

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Informaatika õppekava

Mattias Nurk

**Programmeerimise õpetamise meetodid ja
nende rakendamine gümnaasiumi valikkursusel**

Bakalaureusetöö (6 EAP)

Juhendaja: Eno Tõnisson, MSc

Tartu 2015

Programmeerimise õpetamise meetodid ja nende rakendamine gümnaasiumi valikkursusel

Lühikokkuvõte:

Selles töös antakse ülevaade kuuest programmeerimise õpetamise meetodist ja nendest kolme (paarisprogrammeerimine, mängupõhine programmeerimine, kommenteeeri-ja-kodeeri) rakendamisest gümnaasiumi valikkursusel. Samuti kirjeldatakse hetkeolukorda programmeerimise õpetamises Eestis, tutvustatakse käesoleva töö aluseks oleva kursuse läbiviimist ning analüüsitakse õpilaste arvamust rakendatud meetoditest ja programmeerimisest kui karjäärivalikust.

Võtmesõnad:

Programmeerimise õpetamine, õpetamise meetodid, paarisprogrammeerimine, mängupõhine programmeerimine, kommenteeeri-ja-kodeeri, gümnaasium, valikkursus

Programming Teaching Methods and Implementation in a Secondary School Elective Course

Abstract:

This paper gives an overview of six programming teaching methods and the implementation of three of those (Pair programming, Game-themed programming, Comment-first-coding) in a secondary school elective course. Also, the current state of programming teaching in Estonia and the conducting of the elective programming course are described. Lastly, the students' opinion on the implemented methods and programming as a career choice is analysed.

Keywords:

Programming teaching, teaching methods, pair programming, game-themed programming, comment-first-coding, secondary school, elective course

Sisukord

Sissejuhatus	4
1. Programmeerimise õpetamise meetodid	6
1.1 Paarisprogrammeerimine (Pair programming).....	6
1.2 Mängupõhine programmeerimine (Game-themed programming)	8
1.3 Kommenteeri-ja-kodeeri (Comment-first-coding)	8
1.4 Probleemipõhine õppimine (Problem-based learning).....	9
1.5 Puslepõhine õppimine (Puzzle-based learning).....	11
1.6 Kaaslase juhendamine (Peer instruction)	11
2. Programmeerimise valikkursuse läbiviimine	13
2.1 Programmeerimise õpetamise olukord Eestis	13
2.2 Programmeerimise õpetamise meetodite valik.....	14
2.3 Pythonis programmeerimise valikkursuse läbiviimine	15
3. Programmeerimise õpetamise meetodite tulemused	18
3.1 Paarisprogrammeerimise mõju programmeerimise mõistmisele	18
3.2 Kommenteeri-ja-kodeeri mõju programmeerimise mõistmisele.....	19
3.3 Mängupõhise programmeerimise mõju programmeerimise mõistmisele	21
3.4 Õpilaste arvamuse muutus programmeerimisest.....	22
3.5 Kursuse tulemused.....	23
4. Kokkuvõte	24
5. Tsiteeritud teosed	25
Lisad	27
I. Pythonis programmeerimise kursuse ainekava	27
II. Pythonis programmeerimise kursuse küsimustikud	28
III. Litsents	29

Sissejuhatus

Programmeerimist peetakse loogilise ja analüütilise mõtlemise suureks arendajaks, äärmiselt kasulikuks tulevikuoskuseks ning perspektiivikaks karjäärivalikuks [1]. Ometi on tööturul suur puudus haritud IT-spetsialistidest ja vaatamata headele tulevikuväljavaadetele ei õnnestunud näiteks Tartu Ülikoolil möödunud sügisel kõiki informaatika eriala õppekohti ära täita [2, 3]. 150 õppekohta pakkuvale informaatika õppekavale võeti vastu vaid 132 tudengit [4].

Ajaga kaasas käimiseks ja õpilaste huvi programmeerimise vastu suurendamiseks on mitmed Eesti üldhariduskoolid aina rohkem hakanud programmeerimist õpetama. Seni aga on Gümnaasiumi riiklikus õppekavas valikkursusena olemas vaid üks programmeerimise kursus: „Rakenduste loomise ja programmeerimise alused“ [5]. Nimetatud kursusel õpetatakse programmeerimise põhimõtteid programmeerimiskeele Scratch abil. Seejärel on võimalik spetsialiseeruda kas keelele Visual Basic for Applications või Python. Kuigi väga põhjalikud, on viimaste keelte õpetamismaterjalid üles ehitatud eelnevale Scratchi osale ning ei sobi seega eraldi kursusematerjaliks [6]. Gümnaasiumi riiklikus õppekavas leidub ka kursus „Mehhatroonika ja robotika“, kus läheb programmeerimist tarvis, kuid nimetatud kursus ei keskendu programmeerimise õpetamisele.

Seoses Scratchi mängulise kasutajaliidese ja vähese tööstusliku rakendusväärtusega soovivad mitmed koolid õpetada programmeerimist teiste keelte põhjal. Muude programmeerimiskeelte kursused aga peavad üldhariduskoolid õppematerjalide puuduse tõttu ise välja töötama. Paljudes koolides aga programmeerimise kursuseid üldse ei pakutagi, mis võib tuleneda puudulikust kursusematerjalide hulgast ja kvalifitseeritud õpetajate nappusest. Ülejäänud koolides on programmeerimise õpetamise tase ja meetodid kooliti väga erinevad. [7]

Käesoleva bakalaureusetöö eesmärgiks on aidata õpetajaid programmeerimise õpetamise meetodite valikul. Selle nimel annab käesolev töö ülevaate erinevates välismaa ülikoolides rakendatud ja positiivseid tulemusi andnud kuuest programmeerimise õpetamise meetodist ja analüüsib neist kolme rakendamist gümnaasiumi Pythonis programmeerimise valikkursusel, mida õpetas selle töö autor.

Esimeses peatükis kirjeldataksegi lähemalt vaatluse all olevat kuut programmeerimise õpetamise meetodit. Need valiti välja tingimusel, et neid on mõnes ülikoolis või üldhariduskoolis juba rakendatud ja nende meetodite tulemusi analüüsitud. Samuti pidid valitud meetodid

olema hõlpsasti rakendatavad Eesti üldhariduskoolides. Uurimise all olevateks programmeerimise õpetamise meetoditeks on:

- 1) paarisprogrammeerimine (Pair programming),
- 2) mängupõhine programmeerimine (Game-themed programming),
- 3) kommenteeri-ja-kodeeri (Comment-first-coding),
- 4) probleemipõhine õppimine (Problem-based learning),
- 5) puslepõhine õppimine (Puzzle-based learning),
- 6) kaaslase juhendamine (Peer instruction).

Teises peatükis kirjeldatakse üldist programmeerimise õpetamise olukorda Eesti koolides, bakalaureusetöö aluseks oleva programmeerimiskeele Python valikkursuse ülesehitust ja selgitatakse, mille põhjal valiti välja kolm programmeerimise õpetamise meetodit, mida kursuse jooksul kasutada:

- 1) paarisprogrammeerimine (Pair programming) – paarilised istuvad üldjuhul kõrvuti ühe arvuti taga ja lahendavad koos programmeerimisalast probleemi, üks paariline koodi kirjutades ning teine toimuvat jälgides ja järgmisi samme ette mõeldes [8,9];
- 2) mängupõhine programmeerimine (Game-themed programming) – lihtsate graafilise kasutajaliidesega mängude loomine integreeritakse programmeerimise õppimise kursusesse [10], et anda õpilaste kogutud teadmistele praktiline väljund;
- 3) kommenteeri-ja-kodeeri (Comment-first-coding) – programmiteksti kirjutatakse kõigepealt programmi sisu kommentaaridena kirja pannes ja alles seejärel funktsionaalsust lisades [11].

