

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Liisa-Lotte Pehter
Õppematerjalide koostamine ingliskeelsele kursusele
„Objektorienteeritud programmeerimine“
Bakalaureusetöö (9 EAP)

Juhendaja:
Ljubov Jaanuska, PhD

Tartu 2025

Õppematerjalide koostamine ingliskeelsele kursusele „Objektorienteeritud programmeerimine“

Lühikokkuvõte:

Bakalaureusetöö eesmärk oli koostada ingliskeelsed õppematerjalid kahele õppemoodulile kursusel „Objektorienteeritud programmeerimine“. Õppemoodul sisaldab nii lugemismaterjale, õppevideoid, eneseteste, kodutöid kui ka praktikumi ülesandeid. Kursus on mõeldud võõrkeelt kõnelevatele üliõpilastele, kes soovivad õppida objektorienteeritud programmeerimise põhitõdesid. Õppematerjalide väljatöötamisel võeti aluseks ADDIE mudel, mis pakub struktureeritud lähenemist õppematerjalide loomiseks. Materjalide koostamisel tutvuti olemasolevate programmeerimise kursustega ning arvestati varasemates uurimustes leitud teadmiste ja soovitustega. Valminud materjalide sobivust hindasid praktikumijuhendajad ning saadud tagasiside põhjal tehti kursuses muudatusi.

Võtmesõnad: objektorienteeritud programmeerimine, lugemismaterjalid, õppevideod, koduülesanded, praktikumi ülesanded, enesetestid, ADDIE mudel, ümberpööratud õpe

CERCS: P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

Creating Study Materials for the “Object-Oriented Programming” Course in English

Abstract:

The aim of this bachelor’s thesis was to develop study materials for two learning modules in the course “Object-Oriented Programming” in English. Each module includes reading materials, learning videos, self-assessment tests, homework assignments and practical exercises. The course is intended for foreign-language speaking students who wish to learn the fundamentals of object-oriented programming. The development of the study materials was based on the ADDIE model, which provides a structured approach to instructional design. Existing programming courses were reviewed and recommendations from previous studies were taken into account. The appropriateness of the created materials was evaluated by lab instructors and the course was further modified based on their feedback.

Keywords: object-oriented programming, reading materials, learning videos, homework assignments, practice materials, self-assessment tests, ADDIE model, flipped classroom

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

Sisukord

Sissejuhatus.....	6
1. Õppedisaini mudeli valik	7
2. Analüüs	9
2.1 Ingliskeelse kursuse eesmärgid ja sihtrühm.....	9
2.2 Kursus „LTAT.03.003 Objektorienteeritud programmeerimine“	10
2.3 Kursus „LTAT.03.025 Programmeerimine keeles C++“	11
2.4 Kursus „LTAT.06.001 Operatsioonisüsteemid“	11
2.5 Kursuste võrdlus	12
3. Õppematerjalide disain	15
3.1 Valitud õppemeetod	15
3.2 Lugemismaterjalide disain	15
3.3 Videomaterjalide disain	17
3.4 Enesetestide disain	19
3.5 Kodutööde ja praktikumi ülesannete disain	20
4. Õppematerjalide väljatöötamine	22
4.1 Lugemismaterjalide väljatöötamine	22
4.2 Videomaterjalide väljatöötamine	22
4.3 Enesetestide väljatöötamine	23
4.4 Kodutöö ja praktikumi ülesannete väljatöötamine	24
5. Õppematerjalide rakendamine	26
6. Hinnang õppematerjalidele	27
6.1 Tagasiside küsimustike loomine	27
6.2 Lugemismaterjalide ja õppevideote tagasiside analüüs	28
6.3 Enesetestide tagasiside analüüs.....	30
6.4 Kodutööde tagasiside analüüs.....	31
6.5 Praktikumi ülesannete tagasiside analüüs	33
Kokkuvõte.....	35
Viidatud kirjandus.....	36
Lisad.....	40
I. Lugemismaterjalid, õppevideod ja enesetestid	40
II. Koduülesanded	41
III. Praktikumi ülesanded	42

IV. Tehisintellekti kasutamise näide faili sisu genereerimiseks	43
V. Tagasiside küsimustikud	44
Litsents	45

Sissejuhatus

Programmeerimisoskus muutub iga aastaga üha olulisemaks. Seetõttu tuleb programmeerimise õppimise võimalus tagada kõigile üliõpilastele, kes seda soovivad. Objektorienteeritud programmeerimine on programmeerimise paradigma, kus keskendutakse objektidele, mida arendaja soovib mõjutada, mitte üksnes mõjutamiseks vajaminevale loogikale [1]. Rakendades objektorienteeritud programmeerimise põhitõdesid on võimalik koodi taaskasutada ja selle tõhusust maksimeerida. Tartu Ülikoolis õpetatakse juba eesti keeles kursust „LTAT.03.003 Objektorienteeritud programmeerimine“, kuid võõrkeelt kõnelevatel üliõpilastel on selle läbimine keeruline.

Bakalaureusetöö eesmärk on koostada õppematerjalid uuele ingliskeelsele kursusele „Objektorienteeritud programmeerimine“. Kuna õppematerjalide koostamine on aeganõudev protsess, keskendutakse bakalaureusetöö raames kahele nädalale, millel materjalid puuduvad. Valitud nädalate teemad on klassid ja objektid ning sõned, sisend-/väljundvood ja listid. Mõlema nädala jaoks koostatakse järgmised õppematerjalid: lugemismaterjalid, õppevideod, enesetestid, kodutööd ja praktikumi ülesanded. Materjalide loomisel kasutatakse ADDIE mudelit ning läbitakse kõik selle etapid.

Bakalaureusetöö koosneb kuuest peatükist ja nende alapeatükkidest. Esimeses peatükis tutvustatakse põhjalikult ADDIE mudelit. Teises peatükis keskendutakse ADDIE mudeli esimesele etapile ehk analüüsile. Tutvustatakse olemasolevaid kursuseid ning võrreldakse neid. Kolmandas peatükis kirjeldatakse õppematerjalide disainimise ehk kavandamise protsessi. Neljandas peatükis keskendutakse materjalide tegelikule koostamisele ning viiendas peatükis nende rakendamisele. Kuuendas peatükis analüüsitakse kursuse „LTAT.03.003 Objektorienteeritud programmeerimine“ praktikumijuhendajate tagasisidet ja hinnangut loodud õppematerjalidele. Lisades on esitatud veebilingid koostatud materjalidele, näide tehisintellekti kasutamisest failide sisu genereerimiseks ning veebilingid tagasiside küsimustikele.

Tekstide sõnastuse parandamiseks kasutati ChatGPT-d¹. Tehisintellekti abil parandati tekstide loetavust nii õppematerjalides kui ka bakalaureusetöö teoreetilises osas.

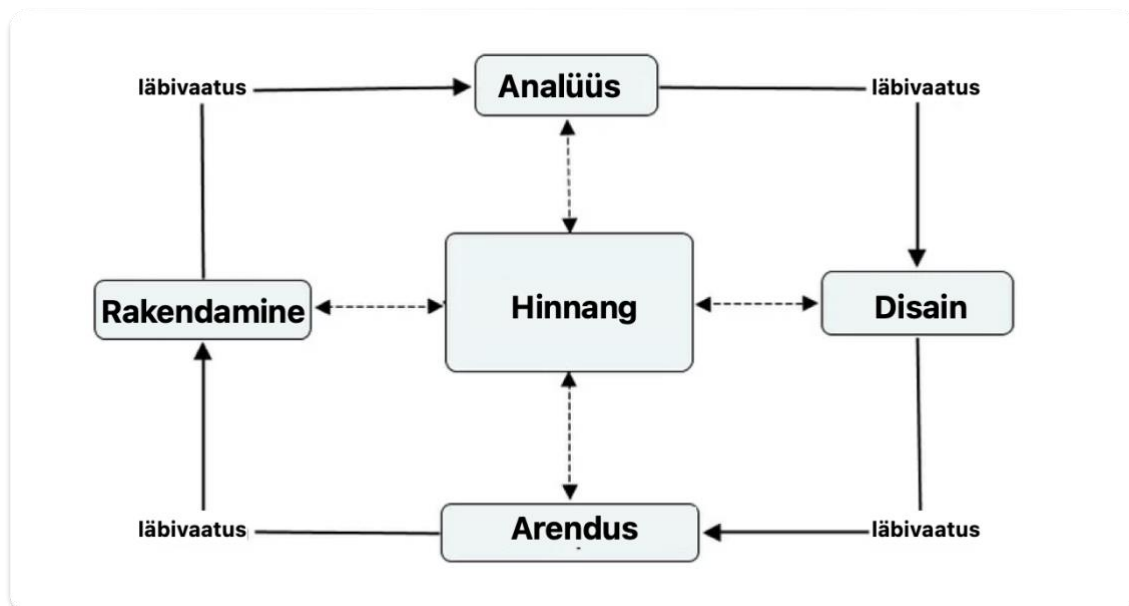
¹ Lõputöö autor on kasutanud tekstide sõnastuse parandamiseks ChatGPT tasuta versiooni. ChatGPT on välja töötatud OpenAI poolt. Lisateave ChatGPT ja OpenAI kohta on aadressil <https://openai.com> (versioon GPT-4; 05.05.2025).

1. Õppedisaini mudeli valik

Enne õppematerjalide koostamist tutvus autor olemasolevate õppematerjalide väljatöötamise mudelitega, nagu ADDIE, Dick ja Carey, Merrilli ning SAMR [2]. Valituks osutus ADDIE mudel, mille eesmärk on aidata kaasa efektiivsete ning tõhusate materjalide koostamisele [3].

ADDIE mudeli kasutamine sobib bakalaureusetöö raames loodavate õppematerjalide väljatöötamiseks, sest see on justkui standardmudel, mis võimaldab efektiivselt koostada erinevaid õppematerjale [4]. ADDIE mudel on paindlik ning selle rakendamist on võimalik kohandada vastavalt projektile. Samuti on selle mudeli kasutamisel võimalik igas etapis materjale täiustada vastavalt tagasisidele, mis aitab luua võimalikult kvaliteetsed materjalid.

ADDIE mudel koosneb viiest etapist: esimene etapp on analüüs, teine on kavandamine ehk disain, kolmas on väljatöötamine, neljas rakendamine ja viies on hinnang mudelile (ingl *analysis* → *design* → *development* → *implementation* → *evaluation*) [5]. Viimast punkti ehk hinnangut mudelile peaks rakendama ka teistes etappides, seetõttu on see joonisel 1 keskmisel positsioonil.



Joonis 1. ADDIE mudeli etapid [5].

Järgnevalt kirjeldatakse ADDIE mudeli etappe lähemalt tuginedes *Digital Learning Institute* käsitlusele [5]. Mudeli analüüsi faasis uuritakse peamiselt, mis on õppematerjalide eesmärk ja miks on neid vaja, millisele sihtrühmale materjale luuakse ning milliseid materjale on juba loodud, mida saaks kasutada. Disaini faasis määratakse, kuidas materjale hakatakse looma,

millised on õpiväljundid, milliseid ülesandeid kasutatakse, et hinnata üliõpilaste arusaama materjalidest ning milline võiks olla materjalide ülesehitus. Väljatöötamise või arenduse etapis toimub õppematerjalide loomine. Rakendamise faasis antakse materjalid üliõpilastele kasutamiseks. Viimases etapis kogutakse õpilaste tagasisidet ning vastavalt sellele viiakse sisse muudatused.

Järgmistes peatükkides käsitletakse ingliskeelse kursuse „Objektorienteeritud programmeerimine“ õppematerjalide väljatöötamist ADDIE mudeli etappide kaupa.

2. Analüüs

ADDIE mudeli esimene etapp on analüüs. Kuna kursust ei ole veel loodud, on autoril vabad käed selle disainimisel. Seetõttu oli enne uute õppematerjalide loomist oluline määratleda kursuse eesmärgid ja sihtrühm. Lisaks analüüsiti erinevaid programmeerimise kursuseid ning nendega seotud tagasisidet, et saada ülevaade olemasolevatest lahendustest. Nendest kursustest koguti inspiratsiooni õppematerjalide koostamiseks. Analüüsitavad kursused valiti vastavalt kursuse ülesehitusele ning jälgiti, et kasutatud oleks mitmekesiseid õppemeetodeid nagu ümberpööratud klassiruum, klassikalised loengud ja videoloengud, kodutööd, praktikumid ning enesetestid. Samuti valiti kursused, mis olid üliõpilaste poolt kõrgelt hinnatud. Tagasisidet käsitletakse peatükis 2.5.

