

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Arvutisüsteemide instituut

Aleksander Laasmägi 213453IACB

# ÕPETAMISPRAKTIKA

Praktika IXX0760

Juhendaja: Vladimir Viies  
Dotsent

Tallinn 2025

## Lühendite ja mõistete sõnastik

|                          |                                                                                                         |
|--------------------------|---------------------------------------------------------------------------------------------------------|
| IDE                      | <i>Integrated Development Environment</i> - Integreeritud programmeerimiskeskond/arenduskeskkond        |
| UML                      | <i>Unified Modeling Language</i> - Ühtne modelleerimiskeel                                              |
| GCC                      | <i>GNU Compiler Collection</i> - GNU kompilaatorite kollektsioon                                        |
| GNU                      | Operatsioonisüsteem                                                                                     |
| Kompileerima/kompilaator | Lähtekeeles (nt C keel) kirjutatud lähtekoodi teisendamine masinkoodi või vahekoodi, nt assemblerkeelde |

## Sisukord

|                                                         |    |
|---------------------------------------------------------|----|
| Lühendite ja mõistete sõnastik .....                    | 2  |
| Sisukord.....                                           | 3  |
| 1 Sissejuhatus .....                                    | 4  |
| 2 Asutusest.....                                        | 5  |
| 3 Ootused praktikale .....                              | 6  |
| 4 Praktikakoha kirjeldus .....                          | 7  |
| 4.1 Töökeskkond .....                                   | 7  |
| 4.2 Praktika maht ja ajad .....                         | 8  |
| 5 Praktika käik.....                                    | 9  |
| 5.1 Iganädalane tundide läbiviimine.....                | 9  |
| 5.2 Minu poolt läbiviidud tunnid.....                   | 10 |
| 5.2.1 Versioonihalduslahenduste lühitutvustus .....     | 10 |
| 5.2.2 Programmeerimiskeele Python lühitutvustus .....   | 11 |
| 5.3 Õppematerjalide koostamine .....                    | 12 |
| 5.4 Kontrolltööde ja eksamite hindamine.....            | 12 |
| 6 Praktika kokkuvõte ja arvamus.....                    | 13 |
| 7 Kasutatud allikad .....                               | 14 |
| Lisa 1 – Minu koostatud kontrolltöö eesti keeles .....  | 15 |
| Lisa 2 – Minu koostatud kontrolltöö inglise keeles..... | 17 |

## **1 Sissejuhatus**

2024. aasta sügissemestril avanes mul hea võimalus sooritada praktika Tallinna Tehnikaülikoolis abiõppejõuna. See sobis mulle suurepäraselt, kuna mul oli kindel soov arendada enesekindlust rahva ees esinemisel. Samuti võimaldas see näha loengute ja praktikumide läbiviimise „köögipoolt“.

## 2 Asutusest

Tallinna Tehnikaülikool (tuntud ka kui TTÜ või TalTech) asutati 1918. aastal ja on Eestis ainuke tehnika- ja IT-teaduse ning -hariduse lipulaev, milles on viis teaduskonda (koos Eesti Mereakadeemiaga) ja 20 instituuti. TalTechi missiooniks on olla teaduse, tehnoloogia ja innovatsiooni edendaja ning juhtiv inseneri- ja majandushariduse andja Eestis. [1]

Arvutisüsteemide instituut on õppe-, teadus- ja arendustööga tegelev instituut, olles üks neljast instituudist infotehnoloogia teaduskonnas. Arvutisüsteemide instituudi üheks prioriteediks on anda tudengitele võimaluse tutvuda IT-erialade praktilisema poolega. [2]

Instituut ühendab mitmeid profiililt lähedasi uurimisrühmasid ja keskuseid [2]:

- Usaldusväärsete arvutisüsteemide keskus
- Arukate süsteemide keskus
- Biorobootika keskus
- Keskkonnaseire tehnoloogiate keskus
- Riistvara turvalisuse keskus
- Nutika riistvara uurimiskeskus
- Sardtehisintellekti labor
- Arvutisüsteemide õppekeskus

