

TARTU ÜLIKOOL

Loodus- ja täppisteaduste valdkond  
Tartu Ülikooli arvutiteaduse instituut  
Informaatika õppekava

Herman Rull

**Programmeerimiskeelte ja infoturbe teemade õppematerjalide  
loomine kursusele „Digitaalne maailmapilt”**

Bakalaureusetöö (9EAP)

Juhendaja: Jaak Vilo

Tartu 2022

# **Programmeerimiskeelte ja infoturbe teemade õppematerjalide loomine kursusele „Digitaalne maailmapilt”**

## **Lühikokkuvõte**

„Digitaalne maailmapilt” on Tartu Ülikoolis 2022.a esmakordselt toimunud kursus, mille eesmärk on tutvustada arvutiteaduse aluseid teiste õppekavade tudengitele. Lõputöö raames loodi kursuse „Digitaalne maailmapilt” programmeerimiskeelte ja infoturbe nädalate loengutele toetavad materjalid ning automaatkontrollitavad iseseisvad ülesanded üliõpilastele. Valminud materjale kasutati kevadsemestril toimunud piloot-läbiviimisel. Selle raames koguti andmeid üliõpilaste tulemuste kohta automaatkontrollides ning küsiti tagasisidet loengumaterjalide kohta. Lõputöö osana analüüsiti kogutud andmeid. Ühtlasi, kuna kursus on sissejuhatav aine arvutiteadusesse, siis üks lõputöö eesmärkidest oli pakkuda üliõpilastele võimalusi edasisteks õpinguteks. Seega uuriti bakalaureusetöös erinevaid kursusi, mis käsitlevad programmeerimiskeelte ja infoturbe teemasid põhjalikumalt, ning koostati soovituskaart jätkukursuste jaoks.

**CERCS:** P175 Informaatika, S270 Pedagoogika ja didaktika, S281 Arvuti õpiprogrammide kasutamise meetodika ja pedagoogika

## **Developing Study Materials for the topics Programming Languages and Cybersecurity for the Course “Digital Technology - an Overview”**

### **Abstract**

“Digital Technology - an Overview” is a course from University of Tartu that for the first time took place in 2022. The course aims to teach introductory level computer science to other fields. Within this thesis study materials and automatic grading tasks for the topics Cybersecurity and Programming Languages were developed for the course “Digital Technology - an Overview”. The developed materials were used in the spring semester of 2022 when the course was piloted. During the piloting of the course students’ feedback and grading results were gathered and analysed as part of the thesis. Since the course is introductory, then one of the main goals was to provide resources for further studies. In this Bachelor’s thesis different courses that cover programming languages and cybersecurity were reviewed and based on the review a suggested study map for students was developed.

**CERCS:** P175 Informatics, S270 Pedagogy and didactics, S281 Computer-assisted education .

## Sisukord

Lühikokkuvõte.....	1
Abstract.....	1
Sissejuhatus.....	3
1. Aine „Digitaalne maailmapilt” teemade varasemad käsitlelused.....	5
1.1 Sarnased kursused õppeainele „Digitaalne maailmapilt” .....	5
1.2 Programmeerimiskeelte kursused.....	6
1.3 Infoturbe kursused.....	8
2. Loengumaterjalide koostamine.....	10
2.1 Õpieesmärkide valimine.....	10
2.2 Loengumaterjalide koostamine.....	11
3. Automaatkontrollimeetodid ja nende rakendus teemades programmeerimiskeeled ning infoturve. .....	15
3.1 Erinevad automaatkontrolli meetodid.....	15
3.2 Automaatkontrollid kursusel „Digitaalne maailmapilt” .....	18
4. Üliõpilaste tagasiside koostatud õppematerjalidele.....	21
4.1 Üliõpilaste tagasiside teoreetiliste õppematerjalide kohta.....	21
4.2 Üliõpilaste tulemused programmeerimiskeelte testidel.....	24
4.3 Järeldused.....	27
5. Kokkuvõte.....	29
5.1 Tulemused.....	29
5.2 Edasiarenduse võimalused.....	30
6. Kasutatud kirjandus.....	31
Lisad.....	32

## Sissejuhatus

Arvutite laialdane kasutus on tekitanud suurema vajaduse infotehnoloogiliste oskuste ja hariduse järele ka muudes valdkondades peale informaatika. Peng [1] kirjutab oma artiklis, et digipädevustega inimesel on lihtsam tööd leida ja tööl püsida. Kuna ülikoolide taset sageli võrreldakse lõpetanute töömäära ja keskmise palga alusel, siis on ka ülikooli huvides, et tudengid saaksid konkurentsivõimelised alusteadmised infotehnoloogiast. Lisaks kirjeldab Johansson [2] oma raamatus Medici efekti, mille kohaselt innovatsioon tekib erinevate valdkondade ristumisel. Andes ülevaate, millega arvutiteadus tegeleb, suunab kursus „Digitaalne maailmapilt” ka üliõpilasi mõtlema erinevatele digilahendustele, kuidas oma valdkonnas protsesse tõhusamalt läbi viia. Seega kursusest, mis võtab lühidalt kokku arvutiteaduse põhialused on kasu nii üliõpilastele kui ka ülikoolile.

„Digitaalne maailmapilt” on 2022. aasta kevadsemestril esmakordselt läbiviidud kursus, mille eesmärk on anda edasi üldised teadmised arvutiteadusest 156 tunni ehk 6 Euroopa ainepunktisüsteemi ainepunkti(EAP) raames. Eeskätt on kursus mõeldud Tartu Ülikooli mitteinformaatika õppekavade bakalaureuse- ja magistritudengitele. Kursus koosneb 16 õppenädalast. Igal nädalal keskendutakse uuele valdkonnale arvutiteaduses. Iganädalased õppematerjalid koosnevad videotest ja iseseisvatest ülesannetest. Samuti on kursuse peale 4 kodutööd. Arvestatud on, et loengutele kulub ligikaudu 32 tundi ning ülejäänud 124 tundi on mõeldud iseseisva õppimise jaoks. Hindamine on eristav, kusjuures 40% hindest moodustavad 4 iseseisvat kodutööd ning 60% hindest kujuneb iganädalaste jooksvate testide põhjal. Kursus „Digitaalne maailmapilt” on Tartu Ülikooli esimene kursus, mis toimub õppekeskkonnas Coursera.

Coursera õppeplatvorm on mõeldud veebipõhiste massikursuste ehk MOOC-ide jaoks. MOOC-ide eripära on see, et nende läbiviimine on kõrgelt automatiseeritud. Loengud ja praktikumid on ettesalvestatud, üliõpilaste tagasisidestamine toimub automaatkontrollide abil ning probleemide korral suunatakse üliõpilasi kasutama foorumit, kus aktiivsed kaasüliõpilased või juhendajad suunavad õigele teele. „Digitaalne maailmapilt” on laia sihtrühmaga, mistõttu MOOC-i põhimõtete rakendamine on vajalik, et kursuse läbiviimine oleks jätkusuutlik ja seejuures ei peaks piirama üliõpilaste arvu. Coursera[3] õppekeskkond pakub sisseehitatud foorumit ja automaatkontrollijaid ning on juba laialdaselt kasutusel nii ülikoolides kui ka üliõpilaste seas.

Bakalaureusetöö eesmärk on luua õppeaine „Digitaalne maailmapilt” programmeerimiskeelte ja infoturbe nädalatele erinevaid õppematerjale ning analüüsida 2022. aasta kevadsemestri piloot-

läbiviimisel üliõpilastelt kogutud tagasisidet koostatud õppematerjalide kohta. Lõputöö raames koostatud õppematerjalid sisaldasid loenguslaide ja konspekti õppejõule kasutamiseks ning automaatkontrollitavaid ülesandeid õpilastele.

Lõputöö jaguneb neljaks peatükiks. Esimeses peatükis antakse ülevaade arvutiteaduslikke aluseid õpetavate kursuste varasematest käsitlustest. Samuti uuritakse põhjalikumalt programmeerimiskeelte- ja infoturbeteemalisi kursuseid. Teine peatükk kirjeldab loengumaterjalide loomist. Kolmandas peatükis keskendutakse automaatkontrollitavate ülesannete loomisele ning neljandas peatükis analüüsitakse tudengite tagasisidet programmeerimiskeelte teema materjalidele.

# **1. Aine „Digitaalne maailmapilt” teemade varasemad käsitlused**

Õppeaine „Digitaalne maailmapilt” ei ole esimene kursus, mille eesmärk on õpetada informaatika alaseid baasteadmisi. Peatüki esimeses pooles võrreldakse õppeainet „Digitaalne maailmapilt” teiste sarnaste Eesti ja välismaa ülikoolide poolt õpetavate kursustega. Peatüki teises ja kolmandas osas käsitletakse spetsiifilisemalt erinevaid programmeerimiskeelte ja infoturbe kursuseid. Seejuures hinnatakse ka kursuste sobivust õppeaine „Digitaalne maailmapilt” jätkukursusena tudengitele, kel on huvi programmeerimiskeelte või infoturbe teemadel ennast edasi harida. Lisas I on toodud soovituskaart kursusele „Digitaalne maailmapilt”, mis koostati peatükis vaadeldud programmeerimiskeelte ja infoturbe kursuste analüüsi põhjal.

## **1.1 Sarnased kursused õppeainele „Digitaalne maailmapilt”**

Digitaalsete oskuste õpetavaid kursusi on mitmeid nii Tartu Ülikoolis, Eestis kui ka mujal maailmas.

### **Sarnased kursused Eestis**

Emeriidotsendi Helle Hein kursus „Sissejuhatus informaatikasse” [4,5] on mõeldud varasema IT kogemusega tudengitele, kes jätkavad oma õpinguid informaatika valdkonnas. Õppeaine on 3 EAP-d ning selle üks olulistest eesmärkidest on motiveerida üliõpilasi edasistel õpingutel arvutiteaduses. Seega keskendutakse teemadele, mis on populaarsed ja huvitekitavad. Kursuse „Digitaalne maailmapilt” osaleja puhul ei saa eeldada, et ta võtab järgnevat kursusi informaatika valdkonnast, mistõttu on kursuse teemade valik pigem laiahaardeline andmaks ülevaate kogu arvutiteadusest.