2.1 Ingliskeelse kursuse eesmärgid ja sihtrühm

Tartu Ülikooli arvutiteaduste instituut pakub üliõpilastele, kelle peeriala ei ole informaatika või kes ei ole instituudi üliõpilased, kahte ingliskeelset programmeerimise kursust: „Programmeerimise alused“² ja „Programmeerimise alused II“³. Mõlemad kursused on üles ehitatud selleks, et tutvustada programmeerimiskeele Python põhitõdesid. Sageli aga soovivad üliõpilased pärast nende kursuste läbimist programmeerimist edasi õppida, kuid seni ei ole loodud ingliskeelset jätkukursust⁴.

Samal ajal astuvad Tartu Ülikooli arvutiteaduse instituudi informaatika ja andmeteaduse magistriõppesse ka paljud välisüliõpilased, kelle programmeerimisoskused võivad oluliselt varieeruda, sõltuvalt nende varasemast haridusest. Seetõttu on ka programmeerimisoskuste ühtlustamiseks vajalik luua ingliskeelne programmeerimise kursus⁴.

Bakalaureusetöö juhendaja sõnul peavad väljatöötatava objektorienteeritud programmeerimise (OOP) kursuse eesmärgid olema kooskõlas eestikeelse samanimelise ainega [6]:

- 1) anda õppijatele alustadmised objektorienteeritud programmeerimise põhimõtetest;
- 2) arendada oskust iseseisvalt programme koostada;
- 3) toetada esmaseid rühmatööoskusi.

² Kursuse veebileht asub aadressil: <https://courses.cs.ut.ee/2024/itp/fall> (09.05.2025).

³ Kursuse veebileht asub aadressil: <https://courses.cs.ut.ee/2024/itpII/fall> (09.05.2025).

⁴ Informatsioon on saadud bakalaureusetöö juhendajalt Ljubov Jaanuskalt.

Kuna antud bakalaureusetöö maht ei võimalda kogu kursuse sisu täielikult katta, keskendub käesolev töö kahe õppemooduli väljatöötamisele. Nende moodulite teemadeks on klassid ja objektid ning sõned, sisend-/väljundvood ja listid.

2.2 Kursus „LTAT.03.003 Objektorienteeritud programmeerimine“

Kursuse „LTAT.03.003 Objektorienteeritud programmeerimine“ eesmärk on „anda alusteadmised objektorienteeritud programmeerimise eripärast, oskused programmide koostamiseks ning esmased rühmatööoskused“ [6]. Kursuse läbinud üliõpilane mõistab objektorienteeritud programmeerimise põhikontseptsioone ning oskab analüüsida ja luua vastavaid programme. Ta tunneb erinevaid andmestruktuure, rakendusteeke ja sündmuspõhise programmeerimise põhimõtteid ning oskab neid praktiliselt kasutada. Lisaks omab kursuse läbinu kogemust rühmaprojektis töötamisest ja programmeerimisülesannete lahendamisest integreeritud arenduskeskkonnas IntelliJ IDEA⁵ [6].

Kursus kestab kokku 16 nädalat, millest viimasel on eksam [6]. Positiivse hinde saamiseks on vajalik koguda vähemalt 51 punkti ning saavutada nõutud lävendid [7]. Punkte on võimalik koguda praktikumide, kontrolltööde, loengute, rühmatööde, lisäülesannete ja eksami eest. Kursuse mõneks läbitavaks teemaks on klassid, objektid, konstruktorid ja isendid, sõnetöötlus, sündmuste töötlemine ja veel paljud teised objektorienteeritud programmeerimises olulisel kohal olevad teemad [6].

Kursus on üles ehitatud ümberpööratud klassiruumi meetodil. Õppeaine materjalid koosnevad videoloengutest, lugemismaterjalidest, kodutöödest ning praktikumi materjalidest [6]. Videoloengud ja testid asuvad Moodle'is, kuid nendele vastavad slaidid on ka kursuse veebilehel olemas [6]. Loengumaterjalid on jaotatud mitmeks osaks. Näiteks on kolmandal nädalal, mil teemaks on „Klassid, isendid, konstruktorid, piiritlejad“, kokku neli erinevat videot. Õpilastel on võimalus video asemel tutvuda ka ainult slaididega või kasutada seda võimalust olukorras, kus on vajalik kiiresti mõnd asja slaididelt üle kontrollida.

Kursuse kodutööd on samal lehel, kus on ka vastava nädala lugemismaterjalid [6]. Õppematerjalides on lisaks seletustele ka koodinäited uue informatsiooni edastamiseks. Koodinäited on kommenteeritud, et üliõpilastel oleks lihtsam mõista, mida erinevad käsud ja koodiread teevad. Materjalides on lisaks kohustuslikele ülesannetele ka vabatahtlikud

⁵ IntelliJ IDEA on integreeritud arenduskeskkond, mille on loonud JetBrains. Veebilink IntelliJ IDEA avalehele: <https://www.jetbrains.com/idea/> (12.05.2025).

ülesanded ning enesetestid. Viimased kontrollivad üliõpilaste arusaama õpetavast teemast ning kontrollivad, kas üliõpilane on materjalidest õigesti aru saanud. Kohustuslikud ülesanded põhinevad õppematerjalidel ning võimalus on kasutada ka murelahendajat. Praktikumi-ülesanded on koduülesannetest natukene keerulisemad ning kinnistavad juba saadud teadmisi.

2.3 Kursus „LTAT.03.025 Programmeerimine keeles C++“

Õppeaine „Programmeerimine keeles C++“ eesmärgiks on „anda alusteadmised programmeerimiseks C++ keeles ja tutvustada programmeerimiseks vajalike vahenditega.“ [8]. Kursus on üles ehitatud ümberpööratud klassiruumi meetodil [9]. Mitmete õpiväljundite seas on näiteks, et üliõpilane on tutvunud programmeerimiskeelega C++ ning saab aru objektorienteeritud programmeerimise põhitõdedest.

Kursus kestab kokku 16 nädalat [9]. Aine läbimiseks on vaja lahendada praktikumi- ja kodutöid, sooritada kontrolltööd ning läbida projekti lävend, milleks on viis punkti. Võrreldes kursusega „Objektorienteeritud programmeerimine“, kus igal nädalal on uus teema, läbitakse osad teemad kahe nädalaga, näiteks on teisel ja kolmandal nädalal teemadeks „Keele põhikonstruktsioonid I“ ja „Keele põhikonstruktsioonid II“.

Kursuse materjalid on jagatud neljaks: üliõpilane saab läbida lugemismaterjalid, osaleda loengus, lahendada kodutöö ja praktikumi harjutused ning olemas on ka videolingid [9]. Lugemismaterjalid koosnevad kirjeldusest, millised teadmised õpilane tänu materjalidele saab, sisukorrast, koodinäidetest ning koodi seletustest ja põhjendustest. Mõnes peatükis on ka lisatud lingid sisukamale selgitusele ning kasutusvõimalustele. Õppematerjalide lõpus on teema kohta ka enesetestid.

Kodutööd on eraldi lehel ning nende raskusaste kasvab jooksvalt - esimesed ülesanded on lihtsamad ning viimased on juba rohkem proovile panevad [9]. Harjutused sisaldavad lühikest kirjeldust, mida õpilane tegema peab ning milline võiks lahenduse väljund olla. Osad kirjeldused sisaldavad ka vihjeid ning kui õpilane peab lahenduse jaoks uurima ka muid materjale, siis on ülesandes nendele juhatav link välja toodud. Ka praktikumi materjalid on eraldi lehel ning põhinevad õppematerjalidele.

2.4 Kursus „LTAT.06.001 Operatsioonisüsteemid“

Tartu Ülikooli õppeinfosüsteemis on kursuse „Operatsioonisüsteemid“ eesmärk sõnastatud järgmiselt: „Kursuse läbinu mõistab, mis ja milleks on operatsioonisüsteem ning kuidas see

töötab. Kursuse läbinu on paigaldanud ja skriptinud vähemasti Windows- ja Unix-tüüpi opsüsteeme.“ [10]. Õpiväljundite seas on näiteks, et üliõpilane mõistab, mis on mäluhaldus ja virtuaalmälu ning õpib erinevate operatsioonisüsteemide paigaldamist ning haldamist [10].

Kursuse veebilehte uurides selgub, et aine õpetamisel ei kasutata ümberpööratud klassiruumi meetodit, vaid õpe toimub traditsioonilisel viisil: kõigepealt toimub kontaktloeng, kus räägitakse vastava nädala materjalidest ning tutvustatakse praktikumi sisu, sellele järgneb Moodle test ja praktikum [11]. Õppeaine kestab kokku 16 nädalat [10]. Üliõpilastel on võimalik punkte koguda praktikumi ülesannete, Moodle testide ja eksami eest [11]. Praktikumi ülesannete lahendamiseks ja esitamiseks on aega kaks nädalat pärast praktikumi toimumist.

Kursuse materjalid on jagatud kolmeks: loengud, praktikumi ülesanded ja Moodle testid [11]. Loengutest tehakse ka otseülekanne ning salvestatakse, et üliõpilased saaksid neid hiljem järgi vaadata. Samuti on võimalik tutvuda ka loengu slaididega. Loengu slaidid sisaldavad erinevaid näiteid, mis aitavad käsitletavaid teemasid paremini mõista.

Selle kursuse puhul on kodutööd võrdsed praktikumi ülesannetega [11]. Tutvudes praktikumide sisuga selgus, et materjalid sisaldavad õpetlikku teksti, mille abil saab üliõpilane ülesandeid lahendada. Praktikumi alguses on lühike selgitus, mida vastavas praktikumis õpitakse. Materjalid sisaldavad koodinäiteid/käsklusi, vihjeid ning teemat illustreerivaid pilte. Koodinäidete ning vihjete kasutamine lõputöö raames koostavates materjalides võib tulla kasuks, et aidata üliõpilastel raskeid teemasid paremini mõista.

2.5 Kursuste võrdlus

Tabelis 1 on välja toodud analüüsitud kursuste võrdlus. Andmeid koguti vastavate kursuste veebilehtedelt. Kõik analüüsitud kursused on eestikeelsed ning mahult võrdsed. Võrreldes kursustel kasutatud õpimeetodeid selgus, et kursused „Objektorienteeritud programmeerimine” [6] ja „Programmeerimine keeles C++“ [9] kasutavad mõlemad ümberpööratud klassiruumi meetodit. Kursus „Operatsioonisüsteemid“ [11] kasutab aga klassikalist õpetamise meetodit, kus üliõpilased saavad osaleda loengutes ning kasutada saadud teadmisi praktikumis ja Moodle testides. Esimese kahe kursuse materjalide hulka kuuluvad lugemismaterjalid, kodutööd, praktikumi harjutused ning ka videoloengud (kursusel „Programmeerimine keeles C++” on ka kontaktloengud).

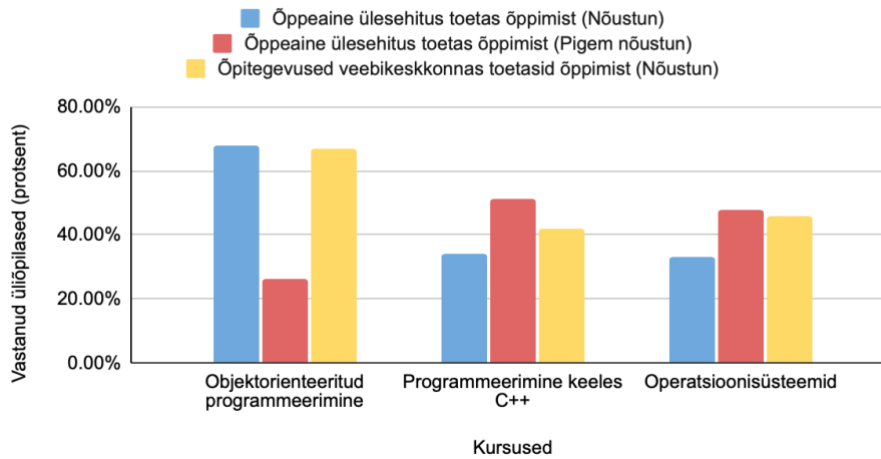
Tabel 1. Analüüsitud kursuste võrdlus.