Kolmandas peatükis analüüsitakse kursuse sooritanud õpilaste vastuseid nii kursuse alguses kui kursuse lõpus esitatud küsimustikule. Muu hulgas uuritakse õpilaste arvamust kasutatud meetodite mõjust programmeerimise mõistmisele ning vaadeldakse õpilaste arvamuse muutust programmeerimisest kui potentsiaalsest karjäärivalikust kursuse vältel.

1. Programmeerimise õpetamise meetodid

Käesolevas peatükis kirjeldatakse lähemalt kuut vaatluse all olevat programmeerimise õpetamise meetodit. Uuritavad meetodid valiti välja selle alusel, et need pidid olema eelnevalt mõnes välismaa ülikoolis või üldhariduskoolis juba rakendatud ja olid programmeerimise õpetamise puhul andnud positiivseid tulemusi.

Põhjaliku otsimise tulemusena paistsid hea dokumentatsiooni poolt silma ja osutusid see-tõttu uuritavateks meetoditeks:

- 1) paarisprogrammeerimine (Pair programming),
- 2) mängupõhine programmeerimine (Game-themed programming),
- 3) kommenteeeri-ja-kodeeri (Comment-first-coding),
- 4) probleemipõhine õppimine (Problem-based learning),
- 5) puslepõhine õppimine (Puzzle-based learning),
- 6) kaaslase juhendamine (Peer instruction).

Järgnevates alapeatükkides tutvustatakse uuritud programmeerimise õpetamise meetodeid lähemalt. Lisaks tuuakse välja meetodite positiivsed ja negatiivsed küljed ning rakendamise tulemused.

1.1 Paarisprogrammeerimine (Pair programming)

Paarisprogrammeerimine on programmeerimise meetod, mis tuleneb väleda tarkvaraarenduse (*agile software development*) põhimõtetest ja seisneb selles, et kaks programmeerijat töötavad üldjuhul külje külje kõrval ühe arvutiga. Paarilised tegelevad koos koodi disaini, kirjapaneku ja testimisega. [8,9]

Enne kätte antud probleemi lahenduse kirjapanekut süvenevad mõlemad partnerid ülesandepüstitusse ja arutavad omavahel läbi potentsiaalsed lahenduskäigud koos nende lahenduste heade külgede ja murekohtadega. Seejärel asuvad paarilised välja valitud ideid programmeerimiskeeles realiseerima ja testima. Koodi kirjapanek paarisprogrammeerimises käib üldjuhul järgmiselt. Kõigepealt jagatakse ära rollid, milleks on „juht“ ja „kaardilugeja“. Juht valitseb olukorda, tema käsutuses on klaviatuur, et kood arvutisse kirja panna. Kaardilugeja aga samal ajal kontrollib juhi tööd potentsiaalsete vigade suhtes, teeb koodialaseid ettepanekuid ja mõtleb alternatiivseid lahendusi välja. [9]

Harvematel juhtudel harrastatakse paarisprogrammeerimist ka virtuaalsel kujul, kus partnerid ei pea töötamiseks füüsiliselt koos olema. Sellisel juhul teevad paarilised koostööd kasutades erinevaid ekraanijagamise ja reaajas suhtlemise tööriistu. Paarisprogrammeerimise puhul vahetavad paarid regulaarsete intervallide tagant rolle, veendumaks et mõlemad paarilised järjepidevalt võrdselt panustavad. [9]

Paarisprogrammeerimise juures nähakse programmeerimise õppimiseks väga palju positiivseid külgi:

1. Sama ülesande lahendamiseks kulub vähem aega [9]. Ajasääst tuleneb sellest, et erinevate teadmistega paarilised saavad tulevase programmi keerulisemad kohad eelnevalt läbi arutada ning tekkinud probleemidele koos lahendus leida.
2. Sama materjali õppimiseks kulub vähem aega, kuna tänu koostööle ja arutelule saab käsilolev teema lihtsamini selgeks [8].
3. Arendatud kood on parema disainiga ja sisaldab vähem vigu [9].
4. Paarilised on paremini motiveeritud käsiloleva ülesandega tegelema ning on enesekindlamad oma lahenduse õigsuses [9].
5. Partnerid on programmeerimise suhtes positiivsemalt meelestatud ega kardavad koodivigu.
6. Paarilised on tõenäolisemad programmeerimise kursust edukalt ära lõpetama [8].

Samas leidub paarisprogrammeerimise juures ka negatiivseid külgi:

1. Kui paariliste teadmistes on väga suur tasemevahe, võib juhtuda, et üks paariline teeb kogu töö ise ära [9].
2. Mõnedele isiksusetüüpidele lihtsalt ei sobi paaristöö. Ka käesoleva bakalaureusetöö aluseks oleval programmeerimise kursusel leidis üks õpilane, kes paarisprogrammeerimisega tegeleda ei soovinud. (Mainitud õpilane ei sooritanud kursust positiivsele hindele.)
3. Mõned paarilised võivad liigagi hästi läbi saada, lobisema hakata ja ebaefektiivseks muutuda.

Paarisprogrammeerimist kasutatakse tänapäeval paljudes Eesti ja välismaa ettevõtetes, mis rakendavad väleda tarkvaraarenduse põhimõtteid.

1.2 Mängupõhine programmeerimine (Game-themed programming)

Mängupõhine programmeerimine on programmeerimise õpetamise meetod, mis integreerib lihtsate graafilise kasutajaliidesega mängude loomise programmeerimise õppimise kursusesse [9].

Mainitud meetodi põhieesmärgiks pole õpetada õpilasi arvutimänge valmis programmeerima. Selle asemel on sihiks programmeerimise põhimõtete õpetamine läbi mõistmise, kuidas arvutimängud tegelikult töötavad [9]. Kuna paljud programmeerimise huvilised mängivad ise regulaarselt arvutimänge, on nad rohkem motiveeritud mängudega töötama. Lisaks sellele, üldjuhul pikemaajalise mängimise kogemusega õpilased on mänguvaldkonna eksperdid ja tänu sellele enda loodud programmi parimad kriitikud. [10]

Motiveerimine läbi kaasahaaravuse pole mängupõhise programmeerimise peamine positiivne külge. Meetod on õpilastele kasulik, kuna mängude loomine nõuab väga paljusid vajalikke tarkvaraarenduse oskusi nagu näiteks efektiivne mälu kasutus, erinevate andmete ja andmestruktuuride manipuleerimine, intuitiivne kasutajaliides, mängupildi ekraanile kuvamine ja tehisintellekti loomine [10]. Paljudel juhtudel antakse õpilastele ülesandekäsuga kaasa ka alusprojekt, mille puuduvad osad peavad õpilased ise ära täitma. Kuigi taolised ülesanded on raskemini mõistetavad ja aeganõudvamad, on õpilased pärast ülesandest arusaamist rohkem motiveeritud ja entusiastlikud. [9]

Tulemuste põhjal on käesolevat meetodi kasutatavate kursuste puhul õpilaste edukus kõrgem [9]. Tõenäoliselt tuleneb see sellest, et mängupõhine programmeerimine annab põneva praktilise väljundi õpilaste kursuse vältel kogutud teadmistele.

1.3 Kommenteeri-ja-kodeeri (Comment-first-coding)

Kommenteeri-ja-kodeeri on vähelevinud programmeerimise meetod, kus õpilasi õpetatakse programmiteksti kirjutama kõigepealt programmi sisu kommentaaridena kirja pannes ja alles seejärel funktsionaalsust lisades [11].

