

TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Infotehnoloogia eriala

**Karl Peedosk**

**Eesti kõrgkoolide programmeerimise algkursused**

Bakalaureusetöö (6 EAP)

Juhendaja: Eno Tõnisson

TARTU 2014

# Eesti kõrgkoolide programmeerimise algkursused

## Lühikokkuvõte

Antud töö eesmärgiks on uurida programmeerimise algkursuseid Eesti kõrgkoolides. Vaatluse alla jäi kümnes Eesti kõrgkoolis 16 erinevat programmeerimise algkursust, mida võtab igal aastal üle tuhande tudengi. Igal kursusel otsiti kursuste kohta käivat üldinfot. Kõrgkoolidele esitati ka küsimustikud, millele neil paluti vastata. Vastuste põhjal koostati kursuste kohta lühikokkuvõtted ning vaadeldi käsitletavate kursuste erinevaid aspekte. Projekti „Kontseptuaalne raamistik suurendamiseks ühiskonna pühendumist IKTsse“ raames uuriti Eesti tudengite kokkupuudet programmeerimisega enne kõrgkooli tulemist. Antud bakalaureusetöö jaoks kättesaadavaks tehtud andmeid analüüsiti esimeste programmeerimise kokkupuudete vaatest.

## Võtmesõnad

Programmeerimise algkursused, esimesed programmeerimise kursused, esimesed programmeerimiskeeled

## Introductory programming courses in Estonian colleges and universities

### Abstract

The purpose of the current bachelor's thesis is to examine introductory programming courses in Estonian colleges and universities. 16 courses in 10 different colleges or universities were observed. These courses are attended by more than a thousand students each year. General information about each course was searched for. In addition colleges and universities were asked to answer a questionnaire. Based on the answers of the questionnaire short summaries of the courses were made and different aspects of the courses were observed. In the context of the project „Conceptual framework for increasing society's commitment in ICT” students were asked about their first contacts with programming before attending colleges or universities. The data that were made available for this bachelor's thesis was analyzed in the perspective of first contacts with programming.

### Keywords

Introductory programming courses, first programming courses, first programming languages

## Sisukord

Sissejuhatus.....	5
1. Mitmeid programmeerimise algkursusi käsitlevad uuringud .....	8
2. Kursuste valik ja andmete kogumine .....	10
3. Kursused .....	13
3.1 Eesti Infotehnoloogia Kolledži kursus programmeerimise algkursus Java baasil .....	13
3.2 Eesti Lennuakadeemia kursus programmeerimise alused.....	15
3.3 Eesti Maaülikooli kursus programmeerimine .....	16
3.4 Lääne-Viru Rakenduskõrgkooli kursus veebiprogrammeerimine I.....	18
3.5 Tallinna Tehnikakõrgkooli kursus programmeerimise alused .....	19
3.6 Tallinna Tehnikaülikooli kursus programmeerimine I.....	20
3.7 Tallinna Tehnikaülikooli kursus programmeerimise põhikursus.....	21
3.8 Tallinna Ülikooli kursus programmeerimise alused .....	22
3.9 Tallinna Ülikooli kursus veebiprogrammeerimine .....	23
3.10 Tartu Ülikooli kursus programmeerimine.....	25
3.11 Tartu Ülikooli kursus programmeerimise alused .....	26
4. Tudengite varasemad kogemused kõrgkooli tulles.....	28
5. Kursuste erinevad aspektid .....	31
5.1 Kõrgkoolide ülesed teemad.....	31
5.2 Kursuse koht laiemas pildis.....	32
5.3 Kursus tervikuna .....	33
5.4 Kursuse osad .....	33
Kokkuvõte.....	34
Kasutatud kirjandus .....	35

Lisad.....	38
Lisa 1 – Andmed küsimused koolidele .....	38
Lisa 2 – Programmeerimiskeelte võrdlus.....	40
Litsents.....	43

## Sissejuhatus

Infotehnoloogia (IT) roll on tänapäeva maailmas järjest suurem. Asutused ja ettevõtted kasutavad aina rohkem tööprotsesside kiirendamiseks arvuteid või liiguvad oma äriга internetti. Samas esineb IT sektoris töötajate puudus vaatamata sellele, et kõrgkoolide arvutiteaduste erialadele vastuvõtu arvud on üsna suured. Näiteks Tartu Ülikooli 2014/2015. õppeaastal on planeeritud vastu võtta 50 arvutitehnikut ja 150 informaatikut. Võrdluseks võib tuua, et arstiteaduse erialale võetakse 160 tudengit [1]. Tartu Ülikoolis on informaatika üks arvuliselt suurimaid erialasid. Infotehnoloogiaga seotud erialasid õpetatakse üsna mitmes Eesti kõrgkoolis, näiteks Arvutikolledž, Eesti Ettevõtluskõrgkool Mainor, Eesti Infotehnoloogia Kolledž, Eesti Lennuakadeemia, Eesti Maaülikool, Lääne-Viru Rakenduskõrgkool, Tallinna Tehnikakõrgkool, Tallinna Tehnikaülikool, Tallinna Ülikool ning Tartu Ülikool. Informaatika, arvutisüsteemide ja muudel sarnastel erialadel on programmeerimine väga tähtsal kohal, olles üks põhiasju, mida stuudiumi algul õpetatakse ning millele põhinevad paljud järgmised teadmised. Lisaks arvutiteaduste erialadel õppijatele võivad programmeerimise algkursused olla ka kohustuslikud teiste erialade tudengitele. Tartu Ülikoolis on programmeerimine kohustuslik näiteks ka füüsika, matemaatilise statistika ja matemaatika erialadel. Erinevused kursustes ning tudengite oskustes tulevad sisse mitmetel põhjustel. Kõik tudengid, kes programmeerimise algkursusi võtavad, ei õpi arvutiteaduste erialadel, seetõttu on tudengite oskused erinevad. Samuti tulevad erinevused programmeerimisega seotud oskustesse sisse juba enne kõrgkooli tulekut. Kõikides keskkoolides ei pühendata sama palju aega programmeerimise õpetamisele ning paljudes koolides ei õpetata programmeerimist üldse. Erinevad ka õpetajate kogemused programmeerimise vallas nii kõrgkoolides kui ka enne kõrgkooli. Samamoodi on ka tudengite endi oskuste, võimete ja pühendumuse tasemed erinevad. Mitmed on õppinud programmeerimist omal käel väljaspool kooli. Üsna tähtis roll jääb kõrgkoolide esimestele programmeerimiskursustele. Need kujundavad tudengite arusaama nii programmeerimisest kui ka tudengite enda võimalustest. Huvi tuleks ka tekitada järgnevate programmeerimise kursuste vastu ning anda baas edasistele õpingutele. Seetõttu on väga tähtis, kuidas on kõrgkoolide esimesed programmeerimise kursused koostatud. Töö sisus mainitud algkursuste all on mõeldud edaspidi just kõrgkoolide algkursusi.

Käesoleva bakalaureusetöö eesmärgiks on uurida Eesti kõrgkoolides õpetatavaid esimesi programmeerimise kursusi, et kaardistada, milline on hetkeseis Eesti kõrgkoolides. Kaugemaks

ideeks, mis jääb sellest bakalaureusetööst välja, on püüda leida võimalus programmeerimise algkursuste korraldamisel teha kõrgkoolide vahel koostööd näiteks ideede ja kogemuste vahetamisel ning materjalide välja töötamisel. Käesolev töö võib olla just selle aluseks. Selle töö raames otsiti esimesi programmeerimise kursusi, mis on 2014/2015 või 2013/2014 õppeaastal kohustuslikud vähemalt ühel bakalaureuseõppe erialal. Leiti 16 erinevat programmeerimise kursust kümnest Eesti kõrgkoolist: Arvutikolledž, Eesti Ettevõtluskõrgkool Mainor, Eesti Infotehnoloogia Kolledž, Eesti Lennuakadeemia, Eesti Maaülikool, Lääne-Viru Rakenduskõrgkool, Tallinna Tehnikakõrgkool, Tallinna Tehnikaülikool, Tallinna Ülikool ja Tartu Ülikool. Kokku võtab neid kursusi igal aastal üle tuhande tudengi. Lisaks viidi läbi projekti „Kontseptuaalne raamistik suurendamiseks ühiskonna pühendumist IKTsse“ raames kolme Eesti kõrgkooli esmatudengite seas küsitlus, mille käigus uuriti ka tudengite esimest kokkupuudet programmeerimisega. Küsitluse käigus kogutud andmed tehti antud bakalaureusetöö kirjutamise jaoks kättesaadavaks. Bakalaureusetöö jaoks analüüsiti läbiviidud küsitluse vastuseid, et välja tuua infot, mis võib olla oluline kursuste täiendamisel.

