

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Stina-Marie Maripuu**

# **Java õppematerjalide loomine gümnaasiumiastmele**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Kristi Salum

Tartu 2024

## **Java õppematerjalide loomine gümnaasiumiastmele**

### **Lühikokkuvõte:**

Java on üks populaarsemaid programmeerimiskeeli maailmas. Võrreldes ingliskeelsete materjalidega, on eestikeelsete Java õppematerjalide hulk pea olematu. Bakalaureusetöö raames loodi E-koolikoti õppematerjalid “Java programmeerimiskeele õppematerjal algajale”, mis annab ülevaate Java programmeerimiskeelest ning programmeerimise algteadmistest nagu muutujad, andmetüübid, tingimuslauseid ning tsükliid. Lisaks E-koolikoti materjalidele loodi kursuse alusfailide allalaadimise jaoks GitHubi repositoorium, mis sisaldab näidiskoodi ja ülesandeid. Materjalide loomisel tugineti Dick ja Carey disainimudelile. Loodud materjale katsetati 24 õpilasega, kes andsid materjalidele üldiselt positiivse hinnangu. Õpilaste tagasiside põhjal tehti materjalides parandusi ning pakuti välja tulevikuvõimalusi.

### **Võttesõnad:**

Programmeerimine, gümnaasiumiaste, õppematerjal, tagasiside, Dick ja Carey disainimudel

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

## **Creating Java Study Materials for Secondary School Level**

### **Abstract:**

Java is one of the most popular programming languages in the world. Compared to study materials in English, the amount of Java study materials in Estonian is almost non-existent. As a part of this thesis, study materials in E-koolikott were created, named “Java Programming Language Study Materials for beginners”. The materials give an overview of Java language and basic programming skills such as variables, data types, conditional statements, and loops. In addition to the E-koolikott materials, a GitHub repository was created to store course base files, which include sample codes and exercises. The creation of the materials was based on the Dick and Carey design model. The created materials were tested with 24 students. The general feedback for the materials was positive. Based on the students’ feedback, improvements were made to the materials, and future improvements were proposed.

### **Keywords:**

Programming, secondary school level, study materials, feedback, Dick and Carey design model

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

# Sisukord

<b>Sissejuhatus</b>	<b>4</b>
<b>1. Programmeerimise õppematerjalid</b>	<b>5</b>
1.1. Veebist leitavad ingliskeelsed Java programmeerimismaterjalid	5
1.2. Veebist leitavad eestikeelsed Java programmeerimismaterjalid	6
1.3. Veebis olemasolevate materjalide kokkuvõte	7
<b>2. Õppematerjalide loomine</b>	<b>8</b>
2.1. ADDIE mudel	8
2.2. Kempfi mudel	8
2.3. Dick ja Carey mudel	9
2.4. Mudeli valik	9
<b>3. Materjalide loomine</b>	<b>11</b>
3.1. Materjalid	11
3.2. Kirjalikud materjalid	11
3.3. GitHubi materjalid	11
3.4. Teemade valik	12
<b>4. Materjalide katsetamine ning tulemused</b>	<b>14</b>
4.1. Näidistund	15
4.2. Eeltest	16
4.3. Tagasiside küsimustik	19
4.4. Tulemused	21
<b>Kokkuvõte</b>	<b>23</b>
<b>Viidatud kirjandus</b>	<b>24</b>
<b>Lisad</b>	<b>25</b>
I. Materjalide sisukord	25
II. Litsents	26

## Sissejuhatus

Java on üks populaarsemaid programmeerimiskeeli maailmas [1]. Selle mitmekülgne kasutusviis, olgu selleks lihtsate veebirakenduste loomine või suuremahuliste tarkvaralahenduste arendamine, teeb Javast väga universaalse programmeerimiskeele, mida iga gümnaasiumiõpilane võiks osata. Oskus programmeerida võimaldab õpilasel tulevikus suuremat valikut tööturul, arendab loogilist mõtlemist, annab uue perspektiivi probleemide lahendamiseks ning avardab silmaringi [2].

Käesoleva töö eesmärgiks on luua Java õppematerjalid gümnaasiumiastmele, et anda õppijatele võimalus antud programmeerimiskeele baasteadmiste omandamiseks. Materjalide läbimisel saab õpilane aru, mida kujutab endast Java programmeerimiskeel ja õpib läbi selgitavate materjalide ning praktiliste ülesannete programmeerimist. Lisaks materjalide loomisele on töö teiseks eesmärgiks loodud materjalide kohta tagasiside kogumine, analüüs ja seeläbi materjalide täiendamine.

Bakalaureusetöö koosneb neljast osast. Esimeses osas antakse ülevaade veebis olemasolevatest nii eesti- kui ka ingliskeelsetest Java õppematerjalidest. Teises osas tutvustatakse õppematerjalide loomise mudeleid. Kolmandas peatükis antakse ülevaade loodud materjalidest ning nende loomise protsessist. Töö viimases peatükis tutvustatakse materjalide põhjal läbi viidud näidistunde, analüüsitakse tunnis materjale katsetanud õpilaste tagasisidet ning antakse ülevaade tehtud parandustest ning edasi arendamise võimalustest.

## 1. Programmeerimise õppematerjalid

Avalikult kättesaadavad ja kasutajasõbralikud õppematerjalid on väga kasulikud inimestele, kellel on huvi programmeerimise vastu. Programmeerimisega alustada sooviv õpilane ei pruugi aga teada kust, mis keelest ja kuidas alustada. Sellisel juhul on oluline, et algajal oleks ligipääs võimalikult paljudele erinevatele programmeerimiskeelte materjalidele, et selle põhjal oma valik teha. Et algajal oleks motivatsiooni ning tahtmist õppimist alustada, on oluline, et tegemist oleks visuaalselt atraktiivse, eestikeelse ning põhiteadmistest alustava kursusega. Järgnevalt tutvustatakse erinevaid olemasolevaid programmeerimise õppematerjale, nii eesti kui inglise keeles, ning võrreldakse nende häid ja halbu külgi.

### 1.1. Veebist leitavad ingliskeelsed Java programmeerimismaterjalid

Veebis olemasolevad materjalid jagunevad enamasti tasuta kättesaadavaks ja tasulisteks. Inglisekeelsetest materjalidest vaatleme Java programmeerimiskeelt tutvustavaid juhendeid W3Schoolsi Java Tutorial<sup>1</sup>, Programiz keskkonnas Learn Java Programming<sup>2</sup>, Geeks for Geeks Java Tutorial<sup>3</sup> ning Codecademy Learn Java<sup>4</sup>. Kõik näitena toodud veebilehed on populaarsed ingliskeelsete programmeerimise materjalide ja juhendite keskkonnad ning on head näited Java õpetamise juhenditest, mis on tasuta kättesaadavad ning kasutajasõbralikud.

Teemad algavad enamasti süntaksi näidetest ja selgitustest ning seejärel vaadatakse üle erinevad sõnede, numbritega, operaatoritega jne seotud tegevused. Juhendites on teemad selgitatud algaja jaoks arusaadavalt ning iga teema juurde on lisatud ka ohtralt näiteid. Lisaks näidetele on teemade lõpus enesekontrollitised, et õppija saaks uusi teemasid paremini kinnistada. Näitena toodud materjalide ülesehitused on omavahel veidi erinevad. W3Schools, Programiz ja Geeks for Geeks materjalid on loogilise teemade järjekorraga ning neid on võimalik läbida Java õppimise juhendina, mis annab algajale hea ülevaate põhiteadmistes. Need keskkonnad on ka väga populaarsed kogemustega programmeerijate seas, sest need sisaldavad rohkelt näiteid ning mõnel juhul võivad anda funktsioonidest parema ülevaate, kui originaalne dokumentatsioon. Codecademy on üles ehitatud kursusena: järgmise teema juurde pole võimalik edasi liikuda enne, kui eelmine on sooritatud. Lisaks on Codecademys võimalik saada automaatkontrolli testidega oma koodile tagasisidet, mõningaid näpunäiteid,

---

<sup>1</sup> <https://www.w3schools.com/java/>

<sup>2</sup> <https://www.programiz.com/java-programming>

<sup>3</sup> <https://www.geeksforgeeks.org/java/>

<sup>4</sup> <https://www.codecademy.com/learn/learn-java>

kuidas paremat koodi kirjutada ning soovi korral kursuse lõpus endale ka kursuse läbimise sertifikaat soetada.

