

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Annabel Maksimov
Pythoni algkursuse õppematerjalid põhikoolile

Bakalaureusetöö (9 EAP)

Juhendaja:
Tauno Palts, PhD

Tartu 2025

Pythoni algkursuse õppematerjalid põhikoolile

Lühikokkuvõte:

Eesti põhikooli riiklikku õppekava järgi on informaatika valikõppeaine, mistõttu sõltub selle õpetamine suuresti iga kooli otsusest. Selleks, et toetada programmeerimisõpet koolides ja muuta see kättesaadavamaks, on vaja vanusele ning praktilistele tundidele vastavaid õppematerjale. Bakalaureusetöö eesmärk on luua põhikooli 6.–7. klassi õpilastele mõeldud Pythoni algkursuse õppematerjalid, mis aitavad neil omandada programmeerimise põhiluseid ja tekitada huvi IT vastu. Merrilli mudeli põhjal loodud materjal koosneb kuuteistkümnest 60-minutilise tunnist, mis on kättesaadav veebikeskkonnas Courses. Kursus arendati ADDIE mudeli põhjal, kus iga tunni lõpus koguti õpilastelt tagasisidet, et hinnata ning täiustada materjale vastavalt õppijate vajadustele. Tagasisidest selgus, et õpilased hindasid kursust kõrgelt ja loodud õppematerjalid on sobivad Pythoni põhialuste õpetamiseks põhikoolis.

Võtmesõnad: Informaatika, põhiharidus, programmeerimise õppematerjalid, Python

CERCS: S270 Pedagoogika ja didaktika, P175 Informaatika, süsteemiteooria

Python Introductory Course Materials for Basic School

Abstract:

According to the Estonian national curriculum for basic schools, informatics is an elective subject, and its teaching depends on the decisions of each school. To support programming education and make it more accessible, age-appropriate and practical learning materials are needed. The bachelor's thesis aims to create introductory Python materials for 6th- and 7th-grade students to help them acquire the basic programming concepts and encourage an interest in IT. Based on Merrill's model, the material consists of sixteen 60-minute lessons, which are accessible via the Courses online platform. The course was developed using the ADDIE model, with student feedback collected after each lesson to evaluate and improve the materials according to learners' needs. The feedback showed that students viewed the course positively, and the created materials are suitable for teaching the basics of Python to younger learners.

Keywords: Informatics, basic education, programming learning materials, Python

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

Sisukord

1. Sissejuhatus.....	5
2. Teoreetiline taust.....	7
2.1 Programmeerimine põhikooli riiklikus õppekavas	7
2.2 Programmeerimiskeel Python.....	9
2.3 Programmeerimiskeskond Thonny	9
2.4 Automaatkontrollikeskkond Lahendus	9
2.5 Varasemad eestikeelsed Pythoni õppematerjalid.....	12
2.6 ADDIE mudel	13
2.7 Merrilli mudel	15
3. Metoodika	17
3.1 Valim.....	17
3.2 Kursuse ja õppematerjalide koostamise protsess ja vahendid	17
3.3 Tagasiside kogumine	19
4. Tulemused.....	21
4.1 Avaküsimustiku vastused.....	21
4.2 Kursuse planeerimine.....	23
4.3 Valminud õppematerjal.....	25
4.4 Õppematerjal ühe tunni näitel	26
4.4.1 Merrilli mudel: ülesande- või probleemikesksus.....	26
4.4.2 Merrilli mudel: aktiveerimine	27
4.4.3 Merrilli mudel: demonstreerimine	28
4.4.4 Merrilli mudel: rakendamine	29
4.4.5 Merrilli mudel: lõimimine.....	30
5. Õpilaste tagasiside kursusele	32
6. Kokkuvõte.....	38
Viidatud kirjandus.....	39
Lisad.....	41
I. Lapsevanema nõusoleku vorm	41
II. Kursuse esimesel tunnil läbiviidud avaküsimustik	42
III. Tunni lõpus läbi viidud tagasiside ankeet.....	45
IV. Kursuse ainekava	46
V. Eestikeelsete Pythoni õppematerjalide analüüs	53

Litsents.....55

1. Sissejuhatus

Digitaalses ühiskonnas puudutab tehnoloogia areng kõiki eluvaldkondi. Nii on üha olulisem, et vajalikud digitaalsed oskused omandataks juba varases eas. Tänu digitehnoloogia kiirele arengule on meie igapäevaelu järjest rohkem seotud arvutite, nutiseadmete ja internetiga, mis muudab programmeerimise oskuse üha olulisemaks nii IT-valdkonnas kui ka igapäevastes tegevustes ja erinevates töövaldkondades.

Pelgalt tehnoloogia tarbijatest ei piisa – ühiskond vajab inimesi, kes mõistavad, kuidas digitaalsed süsteemid toimivad ja suudavad ise luua uusi tehnoloogilisi lahendusi. Programmeerimisoskus ei piirdu üksnes koodi kirjutamisega – see annab noortele võimaluse mõista, kuidas digitaalsed süsteemid toimivad, ning loob aluse tehnoloogiliste lahenduste loomisele. Nii arenevad õppijatest teadlikud tehnoloogia kasutajad ja selle aktiivsed kujundajad ja arendajad.

Ehkki vajadus programmeerimisoskuse järele kasvab, sõltub selle õpetamine Eesti koolides suuresti iga kooli enda otsustest. Eesti põhikooli riikliku õppekavas on programmeerimine üks osa informaatika valikainest, kuid informaatika ei ole kohustuslik õppeaine [1]. Nii jääb koolidele valik kas ja kuidas programmeerimise õpetamist pakkuda. Tihti piirub informaatikatund sageli ainult arvuti igapäevase kasutamise tutvustamisega ning ei keskendu programmeerimisele. See aga ei pruugi õpilastes tekitada piisavat huvi ega oskusi programmeerimise ja IT vastu määral, mis on vajalikud tehnoloogiliste lahenduste loomiseks ja edasiarendamiseks.

Lõputöö püüab sellele probleemile lahendust pakkuda. Töö eesmärk on luua põhikooli õpilastele praktiliselt kasutatavad ja arusaadavad Pythoni algkursuse õppematerjalid.

Bakalaureusetöö koosneb neljast osast. Esimeses osas käsitletakse teoreetilist tausta, mille hulka kuuluvad ülevaade programmeerimise asetusest informaatika õppekavas Eesti põhikooliastmes, Pythoni programmeerimiskeele tutvustus, samuti kasutatavate õpi-keskkondade, nagu Lahendus ja Thonny tutvustus. Antakse ka ülevaade valitud mudelitest nagu ADDIE ja Merrill, mis suunavad kogu õppematerjalide koostamise protsessi. Samuti käsitletakse viit eestikeelset Pythoni õppematerjali ja sobivust algajatele.

Teises osas kirjeldatakse uurimismeetodit, sealhulgas uurimisprotsessi, valimit, kursuse ja õppematerjalide koostamise etappe ning tagasiside kogumist küsitluste abil.

Kolmandas osas esitatakse töö praktilisi tulemusi ja kogutud tagasisidet. Esmalt käsitletakse avaküsimustiku vastuseid, seejärel kirjeldatakse kursuse planeerimist ja loodud õppematerjale. Näitena tuuakse välja ühe tunni ülesehitus, mida analüüsitakse süstemaatiliselt Merrilli mudeli viie etapi alusel.

Töö neljas osa sisaldab õpilaste tagasisidet ja selle analüüsi. Esitatakse tunni lõpus kogutud hinnangud, et selgitada õppematerjali sobivus põhikooli 6.-7. klassile.

Lõputöö lisas on lapsevanema nõusoleku vorm, kursuse esimesel tunnil läbiviidud avaküsimustik, iga tunni lõpu tagasiside ankeet, kursuse ainekava ja eestikeelsete Pythoni õppematerjalide analüüs.

Loodud materjal on avalikult kättesaadav ja võib olla kasulik nii õpilastele, kes saavad omandada Pythoni põhialuseid, kui ka õpetajatele, kes saavad kasutada neid materjale ja kursuse tagasisidet oma õppetöö täiustamiseks ja õpetamisprotsessi tõhustamiseks.

2. Teoreetiline taust

Teoreetilise tausta peatükk annab ülevaate programmeerimise õpetamisest Eesti põhikoolides, keskendudes põhikooli informaatika õppekavale. Lisaks sellele tutvustatakse programmeerimiskeelt Python, mille lihtne süntaks sobib algajatele. Samuti räägitakse keskkondadest, mida kasutati õppematerjalide loomisel ja katsetamisel – programmeerimise keskkonnast Thonny¹, mis on programmeerimiskeskond loodud just algajatele ja automaatkontrolli keskkonnast Lahendus², mis pakub kiiret automaatset tagasisidet programmeerimisülesannete lahendustele.

2.1 Programmeerimine põhikooli riiklikus õppekavas

Tänapäeval on oluline, et õpilased tutvuksid programmeerimisega juba varakult, kuna see oskus on vajalik igapäevaelus, töökohal ja digitaalses maailmas toimetulemiseks [2]. Õpilased alustavad arvutitega tutvumist sageli arvuti igapäevase kasutamise tutvustamisega. Eesti põhikooli riikliku õppekava üldpädevuste arendamise kohustuslik osa on ka digipädevus, mis hõlmab oskust kasutada digitehnoloogiat õppimiseks, suhtlemiseks ja probleemide lahendamiseks [1]. Näiteks II kooliastme puhul tähendab digipädevus oskust kasutada arvutit ja interneti suhtlusvahendina ja vormistada tekste. III kooliastmes eeldatakse, et õpilased suudavad kasutada tehnoloogiat eesmärgipäraselt, teadvustades võimalikke riske ja neid vältides. See pädevus aitab õpilastel arendada oskusi, mis on vajalikud mitte tehnoloogia kasutamiseks ja oma teadmiste rakendamiseks teistes valdkondades.

Kuigi digipädevus on oluline oskus, on informaatika saadaval vaid valikainena [1]. Kooli õppekavas olevad informaatikaalased ained sõltuvad kooli enda otsustest ning võivad varieeruda. Põhikooli riikliku õppekava järgi on valikõppeaine „Informaatika“ eesmärgiks arendada õpilastes oskusi, et nad saaksid kasutada info- ja kommunikatsioonitehnoloogiat ning rakendada neid erinevates valdkondades [3]. Õppekava toob ka välja, et informaatikaõppe sisu koosneb kahest osast: raalmõtlemine, mis keskendub loogilisele probleemilahendusele ja algoritmilisele mõtlemisele, ning disainmõtlemine, mis rõhutab loovust, koostööd ja kasutajakeskset lähenemist lahenduste leidmisel.

Esimeses kooliastmes toimub õpe sageli lõimituna teiste õppeainetega ning kool saab valida nelja õppeteema vahel. „Digiseade töövahendina“ keskendub digiseadme kasutamise

¹ <https://thonny.org> (19.04.2025)

² <https://lahendus.ut.ee> (19.04.2025)

baasoskustele, „Digitaalne ohutus“ tutvustab turvalise ja vastutustundliku digikäitumise aluseid, „Kood“ on sissejuhatus programmeerimisse ja robotikasse ning „Digikunst“ arendab loovust digitaalse meedia kaudu [3].

Teises kooliastmes toimub informaatikaõpe eelistatult eraldi aienena ning seda viib läbi vastava haridusega õpetaja [3]. Õppekavas on neli õppeteemat: „Digiseade töövahendina“, „Programmeerimine“, „Digimeedia“ ja „Digihügieen“. Need teemad aitavad õpilastel arendada digioskusi alates tekstitöötlustest ja meediatöödest kuni programmeerimiseni ning rõhutavad ka küberturvalisuse tähtsust.

Kolmandas kooliastmes õpitakse informaatikat valikainena, milleks on „Infoühiskonna tehnoloogiad“ ja/või digiloovtöö kursus [3]. „Infoühiskonna tehnoloogiad“ kursus keskendub digitaalsete teenuste ja tehnoloogiate praktilisele rakendamisele igapäevaelus. Digiloovtöö kursus keskendub loovusele, praktilistele oskustele ja meeskonnatööle andes õpilastele võimaluse töötada praktilistes projektides.

Informaatika õppeülesannete koostamisel on oluline, et need oleksid seotud igapäevaelu situatsioonidega ja lõimitud teiste õppeainetega, pakkudes õpilastele praktilisi ja mõtestatud õpikogemusi. Riikliku õppekava kohaselt on info- ja kommunikatsioonitehnoloogia integreerimine õpikeskkonda vajalik, kuna see on loomulik osa tänapäeva õppetööst [3]. Selline lõiming aitab ühelt poolt süvendada õpilaste arusaama teiste ainete teemadest, sidudes need informaatikaga, ning teisalt võimaldab tuua tehnoloogia läbiva teemana kõikidesse õppeainetesse.

Programmeerimine ja digioskused muutuvad üha väärtuslikumaks, seega on oluline tagada, et kõigil õpilastel oleks võimalus neid oskusi arendada. Kuna informaatika on praegu õppekavas valikaine ja selle maht võib kooliti suuresti erineda, võib see tekitada olukorra, kus osa õpilasi saab tugeva tehnoloogilise aluspõhja, samas kui teised jäävad sellest ilma. Seetõttu on oluline pakkuda ka eraldi kursuseid või võimalusi iseseisvaks õppeks, et kõik huvilised saaksid oma teadmisi ja oskusi arendada. Just siin mängivad olulist rolli iseseisvad õppematerjalid ja juhendatud kursused, mis annavad õpilastele paindlikkuse omandada programmeerimisoskusi sõltumata kooli pakutavast õppekavast.

2.2 Programmeerimiskeel Python

Python on avatud lähtekoodiga programmeerimiskeel, mis on saanud väga populaarseks eriti algajate seas just tänu oma lihtsusele ja loogilisele keelele. Vähesel määral on võimalik luua praktilisi rakendusi, mis muudab programmeerimise lihtsamaks ja huvitavamaks [4].

Pythoni populaarsust näitab ka TIOBE indeks, mis on üks tuntumaid programmeerimiskeelte populaarsuse hindamise meetodeid. See indeks mõõdab programmeerimiskeelte populaarsust, analüüsides 25 erineva otsingumootori andmeid, et teada saada, kui sageli otsitakse konkreetse keele kohta fraasi „<keel> programming“ [5]. TIOBE indeksi järgi on Python aastal 2024 esikohal, olles saavutanud aasta jooksul 9,3% kasvu [6]. See teeb Pythoni 2024. aasta programmeerimiskeeleks, kuna selle populaarsus on kasvanud kõige enam, ületades teisi keeli.

2.3 Programmeerimiskeskond Thonny

Thonny on Tartu Ülikooli arvutiteaduse instituudis väljatöötatud programmeerimiskeskond, mis on mõeldud eelkõige algajatele Pythoni keele õppimiseks ja õpetamiseks. Selle arendas Aivar Annamaa, kelle eesmärk oli luua kasutajasõbralik ja lihtne keskkond algajatele, mis toetaks programmeerimise õppimist [7]. Thonny kasutajasõbralik disain aitab algajatel kiiresti kohaneda ja keskenduda õppimisele. Lisaks on Thonny veateadete süsteem loodud algajate vajadusi silmas pidades – veateated on selged ja annavad konkreetseid juhiseid, kuidas vigu parandada. Oluline Thonny omadus on võimalus samm-sammult siluda ehk *debug*'ida programmi. See võimaldab astmeliselt jälgida, kuidas väärtused muutuvad ja millised tegevused iga käsu sooritamisel toimuvad, muutes vigade leidmise ja parandamise lihtsamaks.

2.4 Automaatkontrollikeskkond Lahendus

Programmeerimise õppimisel on kandval kohal praktiliste ülesannete lahendamine. Ülesannete lahendamise tõhusamaks muutmiseks kasutatakse sageli automaatkontrolliga keskkondi, mis annavad kiiret automaatset tagasisidet ülesannetele. Üks selline automaatkontrollide keskkond on Lahendus³ – Tartu Ülikooli arvutiteaduse instituudi loodud platvorm, mis on mõeldud programmeerimise kursuste läbiviimiseks [8].

Õpetaja saab luua ülesandeid, lisades ülesande kirjelduse, lävendi ja määrata automaatkontroll testid, mille alusel õpilase lahendusi hinnatakse. Kui õpilased esitavad oma lahendused, siis Lahenduse automaatkontroll hindab need ja annab tagasisidet. See tähendab, et õpetaja ei pea

³ <https://lahendus.ut.ee> (19.04.2025)





igat õpilase tööd käsitsi hindama, kuna süsteem kontrollib lahenduste õigsust automaatselt vastavalt loodud automaatkontrollidele. Kui õpilase lahendus ei vasta ootustele, kuvatakse veateated, mis selgitavad, milles viga seisneb. Samuti on õpetajal võimalus vajadusel ülesanne ümber hinnata või hinnata käsitsi, kui automaatkontroll ei pruugi kõiki nüansse arvesse võtta.

Automaatkontrolli peamised eelised on kohene tagasiside programmile, mis aitab õpilastel kiiresti koodist oma vigu avastada ja parandada, ning õpetaja ajakulu vähenemine, kuna lahenduste käsitsi hindamist on vähem. Lisaks tagab see ühtlase ja objektiivse hindamise, kuna kõik lahendused kontrollitakse samade kriteeriumite alusel. Siiski on automaatkontrollil ka piiranguid. Kõiki ülesandeid ei ole võimalik sel viisil hinnata – näiteks visuaalse väljundiga ülesanded, nagu kilpkonnagraafika, vajavad käsitsi hindamist. Samuti keskendub automaatkontroll ainult lahenduse lõpptulemusele, mitte koodi kvaliteedile, loetavusele või tõhususele. See tähendab, et ka halvasti struktureeritud kood võib testi läbida, kui see annab õige väljundi.

