

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND  
Arvutiteaduse instituut  
Informaatika õppekava

**Siim Puniste**  
**Eesti gümnaasiumides õpetatavad**  
**programmeerimiskursused**  
**Bakalaureusetöö (6 EAP)**

Juhendaja: Eno Tõnisson, MSc

Tartu 2015

# **Eesti gümnaasiumides õpetatavad programmeerimiskursused**

Lühikokkuvõte:

Antud töö eesmärgiks on anda ülevaade Eesti gümnaasiumides õpetatavatest programmeerimiskursustest. Andmete saamiseks koostati küsimustik, mis saadeti 31 koolile. Koolide nimekiri põhines Eesti Hariduse Infosüsteemil, kust filtreeriti välja ained ja kursused, mille nimi sisaldas sõna “programmeerimine”. Tegelikult võidakse programmeerimist õpetada ka teiste kursuste raames, kuid antud töös jäetakse need vaatluse alt välja. Küsimustikule vastati 16 programmeerimiskursuse kohta. Saadud andmete põhjal koostati kursuste ülevaade, milles kirjeldatakse kursuste ülesehitust ja ainekava, kasutatavaid programmeerimiskeeli ja õppematerjale ning koostöövalmidust tulevikuks. Samuti antakse eraldi lühiülevaade igast vastustes väljatoodud programmeerimiskeelest. Töö lõpus tuuakse tulemustel põhinedes välja mõtteid ja ideid, mida võiks edasistel uurimustel aluseks võtta.

Võtmesõnad:

informaatika gümnaasiumides, arvutiõpetus, programmeerimiskursused gümnaasiumides.

## **Programming courses in Estonian secondary schools**

Abstract:

The purpose of the current bachelor's thesis is to give an overview on programming courses taking place in Estonian secondary schools. Data was collected by sending out surveys to a list of schools that, based on Estonian Education Information System, were teaching a course in programming. Answers were collected for 16 different programming courses. Based on gathered information, an overview of the courses was made, including the structure of the course, used programming languages and learning materials, and willingness for cooperation. In addition, a short overview of all mentioned programming languages was included. At the end of the thesis, some points of interests are brought out for possible future research.

Keywords:

computer science in secondary schools, informatics, information and communication technology, programming courses in secondary schools.

# Sisukord

Sissejuhatus.....	4
1. Varasemad uuringud.....	7
1.1. USA ja Iisrael.....	7
1.2. Suurbritannia.....	9
1.3. Uus-Meremaa.....	9
1.4. Eesti muu maailma taustal.....	10
2. Andmete kogumine.....	12
3. Tulemused.....	14
3.1. Kursustest üldiselt.....	14
3.2. Kursuste sisu.....	15
3.3. Kursustel õpetatavad programmeerimiskeeled.....	17
3.4. Kasutatavad õppematerjalid.....	19
3.5. Võimalik koostöö erinevate organisatsioonide vahel.....	19
4. Ülevaade programmeerimiskeeltest.....	21
4.1. Haridusele orienteeritud programmeerimiskeeled.....	21
4.2. Tööstusele orienteeritud programmeerimiskeeled.....	25
5. Kokkuvõte.....	29
6. Kasutatud kirjandus.....	31
Lisad.....	35
I. Programmeerimise kursuste küsimustik.....	35
II. Kursuste nimetused.....	41
III. Õpetajate arvamused.....	42
IV. Näidisülesanded.....	43
V. Koostöövormid.....	44
VI. Litsents.....	44

# Sissejuhatus

Infotehnoloogia (IT) tähtsus ühiskonnas on viimaste aastate jooksul aina kasvanud. Suurenenud nõudlus uute tarkvaralahenduste vastu on muutnud IT arendamisega seotud inimesed väga nõutavateks nii Eestis kui mujal. Ometigi on tarkvara tootmise valdkond enamike inimeste jaoks suhteliselt võõras. Läbi ajaloo on arvutikasutus nõudnud erinevaid oskusi. IT algusaegadel, kui lõppkasutajale mõeldud tarkvara oli veel suhteliselt vähe, oli arvutikasutuse juures oluliseks oskuseks programmeerimine. IT valdkonna arenedes hakati tootma järjest rohkem lõppkasutajale mõeldud tarkvara, mistõttu muutus programmide kasutamisoskus tavakasutajate jaoks programmeerimisest tähtsamaks. Kõik see väljendus ka koolide IT alases õppes. Kui varasemalt tähendas IT õpe programmeerimist ja algoritmide tundmist, siis hiljem nihkusid õppe-eesmärgid programmide kasutamise peale. Näiteks 2002. aastal jõustunud riiklikus õppekavas tuuakse IT õppe-eesmärgidena välja infotehnoloogia kasutamise seostuvate majanduspoliitiliste, sotsiaalsete ja eetiliste aspektide mõistmine ning infotehnoloogiavahendite iseseisev kasutusoskus [1].

Samm programmeerimise õpetamise laiendamiseks tehti taas 2013. aastal rakendunud Eesti gümnaasiumi riikliku õppekavaga [2]. Uus õppekava sisaldab valikkursust “Rakenduste loomise ja programmeerimise alused”. Kursuse eesmärgiks on tarkvaraarenduse meetodikate ja programmeerimise aluste tutvustamine programmeerimiskeelte Scratch, Visual Basic for Applications ja Python abil. Antud kursuse jaoks on Tallinna Tehnikaülikooli informaatika instituudi töörühma poolt loodud ka spetsiaalne õppekomplekt [3].

Kuigi varasemates õppekavades on välja jäetud programmide loomist käsitlevad kursused, ei tähenda see, et selliseid kursusi ei oleks võinud koolid iseseisvalt õpetada. Samuti ei tähenda “Rakenduste loomise ja programmeerimise alused” valikkursuse lisandumine, et koolid peaksid kasutama etteantud õppematerjale või järgima täpselt õppe-eesmärke. On teada, et programmeerimise alaseid kursusi on Eesti gümnaasiumides antud juba mitmeid aastaid, näiteks hakati programmeerimist õpetama Nõo Reaalgümnaasiumis juba 1965. aastal [4].

IKT karjääri projekti [5] raames päriti Eesti Hariduse Infosüsteemist (EHIS) valikkursuste loetelu, mis võiksid sisaldada infotehnoloogiaga seostuvaid teemasid. Selle töö raames valiti

nende kursuste seast välja kursused, mille nimi sisaldas sõna “programmeerimine”. Saadud loetelu põhjal selgus, et Eesti koolides leidub mitmeid kursusi, mis eelduste kohaselt võiks olla pühendunud programmide loomise aspektile.

Käesoleva bakalaureusetöö eesmärgiks on anda ülevaade Eesti koolides õpetatavatest programmeerimise kursustest. Kursusi uuritakse mitmest aspektist – kasutatavad keeled, õppemetoodikad, õpetaja roll aine ülesehitusel ja õpetajate valmidus koostöö vastu. Tulemusena antakse ülevaade hetkeolukorrast ning analüüsitakse koolide erinevaid lähenemisi programmeerimise õpetamisele.

Bakalaureusetöö andmed koguti kolmes osas:

1. Andmepäring EHIS andmebaasist kursuste kohta, mis sisaldasid nimes sõna “programmeerimine”.
2. Info hankimine koolide kodulehtedelt.
3. Täiendava info pärimine õpetajatelt küsimustiku abil.

Bakalaureusetöö esimeses peatükis tutvustatakse erinevate artiklite põhjal olukorda teistes riikides ning tuuakse välja nendes riikides esinenud programmeerimise õpetamise alased kitsaskohad. Teises peatükis kirjeldatakse koolidelt andmete kogumise protsessi ning selgitatakse, millest küsimustiku koostamisel lähtuti. Käesoleva töö kolmandas peatükis antakse ülevaade toimuvatest kursustest ning näidatakse nende põhilisi sarnasusi ja erinevusi. Neljandas peatükis antakse lühiülevaade küsitluse põhjal selgunud õpetatavatest programmeerimiskeeltest ning tuuakse välja nende positiivseid ja negatiivseid külgi programmeerimise tutvustamise aspektist lähtudes. Töö lõpeb kokkuvõttega mis sisaldab endas ka mõtteid edasisteks uurimusteks.

Käesolevas bakalaureusetöös on oluline täpsustada kasutatavat terminoloogiat. Juba valdkonna ja võimaliku õppeaine enda nimetusi on nii võõrkeeltes kui ka eesti keeles kasutusel mitmeid. Artiklis “Computer Science/Informatics in Secondary Education” [6], tutvustatakse põhiliste ingliskeelsete terminite ajalugu ja tagamaid ning tuuakse välja, et termineid “computer science” (*arvutiteadus*), “informatics” (*informaatika*) ja “information communication technology” (*info- ja kommunikatsioonitehnoloogia, IKT*) võiks kasutada sünonüümidenä, kui käsitletav temaatika piirdub haridusega. Lisaks on veel kasutusel ka mõisted “arvutiõpetus” ja “infotehnoloogia” (IT).

Selles bakalaureusetöös eristame arvutiõpetust ja arvutiteadust. Arvutiõpetuseks nimetatakse õppeainet/kursust, mille raames õpetatakse tavakasutaja vajadustele vastavaid oskusi, nagu näiteks andme- ja tekstitöötlus ning programmide käsitlemine. Arvutiteaduse all peetakse silmas kursusi, mis on suunatud programmide loomisele. Olulisteks märksõnadeks võivad olla programmeerimine, algoritmid, robotika ja muu selline. Nii on püütud jaotada ka välismaiseid lähenemisi, kuigi konkreetne otsetõlge oleks andnud midagi muud. Näiteks tõlgitakse *informatics* õpetamine Prantsusmaal arvutiteaduse õpetamiseks, kuna tegemist on olemuselt sama asjaga, mis näiteks USA puhul on *computer science*.

# 1. Varasemad uuringud

Viimaste aastate jooksul on infotehnoloogiaalase hariduse fookus nihkumas arvutiõpetuse pealt arvutiteaduse peale. Leitakse, et lisaks arvuti kasutamisele tuleks õpilastele tutvustada ka arvutiteaduse kontseptsioone nagu näiteks algoritme, andmestruktuure ning programmeerimist. Mitmetes riikides on tehtud juba otsustavaid samme, et seda eesmärki saavutada. [7]

Leidub mitmeid artikleid, mis kirjeldavad arvutiteaduse õpetamist koolides erinevate riikide näitel. Artikkel “School Subject Informatics (Computer Science) in Russia: Educational Relevant Areas” [8] kirjeldab arvutiteaduse õpet Venemaal. Erinevalt paljudest teistest, on Venemaal arvutiteaduse (*informatics*) õpe gümnaasiumis kohustuslik. Koolidel on võimalik valida põhi- (*basic*) ja edasijõudnud (*advanced*) taseme vahel, kuid juba põhitasemel õpetatakse õpilastele algoritme ning programmide kirjutamist ja analüüsi. Artiklis “Programming in School: Look Back to Move Forward” [9] keskendutakse Rootsi infotehnoloogia õpetamise ajaloole. Uuritakse, mis põhjustel on erinevatel perioodidel eelistatud arvutiteaduse (*computer science education*) õpet arvutiõpetusele (*information and communications technology*) ja vastupidi. Prantsusmaa infotehnoloogia haridusele keskendutakse artiklis “Education in French Secondary Schools: Historical and Didactical Perspectives” [10]. Sarnaselt eelmisele artiklile, arutletakse ka selles arvutiteaduse (*informatics*) ja arvutiõpetuse (*information and communications technology*) populaarsuse kõikumise üle, kuid seekord Prantsusmaa koolisüsteemi näitel.

