

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Brigit Parra

**Automaatkontrollide loomine kursuse „Introduction
to Programming” praktikumiülesannetele**

Bakalaureusetöö (9 EAP)

Juhendaja:
Reelika Suviste, PhD

Tartu 2025

Automaatkontrollide loomine kursuse „Introduction to Programming” praktikumiülesannetele

Lühikokkuvõte:

Automaatkontrollid muudavad lahenduste hindamise kiiremaks ning vähendavad õpetajate töökoormust. Nii jääb õpetajatel aega parendada kursust üleüldiselt, täiendades õppematerjale ning keskenduda õppekvaliteedile. Käesolevas bakalaureusetöös tehti automaatkontrolle kursuse „Introduction to Programming” praktikumiülesannetele. Kokku loodi 42 ülesannet, millest 39 on automaatkontrollitavad. Need loodi dünaamilise hindamise põhimõttel. Automaatkontrollide kvaliteeti hinnati eksperdi tagasiside alusel ning testiti ka varasemalt kursust läbinud üliõpilaste lahenduste põhjal. Testimist viidi läbi juhuslikkuse alusel ehk valiti iga ülesande jaoks juhuslikult viis lahendust ning testiti neid automaatkontrollis. Saadud tulemusi analüüsiti ning selgus, et põhilised vead esinesid näidete mittejärgimises ning loovusel (automaatkontroll ei ole sellele paindlik).

Võtmesõnad: Automaatkontroll, programmeerimine, keskkond Lahendus, Type Specific Language

CERCS: P175 Informaatika, süsteemiteooria, S281 Arvuti õpiprogrammide kasutamise meetodika ja pedagoogika

Creating automatic checks for „Introduction to Programming” course practicum exercises

Abstract:

Automated assessments make solution evaluation faster and reduce the workload of teachers. This allows educators to focus on improving the course overall by enhancing learning materials and maintaining the quality of education. In this bachelor's thesis, automated assessments were implemented for the “Introduction to Programming” practical exercises. A total of 42 tasks were created, of which 39 are assessable through automated checks. These tasks were designed based on the principles of dynamic assessment. The quality of automated assessments is evaluated based on expert feedback and tested using solutions submitted by students who had previously completed the course. Testing is conducted randomly, meaning that five solutions were randomly selected for each task and assessed using automated checks. The results were analyzed, revealing that the main errors occurred in failure to creativity (as automated assessments are not flexible in accommodating creative solutions) and following examples.

Keywords: Automated assessments, programming, Lahendus platform, Type Specific Language

CERCS: P175, Informatics, systems theory, S281 Computer-assisted education

Sisukord

Sissejuhatus	5
1. Automaatkontrollid	6
1.1 Olemasolevad automaatkontrollid	6
1.2 Dünaamiline ja staatiline hindamine	7
1.3 Automaatkontrollide kasulikkus	8
2. Tehisintellekt ja plagiaat	10
3. Programmeerimise kursus ja keskkond Lahendus	11
3.1 Programmeerimise kursus	11
3.2 Keskkond Lahendus	11
4. Automaatkontrollide loomise tööprotsess	16
4.1 Praktikumiülesannete tekst	16
4.2 Automaatkontrollide loomine	18
4.3 Tagasiside küsimustik	21
5. Loodud automaatkontrollide kvaliteedi analüüs	22
5.1 Eksperdi tagasiside	22
5.2 Üliõpilaste esitatud lahenduste analüüs	25
Kokkuvõte	30
Kasutatud allikad	31
Lisad	34
Lisa 1. Tagasiside küsimustik	34
Lisa 2. Praktikum 1	35
Lisa 3. Praktikum 2	36
Lisa 4. Praktikum 3	39
Lisa 5. Praktikum 4	41
Lisa 6. Praktikum 5	43
Lisa 7. Praktikum 6	45
Lisa 8. Praktikum 7	47
Litsents	49

Sissejuhatus

Tänapäeva ühiskond on kiires arengus ning töötatakse välja erinevaid tehnoloogiaid. Selleks, et leiduks keegi, kes oskab neid välja mõelda ja arendada on vaja vastavate teadmistega inimesi [1]. Neid teadmisi on võimalik õppida ise, kutsekoolis või kõrgkoolis; kahes viimases ollakse õppejõudude toetava pilgu all. Kuna õppijaid on rohkem kui õpetajaid, siis on viimastel vaja tuge, et kontrollida õppurite tehtud töid ning üheks mugavaks lahenduseks on rakendada ülesannetele automaatkontrolle.

Automaatkontrollide loomine muudab hindamise lihtsamaks ja ajakulu, mis manuaalsel hindamisel oleks kulunud, väheneb [2]. Seetõttu saavad õppejõud keskenduda teistele aspektidele, näiteks täiendada õppematerjale, parendada õpetamist üleüldiselt või vajadusel keskenduda individuaalsele lähenemisele. Käesoleva lõputöö tulem on suureks toeks eelmainitule.

Lõputöö eesmärk on luua automaatkontrollid ingliskeelse õppeaine „*Introduction to Programming* MTAT.03.236” praktikumiülesannetele ja viia ülesanded üle keskkonda Lahendus ning saadud tagasiside põhjal neid parendada. Loodud automaatkontrollid kontrollivad õpilase tehtud programmi korrektsust automaatselt, mis võimaldab õpilasel saada kohest tagasisidet loodud koodi kohta.

Lõputöö on jaotatud viieks peatükiks. Esimeses peatükis antakse ülevaade olemasolevatest automaatkontrollidest ja nende vajalikkusest ning kirjeldatakse dünaamilise ja staatilise hindamise erinevust. Tehisintellekti ja plagiaadi peatükis kirjutatakse tehisintellektist ning kuidas seda ära tunda ja paarist plagiaadituvastuse programmist. Kolmandas peatükis tutvustatakse kursust, mille jaoks automaatkontrolle tehakse ning keskkonda, mida selleks hakatakse kasutama. Tööprotsessi peatükis antakse ülevaade, mida töö raames tehti. Viimane peatükk analüüsib valminud automaatkontrollide kvaliteeti eksperdi tagasiside ning eelnevalt kursust läbinud üliõpilaste ülesannete lahenduste põhjal.

1. Automaatkontrollid

Selles peatükis antakse ülevaade, milliseid automaatkontrolle on tehtud, milliseid on võimalik teha ning mis programmeerimiskeeli kasutatakse enim automaatkontrollide loomiseks. Lisaks kirjeldatakse automaatkontrollide olulisust programmeerimises.

1.1 Olemasolevad automaatkontrollid

Hollingsworthi [3] sõnul tekkis vajadus automaatkontrollide järele juba 1960. aastate paiku, mil üha enam õpilasi alustasid õpinguid programmeerimise valdkonnas. Tema tehtud eksperimendi tulemusel nähti, et õpilased omandasid õpitud efektiivsemalt ning paremini, kasutades automaatkontrolli abi, kui õppides ilma kõrvalise abita. Ühed esimesed automaatkontrollid kontrollisid assembleris kirjutatud koodi [3], kuid hiljem on loodud ka teisi programme, platvorme ning keeli, kus on olemas automaatkontrolli loomise võimalus, näiteks ASSYST, MOOCi platvormid ja programmeerimiskeel C [4, 5].

ASSYST (*ASsessment SYSTem*) on graafiline liides [4]. Sellel liidesel on kaks vaadet, üks on kuvatud õpilasele ning teine õpetajale. Õpilane saab esitada oma töö hindamiseks ning õppejõul on võimalik seda vaadata ja vajadusel anda suuniseid. ASSYST programm hindab õpilase programmi õigsust, tõhusust, stiili, keerukust ja testandmete piisavust ning selle põhjal koostatakse tagasiside. Uurimuse autorid katsetasid erinevaid hindamissüsteeme ning leiti, et tagasiside ei pruugi alati olla õiglane, andes õpilasele täielikult korrektse lahenduse eest maksimumpunktid ja muudel juhtudel mitte ühtegi punkti. Seetõttu on ASSYST süsteemi kirjutatud skaalapõhine hindamine, et tagada töö õiglasem hindamine [4].

Olemas on ka MOOCi (*Massive Open Online Courses*) platvormid, mis toetavad mitmeid erinevaid keeli [5]. See platvorm sobiv hästi nii õpetamiseks, õppimiseks kui ka andmeteadusega seotud tööde tegemiseks, kuna võimaldab visualiseerida koodiga loodud graafikuid, luua tavateksti sisaldavaid dokumente ning reaajas koodi kirjutada. MOOCi platvormis on võimalik luua automaathindamisega ülesandeid ning seeläbi lihtsustada õpetajate tööd, kuna automaatika on võimeline hindama nii efektiivselt kui ka korrektselt [5].

Lisaks on võimalik teha automaatkontrolle programmeerimiskeeles C. Erika Baksáné Varga ning Antal Kristóf Fekete [6] tehtud uurimuse käigus loodi Ungari Ülikoolile kaks

automaatkontrollikeskkonda: üks neist on veebipõhine tööriist, mis on ungarikeelse GUI-ga¹ ning teine asub Ungari Ülikooli GitHubi keskkonnas ning õpetus, kuidas GitHubi klassiruumis kasutada, on kättesaadav GitHub Education veebilehel². Uurimuse tulemusel selgus, et GitHubi keskkonnas olev repositoorium (ingl *repository*) oli efektiivsem, sest selle kaudu oli kergem saada ülevaadet õpilaste tegevustest ning edusammudest.

1.2 Dünaamiline ja staatiline hindamine

Koodi on võimalik hinnata käsitsi, vaadates struktuuri, kirjapanekut ning saadavat väljundit. Lisaks on võimalik hindamiseks kasutada ka oma loodud programmi või juba olemasolevat ning seda täiendada nii, et see hindaks esitatud koodi õigsust, näiteks saadud väljundi alusel. Kõige kergem on teha seda kasutades automaatset hindamist. Automaatkontrollid jagunevad tööpõhimõtte järgi kahte kategooriasse: dünaamiline ning staatiline hindamine [1]:

- Dünaamilisel hindamisel põhinev automaatkontroll toimib järgnevalt: õpilane esitab omakirjutatud programmi automaatkontrolli ning järgmisena käivitatakse esitatud programm eelnevalt automaatkontrolli koostaja poolt määratud argumentidega [1]. Seejärel kuvab kompilaator õpilasele, kas kood kompileerus, vastasel juhul kuvab, mis koodi osa ei kompileerunud, näidates argumente, millega jooksutati ning oodatavat tulemust. Selline hindamine annab õppijale koheselt tagasisidet, mis läks valesti koos ebakorrekse reaga koodis. Saadud tagasiside põhjal saab õppur oma koodi parandada ning uuesti esitada kuniks programm läbib kõik etteantud testid.
- Staatilisel hindamisel põhinev automaatkontroll aga analüüsib koodi ning ei jookсутa seda argumentidega [1]. Selline kontrollimine annab informatsiooni süntaktiliste vigade, koodi stiili ning ka disainivigade kohta. Saadud tagasiside aitab õppida kirjutama puhast koodi³ ning korrektset ülesehitust, info aitab õppida puhast koodi kirjutama ning korrektset ülesehitust, mida on vaja teada suuremamahuliste programmide kirjutamisel. Puhas ja selge kood aitab paremini mõista, mida programm mingis kohas teeb.

