

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Juhan Tamm**

# **Pythonit tutvustav tund III kooliastmele**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Tauno Palts, PhD

Tartu 2022

## **Pythonit tutvustav tund III kooliastmele**

### **Lühikokkuvõte:**

Selle töö eesmärk on tuvastada, millised võimalused on sobilikud tekstipõhise programmeerimiskeele tutvustamiseks kolmandas kooliastmes. Selleks katsetati ühes Tartu põhikoolis, kui hästi kolmanda kooliastme õpilased suudavad Pythoni programmeerimise loogikast aru saada, kui suur on nende huvi programmeerimise vastu, ning ühtlasi, kui kiiresti suudavad õpilased uusi kontseptsioone omandada ning neid ise realiseerida.

### **Võtmesõnad:**

Programmeerimine, õpilased, põhikool

**CERCS:** P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

## **Introductory lesson of Python for Junior high school**

### **Abstract:**

The goal of this Bachelor's thesis is to determine which options are suitable for introducing a text-based programming language in junior high school. For that, an experiment was conducted in one of the schools in Tartu, Estonia, on how well the Junior high students manage to comprehend Python programming logic, how interested they are in programming and how quickly the students manage to obtain the necessary concepts and realize them.

### **Keywords:**

Programming, students, primary school.

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

## Sisukord

1. Sissejuhatus.....	5
2. Teoreetiline taust.....	6
2.1 Informaatika põhikoolis .....	6
2.2 Paarisprogrammeerimine .....	7
2.3 Programmeerimise õpetamine kolmandas kooliastmes .....	8
3. Metoodika .....	10
3.1 Valim.....	10
3.2 Protsess ja vahendid .....	11
4. Tulemused.....	13
4.1 Õpilaste hinnangud Pythoni materjalide raskusastmele.....	13
4.2 Õpilaste programmeerimisoskus .....	14
4.3 Õpilaste varasem kokkupuude programmeerimisega .....	18
4.4 Õpilaste huvi programmeerimise vastu.....	19
4.5 Tunnimaterjalid .....	21
5. Arutelu .....	23
5.1 Sobilik õppevahend III kooliastmele.....	23
5.2 Valitud Pythoni töölehe sobivus III kooliastmele.....	23
5.3 Varasema programmeerimiskogemuse mõju .....	24
6. Kokkuvõte.....	25
7. Viidatud kirjandus.....	26
Lisad.....	28
I. Tööleht .....	28
II. Küsimustik .....	31
III. Litsents .....	33



## 1. Sissejuhatus

Eestis pole informaatika kohustuslik aine, vaid on õppekavasse lisatud valikainena [1]. Paljudes koolides on siiski informaatikaõpe mingil kujul olemas, kuid enamasti piirdub see tegelikult arvutikasutusõpetusega ning programmeerimine jääb tagaplaanile [2]. Üldiselt on põhikoolis kõige populaarsemaks programmeerimiskeeleks visuaalne programmeerimiskeel Scratch<sup>1</sup> [3], mis võib aga kolmandale kooliastmele juba liiga lihtne olla.

Selle töö eesmärk on välja selgitada, kas ja kui sobiv on tekstipõhine programmeerimine Pythoni<sup>2</sup> töölehe kujul kolmandas õppeastmes ning kuidas mõjutab õppetöö tulemusi ja õigete lahendusteni jõudmist varasem kokkupuude. Selleks, et saada informatsiooni tunni õnnestumise kohta, on oluline küsida õpilastelt tagasisidet. Eesmärgist tulenevalt püstitati kolm peamist uurimisküsimust:

1. Milline tööleht sobiks Pythoni õpetamiseks kolmandale kooliastmele?
2. Kuidas valitud Pythoni tööleht sobib kolmanda kooliastme klassidele?
3. Kuidas mõjutab varasem kokkupuude programmeerimisega edukust ülesannete lahendamisel?

Eesmärgini jõudmiseks valiti välja Pythoni tööleht, mis kasutab moodulit Turtle<sup>3</sup>, koostati ankeetküsitlus ning viidi läbi tunnid ühes Tartu põhikoolis.

Selles töös keskendutakse kõigepealt teoreetilisele taustale: käsitletakse varasemaid uuringuid informaatikaõppe üldisest olukorrast põhikoolis, paarisprogrammeerimise mõjust õppetöö tulemustele ning konkreetselt kolmanda õppeastme eripäradest. Seejärel tutvustatakse töö läbiviimiseks kasutatud metoodikat ja tulemusi ning viimasena arutletakse saadud tulemuste üle ning võrreldakse neid varasemate uuringute tulemustega.

---

<sup>1</sup> <https://scratch.mit.edu/>

<sup>2</sup> <https://www.python.org/>

<sup>3</sup> <https://docs.python.org/3/library/turtle.html>

## 2. Teoreetiline taust

Selles peatükis kirjeldatakse varasemaid kooli programmeerimisõppe teemalisi uuringuid. Uuringutes käsitletakse lisaks õpilaste vaatenurgale ka õpetajate oma.

### 2.1 Informaatika põhikoolis

Eesti tööturul on infotehnoloogia (IT) sektoris suur tööjõupuudus [4–6] ja vaja on inimesi, kellel on olemas lisaks tehnoloogia kasutamise oskustele ka selle loomise oskused.

Informaatika õpetamine põhikoolis on kooliti väga erinev. Järgnev lõik põhineb K. Kori raportil „IT-oskuste arendamine Eesti koolides“ [7], kus uuriti, kui palju ning millisel tasemel Eesti põhikoolides ja gümnaasiumites infotehnoloogiaga tegeletakse. Raportis kirjeldatakse, et riikliku õppekava järgi on gümnaasiumites IT-valdkonna temaatikat käsitlevaid kursuseid vähe ja needki on valikained. Põhikoolis on probleem veelgi suurem, sest riikliku õppekava järgi pole seal ühtegi IT-valdkonda kuuluvat ainet ette nähtud. Sellegipoolest mitmetes põhikoolides temaatikat ikkagi käsitletakse, kuid väga pealiskaudselt ja lühiajaliselt. Raportis kirjeldatud uuringu tulemused näitasid, et õpilaste arvates on suurem osa informaatikatundidest teksti- ja tabelitöötlusele suunatud, aga õpetajate arvates on suurem osa suunatud programmeerimisele. Enamik õpilastest olid kõige rohkem huvitatud programmeerimise õppimisest. Programmeerimisega oli kokku puutunud 56% nii 9. kui ka 12. klassi õpilastest, aga see programmeerimise kogemus oli varieeruv. Peaaegu pooled õpilased väitsid, et ei oska sisuliselt midagi programmeerida, kuid oli ka neid, kes väitsid, et suudavad iseseisvalt keerukaid programme luua. Programmeerimises pidasid ennast oskuslikemaks poisid. Põhilisteks huvitekitajateks IT ja programmeerimise vastu osutusid lapsepõlves mängitud arvutimängud ja suguvõsa mõjutused.

Arnold Veltmanni bakalaureusetöös [8] uuriti, kui palju Tartu koolides teises ja kolmandas õppeastmes informaatikaga kokku puututakse ning, kas informaatika võiks olla kohustuslik õppeaine riiklikus õppekavas [1]. 35 kooli 49-st (71,4%) on toonud oma õppekavasse informaatikaalase õppeaine kohustuslikuna vähemalt ühe klassi jaoks teises või kolmandas õppeastmes. Intervjuudest tuli välja, et kõik õpetajad peavad oluliseks informaatika olemasolu õppekavas, kuid kõige suuremaks probleemiks on sobivate materjalide puudus. Järgnevalt kirjeldatakse erinevaid informaatika õpetamise meetodeid.

Informaatika õpetamise meetodeid koolis on uuritud erinevatel tasemetel. Parham-Mocello uuris, kuidas annavad õpetajad edasi kuuendale ja seitsmendale klassile mõeldud arvutiteaduse õppekava teemasid [9]. Leiti, et kummagi klassi õppekavad on küll erinevad, aga sellegipoolest saab teha õpetaja õpetamisstiili põhjal järeldusi. Kuuenda klassi õpetaja kasutas rohkem mängulisust kui seitsmenda klassi õpetaja, mistõttu said kuuenda klassi õpilased materjalist paremini aru. Vartiainen [10] kirjeldab erinevalt Parham-Mocellost spetsiifilisemalt masinõppe valdkonda, ja kuidas ka seda peaks põhikoolis käsitlema lisaks tavapärasele käsupõhiste programmide loomisele. Tulemuseni jõudmiseks laskis Vartiainen õpilastel alguses pakkuda oma ideid sellest, kuidas võiks arvutit üldse õpetada. Pärast, kui nad olid teadmisi juurde saanud, küsiti neilt uuesti. Uurimuse käigus pidid õpilased töötama tiimides: nad pidid arutlema masinõppe kasutuse üle igapäevaelus ning siis tutvustati neile teoreetilist poolt ja lasti neil ka ise interaktiivselt masinõppe probleeme lahendada. Jõuti järelduseni, et selline õppevorm arendab õpilaste uuendusmeelsust. Kahestes tiimides töötamist kirjeldab järgnev paarisprogrammeerimise meetodit tutvustav peatükk.