Järgnevalt anname ülevaate veel nelja riigi, USA, Iisraeli, Suurbritannia ja Uus-Meremaa kohta. Suuremat tähelepanu pööratakse just nendele riikidele, kuna nende kohta oli saadaval kõige paremat ja põhjalikumat infot, mis ühtlasi ühtis ka selle bakalaureusetöö eesmärkidega.

## 1.1. USA ja Iisrael

Artikkel “A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States” [11] kirjeldab arvutiteaduse õpetamist USA ja Iisraelis.

USA hariduspoliitika on riiklikul tasemel väga nõrgalt tsentraliseeritud. Erinevaid hariduspoliitilisi otsused võidakse langetada nii osariiklikul kui ka kooli tasemel. Arvutiteaduse

(*computer science*) õpetamise populariseerimise on üheks eesmärgiks seadnud organisatsioon Computer Science Teachers Association (CSTA) [12]. CSTA on juba mitmeid aastaid läbi viinud uuringuid USA koolides õpetatavate arvutiteaduslike ainete kohta, et pakkuda asjakohast infot riigile, koolidele ja üldsusele. CSTA poolt 2013. aastal läbi viidud küsitlus sisaldas mitmeid huvitavaid teemasid, mis on võetud ka käesolevas bakalaureusetöös vaatluse alla. Näiteks uuriti koolides õpetatavate programmeerimiskeelte kohta. Selgus, et kõige enam kasutati programmeerimise õpetamiseks programmeerimiskeelt Java (49%), millele järgnesid Scratch (34%), Alice (30.2%), Visual Basic (29.7%), JavaScript (15.4%), C++ või C# (18.1%), Python (13.9%) ja App Inventor(12.6%). Väiksem osa koolidest kasutas ka Greenfooti (5.9%), HTMLi (4.2%), Game Maker-it (2.5%) ja Jerood (2.3%). Sellest tulemusest järeldub ka, et paljudes koolides oli õpilastel võimalik tutvuda rohkem kui ühe programmeerimiskeelega. Samuti uuriti naissoost õpilaste osakaalu sissejuhatavates arvutiteaduslikes kursustes. Selgus, et üle poolte vastanud koolide puhul oli kursusel osalenud naissoost õpilaste osakaal alla 20%. CSTA hinnangul on viimaste aastate tulemuste põhjal põhjust rõõmustada – huvi arvutiteaduse vastu on kõigi aegade kõrgeim. Samas lisatakse, et tegelikult puudub neil ikkagi veel täielik ülevaade koolides õpetatavast arvutiteaduse tasemest.

Vastandina USAle on Iisraeli haridussüsteem tugevalt tsentraliseeritud. Arvutiteadus (*computer science*) on olnud Iisraeli õppekavas juba 1970. aastatest. Selle ajal ei peetud arvutiteaduse õpet tähtsusest võrdseks teiste ainete õpetamisega. Muutused toimusid 1980. aastate lõpul, kui Iisraeli haridusministeerium võttis vastu otsuse uuendada arvutiteaduse õppekava ja muuta arvutiteaduse õpe võrdseks teiste ainetega. Ministeeriumi poolt koostatud komitee seadis ette kindlad punktid, mida arvutiteaduse raames pidi käsitlema ning tõi välja selle rakendamisel tekkivad problemaatilised kohad. Põhjalikult uuendatud õppekava läks esmakordselt käiku 1995. aastal ning on peale seda läbinud veel mitmeid uuendusi, nagu näiteks programmeerimiskeele vahetuse Pascalilt Java peale. Kuigi arvutiteaduse õppekava on tsentraliseeritud haridussüsteemis hästi organiseeritud, ei tähenda see, et kõik töö oleks tehtud. Arvutiteaduse õpetamise tähtsuse propageerimiseks tuleb ka praegu olulist tööd teha.



## 1.2. Suurbritannia

Arvutiteaduse õpetamise tagasitulekust Suurbritannia haridussüsteemi räägib artikkel “Restart: The Resurgence of Computer Science in UK Schools” [13]. Suurbritannias puudub ühtne haridussüsteem – suhteliselt sarnased kuid siiski iseseisvad haridussüsteemid on nii Inglismaal, Walesil, Šotimaal kui ka Põhja-Iirimaal.

1980. aastatel, kui personaalarvutid olid veel suhteliselt vähelevinud, oli programmeerimine arvutikasutuse enesestmõistetav osa. Personaalarvutite leviku suurenedes hakati juurde tootma ka lõppkasutajale suunatud tarkvaralahendusi, mis muutis tähtsaks tarkvara kasutamise oskuse. 2000. aastateks oligi Suurbritannias arvutiteaduse (computer science) õpetamine peaaegu täielikult asendunud arvutiõpetusega (*information and communication technology*). Kuna tegemist oli suhteliselt lihtsa ja vähe väljakutseid pakkuva ainega, pidasid õpilased seda igavaks. 2012. aastal tõi The Royal Society [14] välja, et see probleem seisneb kvalifitseeritud õpetajate puuduses. Kuna arvutialaste teadmistega inimestele oli mitmeid kõrgemapalgalisi alternatiivseid töökohti, oli raske neid haridussektoris hoida. Koolid olid sunnitud kasutama vähesema kvalifikatsiooniga inimesi, mis tegi aine kergemaks ja ebahuvitavamaks.

2008. aastal loodi Suurbritannias huvigrupp Computing At School (CAS) [15], et edendada arvutiteaduse õpet sealsetes koolides. Mõisteti, et arvutiteaduse õpetamise tähtsustamiseks ei piisa vaid tööjõu puudusele viitamisest. Vaja oli näidata, et arvutiteadused sisaldavad ülekantavad oskusi, millest on kasu kõikidel õpilastel. Sellisteks oskusteks peeti arvutuslikku mõtlemist (*computational thinking*), loogilist mõtlemist, probleemide lahendamist, programmide silumist (*debugging*) ja algoritmilist mõtlemist. 2011. aastal toimuski läbimurre. Suurbritannia haridusministeerium kuulutas välja hariduspoliitika muudatuse, tühistades eelneva informaatika aine õppekava ja jättes koolidele vabad käed selle ainekava koostamiseks. Aine ise jäi koolides kohustuslikuks.

## 1.3. Uus-Meremaa

2011. aastal hakati Uus-Meremaa koolides õpetama arvutiteaduse (*computer science*) alaseid aineid. Artiklis “A Case Study of the Introduction of Computer Science in NZ Schools” [16]

tuuakse välja kitsaskohad, mis selle käigus ületati ning märgitakse ära aspektid, mis selle võimalikuks tegid.

Nagu enamikes riikides, oli Uus-Meremaal enne 2011. aastat infotehnoloogia õppe peamiseks teemaks arvuti ja tarkvara kasutamise õpetamine. Arvutiteadusliku osa lisamine õppekavva andis õpilastele võimaluse tutvuda ka teiste arvutialaste teemadega. Näiteks leiab uues ainekavas käsitlemist algoritmid, tehisintellektika ja krüpteerimine. Eesmärgiks seejuures pole nende teemade sügav käsitus, vaid õpilaste ülevaatlilik harimine informeeritud karjääriotsuse tegemiseks.

Artiklis tuuakse välja, et arvutiteaduse juurutamiseks koolides oli vaja töötada mitme erineva aspekti kallal. Esiteks, uus ainekava ei tohtinud olla liiga keeruline. See tagas aine parema vastuvõtlikkuse õpetajate poolt ning lihtsustas aineõpetajate koolitamist. Teise punktina peeti oluliseks meedia ja avalikkuse rolli aine populariseerimisel. Ei piisanud, kui aines õpetatu vajalikkusest said aru ainult ametnikud ja õpetajad. Tarvis oli teadvustada ka lapsi ja lapsevanemaid. Oluline oli ka õpetajate toetamine nii täiendkoolituste kui ka ühiste õppematerjalide näol. Kõiki neid kiireid muudatusi peeti võimalikuks ainult tänu heale suhtlusele kõikide osapoolte (riik, kool, ülikoolid ja tööstus) vahel.

#### **1.4. Eesti muu maailma taustal**

Selles jaotises püütakse erinevate maade olukorda kokku võtta ja seda vaadata võrdluses Eesti kontekstiga. Mitmes vaatluse all olevas riigis on tulnud esile sama trend – arvutiteaduse õpe oli viimaste aastakümnete jooksul vahetunud arvutiõppe vastu. Iseenesest on see loomulik, sest praegune tarkvara kasutusoskus on tänapäeva maailmas igati vajalik. Samas on just viimasel ajal hakatud rohkem tähelepanu pöörama arvutiteaduse õpetamise laiendamisele, mille üheks põhjuseks on ilmselt IT tööstuse tööjõu vajadus.

Küll aga on eelnevalt mainitud ainesisu muutus muutnud inimeste arusaama arvutiteaduse põhimõttest, mistõttu on ilmselt iga riigi esmaseks ja põhiliseks väljakutseks inimeste nägemuse muutmine arvutiteaduse kontseptsioonidest [11].

Mujal maailmas on arvutiteaduse põhimõtete tutvustamise ja populariseerimisega tegelenud selle jaoks loodud organisatsioonid. Kuigi Eestis leidub ka mõningaid selliseid organisatsioone ja ülikoole, ei ole siin veel ühtegi nii tugevat organisatsiooni, nagu seda on CSTA USAs [12] või CAS Suurbritannias [15].

Ühe kitsaskohana, mis esines enamikes vaatluse all olevates riikides, oli kvalifitseeritud õpetajate puudus. Suurbritannia näitel peaks ka Eestis julgustama õpilasi kaaluma arvutiteaduse õpetamist kui karjäärivalikut. Lisaks sellele, on tähtsal kohal ka olemasolevate õpetajate toetamine. Näiteks Suurbritannias loodi selle jaoks *CAS Network of Excellence* [15], mille põhieesmärgiks oli luua ühtne õpetajate võrgustik ning pakkuda professionaalset abi ja tuge.

Kuna infotehnoloogia on suhteliselt kiiresti arenev valdkond, tuleb suurt tähelepanu pöörata ka pidevale õppekava täiustamisele. Mõistlik oleks põhjalikumalt analüüsida Venemaa ja Iisraeli kogemust, kus aine õppeeesmärgid on juba pikka aega paigas olnud, kuid samas ollakse avatud väiksematele muudatustele, nagu näiteks kasutatava programmeerimiskeele vahetus vastavalt trendidele.

Lisaks eelnevale ei tohi arvutiteaduse õpetamise puhul mööda vaadata ka vajadusest varustada koole moodsa riist- ja tarkvaraga. Kuna aga üldiselt on Eesti koolides arvuteid vaja läinud juba ka arvutiõpetuse ja teiste ainete raames, ei tohiks see Eesti puhul suureks probleemiks olla.

## 2. Andmete kogumine

Eesti Hariduse Infosüsteem (EHIS) on riiklik register, mis koondab haridussüsteemi puudutavaid andmeid [17]. EHISis on ka andmed õpetajate kohta, kusjuures õpetajate puhul on märgitud, milliseid aineid või kursusi nad õpetavad. IKT karjääri projekti [5] raames päriti infosüsteemist informaatikaalaste ainete ja kursuste nimesid koolide kaupa. Päringusse kaasati ka ained, milles võib aga ei pruugi IT oluline olla, nagu näiteks joonestamine. Saadud nimekirjast valiti selle bakalaureusetöö jaoks välja kursused, mille nimetuses oli kasutatud sõna “programmeerimine”. Programmeerimisega võidi tegeleda ka teistel kursustel, kuid antud töö raames neid ei käsitleta.

