

TARTU ÜLIKOOL
LOODUS- JA TÄPPISTEADUSTE VALDKOND
Tehnoloogiainstituut
Arvutitehnika õppekava

Helen Õunapuu

**Ainekavade ja materjalide koostamine gümnaasiumi
valikainetele „Programmeerimine keeles Java I“ ja
„Programmeerimine keeles Java II“**

Bakalaureusetöö (12 EAP)

Juhendaja: Anne Villems, MSc
Merike Hein, MSc, ING.PAED.IGIP

Tartu 2019

Ainekavade ja materjalide koostamine gümnaasiumiastme valikainetele „Programmeerimine keeles Java I“ ja „Programmeerimine keeles Java II“

Lühikokkuvõte:

Bakalaureusetöö eesmärk on luua gümnaasiumiastmele sobivad ainekavad ning materjalid valikainetele „Programmeerimine keeles Java I“ ja „Programmeerimine keeles Java II“ ning viia loodud ainekavasid ning materjale kasutades läbi kaks valikkursust Tartu Jaan Poska gümnaasiumis. Valikkursused viidi bakalaureusetöö autori poolt läbi ajavahemikus 18. detsember 2018 kuni 12. aprill 2019.

Hindamaks loodud materjalide ning ainekavade sobivust gümnaasiumiastmele, analüüsiti Tartu Jaan Poska gümnaasiumi õpilaste õpitulemusi ja viidi õpilaste seas läbi küsitlus, et selgitada välja nende rahulolu „Programmeerimine keeles Java I“ ja „Programmeerimine keeles Java II“ valikainega ning koguda ettepanekuid ainete parendamiseks. Töö lõpus esitab töö autor soovitusi teiste sarnaste valikkursuste korraldamiseks loodud õppematerjale ning ainekavasid kasutades.

CRESC: S270 Pedagoogika ja didaktika

Võtmesõnad: Java, ainekava, programmeerimise õpetamine, õppematerjal

Abstract:

The purpose of the this bachelor's thesis is to create syllabi and study materials for secondary school elective courses Programming in Java I, Programming in Java II. Syllabus and study materials for first two courses were used in Tartu Jaan Poska Gymnasium during III and IV period, from 18th December 2018 to 12th April 2019. Both elective courses were taught by the author of this bachelor's thesis.

To validate the effectiveness of the materials and the syllabi created, the Tartu Jaan Poska Gymnasium students' course results were analysed and their satisfaction with the courses was surveyed. The author of the thesis also gives recommendations on using created materials.

CERCS: S270 Pedagogy and didactics

Keywords: Java, syllabus, teaching programming, study material

Sisukord

Jooniste loetelu	6
Tabelite loetelu	8
Lühendid, konstandid ja mõisted	9
Sissejuhatus	10
1. Programmeerimise õpetamine	12
1.1. Java taust	12
1.2. Programmeerimise õpetamine Eestis	13
1.3. Programmeerimise õpetamise meetodid	13
1.4. Tartu Jaan Poska gümnaasiumi valikainete süsteemi eripärad	14
2. “Programmeerimine keeles Java I” valikaine	16
2.1. “Programmeerimine keeles Java I” aine iseseisvad ülesanded	16
2.2. “Programmeerimine keeles Java I” valikaines kaetud teemad	17
2.2.1. Sissejuhatus “Programmeerimine keeles Java I” ainesse	17
2.2.2. Muutujad, loogika ja programmi sisend konsooli kaudu	18
2.2.3. Tsüklid ja otsingumootorite kasutamine informatsiooni leidmiseks	18
2.2.4. Andmestruktuurid I. Massiivid ja järjendid	19
2.2.5. Meetodid ja rekursioon	19
2.2.6. Failist lugemine ja faili kirjutamine	19
2.2.7. Java klassid	20
2.2.8. Andmestruktuurid II. HashSet ja HashMap	20
2.2.9. Graafiline kasutajaliides	21
2.2.10. Dokumentatsioon, klassid II ja graafiline kasutajaliides II	21
2.3. “Programmeerimine keeles Java I” aine kontrolltööd	22
2.3.1. Esimene kontrolltöö	22
2.3.2. Teine kontrolltöö	22

2.4. “Programmeerimine keeles Java I” aine projekt	23
3. “Programmeerimine keeles Java II” valikkursus	25
3.1. “Programmeerimine keeles Java II” valikained kaetud teemad	25
3.1.1. Git ehk versioonihaldussüsteem	26
3.1.2. “Programmeerimine keeles Java I” teemade kordamine	26
3.1.3. Lambdad ja Streamid	26
3.1.4. Erindid	27
3.1.5. Graafilise kasutajaliidese head tavad	27
3.1.6. Algoritmid	27
3.1.7. Tarkvara ettevõtted	28
3.1.8. Klassid ja rekursioon	28
3.2. Kontrolltöö	29
3.3. Projekt	29
4. Tartu Jaan Poska gümnaasiumi õpilaste õpitulemuste ja tagasiside analüüs	33
4.1. TJPG õpilaste õpitulemuste analüüs kursustel “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”	33
4.2. Tartu Jaan Poska gümnaasiumi õpilaste tagasiside analüüs kursustel “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”	36
4.3. TJPG õpilaste õpitulemuste ning tagasiside küsitluse analüüsi järeldused	39
Kokkuvõte	41
Tänuavaldused	43
Allikad	44
Lisa 1 “Programmeerimine keeles Java I” kursuse ainekava	46
Lisa 2 “Programmeerimine keeles Java I” aine slaidid	48
Lisa 3. “Programmeerimine keeles Java I” aine ülesanded	59
Lisa 4 “Programmeerimine keeles Java I” aine Moodle testid	67
Lisa 5. “Programmeerimine keeles Java I” aine kirjalik kontrolltöö	68
Lisa 6. “Programmeerimine keeles Java II” ainekava	71

Lisa 7 “Programmeerimine keeles Java II” aine slaidid	73
Lisa 8. “Programmeerimine keeles Java II” aine iseseisvad ülesanded	81
Lisa 9. “Programmeerimine keeles Java II” aine moodle testid	87
Lisa 10. Kursuste tagasiside küsitluste link	88
Lisa 11. Kursuste tagasiside küsitluste suured graafikud	89
Lisa 12. Õpilaste soovitused “Programmeerimine keeles Java I” ning “Programmeerimine keeles Java II” korraldamiseks	92
Litsents	94

Jooniste loetelu

Joonis 1. TJPG õpilaste perioodi hinnete jaotus aines “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”.	34
Joonis 2. TJPG õpilaste keskmised tulemused kogu klassi peale hindelise töö kaupa aines “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”.	35
Joonis 3. TJPG õpilaste vastused küsimusele “Miks soovisid valikaines osaleda?”.	36
Joonis 4. TJPG õpilaste vastused küsimustele ”Millised olid lemmikud Programmeerimine keeles Java I läbitud teemad?” ja “Millised teemad Programmeerimine keeles Java I kursus üldse ei meeldinud?”.	37
Joonis 5. TJPG õpilaste vastused küsimustele ”Millised olid lemmikud Programmeerimine keeles Java II läbitud teemad?” ja “Millised teemad Programmeerimine keeles Java II kursus üldse ei meeldinud?”	38
Joonis 6. Näidis slaid 1 “Programmeerimine keeles Java I” sissejuhatusest	48
Joonis 7. Näidis slaid 2 “Programmeerimine keeles Java I” sissejuhatusest	48
Joonis 8. Näidis slaid 1 “Programmeerimine keeles Java I” “Muutujad” teemast	49
Joonis 9. Näidis slaid 2 “Programmeerimine keeles Java I” “Loogika tehted” teemast	49
Joonis 10. Näidis slaid 1 “Programmeerimine keeles Java I” “Tsüklid ja Google” teemast	50
Joonis 11. Näidis slaid 2 “Programmeerimine keeles Java I” “Tsüklid ja Google” teemast	50
Joonis 12. Näidis slaid 1 “Programmeerimine keeles Java II” “Andmestruktuurid I” teemast	51
Joonis 13. Näidis slaid 2 “Programmeerimine keeles Java I” “Andmestruktuurid I” teemast	51
Joonis 14. Näidis slaid 1 “Programmeerimine keeles Java I” “Meetodid” teemast	52
Joonis 15. Näidis slaid 2 “Programmeerimine keeles Java I” “Rekursioon” teemast	52
Joonis 16. Näidis slaid 1 “Programmeerimine keeles Java I” faili operatsioonide slaididest	53
Joonis 17. Näidis slaid 2 “Programmeerimine keeles Java I” faili operatsioonide slaididest	53
Joonis 18. Näidis slaid 1 “Programmeerimine keeles Java I” “Klassid” teemast	54
Joonis 19. Näidis slaid 2 “Programmeerimine keeles Java I” “Klassid” teemast	54
Joonis 20. Näidis slaid 1 “Programmeerimine keeles Java I” “Andmestruktuurid II” teemast	55
Joonis 21. Näidis slaid 2 “Programmeerimine keeles Java I” “Andmestruktuurid II” teemast	55
Joonis 22 Näidis slaid 1 “Programmeerimine keeles Java II” “GUI” teemast	56