Kokkuvõtvalt bakalaureusetöö jaoks saadi andmed järgnevalt:

- Otsiti artikleid sarnaste uuringute kohta mujal maailmas
- Otsiti algkursuste kohta informatsiooni, kuidas kursused on üles ehitatud
- Leitud kursuste andmeid paluti täiendada ning kontrollida kõrgkoolidel (võimalusel saadeti kiri vastutavale õppejõule, kui see ei olnud võimalik, siis administratiivsele isikule)
- Paluti kõrgkoolidel vastata küsimustikule
- Kirjutatud lühikokkuvõtted paluti kõrgkoolidel üle vaadata ja teha parandusi, kui millestki oldi valesti aru saadud.
- Saadi juurdepääs projekti „Kontseptuaalne raamistik suurendamiseks ühiskonna pühendumist IKTsse“ raames tehtud tudengite küsitluse andmetele

Antud bakalaureusetöö struktuur on järgmine. Teises peatükis vaadatakse programmeerimise algkursusi vaatlevaid uuringuid ning tehakse neist lühikokkuvõtte. Kolmandas peatükis kirjeldatakse põhjalikumalt, kuidas kursused on valitud ning kuidas koguti andmeid kursuste kohta ning tuuakse välja kursuste põhiandmed. Neljas peatükk toob välja küsimustikust saadud

vastuste põhjal kursuste kokkuvõtted. Viiendas peatükis analüüsitakse projekti „Kontseptuaalne raamistik suurendamaks ühiskonna pühendumist IKTsse“ raames tehtud tudengite küsitluse andmeid ning kuendas peatükis vaadeldakse kursuste erinevaid aspekte.

## 1. Mitmeid programmeerimise algkursusi käsitlevad uuringud

Antud bakalaureusetöö jaoks püüti leida mujal maailmas tehtud sarnase sisuga uuringuid, kus vaadeldi mitmeid programmeerimise algkursusi. Enamasti käsitleti sellistes artiklites peamiselt esimestel programmeerimise kursustel kasutatavaid programmeerimiskeeli. Artikkel „Whatever Happened to Richard Reid’s List of First Programming Languages?“ [2] uuris Ameerika Ühendriikide kõrgkoolides programmeerimise algkursusi programmeerimiskeele vaatepunktist. Andmed saadi 393 programmeerimise algkursusel õpetatava keele kohta. Aastal 2012 ilmunud uuringus leiti, et enamik kursusi õpetatakse Javas (197 kursust). Samas oli ka välja toodud, et mitmed kursused liiguvad Java pealt üle mõne muu programmeerimiskeele kasutamise peale. Enamasti oli uueks programmeerimiskeeleks Python (sarnaselt liiguti ka Tartu Ülikoolis Javalt Pythoni peale üle [3]) Pythoni kasutamisele võtmise põhjusteks toodi välja, et Pythonit on kergem õpetada ning tudengitele objektorienteeritud lähenemise kohta õpetamist saab lihtsasti edasi lükata. Artikkel “Introductory Programming Courses in Australia and New Zealand in 2013“ [4] uuris Austraalia ja Uus-Meremaa kõrgkoolides õpetatavaid programmeerimise algkursuseid (38 kursust). Sellest artiklist tuli samuti välja, et järjest rohkem õpetatakse programmeerimise algkursustel Pythonit. Kui veel aastal 2010 õpetati Javat 40% tudengitele ning Pythonit umbes 20% tudengitele, siis aastal 2013 kasutas Pythonit 33.7% ning Javat 26.9% tudengitest. Sarnane muutus on ka toimunud kursuste arvuga. 2013. aastal oli Pythonis ja Javas antavaid algkursusi sama palju – 27.3 %. Võrdluseks võib tuua, et aastal 2010 õpetati Javat 36.4% kursustel ning Pythonit 13.6% kursustel. Pythoni kasutamise eeliseks toodi välja järgmised põhjused:

- Kättesaadavus/ maksumus tudengitele
- Pedagoogilised eelised
- Platvormist sõltumatus
- Lihtsasti kättesaadavad materjalid

Samas Java kasutamise puhul olid peamiseks põhjusteks:

- Objektorienteeritud keel
- Tähtsus tööstuse jaoks



Kuigi Pythonit võib ka lugeda mingil määral objektorienteeritud keeleks, siis Pythonit on võimalik õpetada algtasemel ka ilma objektidele keskendumata. Samas Javat on keeruline õpetada, ilma klassistruktuuri õpetamata. 34 kursust 38-st vastasid küsimustele kasutatavate õppematerjalide kohta. Umbes 80% kursustel kasutati videoloenguid, ning üle 90% kursustel olid loengu slaidid tudengitele kättesaadavad. Kursuste õpetamise suuremate üldiste eesmärkidena toodi välja algoritmilise mõtlemise arendamine, programmeerimise esimese kogemuse andmine ning algteadmiste õpetamine. Artikleid, mis keskenduvad ühele kursusele käsitletakse töö lõpu poole.

## 2. Kursuste valik ja andmete kogumine

Igal aastal võtab programmeerimise algkursusi üle tuhande tudengi. Erialad, millel need kursused on, on erinevad ning samuti on ka kursused erinevad. Bakalaureuse töö eesmärgiks on vaadelda algkursusi ning neid võrrelda. Töö eesmärgiks ei ole väita, et mõni kursus oleks teisest parem või luua pingerida kursustest. Käesoleva bakalaureusetöö jaoks otsiti välja Eesti kõrgkoolide programmeerimise algkursused, mis on 2013/2014. või 2014/2015. õppeaasta õppekavades ning on vähemalt ühes õppekavas kohustuslikud. Pidi leiduma vähemalt üks eriala, kus kursus ei olnud valikmoodulis või valikaine. Kõigepealt tuli leida kõrgkoolid, kus õpetatakse programmeerimist. Selleks kasutati veebilehti haridussilm.ee [5] ning eesti.ee [6]. Haridussilm võimaldas leida kõrgkoolid, mis võtsid 2013/2014. õppeaastal vastu tudengeid kõrghariduse I astmele, kus õppesuunaks olid arvutiteadused. Lisaks vaadati käsitsi läbi eesti.ee lehel välja toodud kõrgkoolid. Leitud kõrgkoolide kodulehtedelt otsiti kõigepealt eriala järgi (otsiti erialasid, mis on arvutiteaduste või infotehnoloogiaga seotud). Vaadati erialade õppekavasid ja sealt otsiti välja programmeerimise algkursused. Kui selle meetodiga ei leitud enam ühtegi algkursust, otsiti kõikide kõrgkoolide õppeinfosüsteemidest kursusi aine nimetuse järgi. Otsingusõnana kasutati „programmeeri“, et leida ka kursusi, kus sõna lõpp võib olla käändes. Enamasti oli õppeinfosüsteemidest kursuste ja erialade kohta infot kergem leida, kui kõrgkoolide kodulehtedelt. Kahjuks ei võimaldanud kõikide koolide õppeinfosüsteemid siseneda külalisena. Leitud kursuste nimekiri ei pruugi olla täielik, kuna võib olla veel algkursusi, mis on eripärase nimetusega või tuleb teha valik mitme kursuse vahel, millest kohustuslik on valida üks. Samas kõrgkoolidele saadetud kirjades paluti teada anda, kui lisaks väljatoodud algkursustele oli veel programmeerimise algkursusi, mida ei olnud mainitud.

Iga kursuse kohta otsiti järgnevat infot:

- Mitmendal semestril reeglina kursus läbitakse
- Millist programmeerimiskeelt kasutatakse kursuse õpetamisel
- Kui suur on aine maht
- Kuidas on aine jaotatud (loengud, praktikumid, iseseisev töö)

- Milliseid avalike õppematerjale on kursuse kohta (kõik lingid avalikele materjalidele lisati ka veebilehele, mis on kättesaadav aadressilt [http://kodu.ut.ee/~kpeedosk/avalikud\\_materjalid](http://kodu.ut.ee/~kpeedosk/avalikud_materjalid))
- Hindamise vorm
- Keskmise üliõpilaste arv ühel õppeaastal.

