

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Malin Püttsepp**  
**Projektijuhendi koostamine kursusele**  
**„Programmeerimine“ LTAT.03.001**  
**Bakalaureusetöö (9 EAP)**

Juhendaja:  
Tauno Palts, PhD

Tartu 2025

# **Projektijuhendi koostamine kursusele „Programmeerimine“**

## **LTAT.03.001**

### **Lühikokkuvõte:**

Projekt on oluline osa „Programmeerimine“ LTAT.03.001 kursuse läbimisest, selle eesmärgiks on realiseerida oma programmeerimisalaseid oskusi rühmakaaslastega koostöös. Bakalaureusetöö eesmärk on luua kursusele projekti etappide kirjeldused, projekti hindamismaatriksid, Courses lehekülg, mis kõiki projektimaterjale koondab. Bakalaureusetöö tulemusena valmisid eelmainitud materjalid ning üliõpilaste vastatud küsimustikust selgus, et juurde oleks vaja projektitööriistadest juhendeid Githubile, moodulitest sooviti kõige rohkem Pygame juhendit ning programmeerimiskeskondadest oli populaarseim Visual Studio Code. Praktikumijuhendajad andsid hindamismaatriksile mitmeid soovitusi, kuidas teha seda õpilastele arusaadavamaks, milliseid kohti peaks täpsustama ning täiendama.

**Võtmesõnad:** programmeerimine, programmeerimise projekt, projektimaterjalid

**CERCS:** P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

## **Creating a project guide for the programming course „Programmeerimine“ LTAT.03.001**

### **Abstract:**

The project is an important part of completing the course „Programming“ LTAT.03.001, to apply programming skills in collaboration with group members. This bachelor's thesis idea is to create descriptions for the project stages of the course „Programming“ LTAT.03.001, along with project evaluation matrices, a Courses page that consolidates all project materials. As a result of the thesis, the materials mentioned earlier were completed, and a student questionnaire revealed that additional guides for project tools, especially for GitHub, were needed. Among the modules, the most requested was a guide for Pygame, and the most popular programming environment was Visual Studio Code. Teaching assistants provided several suggestions for improving the evaluation matrix, including how to make it more understandable for students and which parts should be clarified and expanded.

**Keywords:** programming, programming project, project guide

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

## Sisukord

1. Sissejuhatus.....	5
2. Teoreetiline taust.....	6
2.1 Kursus „Programmeerimine“.....	6
2.1.1 Kursuse „Programmeerimine“ LTAT.03.001 ülevaade.....	6
2.1.2 Programmeerimiskeel Python algajate õpetamisel.....	6
2.1.3 Uute projektimaterjalide vajadus.....	7
2.2 Rühmatööd programmeerimise õppimisel.....	8
2.2.1 Meeskonnapõhine õpe.....	8
2.2.2 Rühmatöö kitsaskohad.....	9
2.2.3 Projekti idee.....	9
2.2.4 Rühmatööde esitlused.....	10
2.3 Projektijuhendi loomise põhimõtted.....	10
2.3.1 Teised projektijuhendid Tartu Ülikooli arvutiteaduste instituudis.....	10
2.3.2 Hindamismaatriksi loomine.....	11
2.3.3 Projektijuhendite loomine.....	12
3. Metoodika.....	13
3.1 Projektijuhendi ja materjalide loomine.....	13
3.2 Tagasisideküsimustik õpilastele.....	13
3.3 Tagasisideküsimustik praktikumijuhendajatele.....	14
4. Tulemused.....	15
4.1 Valminud projektimaterjalid.....	15
4.2 Üliõpilaste tagasiside projektidele.....	16
4.3 Juhendajate tagasiside projektide läbiviimisele ja projekti-materjalidele.....	19
4.4 Arutelu.....	21
5. Kokkuvõte.....	23
Viidatud kirjandus.....	24
Lisad.....	26
Lisa I. Küsimused juhendajatele.....	26
Lisa II. Küsitlus üliõpilastele.....	29
Litsents.....	32

# 1. Sissejuhatus

Tartu Ülikooli õppeinfosüsteemi<sup>1</sup> kohaselt on kursus „Programmeerimine“ LTAT.03.001 üks suurimaid Tartu Ülikooli kursuseid– see on kohustuslik kaheteistkümnele õppekavale ning eeldusaineks 37-le järgnevale kursusele. Kursuse „Programmeerimine“ raames tuleb sooritada projekt, mille eesmärgiks on realiseerida programmeerimisalaseid teadmisi koostöös rühmakaaslastega.

Antud kursusel tehakse rühmatööd paarides. Paaristöö on õppevorm, kus kaks õpilast töötavad ülesande täitmise nimel üheskoos. Teistega koos töötamisega peab üliõpilane kokku puutuma mujalgi, tulevikus töö juures või teistel kursustel. Rühmatööd tehes saab üliõpilane võimaluse arendada enda koostöövõimet ja üksteisest sõltumist, õppida efektiivselt kasutama vastavaid vahendeid ning vastutada enda loodud lahenduse eest. Lisaväärtusena saab õpilane esinemiskogemuse oma projekti rühmakaaslastele esitledes ning kasutada oma kriitilist mõtlemist teistele tagasisidet andes.

Selle bakalaureusetöö eesmärk on luua kursusele „Programmeerimine“ projekti etappide juhendid, hindamiseks hindamismatriksid ja Courses lehekülje, mis koondab kõiki projekti loomiseks vajaminevaid materjale ja infot. Konkreetsed projekti etappide juhendid, materjalid ning hindamismatriksid aitavad üliõpilasel täpsemini kaardistada temalt nõutavaid kriteeriume ning aitavad ka hindajate tööd teha kiiremaks ja vähem subjektiivsemaks. Courses leht aitab üliõpilasel kiirelt leida vajamineva informatsiooni, mida tal projekti tegemiseks vaja võib minna.

Bakalaureusetöö on jaotatud kolmeks peatükiks. Esimene peatükk tutvustab töö teoreetilist tausta, mille alla kuuluvad informatsioon kursusest, rühmatöödest programmeerimise õppimisel ning projektijuhendi loomise põhimõtted. Teine peatükk käsitleb meetodilist poolt materjalide valmimisest ning küsitlustest üliõpilastele ja praktikumijuhendajatele. Kolmas peatükk annab ülevaate tulemustest, millised materjalid valmisid, milline oli tagasiside üliõpilastelt ja juhendajatelt, ning millised on võimalused parandusteks ja edasiarenduseks.

---

<sup>1</sup> <https://ois2.ut.ee>

## 2. Teoreetiline taust

### 2.1 Kursus „Programmeerimine“

#### 2.1.1 Kursuse „Programmeerimine“ LTAT.03.001 ülevaade

Tartu Ülikooli arvutiteaduse instituudi poolt läbiviidav kursus „Programmeerimine“ (LTAT.03.001) on paljudele üliõpilastele esimeseks kokkupuuteks Pythoniga. Kursuse eesmärk on alustadmiste andmine programmeerimises ning esmased oskused algoritmide ja programmide loomiseks<sup>2</sup>. Kursuse jooksul on õppijad kohustatud tegema paaristööna enda valitud teemal projekti, kus hinnatakse valminud projekti funktsionaalset korrektsust, loetavust, lahenduse selgitamisoskust ning raskusastet. Kursuse üheks õpiväljundiks on oskus realiseerida programmeerimisalaseid projekte koostöös rühmakaaslastega. Kursuse „Programmeerimine“ sihtrühmaks on eelkõige reaalvaldkondade üliõpilased. Kursus on kohustuslik informaatika, matemaatika, matemaatilise statistika, füüsika, keemia ja materjaliteaduse üliõpilastele, aga valitav ka kõigile teistele üliõpilastele, kellel on soov läbida programmeerimise algõpe. 2024. aasta sügisel oli kursusele registreerunute arv 405, mis teeb sellest ühe suurima arvutiteaduse instituudi kursuse. Aine õppematerjalid asuvad Moodle'is<sup>3</sup> ning Courses lehel<sup>4</sup>.

