

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Maike Kadaja**  
**Vestlusagendile ProgVestlus andmestiku loomine**  
**aines “Programmeerimine”**  
**Bakalaureusetöö (9 EAP)**

Juhendajad: Tauno Palts, PhD  
Reimo Palm, PhD

Tartu 2023

## **Vestlusagendile ProgVestlus andmestiku loomine aines “Programmeerimine”**

### **Lühikokkuvõte:**

Tartu Ülikoolis on kursustel üliõpilaste kodutööde kohastele korduma kippuvatele küsimustele vastamiseks kasutusel murelahendajad. Murelahendajate loomisprotsess on aga tülikas ja nende alternatiivina tekkis kursuse “Programmeerimine” eestvedajatel idee rakendada tehisintellekti. Tehisintellekt on hariduses kasutusel juba 1970. aastatest ja on olnud just viimastel aastatel suurema tähelepanu all. Tehisintellekti üks populaarsemaid tüüpe on vestlusrobot. Käesoleva bakalaureusetöö eesmärk oli luua andmestik uuele vestlusagendile ProgVestlus, et seda saaks kasutada üliõpilaste abistamisel Tartu Ülikooli kursusel “Programmeerimine”. Selleks analüüsiti kursuse “Programmeerimine” kodutöid ja õppematerjale ning loodi 13 nädala kodutöödele potentsiaalselt tekkivate küsimustele vastused, vihjed ning terminite selgitused. Lisaks kirjeldati kuulsamaid ja aktuaalsemaid vestlusroboteid ning vestlusrobotite klassifitseerimist ja disainitehnikaid.

**Võtmesõnad:** vestlusrobot, programmeerimine, algõpetus

**CERCS:** P175 Informaatika, süsteemiteooria, P176 Tehisintellekt

## **Creating a dataset for the chat agent ProgVestlus in the course "Programming"**

### **Abstract:**

At the University of Tartu, troubleshooters are used in courses to answer frequently asked questions related to students' homework. However, the process of creating troubleshooters is troublesome, and as an alternative, the leaders of the "Programming" course came up with the idea of using artificial intelligence. Artificial intelligence has been used in education since the 1970s and has been under greater attention in recent years. One of the most popular types of artificial intelligence is the chatbot. The purpose of this bachelor thesis was to create a database for the new chat agent ProgVestlus, so that it could be used to help students in the "Programming" course. For this purpose, the homework and study materials of the course "Programming" were analyzed and answers, hints and explanations of terms were created for the questions that could potentially arise for the 13 weeks of homework. In addition, the most famous and current chatbots and their classification and design techniques were described.

**Keywords:** chatbot, programming, teaching basics

**CERCS:** P175 Informatics, systems theory, P176 Artificial intelligence

# Sisukord

Sissejuhatus.....	4
1. Vestlusrobotid.....	5
1.1 Tehisintellekt ja vestlusrobot.....	5
1.2 Vestlusrobotite klassifitseerimine.....	6
1.3 Vestlusrobotite disainitehnikad.....	7
1.4 Kuulsamate vestlusrobotite kirjeldus.....	8
1.4.1 ELIZA.....	9
1.4.2 A.L.I.C.E.....	9
1.4.3 Eugene Goostman ja Turingi test.....	10
1.4.4 Siri.....	10
1.4.5 ChatGPT.....	11
1.4.6 Bing Chat.....	12
2. Kursus “Programmeerimine (LTAT.03.001)”.....	14
2.1 Kursuse kirjeldus.....	14
2.2 Kursuse korraldus.....	15
2.3 Kursuse murelahendajad.....	16
3. Metoodika.....	18
3.1 Keskkond ProgVestlus.....	18
3.2 ProgVestluse tehniline kirjeldus.....	19
3.3 Tööprotsessi kirjeldus.....	22
3.3.1 Programmeerimise aine koduülesanded.....	22
3.3.2 Logide analüüs.....	23
3.3.3 Koduülesannete analüüs ja vastuste loomine.....	25
3.3.4 Ülesannete analüüsimine ChatGPT’ga.....	27
3.4 Üliõpilaste tagasisideküsitlus.....	28
4. Tulemused ja järeldused.....	29
4.1 Tagasisideküsitluse tulemused.....	29
4.2 Järeldused ja arutelu.....	29
4.3 Järgmised võimalikud sammud.....	31
5. Kokkuvõte.....	34
6. Viidatud kirjandus.....	35
Lisad.....	40
I Link vestlusagendi algandmestikule.....	40
II Võrdlustabel kodutööde küsimustest.....	41
III ChatGPT lahenduse ja näidislahenduse võrdlus.....	43
IV Tagasisideküsitlus.....	44
V Tulemuste tabel.....	46
VI Juhend ProgVestlusesse abimaterjali lisamiseks.....	47
V Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks.....	49

# Sissejuhatus

Viimaste aastate jooksul on programmeerimise õppimine üha populaarsem. Oskus programmeerida pakub tööturul laialdaselt võimalusi, arendab loogilist mõtlemist ja probleemide lahendamise võimeid. Siiski on paljude inimeste jaoks programmikoodi kirjutamine keeruline. Üheks peamiseks programmeerimise kursustelt väljakukkumise põhjuseks on see, et materjalist ei saada aru (Kinnunen, 2009). Motivatsiooni säilitamiseks on oluline, et üliõpilane saaks ülesannete lahendamisel võimalikult palju kergesti ligipääsetavat ja adekvaatset abi. Sageli pöörduakse abi saamiseks õppejõudude poole. Õppejõudude korduvatele küsimustele vastamiseks kuluva aja ja energia kokkuhoiuks kasutatakse Tartu Ülikoolis murelahendajaid. Tehisintellekti on aga hariduses kasutatud juba 1970. aastatest (Kay, 2015), see on praegu üldsuse suure tähelepanu all ja omaks üliõpilaste abistamisel murelahendajate ees mitmeid eeliseid. Üks kõige populaarsemaid tehisintellekti tüüpe, mida õpetamise ja õppimise juures kasutatakse on vestlusrobot (Okonkwo ja Ede-Ibijola, 2020). Vestlusrobotid on arvutiprogrammid, mis on disainitud inimestega vestluse simuleerimiseks. Need pakuvad fokuseeritud ja eesmärgile suunatud veebipõhist õppekeskkonda ja juturoboti mugavus ja interaktiivsus motiveerib õpilasi teadmisi omandama (Hanh, Van Hung & Thanh, 2022).

Kursuse “Programmeerimine” eestvedajatel tuli 2021. aastal idee võtta kursusel kasutusele oma vestlusrobot nimega ProgVestlus. Süsteemi tehniline ülesehitus sai valmis aastal 2022. Selle töö eesmärk on, sarnaselt murelahendajate loomisele, analüüsida aine “Programmeerimine” kodutöid ja luua terviklik andmestik vestlusrobotile ProgVestlus. Töö tulemusel on ProgVestlus võimeline andma kasutajale vihjeid mainitud kursuse kodutööde lahenduste kohta ning üldiseid mõistete ning terminite selgitusi, mida kursusel käsitletakse. Osaline vihjetesüsteem toetab õppimisprotsessi ja jätab üliõpilasele võimaluse ülesandeid ise lahendada. ProgVestlus säilitab eestikeelseid programmeerimise väljendeid ning annab üliõpilastele vastuseid, mis on käsitletavale teemale vastavad.

Siin töös antakse ülevaade tehisintellekti ja vestlusroboti olemusest, selgitatakse nende klassifitseerimist ja disainitehnikaid ning kirjeldatakse kuulsamaid vestlusroboteid. Teises peatükis kirjeldatakse kursust “Programmeerimine” ja praegu kasutusel olevaid lahendusi üliõpilaste abistamiseks. Sellele järgnevalt selgitatakse ProgVestluse keskkonna loogikat, töö metoodikat, analüüsiprotsessi ja tulemusi.

# 1. Vestlusrobotid

Selles peatükis defineeritakse tehisintellekti ja vestlusroboti mõisted ja olemus. Kirjeldatakse, kuidas vestlusagente klassifitseeritakse ja milliseid tehnikaid nende disainimisel kasutatakse. Lisaks antakse lugejale kronoloogiline ülevaade vestlusrobotitest, mis on oma valdkonnas märgilise tähtsusega või viimasel ajal rohkem kõneainet pakkunud.

## 1.1 Tehisintellekt ja vestlusrobot

Tehisintellekt (ingl *artificial intelligence*, lüh AI) on viimasel ajal palju kõneainet pakkunud inimintellekti simulatsioon, mis täidab mitmesuguseid ülesandeid – suudab pidada vestlusi, anda faktipõhiseid või ka loomingulisemaid vastuseid ja kirjutada erinevates programmeerimiskeeltes koodijuppe.

Burnsi, Laskowski, Tucci ja Lawtoni (2023) järgi keskendub tehisintellekti programmeerimine kolmele kognitiivsele oskusele: programmi võime õppida, arutleda ja end parandada. Õppeprotsessi peamine osa ehk programmi võime õppida toetub andmete hankimisele ja reeglite loomisele ning lisaks keskendutakse sellele, kuidas muuta kogutud andmed kasutatavaks teabeks. Siinkohal on abiks algoritmid ehk reeglid, mis annavad arvutitele (sh tehisintellektile) samm-sammult juhiseid konkreetsete ülesannete täitmiseks. Arutlemisostkust toetavad protsessid aga keskenduvad soovitud tulemuse saavutamiseks õige algoritmi valimisele. Eneseparandusprotsessid on loodud algoritmide pidevaks peenhäälestamiseks ja tagamaks, et need annavad võimalikult täpseid tulemusi. Tehisintellekte treenitakse üldjuhul suurte andmehulkade läbitöötlemise ja analüüsimise kaudu, et leida võimalikult palju mustreid ja korrelatsioone. Saadud tulemusi võtab tehisintellekt arvesse oma järgmiste otsuste ja ennustuste loomisel.

Tehisintellekti tüüpiliseks näiteks on vestlusrobot (ingl *chatbot*), mis on üks kõige elementaarsemaid ja laialdasemalt kasutatavaid näiteid intelligentsest inimese ja arvuti interaktsioonist (ingl *human-computer integration*, lüh *HCI*) (Bansal & Khan, 2018). Vestlusroboti puhul on tegemist arvutiprogrammiga, millega saab suhelda nagu inimesega, nii kirjalikult kui ka häälega. Lisaks mõistab vestlusrobot loomuliku keeletöötlust (ingl *natural language processing*, lüh *NLP*) abil üht või mitut inimkeelt (Khanna et al. 2015). Oxfordi sõnastik määratleb vestlusrobotit kui “arvutiprogrammi, mis on loodud inimese kasutajaga

vestluse simuleerimiseks, tavaliselt interneti kaudu"<sup>1</sup>. Vestlusroboteid nimetatakse selles töös ka juturobotiteks ja vestlusagentideks. Eelnevalt juba mainitud treeningmeetodiga saavad vestlusrobotid, keda on õigete näidetega treenitud, luua inimestega elutruid vestlusi. Samuti võib pildituvastustööriist, miljoneid näiteid läbi vaadanuna, õppida piltidel olevaid objekte tuvastama ja kirjeldama (Burns et al., 2023).

## 1.2 Vestlusrobotite klassifitseerimine

Hussaini, Ameri Sianaki ja Ababnehi (2019) sõnul on viimase paari aasta jooksul vestlusrobotite valdkond uute tehnoloogiate saabumisega muutunud dünaamiliseks, mistõttu lähtuvalt vestlusrobotite kasutusulatusest on nende täpne klassifitseerimine muutunud subjektiivseks. Vestlusroboteid võib klassifitseerida erinevate parameetrite järgi. Adamopoulou ja Moussiades (2020) on parameetrite kohta teinud ülevaatliku nimekirja (joonis 1), mis sisaldab järgmisi teemasid:

- 1) teadmiste valdkond,
- 2) pakutav teenus,
- 3) vastuse genereerimismeetod,
- 4) inimabi,
- 5) load,
- 6) interaktsioonirežiim/kommunikatsioonikanal,
- 7) kasutusviisid/eesmärgid.

Need parameetrid võivad sisaldada vestlusrobotite põhilist disainifilosoofiat, konteksti hulka, mida tuleb vestluse mõistmiseks salvestada ja arvesse võtta, või hoopis seda, millist tüüpi ja millise eesmärgiga vestluse jaoks vestlusagent luuakse (Ramesh jt, 2017).