## **2.2 Paarisprogrammeerimine**

Paarisprogrammeerimine on programmeerimise liik, kus partnerid vahetavad iga natukese aja tagant rolle nii, et üks programmeerib ning teine jälgib ja edastab juhiseid. Selles lõigus kirjeldatakse J. Denner artiklit [11], keskendudes küsimusele, kuidas mõjutab paarisprogrammeerimine õpilaste võimet programmeerimisel esinevate probleemidega toime tulla. Seejuures uuritakse nii negatiivseid kui ka positiivseid mõjusid. Varasemate ulatuslike uuringute põhjal on täheldatud paarisprogrammeerimise positiivset mõju õpilaste mõtlemisvõime arenemisel [12]. Uuringute põhjal on selle peamiseks põhjuseks asjaolu, et paarisprogrammeerides saavad õpilased oma mõttekäike lahti seletada ning teise inimesega arutada. Mõned teised uuringud on aga viidanud sellele, et ei ole võimalik kindlalt väita, et paarisprogrammeerimisel on sellised positiivsed omadused. Kohati esineb ka hoopis negatiivseid mõjusid - näiteks juhtudel, kui õpilased ei saa omavahel hästi läbi või pole nende eneseväljendusoskus piisavalt hea [13]. Denneri uuringu tulemused kinnitasid varasemate uurimuste tulemusi: visuaalse programmeerimiskeele Alice abil mängude loomisel avaldavad mõned paarisprogrammeerimise aspektid positiivset ning mõned negatiivset mõju.

Positiivset mõju avaldasid:

1. aktiivne infovahetus mõlemalt osapoolelt,
2. sage rollide vahetus ja

3. jälgija rollis oleva partneri julgustav kehakeel.

Negatiivset mõju avaldasid:

1. võimujanu,
2. jälgija rollis oleva paarilise pidev vahele segamine ja
3. jälgivas rollis oleva paarilise soovide ignoreerimine.

J. Denner on kirjutanud sarnase artikli ka 2019. aastal [14], mis keskendub küsimusele, kuidas moodustatud õpilaste paarid suhtlevad ning millest nad omavahel räägivad. See uuring viidi läbi vabatahtliku programmina koolitundide välisel ajal. Kokku osales uuringus 66 kuuenda kuni kaheksanda klassi õpilast, ehk moodustati 33 paari. Uurimusest selgus, et keskmiselt rääkisid moodustatud paarid omavahel programmeerimisest ainult 47% ajast ning keskmiselt 28% ajast paarid lihtsalt ei suhelnudki omavahel. Selgus ka, et suurim mõjutav tegur on moodustatud paari sõpruse tugevus: õpilastel paluti anonüümselt hinnata, kas nad peavad oma paarilist enda sõbraks või mitte ning leiti selge korrelatsioon – kõrgema sõprusehinnanguga paarid suhtlesid omavahel märgatavalt rohkem. Lisaks programmeerimise õpetamise meetoditele on oluline ka programmeerimise aine sisu.

### **2.3 Programmeerimise õpetamine kolmandas kooliastmes**

Üks olulisi erinevusi informaatika puhul, võrreldes teiste õppeainetega, on riiklikult reguleeritud ühtsete õppematerjalide puudus [2]. N. Humble kirjeldab erinevaid programmeerimise õpetamiseks kasutusel olevaid materjale ja meetodeid esimesest kuni kahesteistkümnenda klassini [15]. Ta nendib, et üldiselt on koolides kasutusel nelja liiki lähenemised:

1. tekstipõhised ehk mingit tekstiredaktorit kasutavad,
2. plokipõhised ehk erinevaid käsuplokke sobivasse järjekorda seadvad,
3. päriselulised ehk arvutit mitte kasutavad,
4. füüsilised ehk füüsilisi objekte nagu roboteid või masinaid kasutavad.

Humble analüüsis 26 erinevat õppematerjali: ta valis need andmebaasist välja esialgu lihtsalt märksõnu kasutades ning siis filtreeris spetsiifilisemate meetodite abil üleliigsed välja. Enim kasutatav õppemeetod nendes materjalides on füüsiline programmeerimine, millele järgnevad populaarsuselt teksti- ja plokipõhine programmeerimine ning kõige vähem esindatud on päriseluline programmeerimine. Eesti kolmandas õppeastmes on uuringu järgi kõige populaarsem plokipõhine programmeerimine, millele järgnesid võrdse populaarsusega

tekstipõhine ja päriseluline programmeerimine ning kõige ebapopulaarsemaks osutus füüsiline programmeerimine.

Õppematerjalide kvaliteet ja sisu on loomulikult väga olulised, kuid tähtis on ka see, kuidas ja mis vahenditega õpilastele teadmisi edasi antakse. Seda on kirjeldanud Peter Jacob Rich [3], kelle käsitlusel see lõik tugineb. Uuringus leiti, et kõige levinum programmeerimiskeel põhikoolis programmeerimise õpetamiseks on Scratch ning enamasti on programmeerimine mingil moel integreeritud matemaatikaga. Mõned teised rohkem levinud keeled olid Blockly<sup>4</sup>, Python ja JavaScript<sup>5</sup>. Kolmanda kooliastme puhul oli tekstipõhiste programmeerimiskeelte osakaal mõistagi suurem kui esimeses või teises kooliastmes, kuid siiski oli ka seal kõige populaarsem Scratch. Artikli teises pooles keskenduti õpetajate ettevalmistusele ja varasemale õpetamiskogemusele. Ainult 45% õpetajatest olid läbinud mingit sorti informaatikaalase treeningu enne, kui nad otsustasid programmeerimist õpetada.

Programmeerimiskeele Python kasutamise kohta kolmandas õppeastmes on uuringuid väga vähe, kuid siiski on võimalik midagi leida. Üks sellistest on Kazunari Ito konverentsil põhinev artikkel [16]. Artiklis kirjeldab Ito oma tiimi poolt loodud Pythoni-põhist keskkonda Picthon<sup>6</sup>, millega saab luua piktogramme ning analüüsib keskkonna kasutamise katsetulemusi, mis viidi läbi 131 kolmanda kooliastme õpilase hulgas. Picthon, nagu ka Scratch loodi spetsiaalselt programmeerimise õppimise ja õpetamise eesmärgil ning seetõttu ei sisalda kogu Pythoni funktsionaalsust. Õpilastega viidi läbi seitse Picthoni erinevaid võimalusi ja seejuures programmeerimise erinevaid olemusi tutvustavat tundi ning küsiti neilt tagasisidet. Üldiselt pidasid õpilased sellist õppevormi ja Picthonit lõbusaks ning ei tundnud, et teemad oleksid liiga keerulised. Õpilaste jaoks olid kõige raskemad kontseptsioonid muutujad, järjendid ja koordinaadipõhine graafika. Kokkuvõtvalt saab öelda, et selline graafikat kasutav programmeerimise õppevorm on kolmandasse õppeastmesse sobiv.

Selles töös õpetatakse ja uuritakse tekstipõhist programmeerimist Pythonis, kasutades graafika moodulit Turtle, ühe tunni raames erinevatele III kooliastme klassidele.

---

<sup>4</sup> <https://developers.google.com/blockly>

<sup>5</sup> <https://www.javascript.com/>

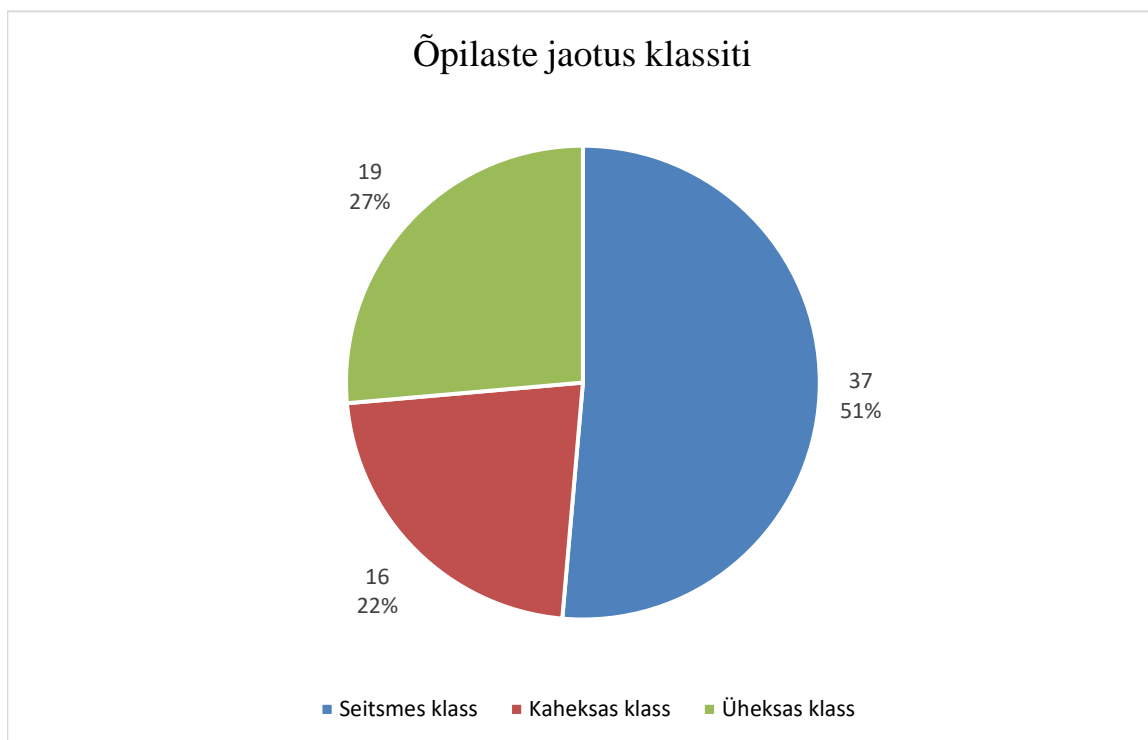
<sup>6</sup> [https://pictogramming.org/?page\\_id=2068](https://pictogramming.org/?page_id=2068)

### 3. Metoodika

Selles peatükis kirjeldatakse uurimistöö läbiviimiseks kasutatud valimit, protsessi, vahendeid ja tulemusi.