Kursus	LTAT.03.003 Objekt-orienteeritud programmeerimine	LTAT.03.025 Programmeerimine keeles C++	LTAT.06.001 Operatsioonisüsteemid
Keel	eestikeelne	eestikeelne	eestikeelne
Maht	6 EAP, 16 nädalat	6 EAP, 16 nädalat	6 EAP, 16 nädalat
Kasutatav õpimeetod	ümberpööratud klassiruum	ümberpööratud klassiruum	klassikaline õpetamismeetod
Lugemismaterjalid või tekstilised selgitused	✓	✓	✓
Õppevideod või videoloengud	✓	✓ (lisaks ka kontaktloengud)	kontaktloengud
Enesetestid lugemismaterjalides	✓	✓	–
Kodutööd	✓	✓	võrdsed praktikumi ülesannetega
Praktikumi ülesanded	✓	✓	✓
Koodinäited õppematerjalides	✓	✓	✓
Joonised või illustreerivad pildid	✓	✓	✓

Kursusel „Operatsioonisüsteemid“ [11] on kasutusel lugemismaterjalid ning kontaktloengud, mida salvestatakse. Samuti peavad üliõpilased lahendama praktikumi ülesandeid, mis on võrdsed kodutöödega. Õppeainetes „Objektorienteeritud programmeerimine“ [6] ja „Programmeerimine keeles C++“ [9] on lugemismaterjalides olemas ka enesetestid. Kõigi analüüsitud kursuste õppematerjalid sisaldavad koodinäiteid ning teemat illustreerivaid jooniseid ja pilte.

Järgmisena võrreldi kursuste tagasisidet õppeaastal 2023/2024. Joonis 2 koostamiseks kasutati Tartu Ülikooli õppeinfosüsteemis olevaid andmeid.

Kursuste tagasiside õppeaastal 2023/2024



Joonis 2. Kursuste tagasiside õppeaastal 2023/2024.

Kuigi ained „Objektorienteeritud programmeerimine“ ja „Programmeerimine keeles C++“ on sarnase struktuuriga, viitab üliõpilaste tagasiside sellele, et aine ülesehitust ja veebikeskkonnas toimuvate tegevuste kvaliteeti tajutakse erinevalt.

Näiteks selgub tulpdiagrammist, et aines „Objektorienteeritud programmeerimine“ vastas väitele „Õppeaine ülesehitus toetas minu õppimist“ vastusega „Nõustun“ 68 protsenti ja „Pigem nõustun“ 26 protsenti üliõpilastest [12]. Õppeaines „Programmeerimine keeles C++“ vastasid samale väitele „Nõustun“ 34 protsenti üliõpilastest ja „Pigem nõustun“ 51 protsenti üliõpilastest [13]. Veelgi suurem erinevus ilmnes väites „Õpitegevused veebikeskkonnas toetasid minu õppimist aines“, kus kursusel „Objektorienteeritud programmeerimine“ nõustus 67 protsenti üliõpilastest [12] ja kursusel „Programmeerimine keeles C++“ 42 protsenti üliõpilastest [13]. Kui vaadelda kolmandat, struktuurilt oluliselt erinevat ainet „Operatsioonisüsteemid“, selgub, et esimesele väitele vastasid „Nõustun“ 33 protsenti üliõpilastest ja „Pigem nõustun“ 48 protsenti üliõpilastest [14]. Väitele „Õpitegevused veebikeskkonnas toetasid minu õppimist aines“ vastasid 46 protsenti üliõpilastest vastusevariandiga „Nõustun“ [14].

Need tulemused viitavad sellele, et kuigi kursused võivad olla struktuurilt sarnased, mõjutavad õppijate kogemust märkimisväärselt ka teised tegurid, nagu materjalide esitusviis, juhendaja juhendamisstiil või interaktiivsuse tase veebikeskkonnas. Nendele aspektidele pööratakse tähelepanu järgmises peatükis – õppematerjalide disain.

3. Õppematerjalide disain

ADDIE mudeli teine etapp on disain ehk kavandamine. Selles etapis valiti kasutatav õppemeetod ning pandi paika õpiväljundid, õppematerjalide koostamise plaan ja materjalide struktuur. Kuna objektorienteeritud programmeerimise kursustel on vaja läbida kindlad teemad, siis mõlema nädala materjalide peatükkide valimisel võeti eeskuju Tartu Ülikooli kursusest „LTAT.03.003 Objektorienteeritud programmeerimine“ [6]. Samuti arvestati materjalide koostamise planeerimisel eelmises peatükis kirjeldatud analüüsiga.

3.1 Valitud õppemeetod

Lähtudes eelmises peatükis toodud analüüsist, üliõpilaste tagasisidest erinevatele kursustele, ja arendava kursuse spetsiifikast otsustati ehitada kursus ümberpööratud klassiruumi meetodil. Ümberpööratud klassiruumi õpetamise struktuur koosneb praktikumile eelnevatest tegevustest ning saadud teadmiste rakendamisest praktikumis [15]. Loenguteks on varem salvestatud videomaterjalid, mille üliõpilane peab iseseisvalt läbi vaatama. Kontaktõppeks on praktikumid, kus kinnistatakse loengutest ning iseseisvatest materjalidest saadud teadmised läbi arutelude, tagasiside ning uute ülesannete või probleemide lahendamise. Ümberpööratud klassiruumi meetodi kasutamisel on aga oluline, et õpetaja valmistatud materjalid on läbimõeldud ning sobivad vastava teema õpetamiseks [16].

Ümberpööratud klassiruumi meetodi kasutamisel on mitmeid positiivseid külgi: näiteks saavad õpilased valida tempo ja aja, mil materjale läbida [15]. Läbiviidud uuringu [17] tulemustest selgub, et õpilased eelistavad ümberpööratud klassiruumi klassikalisele meetodile ning selle kasutamisel on positiivne efekt programmeerimisega seotud kursustele.

Samas on ümberpööratud klassiruumi meetodi kasutamisel ka negatiivseid külgi: õpilased ei pruugi läbida materjale enne tundi (mistõttu on igal õpilasel erinevad teadmised õpetavast teemast) ning õppematerjalide loomine on õpetajale aeganõudev protsess [16]. Nende probleemide leevendamiseks koostatakse kodu- ja praktikumiülesanded astmelise raskusastmega – nii saab iga õppija liikuda edasi vastavalt oma tasemele ja õpetaja saab paremini tuvastada, kellel on vaja lisatuge.

3.2 Lugemismaterjalide disain

Lugemismaterjalid on hea viis olulise informatsiooni edastamiseks. Ümberpööratud klassiruumi meetodi puhul on võimalik kasutada ühe praktikumi eelse tegevusena õpetaja poolt

koostatud lugemismaterjale [18]. Han ja Klein [18] analüüsisid erinevaid olemasolevaid uuringuid ja leidsid, et üliõpilased eelistavad õppejõu poolt koostatud materjale või struktureeritud lugemisjuhendeid rohkem kui iseseisvat õpikust õppimist. Heiner jt [19] loetlevad faktoreid, millega tuleb lugemismaterjalide koostamisel arvestada:

- 1) pöörata tähelepanu sellele, et lugemismaterjalid sisaldaksid sisu, mida läbitakse ka kontaktõppe tunnis;
- 2) selgitada üliõpilastele, mis on lugemismaterjalide eesmärk ja miks on nende lugemine kasulik;
- 3) vältida ebavajaliku informatsiooni lisamist materjalidesse;
- 4) kasutada teste, mille eest saavad üliõpilased hinde;
- 5) kasutada teste, mis kontrollivad kui tähelepanelikult õpilane materjale luges.

Samuti uuriti, milline on üliõpilaste suhtumine lugemismaterjalidesse, et saada parem arusaam, millistel põhjustel neid läbitakse. Samas uuringus [19] analüüsisid autorid ka üliõpilaste hoiakut seoses praktikumi eelsete lugemisülesannetega. Uuringust selgus, et üliõpilased olid motiveeritud materjale läbima järgmistel põhjustel: hinne materjalide läbimise eest, parem arusaamine läbivatest teemadest, uute teadmiste omandamine ning ettevalmistus kontaktõppe tunniks. Ainult vähesed üliõpilased leidsid, et lugemine ei olnud neile kasulik, kuna materjalid olid liiga keerulised või ei pidanud nad lugemist tõhusaks õppimise viisiks. Nende probleemide lahendamiseks luuakse ingliskeelse kursuse jaoks täiendavalt videomaterjalid, mis aitavad keerulisemaid teemasid visuaalselt selgitada, ning koduülesanded, mis toetavad materjali süvendatud mõistmist.

ADDIE mudeli disaini etapis määratakse õppematerjalide õpiväljundid, et suunata õppematerjalide loomist ja tagada soovitud õpieesmärkide saavutamine. Käesoleva kursuse õpiväljundite, teemade ja alapeatükkide koostamisel võeti eeskujuna Tartu Ülikoolis kõrgelt hinnatud kursusest „LTAT.03.003 Objektorienteeritud programmeerimine“. Selle kursuse ülesehitus pakkus selget ja loogilist struktuuri, mis hõlbustas õppeteemade järjestamist nii, et iga uus alapeatükk tugines eelnevatele ja aitas saavutada soovitud õpiväljundeid.

Esimesena määrati teema „Classes and Objects“ õpiväljundid ning alapeatükid. Mõned selle teema õpiväljundid on järgmised (täielik loetelu on arendatava kursuse veebilehel, mis asub aadressil <https://courses.cs.ut.ee/2025/OOPr/fall>):

- 1) üliõpilane oskab koostada klasse ja objekte;

- 2) üliõpilane oskab koostada konstruktoreid ning mõistab nende olulisust;
- 3) üliõpilane oskab kirjeldada andmetüüpide erinevusi.

Alapeatükid sisaldavad järgmisi teemasid: klassid ja objektid, konstruktorid, massiivid, meetodid, piiritlejad ja andmetüübid.

Teisena töötati välja teema „Strings, I/O Streams and Lists“ õpiväljundid ja alapeatükid. Mõned selle teema õpiväljundid on järgmised (täielik loetelu on arendatava kursuse veebilehel):

- 1) üliõpilane oskab kasutada klassi *String* ning selle konstruktoreid ja meetodeid;
- 2) üliõpilane oskab luua, avada ja muuta faile;
- 3) üliõpilane oskab luua liste ja mõistab nende olulisust.

Alapeatükid sisaldavad järgmisi teemasid: sõned, klass *StringBuilder*, sisend ja väljund vood, failist sisu lugemine ning listid.

Väljatöötavate lugemismaterjalide struktuur järgib eestikeelse samanimelise kursuse moodulite ülesehitust⁶: iga mooduli alguses on kirjas, millised teadmised üliõpilane materjalide läbimisel saab ning mooduli lõpus on enesetestid, mis toetavad õpitu kinnistamist ja võimaldavad õppijal oma arusaamist kontrollida. Lisaks on lugemismaterjalide alguses sisukord, mis aitab üliõpilastel orienteeruda teemade järjestuses. Teoreetiline osa sisaldab koodinäiteid, tabelleid ja illustreerivaid pilte, mis peaks aitama keerukamaid kontseptsioone paremini mõista.

3.3 Videomaterjalide disain

Lõputöö raames otsustati koostada õppevideod, et toetada üliõpilaste õppimist. See otsus langetati uurides mitmeid artikleid [20-21] ning põhinedes nende positiivsele mõjule. Samuti arvestati artiklis [19] välja toodud asjaoluga, et osa üliõpilastest ei pea tekstipõhist õppimist endale sobivaks meetodiks. Long jt poolt läbiviidud uuringus [20] andsid üliõpilased positiivse tagasiside õppevideote kasutamisele kui ühele tõhusale ümberpööratud klassiruumi vahendile. Üliõpilaste sõnul oli videote vaatamine efektiivne, sest videod aitasid neil mõista õpetavaid teemasid ning tekitasid klassiruumis õppimisega sarnase tunde. Veel selgub Boutell'i analüüsist [21], et üliõpilased otsustavad videoloengute kasuks, sest see võimaldab neil

⁶ Kuigi struktuur on inspireeritud eestikeelsest kursusest “LTAT.03.003 Objektorienteeritud programmeerimine”, on moodulite sisu unikaalne ja loodud spetsiaalselt ingliskeelse sihtrühma vajadusi silmas pidades.

iseseisvalt planeerida kõige sobivam aeg ja koht videote vaatamiseks. Vastupidiselt kontaktõppena toimuvale loengule, on võimalik videod pausile panna ja omandada materjale omas tempos.