Iga kursuse kohta loodi *Google Drive* leht, kuhu lisati leitud andmed ning ka mõned küsimused, millele paluti koolidel vastata [Lisa 1]. Kuna kursuste kohta ei olnud võimalik kõiki andmeid leida (küstiti informatsiooni, mida tavaliselt koolide kodulehtedel ei kuvata), siis saadeti igale kõrgkoolile e-mail, kus paluti neil vastata küsimustele ning andmed üle vaadata, neid täiendada ja vajaduse korral ka parandada. Kirjad püüti saata võimalikult õigele inimesele, näiteks õppejõule. Kuid alati ei olnud võimalik vastavat õppejõudu leida. Sellisel juhul saadeti kiri administratiivse rolliga isikule. Vastused koolidelt olid erineva detailsuse astmega – mõned olid vastanud põhjalikumalt kui teised. Samuti oli erinevus vastamise detailsuses ka küsimuste kaupa.

Kokku leidis kümnes erinevas Eesti kõrgkoolis 16 erinevat programmeerimise algkursust. Kursused on sorteeritud programmeerimiskeelte järgi.

Need kursused on kohustuslikud 29 erialal (sisse ei ole arvestatud erialasid, kus kursus on valikmoodulis või valikainena ning olukordi, kus kursus on kohustuslik mõne teise kõrgkooli erialal). Enamasti on algkursuste mahuks 4-6 EAP-d ning 3 või vähem EAP-d andvad kursused on mõeldud erialadele, kus programmeerimine ei ole tudengite põhitegevuseks. Tuli ka välja, et üle 4 EAP andvatel kursustel on hindamise vormiks eksam ning 3 või vähem EAP-d andvad kursused on enamasti mitteeristatavad. Lisaks toimuvad kursused tavaliselt I või II semestril, aga on ka kursused, mida võtavad aset III – VI semestril, kuid need on erialad, millel programmeerimine ei ole üks põhitegevustest.

Eriala	Keel	EAP	Tudengeid
Programmeerimine I - Tallinna Tehnikaülikool	C	5	
Programmeerimise alused - Eesti Ettevõtluskõrgkool Mainor	C#	5	30 – 75
Programmeerimise alused Java baasil - Arvutikolledž	Java	4,5	
Programmeerimise algkursus Java baasil - Eesti Infotehnoloogia Kolledž	Java	6	280
Programmeerimise alused - Eesti Lennuakadeemia	Java	4	20
Veebiprogrammeerimine - Tallinna Ülikool	PHP	4	80 – 90
Veebiprogrammeerimine I - Lääne-Viru Rakenduskõrgkool	PHP	2	11
Programmeerimise süvendatud algkursus - Tallinna Tehnikaülikool	Python	5	
Programmeerimise põhikursus - Tallinna Tehnikaülikool	Python	4	
Programmeerimise alused - Tallinna Ülikool	Python	5	100
Programmeerimise alused - Tallinna Ülikool	Python	6	20
Programmeerimine - Tartu Ülikool	Python	3	320
Programmeerimise alused - Tartu Ülikool	Python	4	150
Programmeerimise algkursus - Tallinna Tehnikaülikool	Scratch, Visual Basic	3	
Programmeerimise alused - Tallinna Tehnikakõrgkool	Visual Basic	3	70
Programmeerimine - Eesti Maaülikool	Visual Basic Application	3	17

Tabel 1. Kursuste koondandmed

### 3. Kursused

Kõrgkoolidelt saadud vastuste põhjal koostati lühikokkuvõtted kursustest. Lühikokkuvõttele on lisatud ka kursuse kohta otsitud andmed. Lühikokkuvõtted saadeti ka vastavatele kõrgkoolidele ning uuriti, kas nende vastustest oldi korrektselt aru saadud. Kursused on järjestatud tähestikulises järjekorras kõrgkooli nime järgi.

#### 3.1 Eesti Infotehnoloogia Kolledži kursus programmeerimise algkursus Java baasil

Antud kursus on kohustuslik IT süsteemide administreerimise, IT süsteemide arenduse ja infosüsteemide analüüsi erialadel.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	I semester
<b>Programmeerimiskeel</b>	Java
<b>Aine maht</b>	6 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loengud 32 tundi Harjutus 14 tundi Praktikum 32 tundi Iseseisev töö 46 tundi E-õpe 32 tundi
<b>Hindamise vorm</b>	Eksam
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	280

Tabel 2. Eesti Infotehnoloogia Kolledži kursuse programmeerimise algkursus Java baasil

Java on algkursusel kasutatavaks keeleks alates aastast 2000. Vastava programmeerimiskeele juurde jõuti selle tõttu, et tööturul on suur nõudlus Java arendajate järele ning erialadel, millel kursus on kohustuslik on põhitegevuseks programmeerimine. Kursuse läbimisel on eelteadmised kasulikud, kuid ainet on võimalik läbida ka ilma eelteadmisteta. Kõik vajalikud materjalid on

tudengitele eesti keeles olemas ning lisaks nendele kasutatakse veel ka ingliskeelseid lisamaterjale, erinevaid juhendeid ja õpikut. Kõik kursuse materjalid ja videoloengud on tudengitele kättesaadavad veebi vahendusel. Tudengitel ei ole kohustust käia loengutes, kuid praktikumides käimine on kohustuslik. Praktikumides on peaaegu iga nädal väiksemad ülesanded või tunnikontrollid ja tudengitel tuleb praktikumis ka üks suurem projekt ette võtta. Ülesannete kontrollimiseks on Moodle'i õppekeskkonda integreeritud *JUnit* testid Java programmide testimiseks.

### 3.2 Eesti Lennuakadeemia kursus programmeerimise alused

Antud kursus on kohustuslik lennunduse side- ja navigatsioonisüsteemide käitamise erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	III semester
<b>Programmeerimiskeel</b>	Java
<b>Aine maht</b>	3 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Praktikumid 44 tundi Iseseisev töö 34 tundi
<b>Hindamise vorm</b>	Mitteeristav (plaan muuta eristavaks)
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	20

Tabel 3. Eesti lennuakadeemia kursus programmeerimise alused

Kursus toimub alates 2013/2014. õppeaastast ning on õppekavas eesmärgiga anda tudengitele oskused programmikoodi lugeda ning väiksemal määral ka seda modifitseerida. Java valiti kursusel kasutatavaks programmeerimiskeeleks, kuna Java on üsna laialt kasutusesolev keel ning selle õppimine võrreldes teiste keeltega (näiteks C) tundus jõukohane. Lisaks toetavad ka Eesti Lennuakadeemia robotiplatvormid Javat ning Java kohta on kättesaadaval rohkelt materjale nii eesti kui ka inglise keeles. Eraldi loenguid kursuse raames ei ole ning õppetöö tehakse ära praktikumides, kus on ka teoreetiline sissejuhatus. Seetõttu on kohalkäimine kohustuslik. Praktikumides on kolm korda semestri jooksul tunnikontrollid, mis sisaldavad väikeseid ülesandeid, mille lahendamiseks peaks kuluma umbes 20 minutit. Kui tudengitel on eelnevaid kogemusi programmeerimisega, siis neile antakse keerukamaid lisäülesandeid. Plaanis on kursusele lisada ka mõni suurem projekt ning iganädalased kodutööd, mille lahendamine peaks võtma umbes 120 minutit.

### 3.3 Eesti Maaülikooli kursus programmeerimine

Antud kursus on kohustuslik biotehniliste süsteemide erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	VI semester
<b>Programmeerimiskeel</b>	Visual Basic Application
<b>Aine maht</b>	3 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loenguid 8 tundi Praktikumid 32 tundi Iseseisev töö 39 tundi
<b>Hindamise vorm</b>	Mitteeristav
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	17

Tabel 4. Eesti Maaülikooli kursus programmeerimine

Kui 80ndate lõpus ja 90ndate alguses õpetati veel Turbo Pascalit, mis oli kohustuslikuks aineks maamöötmise erialal. Vahepeal oli auk programmeerimise õpetamisel ja alates 1995. aastast mindi üle Visual Basic Application programmeerimiskeele peale, mida õpetati valikainena. Muudatus tulenes *Microsoft Office*'i laialdasest levikust ning see programmeerimiskeel võeti kasutusele, et näidata rakendustarkvara võimalusi ja õpetada selle kõrvalt ka natukene programmeerimise algtõdesid. Aine oli 3 aasta jooksul küllalt populaarne, sest seda ainet võttis 30 tudengit ja tunnid toimusid kahes rühmas. 90.-ndate lõpus kadus järsku huvi selle aine vastu. Uuesti võeti programmeerimine õppekavasse 2009. aastal geomaatika erialal ja oli seal vaid 3 aastat ning tänaseks päevaks on jäänud vaid biotehniliste süsteemide erialale. Kuna programmeerimine ei ole biotehniliste süsteemide erialal põhitegevuseks, siis on keele valikul just silmas peetud suuremate probleemide kiiremat lahendamist makrode koostamise abil, lootes, et ka tulevikus võetakse oma igapäevaste toimingute lihtsustamiseks mõne makro kirjutamine ette. Loengutes ega praktikumides osalemine ei ole kohustuslik. Kuigi kursuse materjalid on



eestikeelsena internetis saadaval on ilma praktikumides ja loengutes käimata iseseisvat tööd üsna raske teha. Praktikumides tehtud ülesanded on suhteliselt väikesed, kuid iga järgnev ülesanne on eelneva loogiliseks jätkuks. Siiski tuleb ka tudengitel kursuse lõpu poole ette võtta suurem projekt.