Lisafunktsionaalsusena on võimalik kõikides ülaltoodud keskkondades jooksutada Java koodi õppematerjali kõrval. See teeb uue programmeerimiskeele õppimise kordades lihtsamaks, kuna huvilised saavad kohe materjalide sees katsetada koodi ning seda ise muuta, mis teeb koodist aru saamise protsessi kiiremaks ja hõlpsaks. Lisaks tähendab see, et õpilasel ei pea ilmtingimata arvutis koodi kirjutamise keskkond olemas olema.

Veebist ligipääsetavate materjalide nimekirja põhjal on aru saada, et ingliskeelsete materjalide kättesaadavus on piisav - seega pole vajadust vaadelda paberkandjal olevaid versioone.

## **1.2. Veebist leitavad eestikeelsed Java programmeerimismaterjalid**

Võrreldes ingliskeelsete veebist leitavate materjalidega, on eestikeelseid Java õppematerjale veebis väga vähe. Valdav enamus Java materjalidest on avalikud õppematerjalid ülikoolide või koolide kursustele ja ainetele. Materjalid, mis ei ole seotud ülikooliga ning mida pakuvad erinevad koolitusettevõtted, on enamasti tasulised. Tasulised kursused on näiteks BCS koolitus Java programmeerimine (e-õpe)<sup>5</sup>, mille hinnaks on 250€ + käibemaks ja TÜ Teaduskooli E-õppekeskuse kursus Programmeerimine keeltes C/C++ ja Java<sup>6</sup> hinnaga 30€. Tasuta kursustest on veebist leitavad Mario Metsheina Java programmeerimise kursus<sup>7</sup>, TalTechi Java õppematerjalid<sup>8</sup> ning Tartu Ülikooli Objektorienteeritud programmeerimise kursus<sup>9</sup>. Loetletud materjalidest on vaid Mario Metsheina kursus ülikooliväline. Visuaalselt on materjalid sarnased: selgitav tekst, paar koodirida ning mõned olulisemad terminid või funktsioonid on rasvases, teist värvi või teise fondiga kirjas. Koodi jooksutamise funktsionaalsust üheski keskkonnas ei ole.

Arenduskeskkondadena kasutatakse kummagi ülikooli materjalides IntelliJ IDEAd, seda sellepärast, et tudengitel on võimalus kasutada IntelliJ Ultimate rakendust õppe-eesmärkideks tasuta [3]. Mario Metsheina materjalides on keskkonnana kasutuses Eclipse IDE<sup>10</sup>. Samuti tasub mainida, et kõige ajakohasemad on ülikoolide materjalid, kuna Mario Metsheina materjalides on veel toodud näiteid Windows 8 ning varasemate operatsioonisüsteemidesse

---

<sup>5</sup> <https://www.bcskoolitus.ee/koolitus/java-programmeerimine-e-ope/>

<sup>6</sup> <https://e-oppekeskus.ee/kursused/programmeerimine-keeltes-c-c-ja-java/>

<sup>7</sup> <https://www.metshein.com/course/java-programmeerimine/curriculum/>

<sup>8</sup> <https://javadoc.pages.taltech.ee/index.html>

<sup>9</sup> <https://courses.cs.ut.ee/2024/OOP/spring>

<sup>10</sup> <https://eclipseide.org/>

allalaadimisest. Materjalide interaktiivsuse mõttes on kindlasti kõige atraktiivsemad Tartu Ülikooli omad, kuhu on lisatud ka H5P<sup>11</sup> liidesega enesekontrolliteste teksti sisse.

### 1.3. Veebis olemasolevate materjalide kokkuvõte

Inglisekeelseid materjale võrreldes olid esindatud hästi nii kursuse kui üksiku teema põhised juhendid, mis olid samal ajal ka visuaalselt ilusad. Positiivne oli keskkonnas koodi jooksutamise võimalus ning hea ülevaade materjalide sisust ja keskkonna võimalustest.

Eestikeelseid Java materjale on vähem ja nende kasutusvõimalused on piiratumad. Eestikeelsed materjalid jagunesid peamiselt tasuta kättesaadavateks ülikooli kursuste juhenditeks ning tasulisteks kursusteks. Kuna veebist kättesaadavad materjalid on valdavalt mõeldud konkreetsete kursuste jaoks, nõuab nende ülesehitus kõikide teemade järjest läbimist. Visuaalsele poolele pole liialt rõhku pandud, kuna nende eesmärk ei ole olla suunatud laiale publikule ega algajale esimeseks iseseisvaks õppematerjaliks. Võrreldes ingliskeelsete materjalidega jääb eestikeelne valik kindlasti alla ning algajasõbralik, visuaalselt rahuldav ning iseseisvat õppimist toetav materjal on eesti keeles Java kohta puudu.

Pythoni puhul rahuldab neid kriteeriume põhikooli õpilaste jaoks loodud programmeerimise põhiteadmisi Pythoni näitel tutvustav veebikeskkond progema.ee<sup>12</sup>. Materjal valmis Kristiina Kepsi magistritööna Tartu Ülikoolis ning käsitleb teemasid nagu algoritm, funktsioonid, tingimuslause, andmestruktuurid ning objektorienteeritud programmeerimine. Seda materjali on algajal lihtne järgida, kuna tekst on struktureeritud, illustreeritud ning lisatud on asjakohaseid abistavaid näiteid [7].

Võrreldes Pythoniga on Java programmeerimiskeel mõnevõrra keerulisema süntaksi ning reeglitega. See teeb ta heaks teiseks programmeerimiskeeleks, mida võiks gümnaasiumi õpilane õppida. Materjalide loomisel on kindlasti oluline faktor kättesaadavus ning kasutajasõbralikkus. Alates 2015. aastast täidab õppekirjanduse registri rolli E-koolikott [8]. E-koolikoti eesmärgiks on võimaldada tasuta ligipääs õppevarale ühest kohast, et muuta kasutaja jaoks materjalide leidmine ning kasutamine lihtsamaks [8]. Kuna eestikeelsetest Java keeles programmeerimine materjalidest on suur puudus, oleks üheks lahenduseks E-koolikoti funktsionaalsuste kasutamine, mis tagab lihtsa ligipääsu nii loojale kui kasutajale.

---

<sup>11</sup> <https://sisuloome.e-koolikott.ee/>

<sup>12</sup> <https://www.progema.ee/>

## **2. Õppematerjalide loomine**

Õppematerjalide loomiseks on olemas mitmeid erinevaid õppematerjalide koostamise mudeleid [4]. Need mudelid ja protsessid on loodud eesmärgiga teha materjalide loomine võimalikult efektiivsesks ning analüüsivaks [5]. Hans Põldoja on loonud ülevaatliku artikli õppematerjalide koostamise protsessist ning populaarsetest mudelitest. Järgnevalt antakse ülevaade mõningatest Põldoja [4] mainitud mudelitest. Nendest mudelitest valitakse sobivaim, mis võetakse aluseks materjalide loomisel.

### **2.1. ADDIE mudel**

Õppematerjalide koostamiseks on mitmeid lähenemisi, üheks tuntumaks neist on ADDIE mudel [4]. ADDIE on ingliskeelne akronüüm, mis väljakirjutatuna tähendab: analüüsi etappi (ingl Analysis), kavandamise etappi (ingl Design), väljatöötamise etappi (ingl Development), kasutamise etapp (ingl Implementation) ja hinnangu andmise etapp (ingl Evaluation) [4].

Järgnev kokkuvõtte meetodist tugineb Põldoja [4] artiklil. Analüüsi etapis tehakse eeltöö õppematerjalide loomiseks. Pannakse paika sihtrühm ning mõeldakse läbi, mida materjalidelt vajatakse ning luuakse projektiplaan. Kavandamise etapis keskendutakse materjalide eesmärkidele ning teostuse plaanile (eelarve, testimine). Väljatöötamise etapp on mõeldud materjalide koostamiseks ning testimiseks. Luuakse planeeritud sisu, juhendid kasutamiseks ning viiakse läbi testimine, et leida materjalides puudulikke kohti, mida hiljem juhendis täiendatakse. Peale testimisi tuleb kasutamise etapp, kus tutvustatakse materjale õpilastele ning õpetajatele. Viimaseks etapiks on hinnangu andmine, kus määratletakse hinnangukriteeriumid, -vahendid ning viiakse läbi hindamine. Kvaliteedi hindamiseks kasutatakse laialdaselt LORI ja LOE-S hindamismudeleid. LORI mudel on mõeldud mitme eksperdi koondhinnangu loomiseks, kus kõik edastavad oma hinnangu õppematerjale tuginedes üheksale aspektile ning hinnates neid 5-astmelisel skaalal. LOE-S mudel on mõeldud autorile enesehindamiseks, kus tagasiside pärineb õpilastelt 12 väite kohta 5-astmelisel skaalal.