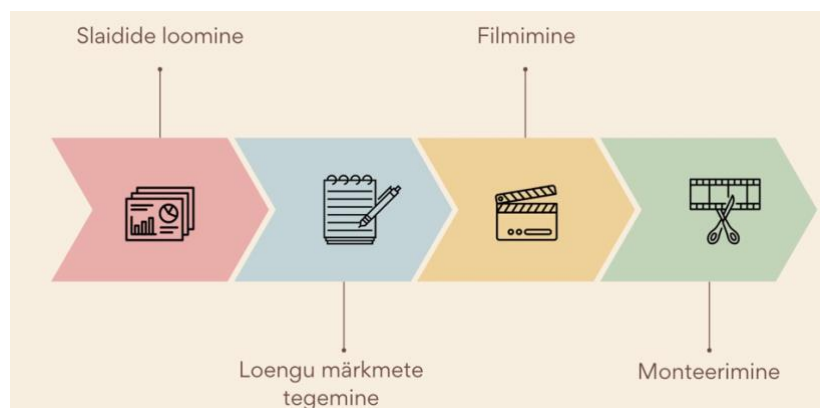
Järgmisena uuriti, kuidas luua võimalikult kvaliteetsed õppevideod. Long jt uuringus [20] otsiti vastust küsimusele, kuidas videoid muuta kaasahaaravamaks. Analüüsist selgub, et videote juures on oluline, et nende pikkus peaks olema maksimaalselt 20 minutit. Veel leiti, et videote koostamisel on oluline, et need valmistaksid üliõpilased tunniks ette, sisaldaksid kõiki õpetavaid teemasid ning oleksid õpilastele kergesti arusaadavad. Guo jt uuringus [22] antakse järgmised soovitusel, kuidas videomaterjale paremini luua:

- 1) videomaterjalide pikkus peab olema lühike, et üliõpilased mõtleksid kogu video vältel kaasa;
- 2) õppejõu entusiasm ja kiire kõne on õpilase jaoks motiveeriv;
- 3) eeltöö videote filmimiseks on väga oluline.

Lähtudes analüüsi etapist, arendava kursuse omapärast ja eelnevalt toodud soovitustest, otsustati luua kokku kuus õppevideot. Nädalale, mille teema on „Classes and Objects“ otsustati koostada järgmiseid teemasid käsitlevad õppevideod vastavalt alapeatükkidele: klassid, objektid ja konstruktorid, meetodid ning piiritlejad ja andmetüübid.

Ka nädalale „Strings, I/O Streams and Lists“ plaaniti koostada kolm videot, mille teemad on: sõned, sisend ja väljund vood ning listid.

Õppevideote loomisel lähtuti joonisel 3 esitatud etappidest.



Joonis 3. Õppevideote koostamise etapid.

Videote salvestamiseks otsustati kasutada arvutisse eelpaigaldatud ekraanisalvestuse rakendust QuickTime Player⁷ ning töötlemiseks videotöötlusrakendust iMovie⁸, sest nende kasutamine on tasuta ning töö autor oli neid varem kasutanud ning jäänud nende võimalustega rahule.

3.4 Enesetestide disain

Disaini etapis tuleb otsustada ja kavandada, milliseid ülesandeid kasutatakse, et hinnata üliõpilaste arusaama materjalidest [5]. Üheks ülesande tüübiks valiti enesetestid. Enese testimine on oluline osa materjalide omandamise juures. See suunab üliõpilasi vastutama õppimise eest, seadma eesmärged ning analüüsima oma oskusi [23]. Programmeerimise kursusel läbi viidud uuringust selgub, et enese testimine aitab üliõpilastel õpitavaid teemasid paremini mõista ning neid on hiljem lihtsam meelde tuletada [24]. Sellel kursusel olid kasutusel näiteks ülesanded, kus üliõpilane pidi valima õige koodi süntaksi või leidma veakohad koodis. Sarnaste ülesannete kasutamine tuleb kasuks ka bakalaureusetöö raames loodava kursuse enesetestide küsimuste koostamisel.

Järgmisena uuriti, milliseid reegleid ja nõuandeid peab valikvastustega küsimuste loomisel järgima. Mõned juhised küsimuste koostamiseks, mille Haladyna jt [25] välja tõid on:

- 1) iga küsimus peaks toetuma kõige olulisemale osale õppematerjalist;
- 2) küsimused ei tohiks üksteisest sõltuda;
- 3) vältima peaks küsimusi, mis nõuavad üliõpilaste arvamust;
- 4) küsimuste tekst peaks olema võimalikult lühike;
- 5) vastusevariantide loomisel on olulised järgmised aspektid: valikus on vähemalt kolm varianti, vastusevariandid ei tohi sõltuda teistest valikutest, valikuvariantide koostamisel tuleks vältida negatiivseid lauseid.

Disaini etapis pandi paika ka enesetestide koostamise plaan:

- 1) lugemismaterjalide läbi lugemine;
- 2) enesetestide küsimuste välja mõtlemine;
- 3) enesetestide koostamine;

⁷ QuickTime Player on Apple'i seadmetele mõeldud rakendus, mille abil saab lisaks videote muutmisele, salvestada ka arvuti ekraani. Juhend QuickTime Playeri kasutamiseks: <https://support.apple.com/en-gb/guide/quicktime-player/welcome/mac> (08.05.2025).

⁸ Apple'i poolt loodud video monteerimise rakendus, mis on mõeldud Mac, iPhone ja iPad seadmetele. Rakenduse abil on võimalik vaadata, lõigata, ühendada ja redigeerida videoklippe ning lisada erinevaid video- ja heliefekte. <https://support.apple.com/en-gb/guide/imovie/mov5ec96da08/mac> (14.04.2025).

- 4) enesetestide lisamine kursuse veebilehele;
- 5) juhendajalt tagasiside küsimine ning vastavalt sellele paranduste tegemine.

Enesetestide loomiseks otsustati kasutada Sisuloome keskkonnas⁹ olevaid H5P malle¹⁰. H5P mallid valiti, sest need on integreeritud ka Tartu Ülikooli Moodle'i keskkonda ning neid saab mugavalt lisada kursuse veebilehele.

Kokku otsustati koostada kaks enesetesti – mõlemale nädalale üks test. Samuti pandi paika, et testid peavad koosnema vähemalt viiest küsimusest, et kontrollida üliõpilaste teadmisi õpetatavast teemast.

3.5 Kodutööde ja praktikumi ülesannete disain

Teiseks viisiks, kuidas üliõpilaste teadmisi kinnistada ja kontrollida valiti kodutööd. Kuna kodutööde ja praktikumi ülesannete disain on sarnane, siis paigutati selle kirjeldus ühte peatükki. Sarnaselt kursusele „LTAT.03.003 Objektorienteeritud programmeerimine“ otsustati luua praktikumi ülesanded natuke keerulisemad kui kodutöö ülesanded [6].

Epstein ja Van Voorhis [26] arutlevad, et mõned põhjused, miks õppejõud annavad kodutöid lahendamiseks on järgmised:

- 1) harjutada tunnis õpitut ning kinnistada saadud teadmisi;
- 2) valmistada õpilasi ette järgmiseks tunniks;
- 3) suurendada õpilaste kaasatust ning rakendada teadmisi praktiliste ülesannetega;
- 4) kasvatada vastutustunnet ning arendada aja planeerimise oskust.

Vatterott [27] väidab, et kodutööde puhul on viis olulist tingimust, mida peab täitma, et luua hea kodutöö. Üheks tingimuseks on, et kodutöödel peab olema kindel ja arusaadav eesmärk, milleks ülesandeid kasutatakse. Teiseks tingimuseks on aga kodutöö tõhusus. Oluline on, et kodutöö oleks kasulik ning vastaks ajale, mis selle lahendamiseks kulub. Veel on oluline, et õpilasel oleks võimalus valida, kuidas ta õpib. Kodutöö peaks olema „kohustus“, mida iga üliõpilane soovib täita. Neljanda punkti mõte on, et tuleb arvestada õpilaste võimetega ja vastavalt sellele määrata kodutöö raskus. Viiendas punktis on välja toodud, et ka kodutöö disain on oluline ning aitab püüda üliõpilaste tähelepanu.

⁹ Sisuloome keskkonnas on võimalik lahendada teiste autorite poolt koostatud teste või koostada ise uusi materjale. Link Sisuloome keskkonna avalehele: <https://sisuloome.e-koolikott.ee/> (31.03.2025).

¹⁰ H5P abil on võimalik luua erinevat tüüpi interaktiivset sisu nagu interaktiivseid videoid, esitlusi või küsimustikke. Link: <https://h5p.org/getting-started> (10.04.2025).

Kodutööde puhul on veel oluline motiveerida üliõpilasi neid lahendama. Kontur ja Terry uuringust [28] selgub, et kodutöö eest punktide jagamine suurendab kodutöö lahendanud üliõpilaste arvu. Ka Planchard jt analüüsi [29] põhjal on kodutööde eest punktide jagamine motiveeriv aspekt, kuid samast uuringust selgub veel, et üliõpilasi motiveerib ka uute teadmiste omandamine. Oluline on arvestada ka teguritega, mis võivad põhjustada üliõpilastes motivatsiooni langemist. Planchard jt [29] uurisid ka motivatsiooni vähendavaid tegureid. Nende uuringus vastasid üliõpilased, et üks põhjus on näiteks arusaamatus, mida ülesande lahendamiseks tegema peab. Teine ja kolmas põhjus on, et üliõpilastele tundub ülesanne liiga raske või mahukas/pikk ning see vähendab koheselt motiveeritust ülesannet lahendada.

Kavandamise käigus pandi paika ka ülesannete koostamiseks järgmised etapid:

- 1) ülesannete teemade väljamõtlemine;
- 2) esialgsete ülesannete kirjelduse koostamine;
- 3) ülesannete lahendamine ja väljundi näidete lisamine ülesannete kirjeldustesse;
- 4) ülesannete kirjelduse parandamine;
- 5) ülesannete lisamine kursuse veebilehele;
- 6) ülesannete esitamine juhendajale;
- 7) ülesannete parandamine vastavalt juhendaja tagasisidele.

Kokku otsustati koostada kahe nädala peale seitse kodutöö ja üheksa praktikumi ülesannet, mis käsitlevad läbitavaid teemasid mitmekülgelt ja erinevate nurkade alt. Järgmises peatükis kirjeldatakse bakalaureusetöö kõige mahukamat osa ehk õppematerjalide väljatöötamise protsessi.

4. Õppematerjalide väljatöötamine

Õppematerjalide väljatöötamise etapis koostati materjalid teemadele „Classes and Objects“ ning „Strings, I/O Streams and Lists“. Selles etapis arvestati analüüsist selgunud tulemustega ning disaini etapis koostatud plaaniga. Mõlema nädala materjalide loomine koosnes viiest etapist: lugemismaterjalide loomine, koduülesannete loomine, praktikumi materjalide loomine, õppevideote loomine ning enesetestide koostamine. Selles peatükis kirjeldatakse iga etapi läbimist.

4.1 Lugemismaterjalide väljatöötamine

Arendus etapis toimus lugemismaterjalide loomine. Kõigepealt valmisid materjalid teemale „Classes and Objects“ ning seejärel teemale „Strings, I/O Streams and Lists“. Arvestades eelmises etapis selgunud õpiväljundite, teema alapeatükkide ja otsusega kuidas materjale struktureerida, koostati dokument, kuhu lisati lugemismaterjalide esialgne raamistik. Seejärel otsiti veebist erinevaid allikaid, et lisada ka õpetlik tekst. Teoreetilist osa täiendati pidevalt koodinäidetega (Joonis 4), et aidata üliõpilastel mõista teooria rakendamist praktikas.

The "this" keyword can also be used to call one constructor from another within the same class:

```
1 //class that describes fruit and what we can do with them
2 public class Fruit {
3     String name; //attribute that holds fruit name
4     int taste; //attribute that holds fruit's taste rating on a 10-point scale
5
6     public Fruit(String name, int taste) {
7         this.name = name;
8         this.taste = taste;
9     }
10
11     public Fruit() {
12         this("fruit", 0); // use "this" keyword to call out the other constructor
13     }
14 }
```

Joonis 4. Koodinäide lugemismaterjalides.

Valmis lugemismaterjalid lisati kursuse veebilehele (Lisa I). Esialgsed lugemismaterjalid anti juhendajale lugemiseks, et tuvastada puudused ning vead. Vastavalt juhendaja tagasisidele tehti kursuses jooksvalt muudatusi.

