

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Sixtina Ladva

**E-kursuse „Programmeerimise alused“ osalejate
hinnangud ülesannetele**

Bakalaureusetöö (9 EAP)

Juhendajad: Merilin Säde, MA
Eno Tõnisson, MSc

Tartu 2017

E-kursuse „Programmeerimise alused“ osalejate hinnangud ülesannetele

Lühikokkuvõte:

Käesolevas bakalaureusetöös analüüsiti e-kursuse „Programmeerimise alused“ osalejate hinnanguid ülesannetele. Töö tulemusena leiti vastused järgnevatele uurimisküsimustele: millised ülesanded on osalejate hinnangul kerged ja millised keerukad, millised ülesanded on osalejate hinnangul huvitavad, milliseid valikülesandeid lahendati kõige rohkem ning milliseid soovitusi annavad kursusel osalejad korraldajatele.

Võtmesõnad:

MOOC, programmeerimise e-kursus, programmeerimise ülesanded

CERCS: P175, S280

E-course “Introduction to Programming” participants’ assessments for exercises

Abstract:

The purpose of this thesis is to analyse e-course “Introduction to Programming” participants’ assessments for exercises. As a result of this thesis, these four research questions were answered: which exercises find the participants easy and which difficult, which exercises are interesting for the participants, which optional exercises the participants solved the most and what recommendations the participants give to the organizers.

Keywords:

MOOC, programming e-course, programming exercises

CERCS: P175, S280

Sisukord

Sissejuhatus	4
1. Ülevaade MOOCist	6
2. MOOC „Programmeerimise alused“	8
2.1 Kursuse ülevaade	9
2.2 Kursuse ülesehitus	9
3. Ülesannete olulisus MOOCil	11
3.1 Programmeerimisülesanded algajatele	11
3.2 Ülesannete keerukus	12
4. Metoodika.....	13
4.1 Valim	13
4.2 Protseduur	13
4.3 Mõõtevahend	14
5. Tulemused	15
5.1 Esimene nädal	15
5.2 Teine nädal	16
5.3 Kolmas nädal	18
5.4 Neljas nädal	20
5.5 Viies nädal	21
5.6 Kuues nädal	23
5.7 Seitsmes nädal	24
5.8 Kaheksas nädal	26
6. Arutelu.....	28
Kokkuvõte	31
Kasutatud kirjandus.....	32
Lisad	34
I. Ülesanded	34
II. Lihtlitsents.....	36

Sissejuhatus

Tehnoloogia kiire arenguga kaasneb programmeerimisoskuse osatähtsuse kasvamine ühiskonnas. Programmeerimisoskus tuleb kasuks tööturul luues paremaid võimalusi. Seda kinnitab haridus- ja teadusministeeriumi poolt läbiviidud uuring [1], kus selgus, et eksisteerib seos oskuste taseme ja palga vahel nendes töökohtades, kus on vaja spetsiifilisi arvutioskusi nagu seda on näiteks programmeerimisoskus. Programmeerimisoskuse olulisus toodi välja ka Eesti Valitsuse poolt koostatud Eesti Euroopa Liidu poliitika 2015-2019 raamdokumendis [2]: „Infotehnoloogiaalase (IT) hariduse edendamine on oluline, et areneva majanduse ja ühiskonna probleeme lahendada ja ootustele vastata. Selleks on tarvis juba varases eas toetada ja soodustada nii tüdrukute kui ka poiste programmeerimisoskuse arendamist, IT-erialade õppejõudude mobiilsust ja teadmiste jagamist ning rahvusvaheliste õppekavade väljatöötamist.“ [2:6-7]. Seega on oluline, et inimestele oleks tagatud võimalus saada algteadmised programmeerimisest. Üks parimaid viise selleks on vaba juurdepääsuga e-kursus ehk MOOC (ingl *Massive Open Online Course*), kus saab korraga õppida väga palju inimesi ja seda tasuta.

Tartu Ülikoolis loodi esimene programmeerimisega seotud MOOC aastal 2014, milleks oli kursus nimega „Programmeerimisest maalähedaselt“. Kuna kursus osutus väga populaarseks, loodi aastal 2016 järgmine kursus nimega „Programmeerimise alused“, mis on võetud ka käesolevas bakalaureusetöös vaatluse alla. Aastal 2017 loodi jätkukursus „Programmeerimise alused II“.

Kuna programmeerimine on praktiline oskus, siis selleks, et seda oskust omandada, tuleb harjutada, sest ainult teooria õppimisest ei piisa. Programmeerimise ülesanded täidavad olulist rolli programmeerimise õpetamisel, kuna nende lahendamine soodustab õpilaste arengut vajalike programmeerimisoskuste ja tehnikate omandamisel. Seega on oluline, et ülesanded oleks lahendajatele põnevad, motiveerivad ja samas õpetlikud.

Kursusel „Programmeerimise alused“ tuli osalejatel lahendada igal nädalal neli kontrollülesannet, millest viimase ülesande puhul oli õppuril valida mitme valikülesande vahel. Tagasiside saamiseks paluti osalejatel täita iga kahe nädala järel vaheküsitlus, mis sisaldas ka osalejate hinnanguid ülesannetele.

Käesoleva bakalaureusetöö eesmärk on analüüsida 2015/2016 õppeaasta kevadsemestril toimunud programmeerimise e-kursuse „Programmeerimise alused“ osalejate hinnanguid ülesannetele ja anda soovitusi ülesannete koostamisel ning e-kursuse paremaks muutmisel.

Eesmärgist tulenevalt esitab töö autor järgnevad uurimisküsimused:

- Millised ülesanded on kerged ja millised keerukad vaba juurdepääsuga e-kursuse “Programmeerimise alused” osalejate endi hinnangul?
- Milliseid valikülesandeid lahendati vaba juurdepääsuga e-kursusel “Programmeerimise alused” kõige rohkem?
- Millised ülesanded on vaba juurdepääsuga e-kursuse “Programmeerimise alused” osalejate jaoks huvitavad?
- Milliseid soovitusi annavad osalejad vaba juurdepääsuga e-kursuse “Programmeerimise alused” kohta?

Töö esimeses peatükis antakse ülevaade MOOCist ja sellest, milliseid programmeerimisega seotud MOOCe pakub Tartu Ülikooli arvutiteaduse instituut. Teises peatükis tutvustatakse lähemalt e-kursust “Programmeerimise alused”. Kolmandas peatükis antakse kirjanduse põhjal ülevaade ülesannete olulisusest, nende keerukusest ning mida tuleks programmeerimisülesannete koostamisel algajatele silmas pidada. Neljandas peatükis kirjeldatakse uurimisel kasutatud metoodikat. Viiendas peatükis esitatakse tulemused ning kuues peatükk keskendub tulemuste arutelule.

1. Ülevaade MOOCist

MOOCi (ingl *Massive Open Online Course*) mõiste ei ole väga rangelt paigas. Erinevates allikates defineeritakse MOOCi mõnevõrra erinevalt. Tänapäeval on ülikoolide poolt pakutavatel e-kursustel kolm iseloomustavat aspekti [3]:

- kursus on reeglina tasuta, kuid on e-kursusi, kus peab sertifikaadi eest tasuma;
- kursusel on tavaliselt ajaline piirang, mille jooksul kursus toimub;
- kursusel kasutatakse õppeprotsessis erinevaid interaktiivseid e-õppe vahendeid.

MOOC kursuste populaarsuse kiire kasv sai alguse 2011. aastal. Dhawal Shahi poolt kirjutatud artiklist [4] selgub, et MOOCide populaarsus on aastast aastasse üha tõusnud. Aastal 2016 korraldati rohkem kui 700 erineva ülikooli poolt 6850 kursust. Võrreldes aastaga 2015, lisandus rohkem kui 2600 uut kursust. Kursustest kokku võttis osa 58 miljonit õppurit, kellest 23 miljoni le õppurile oli see esmakordne kogemus osaleda MOOCil. Kõige rohkem kursusi viidi läbi ärijuhtimise teemal, mis moodustasid 19,3% kõigist korraldatud kursustest. Arvutiteaduse ja programmeerimisega seotud kursused moodustasid 17,4% kõigist korraldatud kursustest. See tähendab, et aastal 2016 viidi läbi ligikaudu 1192 arvutiteaduse ja programmeerimisega seotud MOOCi.

Tartu Ülikooli arvutiteaduse instituut pakub hetkel nelja programmeerimisega seotud MOOCi: “Programmeerimisest maalähedaselt”, “Programmeerimise alused”, mis on suunatud täiskasvanutele, “Programmeerimise alused”, mis on suunatud õpilastele ja „Programmeerimise alused II“. Tegemist on sissejuhatavate kursustega programmeerimisse, kuhu oodatakse kõiki inimesi, kes on huvitatud programmeerimisest. Esimesed kolm kursust on suunatud inimestele, kellel kokkupuude programmeerimisega on vähene või puudub. “Programmeerimisest maalähedaselt” on väiksema mahuga kursus (1 EAP), kus tutvutakse programmeerimise põhimõistete ja konstruktsioonidega [5]. Kursuses “Programmeerimise alused“ seevastu käsitletakse natuke rohkem programmeerimisega seotud teemasid ja neid ka palju põhjalikumalt. Sellest tulenevalt on ka kursuse maht kolm korda suurem ehk 3 EAP. Sel kevadel (3. aprill – 28. mai 2017) alustas ka uus kursus „Programmeerimise alused II“ [6], mille maht on 3 EAP. Võrreldes kahe eelneva programmeerimise kursusega eeldab see kursus eelnevaid programmeerimisalaseid teadmisi. Kursus sobib väga hästi neile, kes on läbinud e-kursuse „Programmeerimise alused“ ning soovivad oma teadmisi veelgi rohkem täiendada. Kursuse

eesmärk on süvendada algteadmisi programmide koostamisest, testimisest ja silumisest ning täiendada esmast programmeerimisoskust.

2. MOOC „Programmeerimise alused“

Selles peatükis antakse ülevaade e-kursusest „Programmeerimise alused“. Käesolevas töös analüüsitakse osalejate hinnanguid antud kursuse ülesannetele.

E-kursus „Programmeerimise alused“ [7] on MOOC, mille eesmärk on osalejatele tutvustada algoritmilist mõtteviisi ja programmeerimist ning sellega seonduvat neile, kellel ei ole varasemat programmeerimise kogemust või kokkupuude programmeerimisega on vähene. Kursus toimub üks kord aastas – tavaliselt kevadsemestril. Kursuse kestus on 8 nädalat ning maht 3 EAP. Programmeerimise õpetamisel kasutatakse programmeerimiskeelt Python.

Kursuse õpiväljundites [7] on sõnastatud, et e-kursuse läbinud õpilane

- oskab programmeerimise baaskonstruktsioone (tingimuslause, tsükkel, funktsioon) esitada nii plokk skeemidena kui ka programmi loikudena programmeerimiskeeles Python;
- oskab lihtsa ülesande realiseerida programmina programmeerimiskeeles Python, kasutades seejuures erinevaid baaskonstruktsioone ja andmetüüpe (täisarvud, ujukomaarvud, sõned, järjendid), muutujaid, operatsioone, graafikavahendeid ning andmevahetust failidega;
- oskab leida ja sõnastada enda eri- või huvialaga seotud programmeerimisülesandeid ja neid lahendada; omab esmast ülevaadet programmeerimise ajaloost, tarkvara tootmise protsessist, erinevatest programmeerimiskeeltest, arvusüsteemidest, regulaaravaldisest ja rekursioonist;
- on mõelnud läbi, kas ta tahab end edaspidi programmeerimisega rohkem kurssi viia ning on teadlik sellekohastest võimalustest.