¹ Graafiline kasutajaliides, ingl *Graphical User Interface*

² <https://education.github.com/case-studies>

³ Kood, mis on selge, arusaadav ning hallatav, ingl *clean code*

2024. aastal tehtud uuringus võrreldi dünaamilise hindamise kasulikkust traditsioonilise staatilise hindamisega [7]. Dünaamilise hindamise fookuses on üleüldine õppimisprotsess, mis staatilise hindamise juures võib jääda puudu. Areneb ka iseseisva õppimise protsess, kuna õppimine toimub järjepidevalt. Uuringu jaoks loodi kaks gruppi: esimeses grupis kasutati dünaamilise hindamise süsteemi ning teist gruppi õpetati traditsioonilisemal pedagoogilisel meetodil (staatiline hindamine), mis moodustas kontrollgrupi. 12-nädalase perioodi lõpuks selgus, et esimese grupi tulemused olid tunduvalt paremad kui kontrollgrupi omad. Kuigi uuring tehti vähese inimeste hulgaga, näitab see siiski dünaamilise hindamise kasulikkust õppijatele, arendades nende probleemide lahendamise oskust ning iseseisvust koodi kirjutamisel.

1.3 Automaatkontrollide kasulikkus

Automaatkontrollide kasulikkust saab hinnata erinevalt - nii juhendaja kui ka õppija vaatest. Juhendajate poolt on automaatkontrollid kasulikud selleks, et lihtsustada nende tööd ning vähendada ajakulu [2], kuna automaatkontrollid kontrollivad juhendaja eest õpilaste esitatud programmide väljundite õigsust. Õpetajatel on võimalik kontrollida ka automaatkontrolli tööd ning hinnata, kas programm on korrektselt vormistatud, milline on stiil ning kui arusaadavalt on kood kirjutatud. Esitatud lahenduste põhjal saab teha ka lahenduste analüüsi, mis annab kasulikku tagasisidet juhendajatele [8]. Analüüsist on võimalik näha õpilaste edasijõudmist ning ka individuaalset arengut. See annab juhendajale teada raskustest, millele võib rohkem tähelepanu juhtida ning ka nõu anda. Lisaks on võimalik tuvastada abi vajavaid õpilasi, kes võib-olla ise ei julgeks abi otsida või küsida.

Automaatkontrollid ei ole ainult juhendajatele kasulikud, vaid ka õpilastele, kuna nad saavad kohest tagasisidet oma koodi kohta [2]. Õpilasel on võimalik korduvalt esitada oma programm kontrollimiseks ning veateadete või ebakorreksete vastuste saamisel (mis erinevad oodatud vastusest) on õpilasel võimalus oma programmi täiendada või muuta. Nad saavad tegutseda omas tempos, mis hoiab motivatsiooni õppimiseks [9]. Areneb ka programmeerimisoskus ja probleemide lahendamise oskus [10].

Kokkuvõttes on automaatkontrollid kasulikud nii juhendajale kui ka õppijale, kuid võib esineda ka kahjulikke külgi. Testid võivad kontrollida koodi erinevate sisenditega või äärejuhtudega, mille peale koodi kirjutaja ise ei oleks tulnud, kuid korduv ebaõnnestumine

võib muuta õpilase kannatamatuks [8]. Sellises olukorras võib juhtuda, et õpilane keskendub automaatkontrollist läbi saamiselening ei süvene detailidesse - miks kood just nii töötab, miks lahendus ei sobinud või mida sellest õpiti. Lisaks võib kaduda otsene suhtlus juhendajaga, mis mõnele õppurile on vajalik, et teemast paremini aru saada.

2. Tehisintellekt ja plagiaat

Viimastel aastatel on tehisintellekt kiiresti edasi arenenud. Tehisintellekti ehk AI⁴ kasutamisel on positiivseid külgi, kuid on ka negatiivseid. Õpilane, kes ei mõista teemat piisavalt hästi, ei oska eristada õiget infot valest [11]. Saadud info võib olla ebaadekvaatne, aegunud või kallutatud. Esile kerkib ka plagiaat, mis on õppimisprotsessi vältel probleem [11]. AI kasutust on peale vaadates keeruline tuvastada, kuid seal on kindlad jooned, mis välja paistavad, näiteks muutujate nimed [12].

Kõige märgatavam viis, kuidas ära tunda tehisintellekti loodud koodi, on vaadata kommentaare. Inimese poolt kirjutatud kommentaarid on struktureeritumad, põhjalikumad, detailsemad ning ka loomulikumad [13]. Samas kui AI loodud kommentaarid on tihtipeale n-ö kohatäite funktsioonis. Need kommentaarid on tihtipeale pinnapealsed, võttes koodi tegevuse kokku ilma sügavamat põhjendust andmata. Ühes varasemas uurimuses [13] toodi välja ka huvitavad mustrid nimetamisreeglites, kuna need on tihtipeale lihtsakoelised ning prioriteediks on selgus, jättes välja loovad või valdkonnaspetsiifilised nüansid, mida inimarendajad nimedes võivad kasutada.

On loodud erinevaid sarnasuse-tuvastajaid, näiteks MOSS ning SSTRANGE [12]. MOSS (*Measure Of Software Similarity*) on 1994. aastal loodud ning 1997. aastal välja antud automaatne süsteem, mis teeb selgeks sarnasuse erinevate programmide vahel [14]. SSTRANGE on sarnasuse tuvastaja, mis tegeleb tõestusmaterjalide kogumisega [15]. See programm toetub süntaktilistele sarnasustele ning seletab kõik leitud sarnasused inimesele arusaadavas keeles. Lisaks on ChatGPT looja OpenAI on valmis teinud AI teksti klassifikaatori, mis kategoriseerib etteantud teksti viite kategooriasse: väga ebatõenäoliselt, ebatõenäoliselt, ebaselge, võimalik, tõenäoliselt, et on AI loodud [16].

Tehisintellekt on võrdlemisi uus asi (ChatGPT tuli laialdasemalt kasutusele 2023. aasta kevadel) ning seda alles õpitakse kasutama tulemuslikult, eesmärgipäraselt ning eetiliselt. Luuakse erinevaid viise ning uuritakse erinevaid meetodeid, kuidas aru saada, kas tekst on loodud õpilase poolt või hoopiski on selle teinud AI.

⁴ ingl k *artificial intelligence*

3. Programmeerimise kursus ja keskkond Lahendus

Selles peatükis tutvustatakse lähemalt, mis kursusele hakatakse automaatkontrolle looma ning mis keskkonda selleks kasutatakse.

3.1 Programmeerimise kursus

Programmeerimise oskus ning algoritmiline mõtlemine on vajalik mitmetes erinevates valdkondades. Seda on võimalik õppida nii iseseisvalt kui ka erinevaid kursuseid läbides. Astudes ülikooli, peab lähtuma kohustuslike ainete ning eeldusainete nõuetest. Mitmetes õppekavades on programmeerimise algkursuse läbimine kohustuslik.

Tartu Ülikool pakub kursust nimega „*Introduction to Programming*”, mis on peamiselt mõeldud infotehnoloogiaõiguse, loodusteaduse ja tehnoloogia, matemaatilise majandusteaduse, ettevõtetmajanduse, linnastunud ühiskonna geoinformaatika ning innovatsiooni ja tehnoloogia juhtimise erialade üliõpilastele [17]. Lisaks võivad kursust läbida ka teiste erialade üliõpilased, kes soovivad programmeerimist õppida.

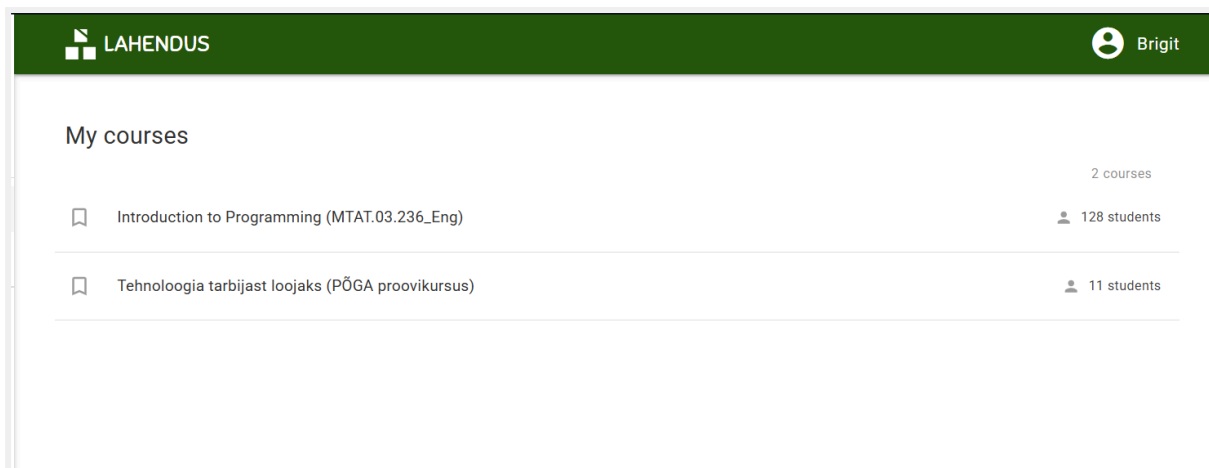
Kursus kestab 8 õppenädalat ning õppeaine maht on 3 EAP⁵-d. Õppeperioodi jooksul õpitakse nii avaldise, loendeid, tingimuslauseid, funktsioone, tsükleid, sõne- ja failitöötlust kui ka graafikat [17]. Õppeaine läbinud üliõpilane oskab luua lihtsamaid arvutiprogramme ning omab algoritmilise mõtlemisviisi aluseid [18].