### 3.4 Lääne-Viru Rakenduskõrgkooli kursus veebiprogrammeerimine I

Antud kursus on kohustuslik majandusinfosüsteemide korraldamise erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	II semester
<b>Programmeerimiskeel</b>	PHP
<b>Aine maht</b>	2 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loengud 26 tundi Iseseisev töö 26 tundi
<b>Hindamise vorm</b>	Eksam
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	11

Tabel 5 Lääne-Viru Rakenduskõrgkooli kursus veebiprogrammeerimine I

PHP on valitud programmeerimiskeeleks, kuna tegemist on üsna levinud programmeerimiskeelega veebirakenduste loomiseks ning PHP järgi on ka nõudlust tööturul. Kursusel tehakse lihtsamaid rakendusi. Kursusel loengutes osalemine kohustuslik ei ole, kuid siiani on ilma kohal käimata hakkama saanud vaid need tudengid, kes töötavad programmeerijatena. Kui kursusel on tudengeid, kes juba tunnevad teemat, siis neile antakse teha eraldi rakendus ning nad ei pea tundides kohal käima, vaid lihtsalt rakenduse ära esitama. Aine materjalid on põhiliselt paber kandjal.

### 3.5 Tallinna Tehnikakõrgkooli kursus programmeerimise alused

Antud kursus on kohustuslik transpordi ja logistika erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	II semester
<b>Programmeerimiskeel</b>	Visual Basic
<b>Aine maht</b>	3 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Praktikumid 32 tundi
<b>Hindamise vorm</b>	Arvestus
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	70

Tabel 6 Tallinna Tehnikakõrgkooli kursus programmeerimise alused

Algkursusel on kasutusel Visual Basic umbes 5 aastat. Visual Basic valiti just seetõttu, et keel on lihtne algajatele ning see on olemas igas arvutis, kuhu on paigaldatud *Microsoft Office*. Tudengitel ei ole vaja hakata midagi lisaks arvutisse paigaldama. Kuna tegemist ei ole erialaga, kus tudengite üheks põhiülesandeks on programmeerimine, siis alustatakse aine õpetamist nullist. Tudengitele on kättesaadavad materjalid, mis on kas eesti- või ingliskeelsed.

### 3.6 Tallinna Tehnikaülikooli kursus programmeerimine I

Antud kursus on kohustuslik arvutisüsteemide, elektroonika ja telekommunikatsiooni, mehhatroonika, integreeritud tehnoloogiate ning elektroonika ja bioonika erialadel.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	Arvutisüsteemid - I Semester Elektroonika ja telekommunikatsioon - I semester Mehhatroonika - I semester Integreeritud tehnoloogiad - III semester Elektroonika ja bioonika - I Semester
<b>Programmeerimiskeel</b>	C
<b>Aine maht</b>	5 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loengud 16 tundi Praktikumid 48 tundi
<b>Hindamise vorm</b>	Eksam

Tabel 7. Tallinna Tehnikaülikooli kursus programmeerimine I

Varem oli programmeerimise algkursus programmeerimiskeeles Pascal, kuid C peale mindi üle, kuna C ja sellelaadsed programmeerimiskeeled (C++, C#, Java) on üsna laialdaselt levinud. Kuna tegemist on siiski algkursusega, siis hakatakse peale algteadmistest. Kui mõni tudeng on juba teemaga hästi kursis, siis korraldatakse erikorras arvestus, et tudeng ei peaks tundides igavlema. Sõltuvalt tudengist pakutakse ka pädevamatele tudengitele lisaks varasema aine sooritamise võimalusele ka keerukamaid ülesandeid ja mõtteid selle kohta, mida võiks veel teha. Kursuse õppetöö toimub nii eesti- kui ka venekeelsena, kuid täiendavad viited materjalidele on enamasti ingliskeelsed.

### 3.7 Tallinna Tehnikaülikooli kursus programmeerimise põhikursus

Antud kursus on kohustuslik äriinfotehnoloogia erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	II semester
<b>Programmeerimiskeel</b>	Python
<b>Aine maht</b>	5 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loengud 32 tundi Harjutused 32 tundi
<b>Hindamise vorm</b>	Eksam

Tabel 8. Tallinna Tehnikaülikooli kursus programmeerimise põhikursus

Tegemist on üsna uue kursusega, mida on antud ainult üks aasta. Programmeerimiskeeleks valiti Python, kuna seda on lihtne õpetada ja kasutada. Seda õpetatakse üsna laialdaselt ning Pythoni kohta on saadaval palju materjale. Lisaks sobib Python programmeerimiseks üsna hästi ja ka tööturul on nõudlust selle oskamise järele. Kursus on ülesse ehitatud eeldusega, et tudengid ei ole programmeerimisega varem kokku puutunud. Kuigi loengutes osalemine ei ole kohustuslik, on loengute lõpus tunnikontrollid, mis annavad boonuspunkte. Sülearvutite kasutamine loengus on igati soovitatav, et iseseisvalt ees tehtavaid programme järgi teha. Internetis on saadaval kursuse materjalid nii eesti- kui ka inglise keeles. Ka harjutustundides käimine ei ole kohustuslik, kuid seal tehtud ülesanded annavad samuti boonuspunkte. Harjutustundides on väikesed ülesanded, mis vahel arendavad edasi varasemaid ülesandeid ning eksamiks tuleb tudengitel koostada üks projektülesanne. Kursusel on ka venekeelsed rühmad.

### 3.8 Tallinna Ülikooli kursus programmeerimise alused

Antud kursus on kohustuslik informaatika erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	I semester
<b>Programmeerimiskeel</b>	Python
<b>Aine maht</b>	4 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	56 tundi
<b>Hindamise vorm</b>	Eksam
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	Viimati oli 100

Tabel 9. Tallinna Ülikooli kursus programmeerimise alused

Algselt õpetati programmeerimise algkursusel Pascalit, kuid alates aastast 2006 või 2007 on kasutusel programmeerimiskeelena Python. Keele vahetuse tingis just vajadus võtta kasutusele keel, mida ehk realselt rohkem Eestis kasutatakse. Lisaks oli ka Pascalil vanamoeline graafiline kasutajaliides, millega õpilased toime ei tulnud ja rolli mängis ka tudengite nurin nii vana keele kasutamise üle. Python sattus valitud keeleks, kuna tegemist oli uuema keelega, kus tudengid ei pea kohe hakkama objektorienteeritud programmeerimiskeeles programmeerima ja Pythonis ei kaasne lihtsamate programmide korral ka palju lisalauseid. Kursusel õpetatakse Pythoni abil programmeerimise põhitõdesid ning Pythoni keelte eripäradesse ei minda. Kursuse põhiõpik on ingliskeelne, millele lisaks on eestikeelsed lühikonspektid. Tundides tutvustatakse materjali vastavalt vajadusele, kuna loenguid ja praktikume eraldi ei ole. Selleks, et tudengid, kes juba varem programmeerimist õppinud on, tundides igavlema ei peaks, on võimalik neil kokkuleppel tundides mitte käia ja teha ära ainult hindelised tööd.