### **2.2. Kempi mudel**

Kempi mudelit iseloomustab selle ringlev loomus, kus materjalide väljatöötamine on pidev protsess ja mis koosneb üheksast osast [5]:

1. Probleemi sõnastus ja eesmärkide kirjapanek.
2. Õpilastele oluliste faktorite välja toomine.



3. Selgitada välja materjali sisu ning viia see vastavusse loodud eesmärkidega.
4. Sõnastada õpiväljundid.
5. Korrastada materjalide sisu loogilisse järjekorda.
6. Luua materjalide kasutamise loogika, eesmärgiga kõigile õpilastele teema selgeks teha.
7. Luua sissejuhatavad ja kasutamist selgitavad juhendid.
8. Luua hindamismeetodid materjalide hindamiseks.
9. Juhendamiseks ja õpetamiseks vajalike ressursside valik.

ADDIE ning Kemp'i mudel sisaldavad mitmeid sarnaseid etappe. Peamine vahe seisneb ülesannete põhjalikkuses. Kemp'i mudelis on konkreetsed sammud kindlas järjekorras.

### **2.3. Dick ja Carey mudel**

Sarnaselt Kemp'i mudelile, on ka Dick ja Carey mudel [6] ringmudel, kuhu on lisatud mitmeid ülevaatamise aspekte, et garanteerida, et materjalide lõpptulemus vastab püstitatud eesmärkidele. Dick ja Carey mudel koosneb kümnest etapist [6]:

1. Materjali eesmärgi tuvastamine.
2. Eesmärgi analüüs.
3. Eelteadmiste vajalikkuse analüüs koos sihtrühma analüüsiga.
4. Materjali sisu ja kvaliteedi sõnastamine.
5. Hindamiskriteeriumite sõnastamine.
6. Materjalide õpetamise strateegia loomine.
7. Sissejuhatavate ja kasutamise juhendite loomine, näiteks õppeväljundite saavutamiseks vajalikud sammud.
8. Loodud materjalide katsetamine ning parandus.
9. Protsessi ja materjalide ülevaatus.
10. Lõpptulemuse analüüs ning sellele hinnangu koostamine.

Võrreldes Kemp'i mudeliga, on Dick ja Carey disainimudel selgemalt tsüklilise protsessina selgitatud ning on hea arusaam ning loogiline järjestus etappide vahel.

### **2.4. Mudeli valik**

Materjalide loomise disainimudeliks valiti Dick ja Carey mudel. Kolmest mudelist on Dick ja Carey mudelis kõige paremini välja toodud arenduse tsükliline protsess. Võrreldes teiste

mudelitega, on Dick ja Carey mudelis protsessid kõige selgemalt sõnastatud ning kõikide etappide vahel on loomulik üleminek.

### **3. Materjalide loomine**

#### **3.1. Materjalid**

Loodud materjalid on avalikult leitavad E-koolikoti keskkonnast nimega “Java programmeerimiskeele õppematerjal algajale” ning ligipääsetavad veebiaadressilt <https://e-koolikott.ee/et/oppematerjal/33935-Java-programmeerimiskeele-oppematerjal-algajale>. Lisaks E-koolikoti materjalidele on kättesaadav ka Google Docsi fail, mille link on leitav E-koolikoti materjalide sissejuhatuses. Toetava materjalina on olemas ka Githubi repositoorium, kust on võimalik alla laadida kursuse jaoks vajalikud alusfailid, koodi- ja struktuurinäited ning ülesannete failid. Materjalidele on lisatud ka orienteeruvalt kaua teemade läbimine / ülesannete lahendamine aega võiks võtta, et võimaldada õpetajal paremini tundi planeerida. Kursuse materjale luues sooviti tagada, et õpilane suudab iseseisvalt ülesandele lahenduse leida. Selleks julgustatakse õpilast hätta jäädes esmalt näidiskoodi vaatama ning vajadusel ka internetist abi otsima.

#### **3.2. Kirjalikud materjalid**

Peamiseks õppevahendiks kursusel on E-koolikoti materjalid. Seal on võimalik õpilasel iseseisvalt nii tunnis kohapeal kui ka kaugõppes teemasid läbida. Õpitut aitavad kinnistada mitmed enesekontrollitised ning ülesannete juures olevad vihjed.

Õppematerjalide loomisel kasutati erinevaid H5P liideseid [8] mitmeks eri ülesandeks: paremaks koodi ja teksti vormistamiseks, kompaktsemaks piltide paigutuseks ning teksti struktuuri ja peatükkide liigendamise parandamiseks. Näidistundi läbiviies olid materjalid loodud tavalise E-Koolikoti stiiliga ning tagasiside põhjal muudeti teksti ja koodi liigendatust ning tekstile lisati värvilisi elemente ning rasvast kirja.

Lisaks E-koolikoti materjalidele on õpetajatel võimalik kasutada ka Google Docsi formaadis olevaid materjale, mis on kättesaadavad E-koolikoti materjalide sissejuhatuses. Need on mõeldud abistavaks vahendiks tunni ettevalmistamisel ning võivad anda parema arusaama materjalide mahust. E-koolikoti materjalid on koopia Docsi materjalidest.

#### **3.3. GitHubi materjalid**

GitHubi on kogu kursuse kohta lisatud kaks repositooriumi, üks õpilasele ning üks õpetajale. Sealt on võimalik saada kogu kursuse jaoks projekti alus, kust on võimalik leida nii

koodinäiteid, ülesannete alusfaile kui ka vormistusnäiteid. Õpetaja jaoks mõeldud repositooriumis on lisaks alusfailidele ka lahendusfailid. See annab ka õpetajale võimaluse valida, kas jagada kohe lahendustega repositooriumi õpilastega, kasutada etteantud lahendusi õpilaste töö kontrollimiseks või kasutada neid alusena, et koos õpilastega ühiselt kood läbi arutada ning rakendada ühisprogrammeerimise põhimõtteid.

Lisaks failide ja projektide jagamisele, annab GitHubi kursusel kasutamine võimaluse ka selle kasulikkust õpilastele näida ning julgustada neid seda juba varakult kasutusele võtma. Õpilastel on kursuse teema 2.3 juures võimalik juhendi järgi luua ka endale oma GitHubi repositoorium ning iga tunni lõpus oma töö üles laadida. See annab õpetajale võimaluse jälgida õpilaste arengut materjalide läbitöötamisel ning hinnata tehtud tööd. Õpilastel võimaldab see oma tööd teises arvutis jätkata ja see GitHubist siis alla laadida.

### **3.4. Teemade valik**

Materjalid jagunevad kuueks peatükiks: Sissejuhatus, Süntaksituvustus, Andmetüübid, Tingimuslaused, Tsüklid ning Lõputeemad.

Esimene peatükk “Sissejuhatus - Mis on Java?” annab üldise ülevaate materjalide sisust, põhjendab miks Javat võiks õppida ning mis oleksid potentsiaalsed väljundid Java programmeerimiskeelele. Lisaks teema sissejuhatamisele antakse peatükis ülevaade kursusel kasutatavatest keskkondadest, milleks on IntelliJ IDEA ning veebipõhine Java koodi kompilaator.

Enne teemade kohapealset katsetamist oli plaan teha keskkonna kasutusjuhend vaid IntelliJ IDEale, kuid tunnis tuli ilmsiks, et selle keskkonna kooliarvutitesse paigaldamine osutub liiga keeruliseks. Seega tuli lisada ka veebipõhise kompilaatori kasutamise juhend. IntelliJ keskkonna kohta antakse materjalide läbimise jooksul ülevaade keskkonna paigaldamisest, projekti ning failide loomisest ning otse IntelliJ keskkonnast failide GitHubi laadimisest. Lisaks eelnevalt mainitule julgustatakse ka õpilast kasutama IntelliJ koodi kirjutamist hõlbustavaid funktsionaalsusi, mis aitavad otseteedega pikka koodi kiiremini kirjutada, koodi lõpetamise funktsionaalsust ja koodi korrigeerimist ning vigade leidmist IntelliJ abiga.