#### 3.1 Valim

Uurimuses kasutati mugavusvalimit (valimit, mis koostatakse sellistest uuritavatest, kes on lihtsasti kättesaadavad), kus ühe Tartu põhikooli kolmanda kooliastme õpilastele viidi läbi programmeerimiskeelt Python tutvustav tund. Valim koosnes kahest seitsmendast, ühest kaheksandast ja ühest üheksandast klassist ning selle kogusuuruseks oli 72 õpilast, kellest 37 (51%) olid seitsmenda klassi õpilased, 16 (22%) olid kaheksanda klassi õpilased ning 19 (27%) olid üheksanda klassi õpilased (vt joonis 1).



Joonis 1. Õpilaste jaotus klassiti

Uurimuses osalevad klassid valiti nende hulgast, mis olid ajaliselt saadaval nii, et esindatud oleks vähemalt üks igast kolmanda kooliastme klassist. Üks seitsmendatest klassidest oli muusikaklass ning kõik ülejäänud olid reaalklassid.

## 3.2 Protsess ja vahendid

Õpilaste programmeerimisoskuse hindamiseks viis töö autor iga osalenud klassiga läbi ühe 45-minutilise tunni, kus tehti umbes viie minuti pikkune sissejuhatav esitlus, mille käigus tutvustati õpilastele programmeerimise olemust, selle põhimõisteid ja tuntumaid programmeerimiskeeli. Järgmisena pidid õpilased paaristöö käigus lahendama programmeerimisülesandeid töölehel (vt lisa alapeatükki I), milleks oli aega ~35 minutit ning seejärel täitma ankeetküsimustiku, mille täitmiseks anti aega umbes viis minutit. Programmeerimisülesanded koostas varasemalt selle töö juhendaja ning neid muudeti vastavalt vajadusele. Kõik ülesanded tuli lahendada programmeerimiskeeles Python, kasutades teeki Turtle ning seda tegid õpilased veebipõhises keskkonnas trinket.io<sup>7</sup>, et ei oleks vaja kooliarvutitesse Pythonit ja arenduskeskkonda paigaldada. Esimeses ülesandes tuli õpilastel teeki kasutades joonistada ruut, kolmnurk, maja ja 2 maja. Teises ülesandes tutvustati ka tsükleid ning tuli joonistada kuusnurk ja ring. Töölehel oli ka lisäülesanne, milles tuli joonistada kuusepuu. Ülesannete täitmise ajal võisid õpilased küsida küsimusi ning vajadusel seletati neile vajalikku loogikat uuesti või anti vihjeid, aga kellelegi ei öeldud ühtegi vastust ette. Ankeetküsimustiku koostas töö autor võttes arvesse juhendaja soovitusi ja piiranguid, et maksimeerida küsimustikust väljaloetavat infohulka.

Küsimustik koosnes kahest plokkist – esimene plokk koosnes 11 küsimusest ning teine plokk koosnes kahest küsimusest (vt lisa alapeatükki II). Esimeses plokkis paluti õpilastel sisestada oma nimi, et saada ülevaade kohalolijatest ning sellest, mitmendas klassis nad käivad. Seejärel uuriti, kuidas õpilased ülesannetega toime tulid, lastes neil hinnata ülesannete raskusastmeid seitsmepallisel skaalal, kus üks tähendab, et ülesanne oli väga lihtne, ja seitse tähendab, et ülesanne oli väga raske. Selleks, et uurida, milline on õpilaste varasem kogemus programmeerimisega, küsiti neilt, kas nad on varem programmeerimisega kokku puutunud ning juhul kui on, siis kuidas. Sellele järgnes tunnist õpitud teadmisi kontrolliv küsimus: ette oli antud kuvatõmmis lühikesest koodijupist, mis joonistab risküliku, ning õpilased pidid ainult koodi põhjal välja mõtlema, mis kujundi see joonistab. Ploki lõpus küsiti, kui suur oli õpilaste huvi programmeerimise vastu enne tundi ja kui suur pärast tundi, mõlemal juhul taaskord seitsmepallisel skaalal, kus üks tähistas huvi puudumist ja seitse tähistas väga suurt huvi. Lisaks paluti anda tagasisidet ja soovitusi läbiviidud tunni kohta. Teises plokkis küsiti

---

<sup>7</sup> <https://trinket.io/>

õpilastelt, kas nad oleksid huvitatud programmeerimisringis osalemisest ning anti võimalus jätta oma elektronposti aadress, et saaks neile tulevikus huviringi kohta infot saata.

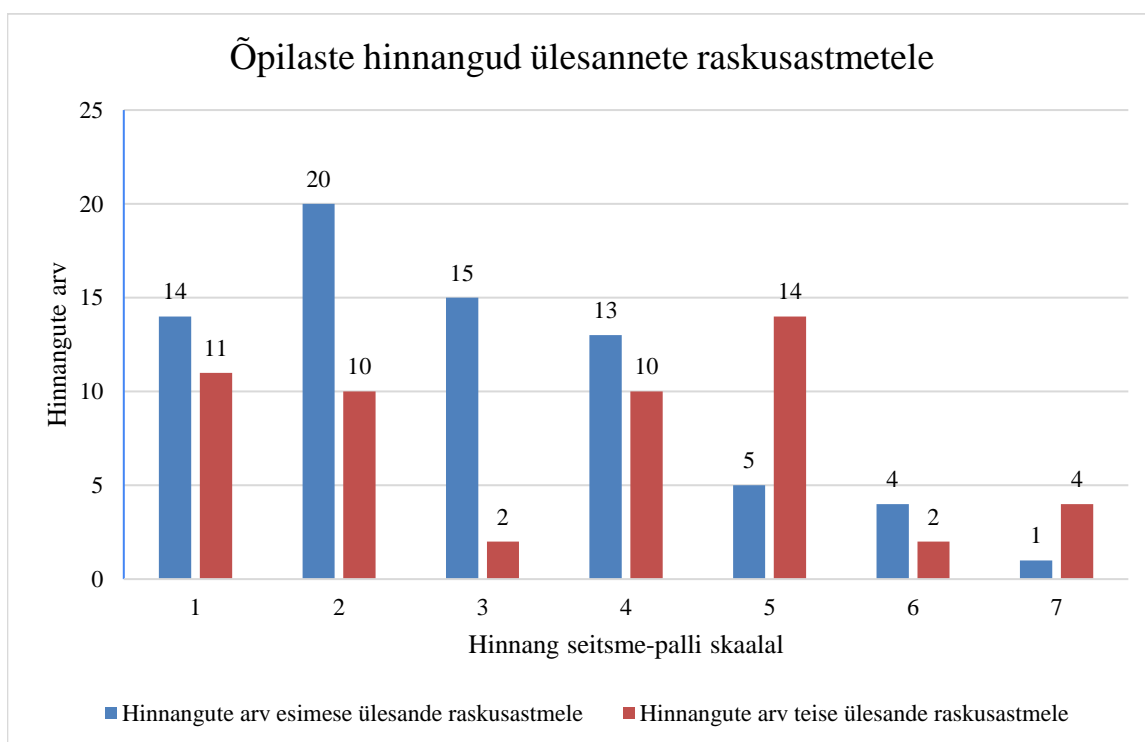
Saadud andmete töötlemiseks ja analüüsimiseks kasutati kirjeldavat statistikat keskkonnas Excel. Kõigepealt eemaldati andmete hulgast sobimatud andmed ehk need, kus õpilane oli meelega üritanud uurimust segada, näiteks kirjutanud vastuselahtritesse roppusi. Seejärel andmed sorteeriti, et graafiku tegemine oleks efektiivsem ja mugavam. Lisaks tuli valmis jõutud kujundite andmetest koostada eraldi tabel, mille baasil graafik luua, sest need andmeväljad sisaldasid rohkem kui ühte vastust.

## 4. Tulemused

Selles peatükis kirjeldatakse uurimuse tulemusi, sealhulgas õpilaste endi arvamusi temaatika mõistmisest, õpilaste võimekust, varasemat kogemust ja üldist huvi programmeerimise vastu. Kõiki tulemusi käsitletakse nii kolmanda õppeastme kohta tervikuna kui ka iga klassi kohta eraldi.