---

<sup>1</sup> Oxfordi sõnastik chatbot - <https://www.oed.com/view/Entry/88357851?redirectedFrom=chatbot>

<b>Chatbot Categories</b>	Knowledge domain	Generic
		Open Domain
		Closed Domain
	Service provided	Interpersonal
		Intrapersonal
		Inter-agent
	Goals	Informative
		Chat based/Conversational
		Task based
	Response Generation Method	Rule based
		Retrieval based
		Generative
	Human-aid	Human-mediated
		Autonomous
	Permissions	Open-source
		Commercial
	Communication channel	Text
		Voice
		Image

Joonis 1. Näide vestlusrobotite klassifitseerimisest ja nimetamisest (Adamopoulou & Moussiades, 2020)

Mõningaid parameetreid nimetatakse inglise keeles mitmel viisil. Seepärast leiab ka uuringut tehes erinevaid vasteid ja allikaid. Samuti on olemas lisaks eelmainitutele veel parameetreid, mida käesolevas bakalaureusetöös ei kirjeldata. Hussaini et al. (2019) sõnul võib laias laastus vestlusroboteid jaotada ülesandepõhisteks ja mitte-ülesandepõhisteks. Esimesed on loodud pidama lühikesi vestlusi ja on ülesande- ning valdkonnaspetsiifilised. Teised suudavad simuleerida pikemat vestlust inimesega ja vestelda meelelahutuslikel eesmärkidel.

### 1.3 Vestlusrobotite disainitehnikad

Siin peatükis kirjeldatakse lähemalt vestlusrobotite disainiliike ja mõnda levinumat tehnikat, mida vestlusagendi loomisel rakendatakse ja siin bakalaureusetöös rohkem mainitakse. Vijaya Saradhi, Swajan Reddy, Arun Reddy, Saikumar ja Adityan (2022) kirjutavad, et

vestlusrobotite disainitehnikad jagunevad kolme kategooriasse: reeglipõhised, otsingupõhised ja generatiivsed. Nende järgi võib kirjeldada kategooriaid järgmiselt. Reeglipõhistel vestlusrobotitel on vastused arendajate poolt eelnevalt defineeritud. Otsingupõhised kopeerivad inimese vastuse eelmisest vestlusest või kasutavad närvivõrkude süsteeme (Hussain et al., 2019) ja generatiivsed vestlusrobotid genereerivad uusi täpsemaid vastuseid, mitte ei kasuta neid vastuseid, mis oleks eelnevalt defineeritud. Seega võib vestlusroboteid jagada kaheks ka selle põhjal, kas nad kasutavad tehisintellekti ja närvivõrke või nende vastused on eelnevalt paika pandud.

Iga kategooria puhul võidakse rakendada erinevaid tehnikaid. Mõned näited enimlevinumatest ja töös mainitustest on näiteks:

- 1) parsimine (ingl *parsing*) - muudab teksti kujul antud sisendi vähemkeeruliste sõnade hulgaks ja eraldab sealt tähendusrikka informatsiooni,
- 2) mustri sobitamine (ingl *pattern matching*) - leiab sisendist eelnevalt käsiti defineeritud mustreid ja tagastab sellele vastavas mallis hoitud vastuse,
- 3) tehisintellekti märgistuskeel AIML (ingl *Artificial Intelligence Markup Language*) - koosneb AIMLi elementidest ja need omakorda koosnevad teemadest ja kategooriatest, milles sisalduvad mustrid ja mallid ning reeglid nende omavaheliseks sobitamiseks,
- 4) loomuliku keele töötlemine (ingl *natural language processing* lüh *NLP*) - võimaldab masinatel tõlgendada inimkeelt, analüüsides süntaksit, semantikat, pragmaatikat jpm,
- 5) lausearvutus (ingl *propositional calculus*) - loogikavaldkond, mida näiteks parsimisel rakendada. Lausearvutust kasutatakse vestlusagendis Progvestlus.

Töö kolmandas peatükis räägitakse lausearvutusest täpsemalt. Järgnevalt kirjeldatakse lähemalt mõnd tuntumat ja päevakajalisemat vestlusrobotit.

## 1.4 Kuulsamate vestlusrobotite kirjeldus

1960. aastatest, kui loodi esimene juturobot, on juturobotite tehnoloogia pidevalt arenenud. Alates lihtsatest reeglipõhistest programmidest kuni keerukate tehisintellektidega virtuaalsete assistentideni on juturobotid saanud inimeste igapäevaelu lahutamatuks osaks. Neid kasutatakse laialdaselt erinevates eluvaldkondades, sealhulgas klienditeeninduses, tervishoius, finantsvaldkonnas ja hariduses. Aja jooksul on mitmed juturobotid omandanud suure



populaarsuse ning muutunud kodumajapidamiste osaks. Selles peatükis on kirjeldatud kuulsamaid ja praegu (2023. a) aktuaalsemaid juturoboteid, nende tausta ja mõju vestlusrobotite valdkonnale.

### 1.4.1 ELIZA

Esimene vestlusrobot avaldati Massachusettsi Tehnoloogiainstituudis professor Joseph Weizenbaumi poolt 1966. aastal keskel. Weizenbaum (Weizenbaum, 1966) kirjutas, et see kasutas märksõnade sobitamise meetodit. ELIZA otsis kasutaja sisendist kindlaid märksõnu ja kasutas sisseehitatud (skriptides<sup>2</sup> kirjeldatud) reegleid, et vastus genereerida. Ta ütleb, et ELIZA kasutas skripti, mis jäljendas psühhoterapeuti (joonis 2) ja ELIZA skriptid ei olnud osa programmist endast, vaid olid kasutusel andmetena. Weizenbaum uskus, et tehisintellekt ei tohiks asendada inimesi ametikohtadel, kus on vaja hoolivust ja lugupidavust (Kinos, 2019). Weizenbaumi juurdles küsimuse üle, kas inimõte on “täielikult arvutatav” (redutseeritav loogiliseks formalismiks). Weizenbaumi sõnul pidavat masinintelligentsi olemus ja loogika sotsiaalses maailmas hakkama toimima kui “aeglase toimega mürk” (Basset, 2019).

```
> Hello, I am Eliza. I'll be your therapist today.  
* Men are all alike  
> What does the similarity suggest to you?  
* They're always bugging us in some way or other.  
> Can you think of a specific example?  
* Well, my boyfriend made me come here.  
> I see. Let's try another topic and we will come back to that issue later.  
* He says I'm depressed much of the time.  
> Did you come to me because you are depressed much of the time?
```

Joonis 2. Näide ELIZAst. (ELIZA teksti ees “>”)

### 1.4.2 A.L.I.C.E.

Alice või Alicebot või A.L.I.C.E. ehk kunstlik lingvistiline interneti-arvuti üksus (ingl *Artificial Linguistic Internet Computer Entity*) on 1995. aastal dr Richard Wallace'i poolt loodud vestlusrobot keeles AIML (Wallace, 2009). AIML on märgendamiskeeke XML alaliik vestlusrobotite loomiseks. Wallace kirjutab (Wallace, 2009), et Alice kasutab AIML-failide

---

<sup>2</sup> Skript - spetsifikatsioon, mis määrab, kuidas kasutajale objekte esitada ja kuidas kasutajalt saadud sisendandmeid töödelda.

komplekti, mis koosneb lihtsatest stiimuli-vastuse moodulitest, mida nimetatakse kategooriateks. Iga kategooria sisaldab mustrit (stiimulit) ja malli (vastus). AIML-tarkvara salvestab need kategooriad puusse, mida haldab objekt nimega Graphmaster. Kui robot saab stiimulina sisendteksti, otsib Graphmaster sobivat mustrit ja väljastab vastusena seotud malli. A.L.I.C.E. oli esimene vestlusagent, mis võitis iga-aastasel Turingi testi võistlusel nimega Loebner Prize kõige inimesesarnasema arvuti auhinna. Kokku on Alice mainitud tiitli pälvinud kolmel korral (BBC). Alice'i peetakse ELIZA laienduseks, aga Alice'i andmestik on tunduvalt suurem (40 000 ja ELIZA'l 200 teadmiskategooriat) (Wallace, 2009).

### 1.4.3 Eugene Goostman ja Turingi test

1950. aastal püstitas Alan Turing küsimuse, et kas arvutiprogramm suudaks rääkida inimestega, ilma et viimased mõistaks, et nende vestluskaaslane on kunstlik - see küsimus kannab nime Turingi test (Oppy & Dowe, 2021). Esimene vestlusrobot, keda peetakse Turingi testi läbinuks, kannab nime Eugene Goostman. Eugene imiteeris 13-aastast ukraina poissi ja loodi aastal 2001 Venemaal, Peterburis, Vladimir Veselovi, Eugene Demchenko ja Sergey Ulaseni poolt, kellest teine on sündinud Ukrainas (Schofield, 2014). Eugene puhul kasutati mustri sobitamist ja (dünaamilist) tokeniseerimist (ingl *dynamic tokenization*) (Masche & Le, 2018). Aastal 2014 viidi Londoni Kuninglikus Seltsis Eugene'i peal läbi Turingi test (Ashford, 2014). Copeland (2014) kirjutab Huffington Postis, et testi läbimise lävendiks lugesid teadlased Huma Shah ja Kevin Warwick seda, kui žürii pidas robotit inimeseks vähemalt 30% juhtudest 5-minutiliste vestluste seeriates. Goostman suutis 33% eduga oma vestluskaaslastest uskuma panna, et ta tõepoolest on päris inimene ja seepärast loetakse teda Turingi testi läbinuks. Copeland toob välja, et tegelikkuses oli see aga Turingi ennustus, et 2000. aastaks suudab vestlusrobot 30% vestluskaaslastest ära petta, mitte testi läbimise piir. Shah ja Warwick aga mõistsid ennustust kui lävendit, mistõttu saab vaielda Eugene'i testiläbimise üle.

### 1.4.4 Siri

Siri avaldati 2010. aastal. Alguses oli ta iseseisev iPhone'i rakendus, mille lõi 24-liikmeline idufirma (Bosker, 2017). 2010. aasta aprillis ostis Apple Inc. Siri ära rohkem kui 200 miljoni dollari eest (Clifford, 2017). Bianca Bosker kirjeldab oma artiklis (Bosker, 2017) Sirit

täpsemalt. Siri loojad Dag Kittlaus (Siri kaasasutaja ja tegevdirektor) ja Harry Sessler (disainiekspert) olid Sirile loonud eraldi isiksuse ja iseloomu - ta vastas küsimustele vaimukalt ja mõnikord ka mõnitavalt. Apple muutis Sirit viisakamaks, vähendas ta temperamenti ja andis talle naishääle ning võimekuse suhelda erinevates keeltes. Oma ilmunisaastal 2010 suutis Siri ühenduda 42 erineva veebiteenusega ja tagastada ühe vastuse, võtnuna arvesse allikates pakutud informatsiooni. Siri kasutab peamiselt kahte tehnoloogiat: kõnetuvastust ja loomuliku keele töötlemist. Kasutaja päringute tõlgendamiseks ja vajaliku informatsiooni hankimiseks teeb Siri kasutaja telefoni protsessori koormamise asemel enamuse tööst pilves asuvate serveritega suheldes (Johnson, 2013).

Sarnaselt Apple'i Sirile on Google loonud Google Assistanti<sup>3</sup>, Amazon Alexa<sup>4</sup> ja Microsoft Cortana<sup>5</sup> isiklikud assistendid.

### 1.4.5 ChatGPT

Järgnev lõik tugineb enamjaolt Will Douglas Heaveni artiklile (2023) MIT tehnoloogia-ajakirjas MIT Technology Review. ChatGPT avaldati 2022. aasta novembri lõpus San Francisco tehisintellektiettevõtte OpenAI poolt. ChatGPT treenimisel on kasutatud tehnikat nimega stiimulõpe inimtagasisidest ehk RLHF (ingl *reinforcement learning from human feedback*). Selles õpetatakse keelemudelit kõikide vastuste või suvalise vastuse asemel tagastama vastus, mis inimkasutajale meeldiks. Inimeste eelistused on välja selgitatud päris inimestega tehtud testis, mille tulemusi kasutati treenimisel. Selleks, et kasutajad ei saaks ChatGPT'd halvasti käituma panna (ingl *jailbreak*), kasutavad OpenAI teadlased ka tehnikat nimega võistlev treenimine (ingl *adversarial training*) (Zhao, Alwidian & Mahmoud, 2022). Edukad rünnakud lisatakse treeningandmestikku, et taolisi tulevikus vältida. ChatGPT mudel on algselt treenitud kasutades andmestikku nimega WebText, mis sisaldab üle 45 terabaidi internetist saadud teksti (How to use ChatGPT, 2023). ChatGPT loojad on üllatunud tema edukuses ja selles, kui palju teda on kasutama hakatud. Üheks üllatuse põhjustajaks on tõsiasi, et enamus kasutatavast tehnoloogiast ei ole uus, vaid GPT-3.5 viimistletud versioon. ChatGPT'le eelnesid GPT-3, GPT-3.5 ja InstructGPT. GPT-3.5 eelkäija oli 2020. a avaldatud GPT-3. Ühtegi ChatGPT'le eelnevat versiooni avalikkusele aga ei esitatud. Peamine

---

<sup>3</sup> Google Assistant - <https://assistant.google.com/>

<sup>4</sup> Alexa - <https://alexa.amazon.com/>

<sup>5</sup> Cortana - <https://www.microsoft.com/en-us/cortana>

edasiarendus GPT-3.5 ja ChatGPT vahel oli vestlusliku andmestiku lisamine ja treenimisprotsessi lihvimine.