Joonis 23. Näidis slaid 2 “Programmeerimine keeles Java II” “GUI” teemast	56
Joonis 24. Näidis slaid 1 “Programmeerimine keeles Java II” “Dokumentatsioon” teemast	57
Joonis 25. Näidis slaid 1 “Programmeerimine keeles Java II” “Klassid II” teemast	57
Joonis 26. Näidis slaid 2 “Programmeerimine keeles Java II” “GUI II” teemast	58
Joonis 27. Näidis slaid 1 “Programmeerimine keeles Java II” “Sissejuhatus ja Git” teemast	73
Joonis 28. Näidis slaid 1 “Programmeerimine keeles Java II” kordamise teemast	74
Joonis 29. Näidis slaid 2 “Programmeerimine keeles Java II” kordamise teemast	74
Joonis 30. Näidis slaid 1 “Programmeerimine keeles Java II” “Streamid” teemast	75
Joonis 31. Näidis slaid 2 “Programmeerimine keeles Java II” “Lambdad” teemast	75
Joonis 32. Näidis slaid 1 “Programmeerimine keeles Java II” “Erindi” teemast	76
Joonis 33. Näidis slaid 2 “Programmeerimine keeles Java II” “Erind” teemast	76
Joonis 34. Näidis slaid 1 “Programmeerimine keeles Java II” “GUI head tavad” teemast	77
Joonis 35. Näidis slaid 2 “Programmeerimine keeles Java II” “GUI head tavad” teemast	77
Joonis 36. Näidis slaid 1 “Programmeerimine keeles Java II” “Algoritmid” teemast.	78
Joonis 37. Näidis slaid 2 “Programmeerimine keeles Java II” “Algoritmid” teemast	78
Joonis 38 Näidis slaid 1 “Programmeerimine keeles Java II” tarkvara ettevõtete teemast	79
Joonis 39. Näidis slaid 2 “Programmeerimine keeles Java II” tarkvara ettevõtete teemast	79
Joonis 40. Näidis slaid 1 “Programmeerimine keeles Java II” “Klassid III” teemast	80
Joonis 41. Näidis slaid 2 “Programmeerimine keeles Java II” “Klassid III” teemast	80
Joonis 42. TJPG õpilaste hinnangud “Programmeerimine keeles Java I” kursusega rahulolu kohta esitatud väidete põhjal	89
Joonis 43. TJPG õpilaste hinnangud “Programmeerimine keeles Java II” kursusega rahulolu kohta esitatud väidete põhjal	90
Joonis 44. TJPG õpilaste vastused küsimusele “Mis üldse ei meeldinud kursuse korralduse juures?” ja “Mis eriti meeldis kursuse korralduse juures”	91

Tabelite loetelu

Tabel 1. “Programmeerimine keeles Java I” projekti hindamisjuhend	32
Tabel 2. “Programmeerimine keeles Java I” iseseisvad ülesanded 1	59
Tabel 3. “Programmeerimine keeles Java I” iseseisvad ülesanded 2	59
Tabel 4. “Programmeerimine keeles Java I” iseseisvad ülesanded 3	60
Tabel 5. “Programmeerimine keeles Java I” iseseisvad ülesanded 4	61
Tabel 6. “Programmeerimine keeles Java I” iseseisvad ülesanded 5	61
Tabel 7. “Programmeerimine keeles Java I” iseseisvad ülesanded 6	62
Tabel 8. “Programmeerimine keeles Java I” iseseisvad ülesanded 7	63
Tabel 9. “Programmeerimine keeles Java I” iseseisvad ülesanded 8	64
Tabel 10. “Programmeerimine keeles Java I” iseseisvad ülesanded 9	65
Tabel 11. “Programmeerimine keeles Java I” teine kontrolltöö	65
Tabel 12. “Programmeerimine keeles Java II” aine iseseisvad ülesanded	81

Lühendid

IT – Infotehnoloogia

TJPG – Tartu Jaan Poska gümnaasium

GUI – *Graphical User Interface* ehk graafiline kasutajaliides

Sissejuhatus

Infotehnoloogia erialad on sisseastujate seas tihti esimeseks valikuks ning nendele on kõigis Eesti suuremates ülikoolides tihe konkurss. Peale sisseastumist avastavad paljud üliõpilased, et infotehnoloogia eriala pole neile sobiv. Seetõttu on IT erialadel palju õpinguid katkestavaid üliõpilasi. Mitmed Eesti üldhariduskoolid on võtnud programmeerimisained oma õppekavasse, et aidata gümnaasiumi lõpetajatel teha teadlik eriala valik. Samas ei ole programmeerimise õpetamine Eestis riiklikult reguleeritud ja enamikele programmeerimiskeeltele, sealhulgas Javale, puuduvad head eestikeelsed gümnaasiumiastme materjalid.

Gümnaasiumiastmes programmeerimist õpetades peaks põhiohk olema noores huvi tekitamisel ja silmaringi avardamisel. Valikainet läbides peaks õpilane ilma teema sügavaid detaile teadmata saama aru, kas õpetatav teema võiks teda ka tulevikus kõnetada. Selline lähenemine gümnaasiumi programmeerimise valikainete õpetamisele lihtsustab õpilasel ülikooli eriala valiku tegemist ja võib aidata vähendada väljalangevust ülikooli esimesel kursusel informaatika ning arvutitehnika erialadel, kuna õpilased on ülikooli astudes oma eriala valikutes teadlikumad.

Bakalaureusetöö eesmärk on luua gümnaasiumiastmes kasutatavad programmeerimiskeele Java õpetamise ainekavad ja materjalid, mille põhjal läbi viidud kursusel osalemise tulemusena on õpilane paremini teadlik IT erialade väljakutsetest ja võimalustest ning ta oskab koostada lihtsamal tasemel Java programme. Bakalaureusetöö autor viib loodud materjale ja ainekavasid kasutades läbi kaks Tartu Jaan Poska gümnaasiumi valikkursust. Kursused läbinud õpilaste õpitulemusi ning tagasisidet analüüsitakse, et teha järeldusi loodud materjalide ja ainekavade sobivusest gümnaasiumiastme valikaineteks.

Õppematerjalide loomisel lähtutakse lihtsusest, loogilisusest ja intuiitsusest – materjalid peaksid õpilasele olema omandatavad ka kontaktundides kohal käimata. Ainekavade loomisel lähtutakse TJPG õppekorralduse eripäradest, silmaringi laiendamise ning huvi tekitamise eesmärgist.

Bakalaureusetöö on jagatud kuude olulisemasse ossa. Esimene osa on “Programmeerimise õpetamine”, kus annab lõputöö autor ülevaate IT erialade, tutvustab programmeerimiskeele Java tausta, kirjeldab programmeerimise õpetamist Eestis, annab ülevaate käesoleva bakalaureusetöö

raames loodud ja läbi viidud kursustel kasutatud õppemetoodikatest ning selgitab Tartu Jaan Poska gümnaasiumi õppekorralduse eripärasid. Bakalaureusetöö teises ja kolmandas osas kirjeldab autor vastavalt “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II” valikkursuste ülesehitust, teemasid ning materjale. Bakalaureusetöö neljandas osas on läbiviidud kursuste õpitulemuste ning kursuste tagasiside analüüs ja selle järeldused. Töö mahukaim osa on töö lisad, kus on kirjas või viidatud kõik aine jaoks loodud ainekavad ning materjalid. Töö lisad on esitatud eraldi failidena. Peamiste allikatena kasutatakse käesolevas bakalaureusetöös 2015. aastal avaldatud uurimust “Mis saab Eesti IT haridusest”, Siim Puniste 2015. aastal avaldatud bakalaureusetööd “Eesti gümnaasiumides õpetatavad programmeerimiskursused“, 2018. aastal avaldatud Neil C. C. Browni ja Greg Wilsoni tööd “*Ten quick tips for teaching programming*“ ning Tartu Jaan Poska gümnaasiumi hindamisjuhendit.

1. Programmeerimise õpetamine

Uuringud ennustavad, et 2020. aastaks on Euroopa Liidus puudu 481 000–1 685 000 IT valdkonna töötajat [1]. 2015. aastal avalikustatud uuringus “Mis saab Eesti IT haridusest?” leiti, et kõrgkoolide IT õppekavadele on suur konkurss – keskmiselt 3,11 avaldust õppekoha kohta. 2014. aastal võeti Tartu Ülikooli, Tallinna Tehnikaülikooli ning IT Kolledžisse IT erialadele kokku vastu 793 üliõpilast. Samuti avastati “Mis saab Eesti IT haridusest?” uuringus, et 2013. aastal katkestas 32% Tartu Ülikooli, Tallinna Tehnikaülikooli ning IT Kolledžisse IT erialadele sisseastunuist esimese õppeaasta jooksul oma õpingud. Väljalangemine oli olnud sarnasel tasemel juba mitu aastat. Peaaegu kolmandik (31%) õpingu katkestanutest tõid põhjusena välja IT eriala mittedobivuse ning asjaolu, et IT eriala ei olnud nende esimene valik. Selgus, et 17% küsitluses osalenute arvates ei vastanud ülikoolis õppimine nende ootustele. “Mis saab Eesti IT haridusest?” uuringu “IT-õpe kõrgkoolides” alamosa kokkuvõttena väidetakse, et üldhariduskoolides, sealhulgas gümnaasiumides, tuleks luua rohkem võimalusi IT valdkonna teemadega tegelemiseks. See suurendab noorte teadlikkust IT valdkonnast ning avardab nende silmaringi. Selle tulemusena saavad gümnaasiumilõpetajad teha ülikooli eriala valikul teadlikumaid otsuseid kui ilma varasema IT valdkonna kokkupuuteta noored. [2]