### 3.9 Tallinna Ülikooli kursus veebiprogrammeerimine

Antud kursus on kohustuslik informaatika erialal.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	I semester
<b>Programmeerimiskeel</b>	PHP
<b>Aine maht</b>	4 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	56 tundi
<b>Hindamise vorm</b>	Eksam
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	3 rühma, kokku 80-90 üliõpilast

Tabel 10. Tallinna Ülikooli kursus veebiprogrammeerimine

Kursust on õpetatud alates 2001. aastast. PHP valiti programmeerimiskeeleks, kuna tegemist on väga levinud keelega ning tööturul on üsna palju nõudlust PHP järele. Programmeerimiskeele valikul võeti arvesse ka tudengite õppesuuna valikut – nii programmeerimise kui graafilise disaini poole õppuritel on veebiga üsna palju kokkupuudet. Tundides käimine ei ole kohustuslik, kuid puudujaid küsitletakse teoriaseminaridel ja eksamil põhjalikumalt. Tunnid algavad tutvustava osaga ning sellele järgneb ülesannete lahendamine kas iseseisvalt või grupiga. Alustuseks on tehtud võimalikult lihtsad näited, mida täiendades ja kombineerides saab suuremad lahendused kokku panna. Ligikaudu pooltes tundides on ka rühmatööd. Vahel vaadeldakse rühmadena tehtud tööd ka tundide lõpus tahvli peal läbi. Samas jaguneb mõni tunniteema mitmele nädalale. Ülesanded on tundides ja kontrolltöodes jaotatud kolmeks tasemeks, mille keerukuse vahe on kolmekordne, nõrgema ja tugevama taseme vahe on järelikult 9-kordne. Seetõttu saavad kõik midagi tehtud ning kõrgema tasemega on ka kogenenumatel nuputamist. Kursuse lõpus on lõpuprojekt 2-4 liikmelise grupiga. Lisaks sellele, et kursuse

materjalid on eestikeelsed ning veebis kättesaadavad, on ka raamatukogus päris korralikud vene- ja ingliskeelsed PHP õpikud.



### 3.10 Tartu Ülikooli kursus programmeerimine

Antud kursus on kohustuslik arvutitehnika, füüsika, informaatika, matemaatika ning matemaatilise statistika erialadel.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	I semester
<b>Programmeerimiskeel</b>	Python
<b>Aine maht</b>	6 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loengud 32 tundi Praktikumid 32 tundi
<b>Hindamise vorm</b>	Eristav
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	320

Tabel 11. Tartu Ülikooli kursus programmeerimine

Veel 3-4 aastat tagasi oli esimese programmeerimiskeelena kasutusel veel Java, kuid pärast suurt diskussiooni leiti, et Python on õpetamiseks parem keel. Lisaks on Pythoni kohta palju kättesaadavaid materjale. Pythonis kirjutatud programme on kergem lugeda ning Pythonis õpetatud kursuse läbivus oli suurem kui Javas antud kursustel [3]. Loengutes ja praktikumides käimine ei ole kohustuslik ning vabariiklikust informaatika olümpiaadi lõppvoorust osavõtjad saavad aine automaatselt sooritatud. Lisaks on videoloengud, eestikeelne õpik ja lühivideod, mis aitavad ülesannete lahendamisel, saadaval internetis, kuigi lühivideosid võiks olla rohkem. Teadmiste kontrollimiseks on loengutes kasutusel ka klikkeriküsimused ning praktikumides on tärnülesanded, mis on raskemad, kuid mille lahendamine ei peaks võtma mitu nädalat. Lisaks tuleb kursuse jooksul teha rühmana kaks suuremat projekti. Ka kontrolltööde kontrollimine on lihtsamaks tehtud, nimelt on Moodle'i keskkonda lisatud *plugin* tööde automaatseks kontrollimiseks.

### 3.11 Tartu Ülikooli kursus programmeerimise alused

Antud kursus on kohustuslik informaatika, majandusteaduse ning materjaliteaduse erialadel.

<b>Mitmendal semestril see aine reeglina läbitakse?</b>	II semester
<b>Programmeerimiskeel</b>	Python
<b>Aine maht</b>	3 EAP
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	Loengud 16 tundi Praktikumid 16 tundi Iseseisev töö 78-96 tundi
<b>Hindamise vorm</b>	Mitteeristav
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	umbes 150

Tabel 12. Tartu Ülikooli kursus programmeerimise alused

Kursus programmeerimise alused loodi 5 aastat tagasi programmeerimise kursuse põhjal tudengitele, kelle põhitegevuseks ei ole programmeerimine. Lühendatud kursus koos järgneva kursusega programmeerimise alused II kokku moodustavad 6 EAP-se programmeerimise kursuse. Python sobis ka sellele kursusele hästi, kuna alguses on Pythonit kergem õpetada, kui näiteks Javat ning kuna kursust antakse erialadel, kus ei õpi põhiprogrammeerijad, siis ei tohiks aine olla keerulisem programmeerimise kursusest. Loengud ja praktikumid on jaotatud nii, et arvestuse saamiseks on vaja saada punkte, mida saab ka loengutes ja praktikumides lisaks kontrolltöödele ja projektile. Nimelt on loengutes tunnikontrollid ja klikkeriküsimused ning praktikumides tuleb tudengitel koostada üks projekt, mille tehnilised nõuded on ette antud, kuid sisend ülesanne tuleb neil endal püstitada.

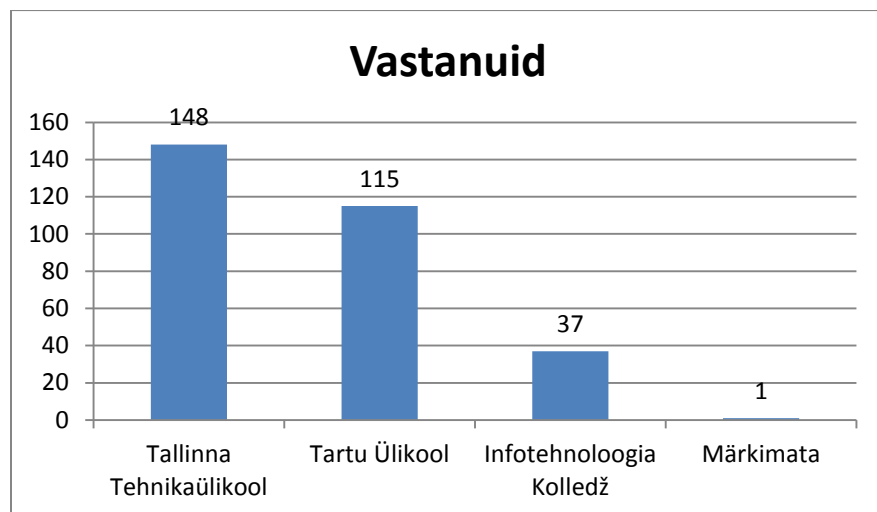
- Ülesande lahendus peab sisaldama vähemalt kahte endakirjutatud funktsiooni ja nende rakendamist.

- Ülesande lahenduses peab olema vähemalt 2 tingimuslauset, kusjuures vähemalt üks peab sisaldama *else*-osa.
- Ülesande lahenduses peab olema vähemalt üks tsükkel.

Ülesannete kontrollimiseks kasutatakse ka Moodle'i keskkonna vahendusel automaatseid. Ülesannete lahendamiseks on ka mõningad tutvustavad videod. Kõik kursuse materjalid ja loengute videod on tudengitele kättesaadavad veebis. Samas õpik [7] on tehtud kursuse jaoks programmeerimine ja vahel on selle kursuse jaoks natuke raske.

#### 4. Tudengite varasemad kogemused kõrgkooli tulles

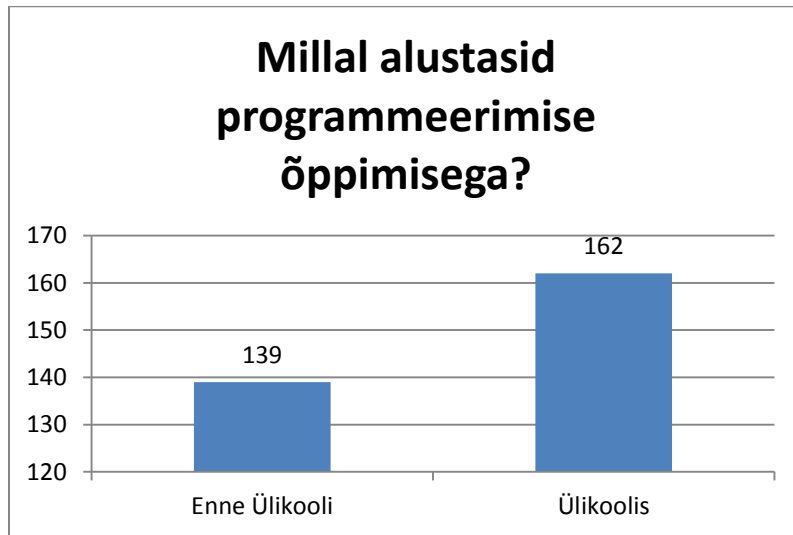
Programmeerimise algkursuste loomisel on tähtis teada ka tudengite eelnevaid kogemusi. Kuna Eestis on koolides programmeerimise õpetamine väga erinev, siis võimaldab sellelaadne teadmine kursusi paremini korraldada. Sealjuures on oluline teada nii tudengite taset kui ka kasutatud programmeerimiskeeli. Selle töö raames vaadatakse projekti “Kontseptuaalne raamistik suurendamiseks ühiskonna pühendumist IKTsse” raames viidi läbi küsitlus Tallinna Tehnikaülikooli, Tartu Ülikooli ja Infotehnoloogia Kolledži esimese aasta tudengite seas. Kuigi on uuritud ainult kolme kooli tudengeid, siis vastanute arv on ikkagi üsna suur. Küsitletud tudengite hulka kuulusid Tallinna Tehnikaülikooli äriinfotehnoloogia, arvutisüsteemide ja informaatika tudengid, Infotehnoloogia Kolledži IT süsteemide administreerimise, IT süsteemide arenduse ja infosüsteemide analüüsi tudengid ning Tartu Ülikooli arvutitehnika ja informaatika tudengid. Kokku saadi vastused 301 tudengilt, kellest 83 (~28%) olid naised ning 214 (~71%) mehed (tudengid ei täitnud kõik välju). Vastajate vanused jäid vahemiku 18 – 43 eluaastat.



