad tekstid ja vajadusel uute meetodite loetelud. Üldjuhul toob see etapp esile koodis olevaid puudusi, nagu liiga väheste uute meetodite kasutamine. Nende kõrvaldamisel tekib tsükliline protsess, kus peab korrigeerima teksti, et see oleks koodiga ühtne tervik ja seejärel jälle üle kontrollima, kas koodis leidub ebaselgeid ridu.

```
record Tass1(int maht, String värv) { }
```

Joonis 1. Kood erilisest `record` klassist `Tass1`. Sellel on vaikimisi loodud isendiväljad `maht` ja `värv`, konstruktor, `get`-meetodid ning meetodid `toString`, `equals` ja `hashCode`.

Mõnes peatükis esineb jooniseid või kuvatõmmiseid. Kümnenenda nädala testide kirjutamise peatükis on viis kuvatõmmist, mis illustreerivad projekti loomist ja testimiseks vajalike

raamistike lisamist. Lisaks on 11. nädalal kaks joonist andmestruktuuride hierarhia kohta, et õppuritel oleks lihtsam mõista kogumite klasside ja liideste vahelisi seoseid.

Peatüki sisulise osa loomisele järgneb juhendaja tagasiside Google Docsi keskkonnas, kus on kommenteerimise võimalus. Peale lisamaterjali parandamist tuleb koostada ka üks enesekontrolli küsimus, millest saab lugeda detailsemalt järgmises alapeatükis. Kõige viimase etapina peab lisamaterjali üle viima Courses'i keskkonda, kus asuvad kõik kursuse kodutööde ja praktikumide materjalid. Google Docsis eksisteerib koodilõikude kuvamise võimalus, aga see pole tavakasutajale kättesaadav (Google, 2022). Alternatiivselt saab kasutada pistikprogramme, nagu Alex Forsythe'i loodud laiendust Code Blocks⁶, millega saab valida programmeerimiskeele ja värvide temaatika ning vastavalt muuta selekteeritud teksti. Courses kasutab enda märgnenduskeelega sisuhaldustarkvara PmWiki⁷. Sellel on sisseehitatud võimalused koodilõikude kujundamiseks, teksti vormindamiseks ning linkide ja piltide kuvamiseks. Seetõttu pole pistikprogrammide kasutamine vajalik.

Kursus „Objektorienteeritud programmeerimine“ toimub eesti keeles, aga Java dokumentatsioon on ingliskeelne. See tähendab, et paljusid mõisteid peab tõlkima. Õigete vastete leidmiseks on kasutatud Cybernetica AS-i (2011) loodud andmekaitse ja infoturbe leksikoni, mis keskendub IT-alastele sõnadele. Klasside, liideste ja meetodite nimesid ei ole tõlgitud, aga mõnikord on nimi ja mõiste sobivas kontekstis piisavalt sarnased, et neid saab kasutada vaheldumisi. Näiteks klass `Vector` ja vektor või liides `List` ja `list`.

3.4 Enesekontrolli küsimuste loomine

Kursuse „Objektorienteeritud programmeerimine“ iga kodutöö lugemismaterjalis on enesekontrollitestedid, mille on loonud Veedla (2022) enda bakalaureusetöö raames. Need on koostatud H5P⁸ raamistikuga E-koolikoti Sisuloome⁹ keskkonnas. Ühtse stiili säilitamiseks on lisamaterjalide enesekontrolli küsimused koostatud samal veebilehel.

E-koolikoti Sisuloome keskkonnas saab luua ülesandeid kümnete erinevate küsimuste tüüpidega. Veedla enda töös toob esile mitme küsimuse (*Quiz (Question Set)*) tüübi

⁶ Code Blocks. https://workspace.google.com/marketplace/app/code_blocks/100740430168

⁷ PmWiki. <https://www.pmwiki.org/wiki/PmWiki/PmWiki>

⁸ H5P. <https://h5p.org/>

⁹ E-koolikoti Sisuloome keskkond. <https://sisuloome.e-koolikott.ee/>

kasutamist, aga sellega loodud alamülesannete konkreetseid tüüpe ei ole kirjeldatud. Seega ei ole alati selge, kuidas ühtset stiili järgida. Näiteks kodutööde enesekontrollides ei ole „Õige“ ja „Vale“ vastusevariantidega küsimuste tüüp õige/vale küsimus (*True/False Question*), vaid valikvastustega küsimus¹⁰ (*Multiple Choice*). Lisamaterjalide enesekontrolli ülesannete tüüpideks kasutatakse peale valikvastustega küsimuste veel sõnade lohistamise (*Drag The Words*) ja lõikude sorteerimise (*Sort the Paragraphs*) tüüpe.