„Digipädevuste alused”[4] on 3 EAP suurune Tartu Ülikooli Pärnu kolledži õppejõu Maarja Pajusalu digitaalsete töövahendite kasutust õpetav aine. Kursusele ligipääs on piiratud õppekavadele „Ettevõtlus ja projektijuhtimine” ning „Sotsiaaltöö ja rehabilitatsiooni korraldus”. Võrreldes selle kursusega on „Digitaalne maailmapilt” teoreetilisem ja kõigile Tartu Ülikooli õppekavadele avatud. Samuti ei oodata kursuse „Digitaalne maailmapilt” läbinult oskusi mõnes spetsiifilises rakenduses, vaid üldist arusaama, mis on arvutitega võimalik korda saata.

Tallinna Tehnikaülikoolis viiakse Tanel Tammeti juhendamisel läbi kursust „Sissejuhatus infotehnoloogiasse”[6]. See on mõeldud eelkõige informaatikutele sissejuhatusena analoogselt Tartu Ülikooli ainega „Sissejuhatus informaatikasse”, kuid räägib infotehnoloogia kujunemisest ajaloolises võtmes.

## **Sarnased kursused välismaal**

Harvardi Ülikool pakub tasuta veebikursust „CS50's Introduction to Computer Science”[7] ja MITil ehk Massachusettsi Tehnoloogiainstituudil on samuti avalik sissejuhatav kursus „Introduction to Computer Science and Programming”[8]. Mõlemad kursused on MOOCid ehk veebipõhised massikursused, kus hindamine on täiesti automaatne ning kursusest saab osa võtta igal ajal ja läbida omas tempos. Kuigi ka „Digitaalne maailmapilt” on laiemale ringkonnale, siis kursus on eesti keeles ja kursusest saab osa võtta vaid semestri sees. Lisaks on oluline eristada, et nii Harvardi kui ka MITi kursuse üks õpiväljund on oskus programmeerida, samas kui „Digitaalne maailmapilt” programmeerimisele nii palju rõhku ei pööra.

### **1.2 Programmeerimiskeelte kursused**

Alampeatükk sisaldab kahe erineva programmeerimiskeelte teemat tervikuna õpetava kursuse lühikirjeldust.

#### **Programmeerimiskeeled**

Courseras on kõrgelt hinnatud kolmeosaline Dan Grossmani ingliskeelne kursus „Programming Languages”. Kogu informatsioon kursuse kohta on võetud õppekeskkonnast Coursera [9]. Kursuse eeldatav maht on ligikaudu 65 tundi. Vajalikud on varasemad programmeerimisoskused. Igas osas keskendutakse uuele programmeerimiskeelele, mille põhjal selgitakse detailselt näidete põhjal, kuidas programmeerimiskeeled töötavad. Esimeses osas vaadatakse staatiliselt tüübitud funktsionaalse programmeerimiskeele Meta Language (ML) näitel, mis on andmetüübid, kuidas programmeerimiskeeled üldiselt toimivad aga ka seda, mis eristab ML teistest keeltest.

Teises osas on vaatluse all programmeerimiskeel Racket. Racket on samuti funktsionaalne keel, ent erinevalt ML-ist määrab see andmetüübid dünaamiliselt ehk programmi jooksutamise hetkel. Selles osas vaadatakse Racketi isepärasusi, luuakse Racketis interpretaator uue keele jaoks ning võrreldakse Racketi ja ML näitel staatiliste ja dünaamiliste keelte nõrkusi ja tugevusi.

Kolmas osa keskendub programmeerimiskeelele Ruby. Erinevalt varasemalt vaadatud keeltest, mis olid funktsionaalsed, on Ruby objektorienteeritud. Kolmandas osas kirjeldatakse tüüpimist Rubys, objektorienteeritud keelte eripärasid Ruby näitel ning võrreldakse funktsionaalseid keeli objektorienteeritud keeltega.

Kursus annab põhjaliku ülevaate kuidas programmeerimiskeeled toimivad. Analüüsides samu aspekte kolme erineva keele vaatenurgast, saab üliõpilane hea ülevaate erinevatest võimalustest

programmeerimiskeelte disainimisel. Selle käigus tekib arusaam disainide nõrkuste ja tugevuste kohta. Eelnevat arvesse võttes sobib hästi üliõpilasele, kes on huvitatud tarkvaraarendusest ning tahab silmaringi avardada.

### **Automaadid, keeled ja translaatorid**

„Automaadid, keeled ja translaatorid” on Tartu Ülikoolis Vesal Vojdani juhtimisel õpetatav kursus mahuga 6 EAP. Info kursuse kohta on kokku pandud Tartu Ülikooli õppeinfosüsteemi (ÕIS) [4] ja Tartu Ülikooli arvutiteaduse instituudi kursusi tutvustava lehe Courses [5] põhjal. Kursus on suunitlusega teise aasta informaatika tudengitele. Seega eeldatakse varasemat kokkupuudet programmeerimise ja Java keelega. Õppeaine käigus saab üliõpilane selgeks, kuidas luua oma programmeerimiskeel ja see arvutis tööle panna. Kursus on jaotatud kolme plokki:

- leksiline analüüs,
- süntaksanalüüs,
- semantiline analüüs ja koodi genereerimine.

Leksilise analüüsi osas käsitletakse, kuidas tuvastada sõnu, mis vastavad etteantud reeglitele. Eesmärgi saavutamiseks tutvustatakse madalama taseme teemasid nagu olekuautomaadid. Selle osa läbinud üliõpilane peaks oskama kirjutada programmi, mis tuvastab etteantud tekstist ära otsitavad sümbolid.

Süntaksanalüüsi nädalatel vaadatakse, kuidas leksilist analüüsi kasutades on võimalik leida grammatikale vastava struktuuriga laused. Ehk kui eelmises osas õpiti kuidas leida teatud osad lausest, siis peale süntaksanalüüsi nädalate läbimist peaks üliõpilane oskama programmeerida lahenduse, mis vastavate reeglite alusel sõnade vahele seosed ära tuvastab ehk loob abstraktse süntaksipuu.

Viimases osas seletatakse, kuidas kirjutada programmi, mis väärtustab abstraktse süntaksipuu. Samuti vaadatakse lühidalt, kuidas implementeerida staatilist ning dünaamilist tüüpimist. Viimase teemana õpetatakse kompilaatori programmi kirjutamist.

Kursus õpetab, kuidas nullist programmeerimiskeelt luua. Protsessi käigus saab praktilise kogemuse läbi selgeks, kuidas programmeerimiskeeled töötavad. See kursus sobib hästi üliõpilasele, kes tahavad asju ise läbi teha või kellel on selge soov töötada programmeerimiskeelte arenduse juures.

### **1.3 Infoturbe kursused**

Alampeatüki kõigi kolme kursuse kohta on informatsioon kogutud lehtedelt Courses[5] ja ÕIS[4].

#### **Infoturve**

„Infoturve” on Kristjan Kripsi 3 EAP suurune õppeaine, mille põhiline sihtrühm on mitteinformaatikud. Kursusel on kolm teemaplokki:

- a) andmete kaitsmine ja privaatsus,
- b) avaliku võtme infrastruktuur ja teenused,
- c) autentimine ja ründed.

Esimeses teemablokis räägitakse sümmeetrilisest krüptograafiast, andmekandjatest ning operatsioonidest nende peal. Analüüsitakse mobiilseadmete nõrkusi ja võrreldakse operatsioonisüsteemide iOS ning Android omadusi. Kolmas suur teema selles blokis on privaatsus veebis. Räägitakse küpsistest, veebibrauseritest, Torist ja VPN-ist.

Teises blokis alustatakse avaliku võtme krüptograafia ja selle jaoks vajaliku taristu kirjeldamisega. Edasi vaadatakse rakendusi, mis kasutavad avaliku võtme krüptograafiat – TLS protokoll, ID-kaart, mobiil-ID ning krüptovaluutad.

Kolmandas ehk viimases blokis käsitletakse autentimisvahendeid ning erinevaid küberrünnakute tüüpe. Ühtlasi vaadatakse kui turvalised on Wi-Fi ja mobiilside.

Õppeaine ei eelda varasemat kokkupuudet infoturbeaga. Sellest hoolimata suudetakse ära katta suur osa infoturbe valdkonnast. Ka keerulisemad teemad nagu e-hääletamine on võetud kursuse skooopi ning nende õpetamisel toetatakse kursusel varasemalt õpitule. Õppeaine jaoks on koostatud konspekt, mis sisaldab muu hulgas ka viiteid materjalidele edasiõppimiseks. „Infoturve” sobib hästi kursuse „Digitaalne maailmapilt” jätkuaineiks, kui eesmärk on saada parem tervikpilt infoturbest.

#### **Andmeturve**

„Andmeturve” on 6 EAP-line õppeaine, mille vastutav õppejõud 2021. aastast on Alo Peets. Kursuse põhiline sihtrühm on informaatika õppekava bakalaureuse tudengid ning sellel on kaks eeldusainet: „Programmeerimine” ja „Operatsioonisüsteemid”. Kursuse õppematerjalid on arvutiteaduse instituudi lehel avalikult kättesaadavad, mistõttu on võimalik ka asjatundlikemal tudengitel „Andmeturve” õppematerjalidega tutvuda. Võrreldes teiste infoturbe kursustega on

„Andmeturve” praktilisem. Suur osa üliõpilase tööst kulgeb arvutis iseseisvalt praktikumimaterjalide põhjal infoturbealaseid ülesandeid tehes.

## **Krüptoloogia I**

„Krüptoloogia I” on Dominique Peer Ghislain Dr Unruh juhendamisel antav ingliskeelne 6 EAP-line aine. Kursus läheneb infoturbele krüptoloogia vaatenurgast. Õppeainel ei ole eeldusaineid, kuid kokkupuude diskreetse matemaatikaga tuleb kasuks.