1.1. Java taust

Programmeerimiskeel Java on maailmas laialt levinud. Tarkvara kvaliteeti uuriv firma TIOBE on välja töötanud programmeerimiskeelte populaarsuse hindamise skaala, mille järgi on Java juba kolm aastat olnud populaarseim programmeerimiskeel maailmas [3]. Javat õpib maailmas 5 miljonit õpilast, kasutab tarkvaraarenduses 10 miljonit arendajat ning Java on ühilduv 10 miljardi seadmega [4]. Kuigi esimene stabiilne Java versioon lasti välja juba 1995. aastal, on Java jätkuvalt aktiivselt arenev programmeerimiskeel – Java kõige uuem versioon, Java 12 tuli välja 2019. aasta märtsis. Java arendamisega tegeleb ettevõtte Oracle. [5]

Java on laialt levinud ka Eestis – edukate tarkvara ettevõtete nagu Playtech [6], Nortal [7] ning Helves [8] kodulehelt võib leida mitmeid töökuulutusi Javat valdavate tarkvara arendajate ja inseneride positsioonidele. Need firmad korraldavad igal aastal praktikaprogramme ja võtavad

vastu praktikante Java arendaja positsioonidele. Lisaks toimub Eestis igal aastal GeekOut, mis on üks Euroopa suurimaid Java konverentse [9].

1.2. Programmeerimise õpetamine Eestis

Programmeerimise õpetamine üldhariduskoolides on Eestis järjest tavaprasem ning juba on koostatud soovitusel infotehnoloogia, sealhulgas programmeerimise, teemade lisamiseks 1. kuni 9. klasside informaatika õppekavasse [10]. 2015. aastal viis S. Puniste läbi uuringu programmeerimise õpetamise kohta Eesti koolides. Eesti gümnaasiumiastmes ei ole programmeerimise õpetamine riiklikult reguleeritud – kasutusel on palju erinevaid materjale. Kasutatakse välismaised õppematerjalide pakkujaid, nagu codeacademy.com ja php.net, ning samuti loovad õpetajad ise sobivaid õppematerjale. Parima programmeerimiskeelte eestikeelsete õppematerjalide allikana on õpetajad toonud välja Progetiigri keskkonna, kust leiab materjale nii Pythoni, Scratchi, App Inventori kui ka JavaScripti jaoks. [11] Java jaoks Progetiigri keskkonnas materjale ei ole [12].

Puniste sõnul keskenduvad enamik Eesti koole programmeerimist õpetades muutujatele, tsüklitele, valiklausetetele ja alamprogrammidele. Võrdlemisi vähe pööratakse tähelepanu IT valdkonna tutvustamisele ja programmeerimise ajaloole. Programmeerimiskeeltest on Eesti gümnaasiumides levinumad Scratch, Python, Java ja JavaScript, kusjuures neist keeltest on Scratch ainus visuaalne programmeerimiskeel. Seitsmeteistkümnest kursusest, mille õpetajad küsitlusele vastasid, käsitletakse Javat viies. [11] Java on üks lubatud programmeerimiskeeltest Eesti informaatikaolümpiaadil [13].

1.3. Programmeerimise õpetamise meetodid

Programmeerimise õpetamiseks on välja töötatud palju erinevaid meetodikaid. Brown ja Wilson on esitanud kümme võtet, mida jälgida programmeerimist õpetades. Nad soovivad lasta klassikaaslastel teineteist õpetada, kuna üks ühele õpetamine on tõenäoliselt kõige efektiivsem õpetamise viis. Kahjuks jääb õpetajal selleks vähe aega, kui klassis on korraga palju õpilasi. Õpilaste kaasamine õpetamisse on hea võimalus rakendada üks ühele õpetamist ja lisaks aitab teiste õpetamine kinnistada juba omandatud teadmisi. Teine meetod, mida Brown ja Wilson mainivad, on õpilastele koodi kirjutamise ettenäitamine. [14] Bakalaureusetöö autori arvates

sobivad selleks suurepäraselt enamikes koolides olemasolevad videoprojektorid – õpetaja saab oma arvutis programmeerida ning tööprotsessi vahetult õpilastele näidata. Brown ja Wilson soovivad programmeerimist õpetades piirduda esialgu ühe programmeerimiskeelega ning rakendada paarisprogrammeerimist, mida on kõige lihtsam rakendada projekti töid tehes, kus ühes meeskonnas on kaks kuni kolm õpilast [14].

D. Clark jt väidavad, et Java õppimise alguses on oluline hakata õpilastele selgitama objektorienteerituse mõistet. D. Clarki jt meelest parim viis klasse ning isendeid selgitada läbi reaaleluliste olukordade, näiteks *“Fido is an instance of the class Dog”* ehk bakalaureusetöö autori poolt tõlgitult “Fido on klassi Koer isend”. Selline lähenemine eeldab õpilastelt üldistamisoskust, mille tõttu võivad klassid ja instantsid õpilastele endiselt keerulised tunduda. D. Clarki jt sõnul on see siiski parem lahendus, kui klasside ja instantside selgitamine mõistete kaudu, kuna objektorienteerituse mõistetest arusaamine on alustajale Java õppijale veelgi keerulisem. [15]

1.4. Tartu Jaan Poska gümnaasiumi valikainete süsteemi eripärad

TJPGs saavad õpilased igal aastal valida ligikaudu 100 erineva valikkursuse hulgast. Gümnaasiumiastme peale kokku peab iga õpilane valima endale 18 kursust. Kool on avatud ka uutele ideedele – kui õpilasel on hea valikkursuse idee ja vähemalt 15 koolikaaslast, kellel on samuti huvi uut ainet läbida, leiab kool ainele õpetaja. Samuti arvestab kool õpilase kooliväliseid ringe ja treeninguid valikkursustena. [16]

TJPGs on perioodõpe. See tähendab, et õppeaasta on jagatud viide perioodi, kusjuures igas perioodis on kuus nädalat õppetööd ja üks nädal arvestustööde sooritamiseks. Igal perioodil peale esimese võib õpilane läbida ühe suuna- ehk valikkursuse ja ühe vabakursuse. Valikkursuste tunnid toimuvad alati teisipäeviti, kolmapäeviti ja neljapäeviti esimese ja viienda tunni ajal. Õppetunni pikkus on 75 minutit. Esimene tund toimub kell 8.20–9.35 ja viies tund toimub kell 14.30–15.45. [16]

Samuti puuduvad TJPGs kohustuslikud suunakursused – iga õpilane saab ise valida, milliseid õppesuunda kujundavaid aineid ta läbib. See tähendab seda, et samas suunas õppivad õpilased võivad läbida väga erinevas mahus aineid. Näiteks võib üks õpilane läbida koos kohustuslike

füüsika kursustega kokku kaheksa erinevat füüsikakursust (sh sellised valikained nagu „Kõõgifüüsika“, „Liiklusfüüsika“, „Astrofüüsika“), samas kui teine õpilane läbib vaid viis riiklikult kohustusliku füüsikakursust. Samuti võivad vabade kohtade olemasolul suunakursuseid läbida ka teiste õppesuundade õpilased. Sellisel juhul arvestatakse neid kursuseid nende õpilaste jaoks vabaainetena. Selline valikainete süsteem tagab õpilastele võimaluse proovida võimalikult palju erinevaid ja neid huvitavaid aineid ning otsida endale meelepärased valdkondi. Õpilastel ei ole kohustust õppida neid suunaaineid, mis õpilasele meelepärased pole. Seetõttu tuleb TJPGs õpetades arvestada asjaoluga, et õpilased on aine valinud vabatahtlikult ja ei läbi seda kooli õppekava tõttu. [16]

TJPGs kasutatakse eristavat hindamisskaalat, millel on kuus positiivset ning üks negatiivne hinne. Positiivsed hinded on A+, A, B, C, D ja E ning negatiivne tulemus on F. Kooli lõpetades arvutatakse hinded ümber viiepallisüsteemi. Viiepallisüsteemis on hinded A+ ning A vastavuses hindegaga 5, B ning C vastavuses hindegaga 4, hinded D ja E vastavuses hindegaga 3 ning hinne F vastavuses hindegaga 2. [17]

2. “Programmeerimine keeles Java I” valikaine

“Programmeerimine keeles Java I” on gümnaasiumiastmele suunatud valikkursus. Lisaks tehnilisele programmeerimisoskusele pööratakse tähelepanu koodi korrektsusele ning loetavusele. Õpilastele selgitatakse koodi kommenteerimise vajalikkust. Aine esimeses pooles kasutavad õpilased konsoolipõhist väljundit. Aine teises pooles tutvustatakse õpilastele graafilise kasutajaliidese loomist. Lisaks programmeerimise õpetamisele on “Programmeerimine keeles Java I” aine eesmärk laiendada õpilaste silmaringi ning seeläbi toetada neid eriala valikute tegemisel. Selleks, et tagada ühtlasema programmeerimistasemega õpilaste grupp, oli TJPGs “Programmeerimine keeles Java I” ainele registreerimise eelduseks TJPG kursuse “Programmeerimine keeles Python I” kursuse läbimine. “Programmeerimine keeles Java I” kursuse ainekava on leitav lisas 1 ning Moodle testid on esitatud lisas 4.