Veebipõhise Java koodi kompilaatori kohta tuuakse esile soovitatud keskkonnad ning mõned erisused, kuidas koodi seal jooksutada. Tagasisidest tuli välja soov õpilaste poolt, et kui kasutada veebipõhist kompilaatorit, on liiga palju keskkondi (E-Koolikott, GitHub ning

kompilaator) korruga lahti, lisati E-Koolikoti peatükkide lõppu ka GitHubis olevad materjalid, et õpilastel oleks kergem koodi kompilaatorisse kopeerida. Samuti on kompilaatorisse lisatavate koodide juures oluline, et need ei sisaldaks tähemärke, mis ei kuulu US-ASCII<sup>13</sup> tähestikku, milleks eesti keeles on täpitähed. Koodi lisamine E-koolikotti elimineerib vajaduse muuta kõik kursusel kasutuses olevad GitHubi koodid täpitähetuks.

Teises, Java süntaksit tutvustavas peatükis tehakse läbi koodinäidete tutvust Javaga ning selle osadega ja eripäradega võrreldes näiteks Pythonis koodi kirjutamisega. Harjutatakse, kuidas uut faili luua, kuidas printida “Tere maailm!” ning kuidas oma koodile kommentaare lisada. Peatüki lõpetuseks tehakse tutvust GitHubiga ning see osa on oluline pigem õpilaste jaoks, kellel on võimalik IntelliJ materjale läbida. Et GitHubi kasutust ja õppimist võimalikult lihtsaks teha, on materjalidesse lisatud ka videoõpetus, kuidas endale GitHubist materjalid alla laadida. Sarnaselt allalaadimisele on olemas ka juhend kuidas luua endale isiklik GitHubi repositoorium oma töö üleslaadimiseks, et seda hiljem jätkata. Repositooriumi tegemine loob õpetajale võimaluse õpilase töö hindamiseks.

Kolmandas peatükis keskendutakse Javas väga olulisele teemale, milleks on andmetüübid. Kuna materjalides ei õpetata objektorienteeritud programmeerimist, keskendutakse vaid olemasolevatele andmetüüpidele. Iga alampeatükk keskendub mingile andmetübile, näiteks *int* ja *float*, *string* ja *char*, *boolean* ja massiivid. Kogu peatüki kohta on ka paar ülesannet, mis jooksvalt lahendades aitavad õpilasel paremini teemadest aru saada ning aitavad kinnistada loetut. Iga alapeatüki jaoks on loodud ka GitHubis vastav fail, mis sisaldab näiteid igast teemast ning seal leidub kõik vajalik, et õpilane saaks ülesanded iseseisvalt lahendatud ja ei peaks internetist koodi otsima hakkama. E-Koolikoti materjalidesse on lisatud ka H5P liidesega enesekontrolliteste, mille eesmärk on teha materjale interaktiivsemaks, kinnistada õpitut teemat ning anda julgustust, et õpilane on tekstist ja näidetest õigesti aru saanud.

Neljandas peatükis käsitletakse tingimuslauseid. Teemade kirjeldused põhinevad suuresti eelneval tõeväärtuse teemal.

Viiendas ehk tsüklite peatükis antakse ülevaade, mida tsükel endast kujutab. Peamiselt kasutatakse näitena korduvalt kirjutatud koodi lihtsamaks muutmist. Selles peatükis liigutakse tagasi eelnevate ülesannete juurde, et sealsed korduvad tegevused tsüklitega ära lahendada. Näiteks lahendatakse tsükliga sisendi korduvalt küsimine ning punktikujundite joonistamine.

---

<sup>13</sup> <https://www.charset.org/charsets/us-ascii>

## 4. Materjalide katsetamine ning tulemused

Dick ja Carey mudeli üheks etapiks on materjalide katsetamine ning paranduste tegemine. Selles peatükis antakse ülevaade näidistunni läbiviimisest, õpilaste tagasisidest materjalidele ning tehtud parandustest.

Loodud materjale katsetati Tallinna Reaalkooli 10. klassi programmeerimissuuna õpilastega programmeerimise tunni raames. Tundi viis läbi materjalide autor, et saada paremat arusaama õpilaste võimekusest materjale läbida nii iseseisvalt kui juhendamisega.

Õpilaste valimis oli 24 õpilast, kes jagunesid kahte rühma. Esimeses rühmas oli 7 ning teises 17 õpilast. Kummagi rühma tunni pikkus oli 2x45 min ilma vahetunnita. Rühmade suuruste erinevus andis võimaluse katsetada tunni läbimist erinevate meetoditega.

Õpilaste arvamustest ja hinnangust parema ülevaate saamiseks täitsid õpilased ära nii eeltesti kui tagasiside küsitluse. Eeltesti täitis 24 õpilast ning tagasiside küsitluse täitis 22, kaks õpilast ei jõudnud tagasisidet ära esitada. Eeltesti eesmärgiks oli õpilaste programmeerimisalaste teadmiste taseme väljaselgitamine. Kuna tegemist oli programmeerimissuuna õpilastega, alustasid nad 10. klassi alguses Pythoni õppimist kasutades Tartu Ülikooli Arvutiteaduse instituudis loodud Programmeerimise õpikut [progeopik.cs.ut.ee](http://progeopik.cs.ut.ee) [9].

Materjalide katsetamise tunniks olid õpilased jõudnud läbida enamuse õpikust. Tänu sellele oli nende jaoks enamus teemasid, mida loodud materjalides käsitleti, tuttavad ega vajanud suurt üleseletamist. Vaatamata sellele, et tegu oli juba Pythonis programmeerimisega kokku puutunud õpilastega, ei olnud neil materjale lugedes igav ning nad said keskenduda süntaksist arusaamisele ning Java programmeerimiskeele õppimisele. Lisaks oskasid nad eelnevatele teadmistele toetudes anda ka näpunäiteid, kuidas nende arvates kõige paremini mingeid teemasid võiks edasi anda ning mille selgitus materjalides küsimusi tekitas. Näiteks tuleks paremini selgitada, mida termin “süntaks” tähendab ning kuigi objektorienteeritud programmeerimist materjalides ei käsitleta, oli seda mitmes kohas mainitud, mis tekitas õpilastes pigem segadust.

Objektorienteeritud programmeerimise väljajätmise soovitus oli väga kasulik tähelepanek, sest algselt materjale luues oli seda mainitud eesmärgiga avardada õpilaste teadmisi ning tuua sisse kasulikke näpunäiteid, et tulevikus teemaga kokku puutudes, oleks aru saada millega tegu on. Peale tagasiside saamist vaadati kõik materjalid üle ning lisati pikem osa, et objektorienteeritud programmeerimist mainitaks ainult paaris kohas, seejuures antakse

ülevaade sellest, mida see endast kujutab, kuidas seda kasutatakse ning mis juhul see kasulik võiks olla.

#### **4.1. Näidistund**

Näidistundi planeerides võeti arvesse, et tegu on juba programmeerimisega tuttavate õpilastega. Oluline oli võtta piisavalt aega eeltesti ning tagasiside küsitluse täitmiseks. Eeltest paluti täita ilma igasuguse eelneva tutvustusega, et saada mõjutamata sisend. Peale eeltesti lõpetamist paluti õpilastel samuti võtta lahti juba tagasiside küsitlus, et julgustada kõik arvamused ning tähelepanekud kohe kirja panna. Järgnevalt tehti üldine sissejuhatus Javasse, mis hõlmas kiiret Java ajalugu ning arengu kirjeldust, õpilaste enda võimalikku kokkupuudet Javaga läbi Java Appletite [10], mida kunagi kasutasid erinevad arvutimängude veebilehed, Java ning Pythoni lühikest ülevaatlisku võrdlust ning hinnangut, miks on Java õppimine kasulik.

Tunni jooksul polnud võimalik kasutada IntelliJ IDEAd, seega kasutati Java veebipõhist kompilaatorit ülesannete lahendamiseks. Materjalide läbimiseks anti õpilastele paar minutit aega, et jõuda järgmise ülesandeni ning siis minna ja tutvuda GitHubis oleva ülesande ning teksti toetava koodiga. Õpilased said hakata iseseisvalt ülesannet lahendama ning mõne aja pärast, kui kõik olid saanud omaette ülesandega tutvuda, hakati üheskoos juhendajaga seda üle vaatama.