Kokku vastas päringule 36 õppeainet 31st koolist. Nendest ainetest üks oli mõeldud I koolastmele, 4 III kooliastmele ja 31 IV kooliastmele. Kuna gümnaasiumide kursused olid nimekirjas väga tugevalt ülekaalus, keskendutakse selle töö raames gümnaasiumide kursustele.

EHISist kogutud andmed sisaldasid endas kooli, kooliastme ja aine nimetust. Järgmise etapina uuriti koolide kodulehti, eesmärgiga leida täiendavat infot EHISist saadud kursuste kohta ja leida kursuse õpetaja kontaktandmed, mille abil oleks võimalik neile täiendav küsimustik edastada. Koolide kodulehed ei osutunud aga väga informatiivseks – 31 koolist vaid neljal puhul suudeti leida täpsustav kirjeldus seal õpetatavatest ainetest ja kursustest.

Täiendava info saamiseks koostati küsimustik. Antud küsimustik on välja toodud käesoleva töö lisas 1. Küsimustik oli jagatud viide ossa ning koosnes kokku 30 küsimusest. Küsimustiku vastustest saadud andmeid kasutatakse antud töös konkreetsetele koolidele viitamata. Küsimustiku täitmiseks arvestatud aeg oli umbes 15-20 minutit.

Küsimustiku esimene osa koosnes sissejuhatavatest küsimustest. Taheti teada kursuse nime, millistele klassidele seda õpetatakse, millised õpilased seda kursust võtavad ja kui pikalt on seda kursust selles koolis õpetatud. Teises osas seati rõhk programmeerimiskeeltele. Paluti välja tuua keeled, mida kursuse raames kasutatakse ja põhjendada seda valikut. Küsimustiku kolmandas osas seati vaatluse alla õppematerjalid. Õpetajatel paluti hinnata kasutatud õppematerjalide kvaliteeti ja kättesaadavust. Lisaks sellele toodi eraldi välja "Rakenduste loomise ja programmeerimise alused" õppekomplekt, ning uuriti, kui palju on antud õppekomplekt koolides kasutust leidnud. Neljas osa keskendus kursuste ülesehitusele. Küsimuste eesmärk oli saada

ülevaade kursuse korralduslikust poolest ja kursusel läbitavatest teemadest. Viimase teemana käsitleti aineõpetajate huvi erinevate välja pakutud võimaluste vastu. Lisaks andsime õpetajatele võimaluse tuua välja nende arvates problemaatilised kohad programmeerimise õpetamisel ning pakkuda välja lahendusi nende parandamiseks.

### 3. Tulemused

Küsimustikule vastati 17 kursuse kohta. Nende 17 kursuse seas leidis ka üks põhikooli kursus, mis jääb selle töö piiridest välja. Igale üksikule küsimusele vastamine ei olnud kohustuslik ning mõne küsimuse puhul sai valida ka mitu varianti, seega ei pruugi iga küsimuse vastuste summa tulla sama arv. Kuna see peatükk põhineb statistiliselt väiksel arvul vastustel, ei tohiks siin välja toodud andmete põhjal teha väga põhjapanevaid järeldusi. Pigem võiks järgnev tekst olla aluseks uute hüpoteeside püstitamiseks edasistes uuringutes. Läbiviidud küsimustiku leiab antud töö lisast 1.

#### 3.1. Kursustest üldiselt

Kursuseid oli mitme erineva nimega. Gümnaasiumi riiklikus õppekavas väljatoodud nime “Rakenduste loomise ja programmeerimise alused” kasutas 16 kursusest vaid kaks. Enamasti toodi nimes välja kas programmeerimiskeel või märgiti, et tegemist on alg- või sissejuhatava kursusega. Kõik küsitluse vastustes märgitud nimed on välja toodud lisas 2.

16 kursusest kuue puhul oli tegemist valikkursusega, mis oli kohustuslik teatud õppesuunale. Enamasti oli selleks kas real- või infotehnoloogia suund. Ülejäänud kümne kooli puhul oli kursus kõikidele gümnaasiumi õpilastele valikuline.

Kursused ei olnud enamike koolide jaoks õppekavas uued. Seitset kursust oli antud rohkem kui neli aastat, viit kursust kaks kuni kolm aastat ning nelja kursust anti esimest korda. Mõned õpetajaid märkisid, et tegelikult on nende koolis programmeerimist õpetatud juba väga kaua, vahepeal oli lihtsalt muutunud kursuse nimi või tehtud sisulisi muudatusi.

Kahte kursust õpetati mitu korda õppeaasta jooksul, ühte harvemini kui korra õppeaastal, ning ülejäänud 13 ühe korra õppeaasta jooksul.

Kursuste maht oli enamasti 35 akadeemilist tundi, mis tähendab keskmiselt ühte tundi nädalas ühe õppeaasta jooksul. Nelja kooli puhul märgiti, et kogu ainekava läbimiseks on ette nähtud rohkem kui üks kursus, mistõttu oli kogu kursuse pikkuseks tegelikult kaks kuni viis korda 35 tundi.

Kuues koolis õpetati ainet ainult kindlale klassile, kuid selleks klassiks võis olenevalt koolist olla nii kümnes, üheteistkümnes kui ka kaheteistkümnes klass. Ülejäänud kümmet kursust oli võimalik õppida mitmes klassis. Kokkuvõtvalt oli kümmet kursust võimalik õppida 10. klassis, kolmeteist 11. klassis ja kaheksat 12. klassis. 12. klassi ebapopulaarsuse üheks põhjuseks võib olla näiteks lõpueksamiteks valmistumine.

Õpilaste arv erinevatel kursustel erines suhteliselt palju. Kõige väiksemal kursusel oli kümme õpilast, samas kõige suuremal kursusel oli 35 õpilast. Aine populaarsuse hindamiseks küsiti õpetaja arvamust kursuse täituvusest. Vastanud õpetajate seast arvas ainult üks, et kursusele ei pääse kõik õpilased, kes seda oleks soovinud. Seega võiks järeldada, et kohtade puudust üldiselt nendel kursustel ei ole.

Küsimustiku käigus uuriti ka naissoost õpilaste osakaalu nendel kursustel. Enamikel kursustel oli naissoost õpilaste osakaal vähemuses. Kolme kursuse puhul oli see alla 10%, kolme puhul vahemikus 20-30%, viie puhul 30-40% ja ühel kursusel 40-50%. Naissoo osakaal oli ülekaalus vaid kolmes koolis, vahemikkudes 50-60%, 60-70% ja 80-90%.

### **3.2. Kursuste sisu**

Küsimustiku vastuste põhjal selgus, et enamikel kursustel tehakse paaris- või rühmatöid ning enamasti tehakse kursuse raames valmis ka suurem iseseisev töö ehk projekt. Kodutööde andmine polnud kursuse raames väga levinud, vaid 20% vastanutest väitis, et kodutöid antakse sageli või pidevalt. Leidus kursusi, kus anti kiirematele õpilastele lisaülesandeid, samas leidus ka kursusi, kus seda töövormi ei kasutatud.

Kaks kolmandikku kursustest olid hindelised ning üks kolmandik arvestuslikud. Kursuse hindamiseks kasutati kõige enam tunnis toimunud tegevuse ja projekti hindamist. Samuti kasutati hinde määramisel kontrolltöid ja mõnel juhul ka kodutöid. Lisaks uuriti, kas leidub erinevusi hindamismeetodites arvestuslike ja hindeliste kursuste puhul. Sellist seost siiski olemasolevate andmete põhjal ei leitud.

Õpetajatel paluti hinnata ka erinevate programmeerimise teemade käsitlemise sügavust. Tabelis 1 on iga teema puhul välja toodud arvamuste populaarsus üle kõikide kursuste. Rohelisega on märgitud teemad, mille käsitlemine oli kõikide kursuste raames suhteliselt põhjalik. Kollasega on

märgitud teemad, mille käsitlus oli kursuste raames erinev, kuid üldpildis olid siiski enamike kursuste poolt mingi tasemeni kaetud. Punasega on värvitud teemad, mille käsitlus oli läbi kõikide kursuste nõrk. Antud teemad ei leidnud ühelgi kursusel põhjalikku käsitlust ning suurel hulgal kursustel ei käsitletud neid üldse.

Tabel 1. Teemade käsitluse põhjalikkus

Teema	Põhjalik käsitlus	Mõõdukas käsitlus	Pealiskaudne käsitlus	Ei käsitleta üldse
Muutujad	7	8	1	0
Valiklausured	9	6	1	0
Tsüklid	7	8	1	0
Alamprogrammid	4	11	1	0
Andmestruktuurid	2	7	7	1
Rekursioon	1	5	6	4
Objektorienteeritud programmeerimine	1	8	5	2
Kasutajaliidese programmeerimine	0	7	6	3
IT valdkonna tutvustus	0	7	6	3
Andmebaasid	0	3	8	5
Programmeerimise ajalugu	0	4	6	6

Lisaks sellele paluti õpetajatel hinnata mõningaid väiteid kursuses osalenud õpilaste kohta. Enamasti nõustuti, et õpilaste huvi programmeerimise vastu kursuse käigus tõuseb, ning enamus õpilasi jäävad kursusega rahule. Nõustuti, et tavaliselt pole enamusel õpilastest kursuse läbimisega probleeme. Arvamused erinesid väite puhul, et kursusel leidub õpilasi, kellele programmeerimine üldse ei sobi. Enamus õpetajaid, kes sellega nõustusid, õpetasid kursust, mis olid mingile õppesuunale kohustuslik, mistõttu võib arvata, et kursust olid sunnitud võtma ka õpilased, kes sellest tegelikult väga huvitatud polnud. Tabeli kõikide vastuste kohta leiab töö lisa 3 alt.



Palusime õpetajatel tuua näiteid ka konkreetsetest ülesannetest, mida õpilased peavad kursuse raames lahendama. Ülesandeid oli mitmeid erinevaid. Näiteks oli lihtsalt sõnastatud arvutöötluse ülesandeid:

Kirjuta funktsioon, mis saab argumendiks täisarvude järjendi, liidab kokku kõik järjendis leiduvad 2-ga ja 3-ga jaguvad elemendid ja tagastab saadud summa.

Samuti tuli õpilastel valmis programmeerida mõni mäng, nagu näiteks linnade põletamine või kuldvillak. Üldiselt tõid õpetajad välja, et ülesannete eesmärgiks oli kasutada kõike eelnevalt õpitut. Rohkem näiteid leiab töö lisast 4.

### **3.3. Kursustel õpetatavad programmeerimiskeeled**

Vastuste põhjal selgus, et kursustel kasutatakse mitmeid erinevaid programmeerimiskeeli. 16 kursuse seas toodi kokku välja 14 erinevat programmeerimiskeelt. Mitmel kursuse kasutati mitut erinevat keelt ning leidus kursusi, kus õpilased said ise mitme keele vahel valida. Levinud praktikaks oli alustada õpetamist mõne lihtsama keelega, mille põhjal oleks hea tutvustada programmeerimise põhitõdesid, ning liikuda sealt edasi mõne keerukama kuid praktilisema keele peale. Tabelis 2 on märgitud kõik keeled, mis vastustes välja toodi, ning mitmel kursusel neid kasutati. Siinkohal tuleb ära märkida, et ühel kursusel võidi kasutada mitut keelt, mis tõttu on ka selle tabeli kogusumma kõvasti üle 16. Näiteks leidus kursus, kus põhitõdede õpetamist alustati Scratchiga, seejärel liiguti edasi Logo peale ning kursuse lõpus tutvustati ka Visual Basicut. Täpsemat kirjeldust keelte kohta saab lugeda antud töö neljandas peatükis.