Igal lisamaterjalide teemal on üks enesekontrolli küsimus. 16-st küsimusest suurem osa on valikvastustega küsimused, mille sisuks võib olla näiteks tõesuse määramine, koodilõigu väljastuse või tagastuse arvamine (joonis 2) ning sobivate meetodite või võtmesõnade leidmine. Kahe teema lõpus on sõnade lohistamise ülesanne, milles on ette antud kindel arv sõnu, mida peab paigutama teksti lünkadesse. Üheksanda nädala enesekontrollis peab lõike sorteerima. Igas lõigus on üks meetod ja need peab järjestama vastavalt ülesande tekstile.

Enesekontroll

Mida tagastab rek(2, 4)?

```
public static int rek(int a, int n) {  
    if (n == 0) return 1;  
    return a * rek(a, n - 1);  
}
```

1

4

8

12

16

24

[Reuse](#) [Manus](#) [Teadmistekontroll](#) [H-P](#)

Joonis 2. Kolmanda nädala enesekontrolli küsimus.

¹⁰ Täpselt samaks tulemuseks peab selekteerima käitumisseadetes küsimuse tüübi rippmenüüst valiku „Üks valik (raadionupp)“. Lisaks on oluline välja lülitada samadest seadetest vastuste juhuslikus järjekorras kuvamise.

4. Üliõpilaste tagasiside

Selles peatükis kirjeldatakse lisamaterjalide küsimustiku (lisa II) koostamist, tagasiside kogumist ja lõpuks analüüsitakse tulemust.

4.1 Küsitluse koostamine ja tagasiside kogumine

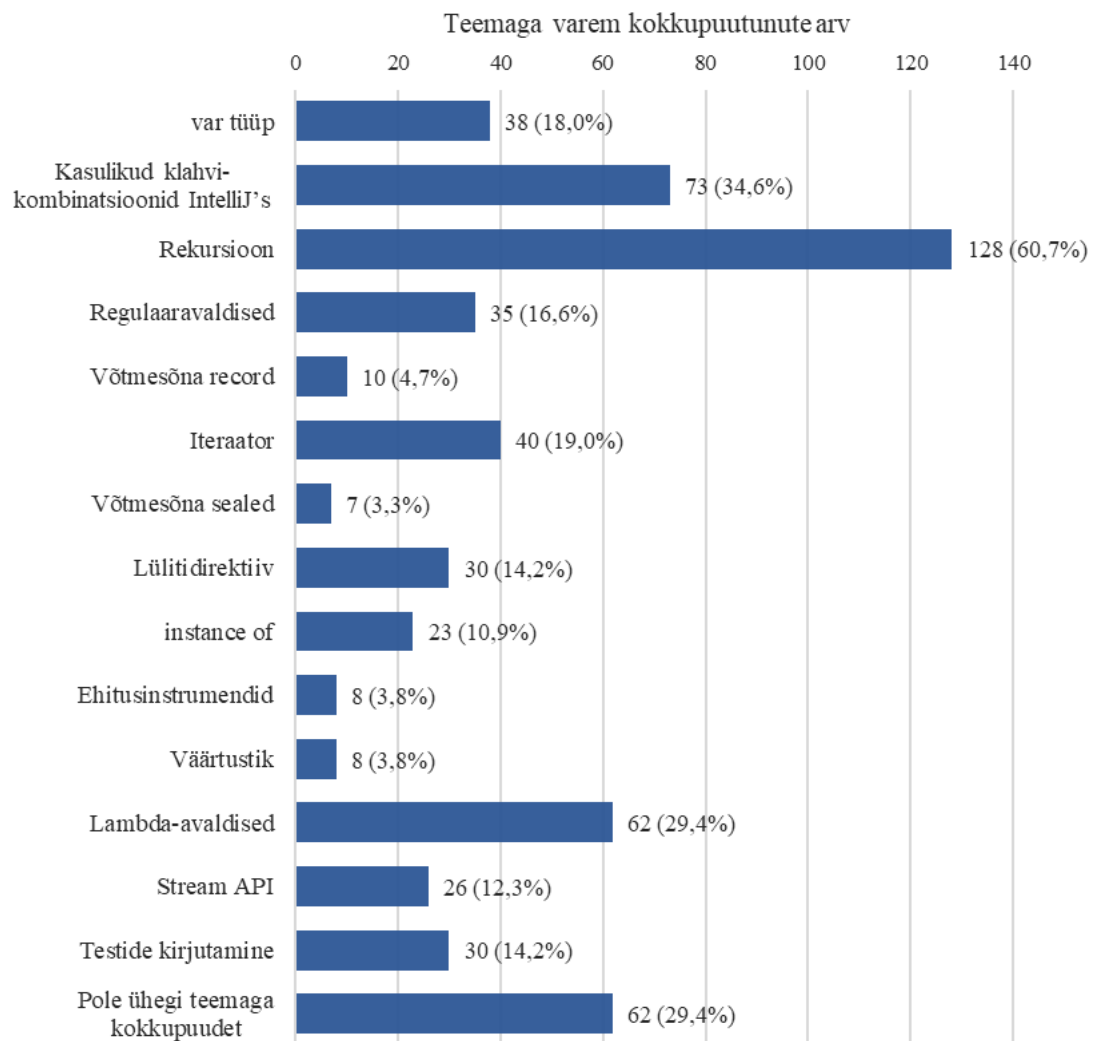
Lisamaterjale kasutati esmakordselt 2024/2025. õppeaasta kevadel. Nende hindamiseks koostati Moodle'i keskkonnas küsimustik, mis oli osa kursuse vabatahtlikust vaheküsitlusest. Selle küsimused hõlmasid AI kasutamist, koduülesandeid, kontrolltöid, praktikume, loenguid, lisamaterjale, ajakasutust ja aine korraldust. Vastamise ligikaudne ajakulu oli 15-20 minutit. See oli üliõpilastele avatud vastamiseks vahemikus 21.-28. aprill ja tagasiside andmisel teenisid nad ühe lisapunkti. Tagasisidet ei küsitud viimase kahe lisamaterjali kohta (veel kogumeid ja virtuaalsed lõimed), kuna vastamisperioodi alguseks olid kursusel käsitlemata andmestruktuurid ja lõimed.