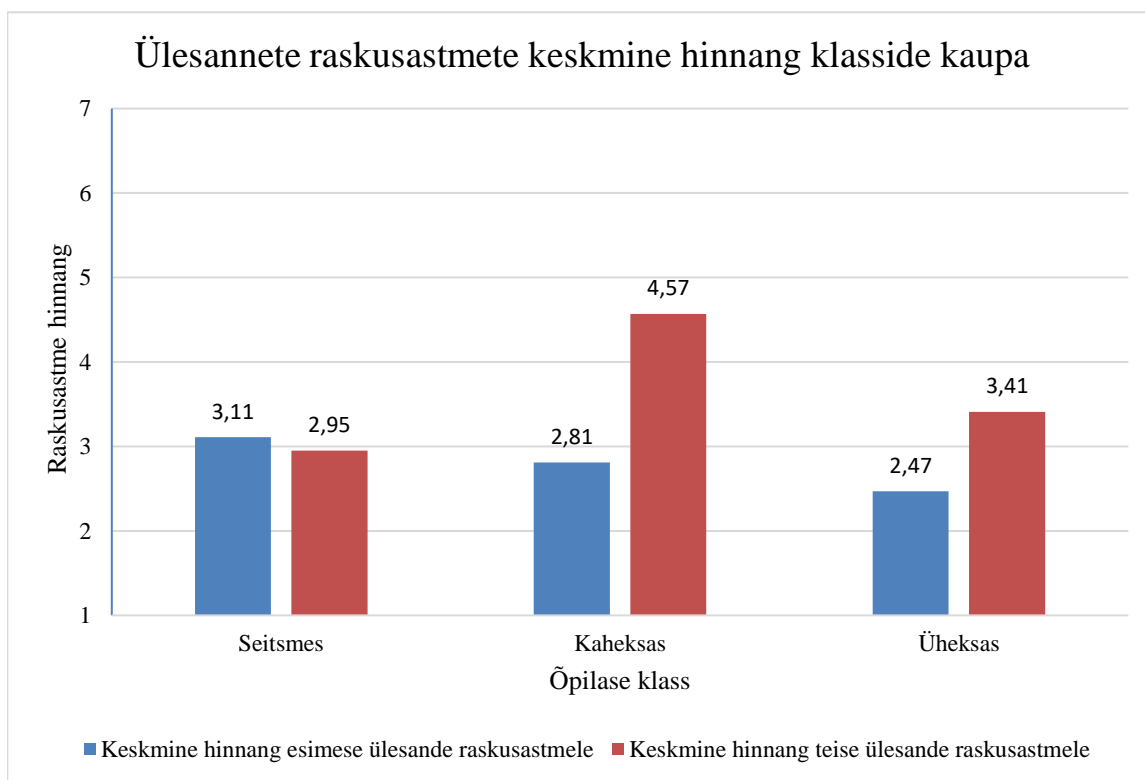
### 4.1 Õpilaste hinnangud Pythoni materjalide raskusastmele

Suurem osa õpilastest pidas esimest ülesannet küllaltki lihtsaks, hinnates raskusastmeks 1–3 palli, kuid olid ka mõned, kes pidasid ülesannet päris raskeks, hinnates raskusastmeks 6–7 palli. Teise ülesande puhul jaotusid raskusastme hinnangud üsna sarnaselt, aga arvestada tuleb sellega, et kõik õpilased ei jõudnud teise ülesandeni (vt joonis 2). Joonisel tähistab üks hinnangut väga lihtne ja seitse hinnangut väga raske.



Joonis 2. Õpilaste hinnang ülesannete raskusastmetele

Vaadeldes andmeid klasside kaupa on näha selgeid erinevusi (vt joonis 3).

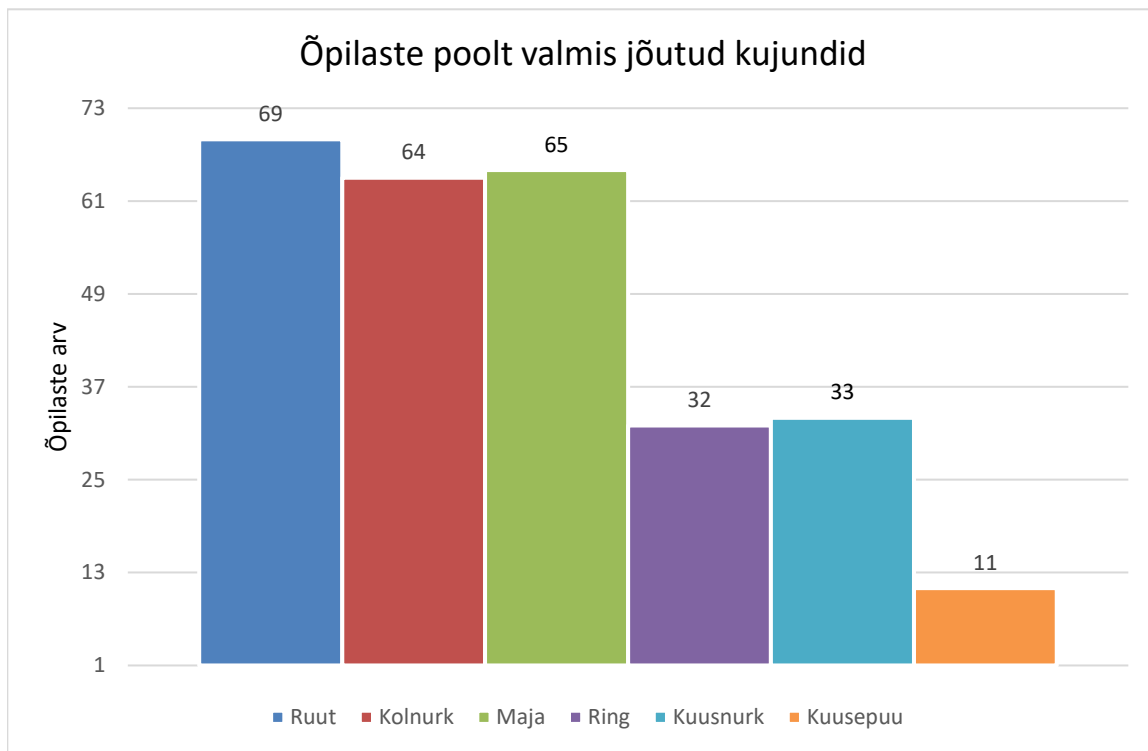


Joonis 3. Õpilaste keskmine hinnang ülesannete raskusastmetele klasside kaupa

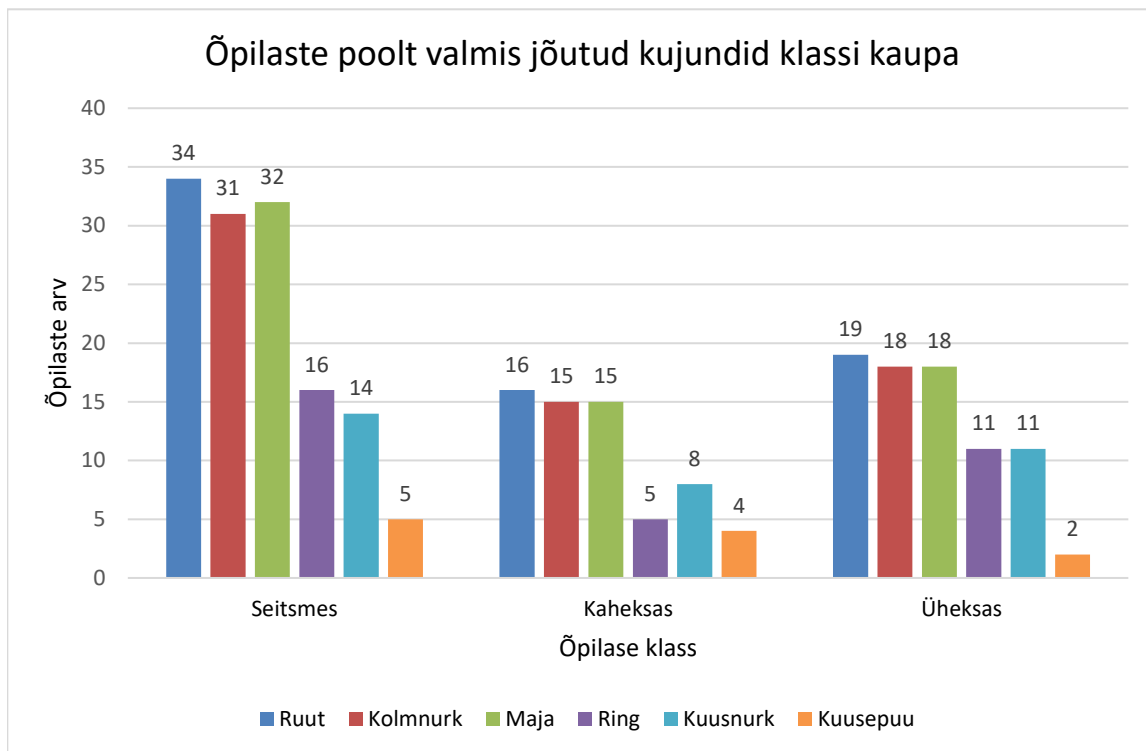
Seitsmenda klassi õpilaste keskmine hinnang esimese ülesande raskusastmele oli ~3,11 palli, aga teise ülesande raskusastmele ~2,95 palli. Kaheksanda klassi õpilaste keskmine hinnang esimese ülesande raskusastmele oli ~2,81 palli, aga teise ülesande raskusastmele ~4,57 palli. Üheksanda klassi õpilaste keskmine hinnang esimese ülesande raskusastmele oli ~2,47 palli, aga teise ülesande raskusastmele ~3,41 palli.