### 3 Ootused praktikale

Nagu varasemalt juba mainitud sai, läksin praktikale ootustega saada uut ja unustamatut kogemust. Avalik esinemine on minu arvates samuti väga vajalik oskus, mida arendada. Just seda õpetamispraktika mulle anda suudabki. Kui ma bakalaureuse teekonda alustasin, siis ei olnud päris täpselt selge, mida ma sooviksin IT valdkonnas saavutada. Valisin selle eriala ootusega saada siit teadmised nii riistvara kui tarkvara kohta. Taipasin teisel aastal, et tarkvara on rohkem minu teema ning selle järgi sai valitud ka õpetamispraktika koht. Juhendajaks sai valitud dotsent Vladimir Viies, kelle juhendada on Programmeerimine I (IAX0583). Teadsin, et olen võimeline andma edasi oma häid kogemusi ja teadmisi programmeerimise maailmast. Ehkki õpetasime peamiselt inseneriteaduskonna tudengeid, üritasin panna neid programmeerimisel mõtlema sarnaselt infotehnoloogiateaduskonna tudengitele. Sellised lihtsamad põhimõtteid nagu näiteks, et programmi kood peaks olema võimalikult puhas, muutujate ja funktsioonide nimetused asjakohased ning funktsioonid võimalikult suures mahus kogu programmi töös korduvkasutatavad, võiksid minu hinnangul olla teada ka inseneriteaduskonna tudengitele.

## 4 Praktikakoha kirjeldus

Programmeerimine I (IAX0583) aine raames õpetatakse riistvara arenduse ja programmeerimise, elektroenergeetika ja mehhatroonika ning integreeritud tehnoloogia õppekavade esmakursuslastele baasteadmisi algoritmilisest mõtlemisest, protseduurilisest programmeerimisest, versioonihaldusplatvormide (nt. GitLab) kasutamisest ja saadud teadmiste rakendamist riistvara programmeerimisel (mikrokontrollerid). Õpilased said tutvust teha C keelega ning hiljem said põgusa ülevaate Pythonist ja selle võimalustest. Koos minuga andis tunde kokku 4 inimest: juhendaja/peaõppejõud Vladimir Viies ja kolm abiõppejõudu, kellest üks olin mina ise. Juhendaja tegi sisulisi ja suuremaid otsuseid ning abiõppejõudude peamiseks ülesandeks oli vaadata, et kõik sujus hästi ning õpilased hakkaksid õppeaine raames käsitletavates teemades hästi orienteeruma. Abiõppejõudude ülesanneteks oli ka materjalide koostamine, tundide iseseisev läbiviimine ning kontrolltööde/eksamite hindamine vastavalt juhendaja korraldustele.

### 4.1 Töökeskond

Loengud ja praktikumid toimusid ICT õppehoones kolmapäeviti ruumis ICT-121/122 ja neljapäeviti ruumis ICT-501. Iga õpilane sai kasutada kooliarvutit, töö käis enamus ajast kasutades Linuxi operatsioonisüsteemi. Töövahenditeks olid õppejõul valitud Geany ja ArgoUML. Python'i käsitlemisel valisin mina töövahendiks Thonny. Geany on IDE, mis on mõeldud koos töötama GCC kompilaatoriga – neid tööriistu kasutasime C keeles programmide arendamiseks. ArgoUML on tööriist, mida kasutasime algoritmide koostamiseks ning nende esitamiseks UML diagrammidena. Thonny-t kasutasime ainult Pythoni tunni raames ning selle otsuse võtsin mina vastu, kuna mina olin selle IDE'ga Pythonit varem kirjutanud ning mul oli lihtsam oma teadmisi just Thonny't kasutades edastada.