#### 2.1.2 Programmeerimiskeel Python algajate õpetamisel

Kursusel „Programmeerimine“ kasutatakse programmeerimiskeelena Pythonit. Pythoni veebilehel kirjeldatakse Pythonit järgnevalt: „Python on võimas... ja kiire; töötab hästi koos teistega; töötab igal pool; on sõbralik & lihtne õppida; on avatud. Need on mõned põhjustest, miks inimesed, kes kasutavad Pythonit ei eelistaks kasutada midagi muud.“ (Autori tõlge)<sup>5</sup>. Pythoni arendas 80-ndate aastate lõpus hobikorras välja Hollandi programmeerija Guido van Rossum [1]. Programmeerimiskeel mängis suurt rolli G. V. Rossumi teoses „Computer Programming for Everybody“. Oma essees kirjutab Rossum, kuidas Python on hea keel, mida õpetada algajale: Pythoni kogukond on saanud palju tagasisidet inimestelt, kes on õpetanud

---

<sup>2</sup> <https://ois2.ut.ee/#/courses/LTAT.03.001/version/lt-2024-autumn-fulltime/details>

<sup>3</sup> <https://moodle.ut.ee/course/view.php?id=500>

<sup>4</sup> <https://courses.cs.ut.ee/2024/programmeerimine/fall>

<sup>5</sup> <https://www.python.org/about/>

oma lastele Pythonit [2]. TIOBE indeksi<sup>6</sup> kohaselt on Python populaarsuselt esimene programmeerimiskeel maailmas.

Python on hea programmeerimiskeel, mida õpetada algajatele, sest [1]:

- sellel on lihtne süntaks mis jääb hästi meelde;
- internetis on palju allikaid, mis on õppimisel abiks;
- Python ühildub paljude nüüdisaja moodulitega;
- lihtne on analüüsida koodi ja leida vigu, kaasa arvatud silurit kasutada.

Tartu Ülikoolis on Courses lehe<sup>7</sup> järgi programmeerimiskeel Python kasutusel mitmetel kursustel, näiteks „Masinõpe“ MTAT.03.227, „Kõrgjõudlusega andmetöötlus“ LTAT.06.026, „Transformerid“ MTAT.06.055 ja „Sissejuhatus andmeteadusesse“ LTAT.02.002.

### 2.1.3 Uute projektimaterjalide vajadus

Kursuse „Programmeerimine“ üheks kandvaks osaks on paaristöö projekt. Varasemalt oli projektijuhend erinevate etappidena, aga puudu oli ühtlus, abijuhendid ja hindamismaatriksid. Ilma hindamismaatriksita võib olla projektide hindamine subjektiivne ja üliõpilasele võib jääda arusaamatuks, mida projektilt oodatakse. Hindamismaatriksi olemasolu aitab õppejõul ja praktikumijuhendajal hindamisprotsessi kiirendada ja lihtsustada, anda tagasisidet ning vähendada vaidlusi hinde üle [3]. Üliõpilasel aitab hindamismaatriks saada aru õppejõu ootustest, mõista õpiväljundit, saada konkreetsemat tagasisidet oma soorituse kohta [3].

Kuna kursusel „Programmeerimine“ õpivad peale informaatikute veel mitmete teiste erialade üliõpilased, kes ei pruugi olla tuttavad arvutiteadustes kasutatavate projektivahenditega, näiteks matemaatiline statistika ning füüsika, keemia ja materjaliteadus, siis nendele oleks vaja rohkem tutvustada, milliseid mooduleid ja tööriistu on võimaik kasutada projekti haldamiseks. Olemasolevad materjalid, mida kursusel kasutati projekti tööriistade, näiteks Trello<sup>8</sup> ja GitHubi<sup>9</sup>, tutvustamiseks on hetkel veebiõpikus „Informaatika valikkursus ja valikmoodulid gümnaasiumitele“, mis vajavad osalist täiendamist. Näiteks kui õpilane kasutab operatsioonisüsteemina MacOSi või Linuxit, siis võib jääda GitHubi materjalidest puudu, sest vajalikus sektsioonis ei ole piisavalt infot ega näiteid võrreldes Windowsi käsurea õpetusega. Samal juhendil on GitHubi veebi- ja töölaueversiooni kasutamise osas vaid YouTube'i videod,

---

<sup>6</sup> <https://www.tiobe.com/tiobe-index/>

<sup>7</sup> <https://courses.cs.ut.ee/courses/index/2024/fall?>

<sup>8</sup> <https://web.htk.tlu.ee/digitalu/tarkvara2/chapter/projektihalduse-keskkond-trello/>

<sup>9</sup> <https://web.htk.tlu.ee/digitalu/tarkvara2/chapter/git/>

mis on paraku ingliskeelsed - ka need vajaksid ekraanitõmmiste ning lühikeste kirjeldustega eestikeelset materjali.

Trello juhendile oleks vaja lisada, milliseid toiminguid oma märkmega teha saab. Praeguses juhendis on ainult info, kuidas märget ühest rühmast teise tõsta, kuigi tegelikult saab sellele määrata vastutajat, aega ning muid kasulikke lisasid.

## **2.2 Rühmatööd programmeerimise õppimisel**

Rühmatööks nimetatakse õppevormi, kus õpilane teeb ühistööd. Järgnev peatükk analüüsib rühmatöö olulisust ning kuidas rühmas töötamine ning meeskonnapõhine õpe saab õppimisele kaasa aidata.

### **2.2.1 Meeskonnapõhine õpe**

Meeskonnapõhine õpe on pedagoogiline strateegia, mis kaasab õpilaste teadmisi individuaalsete testide ning grupitöö kaudu [4]. Sama allikas väidab, et järgides individuaalselt saadud vastuseid jätkavad õpilased gruppides ning töötavad läbi samu probleeme, arutledes vastuste üle. Selline protsess motiveerib õpilasi pannes neid vastutama oma lahenduse ees.

Meeskonnapõhise õppe suureks osaks on rühmatöö. Kõige tavalisem on luua rühmad õppijate omaduste või õpioskustasemete põhjal [5]. Sama allika kohaselt on soovitatav komplekteerida rühmad ikkagi erinevate oskustega õpilastega.

Rühmatöö positiivseteks komponentideks on [5]:

1. üksteisest sõltumine - kogu meeskond püüab saavutada ühte sama eesmärki;
2. individuaalne panustamine - vastutus enda ja teiste ees;
3. koostöö - arutelu ja probleemide lahendamine koos;
4. hindamine - rühma liikmed hindavad üksteise panust ja koostööd.

Raamatu „Kaasav haridus“ [5] kohaselt kaasnevad rühmatöödega ka mõned probleemid. Teatud sihtrühmi võib olla raskem rühmatöösse kaasata, näiteks sotsiaallärevad õppijad. Grupitöös on oluline leida tasakaal sotsiaalse väljakutse ja akadeemilise raskusastme vahel, kordamist ning kinnistamist on parem teha grupina, aga uue teema õppimist on parem teha individuaalselt.

## 2.2.2 Rühmatöö kitsaskohad

Informaatika projektid käituvad õppekeskkonnas natuke kummaliselt, kui neile püütakse rakendada standarditele vastavaid raamistikke [6]. Uuringu autorid R. Putter ja M. Lehner on proovinud kaardistada neli peamist kriitilist edutegurit.

1. Üliõpilased ei ole kogenud projektidega töötama ning seetõttu tekivad tihti probleemid juba projekti algfaasis. Neid probleeme tuleb aga lahendada ja lahendamiseks kuluvat aega ei saa kasutada projekti tegemisel.
2. Üliõpilaste motivatsioon projekti lõpetamiseks varieerub madalast väga kõrgeni.
3. Suhtlus üliõpilase ja praktikumijuhendaja vahel sõltub juhendaja konkreetsetest oskustest ja kokkupuutest antud teemaga ning sellest, kas üliõpilased julgevad varakult küsida näiliselt rumalaid küsimusi.
4. Projekti või projektiidee päritolu.

