

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Raiki Mänd

**Tartu Ülikooli kursuse „Programmeerimine“
arvestuslike testide uuendamine**

Bakalaureusetöö (9 EAP)

Juhendajad: Tauno Palts, PhD
Reimo Palm, PhD

Tartu 2023

Tartu Ülikooli kursuse „Programmeerimine“ arvestuslike testide uuendamine

Lühikokkuvõte:

Töö eesmärgiks on uuendada Tartu Ülikooli kursuse „Programmeerimine“ 1. ja 2. kontrolltöö arvestusliku osa teste. Selle jaoks analüüsiti 2022/2023 õppeaasta sügissemestril toimunud kursuse „Programmeerimine“ 1. ja 2. kontrolltöö arvestusliku osa teste ning tagasisideküsitluste vastuseid. Testide analüüsi eesmärgiks oli tuvastada puudujääke nii testiküsimustes, hindamises kui ka testiformaadis. Lisaks tuvastati õpilaste testiküsimuste vastustes esinenud vead ning prooviti leida eksimuste taga olevaid põhjuseid. Tagasisideküsitluste analüüsis keskenduti tudengite tagasisidele, mis puudutas kontrolltööde arvestusliku osa teste. Töö tulemusena valmisid juhendid mõlema kontrolltöö arvestusliku osa uuendatud testi koostamiseks, et testide koostamine oleks õppejõududele üheselt mõistetav, ning juhendite näitlikustamiseks mõlema kontrolltöö arvestusliku osa tarbeks näidistestid.

Võttesõnad: programmeerimise algkursus, kontrolltöö, testide koostamine, juhend

CERCS: P175 Informaatika, süsteemiteooria; S270 Pedagoogika ja didaktika

Updating Assessment Tests for the Course „Computer Programming“ at the University of Tartu

Abstract:

The aim of the thesis is to update the assessment tests for the 1st and 2nd summative tests of the „Computer Programming“ course at the University of Tartu. For this purpose the assessment tests and feedback surveys were analyzed from the 2022/2023 fall semester. The analysis aimed to identify deficiencies in the test questions, grading and format. In addition, errors in students' test responses were identified and efforts were made to find the causes behind the errors. In the analysis of the feedback surveys, the focus was on the students' feedback regarding the assessment tests. As a result of the work, guidelines for creating the updated assessment tests and sample tests for illustration were developed. These guidelines aim to ensure that creating the tests would be unambiguous for the teaching staff.

Keywords: introductory programming course, test, test creation, guidelines

CERCS: P175 Informatics, system theory; S270 Pedagogy and didactics

Sisukord

| | |
|---|----|
| Sissejuhatus | 5 |
| 1. Teoreetiline taust | 6 |
| 1.1 Kursuse „Programmeerimine“ sisu ja ülesehitus | 6 |
| 1.2 Arvestuslikud testid kursusel „Programmeerimine“ | 7 |
| 1.3 Seotud tööd | 8 |
| 2. Küsimuste tüübid | 10 |
| 2.1 Küsimuste tüübid programmeerimise algkursuste hindamisel | 10 |
| 2.2 Valikvastustega küsimused | 10 |
| 2.3 Lühivastusega küsimused | 11 |
| 3. Metoodika | 13 |
| 3.1 Andmete kogumine ja analüüs | 13 |
| 3.2 Arvestuslike testide juhendite ja näidistestide loomine | 14 |
| 4. Tulemused | 15 |
| 4.1 1. kontrolltöö arvestusliku testi analüüs | 15 |
| 4.1.1 1. kontrolltöö arvestusliku testi analüüs küsimuste kaupa | 18 |
| 4.1.2 1. kontrolltöö tagasiside analüüs | 20 |
| 4.2 2. kontrolltöö arvestusliku testi analüüs | 21 |
| 4.2.1 2. kontrolltöö arvestusliku testi analüüs küsimuste kaupa | 24 |
| 4.2.2 2. kontrolltöö tagasiside analüüs | 26 |
| 4.3 Lõppküsitluse analüüs | 27 |
| 4.4 Kontrolltööde arvestuslike testide koostamise juhendid | 28 |
| 4.5 Kontrolltööde arvestuslike testide näidistestid | 30 |
| 5. Arutelu | 32 |
| Kokkuvõte | 33 |
| Viidatud kirjandus | 34 |

| | |
|--|----|
| Lisad | 36 |
| 1. 1. kontrolltöö arvestusliku testi hindamisskeem 2022 | 36 |
| 2. 1. ja 2. kontrolltöö arvestusliku testi küsimused | 38 |
| 3. 1. ja 2. kontrolltöö arvestuslikes testides esinenud vead | 39 |
| 4. 2. kontrolltöö arvestusliku testi hindamisskeem 2022 | 40 |
| 5. Juhend 1. kontrolltöö arvestusliku testi koostamiseks | 41 |
| 6. Juhend 2. kontrolltöö arvestusliku testi koostamiseks | 44 |
| 7. 1. kontrolltöö arvestusliku testi näidistest | 47 |
| 8. 2. kontrolltöö arvestusliku testi näidistest | 55 |
| 9. Litsents | 62 |

Sissejuhatus

Tartu Ülikooli kursus „Programmeerimine“ on sissejuhatav kursus programmeerimisse, mis kuulub 12 erinevasse õppekavasse ning on seejuures eeldusaineks kuuele õppeainele. Seetõttu on tegemist väga populaarse õppeainega, millele registreerus 2022/2023 sügissemestril 467 üliõpilast [1].

Tartu Ülikooli kursusel „Programmeerimine“ on üheks positiivse hinde saamise kriteeriumiks see, et nii mõlema kontrolltöö kui ka eksami arvestuslik osa peab olema arvestatud. Arvestuslikus osas arvestuse saamiseks tuleb kõik testiküsimused vastata õigesti, küll aga väiksemad puudujäägid antakse andeks. Selline süsteem on olnud kasutusel juba aastaid. 2022. aastal valminud bakalaureusetöö raames loodi juhendid antud kursuse kontrolltööde koostamiseks [2].

Käesoleva bakalaureusetöö eesmärgiks on uuendada kursuse „Programmeerimine“ 1. ja 2. kontrolltöö arvestusliku osa teste. Selle tarbeks analüüsiti töö käigus 2022/2023 sügissemestril toimunud kursuse 1. ja 2. kontrolltöö arvestusliku osa teste ja tagasisideküsitluse vastuseid. Saadud tulemuste põhjal loodi juhendid uute testide koostamiseks ning mõlema kontrolltöö arvestusliku osa tarbeks üks näidistest.

Selle töö esimeses peatükis antakse ülevaade kursuse „Programmeerimine“ sisust ja ülesehitusest ning tutvustatakse arvestusliku osa testi kontseptsiooni. Teises peatükis kirjeldatakse erinevaid küsimusetüüpe ning nende kasutamist teadmiste kontrollimisel. Kolmandas peatükis kirjeldatakse 2022/2023 sügissemestril toimunud programmeerimise kursuse kontrolltööde arvestuslike osade analüüsi ja uute juhendite loomise metoodikat. Neljas peatükk sisaldab analüüsi tulemusi ning tutvustab loodud juhendeid ja näidisteste. Viimases peatükis arutletakse töö tulemuste ning võimalike edasiarenduste üle.

1. Teoreetiline taust

Selles peatükis tutvustatakse kursuse „Programmeerimine“ sisu ja ülesehitust, kursuse arvestuslikke teste ning käesoleva bakalaureusetööga seotud töid.

1.1 Kursuse „Programmeerimine“ sisu ja ülesehitus

Selles peatükis antakse ülevaade 2022/2023 sügissemestril toimunud Tartu Ülikooli kursuse „Programmeerimine“ (LTAT.03.001) sisust ja ülesehitusest, tuginedes kursuse kodulehele [3] ning kirjeldusele ülikooli õppeinfosüsteemis [1]. Tegemist on sissejuhatava kursusega programmeerimisse, mis ei nõua osalejatelt eelnevaid algteadmisi. Õppeinfosüsteemis on välja toodud järgnevad õpiväljundid:

- kursuse läbinud üliõpilane tunneb ja oskab kasutada põhilisi programmeerimiskonstruktsioone: muutuja, avaldis, omistuslause, tingimuslause, tsükel, alamprogramm, rekursioon, andmevahetus kasutaja ja failidega;
- tunneb põhilisi andmetüüpe ja -struktuure (täis- ja ujukomarvud, tõeväärtused, sõned, järjendid jne) ning oskab kasutada vastavaid standardoperatsioone;
- oskab analüüsida ja üksikasjalikult selgitada programmi töö käiku ning programmi muuta, täiendada ja edasi arendada;
- oskab luua lihtsamale ülesandele vastava algoritmi, koostada ja korrektselt vormistada lahendusprogrammi ning seda siluda ja testida;
- oskab realiseerida programmeerimisalaseid projekte rühmakaaslastega koostöös.

Õpe toimub programmeerimiskeeles Python ning programmeerimiskeskkonnaks on Thonny. Õppeaines on kasutusel ümberpööratud klassiruumi meetod. See tähendab, et enne praktikume tuleb iseseisvalt ettenähtud materjalid läbi töötada ning lahendada koduülesanded. Praktikumides keskendutakse vastavate teemade kinnistamisele ning praktikumiülesannete lahendamisele. Ettenähtud mahuks on 156 tundi, millest loengud moodustavad 32, praktikumid 32 ning iseseisev töö 92 tundi. Õpe on ära jaotatud 16-le nädalale, millele järgneb eksamisessioon.

1.2 Arvestuslikud testid kursusel „Programmeerimine“

Õppeaines kasutuses olev hindamissüsteem on kujutatud tabelis 1.

Tabel 1. Hindamissüsteem kursusel "Programmeerimine" [3].

| | Aeg | Alampiir | Max | Kommentaar |
|-------------------|--------------|---------------------------|-----|---|
| Testid | 1.-16. nädal | | 7 | 0,5 punkti testist |
| Kodutöö+praktikum | 1.-16. nädal | 7 | 13 | 0,5+0,5 kodutööst ja praktikumist |
| Projekt | 7.-16. nädal | 5 | 10 | Kahes osas (5+5) |
| 1. kontrolltöö | 6. nädal | Arvestusliku osa arvestus | 20 | Kahes osas: arvestuslik ja punktiline osa |
| 2. kontrolltöö | 12. nädal | Arvestusliku osa arvestus | 20 | Kahes osas: arvestuslik ja punktiline osa |
| Eksam | jaanuaris | Arvestusliku osa arvestus | 30 | Kahes osas: arvestuslik ja punktiline osa |
| Lisaülesanded | | | 10 | |

Üheks aluseks positiivse hinde saamisel on arvestusliku osa arvestus 1. ja 2. kontrolltöö ning eksamitöö puhul. Kõikidel eelmainitud juhtudel on arvestusliku osa aluseks Moodle'i¹ test [5]. Testi sooritamiseks on kõikidel õpilastel üks katse. 1. ja 2. kontrolltöö puhul koosneb test kuuest küsimusest ning lahendamisaega on 45 minutit. Eksami arvestusliku osa test koosneb neljast küsimusest ning lahendamiseks on aega sarnaselt kontrolltöödele 45 minutit. Nii kontrolltööde kui ka eksami puhul on arvestusliku osa lahendamisel keelatud materjalide ja abivahendite kasutamine ning teiste isikutega suhtlemine. Lubatud on kasutada pliiatsit ja paberit. Arvestuse saamiseks tuleb kõikidele küsimustele vastata õigesti. Küll aga on väiksemad eksimused lubatud, kui hindaja arvates on hinnatav õige lahenduse peale tulnud. Sellised vead on seotud lühivastuse tüüpi küsimustega, kus lahendajal tuleb sisestada vastuseks üks või rohkem sõna. Selleks on näiteks eksimused, mis puudutavad kirjavigu muutujate nimetamisel või avaldiste koostamisel.

Kui arvestusliku osa tulemuseks on mitteamvestatud, siis see tähendab ka seda, et kogu kontrolltöö loetakse mitteamvestatuks ning arvestuse saamiseks tuleb järeltööl arvestuslik test uuesti sooritada. Hoolimata mitteamvestuslikust tulemusest ei tühistata punktilise osa tulemust, kuid järeltööl on võimalik ka seda tulemust parandada. Küll aga jääb punktilise osa korral lõplikuks tulemuseks viimane sooritus, mis tähendab, et kui järeltöö tulemus on kehvem kui esialgse töö tulemus, siis jääb kehvem tulemus kehtima.

¹ „Moodle (Modular Object-Oriented Dynamic Learning Environment) on avatud lähtekoodiga tasuta e-õppe rakendus, mis on üldtuntud e-kursuste haldamissüsteemina“ [4].

1.3 Seotud tööd

Järgnev peatükk tugineb 2022. aastal Triinu Augi poolt kaitstud bakalaureusetööle teemal „Kontrolltööde koostamise ja hindamise juhendite loomine Tartu Ülikooli kursusele „Programmeerimine““ [2].

Sarnaselt käesoleva bakalaureusetööga tugines ka Augi töö kursusele „Programmeerimine“. Aug analüüsis töö raames 2021/2022 õppeaasta sügissemestri kursuse „Programmeerimine“ mõlemat kontrolltööd. Analüüsi eesmärgiks, mis puudutas kontrolltööde arvestuslikku osa, oli tuvastada, kas arvestuslikus osas olid kõik küsimused sobiva raskusastmega, arvestades, et see kontrollis minimaalsete teadmiste omandamist. Tema tehtud analüüs, mis puudutas arvestusliku osa teste, sisaldas järgmist:

- arvestusliku osa eest saadud punktide jaotumine;
- arvestusliku osa eest saadud punktide jaotumine küsimuste kaupa;
- arvestusliku osa eest saadud punktide jaotumine praktikumirühmade kaupa;
- arvestusliku osa eest saadud punktide ja programmeerimise osa eest saadud punktide vaheline seos.

Kui mõne arvestusliku osa testiküsimuse tulemus erines teistest märgatavalt, siis analüüsis ta seda küsimust lähemalt: ta kirjeldas lühidalt, mida küsiti ja mille vastu enim eksiti. Selliselt uuris ta 1. kontrolltöö arvestusliku testi puhul vaid 5. küsimust ning 2. kontrolltöö arvestusliku testi puhul vaid 4. küsimust. 2. kontrolltöö 4. küsimuse puhul Aug eksimusi välja ei toonud, vaid piirdus küsimuste kirjeldustega. Praktikumirühmade tulemuste analüüsist polnud järeldusi võimalik teha, sest tulemused olid üsnagi võrdsed erinevate rühmade lõikes. Veel analüüsis ta tagasisideküsitluse vastuseid. Analüüsist tuli välja, et 2. kontrolltöö küsimused olid keerulisemad kui 1. kontrolltöö küsimused, samas küsimuste sõnastus oli olnud sel korral parem kui 1. kontrolltöö puhul. Aug järeldas analüüsi tulemustest seda, et mõlema kontrolltöö arvestusliku osa küsimused olid eesmärgile vastava tasemega.

Augi töö tulemusena valmisid kahele kokkuvõtliku hindamise eesmärgi täitvale kontrolltööle koostamise ja hindamise juhendid, mis puudutasid nii arvestuslikku kui ka kontrolltöö punktilist osa. Siiski oli töö peamine fookus punktilise osa juhendite koostamisel ning arvestusliku osa juhendeid ei ole programmeerimise kursusel seni kasutusele võetud².

² Kursuse „Programmeerimine“ vastutava õppejõu Tauno Paltsi sõnul.