## 4.2 Praktika maht ja ajad

Praktika toimus 2024. aasta sügissemestril, tunde sai antud nii eestikeelsele rühmale kui ka inglisekeelsele rühmale. Õppenädalaid oli kokku 16. Loengud ja praktikumid toimusid koos ühise tunnina ning iga sellise tunni kestus on 3h15min, kokku oli seega igal nädalal 6h30min puhtalt klassiruumis töötegemist. Lisaks sellele sai osaletud peale tundide lõppu 4 korda konsultatsioonis, et aidata kehvemal järjel olevaid õpilasi. Samuti sai iseseisva töö käigus valmistatud ette hulgaliselt uut õppematerjali (slaidiesitlusi, demoprogramme, õpilastele lahendada antavaid ülesandeid) nii eestikeelsele rühmale kui ka ingliskeelsele rühmale. Klassitundide pealt tuleb kokku  $6h30min \times 16 = 104h$ , konsultatsioonide pealt tuleb juurde kokku 4h. Iseseisva tööna lisandub kokku ca 50h tööd tundide ettevalmistusena. Kokku tegin töötunde praktika käigus ca 158h. Arvestades, et 1EAP maht on 26h tööd, teeb see praktika miinimumiks 156h tööd, seega täitsin mahu täiel määral.



## 5 Praktika käik

### 5.1 Iganädalane tundide läbiviimine

Kursus algas esmase enesetutvustamisega ning seejärel said õpilased ülevaate õppeaine õppetöö korraldusest. Esmalt tegid õpilased tutvust algoritimide maailmaga, nad said teadmisi, kuidas panna mingi probleem või suurem tegevus kirja, see osadeks jaotada ning sellest algoritm teha. Samuti asusid õpilased kohe ka praktiliste ülesannete kallale. Tunnis sai lahendatud erinevaid ülesandeid vastavalt sellele, mida juhendaja teha oli andnud. Abiõppejõudude ülesanneteks oli õpilaste küsimuste lahendamine, nende õigetele teele juhatamine ning tehtud tööde kontroll ja lahenduste hindamine.

Mõne nädala möödudes tutvustasime nii ingliskeelsele kui ka eestikeelsele rühmale tehisintellekti, rääkisime, mis lahendusi sellega on võimalik saavutada, kus tekivad vead ning mida kindlasti jälgida tuleks, kui soovida kasutada koodi, mis on koostatud tehisintellekti poolt. Tehisintellekti tutvustamise eesmärgil olin teinud ettevalmistavalt ka mõned demoülesanded. Võrdlesime minu lahendusi mõne tehisintellekti poolt kirjutatud lahendustega, tõime välja erinevused ning mõlema lahendusviisi head ja vead.

Uudne lähenemine Programmeerimine I aines tõi kaasa versioonihaldusplatvormide lühitutvustuse. Kuna minul oli abiõppejõududest kõige suurem kogemus versioonihaldusplatvormidega, siis võtsin selle väljakutse vastu ning võtsin need praktikumid enda juhendada. Kuna ülikool kasutab versioonihaldusplatvormina GitLab'i, siis läks see kasutusse. Antud õppeaine raames oli lubatud ka alternatiivide kasutamine, kuid lühitutvustus sai tehtud ainult GitLab'is. Minule endalegi ootamatult, võeti see soojalt vastu ning seda hakati kasutama eksamite ja kodutööde esitamisel.

Järgmistel nädalatel hakkasid pihta keerulisemad teemad, õpilased pidid hakkama jagama programmi eraldi alamfunktsioonideks ning hiljem kasutama viitasid, et tõhustada alamfunktsioonideks jagamist. Nagu arvata võis, tekitas viitade kasutamine probleeme ning nii alamfunktsioonideks jagamine kui ka viitade kasutamine nõudis oluliselt rohkem pingutust, et see õpilastele selgeks teha.