Esimene „Programmeerimise alused“ [5] e-kursus viidi läbi 11. jaanuarist kuni 6. märtsini 2016, kus osales 295 õppijat, kellest lõpetasid 145. Järgmine kursus algas juba samal kuul - 28. märtsist 22. maini 2016. Osalejate arv oli võrreldes pilootkursusega kuuekordistunud – osavõtjaid oli 1770, kellest 970 lõpetasid. Seega on antud kursus osalejate arvu poolest üks suurimaid Tartu Ülikooli poolt korraldatud programmeerimise e-kursusi. Aastal 2017 alustati jaanuaris kahe kursusega. Neist esimene kursus oli suunatud gümnaasiumivanuses õpilastele ning teine kursus täiskasvanutele.

2.1 Kursuse ülevaade

Kursus kestab 8 nädalat, kus igal nädalal on erinevad teemad, mida osalejatele tutvustatakse.

Kursuse ajakava [7] on järgmine:

- 1. nädal. Sissejuhatus;
- 2. nädal. Tingimuslause;
- 3. nädal. Tsükkel;
- 4. nädal. Sõned ja graafika;
- 5. nädal. Järjend;
- 6. nädal. Funktsioon;
- 7. nädal. Andmevahetus. Lihtne kasutajaliides;
- 8. nädal. Kordamine.

Nädal algab esmaspäevast ja lõpeb pühapäeva hilisõhtul ning selle aja jooksul peab õppija leidma endale aja materjalide läbitöötamiseks ja ülesannete lahendamiseks.

2.2 Kursuse ülesehitus

Kursusel kasutatakse kahte erinevat õpikeskkonda: Tartu Ülikooli arvutiteaduse instituudi õppematerjalide keskkonda *Courses* ja ülikooli õpikeskkonda *Moodle*.

Materjalid on kättesaadavad Tartu Ülikooli arvutiteaduse instituudi õppematerjalide keskkonnas ehk *Courses*'is [8] ning neid on võimalik näha kõigil. Õppematerjalid on väga mitmekesised. Seal leidub programmeerimise materjali, kus selgitatakse programmeerimise põhitõdesid ning õppijal on võimalik vastata enesekontrolli küsimustele, mis aitavad materjali kinnistada. Neid vastuseid keegi ei kontrolli ega salvesta, vaid nende abil saab õppija lihtsalt ja kiiresti oma teadmisi kontrollida. Lisaks programmeerimise materjalile on kursuslastel võimalik lugeda silmaringi materjale, mis aitavad leida seoseid programmeerimise ja ümbritseva maailma vahel. Viimase osa materjalidest moodustavad kontrollülesanded, mis tuleb õppijal iseseisvalt lahendada ja esitada. Kontrollülesanded kontrollivad osaleja arusaamist materjalist. Igal nädalal tuleb lahendada ja esitada neli kohustuslikku kontrollülesannet. Neljanda ülesande puhul antakse õppuril valida, millist ülesannet lahendada. Tavaliselt saab osaleja valida kahe kuni kolme erineva ülesande vahel.

Ülesannete esitamine toimub *Moodle*'i [9] keskkonnas. Lisaks ülesannete esitamisele ja hindamisele, tuleb *Moodle*'is sooritada igal nädalal lõputest, mis koosneb kümnest küsimusest. Õigesti tuleb vastata vähemalt üheksale küsimusele. Ülesannete lahenduste esitamine ja testide tegemiste arv ei ole piiratud. Kursuse läbimiseks tuleb saada arvestatud kõik iganädalased kohustuslikud kontrollülesanded ja nädalatestid. Kuna osalejate arv kursusel on suur, siis osalejate esitatuid töid hindab ja annab tagasisidet automaatkontroll. See tähendab seda, et õppijate programmid peavad vastama teatud nõuetele, mida automaatkontroll eeldab. Seega on oluline, et ülesanded on õppijale arusaadavalt püstitatud ning kursuslane materjali endale selgeks teinud. Kuna tegemist on e-kursusega, siis on ülimalt oluline pakkuda õppijale tuge ja abi küsimuste korral. Üks võimalus selleks on *Moodle*'is olev foorum, kus on võimalik abi küsida või ise teistele kaasõppijatele anda suunavaid lahendusi, kuid mitte ülesande lahendust. Samuti on võimalik küsimuste korral pöörduda kursuse korraldajate poole saates e-maili aadressile prog@ut.ee. Selleks, et kirjade hulka vähendada on loodud ka igale ülesandele murelahendaja, mis annab osalejale abistavaid vihjed, et leida ise oma probleemile lahendus.

3. Ülesannete olulisus MOOCil

Programmeerimise õpetamine on raske, kuna see on praktiline oskus, mille omandamiseks ei piisa ainult teooria teadmistest. Selleks, et seda omandada, tuleb harjutada. See on ilmne, et programmeerimise ülesanded täidavad olulist rolli programmeerimise õpetamisel, kuna nad nõuavad õpilastelt teadmiste rakendamist uutes olukordades [10].

Programmeerimisülesanded võivad täita peale teooria kinnistamise veel olulisi rolle. Näiteks avaldavad nad ka suurt mõju õppijate motivatsioonile. Arto Vihavainen jt [11] kirjutavad, et motivatsiooni tase korreleerub positiivselt õppeedukusega. Kui anda õpilastele programmeerimisülesanded, mis on liiga rasked, siis võib nõrgemate õpilaste motivatsioon oluliselt langeda. Kuid mõõduka keerukusega ja asjakohased ülesanded aitavad õpilasel tõsta sisemist motivatsiooni. Motivatsiooni mõjutab veel tagasiside. Õpilaste motivatsioonile avaldab positiivset mõju see, kui nad saavad oma lahenduse kohta tagasisidet või vihjeid selle kohta, kuidas oma lahendust parandada. Motivatsiooni seisukohast täidavad ülesannete tüüp ja juhiste olemasolu väga olulist rolli programmeerimisülesannete lahendamisel. On leitud, et motivatsiooni puudumine on üks peamisi põhjuseid, miks õpilased kukuvad välja info-tehnoloogiaga seotud kursustest [12]. Kursused, mis on ebahuvitavad ja sisaldavad ülesandeid, mis ei ole motiveerivad, on kõrge väljalangevusega [13]. Seega on oluline, et ülesanded oleksid motiveerivad.

Peale motivatsiooni avaldab tähelepanuväärset mõju õppimisele ka õpilase mugavustase. Mugavustaset mõjutab õpilase enesehinnang ja isiklik efektiivsus. Vajalikud vahendid õpilase mugavustaseme tõstmiseks on sobilikud ülesanded põhjaliku juhendamise ja tagasisidega [11]. Iseseisvalt probleemile lahenduse leidmine ja oskus seda väljendada programmina aitab tõsta õpilase enesekindlust [14].

3.1 Programmeerimisülesanded algajatele

Thomas Staubitzi jt poolt kirjutatud artikli [14] kohaselt algajatele suunatud programmeerimise kursustel, suurem osa õpilaste õpitulemustest tuleneb just programmeerimisülesandeid lahendamises. Algajate programmeerijate kõige suurem puudujääk on oskus mõelda probleemile lahendust ja seda väljendada programmina. Kõige sagedasem viis, kuidas seda oskust parandada on programmeerimisülesannete sagedane lahendamine.

Programmeerimise ülesanded on oluline osa programmeerimise õpetamisel, kuna nende lahendamine soodustab õpilaste arengut vajalike programmeerimise oskuste ja tehnikate omandamisel. Kuid alati ei ole ülesanded koostatud nii, et nad aitaks õppijal saavutada õpieesmärki

[10]. Seega ülesannete disain mängib olulist rolli. Halvasti kavandatud ülesanded ei tekita õppurites tahtmist neid lahendada. Paljud programmeerimisülesanded aitavad õppida kontseptsioone, kuid väga vähesed neist on entusiastlikud ja loovad [15]. Seega on vaja hoolikalt kavandada ülesandeid.

Essi Lahtinen jt [16] kirjutavad, et algajatele programmeerijatele antud programmeerimisülesanded on tihti eluliste probleemide lihtsustatud versioonid. Kui ülesande koostaja ülesannet kavandab, siis tavaliselt on tal olemas ettekujutus soovitud lahendusest ja see lahendus on igale pädevale programmeerijale ilmne, kui ta loeb ülesande kirjeldust. Eriti hoolikas peab olema siis, kui koostatakse ülesandeid algajatele programmeerijatele. Ainuüksi märksõnade valik ülesande kirjelduses on väga oluline. Antud artiklist [16] selgub veel, et ülesannete lihtsustamine ei pruugi parandada õppetulemusi, vaid põhjustada vastupidise efekti. Liiga lihtsad probleemid võivad julgustada õpilasi looma liiga lihtsaid lahendusi, mis ei ole sobivad algse probleemi lahendusele.

3.2 Ülesannete keerukus

Ülesande keerukus on tunnistatud oluliseks ülesande omaduseks, mis mõjutab ja ennustab lahendajate sooritust ja käitumist, kuid keerukuse mõistel ei ole olemas ühtset definitsiooni või mõõdetühikut [17, 18]. Kuid erinevatest seisukohtadest proovitakse seda defineerida. Näiteks Carlos Loría-Sáenz [19] on kirjutanud, et ülesande keerukust on raske hinnata, kuna mõistagi sõltub see siiski inimesest, kes seda proovib lahendada. Sellest seisukohast on keerukus suhteline ja dünaamilise väärtusega. Onur Özgün ja Yaman Barlas [17] väidavad, et ülesande struktuuri seisukohast on ülesande keerukus otseselt seotud ülesande omadustega ja sõltumatu lahendajast. Seega on võimalik defineerida ülesande keerukust selliste näitajate abil nagu elementide arv, lahenduste arv, ülesannete vaheline kokkulangevus või komponentide, milleks on eesmärk, sisend, protsess, väljund, aeg ja esitus, vaheline ühenduvus [18]. Teised teadlased [17] aga peavad ülesande keerukust suhteliseks mõisteks ja peavad keerukust ülesande ja lahendaja vastastikmõju tulemuseks. Kui ülesande keerukus ületab lahendaja suutlikkuse, siis tajub lahendaja, et ülesanne on keeruline. Hoolimata nendest kahest konkureerivast seisukohast, objektiivne ja subjektiivne keerukus võivad koos eksisteerida. Jaques Leplat defineerib kaks keerukust - arvatava (objektiivse) ja tegeliku (subjektiivse). Keerukus, mis on määratud ülesande koostaja poolt on arvatav ning lahendaja vaatenurgast nähtud keerukus on tegelik. Mitmed uuringud näitavad, et subjektiivne keerukus on põhjuslikult sõltuv objektiivsest keerukusest. Objektiivne ülesande keerukus loob subjektiivse keerukuse ja subjektiivne ülesande keerukus kontrollib objektiivset keerukust.

4. Metoodika

Käesolevas peatükis antakse ülevaade sellest, mille põhjal hakatakse analüüsima osalejate hinnanguid ülesannetele. Tulemuste analüüsimiseks ja uurimisküsimustele vastamiseks kasutatakse arvandmeid.