Etteantud ülesannet lahendama asudes mõtleavad õpilased kogu lahenduse läbi ja panevad programmi loogika samm-haaval kommentaaridena kirja. Alles siis, kui õpilased on kogu lahenduse ära kirjeldanud, on neil lubatud seda õpitavas programmeerimiskeeles kirjutama hakata. Kui programmi sammud on paigas, pannakse kirja muutuja omistamised ning põhiline programmistruktuur. Kirjutatakse välja vajalike funktsioonide, if-lausetega ja tsüklite päised, kuid nende sisu asemel pannakse nüüd omakorda kõik sammud kommentaaridena kirja.

Viimaks, kui kogu lahendus on konkreetsete sammudena kommentaaridesse kirja pandud, asub õpilane täitma funktsioonide, if-lausete ja tsüklite sisu. Pärast iga osa täitmist peaks õpilane koodi kompileerima ja käima panema, et veenduda oma senise lahenduse korrektsuses. [11]

Käesoleval meetodil on mitmeid positiivseid külgi:

1. Programmeerimiskeelest sõltumatu – igas keeles on kommenteerimise süntaks, mille abil õpilased saavad ülesande lahenduseks vajaminevaid samme kirjeldada [11].
2. Näitab väga hästi õpilase mõttekäiku – üldjuhul on väga paljude vigaste programmide puhul raske aru saada, kas õpilane ei jõudnud õige lahenduskäiguni puuduliku algoritmilise mõtlemise või spetsiifilise programmeerimiskeele oskuse tõttu. Käesolevat meetodit kasutades kirja pandud koodi puhul aga sellist küsimust ei teki.
3. Valmis arendatud kood on põhjalikult dokumenteeritud, tagades parema koodi loetavuse ja uuesti kasutatavuse võimaluse [11].

Seniste meetodi rakendamiste põhjal esitab rohkem õpilasi realselt kompileeruvaid programme ning sammude kirjapanek emakeeles aitab kaasa õpilaste loovusele [11]. Meetodi negatiivseks küljeks võib pidada, et lihtsamate programmide puhul puudub mõte sammude ükshaaval kirjapanekuks.

1.4 Probleemipõhine õppimine (Problem-based learning)

Probleemipõhine õppimine on programmeerimise õpetamise meetod, kus õpilastele antakse lahendada tavapärastest ülesannetest suuremad probleemid, mille sarnastega puutuvad oma igapäevatoös kokku vastava valdkonna professionaalid [9].

Probleemipõhine õppimine sai alguse hoopiski meditsiinivaldkonnast. Nimelt võeti meetod 1966. aastal McMaster University Medical Schoolis kasutusele, et anda tudengitele parem meditsiinialane haridus. Kuna käesolev meetod pole erialaspetsiifiline, levis see peagi ka teistesse kõrgkoolidesse ja ainevaldkondadesse, sealhulgas programmeerimisse. [12]

Käesolevast meetodist on mitmeid erinevaid variante. Üheks levinud viisiks on järgnev [12]:

1. Õpilased jaotatakse seitsme- kuni kümneliikmelistesse gruppidesse, kellest iga tegevust jälgib tuutor. Üldjuhul on tuutor passiivne, kuid vastab abivalmilt kõikidele küsimustele ning veendub, et grupi arutelu käsiloleva probleemi puhul valesse suunda ei areneks.

2. Gruppidele jagatakse kätte ülesanne. Õpilased viivad end sellega kurssi ja identifitseerivad probleemi.
3. Ajurünnak – grupiliikmed mõtlevad välja erinevaid lahendusi ning õpilased püüavad probleemi siduda oma eelnevate teadmiste ja lahendatud probleemidega.
4. Selgitava mudeli loomine – grupp loob esialgse versiooni probleemi selgitusest ja lahendusest, tähtsamad põhimõtted ja nendevahelised seosed identifitseeritakse.
5. Õppe-eesmärkide paikapanek – õpilased panevad paika teemad ja probleemiosad, mis on lahenduseks vajalikud, kuid mida veel ei osata. Siin punktis peaks tuutor aktiivselt aitama õppe-eesmärkide kinnitamise ja sobiva materjali leidmisega.
6. Iseseisev õppimine – iga grupiliige täidab nädala jooksul iseseisvalt kõik õppe-eesmärgid. Käesolev samm on kõige tähtsam – ilma iseseisva õppimiseta puuduvad käesoleval meetodil tulemused.
7. Arutelu õpitu kohta – grupp tuleb kokku ning arutab nädala jooksul õpitud ja rakendab seda käsilolevale probleemile.

Üldjuhul on käsitletavat probleemi küllaltki mahukad ning lahendatakse ära ühe nädala jooksul ilma reaalselt lahendust valmis programmeerimata. Näidetena võib tuua Olümpiamängude ürituste ja osalejate info talletamise või arvuti failisüsteemi Java klasside abil väljendamise. Suuremate ülesannete puhul või töötava programmi arendamisel võib lahendamine kesta kauem. Probleemipõhise õppimise positiivseteks külgedeks peetakse [12]:

- 1) eluliste probleemide lahendamine suurendab õpilaste motivatsiooni ülesandega tegeleda;
- 2) ülesannete lahendamine gruppides suurendab liikmete vastutust ja kuuluvustunnet;
- 3) õpilased ei ole probleemide lahendamisel ükski ja on tõenäolisemad kursust lõpetama;
- 4) õpitud oskused jäävad pikemaks ajaks meelde ning tagavad tihti paremad tulemused järgmistel kursustel.

Käesoleva meetodi negatiivseteks külgedeks on:

- 1) vajab mitut tuutorit, tehes keerukamaks kursuse läbiviimist [12];
- 2) hea pigem teoreetiliste teadmiste ja põhimõtete, mitte praktilise programmeerimise õppimiseks;
- 3) nõuab palju iseseisvat tööd, mis gümnaasiumi valikkursuse puhul pole soovituslik.

Probleemipõhist õppimist on rakendatud näiteks Helsingi Tehnoloogiaülikooli objektorienteeritud programmeerimise Javas kursusel. Rakendamise tulemusena lõpetas kursuse tavapärasest suurem arv tudengeid ning õppimise kvaliteet suurenes. [12]

1.5 Puslepõhine õppimine (Puzzle-based learning)

Puslepõhise õppimise eesmärgiks on õpetada kriitilist mõtlemist ja probleemi lahendamise tehnikaid. Käesoleva meetodi rakendamisel õpilastele jagatavad ülesanded peaksid olema üldised ja hõlpsasti meelde jäävad, samas pakkuma lahendajale väljakutset ja olema vähemalt näiliselt ilma lihtsa lahenduseta. [9, 13]

Programmeerimise kursustel näib meetodi rakendamine üldjuhul välja järgmiselt [9]:

- 1) õpilasele esitatakse teemakohane probleem;
- 2) probleemi programmiline lahendus tehakse väiksemateks osadeks, nõ pusletükki-deks;
- 3) õpilane asub probleemi lahendama, pannes pusletükke õigesse järjekorda;
- 4) õpetaja või automaatne süsteem kontrollib lahenduse õigsust ja annab õpilasele tagasiside.

Puslepõhine õppimine on hea kontrollimaks õpilaste teoreetilist arusaamist programmeerimise põhimõtetest, kuid ei aita suuremal määral kaasa praktiliste programmeerimisoskuste arendamisele. Küll aga tõusevad antud meetodit kasutades õpilaste huvi ja kursusest aktiivne osavõtt [9].