3.2 Keskkond Lahendus

Üha rohkem inimesi, eriti noori, on huvitatud programmeerimise õppimisest, kuid enamikel õppeainetel on raskesti hallatavad automaatkontrollid, mida oskavad luua ning muuta vaid vähesed õppejõud. Selle murekoha lahendamiseks otsustas Tartu Ülikooli arvutiteaduse instituudi informaatika didaktika töörühm [1] luua intuitiivse kasutajaliidesega keskkonna. Aluseks võeti avatud koodiga tarkvara nimega Lahendus⁶, mida võivad kõik soovijad kasutada ning vajaduspõhiselt edasi arendada.

⁵ Ainepunktide mõõtühik

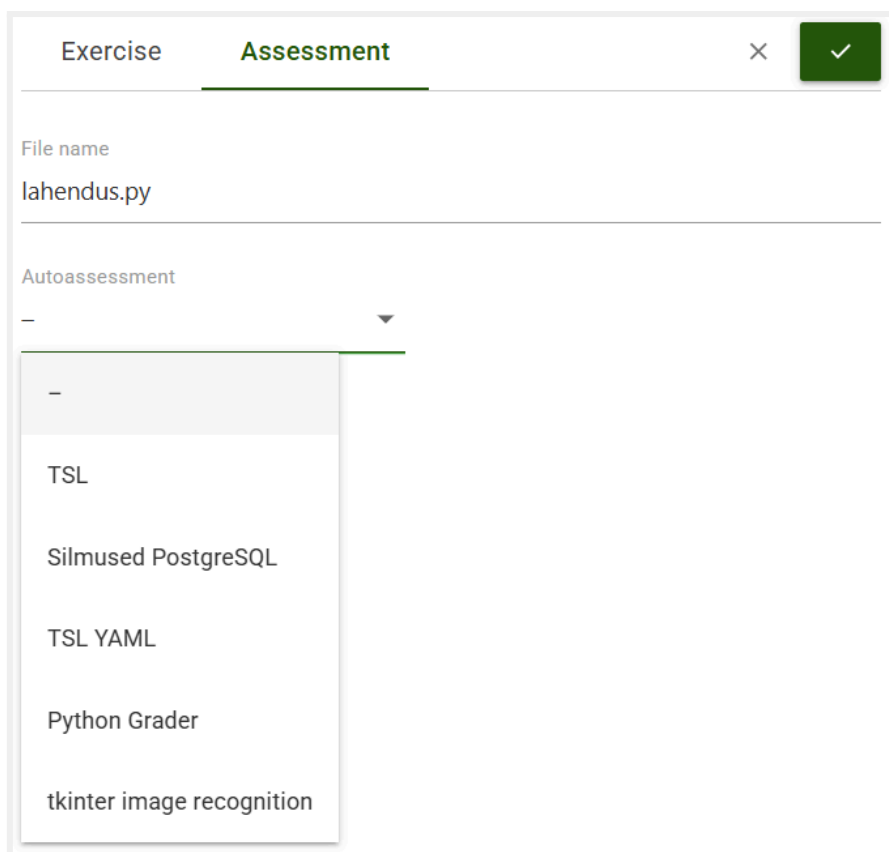
⁶ lahendus.ut.ee



Joonis 1. Keskkond Lahendus

Loodud keskkond Lahendus (vt Joonis 1) on kasutusel mitmetes üldhariduskoolides ning kutsehariduskoolides üle Eesti, sealhulgas Tartu Ülikoolis [1]. Selles keskkonnas on võimalik luua kursuseid, ülesandeid ja automaatkontrolle. Keskkonda on integreeritud Thonny IDE (*Integrated Development Environment*⁷) ning õpilased saavad lihtsasti lahendusi üles laadida ja hindamiseks esitada otse programmeerimiskeskonnast Thonny. Lisaks on keskkond integreeritud ka Moodle'iga, mis võimaldab sünkroniseerimist kahe süsteemi vahel (seda hetkel vaid Tartu Ülikooli Moodle'is). Automaatkontrollide loomiseks on mitu võimalust (vt Joonis 2): Python Grader, TSL (*Test Specific Language*) ning Tkinter pildituvastuse moodul.

⁷ Integreeritud programmeerimiskeskond



Joonis 2. Automaatkontrollide loomise võimalused

Python Grader on Pythoni moodul, mille lõi Karl-Aksel Puulmann 2014. aastal bakalaureusetöö raames [20]. Mooduli loomise eesmärgiks oli lihtsustada automaatkontrollide kirjutamist ning muuta programmeerimisse sissejuhatavate kursuste õpetamist kergemaks.

TSL-märgistuskeele arendas välja Informaatika didaktika töörühm koostöös Eerik Muuli jt [1]. See märgistuskeel on kirjutatud YAML (*YAML Ain't Markup Language*) vormingus, sest see on vähem verboossem⁸ ning kergesti loetavam. Lisaks toetab YAML vaikimisi automaatkontrollide loomiseks enamikke andmetüüpe. Automaatkontrollide loomiseks on vaja ette anda sisend, oodatav väljund, võimalik punktisumma ning kriteeriumid ja väljastatavad sõnumid, mis kuvatakse lähtudes sellest, kas programm läbis testi või mitte [1]. TSL-is kirjutatud koodi näidis on esitatud joonisel 3.

⁸ Verboosne - paljusõnaline

Tests TSL Generated scripts

```
1 {
2   "language": "python3",
3   "validateFiles": true,
4   "requiredFiles": [
5     "lahendus.py"
6   ],
7   "tslVersion": "1.0",
8   "tests": [
9     {
10      "type": "program_execution_test",
11      "name": "Example Chuck",
12      "pointsweight": 1,
13      "visibleToUser": true,
14      "inputs": null,
15      "passedNext": null,
16      "failedNext": null,
17      "id": 3514,
18      "standardInputData": [
19        "Chuck"
20      ],
21      "inputFiles": [
22      ],
23      "genericChecks": [
24        {
25          "id": 3711,
26          "checkType": "ALL OF THESE".
```

Joonis 3. TSL koodi näidis

Tkinter [21] on Pythoni moodul, millega on võimalik luua erinevaid GUI komponente. Nendeks võivad olla näiteks erinevad aknad, nupud, menüüd, tekstiväljad ning pildid. Lisaks on moodulit võimalik kasutada ka automaatkontrollis. Tkinteriga on võimalik luua pildituvastus, mis võrdleb hindamiseks esitatud koodi poolt loodud pilti kontrollis oleva pildiga ning vastavalt sarnasus-protsendile annab tagasisidet. Tkinteri pildituvastuse kasutamise näidis on välja toodud joonisel 4.

```
evaluate.sh  kontroll.py  renamer.py  tester.py  tester_2.py

1 import os
2 import shutil
3 import ast
4 from PIL import Image
5 import io
6
7
8 def getter(widget):
9     widget.update()
10    ps = widget.postscript(colormode='color')
11    img = Image.open(io.BytesIO(ps.encode('utf-8')))
12    img.save("screenshot.jpg")
13
14 def get_student_file():
15     return 'submission.py'
16
17
18 if __name__ == '__main__':
19     student_file = get_student_file()
20     with open(student_file) as f:
21         orig_content = f.read()
22
23     shutil.copy(student_file, 'orig_student_submission.py')
24
25     canvas_name = None
26     tk_name = None
```

Joonis 4. Tkinteri pildituvastuse koodilõigu näidis

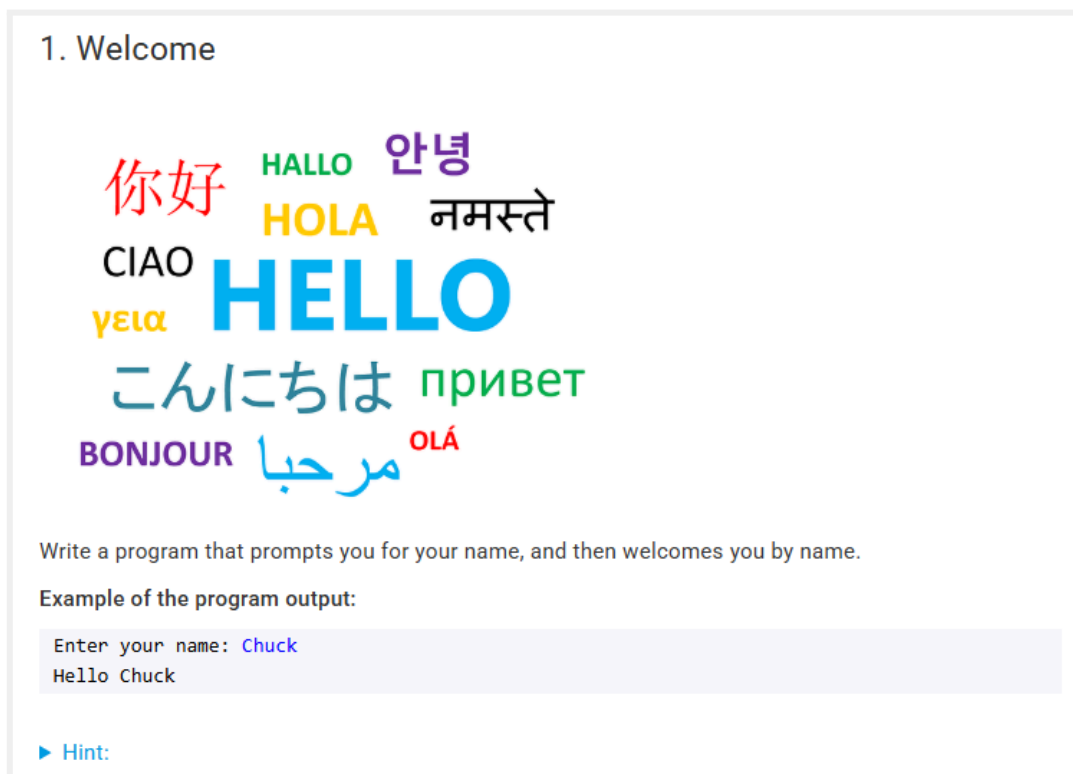
Varasem lahendus automaatkontrollide tegemisel sisaldas mitut Python'i programmi ning ühe kontrolli loomine võttis aega umbes 1-2 tundi [22]. Nüüd on seda võimalik teha keskkonnas Lahendus, kasutajaliidest kasutades, ning ajakulu on umbes 5-15 minutit. Kolmekuulise kasutusperioodi jooksul kogutud tulemused näitasid, et ajakulu kokkuhoid oli rohkem kui 170 tundi [22]. Hetkel ei ole võimalik kõiki ülesandeid kasutajaliidese abil automaatselt kontrollida (näiteks objektorienteeritud ülesanded või rekursioon), kuid keskkond on pidevas arenduses ning tehakse koostööd lõpp-kasutajatega (näiteks praktikumijuhendajad).