Küsimustikus osaleja pidi valikvastuste hulgast märkima, milliste lisamaterjalide teemadega on neil varasem kokkupuude ning juhul, kui kõik teemad olid tundmatud, sai valida „pole ühegi teemaga kokkupuudet“. Tagasisidet lisamaterjalide kasulikkuse ja arusaadavuse kohta küsiti eraldi ning teemade kaupa Likerti 5-pallisel skaalal. Peale skaala numbrite 1–5 oli veel vastusevariant „0“, mis tähistas „ei ole vaadanud“, kuna lisamaterjalidega tutvumine oli täiesti vabatahtlik. Kui lisamaterjal ei olnud üldse kasulik või arusaadav, võis üliõpilane põhjendada oma arvamust tekstiväljas. Nendele küsimustele järgnes viis üldist väidet lisamaterjalidest ja enesekontrolli testidest, millega nõustumist pidi samuti hindama Likerti 5-palli skaalal (5 – olen täiesti nõus, 1 – ei ole üldse nõus, 0 – ei oska vastata, sest pole vaadanud). Võimalike edasiarenduste väljaselgitamiseks esitati kõige lõpus kaks vabatahtlikku avatud küsimust: millistest teemadest tahaksite rohkem teada ja milliseid teemasid võiks veel olla lisamaterjalides.

4.2 Tagasiside analüüs

Kokku vastas küsimustele 211 (66,6%) üliõpilast kursuse 317-st osalejatest. Nendest vastajatest 136 ehk 64,5% olid vähemalt mingi lisamaterjaliga tutvunud ehk küsimuste vastamisel kasutasid vähemalt üks kord väärtusi vahemikus 1–5. Jooniselt 3 on näha, et 60,7% ehk enamus vastajatest olid varem kokku puutunud rekursiooniga, mida käsitleti kursuse eeldusainetes. Veel oli märgatav hulk tutvunud klahvikombinatsioonide (34,6%) ja lambda-

avaldistega (29,4%). Varasemat kogemust oli vähem võtmesõna sealed (3,3%), väärtustike (3,8%) ja ehitusinstrumentidega (3,8%). Vähemalt ühte teemat tundis 70,6% vastajatest.



Joonis 3. Õppurite eelnev kokkupuude lisamaterjalides käsitletud teemadega.

Lisamaterjalide kasulikkuse, arusaadavuse ja teiste väidete hindamiseks kasutati Likerti 5-palli skaalat. Valikut „0“, mida võis valida materjaliga mitte tutvumise korral, ei arvestatud aritmeetilise keskmise ja standardhälve arvutamisel. Kasulikkuse tagasisidet esindavast tabelist 1 selgus, et kõikide teemade keskmine hinnang on üle neutraalse väärtuse „3“ ja valikuvariante „4“ ja „1“ kasutati hindamisel vähe (maksimaalselt vastavalt 9,1% ja 4,2%). Kõige kasulikumaks osutus teise nädala teema kasulikest klahvikombinatsioonidest, mida hindasid 122 ehk 72,1% peatükki vaadanud õppuritest kasulikuks (vastusevariandid „4“ ja „5“). Vähem kasulikuks hinnati viienda nädala teemat võtmesõnast record keskmise hinnanguga 3,04 ja

kolmanda nädala peatükki rekursioonist, kus seda lugenutest 42,2% arvasid, et see ei ole kasulik (vastusevariandid „2“ ja „1“). Üheks madala kasulikkuse põhjuseks oli õppurite sõnul see, et mõnda teemat ei saa väga tihti praktikas kasutada. Kirjutati ka, et Java õppimine pole nende põhifookus ja vajadusel saavad nad ise juurde õppida.