„Krüptoloogia I” eesmärk on tutvustada üliõpilasele krüptograafilisi primitiive. Seejuures kirjeldatakse, kuidas primitiivide tuntumad implementatsioonid töötavad, mis tingimustel need on turvalised ja kus neid otstarbekas kasutada on.

Kursus on teoreetilise kallakuga ning ennekõike mõeldud inimestele, kel on sügavam huvi matemaatika vastu, mis krüptograafiliste primitiivide taga toimub.

## 2. Loengumaterjalide koostamine

Kursusel „Digitaalne maailmapilt” kasutatakse teoreetilise osa õpetamiseks põhiliselt videoloenguid. Iga nädal tuleks vaadata läbi uus kogum videoid, mis keskenduvad teatud arvutiteaduse harule. Kokku on kursusel kuusteist erinevat teemat. Ühe teema videoloengute kogupikkus on ligikaudu 2 tundi. Lõputöö raames koostati 4. nädalal toimuva programmeerimiskeelte loengu ja 14. nädalal toimuva infoturbe loengu jaoks kirjalik konspekt ja slaidiesitus (Lisa I). Konspekti ja slaidide põhjal tegi kursuse vastutav õppejõud loenguvideod. Konspekti ülesanne oli toetada lektorit slaidide sõnumi tõlgendamisel, slide kasutati videoloengut visuaalselt toetava materjalina.

Peatüki esimese pooles käsitletakse õpieesmärkide valikut. Õpieesmärgid on põhilised ideed, mis kursuse läbinud üliõpilane peaks olema omandanud.

Õpieesmärkide põhjal koostati loengumaterjalid. Sisuloome juures lähtuti, et õppevideoid saaks vaadata väikesemahuliste tükkidena. Sisuloome on täpsemalt lahti kirjeldatud peatüki teises pooles.

### 2.1 Õpieesmärkide valimine

Kursuse sihtgrupiks on erineva taustaga ja varasemate arvutiteaduslike teadmisteta inimesed, seega oli eesmärgiks hoida loengud võimalikult lihtsad. Teisalt, kui teema jaoks on ainult üks nädal ja teema on lai, siis on oht jääda pinnapealseks või minna üle ettenähtud aja. Seega tuli õpieesmärkide valikul pidevalt hinnata nende keerukust, ajakulu ja olulisust.

Ülesande lihtsustamiseks oli vastutav õppejõud ette andnud teemad, millest võiks rääkida. Programmeerimiskeelte loengu jaoks olid etteantud erinevad programmeerimis- ja märgistuskeeled koos märksõnadega. Üks selline paar oli näiteks Python ja tüübisüsteem. Neist juhustest lähtudes püstitati õpieesmärgid järgnevate küsimuste ümber:

- a) mida teeb programmeerimiskeel,
- b) mis eristab programmeerimiskeeli tavalistest keeltest,
- c) kuidas erinevad programmeerimiskeeled
  - paradigmade vaatest,
  - jooksutamise seisukohalt(kompileeritav või interpreteeritav),
  - andmetüübi käsitlemise poolest,

d) olulisemad programmeerimiskeeled.

Programmeerimiskeeled on mitmes mõttes sarnased tavalistele keeltele. Pole palju reegleid, mis alati kehtiks. Võimatu on öelda, kui palju on programmeerimiskeeli, sest pole üheselt aktsepteeritud arusaama, mis on tarvilikud tingimused, et üht keelt lugeda programmeerimiskeeleks. Materjalides piirdume kahe tingimusega: Turingi täielikkus ehk keel peab olema suuteline tavalisele arvutile kõiki käske edastama ja keelel peab olema üheselt mõistetav grammatika.

Teiseks koostati lõputöö raames materjale infoturbe teemale. Võrreldes programmeerimiskeeltega on infoturbes vähem vastukäivaid materjale. Palju on erinevaid standardeid ning krüptograafia toetub matemaatilistele tõestustele, mis omakorda tähendab, et on olemas tõeallikas. Teisalt on infoturbe oluliselt laiem teema ning otsus, mida kursusel käsitleda ei olnud nii selge. Õpieesmärkide valikul lähtuti, et teemat võiks vaadelda rohkem arvutiteadlase seisukohalt. Seega valiti ennekõike teemad, mis on teoreetiliselt huvitavad mitte tingimata praktiliselt kasulikud. Infoturbe loengumaterjalid koostati järgmiste õpieesmärkide põhjal:

- a) millega tegeleb krüptograafia,
- b) mida teevad erinevaid krüptograafilised primitiivid:
  - krüpteerimine,
  - räsimine,
  - digiallkirjastamine,
- c) millised on võimalikud ründed krüptograafia tasemel,
- d) millised on levinumad rünnakud süsteemide vastu,
- e) millised on levinumad rünnakud kasutajate vastu,
- f) kuidas töötab ploki ahel.

## 2.2 Loengumaterjalide koostamine

Oluline aspekt, mida videoloengute koostamisel arvestati oli nende tükeldatavus. Tükeldatavus kirjeldab, kui hästi on võimalik materjale jaotada õpitükkideks. Õpitükk on väikseim osa, mille kaupa on võimalik õppematerjale tarbida. Kui loengu lõpus viitatakse näitele, mida tutvustati loengu alguses täpsemalt, siis on loeng halvasti tükeldatav, sest loengu lõpu osa eeldab, et vaataja on hiljuti vaadanud ka loengu algust. Tavalistes ülikooli loengutes ei ole see probleem, sest

üliõpilased kuulavad reeglina kogu loengu ühe korraga ära. Veebiloengute puhul on võimalus panna video pausile ja naasta hiljem. Selleks ajaks on üliõpilane suure tõenäosusega näite ära unustanud ja peaks loengu algusest seda otsima minema. Hästi tükeldatud loengu puhul on viited kapseldatud väiksematesse ajapiiridesse, mistõttu saab ta õppida paindlikumalt.

Guo jt. [10] on suuremahulises MOOC-ide uuringus leidnud, et optimaalne video pikkus maksimaalse vaataja tähelepanu saavutamiseks on kuus minutit. Lühemad videod ei sisalda piisavalt materjali ning pikemate videote puhul hakkab vaataja tähelepanu kaduma. Uuringust ei selgunud, kas oluline on video pikkus või õpitüki pikkus. Kui lindistamisel juba lähtutakse kuue minutilise video põhimõttest, siis kitsendatakse teemat, et see mahuks ära kuue minuti sisse. Seega lühikesed videod tingivad väiksed õpitükid, kuid väiksed õpitükid ei tähenda tingimata lühikesi loenguid. On võimalik, et tegelikult ei mängigi üliõpilaste tähelepanus niivõrd palju rolli video pikkus nagu artiklis väidetakse, vaid hoopis õpitüki suurus. Selle kasuks viitab ka samas uuringus[10] leitud asjaolu, et spetsiaalselt MOOC-i jaoks tehtud videod hoiavad kuulaja tähelepanu pikemalt kui tavalised klassiloengud tükeldatud kuue minutilisteks klippideks. Seda isegi olukorras, kus spetsiaalselt MOOC-i jaoks tehtud videod on pikemad. Eelnevast järeldub, et tükeldamine põhjal on oluline, sest võtab paremini arvesse inimeste tähelepanu pikkust ning parima tükeldatuse saavutamiseks peaks loengud olema kas lühikesed või eristatud selgelt eristatavateks õpitükkideks.

Programmeerimiskeelte ja infoturbe teemades tehti iga loengu jaoks eraldi plokk slaide. Plokk sisaldas vähemalt ühte õpitükki, kuid üldjuhul sisaldasid mitut. Kuigi õpitükid olid teineteisest sõltumatud, siis seoti sarnase sisuga õpitükid ploki tasemel.

Programmeerimiskeelte nädalal slaididel ettevalmistatud kolmest slaidiplokist läksid videote lindistamisel kasutusse kaks. Kolmanda ploki, kus oli eesmärk vaadata tuntumaid programmeerimiskeeli asendas lektor lühiülevaatega programmeerimiskeelte ajaloost, mille raames vaadati ka olulisemaid keeli programmeerimiskeelte ajaloo võtmes. Nende kahe ploki, mis videoloenguteks salvestati, pikkus tuli mõlemal 21 minutit. Esimene video sisaldas kolme õpitükki ning teine video sisaldas nelja(vt tabel 1).

Programmeerimiskeeled		
Plokk	Õpitükk	Kestus
Programmeerimiskeelte olemus	Mis on programmeerimiskeel?	3 minutit
	Millal võib lugeda keelt programmeerimiskeeleks?	10 minutit
	Miks on olemas erinevad programmeerimiskeeled?	7 minutit
Programmeerimiskeelte liigitamine	Kaugus masinkoodist(tasemed)	3 minutit
	Paradigmad	6.5 minutit
	Tüübid	3.5 minutit
	Kompileeritav või interpreteeritav	5 minutit
<i>Tuntumad programmeerimiskeeled</i>	<i>Assmebler</i>	-
	<i>C ja C++</i>	-
	<i>Python</i>	-
	<i>Bash</i>	-
	<i>Javascript</i>	-
	<i>XML, JSON, LaTeX</i>	-

Tabel 1: Programmeerimiskeelte nädala planeeritud struktuur plokide ja õpitükkide kaupa. Kestus on võetud lektori sisseloetud videotest. Kursiivis toodud teemasid loenguks ei salvestatud.

Infoturbe nädala jaoks koostati 4 plokki. Mõeldud oli kirjeldada algul infoturbe olemust ning seejärel vaadelda kolmel erineval tasemel infoturbe eesmärkide saavutamist(vt tabel 2). Loengute tegemisel lõi lektor krüptograafia osa kaheks ning tõstis plokiahela omaette plokki. Seega tuli infoturbe nädalal kokku 6 videot.