#### 4.2 Õpilaste programmeerimisoskus

Enamik õpilasi jõudis esimese ülesande täiesti valmis ning teise ülesandega vähemalt alustada (53 ehk ~73,6% õpilastest). Ruudu jõudsid valmis 69 (95,8%), kolmnurga 64 (88,9%), maja 65 (90,3%), ringi 32 (44,4%), kuusnurga 33 (45,8%), ning kuusepuu 11 (15,3%) õpilast (vt joonis 4). Vaadeldes andmeid klassikaupa, on näha, et kõik klassid tulid umbes sama hästi toime ruudu, kolmnurga ja maja joonistamisega ning ülejäänud kujundite joonistamisel esineb erinevusi (vt joonis 5).



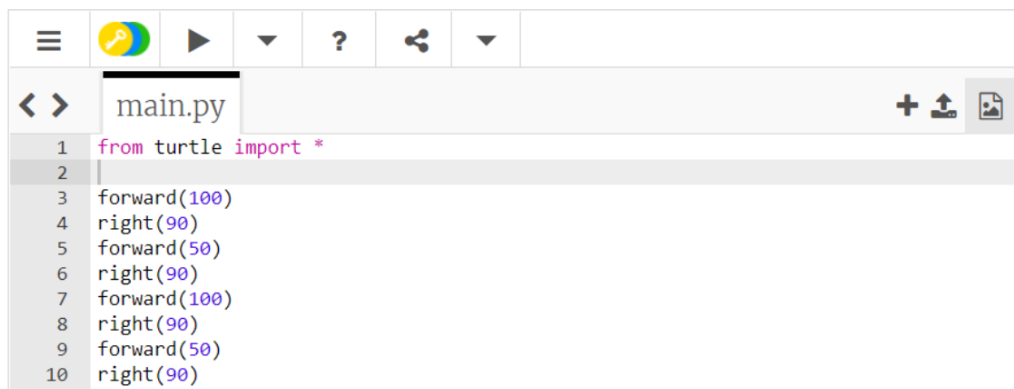
Joonis 4. Õpilaste poolt valmis jõutud kujundid



Joonis 5. Õpilaste arv iga valmis jõutud kujundi kohta ning jaotatult klassikaupa

Ankeetküsimustikus küsitud kontrollülesandega sai enamik õpilastest samuti hakkama, täpsemalt vastas küsimusele õigesti 39 õpilast (54%) ning õigele vastusele kõige sarnasemat vastusevarianti pakkus 21 õpilast (29%) (vt joonis 6).

8) Mis kujundi joonistab pildil näidatud programm? \*

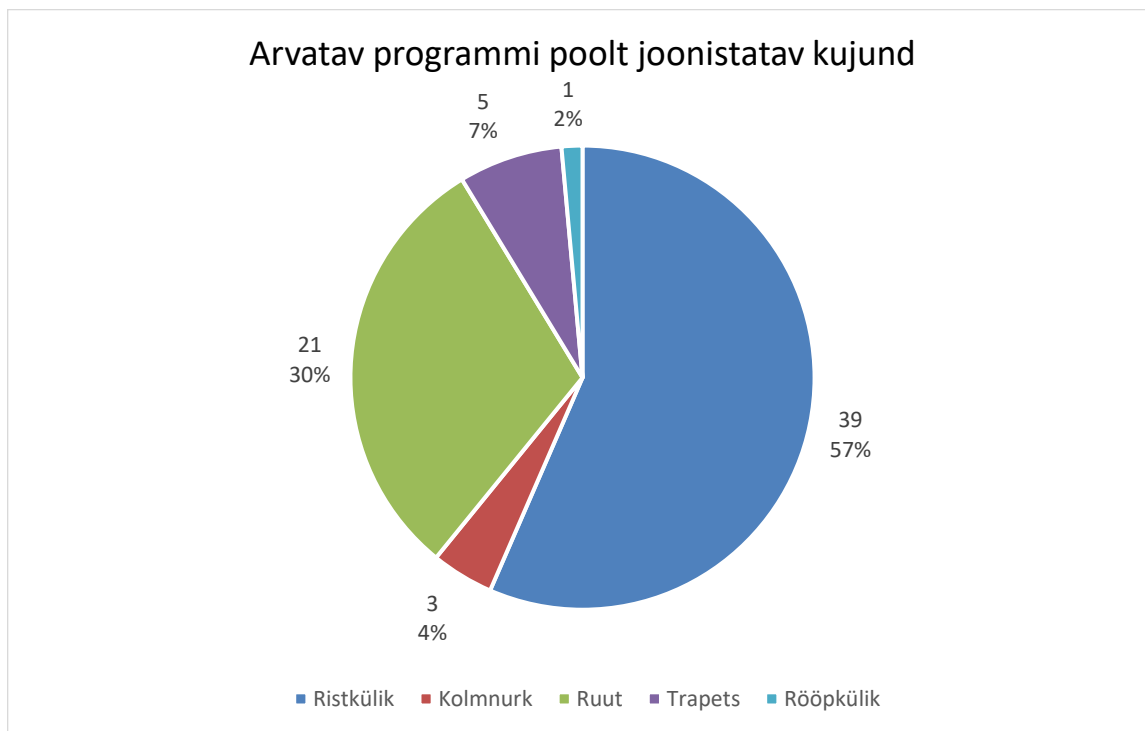


```
1 from turtle import *
2
3 forward(100)
4 right(90)
5 forward(50)
6 right(90)
7 forward(100)
8 right(90)
9 forward(50)
10 right(90)
```

- Ruudu
- Ringi
- Ristküliku
- Trapetsi
- Kolmnurga
- Muu...

Joonis 6. Küsimustiku küsimus programmi poolt joonistatava kujundi kohta

Ülejäänud nelja eeldefineeritud vastusevarianti pakkus 10 õpilast. Õpilastel oli ka võimalik valida vastusevariandiks „muu“ ning siis käsitsi oma pakkumine sisestada, kuid seda tegid ainult 2 õpilast ning mõlemal juhul oli vastus sobimatu (vt joonis 7).



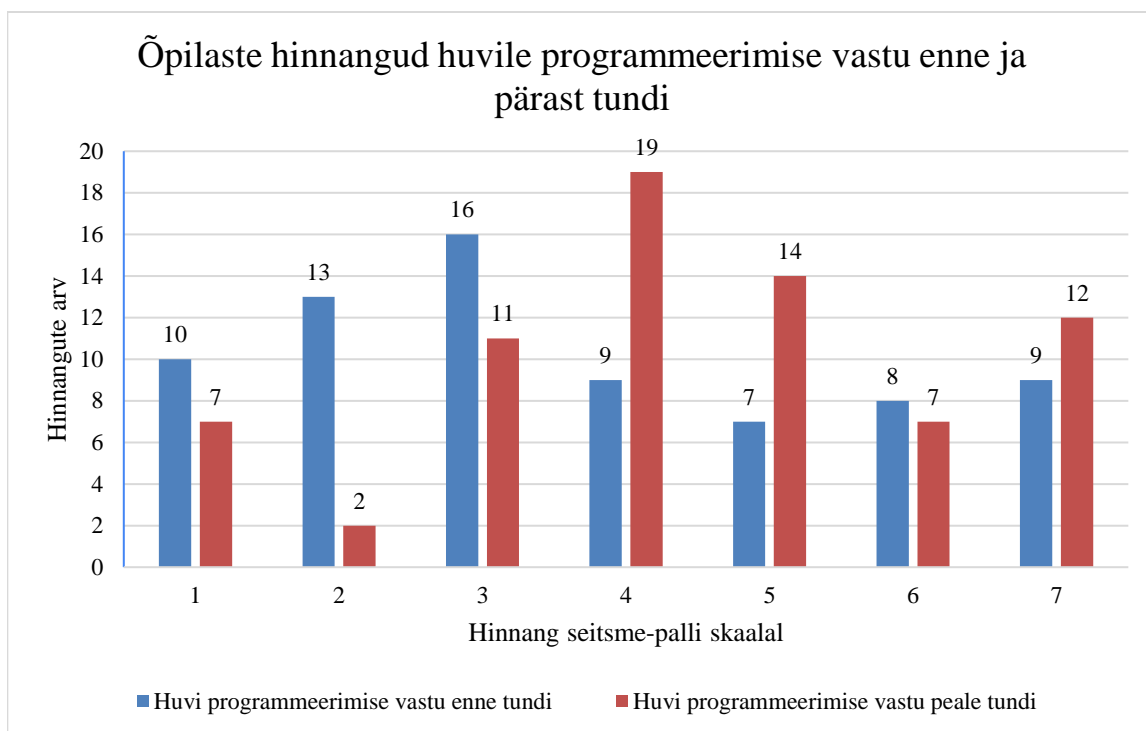
Joonis 7. Arvatav programmi poolt joonistatud kujund

### 4.3 Õpilaste varasem kokkupuude programmeerimisega

Ligikaudu pooled õpilastest (33 ehk ~45,8%) olid varem programmeerimisega tegelenud. Paljud olid viiendas klassis saanud katsetada programmeerimiskeelt Scratch, mõned osalevad aktiivselt programmeerimisringides, kuid oli ka neid, kes olid päris iseseisvalt internetist õpetusi otsinud ja nende abil programmeerinud. Tihti töötas ka vähemalt üks lapsevanematest infotehnoloogia valdkonnas ning oli oma teadmisi lapsele edasi andnud. Igas klassis leidis ka vähemalt üks õpilane, kes oskas nimetada erinevaid programmeerimiskeeli.