1.6 Kaaslase juhendamine (Peer instruction)

Kaaslase juhendamine on programmeerimise õpetamise meetod, kus õpilased omandavad käsiloleva teema iseseisvalt ja tunnis vastavad küsimustele, mille põhjal õpetaja kujundab oma edasised teemakohased selgitused [14].

Kaaslase juhendamise puhul antakse õpilastele ülesanne kindel osa teooriast iseseisvalt omandada. Õppimisejärgselt, kuid enne vastava tunni toimumist, lastakse õpilastel sooritada ka teemakohane test. Seda tehakse selleks, et motiveerida õpilasi materjal endale selgeks tegema ja anda neile tagasisidet nende valmiduse kohta tunnis osalemiseks. Tund ise seisneb suuresti õpitu arutamises ja teemakohastele valikvastustega küsimustele vastamises kas klikkerite, spetsiifilise veebirakenduse nagu Socratic või mõne muu vahendi abil. Küsimused käsitlevad tihti käsiloleva teema põhimõtteid või levinud eksiarvamusi. [14]

Pärast õpetajapoolset küsimuse esitamist toimitakse järgmiselt [14]:

- 1) õpilased mõtlevad iseseisvalt küsimuse peale ja valivad vastuse;
- 2) seejärel arutavad õpilased käsilolevat küsimust väikestes gruppides;
- 3) pärast omavahel arutlemist vastavad õpilased uuesti samale küsimusele;
- 4) kogu klass arutab vastusevariantide üle ja õpilased seletavad oma arvamust.

Eelmainitud protsessi iga osa on väga tähtis. Esimene samm, iseseisev vastamine, kindlustab, et iga õpilane on valinud vastuse ja vähemalt mingil määral küsimusse süvenenud. Teine samm sunnib õpilasi oma arvamust väljendama, argumenteerima ja kogu grupiga õige vastuse valikus konsensuseni jõudma. Selles sammus väljendubki meetodile nime andnud kaaslase juhendamine nende õpilaste puhul, kes esialgu arvasid valesti. Pärast uuesti küsimusele vastamist ja üleklassilist arutelu saab õpetaja selge ülevaate õpilaste arusaamast ning võib soovi korral mõned põhimõtted üle seletada, enne järgmise osa juurde siirdumist. [14]

Arvestatava koguse tunnieelse õppimise tõttu võtab kaaslase juhendamine sama koguse materjali läbimiseks õpilastel mõnevõrra rohkem aega. Samas saavad käsitletavat teemat õpilastel paremini selgeks. [14]

2. Programmeerimise valikkursuse läbiviimine

Käesolevas bakalaureusetöö peatükis kirjeldatakse üldist programmeerimise õpetamise olukorda ja pakutavaid kursuseid Eesti üldhariduskoolides. Samuti selgitatakse, mille põhjal valiti välja Pythoni kursusel rakendamisele kuulunud kolm programmeerimise õpetamise meetodit. Viimasena antakse ülevaade läbiviidud Pythoni valikkursusest. Vaadeldakse kursuse struktuuri, kasutatud õppematerjale ja käsitletud programmeerimise teemasid.

2.1 Programmeerimise õpetamise olukord Eestis

Terve käesolev alapeatükk põhineb Siim Puniste bakalaureusetööl „Eesti gümnaasiumides õpetatavad programmeerimiskursused“ [15]. Nimetatud töös saadeti erinevatesse Eesti koolidesse programmeerimise õpetamist käsitlev küsimustik, millele vastasid õpetajad 17 pakutava kursuse kohta. Neist 16 olid mõeldud gümnaasiumile. Antud uurimistöö käsitles vaid koole, kelle õppekavas oli kursus, mis sisaldas oma nimes sõna programmeerimine. Muude kursuste raames, näiteks informaatika, toimuvat programmeerimise õpet ei uuritud.

Enamjaolt pakutakse programmeerimist valikainena gümnaasiumi kõikidele õpilastele, kuid kuuel juhul oli kursus kohustuslik kindlale õppesuunale, näiteks reaal- või infotehnoloogia-suund. Seitsmel juhul oli kursust pakutud juba üle nelja aasta, vaid kolme kursust pakuti käesoleval kooliaastal esimest korda. Erinevate kursuste õpilaste arv erines väga suurel määral, alates 10 õpilasest kuni 35-ni, ning kõikidel juhtudel peale ühe tundis õpetaja, et kursusele pääsesid kõik õpilased, kes seda soovisid. Samuti tuli välja, et enamikel kursustel oli meessoost õpilasi rohkem.

Selgus, et 16 gümnaasiumikursusel õpetati koguni 14 erinevat programmeerimiskeelt. Populaarseimateks osutusid Scratch, Python, Java ja Javascript. Keelte valikul arvestati nende sobivust õpetamiseks ja hilisemat rakenduslikku väärtust. Kursustel kasutatavate õppematerjalide hulgas oli veelgi suurem erinevus. Paljud õpetajad kasutasid kombinatsiooni enda loodud ja veebis kättesaadavatest materjalidest. Kasulike viidetena toodi välja Tartu ja Tallinna ülikoolide programmeerimise kursuste materjale, samuti õppekeskkondi Progetiiger, w3schools.com ja php.net.

Programmeerimise õppekomplekti „Rakenduste loomise ja programmeerimise alused“ kasutati vaid kolmel kursusel 16-st. Selline tulemus on üllatav arvestades, et tegu on Euroopa Sotsiaalfondi rahastatud ja Tallinna Tehnikaülikooli töörühma poolt koostatud nõ ametlike gümnaasiumi programmeerimise õppematerjalidega.

2.2 Programmeerimise õpetamise meetodite valik

Efektiivseks programmeerimise õpetamiseks ja õpilaste huvi hoidmiseks, otsustati autori poolt läbiviidud Pythoni valikkursusel kasutada kuuest uuritud programmeerimise õpetamise meetodist kolme. Meetodite valikul pidi arvestama sellega, et kursusel on 26 gümnaasisti, kellest mõnedel puudub igasugune eelnev kogemus programmeerimisega.

Eesmärk oli rakendada ja seejärel uurida võimalikult paljusid meetodeid, ilma et nende rohkus hakkaks õpilasi kursusematerjali omandamisel segama. Seetõttu kujunes kasutatud meetodite arvuks just nimelt kolm. Rakendamisele kuulunud õpetamise meetodid valiti välja järgmiste kriteeriumite alusel:

- 1) meetodid pidid olema eelnevalt mõnes välismaa ülikoolis või üldhariduskoolis juba rakendatud;
- 2) meetodid olid programmeerimise õpetamise puhul andnud positiivseid tulemusi;
- 3) meetodid oleksid kursuse raames hõlpsasti rakendatavad;
- 4) meetodid toetaksid, mitte ei konkureeriks üksteisega.

Kuna kursusematerjaliks oli Tartu Ülikooli Programmeerimise kursuse õpik [16] – s.t valitud õppematerjal ja ülesanded olid suunatud informaatikatudengitele, mitte gümnaasistidele –, siis oli tähtis, et raskuste ilmnemisel oleks õpilastel olemas tugi. Selle probleemi lahendamiseks sai üheks õpetamise meetodiks valitud paarisprogrammeerimine.

Selleks et motiveerida õpilasi terve kursuse vältel järjepidevalt õppima ja anda omandatud teadmistele praktiline väljund, rakendati mängupõhist programmeerimist kursuse lõpuprojekti näol. Samuti tutvustati kursuse jooksul kommenteeri-ja-kodeeri meetodit, mille abil läheneda keerukamatele probleemidele takerdumata programmeerimise või Pythoni detailidesse.