Seetõttu keskendub käesolev töö põhjalikult mõlema kontrolltöö arvestusliku osa testidele ning punktilist osa töös ei käsitleta. Käesoleva töö analüüsi käigus ei keskenduta enam aspektidele, mida Triinu Aug 2022. aastal uuris.

2. Küsimuste tüübid

1. ja 2. kontrolltöö arvestusliku osa aluseks on Moodle'i test. Selles peatükis antakse ülevaade enamlevinud küsimusetüüpidest, mida kasutatakse sarnaste testide koostamisel.

2.1 Küsimuste tüübid programmeerimise algkursuste hindamisel

Simon jt [6] poolt läbi viidud uurimuse käigus analüüsiti 20 erinevat programmeerimise algkursuse eksamit, mis olid pärit 10 erinevast institutsioonist ning viiest erinevast riigist. Analüüsitud eksamitöödest 54% moodustasid koodikirjutamise küsimused, 28% lühivastusega küsimused, 17% valikvastustega küsimused ning alla 2% graafilist tüüpi küsimused. Sarnaselt eelmainitud tööle viisid Petersen jt [7] läbi uuringu, mille käigus analüüsiti 15 erinevat eksamitööd 14 erinevast koolist. Analüüsitud eksamitöödest 54% moodustasid koodikirjutamise ülesanded, 36% lühivastusega küsimused, 7% valikvastusega küsimused ning 3% graafilised küsimused.

Tartu Ülikooli kursuse „Programmeerimine“ mõlemad kontrolltööd koosnesid arvestuslikust ja punktilisest osast, kus punktiline osa koosnes praktilistest koodikirjutamise ülesannetest. Arvestuslik osa sarnaseid koodikirjutamise ülesandeid ei sisaldanud ning 2022/2023 aasta sügissemestril toimunud kursusel koosnes arvestuslik osa mõlema kontrolltöö puhul lühivastusega küsimustest. Seega jäetakse käesolevas peatükis koodikirjutamise küsimused vaatluse alt välja.

2.2 Valikvastustega küsimused

Geoffrey Crisp [8] on kirjutanud, et hindamisel on enamlevinud küsimuste tüübiks valikvastustega küsimused. Õpetajate jaoks on valikvastustega küsimusi mugav luua ning nende loomine ei nõua erilist ajakulu ega oskusi. Samas ei pane need õpilasi piisavalt proovile. Crispi sõnul on hoopis väärtuslikumad ehituselt keerukamad ülesanded, mille käigus saab õpilane kasutada erinevaid tööriistu probleemide lahendamiseks. Ka David J. Walker jt [9] väidavad, et valikvastustega küsimused ei ole alati parim lahendus. Õpilaste tagasisidest tuli välja, et mõningatel juhtudel kasutati valikvastuste puhul arvamise meetodit ega ei süvenetudki küsimuse sisusse. Veel tuli välja, et otsiti küsimusele võimalikult sarnase kirja-pildiga vastuseid, eeldades et nende puhul on tegemist õige vastusega. Lisaks peavad nende sõnul olema küsimused hästi sõnastatud, sest halvasti konstrueeritud küsimused võivad põhjustada õppijas tunnet, et teda üritatakse petta.

Shuhidan jt [10] jällegi leidsid, et valikvastustega küsimused ei ole halb meetod algajate programmeerijate kokkuvõtvaks hindamiseks. Läbiviidud uuringu käigus vastasid 66 programmeerimise õppeainete instruktorit üle maailma küsimustikule, mille ühes osas oli neil tarvis analüüsida nelja küsimust: kolme valikvastustega küsimust ning üht lühivastusega küsimust. Kõik küsimused ei nõudnud koodikirjutamise oskust, vaid hoopis koodilugemist ja -mõistmist. Küsimuste puhul tuli hinnata nelja aspekti, milleks olid lahenduseks vajaminevad süntaksialased teadmised, semantilised teadmised, probleemilahenduse oskus ning tõstatatud probleemi raskusaste. Vastustest selgus, et kõigi nelja aspekti vahel esineb korrelatsioon ning instruktorite meelest kontrollivad küsimused kõiki aspekte iga küsimuse puhul võrdsel tasemel. Uuringu tulemustest selgus, et kui valikvastustega küsimused on hoolikalt koostatud, siis need võivad õpilasi tõsiselt proovile panna ning on sobilikud programmeerimisalaste teadmiste kontrolliks. Nende soovitus on see, et õpetajad arvestaksid küsimuste koostamisel eelnevalt mainitud nelja aspektiga.

Baconi [11] poolt läbiviidud uuringu käigus selgus, et valikvastustega küsimustele vastamine võttis õpilastel oluliselt vähem aega kui lühivastustega küsimuste puhul. Hoolimata väiksemast ajakulust ei kannatanud töö usaldusväärsus valikvastustega küsimuste korral. Samuti tuli uurimuses välja see, et valikvastustega küsimuste puhul oli tulemuse ja ajakulu vahel negatiivne korrelatsioon – mida rohkem aega kulutati vastamiseks, seda suurem oli tõenäosus, et vastati valesti.

2.3 Lühivastusega küsimused

Impara ja Fosteri [12] sõnul on lühivastusega küsimused turvalisemad kui valikvastustega küsimused, sest raskendavad mahakirjutamist. Kui küsimused on konkreetsetelt sõnastatud ning oodatud on lühikesed vastused, siis ei võta vastamine nende sõnul kaua aega. Lisaks on selliste küsimuste puhul võimalik kontrollida sügavamaid teadmisi. Samas tõid nad välja, et suure hulga vastuste korral on kontrollimine väga ajamahukas protsess.

Simoni ja Snowdoni [13] läbiviidud uuringust selgus, et samaväärsete valikvastuste ja lühivastusega koodi selgitavate küsimuste puhul õpilaste vastused ei ühtinud. Valikvastustega küsimuste korral saavutasid tudengid paremad tulemused. Uuringus järeldati, et selle põhjustajaks ei olnud õigete vastuste äraarvamine, vaid õpilastel oli raskusi koodi eesmärgi sõnastamisega hoolimata sellest, et nad olid valikvastustega küsimuste puhul näidanud, et nad mõistavad koodi ja oskavad selle tööd selgitava õige vastusevariandi valida.

Lühivastusega küsimuste puhul Moodle'i keskkonnas on probleemiks automaatse hindamissüsteemi kasutamine. Olenevalt küsimusest võivad õpilaste vastused kirja pildilt olla erinevad, isegi kui on tegu õigete vastustega. Seetõttu on küsimuste loomisel keeruline kõikvõimalikud õiged vastused ära määratleda. Sellest tulenevalt võib juhtuda, et suur hulk vastuseid, mille hindamissüsteem on kas valeks või osaliselt õigeks määranud, on tegelikult õiged ning need tuleb käsitsi üle hinnata.

3. Metoodika

Selles peatükis antakse ülevaade 2022/2023 sügissemestril toimunud kursuse „Programmeerimine“ 1. ja 2. kontrolltöö arvestuslike testide tulemuste, küsimuste ja tagasisideküsitluste analüüsi ning uute juhendite ja näidistestide loomise metoodikast.

3.1 Andmete kogumine ja analüüs

Arvestuslike testide andmete kogumiseks ja analüüsiks oli tarvilik ligipääs aine Moodle'i kursusele. Aine vastutav õppejõud andis selleks ligipääsu ning laiendatud õigused käesoleva töö autorile. Laiendatud õigused võimaldasid autoril näha kogu kursusega seotud infot, sealhulgas iga õpilase individuaalseid testisooritusi.

1. kontrolltöö arvestusliku osa testi sooritas 406 ning 2. kontrolltöö testi 353 üliõpilast. Mõlema kontrolltöö arvestusliku osa testi analüüsimiseks laeti aine Moodle'i kursuselt alla Exceli tabel, kus oli mõlema kontrolltöö puhul kirjas iga testisoorituse kohta järgnev info:

1. testisooritaja ees- ja perenimi;
2. testi staatus;
3. testi alustamise kellaaeg;
4. testi lõpetamise kellaaeg;
5. testi sooritamiseks kulunud aeg;
6. testi tulemus skaalal 0.00-1.00;
7. tulemus iga küsimuse eest skaalal 0.00-0.17.

Analüüsimiseks kasutati veel Moodle'i kursuselt kättesaadavat statistikat, mis puudutas näiteks automaatselt ja käsitsi hinnatud vastuste hulka ning erinevate küsimuste variantide esinemist testis. Lisaks sellele analüüsiti kõiki töös esinenud küsimusi ning küsimuste vastuseid, mis olid kas hindamissüsteemi poolt automaatselt osaliselt õigeks või valeks hinnatud või käsitsi hindaja poolt üle hinnatud. Selle jaoks vaadati üle kõikide vastavate küsimuste vastused. Aine Moodle'i kursuselt laaditi alla ka tagasisideküsitluse vastused.

Analüüsi eesmärgiks oli tuvastada kontrolltööde arvestuslike testide puudujäägid ja testides esinenud vead.

Statistika ja jooniste tegemiseks kasutati töös Microsoft Excelit (Microsoft Office Professional Plus 2019), sest osa andmetest olid esitatud vastavas failiformaadis ning autoril oli eelneva kogemuse tõttu antud töövahendit mugav kasutada. Analüüsi käigus uuriti tabelite ja jooniste abil mõlema kontrolltöö arvestusliku testi puhul järgmist:

1. tulemusi;
2. punktide jaotust küsimuste kaupa;
3. lahendamiseks kulunud aega;
4. automaatse ja käsitsi üle hindamise jaotust küsimuste kaupa;
5. küsimuste vastuste koguarvu variantide kaupa;
6. küsimuste punktide jaotust variantide kaupa;
7. küsimuste õigeks loetud vastuste protsenti kõigist vastustest variantide kaupa;
8. küsimuste automaatselt ja käsitsi üle hindamise jaotust variantide kaupa;
9. küsimuste käsitsi üle hindamise protsenti kõigist vastustest variantide kaupa;
10. küsimuste käsitsi üle hinnatud vastuste punktide jaotust variantide kaupa;
11. küsimuste automaatkontrolli poolt mittetuvastatud õigete vastuste arvu variantide kaupa;
12. küsimuste automaatkontrolli poolt mittetuvastatud õigete vastuste protsenti kõigist õigetest vastustest variantide kaupa;
13. igas küsimuses esinenud vigu ja vigade esinemiste arvu variantide kaupa.

Järgnevalt kirjeldatakse arvestuslike testide juhendite ja näidistestide loomise metoodikat.

3.2 Arvestuslike testide juhendite ja näidistestide loomine

Arvestuslike testide juhendite loomisel lähtuti testide analüüsi ja tagasisideküsitluse tulemustest ning võeti arvesse ka Augi [2] poolt loodud juhendeid. Lisaks arvestati 2. peatükis kirjeldatud küsimuste tüüpe, kursuse õpiväljundite ja kursusel läbitud teemade ning õppematerjalide sisuga kursuse koduleheküljel.

Näidistesti loomiseks uuriti 1. ja 2. kontrolltöö testiküsimuste sisu ja sõnastust, testides enamesinenud vigu ning kursuse koduleheküljel olevate õppematerjalide sisu. Näidistesti luues toetuti juhendis välja toodud juhtnööridele. Veenduti, et kõik kontrolltööle eelnevad kursusel läbitud teemad oleksid kaetud ning küsimused ei kontrolliks teadmisi, mida kursuse vältel ei ole tudengitele õpetatud. Arvesse võeti ka käesoleva töö 2. peatükis kirjeldatud küsimusetüüpe.

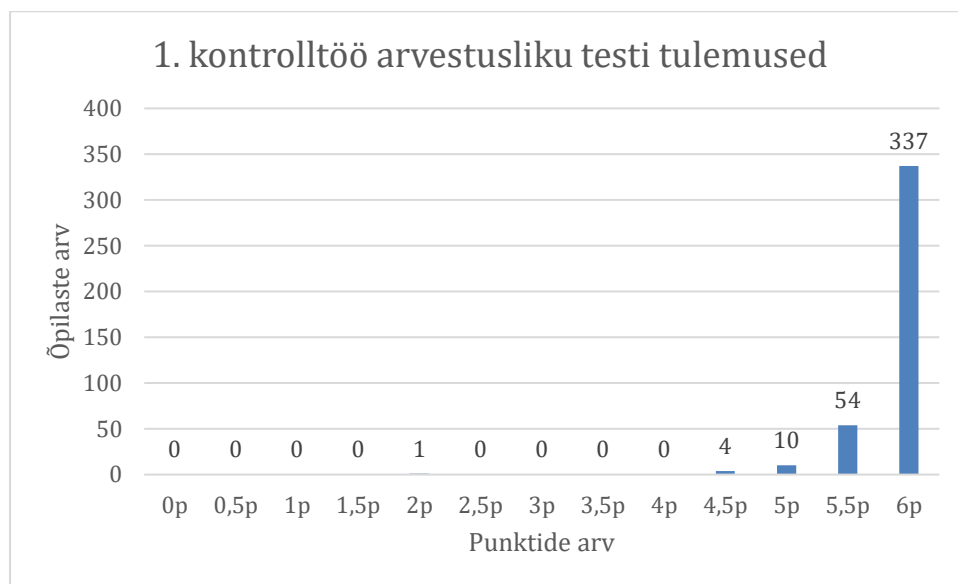
Loodud juhenditele ja näidistestidele küsiti tagasisidet aine vastutavatelt õppejõududelt. Vastutavate õppejõudude tagasiside põhjal viidi sisse vajalikud muudatused.

4. Tulemused

Selles peatükis antakse ülevaade 1. ja 2. kontrolltöö arvestusliku testi tulemuste, küsimuste ja tagasiside analüüsi tulemustest. Lisaks tutvustatakse lähemalt loodud 1. ja 2. kontrolltöö arvestusliku testi juhendite ja näidistestide sisu ja loomise protsessi.