Iga ülesande juures on võimalik näha, millise tulemuse on õpilased saavutanud protsentuaalselt ja värvi skeemi järgi. Ülesannete soorituse taset kuvatakse nii õpilasele kui õpetajale järgmise skeemi alusel (vaata Joonis 1):

- Hall – õpilased, kes pole lahendust esitanud.
- Kollane – õpilased, kes on lahenduse esitanud, kuid pole saavutanud lävendit ületavaid punkte.
- Roheline – õpilased, kelle viimane esitus on saanud lävendit ületavad punktid.
- Sinine – õpilased, kelle lahendusi ei ole veel hinnatud (näiteks käsitsi hinnatavate ülesannete puhul).

Minu kursused > Programmeerimine (näidis) > 1.1 Tervitus

Ülesanne	Katsetamine	Esitused
Mari Mallikas		-/100
Kaarel Karu	8. november 2019, 18:12	100/100 
Pille Piirisild		-/100
Oskar Ohakas	26. september 2019, 12:57	100/100 
Kalle Kaalikas	14. detsember 2019, 22:04	0/100 
Kusti Kuristik	26. september 2019, 12:44	0/100 

Joonis 1. Ülesannete esitluste hindamise vaade Lahenduse keskkonnas [9].

Lahenduse kasutusjuhendis on õpetajale esitatud kõik vajalikud juhised ja samm-sammult selgitused süsteemi kasutamiseks [9]. Lisaks on süsteemis olemas mehhanism, mis võimaldab tuvastada lahenduste vahel sarnasusi, aidates avastada võimalikku plagiaati või leida tüüpilisi lahendusmustreid. On olemas ka hinnetetabel, mis koondab kõikide õpilaste tulemused ühte tabelisse ja annab kiire ülevaate nende edasijõudmisest kursusel. Tabelis on kasutusel sama värviskeem, mis näitab ka ülesannete soorituse taset visuaalselt, aidates õpetajal kiiresti saada ülevaade õpilaste edenemisest.

Thonny keskkonnas on võimalik lisada ka Lahenduse plugin, mis võimaldab esitada Thonnys kirjutatud ülesannete lahendusi otse Lahendusse, ilma keskkondade vahel liikumata [10]. See muudab õppeprotsessi sujuvamaks ning lihtsustab nii õpilastele kui ka õpetajatele ülesannete haldamist.

2.5 Varasemad eestikeelsed Pythoni õppematerjalid

Algajatele programmeerimise huvilistele on õppimiseks loodud mitmeid eestikeelseid õppematerjale, mis pakuvad erinevaid lähenemisviise ja esitlusvorme Pythoni omandamiseks. Materjalid on suunatud erinevatele sihtrühmadele ja pakuvad erinevaid õpivõimalusi. Selle töö raames analüüsiti viit eestikeelset Pythoni õppematerjali algajatele. Lisas V on ülevaatlik tabel, mis toob välja materjalide erinevad aspektid, sealhulgas veebilehe, hinnangulise sihtrühma, materjali vormi, ülesehituse, kirjelduse ja käsitletavat teemat. Järgnevalt on lühiülevaade nendest materjalidest, et paremini mõista, kuidas need õpilastele programmeerimise õpetamisel sobivad.

1. **Digiõpik „Progema“.**⁴ Digiõpik on suunatud põhikooli õpilastele. Materjal on jagatud neljaks põhiteemaks: põhiteadmised, algoritmid, objektorienteeritud programmeerimine ja projektid, mis omakorda jagunevad mitmeks alateemaks. On üles ehitatud selliselt, et katab palju teemasid kiiresti, seetõttu võib see sobida neile, kes soovivad kiiresti arendada või täiendada oma programmeerimisoskusi. Materjal on sobilik kasutamiseks nii huviringis kui ka iseseisvaks õppimiseks, kuna õppija saab vajadusel liikuda omas tempos ja valida teemad vastavalt oma vajadustele.
2. **Digiõpik „Python koolis“.**⁵ Mõeldud põhikooli õpilastele, materjali saab läbida kas iseseisvalt või huviringis. Materjal on üles ehitatud teemade kaupa, katab Pythoni põhitõed ning ulatub edasi keerukamate teemadeni, näiteks Pygame mängude loomine. Õppematerjalis on vähe samm-sammulisi juhendeid, sest rõhk on probleemilahendusel ja iseseisval töö oskuste arendamisel.
3. **Kursus „Teeme ise arvutimänge“.**⁶ Gümnaasiumiõpilastele suunatud kiire tempoga programmeerimise kursus, mis keskendub mängude loomisele Pythoni abil. Kursus on jaotatud kuueks raamatuks, millest igaüks tuleb läbida ühe nädalaga, jättes vähe aega süvenemiseks. Koodinäited on esitatud piltidena, mistõttu peavad õpilased need käsitsi ümber kirjutama, mis võib olla ajamahukas. Kursus sobib hästi motiveeritud õppijatele või huviringis kasutamiseks, kuid võib olla keeruline neile, kes vajavad rohkem aega harjutamiseks.

⁴ <https://www.progema.ee/> (09.02.2025)

⁵ <https://courses.cs.ut.ee/t/pythonkoolis/Main/HomePage> (09.02.2025)

⁶ <https://courses.cs.ut.ee/t/tiam/Main/HomePage> (09.02.2025)

4. **Kursus „Programmeerimisest maalähedaselt“.**⁷ Suunatud pigem täiskasvanutele, aga kasutatav ka põhikooliõpilastega. Materjali saab läbida iseseisvalt või veebi-kursusena. Materjal on hästi struktureeritud ning toetab erinevaid õppimisstiile (tekst, videod, harjutused). Lisaks sisaldab kursus silmaringi laiendamise ja enesekontrolli võimalusi, mis aitavad teadmisi kinnistada ja rakendada. Kursus sobib kõigile, kes soovivad oma programmeerimisoskusi süvendada.
5. **Videokursus „Programmeerimine Pythonis“.**⁸ Suunatud põhikooli III kooliastmele ja gümnaasiumile ning põhineb täielikult videotundidel. Seda saab kasutada nii huviringis kui ka iseseisvalt õppimiseks. Ülesanded ja näidiskoodid esitatakse ainult videotes, mistõttu võib koodi ümbertrükkimine olla keerukas ja ajamahukas. Kursus sobib neile, kes eelistavad visuaalset õpet ja pigem kiiremat tempot.

Kokkuvõttes võib öelda, et Eestis on olemas mitmeid põhikoolis algajatega kasutatavad Pythoni õppematerjale, mis katavad erinevaid õppimisstiile ja tasemeid, kuid igapähe neist on omad puudused. Selle töö eesmärk on luua õppematerjal, mida reaalselt kasutatakse tund-tunnilt 6.-7. klassi õpetamisel põhikoolis, kus õpilased saavad õppida omas tempos, saavad kiiresti tagasisidet ja lahendavad ülesandeid, mis aitavad neil tundide kaupa järk-järgult oma Pythoni oskusi arendada. Selline materjal täiendaks olemasolevaid materjalide valikut ja toetaks õpilaste programmeerimisoskuse arengut.

2.6 ADDIE mudel

Põhikoolile õppematerjali ja kursuse süstemaatiliseks ettevalmistuseks kasutatakse ADDIE (ingl *analyse, design, development, implementation, evaluation*) mudelit [11]. See viiest etapist koosnev lähenemine (analüüs, kavandamine, väljatöötamine, kasutamine ja hinnang) pakub süsteemset raamistikku, mis sobib hästi programmeerimise kursuse loomiseks. Mudeli tugevuseks on selle paindlikkus, mis võimaldab vajadusel etappide vahel edasi-tagasi liikuda ning õppematerjale jooksvalt täiendada [12].

Õppematerjali väljatöötamise etapid ADDIE mudeli järgi on [11] (vaata Joonis 2):

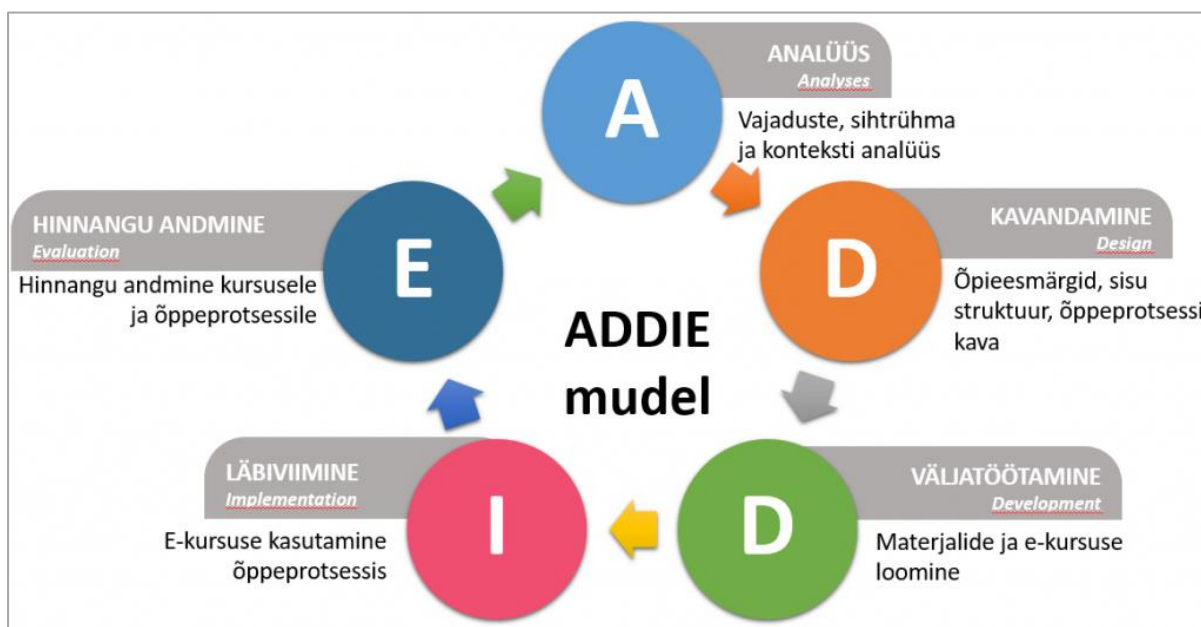
1. **Analüüsi etapp.** Hinnatakse õppijate varasemaid teadmisi ja oskusi, sõnastatakse eesmärgid ning tehakse kindlaks, milliseid ressursse on vaja. Näiteks kui tehakse programmeerimise kursust, siis analüüsitakse, kas õppijatel on eelnev kogemus

⁷ <https://courses.cs.ut.ee/2024/progmaa/fall> (09.02.2025)

⁸ <https://e-koolikott.ee/et/oppematerjal/32909-Videokursus-Programmeerimine-Pythonis> (09.02.2025)

programmeerimisega või nad alustavad täiesti nullist. See tagab, et kursuse sisu oleks kohandatud õppijate vajadustele.

2. **Kavandamise etapp.** Töötatakse välja plaan, mis määratleb õppematerjalide struktuuri ja sisu. Näiteks võib programmeerimise teemad jagada väiksemateks alateemadeks ning lisada praktilisi ülesandeid.
3. **Väljatöötamise etapp.** Luuakse õppematerjalid ja ülesanded, mis on kujundatud nii, et need vastaksid õppijate ootustele ja motiveeriksid praktiliselt kaasa töötama. Näiteks on materjalid koostatud nii, et viivad õppijad samm-sammult ülesannete lahendamiseni.
4. **Kasutamise etapp.** Materjalid viiakse ellu realses õppekeskkonnas, kus hinnatakse nende kasutatavust ja tõhusust. Näiteks viiakse kursus läbi klassiruumis, kus juhendaja jälgib, kas materjalid on piisavalt kaasavad ja kas õpilased suudavad teadmisi rakendada praktilistes ülesannetes.
5. **Hindamise etapp.** Lõpuks, hindamisetapis analüüsitakse õppijate saavutusi, materjalide kvaliteeti. Näiteks tunni või kursuse lõpus hinnatakse tagasiside põhjal, kas õppijad on omandanud vajalikud teadmised ja kas õppematerjalid olid piisavalt arusaadavad ja tõhusad.



Joonis 2. ADDIE mudel [13].

ADDIE mudel valiti, sest see on üks tuntumaid kursuse arendamise mudeleid ning Hidayanto jt uurimus näitas, et mudeli rakendamine võib märkimisväärselt parandada õppijate arusaamist ja motiveerida neid praktilise õppe kaudu [14]. Samas uurimuses leiti, et materjalid mis on loodud vastavalt ADDIE mudeli etappidele on aidanud õpilastel oma teadmisi täiendada ja paremini mõista programmeerimise põhitõdesid.

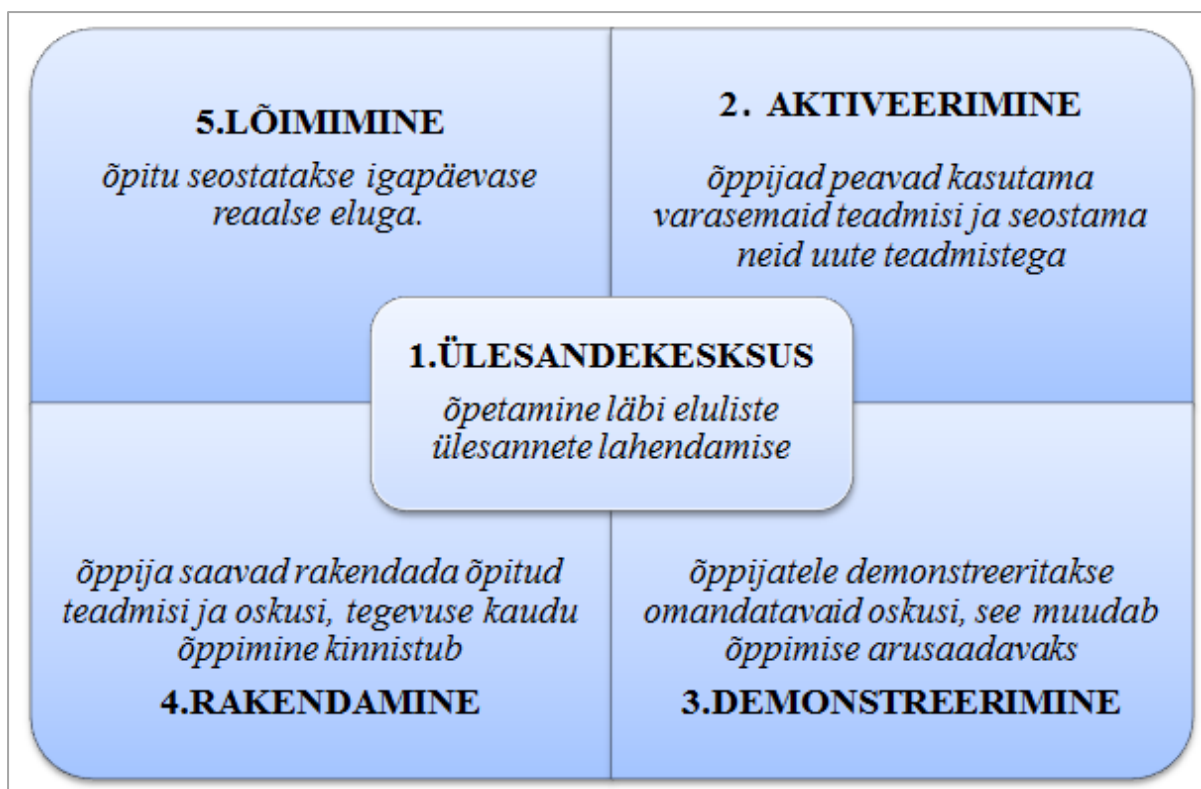
2.7 Merrilli mudel

Õppematerjalide koostamisel on oluline arvestada tõhusate õpikogemuste loomise põhimõtteid. David Merrill toob välja, et kuigi õpetusteooriad ja -mudelid kasutavad erinevat sõnastust ja rõhuasetust, on nende aluseks sageli sarnased põhimõtted [15]. See näitab, et õppimise põhialused on üldiselt sarnased, olenemata teooria taustast või lähenemisest.

Materjalid peaksid olema kujundatud nii, et need keskenduvad õpilaste aktiivsele osalusele ja probleemide lahendamisele, et suurendada õppijate motivatsiooni ja rahulolu [16]. Sellise lähenemise rakendamiseks on loodud Merrilli mudel, mis põhineb viiel põhikomponendil, mis toetavad tõhusat õppimist ja õpikogemust.

Järgnev jaotus põhineb Merrilli esitatud soovitusel [15] (vaata Joonis 3):

1. **Ülesande- või probleemikesksus.** Õppimine algab ülesannetest või probleemidest, mis seovad õpitava reaalse olukordadega. See aitab õppijatel mõista, kuidas teadmisi ja oskusi rakendada praktilises kontekstis. Näiteks õpilased arutavad, kuidas õpitavat teemat saab rakendada igapäevaelu olukordades, näiteks koduste tööde või projektide lahendamiseks.
2. **Aktiveerimine.** Õppimise käigus kasutatakse varasemaid teadmisi ja kogemusi, mis aitavad luua tugeva aluse uute teadmiste omandamiseks ja nende seostamiseks juba olemasolevaga. Näiteks õpilased meenutavad õpitut eelmistest tundidest ja arutavad, kuidas need võivad aidata neil mõista uut teemat.
3. **Demonstreerimine.** Õppematerjalid ja õpetamisprotsess sisaldavad konkreetseid näiteid ja juhendamist, mis aitavad õppijatel mõista uusi kontseptsioone. Näiteks materjalis olev koodinäide või õpetaja näitab ülesande lahendamist ekraanil.
4. **Rakendamine.** Õppijad saavad võimaluse õpitut praktiseerida ja rakendada erinevates ülesannetes, mis kinnistavad teadmisi ja oskusi. Näiteks saavad õpilased ise koostada või lahendada ülesande, mis seotud mingi elulise probleemiga või koduse tööga.
5. **Lõimimine.** Õppijad lõimivad õpitu oma igapäevaellu, jagavad oma teadmisi ja kogemusi ning reflekteerivad õpiprotsessi üle. Näiteks õpilased arutavad, kuidas õppimine on aidanud neil igapäevaelu probleemidega toime tulla.