Uuringu kohaselt üheks faktoriks, mis kõvasti mõjutas üliõpilaste subjektiivset suhtumist oli projekti või selle idee päritolu

## 2.2.3 Projekti idee

Projekti puhul on esimene oluline asi leida ennast huvitav idee. Uuring, mille teostajateks on R. Putter ja M. Lehner [6], käsitles projektide hinnete võrdlust sõltuvalt sellest, kas idee tuli üliõpilastelt või õppejõududelt. Esmapilgul võib tunduda, et üliõpilased, kelle projektid olid nende enda ideedest peaksid saama paremad hinned, kui need, kelle ideed olid pakutud õppejõudude poolt.

Statistilises analüüsis jagati projektid kaheks rühmaks, 226 üliõpilast, kellel oli oma idee ning 220 üliõpilast, kelle projektiidee oli õppejõu pakutud. Hinnete keskmistest selgus, et oma ideega üliõpilaste keskmine hinne oli 1.90 ning õppejõudude poolt pakutud ideega üliõpilaste keskmine hinne oli 1.64 (Austria süsteemi järgi on 1 parim hinne ning 5 halvim). Üllatuslikult olid õppejõudude projektidel paremad tulemused. Autorid sellist tulemust ei oodanud.

Suur erinevus kahe grupi vahel tuleb kogemusest, mis projekti soovitajal on. Paljude üliõpilaste puhul tihti kogemuse puudumise tõttu kipuvad nad vajaminevat pingutust alahindama ja koostavad projekte, mille täitmine osutub lõpuks keeruliseks.

Sellest vaatenurgast näib selge, et õpilaste pakutud ideed on head, aga õppejõudude poolt pakutavate projektide tulemused on tihti paremad. Aga tuleb meeles pidada, et kehvem tulemus ei tähenda vähemat õpikogemust ja oskuste arengut.

## 2.2.4 Rühmatööde esitlused

Rühmatööde valmimisel on hea enda projekti ka teistele tutvustada. Uuringus, kus käsitleti projektipõhise õppimise integreerimist suuliste esitlustega, leiti et õpilaste suuliste esitluste tõttu esitatava sisu kvaliteet paranes [7]. Varem võisid nad võtta infot veebiotsingutest, aga suuline esitlus ajendas neid oma infot rohkem kontrollima ning kuulajate veenmiseks tõendeid hankima. Õpilased on oma esitluses kindlamad, kui esitletav töö põhineb nende endi kogemustel ja konkreetsetel tõenditel.

Projekti esitlust tehes oleks hea sisu kohandada sihtgrupile, et ei oleks liiga palju keerulisi või pinnapealseid selgitusi ja seeläbi ka jälgida publiku reaktsioone [8]. Ettekanne peaks olema selge ning hästi struktureeritud. Selleks, et esitlus oleks mugav nii esinejale kui kuulajale, on soovituslik teha märkmeid, mida rääkida. Esitlus ei tohiks minna liiga pikaks ega jääda lühikeseks, tavaliselt on esitluse aeg ette antud ja selle raamis võiks ka püsida. Et ettekanne läheks ladusalt, võiks esineja harjutada ettekannet ka enne selle esitlemist. Oluline on ka valida õige hääletoon ja tugevus, et kuulajatel huvi püsiks. Hea on lisada esitlusele ka pilte või muud visuaalset sisu, aga siiski peab silmas pidama, et visuaalne pool ei tohi olla segadust tekitav või ebavajalik.

Ettekanne struktuurisoovitus esinejale [7]:

- Räägi publikule, mida sa hakkad neile rääkima (sissejuhatus);
- räägi publikule (esitluse tuum);
- Räägi publikule, mis sa neile rääkisid (kokkuvõte).

## 2.3 Projektijuhendi loomise põhimõtted

### 2.3.1 Teised projektijuhendid Tartu Ülikooli arvutiteaduste instituudis

Selleks, et uuendada juhendeid Tartu Ülikooli arvutiteaduste instituudi (ATI) kursustel tehtavate projektidega, kirjeldame Courses lehelt leitud ATI kursuseid ning nende projektijuhendeid.

Kursusel „Arvutimängude loomine ja disain“ MTAT.03.263<sup>10</sup> on projekti kohta Courses leheküljel, mis kajastab tingimusi ja reegleid. Sellel kursusel ei ole avalikult väljas kindlat

---

<sup>10</sup> <https://courses.cs.ut.ee/2024/gamedev/fall/Main/Projects>

hindamismaatriksit, kuid on seatud kindlad nõuded, mida projektirühm peab täitma. Projekt on jaotatud kolme osasse ning peale igat etappi peavad õpilased tegema oma progressist ülevaate. Kursusel „Objektorienteeritud programmeerimine“ LTAT.03.003<sup>11</sup> on rühmatöid kaks, teine võib olla esimese edasiarendus, aga võivad ka olla täiesti erinevad. Esimest ja teist rühmatööd võib teha ka erinevates rühmades. Courses leheküljel on kursusel toodud mõlemale projekti osale hulk kriteeriumeid, need defineerivad ka hindamise. Teise rühmatöö lõpus peab rühm tegema oma projektist ka esitluse ning seda praktikumis esitlema.

Programmeerimise kursusele sarnases sessioonõppe aines „Programmeerimine“ LTAT.SO.001<sup>12</sup> on antud nõuded, mis on samalaadsed LTAT.03.001 aine projekti nõuetega. Sessioonõppe aines tuleb lisaks projektile esitada ajatabel ja aruanne, kus õppija peab ise analüüsima, kas kõik nõuded said täidetud.

### 2.3.2 Hindamismaatriksi loomine

Hindamismaatrikseid on kahte tüüpi: analüütiline hindamismaatriks ja holistiline hindamismaatriks [3], holistilised hindamismaatriksid hindavad projekti kõiki elemente tervikuna, analüütiline hindamismaatriks hindab aga projekti kõiki kriteeriumeid eraldi. Paremaks arusaamiseks tuuakse allikas näide inimkehast - kui võtta inimese keha tervikuna, siis on tegemist holistilise variandiga, kui aga vaadata eraldi igat elundkonda, siis on tegemist analüütilise hindamismaatriksiga.

Hindamiskriteeriumi loomisel on ka omad ohud [3]: sõnastatakse mitteolulised tunnused, maatriks võib suunata õppijat täitma ainult maatriksis olevaid kriteeriumeid, hindamise ja tagasiside mitmekesisus võib väheneda.

Harilikult on hindamismaatriks jagatud neljaks osaks [9]:

- töö kirjeldus - ülesanne;
- kriteeriumite skaala - tihti kutsutakse ka kvaliteedi hinnanguks;
- dimensioonid - kutsutakse ka hindamiskriteeriumiteks, need kirjeldavad teadmisi või oskusi, mida ülesandes näitama peab;
- dimensioonide kirjeldused - kirjeldab igat hindamiskriteeriumi tulemuslikkuse taset.

---

<sup>11</sup> <https://courses.cs.ut.ee/2024/OOP/spring>

<sup>12</sup> [https://courses.cs.ut.ee/2024/prog\\_so/fall/Main/Projekt](https://courses.cs.ut.ee/2024/prog_so/fall/Main/Projekt)

Hindamismaatriksit tuleks jagada õpilastele koos ülesande kirjeldusega [9]. Nii saab õpilane paremini aru, mida temalt konkreetselt oodatakse.

### **2.3.3 Projektjuhendite loomine**

Juhendi tegemisel peab mõtlema, millele juhendit luuakse, kas rakendusele, ülesandele, masinale või töökohale [10]. Juhendi kirjutamisel tuleks kasutada lihtsat ja arusaadavat sõnastust. Laused võiksid olla lühikesed ja tabavad. Soovitatav on juhendis kasutada ka visuaalset materjali, nii saab lugeja parema vaate olukorrast. Instruktsioon peaks olema hästi struktureeritud, et lugeja oskaks hõlpsasti juhendis orienteeruda.