4.1 1. kontrolltöö arvestusliku testi analüüs

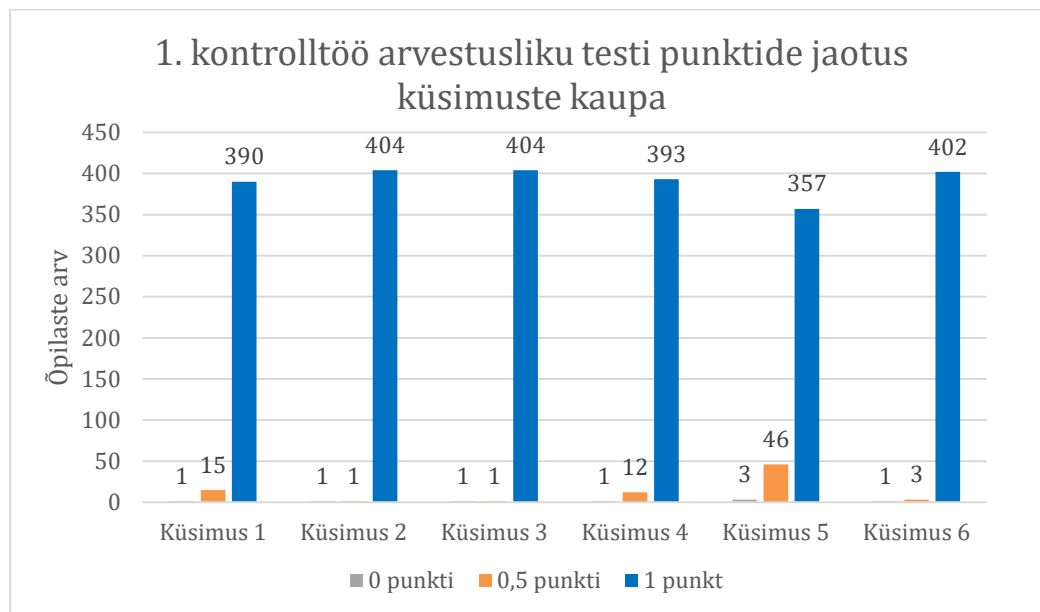
1. kontrolltöö arvestuslikku testi sooritas 406 õpilast. Test koosnes kuuest lühivastusega küsimusest. Iga küsimuse korral toimus õpilastele küsimuse valimine juhuslikult kolme erineva variandi hulgast (erandlikult oli küsimusel 5b omakorda erinevad variandid, kus muutus tsüklimuutuja suurus). 1. küsimuse teemaks oli muutujad, avaldised ja omistuslaused, 2. küsimuse teemaks oli andmevahetus kasutajaga, 3. küsimuse teemaks oli tingimuslaused, 4. küsimuse teemaks oli funktsioonid, 5. küsimuse teemaks oli tsüklid ning 6. küsimuse teemaks oli andmevahetus failiga. Kõigi kuue küsimuse eest oli võimalik teenida kuni üks punkt ehk kogu testi peale oli maksimaalne võimalik punktisumma kuus punkti. Testi edukaks sooritamiseks oli vaja kokku koguda vähemalt 5,5 punkti 6-st. Sellega sai hakkama 391 üliõpilast, kes said testi tulemuseks arvestatud. 15 õpilast said kokku vähem kui 5,5 punkti ning nende tulemuseks märgiti mitteamestatud. Kontrolltöö järeltööd käesolevas töös ei analüüsitud. 1. kontrolltöö arvestusliku testi tulemused on kujutatud joonisel 1.



Joonis 1. 1. kontrolltöö arvestusliku testi tulemused.

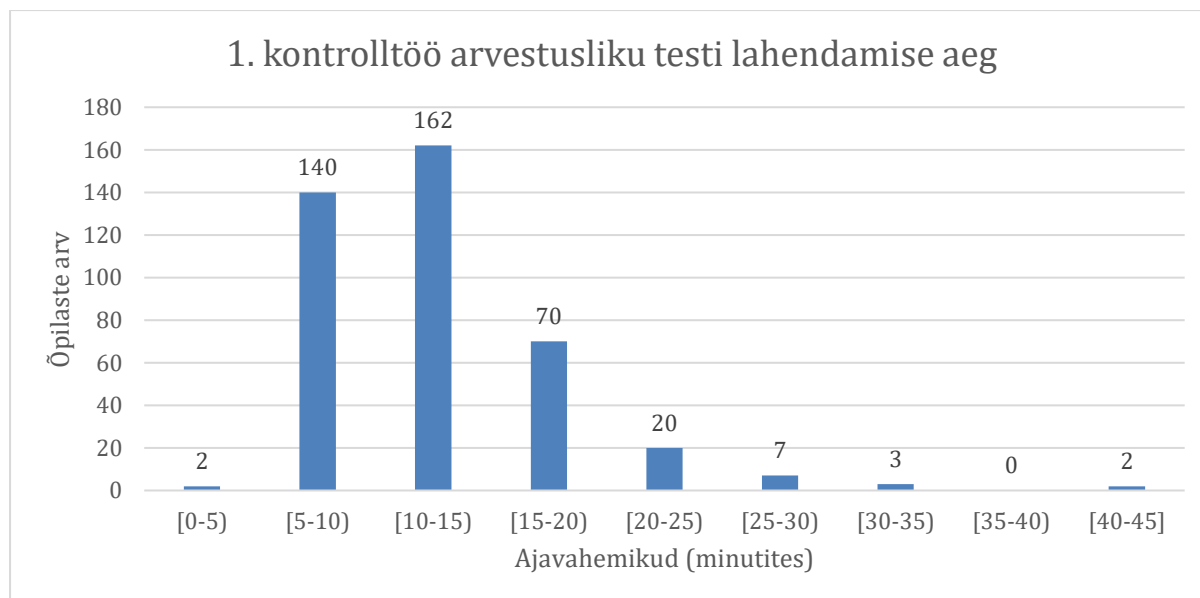
Kõige raskemaks küsimuseks osutus testi viies küsimus, mis kontrollis teadmisi teemal tsüklid. Ülejäänud küsimuste puhul oli punktide jaotus üsnagi võrdne, esile saab tõsta veel esimese

ning neljanda küsimuse, mille puhul said vastavalt 15 ja 12 üliõpilast võimaliku ühe punkti asemel 0,5 punkti. Punktide jaotust küsimuste kaupa iseloomustab joonis 2.



Joonis 2. 1. kontrolltöö arvestusliku testi punktide jaotus küsimuste kaupa.

Testi sooritamiseks oli ette nähtud aega kuni 45 minutit. Õpilaste lahendamiseks kulunud aega iseloomustab joonis 3.



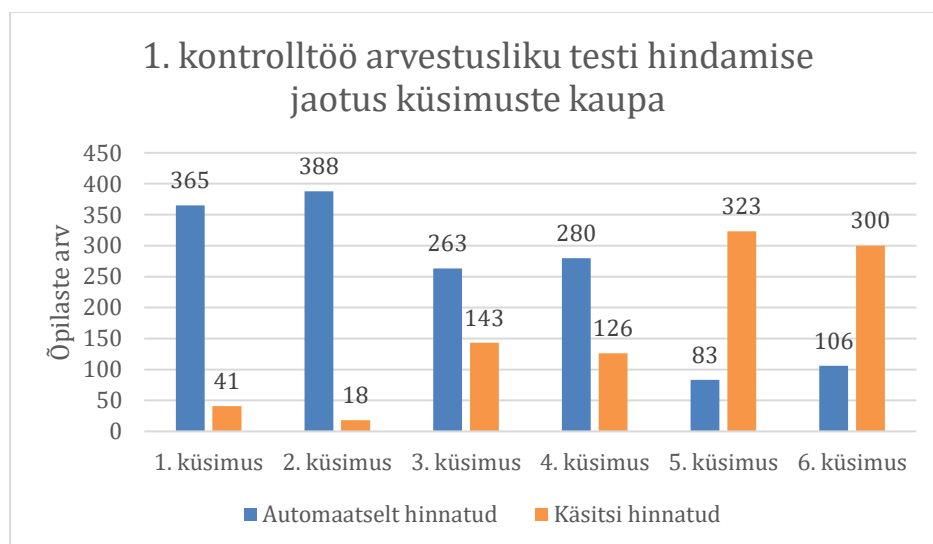
Joonis 3. 1. kontrolltöö arvestusliku testi lahendamiseks kulunud aeg.

Keskmiselt läks testi sooritamiseks aega 12 minutit ja 37 sekundit. Kõige kiirem sooritus oli 4 minutit ja 41 sekundit, kõige pikem sooritus oli 45 minutit.

Kuigi test toimus Moodle'i keskkonnas ning avatud küsimusi ei küsitud, siis käsitsi üle hinnatud vastuste hulk oli ligikaudu 39%. Käsitsi üle hinnatud vastus tähendas kas seda, et vastus ei olnud õige, või seda, et vastus oli õige, aga automaatne hindamissüsteem ei suutnud seda õige vastusena ära tuvastada. Selle põhjustasid järgnevad asjaolud:

- testil oli kõrge hindamislävend: vähemalt üks vale ehk 0-punktiline vastus või kaks osaliselt õiget ehk 0,5-punktilist vastust taganuks mittearvestatud testitulemuse;
- hoolimata kõrgeist lävendist olid hindajad tudengite suhtes leebed: laialisaadetud hindamisskeemi alusel (vt lisa 1) anti väiksemad puudujäägid andeks;
- tudengite vastustes esines palju puudujääke;
- arvestatav hulk tudengite vastuseid olid õiged, aga automaatsele hindamissüsteemile mitteoodatud kujul.

Automaatselt ja käsitsi hinnatud küsimuste jaotust iseloomustab joonis 4.



Joonis 4. 1. kontrolltöö arvestusliku testi hindamise jaotus küsimuste kaupa.

Selgitused edasise analüüsi jaoks:

- kui vastus hinnati õigeks, siis vastuse eest oli tulemuseks 1 punkt;
- kui vastus hinnati osaliselt õigeks, siis vastuse eest oli tulemuseks 0,5 punkti;
- kui vastus hinnati valeks, siis vastuse eest oli tulemuseks 0 punkti;
- automaatselt hinnatud vastus tähendab seda, et automaatse hindamise tulemus jäi lõplikuks tulemuseks;
- käsitsi hinnatud vastus tähendab seda, et vastus hinnati esialgu automaatselt, aga hindaja hindas vastuse käsitsi üle.

Edasi tutvustatakse iga küsimuse analüüsi, mis esines 1. kontrolltöö arvestuslikus testis.

4.1.1 1. kontrolltöö arvestusliku testi analüüs küsimuste kaupa

1. kontrolltöö arvestuslikus testis esinenud küsimused on töö lisades (vt lisa 2).

Kokkuvõtlik tabel, mis koondab 1. kontrolltöö arvestusliku testi iga küsimuse analüüsi tulemusi, on esitatud tabelis 2. Tabelis 2 esitatud veerud tähistavad järgmist:

1. A – küsimuse nimi;
2. B – vastuste koguarv;
3. C-E – lõplik punktide jaotus (vastavalt 0, 0,5 ja 1);
4. F – õigeks loetud vastuste protsent kõigist vastustest;
5. G – automaatselt hinnatud vastuste arv;
6. H – käsitsi üle hinnatud vastuste arv;
7. I – käsitsi üle hinnatud vastuste protsent kõigist vastustest;
8. J-L – käsitsi üle hinnatud vastuste punktide jaotus (vastavalt 0, 0,5 ja 1);
9. M – automaatkontrolli poolt mittetuvastatud õigete vastuste arv;
10. N – automaatkontrolli poolt mittetuvastatud õigete vastuste protsent kõigist õigetest vastustest.

Tabel 2. 1. kontrolltöö arvestusliku testi iga küsimuse analüüsi tulemusi koondav tabel.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|------------------------------------|-----|---|----|-----|--------|-----|-----|-------|---|----|-----|----|-------|
| 1a. Korrutis tegur | 124 | 1 | 13 | 110 | 88.7% | 110 | 14 | 11.3% | 0 | 8 | 6 | 4 | 3.6% |
| 1b. Värv sinine | 135 | 0 | 0 | 135 | 100.0% | 121 | 14 | 10.4% | 0 | 0 | 14 | 3 | 2.2% |
| 1c. 2 ja 3 summa | 147 | 0 | 2 | 145 | 98.6% | 134 | 13 | 8.8% | 0 | 2 | 11 | 8 | 5.5% |
| 2a. Kasutaja nime muutujale | 124 | 1 | 0 | 123 | 99.2% | 110 | 14 | 11.3% | 0 | 0 | 14 | 10 | 8.1% |
| 2b. Väljasta tekst ekraanile | 135 | 0 | 0 | 135 | 100.0% | 133 | 2 | 1.5% | 0 | 0 | 2 | 1 | 0.7% |
| 2c. Kasutaja nimi ekraanile | 147 | 0 | 1 | 146 | 99.3% | 145 | 2 | 1.4% | 0 | 0 | 2 | 0 | 0.0% |
| 3a. Täisarv suurem-võrdne 2 | 149 | 0 | 1 | 148 | 99.3% | 131 | 18 | 12.1% | 0 | 1 | 17 | 2 | 1.4% |
| 3b. Sipsiku omanik | 135 | 0 | 0 | 135 | 100.0% | 109 | 26 | 19.3% | 0 | 0 | 26 | 5 | 3.7% |
| 3c. Paarisarv, mitte 5 ja 10 vahel | 122 | 1 | 0 | 121 | 99.2% | 23 | 99 | 81.1% | 1 | 0 | 98 | 19 | 15.7% |
| 4a. See on paaritu arv | 146 | 0 | 6 | 140 | 95.9% | 80 | 66 | 45.2% | 0 | 6 | 60 | 0 | 0.0% |
| 4b. Taanete suurused | 140 | 1 | 4 | 135 | 96.4% | 130 | 10 | 7.1% | 1 | 4 | 5 | 0 | 0.0% |
| 4c. Sipsiku omanik | 120 | 0 | 2 | 118 | 98.3% | 70 | 50 | 41.7% | 0 | 2 | 48 | 0 | 0.0% |
| 5a. Print A B | 143 | 1 | 26 | 116 | 81.1% | 40 | 103 | 72.0% | 0 | 26 | 77 | 1 | 0.9% |
| 5b. Mitu arvu väljastab | 125 | 1 | 9 | 115 | 92.0% | 23 | 102 | 81.6% | 1 | 4 | 97 | 23 | 20.0% |
| 5c. Paaritud arvud 0-st 10-ni | 138 | 1 | 11 | 126 | 91.3% | 20 | 118 | 85.5% | 0 | 11 | 107 | 62 | 49.2% |
| 6a. Esimese rea lugemine | 135 | 0 | 1 | 134 | 99.3% | 28 | 107 | 79.3% | 0 | 1 | 106 | 1 | 0.7% |
| 6b. Rea kirjutamine | 124 | 1 | 2 | 121 | 97.6% | 59 | 65 | 52.4% | 1 | 2 | 62 | 8 | 6.6% |
| 6c. Sisu sisselugemine | 147 | 0 | 0 | 147 | 100.0% | 19 | 128 | 87.1% | 0 | 0 | 128 | 12 | 8.2% |

Küsimuse 1c puhul hindas automaatkontroll muuhulgas õigeks ka vastused, kus oli kasutatud muutuja „õunu“ asemel teistsugust muutujanime. Seega „õunu“ asemel teistsuguse muutujanime kasutamist analüüsi käigus veaks ei arvestatud. Sarnaselt 1c-le esines ka küsimuse 3a puhul seda, et küsimuse tekstis etteantud muutuja asemel kasutati läbivalt teistsugust muutujanime, kuid seda veaks ei arvestatud ning automaatselt hinnati need küsimused suuremas osas õigeks, kui tingimuslause oli korrektselt koostatud.

Käsitsi hindamise osakaal oli suur küsimuste 3c, 4a, 4c ning kõigi 5. ja 6. küsimuse variantide puhul. Küsimuste 4a, 5c ning 6b puhul põhjustas suure hulga vastuste käsitsi üle hindamist peamiselt see, et õpilastel oli vaja täita kolm lahtrit, mis suurendas eksimisvõimalusi. Lisaks oli automaatkontrolli töö ebaefektiivne küsimuse 5c vastuste hindamisel, kui 49,2% õigetest vastustest jäid tuvastamata.

Küsimuste 4c, 6a ja 6c puhul olid käsitsi üle hindamised seotud küsimuste sõnastusega. Küsimuses 4c oli üheks enamlevinud veaks see, et küsimuse kirjelduses nõutud sõne „Ta on Sipsiku omanik“ väljastamiseks ei kasutatud funktsiooni väljakutsumist, vaid sõne väljastati hoopis print-käsku kasutades. Seda esines 21 korral, millest võib järeldada, et küsimuse sõnastus võis olla eksitav. Sellest tulenevalt oli käsitsi üle hindamise osakaal suur.