4. Automaatkontrollide loomise tööprotsess

Selles peatükis antakse ülevaade automaatkontrollide loomise protsessist. Esimene alampeatükk annab ülevaate ülesannete teksti koostamisest ning teine alampeatükk tutvustab lähemalt automaatkontrollide loomist.

4.1 Praktikumiülesannete tekst

Praktikumiülesannete tekst võeti „*Introduction to Programming*” Courses keskkonnast [17]. Teksti muudeti juhendajaga kooskõlastatult arusaadavamaks ning selle käigus kujunes välja ka ühtne stiil. Ülesanded on kirjutatud AsciiDoc⁹ formaadis [23].



1. Welcome

你好 HALLO 안녕하세요
HOLA नमस्ते
CIAO HELLO
γεια
こんにちは ПРИВЕТ
BONJOUR مرحبا OLÁ

Write a program that prompts you for your name, and then welcomes you by name.

Example of the program output:

```
Enter your name: Chuck  
Hello Chuck
```

▶ Hint:

Joonis 5. Illustreerivat pilti sisaldav praktikumiülesanne

Ülesannete tekstile lisati juurde illustreerivaid pilte (vt Joonis 5), kasutades vabavaralisi pilte veebilehelt Pixabay [24]. Mõnes praktikumiülesandes on pildi asemel tabel (vt Joonis 6), mis sisaldab kasulikku informatsiooni ülesande lahendajale. Üksikud ülesanded (näiteks näidiseksami ülesanded) jäeti piltideta, kuna leiti, et pildi olemasolu võib olla rohkem eksitav kui kasulik.

⁹ Inimloetav dokumendiformaat

4. Values and types

Basic built-in data types

type	set of values	common operators	sample literals
int	<i>integers</i>	+ - * // % **	99; 12; 2147483647
float	<i>floating-point numbers</i>	+ - * / **	3.14; 2.5; 6.022e23
bool	<i>true-false values</i>	and or not	True False
str	<i>sequences of characters</i>	+	'AB' 'Hello' '2.5'

Assume that we execute the following assignment statements:

```
width = 17
height = 12.0
```

For each of the expressions below, print the value of the expression and the type (of the expression value).

1. width/2
2. width/2.0
3. width/17
4. width//2
5. width//2.0
6. width%2
7. height/3
8. 1 + 2 * 5

Use the Python interpreter to check your answers.

Example:

```
width = 10.0
height = 8

print('1.', 5.0, 'float')
print('2.', 5.0, 'float')
...
```

Joonis 6. Tabelit sisaldav praktikumiülesanne

Iga ülesande juures on ka näidisväljundid, millele viitab paksemas tekstis olev sõna „*Example*” ning sellele järgneb hallil taustal programmijupp (vt Joonised 5 ja 6). Kui tegemist on kasutaja sisendit küsiva ülesandega, siis sisendiks antav sõna on märgitud sinises kirjas (vt Joonis 5). Mõne ülesande juures on ka vihjed, millele viitab klikitav sinises tekstis „*Hint*”, mis avab tekstivälja, kus on kirjas üliõpilast suunav vihje (vt Joonis 7).

▼ Hint:

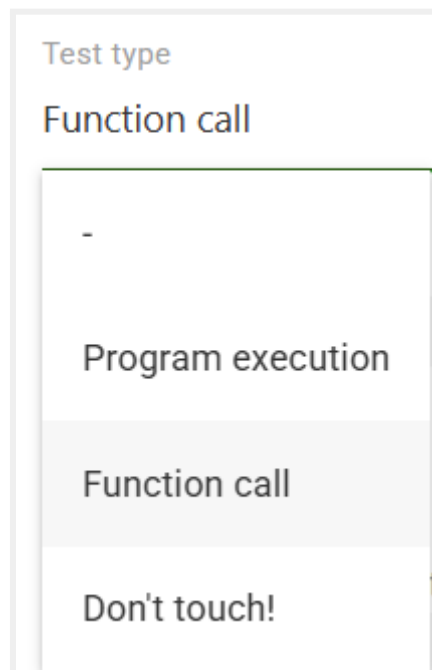
Use `input()` to ask user for input.

Joonis 7. Vihjet sisaldav lõik ülesandest

Praktikumiülesannete tekstid kohandati juhendajaga kooskõlastades selgemaks ja stiililt ühtsemaks, täiendades neid visuaalsete ja tekstiliste elementidega (sh illustreerivad pildid, tabelid, näidiseväljundid ja vihjed), et toetada õppijate programmeerimisalaste teadmiste ja oskuste omandamist.

4.2 Automaatkontrollide loomine

Ülesannete automaatkontrollid luuakse keskkonnas Lahendus. Joonisel 8 on kujutatud erinevad testitüübid, mida on võimalik kasutajaliidest kasutades teha. „*Program execution*” kutsub välja programmi enda, „*Function call*” kutsub aga välja ainult ette antud funktsiooni ning „*Don't touch!*” on mõeldud kontrollidele, mida on vaja käsitsi kirjutada. Selliste automaatkontrollide jaoks võeti abiks Notbergi bakalaureusetöö [25]. Selle töö raames koostati TSL keeles kirjutatavate automaatkontrollide jaoks dokumentatsioon.



Joonis 8. Valitavad testi tüübid

Enne, kui hakati automaatkontrolle tegema, loodi kõigepealt ülesande teksti järgides lahendatud programm. Selleks kasutati tehisintellekti (täpsemalt ChatGPT [26]) abi. Joonisel 9 on kujutatud üht vestlust juturobotiga, kus paluti tal ülesanne ära lahendada, kuid kuna ta tegi ülesande lahenduse funktsioonina, siis öeldi talle, et ta seda ei teeks. Automaatkontrollid loodi dünaamilise hindamise põhimõttel, kus õpilane esitab oma loodud programmi ning seejärel kontrollitakse sisendite põhjal programmi väljundit.

Write a program that prompts the user for his/her name and actual foot length in cm. The program must compute and output a suitable shoe size (use the following formula: shoe size = 1.5 * (foot length + 1.5)). Round the answer (you can use round for that, e.g. round(2.5) will give 3).

Example
Enter your name: Marina
Enter your foot length (cm): 23.5
Dear Marina, your suitable shoe size is 38

Failed to generate

`</>` Shoe Size Calculator

This program prompts the user for their name and foot length in centimeters, then calculates their suitable shoe size using the given formula. The result is rounded and displayed in a friendly message. Let me know if you'd like any modifications!

please don't do functions

```
</> Shoe Size Calculator
```

```
1 # Get input from the user
2 name = input("Enter your name: ")
3 foot_length = float(input("Enter your foot length (cm): "))
4
5 # Calculate shoe size using the given formula
6 shoe_size = round(1.5 * (foot_length + 1.5))
7
8 # Display result
9 print(f"Dear {name}, your suitable shoe size is {shoe_size}")
10
```

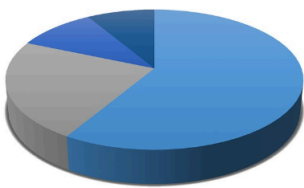
Edited Shoe Size Calculator

I've removed the function and kept the logic in a straightforward script format. Let me know if you need any further modifications!

Joonis 9. Prakikum 1 ülesande 5 lahendus

Kui programm oli valmis, hakati selle põhjal looma automaatkontrolli. Esimesed testid tehti ülesandes olevate näidete põhjal ning lisati ka teistsuguste argumentidega testjuhud. Bakalaureusetöö autor otsustas, et ei arvesta automaatkontrollide loomisel ebakorrektsid sisendeid, kui nende käsitlemist ei ole ülesande tekstis nõutud. Selline otsus tulenes järgnevast: kuna tegemist on algkursusega ning seda võtavad enamasti üliõpilased, kellelt ei eeldatud varasemat programmeerimiskogemust, siis igasugused lisanõuded (mis pole tekstis välja toodud) võivad tekitada segadust.

2. ECTS and hours



Write a program that asks the student about the ECTS and the duration of the present course in weeks. The program outputs the hours per week outlined in this course. (1 ECTS is 26 hours)

Example of the program output:

```
Enter ECTS: 3
Enter weeks: 9
Hours per week - 8.666666666666666
```

▼ Hint:

```
hours = (ECTS * 26 / weeks)
```

File name: lahendus.py
 Autoassessment: TSL
 Allowed execution time: 7 sec
 Allowed memory usage: 30 MB

Tests	TSL	Generated scripts
Example		
6 ECTS, 7 weeks		
6 ECTS, 9 weeks		
9 ECTS, 5 weeks		
9 ECTS, 9 weeks		

Joonis 10. Praktikum 1 ülesande 2 tekst ja testjuhud

Joonisel 10 on kujutatud ülesande tekst ning automaatkontrollis olevad testjuhud. Ülesande lahendust kontrollitakse kõigepealt näitega, mis on tekstis välja toodud. Seejärel ka mitme erineva testjuhuga, et hinnata esitatud programmi korrektsust väljundite põhjal võimalikult hästi. Selleks, et ei toetutaks liialt juturoboti lahendustele, katsetati automaatkontrolle ka 2024. aastal kursust läbinud üliõpilaste lahenduste peal. Vajadusel muudeti või täiendati testjuhtusid.

```

"tests": [
  {
    "type": "program_contains_loop_test",
    "name": "Program contains loop test",
    "pointsWeight": 1,
    "visibleToUser": true,
    "inputs": null,
    "passedNext": null,
    "failedNext": null,
    "id": 1,
    "programContainsLoop": {
      "mustNotContain": false,
      "beforeMessage": "",
      "passedMessage": "Program contains loop",
      "failedMessage": "Program doesn't contain loop"
    }
  },

```

Joonis 11. Automaatkontrolli programmilõik

Enamik automaatkontrolle oli võimalik luua kasutajaliidese abil, kuid mõne kontrolli jaoks oli vaja kirjutada test käsitsi. Üheks selliseks on näiteks neljanda praktikumi neljas ülesanne, kus on vaja kontrollida, kas programm sisaldab tsüklit või mitte. Seda kontrolli on kujutatud joonisel 11.

4.3 Tagasiside küsimustik

Bakalaureusetöö raames valminud praktikumiülesannete teksti kvaliteedi hindamiseks koostati tagasiside küsimustik. Küsimused kohandati lähtudes LORI (*Learning Object Review Instrument*) hindamismudelist [27]. Ülevaatajad saavad hinnata materjali lähtudes kaheksast punktist¹⁰: sisu kvaliteet (*Content Quality*), kooskõlastatus õpieesmärkidega (*Learning Goal Alignment*), tagasiside ja kohandatus (*Feedback and Adaptation*), motiveerivus (*Motivation*), kujundus (*Presentation Design*), interaktiivsus (*Interaction Usability*), kättesaadavus (*Accessibility*) ja standartitele vastavus (*Standards Compliance*).