Seega osutusid valitud meetoditeks:

- 1) paarisprogrammeerimine (Pair programming),
- 2) mängupõhine programmeerimine (Game-themed programming),
- 3) kommenteeri-ja-kodeeri (Comment-first-coding).

Probleempõhine õppimine ei leidnud rakendust, kuna see vajab mitut tuutorit – igale gruppile üht –, mida koolitingimustes on väga keeruline saavutada. Samuti nõuab mainitud mee-

tod õpilastelt palju iseseisvat tööd. Võttes arvesse, et programmeerimist õpetati valikkursusel, kus gümnaasistid pole veel oma huvis programmeerimise vastu kindlad, ei tohtinud töömaht väga suur olla.

Kaaslase juhendamise meetod ei osutunud valituks, sest niigi lühikese valikkursuse puhul pidi materjali pidevalt edasi võtma ja ei saanud väga kauaks ühe teema peale pidama jääda. Samuti, kuna juba oli valitud paarisprogrammeerimine, mille puhul õpilased saavad uue teema omavahel läbi arutada, tundus kaaslase juhendamine üleliigsena.

Puslepõhist õppimist ei rakendatud põhjusel, et see nõuab eelnevat ülesannete ja pusletükide väljamõtlemist, sobivat veebikeskkonda puslede lahendamiseks ja ei toeta oluliselt valituks osutunud meetodeid.

2.3 Pythonis programmeerimise valikkursuse läbiviimine

Käesoleva bakalaureusetöö raames viis töö autor ühes Eesti gümnaasiumis läbi Pythonis programmeerimise valikkursuse. Kursuse ainekava on leitav lisast 1.

Kõnealuses koolis pakuti juba eelnevalt Scratchis ja Logos programmeerimise ning Javas programmeerimise kursust. Töö autori ettepanekul loodi uus aine, Programmeerimine Pythonis, et ühtlustada üleminekut mainitud kursuste vahel. Programmeerimise õpetamise keeleks sai Python valitud järgmistel põhjusel:

1. Hea loetavus – Pythoni süntaks on programmeerimiskeelte hulgas üks selgemaid ja kirjutatud kood kergemini loetavaid. [15]
2. Toetus – Pythoni puhul on veebis leitav põhjalik dokumentatsioon ning suur hulk õpetusi ja probleemilahendusi. [15]
3. Lihtsus – õpilased ei pea ülemäära süvenema erinevatesse andmetüüpidesse või muudesse enamlevinud keeltega kaasnevatesse detailidesse.
4. Lihtne arenduskeskkond – Pythoni arenduskeskkond IDLE on kergesti õpitav ja vähe segadusse ajav.

Samuti on Pythoni puhul internetist hõlpsasti kättesaadav eestikeelne programmeerimise õpik [16]. Nimetatud Aivar Annamaa e-õpik käsitleb põhjalikult programmeerimise põhimõtteid ja oligi suuresti Pythonis programmeerimise valikkursuse õppematerjaliks. Kursuse piirarv, 20 osalejat, täitus kiiresti ning lõpuks osales kokkuleppel õpetajaga kursusel 26 õpilast 10. ja 11. klassidest. Osalejatest 9 olid naissoost ja 17 meessoost. Kursusel puudusid

eeldusained või -tingimused, kuid õnnelikul kombel oli enamus õpilastest juba programmeerimisega kokku puutunud.

Kursusele eelnenud perioodil läbisid Scratchis ja Logos programmeerimise kursuse 14 õpilast, kellest kolm olid lisaks Tartu Ülikooli Teaduskooli või TTÜ koolituse „Teeme ise arvutimänge“ kaudu programmeerimisega põhjalikumalt kokku puutunud. Õpilasi, kes olid kursusele eelnevalt Pythoni programmeerimiskeelega kokku puutunud, oli kaheksa. Igasugune programmeerimise kogemus puudus viiel õpilasel.

Kursuse kestuseks oli seitse nädalat ning tunnid toimusid kolm korda nädalas ja kestsid 75 minutit, lisaks leidsid õpetajaga kokkuleppel kord nädalas aset konsultatsioonid. Esimesed neli nädalat õpiti intensiivselt programmeerimise põhimõtteid. Järgmised kaks nädalat tegeleti lõpuprojektidega, mida viimasel nädalal kaitsti. Kursuse vältel läbitud teemadeks olid:

- 1) tehted arvudega ja sõnetöötlus,
- 2) muutujad ja tõeväärtused,
- 3) sisend ja väljund,
- 4) tingimus- ja korduslauseid,
- 5) funktsioonid ja järjendid.

Lühikese õppeperioodi tõttu ei jõutud kursuse jooksul käsitleda teemasid nagu failioperatsioonid ja mitmemõõtmelised järjendid. Vähemtähtsaks pidas õpetaja ka hulkade ja sõnaslike kasutamise õpetust. Rekursiooni ei õpetatud põhjusel, et see võib gümnasistidele liiga keeruliseks osutada. Teemade käsitus nägi üldjuhul välja järgnevalt:

1. Kui jõuti uue teema juurde, esitleti seda tunni alguses.
2. Seejärel näitas õpetaja väiksemaid näiteid tervele klassile ette, projitseerides enda arvutis toimuvat klassi ees olevale seinale.
3. Kohati lahendas õpetaja ka esimese teemakohase ülesande ja jättis koodinäite seinale nähtavaks.
4. Siis võtsid õpilased nädala alguses paikapandud paaridesse või moodustas uued paarid õpetaja.
5. Viimaks asuti ülesandeid lahendama. Sel ajal käis õpetaja klassis ringi ja pakkus õpilastele abi.

Neljanädalase õppeperioodi jooksul rakendati aktiivselt paarisprogrammeerimist, et materjali omandamist hõlbustada. Käesoleva töö juhendaja soovitusel moodustati võimalusel soolises mõttes segapaarid näiteks eesnime esimese tähe või sünnikuupäeva järgi. Iga nädala alguses moodustati uued paarid.

Neljanda nädala alguses tutvustati õpilastele kommentaari-ja-kodeeri meetodit, viies ja kuues nädal olid iseseisva projektiga tegelemiseks. Selleks paluti õpilastel moodustada üldjuhul kahe- või kolmeliikmelised rühmad. Seitsmendal nädalal kaitsti valminud lõpuprojekti. Projektile erilisi piiranguid ei seatud. Küll aga pidi see rahuldama vähemalt kahte tingimust järgnevatest:

- 1) Tihe seos mõne muu ainega;
- 2) Graafiline kasutajaliides;
- 3) Programm, mis kirjutab/loeb failist;
- 4) Programm, mis töötleb kahe- või kolmemõõtmelisi järjendeid.

Kursuse sooritas 24 õpilast 26-st. Ühe läbikukkumise põhjuseks olid tervislikud probleemid, mistõttu õpilane jõudis esimest korda tundi alles õppeperioodi neljandal nädalal, ja maha jäämus teistes ainetes, mille tulemusel otsustas õpilane pärast õpetajaga läbi arutamist kursuse pooleli jätta. Teisel juhul oli tegu õpilasega, kes puudus mitmest tunnist põhjuseta, keeldus osa võtmast paarisprogrammeerimisest ning sooritas kontrolltöö mitterahuldavalt.