Küsimuse 6a sõnastus võis olla kaheti mõistetav: küsimuse püstitusest ei tule näiteks välja, kas kirjutama peab tervikliku programmi või piisab lihtsalt omavahel mittesõltumatutest lausetest, kas faili esimese sisseloetud rea tulemust peab muutujasse salvestama või mitte. Sellest tulenevalt esines näiteks 52 käsitsi üle hinnatud vastuse puhul asjaolu, et faili esimese rea sisselugemise tulemust ei salvestatud muutuja väärtuseks, mida automaatkontroll vastusest ootas.

Küsimuses 6c oli 10 korral vastuses funktsiooni `read()` asemel kasutatud funktsiooni `readlines()`. Kuigi automaatkontroll ootas vastuses `read()` funktsiooni kasutamist, siis küsimuse kirjeldusest ei tule välja, et funktsiooni `readlines()` kasutamine oleks väär. Funktsioon `readlines()` loeb sarnaselt funktsioonile `read()` terve faili sisu, erinevus seisneb selles, et kui `read()` loeb faili sisse ühe sõnena, siis `readlines()` teeb seda järjendina, kus elementideks on failis esinenud read. Seega käesolevas analüüsis loetakse ka eelmainitud vastused õigeks. Üheks enamlevinud veaks osutus see, et faili sisu ei loetud sisse ehk ei salvestatud muutuja väärtuseks. Selle põhjuseks võis olla see, et küsimuse kirjelduses ei olnud terve faili sisu salvestamiseks muutujat ette antud.

Küsimuse sisu põhjustas hindamisel probleeme küsimuste 3c, 5a ja 5b korral. Hoolimata sellest, et suure hulga küsimuse 3c vastuste puhul kasutati etteantud muutuja asemel teistsugust muutujat, mida automaatkontroll vaele ei hinnanud, oli käsitsi üle hinnatud vastuste hulk suur. See oli tingitud sellest, et küsimus nõudis tudengite jaoks keerulise tingimuslause koostamist, millega suurem osa õpilastest hakkama ei saanud.

Kuna küsimuse 5a puhul esines palju käsitsi hindamist, siis käesoleva küsimuse oleks võinud asendada mõne muu küsimusega. Näiteks küsimusega, kus `for`-tsükli täitmisel muutuja `x` väärtus muutub (suureneb või väheneb) ning tsükli lõppedes väljastatakse muutuja `x` väärtus ekraanile. Õpilane peab vastama küsimusele, et mis väljastatakse programmi käivitamisel ekraanile. Sellise küsimuse puhul oleks lihtne automaatselt arvulist vastust kontrollida, lisaks nõuaks õige vastuseni jõudmine sarnaselt 5a küsimusele teadmisi, kuidas tsükli täitmine töötab.

Variant 5b käsitsi üle hinnatud vastustest 23 olid õiged, kuid automaatkontroll oli need vaele hinnanud, sest oodatud vastuseks oli küsimuse koostaja poolt määratud ekslikult vale vastus. Hoolimata mainitud eksimusest ei tekkinud sellist olukorda, kus automaatsel hindamisel oleks mõni tudeng saanud vaele vastusest hoolimata automaatkontrollis õige tulemuse. Mitmetel juhtudel olid tudengid lähedal õigele vastusele, aga eksimus võis olla tingitud sellest, et programmis läbiti tsükli kümneid tuhandeid kordi, mida oli võimatu kas peas või paberil läbi mängida. Antud küsimuses võiks muuta tsükli selliselt, et õpilastel oleks võimalik kogu tsükli töö peas või paberil läbi teha. Selliselt jääks küsimuse eesmärk samaks, kuid oleks õpilastele jõukohasem.

Analüüsi käigus vaadati üle kõik sellised testiküsimuste vastused, mis olid kas käsitsi üle hinnatud või olid automaatsel hindamisel kas vaele või osaliselt õigeks hinnatud (lõplik tulemus). Vastustes esinenud vead lahterdati veatüüpide järgi ning loeti kokku vastavate vigade esinemiste arvud, et saada ülevaade, mille vastu õpilased enim eksivad. Lisaks võrreldi samasse kategooriasse kuulunud vigade puhul vastuste lõplikke tulemusi, mis mitmetel juhtudel erinesid. 1. kontrolltöö arvestusliku testi igas küsimuses esinenud vigu kirjeldavad tabelid on töö lisades (vt lisa 3). Lõplikke tulemusi võrreldi ka sügisel kasutatud hindamisskeemiga (vt lisa 1), et kontrollida, kas hindamine on üheselt mõistetav. Hindamises esines mitmeti mõistmist, kus töö autor oleks hindamisskeemi alusel vastuseid teistmoodi hinnanud.

4.1.2 1. kontrolltöö tagasiside analüüs

1. kontrolltöö tagasisideküsitlusele vastas 109 üliõpilast. Õpilased hindasid kontrolltöö arvestusliku testi raskusastet skaalal väga lihtne, lihtne, pigem lihtne, pigem keeruline,

keeruline ning väga keeruline. Ligikaudu 37,6% vastanutest pidas testi pigem lihtsaks, lisaks hindas 18,3% vastanutest testi lihtsaks ning 11% väga lihtsaks. 26,6% vastanutest pidas testi pigem keeruliseks ja 6,4% keeruliseks. Väga keeruliseks ei hinnanud testi mitte ükski küsitlusele vastanu.

Tudengite käest küsiti veel seda, et millised küsimused olid nende arvates kõige raskemad. 45% õpilastest hindas kõige raskemateks küsimusteks 5. küsimust, mis kontrollis teadmisi teemal tsüklid. See on ka vastavuses arvestusliku testi punktide jaotusega küsimuste kaupa. Veel hinnati kõige raskemateks küsimusteks 6. küsimust (34,9% vastanutest), 4. küsimust (12,8% vastanutest) ning 3. küsimust (9,2% vastanutest). 1. ja 2. küsimust ei pidanud mitte ükski küsitlusele vastanu kõige keerulisemateks küsimusteks ning 30,3% vastanu arvates ei olnud ükski küsimus raske.

Vabas vormis esitatud kommentaaride hulk oli väike. Tudengid tõid välja järgmist:

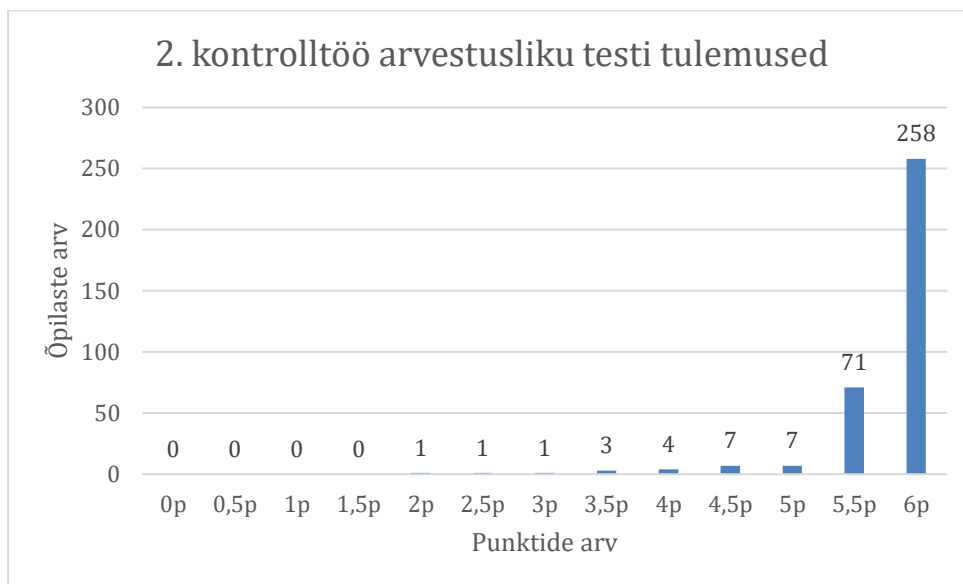
1. tsüklite teema valmistas probleeme ning seda tasuks loengutes/praktikumides rohkem käsitleda (esines 4 korda);
2. hakkasin üle mõtlema ja unustasin asjad ära (esines 2 korda);
3. küsimused olid kohmakalt sõnastatud (esines 2 korda);
4. test ei olnud keeruline (esines 2 korda);
5. faili ridade lugemine tekitas kahtlusi, aga muidu sai test kiiresti läbitud (esines 1 kord);
6. väga vähe oli aega (esines 1 kord);
7. oleks tahtnud rohkem vabadust, kuidas küsimusi lahendada (esines 1 kord);
8. koodi kirjutamine on keeruline, kui ei ole võimalik saada kohest tagasisidet, et kas süntaks on õige või mitte (esines 1 kord);
9. lävend võiks olla väiksem kui 100% (esines 1 kord);
10. test on ebavajalik (esines 1 kord).

Järgnevalt tutvustatakse 2. kontrolltöö arvestusliku testi analüüsi tulemusi.

4.2 2. kontrolltöö arvestusliku testi analüüs

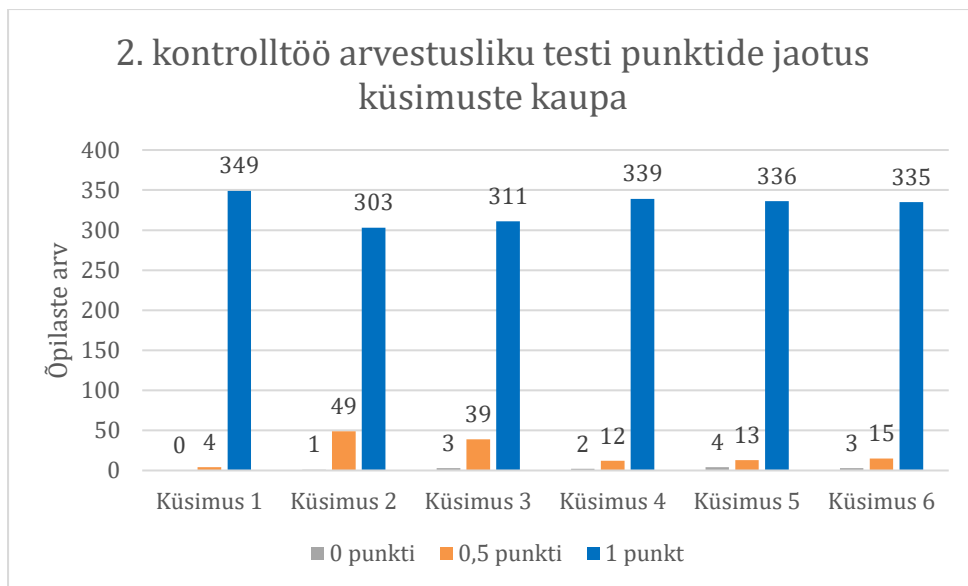
2. kontrolltöö arvestuslikku testi sooritas 353 õpilast. Test koosnes kuuest lühivastusega küsimusest. Iga küsimuse korral toimus õpilastele küsimuse valimine juhuslikult kolme erineva variandi hulgast (erandlikult oli küsimusel 3a omakorda erinevad variandid, kus muutusid funktsiooni `range` argumentid). 1. küsimuse teemaks oli järjendi element ja indeks, 2. küsimuse teemaks oli järjendi läbimine, 3. küsimuse teemaks oli kahekordne tsükkel, 4.

küsimuse teemaks oli maatriksid, 5. küsimuse teemaks oli sõnastikud ning 6. küsimuse teemaks oli ennikud ja hulgad. Kõigi kuue küsimuse eest oli võimalik teenida kuni üks punkt ehk kogu testi peale oli maksimaalne võimalik punktisumma kuus punkti. Testi edukaks sooritamiseks oli vaja kokku koguda vähemalt 5,5 punkti 6-st. Sellega sai hakkama 329 üliõpilast, kes said testi tulemuseks arvestatud. 24 õpilast said kokku vähem kui 5,5 punkti ning nende tulemuseks märgiti mittearvestatud. Sarnaselt 1. kontrolltööga kehtis reegel, et kui 2. kontrolltöö arvestusliku osa tulemuseks oli mittearvestatud, siis kogu kontrolltöö loeti mittearvestatuks ja arvestuse saamiseks pidi järeltööl arvestusliku testi uuesti sooritama. Kontrolltöö järeltööd käesolevas töös ei analüüsitud. 2. kontrolltöö arvestusliku testi tulemused on kujutatud joonisel 5.



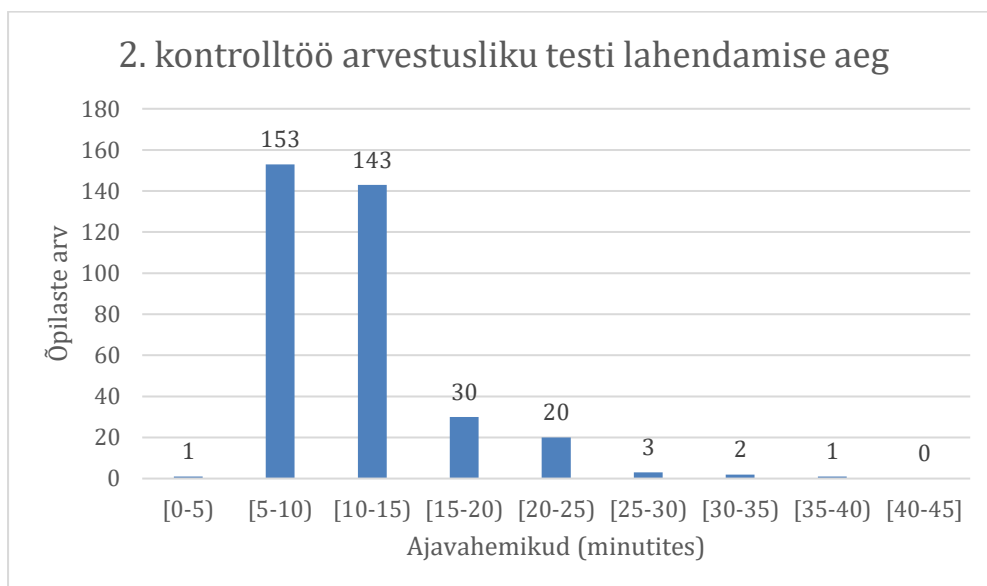
Joonis 5. 2. kontrolltöö arvestusliku testi tulemused.

Kõige enam kaotati punkte teise ja kolmanda küsimusega, kõige lihtsamaks küsimuseks osutus esimene küsimus ning ülejäänud küsimuste puhul oli punktide jaotus üsnagi võrdne. Punktide jaotust küsimuste kaupa iseloomustab joonis 6.



Joonis 6. 2. kontrolltöö arvestusliku testi punktide jaotus küsimuste kaupa.