4.2 Videomaterjalide väljatöötamine

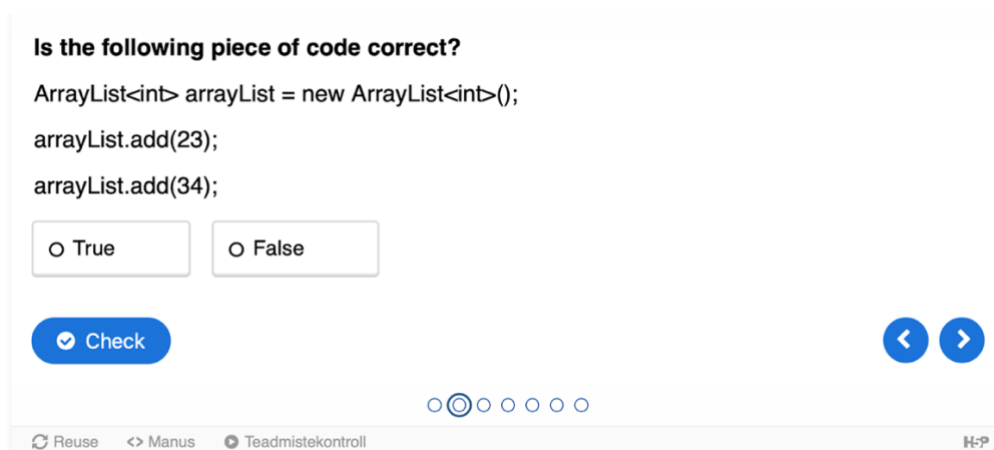
Järgmisena valmisid õppevideod. Videotes läbitakse materjalid kiirelt ning nende pikkus vastab peatükis 3.3 toodud soovitudele: kõige lühem video on alla kolme minuti ning kõige pikem on peaaegu seitse minutit pikk.

Kõigepealt koostati slaidid, mis võtavad kiirelt kokku lugemismaterjalides kirjeldatu ning sisaldavad erinevaid koodinäiteid, lühikesi selgitusi ning tabeleid. Järgmisena koostati loengu märkmed, et filmides ei tekiks kõnes pause ning selgitused oleksid kõigile arusaadavad. Samuti aitas see kiirendada monteerimise protsessi.

Videoid filmiti formaadis, kus üliõpilane näeb slaididel olevat informatsiooni ning kuuleb täpsemaid selgitusi. Salvestamiseks kasutati arvutisse eelpaigaldatud ekraanisalvestuse keskkonda. Viimases etapis toimus õppevideote monteerimine. See koosnes videote lõikamisest ning heli parandamisest, puudu olevate klippide lisamisest ning vajadusel uute videolõikude salvestamisest. Valmis õppevideod lisati kursuse veebilehele (Lisa I) ning juhendaja andis neile tagasisidet. Vastavalt tagasisidele tehti videotes muudatusi. See sisaldas näiteks slaidide muutmist, uute videolõikude filmimist ning monteerimist.

4.3 Enesetestide väljatöötamine

Arvestades disaini faasis paika pandud enesetestide koostamise plaaniga, loeti kõigepealt läbi lugemismaterjalid. Lugemise käigus tehti märkmeid, milliseid küsimusi koostada ja mis teemasid need sisaldama peaks. Järgmisena mõeldi välja enesetestide küsimused. Enesetestides küsitakse näiteks, mida esitatud kood välja prindib, kas esitatud väide on korrektne või kas esitatud koodis on mõni viga (Joonis 5).



Joonis 5. Näide enesetestide küsimusest.

Kolmandas etapis toimus enesetestide koostamine Sisuloome keskkonnas. Peamiselt koostati mitme valiku variandiga küsimusi, kuid kasutati ka küsimusi, kus vastus sisestati vastaja poolt või pidi valima, kas väide on õige või vale. Valmis testid lisati kursuse veebilehele. Viimases etapis küsiti juhendaja arvamust ning vastavalt sellele tehti küsimustes parandusi.

Kokku valmis kaks enesetesti, mõlemale nädalale üks test (Lisa I). Nädala „Classes and Objects“ testis on kuus küsimust ning nädala „Strings, I/O Streams and Lists“ testis on kokku seitse küsimust.

4.4 Kodutöö ja praktikumi ülesannete väljatöötamine

Kodutööde ja praktikumi ülesannete väljatöötamisel läbiti kõik etapid, mis pandi paika disaini etapis. Esiteks mõeldi välja koduülesannete teemad. Teemad valiti nii, et need oleksid üliõpilastele huvitavad ning seotud nädala õppematerjalidega. Arvestati ka sellega, et teemad põhineksid igapäevastel tegevustel ja üliõpilased saaksid ülesandeid seostada igapäevaeluga ning ei peaks kontekstist eraldi aru saama. Näiteks on nädala „Classes and Objects“ koduülesannete läbivaks teemaks puuviljad ning nende müümisega seotud toimingud (Joonis 6).

Exercise 3

Create a new class called `FruitShop`. Define the following class fields:

1. `shopName` (`String`)
2. `shopAddress` (`String`)
3. `openDays` (An array of `String` objects (weekdays should be written as 'MO', 'TU', 'WE', etc))
4. `sellers` (An array of class `Seller` objects)
5. `fruits` (An array of class `Fruit` objects)

All of these fields should have `get` methods. `ShopAddress`, `sellers` and `fruits` fields should also have a `set` method. Add a constructor and a `toString()` method and make the following methods:

1. The `set` method for `fruits` should control if the new array has at least three objects.
2. Instance method for calculating the earnings from one fruit.
 - The method should take two parameters: a class `Fruit` object and the number of products sold (`double`)
 - The method should return a double type number of how much the shop earned from selling the fruit
3. Instance method for identifying if the shop is open:
 - The method should take one parameter: the day the user wants to know if the shop is open (`String`).
 - The return type should be `boolean`, if the shop is open then the method should return true otherwise false.
 - Tip: you can use a loop to go through the `openDays` array and use method `equals()` when comparing strings.
4. An instance method that prints out all of the `sellers` array objects.
 - The method return type should be `void`.
 - Use the `Seller` class's `toString()` method.

Joonis 6. Näide nädala „Classes and Objects“ kolmandast koduülesandest.

Järgmises etapis koostati esialgsed ülesannete kirjeldused. Tekstid sõnastati nii, et üliõpilased saaksid aru, mida nad tegema peavad ning jälgiti, et ülesanded ei tunduks liiga mahukad või pikad. Pärast tekstide koostamist lahendati ülesanded ning lisati näited programmi väljundist. Viimase käigus tehti ka märkmeid, kuidas võiks sõnastust muuta ning seejärel viidi need

muudatused sisse. Valmis ülesanded lisati kursuse veebilehele (Lisa II ja Lisa III) ning esitati juhendajale. Vastavalt juhendaja tagasisidele tehti ülesannetes muudatusi.

Nädala „Strings, I/O Streams and Lists“ kodutöö ja praktikumi ülesannete jaoks koostati kaks tekstifaili, mis olid ülesande lahendamiseks vajalikud. Failide sisu koostati nii, et kõigepealt mõeldi välja, milliseid andmeid peab fail sisaldama ning koostati paar esimest rida. Seejärel kasutati ülejäänud sisu koostamiseks ChatGPT-d¹¹ (Lisa IV) .

¹¹ Failide sisu pärineb ChatGPT-st, keelemudelist, mille väljaõpe põhineb suurel hulgal erinevatel tekstiallikatel. ChatGPT on välja töötatud OpenAI poolt. Lisateavet ChatGPT ja OpenAI kohta leiate aadressilt <https://openai.com> (versioon GPT-4; 22.01.2025).

5. Õppematerjalide rakendamine

Õppematerjalide rakendamine ei olnud võimalik, sest kursus toimub esmakordselt 2025/2026 õppeaasta kevadsemestril. Selle asemel paluti materjale lahendada kursuse „LTAT.03.003 Objektorienteeritud programmeerimine“ praktikumijuhendajatel.

Materjalide efektiivsuse ja sobivuse hindamiseks lisati mõlema nädala lugemismaterjalide, õppevideote ja enesetestide ning kodutööde ja praktikumi ülesannete lõppu link tagasiside küsimustikule. Täpsem kirjeldus tagasiside küsimustike koostamise ja praktikumijuhendajate vastuste kohta on järgmises peatükis.

6. Hinnang õppematerjalidele

ADDIE mudelit kasutades toimub hinnangu andmine igas etapis [5]. Õppematerjale välja töötades hindas autor kõigepealt ise tehtud tööd ning seejärel andis hinnangu ja soovitusel parandusteks ja täiendusteks bakalaureusetöö juhendaja. Viimasena andsid tagasiside kursuse „LTAT.03.003 Objektorienteeritud programmeerimine“ praktikumijuhendajad. Selles peatükis kirjeldatakse, kuidas tagasiside küsimustikke loodi ning millised olid praktikumijuhendajate vastused. Vastavalt saadud tagasisidele ja soovitudele, mille asjakohasust esmalt hinnati, tehti kursuse õppematerjalides muudatusi.

6.1 Tagasiside küsimustike loomine

Tagasiside küsimustike koostamisel lähtuti eesmärgist hinnata bakalaureusetöö raames loodud materjale võimalikult efektiivselt ning tagada, et küsimustike täitmine ei võtaks liigselt aega. Selleks võeti aluseks Sarapi bakalaureusetöös [30] loodud küsimustik, mis tõlgiti inglise keelde ja kohandati vastavalt vajadusele. Kodutööde ja praktikumi ülesannete tagasiside küsimustike koostamisel valiti mõned küsimused ka Zakatovi lõputöös [31] kasutatud küsimustikust. Küsimused vaadati koos juhendajaga üle ning vajadusel tehti parandusi.

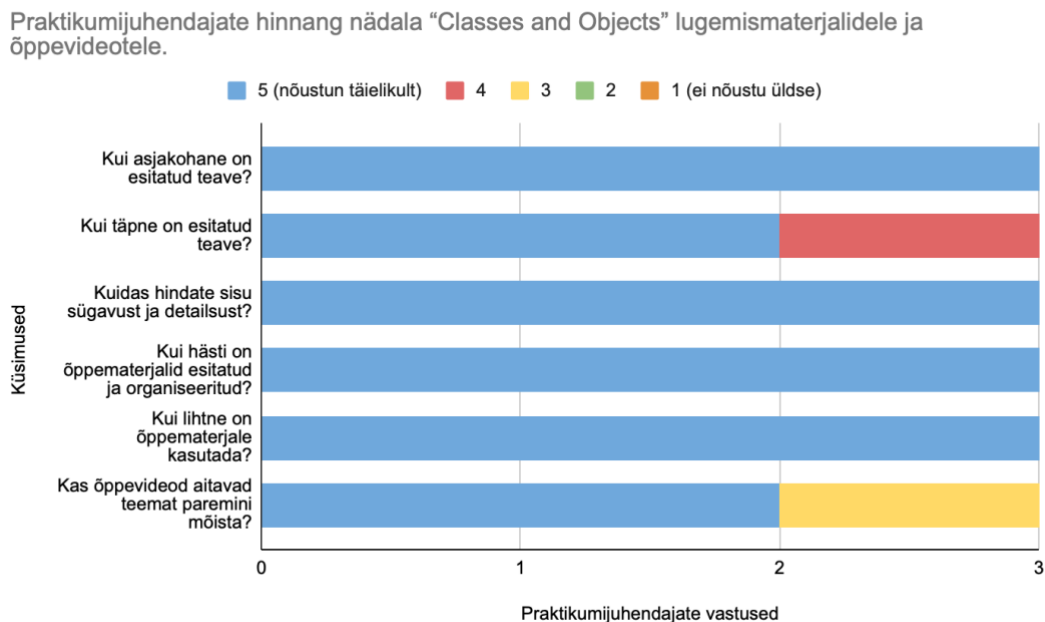
Kokku koostati LimeSurvey¹² keskkonnas kaheksa küsimustikku – eraldi küsimustikud mõlemale teemale ning iga materjalitüübi (lugemismaterjalid ja õppevideod, enesetestid, kodutööd ja praktikumi ülesanded) kohta (Lisa V). Lugemismaterjalide ja õppevideote küsimustikes oli kokku 13 küsimust, kus uuriti arvamust näiteks materjalide sisu, täpsuse ja paigutuse kohta. Enesetestide küsimustikes oli kokku kuus küsimust ning hinnata tuli näiteks, kas enesetestid aitavad materjale paremini mõista või kui keerulised on enesetestide küsimused. Kodutöö ja praktikumi ülesannetele koostati eraldi küsimustikud, kuid nende sisu oli sarnane. Hinnata sai näiteks kodutöö või praktikumi ülesannete raskusastet, arusaadavust ja kui põnev oli neid lahendada.