Materjalid ning ainekava on loodud selliselt, et nende põhjal oleks võimalik Javat õpetada ka esimese programmeerimiskeelena. Aine hindamist saab kujundada kooli ning õpetaja vajadustele vastavaks ning hindamissüsteemis kasutada kas eristavat või mitteeristavat hindamist. Töö autori välja töötatud hindamisjuhend “Programmeerimine keeles Java I” kursuse läbiviimiseks TJPGs on toodud lisas 1. TJPG õpilastele ei olnud kontakttunnid kohustuslikud, kuid vastava tunni iseseisvad ülesanded pidid järgmise tunni alguseks tehtud olema. Selline korraldus jättis õpilastele võimaluse ise planeerida oma õppimise aega.

2.1. “Programmeerimine keeles Java I” aine iseseisvad ülesanded

Iseseisvad ülesanded on jagatud plokkidesse ehk gruppidesse teemade järgi. Igas grupis on kerge, keskmise ning raske taseme ülesanded. Kerged ülesanded on mõeldud uue teema sissejuhatuseks, et õpilased saaksid uute võimaluste ja süntaksi eripäradega tutvuda. Keskmise taseme ülesanded vastavad tasemele, mille iga õpilane võiks aine lõpuks saavutada. Raskema taseme ülesanded on mõeldud õpilastele, kelle jaoks kergemad ja keskmised ülesanded liiga lihtsaks jäävad. Samuti annavad raskema taseme ülesanded õpilastele võimaluse rohkem punkte teenida. Enamik iseseisvaid ülesandeid on loodud selliselt, et neid saab kasutada

automaatkontrolliga esitamisvormides ehk süsteemides, mis kontrollivad automaatselt üles laetud programmi mistahes väljundi vastavust ettenähtule fikseeritud sisendi korral.

Igale ülesandele vastab ülesande identifikaator. Ülesande identifikaatori formaat on järgmine: keerukuse tähis, millele järgneb ülesande number. Kergete ülesannete keerukuse tähis on “a”, keskmistel ülesannete “b” ning rasketel ülesannete “c”. Näiteks ülesande kood a7 tähendab, et tegu on järjekorras seitsmenda kerge taseme ülesandega.

Tartu Jaan Poska gümnaasiumi “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II” valikkursustel täidavad iseseisvad ülesanded nii tunni ülesannete kui ka kodutööde eesmärgi – kui õpilane on tunnis kohal ning töötab aktiivselt kaasa, on tal enamasti võimalik kohustuslik hulk ülesandeid ära lahendada kohapeal ning kodust tööd ei jää. Kui õpilane otsustab, et ta ei soovi tundides kohal käia, siis peab ta kohustusliku hulga ülesandeid ära lahendama iseseisvalt kodus.

Iseseisvate ülesannete eesmärk on kinnistada tunnis õpitut ning arendada infopädevust lisainformatsiooni otsimist vajavate ülesannete abil. Samuti saavad õpilased iseseisvaid ülesandeid lahendades koguda ideid aine lõpus esitatava projekti teema jaoks.

2.2. “Programmeerimine keeles Java I” valikaines kaetud teemad

“Programmeerimine keeles Java I” aines on kokku kaksteist teemat, millest esimese üheksa kohta tuleb õpilastel lahendada iseseisvaid ülesandeid. Valikkursuse teemad on valitud selliselt, et kursuse lõpuks saaksid õpilased ise hakkama lihtsama reaalelus kasutava Java programmi loomisega. Iseseisvate ülesannete eesmärk on toetada materjali omandamist. Kõik ainega kokkukäivad slaidid on esitatud lisa 2.

2.2.1. Sissejuhatus “Programmeerimine keeles Java I” ainesse

Esimene teema “Programmeerimine keeles Java I” valikaines on sissejuhatus. Sissejuhatus tunnis tutvustab õpetaja õpilastele aine korraldust (sh hindamist) ning kasutatavat tarkvara. Soovituslik on kasutada IntelliJ IDEA keskkonda. Seejärel tutvutakse Java struktuuri ning süntaksi ja kirjeldatakse lihtsamaid andmetüüpe *String*, *int* ja *double* ning programmi väljundit.

Iseseisvate ülesannete käigus õpitakse tekstilise väljundi printimist konsoolis ja eritähenduslike sümbolite, näiteks jutumärgid ning länkriips, kujutamist. Esimesse ploki kuuluvad ülesanded a1, a2, b1, b2, c1 ning c2, mis on leitavad lisa 3 tabelist 2. Kõik ülesanded keskenduvad otseselt tunni teemade kinnistamisele. Ülesanne c2 on lisaks esimene ülesanne, mille lahendamise jaoks on õpilasel tarvis iseseisvalt leida lisamaterjali internetist.

2.2.2. Muutujad, loogika ja programmi sisend konsooli kaudu

Teine teema, mida “Programmeerimine keeles Java I” kõlmab, on muutujad, loogika ja sisend. Tutvutatakse uut andmetüüpi *boolean*, tingimuslauseid, loogikaavaldiste koostamist ning kasutajalt sisendi küsimist. Tingimuslausete ja loogikaavaldiste loomine on esimene teema, mis aitab õpilastel arendada süsteemset probleemide lahendamise oskust.

Teemale vastavaid iseseisvaid ülesandeid on üheksa: a3, a4, a5, b3, b4, b5, c3, c4, c5. Ülesannete kirjeldused on leitavad lisa 3 tabelist 3. Teise ploki iseseisvad ülesanded keskenduvad kasutajalt sisendi küsimisele ning kergemate tingimuslausete koostamisele. Tsüklite kasutamine ülesannete lahendamisel ei ole veel nõutud. Ka selle ploki raskemate ülesannete lahendamine vajab internetist informatsiooni otsimist.

2.2.3. Tsüklid ja otsingumootorite kasutamine informatsiooni leidmiseks

Kolmandas teemas õpetatakse õpilastele kolme erineva tsükli tüübi – *while*, *for* ja *for each* – loomist ning kasutamist. Oluline on, et õpilased saaksid aru, mille poolest need tsükli tüübid üksteisest erinevad. See aitab hiljem kirjutada paremini loetavat ning jätkusuutliku koodi. Selle teema all selgitatakse internetist informatsiooni otsimise põhitõdesid ning ohtusid.

Kolmandasse gruppi kuuluvad iseseisvad ülesanded a6, a7, b6, b7, c6, c7, mis on leitavad lisa 3 tabelist 4. Selle grupi iseseisvad ülesanded on mõeldud tsüklite õppimiseks. Õpilased saavad ise valida, kas eelistavad *while*- või *for*-tsükleid, ülesande tekstid seda ette ei määra. Samas tuleks iseseisvaid ülesandeid üle vaadates anda õpilastele soovitusi kui nad kasutasid tsükleid ebaefektiivselt – näiteks löid lõpmatu tsükli *for* tsükli tüübi abil. Selline lahendus on võimalik, kuid sellest võib võõral koodilugejal olla raske aru saada. Vastava tunni teema sisaldab ka

internetist informatsiooni otsimise põhitõdede ülevaadet, seega on iga raskusastme ühe ülesande lahendamiseks õpilastel tarvis otsida informatsiooni internetist.

2.2.4. Andmestruktuurid I. Massiivid ja järjendid

Järgmise teemana õpivad õpilased tundma lihtsamaid andmestruktuure massiiv (*array*) ja järjend (*List*). Oluline on õpilastele selgitada nende kahe andmestruktuuri erinevusi ning sarnasusi.

Sellesse gruppi kuuluvad iseseisvad ülesanded a8, a9, b8, b9, c8, c9. Need ülesanded on kirjas lisa 3 tabelis 5. Neljanda grupi iseseisvad ülesanded keskenduvadki massiividele (*array*) ning järjenditele (*List*). Õpilased õpivad massiive ning järjendeid looma ning modifitseerima. Iseseisvad ülesanded aitavad õpilastel mõista järjendite ning massiivide erinevaid kasutusjuhte.

2.2.5. Meetodid ja rekursioon

Viies teema, mida käesolevas aines käsitletakse, on meetodid ja rekursioon. Meetodite kasutamine muudab Java koodi kergemini loetavaks ning muudetavaks. Samuti saab tänu meetodite kasutamisele vältida koodi kopeerimist. Rekursiooni õppimine on õpilastele “Programmeerimine keeles Java I” aines vabatahtlik ning selle kohta on vaid üks raskema taseme ülesanne. Rekursiooni õpetamise eesmärgiks on pakkuda väljakutset õpilastele, kellele tavaliste meetodite õppimine liiga lihtne tundub. Viienda kursuse teemaga käivad kokku iseseisvad ülesanded a10, a11, b10, b11, c10, c11 (vt lisa 3 tabel 6).