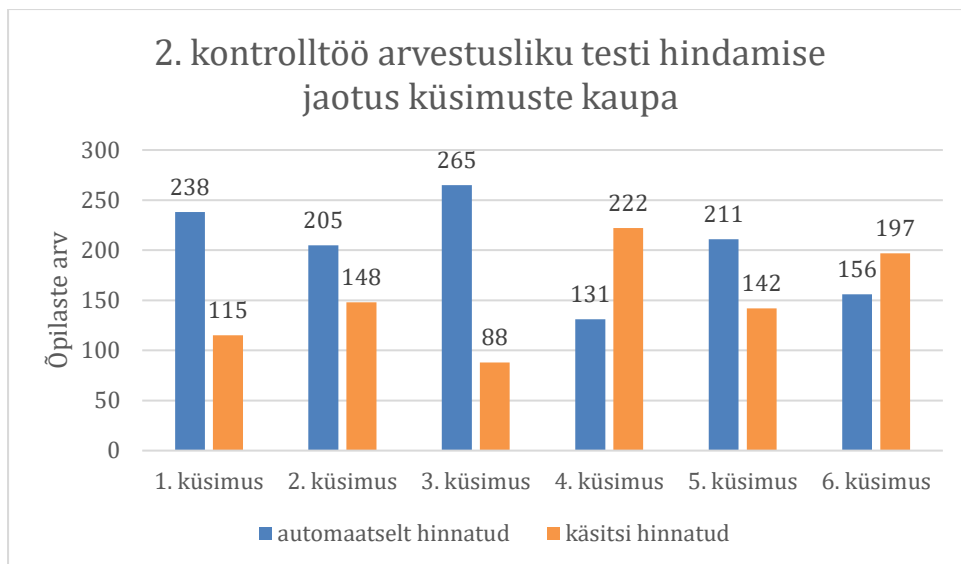
Testi sooritamiseks oli ette nähtud aega kuni 45 minutit. Õpilaste lahendamiseks kulunud aega iseloomustab joonis 7.



Joonis 7. 2. kontrolltöö arvestusliku testi lahendamiseks kulunud aeg.

Keskmiselt läks testi sooritamiseks aega 11 minutit ja 38 sekundit. Kõige kiirem sooritus oli 4 minutit ja 52 sekundit, kõige pikem sooritus oli 35 minutit ja 12 sekundit.

2. kontrolltöö arvestusliku testi puhul hinnati käsitsi üle ligikaudu 43% vastustest. Automaatselt ja käsitsi hinnatud küsimuste jaotust iseloomustab joonis 8.



Joonis 8. 2. kontrolltöö arvestusliku testi hindamise jaotus küsimuste kaupa.

Järgmiseks tutvustatakse 2. kontrolltöö arvestusliku testi analüüsi tulemusi küsimuste kaupa.

4.2.1 2. kontrolltöö arvestusliku testi analüüs küsimuste kaupa

2. kontrolltöö arvestuslikus testis esinenud küsimused on töö lisades (vt lisa 2).

Kokkuvõtlik tabel, mis koondab 2. kontrolltöö arvestusliku testi iga küsimuse analüüsi, on esitatud tabelis 3. Tabelis 3 esitatud veerud tähistavad järgmist:

1. A – küsimuse nimi;
2. B – vastuste koguarv;
3. C-E – lõplik punktide jaotus (vastavalt 0, 0,5 ja 1);
4. F – õigeks loetud vastuste protsent kõigist vastustest;
5. G – automaatselt hinnatud vastuste arv;
6. H – käsitsi üle hinnatud vastuste arv;
7. I – käsitsi üle hinnatud vastuste protsent kõigist vastustest;
8. J-L – käsitsi üle hinnatud vastuste punktide jaotus (vastavalt 0, 0,5 ja 1);
9. M – automaatkontrolli poolt mittetuvastatud õigete vastuste arv;
10. N – automaatkontrolli poolt mittetuvastatud õigete vastuste protsent kõigist õigetest vastustest.

Tabel 3. 2. kontrolltöö arvestusliku testi iga küsimuse analüüsi tulemusi koondav tabel.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|-------------------------------------|-----|---|----|-----|--------|-----|-----|-------|---|----|-----|----|-------|
| 1a. Värvid | 108 | 0 | 0 | 108 | 100.0% | 90 | 18 | 16.7% | 0 | 0 | 18 | 0 | 0.0% |
| 1b. Paarisarv | 115 | 0 | 1 | 114 | 99.1% | 105 | 10 | 8.7% | 0 | 1 | 9 | 0 | 0.0% |
| 1c. Värvid | 130 | 0 | 3 | 127 | 97.7% | 43 | 87 | 66.9% | 0 | 3 | 84 | 0 | 0.0% |
| 2a. Negatiivne täisarv | 115 | 0 | 4 | 111 | 96.5% | 79 | 36 | 31.3% | 0 | 3 | 33 | 3 | 2.7% |
| 2b. Vastand arvud | 112 | 0 | 37 | 75 | 67.0% | 20 | 92 | 82.1% | 0 | 34 | 58 | 33 | 44.0% |
| 2c. Ühe võrra suuremad | 126 | 1 | 8 | 117 | 92.9% | 106 | 20 | 15.9% | 0 | 2 | 18 | 6 | 5.1% |
| 3a. range-range | 115 | 1 | 13 | 101 | 87.8% | 74 | 41 | 35.7% | 0 | 13 | 28 | 0 | 0.0% |
| 3b. Täisarvude järjendid | 115 | 0 | 15 | 100 | 87.0% | 92 | 23 | 20.0% | 0 | 15 | 8 | 0 | 0.0% |
| 3c. Sõnede järjendid | 123 | 2 | 11 | 110 | 89.4% | 99 | 24 | 19.5% | 1 | 11 | 12 | 0 | 0.0% |
| 4a. Maatriksi elemendid rida-veerg | 128 | 1 | 9 | 118 | 92.2% | 19 | 109 | 85.2% | 1 | 8 | 100 | 0 | 0.0% |
| 4b. Maatriksi elemendid element | 109 | 0 | 0 | 109 | 100.0% | 85 | 24 | 22.0% | 0 | 0 | 24 | 1 | 0.9% |
| 4c. Vastasnurkade elemendid võrdsed | 116 | 1 | 3 | 112 | 96.6% | 27 | 89 | 76.7% | 0 | 3 | 86 | 22 | 19.6% |
| 5a. Lemmikviljad | 118 | 3 | 4 | 111 | 94.1% | 38 | 80 | 67.8% | 3 | 4 | 73 | 35 | 31.5% |
| 5b. Telefoninumbrid | 121 | 0 | 5 | 116 | 95.9% | 89 | 32 | 26.4% | 0 | 5 | 27 | 7 | 6.0% |
| 5c. Tooted | 114 | 1 | 4 | 109 | 95.6% | 84 | 30 | 26.3% | 1 | 4 | 25 | 11 | 10.1% |
| 6a. Jooksjad | 105 | 0 | 2 | 103 | 98.1% | 67 | 38 | 36.2% | 0 | 1 | 37 | 0 | 0.0% |
| 6b. Tooted | 117 | 1 | 7 | 109 | 93.2% | 74 | 43 | 36.8% | 1 | 7 | 35 | 1 | 0.9% |
| 6c. Ladu | 131 | 2 | 6 | 123 | 93.9% | 15 | 116 | 88.5% | 2 | 6 | 108 | 87 | 70.7% |

Üle kolmandiku vastustest kontrolliti käsitsi üle küsimuste 1c, 2b, 3a, 4a, 4c, 5a, 6a, 6b ja 6c puhul. Neljal juhul, küsimuste 2b, 4c, 5a ja 6c korral, oli üheks põhjuseks automaatkontrolli ebaefektiivsus õigete vastuste tuvastamisel. Lisaks nõudis küsimuse 2b üks osa järjendi elementide asendamist nende vastand arvudega, mis ei ole seotud 2. küsimuse teemaga. See põhjustas osa tudengitele raskusi. Ka teemakohane osa, mis nõudis tsüklipäise koostamist, osutas mitmekümnele tudengile probleeme. Need olid põhjused, miks küsimuse 2b puhul esines palju käsitsi hindamist.

Küsimus 1c osutus keeruliseks küsimuseks: kuigi suurem osa tudengitest valdas indekseerimise teemat, siis ei osatud vastust järjendi kujul väljastada. Sellegipoolest hinnati vastused valdavalt õigeks.

3. küsimuse puhul esines palju osaliselt õigeid vastuseid. See oli suuresti tingitud sellest, et kõikides küsimustes oli vaja täita 3 lahtrit, millega suurenes eksimisvõimalus. Seejuures esines küsimusel 3a erinevaid variante, kus muutus vahemik, mis lõigust tsüklimuutuja väärtusi omandas. Muus osas jäi küsimus samaks. Küsimuses 3c arvestas automaatne hindamissüsteem õigeks vastused, kus oli ekraanile väljastus ümbritsetud jutumärkidega. Seega jutumärkide kasutamist küsimuses 3c siin töös veana ei arvestata.

Küsimus 4a nõudis kahe lahtri täitmist, mis osutusid tudengitele keerukaks. Seetõttu esines paljudes vastustes puudujääke ning vastused hinnati käsitsi üle. Lisaks automaatkontrolli ebaefektiivsusele küsimuse 4c puhul võis üliõpilasi segadusse ajada ka küsimuse sõnastus: küsimuses palutakse täiendada programmi ning oodatakse vastust, kus kasutatakse muutujaid m ja n. Samas ei ole küsimuses näha, et muutujad m ja n oleksid eelnevalt programmis defineeritud. Peamiselt jäid tudengid hätta maatriksi viimase elemendi indekseerimisega.

Olgugi et küsimuse 5a puhul jäi automaatkontrollil 31,5% õigetest vastustest tuvastamata, esitas see sooritajatele väljakutse, millega kaasnes üksjagu eksimusi. Seda ilmestab ka fakt, et kolm vastust olid puudulikud ning hinnati valeks.

Kõigi kolme 6. küsimuse variandi puhul hinnati üle kolmandiku vastustest käsitsi üle. Küsimuses 6a oli sooritajatel vaja täita 2 lahtrit, 6b ja 6c korral 3 lahtrit. Esile tõusis küsimus 6c, kus 88,5% vastustest hinnati käsitsi. Seejuures ei tuvastanud automaatkontroll ära 87 õiget vastust.

Sarnaselt 1. kontrolltööga vaadati 2. kontrolltöö arvestusliku testi analüüsi käigus üle kõik sellised testiküsimuste vastused, mis olid kas käsitsi üle hinnatud või olid automaatsel hindamisel kas valeks või osaliselt õigeks hinnatud (lõplik tulemus). Vastustes esinenud vead lahterdati veatüüpide järgi ning loeti kokku vastavate vigade esinemiste arvud, et saada ülevaade, mille vastu õpilased enim eksivad. Lisaks võrreldi samasse kategooriasse kuulunud vigade puhul vastuste lõplikke tulemusi. Mitmel juhul oli tulemusi erinevalt hinnatud. 2. kontrolltöö arvestusliku testi igas küsimuses esinenud vead on töö lisades (vt lisa 3). Lõplikke tulemusi võrreldi ka sügisel kasutatud hindamisskeemiga (vt lisa 4), et kontrollida, kas hindamine on üheselt mõistetav. Hindamises esines mitmeti mõistmist, kus töö autor oleks hindamisskeemi alusel vastuseid teistmoodi hinnanud.

4.2.2 2. kontrolltöö tagasiside analüüs

2. kontrolltöö tagasisideküsitlusele vastas 40 üliõpilast. 14 tudengit hindasid 2. kontrolltöö arvestuslikku testi pigem lihtsaks, 5 lihtsaks ning 1 väga lihtsaks. 13 tudengi arvates oli test pigem keeruline ning 6 õpilase arvates keeruline. Väga keeruliseks ei hinnanud testi mitte ükski küsitlusele vastanu ning üks vastanu jättis hinnangu andmata. Kuna 2. kontrolltöö tagasisidele vastanud oli ligikaudu 2,5 korda vähem kui 1. kontrolltöö puhul, siis mõlema kontrolltöö arvestusliku testi raskusastme hinnanguid käesolevas töös võrdlema ei hakatud.

Kõige raskemate küsimuste hindamisel toodi välja kõik küsimused. Enim hinnati rasketeks 5. ja 6. küsimust (13 vastanut), nendele järgnesid 4. küsimus (10 vastanut), 2. küsimus (6 vastanut), ning 1. ja 3. küsimus (mõlemal juhul 2 vastanut). 9 tudengi arvates ei olnud ükski küsimus raske. Testi punktide jaotusest küsimuste kaupa tuleb aga välja, et enim probleeme valmistasid 2. ja 3. küsimus.

Vabas vormis esitatud kommentaaride hulk oli sarnaselt 1. kontrolltöö tagasisidega väike. Õpilased tõid välja järgmist:

1. eestikeelse terminoloogia ümbertõlkimine oli raske (esines 1 kord);
2. test oli okei (esines 1 kord);
3. kõik on väga raske (esines 1 kord);
4. isegi kui põhimõtet tean, siis ei tule õiged sümbolid meelde (esines 1 kord);
5. kardeti lohakusvigu (esines 1 kord).

Järgnevalt kirjeldatakse lõppküsitluse analüüsi tulemusi.

4.3 Lõppküsitluse analüüs

Kursuse „Programmeerimine“ lõppküsitlusele vastas 92 üliõpilast. Vabas vormis kommentaaridena tõid tudengid kõige enam välja seda, et arvestusliku osa hindamissüsteem, kus arvestuse saamiseks on vaja vastata kõikidele küsimustele õigesti, hirmutas ning tekitas lisapingeid ja stressi (esines 13 korda). Veel tõid õpilased välja järgmist:

1. ei mõistetud seda, miks arvestuslik osa vajalik on (esines 4 korda);
2. ei meeldinud, et eksamile pääsemise tingimuseks oli see, et arvestuslik osa peab olema arvestatud (esines 2 korda);
3. arvestuslik osa meeldis (esines 1 kord);
4. arvestusliku osa ajalimiit võiks olla väiksem, sest muidu kulutatakse testi peale liiga kaua aega (esines 1 kord);
5. arvestuslikus osas võiks olla lubatud materjalide kasutamine, sest ka programmeerijad kasutavad palju googeldamist (esines 1 kord);
6. testi küsimused olid halvasti sõnastatud (esines 1 kord).

Edasi tutvustatakse loodud kontrolltööde arvestuslike testide juhendite sisu.

4.4 Kontrolltööde arvestuslike testide koostamise juhendid

Kontrolltööde arvestusliku testi koostamise juhendis on esmalt välja toodud testi eesmärk. Testi eesmärk 1. kontrolltöö puhul on kontrollida järgnevate õpiväljundite saavutamist vähemalt minimaalsel tasemel:

- tunneb põhilisi programmeerimiskonstruktsioone: muutuja, avaldis, omistuslause, tingimuslause, tsükkel, alamprogramm, andmevahetus kasutaja ja failidega;
- tunneb põhilisi andmetüüpe ja -struktuure (täis- ja ujukomaarvud, tõeväärtused, sõned) ning vastavaid standardoperatsioone;
- oskab analüüsida programmi töö käiku ja programmi täiendada.

Testi eesmärk 2. kontrolltöö puhul on kontrollida järgnevate õpiväljundite saavutamist vähemalt minimaalsel tasemel:

- tunneb põhilisi programmeerimiskonstruktsioone: tsükkel (sh kahekordne tsükkel);
- tunneb põhilisi andmetüüpe ja -struktuure (järjendid, maatriksid, ennikud, hulgad, sõnastikud) ning vastavaid standardoperatsioone;
- oskab analüüsida programmi töö käiku ja programmi täiendada.

Arvestuslik test ei kontrolli kaht viimast õppeinfosüsteemis väljatoodud õpiväljundit [1], sest ühe neist realiseerib kontrolltöö punktiline osa ning teise kursuse vältel rühmatöö raames loodud projekt. Lisaks ei kontrolli arvestuslikud testid esimese kolme õppeinfosüsteemis väljatoodud õpiväljundi järgnevate osade saavutamist: oskab kasutada põhilisi programmeerimiskonstruktsioone, oskab kasutada põhiliste andmetüüpide ja -struktuuride vastavaid standardoperatsioone, oskab üksikasjalikult selgitada programmi töö käigu ning programmi muuta ja edasi arendada. Arvestuslikus testis kasutatavad küsimuste tüübid ei võimalda neid tingimusi täita. Küll aga kaetakse need õpiväljundite osad kursuse muude tegevuste käigus, nagu näiteks kontrolltööde punktilises osas, projektis ning kodutöodes. Õpiväljundite osad, mida kursusel ei käsitleta enne 2. kontrolltöö toimumist, võiksid sisalduda eksamitöö arvestuslikus osas.