Joonis 3. Merrilli mudel [16].

Badali jt uurimus on näidanud, et Merrilli printsiipide kasutamine õppematerjalide koostamisel, näiteks MOOC-ide (ingl *massive open online courses*) kujundamisel, võib tõsta õppijate õpitud taset ja rahulolu [17]. Samas uurimuses rõhutatakse ka vajadust õppijate eelnevaid teadmised arvesse võtta ja nende sidumist uute teadmistega, kuna see loob tugeva aluse uute teadmiste omandamiseks. Samuti peab uusi teadmisi edastama praktiliselt ja siduma õppijate isiklike kogemuste või igapäevaeluga.

Kui kasutatakse Merrilli mudelit, siis rakendatakse ülesande- või probleemikeskset lähenemist, mis soodustab probleemilahendusoskuste arendamist [18]. Selline õppeviis pakub õppijatele võimaluse lahendada reaalses kontekstis olevaid ülesandeid, mille kaudu luuakse seos teooria ja praktilise rakenduse vahel. See omakorda toetab iseseisvat ja kriitilist mõtlemist.

3. Metoodika

Selles peatükis kirjeldatakse kasutatud meetodeid ja vahendeid, mille abil viidi läbi bakalaureusetöö raames loodud Pythoni algkursus põhikooliõpilastele. Kirjeldatakse kursuse loomise ja läbiviimise protsessi, valimi moodustamist, tagasiside kogumist ja analüüsimist.

3.1 Valim

Bakalaureusetöö raames koostatud kursusel kasutati mugavusvalimit, kuna õpilased registreerusid ise õppeaasta alguses Pythoni programmeerimise algkursusele, mis oli avatud kõigile põhikooli õpilastele.

Kokku registreerus kursusele üheksa õpilast, kuid esimeses tunnis osales neist kaheksa. Umbes kolme nädala jooksul selgus, et ühel registreerunud õpilasel ei olnud võimalik kursusega jätkata, mistõttu kujunes lõplikuks valimiks seitse õpilast. Õpilased jaotusid kahte gruppi – kolm ja neli õpilast. Kõik osalejad olid 6.-7. klassi õpilased.

Uuringu läbiviimiseks täitsid kõik õpilaste vanemad eelnevalt nõusoleku vormi, milles andsid loa oma laste osalemiseks kursusel ning ka lõputöös (vaata Lisa I).

3.2 Kursuse ja õppematerjalide koostamise protsess ja vahendid

Kursus algas 2024. aasta oktoobri teisel nädalal. Kursuse loomisel kasutati ADDIE mudelit, et tagada süsteemne lähenemine õppetöö korraldamisele. Analüüsi etapis, koostati algeline kursuse ainekava, kus määratleti põhiteemad ja eesmärgid, mida kursuse jooksul käsitleda. Õpilaste teadmisi hinnati esimesel tunnil avaküsimustikuga (vaata Lisa II), et mõista nende teadmisi.

Kursuse jooksul iganädalaselt toimus ADDIE mudeli järgi kolm etappi: kavandamine, väljatöötamine ja kasutamine, et tagada kursuse pidev kohandamine vastavalt õpilaste vajadustele. Iga etapi raames arvestati ka Merrilli mudeli põhimõtteid, et materjalid oleksid struktureeritud ja praktilised, pakkudes õpilastele võimalust omandatud teadmisi mõtestada ning neid ülesannetes tõhusalt rakendada.

Kavandamise etapis määrati iga nädala eesmärgid ja tegevused. Tähtis oli, et iga teema oleks seotud reaalse maailma probleemidega, mis suurendab õppijate kaasatust ja motiveerib neid aktiivselt õppima. Suuremad tunni teemasid jagati alateemadeks, et õppijad saaksid iga tunni jooksul omandada uusi teadmisi samm-sammult. Merrilli mudeli järgi alustati tunde

kordamisest ja lihtsamatest ülesannetest, et õpilased saaksid kinnistada oma teadmisi. Seejärel liiguti keerukamate ülesannete juurde, mis vajasisid juba varem õpitud teadmiste rakendamist.

Väljatöötamise etapis loodi õppematerjalid ja ülesanded, mis olid välja töötatud vastavalt ADDIE mudeli eesmärkidele ja Merrilli mudeli põhimõtetele. Iga õppetund sisaldas praktilisi ülesandeid ja harjutusi, kus õpilased said rakendada oma teadmisi ja oskusi. Materjalide loomisel arvestati ka Merrilli põhimõtetega, et iga õppetund toetaks õpilaste aktiivset osalust. Slaidid loodi keskkonnas Google Slides ja pandi ülesse TÕ arvutiteaduse instituudi PmWiki põhisesse Courses⁹ keskkonda, kus need olid avalikult kättesaadavad õpilastele kogu kursuse vältel. Automaatkontrollid koostati Lahenduse keskkonnas, kus need olid saadaval alates vastava tunni algusest.

Kasutamise etapis viidi loodud õppematerjalid ja ülesanded ellu reaalses klassitunnis. Õpilased lahendasid ülesandeid, saades kohe tagasisidet, mis aitas neil mõista, kas nad on õigesti arusaama omandanud. Kasutamise etapis oli oluline Merrilli mudeli rakendamine, et julgustada õpilasi oma õpikogemust igapäevaelus rakendada ja integreerima. Õpetaja jälgis õpilaste edusamme, pakkus individuaalset tuge, vajadusel täpsustas ülesandeid ja suunas õpilaste tähelepanu kõige olulisematele aspektidele. Lisaks tagas õpetaja, et kõigil tunnis olijatel oleks vähemalt ülesannete minimaalne lävend ületatud, et kindlustada nende edasijõudmine kursusel.

Analüüsi etapis, iga tunni lõpus, täitsid õpilased tagasisideküsimustiku, mille abiga hinnati õppematerjalide kvaliteeti, õpilaste edusamme ja vajadust kursuse sisu täiendamiseks või muutmiseks.

Iga õppetunni kestus oli 60 minutit, mille jooksul keskenduti nii eelmiste tundide kordamisele, uute temade tutvustamisele kui ka praktiliste ülesannete lahendamisele. Tundides kasutati programmeerimiskeskonda Thonny, mis on oma lihtsuse ja kasutajasõbralikkuse tõttu sobiv eelkõige algajatele. Õpilased kasutasid lahenduste kontrollimiseks automaatkontrolli keskkonda Lahendus, mis võimaldas neil saada kohest tagasisidet oma töö kohta. Lisaks sellele olid slaidid, mida kasutati tunnis kättesaadavad veebilehel <https://courses.cs.ut.ee/t/it/Progpython>. Õpilased said materjalidele igal ajal ligi pääseda ja vajadusel neid omas tempos üle vaadata.

⁹ <https://courses.cs.ut.ee> (10.10.2024)

Tundide materjalid, ülesanded ja automaatkontrollid koostati jooksvalt kogu kursuse vältel, lähtudes toimunud praktikumi kogemustest.

3.3 Tagasiside kogumine

Kursuse esimesel tunnil täitsid õpilased anonüümselt avaküsitluse, mille eesmärk oli saada ülevaade nende varasemast kogemusest programmeerimisega ning mõista, millised on nende isiklikud huvid ja motivatsioon kursuse osas (vaata Lisa II).

Küsimustik sisaldas järgmisi küsimusi:

- Millises grupis oled?
- Kas sa oled varem programmeerimisega kokku puutunud?
- Milliseid programmeerimiskeeli oled varem kasutanud?
- Kas sul kodus on võimalik kasutada arvutit programmeerimisülesannete lahendamiseks?
- Kuidas nõustud järgmiste väidetega?
- Millised on su ootused ja soovid käesoleva kursuse suhtes?
- Mis võib olla suurim takistus kursuse läbimisel?

Väidetele said õpilased hinnangu anda Likerti skaalal, kus „1“ tähendas täielikku mittenõustumist ja „5“ täielikku nõustumist. Lisaks oli lisatud „0“, mis tähendas, et õpilane ei osanud vastata. Skaala abil saab mõõta õpilaste motivatsiooni, ootusi ja hinnanguid õppeprotsessi erinevates aspektides, võimaldades neil väljendada oma arvamust mitmel tasandil, mitte ainult variantidega „jah“ või „ei“. See aitab tabada ka väikeseid, kuid olulisi erinevusi vastajate hinnangutes. Joshi jt rõhutavad, et Likerti skaala sobib eriti hästi hoiakute uurimiseks, kuna see võimaldab eristada vastajate hinnangute väikeseid nüansse [19].

Avaküsitluses paluti õpilastel hinnata järgmisi väiteid:

- Mulle meeldib mängida arvutimänge.
- Mulle meeldib matemaatika.
- Mulle meeldib lahendada erinevaid loogilisi ülesandeid.
- Arvan, et programmeerimine on raske.
- Olen huvitatud programmeerimise õppimisest.
- Kasutan igapäevaselt oma telefoni/arvutit/muud nutiseadet.

Kursuse ootused ja soovid ning suurim takistus kursuse läbimisel olid avatud küsimused, kus õpilased said oma arvamusi ja seisukohti soovi korral kirjalikult väljendada. Ülejäänud küsimused olid valikvastustega ja õpilased said soovi korral ise vastuse kirjutada.

Iga tunni lõpus viidi läbi tunni kohta tagasisideküsimustik, et koguda õpilaste arvamusi ja hinnanguid õppeprotsessi kohta (vaata Lisa III). Küsimustiku eesmärk oli mõista tunni raskusastet, hinnata õpilaste arusaamade selgust, õpetaja selgituste tõhusust ning uurida, kui hästi õpilased suudavad õpitut iseseisvalt rakendada.

Küsimustikus olid järgmised küsimused:

- Millises grupis oled?
- Tunni number

Seejärel hindasid õpilased järgmisi väiteid Likerti skaalal, kus „1“ tähendas täielikku mittenõustumist ja „5“ täielikku nõustumist. Lisaks oli skaalal ka „0“, mis tähendas, et õpilane ei osanud vastata. Vastusevariant „0“ jäeti tulemuste keskmise arvutamisel välja. Tagasisideküsimustikus paluti õpilastel hinnata järgmisi väiteid:

- Mulle meeldis tänane tund.
- Ülesanded olid minu jaoks jõukohased.
- Jõudsin kogu tunni materjaliga valmis.
- Arvan, et programmeerimine on raske.
- Õpetaja seletas teemat piisavalt lahti.
- Tunnen, et suudan õpitut iseseisvalt kasutada.
- Sain täna midagi uut ja huvitavat teada.
- Ootan juba järgmist tundi!

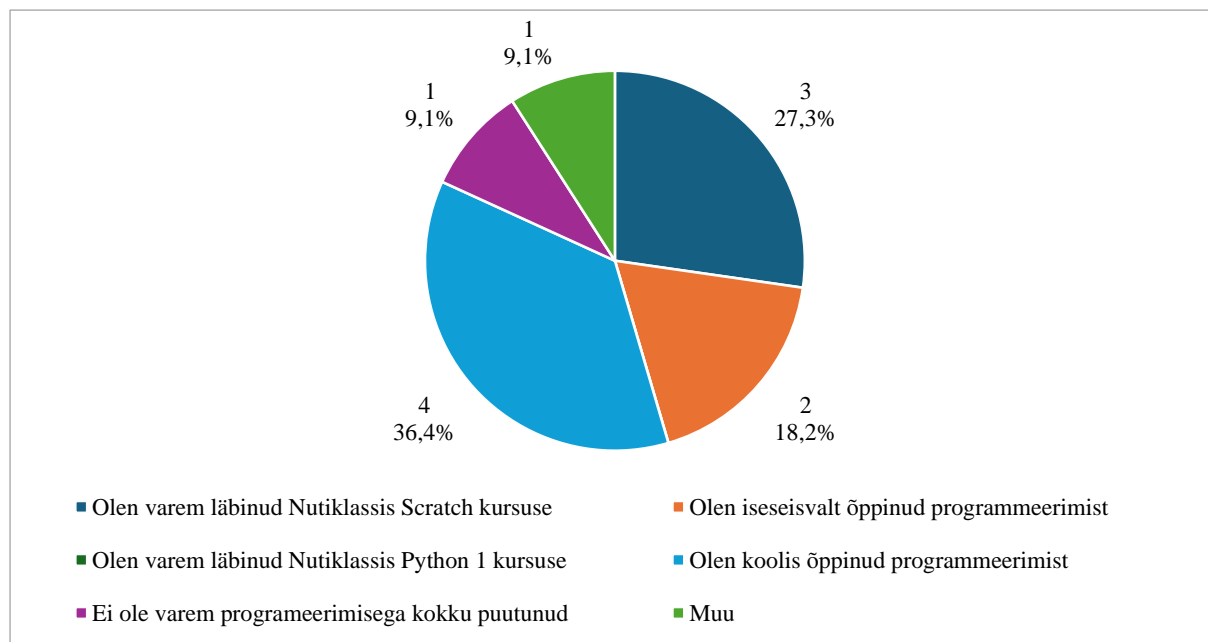
Kogutud tagasiside analüüsimiseks kasutati kirjeldavat statistikat. Andmed, mis saadi Likerti skaalal hinnatud väidete kaudu koondati Exceli tabelisse, kus arvutati keskmised hinnangud ja jälgiti muutusi nädalate lõikes. Nii sai tuvastada keerulisemaid teemasid ning hinnata õpilaste motivatsiooni ja hoiakute muutumist kursuse jooksul. Kuna valim oli väike, siis statistiliselt oluliste seoste leidmine väidete vahel ei olnud võimalik.

4. Tulemused

Selles peatükis kirjeldatakse bakalaureusetöö raames koostatud programmeerimise õppematerjale ja kogutud tagasisidet. Alustuseks antakse ülevaade avaküsimustiku vastustest, mille abil kaardistati õpilaste eelnev kogemus ja ootused. Seejärel kirjeldatakse kursuse kavandamise protsessi ning valminud õppematerjale, kus tuuakse näitena välja ühe tunni ülesehitus Merrilli õpetamismudeli viie etapi alusel. Peatüki lõpetab õpilastelt kogutud tundide tagasiside ja selle analüüs.

4.1 Avaküsimustiku vastused

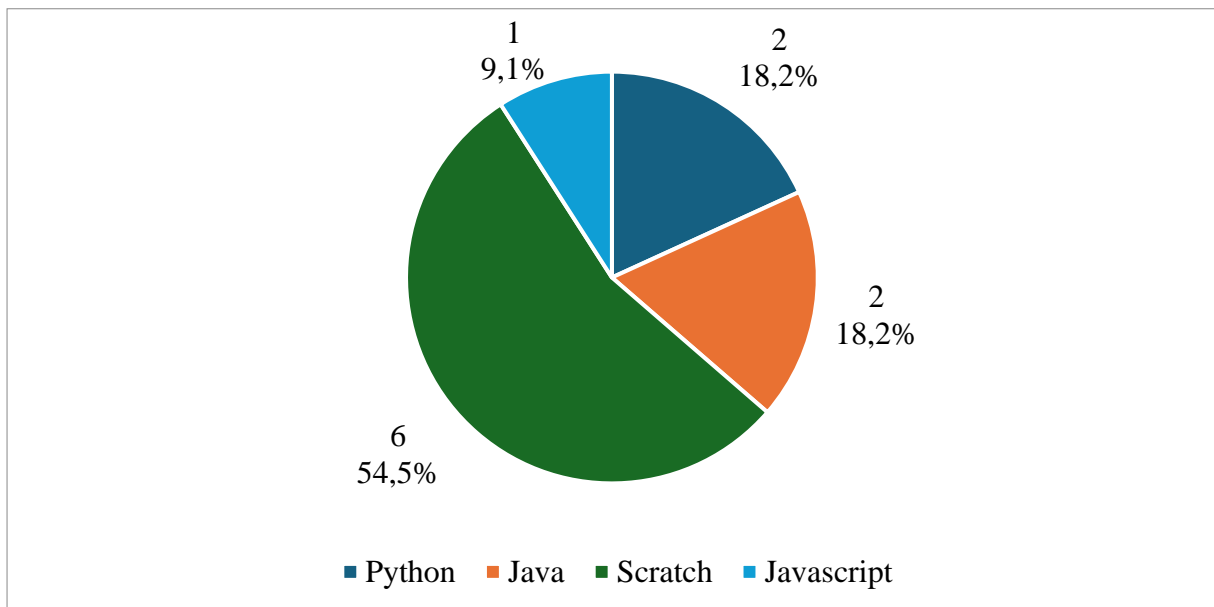
Kursuse esimeses tunnis paluti õpilastel vastata avaküsimustikule, millele vastasid kaheksa õpilast (vaata Lisa II). Avaküsimustik aitab saada ülevaate õpilaste varasemast kokkupuutest programmeerimisega, huvist ja ootustest kursuse suhtes. Küsimustikust selgus, et neli (36,4%) õpilast olid varem koolis õppinud programmeerimist, kolm (27,3%) õpilast olid varem läbinud Scratchi¹⁰ kursuse, kaks (18,2%) olid varem iseseisvalt programmeerimist õppinud, üks (9,1%) õpilane ei olnud varem programmeerimisega kokku puutunud ja üks (9,1%) õpilane oli iseseisvalt Youtube'i abiga kunagi Pythonit proovinud õppida. Joonisel 4 on kujutatud õpilaste protsentuaalne jaotus vastavalt nende varasemale kokkupuutele programmeerimisega.



Joonis 4. Kursusel osalenud õpilaste varasem kokkupuude programmeerimisega (n=8).