4.1 Valim

Valimi moodustavad õppurid, kes osalesid Tartu Ülikooli vaba juurdepääsuga e-kursusel „Programmeerimise alused“ ning vastasid vaheküsitlusele. Kursus toimus 2016. aastal 28. märts kuni 22. mai. Osalejaid oli kokku 1770 ning neist lõpetas 970. Iga kahe nädala järel paluti osalejatel täita vaheküsitlus, mis andis korraldajatele tagasisidet. Esimesele vaheküsitlusele vastas 1003 inimest vanuses 14-76 aastat, kellest 505 olid mehed ja 498 naised. Teisele küsitlusele vastas 890 inimest vanuses 14-76 aastat, kellest 453 olid mehed ja 437 naised. Kolmandale küsitlusele 678 inimest vanuses 14-61 aastat, kellest 327 olid mehed ja 351 naised. Neljandale vaheküsitlusele vastas kokku 604 inimest vanuses 14-61 aastat, kellest 285 olid mehed ja 319 naised.

4.2 Protseduur

Igal nädalal tuli osalejatel lahendada neli kontrollülesannet. Neljanda ülesande puhul oli õppuril võimalus valida lahendamiseks tavaliselt kahe kuni kolme erineva ülesande vahel. Ainus erand oli neljas nädal ja kaheksas nädal. Neljandal nädalal tuli lahendada kaks kontrollülesannet, kus viimase ülesande puhul oli õppuril võimalik valida kolme erineva ülesande vahel, mida lahendada. Kaheksandal nädalal oli osalejatel võimalik valida kas teha loovtöö või lahendada arvestusülesanne. Käesolevas töös võetakse vaatluse alla kõik 41 kontrollülesannet (vt Lisa D).

Andmete kogumiseks kasutati korraldajate poolt koostatud tagasisideküsitlust ehk vaheküsitlust, mida osalejad pidid täitma üle nädala. Kokku oli neli vaheküsitlust. Esimene vaheküsitlus viidi läbi kolmanda nädala alguses, kus küsiti tagasisidet esimese ja teise nädala kohta. Järgmine vaheküsitlus oli viienda nädala alguses, kus osalejad pidid avaldama arvamust kolmanda ja neljanda nädala kohta ning sellist tagasiside küsimist jätkati kursuse lõpuni.

Käesolevas töös valis autor välja järgnevad küsimused, mida antud töös hakatakse analüüsima:

- Palun hinnake X.nädala kontrollülesannete keerukust.
- Milline oli kõige huvitavam teema/ülesanne/küsimus?
- Millised teemad/ülesanded/küsimused vajaksid Teie arvates muutmist?

X – tähistab antud nädalat.

Esimese küsimuse puhul tuli osalejatel hinnata vastava nädala kontrollülesannete keerukust. Iga kontrollülesande kohta tuli anda hinnang 5-palli süsteemis, kus hinnang 1 tähistas „liiga lihtne“ ning hinnang 5 „liiga keeruline“. Samuti said õppurid oma vastust ka kommenteerida ning seda võetakse antud töö analüüsis ka arvesse. Antud küsimused valiti välja just sellepärast, et nende küsimuste vastused annavad korraldajatele väga head tagasisidet selle kohta, mida oleks kursusel vaja muuta, et seda teha paremaks.

Lõputöö eesmärgiks on analüüsida e-kursuse „Programmeerimise alused“ osalejate hinnanguid ülesannetele. Selle saavutamiseks läbiti etapid, mida uurimisküsimuste kaupa järgnevalt loetletakse.

Esimesele uurimisküsimusele „Millised ülesanded on kerged ja millised keerukad vaba juurdepääsuga e-kursuse „Programmeerimise alused“ osalejate enda hinnangul?“ vastamisel koostati antud andmete põhjal joonised, mis näitavad milliseid hinnanguid osalejad ülesannetele andsid. Seejärel tõlgendati jooniselt tulemusi.

Uurimisküsimusele „Milliseid valikülesandeid lahendati vaba juurdepääsuga e-kursusel „Programmeerimise alused“ kõige rohkem?“ vastamiseks leiti antud andmete põhjal iga valikülesande puhul mitu protsenti vastanutest lahendas antud ülesannet.

Uurimisküsimustele „Milliseid soovitusi annavad osalejad vaba juurdepääsuga e-kursuse „Programmeerimise alused“ kohta?“ ja „Millised ülesanded on vaba juurdepääsuga e-kursuse „Programmeerimise alused“ osalejate jaoks huvitavad?“ vastamiseks analüüsiti osalejate kommentaare.

4.3 Mõõtevahend

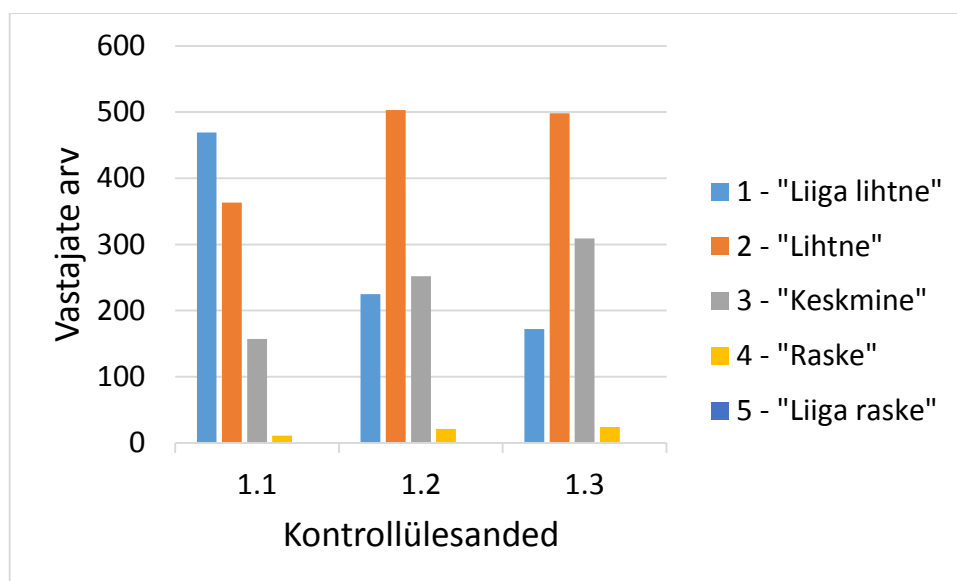
Kokku oli neli vaheküsitlust, mis koosnes väga erinevatest küsimustest. Küsitluses oli 13 küsimust, mis olid nii lahtised kui kinnised küsimused. Näiteks küsiti kursusele kulunud ajakulu, kui võrd ollakse kursusega graafikus ning osalejate hinnanguid teemade, ülesannete ja testide keerukuse kohta 5-palli skaalal. Samuti said osalejad mõne küsimuse juures soovi korral kommenteerida oma vastust.

5. Tulemused

Käesolevas peatükis tuuakse välja osalejate hinnangud iga nädala ülesannete kohta. Ülesannete lingid on leitavad lisas (vt Lisa I).

5.1 Esimene nädal

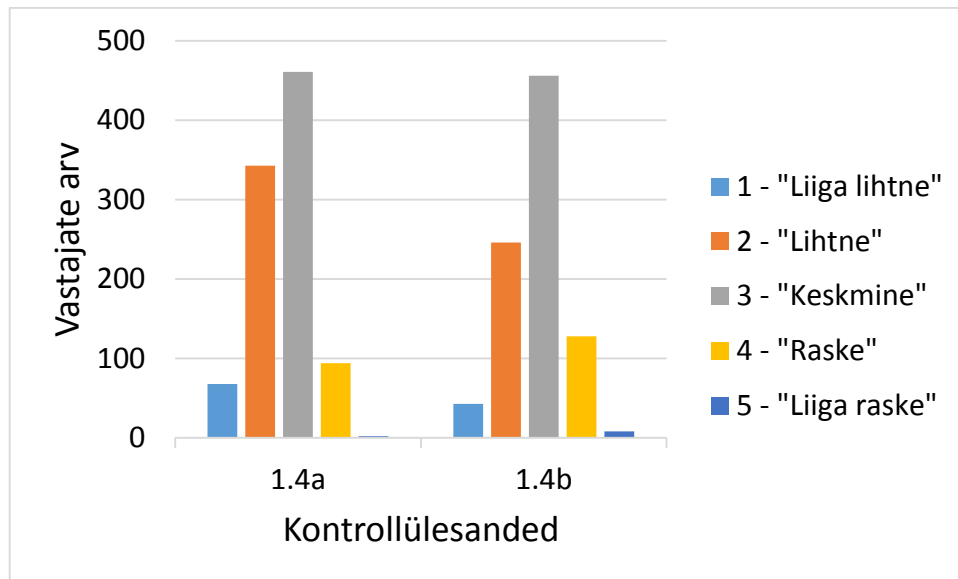
Esimese nädala teemaks olid andmetüübid, muutujad ning sisendi küsimine kasutajalt. Sellel nädalal kirjutasid osalejad oma esimesed programmid. Joonisel 1 on kujutatud osalejate hinnangud esimese nädala kontrollülesannete 1.1 Tervitus, 1.2 Rasvatihane ja 1.3 Astendamine keerukusele.



Joonis 1. Osalejate hinnangud esimese nädala kontrollülesannete keerukusele.

Hinnangu, et ülesanne 1.1 oli liiga lihtne, andsid 47% osalejatest. Täpselt 50% vastajatest hindasid nii ülesande 1.2 kui ka 1.3 pigem lihtsaks. Jooniselt 1 on näha, et esimene ülesanne on selgelt liiga lihtne osalejate jaoks. Samuti on näha, et ülesannete raskus järjest kasvab, kuna hinnangute 3 ja 4 osakaal iga järgneva ülesande juures suureneb.

Esimesel nädalal said osalejad valida kahe valikülesande vahel. Lahendada tuli vähemalt üks. Joonisel 2 on esitatud osalejate hinnangud esimese nädala valikülesannete 1.4a Nädala ajakulu ja 1.4b Kiiruseületamise trahv keerukusele.



Joonis 2. Osalejate hinnangud esimese nädala valikülesannete keerukusele.

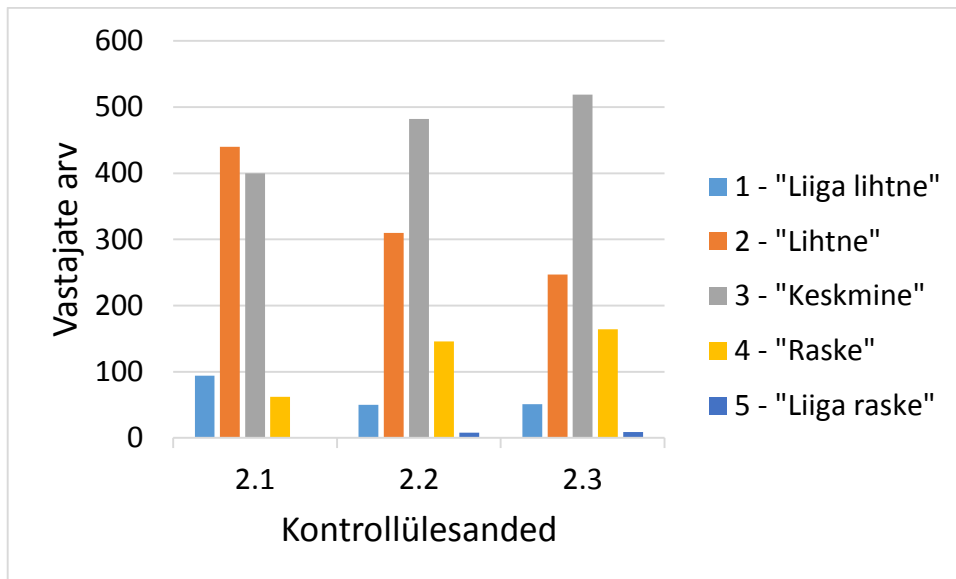
Ülesannet 1.4a lahendas 97% ning 1.4b 88% vastajatest. Nendest ülesande 1.4a puhul 48% ning ülesande 1.4b korral 52% arvas, et ülesanne on keskmise keerukusega. Jooniselt 2 on näha, et valikülesannete keerukus on võrreldes esimeste ülesannetega kasvanud. Kui joonisel 1 domineeris pigem hinnang 2, siis joonisel 2 on näha, et ülekaalus on hinnang 3.

