

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

Aleksander Laasmägi 253134IAPM

ÕPETAMISPRAKTIKA

Praktika ITI8750

Juhendaja: Tarmo Robal
PhD

Tallinn 2026

Lühendite ja mõistete sõnastik

<i>Branch</i> (haru)	Versioonikontrolli mõiste, mis tähistab eraldiseisvat arendusliini koodivaramus
C#	Microsofti loodud objektorienteeritud programmeerimiskeel .NET ökosüsteemis
<i>Commit</i>	Versioonikontrollisüsteemi muudatus, mis salvestab koodimuutused koodivaramusse
EAP	Euroopa ainepunkt — ülikooli õppemahu mõõtühik, millele vastab 26 tundi tööd
EF Core (Entity Framework Core)	Microsofti objekt-relatsiooniline andmebaaside halduse raamistik .NET ökosüsteemis
GitHub Classroom	GitHubi õppekeskkond, mis võimaldab õppejõududel tudengite koodivaramuid hallata
IDE	<i>Integrated Development Environment</i> - Integreeritud programmeerimiskeskond/arenduskeskkond
Moodle	Avatud lähtekoodiga õpiahaldussüsteem, mida TalTech kasutab õppematerjalide jagamiseks
.NET	Microsofti tarkvaraarenduse raamistik, mis toetab mitut programmeerimiskeelt
OOP	Objektorienteeritud programmeerimine — programmeerimisparadigma, mis põhineb objektidel ja klassidel
Razor Pages	Microsofti veebirakenduste loomise raamistik .NET ökosüsteemis
Sprint	Agiilse tarkvaraarenduse lühike ja fikseeritud pikkusega iteratsioon
UML	<i>Unified Modeling Language</i> — ühtne modelleerimiskeel tarkvarasüsteemide visuaalseks kirjeldamiseks
VB.NET	Visual Basic .NET — Microsofti programmeerimiskeel .NET raamistikus
WPF	Windows Presentation Foundation — Microsofti kasutajaliideste loomise raamistik .NET ökosüsteemis

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.....	7
5 Praktika käik.....	9
5.1 Õppematerjalide kaasajastamine	9
5.2 Praktikumide läbiviimine	10
5.3 Tudengite koodivaramute ülevaatamine.....	11
6 Praktika kokkuvõte ja hinnang	12
7 Kasutatud allikad	13

1 Sissejuhatus

2025/2026 õppeaasta kevadsemestril avanes mul võimalus sooritada praktika Tallinna Tehnikaülikoolis abiõppejõuna. See sobis mulle hästi, kuna mul oli varasem kogemus juba abiõppejõuna olemas ning ma olen alati avatud võimalusele arendada endas oskust teadmisi ja informatsiooni arusaadavalt teistele edasi anda. Olulist rolli mängis ka asjaolu, et abiõppejõu koht pakuti mulle just tarkvaratehnika õppeaine raames. Tarkvaratehnika oli minu bakalaureuseõppe üks kasulikumaid aineid, kuid vajab minu hinnangul ammu kaasajastamist – millega saingi praktika käigus tegelema asuda.

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. TalTech'i 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
- Biorobotika keskus
- Keskkonnaseire tehnoloogiate keskus
- Riistvara turvalisuse keskus
- Nutika riistvara uurimiskeskus
- Sardtehisintellekti labor
- Arvutisüsteemide õppekeskus

3 Ootused praktikale

Läksin praktikale ootusega, et saan seekord rohkem vastutust ja otsustusõigust õppematerjali kujundamisel. Varasemas abiõppejõu rollis olin pigem täitja – juhendasin ja abistasin tudengeid juba valmis materjali põhjal. Seekord lootsin, et saan olla rohkem kaasatud ka sisulistesse otsustesse: millist materjali kaasajastada, kuidas ülesandeid koostada ning kuidas ainet tervikuna paremaks muuta.

Samuti ootas, et bakalaureuseõppe ja senise magistriõppe jooksul omandatu saab lõpuks praktilise väljundi. Teoorias arutletud põhimõtteid on hoopis teine tunne päriselt rakendada – eriti sellises aines, mis kätkeb endas tarkvara arendamise protsessi tervikuna. Lootsin, et praktika käigus tekib ka selgem arusaam sellest, kuidas ülikooliõpe ja tööstuse tegelikud vajadused omavahel joonduvad – või ei joondu.

4 Praktikakoha kirjeldus

Tarkvaratehnika (IAS0110) õppeaine raames õpetatakse riistvara arenduse ja programmeerimise, ehitiste projekteerimise ja ehitusjuhtimise ning tootearenduse ja robotika õppekavade teise kursuse tudengitele tarkvaraarenduse terviklikku elutsüklit — alates nõuete analüüsist ja UML-modelleerimisest kuni erinevate arendusmetoodikate ning kvaliteedijuhtimise mudeliteni. Kursuse praktilise osana õpivad tudengid kirjeldama lähteülesannet UML-keeles ning realiseerima nõutava funktsionaalsuse komponenttehnoloogias. Minu praktika keskendus õppeaine praktilise poole sisu kaasajastamisele ning praktikumitundide läbiviimisele kaasaaitamisele.