Kursuse alguses ja lõpus pidid õpilased täitma küsimustiku. Muuhulgas vastasid õpilased küsimustele iga rakendatud meetodi kasulikkuse kohta, miks registreeriti end Pythoni valikkursusele ja kas kaaluti programmeerimist ka tulevikutööna. Kooli juhtkonna poolt läbi viidud tagasiside küsimustiku põhjal jäi enamus õpilastest kursusega väga rahule. Kiideti õpetaja meisterlikkust, õhkkonda ning õppemeetodeid. Küll aga kurdeti kursuse liiga kiire tempo pärast.

3. Programmeerimise õpetamise meetodite tulemused

Käesolevas peatükis analüüsitakse kursuse sooritanud 24 õpilase vastuseid kursuse alguses ja lõpus esitatud küsimustikele, mis on leitavad lisast 2. Esimese ankeedi täitsid õpilased kohe esimese tunni alguses, teise pärast lõpuprojekti kaitsmist.

Muuhulgas uuriti õpilaste käest nende eelnevaid programmeerimise ja paaristöö kogemusi, miks registreeruti antud kursusele ja nende ootuseid selle suhtes ning nende arvamust rakendatud meetodite suhtes. Järgnevates alapeatükkides uuritaksegi lähemalt õpilaste arvamust kasutatud meetodite mõjust programmeerimise mõistmisele ning vaadeldakse õpilaste arvamuse muutust programmeerimisest kui potentsiaalsest karjäärivalikust kursuse vältel.

3.1 Paarisprogrammeerimise mõju programmeerimise mõistmisele

Kursuse alguse küsimustikust selgus, et õpilaste eelnev kogemus paaristööga, mida paarisprogrammeerimine endas sisuliselt kujutab, oli pigem positiivne.

Kuigi muretseti olukordade pärast, kus paariline enda osa tegemata jätab, ning oldi natuke skeptilised, kuidas paaristöö programmeerimises välja näeks, siis üldiselt oli õpilaste suhtumine kursuse raames potentsiaalselt tehtavasse paaristöösse hea. Toodi välja, et mitmekesi töötades tekib rohkem ideid ja saab kuulata kellegi teise arvamust antud teema suhtes. Samuti mainiti, et kogemus on parem, kui paariliseks on taibukas inimene, keda tuntakse ja kes pole ükskõikne.

Kursuse lõpu küsimustikus uuriti õpilaste käest, kui võrd aitas paarisprogrammeerimise rakendamine kursuse esimesel neljal nädalal kaasa programmeerimise mõistmisele/ülesannete lahendamisele. Jooniselt 1 on näha, et üle poole õpilastest arvas, et käesolev meetod aitas pigem palju või väga palju. Vaid üks õpilane 24-st arvas, et meetod aitas pigem vähe.



Joonis 1. Õpilaste arvamus paarisprogrammeerimise kasulikkusest.

Samuti selgus, et kogemus paarisprogrammeerimisega oli üldjuhul väga positiivne, seda ka nende õpilaste puhul, kellele eelnevalt paaristöö tegemine eriti ei meeldinud. Toodi välja, et paaris oli programmeerida palju lõbusam, genereeriti rohkem ideid ja lahenduskäike ning ülesanded said kiiremini ja vähemate vigadega valmis. Lisaks tutvuti uute õpilastega teistest klassidest.

Autori meelest aga oli tähtsaimaks tagasiside punktiks ülesannete kahekesi läbiarutamine. Kirjutati, et mõlemal partneril olid omad arvamused ja mõtted, mille väljendamine paarilisele tekitas kohati raskusi. See tähendab, et lisaks programmeerimise õppimisele arendab käesolev meetod ka valdkonnaalast eneseväljendusoskust. Kui mõlemad partnerid olid oma arvamusi väljendanud ja põhjendanud, jõuti üldjuhul kõige mõistlikuma lahenduskäigu suhtes konsensusele.

3.2 Kommentaari-ja-kodeeri mõju programmeerimise mõistmisele

Pythonis programmeerimise kursuse neljanda nädala alguses tutvustati õpilastele kommentaari-ja-kodeeri meetodit.

Kuna neljanda nädala lõpus toimus kontrolltöö ja õpilaste mõtted olid tõenäoliselt suuresti tuleva töö juures, ei pälvinud see märkimisväärselt head vastuvõttu ega leidnud õpilaste seas laialdast kasutust. Samuti oli puudumiste arv kõne all oleva nädala tundidest üpris suur,

mistõttu ei pruukinud mõned õpilased kursuse lõpuks täpselt olla aru saanud, mida meetod endast tegelikult kujutab.

Autor arvab, et järgmisel kursuse läbiviimisel tuleks näidisülesannete ettelahendamisel kohe kursuse alguses õpetajal antud meetodit kasutama hakata. Samuti tasuks tugevamalt julgustada õpilasi sellisel viisil koodi kirjutama. Samas tasub arvestada, et kui kursusel juba rakendatakse paarisprogrammeerimist ja õpilased ülesande lahenduskäigu eelnevalt läbi arutavad, pole sammude inimkeeles kirjapanekul enam nii suurt kasu. Sellisel juhul on õpilased oma lahenduse, esitledes seda paarilisele, juba läbi mõelnud.

Kursuse lõpu küsimustikus uuriti õpilaste käest, kuiõrd aitas kommenteeri-ja-kodeeri meetodi rakendamine kaasa programmeerimise mõistmisele/ülesannete lahendamisele. Kontrastina paarisprogrammeerimisele arvas enamus kursuse lõpetanutest kas pigem või väga vähe. Samas on jooniselt 2 näha, et õpilastest kolmandiku arvates oli meetod ikkagi kasulik.



Joonis 2. Õpilaste arvamus kommenteeri-ja-kodeeri meetodi kasulikkusest.

Neljandal nädalal oli õpilastel vaid kahes tunnis võimalus käesolevat meetodi kasutada ja viiendast nädalast alustasid õpilased oma lõpuprojektidega tegelemist, kus vähenes keeruliste algoritmide väljamõtlemise osakaal ja seega meetodi kasutamise vajadus. Vähesest kasutamisest tulenevalt spekulereib autor, et õpilaste vastused antud küsimusele ei pruukinud olla väga põhjalikult läbi mõeldud ning et tugevaid järeldusi antud andmete põhjal teha pole soovituslik.

3.3 Mängupõhise programmeerimise mõju programmeerimise mõistmisele

Mängupõhise programmeerimise meetodit rakendati kursusel lõpuprojekti näol. Kuigi õpilastel polnud kohustust tingimata mäng programmeerida, siis kõik rühmad peale mõne üksiku tegid projektina midagi täpselt nimetatud valdkonnast. Seetõttu võib kursuse lõpuprojekti pidada justnimelt mängupõhise programmeerimise meetodi rakendamiseks.

Muuhulgas löid õpilased:

- 1) kaks ussimängu, kus ussi peab juhtima söögini, minemata seejuures pihta seintele;
- 2) kolm mängulist õppekomplekti näiteks leelismuldmetallide või matemaatika valemitte õppimiseks;
- 3) kosmoselaeva läbi asteroidisaju juhtimise mängu;
- 4) põhjalikes 2D labürintides üksiku ruudu koju juhatamise mängu.

Kursuse lõpuküsimustikus uuriti õpilaste käest, kuivõrd aitas programmeerimise mõistmisele kaasa lõpuprojekt. Jooniselt 3 on näha, et valdava enamuse ehk 24 kursuse lõpetanust 21 arvates aitas projekt pigem palju või väga palju ja mitte ühegi õpilase arvates pigem või väga vähe.



Joonis 3. Õpilaste arvamus lõpuprojekti kasulikkusest.