#### 1.4.6 Bing Chat

Microsoft avaldas vestlusroboti nimega Bing Chat 7. veebruaril 2023 (Rusen, 2023). Arvati, et see on ametlikult avaldamata GPT-4 versioon. OpenAI ütleb aga, et Bing põhineb ühel OpenAI loodud järgmise põlvkonna mudelil, mille Microsoft kohandas spetsiaalselt otsingu jaoks ning see sisaldab ChatGPT ja GPT-3.5 arendusi ja andmeid (Heaven, 2023). Rusen (2023) toob välja peamised erinevused Bing Chati ja ChatGPT vahel:

- Bing Chat on ühenduses internetiga läbi Microsoft Bingi lisaks GPT'ga ühisele andmestikule.
- Bing Chati kasutades saab kasutaja valida kolme vestlusstiili vahel, milleks on loominguline (ingl *creative*), tasakaalustatud (ingl *balanced*) ja täpne (ingl *precise*).
- Bing Chat annab kaasa linke ja viiteid allikatele, kust ta on oma informatsiooni saanud. ChatGPT seda ei tee ja küsimuste korral võib linke isegi välja mõelda. ChatGPT annab pikemaid loomingulisi vastuseid, mis võib vähendada vastuste faktuaalset täpsust.
- Aegadel, kui kasutusnõudlus on kõrge, vastab ChatGPT ainult neile, kes kasutavad tasulist versiooni. Bing Chatiga saab pidada limiteeritud arvu vestlusi päevas (2023. a märtsi seisuga 20 vestlust sessioonis ja 200 vestlust päevas (Ortiz, 2023)).
- ChatGPT eesmärk on pakkuda tasulist toodet, mida ettevõtted oma äriprotsessidesse integreeriks. Bing Chati eesmärk on aga pakkuda tasuta virtuaalset assistenti.
- ChatGPT tehnoloogiat arendab OpenAI üksinda, aga Bing Chati arendab Microsoft OpenAIlt saadud keelemudeli põhjal.
- ChatGPT on kättesaadav vaid veebilehel chat.openai.com. Bing Chat aga veebilehel bing.com ja Bingi telefonirakenduses ning integreerituna ka Skype'i, Windows 11 ja Microsoft Edge'i programmidesse. Bing Chatil on aga ootejärjekord, kuid ChatGPT kasutus on koheselt võimalik.

Bing Chati loojad avastasid, et võimalike vestluste piiramatul arvul tekitab robotile segadust ja limiteerisid kasutust. Kasutuslimiiti tõstetakse arenduse käigus ja sellest jagab oma jälgijatele

aeg-ajalt informatsiooni oma Twitteri kontol Microsofti asepresident Michael Schechter (Ortiz, 2023).

Olgugi et vestlusrobotid on suure tähelepanu osaks saanud viimase 10-15 aasta jooksul, on nende areng on toimunud juba viimased 60 aastat, alates sellest, kui arvutid said piisavalt võimsaks. Vestlusagente on väga palju ja nende arvu on pea võimatu hinnata, kuna praegu võib igaüks oma ise vestlusagendi luua (Kumar, 2023). Tulenevalt nende populaarsusest tuli juturoboti loomise idee ka kursuse “Programmeerimine” loojatele. Järgnevas peatükis kirjutataksegi lähemalt mainitud kursusest, selle korraldusest ja üliõpilastele abi pakkumise lahendusest.

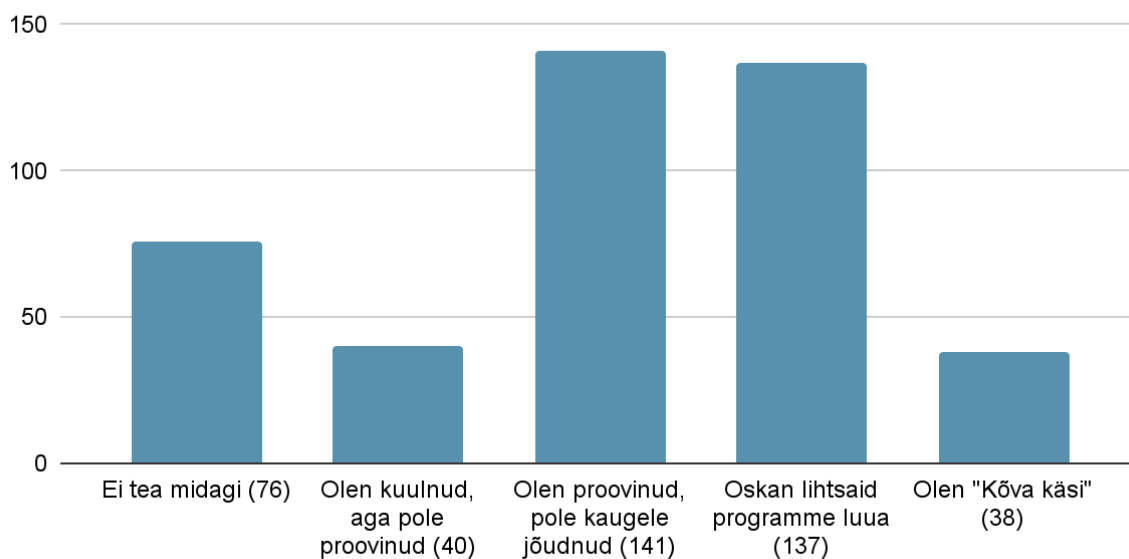
## **2. Kursus “Programmeerimine (LTAT.03.001)”**

### **2.1 Kursuse kirjeldus**

Kursus “Programmeerimine” (LTAT.03.001) on Tartu Ülikooli programmeerimise sissejuhatav kursus, mille eesmärgiks on anda osalejaile alusteadmised programmeerimise põhikonstruktsioonidest, algoritmidest ja programmide loomisest (Programmeerimine, 2022). Kursusel eeldusaineid ei ole ja mingeid varasemaid programmeerimisoskusi ei nõuta. Õppeaine on kohustuslik informaatika, andmeteaduse, matemaatika, matemaatilise statistika, arvutitehnika, matemaatika- ja informaatikaõpetaja, infotehnoloogia mitteinformaatikutele, küberkaitse, infotehnoloogiliste süsteemide arenduse, füüsika, keemia ja materjaliteaduse, ettevõtluse ja digilahenduste ning molekulaarsete biolahenduste õppekavades. Lisaks on aine ka eeldusaineks paljudele teistele õppeainetele, nagu näiteks “Programmeerimine II”, “Objektorienteeritud programmeerimine”, “Programmeerimine keeles C++” ja “Eesti keele töötlus Pythonis”. Kursusel keskendutakse programmeerimise põhitõdede selgestegemisele programmeerimiskeeles Python. 2022. aasta sügissemestril osales kursusel 469 üliõpilast, mis teeb ainest ühe suurema osavõtjate arvuga kursuse Tartu Ülikoolis. Kursuse läbinuna on üliõpilane omandanud teadmised põhilistest programmeerimisfunktsioonidest, andmetüüpidest ja -struktuuridest, suudab mõista programmi töö käiku, seda analüüsida ja muuta, luua ise vajalikke programme ning rühmatöös kaaslastega programmeerimisprojekte.

Kursusel osalenute vahel viidi 2022. sügissemestri alguses läbi küsitlus, saamaks selgemat ettekujutust nende tasemest ja programmeerimiskogemusest (joonis 3).

### Kuidas hindad oma programmeerimisoskusi / teadmisi enne selle kursuse algust?



Joonis 3. Stardiküsitluses antud vastused varasema programmeerimiskogemuse kohta

Stardiküsitluses oli vastajaid 432, kellest 76 ehk 18% ei teadnud programmeerimisest mitte midagi ja 40 ehk 9% ei olnud programmeerimist proovinud, aga oli sellest kuulnud. 9% (38 inimest) vastanuist arvasid, et nad kursusel uusi teadmisi ei saa. Kuna aine on, nagu eelnevalt mainitud, paljudele teistele ainetele eeldusaineks, on üliõpilastel tarvis nendes ainetes positiivsete tulemuste saamiseks siin põhitõed selgeks saada. Selleks tuleb neile pakkuda häid õppimismaterjale ja tuge.

## 2.2 Kursuse korraldus

Õppetöö aines "Programmeerimine" toimub eesti keeles, hindamine on eristav ning õppevormiks päevaõpe. Koodikirjutamist õpitakse programmeerimiskeeles Python. Õppetöö vorme on kolm: loengud, praktikumid ja iseseisev töö. Iga mainitu maht vastavalt 32, 32 ja 92 tundi (Programmeerimine, 2022). Kursuse õpetamise vormi nimetatakse ümberpööratud klassiruumi vormiks. Selles tuleb iga praktikumi eel teema õpikust läbi töötada, koduülesanded lahendada ning lahendused esitada (Kursuse "Programmeerimine" koduleht, 2022). Praktikumides lahendatakse teemade kinnistamiseks veel käsitletavale materjalile tuginevaid ülesandeid. Seejärel tuleb üliõpilastel taas uue teemaga tegelema hakata.

Hinde kujundamiseks saab punkte seitsmest kategooriast: testid, kodu- ja praktikumitöö, projekt, kaks kontrolltööd, eksam ning lisapunktid. Kontrolltöödel ja eksamil toimub hindamine kahes osas, millest üks on arvestuslik ja teine punktiline. Kodutööd on head ülesanded üliõpilasele ise pusimiseks ja ei anna väga palju punkte (13 punkti 100st). Seevastu on programmeerimise õppimisel iseseisev töö väga tähtis ja Kinnuneni uuringust (2009) selgus, et üheks peamiseks programmeerimise kursusest väljakukkumise põhjuseks on see, et materjalist ei saada aru. Motivatsiooni säilitamiseks on oluline, et õppija saaks ülesannete lahendamisel võimalikult palju, kergesti ligipääsetavat ja adekvaatset abi. Nagu eelnevalt mainitud, on tegemist väga suure osavõtjate arvuga kursusega. See tähendab, et on palju inimesi, kes kodutööd lahendavad ja potentsiaalselt kuhugi takerduvad. Et üliõpilaste abistamiseks ja sagedastele küsimustele vastamiseks ei rakendataks liiga palju õppejõudude aega ning vastused oleksid kiiremini kättesaadavad, võeti Tartu Ülikoolis MOOCi (ingl *Massive Online Open Courses*) kursustel kasutusele murelahendajad. Aasta 2022 sügissemestriks olid kodutööde uuendamise tõttu alles vaid mõnede koduülesannete murelahendajad ning kursusele plaaniti uuendust.

## 2.3 Kursuse murelahendajad

Murelahendajad (ingl *troubleshooter*) võeti Tartu Ülikooli arvutiteaduste instituudis 2016. aastal kasutusele MOOCides ehk vaba ligipääsuga ja osalejate arvu poolest suuremahulistel tasuta e-kursustel (Hollo, 2016). Murelahendaja abistab kursusest osavõtjaid, andes neile suunavaid vihjeid ülesannete lahendamiseks. Vello Vaherpuu programmeeris 2016. aastal oma bakalaureusetöö raames keskkonna murelahendajate koostamiseks. Murelahendaja tähistabki selle keskkonna abil loodud konkreetse programmeerimisülesande kohta käivate vihjete komplekti (Hollo, 2016). Varasemalt murelahendajaid erinevatel kursustel rakendades on uuritud ka nende mõju üliõpilaste tulemustele ja selle põhjal saadud tagasiside murelahendajate kasulikkuse kohta. MOOCi kursusel “Programmeerimisest maalähedaselt” vähenes peale murelahendajate lisamist abiliinile pöördumiste arv 29% (Lepp et al., 2018). Aastal 2021 kirjutas Joosep Kaimre oma bakalaureuse lõputöö murelahendajate koostamisest kursusele “Programmeerimine” ja lõi murelahendajad kaheksa nädala kodutöödele (Kaimre, 2021). Siinne töö tegeleb sama kursusega ning osalt on säilinud Kaimre poolt loodud vihjed.



Aastate möödudes on kursuse koduülesandeid uuendatud ning uutele ülesannetele murelahendajaid ei leidu.

Murelahendajate koostamine ja uuendamine on aga tülikas ja võtab palju aega ning energiat. Esmalt tuleb tutvuda kodutöös püstitatud ülesannetega ning kursusel läbitud materjaliga. Siis tuleb kodutöö lahenduskäiku analüüsida ning leida potentsiaalsed murekohad, see eeldab vähemalt mõneti kodutöö läbilahendamist. Kodutööde analüüs ja vastuste koostamine ongi sellise abimeetme kõige olulisem osa. Murelahendaja lineaarse ülesehituse tõttu tuleb paika panna ka vihjete andmise järjekord, mis oleks üliõpilase lahendusprotsessiga vastavuses. Üliõpilastele abipakkumise hõlbustamiseks otsustati luua ainesse “Programmeerimine” uus, Murelahendajale sarnase funktsiooniga programm ProgVestlus. Sellest on kirjutatud lähemalt järgnevas peatükis.

## 3. Metoodika

Selles peatükis räägitakse lähemalt, kuidas toimib ProgVestluse vestlusagent, selgitatakse kodutööde analüüsiprotsessi ning kirjeldatakse samme, mida tuleb teha, et keskkonda vastuseid lisada.