Tabel 2. Kursustel kasutatud programmeerimiskeeled

Programmeerimiskeel	Kursuste arv
Scratch	7
Python	6
Java	5
JavaScript	5
C/C++	4
PHP	4
Visual Basic	4
Logo	2
NXC/robotika	2
App Inventor	1
C#	1
Perl	1
VBA	1
Pascal	1

Võrreldes saadud tulemusi CSTA [12] uuringu tulemustega on näha, et üldiselt on tabeli eesosas samad keeled. Populaarsuselt esimese kaheksa keele hulgas on vaid üks erinevus – USAs oli populaarseks programmeerimiskeeleks hariduslik keel Alice [18], mis selle küsitluse põhjal Eestis ühelgi kursusel kasutust ei leidnud. Samuti tuleks ära märkida, et vaid 16 vastanuga saadi juba sama värvikas valik keeltest kui CSTA 1246 vastanu puhul.

Keele valikute põhjusteks toodi välja keele sobivust õpetamiseks ja keele tähtsust IT-maailmas. Näiteks põhjendati Java õpetamist järgnevalt:

Sest tegu on väga levinud keelega, millega saab teha paljusid erinevaid programme (s.h Androidi mängu) ja mille järele on ka tööturul nõudlus.

Lisaks sellele oli oluline õpetaja enda kogemus antud keeltega ja õppematerjalide kättesaadavus. Varasem kokkupuude antud keeltega oli õpetajatel enamasti kas ülikooli ajast või iseseisvast õppimisest. Mõni üksik õpetaja oli vastava keele jaoks ka täiendkoolituse läbinud.

Õpetajatel paluti rääkida ka teistest keeltest, mida on varasemalt õpetatud või mida võiks tulevikus õpetada. Üks õpetaja tõi välja, et nad on varasemalt õpetanud ka Javat, kuid see keel oli algajate jaoks liialt keeruline. Kõige rohkem arvati, et tulevikus võiks õpetada ka C ja Pythoni

programmeerimiskeelt. C keele puhul toodi põhjuseks selle kasutus robotika valdkonnas ja Pythoni puhul tema populaarsuse tõus hariduslikus valdkonnas.

### **3.4. Kasutatavad õppematerjalid**

Õppematerjalide puhul oli tegemist peaaegu sama värvika valikuga, kui programmeerimiskeelte puhul. Paljud õpetajad kasutasid kursustel segu enda koostatud ja veebis kättesaadavatest materjalidest. Kõige parema õppematerjalide allikana toodi välja Progetiigri veebilehekülg [19]. Progetiigri programmi eesmärgiks on õppijate tehnoloogilise kirjaoskuse ja digipädevuse tõstmine. See hõlmab endast õppematerjalide koostamist, kohendamist ja tõlkimist, õpetajate ja juhendajate koolitamist, ning teavitus- ja populariseerimistegevusest. Progetiigri veebilehelt leiab õppematerjale mitmele eespool välja toodud keelele, näiteks Scratch, App Inventor, Python ja JavaScript.

“Rakenduste loomise ja programmeerimise alused” õppekomplekti kasutati kolme kursuse raames. Arvestades, et tegu on Euroopa Sotsiaalfondi poolt rahastatud ja TTÜ töörühma poolt koostatud nõ ametlike õppematerjaliga programmeerimise õpetamiseks Eesti koolides, on see arv üsnagi väike. Portaalil koolielu.ee on valik neljanda kooliastme informaatika kursuste jaoks suhteliselt väike [20]. Leidub mõningaid konsepte ja harjutusi vaid teatud keelte, nagu Java ja JavaScript ja Python, kohta. Välismaiste õppematerjalide allikatena toodi välja veel mõned leheküljed, nagu näiteks *php.net*, *w3schools.com* ja *codecademy.com*, mis pakuvad õppematerjale veebiprogrammeerimise suunale. Kursustel kasutati ka mugandatud kujul materjale nii Tartu kui ka Tallinna ülikoolidest.

### **3.5. Võimalik koostöö erinevate organisatsioonide vahel**

Antud bakalaureusetöö üheks oluliseks eesmärgiks oli tuua välja kohti, mida saaks programmeerimise õpetamisel parandada. Õpetajatelt päriti, millist tuge ootaksid nemad nii riigi kui ka ülikoolide poolt, et programmeerimise õpetamist Eestis parandada. Vastusevariantidena pakuti välja ka võimalikud koostöövormid, ning paluti õpetajatel hinnata huvi nende vastu. Koostöövormid ja huvi nende vastu on välja toodud töö lisas 5. Välja pakutud variantide puhul on tegemist ideedega, mida on juba osaliselt rakendatud või mille vajaduse üle on varasemalt mõeldud.

Tihedamast koostööst kõrgkoolidega olid õpetajad enamasti väga huvitatud. Kõrgkoolidelt oodati ühiste õppematerjalide koostamist ja täiendkoolituste korraldamist. Kuigi uute ühiste õppematerjalide loomise poolt olid kõik vastanud õpetajad, ei peetud ka praegu kasutatavate õppematerjalide kvaliteeti halvaks.

Küsimustikus oli välja pakutud erinevaid võimalusi IT maastikuga tutvumiseks. 12 õpetajat 14st oli huvitatud ekskursioonide vastu IT firmadesse ja kõrgkoolidesse. IT maailma tutvustava dokumentaalfilmi vastu tundis huvi 10 õpetajat.

Küsimustiku lõpus küsiti õpetajatelt veel täiendavalt, mida nende arvates tuleks teha, et parandada programmeerimise õpetamist Eesti koolides. Kaks õpetajat arvasid, et Eesti riik peaks programmeerimise õpetamisega rohkem tegelema, sest puudus on tsentraliseeritud juhtimisest. Kolm õpetajat arvas, et õpilasi ja õpetajaid on vaja julgustada ja motiveerida programmeerimisega tegelema. Samuti arvas üks õpetaja, et programmeerimise õpetamisega peaks koolides alustama juba palju varem.

## 4. Ülevaade programmeerimiskeeltest

Küsimustiku vastustest selgus, et kursustel kasutatakse programmeerimise õpetamiseks mitmeid erinevaid keeli. Selles peatükis jagatakse need keeled kahte gruppi. Haridusele orienteeritud keelteks loetakse keeli, mille loomise eesmärgiks oli õpetada programmeerimist. Tööstusele orienteeritud programmeerimiskeelte all mõeldakse keeli, mille eesmärgiks on olnud tarkvaralahenduste loomine. Käesolevas peatükis tehakse nendele keeltele lühitutvustus ning hinnatakse keelte sobivust esimeseks õpitavaks programmeerimiskeeleks. Samuti tuuakse Karl Peedoski bakalaureusetöö “Eesti kõrgkoolide programmeerimise algkursused” [21] põhjal välja, mis keeli kasutavad erinevad ülikoolid programmeerimise algkursustel. Iga keele puhul märgitakse ka tema koht TIOBE Programming Community edetabelis [22]. TIOBE Programming community indeks, mille põhjal iga keele koht edetabelis arvutatakse, põhineb keele kasutamise populaarsuses. Indeksi arvutamisel analüüsitakse inseneride, pakutavate kursuste ja toetavate üksuste osakaalu maailmas. Antud indeksi puhul tuleb arvestada, et haridusele orienteeritud keelte puhul on tegemist suhteliselt vähetähtsa näitajaga. Haridusele orienteeritud keeltele puudub tööstuslik väärtus, mistõttu pole antud keelte populaarsus võrreldav tööstusele orienteeritud keeltega.

### 4.1. Haridusele orienteeritud programmeerimiskeeled

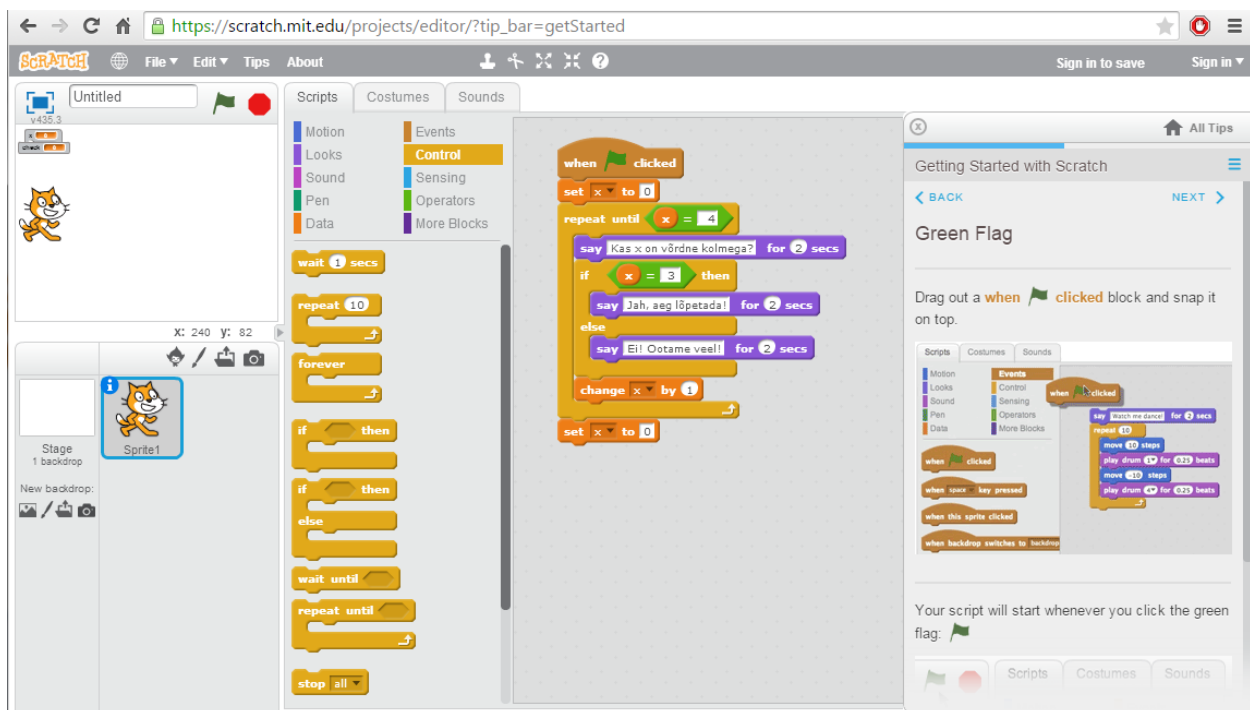
Haridusele orienteeritud programmeerimiskeelteks nimetatakse keeli, mis on loodud spetsiaalselt õpetamise otstarbel. Artikli “Choosing the First Educational Programming Language” [23] kohaselt peaks hariduslikud keeled olema selge loogikaga ning sisaldama kõiki programmeerimise kontseptsioone. Haridusliku keele õppimine peaks lihtsustama tööstusele orienteeritud keele õppimist.