¹⁰ <https://nutiklass.ee/info> (01.02.2025)

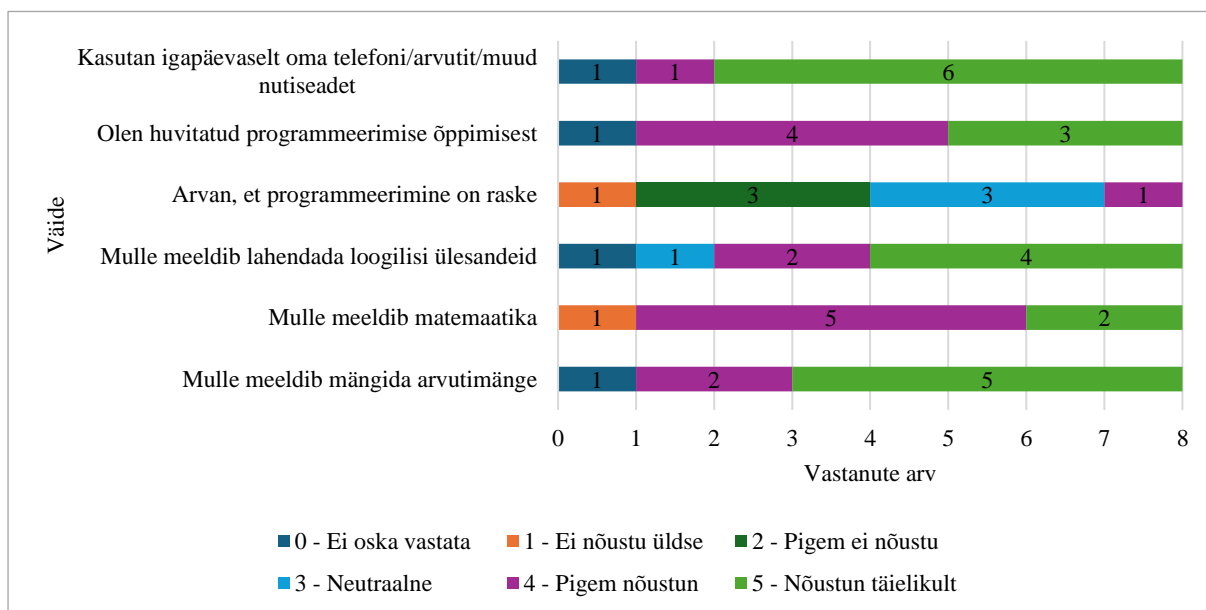
Kõik õpilased olid varem mingil moel programmeerimisega kokku puutunud. Õpilased olid varem proovinud järgmisi programmeerimiskeeli: Scratch, Python, Java ja Javascript. Joonisel 5 on esitatud õpilaste vastuste jaotus vastavalt varasematele kogemustele erinevate programmeerimiskeeltega, andes ülevaate õpilaste eelnevate teadmiste tasemest.



Joonis 5. Küsimuse „Milliseid programmeerimiskeeli oled varem kasutanud?” vastused (n=8).

Avaküsimustikust selgus ka, et kolmel õpilasel ei ole võimalik kodus kasutada arvutit programmeerimise ülesannete lahendamiseks. Selle tõttu võeti vastu otsus mitte anda kursusel kodutöid, et tagada võrdne osalemisvõimalus kõigile õpilastele.

Uuriti ka õpilaste erinevaid hoiakuid. Vastuseid pidid hindama Likerti skaalal. Väidete vastused on nähtavad joonisel 6.



Joonis 6. Küsimuse „Kuidas nõustud järgmiste väidetega?” vastused (n=8).

Kõik, välja arvatud üks õpilane, nõustusid, et neile meeldib mängida arvutimänge, lahendada matemaatilisi ja loogilisi ülesandeid. Programmeerimise keerukuse kohta olid arvamused erinevad. Enamik õpilasi ei pidanud programmeerimist liiga keeruliseks samas kui üks õpilane oli veidi rohkem nõus, et programmeerimine on raske. Enamik õpilastest väljendas samuti suurt huvi programmeerimise õppimise vastu ning kasutas igapäevaselt oma telefone, arvuteid või muid nutiseadmeid.

Avaküsimustikus uuriti õpilaste ootusi kursusele. Kõik kaheksa õpilast väljendasid soovi õppida programmeerimist. Sooviti ise proovida midagi praktilist, näiteks mängude loomist. Õpilased soovisid omandada oskusi, mida saaks ka tulevikus rakendada ja millel oleks reaalne väärtus. Samuti uuriti, et mis võiks olla suurim takistus kursuse läbimisel. Siin mainiti takistusteks matemaatika keerukust, kodutööde kogust ja arusaamise raskusi mõningatest teooriatest. Need vastused on olulised kursuse kujundamiseks. Takistuste mainimine viitab sellele, et ülesannete maht ja raskusaste tuleks hoida tasakaalus – need peaksid pakkuma väljakutset, kuid samas jääma jõukohaseks.

4.2 Kursuse planeerimine

Kursuse planeerimisel loodi kursuse ainekava. Kursuse ainekavas (vaata Lisa IV) on kõikide teemade kohta välja toodud tundide arv, põhimõisted, tunnis toimuv teoreetilised ja praktilised tegevused, soovitatavad õpitulemused ja hindamine, lõiming muude ainetega ja kasutatav materjal. Kursuse eesmärgiks on tutvustada õpilastele Pythoni programmeerimiskeele põhialuseid, et anda neile esimene kogemus programmeerimise valdkonnas.

Enne kursuse algust koostati kursuse ajaline kava, kus määrati kindlaks põhiteemad ja eesmärgid, mida kogu õppeperioodi jooksul käsitleda. Kursuse käigus täiendati ja muudeti kava jooksvalt, lähtudes õpilaste arusaamadest ja vajadustest. ADDIE mudel aitab tagada, et kursuse koostamisel ja läbiviimisel oleks järgitud struktureeritud lähenemisviisi, mis kohandub õpilaste varasemate teadmiste ja vajadustega. Tundide struktuuri ja sisu kohandamine võimaldas paindlikult reageerida õpilaste arengule, et tagada sujuv õppimisprotsess. Kursuse teemade lõplik järjestus ning jaotumine nädalate lõikes on esitatud tabelis 1.

Tabel 1. Lõplik teemade järjestus kursusel ja jaotumine nädalate kaupa.

Nädal	Teema	Sisu
1.	Sissejuhatus, algoritm	Programmeerimise põhialused, programmeerimiskeeled, avaküsimustik. Algoritmi mõiste, algoritmide koostamine ja realiseerimine
2.	Arenduskeskkond, andmed	Arenduskeskkond Thonny, selle seadistamine ja kasutamine. Erinevad andmetüübid Pythonis. Andmete esitamine ja töötlemine programmides.
3.	Muutujad	Muutujale väärtuse omistamine ja kasutamine.
4.–5.	Sisend ja väljund, valiklauseid	Kasutajalt sisendi võtmine ja tulemuste kuvamine ekraanil. Tingimuslauseid (<i>if</i> , <i>elif</i> , <i>else</i>) otsuste tegemiseks programmides.
6.–7.	Tsüklid	Korduvate tegevuste automatiseerimine tsüklite (<i>while</i> ja <i>for</i>) abil.
8.–10.	Andmestruktuurid	Järjendid ja sõnastikud andmete organiseerimiseks ja töötlemiseks.
11.–12.	Funktsioonid	Koodi korduvkasutus ja organiseerimine funktsioonide abil.
13.–14.	Failid ja andmevahetus, veahaldus	Andmete lugemine ja salvestamine failidesse. Vigade käsitlemine (<i>try</i> ja <i>except</i>).
15.–16.	Kilpkonnagraafika	Visuaalsete jooniste ja kujundite loomine kilpkonnagraafika (Turtle) abil.

Kursuse käigus kaeti need teemad ära kuueteistkümnelt 60-minutilise tunni vahel. Kursusel käsitletavate teemade järjestus sarnaneb teiste programmeerimise algkursuste ülesehitusele.

See võimaldab õppijal liikuda sujuvalt lihtsamatelt teemadelt keerukamateni, toetades Pythoni programmeerimise aluste mõistmist ja edasiste oskuste arendamist.

Kursuse läbinud õpilane:

- mõistab programmeerimise põhialuseid ja oskab selgitada programmeerimisega seonduvaid mõisteid,
- oskab kasutada arenduskeskkonda Thonny,
- oskab kirjutada lihtsaid programme,
- suudab luua ja kasutada muutujaid ning erinevaid andmetüüpe,
- oskab lugeda ja salvestada andmeid failidesse,
- suudab luua ja kasutada funktsioone,
- mõistab veahaldust ja oskab kasutada *try-except* struktuuri,
- suudab luua lihtsamaid visuaalseid jooniseid ja kujundeid, kasutades kilpkonna-graafikat.

4.3 Valminud õppematerjal

Kursuse läbiviimiseks loodi 16-tunnine õppematerjal mis on kättesaadav Tartu Ülikooli arvutiteaduse instituudi PmWiki keskkonnas Courses (<https://courses.cs.ut.ee/t/it/Progpython>). Kursuse kavandamisel toetuti ADDIE mudelile ja materjalide koostamisel Merrilli mudelile. ADDIE mudel pakub kursusepõhiselt struktureeritud lähenemist, mille järgi kohandatakse õppematerjalid õppijate varasemate teadmiste ja vajadustega. Selle mudeli kaudu saab tagada, et õppetöö oleks hästi planeeritud ja vastaks õpilaste tasemele vastavalt avaküsitlusele. Merrilli mudel keskendub tunnipõhiselt sellele, et õpilased oleksid aktiivselt seotud õppimisprotsessiga, näiteks lahendades reaalseid probleeme keskenduvaid ülesandeid ja arutades lahendusi kaasõpilastega. See mudel aitab tõsta õpilaste motivatsiooni ja õpikogemust, muutes õppe kaasahaaravamaks.

Kursus on mõeldud toimumiseks klassiruumis, kuna õppetöö keskendub praktilisele tööle, kus slaidide ja õpetaja selgituste kõrval saavad õpilased arvuti taga iseseisvalt harjutada ja ülesandeid lahendada. Kodutöid selle kursuse raames ette nähtud ei ole. Kursusel kasutatakse programmeerimiskeelt Python ja programmeerimiskeskkonda Thonny. Praktiliste ülesannete kontrollimiseks loodi Lahenduse¹¹ keskkonda kokku 37 automaatkontrolli, millest 28 olid töö

¹¹ <https://lahendus.ut.ee/> (19.04.2025)

autori loodud ja 9 olid laenatud gümnaasiumi programmeerimise kursusel ja vastavalt vajadustele kohandatud.

Kursuse peamine eesmärk oli töötada välja praktiliselt kasutatavad ja arusaadavad tund-tunnilt kasutatavad Pythoni algkursuse õppematerjalid, mis on suunatud põhikooli õpilastele. Õppematerjali eesmärk on anda õpilastele esimene kogemus Pythoni programmeerimis-keelega, tutvustades neile programmeerimise põhialuseid.

4.4 Õppematerjal ühe tunni näitel

Selles peatükis kirjeldatakse näitena viienda tunnimaterjali sisu vastavalt Merrilli mudelile, mis keskendub järk-järgulisele õppimisele: esmalt aktiveeritakse varasemad teadmised, seejärel demonstreeritakse uut materjali, võimaldatakse selle rakendamist ning lõpuks lõimitakse õpitu iseseisvatesse ülesannetesse. Viienda tunni materjalid ja siin peatükis näidatud slaidid on kättesaadavad Courses keskkonnas materjali 5. tunni alt aadressilt <https://courses.cs.ut.ee/t/it/Progpython/Tund5>.

4.4.1 Merrilli mudel: ülesande- või probleemikesksus

Merrilli mudeli põhjal on tunni keskmeks praktiline probleemilahendus, milleks on näiteks viiendas tunnis tingimuslausete rakendamine erinevates olukordades. Tund on üles ehitatud nii, et õpilased ei õpi pelgalt tingimuslausete süntaksit, vaid mõistavad ka nende praktilist rakendust ja olulisust igapäevaeluliste probleemide lahendamisel. Õppetöö käigus lahendavad õpilased mitmesuguseid praktilisi ülesandeid, mis aitavad neil rakendada omandatud teadmisi reaalses kontekstis.

Näiteks:

- „Lennupileti soodustused” – ülesanne, mis arvutab lennupileti hinna sõltuvalt reisija vanusest.
- „Seifi parool” – ülesanne, mis kontrollib, kas kasutaja sisestatud parool on õige.
- „Temperatuuri põhjal riietussoovitus” – ülesanne, mis annab kasutajale riietumisnõuandeid vastavalt õhutemperatuurile.

Kuna ülesanded on seotud reaalse eluga, aitavad need õpilastel paremini mõista tingimuslausete sisu ja tähtsust programmeerimisel.

4.4.2 Merrilli mudel: aktiveerimine

Merrilli mudeli põhjal on oluline uute teadmiste sidumine varasemate kogemustega, et õppijad saaksid paremini mõista uut materjali ja seostada seda juba omandatud teadmistega. Viies tund algab eelmise tunni materjali kordamisega, et tuletada meelde varasemalt õpitud teadmised. Meenutatakse, kuidas küsida sisendit ja milliseid sisendi liike on olemas (vaata Joonis 7). Arutatakse, kuidas erinevaid sisenditüüpe saab kasutada erinevates projektides, näiteks millal on vajalik küsida numbrilisi või tekstilisi andmeid. See aitab õpilastel seostada uut teemat varasemate teadmistega.

Muud tüüpi sisend

```
sõne = input("Sisesta sõne: ")  
täisarv = int(input("Sisesta täisarv: "))  
komaarv = float(input("Sisesta ujukomaarv: "))
```

Kui sisestada midagi, mida nt int ei saa täisarvuks teisendada, siis ilmub veateade

Eeldame, et kasutaja sisestab seda tüüpi asju, mida küsitakse

Joonis 7. Viienda tunni viies slaid õpilaste aktiveerimiseks.

Kui tunni teemaga on juba tutvunud, siis saavad õpilased suulise testi käigus aktiveerida just õpitud teadmisi, analüüsides slaidil esitatud koodijuppi (vaata Joonis 8). Õpilastel on lubatud omavahel arutada, kuna see võimaldab üksteiselt õppida. Vajadusel arutatakse vastused koos õpetajaga läbi, et selgitada välja õiged lahendused. Õpilased saavad ka ise koodijuppi muuta ja testida, et kontrollida vastuseid praktiliselt.

Test

- 1
- a
- 2
- b
- Olen siin
- Veateade

Mis väljastatakse ekraanile?

```
a = 1
b = 2
if a > b:
    if a == 1:
        print(a)
    else:
        print(b)
else:
    print("Olen siin")
```

Joonis 8. Viienda tunni kümnes slaid õpilaste aktiveerimiseks ehk vaheküsimuse näide.

4.4.3 Merrilli mudel: demonstreerimine

Demonstreerimine on Merrilli mudelis oluline etapp, kuna see aitab õppijatel mõista uusi teadmisi läbi praktiliste näidete, enne kui nad asuvad seda ise rakendama. Selles etapis tutvustab õpetaja uut teemat – tingimuslauset, esitades slaidil koodinäite (vaata Joonis 9) ning selgitades selle struktuuri ja toimimist. Õpetaja arutleb, kuidas tingimuslause iga osa, nagu koolon ja taanded, mõjutavad tingimuslause toimimist. Õpilased jälgivad õpetaja näidet ja proovivad sama koodijuppi ise kirjutada ja käivitada.

Tingimuslause näide

Kindlasti ei tohi unustada koolonit (:) if-rea lõpust

Print on taandatud võrreldes eelmise reaga paremale

Selleks tuleb vajutada ühe korra TAB-klahvi või 4 korda tühikut. Sellega määrame, et print käsk täidetakse if-tingimusest sõltuvalt

```
1 sisestatud_pin = input("Sisesta PIN-kood: ")
2
3 if sisestatud_pin == "1234":
4     print("Sisenesid pangaautomaati!")
```

Käsurida ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Sisesta PIN-kood: 1234
Sisenesid pangaautomaati!
```

Joonis 9. Viienda tunni kuues slaid tingimuslause demonstreerimine.

Pärast baasteadmiste omandamist tutvustatakse pesastatud tingimuslauseid. Õpetaja tutvustab, kuidas selline ülesehitus töötab ning miks ja millistes olukordades võib olla vajalik kasutada sellist lähenemist. Koos klassiga arutatakse, millised päriselus esinevad probleemid võiksid seda lähenemist nõuda. Pärast selgitust käiakse koos õpilastega läbi näide, mille õpetaja on ette valmistanud (vaata Joonis 10) ja õpilased saavad praktiliselt näha, kuidas pesastatud tingimuslause struktuur töötab.

Näide

```
1 päev="laupäev"
2 ilm="vihmane"
3
4 if päev == "laupäev":
5     if ilm == "päikesepaisteline":
6         print("Lähme õue!")
7     else:
8         print("Võib-olla jääme tuppa.")
9 else:
10    print("Täna on koolipäev.")
11
```

Joonis 10. Viienda tunni üheksas slaid pesastatud tingimuslause demonstreerimine.

4.4.4 Merrilli mudel: rakendamine

Merrilli mudeli järgi on rakendamine oluline samm õppeprotsessis, kuna see võimaldab õppijatel omandatud teadmisi praktiliselt kasutada ja seeläbi õpitud kinnistada. Selles tunnis saavad õpilased rakendada oma teadmisi lahendades praktilise ülesande „Lennupileti soodustused”, kus nad kirjutavad koodi, mis määrab lennupileti hinna vastavalt erinevatele tingimustele (vaata Joonis 11). Õpilased esitavad oma lahendused automaatkontrolliks ning õpetaja aitab analüüsida võimalikke vigu. Selle ülesande eesmärk on anda õpilastele võimalus kohe rakendada õpitud juhendatud keskkonnas, kus õpetaja saab vajadusel suunata.

Ülesanne: Lennupileti soodustused

Kirjuta programm, mis küsib kasutajalt tema vanust ja kuvab reisimisvõimalusi ning pileti soodustusi vastavalt vanuserühmale.

Lisaks pakub programm 12–64-aastastele kasutajatele võimalust sisestada sooduskood, mis annab täiendavat allahindlust.

Esita lahendus automaatkontrolliks. Ülesanne "Lennupileti soodustused"

Joonis 11. Viienda tunni üheteistkümnes slaid õpitud teadmiste rakendamine.

Samuti käsitletakse loogilisi operaatoreid (*and*, *or*, *not*) ja nende kasutamist tingimuslausetes koostamisel. Pärast teoreetilist selgitust tehakse suuline test (vaata Joonis 12), kus õpilased analüüsivad koodijuppe ja arutavad, millised väärtused erinevate sisendite korral kuvatakse. Vajadusel käivitatakse kood ja kontrollitakse tulemusi.

Mis tuleb ekraanile?