Tagasiside küsimustik tehti Google'i vormina ning saadeti seitsmele praktikumijuhendajale 11. aprillil 2025 ning vastuseid oodati 20. aprillini. Selles esitati seitse küsimust praktikumi ülesannete kohta, näiteks kas tekst on arusaadav ning kas pilt on illustreeriv ja ei ole eksitav. Küsimused on välja toodud lisas 1 (vt Lisa 1).

¹⁰ Indrek Avi magistritöös [28] on LORI mudel tõlgitud eesti keelde ning paremini mõistetav.

Esimene küsimus on teksti kohta: kui kergesti on tekst visuaalselt jälgitav ning mida oleks vaja parandada. Teine küsimus on piltide kohta: kui sobilikud on ülesandeid illustreerivad pildid ning millised võiks ära vahetada. Kolmas küsimus on motiveerituse kohta: kas ülesannete juures on piisavalt vihjeid, et hoida õpilase motiveeritust ja toetada lahendamist ning millistele ülesannetele võiks rohkem vihjeid olla. Neljas ja kuues küsimus on arusaadavuse kohta: kui arusaadav on ülesande tekst õppijale, mida võiks parandada ning mis kohad võivad tekitada segadust. Viies küsimus on üldise stiili kohta: kas stiil on ühtlane. Seitsmenda küsimusega on oodatud kõik ettepanekud, mis eelnevate küsimustega kaetud ei saanud.

Bakalaureusetöö raames praktikumiülesannete teksti kvaliteedi hindamiseks koostatud tagasisideküsitlus võimaldas koguda sisulist ja struktuurilist tagasisidet ülesannete arusaadavuse, visuaalse esituse ja pedagoogilise tõhususe kohta.

5. Loodud automaatkontrollide kvaliteedi analüüs

Selles peatükis tehakse eksperdilt saadud tagasiside põhjal parandusi: kirjutatakse, mis olid ettepanekud ning kuidas neid rakendati. Analüüsitakse bakalaureusetöö raames loodud automaatkontrollide tööd eelnevalt kursust läbinud üliõpilaste loodud ülesannete lahenduse põhjal.

5.1 Eksperdi tagasiside

Tagasiside küsimustikku täitis ainult üks juhendaja, kes edaspidi on kirjeldatud kui ekspert. Eksperdi hinnangul oli praktikumiülesannete tekst visuaalselt jälgitav ning õppijale arusaadav. Lisaks on ülesannete stiil sobivalt ühtlane, tsitaat vastusest „*Ülesande tekstid on vormistatud viisil, et neid on hea visuaalselt jälgida.*”

Teise küsimuse juures öeldi, et üldjuhul on illustreerivad pildid sobivad, kuid mõne ülesande juures võiks muuta pilti selliselt, et need toetaks ülesande teksti rohkem. Välja toodi praktikum 2 ülesanne „*Pizza price (€/cm²)*” ning praktikum 4 ülesanne „*The three-year old child simulator*”. Sellele vastavalt täiendati ja muudeti pilte.

Pitsa ülesande muudetud pilt on kujutatud joonisel 12. Kujutamaks 1 cm² ala kasutati rakendust Paint ning laeti pilt ImgBB [29] lehele üles, mis on tasuta üleslaadimist pakkuva lehekülge. Pildi põhi jäi samaks ning on pärit Pixabay [24] vabavaralisi pilte sisaldavalt lehel. Muudatused tehti vastavalt eksperdi soovitusel, mis oli järgnev: „Näiteks ülesandes *Pizza price (€/cm²)* võiks pildile lisada ala, mis on 1 cm² ja näidata, et selle hind tuleb leida.”



Joonis 20. Ülesande „*Pizza price (€/cm²)*” pilt

Ekspert tõi välja, et ülesandes „*The three-year old simulator*” olev pilt on liiga intensiivne. Küsimuse 2b vastus oli järgnev: „5. *The three-year old child simulator. Pilt on sobilik, aga natuke liiga tähelepanu võttev ja intensiivne seoses kirjususega.*”. Selles ülesandes otsustas autor muuta pilti vähem intensiivsemaks, kuid jättes põhikomponendid, küsimärgid, taustal alles. Muudatust on näha joonisel 21, kus ülemine pilt näitab seda, mis asemele pandi, ning alumine pilt kujutab seda, mida muudeti. Mõlemad pildid on võetud leheküljelt Pixabay [24].



Joonis 21. Ülesande „*The three-year old simulator*” pildi muudatus

Ekspert tõi välja paar ülesannet, kuid modifitseerimist vajavaid pilte oli veel ning autor vaatas kõik ülesanded uue pilguga üle. Ülesanded, mille piltides tehti muudatusi on järgnevad: „*Shoe size*”, „*Gender from an Estonian Personal Identifications Number*” ning „*Business card*” - kokku kolm ülesannet.

Kolmas küsimus oli vihjete kohta, kas neid on ülesannetes piisavalt. Üldjoontes oli eksperdi sõnul vihjeid piisavalt, kuid tärnülesannetes¹¹ võiks rohkem olla: „*Ülesannetes on piisavalt vihjeid, aga tärnülesannetes võiks vihjeid veidi rohkem olla: 6*. The soup is getting cold; 4*. Euros and cents*”. Välja toodud ülesanded olid „*The soup is getting cold*” ning „*Euros and cents*”. Esimesena nimetatud ülesandele lisati vihje, et näiteks on võimalus kasutada *while* tsükli, jälgida millal saabub oodatud väärtus ning siis tsükkel lõpetada. Teisena nimetatud ülesandes täiendati olemasolevat vihjet näidetega. Ekspert ei toonud rohkem ülesandeid välja,

¹¹ Tavapärasest ülesandest keerukam ülesanne, mille lahendamine on rangelt soovituslik

seetõttu ei vaadatud ka teisi ülesandeid, kuna leiti, et tegemist on siiski tärnülesannetega, mis peaksidki olema raskemad tavapära-stest ülesannetest.

Viimases küsimuses sooviti teada, kas on veel mingeid ettepanekuid ülesannete parendamiseks. Välja toodi kaks ülesannet „*Money change with GUP*” ning „*Processing records*”. Vastuse esimene pool oli järgnev: „*2. Money change with GUI - siin viib tekstisisene viide coursesi lehele. Ma arvan, et võiks näidistfunktsiooni money change ette anda.*”. Toodi välja, et viide viib Courses’i lehele, see viga parandati ning nüüd viib link keskkonnas Lahendus olevasse ülesandesse. Lisaks soovitati anda näidistfunktsioon ette, kuna ülesanne põhineb varasema lahenduse olemasolule ning kõigil ei pruugi see olla koheselt olemas. Tsiteerides vastuse teist pool „*4a. Processing records - If you don’t have Excel, use this file instead. - fail viitab coursesi lehele.*”, siis seal välja toodud ülesandes oli ka lingi probleem, et viitab valele lehele. Seda linki ei muudetud.

5.2 Üliõpilaste esitatud lahenduste analüüs

Kuigi automaatkontrollid on välja töötatud, on vajalik hinnata nende toimimist ning veenduda, et need täidavad kavandatud eesmärki tõhusalt ja usaldusväärset. Selleks võetakse 2024. aasta sügissemestril kursust läbinud üliõpilaste lahendused, iga ülesande jaoks valitakse juhuslikult välja viis lahendust ning kävitatakse need automaatkontrollis. Iga praktikumi kohta tehakse tabel ning tuuakse välja, mis on automaatkontrolli tulemus ning milles viga seisnes.

Esimese praktikumi ülesannetest testiti neli ülesannet viiest, sest üks ülesanne muudeti alles bakalaureusetöö käigus automaatkontrollitavaks. Varasemalt ülesanne ei nõudnud otsest väljundit. Lisas 2 on tabel, mis kujutab esimese praktikumi tulemusi. Juhuslikult valitud ülesanded läbisid automaatkontrolli hästi. Kaks lahendust ülesandes „*ECTS and hours*” ei saanud maksimaalset tulemust järgnevatel põhjustel: vastust ei väljastatud ekraanile, ei arvestatud kasutaja sisendiga või oli tehtud ümardamine.

Teise praktikumi ülesannetest oli võimalik kõiki testida. Lisas 3 on tabel, milles on kontrolli tulemused. Esimeses ülesandes, pealkirjaga „*Greetings*”, läbisid lahendused kõik testid edukalt. Ülesande „*Pizza price (€/cm²)*” puhul oli näha, et peamine viga tuli sellest, et üliõpilased ei kasutanud pii täpset väärtust, vaid ligikaudset. Kolmandas ülesandes „*Grading*”

system” lahenduse A korral oli näha, et automaatkontroll annab valepositiivse tulemuse hinde E korral, kuna ebakorrektnel lahendus võib väljastada sõna või lause, mis sisaldab tähte E. Ülesandes „*What is the name of the month*” tuleb esile hooletusvead lahendustes, mis on tekkinud ülesande teksti ebakorrektest järgimisest. Viienda ülesande „*Students*” automaatkontroll testiti nelja lahendusega, kuna rohkem täispikkuses lahendusi ei leidunud. Tulemustest tuli välja, et automaatkontroll võib olla ebakorrektnel, kuna kõigis lahendustes olid täpselt samad vead. Kõigi ülesannete ülene probleem oli see, et automaatkontroll ei ole paindlik väljastatavates vastustes, vaid need peavad olema täpselt sellised nagu on näidisväljundis.

Kolmandas praktikumis oli viis ülesannet, millele koostati automaatkontroll. Esimeses ülesandes „*Money change*” tehti üliõpilaste lahendustes parandused, kuna ülesande lahendamise hetkel oli rahakurss 1.09, kuid automaatkontrollide kirjutamise hetkel oli see 1.19. Lahenduste esitamisel oli näha, et põhiprobleemiks oli ümardamise puudumine, mida oli näitest aru saada (ümardati kolm kohta pärast koma). Ülesandes „*What is the name of the month*” olid enamus ülesandeid väga hästi lahendatud, kuid lahenduses A olid alates viiendast kuust nummerdus ebakorrektnel ning lahenduses D ei tagastanud funktsioon väärtust, mida oli ülesandes tahetud. Kolmandas ülesandes „*Date as string*” puudus veateate „*Invalid date*” käsitlus, kuna see lisati sinna automaatkontrollide koostamisel. Tulemustest on näha ka teisendusprobleemid (*integer*-ile tahetakse liida *string*) ning kirjavead. Ülesandes „*Budget*” oli peamine viga viimases automaatkontrolli juhuses, kus *minimum budget* tuleb suurem kui *maximum budget*. Seal oodatakse, et üliõpilane teeb kontrolli, kus võrdleb neid ning omistab õige väärtuse vastavalt nimetusele. Lahenduses D oli õpilane liiga loov ja kirjutanud funktsiooninime „*budget*” asemel „*budget*”, loovat nimetust automaatkontroll ei tuvastanud ning seetõttu andis vea. Viimane ülesanne „*Money*” oli hästi lahendatud, kuid lahenduses D oli võrdluses tekkinud viga, „suurem kui” asemel oli pandud „suurem-võrdne” ning automaatkontroll ei saanud õiget vastust.