Juhendi loomisel peaks mõtlema järgmistele aspektidele [10]:

1. Milline on kasutajaskond? Millise täpsusega peaks juhend olema ning kui detailne?
2. Millist probleemi peaks juhend lahendama?
3. Juhend võiks olla sammudena - see aitab kasutajal paremini instruktsioone järgida.
4. Tuleb kaardistada kasutaja teekond - võiks uurida, kuidas kasutaja seda juhendit kasutaks ning aru saaks.
5. Tuleb kirjutada konkreetselt ja lühidalt - siis on kasutajal lihtsam juhendit järgida.
6. Kõiki kasutajaid tuleks võtta algajatena - ei saa eeldada, et kasutajad on samasuguse ning vastava taustaga.
7. Juhendit tuleks testida algajate peal.
8. Juhend peaks sisaldama ka näiteid.
9. Tundmatuid sõnu ja sümboleid tuleks selgitada võimalikult vara.

### 3. Metoodika

#### 3.1 Projektijuhendi ja materjalide loomine

Projektijuhendi ja materjalide loomine jagunes viieks osaks:

1. projekti etappide kirjeldused;
2. hindamismaatriksid;
3. vormistamisnõuded;
4. tööriistade juhendid;
5. Pythoni moodulite otsimine.

Iga osa juures olid olemas eelnevate aastate materjalid, mis vajasis täiendamist. Kõigepealt tuli analüüsida olemasolevaid materjale ning mõelda nende puudujääkidele. Projekti etappide kirjelduste parandamise juures tuli panna õpilasele täpsemini ja arusaadavamalt kirja, mida igal etapil esitama peab. Nagu eelnevalt mainitud ei olnud kursusel varem hindamismaatriksit projektide hindamiseks. Hindamismaatriksid valmisid juhendajaga koostöös ning pideva tagasisidega teistelt praktikumijuhendajatelt. Esimese hindamismaatriksi kogemuse põhjal valmis ka teine ning ka selle valmides saadeti see üle vaatamiseks teistele praktikumijuhendajatele, kes sellele siis oma tagasisidet andsid. Vormistamisnõudeid projektile samuti polnud, nende olemasolu teeb hindajale ja lugejale lihtsamaks koodist arusaamise ning selle jätkusuutlikkusele, kui peaks olema tahtmine veel loodud programmi edasi arendada. Tööriistade juhendite loomisel kasutati olemasolevat materjali, mida oli vaja kohandada vastavalt kursuse vajadustele. Pythoni moodulite näited olid varasemalt olemas, aga nende kohta polnud kirjeldusi ning osad olid ka aegunud. Kõik materjalid pandi ülesse pmWiki põhisesse Tartu Ülikooli arvutiteaduste instituudi Courses-keskkonda, aadressile: <https://courses.cs.ut.ee/2024/programmeerimine/fall/Main/projekt>. Sellel veebilehel olevaid materjale nägid ainult praktikumijuhendajad ning said ka anda sellele tagasisidet.

#### 3.2 Tagasisideküsimustik õpilastele

Selles lõputöös koostati tagasiside küsimustik õpilastele, et saada infot üliõpilaste kasutatud moodulite ja tööriistade kohta ning tagasisidet loodud materjalidele. Kõiki loodud materjale üliõpilased ei näinud, seega on tagasiside vaid projektietappide ja hindamismaatriksi kohta. Küsimustikku said üliõpilased täita keskkonnas Moodle, see oli vabatahtlik, aga selle täitmisel teenis juurde 0,4 punkti projekti kogupunktidest, mis on marginaalne osa kursuse kogupunktisummast 100.

Küsimustiku loomisel alustati küsimustest, mis aitaksid ka praktikumijuhendajaid. Küsimustiku teine pool koosnes suuresti üliõpilaste vajaduste kaardistamisest. Kuna küsimustiku loomise hetkel ei olnud veel kõik projektimaterjalid valmis ega üliõpilastele nähtavad, siis küsimused puudutasid pigem vajadusi.

Tagasisideküsimustik oli praktikumijuhendajatele nimeline, aga lõputöö kirjutaja jaoks anonüümne. Küsimustik loodi nii, et praktikumijuhendajatel oleks võimalus saada tagasisidet töö sujumise ja mahu kohta. Lõputöö tarbeks otsustati üliõpilaste eristamiseks kasutada ainult rühma tähist.

Üliõpilastelt tagasisideks saadud küsimustikus leidis ühele konkreetsele küsimusele vähemalt 5 valenegatiivset vastust. Neid vastuseid ei eemaldatud valimist, sest valenegatiivsed olid ainult ühe konkreetse küsimuse kohta, samade üliõpilaste teisi vastuseid lugedes tundusid need igati adekvaatsed ja asjalikud.

Tagasisideküsimustiku vastuseid analüüsiti tabelite ja graafikutega ning kõik kommentaarid said ka individuaalselt läbi loetud.

Tagasisideküsimustiku küsimused üliõpilastele on saadaval Lisa II all.

### **3.3 Tagasisideküsimustik praktikumijuhendajatele**

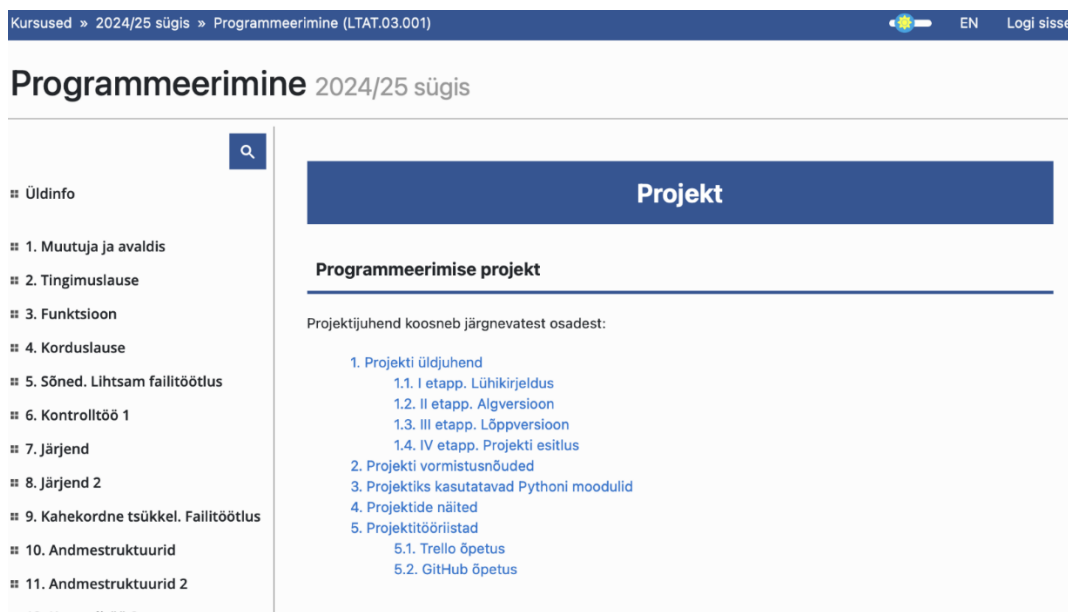
Tagasiside küsimustik juhendajatele sisaldas endas küsimusi nii nende materjalide kohta, mida üliõpilased nägid kui ka Courses lehe kohta. Tagasiside küsimustik loodi keskkonnas Google Forms, mis on loodud küsimustike koostamiseks. Tagasisideküsimustik koosnes viiest osast ning sisaldas endast 16 küsimust. Esimeses osas sooviti tagasisidet projekti lühikirjeldusele. Teises ja kolmandas osas tuli põhiline rõhk panna hindamismaatriksi tagasisidele. Hindamismaatriksi kohta tuli küsida palju infot, seega otsustati suure tabeli kasuks, kus igit hindamiskriteeriumit saab eraldi analüüsida.