Joonistest 1 ja 2 selgub, et esimese nädala kõige raskem ülesanne oli 1.4b. Vastajatest 1% arvas, et tegu oli liiga raske ülesandega. See toodi välja ka kommentaarides, et üheks raskemaks ülesandeks osutus kontrollülesanne 1.4b. Hoolimata sellest valiti see ülesanne kommentaaride põhjal kõige huvitavamaks.

Kommentaaridest selgus, et osalejatele meeldis, et ülesanded on üsna elulised, mistõttu tekib rohkem seoseid antud teemaga ning valikülesannete puhul saab valida endale sobiva ülesande. Samuti toodi välja, et valikülesandeid või lisaharjutusi võiks olla rohkem ja erinevate raskusastmetega.

5.2 Teine nädal

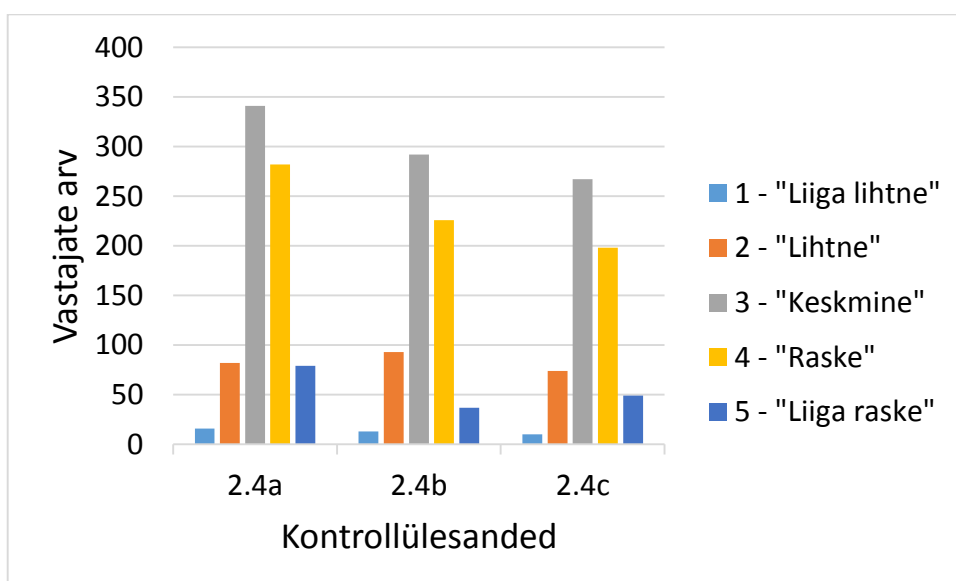
Teise nädala teemaks oli tingimuslause. Joonisel 3 on toodud osalejate hinnangud teise nädala kontrollülesannete 2.1 Jäätumine, 2.2 Allveelaev ja 2.3 Leedu perenimed keerukusele.



Joonis 3. Osalejate hinnangud teise nädala kontrollülesannete keerukusele.

Esimese kontrollülesande puhul pidas 44% vastajatest ülesannet pigem lihtsaks ning 40% arvas, et ülesanne on keskmise keerukusega. Nii ülesande 2.2 kui ka 2.3 puhul arvas vastavalt 48% ning 52% vastajatest et ülesanne on paraja keerukusega. Mõlema ülesande puhul arvas 1% vastajatest, et ülesanne on liiga raske. Jooniselt 3 on selgelt näha, et hinnangute 1 ja 2 osakaal iga järgneva ülesande puhul kahaneb ja hinnangute 3, 4, 5 osakaal kasvab.

Teisel nädalal said osalejad valida kolme erineva valikülesande vahel. Lahendada tuli vähemalt üks ülesanne. Joonisel 4 on kujutatud osalejate hinnangud teise nädala valikülesannete 2.4a Pulss, 2.4b Tribüün ja 2.4c Busside logistika keerukusele.



Joonis 4. Osalejate hinnangud teise nädala valikülesannete keerukusele.

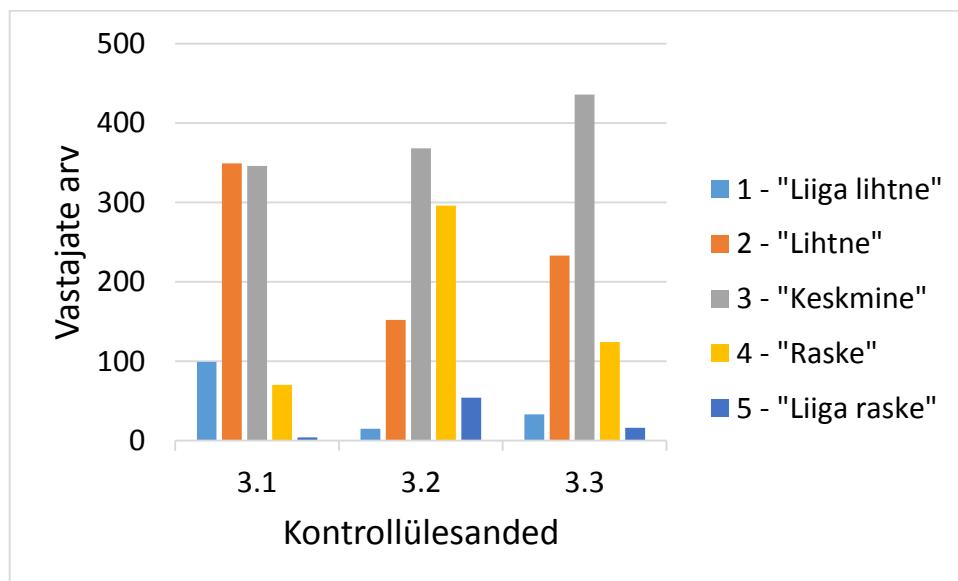
Ülesannet 2.4a lahendas 98% vastajatest ning nendest 43% hindas, et ülesande keerukus on keskmine. Vastajatest 10% arvas, et nende jaoks oli ülesanne liiga raske. Jooniselt 4 on näha, et nii ülesande 2.4b kui ka 2.4c hinnangud on üsna sarnased. Mõlema ülesande puhul suurem osa vastajatest hindas ülesandeid keerukusega 3 ja 4. Hoolimata sellest, et ülesannet 2.4a hindasid osalejad kõige raskemaks, lahendas seda ülesannet 98% vastajatest. Ülesannet 2.4b ja 2.4c lahendasid vastavalt 77% ja 60% vastajatest. Kui võrrelda jooniseid 3 ja 4, siis on näha, et kui joonisel 3 domineerisid hinnangud 2 ja 3, siis joonisel 4 on ülekaalus hinnangud 3 ja 4.

Kommentaaridest selgus, et osalejate jaoks on just viimased ülesanded keerulisemad ning seda on ka jooniselt 4 näha. Põhjuseks toodi see, et kohati jääb puudu oskusest mõelda, kuidas lahendus kirja panna ning kui mingi ülesanne nõuab matemaatilisi teadmisi, siis tekivad väiksed raskused. Samuti tõi mitu inimest välja, et tema jaoks sõltub ülesande keerukus väsimuse astmest – peale rasket tööpäeva tundus tema jaoks iga ülesanne raske. Ülesande 2.4a puhul toodi välja, et segadust tekitas juhistest aru saamine.

Kuid üldiselt toodi välja, et ülesanded on olnud paraja keerukusega, kuna nuputamist on parajalt olnud. Kommentaaride põhjal teise nädala kõige huvitavamateks ülesanneteks olid 2.3, 2.4a ja 2.4c.

5.3 Kolmas nädal

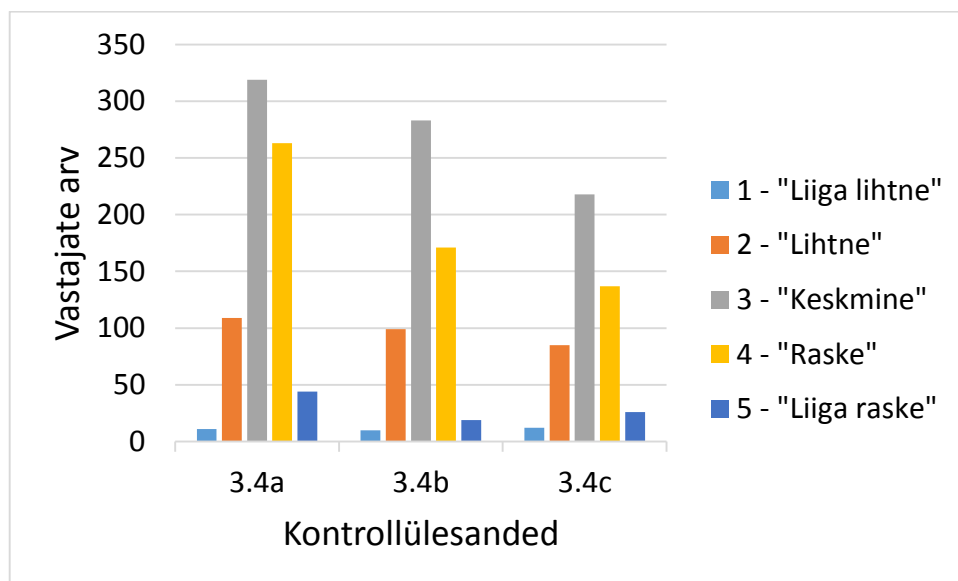
Kolmanda nädala teemaks oli tsükkel. Joonisel 5 on esitatud osalejate hinnangud kolmanda nädala kontrollülesannete 3.1 Vaateratas, 3.2 Lillede arv ja 3.3 Täringumäng keerukusele.



Joonis 5. Osalejate hinnangud kolmanda nädala kontrollülesannete keerukusele.

Ülesannet 3.1 pidasid lihtsaks 40% vastajatest. Sama suur hulk vastajatest arvas ka, et ülesanne on keskmise keerukusega. Jooniselt 5 on näha, et ülesanne 3.1 on võrreldes ülesannetega 3.2 ja 3.3 selgelt pigem kerge raskusastmega ning ülesanne 3.2 nendest kolmest ülesandest kõige keerukam. Kommentaaridest selgus, et selle ülesande tegi keerukaks matemaatiline osa. Ülesandele 3.3 andsid 52% vastajatest hinnangu 3, mis tähendab, et see ülesanne oli pigem paraja keerukusega.

Kolmandal nädalal said osalejad valida kolme erineva ülesande vahel ning lahendada tuli neist vähemalt üks. Joonisel 6 on näha osalejate hinnanguid kolmanda nädala valikülesannete 3.4a Lilled arv, 3.4b Vabavisked ja 3.4c Male keerukusele.



Joonis 6. Osalejate hinnangud kolmanda nädala valikülesannete keerukusele.