Neljas praktikum oli üldiselt hästi tehtud ning enamik vigu tulenes sellest, et ülesannete teksti ja oodatud lahendust muudeti automaatkontrolle tehes. Esimeses ülesandes „*Mantra*” oli lahenduses E pandud sisendit hoidev väärtus jutumärkidesse ning seetõttu ei kuvatud ekraanil õiget lahendust. Teises ja kolmandas ülesandes („*Triangle*” ja „*Integers that are not divisible by 10*”) viga ei esinenud. Ülesandes „*Celsius and Fahrenheit*” olid vead tingitud sellest, et lahenduste lahendamise hetkel ei olnud nõutud eraldi funktsiooni, kuid automaatkontroll

kontrollis funktsiooni olemasolu. Viienda ülesande „*The three-year old simulator*” vead tulid sarnasel põhjusel: automaatkontroll ootas programmi töö lõppedes sõna „*Okay*”, kuid lahendamise hetkel nõue puudus. Ülesande „*The soup is getting cold*” automaatkontrolli kontrolliti nelja lahendusega. Lahendus A väljastas väärtused ainult viie minuti kohta, kuid ülesandes oli nõue, et supi- ja toatemperatuurid peavad olema võrdsed. Lahenduse B väljundis oli puudu „:” märk, mistõttu ei lugenud automaatkontroll väljundit õigeks. Lisaks oli tsükli lõpu tingimuses väärtused tehtud *integer*-iks ning programm lõpetas töö liiga vara ära, kuna ei ümardanud väärtusi ühe komakoha täpsusega (uues juhendis on lisatud vihje selle kohta). Lahendused C ja D töötasid vastavalt 60 minutit ning 173 minutit, mis väljastas liiga palju väärtuseid, kuid automaatkontroll luges mõlemad lahendused õigeks.

Viienda praktikumi esimene ülesanne „*Word Square*” oli väga hästi lahendatud ning kõik lahendused said maksimumpunktid. Teise ülesande „*Gender from an Estonian Personal Identifications Number*” põhiprobleemiks oli väljundi ja oodatud tulemuse erinevus, automaatkontroll ootas väljundisse „*Female*” ja „*Male*”, kuid lahendustes oli see mitmel erineval viisil kirjas (nt „*Woman*”, „*female*”, „*a woman*”). Nüüd on selles ülesandes ka näidisväljundid ning probleem võiks olla lahendatud. Lisaks lahenduses C ei muudetud isikukoodi esimest numbrit *integer*-iks, mis näitas et üliõpilane oma koodi suure tõenäosusega ei käivitanud. Ülesande „*Business card*” põhiveaks oli väljundis oleva kaardi laiuse erinevus näitest. Olid ka kasutatud sümbolite erinevused, näiteks „-” asemel „_” või mõnel real „|” asemel „+”. Neljandas ülesandes „*Euros and cents*” ei läinud automaatkontrollist läbi lahendused C ja E. Lahenduses C oli arvutusviga, 100 asemel tehti arvutus 10-ga, ja lahenduses E puudus mitmuse ja ainsuse eristus ning eurode ja sentide vahel puudus sidesõna „*and*”. Viimast ülesannet „*Palindrome*” katsetati nelja lahendusega. Kolmel lahendusel puudus väljundis sõna esiletoomine, et mida vaadeldi, ning neljandas lahenduses oli kasutatud ülakoma („ ‘ ‘) asemel jutumärke („ ‘ ‘). Need vead on kergesti välditavad, kui on näide ees, seda lahenduste lahendamisel ei olnud.

Kuues praktikum keskendus peamiselt failitöötlusele ning seetõttu suurem osa vigu tekkisid automaatkontrollis oleva failinime ja lahenduses oleva failinime erinevusest. Paljudes ülesannetes ei olnud eelnevalt ka näidislahendust, mis nüüd on lisatud ning mille järgi automaatkontroll kontrollis. Esimeses ülesandes „*Odd or even*” just need vead esinesidki. Ülesandes „*Summing*” oli lisaks teksti erinevusele ka teised vead. Nendeks olid väljundis kogu summa puudumine, ei eristatud paaritut ja paarisarvu ning koodiviga, kus kasutatavaid

muutujaid ei oldud eelnevalt defineeritud. Kolmandas ülesandes „*Filtering names*” oli peamiseks probleemiks see, et ei arvestatud alamstringi töötusega. Teiste vigade seas olid muutuja vale algväärtus (väärtus algas 1, kuigi oleks pidanud algama 0) ning failinimi oli põhimõtteliselt vale („nimed.py”, faililaiendiga „.py” ei hoita andmeid). Ülesandes „*Upgraded budget*” esinenud vead olid seotud näite mittejärgimisest, näiteks puudus hinna juures „EUR” või ei väljastatud kutsutud ning osalevaid inimesi. Viimases ülesandes „*Horizontal bar*” puudus enamikel lahendustel täрни (*) juures tühik, automaatkontroll ootas kujul ‘* ’, kuid lahenduses A oli see olemas. Samas tuli välja, et kui tühik asub enne täрни, siis selline lahendus ei läbi automaatkontrolli.

Seitsmenda praktikumi esimene ülesanne „*Using list operations*” oli hästi lahendatud. Ülesandes „*What is the name of a month*” oli kahel lahendusel puudu veahaldus juhul, kui sisendina antud kuu number on suurem kui 12. Teiste vigade seas olid kuu nimetuse erinevused näidisväljundist, näiteks kirjavead, lühendina kirjutatud kuu nimetus ning väiketähtedega algavad kuu nimed. Kolmanda ülesande „*Date*” põhiveaks oli veahalduse puudumine, see esines kõigis kontrollitud lahendustes. Lahendustes esines ka vigu, näiteks kirjavead, nurksulgude asemel kasutati ümarsulge ning funktsioon ei tagastanud väärtust. Ülesande 4a „*Processing records*” lahendustest kolmel esines vigu eraldusjoonega andmete vahel, kas ei ühtinud näitega või puudus täielikult. Kasutati ka valet eraldusmärki, ette antud failis oli eraldusmärgiks „ , “, kuid kasutati „ ; “. Veaks loeti ka vale faililaiend, kuna ülesande eesmärk on töödelda “.csv” faililaiendiga faile, kuid näiteks lahenduses E töödeldi “.txt” laiendiga faili. Ülesanne 4b otsustati jätta analüüsimata, kuna ülesannet muudeti oluliselt ümber, et see saaks olla eraldiseisev ülesanne. Sellest tulenevalt ei täidaks see automaatkontrolli kvaliteedi hindamisel oma eesmärki. Viiendat ülesannet „*All words*” oli võimalik kontrollida ainult kahe lahendusega. Lahendus B saavutas maksimumtulemuse, kuid lahendus A automaatkontrolli ei läbinud, kuna ei küsinud kasutajalt failinime sisendina.

Tulemuste analüüsist selgus, et automaatkontroll ei ole paindlik erinevate väljundite suhtes, näiteks ei loe automaatkontroll õigeks sõna „*female*” kui on oodatud „*Female*”. Paindlikkust on võimalik lisada, kuid kõigi erijuhtude lisamine on aeganõudev ning tülikas, kuna kõigi erisuste peale ei pruugita kohe tulla. Samas võib eeldada, et üliõpilased järgivad näidet, kui see on ette antud, ning automaatkontrolli kuvatud veateated aitavad leida vigast rida, mille tõttu lahendus kontrolli ei läbinud. Üldiselt võib väita, et automaatkontrollid täitsid oma eesmärki, leides üles erinevad vead lahendustes.

Soovitusi kursuse korraldajatele on mitmeid. Üle tuleks vaadata teise praktikumi viies ülesanne, kohandada või täpsustada teksti/lähteülesannet, kuna analüüsi käigus selgus, et automaatkontroll võib olla vigane. Lisaks tuleks üle vaadata kaheksas praktikum, kuna seda ei olnud võimalik üle kontrollida. Soovi korral võib mitmekesistada kõigi automaatkontrollide aksepteeritavaid väljundeid, lisades erinevaid alamjuhte, mis sisuliselt on korrektsed, näiteks oodatava väljundi „*Female*” korral aksepteerib väljundeid nagu „*female*”, „*woman*”, „*Woman*” või „*a woman*”. See jätab alles üliõpilaste loomingulisuse.

Kokkuvõte

Automaatkontrollid on vajalikud, et lihtsustada õppejõudude tööd, andes neile lisaega, et parandada kursuse õppematerjale, ülesandeid ning vajadusel rohkem keskenduda keerukamatele teemadele. Automaatkontrollid on kasulikud ka õpilastele, sest annavad kohest tagasisidet ning õpilane saab omas tempos uurida ja lahendada ülesannet.

Töö käigus anti ülevaade automaatkontrollide vajalikkusest ning kahest erinevast õpetamise meetodist. Lühike ülevaade anti ka tehisintellektist ning sellega seonduvast plagiaadiprobleemist. Lisaks tutvustati käsitletavat ainet ning keskkonda Lahendus, kus automaatkontrollid loodi.

Bakalaureusetöö eesmärk oli luua automaatkontrollid kursuse „*Introduction to Programming MTAT.03.236*” praktikumiülesannetele ning viia need üle keskkonda Lahendus. Kokku viidi üle 8 praktikumi, mis sisaldas 42-te ülesannet, millest 39-le loodi automaatkontrollid. Automaatkontrollid loodi dünaamilise hindamise põhimõttel. Vaadati üle olemasolevad ülesannete püstitused, ühtlustati nende stiili ning lisati abistavaid materjale nagu pildid ning vihjed.

Praktikumiülesannete teksti kvaliteedi kontrolliks koostati tagasiside küsimustik ning saadud vastuste põhjal tehti ülesannete tekstis parandusi. Valminud automaatkontrolle testiti varasemalt kursust läbinud üliõpilaste lahendustega ning analüüsiti saadud tulemusi. Tulemustest oli näha, et automaatkontroll ei ole paindlik ega vastuvõtlik loovusele. Üldjoontes automaatkontrollid täitsid oma eesmärgi, tuues välja lahendustes esinenud vead.