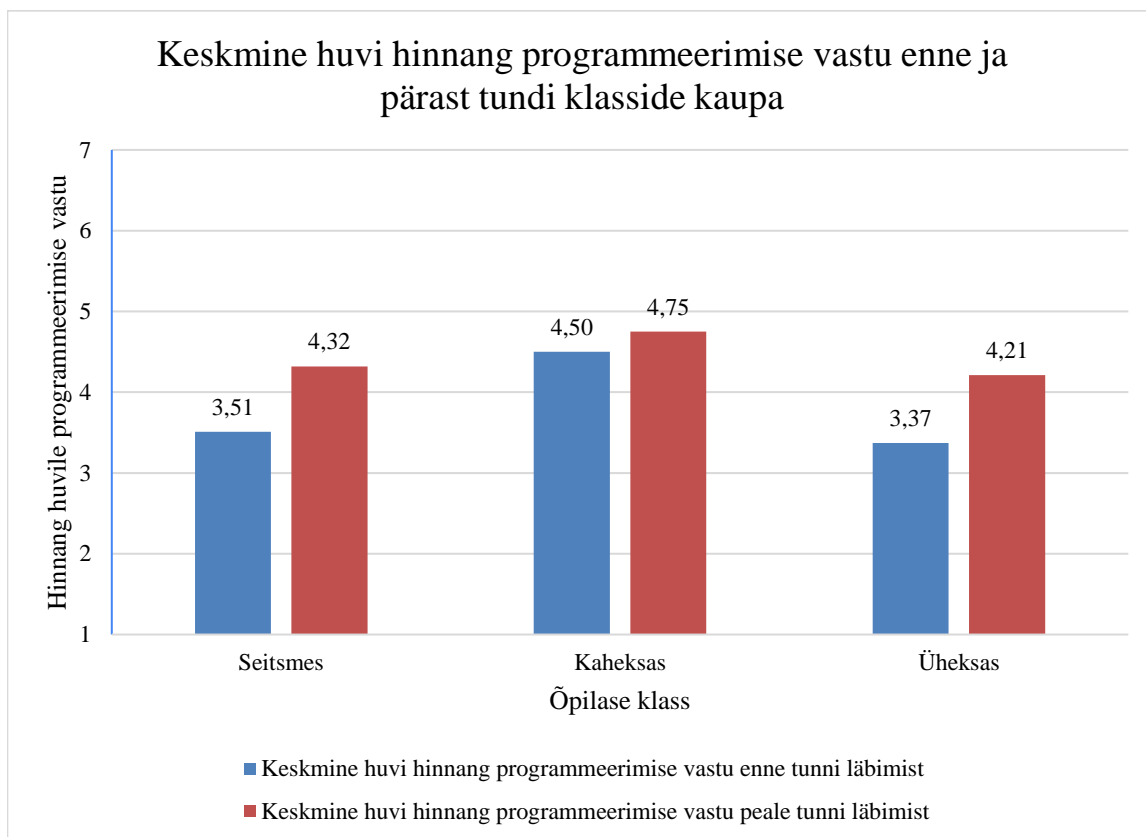
#### 4.4 Õpilaste huvi programmeerimise vastu

Üldiselt õpilaste huvi programmeerimise vastu suurenes pärast tunni läbimist, kuid oli ka neid, kes pärast tunni läbimist olid programmeerimisest vähem huvitatud kui enne. Enne tunni läbimist hindas 39 õpilast (54,2%) huvi suuruseks 1-3 palli, 9 õpilast (12,5%) 4 palli ning 24 õpilast (33,3%) 5-7 palli (vt joonis 8). Pärast tunni läbimist hindas 20 õpilast (27,8%) huvi suuruseks 1-3 palli, 19 õpilast (26,4%) 4 palli ning 33 õpilast (45,8%) 5-7 palli (vt joonis 8).



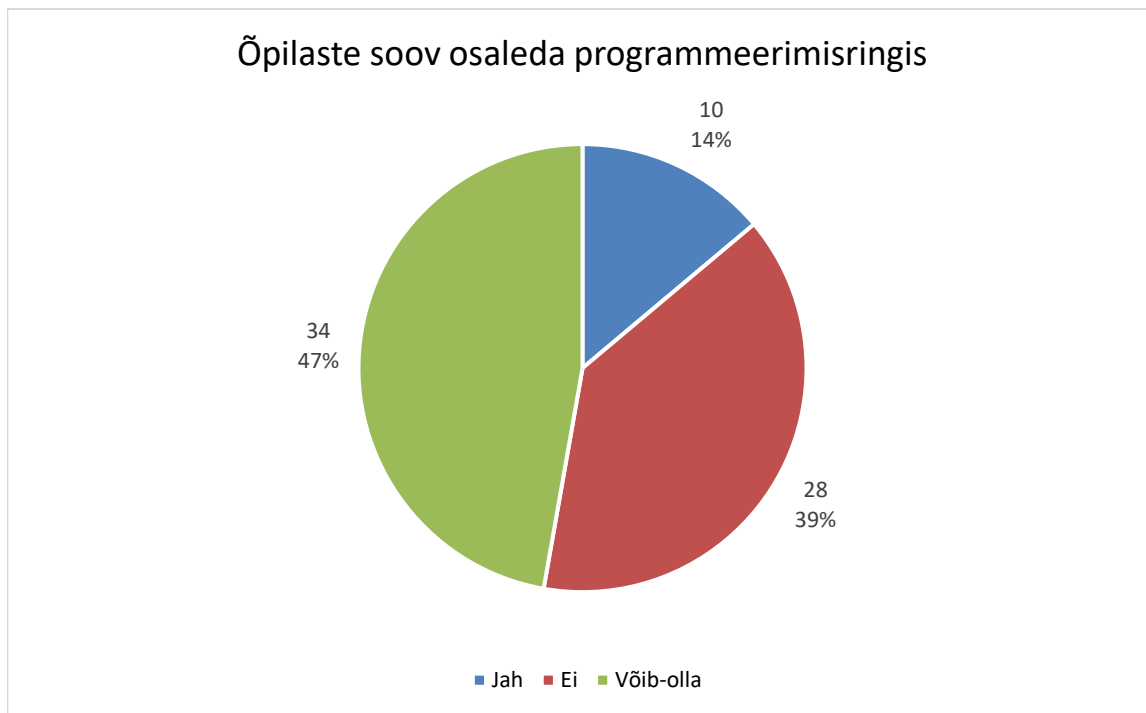
Joonis 8. Huvi programmeerimise vastu enne ja pärast tunni läbimist

Kõige suurem keskmine huvi programmeerimise vastu oli kaheksanda klassi õpilastel, nii enne kui ka pärast tundi. Kaheksandale klassile järgnes seitsmes klass ning kõige väiksem keskmine huvi programmeerimise vastu oli üheksanda klassi õpilastel. Kõige suurem keskmine huvi kasv oli aga sellegipoolest üheksandal klassil. Enne tundi oli nende keskmine huvi hinnang 3,37 ning pärast tundi 4,21 palli ehk kasv oli 24,93%. Huvi kasvult järgmisel kohal oli seitsmes klass, kelle keskmine huvi hinnang enne tundi oli 3,51 ja pärast tundi 4,32 palli ehk kasv oli 23,08%. Kõige vähem kasvas kaheksanda klassi keskmine huvi hinnang, mis oli enne tundi 4,50 ja pärast tundi 4,75 palli, ehk kasv oli 5,56% (vt joonis 9).



Joonis 9. Keskmine hinnang huvile programmeerimise vastu enne ja pärast tundi klasside kaupa

Kui õpilastelt küsiti, kas nad oleksid huvitatud programmeerimisringis osalemisest, siis enamik valis vastusevariandi „Võib-olla“ – täpsemalt 34 õpilast (47%). Populaarsuselt teine vastusevariant oli „Ei“, mille valis 28 õpilast (39%), ning vastusevariandi „Jah“ valis 10 õpilast (14%) (vt joonis 10).



Joonis 10. Õpilaste soov osaleda programmeerimisringis

Kasutatud materjalide ja saadud tulemuste põhjal loodi ka teemalehekülg.

#### 4.5 Tunnimaterjalid

Tartu Ülikooli arvutiteaduste instituudi kursuste keskkond<sup>8</sup> sisaldab enamiku Tartu Ülikooli IT-erialade kursuste lehekülgi, kuid leidub ka lehekülgi muude kursuste jaoks. Samasse keskkonda loodigi teemalehekülg „Programmeerimise tunnid“ (leitav aadressilt <https://courses.cs.ut.ee/t/progtunnid>) erinevate programmeerimise tundide materjalide jaoks, mida on võimalik põhikooli õpetajatel kasutada tundide läbiviimiseks. Lehekülje mõte pikemas perspektiivis on see, et seda saab tulevaste uurimuste käigus lihtsasti täiendada uute

---

<sup>8</sup> <https://courses.cs.ut.ee/>

materjalidega ning seeläbi luua avatud juurdepääsuga laiahaardeline keskkond kõigile huvilistele. Selle uurimuse materjalide jaoks loodi sinna alamlehekülg „Pythoni tund III kooliastmele“, kus on detailselt kirjeldatud tunnikava, tunnimaterjalid (näidisslaidid ja tööleht) ning statistika seniste tulemuste kohta.

## 5. Arutelu

Järgnevas peatükis analüüsitakse uurimisküsimusi, mis tõstatati töö algul ning arutletakse tulemuste üle.