```
1 y=8
2 x=5
3 if x!=10 or y>0:
4     print("a")
5 else:
6     print("b")
```

```
1 y=8
2 x=3
3 if x < 5 or y > 10:
4     print("a")
5 else:
6     print("b")
```

```
1 y=8
2 x=5
3 if x!=10 and y<=0:
4     print("a")
5 else:
6     print("b")
```

```
1 y=3
2 x=5
3 if not x < 10 or y > 5:
4     print("a")
5 else:
6     print("b")
```

Joonis 12. Viienda tunni neljateistkümnes slaid õpitud teadmiste rakendamine.

4.4.5 Merrilli mudel: lõimimine

Merrilli mudeli kohaselt on lõimimine õppeprotsessi oluline etapp, kuna see võimaldab õpilastel siduda uusi teadmisi varasemate kogemustega ning rakendada õpitut praktilistes ülesannetes. Tunni alguses õpivad õpilased tingimuslaused, mida nad hiljem iseseisvalt

ülesannetes, nagu „Seifi parool” ja „Temperatuuri põhjal riietussoovitus” rakendavad. Õpilased esitlevad oma lahendusi automaatkontrolliks ning õpetaja annab vajadusel lisavihjeid. Need ülesanded esitatakse tunni lõpus iseseisvaks lahendamiseks, kuna need nõuavad tunnis õpitud seoste rakendamist ja varasemate teadmiste integreerimist.

Kuna ülesannete kirjeldused ei pruugi slaididele ära mahtuda, on Courses keskkonda integreeritud Lahenduse¹² keskkond (vaata Joonis 13), kus on ülesannete täpsed kirjeldused, töö näited ja raskemate ülesannete puhul ka vihjed. Selles keskkonnas saavad õpilased ka oma lahendusi esitada ja automaatkontrolli kaudu tagasisidet saada.

Lennupileti soodustused

Kirjuta programm, mis küsib kasutajalt tema vanust ja kuvab reisimisvõimalusi ning pileti soodustusi vastavalt vanuserühmale. Lisaks pakub programm 12–64-aastastele kasutajatele võimalust sisestada sooduskood, mis annab täiendavat allahindlust.

- Kui vanus on alla 12, siis kuvatakse teade: Saad lapsepileti ja reisida ainult koos täiskasvanuga!
- Kui vanus on vahemikus 12 kuni 64, siis kuvatakse teade: Saad tavahinnaga pileti, sest kuulud täiskasvanute vanuserühma!
 - Kasutajalt küsitakse, kas tal on sooduskood (jah/ei).
 - Kui kasutaja sisestab kehtiva sooduskoodi "SOODUS20", kuvab programm täiendava allahindluse teate: Sooduskood aktsepteeritud, saad 20% allahindlust!
 - Kui sooduskood on vale või seda ei sisestata, jätkab programm ilma soodustust arvestamata.
- Vanusega 64 ja rohkem kuvab teate: Teile kehtib sooduspilet vanuse alusel!

► Näide programmi tööst

[Lennupileti soodustused · Lahendus](#)

Lahendus | #849

Joonis 13. Ekraanipilt Courses keskkonnast ülesande „Lennupileti soodustused” kirjeldusest.

Tunni lõpus tehakse kokkuvõtte ning arutatakse läbi keerulisemad kohad. Õpilastel on võimalus esitada küsimusi, kui midagi jäi segaseks. Samuti täidavad nad tunni kohta tagasisideküsimustiku (vaata Lisa III), et anda õpetajale tagasisidet tunni sisu, ülesannete keerukuse ja oma arusaamade kohta.

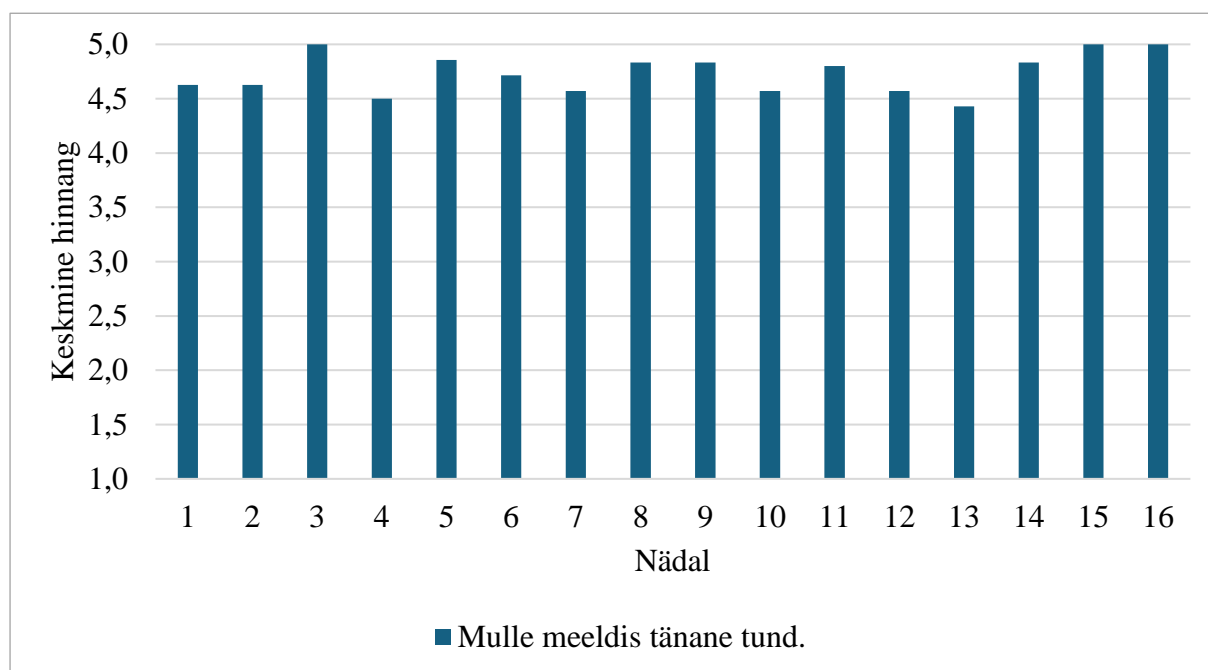
¹² <https://lahendus.ut.ee/> (19.04.2025)

5. Õpilaste tagasiside kursusele

Kursuse vältel koguti igas tunnis kõigilt kohalolijatelt tagasisidet tunni kohta (vaata Lisa III). Õpilased hindasid Likerti skaalal tundi kaheksa väite abil, mis on välja toodud peatükis „Tagasiside kogumine”.

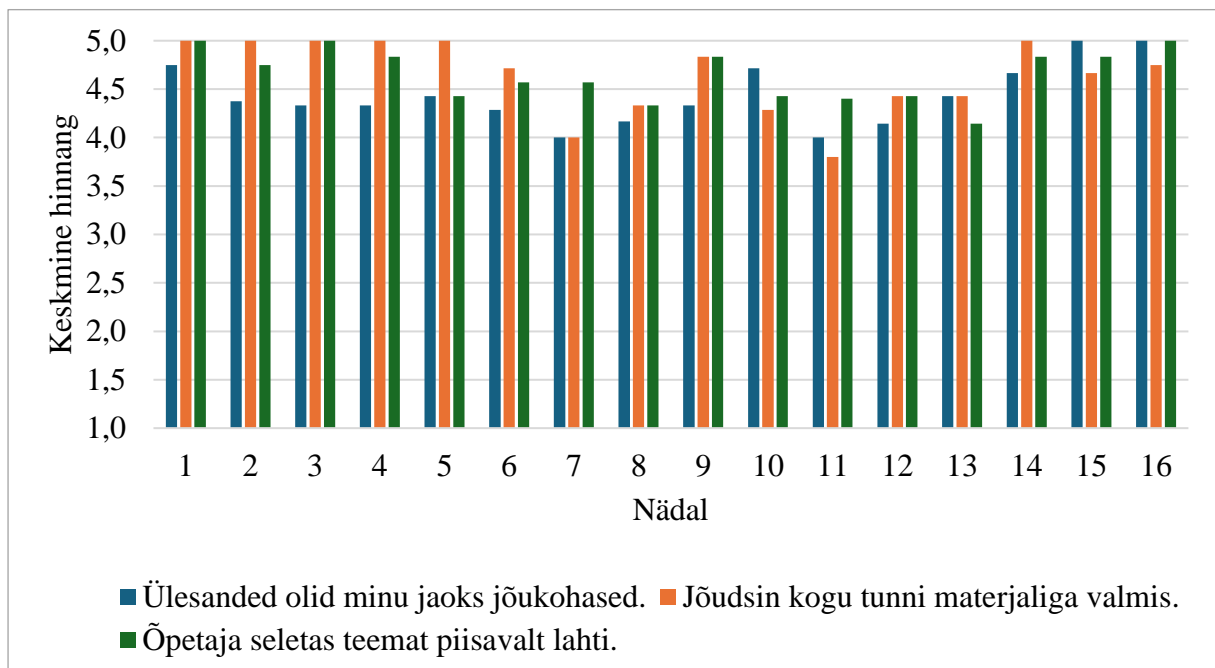
Joonisel 14 on näha, et õpilastele üldiselt tunnid meeldisid. Keskmise hinnang tundide meeldimise kohta viie punkti skaalal oli 4,71, mis viitab positiivsele õpikogemusele. Enim meeldisid õpilastele kolmas, viieteistkümnes ja kuueteistkümnes tund. Kolmanda tunni teemaks oli muutujad, mis on programmeerimise üks põhialuseid ning tõenäoliselt oli see teema hästi mõistetav ja praktiline. Viieteistkümnenenda ja kuueteistkümnenenda tunni teemad olid kilpkonnagraafika. See on rohkem visuaalne ja mängulisem õppimisviis ning seetõttu meeldis ka õpilastele rohkem.

Kõige vähem meeldis õpilastele kolmeteistkümnes tund. Selle tunni teemaks oli andmevahetus I ehk failidest lugemine. Tulemus võib viidata sellele, et teema oli alguses keerulisem või vähem kaasahaarav võrreldes visuaalsemate teemadega nagu kilpkonnagraafika, mis oli nädalatel viisteist ja kuusteist. Failidega töötamine on aga oluline oskus ning kuigi tund ei olnud võib-olla kõige populaarsem, oli see siiski õpetlik ja vajalik. Tund neliteist oli samuti seotud failide töötlemisega (faili kirjutamine ja veahaldus), kuid õpilased leidsid, et see tund meeldis juba rohkem, kuna teema sai selgemaks.



Joonis 14. Keskmise hinnang tundide meeldimise kohta nädalate kaupa.

Joonisel 15 on esitatud õpilaste hinnangud kolmele küsimusele: „Ülesanded olid minu jaoks jõukohased“, „Jõudsin kogu tunni materjaliga valmis“ ja „Õpetaja seletas teemat piisavalt lahti“.



Joonis 15. Keskmine hinnang väidetele: „Ülesanded olid minu jaoks jõukohased.“, „Jõudsin kogu tunni materjaliga valmis.“ ja „Õpetaja seletas teemat piisavalt hästi.“ nädalate kaupa.

Keskmine hinnang ülesannete jõukohasuse kohta oli 4,43, mis viitab sellele, et enamik õpilasi pidas kursuse materjali nende jaoks jõukohaseks. Kõrgeimad hinnangud anti tundidele viiasteist ja kuusteist, kus teemaks oli kilpkonnagraafika. Kõige madalamad hinnangud (4,00) anti aga tundidele seitse ja üksteist, kus teemadeks olid vastavalt tsüklid ja funktsioonid. Need teemad võivad algajatele tunduda keerukamad, mis selgitab madalamat hinnangut.

Keskmine hinnang materjaliga valmis jõudmisel oli 4,64, mis näitab, et enamik õpilasi jõudis tunni jooksul kogu materjaliga valmis. Kõik õpilased jõudsid kogu materjaliga valmis tundides üks, kaks, kolm, neli, viis ja neliteist. Järelikult olid need tunnid hästi struktureeritud ja õpilased suutsid kõik õpitud materjalid omandada. Kõige madalam hinnang (3,80) oli tunnile üksteist, kus teemaks oli funktsioonid I. Tunnis kaksteist, kus teemaks oli funktsioonid II, tõusis hinnang materjaliga valmis jõudmisel. See võib viidata sellele, et õpilased omandasid eelmises tunnis teadmised funktsioonide alustest ja seetõttu tundus teema neile selgem ja nad jõudsid ülesannetega kaugemale.

Õpetaja seletuste keskmine hinnang oli 4,63, mis viitab sellele, et õpilased olid üldiselt rahul õpetaja selgitustega. Kõrgeim hinnang anti tundides üks, kolm ja kuusteist, kus õpilased

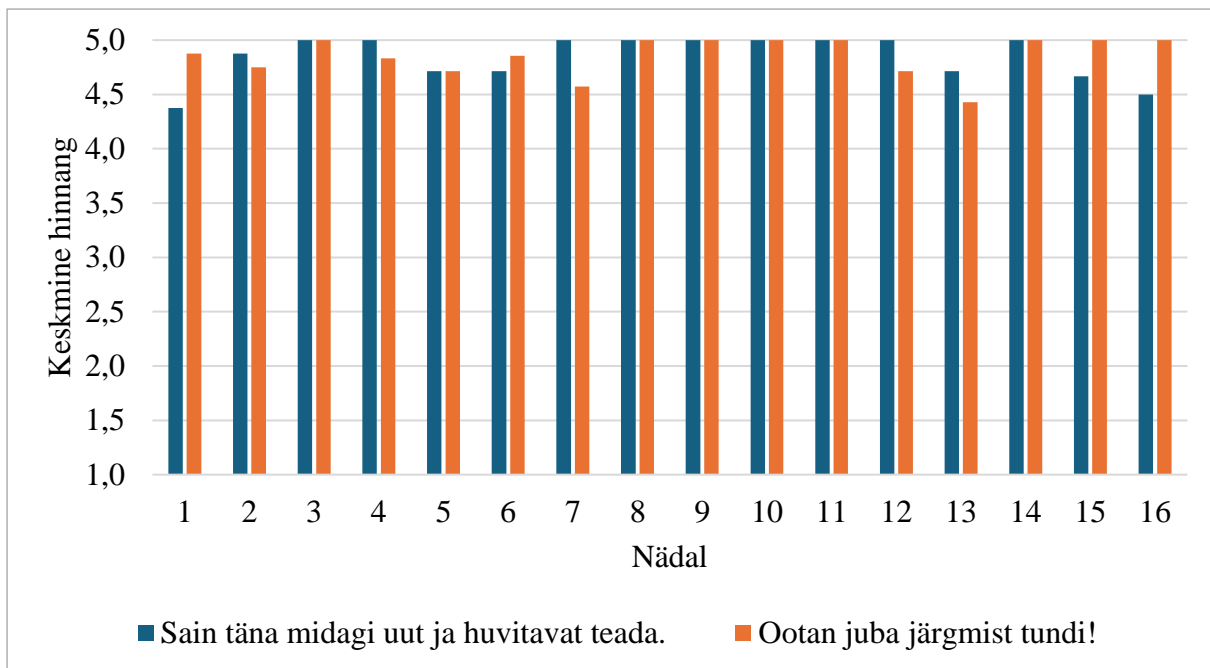
tunds, et õpetaja selgitused olid piisavad ja selged. Madalam hinnang (4,14) anti kolmeteistkümnendas tunnis, kus käsitleti andmevahetuse teemat. Esimene andmevahetuse tund, kus oli teemaks failidest lugemine, võis olla õpilaste jaoks keerulisem, kuna teema oli õpilastele võõras. Neljateistkümnendas tunnis, kus oli teemaks faili kirjutamine ja veahaldus, õpetaja seletuse keskmine hinnang kohe tõusis. See võib viidata sellele, et õpilased suutsid läbi erinevate näidete paremini mõista, kuidas failide töötlemine toimib tervikuna ja õpetaja selgitused olid selgemad, kuna õpilased olid juba eelmisest tunnist rohkem teadlikud failide töötlemisest. Väitele „Õpetaja seletas teemat piisavalt hästi.“ oli kaks vastust, kus valiti skaala väärtuseks „0“ ehk „Ei oska vastata“. Kuna seda ei saa arvestada keskmiste leidmisel, jäeti need tulemuste analüüsist välja.

Kokkuvõttes on tulemused positiivsed, näidates, et õpilased olid rahul ülesannete keerukuse, tunni materjaliga valmis jõudmise ja õpetaja selgituste tasemega. Kõrgemad hinnangud andsid õpilased neile tundidele, kus teema oli praktiline või visuaalne, samas kui madalamad hinnangud viitavad keerukamatele teemadele, nagu andmevahetus, kus õpilased vajasisid rohkem tuge ja täiendavaid selgitusi.

Joonisel 16 on näha hinnanguid küsimustele „Ootan juba järgmist tundi!“ ja „Sain täna midagi uut ja huvitavat teada.“. Mõlema väite keskmine hinnang on 4,84, mis näitab, et õpilased olid üldiselt väga rahul oma õpikogemusega. Väitele „Sain täna midagi uut ja huvitavat teada.“ anti üks vastus valikuga „0“ ehk „Ei oska vastata“, kuid see jäeti analüüsist välja, kuna seda ei saa arvestada keskmise arvutamisel.

Kõrgeim hinnang järgmise tunni ootusele, anti tundides kolm, kaheksa, üheksa, kümme, üksteist, neliteist, viisteist ja kuusteist, kus õpilased tundsid, et õpikogemus oli piisavalt huvitav ja motiveeriv, et oodata järgmist korda. Kõige madalam hinnang (4,43) anti aga kolmeteistkümnendas tunnis, kus teemaks oli andmevahetus, mis võib viidata sellele, et teema ei olnud õpilastele sama põnev kui teised.

Kõrgeim hinnang uutele ja huvitavatele teadmistele anti tundides kolm, neli, seitse, kaheksa, üheksa, üksteist, kaksteist ja neliteist, kus õpilased leidsid, et nad omandasid uusi teadmisi ja olid rahul õpikogemusega. Madalaim tulemus (4,38) oli tunnis üks. See oli sissejuhatav tund, mis võib-olla ei pakkunud veel nii palju uusi ja huvitavaid teadmisi võrreldes praktilisemate ja keerukamate teemadega. Kokkuvõttes on tulemused positiivsed, andes märku, et õpilased olid õpikogemusega rahul ning ootasid huviga järgmist tunde.



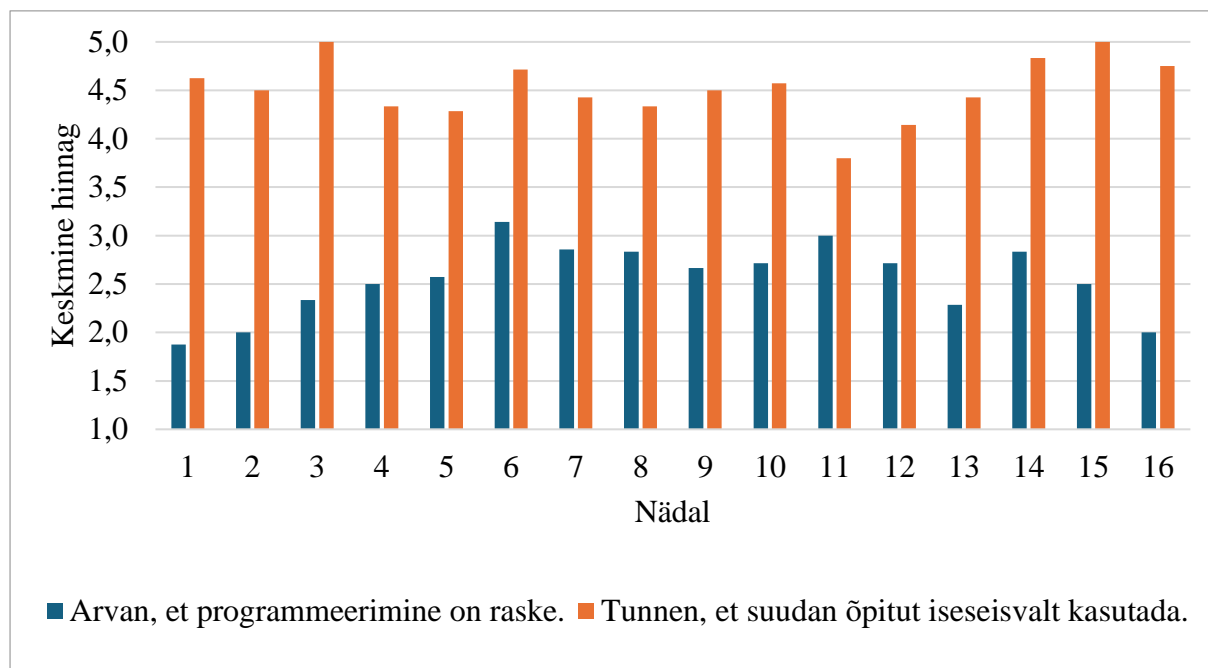
Joonis 16. Keskmine hinnang väidetele: „Ootan juba järgmist tundi!“ ja „Sain täna midagi uut ja huvitavat teada.“ nädalate kaupa.

Joonisel 17 on näha keskmised hinnangud väidetele „Arvan, et programmeerimine on raske“ ja „Tunnen, et suudan õpitut iseseisvalt kasutada“. Väitele „Arvan, et programmeerimine on raske“ vastati kahel korral skaala valikuga „0“ ehk „Ei oska vastata“. Kuna neid vastusevariante ei olnud võimalik panna keskmise arvutamisel arvulisele skaalale, jäeti need analüüsist välja.

Esimese väite keskmine hinnang on 2,55, mis viitab sellele, et enamik õpilastest ei pidanud programmeerimist eriti raskeks. Kõrgeim hinnang (3,14) anti tunnis kuus, kus teemaks oli tsüklid I. See tulemus võib olla seotud sellega, et tsüklite õppimine oli esimene keerulisem teema, mis nõudis õpilastelt rohkem pingutust ja arusaamist. Tsüklite II tunnis langes hinnang programmeerimise raskusele madalamaks, kuna õpilased olid juba tuttavad teema põhialustega ja suutsid neid paremini omandada. Madalaim hinnang (1,88) anti aga esimeses tunnis, kus õpilased ei pidanud programmeerimist raskeks, kuna tegemist oli sissejuhatava tunniga.

Küsimuses „Tunnen, et suudan õpitut iseseisvalt kasutada“ oli keskmine hinnang 4,50, mis näitab, et enamik õpilastest uskusid, et nad suudavad õpitut iseseisvalt rakendada. Kõrgeim hinnang, anti tundides kolm ja viisteist, kus õpilased tundsid, et nad suudavad õpitut praktikas kasutada. Madalaim hinnang (3,80) anti tunnis üksteist, kus teemaks oli funktsioonid I. Järgmisel tunnil, funktsioonid II, tõusis hinnang, mis viitab sellele, et õpilased suutsid paremini

mõista ja rakendada eelmisel tunnil õpitud teadmisi ning olid kindlamad oma võimes iseseisvalt õppimist rakendada.



Joonis 17. Keskmine hinnang väidetele: „Arvan, et programmeerimine on raske.” ja „Tunnen, et suudan õpitut iseseisvalt kasutada.” nädalate kaupa.

Kursuse üldine hinnang õpilaste poolt oli väga positiivne, keskmine hinnang tunni meeldimise kohta oli 4,71, mis viitab sellele, et õpilased said positiivse õpikogemuse. Kõige rohkem meeldisid õpilastele tunnid, kus käsitleti visuaalseid ja praktilisi teemasid. Need tunnid pakkusid õpilastele rohkem lõbu ja loovust, mis aitas suurendada huvi õpitava vastu. Kilpkonnagraafika on hea näide sellest, kuidas visuaalsed ja interaktiivsed tegevused suudavad motiveerida õpilasi ning muuta programmeerimise atraktiivsemaks eriti algajatele.

Tagasiside järgi olid tunnid üksteist ja kolmeteistkümnes õpilaste jaoks kõige keerukamad. Teemad nagu andmevahetus ja funktsioonide mõisted võivad algajatele tunduda esialgu keerulised, kuna need nõuavad rohkem teoreetilist mõistmist. Samuti võivad õpilased tunda, et see teema on vähem kaasahaarav, kuna puudub visuaalne või mänguline efekt, nagu kilpkonnagraafika puhul. Siiski tuleb arvestada, kuna kursuse eesmärgiks on anda õpilastele teadmisi programmeerimise alustest, mis on rakendatavad ka edasistes kursustes on selliste kontseptsioonide õpetamine vajalik. Kuid järgmistest tundidest oli näha, et teema sai selgemaks ja tõusis ka õpilaste hinnang. Tundide järkjärguline ülesehitus ja pidev selgitamine aitas vähendada segadust ja suurendasid õpilaste usku oma võimetusse keerukamate ülesannetega toime tulla.