Joonis 1.

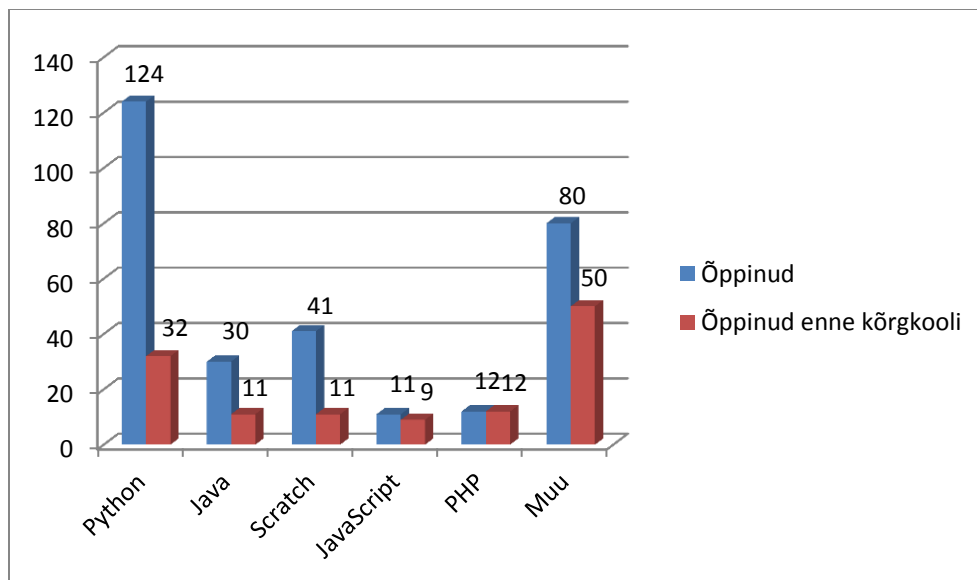
Tudengitelt uuriti, kas nad on enne kõrgkooli astumist tegelenud programmeerimisega. Vastustest selgus, et 139 tudengit (ligikaudu 46% vastanutest) olid enne kõrgkooli astumist tegelenud programmeerimisega. Täiendava küsimusena uuriti, kui vanalt nad esimest korda programmeerimisega kokku puutusid. Siinkohal vaadeldi kahte gruppi tudengeid, kuni 16-aastased (v.a) ja kuni 19-aastased (k.a). Need vahemikud on valitud sotsiaalkindlustusameti

peretoetuste maksmise alusel [8]. Siinkohal on eeldatud, et kuni 16-aastane isik õpib põhikoolis ja kuni 19-aastane isik õpib keskkoolis. Enne 16. eluaastat oli programmeerimisega kokku puutunud 35 tudengit (~12%) ning 19. eluaasta lõpuks oli juba 125 tudengil (~42%) olnud kokkupuude programmeerimisega. Samas ainult 25 tudengit (~8%) vastasid, et koolis oli informaatika või arvutiõpetuse tundide peamiseks sisuks programmeerimine ja 229 tudengit (~76%) vastasid, et enamasti tegeleti nende ainete raames teksti- ja tabelitöötlemisega. Siit järeldub, et peaaegu pooled tudengitest on enne arvutiteaduste erialale õppima asumist kokku puutunud programmeerimisega ning üsna paljud neist on seda teinud iseseisvalt.



Joonis 2.

Tudengitelt uuriti ka millised olid esimesed programmeerimiskeeled, mida nad õppisid.



Joonis 3.

Kõige suurem arv tudengeid oli esimese programmeerimiskeelena õppinud Pythonit (124 tudengit – ~41%), kuid tuleb välja tuua, et neist ainult 32 (~26%) olid enne ülikooli programmeerimisega tegelema. Java oli esimeseks programmeerimiskeeleks 30 tudengil (~10%) ning 11 (~37%) neist olid enne kõrgkooli astumist programmeerimisega kokku puutunud. Scratch oli esimeseks programmeerimiskeeleks 41 tudengil (~13%) ning enne kõrgkooli astumist oli neist 11 (~27%) programmeerimisega kokku puutunud. Javascript oli esimeseks programmeerimiskeeleks 11 tudengil (~4%), kellest 9 (~82%) olid sellega enne kõrgkooli kokku puutunud. PHP oli esimeseks programmeerimiskeeleks 12 tudengil (~4%) ning kõik neist olid enne kõrgkooli astumist programmeerimisega tegelema. Muudest programmeerimiskeeltest populaarsemad olid C – 18 tudengil (~6%), C++ – 16 tudengil (~5%), HTML – 17 tudengil (~6%), Pascal – 12 tudengil (~4%) ning Visual Basic - 10 tudengil (~3%).

## 5. Kursuste erinevad aspektid

Kõrgkoolidelt saadud vastustest on võimalik välja ka tuua sarnasusi, erinevusi ning teemasid, mida oleks võimalik tulevikus koostöös edasi arutada. Vastused on jaotatud tinglikesse kategooriatesse: kõrgkoolide ülesed teemad, kursused laiemas pildis, kursus tervikuna ning kursuse osad. Kui mõni teema puudutas mitut kategooriat, siis toodi see välja kõige laiemas kategooria juures. Lisaks tuuakse ka välja mõned uuringud, mis on sarnastel teemadel läbi viidud. Vastavaid uuringuid pikemalt selle töö raames ei käsitleta.

### 5.1 Kõrgkoolide ülesed teemad.

Mitmed kõrgkoolid tõid välja enda vastustes või hilisemas kirjavahetuses, et edasine koostöö oleks kasulik. Osad kõrgkoolid pakkusid ka välja loengu konsepte ning materjale teistele kõrgkoolidele kasutamiseks. Lisaks näidati huvi koostöö raames ülesannete baasi suurendada ning uusi õppematerjale luua. Näiteks toodi välja veebiraamistikke katmine, JavaScripti abil kasutajaliideste koostamine ning JQuery. Tallinna Ülikooli veebiprogrammeerimise kursuse vastustes toodi välja, et lühikeste ning terviklike töötavate näidete põhjal õppimine on tõhusam ning igati mõjusam. Tartu Ülikoolis on kasutusel lühivideod, mis aitavad ülesannete lahendamisel. Samas võiks olla rohkem materjale just aeglaselt alustajatele, mis oleks eriti lihtsustatud. Lisaks tunti huvi, kuidas oleks võimalik tudengid rohkem tundides kaasa mõtlema saada. Tartu Ülikoolis on loengutes võetud kasutusele klikkeriküsimused, millele tudengid jooksvalt loengu käigus vastavad ning ka tunnikontrollid. Lisaks toodi välja, et teistes kõrgkoolides võiks kaaluda ülesannete automaattestimist, mida juba kasutatakse mõne kursuse raames, näiteks Infotehnoloogia Kolledžis ja Tartu Ülikoolis. Võiks ka kaaluda tudengitele projektide andmist, kus on ette antud tehnilised nõudmised, kuid ülesande sisu tuleb neil endal püstitada.

## 5.2 Kursuse koht laiemas pildis

Erialade mõttes on kursused üsna erinevad. Mõned kursused on kohustuslikud mitmel erialal, mõned ainult ühel ning mõne eriala puhul tuleb teha valik mitme kursuse vahel. Kursused varieeruvad ka selles, mitmendal semestril ainet võetakse (1-6 semester). Samas IT-ga rohkem seotud erialadel on programmeerimise algkursused tavaliselt esimesel või teisel semestril. Erialast sõltub ka osaliselt programmeerimiskeel. Mitmed kõrgkoolid tõid välja, et programmeerimiskeele valikul oli arvestatud erialaga, kus seda õpetatakse. Kokku on programmeerimise algkursustel kasutusel Eesti kõrgkoolides 7 erinevat programmeerimiskeelt C, C#, Java, PHP, Python, Scratch ja Visual Basic (Visual Basic ja Visual Basic Application on arvestatud ühe keelena).