### 3.1 Keskkond ProgVestlus

Mõte interaktiivne vestlusagent luua on olnud programmeerimise algainetes juba pikemat aega, sest vestlusagendil oleks praegu kasutatavate murelahendajatega mitmeid eeliseid. Konkreetne plaan tekkis 2021. aasta detsembris, kui käivitusid “Programmeerimise” aine arendamise koosolekud. Esimene versioon sai valmis 2022. aasta veebruaris, selle tegi Sven Aller. Siis toimus keskkonna katsetamine, mille põhjal tekkisid täpsemad ideed, kuidas süsteem võiks toimida. 8. märtsist 2022 on rakendus saadav aadressil progvestlus.cs.ut.ee. Pärast suvevaheaega toimus märksõnade käsitlemise loogika arendamine ja 13. oktoobril 2022 valmis teine versioon, mis on ka praegu kasutusel.

Vestlusagendil on murelahendaja ees nii mõnedki eelised. Kõigepealt võimaldab see kursusel kasutatavad murelahendajad viia ühtsesse süsteemi, kus neid on lihtsam muuta ja täiendada. Murelahendajatel on võrreldes vestlusagendiga rida piiranguid ja ebamugavusi: murelahendaja on seotud konkreetse ülesandega ning näiteks ülesande asendamisel teisega muutub murelahendaja kasutuks; murelahendaja koostamisel tuleb ära arvata, mis küsimused kasutajal tekivad, ja välja mõelda, millises järjekorras neile lahendusi pakkuda; kasutajal võib olla vaja mitu ekraani läbi klõpsida enne, kui oma probleemile lahenduse leiab või saab selgeks, et murelahendaja vihjet ei pakugi. ProgVestluses on kõik ühel ekraanil, vihjete järjekord ei ole oluline ja vihjeid on lihtne muuta või täiendada. Vestlusagendi kasutamine on seega Murelahendajaga võrreldes märksa mugavam, olgugi et kasutajal tekkivate küsimuste äraarvamise aspekt säilib.

Lisaks võimaldab vestlusagent koguda paremini andmeid õppijate tegelike probleemide ja vajaduste kohta. Nende andmete põhjal saab näha, kus on kursuses raskemad kohad, ja vajadusel teha muudatusi, näiteks rõhutada või selgitada mõnda asja rohkem, lisada uusi või ühendada olemasolevaid teemasid jne. Välise privaatse vestlussüsteemi nagu ChatGPT

kasutuslogidele juurdepääsu pole ning seal oleks ka raske eristada, kas päring tuli seoses konkreetse õppeainega või mingil muul põhjusel.

ChatGPT antavatel vastustel puudub garantii, et nad on usaldusväärsed. Õppijale, kes vastavat teemat veel hästi ei tunne, võib see olla isegi ohtlik, sest niimoodi võib ta tõe pähe võtta asju, mis tegelikult ei kehti. Progvestluse vastused saab sõnastada nii, et nad on kindlasti õiged.


Vaatamata praegusele melule ChatGPT ümber pole ette näha, et sellel lähemal ajal tekiks mõistlik suutlikkus lahendada keerulisemaid ja harvaesinevamaid ülesandeid. Mida spetsiifilisem on teemavaldkond, seda vähem on olemas andmeid, mille põhjal saaks mudeleid treenida. Niisugused teemavaldkonnad on aga olulised just ülikoolis, sest ülikool on koht, kus uusi teadmisi luuakse ja õppijatele edasi antakse. Kui ChatGPT on läbi lugenud kogu interneti ja saadud teadmiste baasil oskab lahendada ainult lihtsaid standardseid ülesandeid, siis kui palju aega kulub, et sellel tekiks võime lahendada mittestandardseid ülesandeid? Õieti, nagu nüüd kevadsemestril ingliskeelses programmeerimise aines on täheldatud, ei suuda see alati õigesti lahendada ka lihtsamaid, programmeerimise algkursuse ülesandeid. Samuti ei suuda see lahendada suuremahulisi ülesandeid. ChatGPT võib küll abi anda sellise ülesande osade juures, aga loota, et ta ise mingi suure süsteemi valmis teeks, on praegu ebareaalne. Lisaks annab ChatGPT lahenduse, aga õppimisprotsessi toetaks paremini vajalik osaliste vihjete süsteem, et üliõpilasele jääks ikkagi võimalus iseseisvalt ülesannet lahendada vaid osalise abiga.


### **3.2 ProgVestluse tehniline kirjeldus**


ProgVestluse keskkonnal on kaks liidest. Üks neist on kasutamiseks üliõpilasele, kes vestlusagendilt abi palub, ning teine administraatorile. Link üliõpilase kasutatavale lehele on <http://prog.keeleressursid.ee/progvestlus/index.php> ja administraatori lehele <http://prog.keeleressursid.ee/progvestlus/admin/index.php>. Administraatori keskkonnale ligipääsu saamiseks tuleks võtta ühendust Sven Alleriga. Abiküsija vaates on küsimuste sisestamiseks vaba teksti väli ja nupp küsimuse edastamiseks (joonis 4). Administraatori lehel on võimalik sisestada vastuseid nende pealkirja, lausearvutusvalemi ja nädalaga (joonis 5).

Administraator saab määrata vastustele ajaraame ja vaadata logisid, mis on tekkinud kasutajate loodud päringutest.

## ProgVestlus

 Tervist, püüan aidata programmeerimisülesannete lahendamisel.

 appi

 Palun kirjelda oma probleemi täpsemalt.

Joonis 4. üliõpilasevaade ProgVestlus keskkonnas

## ProgVestlus: administreerimine [Logi välja]

### Vastused

Vastus (sh kood)	Lausearvutusvalem	Nädal	
☑ Üldine abiküsimine	abi vajamine	0	👉 ✖
☑ Üldine abiküsimine	abi   abistamine   aitamine   appi   vihje   probleem   raskus   lahendamine	0	👉 ✖
☑ Üldine abiküsimine	kodutöö (abi   abistamine   aitamine   appi   vihje   probleem   raskus   lahendamine)	0	👉 ✖
☑ Üldine abiküsimine	(ülesanne   koduülesanne) (abi   abistamine   aitamine   appi   vihje   probleem   raskus   lahendamine)	0	👉 ✖
☑ Üldine abiküsimine	ülesanne   koduülesanne	0	👉 ✖
☑ Üldine abiküsimine	kodu1   kodu2   kodu3	0	👉 ✖
☑ Üldine abiküsimine	(automaatkontroll   automaattest   test) viga	0	👉 ✖

Joonis 5. Administraatori vaade ProgVestlus keskkonnas

Vastuse lisamiseks adminikeskkonnas tuleb klõpsata nupul “[lisa vastus]”. Sealt avaneb vaade (joonis 6), kus on väljad, mis vastuse sisestamiseks täita tuleb. Kui kõik soovitud andmed on sisestatud, saab lisada vastuse andmebaasi. Lisaks uute vastuste sisestamisele saab ka vanu vastuseid muuta ja kustutada. Selleks on iga vastuse taga vastavalt ringikujuline noolega nupp ja ristikujuline nupp.

**Vastused**

Pealkiri:	<input type="text"/>		
Vastus:	<div> <span>↩</span> <span>↪</span> <span>&lt;KOOD&gt;</span> <span><i>I</i></span> <span><i>o</i></span> <span>&lt;&gt;</span> </div> <div> <p>p</p> </div>		
Lausearvutusvalem:	<input type="text"/>		
Nädal:	<input type="text"/>		
DNK:	<input type="text"/>		
<div> <a href="#">Lisa vastus andmebaasi</a> <a href="#">Tühista</a> </div>			
<b>Vastus (sh kood)</b>	<b>Lausearvutusvalem</b>	<b>Nädal</b>	
☑ Üldine abiküsimine	abi vajamine	0	🔍 ✖
☑ Üldine abiküsimine	abi   abistamine   aitamine   appi   vihje   probleem   raskus   lahendamine	0	🔍 ✖

Joonis 6. Vastuse sisestamine ProgVestlus keskkonnas

ProgVestluse programm kasutab lausearvutuse meetodeid. Vastused on süsteemis kirja pandud lausearvutusvalemitega, kus tühik tähistab konjunktsiooni, püstkriips disjunktsiooni ja sidekriips eitust. Võimalik on kasutada sulge. Tehted võivad asuda üksteise sees ükskõik kui sügaval. Süsteem teisendab valemi disjunktiivsele normaalkujule, mis on komplekt sõnade hulki. Igale sõnade hulgale vastab sama vastus. Kui kasutaja sisestab päringu, teisendab programm kõik päringus sisalduvad sõnad sisseehitatud lemmatiseerija abil algvormi. Lemmatiseerimise puhul on programmisiseselt liitsõnade osad üksteisest lahutatud, et administraator ei peaks arvestama võimalikke kokku- ja lahkukirjutamise vigu, mida kasutaja teha võib. Peale lemmatiseerimist programm otsib saadud normaalkujust liiget (elementaarkonjunktsiooni), mis on kasutaja küsimuse peal tõene, st kõik eitusega (positiivsed) sõnad on küsimuses olemas ja ühtegi eitusega (negatiivset) sõna pole. Niimoodi vaatab läbi kõigi valemite kõik liikmed. Kõigist sobivatest liikmetest valib lõpuks selle, milles eitusega sõnade arv on kõige suurem, ja annab selle valemiga seotud vastuse. Näiteks kui kasutaja sisestab “Kuidas sõne väljastada?”, muudab programm selle enese tarbeks lauseks “kuidas sõne väljastamine”. Andmebaasis leidub lausearvutusvalem “sõne väljastamine”, mille liikmed on vastavuses kasutajasisestusega ning süsteem tagastab andmebaasi sisestatud vastuse selle lausearvutusvalemi realt. Administraatorikeskkonnas

lausearvutusvalemiste loomisel ei saa kasutada alakriipse ega langkriipse. Samuti ei tunne süsteem ära, kui on sisestatud punktiga eraldatud arve (näiteks 3.2 jms).

### **3.3 Tööprotsessi kirjeldus**

Töö eesmärgist lähtuvalt oli vaja luua sisu keskkonnale ProgVestlus. Keskkond ja selle andmebaas oli uus ja sisaldas vähe sissekandeid (peamiselt vaid loojate poolt testimiseks tehtuid). Kuna sellises vormis ei olnud sellest kasu, oligi tarvis tekitada andmebaasi andmeid. Töö autor analüüsis saadaval olevate logide, kursuse materjali ja olemasolevate murelahendajate toel ise kõik kodused tööd läbi. Kuna ChatGPT puhul on palju kõneainet pakkunud tema mitmekülgne võimekus, siis lasti võrdluseks ka ChatGPT'l nende samade ülesannete kohta küsimusi genereerida ja võrreldi tulemusi. Siin peatükis kirjeldatakse seda, kuidas analüüsi protsess kulges.

#### **3.3.1 Programmeerimise aine koduülesanded**

Kursuse ajakava (joonis 7) on üliõpilastele nähtav keskkonnas [courses.cs.ut.ee](https://courses.cs.ut.ee). Aine kestab 16 nädalat, millest enamikel on tarvis lahendada kodutöid. Kahel nädalal on kodutöö asemel antud üliõpilastele aega kontrolltööks valmistuda ning üks nädal on jäetud projektiesitluseks valmistumiseks. Seega on kursusel 13 komplekti kodutöid. Iga nädala kodutöökomplektis on üldjuhul kaks kuni neli programmeerimisülesannet, mis rakendavad vastaval nädalal õpitud teemasid. Kõigi eelmainitute jaoks valmisid bakalaureusetöö käigus sisendid andmebaasi. Sisendite koostamisest ja loomisest kirjutatakse järgnevalt lähemalt.