Kursuse tagasiside viitab üldiselt positiivsele õpikogemusele, kuid teatud kitsaskohad vajavad tähelepanu. Tuleb meeles pidada väikest valimit – seitse õpilast. Selline valim võib mõjutada kursuse üldhinnangut, kuna väikese grupi puhul on tagasiside kallutatud ja ei pruugi kajastada kogu kogemust. Väikese grupi eelised, nagu rohkem aega individuaalseks juhendamiseks ja suurem võimalus õpilastele tähelepanu köita, võivad küll olla positiivsed, kuid ei pruugi anda adekvaatset ülevaadet kõigi õpilaste arvamustest. Üks võimalik lahendus sellele võiks olla järgmistel kursustel suurema õpilasgrupi kaasamine või kursuse läbiviimine korduvalt erinevate gruppidega. Suurema õpilasgrupi puhul saaks paremini hinnata, kuidas õpilased erinevaid teemasid omandavad, millised on levinumad raskused ja kas kõik õpilased on saanud piisavalt tuge keerukamate kontseptsioonide mõistmiseks.

Saadud tagasiside kinnitab, et kursuse eesmärk – luua Pythoni õppematerjalid, mis on arusaadavad ja sobivad põhikooli 6.–7. klassi õpilastele, saavutati. Õpilased said praktilise kogemuse programmeerimisega, tutvuti põhialuste ja lihtsate ülesannetega, mis arendasid nende probleemide lahendamise oskusi ja loogilist mõtlemist.

Järgmistes kursustes on veel ruumi täiendamiseks. Kõige olulisem on pakkuda rohkem praktilisi ja visuaalseid tegevusi, mis aitaksid keerukamate teemade õpetamisel. Näiteks võiks tutvustada uusi kontseptsioone interaktiivsete ülesannete kaudu, mis ühendaksid teooriat ja praktikat. Samuti saaks rohkem arvestada õpilaste individuaalsete vajadustega ja kohandada ülesandeid vastavalt sellele, millised teemad neile rohkem väljakutseid pakkuvad. Kui õpilaste tasemed varieeruvad, on tähtis luua täiendavad toetavad ülesanded ja ülesehitused, mis aitaksid igal õpilasel edasi liikuda omas tempos. Näiteks lisada võimalusi iseseisvateks projektideks või rühmatöödeks, kus õpilased saavad oma ideid ellu viia ja programmeerimisoskusi loovalt rakendada.

6. Kokkuvõte

Bakalaureusetöö raames loodi Pythoni algkursuse õppematerjalid, mis on mõeldud põhikooli 6.–7. klassi õpilastele ja aitavad neil omandada Pythoni programmeerimiskeele põhialused. Koostatud õppematerjal (<https://courses.cs.ut.ee/t/it/Progpython>) koosneb kuueteistkümnest 60-minutilise tunnist, millest igaüks sisaldab nii teoreetilist poolt kui ka praktilisi ülesandeid.

Kursus valmistati ette ADDIE mudelist lähtuvalt ja õppematerjalide koostamisel lähtuti Merrilli mudelist, mille keskmes on teadmiste rakendamine reaalses kontekstis. Mudeli põhimõtted on integreeritud slaidide ülesehitusse: tunnid algavad juba õpitu meelde tuletamisega, koodinäited toetavad kontseptsioonide demonstreerimist, praktilised ülesanded võimaldavad õpilastel uusi teadmisi rakendada, testküsimused aitavad kontrollida arusaamist ja toetavad teadmiste integreerimist. Igas tunnis andsid õpilased ka tagasisidet materjali ja seletuste kohta, et aidata paremini mõista nende vajadusi ja kogemusi kursuse jooksul.

Tulemused näitasid, et õpilased hindasid kursust positiivselt, keskmine hinnang tunni meeldimise kohta oli Likerti skaalal 4,71, mis viitab sellele, et õpilased said positiivse õpikogemuse. Kõige rohkem meeldisid neile tunnid, kus käsitleti visuaalseid teemasid, nagu kilpkonnagraafika, mis muutis programmeerimise lõbusaks ja arusaadavaks. Kuigi tagasiside viitas ka keerukamatele teemadele, nagu andmevahetus ja funktsioonide mõisted, näitas õpilaste hinnangute tõus pärast järgmisi tunde, et need kontseptsioonid said lõpuks selgemaks.

Võib öelda, et kursuse eesmärk saavutati, kuna õpilased said teadmised Pythoni programmeerimiskeelest, mis võivad olla vajalikud edasistes õpingutes. Siiski tuli tagasisidest välja, et keerukamate teemade õpetamine võib vajada rohkem praktilisi tegevusi ja visuaalseid näiteid, et õpilased saaksid neid paremini omandada.

Bakalaureusetöö võib olla kasulik nii õpilastele kui ka õpetajatele. Õpilased saavad õppematerjalidest tuge, et omandada Pythoni põhialuseid, arendades oma praktilisi oskusi ja teoreetilisi teadmisi. Õpetajad saavad kasutada tund-tunnilt loodud materjalide struktuuri, ülesannete jaotust ja tagasisidet, et viia läbi enda kursus.

Edasiarendusena saaks kursuse ülesehitust täiustada, lisades rohkem interaktiivseid ülesandeid ja visuaalseid elemente, et keerukamad kontseptsioonid oleksid veelgi selgemad. Samuti võiks kursust täiendada rohkemate praktiliste projektidega, mis annaks õpilastele võimaluse teha enda valitud projekte ning rakendada õpitud iseseisvalt. Töö autor kavatses materjali kasutada ka edaspidi ja täiustada seda vastavalt vajadusele.

Viidatud kirjandus

- [1] Põhikooli riiklik õppekava. Riigi Teataja. 2024.
<https://www.riigiteataja.ee/akt/110082024002> (21.01.2024)
- [2] Bahromova Muhayyo Mansurjon qizi. The Importance and Necessity of Teaching Computer Science and Programming for Primary School Students. *Current Research Journal of Pedagogics*, 2021, Vol. 2, No. 9, p. 192-196. doi.org/10.37547/pedagogics-crjp-02-09-40
- [3] Põhikooli riiklik õppekava. Valikõppeaine „Informaatika“. 2024.
https://www.riigiteataja.ee/aktilisa/1100/8202/4002/18m_pohi_lisa10.pdf# (21.01.2025)
- [4] Shein, E. Python for beginners. *Communications of the ACM*, 2015, Vol. 58, No. 2, p. 19-21, doi.org/10.1145/2716560
- [5] TIOBE Programming Community Index Definition. https://www.tiobe.com/tiobe-index/programminglanguages_definition/ (18.12.2024)
- [6] TIOBE Index for December 2024. <https://www.tiobe.com/tiobe-index/>(21.01.2025)
- [7] Annamaa, A. Introducing Thonny, a Python IDE for Learning Programming. *In Proceedings of the 15th Koli Calling Conference on Computing Education Research*, 2015, p. 117-121. doi.org/10.1145/2828959.2828969
- [8] Keskkonna lahendus.ut.ee kasutajatingimused. Tartu Ülikooli Arvutiteaduste instituut. https://docs.google.com/document/d/1dk1Pp3hXJEX7HlIQFdMFo5AXhgzy4zhZv3Qt6-xI_CI/edit?tab=t.0#heading=h.df01thmxe31v (28.11.2024)
- [9] Tartu Ülikooli Informaatika didaktika töörühm. Lahendus.ut.ee kasutusjuhend õpetajale. <https://didaktika.cs.ut.ee/lahendus-kasutusjuhend-opetajale/> (10.04.2025)
- [10] Thonny plugin for lahendus.ut.ee. <https://pypi.org/project/thonny-lahendus/> (19.04.2025)
- [11] Villems A., Aluoja L., Pilt L., Naulainen M.-M., Kusmin M., Rogalevitš V., Tokko U. Digitaalse õppematerjali loomise soovitused - Juhend digitaalse õppematerjali autorile.
<https://oppevara.edu.ee/kvaliteet> (18.12.2024)

- [12] Kurt S. ADDIE Model: Instructional Design, 2017.
<https://educationaltechnology.net/the-addie-model-instructional-design> (20.04.2025)
- [13] Eesti Kõrg- ja Kutsehariduse Kvaliteediagentuur e-kursuse kvaliteedi töörühm. Juhend kvaliteetse e-kursuse loomiseks. 2021. <https://oppevara.edu.ee/ekursus> (27.02.2025)
- [14] Hidayanto, D.R., Rahman, E.F., Kusnendar, J. The application of ADDIE model in developing adventure game-based multimedia learning to improve students' understanding of basic programming. In *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 2017, pp. 307-312. doi.org/10.1109/ICSITech.2017.8257130
- [15] Merrill, M. D. First Principles of Instruction. *Educational Technology Research and Development*, 2002, Vol. 50, No. 3, pp. 43-59. <https://mdavidmerrill.wordpress.com/wp-content/uploads/2019/04/firstprinciplesbymerrill.pdf> (19.04.2025)
- [16] Maia, L. Merrilli õppedisaini mudel. 2015.
<https://maialusttlu.wordpress.com/2015/10/17/merrilli-oppedisaini-mudel/comment-page-1> (27.02.2025)
- [17] Badali, M., Hatami, J., Farrokhnia, M., Noroozi, O. The effects of using Merrill's first principles of instruction on learning and satisfaction in MOOC. *Innovations in Education and Teaching International*, 2022, Vol. 59, No. 2, pp. 216-225.
doi.org/10.1080/14703297.2020.1813187
- [18] Merrill, M. D. A Task-Centered Instructional Strategy. *Journal of Research on Technology in Education*, 2007, Vol. 40, No. 1, pp. 5-22.
[https://app.nova.edu/toolbox/instructionalproducts/edd8124/articles/2007-Merrill-A%20task-centered Instructional Strategy.pdf](https://app.nova.edu/toolbox/instructionalproducts/edd8124/articles/2007-Merrill-A%20task-centered%20Instructional%20Strategy.pdf) (19.04.2025)
- [19] Joshi, A., Kale, S., Chandel, S., Pal, D.K. Likert Scale: Explored and Explained. *Current Journal of Applied Science and Technology*, 2015, Vol. 7, No. 4, pp. 396-403.
doi.org/10.9734/BJAST/2015/14975

Lisad

I. Lapsevanema nõusoleku vorm

Pildi abil on esitatud lapsevanema nõusoleku vorm.

Lugupeetud lapsevanem!

Olen Nutiklassi Python ja Minecraft PI huviringide juhendaja Annabel Maksimov. Õpin Tartu Ülikoolis bakalaureuseõppes informaatikat ja oma lõputöö raames koostan õppematerjale, mis on mõeldud põhikooli õpilastele Pythoni programmeerimiskeele õppimiseks.

Soovin Teie luba kasutada lõputööks Teie lapse õppetöös osalemisega seotud õpitulemusi ja arvamust tundide ja materjali kohta.

Lõputöös kasutatavad andmed on konfidentsiaalsed ning neid näen ainult mina, töö autor. Andmeid kogun eesmärgiga uurida, kuidas õpilased materjale omandavad ning kuidas õppematerjale paremini kohandada õpilaste vajadustele.

Täiendavatele küsimustele vastan e-maili teel: annabel@nutiklass.ee.

Palun täidetud vormi tagasi hiljemalt järgmiseks huviringi tunniks.
Aitäh!

Mina, (nimi), luban oma lapsel (nimi) osaleda Annabel Maksimovi lõputöös, mis käsitleb Pythoni õppematerjalide loomist. Olen teadlik, et kõik andmed on konfidentsiaalsed ning mul on hiljem võimalus tutvuda lõputööga.

.....

Allkiri

.....

Kuupäev

II. Kursuse esimesel tunnil läbiviidud avaküsimustik

Piltide abil on esitatud kursuse alguses läbiviidud küsimustik uurimaks õpilaste tausta.

Mis rühmas oled? *

- Python 1 (Esmaspäevane grupp)
- Python 1 (Neljapäevane grupp)

Kas sa oled varem programmeerimisega kokku puutunud? *

- Olen varem läbinud Nutiklassis Scratch kursuse
- Olen varem läbinud Nutiklassis Python 1 kursuse
- Olen iseseisvalt õppinud programmeerimist
- Olen koolis õppinud programmeerimist
- Ei ole varem programmeerimisega kokku puutunud
- Muu: _____

Milliseid programmeerimiskeeli oled varem kasutanud?

- Python
- Java
- C, C++
- HTML
- Scratch
- CSS
- Muu: _____

Kas sul kodus on võimalik kasutada arvutit programmeerimisülesannete lahendamiseks?

- Ei
- Jah
- Muu: _____

Kuidas nõustud järgmiste väidetega? *

	0 - Ei oska vastata	1 - Ei nõustu üldse	2 - Pigem ei nõustu	3 - Neutraalne	4 - Pigem nõustun	5 - Nõustun täielikult
Mulle meeldib mängida arvutimänge	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mulle meeldib matemaatika	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mulle meeldib lahendada erinevaid loogilisi ülesandeid	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Arvan, et programmeerimine on raske	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Olen huvitatud programmeerimise õppimisest	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kasutan igapäevaselt oma telefoni/arvutit/muud nutiseadet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Millised on su ootused ja soovid käesoleva kursuse suhtes? *

Teie vastus

Mis võib olla suurim takistus kursuse läbimisel? *

Teie vastus

III. Tunni lõpus läbi viidud tagasiside ankeet

Pildi abil on esitatud iga tunni lõpus läbiviidud tagasiside ankeet.

Mis rühmas oled? *

Python 1 (Esmaspäevane grupp)

Python 1 (Neljapäevane grupp)

Vali tunni number *

Valige ▼

Kuidas nõustud järgmiste väidetega? *

	0 - Ei oska vastata	1 - Ei nõustu üldse	2 - Pigem ei nõustu	3 - Neutraalne	4 - Pigem nõustun	5 - Nõustun täielikult
Mulle meeldis tänane tund.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ülesanded olid minu jaoks jõukohased.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jõudsin kogu tunni materjaliga valmis.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Arvan, et programmeerimine on raske.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Õpetaja seletas teemat piisavalt lahti.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tunnen, et suudan õpitut iseseisvalt kasutada.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sain täna midagi uut ja huvitavat teada.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ootan juba järgmist tundi!	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

IV. Kursuse ainekava

Piltide abil on esitatud kursuse ainekava.

Ainekava: Pythoni algkursuse materjalid

Vanus: 14-18 aastased

Tundide arv: 16 tundi (60 minutilised)

Kasutatav õppekirjandus: <https://courses.cs.ut.ee/t/it/Progpython>

Vajalik taristu: igale õpilasele arvuti, interneti võimalus, arenduskeskkond Thonny

Läbitavad teemad:

- **Sissejuhatus:** Programmeerimise põhialused, programmeerimiskeeled, avaküsimustik.
- **Algoritm:** Algoritmi mõiste, algoritmide koostamine ja realiseerimine.
- **Arenduskeskkond:** Arenduskeskkond Thonny, selle seadistamine ja kasutamine.