Peale viitade kasutamist ja alamfunktsioonideks jagamist oli õpilastel enam-vähem selged baasteadmised C keeles programmeerimisest ning sai hakata näitama, milleks C

keel tegelikult loodud on – toimus mikrokontrollerite labor. Labori alguses pidid õpilased end paarideks jagama ning igale paarile anti mikrokontroller. Esmalt selgitasime, kuidas mikrokontrollerile programmi peale laadida, seejärel tegime demoprogrammi, et õpilased näeksid, kuidas sellele laborile lähenema peaks. Jaotasime kätte ülesanded, mille lahendusi hakkasime hiljem kontrollima ja hindama. Õpilastele meeldis selline praktiline lähenemine ning ka minul oli huvitav abiõppejõuna selles tunnis õpetada.

Peale mikrokontrollerite laborit toimus 12. nädalal kontrolltöö, kus õpilased said koguda eksamieelduse tarbeks punkte. Eeleksamile järgnevatel nädalatel toimusid loengud ja praktikumid, mille käigus jagati lisateadmisi, kuid mis polnud selle kursuse raames kohustuslikud. Rääkisime tekstifailidest lugemisest ja nendesse kirjutamisest, programmi mitmesse faili jagamisest (header-failid ja koodifailid) ja Pythoni programmeerimiskeelest.

Viimastel õppenädalatel oli õpilastel võimalus parandada sooritusi ning sooritada eeleksameid. Abiõppejõududena me küll nendel nädalal midagi otseselt ei õpetanud, kuid olime siiski kohal juhuks, kui õpilastel tekkis mingeid küsimusi või muresid eksami või mõne muu teemaga seoses.

## **5.2 Minu poolt läbiviidud tunnid**

Lisaks üldisele õpilaste abistamisele oli ka mõnikord vaja läbi viia loenguid ja praktikume abiõppejõu vaatevinklist iseseisvalt. Kuna versioonihalduslahendused on mulle väga tuttavad ja ma valdasin neid sel ajahetkel teistest abiõppejõududest veidi rohkem, siis anti minule ülesandeks läbi viia versioonihalduslahenduste tutvustamise tund. Lisaks sellele anti mulle juhendada ka Pythoni keelt tutvustav tund, kuna mul oli Pythoniga varasem kokkupuude olemas juba gümnaasiumi aegadest.

### **5.2.1 Versioonihalduslahenduste lühitutvustus**

Versioonihalduslahenduste lühitutvustuse tunni eesmärgiks oli õpilastele anda ülevaade peamistest Git'i tööpõhimõtetest ning ülesseadmisest. Enne loengut rõhutasin, et rangelt soovituslik on Git üles seada just enda isiklikku arvutisse, kuna ülikooli arvutis töötamiseks peab iga kord uue SSH võtme Giti lisama. Loeng ja demo-sooritus andis ülevaate peamistest Git'i käsklustest käsureaal (git clone, git add, git commit, git push, git pull). Need käsklused on kõige olulisemad ning just neid kasutatakse enamus ajast.

Demo-sooritus kestis pikalt, kuna ma tegin seda õpilastega samas rütmis ja samm-sammult. Teised abiõppejõud jälgisid, et kõik jõuaksid sammu pidada ning enne ma edasi ei läinud, kui kõik jõudsid minuga toleks hetkeks samasse kohta. Minu jaoks oli see kogemus esmakordne niimoodi suuremale hulgale korraga midagi õpetada sellises loenguformaadis. Arvan, et sain sellega hästi hakkama, kuna päris suur hulk õpilasi võttis Git'i kohe ka automaatselt kasutusse oma edasistes tunnitöodes, kodutöodes, kontrolltöodes ja eksamites. Git on minu arvates kõigile programmeerimisega tegelevatel inimestel üks olulisimaid töövahendeid ja seetõttu oli see tund väga asjakohane.