Tabel 1. Tagasiside lisamaterjalide kasulikkuse kohta teemade kaupa. 5 – väga kasulik, 1 – ei ole üldse kasulik, 0 – ei ole vaadanud.

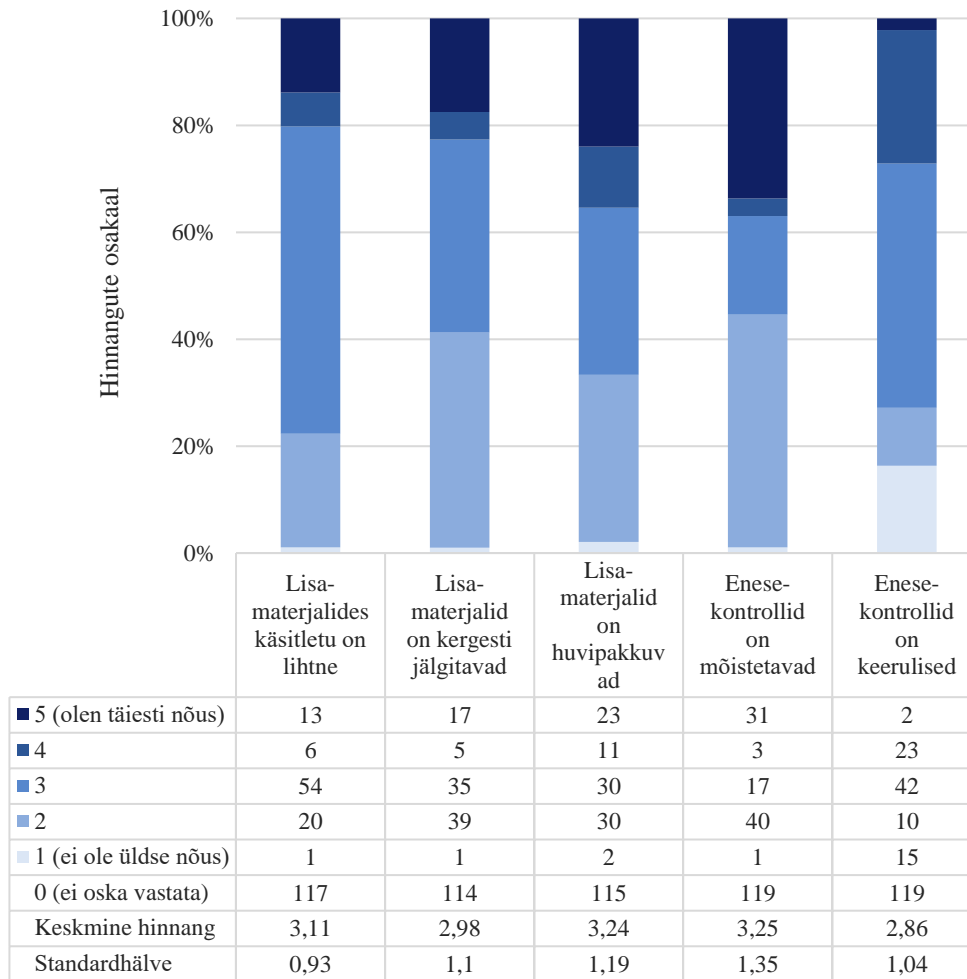
Teema	5	4	3	2	1	0	Keskmine hinnang	Standardhälve
2. nädal. var tüüp	18	3	29	18	3	140	3,21	1,21
2. nädal. Kasulikud klahvikombinatsioonid IntelliJ's	85	3	10	24	0	89	4,22	1,24
3. nädal. Rekursioon	23	2	34	31	2	119	3,14	1,2
4. nädal. Regulaaravaldised	16	2	19	27	0	147	3,11	1,21
5. nädal. Võtmesõna record	10	4	18	16	3	160	3,04	1,2
5. nädal. Iteraator	22	1	22	19	2	145	3,33	1,29
6. nädal. Võtmesõna sealed	13	8	16	13	2	159	3,33	1,22
7. nädal. Lülitidirektiiv	15	6	19	25	1	145	3,14	1,2
7. nädal. instance of	19	3	21	15	1	152	3,41	1,23
7. nädal. Ehitusinstrumendid	15	2	18	15	2	159	3,25	1,27
8. nädal. Väärtustik	13	4	21	12	2	159	3,27	1,19
8. nädal. Lambda-avaldised	38	5	22	20	1	125	3,69	1,29
9. nädal. Stream API	21	1	16	20	1	152	3,36	1,32
10. nädal. Testide kirjutamine	27	0	14	17	2	151	3,55	1,4

Üldiselt üliõpilaste tagasiside lisamaterjalide arusaadavuse kohta (tabel 2) ei erinenud kasulikkuse hinnangust. Enamasti erines arusaadavuse keskmine hinnang kasulikkuse kohta kuni $\pm 0,2$ punkti. Kõige märgatavam muutus toimus kaheksanda nädala teemas lambda-avaldised, mille keskmine hinnang langes 0,4 punkti väärtusele 3,29 (suhtelise muutusega 10,8%). Rekursiooni peatüki keskmine hinnang jäi samaks, aga neutraalse hinnangu „3“ asemel esines rohkem vastuseid „4“ ja „2“. Üheksanda nädala teema kohta Stream API toodi välja, et on raske aru saada, kuidas sisend- ja väljundvood toimivad.