2.2.6. Failist lugemine ja faili kirjutamine

Kuues teema “Programmeerimine keeles Java I” aines on failist lugemine ning faili kirjutamine. Lihtsuse mõttes kasutatakse aines ainult tekstifaile (faililaiendiga .txt), kuid saadud teadmised on rakendatavad ka teistele failitüüpidele. Failist lugemise ja faili kirjutamise teema juures on oluline rääkida õpilastele suhtelisest faili asukohast ning selle erinevustest absoluutse faili asukohaga. Absoluutne faili asukoht märgib teed failini seadme juurkaustast ning pole tingimata kättesaadav, kui jooksutada programmi mõnes teises seadmes. Selle aine raames peaksid õpilased *FileNotFoundException* erindi deklareerima, mitte seda käsitlema, kuna “Programmeerimine keeles Java I” aines ei õpita süvitsi erindeid.

Kuuenda grupi iseseisvad ülesanded on a12, a13, b12, b13, c12, c13, mis on leitavad lisa 3 tabelist 7. Ülesanded keskenduvad failidesse kirjutamise ning failist lugemise kasutamisele ning eripäradele. Osades ülesannetes tuleb õpilastel lahenduse leidmiseks otsida abi internetist.

2.2.7. Java klassid

Seitsmes teema “Programmeerimine keeles Java I” aines on Java klassid. “Klassid” on teema, kus kirjeldatakse õpilastele täpsemalt, mis on objektid ning mida tähendab väljend “Java on objektorienteeritud programmeerimiskeel”. Oluline on tuua objektide mõiste õpilastele võimalikult eluliselt lähedale. Näiteks saab Java objekte selgitada läbi kodumasinade. Igal ühel on kodus kodumasin, kus hoitakse kergesti halvaks minevat toitu. Seda nimetatakse külmkapiks. Igal külmkapil on mingid omadused – näiteks värvus, kaal, kõrgus. Iga (töötav) külmkapp jahutab oma sisemust. Java klassidel on ka teatavad omadused. Neid omadusi hoitakse muutujates, mida nimetatakse isendiväljadeks. Kui teha Java klass Külmkapp, siis sellel võiks olla kolm isendi välja: *String* värvus, *double* kaal ja *double* kõrgus ning *void* tüüpi meetodid *avaUks()* ning *sulgeUks()*, mis prindivad konsooli vastavalt tekstid “Külmkapi tuli läks põlema” ning “Külmkapi tuli kustus”. Eelnevalt kirjeldatud ning täiendatud näidisfaili juurde juhatav link on esitatud lisa 3.

Sarnased elulised võrdlused võiksid aidata õpilastel klasside olemusest paremini aru saada. Õpilastele tuleb selgitada, mis on konstruktor ning kuidas seda kasutada. Selles teemas ei ole tarvis veel teada *set* ja *get* meetodeid, klasside hierarhiat ega keerulisemat tüüpi klasse (nt abstraktseid klasse).

Klasside teemal saavad õpilased lahendada iseseisvaid ülesandeid a14, a15, b14, b15, c14, c15. Need ülesanded on leitavad lisa 3 tabelist 8. Klasse käsitlevate ülesannete eesmärk on survestada õpilasi looma objektorienteeritud koodi, kasutades klasse nii iseseisvates ülesannetes kui hiljem enda projektis.

2.2.8. Andmestruktuurid II. HashSet ja HashMap

Kaheksas teema keskendub andmestruktuuridele *HashMap* ehk sõnastik ja *HashSet* ehk hulk. Nende andmestruktuuride õpetamise eesmärk on näidata õpilastele, et andmestruktuuride valik ei

piirdu vaid massiivi ja järjendiga ning uutes olukordades võib olla kasulik leida uut tüüpi andmestruktuur. “Andmestruktuurid II” teemaga käivad kokku iseseisvad ülesanded a16, a17, b16, b17, c16, c17, mis on leitavad lisa 3 tabelist 9. Ülesanded on loodud selliselt, et õpilased saaksid aru erinevate andmestruktuuride kasutusjuhtudest ning erinevustest.

2.2.9. Graafiline kasutajaliides

Üheksas ehk viimane teema, kus õpilased lahendavad iseseisvaid ülesandeid, on “Graafiline kasutajaliides”. Graafilise kasutajaliidese abil saavad õpilased luua reaalselt kasutatavaid programme ning selle kasutamine on üks projekti esitamise nõuetest. Graafilise kasutajaliidese põhitõdede kohta on lõputöö autor loonud lühikese helita õppevideo, milles kasutatakse IntelliJ IDEA arenduskeskkonda. Video on kättesaadav Google Photos keskkonnast: <https://photos.app.goo.gl/HBHNDTwndL3q9Kih7>.

Graafilise kasutajaliidese teemale vastavas iseseisvate ülesannete ploki on kokku kolm ülesannet – üks igale raskusastmele. Vastavad ülesanded on a18, b18, c18, mis on leitavad lisa 3 tabelist 10. Ülesannete abil saavad õpilased ise proovida graafilise kasutajaliidese loomist enne projektiga tegelema hakkamist.

2.2.10. Dokumentatsioon, klassid II ja graafiline kasutajaliides II

Dokumentatsioon, klassid II ja graafiline kasutajaliides II on lisateemad, mis toetavad õpilastel projektiga tegelemist. Dokumenteerimise õpetamine aitab õpilastel kujundada harjumust kirjutada loetavat ning arusaadavat koodi. Koodi kommenteerimine aitab sellest aru saada nii võõral lugejal kui ka õpilasel endal, eriti kui koodi juurde tulla tagasi mitme aasta pärast. Dokumentatsiooni teemas tehakse õpilastele ülevaade IntelliJ kiirklahvidest.

Klassid II teemas tutvustatakse õpilastele klasside hierarhiat ning *get* ja *set* meetodeid. Klasside hierarhia kasutuselevõtt aitab õpilastel vajadusel luua keerulisema struktuuriga projekti kui ilma selleta. Graafilise kasutajaliidese teises teemas antakse õpilasele suunitlusi erinevate graafilise kasutajaliidese komponentide, nagu *textfield* ja *radio button*, kasutamiseks IntelliJ IDEA keskkonnas. Kui õpilastel on projektidega seoses tekkinud huvi mõne keerulisema graafilise kasutajaliidese lahenduse vastu, näiteks animatsioonide, siis on see hea tund, kus jagada

lisamaterjale, välislinke ning soovitusi. Nende kahe teema kohta õpilased iseseisvaid ülesandeid lahendama ei pea.

2.3. “Programmeerimine keeles Java I” aine kontrolltööd

“Programmeerimine keeles Java I” aines peavad õpilased sooritama kaks kontrolltööd. Esimene kontrolltöö peaks toimuma peale teema “Meetod ja rekursioon” läbimist ning teine kontrolltöö peale teema “Failist lugemine ja faili kirjutamine” läbimist. Kontrolltööde eesmärk on kontrollida õpilaste omandatud teadmisi.

2.3.1. Esimene kontrolltöö

“Programmeerimine keeles Java I” aine esimene kontrolltöö on paberkujul pigem teooria teemadele keskenduv kontrolltöö. Kontrolltöö esimeses osas kontrollitakse õpilaste teadmisi põhiliste Java teooria temadel. Õpilased peavad tegema vahet lihtsamatel andmetüüpidel (*int, double, String, boolean*) ja andmestruktuuridel (massiv, järjend, *HashMap, HashSet*) ning nende kasutusjuhtudel. Õpilased peavad teadma kolme erinevat tsükli tüüpi (*for, for each, while*) ning nende iseärasusi.

Kontrolltöös kontrollitakse õpilaste arusaama meetodi ülesehitusest ning selle komponentide eesmärkidest. Samuti võiks esimene kontrolltöö osa sisaldama küsimust *File* ja *Path* tüüpi isendite loomise kohta. Kontrolltöö teises osas peavad õpilased lugema valmis kirjutatud koodi ning sellest leidma vigu. Vigade otsimise ülesandeid on kaks – esimene ülesanne keskendub rohkem kergetele süntaksi vigadele, näiteks semikooloni puudumine rea lõpust, teine ülesanne sisaldab rohkem loogikavigu. Esimeses vigade otsimise ülesandes on vaja leida vähem vigu ning vead peaksid olema lihtsamini leitavad. Kontrolltöö kolmas osa on lisaülesanded ning ei kata läbitud kohustuslike teemasid. Lisaküsimustele oskab õpilane vastata eelkõige juhul, kui on lahendanud raskema taseme iseseisvaid ülesandeid või uurinud muul põhjusel lisa internetist. Esimeses kontrolltöös on lisamaterjalide kasutamine keelatud. Kontrolltöö näidis on lisas 5.

2.3.2. Teine kontrolltöö

Aine teine kontrolltöö on suunatud õpitud programmeerimisoskuste kontrollimiseks. Kontrolltöö katab kõiki olulisemaid teemasid aines “Programmeerimine keeles Java I”, välja arvatud GUI.

Teine kontrolltöö koosneb sarnaselt esimesele kontrolltööle kolmest osast. Teises kontrolltöös on õpilastel lubatud kasutada lisamaterjale ning interneti. TJPG kursuse “Programmeerimine keeles Java I” jaoks loodud kontrolltöö näidis on esitatud lisa 3 tabelis 11.