### **5.2.2 Programmeerimiskeele Python lühitutvustus**

C keelt õppides on tore asi see, et päris suur hulk keeli on C keele põhjal tehtud ehk C keelt õppides on hiljem lihtsam õppida muid C keele baasil tehtud teisi programmeerimiskeeli (kõige enamkasutatavad neist on nt. Java ja C#). Python on aga hoopis teistsugune programmeerimiskeel. Pythonil on hoopis teised reeglid kui C keele põhistel programmeerimiskeeltel, seega oli Programmeerimine I kursuse raames toimunud Python'i kiire lühitutvustuse idee igati hea mõte.

Enne tundi valmistasin ette põhjalikuma slaidiesitluse, mis käsitles Pythoni ja C keele peamiste erinevuste ning koodinäiteid. Lisaks tegin ühe varasemalt tunniülesandena lahendatud programmi näite Pythonis ning tunni käigus võrdlesin seda C keeles kirjutatud programmiga, et õpilastele teooriat paremini edasi anda.

Peale demo-võrdlust andsin esimeseks ülesandeks lihtsa ülesande, mille käigus tuli õpilastel realiseerida vastavalt algoritmile üks programm Pythoni keeles, palusin neil tehisintellekti abi mitte kasutada, kuid minu materjalide kasutamine olid lubatud. Hilisemates ülesannetes oli lubatud tehisintellekti kasutamine.

Õpilastele tundus Python'i õppimine lihtne, kuna Python on oma olemuselt oluliselt lihtsam keel, kui C keel. Python'i puhul käsitlesime ainult protseduurilist lähenemist, objektorienteeritud programmeerimine oleks minu hinnangul paljusid liigselt segadusse ajanud. Hiljem tunnitöid kontrollides sai selgeks, et õpilased võtsid sellegi tunni sisu väga hästi vastu ning kirjutasid päris häid lahendusi.

### 5.3 Õppematerjalide koostamine

Õppematerjalide koostamine oli üks osa minu iseseisva tööst selle praktika raames. Kuna ma juhendasin abiõppejõuna kahes keeles rühmi, siis tuli ka õppematerjale ümbertõlkida. Esmalt koostasid õppematerjalid inglise keeles ning seejärel üritasin need tõlkida tehniliselt korrektsesse ja üheselt mõistetavasse eesti keelde.

Esimesed õppematerjalid tulid kasutusse kohe esimestel nädalatel, kirjutasin kodus mitmeid demoprogramme, paar slaidiesitlust, näiteülesandeid ning nende lahendusi. Nende kohta küll õpilastelt tagasisidet ei saanud, kuid loodan, et need materjalid aitasid neid teadmiste omandamisel.

Lisaks otseste õppematerjalide koostamisele oli oluline ka enda ettevalmistamine, et mõelda läbi, millest ma Giti ja Pythoni tunnis näiteks räägin. Vaatamata sellele, et slaidid on põhjalikud, tuli suur osa teooriast minu jutust. Üritasin vältida slaidide kordamist ja proovisin märkida slaididele rohkem üldist teavet, millele ma ise suuliselt juurde räägin.

Õppematerjalidele lisaks oli mul au valmistada ette ka üks kontrolltöövariant. Kontrolltöö sisule seadis ette nõuded juhendaja ning selliste suuniste järgi hakkasingi. Kontrolltöös pidi olema kolm ülesannet: UML diagrammi järgi programmikoodi kirjutamine, programmeerimisülesanne ja keerukam programmeerimisülesanne koos algoritmi loomisega. Koostasid kontrolltöö nii eestikeelses versioonis (vt. Lisa 1) ja ingliskeelses versioonis (vt. Lisa 2).

### 5.4 Kontrolltööde ja eksamite hindamine

Minu ülesannete hulka kuulus ka kokku kuue kontrolltöö hindamine. Täpsemaks ülesandeks oli vaadata igat tööd ning analüüsida, kui hästi on õpilane hakkama saanud, tuua välja töö puudused ning teha nendest kokkuvõte. Lisaks pidi andma hinnangu, mitme punkti vääriline antud töö on. Punktid ja arvamus pidi seejärel juhendajale edastama ning seejärel tegi juhendaja otsuse, kas ta on minu arvamus ja punktisummaga nõus või mitte. Minu tagasiside oli konstruktiivne ning tavaliselt läksid need punktid ka päriselt tulemusena kirja, mille mina olin mingi konkreetse töö eest andnud.