Kuigi kontrolltöö punktiline osa kontrollib ka teatud määral eelnevalt kontrolltööde juhendites välja toodud õpiväljundite saavutamist, siis on arvestuslik test kontrolltöös vajalik. Kontrolltöö punktilisel osal puudub lävend, mis tähendab, et arvestuslikku testi arvestamata võib juhtuda, et õpilane läbib kursuse omandamata mõningaid kursusel õpetatud teemasid ning seeläbi ka

kõiki õpiväljundeid saavutamata. Seega on arvestuslik test õpiväljundite saavutamiseks kursusel vajalik.

Seejärel on juhendis väljatoodud testis sisalduvad teemad. 1. kontrolltöö puhul on teemadeks enne 1. kontrolltöö toimumist kursusel läbitud teemad. 2. kontrolltöö puhul on teemadeks 1. kontrolltöö järel ja enne 2. kontrolltöö toimumist läbitud teemad. Mõlema kontrolltöö arvestuslikus testis on kuus teemat, mis jagunevad kuueks teemaplokiks. Iga teemaplokk koosneb kahest küsimusest. Juhendis on välja toodud märksõnadena olulisimad aspektid, mida etteantud teemade puhul küsimuste koostamisel arvesse võtta.

Sügisel toimunud kursuse kontrolltööde arvestuslike testide analüüsi ning tagasisideküsitluse tulemustest lähtuvalt on muudetud hindamissüsteemi, mis on mõlemas juhendis kajastatud. Seni oli arvestuse saamiseks vaja koguda 5,5 punkti 6-st. Samas tudengitele antud testi lävendit selliselt ei tutvustatud. Õpilastele sõnastati lävend selliselt, et kõikidele küsimustele tuleb vastata õigesti, aga väiksemad eksimused tulemusele mõju ei avalda. Samas tõid mitmed tudengid tagasisides välja, et justkui igasugune eksimisruum testis puudub. Tagasisideküsitlusest tuli välja, et selline lävend hirmutas üliõpilasi ning põhjustas lisapingeid. Lisaks mõjutas lävend ka hindajaid, kes olid hindamisel väga leebed. Uue juhendi kohaselt on arvestuse saamiseks vaja vastata iga teemaploki puhul vähemalt ühele küsimusele õigesti. Kuna iga teemaploki kaks küsimust kontrollivad teadmisi samal teemal, siis vähemalt ühele küsimusele õigesti vastates võib eeldada, et tudeng on saavutanud teadmised vastaval teemal vähemalt minimaalsel tasemel. Selline formaat vähendab tudengitel lisapingete tekkimist, sest arvestuse saamiseks on olemas arvestatav eksimisruum.

Juhendis on välja toodud ka suunised küsimuste koostamiseks. Vastavalt programmeerimise algkursuste eksamitel enamlevinud küsimuste tüüpidele on testi koostamiseks soovituslik kasutada vastuse valimise ja sisestamise küsimusi. Vastuse valimise küsimuste all mõeldakse siin küsimusi, kus võimalikud vastused on ette antud. Näiteks küsimused, kus etteantud vastusevariantide hulgast tuleb valida õige(d) või kus etteantud vastused tuleb õigesse järjekorda panna. Vastuse sisestamise küsimuste all mõeldakse küsimusi, kus vastuse lahtrisse tuleb sisestada üks sõna, arv või fraas, sellised võivad olla näiteks ka lünkade täitmise küsimused, mis nõuavad mitme lahtri täitmist.

Testide puhul on üheks eesmärgiks veel ka käsitsi üle hindamise minimeerimine. Selleks on tarvilik vastuse sisestamist nõudvate küsimuste vähendamine või täieliku automaathindamise tagamiseks selliste küsimuste mittekasutamine. Tekstilist vastust sisaldavaid küsimusi võib

kasutada arvulist väärtust nõudvate vastuste puhul, sest neid on lihtne automaatselt kontrollida. Muudel puhkudel on tekstilist vastust automaatselt keeruline kontrollida, sest tudengid võivad lihtsasti süntaksiga eksida. Samas ei tohiks süntaksialaseid eksimusi veana arvestada. Programmeerides annab programmeerimiskeskond sellistel puhkudel kohest tagasisidet näiteks veateadete näol ning seejärel on võimalik kas keskkonna poolt väljapakutud parandusi tehes või internetti kasutades vigu parandada. Arvestusliku testi sooritamisel on aga keelatud abimaterjalide kasutamine ning teistega suhtlemine, lubatud on kasutada vaid paberit ja pliiatsit. Seetõttu ei ole otstarbekas süntaksialaselt tudengite mälu kontrollida. Lisaks on hiljem keeruline hinnata, kas vale süntaksi korral on õpilane küsimusele piisavalt õigesti vastanud. Kui aga testi koostaja hinnates on tarvilik taolise vastuse sisestamise küsimuse kasutamine, siis juhendis on välja toodud suunised, millest tuleb nende koostamisel lähtuda. Lisaks on juhendites välja toodud suunised ka vastuse valimise küsimuste puhul.

Juhendis on kirjas testi ajalimiit. Ajalimiiti ei muudetud, sest kiirematel sooritajatel on võimalik koheselt punktilise osa lahendamisele edasi liikuda. Küsimuste koostamise suuniste üks eesmärkidest on ka see, et testis kasutatavatele küsimustele vastamine võtaks vähem aega kui pikemate vastuste sisestamine või mitmete lahtrite täitmine.

Veel on juhendis kirjeldatud, millistele nõuetele peavad testiküsimused vastama, missugune on testi üldine ülesehitus ja milliseid põhimõtteid tuleb testi koostamisel järgida.

Juhendid edastati tagasisidestamiseks kursuse vastutavatele õppejõududele. Tagasiside põhjal teostati parandused ning lõplikud juhendid on leitavad töö lisades (vt lisa 5 ja lisa 6).

4.5 Kontrolltööde arvestuslike testide näidistestid

Lisaks mõlema kontrolltöö juhendile loodi testi koostamise näitlikustamiseks näidistestid. Näidistestid tuginesid juhendis välja toodud aspektidele ning on leitavad töö lisades (vt lisa 7 ja lisa 8).

Näidistestide küsimuste loomisel lähtuti analüüsitud küsimuste ning kursuse kodulehel olevate õppematerjalide sisust [3]. Lisaks uuriti 1. ja 2. kontrolltöö arvestuslikus testis esinenud vigu ja küsimuste sõnastust.

Tähelepanu pöörati sellele, et küsimused oleksid suures osas automaatselt hinnatavad. Vastuse sisestamise küsimuste korral, mis ei nõudnud arvulist vastust, prooviti oodatav vastus hoida konkreetse ja lühikesena. Pikemat sisendit nõudva küsimuse korral lähtuti sellest, et sarnase kirjaipildiga vastuste korral oli analüüsitud küsimuste käsitsi üle hinnatud vastuste hulk väike.

Prooviti luua uudseid testiküsimusi, mis erineksid 2022/2023 sügissemestril kontrolltööde arvestuslikes testides esinenud testiküsimustest. Küsimuste koostamisel jälgiti, et küsimuste sõnastus oleks selge ning küsimused ei oleks liialt keerulised. Seda seetõttu, et tegemist on sissejuhatava kursusega programmeerimisse ning kursusel on osalejaid, kel puudub varasem kokkupuude programmeerimisega. Lisaks on võrreldes eelneva testiformaadiga kaks korda rohkem testiküsimusi, mistõttu ei tohiks küsimustele vastamine võtta palju aega. Järgides juhendites välja toodud nõudeid ning suuniseid küsimuste koostamisel, on eeldused selleks tingimuseks täidetud.

Veenduti, et iga küsimus kontrolliks vähemalt ühe juhendis väljatoodud õpiväljundi saavutamist ning kõik testiküsimused kataksid üheskoos kõik juhendis kirjeldatud õpiväljundid.

Näidistestid edastati sarnaselt juhenditele kursuse vastutavatele õppejõududele tagasisidestamiseks ning tagasiside põhjal viidi sisse parandused.

5. Arutelu

Selles peatükis kirjeldatakse töö autori soovitusi juhendite rakendamiseks ning ideid töö võimalikeks edasiarendusteks.

Autoripoolne soovitus on loodud juhendeid tutvustada kõikidele 2023/2024 toimuva kursuse „Programmeerimine“ praktikumijuhendajatele. Juhendite ning näidistestide näitel võiks praktikumijuhendajaid kaasata aktiivselt uute testiküsimuste loomisse. Selliselt oleks garanteeritud piisav hulk uuele testiformaadile sobivaid testiküsimusi, et neid jätkuks lisaks põhitööle nii järeltööde kui ka järgnevate aastate jaoks.

Analüüsi käigus tuvastatud õpilaste töös esinenud vigu tuleks aine vastutavatel õppejõududel lähemalt uurida. Selliselt on võimalik kindlaks teha teemad, mis valmistasid õpilastele enim probleeme ning nendele järgmisel, 2023/2024 sügissemestril toimuval kursusel rohkem tähelepanu pöörata.

Kui käesoleva bakalaureusetöö raames valminud juhendid leiavad aktiivset kasutust, siis võiks nende baasil luua ka juhendid kursuse eksami arvestuslikule testile.

Kokkuvõte

Bakalaureuse töö käigus uuendati Tartu Ülikooli kursuse „Programmeerimine“ 1. ja 2. kontrolltöö arvestusliku osa teste ning loodi vastavad juhendid uuendatud testide loomiseks, et testide koostamine oleks kursuse õppejõududele üheselt mõistetav. Juhendite näitlikustamiseks valmisid mõlema kontrolltöö arvestusliku testi tarbeks näidistestid.

Testide uuendamiseks tutvuti esmalt kursuse „Programmeerimine“ sisu ja ülesehitusega ning arvestusliku osa testidega. Seejärel uuriti, milliseid küsimusetüpe kasutatakse programmeerimise algkursuste hindamisel ning tutvuti lähemalt neis enamesinenud küsimuste tüüpidega. Järgneva sammuna analüüsiti 2022/2023 sügissemestril toimunud kursuse „Programmeerimine“ arvestusliku osa teste ning tagasisideküsitluste vastuseid. Testide analüüsi eesmärgiks oli tuvastada puudujääke nii testiküsimustes, hindamises kui ka testiformaadis. Lisaks tuvastati õpilaste testiküsimuste vastustes esinenud vead ning prooviti leida eksimuste taga olevaid põhjuseid. Tagasisideküsitluste analüüsis keskenduti tudengite tagasisidele, mis puudutas kontrolltööde arvestusliku osa teste. Tulemuste põhjal muudeti mõlema kontrolltöö arvestusliku osa testiformaati ning loodi kontrolltööde arvestusliku osa testide juhendite ja näidistestide mustandid. Mustandid edastati tagasisidekamamiseks kursuse vastutavatele õppejõududele. Saadud tagasiside põhjal viidi sisse parandused. Töö tulemusena valmisid juhendid mõlema kontrolltöö arvestusliku osa uuendatud testi koostamiseks, et testide koostamine oleks õppejõududele üheselt mõistetav, ning juhendite näitlikustamiseks mõlema kontrolltöö arvestusliku osa tarbeks näidistestid.

Loodud juhendites on välja toodud, mis on arvestusliku testi eesmärk, millistele nõuetele peavad testiküsimused vastama, missugustest teemadest peab test koosnema, missugune on testi üldine ülesehitus ja hindamine ning milliseid põhimõtteid tuleb testi koostamisel järgida. Lisaks on juhendites kirjas suunised uute testiküsimuste loomiseks.

Näidistestid tuginevad loodud juhenditele ning mõlema kontrolltöö arvestusliku testi illustreerimiseks on näidistestis iga teema kohta välja toodud üks küsimus.

Kuna juhendite loomise hetkeks oli 2022/2023 sügissemestril toimunud programmeerimise kursus juba lõppenud, siis õpilaste tagasiside uuendatud testidele puudub. On võimalik, et järgneval kursusel, mis leiab aset 2023/2024 sügissemestril, tuleb pärast kontrolltööde toimumist juhendeid parandada/täiendada.

Viidatud kirjandus

- [1] Tartu Ülikooli õppeinfosüsteem. Programmeerimine (6 EAP) LTAT.03.001 2022/2023 sügis. <https://ois2.ut.ee/#/courses/LTAT.03.001/version/5ce5cc23-3331-cfb9-2784-3f820084cb31/details> (10.12.2022)
- [2] Aug, T. Kontrolltööde koostamise ja hindamise juhendite loomine Tartu Ülikooli kursusele „Programmeerimine“. *Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö*, 2022.
https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=74278&year=2022 (26.02.2023)
- [3] Kursuse „Programmeerimine“ kodulehekülg.
<https://courses.cs.ut.ee/2022/programmeerimine/fall> (10.12.2022)
- [4] Tartes T. Mis on Moodle? *Tartu Ülikooli e-õppe ajakiri*, 2010.
<https://etu.ut.ee/kevad-2010/mis-on-moodle/> (10.12.2022)
- [5] Tartu Ülikooli Moodle. Programmeerimine (LTAT.03.001).
<https://moodle.ut.ee/course/view.php?id=500> (10.12.2022)
- [6] Simon, Sheard, J., Carbone, A., Chinn, D., et al. Introductory programming: examining the exams. *Proceedings of the Fourteenth Australasian Computing Education Conference (ACE2012)*, Vol. 123, 2012, 61-70.
- [7] Petersen, A., Craig, M., Zingaro, D. Reviewing CS1 Exam Question Content. SIGCSE '11: *Proceedings of the 42nd ACM technical symposium on Computer science education*, 2011, 631-636.
- [8] Crisp G. Interactive e-assessment – practical approaches to constructing more sophisticated online tasks. *Journal of Learning Design*, Vol. 3, No. 3, 2010, 1-10.
- [9] Walker, D. J., Topping, K., Rodrigues, S. Student reflections on formative e-assessment: expectations and perceptions. *Learning, Media and Technology*, Vol. 33, No. 3, 2008, 221-234.
- [10] Shuhidan, S., Hamilton, M., D’Souza, D. Instructor perspectives of multiple-choice questions in summative assessment for novice programmers. *Computer Science Education*, Vol. 20, No. 3, 2010, 229–259.
- [11] Bacon, D. R. Assessing Learning Outcomes: A Comparison of Multiple-Choice and Short-Answer Questions in a Marketing Context. *Journal of Marketing Education*, Vol. 25, No. 1, 2003, 31-36.

- [12] Impara, J. C., Foster, D. Item and Test Development Strategies to Minimize Test Fraud. *Handbook of Test Development*, 2006, 91-114.
- [13] Simon, Snowdon, S. Multiple-choice vs free-text code-explaining examination questions. *Koli Calling '14: Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, 2014, 91-97.

Lisad

1. 1. kontrolltöö arvestusliku testi hindamisskeem 2022

1. küsimus

Omistamisavaldis, aga ei kasuta ülesande andmeid 0,5

Sobiv arvutus, aga tulemust ei omistata muutujale 0,5

2. küsimus

Punktide arv vastavalt õigete lahtrite arvule. Peaaegu kõik vastused olid õiged.