Infoturve		
Plokk	Õpitükk	Kestus
Sissejuhatus infoturbesse	Infoturbe olulisus	6 minutit
	Infoturbe eesmärgid	20 minutit
	Tasemed infoturbes	2 minutit
Turvalisuse tagamine krüptograafia tasemel	Põhimõisted	6 minutit
	Krüptograafilised primitiivid	6.5 minutit
	Sümmeetriline krüptograafia	4 minutit
	Võtmevahetus	7 minutit
	Avaliku võtme krüptograafia	30 minutit
	Räsimine	5 minutit
	Kvantarvuti	3 minutit
Turvalisuse tagamine rakenduse tasemel	Implementatsioonivead	4.5 minutit
	Digiallkirjastamine	5 minutit
	Plokiahel	25 minutit
Turvalisuse tagamine lõppkasutaja tasemel	Pahavara, õngitsemine ja halvad paroolid.	6 minutit

Tabel 2: Infoturbe nädala planeeritud ülesehitus plokkide ja õpitükkide kaupa.

Loengute sisu omandamist kontrollitakse kursusel „Digitaalne maailmapilt” automaatkontrollidega, mille koostamist vaadatakse järgmises peatükis.

### 3. Automaatkontrollimeetodid ja nende rakendus teemades programmeerimiskeeled ning infoturve.

Oluline osa õppeprotsessis on saada tagasisidet õpieesmärkide saavutamise kohta. Kursuse „Digitaalne maailmapilt” osalejate arv ei ole piiratud, mistõttu on personaalse tagasiside andmiseks vajalik protsessi automatiseerida.

Peatükis antakse ülevaade erinevatest automaatkontrollide koostamise meetoditest. Seejuures analüüsitakse nende plusse, miinuseid ja sobivust kursuse „Digitaalne maailmapilt” jaoks. Peatüki teises pooles kirjeldatakse täpsemalt, kuidas kursusel automaatkontrollide tegemine käis.

#### 3.1 Erinevad automaatkontrolli meetodid

Bey jt [11] toovad oma artiklis välja kolm põhilist viisi automaatseks hindamiseks: valikvastustega küsimustikud, kaaslaste hindamine (ingl *peer grading*), kus üliõpilased hindavad teineteise töid, ja automaattestid.

#### Valikvastustega küsimustikud

Igale küsimusele on küsimustiku koostaja ettevalmistanud kogumi vastustest. Neist vähemalt üks on tõene. Kontrollitav peab oma teadmiste tõestamiseks valima õige(d) vastuse(d) kogumist välja. Samuti määrab koostaja igale küsimusele väärtuse, mis kontrollitav õige vastuse korral kätte saab. Valede vastuse korral on koostaja otsus, kas ta tahab anda miinuspunkte ja mis ulatuses. Töö lõpus liidetakse kontrollitava punktid kokku ja hindamismaatriksi alusel saab üliõpilane hinde (vt joonis 1).

##### 1. Märki tõene väide, mis kehtib assemblerkeele kohta

1 point

- Assemblerkeelt kasutatakse kiirete prototüüpide tegemiseks tarkvaraarenduses
- Ühele protsessorile kirjutatud assembleri programm on lihtsasti porditav ka teist tüüpi protsessoritele.
- Assembler keele käsud on üks ühele vastavuses protsessori käskusega
- Tänapäeval enam ei kasutata assemblerit.

Joonis 1: Näide valikvastustega küsimuste kohta. Pakutud neljast variandist on vaid üks õige. Kui üliõpilane oskab õigesti vastata, siis saab ta ettenähtud koguse punkte. Antud juhul saab õige vastuse eest ühe punkti.

Valikvastustega küsimustikel on mitmeid häid külgi. Neid saab kasutada kõikjal, kus on võimalik kategoriseerida vastuseid õigeks ja valeks. Ühtlasi on see paljudesse õppekeskkondadesse nagu Moodle ja Coursera juba sisse integreeritud. Seega on nii kursuse läbiviijal kui ka üliõpilastel mugav neid kasutada, kuna kõik on ühes kohas. Kolmandaks on valikvastustega küsimused üliõpilastele mugavad, kuna õppuril ei ole vaja ise midagi kirjutada ning võtavad võrdlemisi vähe aega.

Valikvastustega küsimustikel on kolm suurt probleemi. Esiteks hakkavad kontrollitavad õigeid vastuseid omavahel jagama. Üks lahendus oleks vastused jätta salastatuks. Ehk kui üliõpilane esitab oma töö, siis ta saab teada punktid, kuid tagasisidet selle kohta, mis ta valesti vastas, ei saa. Tegemist siiski ei ole päris töökindla lahendusega, maksimumpunktidega üliõpilased teavad õigeid vastuseid ning kui neid on piisavalt palju saadakse lahendatud küsimustik kätte. Ühtlasi on tagasiside üliõpilase jaoks oluline ning selline kollektiivne karistusvorm, kus keegi ei saa tagasisidet petturi pärast on ebaõiglane ausate üliõpilaste suhtes. Teine mure valikvastustega küsimustikega on see, et osav vastaja võib välistades ilmselgelt valesid vastuseid, vastata õigesti küsimustele, millele ta ilma variante nägemata ei oskaks vastata. Kuna ta ei pea andma põhjendust, siis on ka raske tuvastada, kas üliõpilane oskab teemat või oskab hästi teste teha. Võrreldes teiste automaatkontrolli vormidega on see koostajale üsna aeganõudev. Ei piisa ainult heast küsimusest, vaid tuleb välja mõelda ka vastusevariandid, mis eristaks juhupakkujad õppinud testitavatest.

Kursusel „Digitaalne maailmapilt” sobiks kasutada selles vormis kontrollle loengujärgselt andmaks üliõpilasele kiire esmane tagasiside, kas loengust jäi oluline meelde. Selle kasutuse jaoks sobivad valikvastustega küsimused hästi, sest üliõpilase jaoks on see mugav, mis tähendab, et ta teeb neid suurema tõenäosusega ka siis, kui punkte selle eest ei anta. Samas kui punkte ei ole arvel, ei ole ka üliõpilasel motivatsiooni sohki teha.

### **Kaaslaste hindamine**

Kaaslaste hindamisel parandab iga õppur vähemalt ühte oma kaasõppuri tööd ning vähemalt üks kaasõppur hindab tema tööd. Nii palju, kui on kohustuslikus korras ette nähtud igal üliõpilasel hinnata, nii mitu erinevat tagasisidet saab ka iga üliõpilane. Kursuse läbiviija roll on anda hindamismatriks, mille põhjal osalejad saavad töid hinnata, ja tagada, et hindajad leiaks hinnatava(d) üles.

Meetod on väga universaalne, seda saab kasutada igas valdkonnas ja kui sobiv hindamismatriks koostada, siis puudub ka vajadus õiged vastused ära täpsustada. See on oluline näiteks kirjatööde

puhul, kus vaja hinnata subjektiivselt hinnatavaid aspekte. Samuti on võimalik kontrollida teadmisi palju sügavamal tasemel. Näiteks programmeerimiskeelte nädala õpieesmärkide kontrollimisel on sageli põhjendus olulisem kui vastus, sest sõltuvalt vaatepunktist võib ühele küsimusele olla mitu õiget vastust.

Põhiline probleem kaaslaste hindamisel on see, et see on üliõpilaselt üliõpilasele, mistõttu ei ole tagatud õiglane hindamine. Isegi, kui anda täpne maatriks, siis igal hindajal on oma arvamus, mis kallutab teda ühele või teisele poole. Seega kontrollitava hinne sõltub suurel määral, kes juhtub teda hindama. Lahendus sellele oleks suurendada tööde arvu, mis iga inimene peab hindama. Niimoodi koguneb rohkem hinnanguid igale tööle ja saab tasakaalustuma keskmise, mis on õiglasem. Paraku kuskil jookseb piir, kui palju võib kursusel osalejatel teiste töid lasta parandada. Nimelt tuleb see aeg võtta mingi muu õpiväljundi arvelt, et 6 EAP sisse ära mahtuda.

Lõputöö autori koostatud materjalides kaaslaste hindamist ei rakendata, sest see ei mahuks ühe nädala jaoks ettenähtud aja sisse.

## Automaattestid

Automaattestide kasutatakse arvutiprogrammide testimiseks. Eristatakse kahte sorti automaattestide: staatilised ja dünaamilised. Staatilised testid kontrollivad programmi ilma jooksutamata. Selleks võrdlevad nad koodi paljude korrektsete näidislahendustega. Dünaamilised testid käitavad koodi erinevate sisenditega ning kontrollivad, kas programm annab oodatud vastuse [11]. Vale vastuse korral reeglina tagastatakse veateade ja antakse võimalus uuesti esitada. Joonistel 2 ja 3 on toodud kursuse „Digitaalne maailmapilt” jaoks loodud näidiskontrollifunktsioon koos dünaamiliste automaattestidega, mille eesmärk oli näidata, kuidas automaattestid töötavad.

```
def demo_liitja(a,b):  
    """  
    Näidis, kuidas automaattestid töötavad.  
    test_demos on kirjutatud testid, mille vastu funktsiooni kontrollitakse. Jooksuta faili test_demo erinevate  
    tagastatavate väärtustega ja vaata kas testid ebaõnnestuvad.  
    """  
    return a+b  
    # return abs(a)+abs(b)  
    # return a+b+1
```

Joonis 2: Näidis kursusel üliõpilastele automaattestid tutvustamiseks. Lihtne funktsioon, mille ülesanne on liita kaks arvu.

```

from importlib import import_module
yl = import_module('14_Andmeturve.main')

def test_demo_positiivsed_arvud():
    assert yl.demo_liitja(4, 5) == 9

def test_demo_negatiivsed_arvud():
    assert yl.demo_liitja(-2, -3) == -5

```

Joonis 3: Automaattestid, mis kontrollivad funktsiooni `demo_liitja` rakendamist.

Automaattestidega saab üliõpilasele anda personaalset tagasisidet. Kontrollides lahenduse väikseid osasid saab tudeng täpselt teada, mis läks õigesti ja mis valesti. Samuti annab programmeerimisülesanne hea arusaama, kas tudeng on teemast aru saanud. Kui ei ole, siis automaattestide pideva kiire tagasisidestamise tõttu, on võimalik ka kiiremini õppida. Programmeerimisülesannete puhul on võimalikke lahendusi mitmeid, mistõttu on ebatõenäoline, et kaks tudengit lahendaksid ülesande täpselt samamoodi. Seetõttu on plagiaadivastajat kasutades võimalik vähendada juhtumeid, kus esitatakse kaaslane töö oma nime all.