## **6 Praktika kokkuvõte ja arvamus**

Läbitud praktika andis mulle hea ettekujutuse õppejõu tööst. Lisaks võimaldas see mul uuesti üle korrata varem õpituid teadmisi selles aines. Arvan, et see praktika andis mulle väga hea tõuke enesearengu mõttes, ma polnud varem midagi niimoodi sellise hulga rahva ees otseselt õpetanud. Töö nõudis küll kannatlikkust, kuid hiljem mõistsin, et minul oli otsene roll nende õpilaste teadmiste portfelli kujunemisel. Koostöö juhendajaga sujus suurepäraselt ning õpilaste tagasisidest lähtudes sain oma ülesannetega hästi hakkama. Leian, et õpetamispraktika on suurepärase alternatiiv tavapärasele tööstuspraktikale ning soovitan õpetamispraktikat just neile, kel on vaja arendada kannatlikkust ning enesekindlust ja veenvust suurema hulga rahva ees esinemisel.

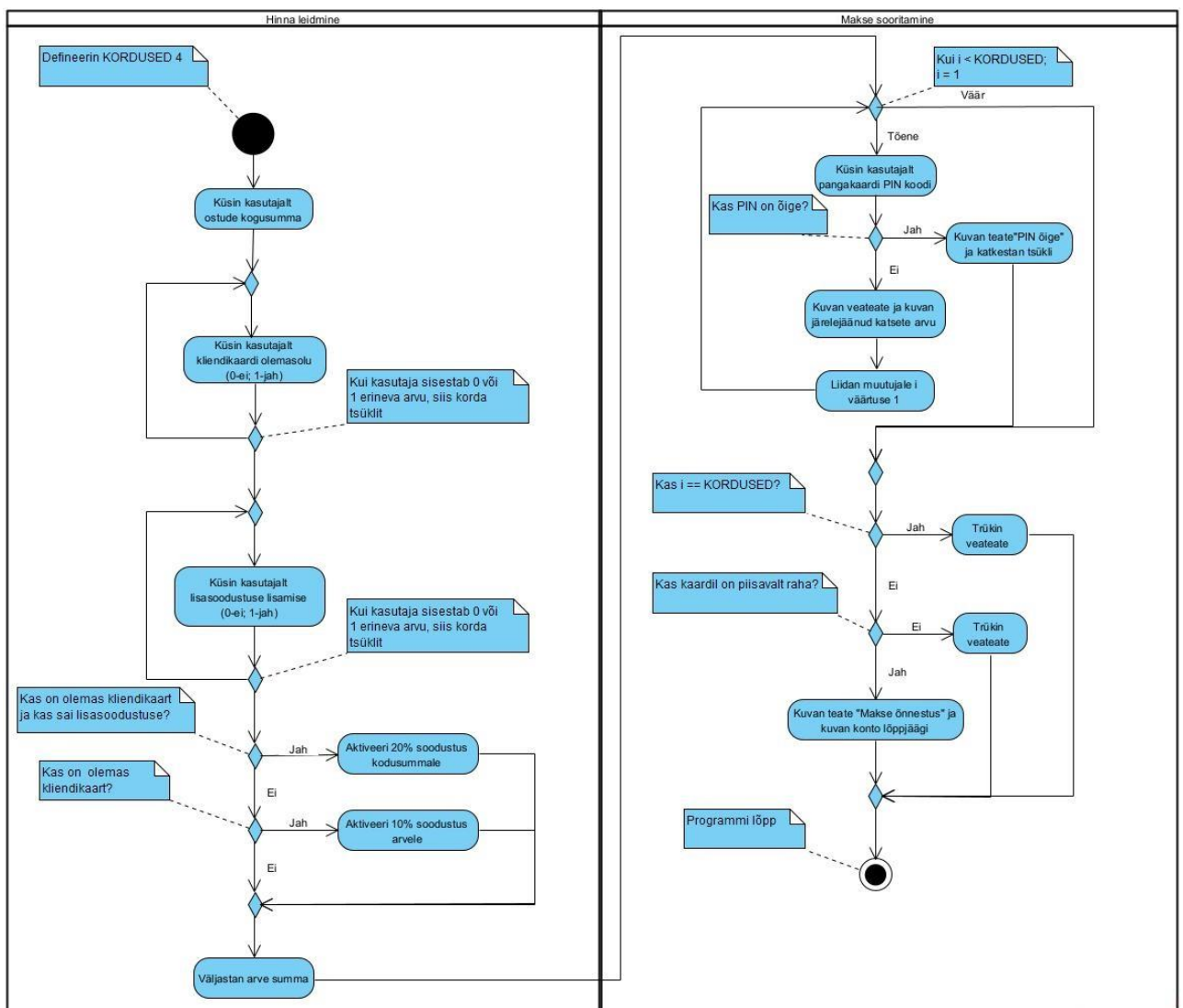
## 7 Kasutatud allikad

- [1] „Arvutisüsteemide instituut | Tallinna Tehnikaülikool (TalTech),“ TalTech, [Võrgumaterjal]. Kättesaadav: <https://taltech.ee/arvutisusteemide-instituut/instituut>. [Kasutatud 14.01.2025].
- [2] „Ülikooli lugu | Tallinna Tehnikaülikool (TalTech),“ TalTech, [Võrgumaterjal]. Kättesaadav: <https://taltech.ee/ulikool/ulikooli-lugu>. [Kasutatud 14.01.2025].

## Lisa 1 – Minu koostatud kontrolltöö eesti keeles

### Kontrolltöö

**ÜLESANNE 1:** Allpool on antud UML diagramm (algoritm), mis kirjeldab supermarketi kassasüsteemi: leiab toodete hinna, arvutab soodustused ning tegeleb maksetega. Kirjuta C-keeles sellele vastav programmi kood kasutades seni õpituid parimaid praktikaid (funktsioonideks jagamine, kordused jms). (10p)



**ÜLESANNE 2:** Kirjuta C-keeles programm, mis: võtab kasutajalt täisarvude massiivi, kasutab funktsioone ja viitasid, et eraldada paaris- ja paaritud arvud erinevatesse massiividesse ja väljastab paaris- ja paaritud arvud eraldi. Kasuta seniõpituid parimaid praktikaid (funktsioonideks jagamine, viidad, kordused jms). (10p)