Tundi viidi läbi kahe erineva rühmaga, esimeses rühmas oli 7 ning teises 17 õpilast. Grupi suuruste tõttu oli võimalik proovida erinevaid õpetamisstiile. Kuna esimene grupp oli väiksem, oli võimalik nendega rakendada ühisprogrammeerimist ning anda neile võimalus kasutada IntelliJ keskkonda juhendaja arvutis. Õpilastele anti aega materjalide ja teemadega kokku puutuda ning hiljem kordas juhendaja tähtsamad osad üle. Ekraanipilt edastati projektoriga tahvlile ning õpilased said vaheldumisi arvutis kirjutada koodi ning koos arutades lahenduseni jõuda. Juhendaja andis vihjajaid näpunäiteid, mida IntelliJ enda teated tähendavad ning kui mingi osa jäi segaseks, suunati õpilased näidiskoodi vaatama ning soovi korral ka guugeldama. Selline programmeerimisstiil paistis õpilastele meeldivat, kuna tekkis avatud vestlus ning neil oli lihtsam küsimusi küsida.

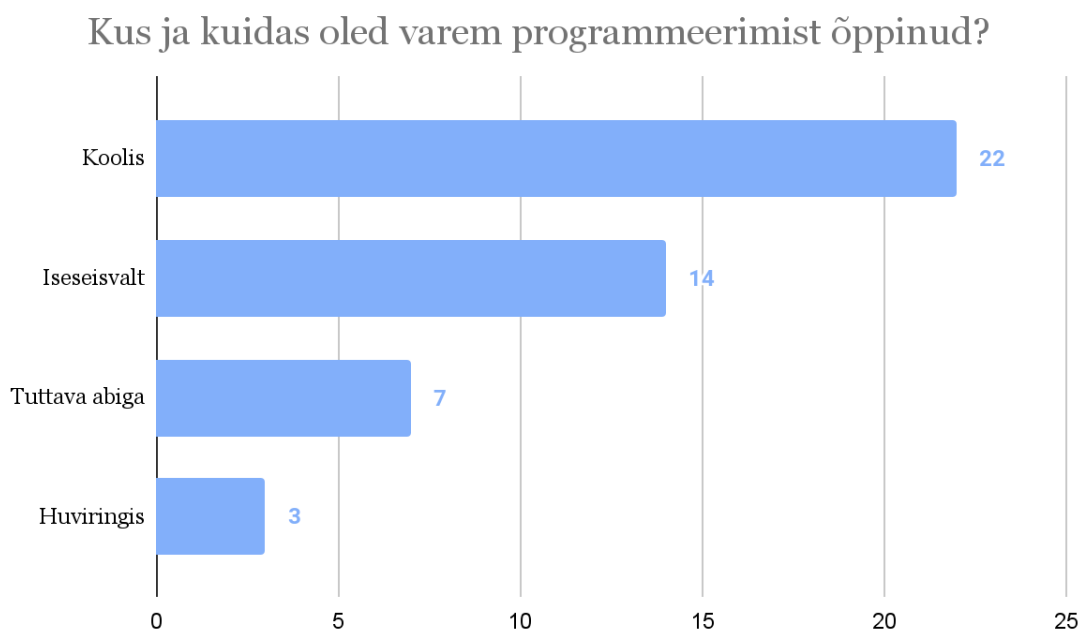
Teises grupis oli palju rohkem õpilasi - seega ühisprogrammeerimine ei oleks selle grupiga mõttekas olnud. Sarnaselt esimese grupiga, said õpilased iseseisvalt tutvuda materjalidega. Seekord aga suunati nad iseseisvalt ülesannet lahendama ning peale mõningat pusimist vaadati üheskoos juhendajaga ülesanne ning selle lahendus üle. Projektoriga näidati IntelliJ

keskkonda, et tekiks arusaam, mida selle keskkonna kasutamine endast ette kujutab. Ühine läbivaatus andis samuti kinnitust mõnele õpilasele, et oli ka enda ülesande õigesti lahendanud ning klassilt suunavate küsimuste küsimine lõi keskkonna, kus rohkem õpilasi julgesid selgitavaid küsimusi küsida, ning aitas kinnistada õpitut.

## 4.2. Eeltest

Enne tunni sissejuhatavat osa paluti kõigil õpilastel täita eeltest. Test viidi läbi enne tunni algust, et saada võimalikult mõjutamata vastused. Eeltesti eesmärgiks oli saada ülevaade näidistunnis osalevate õpilaste programmeerimisalastest eelteadmistest. Lisaks senistele teadmistele programmeerimiskeelte kohta uuriti, kust ja kuidas nad programmeerimist olid õppinud, mis programmeerimiskeeltest nad tulevikus huvitatud oleksid ning kas nad õpivad enamasti eesti-, inglise- või venekeelsetest materjalidest. Kuna käesoleva töö eesmärgiks on luua mitmekülgsed materjalid, et neid saaks kasutada nii iseseisvaks kui tunniõppeks, oli oluline saada paremat ülevaadet mis keskkondi ning meediume õpilased enim kasutavad. Eeltestile vastas 24 õpilast.

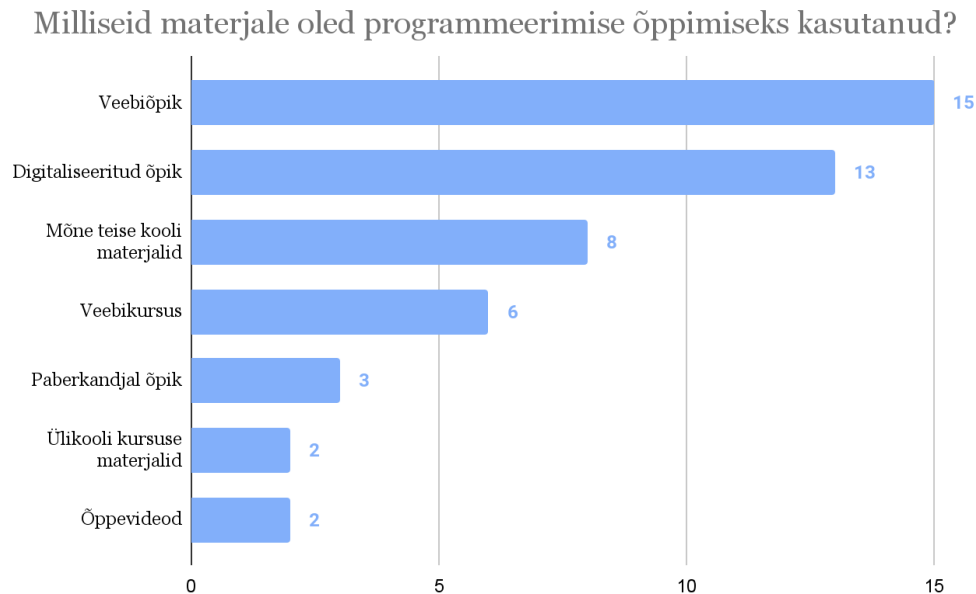
Vastanud õpilaste seast olid kõik programmeerimisega varasemalt kokku puutunud. Kaks oskasid enda hinnangul hästi programmeerida, 17 õpilast tundis et oskab mingeid asju, aga ei tunne ennast veel kindlalt ning 5 on natukene õppinud programmeerimist.



Joonis 2. Kokkupuude programmeerimisega

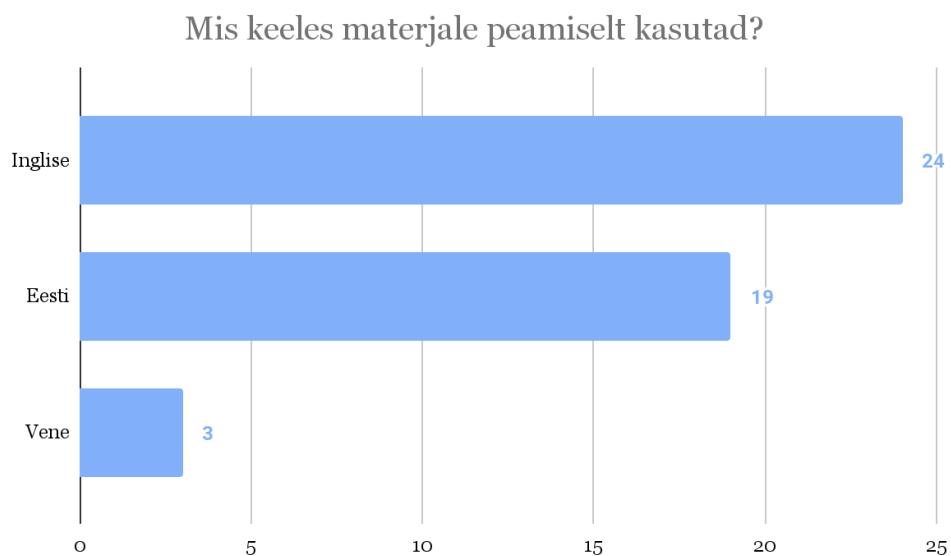


Valikvastustega küsimusele “Kus ja kuidas oled varem programmeerimist õppinud?” vastanutest oli 22 õpilast, kes olid õppinud programmeerimist koolis, 14 iseseisvalt, 7 tuttava abiga ning 3 huviringis (Joonis 2).