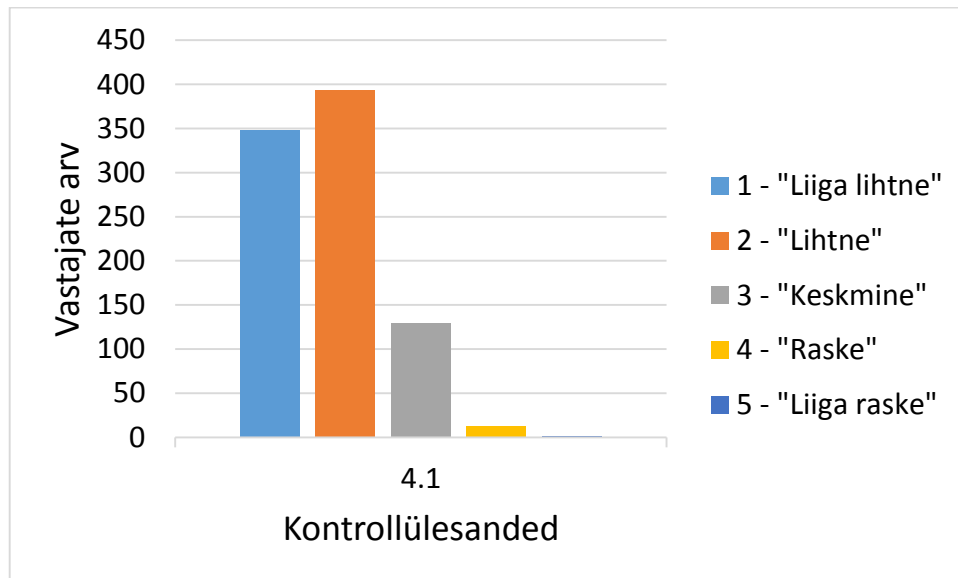
Ülesannet 3.4a lahendati valikülesannetest kõige rohkem ehk 84% vastajatest. Jooniselt 6 selgub, et nendest kolmest ülesandest, oli see osalejate jaoks ka kõige keerulisem ülesanne. Selle ülesande puhul on 6% vastajatest arvanud, et ülesande keerukus on liiga raske. Suurem osa ehk 43% vastajatest arvas siiski, et tegu on paraja keerukusega. Ülesannet 3.4b ja 3.4c lahendas vastavalt 65% ja 54% vastajatest.

Kommentaaridest selgus, et selle nädala ülesanded tegi keeruliseks matemaatiline ja loogiline mõtlemine. Mitmed osalejatest tõid välja, et kuna pole ammu matemaatika ülesannetega kokku puutunud, siis võttis aega just erinevate valemite meelde tuletamine.

Kommentaaridest selgus, et kõige huvitavamad ülesanded olid 3.2, 3.4a ja 3.4c.

5.4 Neljas nädal

Neljanda nädala teemaks olid sõned ja graafika. Sellel nädalal pidid osalejad lahendama ainult kaks kontrollülesannet. Joonisel 7 on toodud osalejate hinnangud neljanda nädala kontrollülesande 4.1 Suured tähed keerukusele.

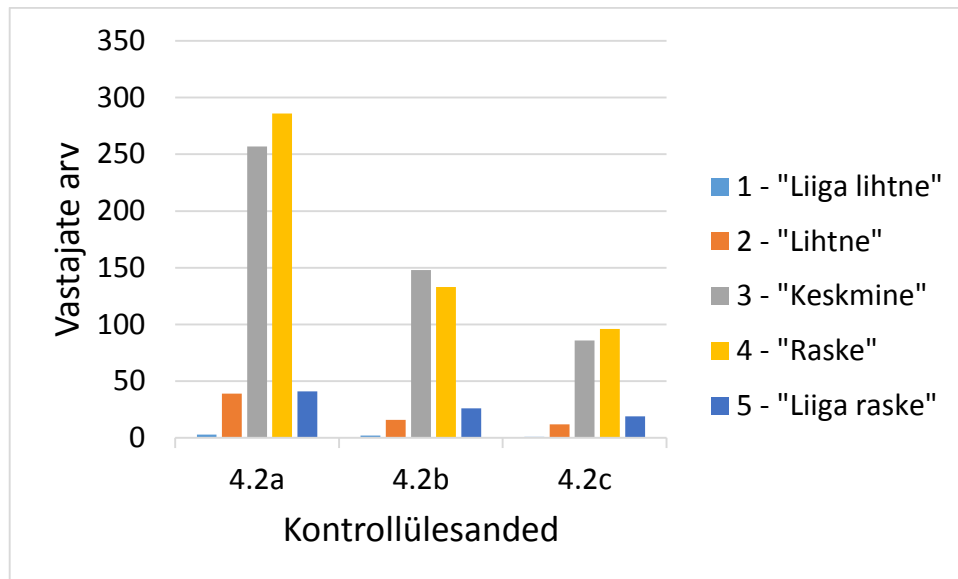


Joonis 7. Osalejate hinnangud neljanda nädala kontrollülesande keerukusele.

Ülesandele 4.1 andsid hinnangu, et tegu on liiga lihtsa ülesandega 39% vastajatest ning 45% vastajatest arvasid, et tegu on pigem lihtsa ülesandega. Ühe inimese jaoks oli tegu ka liiga raske ülesandega.

Kommentaaridest selgus, et vahelduseks oli hea lahendada ka üks kergem ülesanne, mis tõi motivatsiooni kursuse jätkamiseks tagasi.

Neljandal nädalal said osalejad valida kolme erineva valikülesande vahel ning lahendada tuli neist täpselt üks. Joonisel 8 on näha osalejate hinnanguid neljanda nädala valikülesannete 4.2a Eesti haldusüksuse lipp, 4.2b Liiklusmärk ja 4.2c Seadme paneel keerukusele.



Joonis 8. Osalejate hinnangud neljanda nädala valikülesande keerukusele.

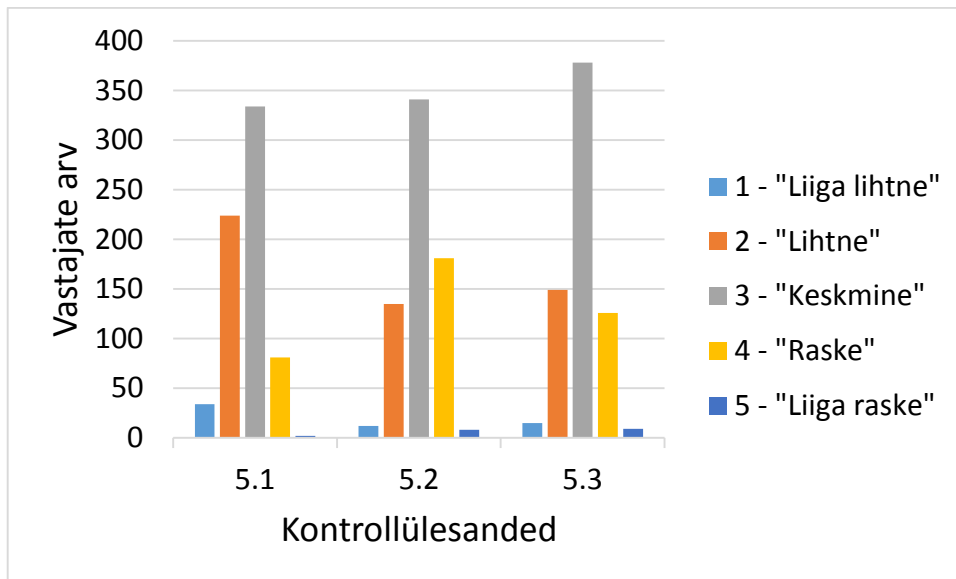
Valikülesande 4.1a lahendasid 70% vastajatest. Nendest 46% hindasid ülesande keerukuse pigem raskeks ning 41% arvasid, et ülesanne on keskmise raskusega. Ülesande 4.2b lahendasid 37% vastajatest. Nendest 45% hindasid ülesande keerukuse keskmiseks ning 41% neist arvasid, et tegu on pigem keerulise ülesandega. Ülesannet 4.2c lahendati kõige vähem ehk 24% vastajatest, kuid ka siin domineerisid kõige rohkem eelnevalt nimetatud hinnangud ehk 45% lahendajatest pidasid ülesannet pigem keeruliseks ning 40% keskmiseks.

Kommentaaridest selgus, et graafika osas tekitas raskusi koordinaatide paigutusest arusaamine. Väga paljud tõid välja, et *tkinteriga* seotud materjal võiks olla põhjalikum – rohkem näiteid kuidas joonistada kaari ja keerulisemaid funktsioone. Kui muidu olid kõik näited arusaadavalt seletatud ja samm-sammult keerukamaks tehtud, siis toodi välja, et lipu joonistamise näidet oli keeruline mõista, kuna tuli korraga aru saada teistsugusest programmi ülesehitusest ja numbritest, mis korraldajad olid välja mõelnud.

Kommentaaride põhjal selgus, et osalejate jaoks olid kõige huvitavamad ülesanded 4.2a ja 4.2c. Viimase ülesande puhul toodi välja, et põnevaks tegi selle ülesande see, et sai teha oma loomingut ning esitada endale väljakutse.

5.5 Viies nädal

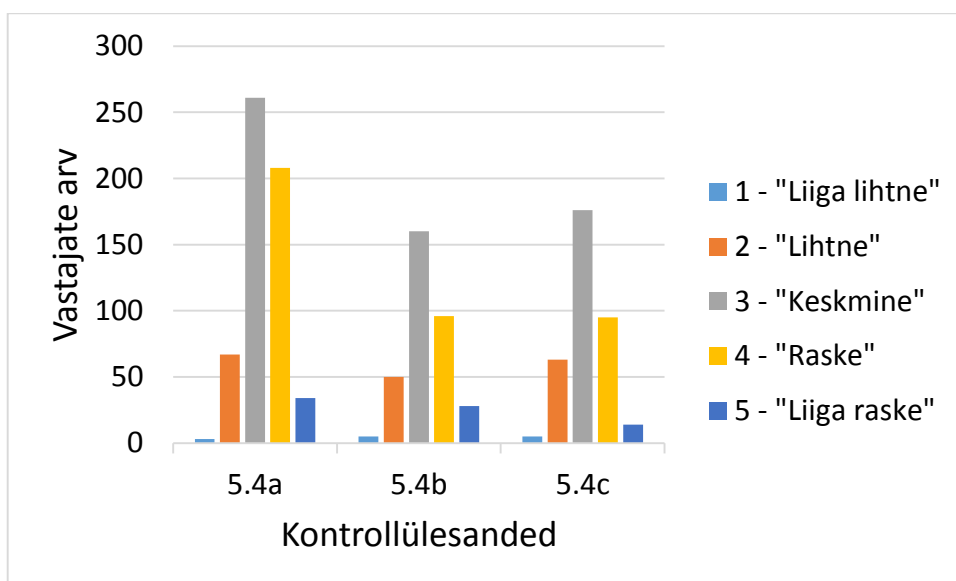
Viienda nädala teemaks olid järjendid ja for-tsüklid. Joonisel 9 on toodud osalejate hinnangud viienda nädala kontrollülesannete 5.1 Mopeedid, 5.2 Lillede arv ja 5.3 Sissetulekud keerukusele.



Joonis 9. Osalejate hinnangud viienda nädala kontrollülesannete keerukusele.