3. küsimus

Lause üldkuju +0,5

Tingimuse üldkuju +0,5

4. küsimus

4a.

Päis +0,5

Sisu +0,5

4b.

Pole rohkem kui üks taandeviga 1

Vigane, kuid on näha korrektset struktuuri 0,5

Taanded suvaliselt 0

4c.

Tingimus +0,5

Väljakutse +0,5

5. küsimus

5a.

Vastusest nähtub, et tsükli sisu täidetakse mitu korda 1

Programmi sisu täidetakse ainult üks kord käskude õiges järjekorras 0,5

Programmi sisu täidetakse ainult üks kord käskude suvalises järjekorras 0

5b.

Esimene lahter +0,5

Teine lahter +0,5

Teises lahtris vigu +/- 3 ei arvesta. Kui viimane print loetakse asuvaks tsükli sees, siis seda viga ei arvesta.

5c.

Kombineeritud tsükli jätkamistingimus ja tingimuslause tingimus +0,5

Arvestatud, et tsükli sisu täidetakse korduvalt +0,5

Arvutus- ja loendamisvigu ei arvesta.

6. küsimus

1 - Vähemalt kaks kolmest failitegevusest õiged

0,5 - Üks kolmest failitegevusest õige

0 - Kõik failitegevused valed

Lugeda arvestatuks kõik need tulemused, mis on vähemalt 5,5/6.

2. 1. ja 2. kontrolltöö arvestusliku testi küsimused

1. kontrolltöö arvestusliku testi küsimused on leitavad siit:

<https://drive.google.com/file/d/16EK4Jetj6voj8bTdcUX3O5k1bJwX9gtv/view?usp=sharing>

2. kontrolltöö arvestusliku testi küsimused on leitavad siit:

https://drive.google.com/file/d/1ImIC080rCypZF7cV2mrQI7wt_WMrdyxw/view?usp=sharing

Lisadele ligipääsemiseks võtke ühendust kursuse „Programmeerimine“ vastutava õppejõu Tauno Paltsiga.

3. 1. ja 2. kontrolltöö arvestuslikes testides esinenud vead

1. kontrolltöö arvestuslikes testides esinenud vead on leitavad siit:

<https://drive.google.com/file/d/1T2b92hNdTvkj6w2HLP81sjCpSYCI3f8V/view?usp=sharing>

2. kontrolltöö arvestuslikes testides esinenud vead on leitavad siit:

<https://drive.google.com/file/d/1Byf6I-amGThVKJCLeaRESN178D6HjfWw/view?usp=sharing>

Lisadele ligipääsemiseks võtke ühendust kursuse „Programmeerimine“ vastutava õppejõu Tauno Paltsiga.

4. 2. kontrolltöö arvestusliku testi hindamisskeem 2022

1. küsimus

Oskab võtta järjestist elementi indeksi järgi: 1 p

Oskab võtta elementi, aga mujalt: 0,5 p

2. küsimus

Oskab organiseerida tsükli üle järjendi: +0,5 p

Teeb vahet elemendi ja indeksi vahel: +0,5 p

3. küsimus

Väljastatavad arvud õiged: +0,5 p

Ridade arv õige: +0,5 p

Arvutusvigu ja vigu ± 1 ei arvesta.

4. küsimus

Variandid a ja b.

Välimine tsükkel arvestab maatriksit: +0,5 p

Sisemine tsükkel arvestab välimist tsükli: +0,5 p

Variant c.

Maatriksist elemendi võtmine kahe indeksi abil: 1 p

Maatriksist elemendi võtmine ühe indeksi abil: 0,5 p

Maatriksist pole elementi võetud: 0 p

5. küsimus

Teeb vahet võtme ja väärtuse vahel: +0,5 p

Oskab sõnastikuga tehteid teha: +0,5 p

6. küsimus

Oskab tegutseda ennikuga: +0,5 p

Oskab tegutseda hulgaga: +0,5 p

Lugeda arvestatuks kõik need tulemused, mis on vähemalt 5,5/6.

5. Juhend 1. kontrolltöö arvestusliku testi koostamiseks

Juhend Tartu Ülikooli kursuse „Programmeerimine“ 1. kontrolltöö arvestusliku testi koostamiseks

I. 1. kontrolltöö arvestusliku testi eesmärgiks on kontrollida järgnevate õpiväljundite saavutamist vähemalt minimaalsel tasemel:

- tunneb põhilisi programmeerimiskonstruktsioone: muutuja, avaldis, omistuslause, tingimuslause, tsükkel, alamprogramm, andmevahetus kasutaja ja failidega;
- tunneb põhilisi andmetüüpe ja -struktuure (täis- ja ujukomaarvud, tõeväärtused, sõned) ning vastavaid standardoperatsioone;
- oskab analüüsida programmi töö käiku ja programmi täiendada.

II. Iga küsimus peab olema seotud vähemalt ühe õpiväljundiga punktis I, sh kõik testiküsimused peavad üheskoos katma kõik väljatoodud õpiväljundid punktis I.

III. Test koosneb 12 küsimusest ning ajalimiit on 45 minutit. Kiirematel sooritajatel peab olema võimalus pärast testi lõpetamist liikuda koheselt edasi kontrolltöö punktilise osa lahendamise juurde.

IV. Test koosneb kuuest teemaplokist, mis tuginevad kursusel vahetult enne kontrolltöö toimumist läbitud teemadele. Iga teemaplokk koosneb kahest küsimusest. Teemaplokkideks on:

- 1) avaldised ja lihtlauseid (muutujad, avaldised, omistuslause, andmetüübid, sisend ja väljund)
- 2) tingimuslauseid (if, if-else, if-elif-else, tingimuslauseid üksteise sees, tõeväärtustehted)
- 3) funktsioonid (funktsioonide defineerimine, rakendamine, lokaalsed ja globaalsed muutujad, parameetrid, väärtuse tagastamine)
- 4) korduslauseid (while- ja for-tsükkel)
- 5) sõnetöötlus (tehted sõnedega, sõnefunktsioonid)
- 6) lihtsam failitöötlus (failist lugemine, faili kirjutamine (mõlemad ilma tsükliita))

V. Teemaplokid võiksid töös olla järjestatud vastavalt kursusel käsitletud järjekorrale ning teemaploki mõlemad küsimused võiksid olla järjestikku. Selliselt on testil loogiline ülesehitus.

VI. Küsimused võiksid lähtuda vastava teema olulisimatest aspektidest, mis on välja toodud ülal teemaplokkide kirjeldustes punktis IV.

VII. Töösse küsimusi valides tuleb kontrollida, et kõik küsimused oleksid üheselt mõistetavad ning ühe teema küsimused oleksid sarnase raskusastmega. Kui ei ole, tuleks neid siis muuta, need kustutada või nende asemel luua uued. Kui küsimuse lahendamiseks on tarvilik kasutada

küsimuse püstituses välja toodud muutujaid, siis peavad need olema küsimuse tekstis selgelt välja toodud ning eristatud. Selleks on soovituslik kasutada paksu kirja.

VIII. Töösse võib valida olemasolevaid küsimusi Moodle'i küsimustepangast, kuid tuleb veenduda, et neid ei ole eelneval aastal kasutatud. Soovituslik on testiküsimusi iga-aastaselt uuendada.

IX. Iga küsimuse kohta peab töös olema 3 erinevat varianti, et vähendada mahategemist. Küsimuse erinevad variandid peavad olema sama raskusastmega. Küsimus valitakse töösse juhuslikult kolme erineva variandi seast.

X. Järeltöös ei tohi olla samad küsimused, mis olid põhitöös. Seega tuleb küsimustepangast küsimusi valides veenduda, et järeltöö küsimused ei kattu põhitöö küsimustega. Põhitööga samasuguseid küsimusi võib kasutada juhul, kui neid muudetakse.

XI. Testi tulemuseks märgitakse „arvestatud“, kui iga teemaploki puhul on vastatud vähemalt ühele küsimusele õigesti. See tähendab, et küsimuse puhul, millel on mitu õiget vastusevarianti, peavad kõik õiged ja ainult õiged vastusevariandid olema valitud. Sellest lähtuvalt võib iga küsimuse tulemuse arvestada kas õigeks või valeks (1 või 0 punkti).

Suunised küsimuste koostamiseks

Testi koostamiseks on soovituslik kasutada vastuse valimise ja sisestamise küsimusi. Vastuse valimise küsimuste all mõeldakse siin küsimusi, kus võimalikud vastused on ette antud. Näiteks küsimused, kus etteantud vastusevariantide hulgast tuleb valida õige(d) või kus etteantud vastused tuleb õigesse järjekorda panna. Vastuse sisestamise küsimuste all mõeldakse küsimusi, kus vastuse lahtrisse tuleb sisestada üks sõna, arv või fraas, sellised võivad olla näiteks ka lünkade täitmise küsimused, mis nõuavad mitme lahtri täitmist.

Testide puhul on üheks eesmärgiks veel ka käsitsi üle hindamise minimeerimine. Selleks on tarvilik vastuse sisestamise küsimuste vähendamine või täieliku automaathindamise tagamiseks nende mittekasutamine.

Tekstilist vastust nõudvaid sisestamise küsimusi võib kasutada arvulist väärtust nõudvate vastuste puhul, sest neid on lihtne automaatselt kontrollida. Muudel puhkudel ei tohi tekstiliste vastuste korral süntaksialaseid vigu veana arvestada. Sellistel puhkudel on tarvilik koostada üheselt mõistetavad hindamiskriteeriumid ja need kõikidele hindajatele edastada.

Vastuse sisestamise küsimuste koostamisel tuleb lähtuda lisaks punktis II ja VII mainitule järgmistest aspektidest:

- ühe küsimuse kohta peab täidetavate lahtrite arv olema minimaalne;
- lahtrite täitmine nõuab ainult vastava teemaploki laseid teadmisi;
- lahtri täitmiseks nõutav sisend peab olema lühike ja konkreetne.

Vastuse valimise küsimuste koostamisel peab lähtuma lisaks punktis II ja VII mainitule järgmistest aspektidest:

- küsimuse kirjelduses peab olema selgelt välja toodud, kas õigeid vastusevariante on üks või rohkem;
- vastusevariantide soovituslik arv on olenevalt küsimusest 3-5, et vähendada juhuslikul teel õige vastuse valimise tõenäosust ning säilitada peibutusvariantide kvaliteet;
- valed vastusevariandid ehk peibutusvariandid peaksid välja nägema õige vastusega võimalikult sarnased; eri variandid võiksid olla sõnastatud võimalikult ühtses stiilis ning selliselt, et pealiskaudsel lugemisel tunduksid nad kõik usutavad;
- valede vastusevariantide ehk peibutusvariantide koostamiseks võib lähtuda eelnevatel aastatel testides esinenud vigade näidetest, et valed vastusevariandid oleksid ajakohased.

6. Juhend 2. kontrolltöö arvestusliku testi koostamiseks

Juhend Tartu Ülikooli kursuse „Programmeerimine“ 2. kontrolltöö arvestusliku testi koostamiseks

I. 2. kontrolltöö arvestusliku testi eesmärgiks on kontrollida järgnevate õpiväljundite saavutamist vähemalt minimaalsel tasemel:

- Tunneb põhilisi programmeerimiskonstruktsioone: tsükkel (sh kahekordne tsükkel);
- tunneb põhilisi andmetüüpe ja -struktuure (järjendid, maatriksid, ennikud, hulgad, sõnastikud) ning vastavaid standardoperatsioone;
- oskab analüüsida programmi töö käiku ja programmi täiendada.

II. Iga küsimus peab olema seotud vähemalt ühe õpiväljundiga punktis I, sh kõik testiküsimused peavad üheskoos katma kõik väljatoodud õpiväljundid punktis I.

III. Test koosneb 12 küsimusest ning ajalimiit on 45 minutit. Kiirematel sooritajatel peab olema võimalus pärast testi lõpetamist liikuda kohe edasi kontrolltöö punktilise osa lahendamise juurde.

IV. Test koosneb kuuest teemaplokist, mis tuginevad kursusel vahetult pärast 1. kontrolltöö toimumist ja enne 2. kontrolltöö toimumist läbitud teemadele. Iga teemaplokk koosneb kahest küsimusest. Teemaplokkideks on:

- 7) järjendid I (järjendioperatsioonid, nt: `len`, `count`, `min` ja `max`, `sorted`, `sum`, `list`, `range`), loogilised operatsioonid (`in`), indeksid (k.a negatiivsed), järjendite viilutamine)
- 8) järjendid II (järjendioperatsioonid, nt: `index`, `append`, `extend`, `insert`, `remove`, `pop`, `clear`, `sort`, `reverse`, `copy`), järjendite läbimine, järjendite liitmine, elemendi lisamine järjendi lõppu, järjendi filtreerimine, järjendite kombineerimine, järjendi viit)
- 9) kahekordne tsükkel (maatriksid, läbimine elementhaaval, läbimine indeksitega, järjend ja funktsioon, failist järjendisse lugemine)
- 10) ennikud (loomine, indeksid, operatsioonid (`len`, `in`, `min` ja `max`), viilutamine)
- 11) hulgad (loomine, võrdsus, läbimine, elemendi lisamine ja kustutamine, hulga uuendamine ja tühjendamine, kopeerimine, operatsioonid (nt `len`, `min` ja `max`, `remove`, `update`, `pop`, `clear`, `copy`), hulgaoperatsioonid)
- 12) sõnastikud (andmete lisamine, läbimine, operatsioonid (`len`, `in`), kirje kustutamine, läbimine paaride hulgana (`items()`))

V. Teemaplokid võiksid töös olla järjestatud vastavalt kursusel käsitletud järjekorrale ning teemaploki mõlemad küsimused võiksid olla järjestikku. Selliselt on testil loogiline ülesehitus.

VI. Küsimused võiksid lähtuda vastava teema olulisimatest aspektidest, mis on välja toodud ülal teemaplokkide kirjeldustes punktis IV.

VII. Töösse küsimusi valides tuleb kontrollida, et kõik küsimused oleksid üheselt mõistetavad ning ühe teema küsimused oleksid sarnase raskusastmega. Kui ei ole, tuleks neid siis muuta, need kustutada või nende asemel luua uued. Kui küsimuse lahendamiseks on tarvilik kasutada küsimuse püstituses välja toodud muutujaid, siis peavad need olema küsimuse tekstis selgelt välja toodud ning eristatud. Selleks on soovituslik kasutada paksu kirja.

VIII. Töösse võib valida olemasolevaid küsimusi Moodle'i küsimustepangast, kuid tuleb veenduda, et neid ei ole eelneval aastal kasutatud. Soovituslik on testiküsimusi iga-aastaselt uuendada.

IX. Iga küsimuse kohta peab töös olema 3 erinevat varianti, et vähendada mahategemist. Küsimuse erinevad variandid peavad olema sama raskusastmega. Küsimus valitakse töösse juhuslikult kolme erineva variandi seast.

X. Järeltöös ei tohi olla samad küsimused, mis olid põhitöös. Seega tuleb küsimustepangast küsimusi valides veenduda, et järeltöö küsimused ei kattu põhitöö küsimustega. Põhitööga samasuguseid küsimusi võib kasutada juhul, kui neid muudetakse.

XI. Testi tulemuseks märgitakse arvestatud, kui iga teemaploki puhul on vastatud vähemalt ühele küsimusele õigesti. See tähendab, et küsimuse puhul, millel on mitu õiget vastusevarianti, peavad kõik õiged ja ainult õiged vastusevariandid olema valitud. Sellest lähtuvalt võib iga küsimuse tulemuse arvestada kas õigeks või valeks (1 või 0 punkti).