Koos minuga andis praktikumitunde kokku 5 inimest: kaks õppejõudu ja kolm abiõppejõudu, kellest üks olin mina ise. Õppejõud tegid sisulisi ja suuremaid otsuseid ning abiõppejõudude peamiseks ülesandeks oli vaadata, et kõik sujuks hästi ning tudengid hakkaksid õppeaine raames käsitletavates teemades hästi orienteeruma. Lisaks materjalide kaasajastamisele ja praktikumitundide läbiviimisele kaasaaitamisele oli minu ülesandeks ka tudengite praktikumide/projekti soorituste hindamine vastavalt õppejõudude korraldustele.

4.1 Töökeskkond

Peamiseks arendusvahendiks oli õppejõudude poolt valitud Visual Studio, mis on Microsofti loodud võimas ja laialt kasutatav integreeritud arenduskeskkond (IDE). Visual Studio toetab C# programmeerimiskeelt, mida kursuse raames kasutati komponenttehnoloogia põhimõtete õpetamiseks. C# sobib selleks hästi, kuna tegemist on tänapäevase objektorienteeritud keelega, mis toetab komponentpõhist arendust ning on tööturul laialdaselt kasutusel.

4.2 Praktika maht

Praktika kogumaht jagunes kaheks suureks osaks: õppeaine materjalide kaasajastamine ning praktikatundide läbiviimine ja tudengitega tegelemine.

Õppematerjale, mida oli vaja kaasajastada, oli nelja slaidikomplekti jagu, iga slaidikomplekt sisaldas keskmiselt 88 slaidi. Materjalide kaasajastamise töötundide mahuks tuli ca 110h.

Praktikatunnid toimusid 2025/2026 õppeaasta kevadsemestril, tunde sai antud nii eestikeelsele rühmale kui ka inglisekeelsele rühmale. Õppenädalaid oli kokku 16, kuid viimasel üheksal nädalal toimusid praktikatunnid vaid paaritutel õppenädalatel. Praktikatunnid toimusid järjestikuste paarissessioonidena. Iga selline sessioon kestis 3h15min, kokku oli seega igal nädalal 6h30min puhtalt klassiruumis töötegemist. Lisaks sai panustatud aega ca 30min igal nädalal tudengite koodivaramutega tutvumisel ning tehtud töö kontrollimisel. Praktikatundide ja tudengite tööde kontrollimise kogumahuks tuli arvutuslikult:

$$7 * 6,5h + 4 * 6,5h + 16 * 0,5h = 45,5h + 26h + 8h = \mathbf{79,5h}.$$

Kokku sai sooritatud töötunde praktika käigus $110h + 79,5h = \mathbf{189,5h}$. Arvestades, et 1EAP maht on 26h tööd, teeb see praktika miinimumiks 156h tööd, seega täitsin mahu täiel määral.

5 Praktika käik

Käesolev peatükk annab ülevaate praktika käigust kolmes alapeatükis: õppematerjalide kaasajastamine, praktikumide läbiviimine ning tudengite koodivaramute ülevaatamine. Kõik kolm tegevusvaldkonda olid omavahel tihedalt seotud — kaasajastatud materjalid said kohe praktikumides kasutusele võetud ning koodivaramute ülevaatamine toimus paralleelselt nii juhendite lahendamise kui ka projektitöö faasis.

5.1 Õppematerjalide kaasajastamine

Algsed õppematerjalid koosnesid neljast eestikeelsest slaidikomplektist, mis olid koostatud Visual Basic .NET (VB.NET) programmeerimiskeele, .NET Framework 4.7.2 raamistiku ja Windows Forms kasutajaliideste loomise raamistiku baasil. Kuna eelpool nimetatud tehnoloogiad on infotehnoloogiasektori tööturul kasutatavate tehnoloogiate hulgast väljas, tegin õppejõududele ettepaneku kogu Tarkvaratehnika õppeaine praktilise osa materjalid põhjalikult üle vaadata ja kaasajastada.

Planeerisime õppejõududega koostöös pikalt värskemate tehnoloogiate valikut ning kursuse uut ülesehitust. Eesmärgiks oli säilitada kogu .NET raamistiku põhine struktuur, kuid vahetada välja programmeerimiskeel, .NET raamistiku versioon ja kasutajaliideste loomise raamistik.