Täpselt 50% vastajatest arvas, et ülesanne 5.1 on keskmise keerukusega. Suur osa lahendajatest ehk 35% arvas, et ülesanne on pigem lihtne. Ülesande 5.2 puhul arvasid pooled ehk 50% vastajatest, et ülesanne on paraja raskusega. Ka ülesanne 5.3 oli 53% jaoks osalejatest paraja raskusega.

Viiendal nädalal said osalejad valida kolme erineva valikülesande vahel ning lahendada tuli neist vähemalt üks. Joonisel 10 on näha osalejate hinnanguid viienda nädala valikülesannete 5.4a Reisidiilid, 5.4b Kümnevõistlus ja 5.4c Tahvli juurde keerukusele.



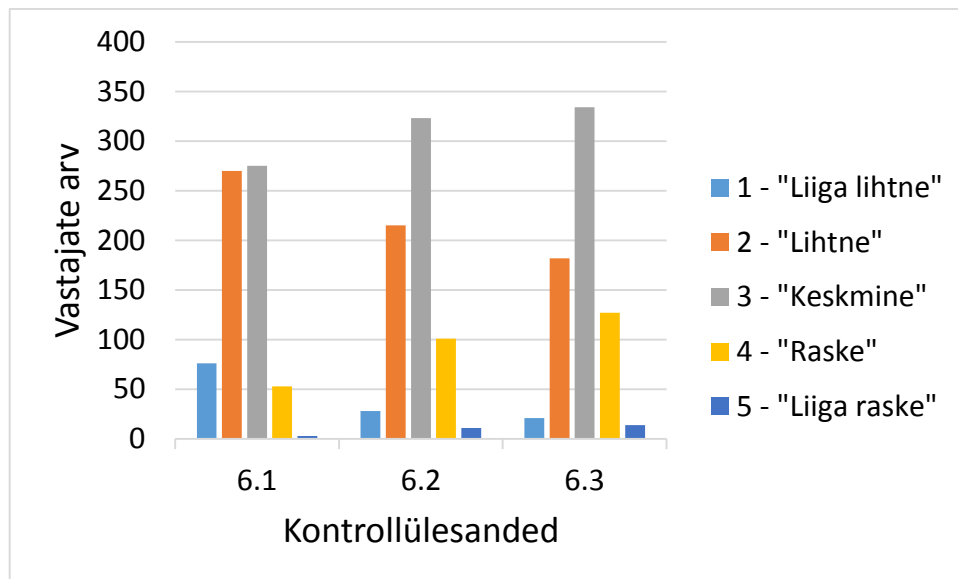
Joonis 10. Osalejate hinnangud viienda nädala valikülesannete keerukusele.

Ülesande 5.4a lahendas 85% vastajatest ning nendest 46% arvas, et keerukus on paras ja 6%, et ülesanne on liiga raske. Ülesannet 5.4b lahendasid täpselt pooled vastajatest ning nendest 46% jaoks oli ülesande keerukus paras ning 8% jaoks liiga keeruline. Ülesande 5.4c lahendasid 52% vastajatest ning lahendajatest täpselt pooled arvasid, et ülesande keerukus on keskmine.

Kommentaaridest selgus, et selle nädala kõige huvitavamateks ülesanneteks olid 5.4a ja 5.4c.

5.6 Kuues nädal

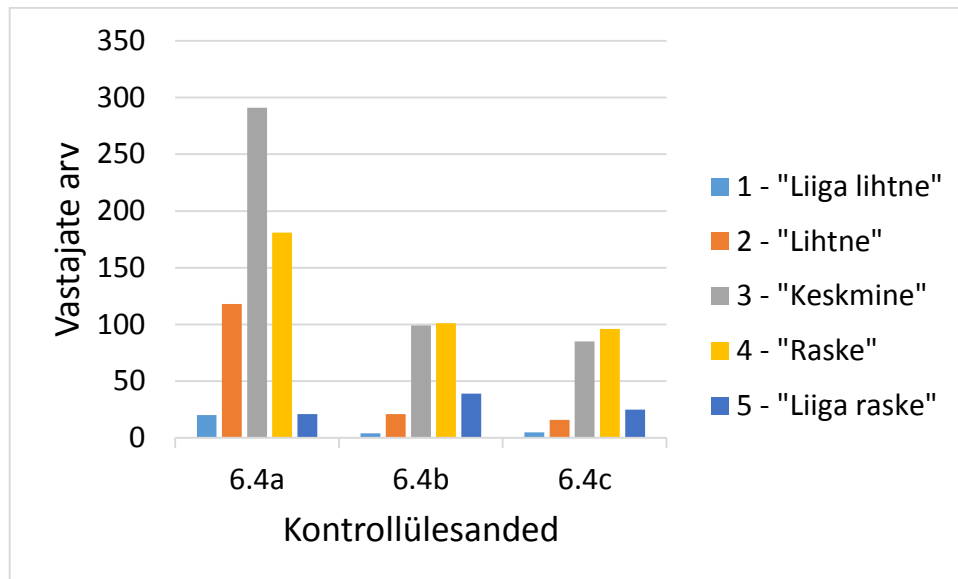
Kuuenda nädala teemaks oli funktsioon. Joonisel 11 on näha osalejate hinnangud kuuenda nädala kontrollülesannete 6.1 Tervitused, 6.2 Teleri suurus ja 6.3 Peo eelarve keerukusele.



Joonis 11. Osalejate hinnangud kuuenda nädala kontrollülesannete keerukusele.

Ülesandele 6.1 andis hinnangu, et tegemist on lihtsa ülesandega 40% lahendajatest ning 40% arvasid, et keerukus on keskmine. Ülesanded 6.2 ja 6.3 olid paraja keerukusega vastavalt 48% ja 49% jaoks lahendajatest.

Kuuendal nädalal said osalejad valida kolme valikülesande vahel ning nendest tuli lahendada vähemalt üks. Joonisel 12 on näha osalejate hinnangud valikülesannete 6.4a Tervitused mõtisklustega, 6.4b Mündid ja 6.4c Kuupäev keerukusele.



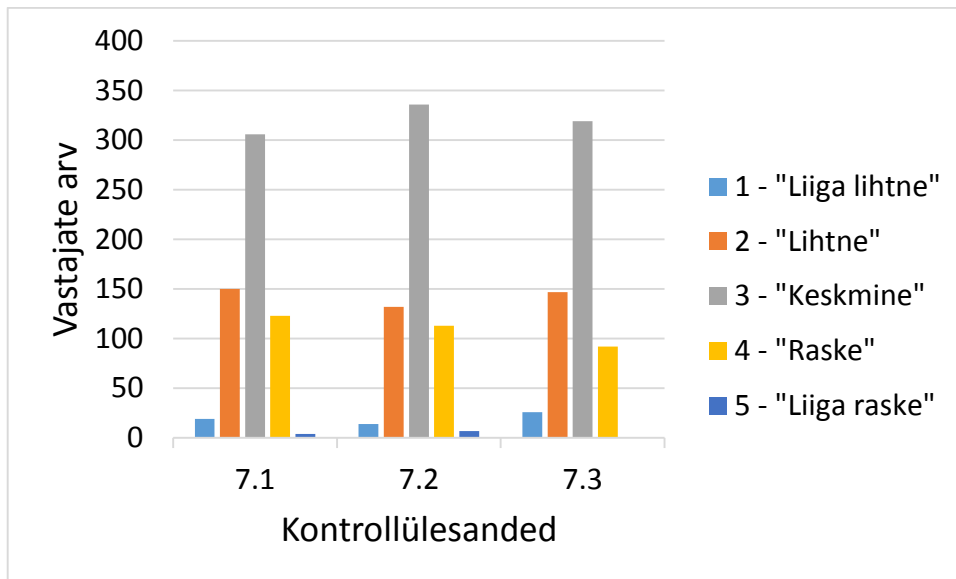
Joonis 12. Osalejate hinnangud kuuenda nädala valikülesannete keerukusele.

Ülesannet 6.4a lahendas 93% vastajatest. Lahendajatest 46% jaoks oli ülesande keerukus keskmine. Ülesannet 6.4b lahendas 39% vastajatest ning nendest 15% hindas, et ülesanne on liiga raske. Ülesannet 6.4c lahendas 33% vastajatest.

Joonisel 12 on näha, et nii ülesannete 6.4b kui ka 6.4c puhul domineerivad hinnangud 3 ja 4. Ka kommentaarides mainiti, et kõige raskemad ülesanded olid kuuendal nädalal 6.4b ja 6.4c. Paljud osalejad kirjutasid kommentaarides, et automaatkontrolli nõudmised võiksid olla eelnevalt nimetatud ülesannete puhul paremini kirjeldatud. Kui Thonnys töötab programm nii nagu vaja, siis automaatkontrolli tagasisidest ei saadud aru, mis täpselt valesti on. Kuigi ülesanded 6.4b ning 6.4c olid selle nädala ühed raskemad ülesanded osalejate jaoks, selgus kommentaaride põhjal, et ühtlasi oli need ka kõige huvitavamad, kuna pakkusid piisavalt nuputamist. Samuti meeldis osalejatele ülesanne 6.3.

5.7 Seitsmes nädal

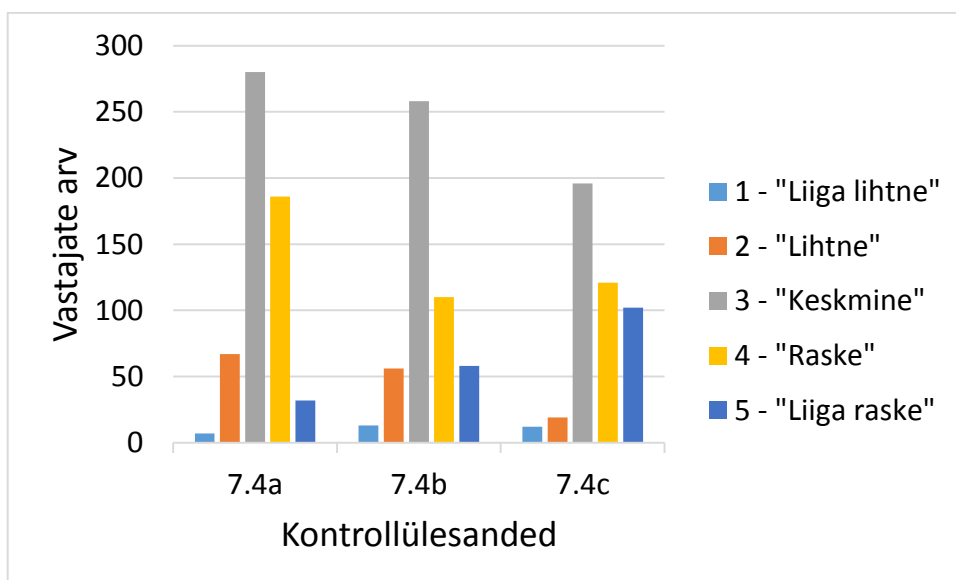
Seitsmenda nädala teemaks oli andmevahetus ja lihtne kasutajaliides. Joonisel 13 on näha osalejate hinnangud seitsmenda nädala kontrollülesannete 7.1 Telegramm, 7.2 Päevik ja 7.3 Kalkulaator keerukusele.



Joonis 13. Osalejate hinnangud seitsmenda nädala kontrollülesannete keerukusele.

Ülesannete 7.1, 7.2 ja 7.3 hinnangute osakaalud on väga sarnased. Kõikide ülesannete puhul arvas üle poolte lahendajatest, et ülesanded on keskmise keerukusega.