Automaatteste saab kasutada vaid arvutiprogrammide testimiseks. Seega ei pruugi õppekeskkonda sisse olla integreeritud eraldi automaattesti keskkonda, kus üliõpilased saaksid lahendusi esitada. Dünaamiliste testide puhul peab olema olemas korrektne näidislahendus, mille alusel on võimalik genereerida sisend-vastus paare, mille vastu üliõpilaste programme testida. Staatilise testimise puhul on vaja korrektseid näidislahendusi rohkem. Täpne kogus sõltub programmi pikkusest ja keerukusest. Mõlemal juhul peab koostaja palju aega panustama, et automaattestikeskkond tööle saada.[11]. Samuti on vaja hoida pidevalt töös arvutit, mis testib lahendusi ja annab tagasisidet.

Automaatkontrollide koostamisel kasutati automaatteste kõikide programmeerimisülesannete juures. Kuigi kursuse „Digitaalne maailmapilt” prioriteet ei ole programmeerimise õpetamine, siis arvutiteadus on tugevasti programmeerimisega põimunud ning sageli on programmeerimisülesanded parim viis teadmiste kontrollimiseks.

### **3.2 Automaatkontrollid kursusel „Digitaalne maailmapilt”**

Lõputöö raames koostati valikvastustega küsimustik programmeerimiskeelte loengu jaoks ning programmeerimisülesanded koos testidega infoturbe loengu jaoks.

## **Programmeerimiskeelte küsimustik**

Programmeerimiskeelte loengutes omandatud teadmisi otsustati kontrollida valikvastustega küsimustikuga. Loengus käsitleti teemasid teoreetiliselt, mistõttu ei olnud võimalik koostada asjakohaseid programmeerimisülesandeid.

Programmeerimiskeelte test sisaldas 18 küsimust. Küsimustik koostati pärast videote lindistamist, mistõttu lähtuti küsimuste koostamisel loengust.

Küsimustik sai koostatud Coursera malli järgi. Õppekeskkond Coursera võimaldab oma integreeritud autokontrollijas lisaks tavapärasele üks kuni mitu õiget vastust küsimustele, kasutada ka kasutaja sisendi põhiseid küsimusi. Tekstilise sisendi korral kontrollitakse vastust regulaaravaldisega ning arvulise sisendi korral peab üliõpilase vastus kuuluma etteantud arvuvahemikku. Samuti nõudis mall, et iga vastusevariandi juures oleks täpsustatud, miks on see õige või vale.

## **Infoturbe programmeerimisülesanded**

Infoturbe on olemuselt praktilisem kui programmeerimiskeeled, mistõttu oli seal otstarbekas testida teadmisi programmeerimisülesannetega ja jätta tagasisidestamine automaattestidele. Et tudengitel ei ole kursusele eelnevalt programmeerimiskogemusi, siis ükski viiest ülesandest ei vaja programmeerimisoskusi. Samas tuleks programmeerides õige vastus oluliselt kiiremini välja. Lisas I on kursuse koodivaramu link, kust leiab nii koostatud ülesanded kui ka testid.

Esimeses ülesandes tuli leida räsi, mis algab nelja null bitiga. Vormis on etteantud meetod, mis leiab suvalisele sisendile vastava räsi. Seega tudeng peab erinevaid variante läbi katsetama ja vaatama, kas vastus sobib. Keskmiselt tuleb proovida 16 erinevat varianti, ent ülesanne võiks suunata neid kirjutama tsüklit, mis leiaks sobiva räsi. Ülesande käigus veendub tudeng, et otsitavat räsi ei ole võimalik kõiki variante läbi proovimata genereerida.

Teises ülesandes on antud meetod sõnumi Caesari šifriga krüpteerimiseks ning tudengi ülesanne on murda lahti etteantud krüpteeritud sõnum. Ülesande eesmärk on kontrollida arusaama krüpteerimise ja dekrüpteerimise mõistest.

Kolmandas ülesandes on antud meetodid veebilehe sertifikaatide kontrollimiseks ja üliõpilase ülesanne on leida üks veebileht, mis ei läbi kontrolli. Erinevalt varasematest ülesannetest, kus kontrolliti teadmisi, on selle ülesande eesmärk harida, mida loengus mainitud lukumärgid veebibrauserites päriselt tähendavad.

Neljandas ja viiendas ülesandes on tudengitele ette antud räsitud ja viiendas ülesandes ka soolatud paroolid ning vajalikud meetodid räside genereerimiseks. Mõned räsied on tekitatud levinud paroolidest, teised on väga lühikesed. Üliõpilaste ülesanne on kas toorjõurünnakuga või levinuid parooli ära arvates 3 salasõna lahti murda. Harjutuse eesmärk on näidata tugeva parooli vajadust ning soolamise ja räsamise kasulikkust.

Ülesannete jaoks koostati automaattestid. Ülesanded ei olnud tegelikult sisendist sõltuvad, mistõttu piisas iga ülesande kontrollimiseks vaid ühest testist.

2022. aasta kevadsemestri kursusel ei kasutatud programmeerimisülesandeid, sest Courseras ei olnud võimalik integreerida ülesandeid õpikeskkonda. Oleks olnud võimalus kasutada välist lahendust, kus tudengid lahendavad ülesanded ning sealt jõuavad hinded Courserasse. Sellise lahenduse implementeerimist hinnati piloot-läbiviimise jaoks liiga keeruliseks.

Järgmises peatükis vaadatakse, kuidas tudengitel automaatkontrollides läks nii statistika kui ka nende endi subjektiivse hinnangu järgi. Ühtlasi analüüsitakse üliõpilaste tagasisidet üldisemalt ja pakutakse välja lahendusi, mida tulevastel aastastel teistmoodi teha.

## **4. Üliõpilaste tagasiside koostatud õppematerjalidele**

Üliõpilaste tagasisidet õppematerjalide kohta koguti 2022. aasta kevadsemestril toimunud kursuse raames (vt Lisa II). Peatükis analüüsitakse üliõpilaste antud tagasisidet programmeerimiskeelte nädala õppematerjalidele ja testidel. Lõpuks hinnatakse mida tagasisidest järeltada ning tulevikus parandada. Kuna lõputöö esitamise hetkel ei olnud 2022. aasta kevadsemestri infoturbe teemat veel käsitletud, siis tagasisidet sellele nädalale lõputöö raames ei analüüsita.

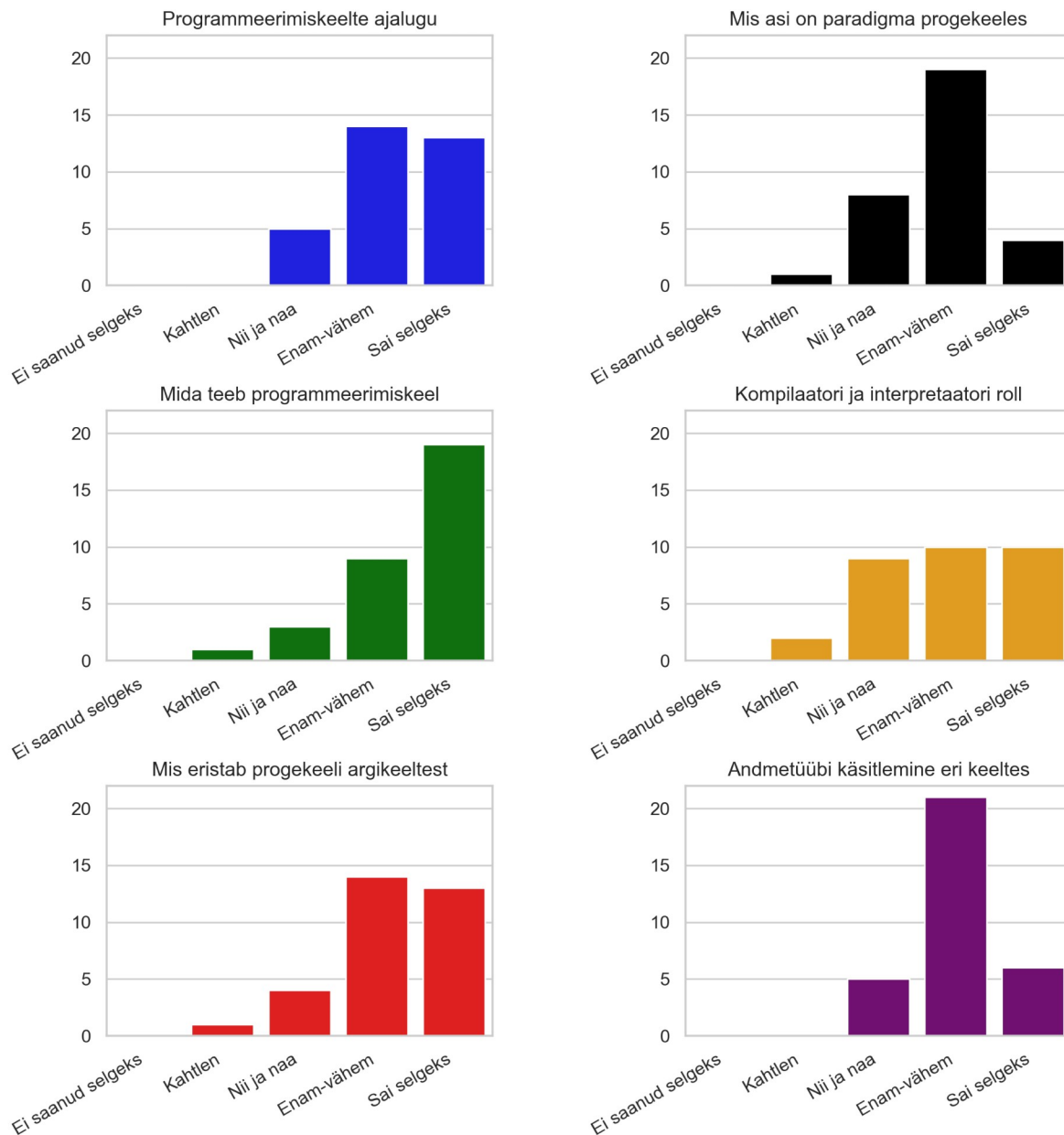
### **4.1 Üliõpilaste tagasiside teoreetiliste õppematerjalide kohta**

Teoreetiliste õppematerjalide kohta koguti tagasisidet küsimustike abil. Küsimustiku täitmine oli vabatahtlik ning selle eest boonuspunkte ei antud. Seetõttu koostati küsimustikud võimalikult lühikesed ja avatud küsimused olid vabatahtlikud, et üliõpilased ikkagi leiaksid aja ja motivatsiooni küsimustikku täita. Kokku andis tagasisidet 32 inimest. Tagasisides uuriti põhiliselt kolme järgneva aspekti kohta:

- a) üliõpilaste endi subjektiivne hinnang, kui hästi nad õpieesmärkidest aru said;
- b) üliõpilaste hinnang õpetamisele;
- c) ajakulu programmeerimiskeelte teema läbimiseks.