Tabel 2. Tagasiside lisamaterjalide arusaadavuse kohta teemade kaupa. 5 – väga arusaadav, 1 – ei ole üldse arusaadav, 0 – ei ole vaadanud.

Teema	5	4	3	2	1	0	Keskmine hinnang	Standardhälve
2. nädal. var tüüp	24	1	15	27	1	143	3,29	1,35
2. nädal. Kasulikud klahvi-kombinatsioonid IntelliJ's	70	0	11	31	0	99	3,97	1,36
3. nädal. Rekursioon	23	5	24	38	0	121	3,14	1,22
4. nädal. Regulaaravaldised	15	1	17	24	2	152	3,05	1,27
5. nädal. Võtmesõna record	16	1	12	20	2	160	3,18	1,35
5. nädal. Iteraator	21	2	11	26	3	148	3,19	1,4
6. nädal. Võtmesõna sealed	15	1	13	20	2	160	3,14	1,33
7. nädal. Lülitidirektiiv	19	4	15	22	1	150	3,3	1,3
7. nädal. instance of	20	3	13	15	1	159	3,5	1,32
7. nädal. Ehitusinstrumendid	13	3	12	20	1	162	3,14	1,27
8. nädal. Väärtustik	12	1	14	18	3	163	3,02	1,3
8. nädal. Lambda-avaldised	23	5	20	27	1	135	3,29	1,27
9. nädal. Stream API	15	2	15	23	2	154	3,09	1,29
10. nädal. Testide kirjutamine	19	0	18	17	1	156	3,35	1,29

Peale iga teema eraldi hindamist pidid üliõpilased avaldama arvamust lisamaterjalide kohta tervikuna (joonis 4). Tagasisidest selgus, et materjalis käsitletud teemad on pigem lihtsamad, kuigi 57,4% hindasid raskusastet parajaks (vastusevariant „3“). Enesekontrolli küsimuste keerukuse hinnang oli rohkem polariseeritud. 45,7% väite hindajatest olid neutraalsed ja keskmise hinnangu 2,86 järgi on enesekontrollid sobiva raskusega või natuke lihtsad. Lisamaterjalide jälgitavust hinnati keskpäraseks ja enesekontrollile pigem mõistetavaks. Suurem osa (35,4%) väitsid, et lisamaterjalid on huvipakkuvad (vastusevariant „4“ ja „5“) ja 33,3% arvasid vastupidist.



Joonis 4. Üldine tagasiside lisamaterjalide ja enesekontrollitestide kohta.

Küsimustiku viimastes avatud küsimustes said vastajad kirjutada, millistest teemadest nad sooviksid rohkem teada ja milliseid teemasid nad veel tahaks lisamaterjalidesse. Selgus, et rohkem soovitakse teada klahvikombinatsioonidest, võtmesõnast record, võtmesõnast instanceof, lambda-avaldistest ja Stream API-st. Veel võiks üliõpilaste sõnul käsitleda projekti soovituslikku arhitektuuri, Spring Booti, andmete parsimist ja rohkem graafiliste programmide näiteid või lahendusi.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli koostada Tartu Ülikooli kursusele „Objektorienteeritud programmeerimine“ lisamaterjale, mis tutvustavad uusi programmeerimiskeelega Java seotud teemasid. Nende loomisel toetus töö autor peamiselt Java dokumentatsioonile ja olemasolevatele kursuse materjalidele. Lisamaterjalide teemade ja nende esitamise järjekorra valimisel arvestati kursusel juba käsitletavate teemadega. Töö käigus koostati Courses'i keskkonda lisamaterjale 16-st teemast 11-le nädalale. Nendeks teemadeks olid näiteks lambda-avaldised, iteraator, testide kirjutamine ja virtuaalsed lõimed.

Üldiselt loodi iga lisamaterjalide peatükk järgneva ülesehitusega: sissejuhatav lõik, koodinäited ja kõige lõpus enesekontrollitest. Sissejuhatavas lõigus seostatakse konkreetse nädala materjal lisamaterjali teemaga. Koodinäiteid täiendavad selgitavad tekstid ja väljundid või tagastused. Enesekontrollitestidest pidi leidma valikuvariantidest sobivad vastused, otsustama, kas väide on õige või vale, täitma lünkasid, määrama järjekorda. Mõnes peatükis esines ka erinevaid jooniseid ja meetodite loetelusid. Lisamaterjalid leidsid rakendust kursusel „Objektorienteeritud programmeerimine“ esmakordselt 2024/2025. õppeaasta kevadel.