Seitsmendal nädalal said osalejad valida kolme valikülesande vahel, millest lahendada tuli vähemalt üks. Joonisel 14 on näha osalejate hinnangud seitsmenda nädala valikülesannete 7.4a Täiendatud peo eelarve, 7.4b Nimepäev ja 7.4c Elutee number keerukusele.



Joonis 14. Osalejate hinnangud seitsmenda nädala valikülesannete keerukusele.

Ülesande 7.4a lahendasid 95% vastajatest ning 6% nendest arvas, et tegu on liiga raske ülesandega. Joonisel 14 on näha, et selle ülesande puhul domineerisid hinnangud 3 ja 4. Ülesannet 7.4b lahendas 82% vastajatest ning nendest 52% hindas ülesande keerukust keskmiseks.

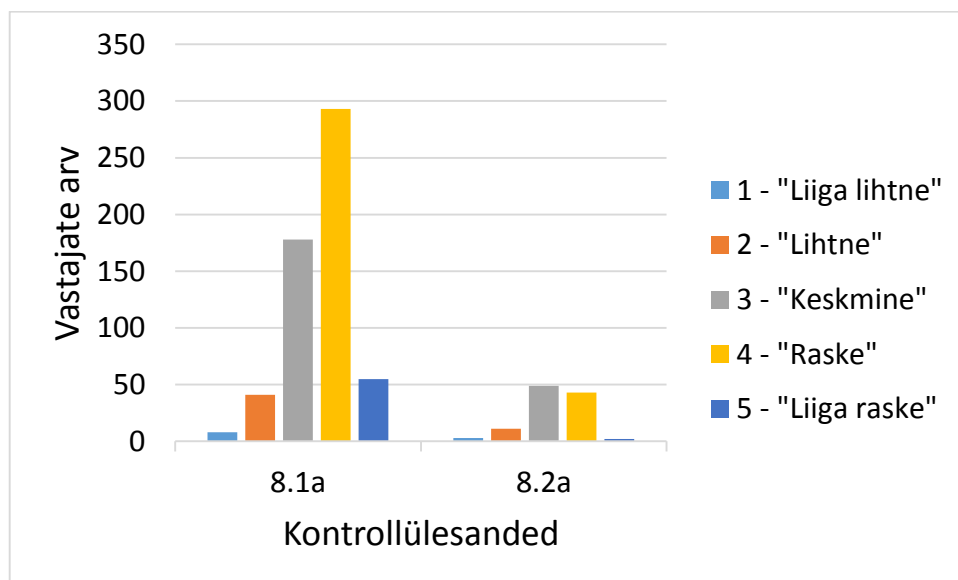
Lahendajatest 12% jaoks oli ülesanne liiga raske. Ülesande 7.4c lahendas 75% osalejatest ning nendest 45% arvas, et ülesande keerukus on keskmine ning 23%, et ülesanne on liiga raske.

Kommentaaridest selgus, et seitsmenda nädala valikülesanded olid ühed raskemad terve kursuse jooksul. Põhjuseks toodi see, et selle nädala teemade kohta oli materjali vähe ning pidi internetist juurde otsima. Nii failist kui veebist lugemise teema juures sooviti rohkem ülesandeid kasulike näidetega. Samuti mainiti, et kuna failist lugemisel on mitmeid variante, siis oleks hea kui kirjeldatakse erinevate variantide plusse, miinuseid ja kasutusvõimalusi. Teine põhjus, mis osalejatele probleeme tekitas oli tekstifailide kodeeringud.

Kõige huvitavamad ülesanded osalejate hinnangul olid 7.2, 7.3 ning 7.4a.

5.8 Kaheksas nädal

Kaheksanda nädala teemaks oli kordamine. Sellel nädalal oli osalejatel võimalik valida kahe ülesande vahel – arvestusülesanne või loovtöö. Arvestusülesanne on koostatud korraldajate poolt, mis on ka ühtlasi näidiseks, mida peaksid sooritama need Tartu Ülikooli üliõpilased, kes soovivad seda ainet oma õpingute osana arvestada. Loovtöö puhul saab aga osaleja mõelda välja ise ülesande, kuid see peab vastama korraldajate poolt loodud nõuetele. Mõlema ülesande puhul peavad osalejad esitama ka vastava ülesande lahendamise protsessi kirjelduse. Joonisel 15 on näha osalejate hinnangud kaheksanda nädala valikülesannete 8.1a Arvestusülesanne ja 8.2a Loovtöö keerukusele.



Joonis 15. Osalejate hinnangud kaheksanda nädala valikülesannete keerukusele.

Ülesannet 8.1a ehk arvestusülesannet lahendas 96% vastajatest ning üle poolte nendest arvas, et ülesande keerukus on pigem raske. Vastajatest 10% jaoks oli ülesanne liiga raske. Ülesannet 8.2a ehk loovtööd lahendas vaid 18% vastajatest ning nendest 45% arvas, et ülesanne on keskmise raskusega ning 40% jaoks pigem raske.

Kommentaaridest selgus, et arvestusülesandeid oleks võinud olla kaks analoogset, millest üks oleks kohustuslik. See annaks neile, kes soovivad veel ülesandeid lahendada ja teemasid korrata veel rohkem enesekindlust. Kuus osalejat tahtsid, et arvestusülesanne oleks mahukam ja raskem. Samuti selgus, et kui muidu ollakse murelahendajast abi saadud, siis viimasel nädalal 8.1a ülesande korral sellest abi ei olnud.

Kommentaaridest selgus, et mõlemad valikülesanded olid huvitavad. Arvestusülesanne meeldis, kuna seal sai kõike õpitut rakendada. Loovtöö puhul meeldis see, et sai panna ennast proovile, mis on kursuselt meelde jäänud ning teha omaloomingut.

6. Arutelu

Antud töö eesmärgiks oli analüüsida e-kursuse „Programmeerimise alused“ osalejate hinnanguid ülesannetele. Töös esitati neli uurimisküsimust ning selles peatükis arutletakse saadud tulemuste üle.

Esimeseks uurimisküsimuseks oli, et millised ülesanded on kerged ja millised keerukad vaba juurdepääsuga e-kursuse “Programmeerimise alused” osalejate endi hinnangul. Jooniste tõlgendamisel ja kommentaaride analüüsimisel selgus, et iga nädala esimene ülesanne oli võrreldes teiste ülesannetega kõige lihtsam. Ülesanded, mis olid selgelt lihtsad osalejate jaoks, olid esimese nädala kolm esimest kontrollülesannet. Tuginedes eelnevalt kirjutatud teooriale, on tähtis, et kursuse alguses oleksid ülesanded pigem lihtsad, et osalejad saaksid enesekindlust ülesannete lahendamisel. Lisaks selgus kommentaaridest veel, et neile, kes on varem programmeerimisega kokku puutunud või osalenud kursusel „Programmeerimisest maalähedaselt“, olid ülesanded väga lihtsad, kuid nende jaoks kes olid algajad, oli raskusaste sobiv. Toodi välja, et pigem on see hea, et ülesanded on alguses lihtsad, kuna siis saab ülesannete lahendamisel enesekindlust. Neljandal nädalal tuli lahendada vaid kaks ülesannet, millest esimene oli osalejate jaoks pigem liiga lihtne. Kui eelnevate nädalate jooksul on ülekaalus olnud pigem hinnangud 3 ja 4, siis sel nädalal 1 ja 2. Kuigi osalejate hinnangul oli vahelduseks hea lahendada ka üks kergem ülesanne, mis tõi tagasi motivatsiooni kursuse jätkamiseks.

Tulemustest selgus, et enamasti osutusid keerukateks ülesanneteks just valikülesanded, millest enamasti esimene valikülesanne oli kõige keerulisem. Seitsmenda nädala ülesanded olid osalejate arvates kursuse ühed raskemad. Näiteks oli kindlasti 7.4c Elutee number üks keerulisemaid ülesandeid sellel kursusel lahendajate jaoks.

Kommentaaridest selgusid ka põhilised põhjused, mis teeb ülesande lahendamise keeruliseks. Esiteks esimestel nädalatel tekitas mitmetel segadust, kuidas oma lahendus programmina kirja panna. Teiseks mõeldi tihti ülesanne keerulisemaks, kui see tegelikult oli. Kolmas väga populaarne põhjus, mis inimestele tekitas raskusi, oli matemaatiline ja loogiline mõtlemine. Neljas põhjus oli kas materjalide vähesus (näiteks seitsmenda nädala puhul) või ülesande juhised olid osalejate jaoks segased (ülesande 2.4a Pulss puhul).

Teine uurimisküsimus oli, et milliseid valikülesandeid lahendati vaba juurdepääsuga e-kursusel “Programmeerimise alused” kõige rohkem. Valikülesannete puhul selgus, et kõige rohkem lahendatakse esimest ülesannet. See kehtis iga nädala puhul, et valiti esimene valikülesanne, kuigi see ülesanne ei pruugi olla kõige lihtsam võrreldes teiste valikülesannetega. Kõige rohkem lahendati seitsmenda nädala valikülesandeid. Sellel nädalal lahendas ülesandeid 7.4a Täiendatud peo eelarve 95%, 7.4b Nimepäev 82% ja 7.4c Elutee number 75% vastajatest, mis on võrreldes eelnevate nädalatega väga kõrged protsendid. Kommentaaridest selgus, et seitsmenda nädala ülesanded olid ühed raskemad terve kursuse jooksul, kuid sellest hoolimata lahendati neid kõige rohkem.

Kolmas uurimisküsimus oli, et millised ülesanded on vaba juurdepääsuga e-kursuse “Programmeerimise alused” osalejate jaoks huvitavad. Selgus, et kõige huvitavamad olid ülesanded, mis olid pigem raskemad. Enamasti olid nendeks valikülesanded. Kuid leidis ka ülesandeid, mis olid osalejatele huvitavad just sellepärast, et nad olid elulised ja praktilised. Näiteks seitsmenda nädala ülesanded 7.1 Telegramm ja 7.2 Päevik meeldisid lahendajatele väga. Samuti oli lahendajate jaoks huvitavad ülesanded, kus osalejad said teha omaloomingut (4.2c Seadme paneel ja 8.1b loovtöö ülesanne).

Neljas uurimisküsimus oli, et milliseid soovitusi annavad osalejad vaba juurdepääsuga e-kursuse “Programmeerimise alused” kohta. Valikusse osutasid soovitusel, mida esines mitmeid kordi ning mis tundusid töö autori arvates kasulikud kursuse parandamiseks.

Kuna kursusel osalejate teadmised on väga erinevad, siis oli neid, kes vajasisid mõne ülesande juures rohkem näiteid ja ka neid, kes neid üldse ei vajanud. Seega leidis väga palju neid, kelle arvates võiks näiteid ja vihjeid rohkem olla. Teisalt jälle leidis ka neid, kelle arvates oli neid liiga palju ja iseseisvat nuputamist võiks rohkem olla. Seega võiks olla näiteks võimalus, kus vihjeid saavad näha vaid need, kes seda soovivad. Kui praegu on iga ülesande juures vihjed kohe nähtaval, siis üks võimalus oleks näiteks teha näiteks nii, et vihjete jaoks oleks eraldi leht ning alles sinna peale klikkides, avaneksid vihjed.

Väga palju leidis neid, kes soovisid näha ülesannete korrektset lahendust. Samuti toodi välja, et põnev oleks näha teiste osalejate lahendusi. Ülesannete lahenduste avalikustamine just sellel kursusel on mõnevõrra riskantne, kuna osalejad saavad oma lahendusi esitada ja parandada kursuse lõpuni. Seega üks võimalus oleks mõne ülesande lahenduse avalikustamine kursuse lõpus, kuid sellisel juhul tuleks avaldatud ülesanne järgmisel kursusel asendada uue sarnase ülesandega.