## Ajakava

Nädal	Teema	Kodutöö	Praktikum	Harjutused	Projekt
1.	Muutuja ja avaldis	Kodutöö 1	1.09	Harjutused	
2.	Tingimuslause	Kodutöö 2	8.09	Harjutused	
3.	Funktsioon	Kodutöö 3	15.09	Harjutused	
4.	Korduslause	Kodutöö 4	22.09	Harjutused	
5.	Lihtsam failitöötlus. Sõned	Kodutöö 5	29.09	Harjutused	
6.	Kordamine. KONTROLLTÖÖ	Kontrolltööks valmistumine	6.10	Kontrolltöö	
7.	Järjend	Kodutöö 7	13.10	Harjutused	Projekti algus
8.	Järjend II	Kodutöö 8	20.10	Harjutused	Projekti kirjeldus
9.	Kahekordne tsükkel. Failitöötlus	Kodutöö 9	27.10	Harjutused	
10.	Andmestruktuurid	Kodutöö 10	3.11	Harjutused	
11.	Andmestruktuurid II	Kodutöö 11	10.11	Harjutused	Projekti algversioon
12.	Kordamine. KONTROLLTÖÖ 2	Kontrolltööks valmistumine	17.11	Kontrolltöö	
13.	Rekursioon	Kodutöö 13	24.11	Harjutused	
14.	Objektorienteeritud programmeerimine	Kodutöö 14	1.12	Harjutused	
15.	Objektorienteeritud programmeerimine II	Kodutöö 15	8.12	Harjutused	Projekti lõppversioon
16.	Kordamine. Projektide esitlused	Projekti esitluse loomine	15.12	Projekti esitlus	

Joonis 7. Kursuse ajakava ja teemad (Kursuse “Programmeerimine” koduleht, 2022)

Selleks, et kokku koguda kogu ProgVestluse vestlusagendi andmebaasi sisestatud materjal ning mustandstaatuses selles muudatusi teha, loodi esmalt Google Sheetsi arvutustabel, mis asub lisas I toodud lingil. Arvutustabelisse tehti 14 vahelehte. Esimesest 15. nädalani koostati iga kodutöödega nädala jaoks vaheleht vastava nädala kodutööde jaoks ning lisaks neile loodi üks vaheleht mõistete, veateadete, organisatsioonide ja otseselt programmeerimise välise materjali hoiustamiseks. Arvutustabelis on, nagu mainitud, mustand ja see sisaldab märkmeid ka töö autori isiklikuks tarbeks. ProgVestluses sisalduv materjal on kättesaadav kogu kursuse vältel. Küll aga on võimalik iga sissekande juures täpsustada nädal, millal see käsitlusel on. Nädala number seab selle nädala vastused defineeritud perioodil prioriteetsemaks.

### 3.3.2 Logide analüüs

Logid asuvad administreerimise keskkonnas. Neis näeb infot sisestuste kuupäevade, kasutaja sisestatud teksti, IP-aadressi ja osaliselt ProgVestluse antud vastuse kohta. Nime, ega muud taolist isiklikku informatsiooni ProgVestlus ei kogu. Kodutööde analüüsis kasutati logisid, et aru saada, milleks olid üliõpilased juba üritanud ProgVestluse vestlusagendi kasutada. Ajal,

mil nad vestlusagenti proovida said, ei olnud see veel oma funktsionaalsuselt päris täpne ja üliõpilaste päringuid oli vähe. Sellegipoolest olid pöördumised logides hästi säilinud.

Enne logidesse süvenemist olid kodutööde analüüsiks loodud Google Sheetsi vahelehtedele jaotised ja alapealkirjad ülesannete kohta, et oleks võimalik kasutajate sisestusi õige teema juurde märkida. Logide uurimisel tuli välja, et üliõpilaste küsimismeetodid olid erinevad. ProgVestluse esimestel katsetusel olid üliõpilased huvitatud vestlusagendiga programmeerimist mittepuudutavatel teemadel suhtlemisest. Ootuspäraselt olid üldjoontes küsimused siiski seotud Pythonis programmeerimise, programmeerimise mõistete ja kodutöödega, kuid logidest oli tajuda ka üliõpilaste frustratsiooni, kui vestlusagent neid ei mõistnud. Kasutajakogemuse parendamiseks lisati ülesande üldise vihje juurde nimekiri küsimustest (joonis 8), mis kodutöö lahendamisel võivad tekkida. Küsimuste läbilugemine oli üliõpilasele ühtlasi esimeseks vihjeks, kui tal tekstis midagi märkamata oli jäänud. Küsimuste loetlemine ei piiranud kasutajat mõistete ega terminite kohta uurimast. Lisaks saadi kinnitust, et lausearvutusvalemid peaksid sisaldama võimalikult palju sünonüüme, või alternatiivina võiks tulevikus lisada ProgVestlusele sünonüümisõnastiku toe.

## ProgVestlus: administreerimine [Logi välja]

### Vastused

#### Muutmine: vastus "Ül 10.1"

Pealkiri:	Ül 10.1
Vastus:	<div><div>&lt;KOOD&gt; <i>I</i> <i>o</i> &lt;&gt;</div><div>Ülesande 10.1 juures saab küsida järgmisi küsimusi: 1) Kuidas defineerida funktsioon erinevad_sümbolid ja mis on tema argument? 2) Kuidas sõne sümbolite hulka lisada? 3) Kuidas defineerida funktsioon sümbolite sagedus ja mis on tema argument? 4) Kuidas luua tühja sõnastikku? 5) Kuidas loendada sümbolite esinemist? 6) Kuidas defineerida funktsioon grupeeri ja mis on tema argument? 7) Kuidas lisada sõnastikku võtit, millele vastab hulk? 8) Kuidas hoiustada sümbolite sagedust? 9) Kuidas saab sõnastikust kirje kätte? 10) Kuidas kontrollida sümboli kuulumist häälikute hulka? 11) Kuidas lisada sõnastiku võtme hulgale väärtusi?  Kui Sul on küsimusi seoses terminite või mõistetega, küsi julgelt! (nt "input", "print" jne.)</div><div>P</div></div>
Lausearvutusvalem:	(Ülesanne 10.1)   (Ülesanne 1)   (Teksti analüüs)
Nädal:	10
DNK:	

Joonis 8. Näide küsimustest ülesande 10.1 juures

Ehkki logidest kõikide ülesannete puhul tekkivate küsimuste kohta näiteid ei leitud ja logid sisaldavad ka loojate katsetusi, andsid olemasolevad sisestused siiski olulist informatsiooni.



### 3.3.3 Koduülesannete analüüs ja vastuste loomine

Kaspar Hollo bakalaureustöös (Hollo, 2016) käsitletud küsimuste analüüsist selgub, et põhilised kohad, millega seoses küsimusi tekib, on sageduse kahanevas järjekorras järgmised:

- 1) muutuja viga,
- 2) funktsiooni väljakutse,
- 3) ümardamine,
- 4) funktsiooni argument puudu või kasutamata,
- 5) sisendiviga,
- 6) tüübiviga,
- 7) süntaksi- ja treppimise vead,
- 8) arvutusviga.

Nendele kohtadele katsuti teemade kattuvuse korral analüüsimise protsessis tähelepanu pöörata. Analüüsiti ka koduülesannete tekstide ülesehitust, leidmaks kohti, mis võivad jääda märkamata või segaseks. Üliõpilase hetkepädevusest ettekujutuse saamiseks ning teemas püsimiseks tutvuti õpikumaterjalidega ja mõistete selgitusel toetuti õpikus või kursusel kasutusel olevatele definitsioonidele. Et internetist leitav programmeerimise õppematerjal on suuresti ingliskeelne, peeti definitsioonide puhul silmas ka, et säiliks eestikeelsed terminid (joonis 9). Mõningate kodutööde juures olid säilinud varasematest aastatest murelahendajad. Murelahendajate informatsioon kanti samuti ProgVestlusesse sisse. Materjali analüüsimisel ja vihjete koostamisel tuli silmas pidada, et ProgVestlus ei saa käsitleda alakriipsu sisaldavaid nimesid. Paljud koduülesannetes loodavad funktsioonid aga on kirjas kujul “esimene\_teine\_kolmas”, kus sõnu eraldavad alakriipsud. Seega ei saa ProgVestluselt küsida otseselt funktsiooni kohta funktsiooni nime kasutades. Töö käigus ilmnis ProgVestluse keskkonnas probleeme, mida töö autori algatusel Sven Aller lahendas. Näiteks katsetati alakriipsude kasutamise võimaldamist, kuid nendega hästi toimiv lahendus vajaks suuremaid süsteemisisesid muudatusi.



mis on ennik?



Ennik (ingl tuple) on Pythoni andmetüüp, mis on järjendiga väga sarnane. Kõige suurem vahe enniku ja järjendi vahel on see, et järjendeid saab muuta, ennikuid aga mitte. Enniku elemendid kirjutatakse ümarsulgude vahele, aga nende indekseerimiseks kasutatakse siiski nurksulge. Ennikus võivad esineda duplikaadid ja ka erinevad andmetüübid. Ennikust saab elemente kätte samamoodi nagu järjendist.

```
ennik = ("sõne", "element", 33, False)
print(ennik[2])
33
```



mis on tuple?



Ennik (ingl tuple) on Pythoni andmetüüp, mis on järjendiga väga sarnane. Kõige suurem vahe enniku ja järjendi vahel on see, et järjendeid saab muuta, ennikuid aga mitte. Enniku elemendid kirjutatakse ümarsulgude vahele, aga nende indekseerimiseks kasutatakse siiski nurksulge. Ennikus võivad esineda duplikaadid ja ka erinevad andmetüübid. Ennikust saab elemente kätte samamoodi nagu järjendist.

```
ennik = ("sõne", "element", 33, False)
print(ennik[2])
33
```

Saada

Joonis 9. Näide enniku definitsioonist ProgVestluses

Koduülesannete analüüsil ja vastuste loomisel lahendati koduülesanded õpitud materjali kasutades läbi. Autori enda lahendusi võrreldi koduülesannete näidislahendustega ja vihjete andmisel lähtuti kasutatud programmeerimisvõtetest. Lisaks mainitule püüti kirja panna ka vastuseid ka küsimustele, mis võivad tekkida, kui üliõpilane on segaduses ja nende küsimuste vastustega teda õigele teele tagasi suunata. Segaduses üliõpilast on keeruline imiteerida, mistõttu sai neil puhkudel suuniseid logidest.

### 3.3.4 Ülesannete analüüsimine ChatGPT'ga

Koduülesannete probleemkohtade väljaselgitamine ja analüüs on üpriski keeruline ja aegavõttev tegevus. Sellepärast oli töö autor huvitatud uurima alternatiivseid lahendusi protsessi kiiremaks ja mugavamaks tegemiseks. Kuna ChatGPT'd peetakse võimekaks mitmetes valdkondades ja see sobitub oma olemuselt töö teemaga, paluti ka sellel ülesannete kohta küsimusi genereerida. Vaadatavad ülesanded olid “3.3 Moosi keetmine”, “7.2 Takso hinnad” ja “10.1 Teksti analüüs”. Need võeti erinevatest kursuse etappidest ja eri teemadelt. Lisas II on toodud tabel, kus on töö autori mõeldud küsimused ning küsimused, mille pakkus välja ChatGPT. ChatGPT'lt abi küsimiseks küsiti talt “Millised tehnilised küsimused võivad tekkida algajal programmeerijal järgmist ülesannet lahendades?” ja seejärel kopeeriti koduülesande tekst.

Nende tabelite võrdlus näitab, et ChatCPT koostab hästi konteksti sobituvaid küsimusi. Mida tähele panna, on et ChatGPT genereerib igal korral eelmisest erineva vastuse. Seetõttu, kui üliõpilased seda kasutaksid, ei saaks nad võrdselt abi. ChatGPT'l saab paluda ka oma esitatud küsimustele vastuste andmist. Pakutavad vastused sisaldavad sageli ka koodijuppi, mis oleks lahendajale päris suureks vihjeks. Ühtlasi saab ilma ise mõtlemata paluda ChatGPT'l juba ära lahendada terve ülesanne. Ühe kodutöö puhul katsetati ka, kuidas ChatGPT selle tervikuna ära lahendaks. ChatGPT lahendust ja kursusel antud näidislahendust saab võrrelda lisas III asuva joonise põhjal.

ChatGPT loodud küsimusi saab ProgVestluse andmestiku koostamisel kindlasti appi võtta ja enda loodud materjalidega kombineerida, kuid samal ajal tuleb säilitada kriitiline meel olukordades, kus otsustatakse antavate vihjete hulk. Programmeerimise algkursustel on materjali selgekssaamise vastutus väga suuresti õppija õul, sest kodutööde põhjal ei pruugi õppejõud enam tegelikest oskustest ülevaadet saada. Et ChatGPT ja muud sarnased abivahendid on aga kõigile lihtsasti kättesaadavad, siis peaks taolistel algteadmisi õpetavatel kursustel nagu “Programmeerimine” korraldusliku poole pealt mõningaid ümberkorraldusi tegema.

### **3.4 Üliõpilaste tagasisideküsitlus**

Vestlusagendi ProgVestlus kasutajakogemuse uurimiseks viidi aine “Programmeerimise alused II” osalejate seas läbi küsitlus (lisa IV). “Programmeerimise alused II” valiti selle pärast, et see on kevadsemestritel läbiviidav aine ja kursus “Programmeerimine” toimub vaid sügissemestritel. Üliõpilastel paluti proovida ProgVestluse käest küsimusi küsida ja kirjeldada oma oodatud vastuseid, samuti seda, kuivõrd ProgVestlus nende ootuseid täitis, ning mis neile meeldis ja mis mitte. Lisaks sellele said vastajad võrrelda ProgVestlust ChatGPT ja Murelahendajatega ning välja pakkuda võimalikke edasiarendusi.

## 4. Tulemused ja järeldused

Bakalaureusetöö tulemusena valmis andmestik ainel “Programmeerimine” kasutuselevõetavale ProgVestlus vestlusagendile. Kursus kestab 16 nädalat, millest 13 nädalal on üliõpilastel tarvis lahendada koduülesandeid. Igal nädalal on kodus lahendamiseks jäetud kaks kuni neli koduülesannet, mis tähendab, et abimaterjal loodi 33 kodutööle, mõnele lisaülesandele ja üle 160 teemade juurde kuuluvate mõiste ja vihje jaoks, mille kohta kasutajad küsida saavad. Ülevaatliku tabeliga, loodud vihjete arvu kohta, saab tutvuda lisa V. Kokku sisaldab ProgVestlus töö tulemusena üle 340 vastuse. Lisaks valmis vestlusagentide teoreetilise tausta ülevaade ja juhend ProgVestluse administraatori sissekannete loomiseks (lisa VI).