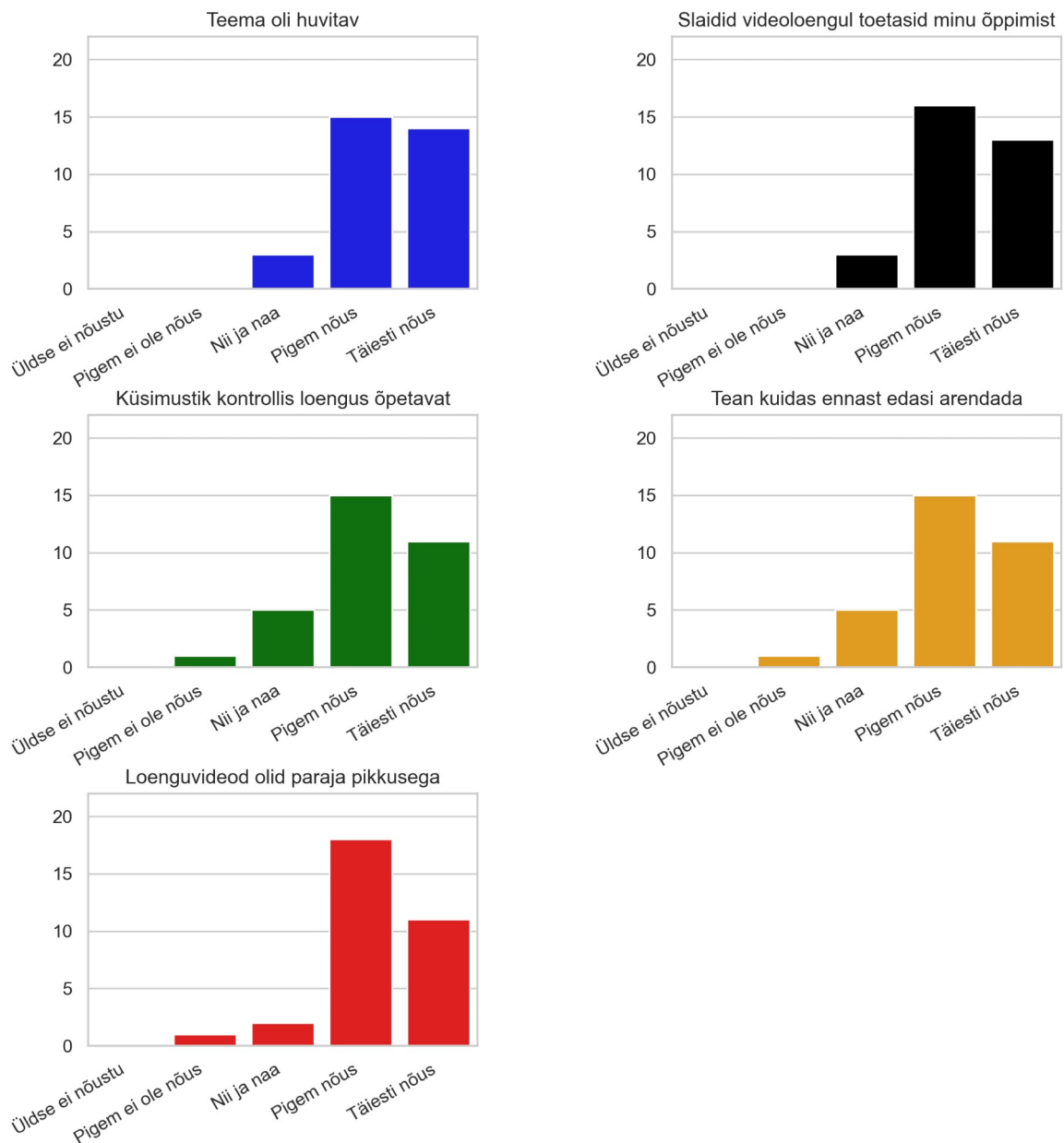
Subjektiivne tunnetus õpieesmärkide saavutamisest on oluline, sest teemabloki lõpetanud inimene võiks tunda, et on targemaks saanud ning tal võiks olla enesekindlus vastavate teemade raames suhtlemisel. Kokku oli kuus õpiväljundit ning küsimustikus uuriti üliõpilaste tagasisidet iga osa kohta viiepallisüsteemis.

Joonisel 4 on kujutatud üliõpilaste tagasiside jaotus. Üliõpilased tundsid, et kõige paremini said nad aru sellest, mida programmeerimiskeel teeb. Rohkem kui pooled üliõpilased hindasid oma teadmisi kõrgeima hindega ning keskmine hinne oli 4,44. Kõige raskemateks teemadeks osutusid programmeerimisparadigmad ning kompilaatori ja interpretaatori erisuste käsitus. Mõlema keskmise hinne oli 3,81. Teemadel andmetüüpide kasutus, programmeerimiskeelte ajalugu ning programmeerimiskeelte erinevus tavakeeltest jäi keskmine 4,03 ja 4,25 vahele.



Joonis 4: Koondülevaade üliõpilaste tagasisidest õpiväljundite saavutamise kohta. Y-teljel on kujutatud üliõpilaste arv ning x-teljel on kujutatud erinevad valikuvariandid ühest viieni teisendatud fraasilisteks hinnanguteks.

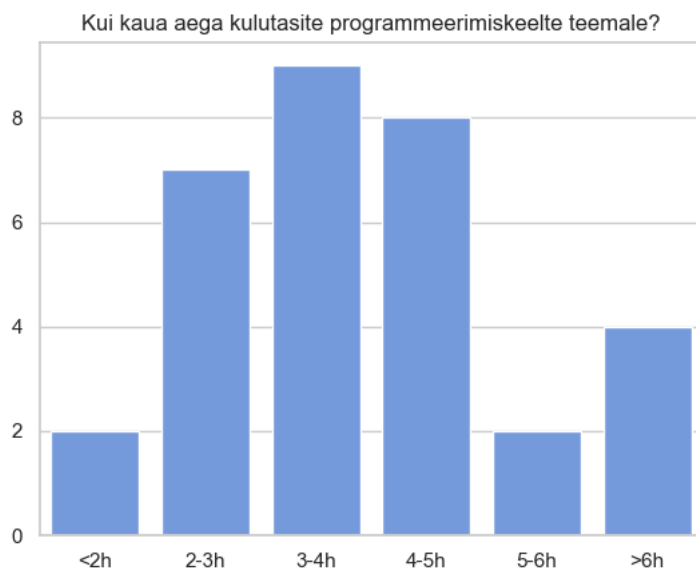
Teiseks uuriti üliõpilaste tagasisidet materjalide kohta üldiselt. Joonisel 5 on näidatud üliõpilaste tagasiside jaotus. Tagasiside oli valdavalt positiivne. Vähemalt nelja punkti vääriliseks hindasid 90,6% üliõpilastest programmeerimiskeelte teema huvitavust. Sama suures ulatuses pidasid inimesed ka videoloengute pikkust sobivaks ning slide toetavaks. Mõnevõrra kriitilisemalt suhtuti kontrollküsimustesse ja edasiste õppimisvõimaluste tutvustamisesse. 81,3% vastanutest nõustusid, et küsimustik kontrollis loengus õpetatud teadmisi ning sama protsent üliõpilastest vastas, et teavad, kuidas ennast edasi arendada.



Joonis 5: Koondülevaade üliõpilaste tagasisidest õpetamisele. Y-teljel on kujutatud üliõpilaste arv ning x-teljel on kujutatud erinevad valikuvariandid ühest viieni teisendatud fraasilisteks hinnanguteks.

Kolmandaks uuriti küsimustikus üliõpilastelt ajakulu kohta (vt joonis 6). Võttes arvesse, et lisaks iganädalastele loengutele ja kontrollküsimustikele on üliõpilastel ka kursusel „Digitaalne maailmapilt” 4 mahukat kodutööd, siis pelgalt programmeerimiskeelte teema jaoks võiks minna ligikaudu neli tundi. Joonisel 6 on kujutatud üliõpilaste ajakulu jaotust. Rohkem kui 4 tundi läks 14

üliõpilasel, mis moodustab vastanutest 43.75%. Neist neljal üliõpilasel kulus rohkem kui kuus tundi.



Joonis 6: Ajakulu jaotus küsimustiku täitnud üliõpilaste seas. Koondülevaade üliõpilaste tagasisidest õpiväljundite saavutamise kohta. Y-teljel on kujutatud üliõpilaste arv ning x-teljel on kujutatud nende ajakulu tundides.

## 4.2 Üliõpilaste tulemused programmeerimiskeelte testidel

Tudengitel ei olnud ajalimiiti testi tegemiseks ning nad said sooritada testi lõpmata palju kordi. Ainuke piirang oli, et 8 tunni jooksul saab ühte testi kolm korda proovida. Testi läbimiseks oli vaja saada 50% punktidest. Teste sai sooritada Courseras.