Tagasiside küsimused praktikumijuhendajatele on leitavad Lisa I all.

## 4. Tulemused

### 4.1 Valminud projektimaterjalid

Töö peamiseks tulemuseks on valminud Courses lehekül<sup>10</sup> ning selles olevad projektimaterjalid. Terve lehekül on valminud kasutades sama kursuse, „Programmeerimine“ (LTAT.03.001), Courses lehe stiili ja vormistust.



Joonis 1. Kuvatõmmis valminud materjalide koondlehest Courses keskkonnas

Valminud projektijuhend koosneb viiest peamisest osast:

1. projekti üldjuhend;
2. projekti vormistamisnõuded;
3. projektiks kasutatavad Pythoni moodulid;
4. projektide näited;
5. projektitööriistad.

Projekti üldjuhendi alla kuuluvad neli alapeatükki, projekti lühikirjeldus, projekti algversioon, projekti lõppversioon ning esitlus, millest iga on projekti esitamise üks osa. Projekti üldjuhend on juhend, mis koondab endas kõike infot, mida üliõpilane peab projekti kohta teadma. Üldjuhendi juurest leiab ka viited nelja alapeatüki juurde. Projekti alg- ja lõppversiooni juures on ka välja toodud teooriale tuginedes loodud hindamismaatriksid, mis on saadaval bakalaureusetöö lisades. Hindamismaatriksid valmisid jooksvalt kursuse toimumise jooksul, kus neid ka kasutati.

Projekti vormistamisjuhendi all on üliõpilastele suunised, kuidas peaks oma koodi õigesti vormistama. Vormistamisjuhendi alla on kokku pandud nii koodi kommenteerimine kui ka muutujanime valimine.

Projektis kasutatavate Pythoni moodulite peatükk käsitleb endas võimalikke pistikprogramme, mida Python pakub. See leht on jagatud ka moodulite omaduste järgi, näiteks eraldi on andmetöötlus ja graafilised liidesed. Iga pistikprogrammi juures on väike kirjeldus, lingid dokumentatsioonile ja paarile juhendile, graafiliste kasutajaliideste juures ka pildid liidese väljanägemisest.

Projektide näidete alla on kogutud silma jäänud projekte eelmistest aastatest. Iga näite juures on pilt rakendusest või mängust ning lühike kirjeldus programmi tööst.

Projektitööriistade all on kasulikud keskkonnad, mida oma rühmaga projekti tarbeks kasutada saab, näiteks versioonihaldus, arendus ning märkmete ja tehtavate tegevuste kategoriseerimine. Iga tööriista juures on lühike kirjeldus selle kohta ning peale klikkides avaneb sisukas kasutusjuhend piltide ja näidetega. Trello ja Githubi juhendid olid enne olemas, aga kohati polnud piisavalt infot, neid täiendati nii piltide kui ka infoga. PyCharmi juhend on loodud eelmisel aastal bakalaureusetöö raames.

## 4.2 Üliõpilaste tagasiside projektidele

2024. aasta sügisel osales kursusel „Programmeerimine“ 405 üliõpilast, kellest tagasisideküsimustikule vastas 144. Kuna küsitluses ei olnud täpsustatud, et vastamine on individuaalne, siis ei saa eeldada, et kõik rühmaliikmed vastasid küsimustikule.

Tabel 1. Programmeerimiskeskkondade kasutus ja juhendite vajadus

Keskkond	Mitu üliõpilast kasutas?	Mitu üliõpilast sooviks juhendit?
Thonny	95 (65,97%)	22 (15,28%)
PyCharm	18 (12,5%)	55 (38,19%)
Visual Studio Code	48 (33,33%)	67 (46,52%)
muu	6 (4,17%)	-
Rohkem kui üks keskkond	23 (15,97%)	-

Tabel 1 näitab, kui palju üliõpilastest, milliseid programmeerimiskeskondi kasutas ning milliste keskkondade juhendite järgi on nõudlus. Tabelis on antud nii õpilaste arv, kui ka osakaal protsendina. Tabelist selgub, et kõige rohkem kasutasid üliõpilased oma projekti loomiseks programmeerimiskeskonda Thonny. Sooviti, et oleks rohkem juhendeid Visual Studio Code keskkonna kohta. Lisaks tabelis toodud keskkondadele kasutasid õpilased veel koodi kirjutamiseks järgnevaid keskkondi: Google Colab, Jupyter Notebook, Cursor ja Kate.

Tabel 2. Populaarseimad Pythoni moodulid ja nende lisamaterjalide vajadus

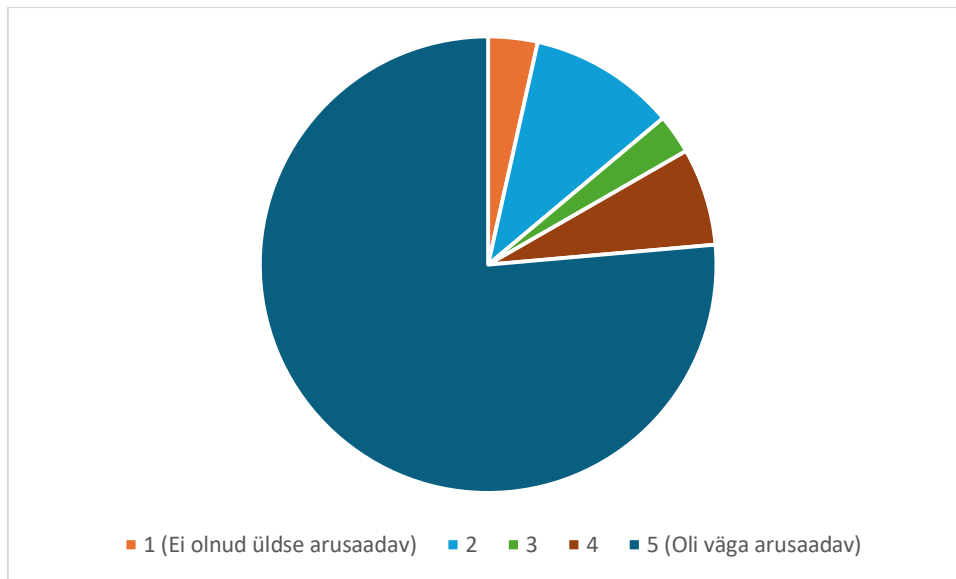
Moodul	Mitu üliõpilast kasutas?	Mitu üliõpilast sooviks juhendit?
Pandas (andmestikega töötamine)	14 (9,72%)	36 (25%)
Numpy (andmete töötlus)	7 (4,82%)	28 (19,44%)
Pygame (mängude loomine)	42 (29,17%)	54 (37,5%)
Tkinter (graafiline kasutajaliides)	70 (48,61%)	49 (34,03%)
Matplotlib (visualiseerimine)	15 (10,42%)	33 (22,92%)
Opencv (pilditöötlus)	2 (1,39%)	15 (10,42%)
Flask (veebirakenduse loomine)	3 (2,08%)	34 (13,61%)
Easygui (graafiline kasutajaliides)	2 (1,39%)	23 (15,97%)
muu: Selenium	7 (4,82%)	2 (1,39%)
muu: BeautifulSoup4 (bs4)	10 (6,94%)	2 (1,39%)
muu: Json	7 (4,82%)	0 (0%)

Tabel 2 näitab, milliseid lisamooduleid kasutasid üliõpilased oma projekti tarbeks ning millistele moodulitele soovisid üliõpilased juurde juhendeid. Tabelis on antud nii õpilaste arv, kui ka osakaal protsendina. Antud tabeli järgi oli kasutamiseks populaarseim pistikprogramm tkinter, mida kasutas oma projektis 70 vastanud üliõpilast. Lisamaterjale sooviti kõige rohkem pygame ja tkinter moodulite kohta. Etteantud pistikprogrammidele lisaks olid kõige silmapaistvamad üliõpilaste välja toodud moodulid: Selenium, mida kasutatakse veebist informatsiooni saamiseks, Beautifulsoup4, millel on eelnevaga sama eesmärk, ning Json, mida kasutatakse andmete transportimiseks. Lisaks tabelis loetletud moodulitele kasutasid õpilased veel järgnevaid mooduleid: Streamlit, Beeware, Customtkinter, Face-recognition, Librosa, Datetime, PySimpleGUI.