Haridusele orienteeritud programmeerimiskeeli on nii tekstilisi (*text-based*) kui ka visuaalseid (*graphical/visual*) [24]. Visuaalsete programmeerimiskeelte puhul luuakse programme ette antud graafiliste elementidega manipuleerimisel, samas kui tekstiliste programmeerimiskeelte puhul kirjutatakse programmi elemendid välja tekstiliselt. Kuna haridusele orienteeritud programmeerimiskeeled leiavad kasutust enamasti vaid koolides, ei tohiks nende madalaid kohti TIOBE indeksi edetabelis üle tähtsustada.

Visuaalsetest programmeerimiskeeltest kõige tuntum on Scratch [25]. Scratch on 2006. aastal avaldatud vabavaraline visuaalne keel, mis loodi Massachusettsi Tehnoloogia Instituudis (MIT) spetsiaalselt programmeerimise õpetamiseks algajatele. Scratch on kasutusel üle 150 riigis ning saadaval rohkem kui 40 erinevas keeles, sealhulgas ka eesti ja vene keeles. [26]

Scratch on mõeldud inimestele, kellel puudub varasem kokkupuude programmeerimisega. Scratchi kasutajaliidese komponendid asuvad kõik ühes aknas, mis tagab lihtsa navigeerimise ning hoiab kõik vajaliku alati nähtaval. Ekraanitõmmisel 1 on näha Scratchi programmeerimise keskkonda. Kasutaja saab elementidele klikkides kohese tagasiside toimuvast ning võib isegi programmi töötamise ajal seda täiendada. Programmi töötamise ajal ümbritsetakse hetkel töötav element valge joonega, mis tagab kasutajale hea ülevaate, mis element igal töös on. Scratchis puuduvad ka veateated, kuna kasutajal võimaldatakse omavahel sobitada ainult kokku sobivaid elemente. [26]

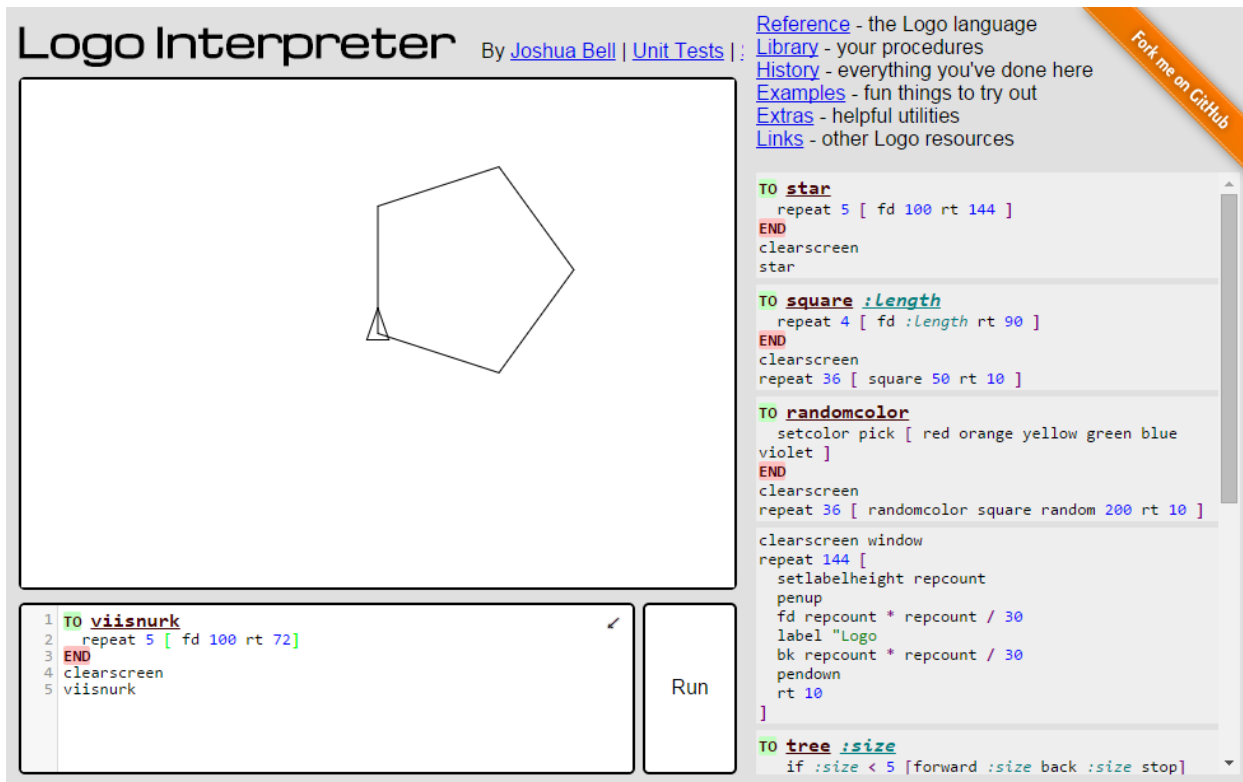
Scratchi õpetati kokku seitsmel kursusel 16st. Scratch on TIOBE Programming community indeksi põhjal populaarsuselt 33. keel maailmas.



Ekraanitõmmis 1 - Scratchis koostatud programm, mis muudab muutuja x väärtust nullist kolmeni.

Suhteliselt sarnane Scratchile on MIT App Inventor [27]. Ka see programmeerimiskeel on vabavaraline visuaalne keel, mis loodi MIT poolt. Kasutajaliidese juures on lähtunud samadest põhimõtetest mis Scratchi puhul. Erinevuseks on, et MIT App Inventoriga saab luua rakendusi Androidi tarkvaraga seadmetele. Artiklis “App Inventor and Real-World Motivation” [28] tuuakse välja, et App Inventori suurim eelis on see, et õpilased saavad luua rakendusi nendele kõige väärtuslikumale seadmele, telefonile. See muudab programmeerimise õpilaste jaoks palju päriselulisemaks, mis lisab motivatsiooni. Küsitluse vastuste põhjal selgus, et MIT App Inventorit kasutati ühel kursusel. Kuna tegemist on suhteliselt uue ja haridusele orienteeritud keelega, ei mahtunud see TIOBE Programming community edetabelisse.

Pikema ajalooga on 1967. aastal loodud Logo [29] programmeerimiskeel. Logo loomise eesmärgiks oli õpetada lastele programmeerimist. Logo on tuntud oma kilpkonnagraafika poolest. “Kilpkonn” oli justkui robot, mida juhitakse kirjutatud koodi abil, moodustades nii graafilisi kujundeid. Sarnast kilpkonnagraafikat on kasutatud ka teistes programmeerimiskeeltes, nagu näiteks Python. Lisaks allalaetavale programmeerimiskeskonnale MSWLogo [30] on Logo jaoks tehtud ka mitmeid veebis töötavaid interpretaatorid, nagu näiteks Logo Interpreter [31] ja TurtleAcademy [32] mis teevad Logoga tutvumise väga lihtsaks. Ekraanitõmmisel 2 on näha Logo Interpreteriga tehtud lihtne programm, mis joonistab ekraanile viisnurka. Logo programmeerimiskeelt kasutati vastanute seas kahel kursusel. Logo on TIOBE Programming community indeksi põhjal populaarsuselt 26. keel.



Ekraanitõmmis 2 - Logo keeles kirjutatud kood, mis tekitab kilpkonnagraafika abil ekraanile viisnurka.

NXC (Not eXactly C) [33] on loodud Lego Mindstormi programmeeritava robotika komplekti NXT programmeerimise jaoks. NXC kasutamist mainis küsimustike vastustes üks õpetaja, kui robotika tunde oli tegelikult rohkem, nii et võib eeldada, et antud keel leidis kasutust veel mitmetes koolides. Kuna NXC puhul on tegemist väga kitsa valdkonna keelega, ei ole seda TIOBE edetabelis välja toodud.

Pascal [34] loodi algselt haridusliku keelena. Keel pidi olema hästi loetav ja arendama algoritmilist mõtlemist. Pascalit peeti pikka aega parimaks hariduslikuks keeleks. [23] Turbo Pascal [35] on 1983. aastal kasutusele tulnud tarkvaraarendussüsteem, mis ühendab endas kompilaatori integreeritud arenduskeskkonna ja Pascali keele. Turbo Pascali keskkond on praeguseks suhteliselt aegunud, mistõttu on langenud ka tema populaarsus [23]. Turbo Pascali toel õpetati programmeerimist ühes koolis. TIOBE indeksi põhjal on Pascali keel 20. kohal.



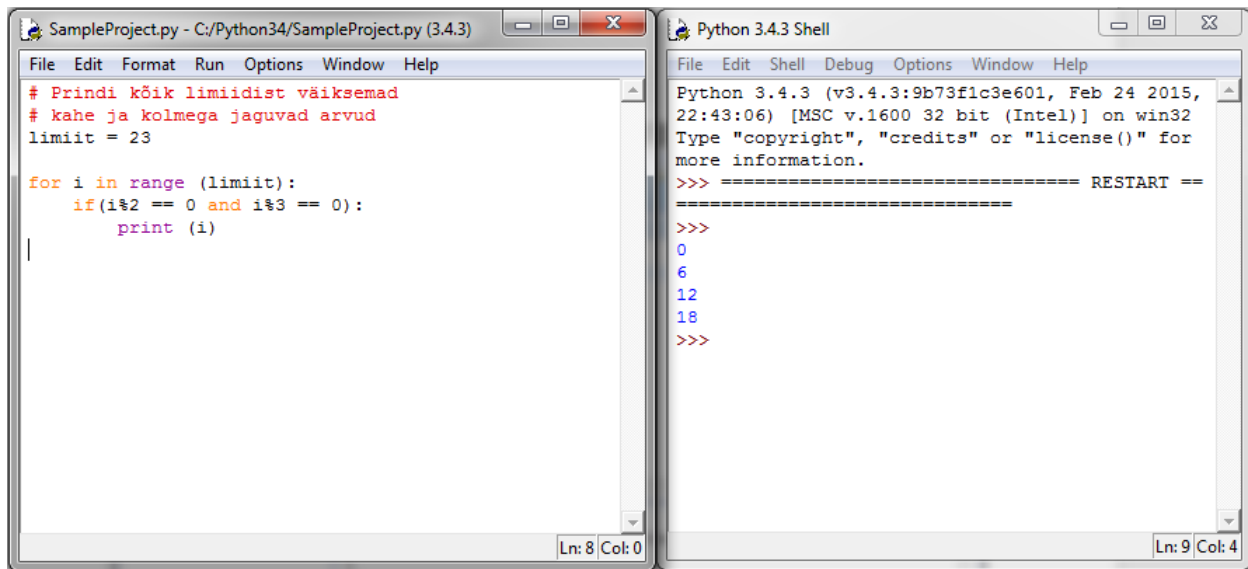
## 4.2. Tööstusele orienteeritud programmeerimiskeeled

Erinevalt haridusele orienteeritud programmeerimiskeeltest on tööstusele orienteeritud keelte eesmärk olla piisavalt võimekad keerukate programmide ja süsteemide loomiseks. Need keeled on tavaliselt keerukama ülesehitusega ning nende õppimine võtab rohkem aega. Samas on nende keelte oskamisel suurem praktiline väärtus, kuna kõik suuremad tarkvara lahendused kirjutatakse just nende abil.

Python [36] on 1991. aastal kasutusele tulnud kõrgtasemeline keel, mis loodi algselt skriptimise eesmärgil. Artiklis “Why Complicate Things? Introducing Programming in High School Using Python” [37] tuuakse välja Pythoni mitmeid eeliseid:

- Pythoni kood on kergesti loetav, mistõttu on lihtsate programmide lugemine ja kirjutamine selles mugav.
- Python sunnib programmeerijaid kasutama taandeid, mis on hästi struktureeritud koodi aluseks.
- Pythoni jaoks on veebis olemas ka suur hulk õpetusi, raamatuid, harjutusi ja dokumentatsiooni.