### 4.1 Tagasisideküsitluse tulemused

Tagasisideküsitlusele ei saadud palju vastuseid. Kursusel, mille osalejad tagasisideküsitlust täitma kutsuti, oli vastamisperioodil vaba nädal ja see võis tingida üliõpilaste vähese aktiivsuse. Küsitlusele vastas üks üliõpilane, kes oma antud vastuste põhjal ei olnud täiesti algaja programmeerija. Tema antud vastustest saadi kinnitust, et kui inimene igapäevaselt pikema perioodi vältel on juba Pythonis programmeerimisega tegelenud, siis pole tal elementaarsete mõistete puhul küsimusi ja ProgVestluselt eriti abi ei saa. Vastaja ootas vastuseid oma reaalse isiklike projektide puhul tekkinud küsimustele, mis on hetkel ProgVestluse pädevusest väljaspool. Üliõpilane küsis näiteks, kuidas Exceli failis Pythoni abil lahtreid muuta ja tõlkida. Olgugi et vastanute arv oli tagasihoidlik, saadi süsteemi muutmiseks natuke kasutajakogemuse tagasisidet. Ilmnes, et ProgVestlusele tuleks lisada mõningaid vihjeid või suuniseid selle kasutamiseks või iga vastuse puhul anda ette näiteks teemaga haakuvaid märksõnu, mis suunaks kasutajat järgmiste küsimusteni.

### 4.2 Järeldused ja arutelu

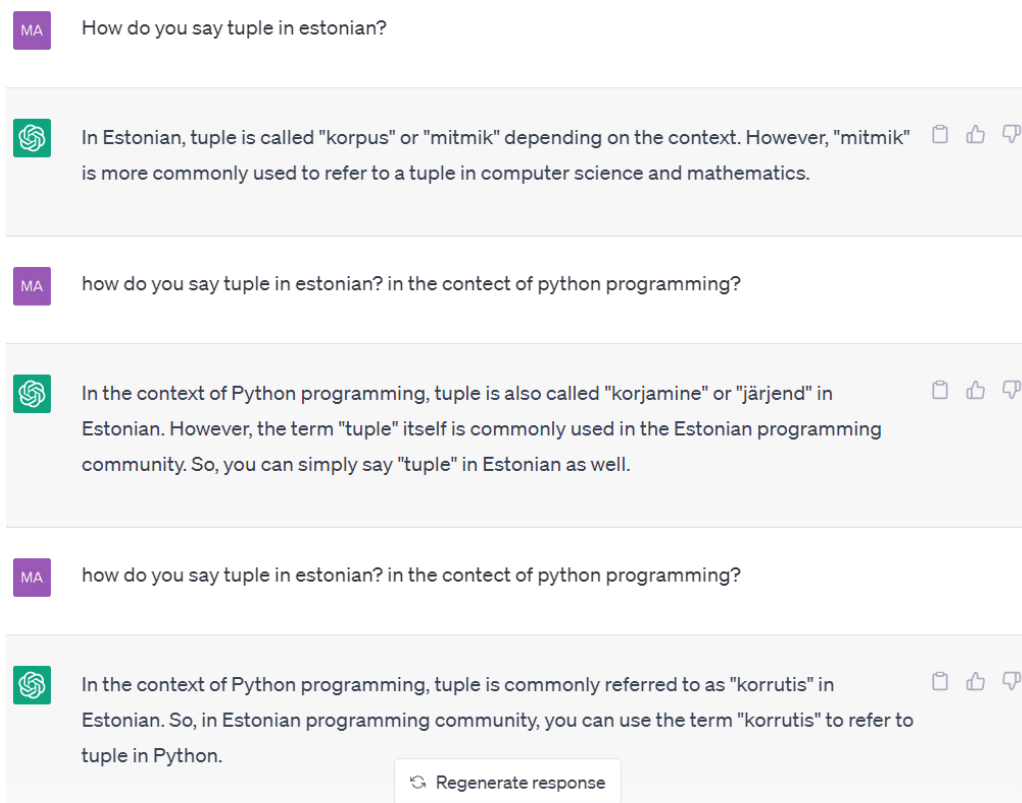
ProgVestlus on oma eesmärgilt sarnane selles aines ja muudel MOOCidel kasutatav Murelahendajate süsteem. Mõlemad on abiks üliõpilasele kodutööde lahendamiseks, püüavad vastata korduma kippuvatele küsimustele ja vähendada seejuures õppejõudude koormust. ProgVestlus on aga oma omadustelt Murelahendajast paindlikum ja interaktiivsem.

Vestlusagendilt on võimalik kasutajatel küsida selgitusi rohkemate mõistete ja murekohtade kohta, kui Murelahendaja range skoobi puhul. Samuti ei ole erinevalt Murelahendajast ProgVestlusel vaja vihjete kuvamise järjekorda defineerida. Sellegipoolest on vältimatu osa see, mis on ühtlasi ka murelahendajate ja ProgVestluse koostamisel kõige keerulisem - küsimuste ennustamine. Algajatel programmeerijatel võivad tekkida küsimused elementaarsetes kohtades ja veateadetes on sageli erinevad põhjused. Vestlusroboti ProgVestlus programm töötab kasutades lausearvutust ja lemmatiseerimist. ProgVestluse vastused on vastavalt lausearvutuse valemite sisalduvatele komponentidele eelnevalt defineeritud, mis teeb sellest reeglipõhise ja valdkonnaspetsiifilise juturoboti.

Selles töös leiti, et analüüsimisel saab abijõuna kasutada ChatGPT pakutavaid küsimusi. Lisaks, võivad tekkida üliõpilastel ka erilised küsimused, mida korduvate küsimuste hulka liigitada ei saa ja kursusel nii täpselt kirjeldatud ei ole. Selliste olukordade lahendamiseks saaks tulevikus ProgVestlust veel täiendada. Järgmistest võimalikest sammudest kirjutatakse järgnevas peatükis.

Lugejal võib tekkida küsimus “Miks peaks üliõpilane kasutama ProgVestlust ja mitte ChatGPT’i?”. See küsimus on tõepoolest adekvaatne, kuid õieti on mõlemal agendil omad head ja vead. Kursusel “Programmeerimine” ning üldiselt Tartu Ülikoolis pööratakse rõhku eesti keele kasutusele. ProgVestlus sisaldab õppematerjalides kasutatud termineid ja tagastab alati ühese vastuse. ChatGPT suudab aga genereerida igal korral uue vastuse, mõtleb sageli “fakte” välja ja ei ole kursis eestikeelsete programmeerimisterminitega, mistõttu võib kasutaja erineva informatsiooni esitamise puhul segadusse sattuda. Joonis 10 illustreerib ChatGPT antud vastuseid, kui temalt küsida *tuple* (ennik) eestikeelse tähenduse kohta. Et ChatGPT vastused ka samale küsimusele erinevad, tähendab see, et üliõpilased saaksid teda abijõuna kasutades ebavõrdsetes mahtudes juhiseid. ChatGPT on võimeline analüüsima talle esitatud tekste ja lahendama programmeerimisülesandeid (vt lisa III). Kui üliõpilane ChatGPT’lt abi küsib, vastab viimane põhjalikult ja sageli ka koodinäidetega. Programmeerimise aine mõte on aga, et üliõpilased ise harjutaksid ja õpiksid. ProgVestlus tagastab kontrollitud mahus vihjeid ilma koduülesannete lahendamist liiga lihtsaks tegemata. Veel üks põhjus (vähemalt praegu) on ka see, et Progvestlus on vabalt kõigile kasutatav, samas kui ChatGPT nõuab juurdepääsemiseks oma andmete andmist ülikoolivälisele süsteemile.

Üheks ProgVestluse miinuseks on see, et ta ei pruugi tingimata alati kasutaja päringuid mõista. Seda siis, kui päringus sisaldavaid märksõnu andmebaasis ei ole. ChatGPT on aga suurel hulgal andmetel treenitud ja annab igal juhul mingi vastuse, kusjuures arvestab ka kasutaja võimalikke eelistusi. ProgVestluse säilivate logide põhjal on võimalus koguda andmeid õppijate tegelike probleemide ja eksiarvamuste kohta ning baasi nendest lähtuvalt täiendada.



Joonis 10. ChatGPT vastuste erisused

## 4.3 Järgmised võimalikud sammud

ProgVestluse vestlusagent on aines “Programmeerimine” veel uus ja mitte eriti palju kasutust näinud lisandus. Seetõttu on ka palju külg, mida selle juures saaks edaspidi arendada. Kindlasti tuleks täiendada ProgVestlust uute koduülesannete lisamisega ja asendatava ülesande vastused kustutada. Hea oleks uurida üliõpilaste kasutajakogemust ja ProgVestluse mõju õpitulemustele. Selleks tuleks ehk lisada ProgVestluse agendile näiteks valik nuppudest nagu “Vajan veel abi” ja “Aitäh, sain hakkama” sarnaselt murelahendajale, et tänava valiku klikkimiste puhul monitoorida agendi kasulikkust. Nuppude juures peaks kindlasti säilima

võimalus teksti sisestada ja ise küsida. Monitooringu lahendus jääb aga selle mõelda, kes seda tegema hakkab. Tõenäoliselt on nupuvariandist kindlamaid ja kavalamaid lahendusi, mille üle autor siin edasi ei arutle. Veel oleks võimalik ProgVestlusesse ehitada sisse puu-süsteem, kus vestlusagent küsiks kasutajalt küsimusi vastu ja olenevalt kasutaja vastusest kuvaks vihjeid.

Administraatori kasutajakogemuse parendamiseks võiks administraatorikeskkonnas samuti mõningaid mugandusi teha. Esiteks võiks lisada vihjete tabelile pealkirjade või nädalate järgi filtreerimise võimaluse. Praegu on tabelisse uue vastuse sisestamise nupp tabeli all, mis eeldab, et iga kord keritakse lehekülje allosa. Vastuse lisamise nupu võiks aga tõsta ülespoole ning see muudaks juba sisestusprotsessi oluliselt mugavamaks. Veel võiks sisestustele lisada eelvaate võimaluse, et iga kord, kui mingi lausearvutuse vastust vaadata soovitakse, ei peaks avama muutmisrežiimi ja jälle lehel üles-alla kerima. Vastuse sisestamisel võiks olla linnukesega valik “lisa järgmine sama nädala vihje”, mis siis, kui on valitud, parasjagu sisestamisel oleva vihje salvestamisel jätab lisamise akna lahti nii, et nädal (ja võib-olla ka pealkiri) on juba eelneva vihje informatsiooniga täidetud. Näiteks, kui sisestatakse parasjagu seitsmenda nädala vihjet ja pealkiri on “Ül 7.1”, siis need väljad jäävad täidetuks, eelnev vastus salvestub ja järgmise lisamiseks on vaja vaid kirjutada lausearvutusvalem ja selgitus.

Kasutajatel võib tekkida küsimusi, millele baasis vastust ei ole, mistõttu oleks üks võimalik edasiarendus ka inimabi kaasamine. Näiteks oleks võimalik edastada küsimus kuhugi õppejõududele ühiselt ligipääsetavasse süsteemi ja neist kiireim saaks küsimusele vastuse anda ja see saadetak üliõpilase e-mailile. Või edaspidi täiendada selliselt, et ProgVestlus näiteks juhul, kui ta ise ei oska küsimusele vastata, teeb päringu mõnesse välisesse vestlussüsteemi ning kasutab saadud vastust oma vastuse formuleerimisel. Samuti võib ta teha päringuid Vikipeediasse või otsingumootoris faktide otsimiseks, mida näiteks ChatGPT ei tee. Kuna ProgVestlusel on kasutusel lausearvutusvalemid, kus peab arvestama mitmeid erijuhte, kuidas kasutaja võiks millegi kohta küsida, siis veel üks võimalik lisandus, mida programmile juurde panna saaks, oleks näiteks sünonüümisõnastiku tugi. See teeks lausearvutusvalemid lühemaks, nende kirjapaneku lihtsamaks ja aitaks katta rohkemaid olukordi. Sünonüümide kasutamisel tuleb muidugi arvestada juhte, kus programm võib hakata sisendit hoopis teemaväliselt või valesti tõlgendama. Kui mitte eeldada kasutajate korrektset keelekasutust, oleks kasulik ka lisada programmile võimekus toimida olukordades, kus kasutaja sisestab küsimuse täpitähtedeta, kui neid tegelikult vaja oleks. Funktsioonide



kohta nende nimedega abi küsimiseks võiks lisada ProgVestlusele võimekuse tegeleda sisendite ja lausearvutusvalemitega, mille komponendid sisaldavad alakriipsu. Alakriipsu toe lisamine aga eeldab süsteemi suuremat ümberehitamist, kuna lemmatiseerimine on sisemiselt osaliselt alakriipsudele üles ehitatud. Viimane siin väljapakutav täiendus oleks lisada võimalus faili importimiseks. Kui esialgu mustandina luukse kõik andmestik arvutustabelisse, siis selle asemel, et kõike ükshaaval üle kanda, oleks mugav, kui saaks lihtsalt näiteks nädala kaupa materjali importida. Ehk võiks ka küsija saada importida oma programmifaili ning selle kohta abi küsida ja süsteem näitaks ette programmis esinevad vead.