Tabel 3. Projekti tööriistade kasutamine ja nende kohta materjalide vajadus

Tööriist	Mitu üliõpilast kasutas?	Mitu õpilast soovis lisamaterjale?
Discord	52 (36,11%)	7 (4,82%)
Github	75 (52,08%)	67 (46,53%)
Trello	2 (1,39%)	25 (17,36%)
Zulip	4 (2,78%)	17 (11,81%)
Moodle'i vahendid	33 (22,92%)	9 (6,25%)
muu: Facebook Messenger	19 (13,19%)	0 (0%)
muu: Gitlab	7 (4,82%)	1 (0,69%)
muu: Google drive	3 (2,08%)	0 (0%)

Tabelist 3 näeb, milliseid projektitööriistu üliõpilased oma töös kasutasid ning milliste kohta oleks vaja juurde luua materjale. Tabelis on antud nii õpilaste arv, kui ka osakaal protsendina. Tabelist selgub, et populaarseimad tööriistad olid Discord ja Github. Üliõpilaste enda kasutatud tööriistadest oli kõrgeimal kohal messenger. GitHubile lisaks kasutati versioonihalduseks ka Gitlabi. Suurim vajadus oli Githubi juhendile, selle kohta kirjutasid paar üliõpilast kommentaarina juurde, et nad oleksid tahtnud seda kasutada, aga ei osanud. Lisaks tõid üliõpilased oma kasutuseelistuste seas välja Google Drive keskkonna ning päris mitu kasutas paarisprogrammeerimisvõtet.



Joonis 2. Projekti hindamismaatriksi arusaadavus (n=144)

Hindamismaatriksi arusaadavust sai hinnata skaalal ühest viieni, kus üks tähistas „ei olnud arusaadav“ ning viis tähistas „oli väga arusaadav“. Jooniselt on näha, et hindamismaatriks oli õpilastele pigem arusaadav. Kõik üliõpilased, kes vastasid „ei olnud arusaadav“ lisasid kommentaariks juurde, et kõik oli hästi ja midagi ei peaks täpsustama. Keskmine hinne hindamismaatriksile oli 4,42, kui eemaldada viis valenegatiivset vastust, siis on keskmine 4,54. Vastanute jäetud kommentaaridest selgus, et peamised murekohad hindamismaatriksil ja projekti kirjeldusel olid projekti maht ja keerukus. Välja toodi ka, et inimeste kogemus programmeerimisega on erinev, sellest tulenevalt ka töötempo ning kas sellest sõltub siis ka projekti keerukus. Natuke segaseks jäi ka, see et kuidas hinnatakse projekti kasulikkust, eriti kui projektiks loodi mäng.

### 4.3 Juhendajate tagasiside projektide läbiviimisele ja projekti-materjalidele

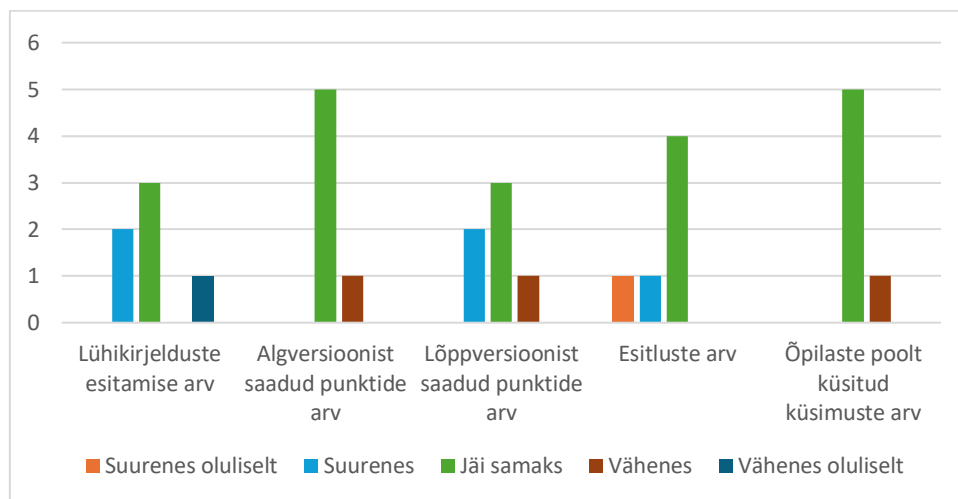
2024. aasta sügisel oli kursusel „Programmeerimine“ 10 praktikumijuhendajat, kellest tagasiside küsimustikule vastas 7.

Esimese etapi ehk projekti lühikirjelduse tööjuhend hinnati pigem sobivaks, kommentaaridena toodi välja, et juhend on lühike, võiks natuke pikemalt ülesannet lahti seletada ning kuskil võiks kirjas olla kui pikalt ja põhjalikult peab küsimusele vastama.

Teise etapi ehk projekti algversiooni esitamise juhend hinnati samuti sobivaks. Vastuste kohaselt sama etapi hindamismaatriks toetas projektide tegemist. Kommentaaridest selgus mitu näpunäidet kuidas hindamismaatriksit muuta, näiteks võiks vähem rõhku panna projekti mahule ja rohkem kasutajasõbralikkusele. Probleemiks osutus ka info puudumine selle kohta, kas kriteerium käib ainult algversiooni või kogu projekti kohta, näiteks projekti maht, kas see käib algversiooni mahu kohta, mis ilmselgelt on kõigil väga erinev või valmiva projekti oletatava mahu kohta.

Kolmas etapp - projekti lõppversiooni juhend sai pigem positiivse hinde. Täiendada saaks juhendit selgitustega, toodi välja murekoht, et pole aru saada, kas küsimusele peab vastama individuaalselt või terve rühma peale üks.

Neljanda etapi ehk esitluse juhend oli sobiv, kuid leidis kohti, mida paremaks muuta. kommentaaridest selgub, et jäi segaseks kas ülesanne on individuaalne või rühma peale.

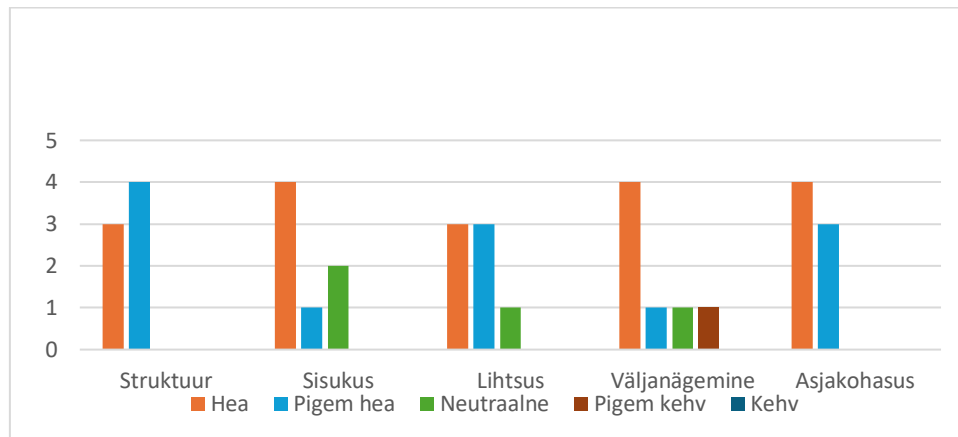


Joonis 3. Kursuse „Programmeerimine“ erinevate etappide esituste arv võrreldes varasemaga (n=6)

Antud joonise põhjal ei saa lõplikult otsustada, kas etappide kirjeldused aitasid kaasa erinevate esituste arvu suurenemise või vähenemisega. Kõige domineerivam on siiski vastus „jäi samaks“ ning 7 vastanu põhjal ei saa statistikat teha.