Pythoni näidiskood on väljatoodud ekraanitõmmisel 3. Kuna erinevalt Scratchile ja Logole, on Pythonil ka reaalne tööstuslik väärtus, võib teda pidada üsnagi heaks esimeseks programmeerimiskeeleks. Pythonit kasutati kuuel vastanud kursusel. Samuti kasutatakse Pythonit mitme algkursuse raames nii Tartu Ülikoolis kui ka Tallinna Tehnikaülikoolis. TIOBEi edetabelis asub Python kaheksandal kohal.

The image shows two side-by-side windows from a Python IDE. The left window, titled 'SampleProject.py - C:/Python34/SampleProject.py (3.4.3)', contains the following Python code:

```
# Prindi kõik limiidist väiksemad
# kahe ja kolmega jaguvad arvud
limit = 23

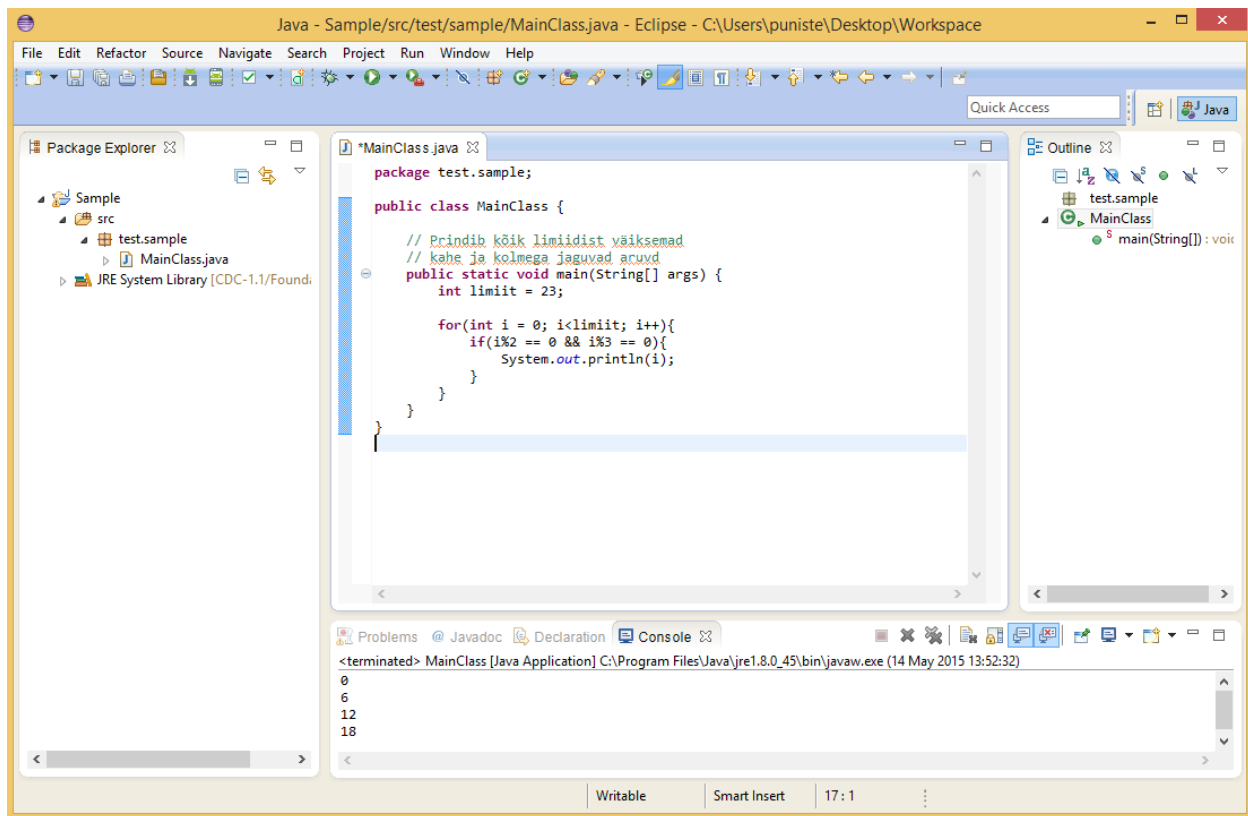
for i in range (limit):
    if(i%2 == 0 and i%3 == 0):
        print (i)
```

The right window, titled 'Python 3.4.3 Shell', shows the execution output:

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015,
22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for
more information.
>>> ===== RESTART =====
>>>
0
6
12
18
>>>
```

Ekraanitõmmis 3 - Pythoni keeles kirjutatud programm, mis kirjutab ekraanile kõik ette antud piirangust väiksemad kahe ja kolmega jaguvad arvud.

Java [38] on maailmas üks kõige levinuimaid objektorienteeritud programmeerimiskeeli. Java 1.0 esimene ametlik väljalase oli 1995. aastal. Java üheks suureks eeliseks on tema sõltumatus platvormist – korra kompileeritud programmi saab avada kõikidel Javat toetavatel platvormidel ilma ümberkompileerimata. Samas on Java suhteliselt keeruline keel. Artiklis “Java as First Programming Language: A Critical Evaluation” [39] uuritakse Java erinevusi keelest C++ õpetamise aspektist lähtudes. Selgus, et sarnasel keelele C++ oli ka Java puhul 30% varasema kogemusega õpilastel raskusi korrektse süntaksi kirjutamisega. Raskusi valmistasid lihtsad tsüklid, järjendid ja valiklauseid. Ka küsimustiku vastustes märkis üks õpetaja, et nende koolis lõpetati Java õpetamine, kuna see oli algajate jaoks liiga raske. Vaatamata sellele kasutati Javat siiski viiel gümnaasiumi kursusel. Samuti kasutatakse Javat mitmel kõrgkooli algkursusel, nagu näiteks Lennuakadeemias, Arvutikolledžis ja Eesti Infotehnoloogia Kolledžis. Ekraanitõmmis 4 näitab lihtsat Javas kirjutatud programmi, mis kirjutab ekraanile kõik limiidist väiksemad kahe ja kolmega jaguvad arvud. Aprillikuu seisuga on Java TIOBE Programming community indeksi põhjal üle pooleteise aasta taaskord esimesel koha. Põhjuseks arvatakse olevat mobiilse platvormi Android populaarsuse kasv, millele kirjutatud rakendused on justnimelt Java programmeerimiskeeles.



Ekraanitõmmis 4 - Java keeles kirjutatud programm, mis kirjutab ekraanile kõik limiidist väiksemad kahe ja kolmega jaguvad arvud.

Visual Basic (VB) [40] on Microsofti poolt loodud üldotstarbeline objektorienteeritud keel. VB on algajatele suhteliselt lihtsasti õpitav jäädes samas võimekaks keeleks professionaalide jaoks [40]. Kokku oli küsimustikule vastanute seas 4 kursust, kes kasutasid programmeerimise õpetamisel Visual Basicut. Karl Peedoski bakalaureusetööst selgus, et Visual Basicut kasutatakse ka Tallinna Tehnikaülikoolis programmeerimise aluste kursusel. Visual Basic asub TIOBE Programming community indeksi põhjal üheksandal koha. Visual Basicu kõrvalarendus on Visual Basic for Applications (VBA), mida kasutatakse skriptide kirjutamiseks erinevate Microsofti rakenduste, nagu näiteks Excel ja Word, jaoks. VBA oli kasutusel ühel kursusel.

C [41] on laiaotstarbeline keel, mis loodi 1970. aastate alguses. C põhjal on arendatud ka mitmeid teisi keeli. C++ [41] on C järeltulija, mis tõi keelde sisse klassid. Tänapäeval on C ja C++ ühed kõige enam kasutust leidvad keeled, ning neid saab programmeerida peaaegu igal operatsioonisüsteemil. Samas on C ja C++ puhul tegemist ühtede kõige keerulisemate keeltega, mida on võrreldes teiste keeltega suhteliselt raske õppida [39]. Küsimustiku vastuste põhjal

õpetati C võid C++ keelt kokku neljal kursusel. C keelt kasutatakse ka Tallinna Tehnikaülikoolis “Programmeerimine I” aine raames. TIOBE indeksi järgi asub C keel teisel ja C++ kolmandal kohal.

C# (loetakse C Sharp) [42] on Microsofti poolt arendatud objektorienteeritud programmeerimiskeel, mille nimi on tuletatud tema algsest sarnasusest C keelega. C# on disainilt lihtne, moodne ja laialdase kasutusega. C# õpetati küsitlusele vastanute seas ühel kursusel, ning C#i kasutatakse ka Eesti Ettevõtluskõrgkool Mainori sissejuhataval kursusel “Programmeerimise alused”. TIOBE tabelis on C# viiendal kohal.

Kahe veebiprogrammeerimise keelena toodi välja PHP [43] ja JavaScript [44]. PHP on 1995. aastal avaldatud skriptimiskeel, mida kasutatakse enamasti serveripoolses osas. PHP koodi on lihtne siduda HTMLga, mis on teinud sellest ühe kõige enamkasutatava keele dünaamiliste veebilehtede loomisel. 2013. aasta seisuga oli PHPd kasutatud üle 240 miljonil veebilehel. PHP süntaks on põhiliselt laenatud Java, C ja Perli keeltest. Küsitluse tulemuste järgi õpetati PHPd neljal kursusel ja ta on TIOBE indeksi põhjal populaarsuselt seitsmes keel. PHPd kasutatakse ka Tallinna Tehnikaülikoolis ja Lääne-Viru Rakenduskõrgkoolis veebiprogrammeerimise algkursustel.

Teiseks kasutatud veebiprogrammeerimise keeleks oli JavaScript. Sarnaselt PHPle, on JavaScript 1995. aastal avaldatud skriptimiskeel, mis on loodud veebilehtede dünaamilisuse implementeerimiseks. Nimetusest hoolimata ei ole JavaScript Javaga otseselt seotud. JavaScripti õpetati viiel kursusel ning ta on TIOBEi indeksi põhjal kuuendal kohal. Veebiprogrammeerimise üldiseks eeliseks toodi tulemuse lihtne jagamine ja rakenduvus, näiteks mobiilsetel platvormidel.

Perl [45] on 1987. aastal avaldatud üldotstarbeline programmeerimiskeel, mida kasutatakse mitmes eri valdkonnas. Perl, olles siiski vähem populaarsem kui PHP ja JavaScript, on üks olulisemaid veebiarenduse keeli. Keele võimsus ja lihtsus on toonud talle suure kasutajaskonna. Perli kasutatakse veel näiteks süsteemiadministratsiooni ja automaatse testimise juures. Küsitlusele vastanutest õpetati Perli vaid ühe kursuse raames. TIOBE indeksi järgi on Perl 12. kohal.

## 5. Kokkuvõte

Käesoleva bakalaureusetöö raames uuriti programmeerimise õpetamist Eesti gümnaasiumides. Andmeid koguti küsimustiku kaudu, mis saadeti koolidele, kus EHISi riikliku registri põhjal õpetati kursust, mille nimetus sisaldas sõna “programmeerimine”. Küsimustiku koostamise eesmärk oli saada täpset infot Eesti koolides õpetatavate programmeerimiskursuste sisu kohta. Vastuseid saadi 16 kursuse kohta. Kuna 16 kursust ei ole kindlasti piisav hulk, mille põhjal saaks teha kindlaid järeldusi üldise seisuga, toome siinkohas välja ka mõningaid mõtteid edasisteks uuringuteks.

Vastustest selgus, et Eesti üldhariduskoolides õpetatavad kursused erinevad üksteisest märgatavalt. Kursuste ülesehitusest ja sisust oli näha, et kursustel puudub riigipoolne reguleerimine. Üldiselt oli tegemist valikkursustega, mis olid mõnes koolis teatud õppesuuna kohustuslikud. Kursustel leidsid kasutust mitmed erinevad programmeerimiskeeled, mida ilmestab fakt, et 16 vastanu kohta mainiti kokku 14 erinevat programmeerimiskeelt. Programmeerimiskeele valiku puhul pidasid õpetajad kõige tähtsamaks sobivust õpetamiseks ja tähtsust IT maailmas. See mõtteviis tuli välja ka keelte populaarsuses – edetabeli tipus olid Scratch, Python, Java ja JavaScript. Samas tundus keelte erinevuse põhjuseks olevat õpetajate erinevad varasemad kogemused ja eelistused. Kas oletust, et kursustel kasutatav programmeerimiskeel on tingitud õpetaja eelistustest saaks laiendada ka kõikide Eestis õpetatavate programmeerimiskursuste peale? Samuti oleks huvitav teada, kas kasutatud programmeerimiskeel on mingi seos näiteks tulemuste või õpilaste rahuloluga.

Kursuste sisu oli suhteliselt erinev. Põhjalikku käsitlust leidsid enamikes koolides küll programmeerimise põhiteemad, nagu muutujad, valiklause, tsüklid ja alamprogrammid, kuid teiste teemade käsitluse sügavus võis erineda kursuseti päris palju. Enamikel kursustel oli tahaplaanile jäänud spetsiifilisemad andmebaaside ja kasutajaliideste loomise ning üldisemad programmeerimise ajaloo ja IT valdkonna tutvustuse teemad. Õppematerjalide puhul leidsid mitmeid erinevaid lähenemisviise. Kasutati ProgeTiigri, koolielu.ee-s saadaval olevaid, isekoostatud ja ülikooli mugandatud materjale. Üldiselt ei kurdetud praeguste õppematerjalide üle, kuid samas oodati ülikoolide poolt uusi ja ühiseid õppematerjale.

Küsimustiku raames taheti teada ka koolide huvi koostöö vastu ülikoolide ja IT tööstusega. Selgus, et koostööst olid huvitatud peaaegu kõik koolid. Populaarsemateks vastusevariantideks olid külastused IT firmadesse ja juba eelnevalt mainitud ühiste õppematerjalide koostamine. Samuti näidati huvi üles ekskursioonide vastu IT-d õpetavatesse kõrgkoolidesse ja IT maailma tutvustava dokumentaalfilm kohta. Edasistes uuringutes tasuks keskenduda täpsemalt mõnele väljapakutud variandile ning ka reaalseid samme selle edendamiseks teha. Samuti võiks tulevikus uurida IT firmade soovi ja valmidust sellise koostöö läbiviimiseks.

Lisaks eelnevalt mainitud edasise uurimustöö mõtetele, võib vastavalt huvile leida veel huvitavaid aspekte, mida suurema valimi peal uurida. Näiteks võiks uurida töövormide või käsitluses olevate temade mõju õpilastele ja nende tulemustele.

## 6. Kasutatud kirjandus

- [1] Riigi Teataja - Põhikooli ja gümnaasiumi riiklik õppekava (2002)  
<https://www.riigiteataja.ee/akt/12888846> (Vaadatud 5.05.2015)
- [2] Riigi Teataja - Põhikooli ja gümnaasiumi riiklik õppekava (2010)  
<https://www.riigiteataja.ee/akt/13272925> (Vaadatud 4.05.2015)
- [3] Rakenduste loomise ja programmeerimise alused õppekomplekt  
<http://koolielu.ee/waramu/view/1-5171dab3-4720-4363-a956-b7602de68489> (Vaadatud 4.05.2015)
- [4] Õpetajate Leht - Nõo reaalhariduse esimene poolsajand  
<http://opleht.ee/19371-noo-reaalhariduse-esimene-poolsajand/>
- [5] Projekti „Kontseptuaalne raamistik suurendamiseks ühiskonna pühendumist IKTsse: IKTga seonduvaid karjäärivalikuid motiveerivad ning IKT rakendamiseks ja arendamiseks vajalikku kompetentsust arendavad lähenemised üld- ja kõrghariduses”  
<https://sisu.ut.ee/ikt> (Vaadatud 4.05.2015)
- [6] ] P. Hubwieser, M. Armoni, T. Brinda, V. Dagiene, I. Diethelm, M. N. Giannakos, M. Knobelsdorf, J. Magenheimer, R. Mittermeir. ja S. Schubert. Computer science/informatics in secondary education.  
<http://dl.acm.org/citation.cfm?id=2078859>
- [7] P. Hubwieser, M. Armoni, M. N. Giannakos ja R. T. Mittermeir. Perspectives and visions of computer science education in primary and secondary (k-12) schools.  
<http://dl.acm.org/citation.cfm?id=2602482>
- [8] E. Khenner ja I. Semakin. School Subject Informatics (Computer Science) in Russia: Educational Relevant Areas  
<http://dl.acm.org/citation.cfm?id=2602489>
- [9] L. Rolandsson ja I.-B. Skogh. Programming in school: Look back to move forward. Trans. Comput. Educ., 14(2):12:1–12:25, June 2014.  
<http://dl.acm.org/citation.cfm?id=2602487>
- [10] G.-L. Baron, B. Drot-Delange, M. Grandbastien ja F. Tort. Education in French Secondary Schools: Historical and Didactical Perspectives  
<http://dl.acm.org/citation.cfm?id=2602486>
- [11] J. Gal-Ezer ja C. Stephenson. A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States  
<http://dl.acm.org/citation.cfm?id=2642651.2602483&coll=DL&dl=GUIDE&CFID=454665033&CFTOKEN=15120117>

- [12] Organisatsiooni Computes Science Teacher Association veebilehekül  
<http://csta.acm.org/> (Vaadatud 03.04.2015)
- [13] N. C. C. Brown, S. Sentance, T. Crick ja S. Humphreys. Restart: The Resurgence of Computer Science in UK Schools.  
<http://dl.acm.org/citation>.
- [14] Selts Royal Society veebilehekül  
<https://royalsociety.org/> (Vaadatud 03.04.2015)
- [15] Organisatsioon Computing At Schools veebilehekül  
<http://www.computingatschool.org.uk/> (Vaadatud 03.04.2015)
- [16] T. Bell, P. Andreae, ja L. Lambert. Computer science in New Zealand high schools.  
<http://dl.acm.org/citation.cfm?id=2602485>
- [17] Eesti Hariduse Infosüsteemi veebilehekül  
<http://www.ehis.ee/> (Vaadatud 03.04.2015)
- [18] Programmeerimiskeele Alice veebilehekül  
[http://www.alice.org/index.php?page=what\\_is\\_alice/what\\_is\\_alice](http://www.alice.org/index.php?page=what_is_alice/what_is_alice) (Vaadatud 03.04.2015)
- [19] ProgeTiiger veebilehekül  
<http://progetiiger.ee/> (Vaadatud 03.04.2015)
- [20] Koolielu õppematerjalid informaatika IV kooliastmele  
<http://koolielu.ee/waramu/search/sort/created/curriculumSubject/61197430> (Vaadatud 12.05.2015)
- [21] K. Peedosk. Eesti kõrgkoolide programmeerimise algkursused  
[http://comserv.cs.ut.ee/forms/ati\\_report/downloader.php?file=403F643F5D755E7A2D83603FB6503DEF57855B0C](http://comserv.cs.ut.ee/forms/ati_report/downloader.php?file=403F643F5D755E7A2D83603FB6503DEF57855B0C)