Tagasisideküsimustele oodati vastuseid Likerti 5-palli skaalal, kus „1“ tähistas mittenõustumist ja „5“ täielikku nõustumist. Iga küsimuse järel oli vastajal võimalus oma vastust täpsustada või täiendada. Küsimustikule vastamine oli vabatahtlik. Uuritavate anonüümsuse säilitamiseks ei kogutud ühtegi identifitseerivat teavet. Vastuseid oodati perioodil 01.04.2025 kuni 27.04.2025.

¹² LimeSurvey on keskkond veebipõhiste küsimustike koostamiseks ja läbiviimiseks. Veebilink: <https://survey.ut.ee/index.php/> (20.03.2025).

6.2 Lugemismaterjalide ja õppevideote tagasiside analüüs

Lugemismaterjalide ja õppevideote küsimustikele vastas kolm praktikumijuhendajat. Nädala „Classes and Objects“ vastused on esitatud joonise 7 abil.



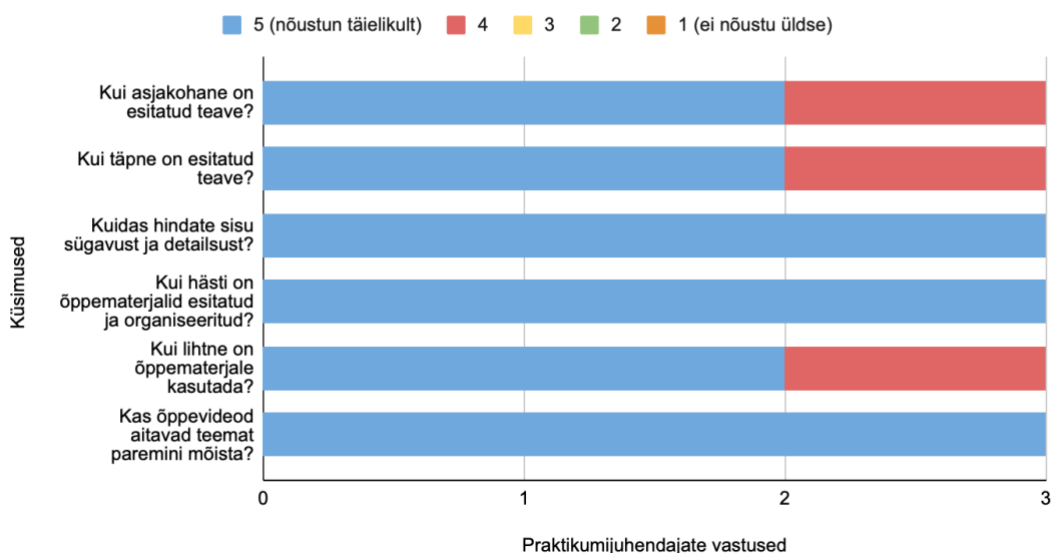
Joonis 7. Praktikumijuhendajate hinnang nädala „Classes and Objects“ lugemismaterjalidele ja õppevideotele.

Joonis 7 põhjal selgub, et praktikumijuhendajate arvates on õppematerjalides esitatud teave väga asjakohane. Samuti toodi välja, et õppevideod selgitavad teemat hästi ja lugemismaterjalid täiendavad videoid ning on kirjutatud selges ja arusaadavas keeles. Esitatud informatsiooni täpsust hinnati samuti kõrgelt. Täpsustati, et esineb grammatilisi vigu ja mõni kasutatav sõna võib olla algaja jaoks segadust tekitav. Kõigi praktikumijuhendajate arvates oli sisu sügavus ja detailsus sobiv. Samuti leiti, et õppematerjalid on väga hästi esitatud ja organiseeritud. Küsimusele kui lihtne on õppematerjale kasutada ja neist vajalikku informatsiooni leida vastasid kõik juhendajad, et nõustuvad sellega täielikult. Lisaks toodi välja, et materjalidesse võiks lisada rohkem värve, et üliõpilased saaksid kiire lugemise korral teksti paremini eristada. Kahe juhendaja hinnangul aitasid õppevideod teemat paremini mõista, kuid üks juhendaja sellega täielikult ei nõustunud. Lisati, et kuigi videod on informatiivsed, oli neid kuulates lihtne tähelepanu kaotada. Positiivsena toodi välja, et lugemismaterjalides ja õppevideotes kasutatud näited ei kordu. Küsimusele, kas mõni teema puudub õppematerjalidest või sai liiga vähe tähelepanu vastasid kaks praktikumijuhendajat, et nende

arvates ei puudu ükski teema. Kolmas praktikumijuhendaja aga lisas, et materjalid võiksid sisaldada veel näiteks selgitusi, kus objektid mälus asuvad ning mis juhtub kasutamata objektidega.

Nädala „Strings, I/O Streams and Lists“ lugemismaterjalide ja õppevideote küsimustike vastused on esitatud joonise 8 abil.

Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists“ lugemismaterjalidele ja õppevideotele.

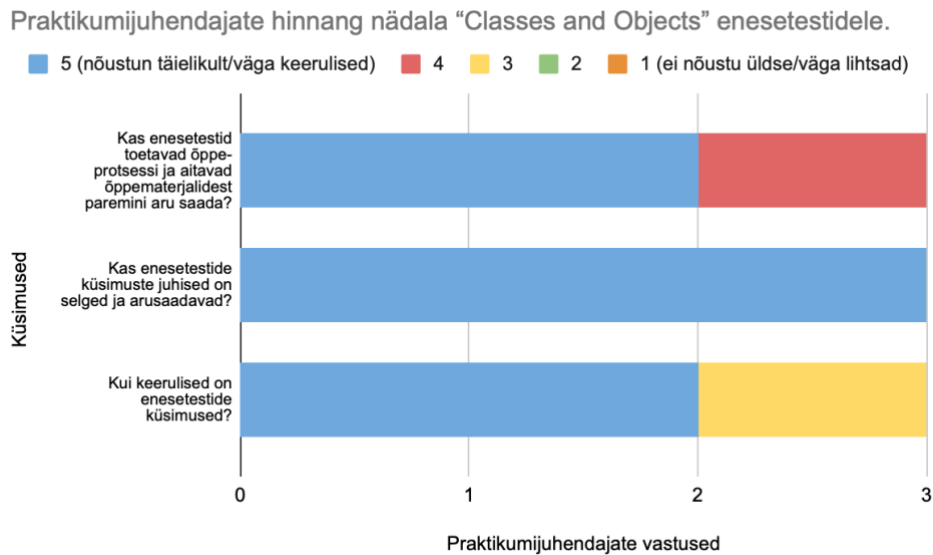


Joonis 8. Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists“ lugemismaterjalidele ja õppevideotele.

Joonis 8 põhjal on näha, et praktikumijuhendajate hinnangul on esitatud teave asjakohane. Ka sisu täpsuse osas leiti, et see on täpne. Samas tehti ettepanek täpsustada lugemismaterjalides failide loomisel erindite käsitlemist. Juhendajate arvates on materjalide sisu väga detailne ning need on hästi esitatud ja struktureeritud. Õppematerjalide kasutamise lihtsuse kohta toodi välja, et tekst võiks sisaldada rohkem värve. Samuti soovitati muuta veebilinkide seadistust nii, et need avaneksid uues aknas. Kõik juhendajad hindasid õppevideoid väga kasulikeks. Küsimusele, kas mõni teema puudub õppematerjalidest või sai liiga vähe tähelepanu vastasid kaks praktikumijuhendajat, et nende hinnangul on kõik vajalikud teemad olemas. Kolmanda praktikumijuhendaja arvates võiks materjalidesse lisada täiendavaid selgitusi teiste failide lugemiseks või kirjutamiseks kasutatavate klasside kohta.

6.3 Enesetestide tagasiside analüüs

Enesetestide mõlema nädala tagasiside küsimustikule vastas kolm praktikumijuhendajat. Nädala „Classes and Objects“ tagasiside küsimustiku vastused on esitatud joonise 9 abil.

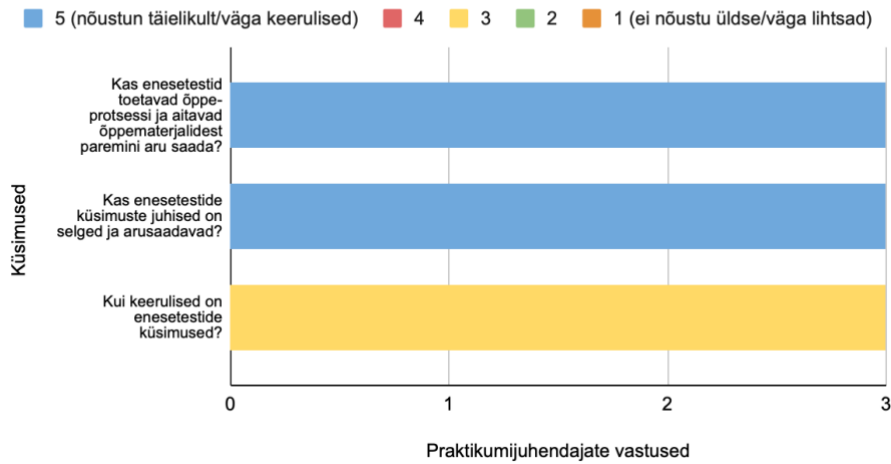


Joonis 9. Praktikumijuhendajate hinnang nädala „Classes and Objects“ enesetestidele.

Joonis 9 näitab, et kaks praktikumijuhendajat nõustusid täielikult, et enesetestid on kasulikud ja aitavad üliõpilastel õppematerjalidest paremini aru saada. Üks juhendaja lisas, et enesetestid küll aitavad aga võiks lisada ka selgitavaid kommentaare, mis aitaksid üliõpilastel mõista, miks nende vastus oli vale. Kõik praktikumijuhendajad nõustusid täielikult, et enesetestide küsimuste juhised on selged ja arusaadavad. Küsimuste raskusastet hinnati järgmiselt: üks praktikumijuhendaja leidis, et ülesannete raskus on keskmine, kuid kaks juhendajat pidasid küsimusi väga keerulisteks.

Nädala „Strings, I/O Streams and Lists“ tagasiside küsimustiku vastused on esitatud joonise 10 abil, millest selgub, et kõigi praktikumijuhendajate hinnangul toetasid ka selle nädala enesetestid õppimist ning aitasid õppematerjalidest paremini aru saada. Samuti nõustusid kõik juhendajad, et enesetestide juhised on selged ja arusaadavad. Eraldi toodi esile, et küsimuste selgitused on väga head.

Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists” enesetestidele



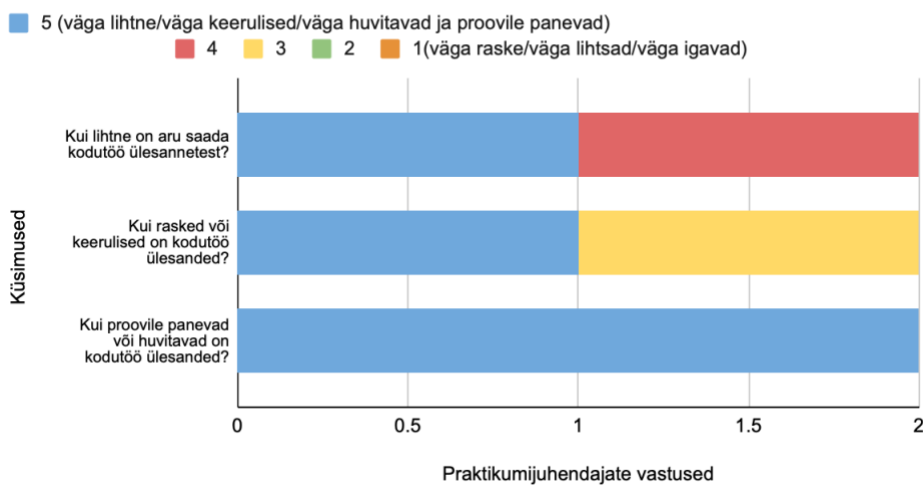
Joonis 10. Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists“ enesetestidele.

Enesetestide raskusastme hindamisel vastasid kõik juhendajad ühtemoodi ning leidsid, et küsimuste raskusaste on keskmisel tasemel.