- **Andmed:** Erinevad andmetüübid Pythonis. Andmete esitamine ja töötlemine programmides.
- **Muutujad:** Muutujale väärtuse omistamine ja kasutamine.
- **Sisend ja väljund:** Kasutajalt sisendi võtmine ja tulemuste kuvamine ekraanil.
- **Valiklaused:** Tingimuslikud laused (if, elif, else) otsuste tegemiseks programmides.
- **Tsüklid:** Korduvate tegevuste automatiseerimine tsüklite (while ja for) abil.
- **Andmestruktuurid:** Järjendid ja sõnastikud andmete organiseerimiseks ja töötlemiseks.
- **Funktsioonid:** Koodi korduvkasutus ja organiseerimine funktsioonide abil.
- **Failid/Andmevahetus:** Andmete lugemine ja salvestamine failidesse.
- **Veahaldus:** Vigade käsitlemine (try ja except).
- **Kilpkonnagraafika:** Visuaalsete jooniste ja kujundite loomine kilpkonnagraafika (Turtle) abil.

Õpitulemused kursuse läbimisel:

- Õpilane mõistab programmeerimise põhialuseid ja oskab selgitada programmeerimisega seonduvaid mõisteid,
- Õpilane oskab kasutada arenduskeskkonda Thonny,
- Õpilane oskab kirjutada lihtsaid programme,
- Õpilane suudab luua ja kasutada muutujaid ning erinevaid andmetüüpe,
- Õpilane oskab lugeda ja salvestada andmeid failidesse,
- Õpilane suudab luua ja kasutada funktsioone,
- Õpilane mõistab veahaldust ja oskab kasutada try-except struktuuri,
- Õpilane suudab luua lihtsamaid visuaalseid jooniseid ja kujundeid, kasutades kilpkonnagraafikat.

Ainekava tabelina

Õppeteema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
Sissejuhatus ja Algoritm	1	programmeerimine, algoritm, plokk skeem	Tunni esimeses osas toimub õpilastega tutvumine, arvutisse kasutajate loomine, avaküsimustiku täitmine, et näha õpilaste eelnevat kogemust ja meeletatust programmeerimise õppimisest. Tunni teises osas selgitab õpetaja slaidide ja näidete abiga, mis on programmeerimine, algoritm, plokk skeem. Tunni kolmandas osas joonistavad õpilased ise algoritmi paberile. Ülesanded: Mis on programmeerimine? Mis on algoritm? Mis on plokk skeem? Kirjuta algoritm ühe enda tegevuse kohta. Tee algoritm plokk skeemiks.	Õpilane koostab algoritmi ühest enda tegevusest. Teeb sama algoritmi ka plokk skeemis. Hindamine vastavalt, kas on tehtud või tegemata.	Algoritmi võib koostada erinevatest elulistest näidetest- kirjanduses raamatu sisu, kehalises kasvatuses trennikava koostamine jne.	https://courses.cs.ut.ee/t/it/Progpython/Tund1 Veebilehel on saadaval slaidid
Arenduskeskkond	0.5	Arenduskeskkond, programm	Tunni esimeses osas toimub eelmise tunni kordamine, siis seletab õpetaja mis on programm. Teises osas tutvustab õpetaja erinevaid arenduskeskkondi ja Thonnyt.	Õpilane teab ja oskab nimetada erinevaid arenduskeskkondi, konto on loodud automaatkontrolli keskkonda	Kasutades programmeerimis keskkonda nagu Thonny, saab matemaatikaga lõimida lihtsate	https://courses.cs.ut.ee/t/it/Progpython/Tund2

Õppeteema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>Tunni lõpuks juhend koduseks Thonny installimiseks. Ning automaatkontrolli keskkonda (lahendus.ut.ee) kontode loomine.</p> <p>Ülesanded: Mis on programm? Mis on arenduskeskkond?</p> <p>Ülesanne: "Tere maailm!"</p>	<p>Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud või tegemata</p>	<p>matemaatiliste probleemide lahendamise programme jne</p>	<p>Veebilehel on saadaval slaidid</p>
Andmed	0.5	<p>Andmetüüp</p> <p>Täisarvud (int, integer)</p> <p>Ujukomaarvud (float, floating-point number)</p> <p>Sõned (str, string)</p> <p>Tõeväärtused (bool, boolean)</p>	<p>Tunni esimeses osas seletab õpetaja mis on andmetüüp ja 4 peamist andmetüüpi (täisarvud,ujukomaarvud, sõned, tõeväärtused). Õpilased arvavad andmetüüpe</p> <p>Tunni teises osas teevad õpilased õpetaja abiga esimese programmi, kus kontrollivad ja väljastavad andmetüüpe.</p> <p>Tunni lõpus seletab õpetaja andmetüüpide teisen-dust ja katsetavad koos õpetajaga. Pärast ülesanne ja suuline test teisen-duste kohta.</p> <p>Ülesanded: Millised on andmetüübid? Kuidas andmetüüpe Pythonis kontrollida? Kuidas andmetüüpe teisen-dada ja liita?</p> <p>Ülesanne: "Erinevad andmetüübid"</p>	<p>Õpilane teab nelja andmetüüpi, oskab neid Pythonis kontrollida</p> <p>Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud või tegemata</p>	<p>Arvude mõistmine: Täisarvude (int) jaujukomaarvude (float) käsitlemine programmeerimis es toetab matemaatika tundides õpitud.</p> <p>Tõeväärtused (bool) on abiks loogika ja tingimuste mõistmisel</p>	<p>https://course.s.cs.ut.ee/t/it/Progpython/Tund2</p> <p>Veebilehel on saadaval slaidid</p>

Õppeteema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>Ülesanne: "Andmetüüpide teisen-damine"</p>			
Muutujad	1	<p>Muutujad</p> <p>Muutuja määramine</p> <p>Tehted sõnede ja arvudega</p>	<p>Tunni esimeses osas toimub eelmise tunni kordamine ja uude tundi sissejuhatus: muutujad. Arutatakse, et millised võivad olla muutuja nimed?</p> <p>Tunni teises osas käsitletakse muutujate määramist, kasutamist, järjekorda ning muutumist. Lahendatakse iseseisvalt kaks ülesannet.</p> <p>Tunni lõpu poole selgitatakse kuidas saab korraga määrata mitu muutujat. Õpetaja näidete abil vaadatakse sõnedega erinevaid tehteid ja pärast selle koha ka iseseisvalt ülesanne.</p> <p>Ülesanne: "Aasta liblikas 2024"</p> <p>Ülesanne: "Tehted arvudega"</p> <p>Ülesanne: "Tehted sõnedega"</p>	<p>Õpilane teab mis on muutuja ja kuidas seda kasutada.</p> <p>Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud või tegemata.</p>	<p>Muutujad programmeerimis es on nagu igapäevaelus objektid, millele anname nimed.</p> <p>Need nimed aitavad väärtusi määrata ja neid hiljem muuta.</p> <p>Nii nagu elus muutuvad asjad, võivad ka muutujate väärtused programmeerimis es muutuda.</p>	<p>https://course.s.cs.ut.ee/t/it/Progpython/Tund3</p> <p>Veebilehel on saadaval slaidid</p>
Sisend ja väljund	0.5	<p>Sisend</p> <p>Väljund</p>	<p>Tunni esimeses osas toimub eelmise tunni kordamine ja uude tundi sissejuhatus: sisend ja väljund</p>	<p>Õpilane teab, kuidas küsida kasutajalt erinevat tüüpi sisendeid.</p>	<p>Matemaatika, kus õpilane kasutab sisendite kaudu saadud väärtusi</p>	<p>https://course.s.cs.ut.ee/t/it/Progpython/Tund4</p>

Õppe-teema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
		Kasutajalt erinevate sisendite küsimine	Tehakse õpetajaga läbi erinevaid näiteid. Iseseisvalt ülesanne "Sisenditega arvutamine". Peale seda tuleb uus teema - tingimuslause Ülesanne: "Sisenditega arvutamine"	Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud või tegemata.	lihtsate arvutuste tegemiseks	Veebilehel on saadaval slaidid
Valiklauseid / Tingimuslause	1.5	Valiklause Tingimuslause Tingimuslause tingimuslause sees Loogilised avaldised	See teema algab samas tunnis, mis sisend ja väljund. Õpetaja seletab, mis on tingimuslause, selle struktuuri (if, elif, else, koolon, taanded jne) ja milliste sümbolitega saab võrrelda. Tehakse kaks suulist testi tingimuslause tulemuse kohta. Ülesanne: "Arvu arvamine" Ülesanne "Ülikooli hindamine" Teine tund algab eelmise tunni kordamisega. (Sisend ja väljund, tingimuslause ja näited) Siis seletab õpetaja, et tingimuslause saab tingimuslause sees ka olla ja näide koos õpilastega	Õpilane teab, kuidas kasutada tingimuslauseid. Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud või tegemata.	Valiklauseid saab siduda igapäevaste otsustega, näiteks eelarve haldamine: "Kui toote hind ületab eelarvet, siis kuvab 'Ei saa osta', kui hind on eelarves, siis 'Saab osta'". Tingimuslauseid saab lõimida ka matemaatikaga, näiteks nõuavad teatud arvuliste	https://course.s.cs.ut.ee/t/it/Progpyhton/Tund4 https://course.s.cs.ut.ee/t/it/Progpyhton/Tund5 Veebilehel on saadaval slaidid

Õppe-teema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			ning suuline test selle kohta. Seejärel teevad õpilased ülesande. Tunni keskel seletab õpetaja loogilisi tehted ja avaldise, suulised testid selle kohta. Tunni lõpus kaks iseseisvat ülesannet vajadusel õpetaja abiga. Ülesanne: "Lennupileti soodustused" Ülesanne: "Seifi parool" Ülesanne: "Temperatuuri põhjal riietussoovitus"		kriteeriumite kontrollimist.	
Tsüklid	2	Tsükel Korduslause Paaris ja paaritu arv Juhusliku numbri genereerimine	Tund algab sissejuhatusena korduvasse tegevusse (while tsükliks). Õpetaja seletab jätkamistingimust, while tsükli struktuuri ja toimimist. Vaadatakse läbi ka väärtuse muutmise lühivariandid (nt arv += 1). Peale seda suuline test jätkamistingimuste kohta, ja while tsükli ülesanne "Trenn". Peale ülesannet veel suuline test selle kohta, mis tuleb ekraanile. Siis õpetaja seletab, kuidas saab leida paaris ja paaritu arvu, ning sellega ülesanne "Paaris ja paaritu arv".	Õpilane teab, kuidas kasutada while tsükli ja leida paaris ja paaritu arvu ning juhuslikku numbrit randinti abil. Õpilane teab, kuidas kasutada alati tõest tsükli (while True). Oskab rakendada break ja continue käsk.	Elus on korduslauseid seotud igapäevaelu tegevustega, kus on vaja midagi korrata või jälgida, kuni teatud tingimus on täidetud. Tsüklid on tihti aluseks	https://course.s.cs.ut.ee/t/it/Progpyhton/Tund6 https://course.s.cs.ut.ee/t/it/Progpyhton/Tund7

Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>Täiendatakse ka eelmises tunnis lahendatud ülesande Seifi parooli, kus lisatakse tsüklitel, et kasutaja saab kolm korda proovida parooli sisestada.</p> <p>Tunni lõpus õpetaja seletab, kuidas saab genereerida suvalist arvu (randint) ja sellega ülesanne "Numbri arvamine".</p> <p>Teine tund algab eelmise tsükli tunni kordamisega. Kordamiseks randint ülesanne "Juhuslik summa".</p> <p>Siis õpetaja seletab alati tõest tsüklit koos näitega (while True, käsud break ja continue). Siis täiendatakse näidet ja õpilased lahendavad ülesande "Korrutustabel" ja "Jagamine 2 ja 3 järgi". Koos arutletakse ka küsimuste üle.</p> <p>Tundi lõpetab ülesanne "Jänesevanemate mure".</p> <p>Ülesanne "Jänesevanemate mure" versioon 2 on lisaülesanne kiirematele.</p> <p>Ülesanne: "Trenn"</p> <p>Ülesanne: "Paaris ja paaritu arv"</p>	<p>Oskab kasutada tsüklite ja tingimuste abil keerukamate ülesannete lahendamiseks, nagu "Jänesevanemate mure".</p> <p>Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud/tegemata</p>	<p>automaatsetele süsteemidele (nt kodumasinad, mis töötavad seni, kuni kindel tingimus on täidetud).</p> <p>Tsüklitel, kus kasutaja saab mitu korda proovida parooli sisestada, samaneb päris elu olukorraga, kus meil on piiratud arv katseid, et täita mõni turvameede (nt PIN-kood või salasõna).</p>	Veebilehtedel on saadaval slaidid

Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>Ülesanne: "Seifi parool 2"</p> <p>Ülesanne: "Numbri arvamine"</p> <p>Ülesanne: "Juhuslik summa"</p> <p>Ülesanne: "Korrutustabel"</p> <p>Ülesanne: "Jagamine 2 ja 3 järgi"</p> <p>Ülesanne: "Jänesevanemate mure" ja "Jänesevanemate mure ver 2"</p>			
Andmestruktuurid	3	<p>Andmestruktuur</p> <p>Järjend</p> <p>Indeksid ja nende leidmine</p> <p>Alamjärjendid (viilutamine)</p> <p>Sõnastik</p> <p>Võti</p>	<p>Tund algab eelmise tunni lisaülesande selgitusega (Jänesevanemate mure ver 2). Siis sissejuhatus uude teemasse - andmestruktuurid. Mis on andmestruktuur? Mis on järjend, kuidas see käib? Leiame järjendi tüübi.</p> <p>Siis õpetaja selgitab, mis on indeksid ja kuidas saab elementi kätte indeksi abil (+näide). Suuline test: "Mitmes indeks on?". Jätkub teema negatiivsetest indeksitest, elemendi indeksi leidmisest, järjendi pikkuse leidmisest (+näited). Suuline test, et mis ilmub ekraanile.</p>	<p>Õpilane teab:</p> <p>Mis on andmestruktuur ja milleks neid kasutatakse.</p> <p>Mis on järjend ja kuidas seda kasutada.</p> <p>Kuidas töötada indeksite ja negatiivsete indeksitega.</p>	<p>Andmestruktuurid aitavad meil andmeid korraldada, nagu hoiame asju kindlates kohtades.</p> <p>Indeksid on nagu aadressid, mis näitavad, kus midagi asub.</p> <p>Sõnastik lõimub eluga, sest see</p>	<p>Järjendi slaidid:</p> <p>https://course.s.cs.ut.ee/t/it/Progpyhton/Tund8</p> <p>https://course.s.cs.ut.ee/t/it/Progpyhton/Tund9</p>

Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