Kontrolltöö esimene osa võiks katta failist lugemist ning faili kirjutamist ning olla üles ehitatud selliselt, et see suunaks õpilast kasutama sõnastikke või hulki. Õpilastele tuleks koos kontrolltöö ülesandega esitada ka sisendfaili(de) näidised, et nad teaksid kindlalt, millisel kujul sisendiga arvestada tuleks. Sisendfail võiks sisaldada üleliigseid ridu, et õpilased ei saaks järjest kogu faili lugeda, vaid peaksid saadud andmeid natuke töötleva. Näiteks sobiks selleks erinevate faili osade eraldamine “---” reaga.

Kontrolltöö teine osa peaks olema esimese osa edasiarendus, et õpilastel ei oleks vajadust ühe töö jooksul mitmesse ülesandesse süveneda. Teises kontrolltöö osas tuleb õpilastel demonstreerida klasside loomise oskust. Õpilased peavad teadma mõisteid isendiväli, isend ning konstruktor ja oskama lahendada mõistetega seotud ülesandeid. Õpilased peaksid oskama klassi lisada ka meetodit ning demonstreerima selle kasutust varasemalt loodud peaklassis.

Kolmas kontrolltöö osa on õpilastele vabatahtlik ning annab korrektsel lahendamisel ainesse lisapunkte. Lisapunktide teenimiseks peavad õpilased varasemalt loodud isendiklassi lisama vähemalt ühe parameetriga meetodi, lisama ilma kõigi isendiväljateta konstruktori ning kirjutama *set* ja *get* meetodid vähemalt ühele isendiväljale.

2.4. “Programmeerimine keeles Java I” aine projekt

“Programmeerimine keeles Java I” aine lõpuks peavad õpilased esitama projekti. Projekt on ettenähtud rühmatööna, kusjuures igas rühmas on 2–3 õpilast. Projekti eesmärk on kinnistada kursuses õpitut ning praktiseerida paarisprogrammeerimist. Projekti tegemine üksinda on lubatud vaid erandkorras. Projekti näidiseid saab vaadata failist ProjektiIdeid, mis on leitav Google Drive’ist lingilt: <https://drive.google.com/open?id=1aL9m80zXKqFe0ItQINjSROzOOMN-DP2i>.

Õpilastel tuleb esitada projekti teema ning meeskond neljanda õppenädala teiseks tunniks. Peale seda jääb õpilastel kolm nädalat aega projekti teostamiseks. Selle vältel ei ole õpilastel muid iseseisvaid ülesandeid ning tunni teemad on mõeldud projekti teostusele kaasa aitamiseks. Nii

viiel ja kuuel nädalal on ettenähtud üks nn projektitund, kus ei tutvustata uut teemat ega toimu kontrolltööd. Need tunnid on vajalikud, et õpilased saaksid projektidega seoses abi küsida.

Kõik projekti meeskonna liikmed saavad projekti eest sama palju punkte, mis tähendab seda, et iga õpilase tööd ei hinnata persooniti. Projekt võiks sisaldada graafilist kasutajaliidest. Projekte esitades õpilased demonstreerivad, mida nende loodud programm teeb, selgitavad teistele õpilastele, milleks on see vajalik ning tutvustavad projekti koodi. Kirjutatud Java kood peab olema loetav ning kommenteeritud ja dokumenteeritud vastavalt vajadusele. Õpetajal on õigus eriti heade projekti lahenduste eest anda kogu tiimile lisapunkte.

“Programmeerimine keeles Java I” kursus on õpilaste projekti loomiseks aega paar nädalat ning mitmed õpilased võivad tunda, et sellest jääb väheks. “Programmeerimine keeles Java I” aines loodud projekti võivad õpilased huvi korral edasi arendada “Programmeerimine keeles Java II” aine raames, mille kirjelduse leiab järgmisest peatükis.

3. “Programmeerimine keeles Java II” valikkursus

“Programmeerimine keeles Java II” aine on loodud jätkukursusena “Programmeerimine keeles Java I” ainele. “Programmeerimine keeles Java II” kursust on soovitatav läbida õpilasel, kellele “Programmeerimine keeles Java I” kursus meeldis ning ta sooviks Javast ning IT valdkonnast rohkem teada saada. Juhul, kui õpilane ei ole “Programmeerimine keeles Java I” kursust läbinud, peab tal “Programmeerimine keeles Java II” kursusel osalemiseks olema eelnev Java programmeerimise kogemus. “Programmeerimine keeles Java II” kursuse ainekava on esitatud lisas 6 ning Moodle testid on esitatud lisas 9.

Kogu “Programmeerimine keeles Java II” kursuse jooksul tegelevad õpilased oma projekti loomise ning parendamisega. Iganädalasel toimub üks tund, kus tegeletakse ainult projektiga. Õpetaja roll selles tunnis on nõustaja ning suunaja. Lisaks projektile peavad õpilased lahendama iseseisvaid ülesandeid ning sooritama ühe kontrolltöö. Aine käigus saavad õpilased sooritada kolm Moodle testi, mille eesmärk on toetada kontrolltööks õppimist, ning üks Moodle test, mille eesmärk on eneseanalüüs ehk tagasivaade tehtud projektile ning selle käigus õpitud oskustele.

Aine materjalide põhjal on võimalik hindamist kujundada iga kooli ning õpetaja vajadustele vastavaks. Aines saab kasutada nii eristavat kui ka mitteeristavat hindamist. Töö autori välja töötatud hindamisjuhend “Programmeerimine keeles Java II” kursuse läbiviimiseks TJPGs on toodud lisas 6. TJPG õpilastel oli tundides kohalkäimine vabatahtlik, kuid projekti tundides osalemine oli kohustuslik ning iga versteposti esitamise tunnis peadi olema kohal vähemalt üks meeskonnaliige.

3.1. “Programmeerimine keeles Java II” valikaines kaetud teemad

“Programmeerimine keeles Java II” aine teemade ning iseseisvate ülesannete eesmärgiks on tutvustada õpilastele keerulisemaid Javaga seotud konseptsioone, toetada projekti loomist ning korrata seni õpitut. Õpilaste silmaringi avardamiseks ning võimaluste tutvustamiseks, on ainesse planeeritud tarkvara ettevõtteid tutvustav teema. “Programmeerimine keeles Java II” aine on pigem projektipõhine aine ning iseseisvate ülesannete hulk on väiksem kui “Programmeerimine keeles Java I” aines. Kõik aine teemade slaidid on leitavad lisast 7.

3.1.1. Git ehk versioonihaldussüsteem

“Programmeerimine keeles Java II” aine esimene teema on Git ehk versioonihaldussüsteem. Versioonihaldussüsteem on tarkvara, mis lubab salvestada muudatuste ajalugu ning tagab kasutajate ligipääsu projektile igast internetiühendusega seadmest. [18] See on “Programmeerimine keeles Java II” aines oluline, kuna iga nädal toimuvad projekti tunnid. Kasutades versioonihaldussüsteemi ei teki olukorda, kus meeskond ei saa tunnis tööd teha, kuna projektifailid ununesid kodusse arvutisse.

Versioonihaldussüsteemi teemale vastab üks iseseisev ülesanne j1, mis on leitav lisast 8. Ülesande eesmärk on lahendada avalik Git-i õppimise keskkonna juhend Git-ist. Juhend tutvustab õpilasele põhilisi Git-i käske ning tavasid. Õpetajale peab esitama juhendi järgi loodud repositooriumi lingi.

3.1.2. “Programmeerimine keeles Java I” teemade kordamine

“Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II” ainete läbimise vahele võib jääda pikem aeg, näiteks juhul kui õpilane läbib “Programmeerimine keeles Java I” aine 10. klassis ja “Programmeerimine keeles Java II” 12. klassis, sest kursuse algusesse on planeeritud “Programmeerimine keeles Java I” aine teemasid kordav tund. Õpilastele tuletatakse meelde programmi sisendi ja väljundi reeglid, failist lugemine ja faili kirjutamine, neli õpitud andmestruktuuri, kõik kolm tsükli vormi, tingimuslaused, meetodid ning klassid. Õpilased peavad kordava teema kohta lahendama iseseisvad ülesanded j2 ning j3, mis on leitavad lisast 8.

3.1.3. *Lambdad ja Streamid*

“Programmeerimine keeles Java II” aines õpetatakse õpilasi kasutama *Lambdasid* ja *Streamid* ehk andmevoogusid. *Lambdasid* ja andmevooge kasutades on võimalik kirjutada selgemat ja paremini loetavat koodi kui ilma nendeta. Andmevoogudega töötamine avardab arusaama andmehulkadega töötamisest ning on hea esimene teema, millega soovi korral tutvustada õpilastele paralleelse programmi loomist. “Programmeerimine keeles Java II” aine raames paralleelsusest ei räägita. “*Lambdad ja Streamid*” teema kohta peavad õpilased lahendama iseseisvad ülesanded j4 ning j5, mis on leitavad lisast 8.

3.1.4. Erindid

Neljanda teemana õpetatakse õpilastele erindeid. Pikemate ning keerukamate projektide loomisel, nagu seda on “Programmeerimine keeles Java II” projekt, tekib olukordi, kus võib tekkida erindeid. Selleks, et õpilased saaksid oma projektides erindeid õigesti deklareerida või käsitleda, tuleb “Erindid” teemas õpilastele selgitada, et *try-catch* meetoditega erindite käsitlemine on ohtlik, kuna nii võivad programmi tekkida vead, mille olemasolust programmi kirjutaja teadlik ei ole. Erindite teema kohta peavad õpilased lahendama iseseisvad ülesanded j6 ning j7, mis on leitavad lisast 8. Ülesandes j6 ei pea õpilased ise ülesande koodi valmis kirjutama, vaid peavad parandama vigast etteantud koodi.