## 5. Kokkuvõte

Vestlusagendid on olnud pidevas arengus juba pea rohkem kui 60 aastat, alates esimese juturoboti ELIZA loomisest ja ajast, mil arvutid said piisavalt võimekaks. Tehisintellekti on hariduses kasutatud 1970. aastatest ja juturoboti mugav ning interaktiivne kasutus motiveerib õpilasi teadmisi omandama.

Bakalaureusetöö eesmärk on luua terve kursuse ulatuses küsimusi, vihjeid ja selgitusi uue vestlusagendi ProgVestlus andmebaasi seoses kursusega “Programmeerimine”. ProgVestlus on spetsiaalselt 2022. aastal Tartu Ülikoolis välja töötatud vestlusagent, mille eesmärk on täita murelahendajatele sarnast eesmärki. Erinevalt murelahendajatest on ProgVestlusel nii mõnedki eelised ja seetõttu on ta mugavam alternatiiv tudengite kodutööde lahendamise jooksul korduma kippuvatele küsimustele vastuste andmiseks. Vestlusroboti ProgVestlus programm töötab lausearvutuse põhimõttel. Eeldefineeritud vastused ja reeglipõhise mudeli kasutus teevad ProgVestlusest algelise ja valdkonnaspetsiifilise vestlusroboti.

Töö raames loodi vihjetekomplekte 13 nädala kodutöödele ja kogu kursuse õppematerjalile (160+ terminit, mõistet ja abistavat vihjet). Igal nädalal on kaks kuni neli ülesannet, mis tähendab, et lisaks kursuse teoreetilisele materjalile analüüsiti 33 koduülesannet ja mõnda lisaülesannet. Analüüsiprotsessi käigus loodi mustandiks Google Sheetsi arvutustabel, mainitud vihjetekomplektid ja mõistete selgitused ning mõningatel juhtudel uuriti ChatGPT poolt pakutavaid lahendusi. Töös anti ka ülevaade tehisintellekti ja vestlusagentide olemusest, kirjeldati vestlusagentide klassifitseerimist ja disainitehnikaid ning anti kokkuvõtlik kronoloogiline ülevaade kuulsamatest ning tänapäeval aktuaalsematest vestlusagentidest. Lisaks võrreldi vestlusrobotit ProgVestlus Murelahendajate ja ChatGPT’ga ning koostati juhend vihjete sissekandmiseks.

ProgVestlusel on veel mitmeid edasiarenduse võimalusi, mis tähendab, et see ei ole veel saavutanud oma täispotentsiaali. Algajatele programmeerimise õppijatele võiks ProgVestlus eelkõige kodutööde lahendamisel pakkuda palju tuge ja motivatsiooni õpinguid jätkata.

## 6. Viidatud kirjandus

Adamopoulou, E., Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2.

<https://doi.org/10.1016/j.mlwa.2020.100006> (06.05.2023)

Alice chatbot wins for third time. (20.09.2004). *BBC News*.

<http://news.bbc.co.uk/1/hi/technology/3672424.stm> (12.04.2023)

Ashford, W. (09.06.2014). Turing Test 2014 marks historic milestone in artificial intelligence. *ComputerWeekly.com*.

<https://www.computerweekly.com/news/2240222178/Turing-Test-2014-marks-historic-milestone-in-AI> (06.05.2023)

Bansal, H., Khan, R. (2018). A Review Paper on Human Computer Interaction. *International Journal of Advanced Research in Computer Science and Software Engineering*. 8. 53-56.

[https://www.researchgate.net/publication/325534924\\_A\\_Review\\_Paper\\_on\\_Human\\_Computer\\_Interaction](https://www.researchgate.net/publication/325534924_A_Review_Paper_on_Human_Computer_Interaction) (02.04.2023)

Bassett, C. (2019). The computational therapeutic: exploring Weizenbaum's ELIZA as a history of the present. *AI & Society*, 803–812.

<https://link.springer.com/article/10.1007/s00146-018-0825-9> (04.05.2023)

Bosker, B. (06.12.2017). SIRI RISING: The Inside Story Of Siri's Origins -- And Why She Could Overshadow The iPhone. *Huffpost*.

[https://www.huffpost.com/entry/siri-do-engine-apple-iphone\\_n\\_2499165](https://www.huffpost.com/entry/siri-do-engine-apple-iphone_n_2499165) (06.05.2023)

Burns, E., Laskowski, N., Tucci, L., Lawton, G. (Märts 2023) What is artificial intelligence (AI)? *Techtarget*.

<https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence> (06.05.2023)

Clifford, C. (24.05.2017). 4 keys to launching a successful business, according to this entrepreneur who sold Siri to Steve Jobs. *CNBC*.

<https://www.cnn.com/2017/05/24/keys-to-a-successful-business-from-a-guy-who-sold-siri-to-steve-jobs.html> (06.05.2023)

Copeland, J. (10.08.2014) Why Eugene Goostman Did Not Pass the Turing Test. *Huffpost*.

[https://www.huffingtonpost.co.uk/jack-copeland/turing-test-eugene-goostman\\_b\\_5478112.html](https://www.huffingtonpost.co.uk/jack-copeland/turing-test-eugene-goostman_b_5478112.html) (06.05.2023)

Hanh, D. T. M., Van Hung, T., Thanh, N. T. (2022). Chatbot in Education: Experience in Teaching Programming for Freshmen at University. *International Journal of Science and Research*, 11, 8.

<https://www.ijsr.net/archive/v11i8/SR22814071943.pdf> (06.05.2023)

Heaven, W. D. (03.03.2023). The inside story of how ChatGPT was built from the people who made it. *MIT Technology Review*.

<https://www.technologyreview.com/2023/03/03/1069311/inside-story-oral-history-how-chatgpt-built-openai/> (06.05.2023)

Hollo, K. (2016). Programmeerimise e-kursusel osalejate küsimuste analüüs ja selle põhjal murelahendajate koostamine. TÜ arvutiteaduste instituudi bakalaureusetöö.

[https://comserv.cs.ut.ee/ati\\_thesis/datasheet.php?id=53218](https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=53218) (06.04.2023)

How to Use ChatGPT to Create Dataset 2023 [Updated]? (2023). *ImgReverse*.

<https://imgreverse.com/blog/how-to-use-chatgpt-to-create-dataset-2023/> (06.05.2023)

Hussain, S., Ameri Sianaki, O., Ababneh, N. (2019). A Survey on Conversational Agents/Chatbots Classification and Design Techniques. *Web, Artificial Intelligence and Network Applications*, 946–956, Springer, Cham.

[https://link.springer.com/chapter/10.1007/978-3-030-15035-8\\_93](https://link.springer.com/chapter/10.1007/978-3-030-15035-8_93) (05.05.2023)

Johnson, B. (06.05.2023). How Siri Works. *HowStuffWorks*.

<https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/siri2.htm> (06.05.2023)

Kaimre, J. (2021). Murelahendajate koostamine Tartu Ülikooli kursuse „Programmeerimine“ jaoks. TÜ arvutiteaduste instituudi bakalaureusetöö.

[https://comserv.cs.ut.ee/ati\\_thesis/datasheet.php?id=71614](https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=71614) (06.05.2023)

Kay, J. (2015) Whither or wither the AI of AIED? *Workshop Proceedings*, 4, 85, 1 -h 10.

[https://ceur-ws.org/Vol-1432/ai\\_ed\\_pap1.pdf](https://ceur-ws.org/Vol-1432/ai_ed_pap1.pdf) (06.05.2023)

Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., Das, T. (2015). A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence. *International Journal of u- and e- Service, Science and Technology*, 8, 7, 277-284.

[http://article.nadiapub.com/IJUNESST/vol8\\_no7/28.pdf](http://article.nadiapub.com/IJUNESST/vol8_no7/28.pdf) (20.12.2022)

Kinnunen, P. (2009). Challenges of teaching and studying programming at a university of technology - Viewpoints of students, teachers and the university.

<http://urn.fi/URN:ISBN:978-952-248-195-5> (06.05.2023)

Kinos, T. (2019) Ethics of AI: Preserving and growing human purpose. *Getjenny*.

<https://www.getjenny.com/blog/ethics-of-ai> (06.05.2023)

Kumar, C. (22.02.2023) 9 Tools to Create Your Personal or Business Chatbot. *Geekflare*.

<https://geekflare.com/create-chatbot/> (08.05.2023)

Kursuse „Programmeerimine“ koduleht (2022).

<https://courses.cs.ut.ee/2022/programmeerimine/fall> (06.12.2022)

Lepp, M., Palts, T., Luik, P., Papli, K., Suviste, R., Säde, M., Hollo, K., Vaherpuu, V., Tõnisson, E. (2018). Troubleshooters for Tasks of Introductory Programming MOOCs. *International Review of Research in Open and Distributed Learning*, 19, 4.

<http://www.irrodl.org/index.php/irrodl/article/view/3639/4728> (06.05.2023)

Masche, J., Le, N.-T. (2011). A Review of Technologies for Conversational Systems. *Advances in intelligent systems and computing*, 212-225.

[https://link.springer.com/chapter/10.1007/978-3-319-61911-8\\_19](https://link.springer.com/chapter/10.1007/978-3-319-61911-8_19) (05.05.2023)

Okonkwo, C. W., Ade-Ibijola, A. (2021) Chatbots applications in education: A systematic review. *Computers & Education: Artificial Intelligence*, 2, 100033.  
<https://doi.org/10.1016/j.caeai.2021.100033> (06.05.2023)

Oppt, G., Dowe, D. (2021). The Turing Test. *Stanford Encyclopedia of Philosophy*.  
<https://plato.stanford.edu/entries/turing-test/> (02.04.2023)

Ortiz, S. (29.03.2023). Microsoft increases Bing chat limit -- again. *Zdnet*.  
<https://www.zdnet.com/article/microsoft-increases-bing-chat-limit-again/> (26.03.2023)

Programmeerimine (2022). *Tartu Ülikooli õppeinfosüsteem*  
<https://ois2.ut.ee/#/courses/LTAT.03.001/version/700527cd-b682-45a8-254a-b3eaa4a04a0b/details> (06.05.2023)

Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K. (2017). A Survey of Design Techniques for Conversational Agents. *Information, Communication and Computing Technology*, Springer, Singapore, 336–350.  
[https://link.springer.com/chapter/10.1007/978-981-10-6544-6\\_31](https://link.springer.com/chapter/10.1007/978-981-10-6544-6_31) (03.04.2023)

Rusen, C. A. (13.03.2023) Bing Chat and ChatGPT come to Windows: Here's how they work. *Digital Citizen*.  
<https://www.digitalcitizen.life/bing-chat-chatgpt-windows/>. (06.05.2023)

Schofield, J. (08.06.2014). Computer chatbot 'Eugene Goostman' passes the Turing test. *Zdnet*.  
<https://www.zdnet.com/article/computer-chatbot-eugene-goostman-passes-the-turing-test/> (22.04.2023)

Zhao, W., Alwidian, S. A., Mahmoud, Q. H. (2022). Adversarial Training Methods for Deep Learning: A Systematic Review. *Algorithms*, 15(8), 283.  
<https://doi.org/10.3390/a15080283> (03.03.2023)

Vijaya Saradhi, M.V., Swajan Reddy, G., Arun Reddy, C., Saikumar, C., Adityan, T. (2022). Survey on Chatbot Classification and Technologies. *International Research Journal of Engineering and Technology (IRJET)*, 9, 11, 126-131.

<https://www.irjet.net/archives/V9/i11/IRJET-V9I11118.pdf> (06.05.2023)

Wallace, R.S. (2009). The Anatomy of A.L.I.C.E. *Parsing the Turing Test*, 1-35, Springer, Dordrecht.

[https://link.springer.com/chapter/10.1007/978-1-4020-6710-5\\_13](https://link.springer.com/chapter/10.1007/978-1-4020-6710-5_13) (05.05.2023)

Weizenbaum, J. (1966). ELIZA A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Communications of the ACM*, 9, 1, 36–45.