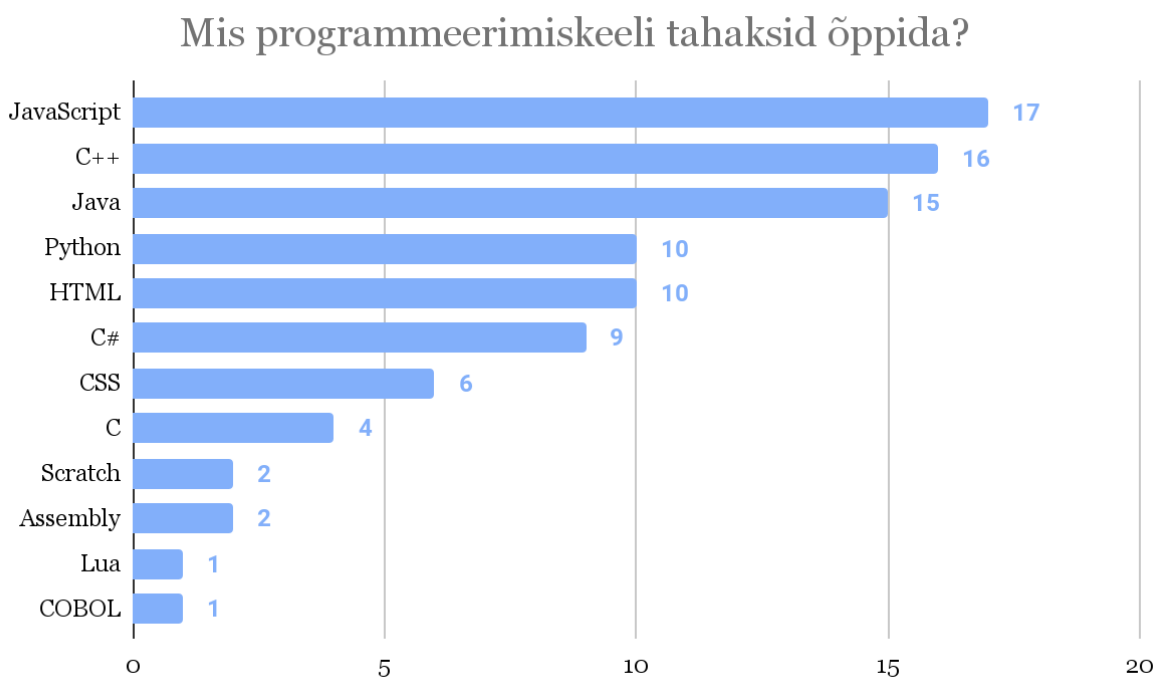
Joonis 3. Programmeerimise õppimiseks kasutatud materjalid

Õppimiseks kasutasid veebiõpikut nagu progeõpik.cs.ut.ee, W3Schools ning progema.ee 15 õpilast, digitaliseeritud õpikut 13, mõne kooli kursuse materjale, näiteks ülesannetekogu, on kasutanud 8, veebikursuseid nagu Coursera ning Codecademy 6, paberkandjal õpikut 3 ning õppevideoid ning ülikooli kursuste materjale 2 õpilast (Joonis 3).



Joonis 4. Materjalide keel

Kõik küsimustikule vastajad kasutavad ingliskeelseid materjale, 19 õpilast kasutavad eestikeelseid ning 3 venekeelseid (Joonis 4). Kirjaliku tagasiside põhjal kasutavad enamus õpilasi ingliskeelseid materjale, kuna inglise keel on universaalsem ja sellega on internetist kergem probleemidele vastuseid leida. Lisaks toodi välja ingliskeelsete materjalide paremat kvaliteeti, algajasõbralikkust ning puudust eestikeelsetest terminitest. Mitmed õpilased mainisid, et eesti keeles on lihtsam aru saada ja õppida, kuid koodinäiteid ning lahendusi probleemidele tuleb siiski inglise keeles otsida. Üks õpilane, kes oli märkinud üheks keeleks ka vene keele, kirjutas, et koolis õpib ta programmeerimist eesti keeles, programmeerimisalaseid videoid vaatab inglise keeles ning vastuseid küsimustele ja muredele otsib vene keeles, sest ka vene keeles on palju infot ning tema jaoks on vajaliku info leidmine nii lihtsam.



Joonis 5. Huvi tekitavad programmeerimiskeeled

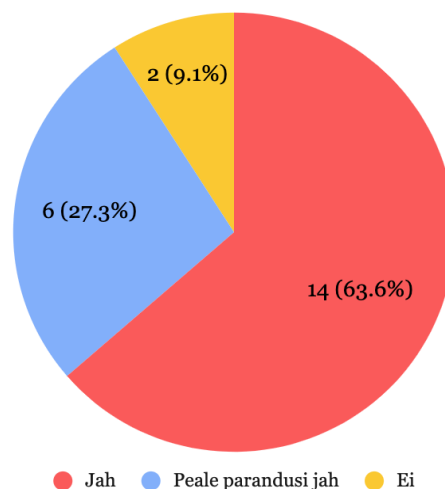
Lisaks praegustele oskustele paluti õpilastel valida, mis programmeerimiskeeli nad veel sooviksid õppida. Kõige populaarsemateks keelteks olid JavaScript, C++ ning Java, neid valiti vähemalt 15 korda, 10 korda valiti Python ja HTML, 9 korda C#, 6 korda CSS, 4 korda C, 2 korda Scratch ning Assembly ja 1 kord Lua ning COBOL (Joonis 5). Keelte valikuks oli peamised põhjused: tulevikus tuleb tööturul kasulikuks, ollakse kõige rohkem nende keelte kohta kuulnud, soov veebiarenduses paremaks saada ning programmeerimiskeel tundub huvitav.

Viimase küsimusena uuriti, kas õpilane on kunagi alustanud mõne keele õppimist ning millegipärast pooleli jätnud. 2 õpilast vastas, et olid enne gümnaasiumi proovinud Pythonit, kuid kadus motivatsiooni ja ilma sobivate allikateta jäi õppimine pooleli. Keeltest oli veel mainitud C#, HTML, CSS, JavaScript ning kahel korral ka Java, õppimine jäi pooleli ajapuuduse tõttu ning samuti osutus Java algajale liiga keeruliseks.

### 4.3. Tagasiside küsimustik

Dick ja Carey õppematerjalide loomise mudeli järgi on üheks protsessi etapiks materjalide testimine ja paranduste sisseviimine. Materjalide paremaks tegemise eesmärgil paluti õpilastel ausalt hinnata oma kasutajakogemust materjale läbides. Hinnangu saamiseks paluti õpilastel peale materjalide läbimist täita tagasiside küsimustik. Küsimustiku eesmärgiks oli saada arusaam materjalide kasutajasõbralikkusest ja teemade, raskusastme ning keskkondade sobivusest. Tagasiside küsimustikule vastas 22 õpilast. Lisaks eelnevalt mainitud teemadele anti õpilastele võimalus anda ka vabas vormis tagasisidet materjalide sisu kohta, mida käsitletakse peatüki lõpus.