Suunised küsimuste koostamiseks

Testi koostamiseks on soovituslik kasutada vastuse valimise ja sisestamise küsimusi. Vastuse valimise küsimuste all mõeldakse siin küsimusi, kus võimalikud vastused on ette antud. Näiteks küsimused, kus etteantud vastusevariantide hulgast tuleb valida õige(d) või kus etteantud vastused tuleb õigesse järjekorda panna. Vastuse sisestamise küsimuste all mõeldakse küsimusi, kus vastuse lahtrisse tuleb sisestada üks sõna, arv või fraas, sellised võivad olla näiteks ka lünkade täitmise küsimused, mis nõuavad mitme lahtri täitmist.

Testide puhul on üheks eesmärgiks veel ka käsitsi üle hindamise minimeerimine. Selleks on tarvilik vastuse sisestamise küsimuste vähendamine või täieliku automaathindamise tagamiseks nende mittekasutamine.

Tekstilist vastust nõudvaid sisestamise küsimusi võib kasutada arvulist väärtust nõudvate vastuste puhul, sest neid on lihtne automaatselt kontrollida. Muudel puhkudel ei tohi tekstiliste

vastuste korral süntaksialaseid vigu veana arvestada. Sellistel puhkudel on tarvilik koostada üheselt mõistetavad hindamiskriteeriumid ja need kõikidele hindajatele edastada.

Vastuse sisestamise küsimuste koostamisel tuleb lähtuda lisaks punktis II ja VII mainitule järgmistest aspektidest:

- ühe küsimuse kohta peab täidetavate lahtrite arv olema minimaalne;
- lahtrite täitmine nõuab ainult vastava teemaploki alaseid teadmisi;
- lahtri täitmiseks nõutav sisend peab olema lühike ja konkreetne.

Vastuse valimise küsimuste koostamisel peab lähtuma lisaks punktis II ja VII mainitule järgmistest aspektidest:

- küsimuse kirjelduses peab olema selgelt välja toodud, kas õigeid vastusevariante on üks või rohkem;
- vastusevariantide soovituslik arv on olenevalt küsimusest 3-5, et vähendada juhuslikul teel õige vastuse valimise tõenäosust ning säilitada peibutusvariantide kvaliteet;
- valed vastusevariandid ehk peibutusvariandid peaksid välja nägema õige vastusega võimalikult sarnased; eri variandid võiksid olla sõnastatud võimalikult ühtses stiilis ning selliselt, et pealiskaudsel lugemisel tunduksid nad kõik usutavad;
- valede vastusevariantide ehk peibutusvariantide koostamiseks võib lähtuda eelnevatel aastatel testides esinenud vigade näidetest, et valed vastusevariandid oleksid ajakohased.

7. 1. kontrolltöö arvestusliku testi näidistest

1. kontrolltöö arvestusliku testi näidistest

1. Avaldised ja lihtlaused

Küsimus 1

Vali kõik õiged Pythoni programmid, mille käivitamisel salvestatakse kasutaja poolt sisestatud kodulinn muutujasse **linn** ning seejärel väljastatakse see ekraanile. Õigeid vastusevariante võib olla rohkem kui 1!

- 1) `linn = "Sisesta enda kodulinn:"`
`print(linn)`
- 2) `linn = str('Sisesta enda kodulinn: ')`
`print(linn)`
- 3) `linn == 'Sisesta enda kodulinn:'`
`print(linn)`
- 4) `linn == print(input("Sisesta enda kodulinn:"))`
- 5) `linn = input("Sisesta enda kodulinn:")`
`print(linn)`

Küsimus 2

Kirjuta Pythoni lause, mille täitmisel suureneb muutuja **hind** väärtus muutuja **hinnatõus** võrra. Kasuta küsimuses etteantud muutujanimesid **hind** ja **hinnatõus**.

Vastus: _____

2. Tingimuslaused

Küsimus 3

Millistel tingimustel on järgnev Pythoni avaldis tõene? Õigeid vastusevariante võib olla rohkem kui 1!

```
if a >= 2 and b < 2 or c != d:
```

- 1) Kui a on 2, b on 3, c on True ja d on True
- 2) Kui a on 2, b on 2, c on True ja d on False
- 3) Kui a on 4, b on 1, c on False ja d on False
- 4) Kui a on 1, b on 4, c on True ja d on True
- 5) Kui a on 5, b on -5, c on True ja d on False

Küsimus 4

Käivitatakse järgnev Pythoni programm:

```
x = 10
y = 10
arv = 0
if y > 0:
    arv += 1
if x > y:
    arv += 3
elif x < y:
    arv -= 3
else:
    arv += 10
print(arv)
```

Mis on muutuja **arv** väärtus programmi töö lõppedes?

Vastus: _____

3. Funktsioonid

Küsimus 5

Vali kõik õiged Pythoni programmid, mille käivitamise tulemusena väljastatakse ekraanile sõne "Ta on president". Õigeid vastuseid võib olla rohkem kui 1!

- 1)

```
def president(nimi):  
    if nimi == "Karis":  
        return "Ta on president"  
    return "Ta ei ole president"  
  
print(president("Karis"))
```
- 2)

```
def president(nimi):  
    if nimi == "Karis":  
        return "Ta on president"  
    else:  
        return "Ta ei ole president"  
  
print(president("Karis"))
```
- 3)

```
def president(nimi):  
    if nimi == "Karis":  
        return "Ta on president"  
    return "Ta ei ole president"  
  
president("Karis")
```
- 4)

```
def president(nimi):  
    if nimi == "Karis":  
        "Ta on president"  
    else:  
        "Ta ei ole president"  
  
print(president("Karis"))
```
- 5)

```
def president(nimi):  
    if nimi == "Karis":  
        print("Ta on president")  
    else: print("Ta ei ole president")  
  
president("Karis")
```

Küsimus 6

Ette on antud päevatasu väljastamise programm, millel on funktsiooni defineerimine jäänud poolikuks:

```
def _____:
    return tunnitasu * tunnid

print(päevatasu(5))
```

Täienda programmi selliselt, et programmi käivitades väljastatakse ekraanile 40.

4. Korduslaused

Küsimus 7

Käivitatakse järgnev programm:

```
i = 0
while i < 32:
    print(i)
    i += 4
print(i)
```

Mitu rida see programm ekraanile väljastab?

Vastus: _____

Küsimus 8

Täienda programmi selliselt, et programmi käivitamisel väljastatakse ekraanile arvud 0-st 4-ni (0, 1, 2, 3, 4).

```
for i in _____(____):
    print(i)
```

5. Sõnetöötlus

Küsimus 9

Käivitatakse järgnev programm:

```
sõna = "Auto"  
sõna += "bussi"  
sõna += "Koondis"  
if sõna == "Auto":  
    print(sõna.upper())  
else:  
    print(sõna.lower())
```

Mis sõne väljastatakse ekraanile? Vali õige vastusevariant.

- 1) Auto
- 2) AUTO
- 3) Auto bussi Koondis
- 4) autobussikoondis
- 5) AUTOBUSSIKOONDIS
- 6) auto bussi koondis

Küsimus 10

Käivitatakse järgnev programm:

```
a = " auto "  
a = a.strip()  
a *= 2  
b = int('2')  
print(len(a) + b)
```

Mis väljastatakse ekraanile?

Vastus: _____

6. Lihtsam failitöötlus

Küsimus 11

Vali kõik õiged Pythoni programmid, mis avavad faili **tekst.txt**, loevad korraga sisse kogu faili sisu, mis salvestatakse muutujasse **sisu**, ning sulgevad faili. Õiged vastused võib olla rohkem kui 1!

- 1)

```
fail = open("tekst.txt")
sisu = fail.readline()
fail.close()
```
- 2)

```
fail = open('tekst.txt')
sisu = fail.read()
fail.close()
```
- 3)

```
fail = open('tekst.txt', 'r')
sisu = fail.readlines()
fail.close()
```
- 4)

```
fail = open("tekst.txt", "read")
sisu = fail.readlines()
fail.close()
```
- 5)

```
fail = open("tekst.txt", "r")
sisu = fail.read()
fail.close()
```

Küsimus 12

Vali kõik õiged Pythoni programmid, mis avavad faili **tekst.txt**, kirjutavad sinna sõne „Teretere vanakere“ ning seejärel sulgevad faili. Õigeid vastuseid võib olla rohkem kui 1!

- 1)

```
fail = open('tekst.txt')
fail.write("Teretere vanakere")
fail.close()
```
- 2)

```
fail = open('tekst.txt', 'r')
fail.write("Teretere vanakere")
fail.close()
```
- 3)

```
fail = open("tekst.txt", "w")
fail.write("Teretere vanakere")
fail.close()
```
- 4)

```
fail = open("tekst.txt", "w+")
fail.write("Teretere vanakere")
fail.close()
```
- 5)

```
fail = open('tekst.txt', "r+")
fail.write("Teretere vanakere")
fail.close()
```

8. 2. kontrolltöö arvestusliku testi näidistest

2. kontrolltöö arvestusliku testi näidistest

1. Järjendid I

Küsimus 1

Käivitatakse järgnev Pythoni programm:

```
tähed = ["a", "b", "c", "d", "e", "f"]  
print(tähed[1:3])
```

Mis väljastatakse ekraanile? Vali õige vastusevariant.

- 1) ['b', 'c']
- 2) ['b', 'c ', 'd']
- 3) ['a', 'b']
- 4) ['a', 'b', 'c']
- 5) ['b']
- 6) ['c']

Küsimus 2

Käivitatakse järgnev Pythoni programm:

```
tähed = ["a", "b", "c", "d", "e", "f"]  
print(tähed[-3])
```

Mis väljastatakse ekraanile?

Vastus: _____

2. Järjendid II

Küsimus 3

Täienda järgnevat programmi selliselt, et ekraanile väljastatakse kõik järjendi **tähed** elemendid.

```
tähed = ["a", "b", "c", "d", "e", "f"]  
for _____ in _____:  
    print(täht)
```

Küsimus 4

Käivitatakse järgnev programm:

```
arvud = [1, 2, 3, 4, 5]  
for i in range(len(arvud)):  
    arvud.pop()  
print(len(arvud))
```

Mis väljastatakse ekraanile?

Vastus: _____

3. Kahekordne tsükkel

Küsimus 5

Käivitatakse järgnev Pythoni programm:

```
a = list(range(5))  
b = list(range(4))  
for i in a:  
    for j in b:  
        print(j)
```

Mitu arvu väljastatakse ekraanile?

Vastus: _____

Küsimus 6

Vali kõik õiged Pythoni programmid, mille käivitamisel väljastatakse maatriksi **maatriks** kõik elemendid. Õigeid vastusevariante võib olla rohkem kui 1!

- 1)

```
maatriks = [[1, 2, 3],
             [4, 5, 6],
             [7, 8, 9]]

for i in range(len(maatriks)):
    for element in i:
        print(element)
```
- 2)

```
maatriks = [[1, 2, 3],
             [4, 5, 6],
             [7, 8, 9]]

for rida in maatriks:
    for element in rida:
        print(element)
```
- 3)

```
maatriks = [[1, 2, 3],
             [4, 5, 6],
             [7, 8, 9]]

for i in range(len(maatriks)):
    for j in range(len(maatriks[i])):
        print(maatriks[i][j])
```
- 4)

```
maatriks = [[1, 2, 3],
             [4, 5, 6],
             [7, 8, 9]]

for rida in maatriks:
    for i in range(len(rida)):
        print(rida[i])
```
- 5)

```
maatriks = [[1, 2, 3],
             [4, 5, 6],
             [7, 8, 9]]

for rida in range(len(maatriks)):
    for element in rida:
        print(rida[element])
```

4. Ennikud

Küsimus 7

Vali kõik õiged väited. Õigeid vastusevariante võib olla rohkem kui 1!

- 1) Ennik ei ole muudetav andmestruktuur
- 2) Ennikusse saab elemente lisada käsu `append()` abil
- 3) Ennikusse saab elemente lisada käsu `add()` abil
- 4) Enniku elemente on võimalik sorteerida käsu `sort()` abil
- 5) Ennikust saab elemente eemaldada käsu `remove()` abil

Küsimus 8

Vali õige Pythoni lause, millega salvestatakse muutuja **ennik** väärtuseks kolme-elementiline ennik, mis koosneb muutujatest **sugu**, **vanus** ja **pikkus**. Ainult üks vastus on õige.

- 1) `ennik = [sugu, vanus, pikkus]`
- 2) `ennik = {sugu, vanus, pikkus}`
- 3) `ennik = (sugu, vanus, pikkus)`
- 4) `ennik = tuple{sugu, vanus, pikkus}`
- 5) `ennik = tuple(sugu, vanus, pikkus)`

5. Hulgad

Küsimus 9

Vali kõik õiged väited. Õigeid vastusevariante võib olla rohkem kui 1!

- 1) Hulk on järjestatud andmestruktuur
- 2) Tühja hulga loomiseks tuleb kasutada loogelisi sulge { }
- 3) Hulgad on muudetavad
- 4) Hulgas ei ole korduvaid elemente
- 5) Hulkadega on võimalik teha selliseid operatsioone, mida järjendite või ennikutega teha ei saa, näiteks on võimalik leida kahe hulga ühisosa või ühend.

Küsimus 10

Kävitatakse järgnev Pythoni programm:

```
a = set([1, 2, 3, 4])  
b = set([3, 2, 1, 4])  
print(a == b)
```

Mis väljastatakse ekraanile?

Vastus: _____

6. Sõnastikud

Küsimus 11

Vali kõik õiged väited. Õigeid vastusevariante võib olla rohkem kui 1!

- 1) Sõnastikud on muudetavad
- 2) Sõnastiku väärtused võivad olla mistahes tüüpi
- 3) Sõnastiku võtmed võivad olla mistahes tüüpi
- 4) Sõnastikus võib olla korduvaid võtmeid
- 5) Sõnastikku on võimalik andmeid lisada käsu `add()` abil

Küsimus 12

Vali kõik õiged Pythoni programmid, mis käivitamisel väljastavad ekraanile kõik sõnastikus **autod** olevad võtmed ja nendele vastavad väärtused (ühel väljastatud real on sõnastiku üks võti ja temale vastav väärtus). Õigeid vastusevariante võib olla rohkem kui 1!

- 1)

```
autod = {"BMW": 8.0, "Audi": 7.2, "Kia": 4.6, "Toyota": 5.4}
for i in autod:
    print(i, autod[i])
```
- 2)

```
autod = {"BMW": 8.0, "Audi": 7.2, "Kia": 4.6, "Toyota": 5.4}
for i in range(len(autod)):
    print(i, autod[i])
```
- 3)

```
autod = {"BMW": 8.0, "Audi": 7.2, "Kia": 4.6, "Toyota": 5.4}
for auto, kütusekulu in autod.items():
    print(auto, kütusekulu)
```
- 4)

```
autod = {"BMW": 8.0, "Audi": 7.2, "Kia": 4.6, "Toyota": 5.4}
for auto in autod:
    print(auto, autod.value(auto))
```
- 5)

```
autod = {"BMW": 8.0, "Audi": 7.2, "Kia": 4.6, "Toyota": 5.4}
for auto in autod:
    print(auto, autod)
```

9. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Raiki Mänd,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Tartu Ülikooli kursuse „Programmeerimine“ arvestuslike testide uuendamine**, mille juhendajateks on Tauno Palts ja Reimo Palm, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Raiki Mänd

09.05.2023