Üliõpilaste tagasiside küsitluse tulemuste järgi on üldiselt lisamaterjalide kasulikkus, arusaadavus, teemade keerukus, jälgitavus, huvipakkuvus üle keskmise taseme. Enesekontrollitestide mõistetavus oli keskmine ja keerukus oli paras. Suuremal osal vastanutest oli varasem kokkupuude vähemalt ühe teemaga.

Bakalaureusetöö käigus valminud lisamaterjalidel on mitu edasiarenemise võimalust. Täiendada saab mitut tagasisides esile toodud teemat, nagu klahvikombinatsioonid, võtmesõna `record`, võtmesõna `instanceof`, lambda-avaldised ja Stream API. Veel võib lisada uusi teemasid, näiteks projekti soovituslik arhitektuur, Spring Boot ja andmete parsimine. Samuti võiks programmeerimiskeele Java uute versioonide väljalaskmisel koostada värskaid lisamaterjale, mis hõlmaksid uudseid teemasid. Lisaks saab kõikide teemade harjutamiseks ja kinnistamiseks koostada ülesandeid ning veel rohkem enesekontrolli teste.

Viidatud kirjandus

- Baweja, A. (2020). Physical Programming Language. *International Journal of Scientific & Technology Research*, 9, 421–424. <https://doi.org/10.6084/m9.figshare.13303448.v3>
- Cybernetica AS. (2011, 2025). *Andmekaitse ja infoturbe portaal*. <https://akit.cyber.ee/>
- Delgado, P., Vargas, C., Ackerman, R., & Salmerón, L. (2018). Don't throw away your printed books: A meta-analysis on the effects of reading media on reading comprehension. *Educational Research Review*, 25, 23–38. <https://doi.org/10.1016/j.edurev.2018.09.003>
- Dodd, A. R., Camacho, G. K., Morocho, E. L., Paredes, F. M., Zuniga, A., Pinza, E. I., Toro, L. V., Vargas, A. B., Benitez, C. D., & Rogers, S. (2015). The Use of Supplementary Materials in English Foreign Language Classes in Ecuadorian Secondary Schools. *English Language Teaching*, 8(9), p187. <https://doi.org/10.5539/elt.v8n9p187>
- Dutta, S., & Mathur, R. (2011). Computer programming—A building block of STEM. *2011 Integrated STEM Education Conference (ISEC)*, 2B-1-2B – 4. <https://doi.org/10.1109/ISECon.2011.6229628>
- Eck, D. J. (2022). *Introduction to Programming Using Java*. Vaadatud 09.05.2025 <https://math.hws.edu/javanotes/>
- Endres, M., Fansher, M., Shah, P., & Weimer, W. (2021). To read or to rotate? Comparing the effects of technical reading training and spatial skills training on novice programming ability. *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 754–766. <https://doi.org/10.1145/3468264.3468583>

- Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The Language of Programming: A Cognitive Perspective. *Trends in Cognitive Sciences*, 23(7), 525–528. <https://doi.org/10.1016/j.tics.2019.04.010>
- GeeksforGeeks. (s.a.). *Java Tutorial*. Vaadatud 23.04.2025
<https://www.geeksforgeeks.org/java/>
- Google. (2022). Easily format and display code in Google Docs. *Google Workspace Updates*. Vaadatud 23.04.2025 <http://workspaceupdates.googleblog.com/2022/12/format-display-code-google-docs.html>
- Hačatrjana, L., & Linde, I. (2023). Piloting Supplementary Materials Aimed at Developing Students' Problem-Solving and Self-Regulated Learning Skills. *International Journal of Learning, Teaching and Educational Research*, 22(6), 475–493.
<https://doi.org/10.26803/ijlter.22.6.25>
- Ho, H.-T., Ly, D.-T., & Nguyen, L. V. (2024). Mitigating Hallucinations in Large Language Models for Educational Application. *2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 1–4. <https://doi.org/10.1109/ICCE-Asia63397.2024.10773965>
- Karki, T. M. (2018). Supplementary Resources Materials in English Language Classrooms: Development and Implementation. *Tribhuvan University Journal*, 32(1), 251–260.
<https://doi.org/10.3126/tuj.v32i1.24791>
- Leping, V., Lepp, M., Niitsoo, M., Tõnisson, E., Vene, V., & VILLEMS, A. (2009). Python prevails. *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, 1–5.
<https://doi.org/10.1145/1731740.1731833>