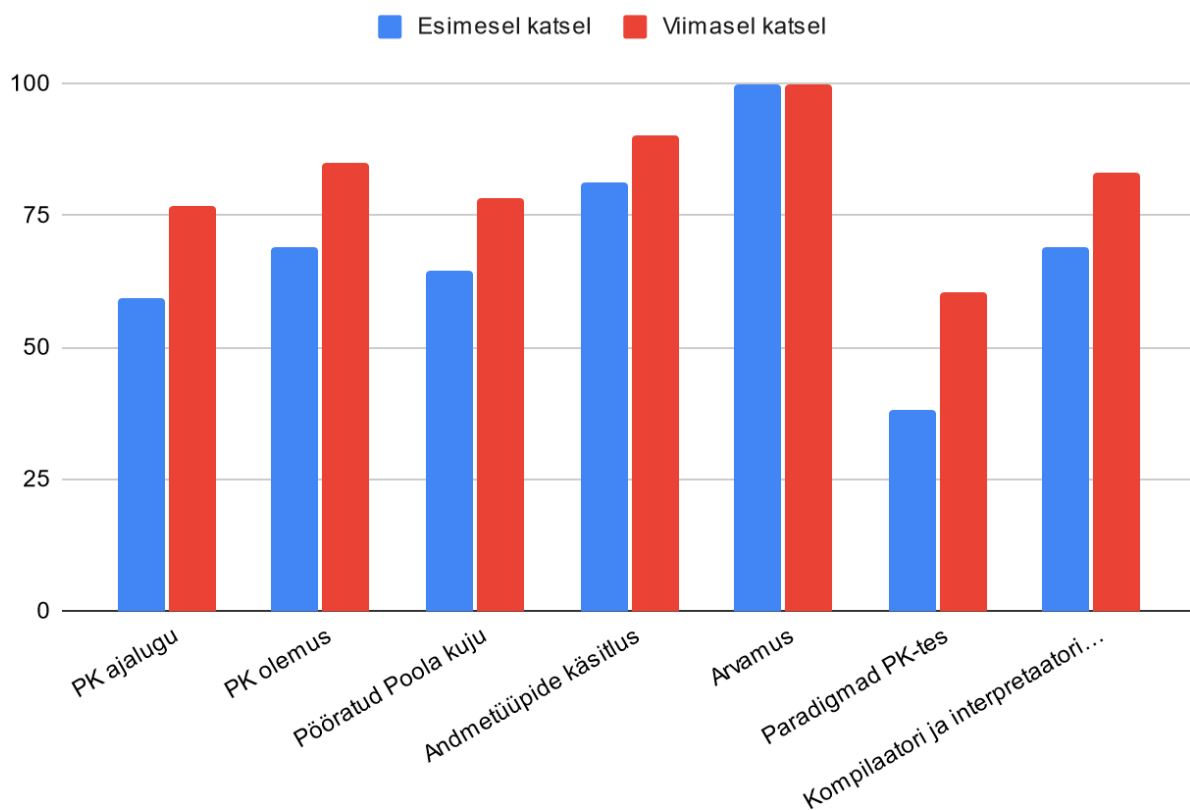
Kokku sooritas testi 109 üliõpilast. Keskmiselt sooritati testi 1,7 korda, kusjuures esimese lahenduskorra keskmine tulemus oli 68% ning viimase soorituse keskmine oli 82% maksimumskoorist.

Testi küsimused jaotati sõltuvalt õpiväljundist, mida nad kontrollivad, erinevatesse kategooriatesse. Lisatud said kategooriad arvamusküsimuste ja pööratud Poola kuju kohta käivate küsimuste kohta. Arvamusküsimustes said üliõpilased juba vastamise eest täispunktid. Pööratud Poola kujust räägiti pinu põhjal töötavate arvutite juures programmeerimiskeelte ajaloo kontekstis. Pööratud Poola kuju kohta sai koostatud 4 küsimust ning kuna need ei kontrollinud otseselt programmeerimiskeelte ajalugu, siis sai loodud neile omaette kategooria. Teiseks liideti mõned õpiväljundit kokku, kuna

küsimusi oli nende kohta liiga vähe. Küsimused selle kohta, mida programmeerimiskeeled teevad ja mille poolest nad erinevad tavakeeltest, seoti kokku kategooriasse programmeerimiskeele olemus.

Joonisel 7 on kujutatud üliõpilaste tulemus testil sõltuvalt kategooriast. Kui arvamusküsimused välja jätta, siis kõige paremini vastasid üliõpilased küsimustele andmetüüpide kohta. Seevastu kõige keerulisemaks osutusid küsimused paradigmatel. See oli vastavuses ka üliõpilaste enda subjektiivse hinnanguga, mis teema neil arusaamatuks jäi. Teine teema, mis tudengitele tundus keeruline oli kompilaatori ja interpretaatori roll. Samas testist ei tulnud välja, et neil oleks olnud suuremaid probleeme selle kategooria küsimustele vastamisel võrreldes teiste temadega. Kompilaatori ja interpretaatori rolli küsimustele oli vastatud paremini kui küsimustele programmeerimiskeelte ajaloo ja olemuse kohta, ehkki subjektiivselt olid need teemad arusaadavamad.

### Õpilaste tulemus erinevate kategooriate lõikes, mõõdetud protsentides maksimumkoorist



Joonis 7: Programmeerimiskeelte(PK) valikvastustega küsimustikus õigesti vastanute osakaal erinevate kategooriate lõikes. Y-teljel on kujutatud õpilaste protsentuaalne tulemus maksimumkoorist.

Kogutud andemete põhjal ei olnud üliõpilaste subjektiivne hinnang oma õpieesmärkide saavutamise kohta statistiliselt olulises korrelatsioonis testi tulemustega. Tabelis 3 on näidatud, et nii Spearmani korrelatsioonikordaja kui ka lineaarne korrelatsioonikordaja viitasid nõrgale positiivsele seosele üliõpilaste testi tulemuste ja nende endi hinnangu teemade arusaadavuse kohta. Lineaarse korrelatsioonikordaja ja olulisuse leidmiseks võrreldi üliõpilaste testi tulemuste keskmist (ÜTTK) erinevate kategooriate lõikes üliõpilaste subjektiivse hinnanguga vastava kategooria arusaadavuse kohta (ÜSHK). Näitaja ÜTTK arvutati testi esimese ja viimase soorituse jaoks eraldi. Spearmani korrelatsioonikordaja arvutamiseks toetuti samadele andmetele, ent ÜTTK ja ÜSHK keskmised väärtused paigutati järjekorda – kõige madalamale keskmisele väärtusele vastas järjekorra number 1 ja kõige paremini sooritatud testi kategooria sai kõige kõrgema järjekorra numbri. Tabelis 3 on näitajast ÜTTK tuletatud pingerea koht ehk Spearmani astak markeeritud lühendiga ÜTTA (üliõpilaste testi tulemuste keskmiste Spearmani astak) ning näitajast ÜSHK tuletatud astak tähistatud lühendiga ÜSHA (üliõpilaste subjektiivse hinnangu keskmiste Spearmani astak). Selleks, et oleks võimalik uurida küsimustiku õpieesmärkide saavutamist testi kategooriatega, jäeti statistikast arvamusküsimused ja pööratud Poola kuju küsimused välja. Samuti leiti kahe õpieesmärgi, programmeerimiskeelte tööpõhimõte ning programmeerimiskeelte erisus tavakeeltest, tulemustest keskmine, et oleks võimalik võrrelda üldisema kategooriaga – programmeerimiskeelte olemus.

Kategooria	Lineaarse korrelatsioonikordaja leidmine			Spearmani korrelatsioonikordaja leidmine			
	ÜTTK(%)		ÜSHK	ÜTTA		ÜSHA	
	esimene katse	viimane katse		esimene katse	viimane katse		
Programmeerimiskeelte ajalugu	59,49	76,86	4,25	2	2	3	
Programmeerimiskeelte olemus	69,21	84,85	4,32	4	4	4	
Andmetüüpide käsitus	81,23	90,28	4,03	5	5	2	
Paradigmad programmeerimiskeeltes	38,22	60,42	3,81	1	1	1	
Kompilaatori ja interpretaatori roll	69,10	83,31	3,81	3	3	1	
	esimene katse		viimane katse		esimene katse		viimane katse
<b>Korrelatsioonikordaja(r)</b>	0,3325		0,396		0,4104		0,4104
<b>Statistiline olulisus(p)</b>	0,4210		0,3315		0,4925		0,4925

Tabel 3: Statistika õpieesmärkide saavutamise kohta üliõpilaste subjektiivsel hinnangul ja testi tulemuste põhjal.

### 4.3 Järeldused

Nii tagasiside küsimustikust kui ka testitulemustest selgus, et kõige problemaatilisem teema oli paradigmade käsitus programmeerimiskeeltes. Võttes arvesse, et 14 inimesel 32-st läks üle eeldatud ajalimiidi, siis üks lahendus oleks teema kursuse skoobist täiesti välja jätta. Teise lahendusena oleks võimalik teemat kitsendada. Kursuse piloot-läbiviimisel prooviti anda täielik ülevaade erinevatest paradigmadest. Käsitleti nii imperatiivseid ja deklaratiivseid keeli koos nende alamkategoriatega kui ka kirjeldatud metaprogrammeerimise põhimõtet. Keskendudes vaid imperatiivsete ja deklaratiivsete keelte võrdlusele oleks võimalik laiali valgumata paradigmade rolli kirjeldada.

Üliõpilaste hindasid tagasiside küsimustikus kõige madalamalt loengujärgsete ülesannete seost loengumaterjalidega ja edasiõppimisvõimaluste tutvustamist. Probleemile loengujärgsete ülesannete ja loenguteemade vahel viitab ka nõrk korrelatsioon üliõpilaste subjektiivse arusaama õpieesmärkide saavutamise ja testi tulemuste vahel. Kui küsimused oleksid täpselt loenguteemade kohta küsitud ja sama keerukuse tasemega, siis peaks olema oodata, et ka kõige halvemini vastatud küsimuste kategooria kattub vastava õpiväljundiga. Samuti küsimuste kategooriate ja õpieesmärkide eristamine on märk sellest, et küsimused päris täpselt ei kattu õpetatuga. Üliõpilaste

edasiõppimisvõimaluste puudumine võis olla tingitud sellest jätkukursuste soovituskaart ei olnud 2022. aasta kevadsemestri tagasiside küsimise hetkeks valmis.