### 5.1 Sobilik õppevahend III kooliastmele

Selleks, et õpilastel oleks programmeerimise tunnis huvitav ja tunni raskusaste oleks neile jõukohane, on oluline koostada vastavale kooliastmele sobilik tunnikava ja materjalid. Loomulikult loeb ka see, kuidas õpetaja õpilastega suhtleb ja temaatikat seletab, aga materjal – siinse uurimistöö puhul tööleht – peaks olema lihtsasti mõistetav ka ilma õpetaja abita. See ei tähenda, et ülesanded peaksid olema lihtsad, aga ülesanded peaksid olema püstitatud ja koostatud loogiliselt, et oleks selgelt aru saada, mida õpilaselt nõutakse.

Põhikooliõpilaste puhul on vajalik panna lisarõhku huvi tekitamisele – selleks on ilmselt üks lihtsamaid mooduseid teha materjalid mänguliseks. Arvestama peaks ka seda, et mida madalama kooliastmega on tegu, seda mängulisemad võiksid materjalid olla. Kolmanda õppeastme jaoks tuleks leida hea tasakaal mängulisuse ja raskusastme vahel, et õpilased ei peaks ülesandeid lapsikuteks ega liiga lihtsateks. Selle lõputöö tulemused kinnitavad, et mängulisus teeb õpilaste jaoks õppimise ja ka õpitava teema huvitavamaks, nagu tuli välja ka J. Parham-Mocello artiklist [9].

Selle töö jaoks valiti Pythoni tööleht, sest Python on algajatele väga sobiv programmeerimiskeel, mistõttu alustatakse ka ülikoolis informaatika õpet Pythoniga. Lisaks on Pythonil mitu ametlikku moodulit, mis on loodud eelkõige õppimise eesmärgil. Üks sellistest moodulitest ongi Turtle, mida selles töös kasutatakse, sest see kasutab graafikat.

### 5.2 Valitud Pythoni töölehe sobivus III kooliastmele

Valitud tööleht võiks olla kolmandale kooliastmele hea valik, sest õpilased saavad rakendada oma matemaatikas õpitud geomeetria teadmisi ning samas on pooleldi justkui tegemist kunsti tunniga. Selleks, et joonistada ekraanile kujundeid tuleb kasutada Turtle teegi käskude ja arvutada välja kujundite õiged nurgad, näiteks võrdkülgse kolmnurga puhul tuleb aru saada, et kõik nurgad on 60 kraadi. Ülesanded teevad kunstiliseks erinevad käsud – näiteks on võimalik muuta joonistatud kujundite värvi. Küsimustiku vastustest ja tulemustest ilmneb, et töölehe raskusaste tundub olevat kolmanda kooliastme õpilastele paras.

Loomulikult on töölehe sobivuse hindamine keeruline, sest enamikul õpilastest ei teki võrdlusmomenti mingite teiste töölehtedega ning nad ei oska ilmselt ka väga täpselt hinnata, kui hea või halb tööleht nende jaoks on. Lisaks on raske ennustada, kas õpilaste huvi oleks mõne teise töölehe puhul rohkem kasvanud või pigem just vähenenud. Selle teada saamiseks, tuleks sarnaseid tunde läbi viia ka teistsuguste töölehtedega ning siis tulemusi omavahel võrrelda. Sellegipoolest, vaadates küsimustiku vastuseid, on näha, et enamiku õpilaste huvi teema vastu tõusis ning ülesannetega jäid hätta ainult mõned üksikud õpilased.

### **5.3 Varasema programmeerimiskogemuse mõju**

Sellest uurimusest tuleb välja, et varasema programmeerimiskogemusega õpilased edenevad paremini kui need, kellel varasem kogemus puudub. Nad lahendavad ülesandeid kiiremini, aga pikas perspektiivis võib see mitte kehtida, sest varasem kogemus on ajutine eelis. Mõni õpilane võib olla väga tark, aga lihtsalt sellepärast, et ta ei oma varasemat programmeerimise kogemust, lahendada ülesandeid aeglasemalt. See tuleneb peamiselt kontseptsiooni uudsusest ning sellest, et nii programmeerimise üldise loogikaga kui ka kindla programmeerimiskeelega harjumine ja selle kasutama õppimine võtab aega. Selle töö valimis oli näiteks üks õpilane, kes polnud varem ise midagi programmeerinud, aga teadis arvutitest päris palju ning suutis tänu sellele ülesanded väga kiiresti lahendada.

Teisest küljest võib varasem kogemus osutada ka segavaks, sest võib tekkida liigne enesekindlus ning sellest omakorda tunne, et ülesanded on liiga lihtsad ja need võib vahele jätta. Tegelikult võivad ülesanded lahendama hakates osutada keerulisemaks, kui õpilane esialgu arvas.

## 6. Kokkuvõte

Uurimistöö eesmärk oli tuvastada, kui hästi sobib tekstipõhine programmeerimine Pythonis kolmandasse õppeastmesse. Eesmärgini jõudmiseks viidi ühes Tartu põhikoolis läbi uurimus, mille raames said õpilased õppida ja ise proovida programmeerimist Pythoni mooduliga Turtle. Katse jaoks koostatud valim koosnes 73 õpilasest ning seejuures kahest seitsmendast klassist, ühest kaheksandast klassist ja ühest üheksandast klassist. Selleks, et oleks võimalik tulemusi analüüsida, paluti õpilastel täita ankeetküsimustik, kust küsiti neilt tagasisidet.

Tulemuste ja varasemate uuringute põhjal võib öelda, et õpetamiseks kasutatav tööleht võiks olla mänguline ja kindlasti võiks olla kasutatud mingit visuaalset väljundit. See suurendab õpilaste huvi teema vastu ning teeb ülesannete lahendamise lõbusaks.

Selles uurimuses kasutatud tööleht tundub tulemuste põhjal olevat kolmandale kooliastmele üsna sobilik. Kõikide klasside puhul sai enamik õpilastest vähemalt esimese ülesande valmis, mis oligi oodatud tulemus, ning kõigi klasside puhul suurenes keskmiselt ka huvi programmeerimise vastu.

Õpilased, kes olid varem programmeerimisega kokku puutunud, said kindlasti ülesannetega kiiremini ja kergema vaevaga hakkama, kuid see ei tähenda automaatselt, et nad programmeerimises teistest tugevamad on. See näitab lihtsalt, et neil on teiste ees ajutine eelis.

Uurimistöö võib olla eelkõige kasulik põhikooli õpetajatele, sest selle abiga on neil võimalik kujundada ka ise programmeerimise tunde. Laiemas mõttes võiks aga töö olla ka inspiratsiooniks Eesti haridussüsteemi edasiarendamisel. Selle töö jätkuna oleks kasulik teha uurimusi ka teiste programmeerimiskeelte ja vahendite toel, et tekiks võrdlusmoment ja saaks luua hästi sobivaid materjale.

## 7. Viidatud kirjandus

- [1] Põhikooli riiklik õppekava – Riigi Teataja 2011. <https://www.riigiteataja.ee/akt/129082014020> (accessed March 10, 2022).
- [2] Mis saab Eesti IT haridusest? Kes tuleb õppima? Kes kuidas õpib? Kes langeb välja? Mida saab keegi teha? Raport 2015.
- [3] Rich PJ., Browning SF., Perkins MK., Shoop T., Yoshikawa E., and Belikov OM., Coding in K-8: International Trends in Teaching Elementary/Primary Computing. *TechTrends* 2019;63:311–29. <https://doi.org/10.1007/S11528-018-0295-4>.
- [4] Tulevikuvaade tööjõu- ja oskuste vajadusele: info- ja kommunikatsioonitehnoloogia valdkond – Tööjõuvajaduse seire- ja prognoosisüsteem OSKA 2021.
- [5] Eero Tohver: meil on Eestis vaja 18 000 IT-spetsialisti n.d. <https://www.accelerista.com/uudis/eesti/eero-tohver-meil-on-eestis-vaja-18-000-it-spetsialisti/> (accessed April 13, 2022).
- [6] Värske tööjõuvajaduse uuring näitab oskustöölise pouda – Töötukassa n.d. <https://www.tootukassa.ee/et/uudised/varske-toojouvajaduse-uuring-naitab-oskustooliste-pouda> (accessed April 13, 2022).
- [7] Kori K., Beldman P., Tõnisson E., Luik P., Suviste R., Siiman L., and Pedaste M., IT oskuste arendamine Eesti koolides. 2019.
- [8] Veltmann A., Informaatikaõpe II ja III kooliastmes Tartumaa koolide näitel Bakalaureusetöö (9 EAP). 2021.
- [9] Parham-Mocello J., Erwig M., and Niess M., Teaching CS Middle School Camps in a Virtual World. 2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) 2021:1–4. <https://doi.org/10.1109/VL/HCC51201.2021.9576271>.
- [10] Vartiainen H., Toivonen T., Jormanainen I., Kahila J., Tedre M., and Valtonen T., Machine learning for middle schoolers: Learning through data-driven design. *International Journal of Child-Computer Interaction* 2021;29. <https://doi.org/10.1016/J.IJCCI.2021.100281>.