.NET'i raamistik toetab kolme programmeerimiskeelt – Visual Basic, C# ja F#. Kuna Visual Basic on ajale jalgu jäänud, oli valida C# ja F# programmeerimiskeelte vahel. F# on mõeldud funktsionaalprogrammeerimiseks, mis langeb Tarkvaratehnika õppeaine skoobist välja ning seega sai valitud uueks kursuse raames käsitletavaks programmeerimiskeeleks C#, mis on .NET'i ökosüsteemis domineeriv ja tööturul laialdaselt nõutud objektorienteeritud programmeerimiskeel.

Kasutajaliidese raamistiku valikul lähtuti kahest peamisest kriteeriumist: raamistik pidi töötama ICT-501 klassiruumis asuvatel lauaarvutitel ning sobima tudengitele, kellel puudub varasem kasutajaliideste arendamise kogemus. Nende kriteeriumide põhjal valiti Windows Presentation Foundation (WPF), mis on Microsofti loodud kasutajaliideste raamistik .NET'i ökosüsteemis. WPF-i peamine eelis on selle visuaalne arenduskeskkond, kus tudengid saavad kasutajaliideseid luua lihtsalt hiirega komponente

lohistades — see teeb algajatel alustamise oluliselt lihtsamaks. Lisaks toetasid WPF-i kasutamist ka klassiruumis olevad arvutid, mistõttu polnud vaja täiendavaid tarkvaramuudatusi teha.

Viimaseks tuli valida kasutatav .NET raamistiku versioon. Siin oli otsustav kriteerium üks — klassiarvutite toetatavus. Sellest lähtuvalt valiti .NET 9.0.

Pärast valikute tegemist sai hakata praktikatundide materjale kaasajastama. Kaasajastamise käigus asendati kõik VB.NET-spetsiifilised koodinäited ja süntaksiselgitused C# keelele vastavate alternatiividega. Lisaks uuendati materjale kaasaegsete lähenemisviisidega — tutvustati C# tänapäevaseid keelekonstruktsioone ning võeti kasutusele Entity Framework Core'i (EF Core) andmebaasikihiga töötamise lähenemine, mis esindab tänapäevast tarkvaraarenduse praktikat objekt-relatsioonilise andmebaaside halduse valdkonnas.

Muutmata jäeti objektorienteeritud programmeerimise (OOP) teoreetiline osa, kuna OOP aluspõhimõtted — nagu kapseldamine, pärimine ja polümorfism — on programmeerimiskeelest sõltumatud ning kehtivad võrdväärselt nii VB.NET-i kui ka C# kontekstis.

Materjalide kaasajastamine oli mahukam protsess, kui esmapilgul võis tunduda — pelgalt süntaksi asendamisest ei piisanud, kuna C# ja VB.NET erinevad üksteisest nii ülesehituse kui ka parimate praktikate poolest. Kogu töö tuli teostada sisulise kvaliteedi säilitamise eesmärgil, tagades, et uuendatud materjalid ei oleks mitte üksnes keeleliselt korrektsed, vaid ka pedagoogiliselt terviklikud.

5.2 Praktikumide läbiviimine

Praktikumid olid üles ehitatud nii, et tudengid said juhendmaterjalid Moodle'i õpikeskkonnast ning asusid seejärel iseseisvalt nende põhjal lahendusi realiseerima. Abiõppejõudude rolliks oli tudengeid jälgida, tekkivatele küsimustele vastata ning vajaduse korral suunata neid õige lahenduse suunas — sekkumata liigselt, et tudengid jõuaksid ise järeldusteni.

Lisaks juhendmaterjalide lahendamisele oli üks praktikum pühendatud UML-diagrammide koostamisele, kus tudengid harjutasid oma lahenduste visuaalset

modelleerimist. Samuti toimus kursuse lõpuosas eraldi praktikum, milles tudengid koostasid oma lõpuprojekti tarbeks kasutajalood — agiilses tarkvaraarenduses üks kesksemaid vahendeid nõuete kirjeldamiseks.

Kuna tudengid lahendasid praktikumide raames uuendatud juhendmaterjale, oli tagasiside kogumine eriti oluline. Pärast iga juhendi sooritamist kogusin tudengitelt tagasisidet — küsisin, mis oli arusaadav, mis tekitas segadust ning kas juhendi ülesehitus oli loogiline. See aitas tuvastada kohti, kus materjalid vajasisid täiendavat selgitust või ümberstruktureerimist. Kogutud tagasiside edastas tulevastele versioonidele kasulikku infot selle kohta, kuidas tudengid materjali tegelikkuses tajuvad ning millised osad vajavad edasist lihvimist.

Lisaks tagasiside kogumisele oli minu ülesandeks hinnata iga tudengi sooritust vastavalt õppejõudude poolt kehtestatud kriteeriumitele. Hindamine ei piirdunud üksnes tulemuse kontrollimisega — vajadusel andsin tudengitele ka sisulist tagasisidet nende lahenduse kohta, et toetada nende edasist arengut.