3.1.5. Graafilise kasutajaliidese head tavad

“Programmeerimine keeles Java II” viiendas teemas õpetatakse õpilasi kasutama IntelliJ keskkonnas Open Javafx 11, mis on Java graafiliste kasutajaliideste loomise platvorm. Lisaks tehnilisele Java projekti ülesseadmise juhendamisele, arutletakse selles tunnis õpilastega heade ning halbade kasutajaliideste lahenduste üle Los Angelese parkimistahvlite näitel. On oluline, et klassis tekiks diskussioon ning õpilased saaksid aru, millistel juhtudel on tegemist halva kasutajaliidese praktikaga. Teemaga on seotud iseseisev ülesanne j8, mis on leitav lisast 8.

3.1.6. Algoritmid

“Programmeerimine keeles Java II” aine kuues teema on algoritmid. Algoritmide teemas ei õpetata ühtegi konkreetset algoritmi kasutama ega programmeerima, vaid räägitakse üldiselt, mis on algoritmid ning milleks on neid vaja. Erinevate algoritmide vahet selgitatakse lihtsate astendamise, otsimise ning sorteerimise näidetega. On oluline, et õpilased saaksid aru, et erinevalt kirjutatud koodid võivad olla erineva kiirusega ning ilma kasutatud algoritmi teadmata ei ole võimalik varasemate programmi kiiruste põhjal arvutada iga juhu töökiirust. Algoritmide teema kohta õpilastel ühtegi iseseisvat ülesannet ei ole, kuid selles tunnis tuleks õpilastele anda ülesanne j9. Ülesandes j9 peavad õpilased demonstreerima internetist andmete laadimist. Lahenduskäiguks vajaliku informatsiooni peavad nad ise internetist leidma. Ülesanne on leitav lisast 8.

3.1.7. Tarkvara ettevõtted

Seitsmendas kursuse teemas tutvustatakse õpilasele tarkvara ettevõtteid ning praktilise kogemuse saamise võimalusi. Aine üks põhieesmärke on avardada õpilaste silmaringi ning aidata neil teha kaalutletud otsus edasiste õpingute suhtes. Kindlasti on kasu sellest, kui õpilased teavad, et tarkvara ettevõttes on lisaks arendajale ja testijale ka palju teisi rolle. Samuti tuuakse välja tarkvara ettevõttes töötamisega kaasnevad hüved ning kirjeldatakse ühe arendaja ning analüütiku päeva. Analüütiku päeva kirjeldus on kirja pandud lõputöö autori enda isiklikule kogemusele toetudes, sest autor töötab tarkvara analüütikuna. Arendaja päeva kirjeldus on saadud vabas vormis intervjuu käigus Nortali tarkvaraarendajalt.

Seitsmendas teemas selgitatakse ka suvepraktika programme, mida enamik suuremaid Eestis tegutsevaid tarkvara ettevõtteid igal aastal korraldavad. Suvepraktika programmide käigus valitakse praktikale igas vanuses ja erineva taustaga inimesi. Praktika programm on ideaalne võimalus aktiivsel gümnaasiumi noorel varajase töökogemuse saamiseks. Selle teema kohta pole õpilastel vaja esitada ühtegi iseseisvat ülesannet.

3.1.8. Klassid ja rekursioon

Kaheksandas “Programmeerimine keeles Java II” kursuse teemas käsitletakse taas klasse ning rekursiooni. Klasside teema läheb edasi abstraktsete klasside mõistega. “Programmeerimine keeles Java II” aine on projektikeskne ja õpilastel on palju aega tegeleda projektiga, sest on tõenäoline, et projektid kasvavad suureks. Abstraksed klassid on hea moodus, kuidas väga mahukaks projektis hoida selget klassi hierarhiat ning vältida koodi kopeerimist. Abstraktsete klasside teema peale õpilastel iseseisvaid ülesandeid ei ole.

Kui “Programmeerimine keeles Java I” aines oli rekursiooni teema vabatahtlik ning sellele vastas raskema taseme ülesanne, siis “Programmeerimine keeles Java II” aines on rekursiooni teema kohustuslik kõigile õpilastele, kuid sellele vastav iseseisev ülesanne j10, mis on leitav lisast 8, on kergem, kui seda oli “Programmeerimine keeles Java I” aine rekursiooni ülesanne c10. Rekursiooni teema õpetamisel on õpilastele kindlasti kasuks kui tahvli peal arutleda läbi lihtsama rekursiivse meetodi käik. Näiteks saab arutlemiseks kasutada “Programmeerimine keeles Java II”, “Klassid ja rekursioon” teema slaididel olevat rekursiivset meetodit

“arvudeSummaAkuniB(int A, int B)”, kuna see meetod on oma olemuselt lihtne. Slaidid on leitavad lisast 7.

3.2. Kontrolltöö

“Programmeerimine keeles Java II” aines on üks kontrolltöö. Kontrolltöö koosneb kahest osast – programmeerimise osa (maksimaalselt 5 punkti) ning veebipõhine test (maksimaalselt 12 punkti). Programmeerimise osa kontrollib “Streamide ja Lambdad” ning “Erind” teemade tundmist ning õpilaste oskust lugeda ja mõista võõrast koodi. Lisaks peavad õpilased mäletama aines “Programmeerimine keeles Java I” õpitud põhiteadmisi järjendite kohta. Kontrolltöö teooria osa katab teemasid kuni “Klassid ja rekursioon” teemani. Kontrolltöö näidisülesanne kt1 on leitav lisast 8.

Kontrolltöös näidisülesandes on õpilastele ette antud kood, milles on kaks viga. Etteantud kood loeb postimees.ee lehelt sisse kõik uudiste pealkirjad ning üritab nende seast tuvastada pealkirjas, milles on mainitud Eestit. Õpilased peavad vigase koodi ära parandama ning seda täiendama nii, et kood leiaks kõigi Eestist mitte-rääkivate pealkirjade keskmise pikkuse.

Kontrolltöö programmeerimise osa hindamisjuhend:

- 1 punkt – Esimese vea parandamine;
- 2 punkti – Teise vea parandamine;
- 2 punkti – Koodi täiendamine.

Kontrolltöö teooria osa annab kokku 12 punkti, seega kokku võib õpilane teenida 17 punkti. Aine arvestusse läheb kirja maksimaalselt 15 punkti – seega kui õpilane teenib rohkem kui 15 punkti, siis läheb kirja tulemus 15 punkti. See annab madalamate punktide saajatel võimaluse saada paremat hinnet ka juhul, kui mõni kontrolltöö osa ei õnnestunud.

3.3. Projekt

“Programmeerimine keeles Java II” aine projekt moodustab 55% kogu aine hindest. Projekt on rühmatöö ning tänu projekti tundides tehtavale paarisprogrammeerimisele areneb õpilasel koodi lugemise ning headele tavadele vastava koodi kirjutamise oskus. Rühmatöö annab ettekujutuse

tarkvara ettevõttes töötamisest, kuna ka seal tehakse projekte rühmades. Iga projekti teema on erinev, sest projekt suunab õpilasi lisainformatsiooni saamiseks kasutama internetti, arendades seeläbi nende digipädevusi. Projektiga hakkavad õpilased tegelema juba kursuse esimesel nädalal. Projektis peab olema kasutatud graafilist kasutajaliidest.

Projekti hinne koosneb kahest osast – verstepostide tulemused ning projekti kokkuvõtva esitluse tulemus. Tartu Jaan Poska gümnaasiumis kestab periood 6 nädalat ning esimesel viiel nädalal toimub versteposti esitamise tund. Viimasel nädalal toimuvad projektide kokkuvõtavad esitlused. Verstepostid annavad 25% ning kokkuvõttev esitlus annab 30% kogu aine lõpphindest. TJPG said õpilased iga versteposti eest maksimaalselt 5% aine hindest. Kui kasutada sama projekti ülesehitust teistes koolides, tuleks verstepostide arvu suurendada või vähendada vastavalt õppetsükli pikkusele ning sellega koos ka muuta ühe versteposti hinde väärtust.

Projekt tuleb üles laadida mõnda versioonihaldussüsteemi, näiteks Githubi. Kõik meeskonna liikmed peavad projekti muudatuste üles laadimiseks kasutama versioonihaldussüsteemi. Projekti kohta peab versioonihaldussüsteemis olema viki leht. Viki avalehel peavad õpilased kirjeldama oma projekti teemat ning selgitama projektisisese ülesannete jaotuse. Kirja pandud ülesannete jaotus võib projekti käigus muutuda, kuid on oluline, et õpilastel oleks olemas esialgne plaan.

Iga versteposti kontrollimise tunniks peavad õpilased tegema vikisse uue sissekande ning kirjeldama projekti edenemist. Õpilasi tuleb julgustada viki lehtedel jagama huvitavaid linke artiklitele, foorumitele ja videodele, mis neid projekti loomisel aitasid. Versteposti kirjeldus võiks vastata järgmistele küsimustele:

- Mis sai nädala jooksul tehtud? Kuidas toimus tööde jaotus?
- Mis läks hästi? Mis läks halvasti?
- Mis on plaanis järgmiseks nädalaks?