		Väärtus	<p>Tunni keskel vaadatakse, kuidas saab leida alamjärjendeid koos näitega. Siis ülesanne "Järjend" siiani õpitud kinnistamiseks.</p> <p>Siis vaadatakse ja tehakse läbi järjendiga kasutatavad funktsioonid ja operatsioonid. Tunni lõpetab ülesanne "Järjendite operatsioonid ja funktsioonid".</p> <p>Teine tund algab eelmise tunni kordamisega. Vajadusel lõpetatakse ülesanne: "Järjendite operatsioonid ja funktsioonid", kui see jäi eelmise tunni lõpus poolikuks.</p> <p>Siis vaadatakse näidete abil, et kuidas saab järjendisse elemente lisada (append ja insert) ning eemaldada (remove ja pop). Vaadatakse ka kuidas saab olemasolevat elementi muuta indeksi abil. Peale teooriat harjutus ja suuline test elementide lisamise ja eemaldamise kohta.</p> <p>Tunni keskel käiakse while tsükli abil läbi kõik järjendi elemendid. Siis õpitakse ka for tsükli kasutama. Lahendatakse ülesanne for tsükli abil. Tunni lõpupoole vaadatakse ka üle funktsioon</p>	<p>Kuidas leida elementide indekseid ja järjendi pikkust.</p> <p>Kuidas leida alamjärjendeid.</p> <p>Milliseid funktsioone ja operatsioone saab järjenditega kasutada.</p> <p>Kuidas järjendisse lisada ja eemaldada elemente.</p> <p>Järjendi läbimine while ja for tsükkliga. For tsükkel ja funktsioon range.</p> <p>Mis on sõnastik andmestruktuurina (võtme ja väärtuste seosed).</p> <p>Kuidas sõnastikus saab lisada, muuta ja</p>	aitab korrastada andmeid (nt telefoniraamat), kus iga inimese nimi (võti) on seotud tema telefoninumbri (väärtus).	<p>Sõnastiku slaidid:</p> <p>https://course.s.cs.ut.ee/t/it/Progpython/Tund10</p> <p>Veebilehtedel on saadaval slaidid</p>
Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>range näidete abil. Suuline test range kohta ja ülesanne.</p> <p>Kolmas tund algab järjendi ja for tsükli kordamisega. Siis uus teema - sõnastik. Tutvutakse sõnastikuga (võti, väärtus, omadused) näidete abil. Õpitakse kuidas saab sõnastikust väärtusi leida ja lisada ning kuidas need erinevad järjendist.</p> <p>Tunni keskel suuline test valikvastustega. Peale seda tehakse läbi sõnastiku põhifunktsioonid ja operatsioonid. Järgneb iseseisev ülesanne "Õpilaste vanus". Siis vaadatakse, kuidas saab sõnastikku erinevat moodi for tsükkliga läbida (+näited). Tunni lõpetab iseseisev ülesanne "Kinkide nimekiri". Vajadusel lisäülesanne "Autoloendur".</p> <p>Ülesanne: "Järjend"</p> <p>Ülesanne: "Järjendite operatsioonid ja funktsioonid"</p> <p>Ülesanne: "Summa segajärjendis"</p> <p>Ülesanne: "Arvude summa numbrini"</p> <p>Ülesanne: "Õpilaste vanus"</p>	<p>eemaldada võtme-väärtuse paare.</p> <p>Põhifunktsioonid: .get(), .keys(), .values(), .items() ja kuidas neid kasutada.</p> <p>Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud/tegemata</p>		

Õppeteema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			Ülesanne: "Kinkide nimekirj" Lisaülesanne: "Autoloendur"			
Funktsioonid	2	Funktsioon Sisseehitatud funktsioon Argument Parameeter Tagastamine (return) Funktsioon split	Tund algab eelmise tunni kordamisega (sõnastik, lisaülesanne: "Autoloendur") Peale kordamist sissejuhatus funktsioonidesse. Mis on (sisseehitatud) funktsioon ja parameetrid ehk argumentid. Vaadatakse õpetaja näidete abil, kuidas funktsiooni defineerida? Suuline test, mis read ilmuvad ekraanile. Ülesanne ringi pindala arvutamiseks, kus alguses ei ole argumente. Pärast täiendamise ülesannet, andes ette raadiuse argumenti. Suuline test argumentide kohta. Iseseisev ülesanne "Kassi vanus". Siis vaadatakse näidete abil funktsiooniga tulemuse tagastamist (return). Näide tehakse ka õpetaja abiga läbi. Tunni lõpetab iseseisev ülesanne "Hindamine". Teine tund on keskendunud erinevate ülesannete lahendamisele funktsioonidega. Tund algab	Õpilane teab: Mis on funktsioon, kuidas seda defineerida ja kasutada. Millised on funktsiooni argumentid ja kuidas neid kasutada. Kuidas funktsioonid tagastavad väärtusi (return). Oskab rakendada funktsioonide teadmisi praktilistes iseseisvates ülesannetes. Oskab koostada funktsioone erinevate ülesannete lahendamiseks.	Funktsioonid võimaldavad lihtsalt korduvkasutatavat koodi. Nii saab koodi jagada väiksemateks osadeks ja kood on ka loetavam.	https://course.s.cs.ut.ee/t/it/Progpython/Tund11 https://course.s.cs.ut.ee/t/it/Progpython/Tund12 Veebilehtedel on saadaval slaidid
Õppeteema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			funktsiooni kordamisega ja vajadusel eelmise tunni ülesande "Hindamine" lõpetamisega. Siis tehakse funktsiooni kordamiseks ülesanne "Peo eelarve". Seejärel õpitakse mida teeb funktsioon split() näidete abil. Spliti kordamiseks ülesanne "Kuupäev". Tunni lõpetab ülesanne "Taishäälkute tuvastaja" Ülesanne: "Kassi vanus" Ülesanne: "Hindamine" Ülesanne: "Peo eelarve" Ülesanne: "Kuupäev" Ülesanne: "Taishäälkute tuvastaja"	Oskab jagada teksti split-meetodiga ja töödelda saadud tekstiosi. Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud/tegemata		
Andmevahetus	1.5	Kodeering (encoding UTF-8)	Tund algab vajadusel eelmise teema kordamisega ja ülesande "Taishäälkute tuvastaja" lõpetamisega. Peale eelmise teema kordamist tuleb uus teema - Failist lugemine. Luuakse tekstifail ja õpitakse selle avamist ja sulgemist.	Õpilane teab ja oskab: Mis on tekstifail ja kuidas seda luua. Mis on kodeering (nt UTF-8) ja miks see on oluline.	Failidega töötamise teema võimaldab lõimida programmeerimis oskusi praktiliste ülesannetega erinevates	https://course.s.cs.ut.ee/t/it/Progpython/Tund13

Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>Siis õpitakse, kuidas failist saab andmeid lugeda (read(); readlines()) ja for tsükli abil. Räägitakse ka kodeeringust (UTF-8 ja näited) ning meetodist strip().</p> <p>Näide kuidas saab ka faile avada with open()-iga.</p> <p>Teine tund algab failist lugemise kordamisega. Siis vaadatakse kuidas saab tekstifaili kirjutada. Mis on erinevused avamise tähtedega (a ja w)</p> <p>Õpetajaga koos ülesanne "Tärnide kirjutamine faili".</p> <p>Iseseisvad ülesanded: "Tehnika" ja "Päevik"</p> <p>Ülesanne: "Tühjad read".</p> <p>Ülesanne: "Faili ridade arv"</p> <p>Ülesanne: "Tärnide kirjutamine faili"</p> <p>Ülesanne: "Tehnika"</p> <p>Ülesanne: "Päevik"</p>	<p>Kuidas avada ja sulgeda faili Pythonis.</p> <p>Kuidas saab tekstifailist lugeda ja sinna kirjutada.</p> <p>Mis on strip() meetod ja kuidas seda kasutada.</p> <p>Hindamine vastavalt kas ülesanne automaatkontrolli keskkonnas on ülesanded tehtud/tegemata</p>	<p>valdkondades. Näiteks saab tekstifailist analüüsida andmeid (ilmavaatlused, hinnangud), töödelda tekste (sõnade või täishäälikute loendamine) või luua loovaid projekte, nagu tsitaatide sorteerimine.</p>	<p>https://course.s.cs.ut.ee/t/it/Progpyhton/Tund14</p> <p>Veebilehtedel on saadavad slaidid.</p>

Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
Veahaldus	0.5	Vead Try Except	<p>Räägitakse millised tüüpi vead on olemas (süntaksivead ja erandid).</p> <p>Tekitatakse ja arvatakse õpetaja abiga erinevaid erindeid (ValueError, TypeError, ZeroDivisionError, IndexError jne)</p> <p>Räägitakse kuidas erinevaid erindeid kinni püüda (try-except).</p> <p>Ülesanne: "Erandid arvudega"</p>	<p>Õpilane teab ja oskab:</p> <p>erindite tüüpe ja millal need tekivad.</p> <p>kasutada try-except plokket, et erindeid kinni püüda ja neid lahendada.</p> <p>kuidas kontrollida kasutaja sisendit ja kuvada kasutajasõbralikke veateateid.</p>	<p>Erindite käsitlemine aitab rakendustes välja selgitada, mis on valesti läinud, ja reageerida sellele õigesti, et programm ei katkeks, vaid jätkaks tööd.</p>	<p>https://course.s.cs.ut.ee/t/it/Progpyhton/Tund14</p> <p>Veebilehel on saadavad slaidid.</p>
Kilpkonnagraafika	2	Turtle	<p>Tund algab sissejuhusega mis on kilpkonnagraafika ja kuidas seda importida. Seejärel vaadatakse káske (liikumist ja pöörmist, pliitsi juhtimist ja peitmist, kiirust jne)</p> <p>Tehakse õpetaja abiga ruut (tsükliliga ja ilma). Seejärel joonistavad õpilased ise kujundeid (ruut, kolmnurk, kuusnurk, ring jne).</p> <p>Siis vaadatakse kuidas saab joone värvi muuta ja tervet kujundit värvida. Proovitakse ka juba tehtud kujundite peal läbi.</p>	<p>Õpilased saavad aru, mis on kilpkonnagraafika ja kuidas seda importida ning kasutada.</p> <p>Õpilased oskavad Turtles teha mitut objekti.</p> <p>Õpilased oskavad kasutada juhuslikkust (randint)</p>	<p>Geomeetriselised kujundid (ruut, kolmnurk, kuusnurk) ja nende omadused (külgede pikkus, nurgad, sümmeetria).</p>	<p>https://course.s.cs.ut.ee/t/it/Progpyhton/Tund15</p> <p>https://course.s.cs.ut.ee/t/it/Progpyhton/Tund16</p> <p>https://course.s.cs.ut.ee/t/it/</p>

Õppeema	Tundide arv	Alateemad ja mõisted	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Lõiming	Kasutatav lisamaterjal veebis
			<p>Tunni lõpetab iseseisev ülesanne, kus õpilased peavad joonistama kuusepuu koos kaunistustega.</p> <p>Teine tund algab eelmise tunni kordamisega (kásud, vajadusel kuusepuu joonistuse lõpetamine)</p> <p>Seejärel vaadatakse veel paari kásku ja kuidas mitut objekti teha (Turtle()).</p> <p>Seejärel koos õpetajaga suurem ülesanne: "Kilpkonnade võidujooks".</p>			<p>Progpyhton/Tund17</p> <p>Veebilehtedel on saadavad slaidid.</p>

V. Eestikeelsete Pythoni õppematerjalide analüüs

	Digiõpik „Progema”	Digiõpik „Python koolis”	Kursus „Teeme ise arvutimänge”	Kursus „Programmeerimisest maalähedaselt”	Videokursus „Programmeerimine Pythonis”
Materjali veebileht	https://www.progema.ee/	https://courses.cs.ut.ee/t/pythonkoolis	https://courses.cs.ut.ee/t/tiam/Main/HomePage	https://courses.cs.ut.ee/2024/progmaa/fall	https://e-koolikott.ee/et/oppematerjal/32909-Videokursus-Programmeerimine-Pythonis
Hinnanguline sihtrühm	Põhikool	Põhikool	Gümnaasium	Täiskasvanud	Põhikooli III kooliaste ja gümnaasium
Materjali vorm	Tekst	Tekst	Tekst ja videod	Tekst ja videod	Videod
Materjali ülesehitus	Teemade kaupa	Teemade / tundide kaupa	Teemade raamatute / nädalate kaupa	Teemade / nädalate kaupa	Tundide kaupa
Materjali kirjeldus	Materjali kirjeldus Tekst, näidiskoodid, joonised, enesekontrollid, ülesanded, vihjed	Tekst, näidiskoodid, ülesanded	Tekst, mõni video, koodinäited piltidena	Tekst, videod, näidiskoodid, joonised, enesekontrollid, ülesanded, murelahendaja	Seletused, ülesanded ja näidiskoodid on esitatud videos
Teemad	1. Põhiteadmised 1.1 Sissejuhatus 1.2 Algoritm 1.3 Muutujad ja avaldised 1.4 Tingimuslauseid 1.5 Korduslauseid 1.6 Funktsioonid 1.7 Failid 1.8 Veahaldus 2. Andmestruktuurid 2.1 Järjendid 2.2 Sõnastikud 2.3 Hulgad 2.4 Ennikud	1. Paigaldamine 2. Kommentaarid 3. Muutujad 4. Andmetüübid 5. Sõned 6. Operaatorid 7. Sisend ja väljund 8. Tingimus 9. Tsükkel 10. Algoritm ja plokk skeem 11. Järjend 12. Sõnastik 13. Funktsioon 14. Failioperatsioonid	1. raamat: esimene tutvus Pythoniga, sisend, väljund, muutujad jms sissejuhatus 2. raamat: hargnemine ja otsustuste tegemine, tsüklid ja mängu kavandamine 3. raamat: tsüklid tsüklikes ja listid 4. raamat: funktsioonid, objektid ja ussimäng 5. raamat: graafika ja animatsioon	1. OSA. Algoritm. Programm 2. OSA. Muutujad. Andmetüübid 3. OSA. Valikulause 4. OSA. Sõned. Kilpkonnagraafika 5. OSA. Tsükkel 6. OSA. Regulaaravaldis 7. OSA. Funktsioonid 8. OSA.	1. video: sissejuhatus, Thonny tutvustus, esimesed koodiread, Turtle 2. video: muutujad, koodi kommenteerimine, matemaatilised tehted, andmetüübid 3. video: kasutajasisendid, sõne muutmine täisarvuks, süntaksiviga 4. video: kahoot 5. video: tõeväärtused, tingimuslauseid, võrratuste

	<p>2.5 Mitmemõõtmelised andmestruktuurid 3. Objektorienteeritud programmeerimine 3.1 Klassid ja isendid 3.2 Pärilus 4. Projektid 4.1 Mäng 4.2 Veebirakendus 4.3 Töölauarakendus</p>	<p>15. Moodulid 16. Objektid Pygame 1. Aken 2. Värvid ja koordinaadid 3. Joonistamine 4. Pildid 5. Tekst 6. Animeerimine 7. Juhtimine klaviatuuriga 8. Juhtimine hiirega 9. Heli 10. Õpetusi</p>	<p>6. raamat: veel mängudes kasulikke programmilõike ja heli</p>	<p>Andmevahetus. Suurem programminäide</p>	<p>kehtivuse ja võrdsuse operaatorid 6. video: debugger, loogilised operaatorid 7. video: while tsükkel, tehte lühivormi tutvustus, break käsk 8. video: PyGame, mänguaken, järjend 9. video: järjendisse kuulumine, for tsükkel, append käsk 10. video: funktsioon, argumendid 11. video: funktsiooni näide, internetist abi otsimine, return käsk 12. video: mängu esimese ja teise etapi loomine 13. video: mängu kolmanda etapi loomine, arvupindade kukkuma panemine mänguaknas 14. video: mängu neljanda etapi loomine 15. video: mängu viienda etapi loomine 16. video: kokkuvõttev tund</p>
--	--	--	--	--	--

Litsents

Mina, Annabel Maksimov,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Pythoni algkursuse õppematerjalid põhikoolile**, mille juhendaja on Tauno Palts, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
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.

Annabel Maksimov

14.05.2025