5.3 Tudengite koodivaramute ülevaatamine

Koodivaramute ülevaatamine toimus kahes erinevas faasis. Enne projektide käivitumist vaatasin jooksvalt üle tudengite juhendilahendusi, mida nad esitasid GitHub Classroom'i kaudu. See andis ülevaate sellest, kuidas tudengid materjaliga toime tulevad ning kas esineb korduvaid probleeme.

Alates 7. õppenädalast, kui projektid käivitusid, leidis koodivaramutel põhinev ülevaatamine aset sprintide keskel. Ülevaatuse käigus jälgisin mitut aspekti: commit-sõnumite selgust ja informatiivsust, *commit*'ide suurus, harude (*branch*) kasutamist ning koodi ennast.

Ülevaatuse käigus oli oluline hoida fookus õpieesmärkidele vastavatel aspektidel — eelkõige süntaksil, nimetamiskonventsioonidel ja muudel selgetel koodikirjutamise tavadel. Arhitektuursed valikud ja isiklikud eelistused selle kohta, kuidas ülesannet teisiti lahendada võiks, jätsin teadlikult tagaplaanile, kuna tudengite lahenduste hindamine oma lahendamisstiili prisma läbi oleks olnud ebaõiglane ning pärssiks nende iseseisvat mõtlemist.

6 Praktika kokkuvõte ja hinnang

Praktika käigus kaasajastasin Tarkvaratehnika õppeaine praktikumimaterjalid ning osalesin aktiivselt praktikumitundide läbiviimises ja tudengite tööde hindamises. Kokkuvõttes kujunes praktika positiivseks ja arendavaks kogemuseks, mis pakkus väärtust nii isiklikul kui ka erialalisel tasandil.

Läksin praktikale ootusega saada rohkem vastutust ja otsustusõigust õppematerjali kujundamisel — ning see ootus täitus. Erinevalt varasemast abiõppejõu rollist sain seekord olla kaasatud sisulistesse otsustesse: millist materjali kaasajastada, milliseid tehnoloogiaid valida ning kuidas ülesandeid üles ehitada. Samuti lootsin, et bakalaureuseõppe ja magistriõppe jooksul omandatu saab praktilise väljundi — ka see leidis kinnitust, eriti tehnoloogiavalikute põhjendamisel ja materjalide koostamisel. Küsimus sellest, kuidas ülikooliõpe ja tööstuse tegelikud vajadused joonduvad, sai samuti selgema vastuse — just see pingeväli motiveeris materjalide kaasajastamist ning on ka WPF-i asendamise ettepaneku taga.

Praktika suurimaks isiklikuks kasuks osutus teadmiste ja ideede edastamise oskuse arenemine. Tudengitele materjali selgitamine nõuab, et ise mõistaksid käsitletavat teemat põhjalikult ning suudaksid seda esitada selgelt ja arusaadavalt — see sundis mind oma mõtteid täpsemalt sõnastama. Lisaks kordas praktika C# programmeerimiskeele teadmisi, kuigi tegemist oli juba varasemalt tuttava keelega.

Ettepanekuna tulevaseks arendusvõimaluseks tooksin välja kasutajaliidese raamistiku uuendamise. Praegu kasutatav WPF on funktsionaalne lahendus, kuid selle peamiseks puuduseks on platvormisõltuvus — WPF töötab ainult Windowsis, mis muudab rakenduste käivitamise ja kontrolli teistel platvormidel, näiteks macOS-il, võimatuks. Alternatiivina võiks kaaluda Razor Pages'i, mis õpetaks tudengitele veebirakenduste arendust — tänapäeval üht olulisimat tarkvaraarenduse valdkonda. Razor Pages'i kasutuselevõtt ei ole siiski probleemideta — erinevalt WPF-ist puudub seal visuaalne lohistamiskeskond ning tudengid peaksid kasutajaliideseid looma HTML-i ja Bootstrapi abil, mis on algajatele mõnevõrra vähem intuiitiivne lähenemine. Sellest hoolimata usun, et pikemas perspektiivis oleks see muutus tudengitele kasulik, kuna annaks neile relevantsemat kogemust tööturul oodatavate oskuste arendamiseks.

7 Kasutatud allikad

- [1] TalTech, „Ülikooli lugu | Tallinna Tehnikaülikool (TalTech),“ TalTech, 10.05.2026. [Võrgumaterjal]. Kättesaadav: <https://taltech.ee/ulikool/ulikooli-lugu>. [Kasutatud 10.05.2025].
- [2] TalTech, „Arvutisüsteemide instituut | Tallinna Tehnikaülikool (TalTech),“ TalTech, 10.05.2026. [Võrgumaterjal]. Kättesaadav: <https://taltech.ee/arvutisusteemide-instituut/instituut>. [Kasutatud 10.05.2026].