<https://dl.acm.org/doi/10.1145/365153.365168> (02.02.2023)

# Lisad

## I Link vestlusagendi algandmestikule

Link Google Sheetsis mustandmaterjali hoiustamiseks loodud arvutustabelile:

<https://docs.google.com/spreadsheets/d/1rxAk5rljaIIINupgMUxn7WoByMb3wQElvI5MgCfs0OY/edit#gid=872166197>



## II Võrdlustabel kodutööde küsimustest

Tabel 1. Autori ja ChatGPT kodutööde “3.3 Moosi keetmine”, “7.2 Takso hinnad” ja “10.1 Teksti analüüs” küsimuste võrdlus.<sup>6</sup>

Autor	ChatGPT
<b>3.3 Moosi keetmine</b>	
1) Kuidas defineerida funktsioon moos ja mis on tema argumendid? 2) Kui palju karpe mul kasutada on? 3) Kuidas arvutada vajaminevate karpide arvu? 4) Kuidas ma tean, et moos ei mahu karpidesse?	1) Kuidas saab funktsiooni kirjutada selliselt, et see lahendab antud ülesande? 2) Kuidas saab funktsiooni loogikat kõige paremini struktureerida ja optimeerida? 3) Kuidas saab kontrollida, kas funktsioon tagastab õigeid tulemusi? 4) Kuidas saab käsitleda olukorda, kus funktsioon tagastab -1, st moos ei mahu antud karpidesse? 5) Kuidas saab funktsiooni üldistada selliselt, et see toimiks ka siis, kui kasutada on rohkem kui kahte suurust karpide karpi?
<b>7.2 Takso hinnad</b>	
1) Kuidas küsida kasutajalt ainult üks sisend? 2) Kuidas failist andmeid lugeda? 3) Kuidas töödelda loetud ridu erinevateks andmeteks? 4) Kuidas arvutada iga takso hind? 5) Kuidas leida minimaalne takso hind ja selle takso nimi?	1) Kuidas lugeda failist sisendit? Kas pean kasutama spetsiaalset funktsiooni, et lugeda failist ridu või saan kasutada sisemisi funktsioone, näiteks sisseehitatud funktsiooni "open" ja "read"? 2) Kuidas teisendada tekstikujul olevaid hindu ujukomaarvudeks? Kuidas teha kindlaks, kas sisestatud väärtus on numbriks? Kas on vaja rakendada vigade

<sup>6</sup> Üliõpilane on saanud vastuse ChatGPT abiga (13.03.2023), s.o kasutades keelemudelit, mille väljaõpe põhineb suurel hulgal erinevatel tekstiallikatel. ChatGPT on välja töötatud OpenAI poolt. Lisateavet ChatGPT ja OpenAI kohta: <https://openai.com>. Mudelile esitatud sisend: "Millised tehnilised küsimused võivad tekkida algajal programmeerijal järgmist ülesannet lahendades: <ülesandeTekst>"

	<p>käsitlemist, kui kasutaja sisestab vale tüüpi andmeid?</p> <p>3) Kuidas võrrelda hindu ja valida kõige odavam variant? Kas peaksin kasutama loogilisi operaatoreid nagu "if" või saan kasutada mõnda muud programmi kontrollstruktuuri? Kuidas võin tagada, et programmi väljund vastab ootustele?</p> <p>4) Kuidas tagada, et programm toimib korrektselt, kui failis on vigu? Kuidas käsitleda vigu, kui mõni rida on valesti vormindatud või kui faili ei leita?</p>
<b>10.1 Teksti analüüs</b>	
<p>1) Kuidas defineerida funktsioon erinevad sümbolid ja mis on selle argument?</p> <p>2) Kuidas sõne sümboleid hulka lisada?</p> <p>3) Kuidas defineerida funktsioon sümboleid sagedus ja mis on tema argument?</p> <p>4) Kuidas luua tühja sõnastikku?</p> <p>5) Kuidas loendada sümboleid esinemist?</p> <p>6) Kuidas defineerida funktsioon grupeeri ja mis on tema argument?</p> <p>7) Kuidas lisada sõnastikku võtit, millele vastab hulk?</p> <p>8) Kuidas hoiustada sümboleid sagedust?</p> <p>9) Kuidas saab sõnastikust kirje kätte?</p> <p>10) Kuidas kontrollida sümboli kuulumist häälikute hulka?</p> <p>11) Kuidas lisada sõnastiku võtme hulga väärtusi?</p>	<p>1) Kuidas luua funktsiooni, mis leiab kõigi antud sõnes leiduvate erinevate sümboleid hulga?</p> <p>2) Kuidas luua funktsiooni, mis leiab sõnest kõik tähed ja nende esinemissagedused?</p> <p>3) Kuidas luua funktsiooni, mis grupeerib sõnes täishäälikud, kaashäälikud ja muud sümboleid erinevatesse hulkadesse?</p> <p>4) Kuidas kasutada sõnastikke funktsioonides ja kuidas neid täita?</p> <p>5) Kuidas eristada suuri- ja väiketähti sõnades?</p>

### III ChatGPT lahenduse ja näidislahenduse võrdlus

```
python Copy code

aastatulu = float(input("Sisesta aastatulu: "))

if aastatulu <= 6000:
    maksuvaba_tulu = aastatulu
elif aastatulu <= 14400:
    maksuvaba_tulu = 6000
else:
    maksuvaba_tulu = 6000 - 6000 / 10800 * (aastatulu - 14400)
    if maksuvaba_tulu < 0:
        maksuvaba_tulu = 0

print("Maksuvaba tulu on " + str(round(maksuvaba_tulu, 2)) + " eurot.")
```

Joonis 11. ChatGPT poolt kirjutatud lahendus kodutööle 2.4

```
'''
Kirjuta programm, mis küsib kasutaja aastatulu ja arvutab ning
väljastab ekraanile maksuvaba tulu ümardatuna kahe kohani pärast koma.
'''

# Küsime kasutajalt aastatulu.
sisend = input("Sisesta aastatulu: ")

# Kuna sisestus on algselt tekstiline, teisendame ta ujukomaarvuks.
aastatulu = float(sisend)

# Kontrollime, millisesse vahemikku aastatulu kuulub
# ja vastavalt arvutame maksuvaba tulu.
if aastatulu <= 6000:
    maksuvaba = aastatulu
# Et teise võrdlusesse jõutakse ainult siis, kui esimene
# ei kehti, pole vaja kontrollida 6000-st suurem olemist:
elif aastatulu <= 14400:
    maksuvaba = 6000
elif aastatulu <= 25200:
    maksuvaba = 6000 - 6000 / 10800 * (aastatulu - 14400)
else:
    maksuvaba = 0

print("Maksuvaba tulu on", round(maksuvaba, 2), "eurot.")
```

Joonis 12. Kursusel esitletud näidislahendus kodutööle 2.4

## IV Tagasisideküsitlus

Link tagasisideküsitlusele: <https://forms.gle/JzRZNO5WcZGgCQT58>

“Tere!

Olen TÜ informaatikatudeng Maike Kadaja. Minu bakalaureusetöö tegeleb loodava vestlusagendiga (*chatbot*) ProgVestlus. Ilmselt oled tuttav juba käesoleval või mõnel muul kursusel kasutusel olevate murelahendajatega. ProgVestlus proovib sarnaselt murelahendajale tudengit aidata kodutööde lahendamisel. Praegu ta veel ilmselt Sulle suureks abiks sellel kursusel ei ole, aga loodan, et ta saab aidata tudengeid järgmisel sügisel ainek "Programmeerimine" osavõtjaid.

Selleks, et ProgVestlus saaks võimalikult kasulikuks ja kasutajasõbralikuks, palun Sul ära täita see küsimustik.

**Enne küsimustiku täitmist** (või selle ajal) **proovi vestlusagenti ProgVestlus** siin lingil: <http://prog.keeleressursid.ee/progvestlus/index.php>

ProgVestlus saab selgitada mõisteid ja termineid, millega algaja programmeerija tutvust teeb ja annab vastuseid ka koduülesannete puhul tekkivate küsimuste kohta. Meenuta lihtsamaid Pythoni käskluseid ja mõisteid, küsi küsimusi ning vaata, mida ta Sulle vastab.

Siia küsimustikku palun Sul kirjutada mõtted, mis ProgVestluse vestlusagenti kasutades pähe tulevad.

Küsimustik on anonüümne ja täitmine võib võtta 10-15 minutit. Mida põhjalikum on Su vastus, seda rohkem on sellest abi.

Kui Sulle pakuks ProgVestluse käekäik huvi või oleksid huvitatud sellega (näiteks oma bakalaureusetöös) tegelema, saad küsitluse lõppu jätta oma e-maili.

Aitäh! :)“

Küsitluses küsitud küsimused olid järgmised:

1. Milliseid küsimusi/märksõnu ProgVestluses katsetasid? Milliseid vastuseid ootasid? Milliseid vastuseid said?

2. Kuidas võiks ProgVestlust edasi arendada ja väärtuslikumaks muuta?
3. Millises olukorras Sa ProgVestlust kasutaksid?
4. Mis Sulle ProgVestluse puhul meeldis? Mis mitte?
5. Võrdle ProgVestlust murelahendajatega. Mis on Sinu meelest kursustel kasutamisel kummagi plussid ja miinused?
6. Võrdle ProgVestlust ChatGPT'ga. Mis on Sinu meelest kursustel kasutamisel kummagi plussid ja miinused?

Tahaksin veel öelda, et...

## V Tulemuste tabel

Tabel 2. Nädalate teemade vastavate vihjete arvud

Nädal, teema	Koduülesannete vihjeid	Mõisteid, termineid, vihjeid	Nädala vihjete arv kokku*
1. Muutuja ja avaldis	9	10	12
2. Tingimuslause	9	10	16
3. Funktsioon	12	13	25
4. Korduslause	12	19	31
5. Lihtsam failitöötlus, sõned	12	17	29
7. Järjend	20	14	34
8. Järjend II	17	14	28
9. Kahekordne tsükkel, failitöötlus	9	9	17
10. Andmestruktuurid	18	25	43
11. Andmestruktuurid II	16	1	17
13. Rekursioon	9	5	14
14. Objektorienteeritud programmeerimine	19	8	29
15. Objektorienteeritud programmeerimine II	14	7	19

+ 10 veateadete kirjeldust ja kõikide ülesannete sissejuhatavad vihjed.

Ligikaudu 340 vihjet.

\*Märkus: Osade ülesannete vihjed ja mõisted on kattuvad.

## VI Juhend ProgVestlusesse abimaterjali lisamiseks

Järgnevalt on kirjas punktid, mida on soovitatav järgida, kui on soov ProgVestluse keskkonda koduülesannete vihjeid lisada.

1. Töötada läbi käesoleva nädala materjal (kui eelnevate nädalate kohta pole ProgVestluses sissekandeid, siis tasub ka eelnevad nädalad üle vaadata).
2. Lahendada koduülesanded tuginedes kursusel õpitule.
3. Logisid saab vaadata administraatori lehe kõige alumisel “Näita logi” nupul klikates.
  - a. Kui logides on piisaval hulgal materjali, siis sealt saab korduma kippuvad küsimused välja võtta. Nende grupeerimise näiteks võib võtta Kaspar Hollo 2016. aasta bakalaureusetöö.
4. Soovitatavalt kasutada arvutustabelit vihjete, lausearvutusvalemite ja vastuste hoiustamiseks (parem ülevaade ja kergem muudatusi teha).
5. Sisestada uued mõisted, terminid ja vihjed arvutustabelisse. Selleks:
  - a. Mõelda välja erinevad viisid, kuidas selle termini kohta küsida võidaks (inglise keeles, eesti keeles, ilma alakriipsudeta).
  - b. Koostada neist märksõnadest lausearvutusvalem.
  - c. Kirja panna selgitus ja/või vihje vastuseks. See kuvatakse lõpptulemuses kasutajale.
6. Sisestada lausearvutusvalemid ja vastused ProgVestlus keskkonda. Selleks:
  - a. Logida administraatorina sisse.
  - b. Kerida juba olemasolevate sisestuste tabeli alla ja klikata nupul “[Lisa uus vastus]”.
  - c. Pealkirjaks panna sobiv kokkuvõttev pealkiri, nt “Programmeerimise mõisted”.
  - d. Sisestada vastus soovitud vormistusega selleks ettenähtud tekstiväljale. NB! Kui kopeeritakse teksti, tuleb kindlasti üle vaadata, et stiilid jm kaasa ei tuleks. Selleks on hea kleepida tekst aknasse, mis avaneb, kui klikata nupul “<” lähtekood.
  - e. Lausearvutusvalemi tekstiväljale kirjutada lausearvutusvalem.
  - f. Tekstiväljale “Nädal” kirjutada nädala number, milles seda vastust anda soovitakse.
  - g. Lisamiseks, vajutada nupul “Sisesta vastus andmebaasi”.

7. ProgVestluses sissekannete muutmiseks on iga rea lõpus ringikujuline nupp, mille sees on noolemärk. Kustutamiseks on ristikujuline nupp.
8. Nädalaid saab defineerida selleks ettnähtud jaotises vastuste jaotise järel. Selleks:
  - a. Klikata nupul "Lisa uus nädal".
  - b. Sisestada nädala number ja selle alguse- ja lõpukuupäev.
  - c. Salvestamiseks klikata nupul "Lisa nädal andmebaasi", tühistamiseks nupul "Tühista".
9. ProgVestluse lingi võib lisada iga nädala kodutöö juurde eraldi või kursuse lehele.



## **V Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Maike Kadaja,

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

Maike Kadaja

Tartu, 09.05.2023