Praktikumijuhendajatelt küsiti ka lisajuhendite vajaduse kohta ning küsimustikust selgus, et programmeerimiskeskondadest võiks olla Visual Studio Code juhend ning PyCharm. Moodulite kohapealt oli kõige populaarsem Pygame. Pistikprogrammidest said häält veel Pandas, Tkinter, Easygui ja Flask, ehk peamiselt graafilised kasutajaliidesed. Projektitööriistade koha pealt arvati, et juhendit vajaks Github, kommentaarina toodi juurde, et

kõik õppekavad, kellele kursus „Programmeerimine“ kohustuslik on ei saa Githubi õpet oma „Sissejuhatus erialasse“ kursusel.



Joonis 4. Tagasiside Courses lehele (n=7)

Üldjoones sai projektimaterjal hea tagasiside. Praktikumijuhendajate poolt tuli ka mitmeid näpunäiteid kuidas veebilehte paremaks ja mugavamaks teha. Näiteks soovitati kuidas parandada kohe esimesena nähtavat materjalide järjekorda ning osadel alamlehtedel on liiga vähe infot. Anti ka sisendit, mis võiks Courses lehel lisaks olla ning kuidas eristada ülesandeid, mida peab tegema individuaalselt ning mida projektirühmaga koos. Üldiselt arvati, et loodud materjalid ning veebileht on üliõpilasele sellel kursusel igati abiks ja vajalik.

#### 4.4 Arutelu

Selle bakalaureusetöö eesmärk oli luua kursusele „Programmeerimine“ projektimaterjalid, hindamismaatriksid ning Courses leht, kus on kõik projektiga seonduv kokku koondatud.

Programmeerimise kursusele projektimaterjalide ning hindamismaatriksite loomine aitab õpilastel neile seatud ootusi ja kohustusi paremini mõista ning loodetavasti teeb kogu projekti loomise protsessi lihtsamaks. Projektimaterjalid aitavad üliõpilastel leida vajaminevat infot kasutada olevate keskkondade, moodulite ning tööriistade kohta ühest kohast. Samuti aitavad hindamismaatriksid Praktikumijuhendajatel üliõpilasi paremini ja ausamini hinnata.

Koostatud materjalide ja hindamismaatriksite tagasiside oli üldiselt positiivne, kuid paranduskohti jagub. Õpilastelt saadud tagasisidest selgus, et juhendeid soovitakse Visual Studio Code keskkonnale ja näiteks Zulipile.

Kui üliõpilased on päriselt saanud loodud projektimaterjale juba mõnda aega kasutada, siis ilmnevad kindlasti parandamist vajavad osad. Edasi saab arendada ka hindamismaatrikseid,

näiteks teise projekti hindamismatriksi üks kriteerium on esitluse eest saadavad punktid. Esitluse tegemise kohta on võimalik ka teha omaette hindamismatriks.

## 5. Kokkuvõte

Tartu Ülikooli kursus „Programmeerimine“ koosneb suures osas etteantud ülesannete lahendamisest. Selleks, et üliõpilased oleksid võimelised leidma lahendusi ka reaalelus ettetulevatele probleemidele, on programmeerimise kursusel üheks kandvaks osaks lisatud rühmaprojekt. Rühmaprojekti tegemisega võib kaasas käia palju raskusi, sest tihti pole üliõpilased varem tuttavad tarkvaraprojekti töövahenditega, millega iseseisvalt tutvumiseks kulub palju projektiaega ning motivatsiooni. Projekt kui rühmatöö aitab üliõpilastes arendada koostöövõimet, tehtu eest vastutamist, üksteisega arvestamist, enese ja teiste töö ning panuse hindamist. Bakalaureuse töö eesmärk oli luua kursusele „Programmeerimine“ LTAT.03.001 projektijuhend erinevatele projektietappidele ning hindamismaatriksid projektide tagasisidestamiseks ja hindamiseks. Selleks loodi keskkonda Courses projekti materjale kokkuvõttev lehekülge, mis sisaldab moodulite kirjeldusi, tööriistade juhendeid, vormistusnõudeid, eelnevate aastate projektinäiteid ning etappide juhendeid koos hindamismaatriksitega. Töö käigus analüüsiti olemasolevaid materjale ja etappide juhendeid; loodi uued projektimaterjalid, hindamismaatriksid ning etappide juhendid; küsiti projektimaterjalide kohta tagasisidet üliõpilastelt ning praktikumijuhendajatelt ning tehti materjalides võimalikud vastavad muudatused. Courses lehe juures said positiivset tagasisidet struktuur, asjakohasus ning sisukus.

Töö annab võimaluse edasi uurida üliõpilaste vajadusi projekti tegemisel ning seejärel Courses keskkonnas olevat lehekülge vastavalt täiendada.

## Viidatud kirjandus

- [1] R. T. Narayanan. Novice Programmer to New-Age Application Developer: What Makes Python their First Choice?, 2019. <https://ieeexplore.ieee.org/document/8944583>
- [2] G. V. Rossum. Computer Programming for Everybody, 1999. Kasutatud 8.12.2024. <https://www.python.org/doc/essays/cp4e/>
- [3] A. Niilo ja D. Lõvi. Tehnilised võimalused kriteeriumipõhiseks hindamiseks, 2023 [https://ut.ee/sites/default/files/2023-03/Tehnilised%20v%C3%B5imalused%20kriteeriumip%C3%B5hiseks%20hindamiseks\\_meistriklass%2022.02.23.pdf](https://ut.ee/sites/default/files/2023-03/Tehnilised%20v%C3%B5imalused%20kriteeriumip%C3%B5hiseks%20hindamiseks_meistriklass%2022.02.23.pdf)
- [4] Yale Poorvuu Center for Teaching and Learning, 2021. Kasutatud 7.01.2025. [https://poorvucenter.yale.edu/Team-Based-Learning#:~:text=Team-based%20learning%20\(TBL\),appealing%20when%20they%20are%20incorrect](https://poorvucenter.yale.edu/Team-Based-Learning#:~:text=Team-based%20learning%20(TBL),appealing%20when%20they%20are%20incorrect)
- [5] T. Kivirand, C. Šuman ja P. Nelis. Kaasav hariduskorraldus rühma- või klassiruumi praktikas. *Kaasav haridus*, 2024, lk 97-98.
- [6] R. Putter ja M. Lehner, Project Based Learning in Computer Science - A Review of More than 500 projects, 2011, <https://www.sciencedirect.com/science/article/pii/S1877042811028655>
- [7] P. D. Rodrigues, O. P. Boon ja A. R. M. Deni. Integrating Project Based Learning in Oral Presentation Assessments Learnt and Way Forward. *Malaysian Journal of ELT Research*, 2024, nr 21, lk 31-48.
- [8] P. Schulze ja L. Barton. *Council on Undergraduate Research* Kasutatud 7.04.2025. [https://www.cur.org/wpcontent/uploads/2023/08/How\\_to\\_Give\\_a\\_Good\\_Presentation\\_CUR.pdf](https://www.cur.org/wpcontent/uploads/2023/08/How_to_Give_a_Good_Presentation_CUR.pdf)

- [9] G. Burghart ja R. C. Panettieri. Faculty Guide to Rubrics, *Radiologic Technology*, 2009, lk 266-268.
- [10] J. Prabhakaran, (2025) *Document360: Ultimate Guide to Write Instruvtion for User Manual*. Kasutatud 8.12.2024. <https://document360.com/blog/creating-a-user-manual/>

## Lisad

### Lisa I. Küsimused juhendajatele

1. Ees- ja perekonnanimi (valikuline)

#### I etapp - teema valimine ja lühikirjeldus

Vaata projekti I etapi tööjuhendit siit:

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

2. Hinnake I etapi tööjuhendi sobivust

(1(Ei sobinud)-5 (Sobis hästi))

Kommentaar (nt mis osa tööjuhendist vajaks täiendamist?)