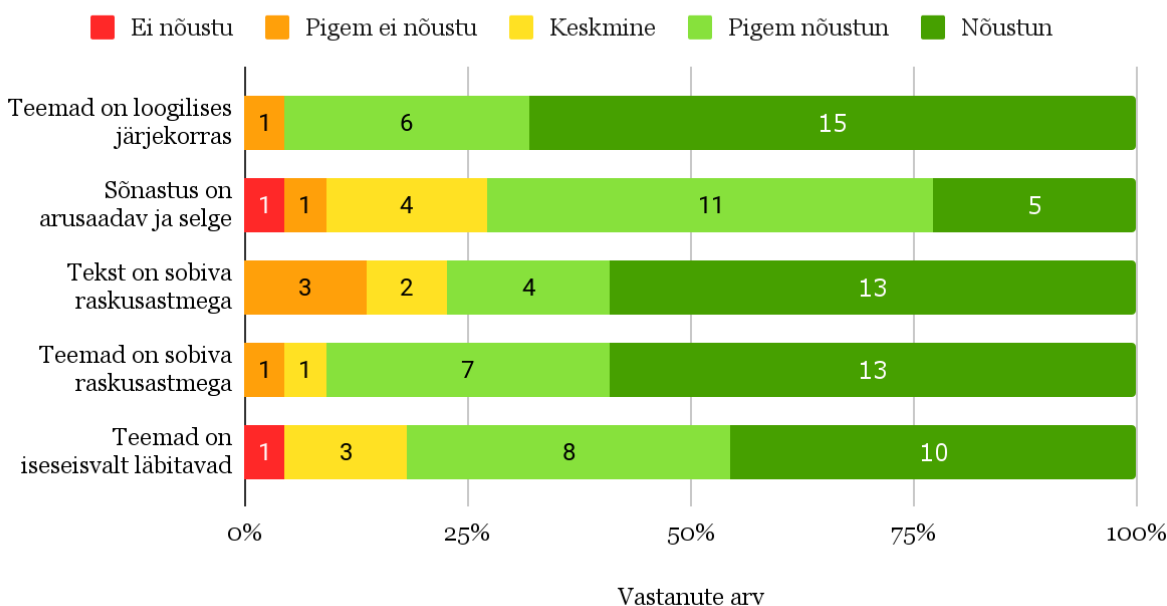
Kas tahaksid neid materjale edasi kasutada?



Joonis 6. Soov materjale edasi kasutada

Küsimusele “Kas tahaksid neid materjale edasi kasutada?” vastasid 14 õpilast ehk 64% vastanutest jaatavalt, 6 õpilast ehk 27% arvas, et peale paranduste sisseviimist jah ning 2 õpilast ehk 9% vastas, et ei sooviks neid materjale edasi kasutada (Joonis 6). Eitavalt vastanud õpilastest üks ei olnud huvitatud Java õppimisest ning teine tundis, et oskab Javat juba piisavalt hästi, et ei saaks materjalidest palju uusi teadmisi. Õpilane siiski mainis, et peale paranduste tegemist ja teemade lisamist oleks see hea materjal Eesti noortele.

## Õpilaste hinnangud materjalidele



Joonis 7. Õpilaste koondhinnangud

Et saada paremat ülevaadet õpilaste arvamustest, paluti neil viie väite kohta anda oma hinnang skaalal ühest viieni, kus üks oli ei nõustu ning viis tähistas nõustumist. Väitega “Teemad on loogilises järjekorras” nõustus 15 õpilast, 6 pigem nõustus ning 1 pigem ei nõustunud (Joonis 7). Väitega “Sõnastus on arusaadav ja selge” nõustus 5, pigem nõustus 11, 4 ei osanud vastata, 1 pigem ei nõustunud ning 1 õpilane ei nõustunud üldse (Joonis 7). Madalama hinnangu selgituseks toodi välja, et mõned terminid, näiteks “süntaks” võiksid olla paremini lahti seletatud ning ülesannete sisu võiks selgem olla. Väitega “Tekst on sobiva raskusastmega” nõustus 13, pigem nõustus 4, ei osanud vastata 2 ning pigem ei nõustunud 3 õpilast (Joonis 7). Väitega “Teemad on sobiva raskusastmega” nõustus 13, pigem nõustus 7, ei osanud vastata 1 ning pigem ei nõustunud 1 õpilane (Joonis 7). Väitega “Teemad on iseseisvalt läbitavad” nõustus 10, pigem nõustus 8, ei osanud vastata 3 ning ei nõustunud 1 õpilane (Joonis 7). Õpilaste vabas vormis hinnangud olid varieeruvad. Mitu õpilast tundsid, et materjalid olid sobilikud ning hindasid neid positiivselt: “Hästi seletatud ja noortepärane.”, “Materjalid olid arusaadavalt kirjutatud, et iga õpilane saaks aru ja teeks õppimise lihtsaks.”, “Kergesti arusaadavad”, “Ülesanded olid väga hea raskustasemega ja hästi seletatud”. Paar õpilast hindas, et materjalid vajaksid parandusi: “Võiks olla algaja jaoks arusaadavamalt tehtud.”, “Ülesannete püstitused võiks olla selgemalt kirjeldatud”.

Et saada tagasisidet ka keskkondade eripärade kohta, uuriti, mis meeldis ning mis ei meeldinud nii E-koolikoti kui GitHubi materjalide juures. E-koolikoti materjalide juures meeldis: “Materjalide lihtsus.”, “Organiseeritus.”, “Kõik ühes kohas.”, “Kergesti navigeeritav.”, “Mulle meeldis, et peatüki lõpus olid väikesed kokkuvõtted. Väga algajasõbralik ja noortepärane.”. E-koolikoti juures ei meeldinud: “Võiks rohkem pilte olla.”, “Mulle meeldiks, kui teooria juures oleks pisike koodijupp, et kohe aru saada, siis ei peaks suure koodi seest otsima”.

GitHubi juures meeldis: “Materjale oli palju ning erinevate raskusastmetega.”, “Näidislahendused.”, “Et oli olemas lahendustega kaust kui hätta jäin.”, “Huvitavad ülesanded”, “Meeldis, et materjalid olid lihtsasti leitavad ja loogilises järjekorras”. GitHubi juures ei meeldinud: “Raske ülesseadmine.”, “GitHubi ja E-koolikoti vahel liikumine ei ole väga mugav. Lisaks võiks muuta ülesannete kirjeldusi nii, et need poleks koodis sees, sest kogu koodi nägemine ja sellest aru saamine muutub raskeks. Ülesande kirjeldus võiks olla eraldi tekstina E-koolikotis.”, “E-koolikotis võiks olla selgitus selle kohta, kuidas ülesandeid teha, et mida kommentaarid tähendavad ja kuidas ülesanded on üles ehitatud.”.

Tunnis osalenud õpilaste hulgas oli ka üks õpilane, kes oli varem Javaga palju kokku puutunud ning ei tundnud, et loodud materjalid talle uusi teadmisi annaksid. Selle eest oli tema tagasiside materjalide kohta väga sisukas. Lisaks tagasisidele andis ta ka mõned ideed edasiarenduseks: iga peatüki juurde võiks lisada lühikese video, mis seletaks teemat lihtsamalt lahti; kursuse juures võiks olla mõistete ja terminite sõnastik või iga teemade juures oleksid mõistete kordamisülesanded; kuna pikad tekstid võivad olla koormavad, võiksid iga peatüki lõpus olla kokkuvõtted stiilis “Kui peatükist jätta kaks asja meelde, siis need oleksid... ”.

#### **4.4. Tulemused**

Õpilaste tagasiside üldiselt oli positiivne ja sisukas. E-koolikoti materjalides parandati mõned kirjavead, selgitati termineid rohkem lahti, täiendati GitHubi kasutusjuhendit, lisati enesekontrolliteste nii teemade kui terminite kohta ning peatükkide lõppu lisati lühikesed kokkuvõtvad laused kõige olulisematest teemadest. E-koolikotti lisati ka veebipõhise kompilaatori kasutusjuhend ja lehtedevahelise liikumise ning ülesannete lahendamise hõlbustamiseks lisati GitHubis olevad koodid ja ülesannete tekstid osaliselt ka E-koolikotti, juhendite lihtsamaks järgmiseks lisati paar lühikest selgitavat videot. GitHubis olevaid ülesandeid täiendati selgitustega, mida ülesandest oodatakse ja millised võiksid olla

lõppväljundid ning E-koolikotti lisati ülesannete ülesehituse ja lahendamise kohta selgitavad juhised.