6.4 Kodutööde tagasiside analüüs

Kodutööde mõlema nädala tagasiside küsimustikule vastas kaks praktikumijuhendajat. Nädala „Classes and Objects“ tagasiside küsimustiku vastused on esitatud joonise 11 abil.

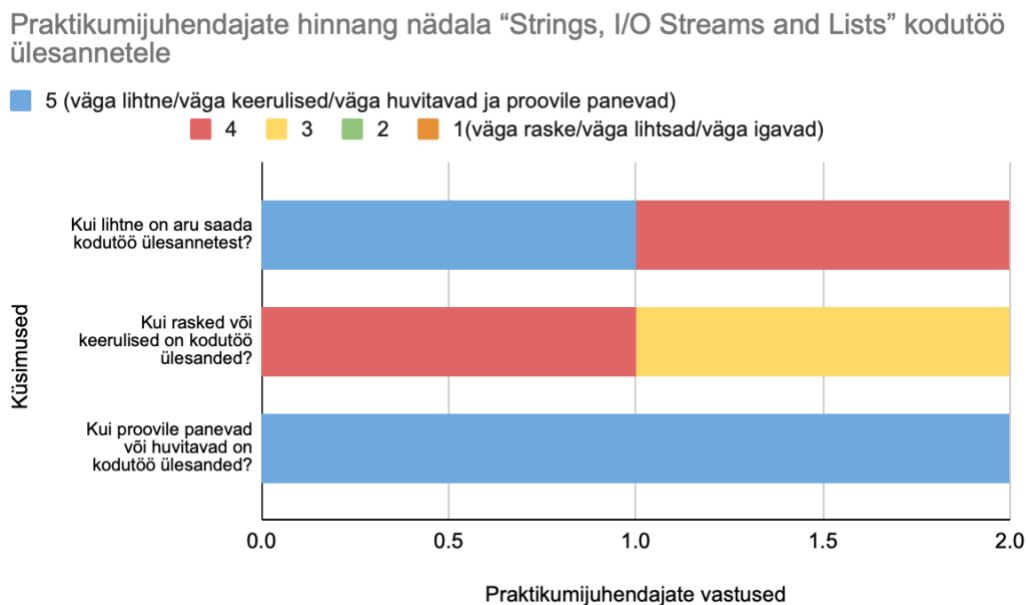
Praktikumijuhendajate hinnang nädala „Classes and Objects” kodutöö ülesannetele



Joonis 11. Praktikumijuhendajate hinnang nädala „Classes and Objects“ kodutöö ülesannetele.

Joonis 11 põhjal selgub, et kodutööde ülesannetest on lihtne aru saada: üks juhendaja hindas neid väga lihtsaks ning teine juhendaja lihtsaks. Veel soovitati lisada juhend, kuidas programmi testida peameetodis, kuna see võib olla keeruline paljudele programmeerimisega alustajatele. Ülesannete keerukuse osas täpsustati, et need ei ole liiga lihtsad ega ka liiga rasked. Lisaks toodi välja, et mitme õpitud teema rakendamine ühes ülesandes on üliõpilastele küll kasulik, kuid võib algajate jaoks tähendada, et nad peavad koodi korduvalt parandama ja sellest vigu otsima. Mõlema praktikumijuhendaja arvates olid ülesanded väga huvitavad ja proovile panevad.

Nädala „Strings, I/O Streams and Lists“ kodutööde tagasiside küsimustiku vastused on esitatud joonise 12 abil.

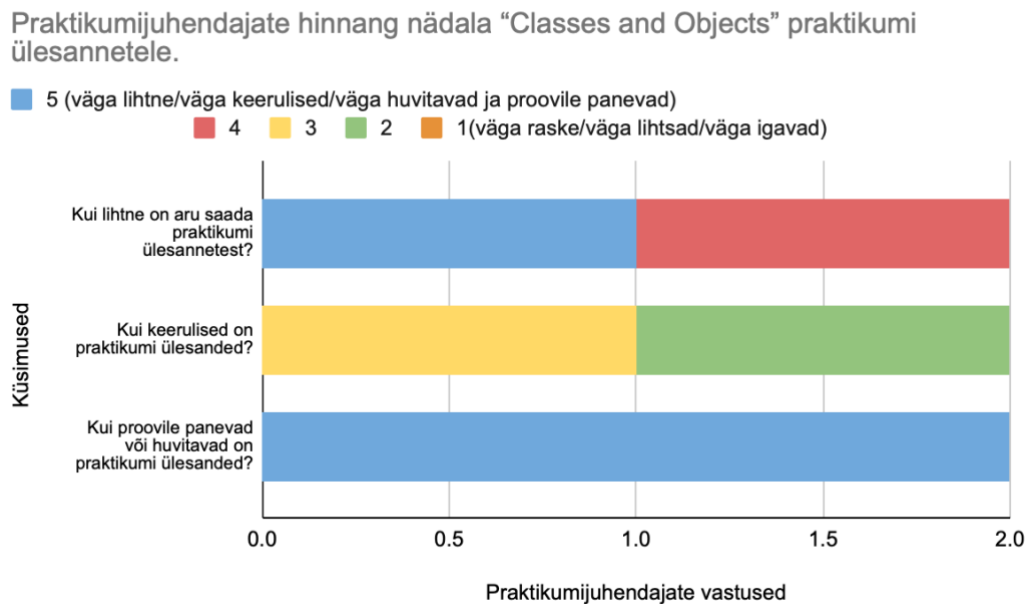


Joonis 12. Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists“ kodutöö ülesannetele.

Joonis 12 näitab, et ka selle nädala puhul leidsid praktikumijuhendajad, et ülesannetest on lihtne aru saada. Samas toodi välja, et ülesannetes on mõned detailid puudu, mis võib algajatele ülesande lahendamise raskemaks teha. Näiteks soovitati täiendada teise ülesande kirjeldust ning lisada täpsem juhised, milliseid sõnu ja kuidas peab üliõpilane listi lisama. Kodutöö ülesannete raskusastet hinnati erinevalt: üks praktikumijuhendaja leidis, et ülesanded on rasked ning teine pidas ülesannete raskusastet keskmiseks. Mõlema praktikumijuhendaja arvates olid ülesanded väga huvitavad ja proovile panevad.

6.5 Praktikumijuhendajate tagasiside analüüs

Mõlema nädala praktikumi ülesannete tagasiside küsimustikule vastas kaks praktikumijuhendajat. Nädala „Classes and Objects“ tagasiside küsimustiku vastused on esitatud joonise 13 abil.

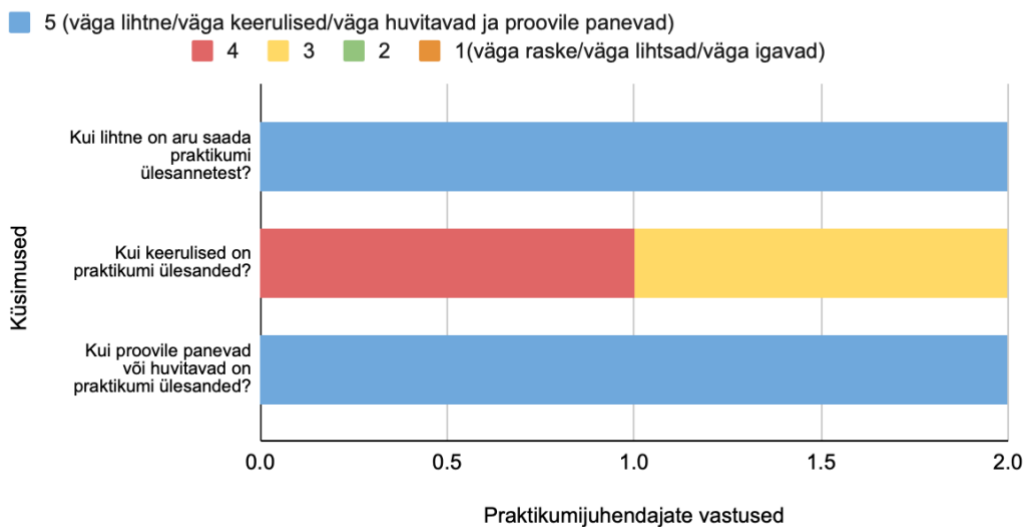


Joonis 13. Praktikumijuhendajate hinnang nädala „Classes and Objects“ praktikumi ülesannetele.

Joonis 13 põhjal selgub, et praktikumijuhendajate arvates on praktikumi ülesannetest lihtne aru saada. Samas soovitati teksti veel täiendada, et ka algajad saaksid ülesandest paremini aru. Praktikumijuhendajate arvates on ülesanded väga huvitavad ja proovile panevad.

Nädala „Strings, I/O Streams and Lists“ tagasiside küsimustiku vastused on esitatud joonise 14 abil.

Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists“ praktikumi ülesannetele.



Joonis 14. Praktikumijuhendajate hinnang nädala „Strings, I/O Streams and Lists“ praktikumi ülesannetele.

Joonis 14 näitab, et mõlema praktikumijuhendaja arvates on ülesannetest väga lihtne aru saada. Praktikumijuhendajate arvates on ülesannete raskusaste nende arvates raske ja keskmine. Lisaks toodi välja, et ülesanded on keerukad, kuid praktikumi juhendaja abiga muutuvad need keskmise raskusastmega ülesanneteks. Ka selle nädala ülesannete puhul vastasid mõlemad praktikumijuhendajad, et ülesanded on väga proovile panevad ja huvitavad.

Tagasisides esitatud soovitusel ja parandused võeti arvesse ning neid käsitleti koostöös bakalaureusetöö juhendajaga. Soovituste põhjal tehti kursuse õppematerjalides järgmised parandused: parandati kirjavead ja asendati algajatele keerulised sõnad lihtsamatega; lugemismaterjalide alapeatükkide pealkirjade kujundust parandati, lisades rohkem värve; eneseteste täiendati selgitustega, mis ilmuvad vale vastuse korral; kodutöö ja praktikumi ülesannete juhiseid täpsustati vastavalt soovitudele ning lisati puuduvad detailid (näiteks lisati täpsem juhiseid, milliseid sõnu ja kuidas peab üliõpilane listi lisama). Kuigi tehti ettepanek lisada nädala „Classes and Objects“ materjalidesse selgitusi objektide asukoha kohta mälu, otsustati, et see võib muuta teemast aru saamise keerulisemaks (ainult üks kolmest praktikumijuhendajast pidas seda vajalikuks). Samuti ei lisatud nädala „Strings, I/O Streams and Lists“ materjalidesse täiendavaid selgitusi teiste failidest lugemiseks või kirjutamiseks kasutatavate klasside kohta, sest sellel õppenädalal toimub alles esmane sissejuhatus teemasse. Failidest ja voogudest räägitakse täpsemalt üheksandal õppenädalal.

Kokkuvõte

Bakalaureusetöö tulemusena koostati ingliskeelsed õppematerjalid kahele õppemoodulile kursusel „Objektorienteeritud programmeerimine“. Loodud materjalide abil on ka võõrkeeli kõnelevatel üliõpilastel võimalik tutvuda objektorienteeritud programmeerimisega. Õppematerjalid koostati teemadele klassid ja objektid ning sõned, sisend/väljund vood ja listid. Materjalide hulka kuuluvad nii lugemismaterjalid, õppevideod, kodutööd, praktikumi ülesanded kui ka enesetestid.

Õppematerjalide loomiseks võeti aluseks ADDIE mudel, mille kõik etapid läbiti ükshaaval. Samuti tutvuti materjalide koostamiseks olemasolevate programmeerimise kursustega ning lugemismaterjalide, õppevideote, enesetestide, kodutööde ja praktikumi ülesannete koostamise soovitustega. Saadud teadmisi rakendati, et luua võimalikult kvaliteetset õppematerjalid.

Loodud õppematerjalidele hinnangu andmiseks küsiti tagasisidet kursuse „LTAT.03.003 Objektorienteeritud programmeerimine“ praktikumijuhendajatelt. Sellest selgus, et õppematerjalid on hästi koostatud ning loogilise ülesehitusega. Juhendajate arvates tuleb materjale täiendada ja parandada minimaalselt. Vastavalt tagasisidele tehti kursuses parandusi.

Bakalaureusetööd on võimalik edasiarendada mitmel viisil. Esiteks on võimalik luua õppematerjalid ka ülejäänud nädalatele, millel materjalid puuduvad. Teiseks oleks vaja koostada koduülesannetele automaatkontrollid, et vähendada õppeaine juhendajate töökoormust. Kolmandaks, murelahendajate koostamine aitaks üliõpilastel saada abi kodutööde lahendamisel ning vähendaks samuti õppejõudude töökoormust.