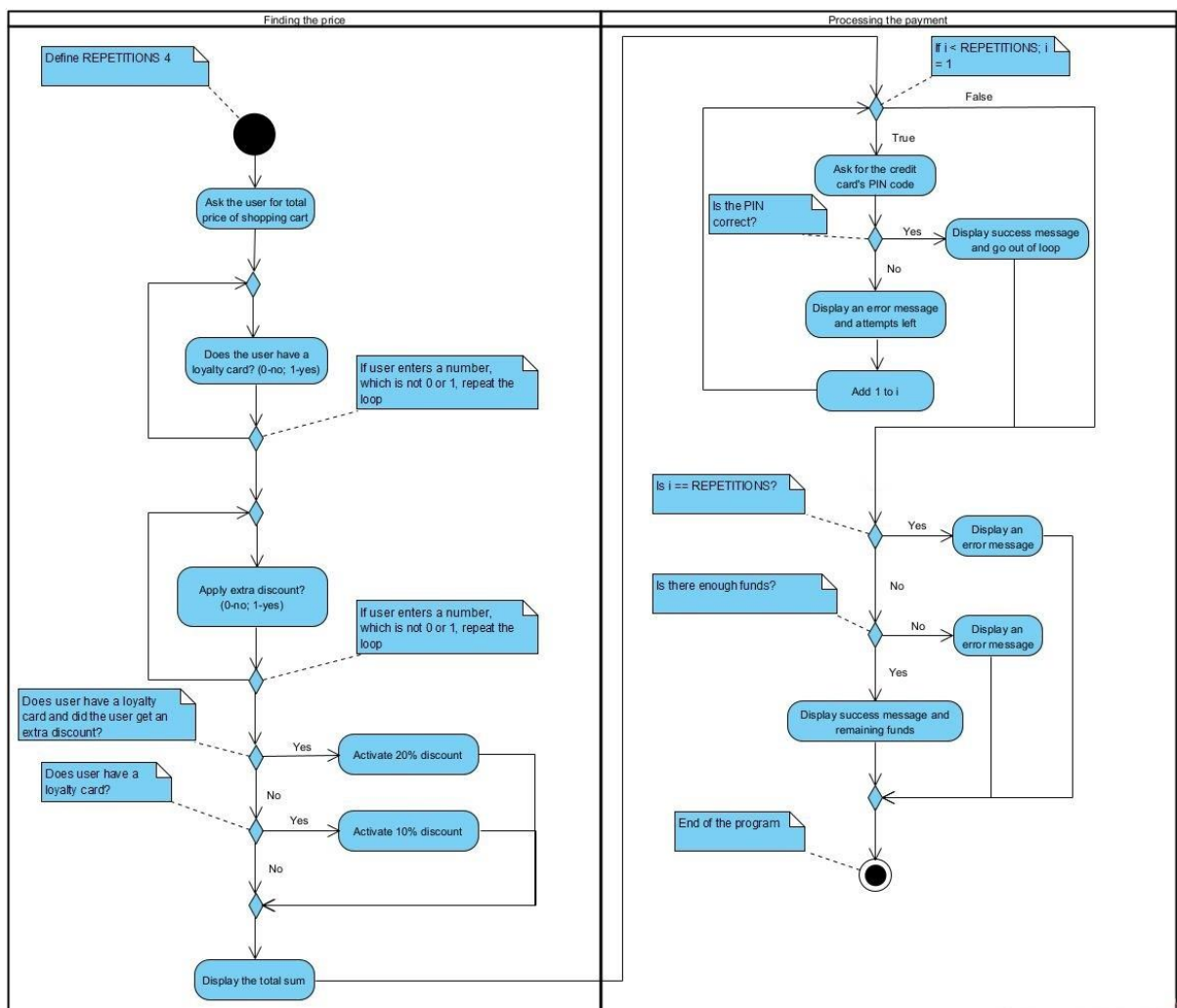
**ÜLESANNE 3:** Koosta algoritm ja kirjuta C-keeles programm, mis: võtab saab sündiks juhuslike täisarvude massiivi, kasutab mullsorti, et sorteerida massiiv kasvavas järjekorras ja väljastab sorteeritud massiivi. Kasuta seniõpituid parimaid praktikaid (viidad ja funktsioonideks jagamine jms). (15p)



## Lisa 2 – Minu koostatud kontrolltöö inglise keeles

### Semi-exam

**TASK 1:** Below is a UML diagram (algorithm) that describes the supermarket cash register system: it finds the prices of products, calculates discounts, and handles payments. Write the corresponding program code in C, applying the best practices you have learned so far (functions besides main(), loops, etc). (10p)



**TASK 2: Write a C program that: takes an array of integers from the user, uses functions and pointers to separate even and odd numbers into different arrays, and outputs the even and odd numbers separately. Use best practices learned so far (such as functions besides main(), pointers, loops, etc). (10p)**

**TASK 3: Create an algorithm and write a C program that: takes an array of random integers as input, uses bubble sort to sort the array in ascending order, and outputs the sorted array. Use best practices learned so far (such as pointers, functions besides main(), etc.). (15p)**