Programmeerimiskeeltest kõige rohkem on kasutusel Python, mida õpetatakse kuuel kursusel ning Pythonit õpib ka kõiges suurem arv tudengeid. Pythoni peamiste eelistena toodi välja, et kättesaadavate materjalide hulk on üsna suur ning Pythonit on lihtne õpetada. Pythoni kasutuselevõtt on ka levimas mujal maailmas. C, C# ja Java puhul võib välja tuua, et tegemist on üsna laialt levinud keeltega ning tööturul on just nende keelte järele kõige rohkem nõudlust, mis oli ka üks põhjus nende valimisel. Siiski on kursuste arv, kus kasutatakse Javat või C-d ka mujal maailmas kahanemas. Samas on näha vastustest, et paljudel kursustel, mida õpetatakse Pythonis, on järgneva aineks objektorienteeritud programmeerimine.

Programmeerimiskeelte paremaks võrdlemiseks lisati koodinäide Hello World! tüüpi programmist, mis kuvaks ekraanil „Hello, World!“ ning lühikirjeldus programmeerimiskeelest [Lisa 2].

Kursuste mahud jäävad 2-6 EAP vahemikku. 4 või rohkem EAP-d andvad kursused on erialadel, mis on seotud IT-ga. Enamasti kursuste juures tudengite eelteadmistega spetsiaalselt arvestatud ei ole. Kuid samas eelmises peatükis tuli välja, et 46% IT tudengitest on varem kokku puutunud programmeerimisega. On küll paar kursust, kus on varasemate kogemustega tudengitega arvestatud. Näiteks neil lubatakse aine varem lõpetada, neile on raskemaid ülesandeid, nad ei pea tundides kohal käima või on selliste tudengite jaoks loodud süvendatud kursus.



### 5.3 Kursus tervikuna

Kursuste suurused on üsna erinevad. On kursusi, mida võtab aastas umbes 10 tudengit ja samas on ka kursusi, mida õpetatakse 320 tudengile. Kursused erinevad ka sisus ning mahus. Enamik kursusi on keskendunud päris algteadmistele, kuid leidis ka paar kursust, kus jõuti kursuse jooksul näiteks rekursioonini. Sellised kursused on 5-6 EAP mahuga. 4-6 EAP mahuga kursustel on hindamise vormiks eksam. Kursuste õpetamiseks esines kaks moodust. Osadel kursused olid loengud ja praktikumid eraldatud ning osadel olid ainult praktikumid, kus vajadusel tutvustati ka teooriat. Toodi välja avalikud materjalid, mida kursuse õpetamisel kasutatakse. Linkidest materjalidele moodustati veebileht [http://kodu.ut.ee/~kpeedosk/avalikud\\_materjalid](http://kodu.ut.ee/~kpeedosk/avalikud_materjalid). Loengutes kohalkäimise enamasti kohustuslik ei olnud ning üsna mitmel kõrgkoolil olid videoloengud või loengute videod internetis kättesaadavad. Samas toodi välja, et mõnel kursusel loengutes kohal käimise eest võib saada lisapunkte. Praktikumides käimine on enamasti kohustuslik. Kursustel kasutatavad materjalid on eesti- või ingliskeelsed, mis on veebi vahendusel kättesaadavad. Samas toodi välja, et materjale võiks veel olla. Mitmetel kursustel on kasutusel ka e-õppe keskkond. Üldiselt on tudengite tööde hindamise ja parandamise automatiseerimisele üsna vähe keskendunud. Leidis ainult paar kursust, kus kasutati automaattestimist. See oleks ka üheks võimalikuks tuleviku koostööpunktiks. Automaattestidega on võimalik tudengitel saada tagasisidet enda töö kohta ning õppejõududel on tööde hindamine kergem. Automaattestide põhjal on ka tudengitele ning õppejõududele anda paremat tagasisidet. Neid teemasid on ka teaduskirjanduses käsitletud. Näiteks automaattestimise räägivad artiklid [9-11] ning kursustelt väljalangemisest räägitakse artiklis [12].

### 5.4 Kursuse osad

Üsna mitmete kursuste raames tuleb tudengitel koostada suurem projekt. Lisaks toodi mitme kursuse vastustes välja, et praktikumides tehtavad väiksemad ülesanded arendavad edasi varasemaid ülesandeid. Leidis ka paar kursust, kus kasutati rühmatöid või paaristöid ning tunnikontrolle. Paaris programmeerimist algkursustel käsitlevad näiteks järgmised artiklid [13,14].

## Kokkuvõte

Antud bakalaureusetöö raames koguti andmeid Eesti kõrgkoolide esimeste programmeerimiskeelte kohta ning analüüsiiti neid. Leidsid 16 erinevat programmeerimise algkursust kümnes erinevas kõrgkoolis. Kõrgkoolide vastuste põhjal koostati lühikokkuvõtted kursuste kohta. Tulemustest selgus, et enamik programmeerimise algkursusi on programmeerimiskeeles Python, kuid samas on ka levinud Java. Sarnane trend on ka levimas ülejäänud maailmas. Koostati küsimustik, millele paluti kõrgkoolidel vastata. Vastuste põhjal loodi lühikokkuvõtted kursustest ning otsiti välja sarnasusi nende vahel. Mitmed kursused kasutasid videoloenguid, kuid samas osade kursuste raames ei olnud loengud ja praktikumid üldse eristatud. Uuriti ka tudengite kokkupuuteid programmeerimisega enne kõrgkooli tulemist. Vastavad andmed saadi bakalaureusetöö kirjutamise jaoks. Nendest andmetest selgus, et umbes 46% tudengitest, kes asuvad õppima infotehnoloogia seotud erialasid on varem õppinud programmeerimist, kuid samas enamikes eesti koolides programmeerimist ei õpetata. Kursused olid üsna erinevad, nii sisult kui ka suuruselt. Kursuste sisust tuli välja mitu punkti, mida oleks võimalik tulevikus edasi arendada, näiteks kursuste materjalide koostamine ning automaattestimine.

## Kasutatud kirjandus

- [1] Tartu Ülikool – Õppekohad 2014 <http://www.ut.ee/et/sisseastumine/bakalaureus/oppekohad>  
(Vaadatud 30.04.2014)
- [2] Siegfried, R. M., Greco, D., Miceli, N., & Siegfried, J. (2012). Whatever Happened to Richard Reid's List of First Programming Languages?. *Information Systems Education Journal*, 10(4), 24. <http://isedj.org/2012-10/N4/ISEDJv10n4p24.pdf> (Vaadatud 9.05.2014)
- [3] Vambola Leping, Marina Lepp, Margus Niitsoo, Eno Tõnisson, Varmo Vene, Anne Villem  
Python Prevails
- [4] Mason, R., & Cooper, G. (2014). Introductory Programming Courses in Australia and New Zealand in 2013-trends and reasons. <http://crpit.com/confpapers/CRPITV148Mason.pdf>  
(Vaadatud 9.05.2014)
- [5] Haridussilm – <https://www.haridussilm.ee> (Vaadatud 5.05.2014)
- [6] Kõrgkoolid – eesti.ee [https://www.eesti.ee/est/kontaktid/korgkoolid\\_1](https://www.eesti.ee/est/kontaktid/korgkoolid_1) (vaadatud 5.05.2014)
- [7] Programmeerimise õpik – Tartu Ülikool <https://programmeerimine.cs.ut.ee/> (Vaadatud 20.05.2014)
- [8] Sotsiaalkindlustusamet – Peretoetuse maksmine õppeaasta alguses. <http://www.sotsiaalkindlustusamet.ee/peretoetuste-maksmine-oppeaasta-alguses/?op=print>  
(Vaadatud 1.05.2014)
- [9] Spacco, J., Hovemeyer, D., Pugh, W., Emad, F., Hollingsworth, J. K., & Padua-Perez, N. (2006, June). Experiences with marmoset: designing and using an advanced submission and testing system for programming courses. In ACM SIGCSE Bulletin (Vol. 38, No. 3, pp. 13-17). ACM. <http://dl.acm.org/citation.cfm?id=1140131> (Vaadatud 18.05.2014)

[10] Ihantola, P., Ahoniemi, T., Karavirta, V., & Seppälä, O. (2010, October). Review of recent systems for automatic assessment of programming assignments. In Proceedings of the 10th Koli Calling International Conference on Computing Education Research (pp. 86-93). ACM.