Viidatud kirjandus

- [1] A. S. Gillis, 'What is Object-Oriented Programming (OOP)?'.
<https://www.techtarget.com/searcharchitecture/definition/object-oriented-programming-OOP> (09.04.2025).
- [2] D. P. Drljača, S. Tomić, and K. Ris, 'Benchmarking Instructional Design Models - ADDIE Wins.', *Economy & Market Communication Review / Casopis za Ekonomiju i Trzisne Komunikacije*, vol. 14, no. 2, pp. 688–698, Dec. 2024, doi:
<https://doi.org/10.7251/EMC2402688D>.
- [3] W. C. Allen, 'Overview and Evolution of the ADDIE Training System', *Advances in Developing Human Resources*, vol. 8, no. 4, pp. 430–441, Nov. 2006, doi:
<https://doi.org/10.1177/1523422306292942>.
- [4] Teachfloor. ADDIE Model: Definition, Benefits, Best Practices - Teachfloor.
<https://www.teachfloor.com/elearning-glossary/addie-model> (07.04.2025).
- [5] The ADDIE Learning Design Process for Instructional Designers.
<https://www.digitallearninginstitute.com/blog/the-digital-learning-design-process-addie-model-for-instructional-design> (16.03.2025).
- [6] Tartu Ülikool. Objektorienteeritud programmeerimine (6 EAP) LTAT.03.003.
<https://courses.cs.ut.ee/2024/OOP/spring/Main/KursuseKorraldus> (24.11.2024).
- [7] Tartu Ülikooli õppeinfosüsteem. Objektorienteeritud programmeerimine (6 EAP) LTAT.03.003.
<https://ois2.ut.ee/#/courses/LTAT.03.003/version/d2bd00a1-232d-5535-d840-b0b78d26496c/details> (24.11.2024).
- [8] Tartu Ülikooli õppeinfosüsteem. Programmeerimine keeles C++ (6 EAP) LTAT.03.025.
<https://ois2.ut.ee/#/courses/LTAT.03.025/version/896c957b-031d-c95d-1211-ce77dd639b75/details> (30.11.2024).
- [9] Tartu Ülikool. Programmeerimine keeles C++ (6 EAP) LTAT.03.025.
<https://courses.cs.ut.ee/2024/cpp/spring> (01.12.2024).
- [10] Tartu Ülikooli õppeinfosüsteem. Operatsioonisüsteemid (6 EAP) LTAT.06.001.
<https://ois2.ut.ee/#/courses/LTAT.06.001/version/d4a1bf94-36f4-9c48-9f6a-c8cc77a13b8f/details> (06.02.2025).

- [11] Tartu Ülikool. Operatsioonisüsteemid (6 EAP) LTAT.06.001.
<https://courses.cs.ut.ee/2024/os/fall/Main/HomePage> (06.02.2025).
- [12] Tartu Ülikooli õppeinfosüsteem. Objektorienteeritud programmeerimine (6 EAP) LTAT.03.003.
<https://ois2.ut.ee/#/courses/LTAT.03.003/version/45b7f8eb-7e9e-cbde-6187-877820687815/feedback> (01.03.2025).
- [13] Tartu Ülikooli õppeinfosüsteem. Programmeerimine keeles C++ (6 EAP) LTAT.03.025.
<https://ois2.ut.ee/#/courses/LTAT.03.025/version/16e3b20b-6f07-1e5c-2bb0-593c9fe3c1f1/feedback> (01.03.2025).
- [14] Tartu Ülikooli õppeinfosüsteem. Operatsioonisüsteemid (6 EAP) LTAT.06.001.
<https://ois2.ut.ee/#/courses/LTAT.06.001/version/71d047e1-1eb6-fbee-f2cf-0b3ef4690a68/feedback> (01.03.2025).
- [15] A. E. Chis, A.-N. Moldovan, L. Murphy, P. Pathak, and C. Hava Muntean, 'Investigating Flipped Classroom and Problem-based Learning in a Programming Module for Computing Conversion Course', *Educational Technology & Society*, vol. 21, no. 4, pp. 232–247, Oct. 2018.
- [16] Y. Fan, 'Flipped classroom: A teaching model', Sep. 2022, doi:
https://doi.org/10.50908/grb.1.0_54.
- [17] M. Fetaji, B. Fetaji, and M. Ebibi, 'Analyses of possibilities of Flipped Classroom in Teaching Computer Science Courses', *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 747–752, 2019, doi: <https://doi.org/10.23919/MIPRO.2019.8757126>.
- [18] E. Han and K. C. Klein, 'Pre-Class Learning Methods for Flipped Classrooms', *American Journal of Pharmaceutical Education*, vol. 83, no. 1, p. 6922, Feb. 2019, doi:
<https://doi.org/10.5688/ajpe6922>.
- [19] C. E. Heiner, A. I. Banet, and C. Wieman, 'Preparing students for class: How to get 80% of students reading the textbook before class', *American Journal Of Physics*, vol. 82, no. 10, pp. 989–996, Nov. 2014, doi: <https://doi.org/10.1119/1.4895008>.

- [20] T. Long, J. Logan, and M. Waugh, 'Students' Perceptions of the Value of Using Videos as a Pre-class Learning Experience in the Flipped Classroom', *TechTrends: Linking Research and Practice to Improve Learning* publication of the Association for Educational Communications & Technology, vol. 60, no. 3, pp. 245–252, May 2016, doi: <https://doi.org/10.1007/s11528-016-0045-4>.
- [21] M. Boutell, 'Choosing Face-to-face or Video-based Instruction in a Mobile App Development Course', in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Seattle Washington USA: ACM, Mar. 2017, pp. 75–80. doi: <https://doi.org/10.1145/3017680.3017774>.
- [22] P. J. Guo, J. Kim, and R. Rubin, 'How video production affects student engagement: an empirical study of MOOC videos', in *Proceedings of the first ACM conference on Learning @ scale conference*, in L@S '14. New York, NY, USA: Association for Computing Machinery, Mar. 2014, pp. 41–50. doi: <https://doi.org/10.1145/2556325.2566239>.
- [23] R. Bourke and M. Mentis, 'Self-assessment as a process for inclusion', *International Journal of Inclusive Education*, vol. 17, no. 8, pp. 854–867, Aug. 2013, doi: <https://doi.org/10.1080/13603116.2011.602288>.
- [24] R. Matthews, H. S. Hin, and K. A. Choo, 'Comparative Study of Self-test Questions and Self-assessment Object for Introductory Programming Lessons', *Procedia - Social and Behavioral Sciences*, vol. 176, pp. 236–242, Feb. 2015, doi: <https://doi.org/10.1016/j.sbspro.2015.01.466>.
- [25] T. M. Haladyna, S. M. Downing, and M. C. Rodriguez, 'A Review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment.', *Applied Measurement in Education*, vol. 15, no. 3, pp. 309–334, Jan. 2002.
- [26] J. L. Epstein and F. L. Van Voorhis, 'More Than Minutes: Teacher's Roles in Designing Homework', *EDUCATIONAL PSYCHOLOGIST*, vol. 36, no. 3, LAWRENCE ERLBAUM ASSOCIATES INC, United States, pp. 181–194, Jan. 01, 2001.
- [27] C. Vatterott, 'Five Hallmarks of Good Homework', *Educational Leadership*, vol. 68, no. 1, pp. 10–15, Sep. 2010.
- [28] F. J. Kontur and N. B. Terry, 'Motivating Students to Do Homework', *The Physics Teacher*, vol. 52, no. 5, pp. 295–297, May 2014, doi: <https://doi.org/10.1119/1.4872413>.

- [29] M. Planchard, K. L. Daniel, J. Maroo, C. Mishra, and T. McLean, 'Homework, Motivation, and Academic Achievement in a College Genetics Course', *Bioscene: Journal of College Biology Teaching*, vol. 41, no. 2, pp. 11–18, Dec. 2015.
- [30] E. Sarap. Lugemismaterjalide ja enesekontrolliülesannete koostamine kursusele „Introduction to Programming”. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö. 2024. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=79641 (02.02.2025).
- [31] A. Zakatov. Abivahenditega varustatud koduülesannete koostamine Tartu Ülikooli kursusele „Objektorienteeritud programmeerimine”. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö. 2023. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=77085 (02.02.2025).

Lisad

I. Lugemismaterjalid, õppevideod ja enesetestid

Teema “Classes and Objects”:

https://courses.cs.ut.ee/2025/OOPr/fall/Main/ReadingMaterials_Classes

Teema “Strings, I/O Streams and Lists”:

https://courses.cs.ut.ee/2025/OOPr/fall/Main/ReadingMaterials_Strings

II. Koduülesanded

Teema “Classes and Objects”:

https://courses.cs.ut.ee/2025/OOPr/fall/Main/Homework_Classes

Teema “Strings, I/O Streams and Lists”:

https://courses.cs.ut.ee/2025/OOPr/fall/Main/Homework_Strings

III. Praktikumi ülesanded

Teema “Classes and Objects”:

https://courses.cs.ut.ee/2025/OOPr/fall/Main/PracticeMaterials_Classes

Teema “Strings, I/O Streams and Lists”:

https://courses.cs.ut.ee/2025/OOPr/fall/Main/PracticeMaterials_Strings

IV. Tehisintellekti kasutamise näide faili sisu genereerimiseks

Näide, kuidas genereeriti andmeid esimese faili jaoks:

Genereeri mulle järgnevas stiilis andmeid mis lähevad txt faili, kokku peab olema 45 rida.
Name,place,age_group. Seega võiksid olla mõned read näiteks sellised (age_group väärtused saavad olla junior, mini ja senior ja place on mis koha osaleja saavutas):
Mati Karu,15,junior
Mark Suli,27,mini

Siin on soovitud tulemus täpselt sellisel kujul:

```
css Copy Edit  
Mati Karu,15,junior  
Andres Valge,8,junior  
Kristel Rand,18,junior  
Peeter Saar,9,junior  
Juhan Lepa,6,junior  
Eliis Vaher,13,junior  
Karl Õun,32,junior  
Merit Puu,20,junior  
Eero Haug,24,junior  
Paul Raud,1,junior  
Katrín Kruus,12,junior  
Keit Sild,4,junior  
Madis Toom,33,junior  
Marleen Luht,19,junior  
Heidi Aare,5,junior  
Hanna Maas,42,junior  
Mark Suli,27,mini  
Kaarel Tamm,11,mini  
Mari-Liis Aas,34,mini  
Raimond Kalju,25,mini  
Ingrid Leht,2,mini  
Tiina Vaik,7,mini  
Gustav Linn,45,mini  
Signe Ruut,37,mini  
Annabel Oja,26,mini  
Lauri Hunt,21,mini  
Riina Meel,30,mini  
Pille Roos,10,mini  
Heli Vaher,46,mini  
Juku Laks,28,mini  
Kert Paas,14,mini  
Tiina Vaik,7,mini  
Rasmus Eerik,49,senior  
Anu Mets,41,senior  
Tarmo Pihl,48,senior  
Peeter Saar,9,senior  
Liina Roos,22,senior  
Silja Taim,17,senior  
Grete Laks,16,senior  
Katrín Kruus,12,senior  
Janne Pärn,36,senior  
Siim Talvik,50,senior  
Lauri Hunt,21,senior  
Tiina Kool,10,mini
```

Kui soovid, võin ka selle salvestada `.txt` faili alla laadimiseks. Kas teha seda?

V. Tagasiside küsimustikud

Lugemismaterjalide küsimustik (kasutati nii nädala “Classes and Objects” kui ka nädala “Strings, I/O Streams and Lists” jaoks):

<https://survey.ut.ee/index.php/211284?lang=en>

Enesetestide küsimustik (kasutati nii nädala “Classes and Objects” kui ka nädala “Strings, I/O Streams and Lists” jaoks):

<https://survey.ut.ee/index.php/229985?lang=en>

Kodutööde küsimustik (kasutati nii nädala “Classes and Objects” kui ka nädala “Strings, I/O Streams and Lists” jaoks):

<https://survey.ut.ee/index.php/629838?lang=en>

Praktikumi ülesannete küsimustik (kasutati nii nädala “Classes and Objects” kui ka nädala “Strings, I/O Streams and Lists” jaoks):

<https://survey.ut.ee/index.php/822582?lang=en>