- [22] TIOBE programmeerimiskogukonna ideksite edetabel  
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (Vaadatud 10.04.2015)
- [23] V. Kruglyk ja M. Lvov. Choosing the First Educational Programming Language  
<http://ceur-ws.org/Vol-848/ICTERI-2012-CEUR-WS-paper-37-p-188-198.pdf>
- [24] From graphical to text-based programming languages  
<http://www.quickstartcomputing.org/secondary/Resources/section4/From%20graphical%20to%20text-based%20programming%20languages.pdf> (Vaadatud 12.05.2015)
- [25] Programmeerimiskeele Scratch veebilehekül  
<https://scratch.mit.edu/> (Vaadatud 03.04.2015)



[26] J. Maloney, M. Resnick, N. Rusk, B. Silverman ja E. Eastmond. The Scratch Programming Language and Environment

<http://dl.acm.org/citation.cfm?id=1868363>

[27] Programmeerimiskeele MIT App Inventor veebilehekülg

<http://appinventor.mit.edu/explore/> (Vaadatud 03.04.2015)

[28] D. Wolber. App Inventor and Real-World Motivation

<http://dl.acm.org/citation.cfm?id=1953329&prelayout=flat>

[29] Programmeerimiskeele Logo veebilehekülg

<http://el.media.mit.edu/logo-foundation/logo/programming.html> (Vaadatud 03.04.2015)

[30] MSWLogo veebilehekülg

<http://mswlogo.en.softonic.com/> (Vaadatud 12.05.2015)

[31] Logo Interpreter veebilehekülg

<http://www.calormen.com/jslogo/> (Vaadatud 12.05.2015)

[32] Turtle Academy veebilehekülg

<http://turtleacademy.com/next/playground/en> (Vaadatud 12.05.2015)

[33] Programmeerimiskeele NXC (Not eXactly C) veebilehekülg

<http://bricxcc.sourceforge.net/nbc/> (Vaadatud 03.04.2015)

[34] Programmeerimiskeele Pascal veebilehekülg

<http://www.pascal-programming.info/index.php> (Vaadatud 12.05.2015)

[35] Tarkvaraarendussüsteemi Turbo Pascal veebilehekülg

<http://turbopascal.org/turbo-pascal-download> (Vaadatud 12.05.2015)

[36] Programmeerimiskeele Python veebilehekülg

<https://www.python.org/> (Vaadatud 03.04.2015)

[37] L. Grandell, M. Peltomäki, R.-J. Back ja T. Salakoski. Why Complicate Things? Introducing Programming in High School Using Python

<http://dl.acm.org/citation.cfm?id=1151880>

[38] Programmeerimiskeele Java veebilehekülg

<https://java.com/en/> (Vaadatud 03.04.2015)

[39] S. Hadjerrouit. Java as First Programming Language: A Critical Evaluation

<http://dl.acm.org/citation.cfm?id=292440>

[40] Programmeerimiskeele Visual Basic veebilehekülg  
<https://msdn.microsoft.com/en-us/library/2x7h1hfk.aspx> (Vaadatud 03.04.2015)

[41] Programmeerimiskeelte C ja C++ veebilehekülg  
<http://www.cprogramming.com/> (Vaadatud 12.05.2015)

[42] Programmeerimiskeele C# veebilehekülg  
<https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> (Vaadatud 12.05.2015)

[43] Programmeerimiskeele PHP veebilehekülg  
<http://php.net/> (Vaadatud 12.05.2015)

[44] Programmeerimiskeele JavaScript veebilehekülg  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript> (Vaadatud 12.05.2015)

[45] Programmeerimiskeele Perl veebilehekülg  
<https://www.perl.org/> (Vaadatud 12.05.2015)

# Lisad

## I. Programmeerimise kursuste küsimustik

### Programmeerimise kursuste küsimustik

Lugupeetud õpetaja!

Arvutiõpetuse/informaatika ja sealhulgas programmeerimise õpetamine on Eestis tsentraalselt suhteliselt nõrgalt reglementeeritud ja koolidel on suur vabadus ise otsustada, mida ja kuidas nad täpsemalt teevad. Samas oleks ülevaade koolides toimuvast vajalik, et pikemas perspektiivis mõistlikumal moel materjale, koolitusi, koostööd jm korraldada.

Sellise ülevaate saamiseks vajame Teie abi. See küsitlus on ka osa Tartu Ülikooli informaatikaüliõpilase Siim Puniste bakalaureusetööst "Programmeerimine Eesti koolides". Bakalaureusetöös esitatakse andmed konkreetse kooliga seostamata.

Kui Teie koolis on mitu programmeerimist õpetavat kursust, siis palume küsimustik täita iga kursuse kohta eraldi! Teistel vastamiskordadel pole vaja vastata küsimustele, mille vastus ainest olenevalt ei muutu!

Küsimustele vastamine võtab aega umbes 15-20 minutit.

Olge Te tänatud koostöö eest!

Täpsustavalt võite julgesti kirjutada siimpuniste@gmail.com (Siim Puniste) või eno.tonisson@ut.ee (Eno Tõnisson).

Mis on Teie kooli nimi?

Mis on Teie koolis õpetatava programmeerimise kursuse nimi?

Kas kursus on kõigile gümnaasiumi õpilastele kohustuslik?

- Kohustuslik kõigile
- Kohustuslik ainult teatud õpilastele
- Valikuline kõigile

Kui valisite, et kursus on kohustuslik ainult teatud õpilastele, siis palun täpsustage, kellele.

Kui suur on kursuse maht akadeemilistes tundides?

Mitmenda(te)s klassi(de)s seda kursust õpetatakse?

- 10. klass
- 11. klass
- 12. klass

Ligikaudu mitu õpilast keskmiselt sellel kursusel igal õppeaastal osaleb?

Ligikaudu kui suur osakaal nendest on naissoost?

- 0 - 10%
- 10 - 20%
- 20 - 30%
- 30 - 40%
- 40 - 50%
- 50 - 60%
- 60 - 70%
- 70 - 80%
- 80 - 90%
- 90 - 100%

Kui sageli seda kursust õpetatakse?

- Mitu korda õppeaasta jooksul
- Kord õppeaastas
- Kord kahe õppeaasta jooksul
- Harvemini
- Other:

Kui kaua on antud kursust Teie koolis õpetatud?

- Õpetatakse esimest aastat
- 2-3 aastat
- 4+ aastat

Kui Teie koolis on olnud varasemalt teisi sarnaseid kursusi, siis palun kirjutage nendest lühidalt.

## Programmeerimiskeeled

Mis keelte baasil programmeerimise õpe antud kursusel toimub?

- C/C++
- Java
- JavaScript
- PHP
- Python
- Scratch
- Visual Basic
- Other:

Põhjendage, miks olete valinud just selle (need) keele(d)?

Milline oli Teie varasem kogemus selle (nende) programmeerimiskeel(t)ega?

Kui te olete varasemalt õpetamisel kasutanud ka mõnda muud keelt, siis millist? Mis põhjusel vahetasite selle keele praeguse keele vastu?

Kas Te arvate, et lähitulevikus võiks kasutada ka mõnda muud keelt programmeerimise õpetamiseks? Miks?

## Õppematerjalid

Milliseid õppematerjale Te antud kursusel kasutate?

Nimi ja autor või viide. (Sh ka enda koostatud materjalid)

Kuidas hindate üldiselt programmeerimise õpetamise õppematerjale?

	Väga hea	Hea	Rahuldav	Halb	Väga halb
Kättesaadavus					
Kvaliteet					
Sobivus					

Kuivõrd olete kasutanud "Rakenduste loomise ja programmeerimise alused" aine jaoks loodud õppekomplekti?

Saadaval siin: <http://koolielu.ee/waramu/view/1-5171dab3-4720-4363-a956-b7602de68489>

- Pidevalt
- Sageli
- Harva
- Mitte kunagi
- Pole sellest õppekomplektist varem kuulnud

## Aine ülesehitus ja hindamine

Kas kursus on hindeline või arvestuslik?

- Hindeline
- Arvestuslik

Mis komponentide põhjal kujuneb kursuse hinne / antakse arvestus?

- Eksam/arvestustöö
- Kontrolltöö(d)
- Suurem iseseisev töö (projekt)
- Tunnikontrollid
- Kodutööd
- Tegevus tunnis
- Other:

Kui tiheidalt leiavad kursusel kasutust järgmised töövormid?

	Pidevalt	Sageli	Harva	Mitte kunagi
Paaristöö/rühmatöö				
Suurem iseseisev töö (projekt)				
Kodutöö				
Lisaülesanded				

Tooge näide ühest ülesandest, mida õpilased kursusel lõpupoole lahendavad. Selgitage ka, mis roll selle ülesandel on.

Näiteks, kas tegemist on kontrolltöö küsimusega, kodutöö küsimusega või tunnis tehtava harjutusega.

Hinnake järgmiste teemade käsitlemise sügavust:

	Põhjalik käsitus	Mõõdukas käsitus	Pealiskaudne ülevaade	Ei käsitleta üldse
Muutujad				
Valiklauseid (if)				
Tsüklid				
Alamprogrammid (funktsioonid, meetodid)				
Andmestruktuurid				
Andmebaasid				
Rekursioon				
Objektorienteeritud programmeerimine				
Kasutajaliidese programmeerimine				
IT valdkonna tutvustus				
Programmeerimise ajalugu				

Hinnake järgmiseid väiteid:

	5 (väga nõus)	4	3	2	1 (ei nõustu üldse)
Enamik õpilasi jääb kursusega rahule					
Enamikel õpilastel pole kursuse läbimisega suuremaid probleeme					
Leidub õpilasi, kes oleks huvitatud programmeerimisega põhjalikumalt edasi tegelema					
Õpilaste huvi programmeerimise vastu kursuse käigus tõuseb					
Kursusele ei pääse kõik õpilased, kes sooviksid					
Kursusel on õpilasi, kellele programmeerimine üldse ei sobi					

## Koostöö tulevikus

Hinnake enda huvi järgmiste võimaluste vastu:

	Väga huvitatud	Huvitatud	Neutraalne	Vähene huvi	Huvi puudub
Koostöö kõrgkoolidega					
Rohkem ühiseid programmeerimise õppematerjale gümnaasiumidele					
Programmeerimise õpetamise täienduskoolitus					
Eesti IT-tööstust tutvustav dokumentaalfilm					
Ekskursioonid kõrgkoolidesse, kus õpetatakse IT erialasid					
Ekskursioonid IT firmadesse					



Kui Te olite huvitatud koostööst ülikoolidega, siis palun kirjeldage, mida saaksid ülikoolid Teie arvates Teile pakkuda?

Millist muud tuge oleks Teie arvates vaja, et muuta programmeerimise õpetamist paremaks?

Kas teie koolis käsitletakse programmeerimist veel mingi(te) õppeaine(te) raames. Millis(t)e?

Kui Teil on veel kommentaare, mis ülal olevate vastuste sisse ei mahtunud, siis palun kirjutage need siia.

## **II. Kursuste nimetused**

- Programmeerimine Logos ja Scratchis
- Programmeerimine Javas
- Veebiprogrammeerimine
- Sissejuhatus programmeerimisse, Veebiprogrammeerimine, Rakenduste loomine C# abil, Robotika.
- Programmeerimine
- Programmeerimise alused I ja II
- Rakenduste loomise ja programmeerimise alused
- Rakenduste loomine
- Majandusarvestus ja programmeerimine
- Programmeerimine
- Programmeerimise algkursus

### III. Õpetajate arvamused

Väide/Hinnang	5 (väga nõus)	4	3	2	1 (ei nõustu üldse)
Enamik õpilasi jääb kursusega rahule	5	9	1	0	0
Enamikel õpilaste pole kursuse läbimisega suuremaid probleeme	2	8	5	0	0
Leidub õpilasi, kes oleks huvitatud programmeerimisega põhjalikumalt edasi tegelema	15	0	0	0	0
Õpilaste huvi programmeerimise vastu kursuse käigus tõuseb	10	3	2	0	0
Kursusele ei pääse kõik õpilased, kes sooviksid	1	0	5	3	5
Kursusel on õpilasi, kellele programmeerimine üldse ei sobi	2	6	5	1	0

## IV. Näidisülesanded

Roll	Ülesanne
Kursuse alguses / kursuse lõpus	Arv arvamise mäng / Kuldvillak
Kontrolltöö	Linnade põletamise kaardimängu kirjutamine.
Kontrolltöö	Kirjuta funktsioon, mis saab argumendiks täisarvude järjendi, liidab kokku kõik 2-ga jaguvad ja kõik 3-ga jaguvad elemendid ja tagastab saadud summa.
Projekt	Õppeteemalise animatsiooni koostamine
Kirjeldus puudus	Javascriptiga kasutades canvasi keskkonda panna ekraanil liikuma pall juhusliku algkiiruse ja suunaga. Põrkamine seintest.
Kirjeldus puudus	Nimede lugemine Kirjutatakse programm, mis kuvab failis antud nimed ekraanile nii, et ees- ja perenimi on ühel real koos.
Kirjeldus puudus	Luu programm, mis kasutab sisendina ultraheli anduri andmeid ja väljundina kahte servomootorit ja peab suutma võimalikult kiiresti tulla välja kaaslaste poolt arvutikastidest ehitatud labürindist (ka hea labürindi ehitamine on ülesanne).
Nii tunnitöös kui ka kontrolltöös	Caesari šiffer · Väljasta sisestatud sõna iga täht tähestikus ühe koha võrra paremale nihutatuna. (kala->lbmb) · Nihuta iga täht tähestikus soovitud arv kohti paremale või vasakule. · Liigutakse ringlevalt tähti koodidega vahemikus 32-127. · Loo vahend tekstifaili tähtede soovitud arvu koodide võrra edasi või tagasi nihutamiseks. · Arvesta nihutamisel ka täpitähti.
MSWLogo viimane töö	MSWLogo viimane töö on teatud värvilise objekti joonistamiseks vajaliku programmi koostamine. Peavad valmis programmeerima mängu.
Kirjeldus puudub	Võetud TÜ kursuse lehelt: <a href="https://courses.cs.ut.ee/2015/OOP/spring/Main/Practice3Harjutused">https://courses.cs.ut.ee/2015/OOP/spring/Main/Practice3Harjutused</a>

## V. Koostöövormid

	Väga huvitatud	Huvitatud	Neutraalne	Vähene huvi	Huvi puudub
Koostöö kõrgkoolidega	6	7	2	0	0
Rohkem ühiseid programmeerimise õppematerjale gümnaasiumidele	9	6	0	0	0
Programmeerimise õpetamise täienduskoolitus	7	5	2	0	1
Eesti IT-tööstust tutvustav dokumentaalfilm	3	8	3	0	1
Ekskursioonid kõrgkoolidesse, kus õpetatakse IT erialasid	6	7	1	1	0
Ekskursioonid IT firmadesse	7	6	1	1	0

## VI. Litsents

### Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Siim Puniste** (sünnikuupäev: 24.08.1991)

(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Eesti gümnaasiumides õpetatavad programmeerimiskursused** (*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, **14/05/2015**