## 5. Kokkuvõte

Peatükis käsitletakse kokkuvõtvalt lõputöös tehtut ning pakutakse erinevaid edasiarendus võimalusi nii kursuse jaoks üldiselt kui ka programmeerimiskeelte ja infoturbe spetsiifiliselt.

### 5.1 Tulemused

Bakalaureusetöö eesmärk oli koostada arvutiteadusesse sissejuhatava kursuse „Digitaalne maailmapilt” jaoks erinevaid õppematerjale ning analüüsida üliõpilaste tagasisidet neile. Õppematerjalid hõlmasid endast infoturbe ja programmeerimiskeelte loengumaterjale, automaatkontrolle ja jätkukursuste soovitusi.

Lõputöös uuriti 3 infoturbealast kursust ning 2 programmeerimiskeelte kursust, mis sobiksid taseme poolest õppeaine „Digitaalne maailmapilt” lõpetanud üliõpilasele jätkukursuseks. Samuti koostati lõputöö osana soovituskaart, mis sõltuvalt üliõpilase eesmärgist, pakub välja sobiva jätkukursuse.

Lõputöö juhendaja ja kursuse „Digitaalne maailmapilt” vastutava õppejõu sisendi põhjal pandi paika õpieesmärgid ehk tuumik, mida kursuse lõpetanud üliõpilane peaks teadma. Nende põhjal koostati toetavad loengumaterjalid lektorile – slaidid ja konspekt. Slaidid koostati arvestades MOOC-i head tava hoida loengud väikeste tükkidena tarbitavad. Kokku tuli 21 õpitüki jagu slaide, mis jagunesid 7 loengu vahel. Slaidide põhjal lektori sisseloetud videoloengud tulid programmeerimiskeeltes keskmiselt 21 minutit ning infoturbes 25 minutit pikad ning sisaldasid konteksti muutusi, kus üliõpilased said tähelepanu hajumisel video pausile panna.

Bakalaureusetöö osana koostati kursuse programmeerimiskeelte ja infoturbe jaoks automaatkontrollid õpieesmärkide saavutamise kontrollimiseks. Programmeerimiskeelte teema jaoks valmistati 18 valikvastustega küsimust ning infoturbe teema jaoks 5 programmeerimisülesannet koos dünaamiliste automaattestidega. Programmeerimiskeelte kontrollküsimusi kasutati 2022. aasta kevadsemestri pilootkursusel, ent dünaamilisi teste ei rakendatud üliõpilastele mugava testikeskkonna puudumise tõttu.

Lõputöös analüüsiti üliõpilaste tulemusi programmeerimiskeelte teema jaoks koostatud ülesannetes. Samuti küsiti tudengitelt subjektiivset tagasisidet õpieesmärkide saavutamise kohta ning võrreldi üliõpilaste subjektiivset tunnetust nende tulemustega ülesannetes. Üldjoontes olid tudengid programmeerimiskeelte käsitlemisega rahul. Õppematerjalide seisukohalt hinnati kõige madalamalt loengute sisu ja automaatkontrollides küsitavate küsimuste seost. Seda kinnitas ka asjaolu, et

teemade, mida üliõpilased raskeks pidasid, ja testide tulemuste vahel ei olnud tugevat seost. Infoturbe teema analüüs jäi lõputööst välja, sest lõputöö esitamise hetkeks ei olnud seda veel 2022. aasta kevadsemestril toimunud.

## **5.2 Edasiarenduse võimalused**

Mõned ideed, kuidas kursust ja loodud materjale saaks edasi arendada:

1. Kontrollküsimused loengutele. Küsimused avanevad loenguaknas kuulajale, kui ta jõuab ettemääratud kohani videos. Ühest küljest suunaks see üliõpilasi mõtlema õpitu üle ja annaks üliõpilasele kiire tagasiside, kas teema sai selgeks. Teisalt võimaldaks kontrollküsimused paremini erinevaid teemasid ühes videos segmenteerida. Küsimus markeerib ühe teema lõppu ja teise teema algust, andes üliõpilasele võimaluse pausi tegemiseks.
2. Lisaks edasistele jätkukursustele võiks koostada ka nimekirja materjalidest õpieesmärgi spetsiifiliselt. Väljapakutud jätkukursuste maht on üsna suur. Kõige väiksem jätkukursus on 3 EAP-d, mida on teise õppekava õppimise kõrvalt päris keeruline üliõpilasel tunniplaani mahutada.
3. Täiendada loengukonspekte, mis loengute jaoks loodi ning teha need tudengitele kättesaadavaks.

## 6. Kasutatud kirjandus

- [1] Peng G. Do computer skills affect worker employment? An empirical study from CPS surveys. *Comput Hum Behav* 2017;74:26–34. <https://doi.org/10.1016/j.chb.2017.04.013>.
- [2] Johansson F. *The Medici Effect: Breakthrough Insights at the Intersection of Ideas, Concepts, and Cultures*. Harvard Business School Press; 2004.
- [3] Coursera’s Mission, Vision, and Commitment to Our Community | Coursera. Coursera’s Mission Vis Commit Our Community Coursera s.a. <https://about.coursera.org/> (3. aprill 2022).
- [4] ÕIS II s.a. <https://ois2.ut.ee/#/dashboard> (15. aprill 2022).
- [5] Courses - Institute of Computer Science s.a. <https://courses.cs.ut.ee/> (30. aprill 2022).
- [6] Sissejuhatus infotehnoloogiasse – Lambda s.a. [http://www.lambda.ee/wiki/Sissejuhatus\\_infotehnoloogiasse](http://www.lambda.ee/wiki/Sissejuhatus_infotehnoloogiasse) (8. mai 2022).
- [7] CS50’s Introduction to Computer Science. EdX s.a. <https://www.edx.org/course/introduction-computer-science-harvardx-cs50x> (8. mai 2022).
- [8] Guttag J. *Introduction to Computer Science and Programming*. MIT OpenCourseWare s.a. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/> (8. mai 2022).
- [9] *Programming Languages, Part A*. Coursera s.a. <https://www.coursera.org/learn/programming-languages> (8. mai 2022).
- [10] Guo PJ, Kim J, Rubin R. How video production affects student engagement: an empirical study of MOOC videos. *Proc. First ACM Conf. Learn. Scale Conf.*, New York, NY, USA: Association for Computing Machinery; 2014, lk 41–50. <https://doi.org/10.1145/2556325.2566239>.
- [11] Bey A, Jermann P, Dillenbourg P. A Comparison between Two Automatic Assessment Approaches for Programming: An Empirical Study on MOOCs. *Educ Technol Soc* 2018;21:259–72.

## **Lisad**

### **I Õppematerjalide lingid**

**Jätkukursuste soovitused -**

<https://courses.cs.ut.ee/2022/digit/spring/Main/J%c3%a4tkukursused>

### **Programmeerimiskeeled**

Loengukonspekti link -

[https://docs.google.com/document/d/1jsQ0bdN6LCKxVOI6X0ESVepVEIn\\_bvIUff1IO4beQSM/edit?usp=sharing](https://docs.google.com/document/d/1jsQ0bdN6LCKxVOI6X0ESVepVEIn_bvIUff1IO4beQSM/edit?usp=sharing)

Programmeerimiskeelte slaidiesituse link enne loengute salvestamist -

<https://docs.google.com/presentation/d/1zixxwyZpdokIIJo4wTjVL5QU3YhixsC19ajkOeffMOQ/edit?usp=sharing>

### **Infoturve**

Loengukonspekti link -

[https://docs.google.com/document/d/1jsQ0bdN6LCKxVOI6X0ESVepVEIn\\_bvIUff1IO4beQSM/edit?usp=sharing](https://docs.google.com/document/d/1jsQ0bdN6LCKxVOI6X0ESVepVEIn_bvIUff1IO4beQSM/edit?usp=sharing)

Slaidiesituse link enne loengute salvestamist -

<https://docs.google.com/presentation/d/1LYScE0xDN-4dxHv7OaKk111PGtXj3T-gBt7fYtwxzn0/edit?usp=sharing>

Programmeerimisülesannete ja automaatsetide link -

[https://github.com/digitMM/2021\\_22\\_kevad/tree/main/14\\_Andmeturve](https://github.com/digitMM/2021_22_kevad/tree/main/14_Andmeturve)

## II Tagasisideküsitlus

### Õpievesmärkide täitmise küsimustik \*

	Ei ole arusaadav	Kahtlen	Nii ja naa	Enamvähem	Sai selgeks
Programmeerimiskeelte ajalugu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mida teeb programmeerimiskeel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mis eristab progekeeli argikeeltest	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mis asi on paradigma progekeeles	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kompilaatori ja interpretaatori roll	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Andmetüübi käsitlemine eri keeltes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Teema õpetamise tagasiside \*

	Üldse ei nõustu	Pigem ei ole nõustu	Nii ja naa	Pigem nõus	Täiesti nõus
Teema oli huvitav	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Küsimustik kontrollis loengus õpetavat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Loenguvideod  
olid paraja  
pikkusega

Slaidid  
videoloengul  
toetasid minu  
õppimist

Tean, kuidas  
ennast edasi  
arendada

Kui kaua aega kulutasite Programmeerimiskeelte teemale? \*

- <2h
- 2-3h
- 3-4h
- 4-5h
- 5-6h
- >6h

Muu tagasiside progekeelte kohta

Your answer

Muu tagasiside aine kohta

Your answer

### III Litsents

Mina, Herman Rull,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

#### **Programmeerimiskeelte ja infoturbe teemade õppematerjalide loomine kursusele „Digitaalne maailmapilt”**

mille juhendaja on Jaak Vilo,

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 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Herman Rull*  
**10.05.2022**