Niivõrd hea tulemus on muljetavaldav. Autor spekulereib, et meetodi edu õpilaste seas tuleneb sellest, et õpilased olid üpris vabad looma seda, mida ise soovisid, ning lõpuprojekt andis õpilastele perspektiivi ja praktilise väljundi kursuse vältel kogutud oskustele.

3.4 Õpilaste arvamuse muutus programmeerimisest

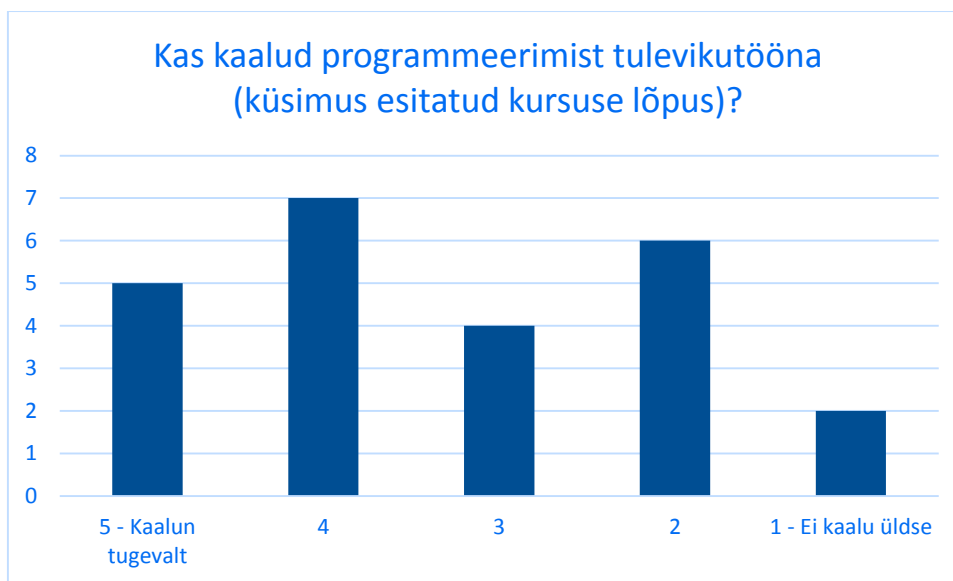
Kuigi mõnede õpilaste jaoks oli kursuse tempo natuke liiga kiire, tundis suur enamus õpilastest, et nende ootused Pythonis programmeerimise kursusele said täidetud.

Toodi välja, et kuigi iga teema jaoks oli aega mõnevõrra vähe, saadi Pythonist korralik ülevaade ja tugevad algteadmised, mille abil on nüüd võimalik lihtsamalt iseseisvalt edasi õppida. Samuti mainiti, et kursus oli põhjalik ja sisukas, õpetaja abivalmis ja oskuslik, ning rõõmustati omaenda mängu loomise üle. Nii kursuse alguse kui lõpu küsimustikus uuriti õpilaste käest, kas nad kaaluvad programmeerimist tulevikutööna. Jooniselt 4 on näha, et õpilaste arvamus kursuse alguses oli küllaltki ühtlaselt jaotunud ja selget mustrit vastustest ei leia.



Joonis 4. Õpilaste arvamus programmeerimisest kui potentsiaalsest karjäärivalikust kursuse alguses.

Kursuse lõpus küsiti sama küsimust uuesti. Jooniselt 5 on näha, et märkimisväärset muutust õpilaste üldises arvamuses ei toimunud.



Joonis 5. Õpilaste arvamus programmeerimisest kui potentsiaalsest karjäärivalikust kursuse lõpus.

Kursuse tulemusena lisandus kaks õpilast, kes vastasid, et pigem kaaluvad või kaaluvad tugevalt programmeerimist tulevikutööna. Kokku muutus arvamus programmeerimisest 10 õpilasel, neist viiel positiivses suunas ja viiel negatiivses suunas. Kokkuvõttes muutus üldine arvamus väga vähesel määral positiivses suunas.

3.5 Kursuse tulemused

Kokkuvõtlikult jäid õpilased pakutud kursusega väga rahule. Kiideti õpetaja meisterlikkust, õhkkonda ning õppemeetodeid, eriti paarisprogrammeerimist ja kursuse lõpuprojekti. Küsimustikust selgus järgnev:

- 1) kõige kasulikumaks meetodiks peeti kursuse lõpuprojekti näol rakendatud mängu-põhist programmeerimist, kus õpilased said ise otsustada, mida nad programmeerivad;
- 2) küllaltki kasulikuks peeti ka paarisprogrammeerimist, mille abil tekkis ülesannete lahendamiseks rohkem ideid ja sai lahenduskäike paarilisega läbi arutada;
- 3) kommentaari-ja-kodeeri meetod tekitas õpilastes vastakaid arvamusi tema kasulikkuse suhtes ning eriti laialt kasutatuks ei osutunud;
- 4) õpilaste suhtumine programmeerimisse kui potentsiaalsesse karjäärivalikusse muutus kursuse vältel minimaalselt, kuid ikkagi positiivses suunas.

Kuigi kurdeti kursuse kõrge tempo üle, jäid õpilased tagasiside põhjal siiski väga rahule.

4. Kokkuvõte

Käesolev bakalaureusetöö annab ülevaate kuuest programmeerimise õpetamise meetodist ning põhineb suuresti ühes Eesti gümnaasiumis töö autori poolt läbi viidud Pythonis programmeerimise valikkursusel.

Töös kirjeldatakse praegust programmeerimise õpetamise olukorda Eestis, mis on võrdlemisi mitmekesine. Järgmisena põhjendatakse töö autori poolt läbi viidud kursusel rakendust leidnud meetodite arvu ja valikut. Lühikese õppeperioodi, kuue nädala jooksul sooviti katsetada võimalikult mitut programmeerimise õpetamise meetodit, ilma et nende rohkus hakkaks õpilaste õppetööd segama. Lõpuks osutus valitud meetodite arvuks kolm.

Töös käsitletud ülejäänud meetoditest eristusid paarisprogrammeerimine, mängupõhine programmeerimine ning kommenteeeri-ja-kodeeri oma seniste heade tulemuste ja hõlpsa rakendatavuse poolest. Lisaks anti ülevaade probleemipõhisest õppimisest, puslepõhisest õppimisest ja kaaslaste juhendamisest, mis erinevatel väljatoodud põhjustel läbi viidud kursusel kasutust ei leidnud.

Pythonis programmeerimise kursuse lõpetas 26 õpilasest 24. Tunnid toimusid kolm korda nädalas ja iga tunni kestuseks oli 75 minutit. Kursuseteemadeks olid tehted arvudega ja sõnetöötlus, muutujad ja tõeväärtused, sisend ja väljund, tingimus- ja korduslaused ning funktsioonid ja järjendid. Kohe kursuse algusest tutvustati õpilastele paarisprogrammeerimist, neljandal nädalal tutvuti kommenteeeri-ja-kodeeri meetodiga ja viimased kaks nädalat tegeleti lõpuprojekti näol mängupõhise programmeerimisega.

Paarisprogrammeerimine ja mängupõhine programmeerimine osutusid õpilaste seas arvestatavalt menukaks. Neist esimese puhul kiideti paarilisega ülesande läbi arutamise võimalust, kiiremat vigade leidmist ning teiste klasside õpilastega tutvumise võimalust. Mängupõhise programmeerimise puhul meeldis õpilastele, et nende kursuse vältel kogutud oskused said praktilise väljundi ning nad said ise valida, mida nad programmeerivad. Seevastu kommenteeeri-ja-kodeeri meetod tekitas õpilastes vastakaid arvamusi ja seda ei peetud tingimata kasulikuks.