Kasutatud allikad

- [1] Muuli E., Palm R., Lepp M., Simplifying the creation and maintenance of automated assessments of programming tasks via Test Specific Language, *In Proceedings of the 2022 6th International Conference on Education and E-Learning (ICEEL '22)*. Association for Computing Machinery, New York, NY, USA, 14–20, 2022, <https://doi.org/10.1145/3578837.3578840> (02.12.2023).
- [2] Skalka J., Drlik M., Obonya J., Automated Assessment in Learning and Teaching Programming Languages using Virtual Learning Environment, pp. 689-697, 2019, https://www.researchgate.net/publication/333501137_Automated_Assessment_in_Learning_and_Teaching_Programming_Languages_using_Virtual_Learning_Environment (13.04.2025).
- [3] Hollingsworth J., Automatic graders for programming classes, *Commun. ACM* 3, 10 (Oct. 1960), pp. 528–529, 1960, <https://doi.org/10.1145/367415.367422> (30.11.2023).
- [4] Jackson D., Usher M., Grading Student Programs Using ASSYST, *In Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education (SIGCSE '97)*, Association for Computing Machinery, New York, NY, USA, 335–339, 1997, <https://doi.org/10.1145/268084.268210> (03.12.2023).
- [5] Elhayany M., Meinel C., Toward Automated Code Assessment with OpenJupyter in MOOCs, *In Proceedings of the Tenth ACM Conference on Learning @ Scale (L@S '23)*, Association for Computing Machinery, New York, NY, USA, pp. 321–325, 2023, <https://doi.org/10.1145/3573051.3596180> (03.12.2023).
- [6] Varga E. B., Kristóf Fekete A., Applications for Automatic C Code Assessment, *2023 24th International Carpathian Control Conference (ICCC)*, Miskolc-Szilvásvárad, Hungary, 2023, pp. 21-26, DOI: 10.1109/ICCC57093.2023.10178987, 2023, <https://ieeexplore.ieee.org/document/10178987> (04.12.2023).
- [7] Hui-Zhi Hu, Li-Guo Zhang, Jia-Hua Zhang, Di Zhang, Jia-Rui Xie, A multidimensional dynamic assessment system for competence in computer science to improve students' academic achievements and learning potential, *Education and Information Technologies*, pp. 1-28, DOI: 10.1007/s10639-024-13290-x, 2024, https://www.researchgate.net/publication/387370544_A_multidimensional_dynamic_assessment_system_for_competence_in_computer_science_to_improve_students'_academic_achievements_and_learning_potential (12.04.2025).

- [8] Muuli E., Lepp M., Palm R., Luik P., Automation of assessment and feedback in IT teaching from the teaching staff perspective, *2021 IEEE Frontiers in Education Conference (FIE)*, Lincoln, NE, USA, pp. 1–9. DOI: 10.1109/FIE49875.2021.9637290, 2021, <https://ieeexplore.ieee.org/document/9637290/authors#authors> (13.04.2025).
- [9] Becker D., Devine M., Automated assessments and student learning, *International Journal of Learning Technology* 3(1): 5-17, DOI:10.1504/IJLT.2007.012364, 2007, https://www.researchgate.net/publication/220497465_Automated_assessments_and_student_learning (10.05.2025).
- [10] Pieterse V., Automated Assessment of Programming Assignments, *In Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research (CSERC '13)*, Open Universiteit, Heerlen, Heerlen, NLD, 45–56, 2013, <https://dl.acm.org/doi/10.5555/2541917.2541921> (10.05.2025).
- [11] Akçapınar G., Sidan E., AI chatbots in programming education: guiding success or encouraging plagiarism, *Discover Artificial Intelligence*, DOI: 10.1007/s44163-024-00203-7, 2024, https://www.researchgate.net/publication/386038983_AI_chatbots_in_programming_education_guiding_success_or_encouraging_plagiarism (10.05.2025).
- [12] Karnalim O., Toba H., Johan M. C., Handoyo E. D., Setiawan Y. D., Luwia J. A., Plagiarism and AI Assistance Misuse in Web Programming: Unfair Benefits and Characteristics, *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, Auckland, New Zealand, 2023, pp. 1-5, DOI: 10.1109/TALE56641.2023.10398397, 2023, <https://ieeexplore.ieee.org/document/10398397> (10.05.2025).
- [13] Bulla L., Midolo A., Mongiovi M., Tramontana E., EX-CODE: A Robust and Explainable Model to Detect AI-Generated Code, *Information* 2024, 15(12), 819, 2024, <https://doi.org/10.3390/info15120819> (11.05.2025).
- [14] Aiken A., MOSS, Stanford University. <https://theory.stanford.edu/~aiken/moss/> (10.05.2025).
- [15] Karnalim O., Maintaining Academic Integrity in Programming: Locality-Sensitive Hashing and Recommendations, *Education Sciences* 13(1), 54, 2023, <https://doi.org/10.3390/educsci13010054> (11.05.2025).
- [16] Elkhatat A. M., Elsaid K., Almeer S., Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text, *International Journal for Educational Integrity* 19, 17, 2023, <https://doi.org/10.1007/s40979-023-00140-5> (11.05.2025).

- [17] Tartu Ülikool, Arvutiteaduse instituut, Arvutiteaduse instituudi kursused 2022/22 sügis, <https://courses.cs.ut.ee/2022/itp/fall/Main/HomePage> (04.05.2025)
- [18] Programmeerimise alused, Õppeinfosüsteem, <https://ois2.ut.ee/#/courses/MTAT.03.236/version/f10b4d86-af09-b65f-fc3d-805f0d1f3e8f/details> (14.04.2025).
- [20] Puulmann K. A., Python module for automatic testing of programming assignments, Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö, 2014, <http://macobo.github.io/python-grader/thesis.pdf> (04.12.2023).
- [21] Sai P. C., Karthik K., Prasad K. B., Sai Pranav C. V., V D. K., Real-Time Task Manager: A Python-Based Approach Using Psutil and Tkinter, *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, Bengaluru, India, 2024, pp. 1-6, DOI: 10.1109/CSITSS64042.2024.10816758, 2024, <https://ieeexplore.ieee.org/document/10816758> (14.04.2025).
- [22] Muuli E., Lepp M., Palts T., Papli K., Palm R., Reflecting on Simplification of the Creation and Maintenance of Automated Assessments for Programming Tasks, *2024 IEEE Global Engineering Education Conference (EDUCON)*, Kos Island, Greece, pp. 1-3, DOI: 10.1109/EDUCON60312.2024.10578575, 2024, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10578575> (24.04.2025).
- [23] AsciiDoc, Ascii süntaksi juhend, <https://docs.asciidoc.org/asciidoc/latest/syntax-quick-reference/> (15.04.2025).
- [24] Pixabay, vabavaraalisi pilte sisaldav veebileht, <https://pixabay.com> (15.04.2025).
- [25] Notberg E., Tudengiprogrammide automaatkontrollid programmeerimise kursustel. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö, 2024, <https://thesis.cs.ut.ee/96bcaba0-ffa2-4e82-a473-459fe26ccd29> (09.03.2025).
- [26] OpenAI (2022), ChatGPT (4), <https://chat.openai.com>
- [27] Nesbit J., Nesbit K., Leacock T., Learning Object Review Instrument (LORI), 2009, https://www.academia.edu/7927907/Learning_Object_Review_Instrument_LORI (05.04.2025).
- [28] Avi I., Digitaalne õppematerjal luu- ja lihaskonna ülekoormushaiguste ennetamiseks, Eesti Maaülikooli tehnikainstituudi magistritöö, 2019, <https://dspace.emu.ee/items/53712939-cb9f-48c3-b2d3-b07890464545> (06.04.2025).
- [29] ImgBB, <https://imgbb.com> (03.05.2025).

Lisad

Lisa 1. Tagasiside küsimustik

- 1a. Kui kergelt on praktikumiülesannete tekst visuaalselt jälgitav?
- 1b. Kuidas oleks võimalik teksti jälgitavust parandada?

- 2a. Kuivõrd sobilikud on ülesandeid illustreerivad pildid?
- 2b. Milliste ülesannete pildid ei ole sobilikud?

3. Kas ülesannetes on piisavalt palju vihjeid, et hoida motivatsiooni ja toetada ülesande lahendamist? Kui ei, siis millistele ülesannetele võiks vihjeid juurde lisada?

- 4a. Kuivõrd arusaadav on ülesande tekst õppijale?
- 4b. Milliste ülesannete tekst vajaks parandamist?

5. Kuivõrd ühtlane tundub ülesannete stiil?

6. Mis osa võib tekitada küsimusi või segadust?

7. On mingeid ettepanekuid, kuidas ülesandeid paremaks teha?

Lisa 2. Praktikum 1

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaat-kontroll (%)	Kommentaar
1. <i>Welcome</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
2. <i>ECTS and hours</i>	A	100	-
	B	100	-
	C	0	- Üliõpilane ei väljastanud vastust ekraanile - Üliõpilane ei arvestanud sisestatud nädalatega
	D	40	- Üliõpilane ümardas arvutused, kuigi seda ei olnud nõutud
	E	100	-
3. <i>Celsius and Fahrenheit</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
5. <i>Shoe size</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-

Lisa 3. Praktikum 2

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaat-kontroll (%)	Kommentaar
1. <i>Greetings</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
2. <i>Pizza price</i> (€/cm ²)	A	100	-
	B	25	- Üliõpilane ei kasutanud pi täpset väärtust - Üliõpilase sõnum vea saamisel on erinev nädisest
	C	25	- Üliõpilane ei kasutanud pi täpset väärtust - Üliõpilane kasutab väärtust, mis ebakorrekse sisendi tõttu jäi määramata
	D	25	- Üliõpilane ei kasuta pi täpset väärtust - Üliõpilase arvutuskäik on vale
	E	100	
3. <i>Grading system</i>	A	18	- Üliõpilase tingimuslause on ebakorrekne - Üliõpilasel puudub veateade, kui tulemus ei ole vahemikus 0 kuni 100
	B	82	- Üliõpilasel puudub vealause juhul, kui sisestatakse ebakorrekne kursusetüüp - Ebakorrekse tulemuse korral erinev üliõpilase lause nädisest
	C	88	- Ebakorrekse tulemuse korral erinev üliõpilase

			lause näidise
	D	76	- Ebakorrektsuse tulemuse korral erinev üliõpilase lause näidise - Mitteeristava hindamise korral üliõpilane ei teisenda sõned integeriks
	E	94	- Puudub tingimuslause hinde F korral
4. <i>What is the name of the month?</i>	A	56	- Sõnalise kuu puhul vealause erineb näitest - Numbrilise kuu puhul peab olema 2 erinevat vealause, kuid on 1
	B	78	- Hooletusviga vealases („The” asemel „The”)
	C	56	- Numbrilise kuu puhul peab olema 2 erinevat vealause, kuid on 1 - Üliõpilase sõnum vea saamisel on erinev näidise
	D	78	- Üliõpilase sõnum vea saamisel on erinev näidise
	E	56	- Numbrilise kuu puhul peab olema 2 erinevat vealause, kuid on 1 - Sõnalise kuu puhul jääb „month” väärtus määramata
5*. <i>Students</i>	A	78	- Ei tule välja erisus, miks valiti väärtused 100 ning 20 - Ei tule välja erisus, miks valiti väärtused 1000 ning 10
	B	78	- Ei tule välja erisus, miks valiti väärtused 100 ning 20 - Ei tule välja erisus, miks valiti väärtused 1000 ning 10
	C	78	- Ei tule välja erisus, miks valiti väärtused 100 ning 20 - Ei tule välja erisus, miks

			valiti väärtused 1000 ning 10
	D	78	- Ei tule välja erisus, miks valiti väärtused 100 ning 20 - Ei tule välja erisus, miks valiti väärtused 1000 ning 10