- Liu, J., Xia, X., Lo, D., Zhang, H., Zou, Y., Hassan, A. E., & Li, S. (2022). Broken External Links on Stack Overflow. *IEEE Transactions on Software Engineering*, 48(9), 3242–3267. <https://doi.org/10.1109/TSE.2021.3086494>
- Liu, Z. (2005). Reading behavior in the digital environment: Changes in reading behavior over the past ten years. *Journal of Documentation*, 61(6), 700–712. <https://doi.org/10.1108/00220410510632040>
- Lyonna, G., Ikhsanudin, I., & Rahmani, E. F. (2023). Developing a Dayak Folk Storybook as Supplementary Material for Teaching Narrative Text. *Acitya: Journal of Teaching and Education*, 5(1), 98–116. <https://doi.org/10.30650/ajte.v5i1.3488>
- Maleki, N., Padmanabhan, B., & Dutta, K. (2024). AI Hallucinations: A Misnomer Worth Clarifying. *2024 IEEE Conference on Artificial Intelligence (CAI)*, 133–138. <https://doi.org/10.1109/CAI59869.2024.00033>
- Mironova, O., Amitan, I., & Vilipõld, J. (2017). Programming Basics for Beginners. Experience of the Institute of Informatics at Tallinn University of Technology. *International Journal of Engineering Pedagogy (iJEP)*, 7(4), 7–18. <https://doi.org/10.3991/ijep.v7i4.7425>
- Oracle Corporation. (s.a.-a). *Java Language Changes*. Vaadatud 23.04.2025 <https://docs.oracle.com/en/java/javase/21/language/java-language-changes-release.html>
- Oracle Corporation. (s.a.-b). *Package java.util.regex*. Vaadatud 09.05.2025 <https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/regex/package-summary.html>

- Pae, H. K. (2020). The Impact of Digital Text. H. K. Pae, *Script Effects as the Hidden Drive of the Mind, Cognition, and Culture* (Kd 21, lk 209–217). Springer International Publishing. https://doi.org/10.1007/978-3-030-55152-0_11
- Pop, M., & Salzberg, S. L. (2015). Use and mis-use of supplementary material in science publications. *BMC Bioinformatics*, *16*(1), 237, s12859-015-0668-z. <https://doi.org/10.1186/s12859-015-0668-z>
- Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K. J., Proulx, V., & Carlisle, M. (2006). Tools for teaching introductory programming: What works? *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, 560–561. <https://doi.org/10.1145/1121341.1121514>
- Ramírez-Echeverry, J. J., Restrepo-Calle, F., & Jiménez, S. T. (2025). Self-Regulated Learning Strategies in Computer Programming Education. *European Journal of Education*, *60*(1), e70052. <https://doi.org/10.1111/ejed.70052>
- Sharmin, N., Houshyar, S., Stevenson, T. R., & Chow, A. K. (2025). Interactive Engagement with Self-Paced Learning Content in a Didactic Course. *Healthcare Informatics Research*, *31*(1), 96–106. <https://doi.org/10.4258/hir.2025.31.1.96>
- Shin, J., Cho, J., & Lee, S. (2021). Tactile-Color System for Accessibility of Color Education: 2.5D UV Printed Supplementary Material for Visually Impaired Students. *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 1–7. <https://doi.org/10.1109/IMCOM51814.2021.9377433>
- Sun, Y., Sheng, D., Zhou, Z., & Wu, Y. (2024). AI hallucination: Towards a comprehensive classification of distorted information in artificial intelligence-generated content.

Humanities and Social Sciences Communications, 11(1), 1278.

<https://doi.org/10.1057/s41599-024-03811-x>

Tartu Ülikool. (2024a). *Objektorienteeritud programmeerimine õppeaastal 2023/24*.

Vaadatud 29.04.2025 <https://courses.cs.ut.ee/2024/OOP>

Tartu Ülikool. (2024b). *Programmeerimine õppeaastal 2024/25*. Vaadatud 23.04.2025

<https://courses.cs.ut.ee/2024/programmeerimine>

Tartu Ülikool. (2024c). *Programmeerimise alused II õppeaastal 2024/25*. Vaadatud

23.04.2025 <https://courses.cs.ut.ee/2025/progalused2/spring>

Tartu Ülikool. (2025). *Objektorienteeritud programmeerimine õppeaastal 2024/25*. Vaadatud

23.04.2025 <https://courses.cs.ut.ee/2025/OOP>

Tartu Ülikooli õppeinfosüsteem. (s.a.). *LTAT.03.003 Objektorienteeritud programmeerimine*

(6 EAP). Vaadatud 23.04.2025 <https://ois2.ut.ee/#/courses/LTAT.03.003>

Veedla, M. (2022). *Enesekontrollitestide koostamine Tartu Ülikooli kursusele*

„Objektorienteeritud programmeerimine“ [Bakalaureusetöö, Tartu Ülikool].