- [11] Denner J., Green E., and Campe S., Learning to program in middle school: How pair programming helps and hinders intrepid exploration. *Learning Sciences* 2021;30:611–45. <https://doi.org/10.1080/10508406.2021.1939028>.
- [12] Hanks B., Fitzgerald S., McCauley R., Murphy L., and Zander C., Pair programming in education: a literature review. <https://doi.org/10.1080/089934082011579808> 2011;21:135–73. <https://doi.org/10.1080/08993408.2011.579808>.
- [13] Simon B., and Hanks B., First-year students' impressions of pair programming in CS1. *Journal on Educational Resources in Computing (JERIC)* 2008;7. <https://doi.org/10.1145/1316450.1316455>.
- [14] Campe S., Denner J., Green E., and Torres D., Pair programming in middle school: variations in interactions and behaviors. <https://doi.org/10.1080/0899340820191648119> 2019;30:22–46. <https://doi.org/10.1080/08993408.2019.1648119>.
- [15] Humble N., The use of Programming Tools in Teaching and Learning Material by K-12 Teachers 2021. <https://doi.org/10.34190/EEL.21.117>.
- [16] Ito K., Picthon: A Learning Environment of Python through Pictogram Content Creation. *Proceedings - Frontiers in Education Conference, FIE 2020;2020-October*. <https://doi.org/10.1109/FIE44824.2020.9273961>.

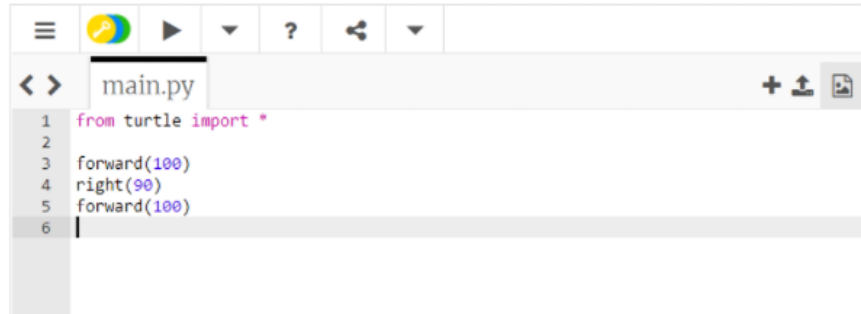
## **Lisad**

### **I. Tööleht**

## Programmeerimine keeles Python

### SISSEJUHATUS

1. Mine leheküljele <https://trinket.io/python/769dfb1238> ja kirjuta esimene testprogramm:



```
1 from turtle import *
2
3 forward(100)
4 right(90)
5 forward(100)
6
```

2. Käivita programm vajutades halli noolega nuppu

### VEEL KÄSKE

**forward(100)** – nool liigub 100 pikslit edasi

**backward(100)** – nool liigub 100 pikslit tagasi

**right(90)** – nool pöörab 90 kraadi paremale

**left(90)** – nool pöörab 90 kraadi vasakule

**up()** – nool enam ei joonista (pliiats tõstetakse üles)



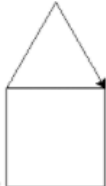
**down()** – nool hakkab jälle joonistama (pliiats lastakse alla)

**speed(1000)** – muudab joonistamise kiiremaks

# – trellide järel olevat koodi ei sooritata. Võib kasutada kommenteerimiseks.

Veel káske: <https://docs.python.org/3/library/turtle.html>

### ÜLESANNE 1:

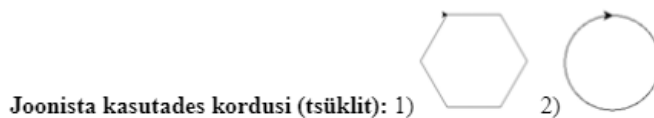
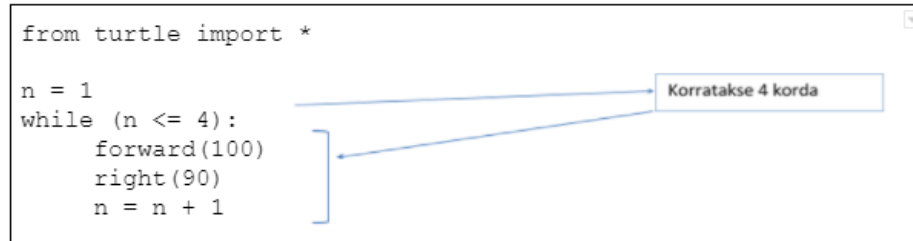
- Joonista: 1)  2)  3)  4) 2 Maja

## ÜLESANNE 2

Ilmselt sai eelmises ülesandes palju kasutatud *copy-paste* meetodit. Tegelikult on selliste olukordade lihtsustamiseks võimalik kasutada kordusi ehk **tsükleid**.

Joonistame näiteks ruudu, korrates käske “forward” ja “right” 4 korda:

```
from turtle import *  
  
n = 1  
while (n <= 4):  
    forward(100)  
    right(90)  
    n = n + 1
```



## JOONE VÄRVIMINE

Proovime nüüd teha programmi natukene lõbusamaks- lisame värve.

Pliiatsi värvi on võimalik muuta nii: `color("värvinimi")`.

Näiteks käsku `color("red")` saaks kasutada punase joone tegemiseks nii:

```
color("red")  
forward(100)
```

Värve saab veel selliselt: `color("#285078")`, kus värvikoodi saab guugeldades “*color codes*”.

## KUJUNDI SEEST VÄRVIMINE

Lisaks on võimalik kujundeid ka seest värvida.

Selleks, et ruut seest punaseks värvida, tuleks panna ruudu joonistamisel selle ette käsk `begin_fill()` ja pärast joonistamist käsk `end_fill()`.

## ISESEISEV ÜLESANNE

Joonista iseseisvalt Pythonis kuusepuu.

Lisa iseseisvalt pildile veel asju. Näiteks võiks pildil veel olla täht kuusepuu tipus, ümmargused jõuluehted, küünlad, kingid, päkapikud jne.

Täida küsitlus: <https://forms.gle/rEDE3F1iuk3DyDK8>

## II. Küsimustik

1) Kirjuta siia oma täisnimi \*

Pikk vastuse tekst

---

2) Mitmendas klassis sa käid? \*

Seitsmendas

Kaheksandas

Üheksandas

3) Millised kujundid valmis jõudsite? \*

Ruut

Kolmnurk

Maja

Ring

Kuusnurk

Kuusepuu

4) Hinda esimese ülesande (ruudu, kolmnurga ja maja joonistamine) raskusastet \*

	1	2	3	4	5	6	7	
Väga lihtne	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga raske

5) Hinda teise ülesande raskusastet (kui selleni jõudsid)

	1	2	3	4	5	6	7	
Väga lihtne	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga raske

6) Kas olid varem programmeerimisega kokku puutunud? \*

Jah

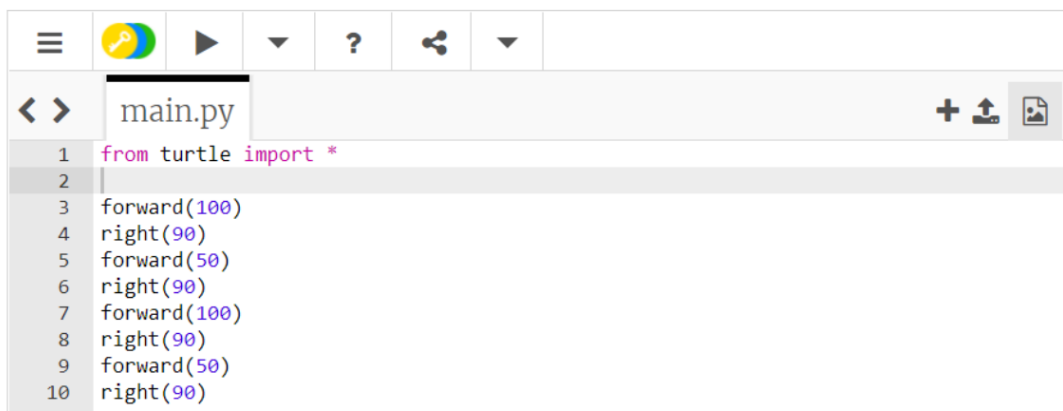
Ei

7) Kui vastasid eelmisele küsimusele jah, siis palun kirjuta siia, kuidas. (Näiteks, et õppisid iseseisvalt Youtube'i abil Python'it vm)

Pikk vastuse tekst

---

8) Mis kujundi joonistab pildil näidatud programm? \*



```
1 from turtle import *
2
3 forward(100)
4 right(90)
5 forward(50)
6 right(90)
7 forward(100)
8 right(90)
9 forward(50)
10 right(90)
```

Ruudu

Ringi

Ristküliku

Trapetsi

Kolmnurga

Muu...

9) Hinda oma huvi programmeerimise vastu enne selle tunni läbimist \*

Ei olnud üldse huvitatud    1    2    3    4    5    6    7    Olin väga huvitatud

10) Hinda oma huvi programmeerimise vastu nüüd \*

	1	2	3	4	5	6	7	
Ei ole üldse huvitatud	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Olen väga huvitatud

11) Kui on soovitusi kuidas sellist tundi paremini läbi viia siis ütle seda siin

Pikk vastuse tekst

---

Kas oleksid huvitatud programmeerimise ringis osalemisest? \*

- Jah
- Ei
- Võib-olla

Kui vastasid eelmisele küsimusele jaatavalt, siis võid siia jätta oma emaili aadressi, kuhu saadan kevade paiku huviringi kohta infot, juhul kui on piisavalt huvilisi

Lühike vastuse tekst

---

### III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Juhan Tamm

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „**Pythonit tutvustav tund kolmandale kooliastmele**“, mille juhendaja on Tauno Palts, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Juhan Tamm*

*10.05.2022*