<http://dl.acm.org/citation.cfm?id=1930480> (Vaadatud 19.05.2014)

[11] de Souza, D. M., Maldonado, J. C., & Barbosa, E. F. (2011, May). ProgTest: An environment for the submission and evaluation of programming assignments based on testing activities. In Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on (pp. 1-10). IEEE.

[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5876088&url=http%3A%2F%2Fieeexplore.ieee.org%2Fexpls%2Fabs\\_all.jsp%3Farnumber%3D5876088](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5876088&url=http%3A%2F%2Fieeexplore.ieee.org%2Fexpls%2Fabs_all.jsp%3Farnumber%3D5876088) (Vaadatud 20.05.2014)

[12] Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. ACM SIGCSE Bulletin, 39(2), 32-36. <http://dl.acm.org/citation.cfm?id=1272879> (Vaadatud 20.05.2014)

[13] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002, February). The effects of pair-programming on performance in an introductory programming course. In ACM SIGCSE Bulletin (Vol. 34, No. 1, pp. 38-42). ACM. <http://dl.acm.org/citation.cfm?id=563353> (Vaadatud 20.05.2014)

[14] Braught, G., Wahls, T., & Eby, L. M. (2011). The case for pair programming in the computer science classroom. ACM Transactions on Computing Education (TOCE), 11(1), 2. <http://dl.acm.org/citation.cfm?id=1921609> (Vaadatud 20.05.2014)

[15] A Very Quick Comparison of Popular Languages for Teaching Computer Programming <http://www.ariel.com.au/a/teaching-programming.html> (Vaadatud 15.05.2014)

[16] Sleiman Rabah, Jiang Li, Mingzhi Liu, Yuanwei Lai A Comparative studies of programming languages <http://arxiv.org/ftp/arxiv/papers/1009/1009.0305.pdf> (Vaadatud 15.05.2014)

[17] PHP: The Ins and Outs, The Pros and Cons <http://talener.com/php-the-ins-and-outs-the-pros-and-cons/> (Vaadatud 15.05.2014)

[18] Visual Basic – Wikiversity

[http://en.wikiversity.org/wiki/Visual\\_Basic#Pros\\_and\\_Cons\\_of\\_VB6](http://en.wikiversity.org/wiki/Visual_Basic#Pros_and_Cons_of_VB6) (Vaadatud 15.05.2014)

## Lisad

### Lisa 1 – Andmed küsimused koolidele

<b>Aine</b>	
<b>Millistele erialadele kohustuslik?</b>	
<b>Mitmendal semestril see aine reeglina läbitakse?</b>	
<b>Programmeerimiskeel</b>	
<b>Aine maht</b>	
<b>Aine jaotus tundides (loengud, praktikumid, iseseisev töö)</b>	
<b>Avalikud õppematerjalid</b>	
<b>Hindamise vorm</b>	
<b>Ligikaudne keskmine üliõpilaste arv selles õppeaines ühel õppeaastal</b>	

1. Milline õppeaine on selle õppeaine loogiliseks jätkuks?
2. Kuivõrd kasutatakse programmide automaattestimist? Milliseid vahendeid?
3. Kui palju on ülesannete hulgas mõnevõrra suuremaid - mitut nädalat hõlmavaid (projekte)?
4. Kui palju rakendatakse paaristöid, rühmatöid?
5. Kui palju kasutatakse suhteliselt väikesi ülesandeid (nt. tunnikontrollides)?
6. Kui kohustuslik on loengutes osalemine? Kas loengus võib kasutada sülearvuteid? Kas loengu videod on osalejatele kättesaadavad?
7. Kui kohustuslik on praktikumides osalemine?
8. Mis põhjustel on valitud aluseks just see programmeerimiskeel?

9. Kui kaua see keel on juba algkursusel kasutusel olnud? Mis keel oli eelmine? Mis tingis ülemineku uuele keelele?
10. Mil määral on keele valikul arvestatud tööstuse trendidega?
11. Kas programmeerimiskeele valikut mõjutas kättesaadavate materjalide hulk.
12. Mil määral on keele valikul arvestatud erialaga (näiteks sellega, kas programmeerimine on nende põhitegevus)?
13. Kuidas on arvestatud tudengite varasemate oskustega? (Tudengid, kes oskavad juba seda programmeerimiskeelt ja tudengid, kes ei ole kunagi kokku puutunud.)
14. Missuguseid materjale veel kasutatakse?
15. Kuivõrd kasutatakse eestikeelseid materjale? Inglisekeelseid? Venekeelseid?
16. Mis kujul on materjalid tudengitele kättesaadavad? (Õpik, videoloengud, ...)
17. Kas on vajadust lisamaterjalidele?
18. Kas on mingeid kursuse elemente (materjale, vahendeid, didaktilisi võtteid vm.), mis teie arvates võiks leida rakendust ka teiste kõrgkoolide kursustel?
19. Kas on mingeid asju, mis teie kursusel veel võiks olla? Kas saaksime neid koostöös ette valmistada?

## Lisa 2 – Programmeerimiskeelte võrdlus

### C

```
#include<stdio.h>

main()

{
    printf("Hello, World!");
}
```

C programmeerimiskeel võib pakkuda väga head jõudlust ning see võimaldab ka madala taseme juurdepääsu riistvarale, kuid tavaliselt algajatel selliseid asju vaja ei ole. C-s kasutatavad viidad on tihti segaduse ja nõrdimuse allikaks algajatel, kuid need on vajalikud isegi triviaalsetes programmides. C puhul tuleb selgitada palju C kohta käivaid asju lisaks programmeerimise õpetamisele [15]. Samas C on tähtis ning võimalusterohke programmeerimiskeel.

### C#

```
public class Hello_World
{
    public static void Main()
    {
        System.Console.WriteLine("Hello, World!");
    }
}
```

C# loodi spetsiaalselt .NET platvormi jaoks. Selle eesmärgiks on pakkuda lihtsat, võimsat ja tugevalt tüübitud programmeerimiskeelt, mis lubaks kiiresti ehitada rakendusi .NET platvormile [16].



## Java

```
class Hello_World {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Java on kasulik keel eri platformidel töötavate graafiliste kasutajaliideste arendamiseks, objektorienteeritud arenduseks ning Java on suur kogumik klassiteeke. Samas tekitab suur klassiteekide arv olukorra, kus palju Java programmeerimisest on lihtsalt õige klassi otsimine. Java nõuab objektorienteeritust, erindite kontrollimist ja ranget tüüpimist. Need on kõik head omadused suurte süsteemide loomisel, kuid väiksemate ülesannete puhul on need aeganõudvaks koormaks. Samas Pythoni eeliseks on, et see on väga laialdaselt levinud keel ning tööturul on selle järgi nõudlust [15].

## PHP

```
<?php echo '<p>Hello, World!</p>'; ?>
```

PHP-d kasutatakse veebiarenduseks. PHP kasutatakse üsna laialdaselt, kuna seda on üsna lihtne õppida ning internetis on üsna palju koodinäiteid saadaval. Samas koodinäited ei ole alati usaldusväärsed. PHP-d peeteks ka vähem turvaliseks kui Javat. [17]

## Python

```
print("Hello, World!")
```

Python vajab vähem aega, koodiridu ning väiksemat arvu õpetatud põhitõdesid ülesannete lahendamiseks. Seetõttu on rohkem aega tegeleda tähtsate asjadega. Samuti läheb Python mööda paljudest algajate programmeerijate põhivigadest nagu puuduvad semikoolonid, lõpetavad sulud, tüüpimine ning mälu jagamine.[15]

## Scratch



Scratchi puhul on tegemist visuaalse programmeerimisega. Scratchi on väga kerge õppida programmeerimine seisneb plokkide kokku panemises. Plokkide kuju on tehtud selliselt, et need mahuks kokku ainult õige süntaksi korral. Samas Scratchiga ei ole võimalik koostada vähegi keerukamaid programme.

## Visual Basic

```
Module Hello_World
  Sub Main()
    MsgBox("Hello, World!")
  End Sub
End Module
```

Visual Basicu üheks suureks eeliseks on, et see on saadaval *Microsoft Office*'ga. Visual Basic on kasutab lähenemist programmeerimisele, mis keskendub tarkvaraarenduse kiirusele ja lihtsusele.

[18]

## Litsents

### **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina **Karl Peedosk** (sünnikuupäev: 16.11.1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Eesti kõrgkoolide programmeerimise algkursused**, mille juhendaja on Eno Tõnisson,
  - 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, **20.05.2014**