Eeltesti tulemuste põhjal võib öelda, et õpilastel on päriselt huvi tulevikus Java õppimise vastu. Põhinedes programmeerimiseks kasutatavate materjalide küsimusele, on õpilased juba tuttavad veebiõpiku formaadi ja ideega ning on varem neid kasutanud õppematerjalidena. Õppekeele küsimusest tuli välja, et mitmed õpilased eelistavad just eesti keeles õppida, kuid väheste materjalide olemasolu tõttu peavad nad pöörduma just ingliskeelsete materjalide poole. Suur iseseisvalt õppijate arv näitab, et on soovi materjalide järele, mida oleks võimalik iseseisvalt ja ilma õpetaja suure osaluseta läbida. Materjalide tagasiside põhjal on loodud materjalid asjakohased, noortepärased ning piisava raskustasemega. Nende järelduste põhjal on eestikeelse Java programmeerimismaterjali loomine väga vajalik ning on olemas sihtrühm, kes neid materjale kasutama võiks hakata ning peale parandusi oleksid huvitatud ka loodud materjalidega õppimist jätkama.

Üheks materjalide loomise protsessi kitsaskohaks võib lugeda õpilaste hulka ning nende varasemat kokkupuudet programmeerimisega. Dick ja Carey mudelile toetudes oleks peale muudatuste sisseviimist järgmise sammuna katsetada materjale klassi peal, kes poleks varasemalt programmeerimisega kokku puutunud. Peale nende peal katsetamist ja muudatuste sisseviimist, oleksid materjalid mitmekülgsemad ning algaja jaoks lihtsamate selgitustega ja paremate kasutusjuhenditega.

Autori arvates sai töö eesmärk täidetud. Valmisid Java õppematerjalid, mis annavad algajale ülevaate Javast ning peale materjalide läbimist suudab õpilane iseseisvalt lihtsamaid programmeerimisülesandeid lahendada. Materjalid on üles ehitatud nii, et õpilane suudaks iseseisvalt neid läbida ning vajalikud koodijupid ja ideed ülesannete lahendamiseks on juba varem koodi sees mainitud.

Loodud materjalidel on palju tulevikuvõimalusi. GitHubi võiks lisada rohkem ülesandeid, et teha õppimist mitmekesisemaks. E-koolikotti võiks lisada teemasid nagu rekursioon, *regex* ning objektorienteeritud programmeerimise alused. Nende temade omandamine võimaldaks kursuse lõpus suurema projekti realiseerimist ning annaks õpilasele väga mitmekülgse teadmiste pagasi, mida saaks omakorda ka teistes programmeerimiskeeletes rakendada.

## Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua Java programmeerimiskeele õppematerjal gümnaasiumiastmele. Töö teiseks eesmärgiks oli loodud materjalide kohta tagasiside kogumine, analüüs ja seeläbi materjalide täiendamine. Töö raames valmis E-koolikoti materjal pealkirjaga “Java programmeerimiskeele õppematerjal algajale”, mis on avalikult kättesaadav E-koolikoti keskkonnas. Lisaks E-koolikoti materjalidele loodi ka GitHubi repositoorium, kust on võimalik alla laadida kursuse jaoks vajalikud alusfailid, koodi- ja struktuurinäited ning ülesannete failid, et õpilasele materjalide läbitöötamine võimalikult lihtsaks teha.

Materjalide loomisel kasutati Dick ja Carey mudelit [6], mis aitas materjalide loomise protsessile läheneda struktuurselt ning analüüsida saadud tulemusi. Valminud õppematerjalis antakse ülevaade Java programmeerimiskeelest ning programmeerimise põhiteadmistest nagu muutujad, andmetüübid, tingimuslaused ning tsüklid. Materjalides on kasutusel IntelliJ IDEA keskkond ning õpilaste jaoks, kellel pole võimalik arvutisse keskkonda laadida, on olemas ka juhend veebipõhise Java kompilaatori kasutamiseks.

Üheks osaks Dick ja Carey mudelis on materjalide katsetamine ning paranduste tegemine. Loodud materjale katsetati näidistunnis 10. klassi programmeerimissuuna õpilastega. Õpilased täitsid enne tundi eeltesti, mille eesmärk oli saada ülevaade nende eelnevatest teadmistest ning pärast tundi tagasiside küsitluse, et hinnata nende rahulolu materjalidega. Üldine tagasiside materjalidele oli positiivne, kuid toodi välja, et materjalidesse võiks mõningaid parandusi lisada. Tagasiside põhjal täiendati GitHubi kasutusjuhendit ja lisati selgitusi, enesekontrolliteste, teemasid selgitavad lühivideod ja veebipõhise kompilaatori juhend.

Tulevikus võiks materjalidesse lisada rohkem ülesandeid ning teemasid nagu rekursioon, regex ning tutvustada objektorienteeritud programmeerimise aluseid. Nende teemade käsitlemine võimaldaks kursuse lõpus mõne suurema projekti realiseerimist.

## Viidatud kirjandus

- [1] TIOBE, Tiobe index, <https://www.tiobe.com/tiobe-index/>, 14.05.2024
- [2] Wong G. K.-W., Cheung H.-Y. Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming, Interact. Learn. Environ., vol. 28, no. 4, pp. 438–450, 2020. doi: 10.1080/10494820.2018.1534245
- [3] JetBrains tudengisoodustus, <https://www.jetbrains.com/community/education/#students>, 14.05.2024
- [4] Põldoja, Õppematerjalide koostamise protsess ja kvaliteet, Tallinna Ülikooli haridustehnoloogia magistriõppe kursus, 2016  
<https://digioppevara.wordpress.com/lugemismaterjalid/oppematerjalide-koostamise-protsess-ja-kvaliteet/>, 14.05.2024
- [5] Edutech Wiki, Kemp'i disainimudel, [https://edutechwiki.unige.ch/en/Kemp\\_design\\_model](https://edutechwiki.unige.ch/en/Kemp_design_model), 14.05.2024
- [6] Edutech Wiki, Systematic Design of Instruction (Dick and Carey model), [https://edutechwiki.unige.ch/en/Systematic\\_Design\\_of\\_Instruction](https://edutechwiki.unige.ch/en/Systematic_Design_of_Instruction), 14.05.2024
- [7] Keps K., Veebirakendus programmeerimise õppimiseks II ja III kooliastmes, [https://comserv.cs.ut.ee/ati\\_thesis/datasheet.php?id=77341](https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=77341), 14.05.2024
- [8] Haridus- ja Teadusministeerium, Õppevara, <https://www.hm.ee/uldharidus-ja-noored/alus-pohi-ja-keskharidus/oppevara#oppekirjanduse-regis>, 14.05.2024
- [9] Tartu Ülikooli Arvutiteaduse instituut, Programmeerimise õpik, <https://progeopik.cs.ut.ee/>, 14.05.2024
- [10] Oracle, Java Applets, <https://www.oracle.com/java/technologies/applets.html#:~:text=An%20applet%20is%20a%20program,is%20included%20in%20a%20page>, 14.05.2024



## Lisad

### I. Materjalide sisukord

1. Sissejuhatus - Mis on Java?
  - 1.1. Kasutusvaldkonnad
  - 1.2. Alustamine - Allalaadimine, ülesseadmine
2. Esimesed koodiread
  - 2.1. Kuhu ja mida kirjutada - Main meetod
    - 2.1.1. Uue faili loomine
    - 2.1.2. "Tere maailm!"
    - 2.1.3. Uus fail + print
  - 2.2. Kommentaarid
  - 2.3. GitHubi kasutamine - aluskoodi allalaadimine
3. Andmetüübid
  - 3.1. Mis on muutujad ja kuidas neid lisada?
  - 3.2. Täisarvud ja ujukomaarvud
    - 3.2.1. Arvutamine: + - \* /
  - 3.3. Sõned ja *charid*
  - 3.4. *Boolean*
  - 3.5. Massiivid
  - 3.6. Kokkuvõte Andmetüüpidest
4. Avaldised + operaatorid - Tingimuslaused
  - 4.1. *If + else*
  - 4.2. *Or + and*
5. Tsüklid
  - 5.1. Mis on tsükkel?
  - 5.2. *While* tsükkel
  - 5.3. *For + for each*
6. Lõputeemad
  - 6.1. Projektide ideed
  - 6.2. Huvitavad meetodid ning näpunäited

## II. Litsents

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

Mina, Stina-Marie Maripuu,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose “Java õppematerjalide loomine gümnaasiumiastmele”, mille juhendaja on Kristi Salum, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Stina-Marie Maripuu*

**15.05.2024**