Kokkuvõtlikult jäid õpilased rakendatud meetoditega rahule ning töö autor soovib julgelt ka teistel programmeerimise õpetajatel käesolevas töös kirjeldatud meetodeid kasutada, eriti paarisprogrammeerimist ja mängupõhist programmeerimist, viimast näiteks kursuse lõpuprojekti näol.

5. Tsiteeritud teosed

- [1] Dan Crow. (2014, veebruar) "Why every child should learn to code", The Guardian. [Online]. <http://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick> (viimane külastus: 03.05.2015)
- [2] Kristi Kivilo. (2014, detsember) ""KoodiTunniga" meelitatakse noori infotehnoloogia juurde", ERR Uudised, Maria-Ann Rohemäe. [Online]. <http://uudised.err.ee/v/majandus/e01b8c0a-23ef-47b6-a315-71a51fd43c9b> (viimane külastus: 03.05.2015)
- [3] Tartu Ülikooli matemaatika-informaatika teaduskonna koduleht, sisseastumine bakalaureuseõppesse. [Online]. <http://www.math.ut.ee/et/sisseastumine/bakalaureuseope> (viimane külastus: 03.05.2015)
- [4] Eesti haridusametusesse sisseastumise infosüsteem (SAIS). [Online]. https://www.sais.ee/avalik/et/PGNMI/et_pingerida_nmi_13_13871.html (viimane külastus: 10.04.2015)
- [5] Gümnaasiumi riiklik õppekava, 2. peatükk, 4. jagu. [Online]. <https://www.riigiteataja.ee/akt/114012011002> (viimane külastus: 03.05.2015)
- [6] Jüri Vilipõld, Kersti Antoi, Irina Amitan. "Rakenduste loomise ja programmeerimise alused" gümnaasiumi valikkursuse õppematerjalid. [Online]. <http://koolielu.ee/waramu/download?rid=1-5171dab3-4720-4363-a956-b7602de68489&attachmentId=6490> (viimane külastus: 03.05.2015)
- [7] Siim Puniste. (2015, juuni) "Eesti gümnaasiumides õpetatavad programmeerimiskursused", TÜ Arvutiteaduse instituudi lõputööde register. [Online]. http://comserv.cs.ut.ee/forms/ati_report/ (tööd kaitstakse alles 2015 kevadel)
- [8] Leo Porter, Mark Guzdial, Charlie McDowell, Beth Simon, (2013, august) "Success in Introductory Programming: What Works?", Communications of the ACM. [Online]. <http://teaching.software-carpentry.org/wp-content/uploads/2013/08/p34-porter.pdf> (viimane külastus: 03.05.2015)
- [9] Sanja Mohorovičić, Vedran Strčić. (2011, september) "An Overview of Computer Programming Teaching Methods", Proceedings of the 22nd Central European Conference on Information and Intelligent Systems. [Online]. http://www.ceciis.foi.hr/app/public/conferences/1/archive2011/EIS_3.pdf (viimane külastus: 03.05.2015)
- [10] Marianne deLaet, James Kuffner, Michael C. Slattery, Elizabeth Sweedyk (2005) "Computer Games and CS Education: Why and How", Proc. SIGCSE, 2005, lk 256-257.
- [11] Arijit Sengupta. (2009) "CFC (Comment-First-Coding) - A Simple yet Effective Method for Teaching Programming to Information Systems Students", Journal of Information Systems Education, Vol. 20(4).
- [12] Esko Nuutila, Seppo Törma, Lauri Malmi. (2005, juuni) "PBL and Computer Programming - The Seven Steps Method with Adaptations", Computer Science Education, Vol. 15, No. 2, lk 123-142. [Online]. http://www.cs.hut.fi/~janne/studio1/PBL_artikkeli.pdf (viimane külastus: 03.05.2015)
- [13] Zbigniew Michalewicz, Matthew Michalewicz. (2008) "Puzzle-Based Learning: An introduction to critical thinking, mathematics, and problem solving". [Online]. <http://cs.adelaide.edu.au/~zbyszek/Papers/intro.pdf> (viimane külastus: 03.05.2015)

- [14] Beth Simon, Quintin Cutts. (2012, veebruar) "Peer Instruction: A Teaching Method to Foster Deep Understanding", Communications of the ACM 55, no. 2 (2012), lk 27-29.
- [15] Tony Jenkins. (2004, juuni) "The First Language - A Case for Python?", Innovation in Teaching and Learning in Information and Computer Sciences, Vol 3, Issue 2. [Online].
<http://journals.heacademy.ac.uk/doi/pdf/10.11120/ital.2004.03020004> (viimane külastus: 03.05.2015)
- [16] Aivar Annamaa (2012) "Programmeerimise õpik". [Online].
<https://programmeerimine.cs.ut.ee/index.html> (viimane külastus: 03.05.2015)

Lisad

I. Pythonis programmeerimise kursuse ainekava

Kursuse nimi: Programmeerimine Pythonis

Osalejate piirarv: 20

Toimumised: teisipäeviti, kolmapäeviti, neljapäeviti kell 8.20–9.35

Kursuse sisu: tehakse tutvust kergemat sorti programmeerimisega Pythoni arenduskeskkonnas IDLE. Kursuse käigus õpitakse ära Pythoni kasutusala ja süntaks ning seejärel luuakse lihtsamaid programme. Kursuse teemadeks on tekstina püstitatud ülesandele algoritmi koostamine, tehted arvudega ja sõnetöötlus, muutujad ja tõeväärtused, sisend ja väljund, tingimus- ja korduslause ning funktsioonid ja järjendid.

Õpitulemused:

- 1) õpilane orienteerub programmeerimisega seonduvas sõnavaras (programm, algoritm, plokk skeem, muutuja, avaldis, omistuslause, tingimuslause, tsükkel, funktsioon);
- 2) õpilane tunneb põhilisi andmetüüpe ja –struktuure (täis- ja ujukomaarvud, tõeväärtused, sõned, järjendid);
- 3) õpilane oskab analüüsida ja selgitada programmi töö käiku ning leida ja parandada programmikoodis esinevaid süntaksivigu;
- 4) õpilane oskab luua lihtsamale ülesandele vastava algoritmi, koostada lahendusprogrammi ning seda testida.

Õppematerjalid, lisamaterjalid, IKT kasutamine: õppematerjalid asuvad Moodle'i keskkonnas spetsiaalselt loodud kursuse alamjaotistes. Kasutatakse personaal- või kooliarvuteid, IDLE arenduskeskkonda ja dataprojektorit.

Hinde kujunemine: 40% hindest moodustab kursuse vahetest, ülejäänud 60% kursuse lõpuprojekt. Hindamine on mitmeeristav, kuid nii vahetest kui lõpuprojekti eest on tarvis koguda vähemalt 50%.

II. Pythonis programmeerimise kursuse küsimustikud

Link kursuse alguseküsimustikule: https://docs.google.com/forms/d/1Ip_T9rJ5a3-Q3khdkhPBmVaEDp_CyfSnU0FaGz46Dkk/viewform?c=0&w=1

Link kursuse lõpuküsimustikule: <https://docs.google.com/forms/d/1-gqAI-byS3Z22Rws324Z-BRbps0qAidjvZ-TyE0jO-zo/viewform>

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Mattias Nurk** (sünnikuupäev: 16.07.1992)
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Programmeerimise õpetamise meetodid ja nende rakendamine gümnaasiumi valikkursusel,
(*lõputöö pealkiri*)

mille juhendaja on Eno Tõnisson,
(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **13.05.2015**