Lisa 4. Praktikum 3

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaat-kontroll (%)	Kommentaar
1. <i>Money change</i>	A	86	- Ei ole ümardamist
	B	86	- Ei ole ümardamist
	C	86	- Ei ole ümardamist
	D	86	- Ei ole ümardamist
	E	43	- Ei ole ümardamist - Funktsioon ei tagasta väärtust
2. <i>What is the name of the month?</i>	A	33	- Alates maikuust on kuu number vale
	B	100	-
	C	100	-
	D	0	- Funktsioon ei tagasta väärtust
	E	100	-
3. <i>Date as string</i>	A	40	- Teisendamisprobleemid ehk integerile tahetakse string liita - Puudub „ <i>Invalid date</i> ” tagastus
	B	60	- Puudub „ <i>Invalid date</i> ” tagastus
	C	40	- Teisendamisprobleemid ehk integerile tahetakse string liita - Puudub „ <i>Invalid date</i> ” tagastus
	D	40	- Puudub „ <i>Invalid date</i> ” tagastus - Kirjaviga „ <i>Febuary</i> ”
	E	0	- Kirjaviga „ <i>february</i> ” - Kood tehtud läbimaks ainult näitega

4. Budget	A	75	- „Maximum budget” ja „Minimum budget” väärtus on vahetuses
	B	75	- „Maximum budget” ja „Minimum budget” väärtus on vahetuses
	C	75	- „Maximum budget” ja „Minimum budget” väärtus on vahetuses
	D	50	- Funktsiooninimetus ei ole nõuetekohane - „Maximum budget” ja „Minimum budget” väärtus on vahetuses
	E	75	- „Maximum budget” ja „Minimum budget” väärtus on vahetuses
5*. Money	A	100	-
	B	100	-
	C	100	-
	D	75	- „Suurem kui” võrdluse asemel on „suurem-võrdne” võrdlus
	E	100	-

Lisa 5. Praktikum 4

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaat-kontroll (%)	Kommentaar
1. <i>Mantra</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	0	- Sisendiks antud mantra nimetus oli <i>print</i> käsus jutumärkides
2. <i>Triangle</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
3. <i>Integers that are not divisible by 10</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
4. <i>Celsius and Fahrenheit</i>	A	33	- Funktsiooni ei ole defineeritud
	B	33	- Funktsiooni ei ole defineeritud
	C	33	- Funktsiooni ei ole defineeritud
	D	33	- Funktsiooni ei ole defineeritud
	E	33	- Funktsiooni ei ole defineeritud
5. <i>The three-year</i>	A	50	- Programm ei lõppe sõnaga

<i>old child simulator</i>			„Okay”
	B	50	- Programm ei lõppe sõnaga „Okay”
	C	50	- Programm ei lõppe sõnaga „Okay”
	D	50	- Programm ei lõppe sõnaga „Okay”
	E	50	- Programm ei lõppe sõnaga „Okay”
<i>6*. The soup is getting cold</i>	A	67	- Programm teeb ainult 5 „minutit” tööd
	B	33	- Väärtuste <i>soup</i> ja <i>room</i> võrdluseks tehakse need <i>integer</i> iks - Programmi väljundis puudub „:”
	C	100	-
	D	100	-

Lisa 6. Praktikum 5

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaatkontroll (%)	Kommentaar
1. <i>Word Square</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
2. <i>Gender from an Estonian Personal Identifications Number</i>	A	0	- Väljundis oli „ <i>female</i> ”, kuid automaatkontroll ootas „ <i>Female</i> ” - Väljundis oli „ <i>male</i> ”, kuid automaatkontroll ootas „ <i>Male</i> ”
	B	0	- Väljundis oli „ <i>Woman</i> ”, kuid automaatkontroll ootas „ <i>Female</i> ” - Väljundis oli „ <i>Man</i> ”, kuid automaatkontroll ootas „ <i>Male</i> ”
	C	0	- Väljundis oli „ <i>a woman</i> ”, kuid automaatkontroll ootas „ <i>Female</i> ” - Väljundis oli „ <i>a man</i> ”, kuid automaatkontroll ootas „ <i>Male</i> ” - Esimene number isikukoodis ei muudetud <i>integer</i> -iks
	D	0	- Väljundis oli „ <i>Gender: Female</i> ”, kuid automaatkontroll ootas „ <i>Female</i> ” - Väljundis oli „ <i>Gender: Male</i> ”, kuid automaatkontroll ootas „ <i>Male</i> ”
	E	0	- Väljundis oli „ <i>a woman</i> ”, kuid automaatkontroll ootas „ <i>Female</i> ”

			- Väljundis oli „a man”, kuid automaatkontroll ootas „Male”
3. <i>Business card</i>	A	0	- Kaardi laius ei ole sama, mis näitel
	B	0	- Kaardi laius ei ole sama, mis näitel - Ülemise ja alumise rea vahel ei ole kõik ääred sümboliga „ ”
	C	0	- Kaardi laius ei ole sama, mis näitel
	D	0	- Kaardi laius ei ole sama, mis näitel - Sümboli „-” asemel kasutati sümbolit „_”
	E	0	- Kaardi laius ei ole sama, mis näitel
4*. <i>Euros and cents</i>	A	100	-
	B	100	-
	C	75	- Arvutus tehtmärgiga „%” ebakorrekne
	D	100	-
	E	50	- Puudub mitmuse ja ainsuse eristus - Eurode ja sentide vahel pole sidesõna „and”
5*. <i>Palindrome</i>	A	0	- Väljundis ei ole näidatud, mis sõna vaadeldakse
	B	0	- Väljundis ei ole näidatud, mis sõna vaadeldakse
	C	0	- Väljundis ei ole näidatud, mis sõna vaadeldakse
	D	0	- Kasutatud jutumärgid ei ole samad, mis näites („ “ “ ja „ ‘ “)

Lisa 7. Praktikum 6

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaat-kontroll (%)	Kommentaar
1. <i>Odd or even</i>	A	0	- Ei ava näidises ette antud nimega faili - Väljundi tekst on erinev näidisest
	B	0	- Väljundi tekst on erinev näidisest
	C	0	- Ei ava näidises ette antud nimega faili - Väljundi tekst on erinev näidisest
	D	0	- Väljundi tekst on erinev näidisest
	E	0	- Ei ava näidises ette antud nimega faili - Väljundi tekst on erinev näidisest
2. <i>Summing</i>	A	0	- Tervik summa on puudu - Väljundi tekst on erinev näidisest
	B	0	- Väljundi tekst on erinev näidisest
	C	0	- Väljundi tekst on erinev näidisest
	D	0	- Väljundi tekst on erinev näidisest - Väljundis ei ole välja toodud, kas arv on paaris või paaritu
	E	0	- Väärtused ei ole defineeritud - Väljundi tekst on erinev näidisest
3. <i>Filtering names</i>	A	0	- „ <i>number</i> ” algväärtus on vale - Ei arvesta alamstringiga

	B	0	- Faili nimi on „nimed.py” mitte „names.txt”
	C	100	-
	D	67	- Ei arvesta alamstringiga
	E	67	- Ei arvesta alamstringiga
<i>4. Upgraded budget</i>	A	0	- Hinna juures puudub „EUR”
	B	0	- Ei väljasta kutsutud ja osalevaid inimesi
	C	100	-
	D	100	-
	E	0	- Hinna juures puudub „EUR”
<i>5. Horizontal bars</i>	A	100	-
	B	0	- Väljundis puudub tühik (* ‘ asemel ‘*)
	C	0	- Väljundis puudub tühik (* ‘ asemel ‘*)
	D	0	- Väljundis puudub tühik (* ‘ asemel ‘*)
	E	0	- Väljundis puudub tühik (* ‘ asemel ‘*)

Lisa 8. Praktikum 7

Ülesande number ja nimi	Esitatud lahenduse nimetus	Automaat-kontroll (%)	Kommentaar
1. <i>Using list operations</i>	A	100	-
	B	100	-
	C	100	-
	D	100	-
	E	100	-
2. <i>What is the name of a month</i>	A	80	- Puudub veahaldus kui number on suurem kui 12
	B	40	- Pikemad kuu nimetused on lühendina
	C	0	- Funktsiooni nimi on vale - Kuu nimed on väikese tähega
	D	100	-
	E	60	- Puudub veahaldus kui number on suurem kui 12 - Kirjaviga kuu „January” korral
3. <i>Date</i>	A	33	- Puudub veahaldus kui number on suurem kui 12 - Funktsioon ei tagasta väärtust
	B	0	- Nurksulgude asmele kasutatud ümarsulge - Kirjaviga („year” asemel „tear”) - Puudub veahaldus kui number on suurem kui 12
	C	0	- Kuu nimed on väikese tähega - Puudub veahaldus kui number on suurem kui 12
	D	67	- Puudub veahaldus kui number on suurem kui 12

	E	67	- Puudub veahaldus kui number on suurem kui 12
<i>4a. Processing records</i>	A	100	-
	B	0	- Andmete vahel puudub eraldusjoon
	C	0	- Andmete vahel puudub eraldusjoon
	D	0	- Andmete vaheline eraldusjoon ei ühti näitega - Eraldusmärk on vale (,, , “ asemel ,, ; “)
	E	0	- Faililaiend on vale
<i>5. All words</i>	A	0	- Ei küsi failinime sisendina
	B	100	-

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Brigit Parra,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Automaatkontrollide loomine kursuse „Introduction to Programming” praktikumiülesannetele**, mille juhendaja on Reelika Suviste, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Brigit Parra

15.05.2025