Kuna kursusel osaleb väga palju neid, kellel puudub varasem kokkupuude programmeerimisega, siis tuleb neile esimestel nädalatel väga palju uut informatsiooni. Esimese vahekuisitluses tõi üks osaleja välja, et võiks olla mingi sõnaraamatulaadne tabel, kus oleksid välja toodud erinevad meetodid (`lower()`, `split()` jne) ning näited nende kohta. Töö autori arvates on see väga hea mõte, kuna igal nädalal lisandub uusi meetodeid ning selline tabel aitaks osalejatel neid paremini meelde jätta ning hoiaks aega ülesannete lahendamisel kokku.

Kuna osalejate tase kursusel on väga erinev siis leidis väga palju neid, kes soovisid, et harjutusülesandeid või valikülesandeid oleks rohkem, mida saaks ise vabatahtlikult lahendada. Samuti toodi välja, et valikülesannete puhul võiks üks ülesanne olla teistest veidi keerulisem, mis nõuaks rohkem iseseisvat nuputamist ning õpetaks osalejat lisainfot internetist otsima.

Hetkel küsitakse osalejate käest tagasisidet iga kahe nädala järel. Paljude osalejate arvates võiks seda teha igal nädalal, kuna eriti ülesannete keerukuse hindamisel on juba ununenud selle raskus või ülesandele kulunud aeg.

Ülesannete koostamisel soovitab töö autor kombineerida ülesandeid ka eelmiste nädalate teemadega, et õpitut rohkem kinnistada. Näiteks seitsmenda nädala ülesanded hõlmasid varem õpitut teemasid ning seetõttu tekkis ka paljudel osalejatel raskusi, kuna olid juba eelnevad teemad unustanud.

Mida nädal edasi, seda rohkem tekib osalejatel raskusi ning mõtteid kursus pooleli jätta. Selleks, et seda vähendada ning õppurite motivatsiooni tõsta, tuleks igal nädalal panna ka mõni kergem ülesanne, mis tõstaks kursusel osalejate motivatsiooni jätkamiseks.

Kursuse kohta on tulnud väga palju positiivset tagasisidet, näiteks üks osaleja kirjutas järgnevalt: „Tore on, et on olemas hakkajad, kes tasuta mõistatusi genereerivad, ja ma ei peagi ristsõnu ja sudokusid ostma.“ Kuna kiidetakse seda, et kursus on eesti keeles ning väga hästi korraldatud, siis on toodud kommentaarides ka välja, et korraldajad võiksid sarnaseid kursusi veel korraldata ka teistel teemadel (java, html).

Kokkuvõte

Käesoleva lõputöö eesmärgiks oli analüüsida e-kursuse „Programmeerimise alused“ osalejate hinnanguid ülesannetele. Vaatluse alla võeti kursus, mis toimus 28. märtsist kuni 22. maini 2016. Andmete kogumiseks kasutati korraldajate poolt koostatud tagasisideküsitlust ehk vahküsitlust, mida osalejad pidid täitma üle nädala. Eesmärgist tulenevalt esitati neli uurimisküsimust, millele vastamiseks uuriti e-kursuse „Programmeerimise alused“ osalejate tagasisideküsitluse vastuseid. Kõigile neljale uurimisküsimustele leiti vastused.

Esimeseks uurimisküsimuseks oli, et millised ülesanded on kerged ja millised keerukad vaba juurdepääsuga e-kursuse „Programmeerimise alused“ osalejate endi hinnangul. Leiti, et iga nädala esimene ülesanne on kõige lihtsam võrreldes teiste ülesannetega ning keerukad ülesanded olid enamasti valikülesanded.

Teine uurimisküsimus oli, et milliseid valikülesandeid lahendati vaba juurdepääsuga e-kursusel „Programmeerimise alused“ kõige rohkem. Selgus, et enamus osalejaid valib alati esimese valikülesande ning kõige rohkem lahendati valikülesandeid seitsmendal nädalal.

Kolmas uurimisküsimus oli, et millised ülesanded on vaba juurdepääsuga e-kursuse „Programmeerimise alused“ osalejate jaoks huvitavad? Selgus, et osalejatele pakuvad kõige rohkem huvi ülesanded, mis on piisavalt rasked, elulised või praktilised.

Neljas uurimisküsimus oli, et milliseid soovitusi annavad osalejad vaba juurdepääsuga e-kursuse „Programmeerimise alused“ kohta. Soovitusi ja kommentaare oli mitmeid, kuid töö autor valis välja nendest soovitusid, mida esines mitmeid kordi ning mis tundusid töö autori arvates kasulikud kursuse parandamiseks. Soovitused on välja toodud peatükis „Arutelu“.

Antud töö alusel saavad kursuse korraldajad teha kursust paremaks muutes vajadusel kursusel olevaid ülesandeid. Samuti saab antud töö võtta aluseks uuele kursusele ülesannete loomisel. Käesolev töö võib olla ka alus uue uurimuse tegemiseks, kus keskendutakse ülesannete lahenduste analüüsimisele.

Kasutatud kirjandus

- [1] Anspal, S., Järve, J., Jürgenson, A., Masso, M., Seppo, I. (2014). Oskuste kasulikkus tööturul: PIAAC uuringu temaatiline aruanne nr 1. Tartu: Haridus- ja Teadusministeerium
https://www.hm.ee/sites/default/files/jarelevalve/oskuste_kasulikkus.pdf (22.02.2017)
- [2] Eesti Euroopa Liidu Poliitika 2015-2019. (2015).
<https://riigikantselei.ee/sites/default/files/content-editors/Failid/EL/elpol-2015-2019.pdf>
(22.02.2017)
- [3] Kaplan, A, M., Haenlein, M. (2016). Higher education and the digital revolution: About MOOCs, SPOCs, social media and the Cookie Monster. *Business Horizons*, Vol 59.
- [4] Shah, D. (2016). By The Numbers: MOOCs in 2016. Class Central:
<https://www.class-central.com/report/mooc-stats-2016> (20.03.2017)
- [5] E-kursuste “Programmeerimisest maalähedaselt” ja “Programmeerimise alused” võrdlus
<https://courses.cs.ut.ee/2016/progmaa/spring/Main/Maalahaalusedvordlus> (08.02.2017)
- [6] Programmeerimise alused II
<https://courses.cs.ut.ee/2017/eprogalused2/spring> (20.03.2017)
- [7] Programmeerimise alused kevadel 2016
<https://courses.cs.ut.ee/2016/eprogalused/spring/Main/HomePage> (08.02.2017)
- [8] Courses
<https://courses.cs.ut.ee/> (20.03.2017)
- [9] TÕ Moodle
<https://moodle.ut.ee> (20.03.2017)
- [10] Carbone, A., Hurst, J., Mitchell, I., Gunstone, D. (2000). Principles for Designing Programming Exercises to Minimise Poor Learning Behaviours in Students
https://www.academia.edu/15917780/Principles_for_designing_programming_exercises_to_minimise_poor_learning_behaviours_in_students?auto=download (06.04.2017)
- [11] Vihavainen, A., Paksula, M., Luukkainen, M. (2011). Extreme Apprenticeship Method in Teaching Programming for Beginners
https://www.cs.helsinki.fi/webfm_send/905 (06.04.2017)
- [12] Courney, M., Teague D., Thomas, R, N. (2010). Engaging Students in Programming
<http://crpit.com/confpapers/CRPITV103Corney.pdf> (06.04.2017)
- [13] Su, X., Wang, T., Qiu, J., Zhao, L. (2015). Motivating students with new mechanisms of online assignments and examination to meet the MOOC challenges for programming

- [14] Staubitz, T., Klement, H., Renz, J., Teusner, R., Meinel, C. (2015). Towards Practical Programming Exercises and Automated Assessment in Massive Open Online Courses
https://www.researchgate.net/publication/295903831_Towards_Practical_Programming_Exercises_and_Automated_Assessment_in_Massive_Open_Online_Courses (09.04.2017)
- [15] Staubitz, T., Teusner, R., Meinel, C., Prakash, N. (2016). Cellular Automata as Basis for Programming Exercises in a MOOC on Testdriven Development.
https://hpi.de/fileadmin/user_upload/fachgebiete/meinel/papers/2016_Staubitz_TALE.pdf (15.04.2017)
- [16] Lahtinen, E., Ruuska, M., Virtanen, A, T. (2005). Enhancing the learning result by adding repetitive complexity to programming assignments.
http://www.cs.tut.fi/~essi/IADAT_8.pdf (06.04.2017)
- [17] Onur Özgün, Yaman Barlas. (2015). Effects of systemic complexity factors on task difficulty in a stock management game
- [18] Liu, P., Li, Z. (2012). Task complexity: a review and conceptualization framework
https://www.researchgate.net/publication/232132658_Task_complexity_A_review_and_conceptualization_framework (01.05.2017)
- [19] Loria-Saenz, C. (2008). On Requirements for Programming Exercises from an e-Learning Perspective.
<https://arxiv.org/pdf/0903.0786.pdf> (16.04.2017)

Lisad

I. Ülesanded

Tabel 1. Ülesannete lingid

Nädal	Kontroll- ülesanded	Kontroll- ülesannete link	Valikülesanded	Valikülesannete link
1. nädal	1.1 Tervitus 1.2 Rasvatihane 1.3 Astendamine	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-sissejuhatus	1.4a Nädala ajakulu 1.4b Kiiruseületamise trahv	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-sissejuhatus
2.nädal	2.1 Jäätumine 2.2 Allveelaev 2.3 Leedu perenimed	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-tingimuslause	2.4a Pulss 2.4b Tribüün 2.4c Busside logistika	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-tingimuslause2
3.nädal	3.1 Vaateratas 3.2 Lillede arv 3.3 Täringumäng	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-whiletsykkel	3.4a Lillede arv v2 3.4b Vabavisked 3.4c Male	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-whiletsykkel2
4. nädal	4.1 Suured tähed	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-graafika	4.2a Eesti haldus üksuse lipp 4.2b Liiklusmärk 4.2c Seadme paneel	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-graafika
5.nädal	5.1 Mopeedid 5.2 Lillede arv 5.3 Sissetulekud	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-jarjendid	5.4a Reisiidid 5.4b Kümnevõistlus 5.4c Tahvli juurde	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-jarjendid2

6.nädal	6.1 Tervitused 6.2 Teleri suurus 6.3 Peo eelarve	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-funktsioon	6.4a Tervitused mõtisklustega 6.4b Mündid 6.4c Kuupäev	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-funktsioon2
7.nädal	7.1 Telegramm 7.2 Päevik 7.3 Kalkulaator	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-failifaili	7.4a Täiendatud peo eelarve 7.4b Nimepäev 7.4c Elutee number	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Kontroll-failifaili2
8.nädal			8.1a Arvestusülesanne 8.1b Loovtöö	https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Arvestusylesanne https://courses.cs.ut.ee/2016/epr/ogalused/spring/Main/Loovtoo

II. Lihtlitsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Sixtina Ladva**,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
E-kursuse „Programmeerimise alused“ osalejate hinnangud ülesannetele,
(lõputöö pealkiri)
mille juhendajateks on Merilin Säde ja Eno Tõnisson,
(*juhendajate nimed*)
 - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **11.05.2017**