#### II etapp - projekti algversioon

Vaata II etapi tööjuhendit ja hindamismatriksit siit:

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

3. Hinnake II etapi tööjuhendi sobivust

(1(Ei sobinud)-5 (Sobis hästi))

Kommentaar (nt mis osa tööjuhendist vajaks täiendamist?)

4. Vaadake projekti algversiooni hindamismatriksit

<https://courses.cs.ut.ee/2024/programmeerimine/fall/Main/IIetapp> ja hinnake kriteeriumite ja nende kirjelduste sobivust:

(valikud: ei sobinud üldse, pigem ei sobinud, neutraalne, pigem sobis, sobis hästi)

1. Projekti kirjeldus ja kommentaar
2. Projekti programm on mõistlik/asjalik
3. Projekti maht on piisav
4. On aru saada, mida funktsioonid teevad sh vajadusel kommenteeritud

5. Kood on arusaadavalt kommenteeritud, alguses on lõik töö nime ja tegijate nimedega ning programmi käivitamisjuhend
6. Muutujanimed on mõistlikud ja muutujaid on mõistlikult kasutatud
7. Programmi struktuur on korras (impordid, funktsioonid, põhiprogramm hästi struktureeritud)

5. Kuivõrd toetas hindamismaatriks projektide tegemist?

(1 (Ei aidanud üldse) - 5 (Aitas hästi))

Kommentaar (nt mis osa hindamismaatriksis vajaks täiendamist?)

### **III etapp - projekti lõppversioon**

Vaata III etapi tööjuhendit ja hindamismaatriksit:

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

6. Hinnake III etapi tööjuhendi sobivust

(1 (Ei sobinud) - 5 (Sobis hästi))

Kommentaar (nt mis osa tööjuhendist vajaks täiendamist?)

7. Vaadake projekti algversiooni hindamismaatriksit

<https://courses.cs.ut.ee/2024/programmeerimine/fall/Main/IIIetapp> ja hinnake kriteeriumite ja nende kirjelduste sobivust:

(valikud: ei sobinud üldse, pigem ei sobinud, neutraalne, pigem sobis, sobis hästi)

1. Projekti kirjeldus ja kommentaar
2. Projekti programm on mõistlik/asjalik
3. Projekti maht on piisav
4. On aru saada, mida funktsioonid teevad sh vajadusel kommenteeritud
5. Kood on arusaadavalt kommenteeritud, alguses on lõik töö nime ja tegijate nimedega ning programmi käivitamisjuhend

6. Muutujanimed on mõistlikud ja muutujaid on mõistlikult kasutatud
7. Programmi struktuur on korras (impordid, funktsioonid, põhiprogramm hästi struktureeritud)
8. Rühmatöö küsimustikule vastamine
9. Projekti esitlus
10. On hinnatud kaastudengite esitlusi

8. Kuivõrd toetas hindamismaatriks projektide tegemist?

(1 (Ei aidanud üldse) - 5 (Aitas hästi))

Kommentaar (nt mis osa hindamismaatriksis vajaks täiendamist?)

#### **IV etapp - esitlus**

Vaata IV etapi esitluste tööjuhendit:

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

9. Hinnake IV etapi tööjuhendi sobivust

(1 (Ei sobinud) - 5 (Sobis hästi))

Kommentaar (nt mis osa tööjuhendist vajaks täiendamist?)

10. Kui andsid Programmeerimise praktikume ka varem, siis hinda kas ja kuidas muutus järgnevate esitluste arv võrreldes varasemaga:

(valikud: suurenes oluliselt, suurenes, jäi samaks, vähenes, vähenes oluliselt)

1. Lühikirjelduste esitamiste arv
2. Algversioonist saadud punktide arv
3. Lõppversioonist saadud punktide arv
4. Esitluste arv
5. Õpilaste poolt küsitud küsimuste arv

**Kuna projektis on võimalik avastada erinevaid mooduleid ja projektitööd hõlbustavaid vahendeid, siis järgnevalt on küsimused projekti toetavate materjalide kohta**

Projektitööd toetavad materjalid on koondatud siia:

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

11. Milliste programmeerimiskeskondade kohta võiks olla projektimaterjalides rohkem juhendeid?

12. Milliste Pythoni moodulite kohta võiks olla rohkem juhendeid?

13. Milliste projekti tööriistade kohta võiks olla rohkem juhendeid?

14. Kogu projekti materjal on saadaval siin:

<https://courses.cs.ut.ee/2024/programmeerimine/fall/Main/Projekt>. Hinda järgnevaid

Courses'i lehe aspekte:

(valikud: hea, pigem hea, neutraalne, pigem kehv, kehv)

1. Struktuur
2. Sisukus
3. Lihtsus
4. Väljanägemine
5. Asjakohasus

15. Mis võiks Courses lehel veel olla? / Mida saaks parandada?

16. Praegu on esitluse punktid samas hindamismaatriksis lõppversiooni punktidega. Kas esitluse etapi jaoks võiks olla eraldi hindamismaatriks või sobib praegune versioon, kus lõppversiooni ja esitluse punktid on ühes hindamismaatriksis?

Vajadusel kommenteeri vastust

Lisakommentaariid projektimaterjalide kohta

## **Lisa II. Küsitlus üliõpilastele**

### **Projekti tagasiside küsimused**

1. Nimi (Moodle'is pole vaja)
2. Projekti lühikirjeldus
3. Kuidas projekti tegemisel koostöö sujus? Kuidas olid rollid jaotatud?
4. Kui palju aega kulus kokku projekti tegemisele inimese kohta (nõutud 15,6 tundi inimese kohta)?
5. Mis õnnestus projekti tegemisel?
6. Mis nii hästi ei õnnestunud?
7. Kust said abi projekti koostamisel?
8. Millises keskkonnas kirjutasite koodi?
  - a) Thonny
  - b) PyCharm
  - c) Visual Studio Code
  - d) Muu ...
9. Millise programmeerimise keskkonna kasutamise kohta sooviksite rohkem juhendeid?
  - a) Thonny
  - b) PyCharm
  - c) Visual Studio Code
  - d) Muu ...
10. Programmide kirjutamisel võis kasutada erinevaid lisamooduleid. Milliseid mooduleid kasutasite projektis?
  - a) pandas
  - b) numpy
  - c) matplotlib
  - d) tkinter
  - e) easygui
  - f) opencv - pilditöötlus;
  - g) pygame - mängude loomine;
  - h) flask - veebirakenduste loomine.
  - i) muu ...

11. Milliste moodulite kohta sooviksite rohkem juhendeid?

- a) pandas
- b) numpy
- c) matplotlib
- d) tkinter
- e) easygui
- f) opencv - pilditöötlus;
- g) pygame - mängude loomine;
- h) flask - veebirakenduste loomine.
- i) muu ...

12. Milliseid projektitööriistu kasutasite projekti koostamisel?

- a) Github
- b) Trello
- c) Zulip
- d) Discord
- e) Moodle'i vahendid
- f) muu ...

13. Milliste tööriistade kasutamise kohta sooviksite rohkem juhendeid?

- a) Github
- b) Trello
- c) Zulip
- d) Discord
- e) Moodle'i vahendid
- f) muu ...

14. Hinnake projekti hindamismaatriksi arusaadavust (1 (oli väga arusaadav)- 5 (ei olnud arusaadav))

15. Mis osa hindamismaatriksis vajaks täpsustamist?

16. Lisakommentaariid

## Litsents

Mina, Malin Püttsepp,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Projektijuhendi koostamine kursusele „Programmeerimine“ LTAT.03.001, mille juhendaja on Tauno Palts,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

1. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commonsi 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;
2. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
3. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Malin Püttsepp

**07.05.2025**