Vaadatud 23.04.2025

https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=74538&year=2022

W3Schools. (s.a.). *Java Tutorial*. Vaadatud 23.04.2025 <https://www.w3schools.com/java/>

Lisad

I. Koostatud lisamaterjalid

2. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa2>
3. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa3>
4. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa4>
5. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa5>
6. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa6>
7. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa7>
8. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa8>
9. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa9>
10. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa10>
11. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa11>
12. nädala lisamaterjal: <https://courses.cs.ut.ee/2025/OOP/Main/Lisa12>

II. Koostatud küsimustik kursusel osalejatele tagasiside kogumiseks

(6/7) Lisamaterjalid

6.1. Milliste lisamaterjalide teemadega olete varem kokku puutunud?

- var tüüp
- kasulikud klahvikombinatsioonid IntelliJ's
- rekursioon
- regulaaravaldised
- võtmesõna record
- iteraator
- võtmesõna sealed
- lülitidirektiiv
- instance of
- ehitusinstrumendid
- väärtustik
- lambda-avaldised
- Stream API
- testide kirjutamine
- pole ühegi teemaga kokku puudet

6.2. Hinda lisamaterjalide kasulikkust teemade järgi.

6.2.1. 2. nädal. var tüüp

O (5) väga kasulik O (4) O (3) O (2) O (1) ei ole üldse kasulik O (0) ei ole vaadanud

6.2.2. 2. nädal. Kasulikud klahvikombinatsioonid IntelliJ's

O (5) väga kasulik O (4) O (3) O (2) O (1) ei ole üldse kasulik O (0) ei ole vaadanud

6.2.3. 3. nädal. Rekursioon

O (5) väga kasulik O (4) O (3) O (2) O (1) ei ole üldse kasulik O (0) ei ole vaadanud

6.2.4. 4. nädal. Regulaaravaldised

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.5. 5. nädal. Võtmesõna record

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.6. 5. nädal. Iteraator

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.7. 6. nädal. Võtmesõna sealed

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.8. 7. nädal. Lülitidirektiiv

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.9. 7. nädal. instance of

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.10. 7. nädal. Ehitusinstrumendid

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.11. 8. nädal. Väärtustik

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.12. 8. nädal. Lambda-avaldised

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.13. 9. nädal. Stream API

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.14. 10. nädal. Testide kirjutamine

(5) väga kasulik (4) (3) (2) (1) ei ole üldse kasulik (0) ei ole vaadanud

6.2.15. Kui vastasite, et mingi lisamaterjal ei olnud kasulik, siis palun täpsustage miks.

6.3. Hinda lisamaterjalide arusaadavust teemade järgi.

6.3.1. 2. nädal. var tüüp

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.2. 2. nädal. Kasulikud klahvikombinatsioonid IntelliJ's

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.3. 3. nädal. Rekursioon

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.4. 4. nädal. Regulaaravaldised

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.5. 5. nädal. Võtmesõna record

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.6. 5. nädal. Iteraator

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.7. 6. nädal. Võtmesõna sealed

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.8. 7. nädal. Lülitidirektiiv

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.9. 7. nädal. instance of

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.10. 7. nädal. Ehitusinstrumendid

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.11. 8. nädal. Väärtustik

O (5) väga arusaadav O (4) O (3) O (2) O (1) ei ole üldse arusaadav O (0) ei ole vaadanud

6.3.12. 8. nädal. Lambda-avaldised

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.13. 9. nädal. Stream API

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.14. 10. nädal. Testide kirjutamine

(5) väga arusaadav (4) (3) (2) (1) ei ole üldse arusaadav (0) ei ole vaadanud

6.3.15. Kui vastasite, et mingi lisamaterjal ei olnud arusaadav, siis palun täpsustage mis.

6.4. Hinda lisamaterjale tervikuna.

6.4.1. Lisamaterjalides käsitletu on lihtne.

(5) olen täiesti nõus (4) (3) (2) (1) ei ole üldse nõus (0) ei oska vastata, sest pole vaadanud

6.4.2. Lisamaterjalid on kergesti jälgitavad.

(5) olen täiesti nõus (4) (3) (2) (1) ei ole üldse nõus (0) ei oska vastata, sest pole vaadanud

6.4.3. Lisamaterjalid on huvipakkuvad.

(5) olen täiesti nõus (4) (3) (2) (1) ei ole üldse nõus (0) ei oska vastata, sest pole vaadanud

6.4.4. Enesekontrolli testid on mõistetavad.

O (5) olen täiesti nõus O (4) O (3) O (2) O (1) ei ole üldse nõus O (0) ei oska vastata,
sest pole vaadanud

6.4.5. Enesekontrolli testid on keerulised.

O (5) olen täiesti nõus O (4) O (3) O (2) O (1) ei ole üldse nõus O (0) ei oska vastata,
sest pole vaadanud

6.5. Millistest temadest tahaksite rohkem teada?

6.6. Milliseid teemasid võiks veel olla lisamaterjalides?

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Marcus Vessel,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Lisamaterjali loomine kursuse „Objektorienteeritud programmeerimine“ jaoks, mille juhendaja on Marina Lepp, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons litsentsiga CC BY NC ND 4.0, 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.

Marcus Vessel
13.05.2025