Projekti kokkuvõtvas esitluses peavad õpilased demonstreerima oma loodud programmi tööd. Ühiselt vaadatakse üle erand olukorrad, näiteks valede sisendite andmine programmile ning koodi kvaliteet. Iga meeskonna liige räägib, mis olid tema vastutusosalad tiimis ning kuidas ta enda meelest nendega hakkama sai.

Projekti hindamine on nii personaalne kui ka meeskonnakeskne. Kokkuvõtva esitluse eest võib iga õpilane maksimaalselt (ilma lisapunktideta) saada 30 punkti (ehk 30% aine lõpphindest). TJPGs läbiviidud kursuse jaoks töötas bakalaureusetöö autor välja projektide lõppesitluste hindamiseks hindamisjuhendi (vt tabel 1 lk 32). Õpetajatele jääb õigus oma äranägemise järgi anda igas kategoorias lisapunkte.

Tabel 1. “Programmeerimine keeles Java I” projekti hindamisjuhend

<p>Tehtud töö ja isiklik panus projekti (personaalne 15p). Hinnatakse kui hästi õpilane täitis endale seatud eesmärgid ja ülesandeid</p>	
13-15p	Õpilane täitis kõik endale võetud ülesanded ja eesmärgid ning tiimikaaslased olid temaga rahul. Või põhjendas puudujäägid mõjuvalt.
8-12p	Õpilane täitis enamiku (üle poolte) endale võetud eesmärkidest ja ülesannetest ning tema tiim oli tema tööga enamasti rahul.
4-7p	Õpilase eesmärkide täitmisel esines olulisi puudujääke ja/või tema tehtud töö oli lohakas ja/või tema tiimikaaslastel on tema töö suhtes argumenteeritud pretensioone.
0-3p	Õpilane ei panustanud absoluutselt rühmatöösse ja/või rühmaliikmetel on väga mõjuvad negatiivsed argumendid.
<p>Kood (meeskondlik 10p) Hinnatakse projekti koodi loetavust, heade tavade järgimist ja koodi korrektset töötavust.</p>	
10p	Kood töötab. Kood on lihtsasti loetav. Kood on vajadusel kommenteeritud. Kaetud on kõik äärehujud.
7,5p	Kood töötab ja tiimis saadakse koodist aru (kõik liikmed ei pea kõigest aru saama, aga iga koodi osa kohta peab olema vähemalt üks õpilane, kes seda koodi mõistab). Osad äärehujud võivad olla katmata.
5p	Programm täidab põhieesmärgi vähemalt ühe sisendiga. Kõik või peaaegu kõik äärehujud on katmata.
2,5p	Kood ei tööta, on enam-vähem loetav ning tõenäoliselt parandatav.
0p	Kood ei tööta, ei ole loetav, ei ole parandatav ning tiim ei ole küsinud abi.
<p>Keerukus (meeskondlik 5p) Hinnatakse projekti keerukuse ja tiimi taseme vastavust</p>	
5p	Projekt vastab keerukuselt õpilaste tasemele.
1p-4p	Projekti keerukus ja õpilaste tase ei ole vastavuses.
0p	Projekt on väga primitiivne – võrdväärne iseseisva ülesande keerukusega.

4. Tartu Jaan Poska gümnaasiumi õpilaste õpitulemuste ja tagasiside analüüs

Selleks, et hinnata loodud materjalide ja ainekavade sobivust gümnaasiumiastmes kasutamiseks, analüüsiti õpilaste õpitulemusi ning tagasisidet loodud “Programmeerimine keeles Java I” ning “Programmeerimine keeles Java II” ainetele. Õpiedukuse analüüsi aluseks võeti õpilaste perioodi hinded ning hindeliste tööde keskmised tulemused. Tagasiside küsitlus viidi õpilaste hulgas läbi peale “Programmeerimine keeles Java II” kursuse lõppu, ajavahemikus 22. aprill kuni 13. mai 2019. Tagasiside küsimustik oli nõnda pikalt avatud, kuna gümnaasiumi uurimistöödega seoses tuli õpilastel järjest vastata mitmetele küsimustikele ning vastamisaktiivsus oli madal. Küsimustiku link on esitatud lisas 10.

4.1. TJPG õpilaste õpitulemuste analüüs kursustel “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”

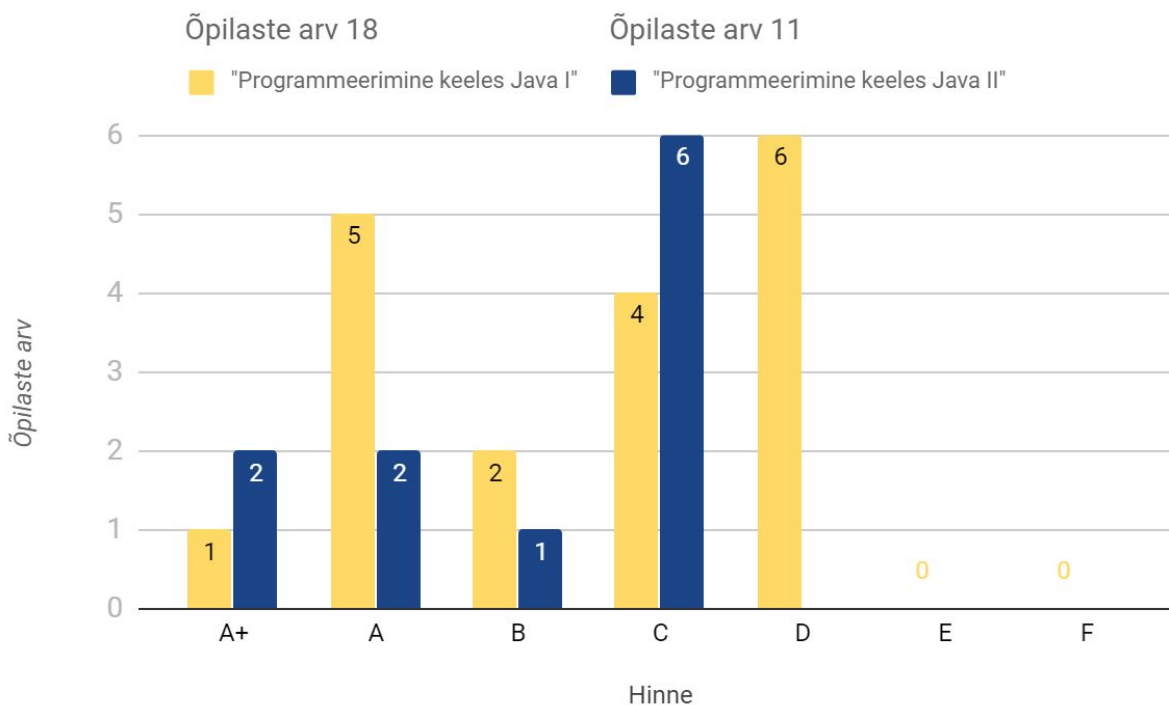
“Programmeerimine keeles Java I” kursus toimus 17. detsember 2018 kuni 15. veebruar 2019, kusjuures viimane nädal ei toimunud tavapärasest õppetööd, kuna õpilastel oli arvestuste nädal. Arvestuste nädalal tundi ei toimunud. “Programmeerimine keeles Java I” kursuse keskel oli vaheaeg ajavahemikus 22. detsember 2018 kuni 06. jaanuar 2019. “Programmeerimine keeles Java II” kursus toimus 18. veebruar kuni 12. aprill 2019 ning vahemikus 23. veebruar kuni 03. märts 2019 oli õpilastel vaheaeg.

Kokku osales “Programmeerimine keeles Java I” kursusel 18 õpilast ning “Programmeerimine keeles Java II” kursusel 11 õpilast. Kursusel “Programmeerimine keeles Java I” said õpilased enim hindeks D (~33% kõigist õpilastest) või A (~28%). Neljal õpilasel tuli enne perioodi hinde välja panemist lahendada lisaülesanne, kuna nad ei olnud õigeaks ajaks esitanud piisavas koguses iseseisvaid ülesandeid. “Programmeerimine keeles Java II” kursuse õpilased said enamik (~55%) hindeks C. Selle aine lõpus ei pidanud ükski õpilane oma hinnet parandama, vaid positiivne hinne saadi esimesel katsel.

Mõlemas aines ei saanud ükski õpilane perioodi hindeks E ega F. “Programmeerimine keeles Java I” kursusel sai üks õpilane (~6%) ning “Programmeerimine keeles Java II” kursusel kaks

õpilast perioodi hindeks A+ (~18%), kuna olid aine lõpuks teeninud piisavalt lisapunkte. Õpilaste perioodi hinnete kokkuvõte on esitatud joonisel 1.

Kui teisendada hinded vastavalt kooliastme lõpetamise hinnete teisendamise reeglitele, saame “Programmeerimine keeles Java I” kursuse õpilaste keskmiseks perioodi hindeks $(5*6+4*6+3*6)/18 = 4,00$ ning “Programmeerimine keeles Java II” keskmiseks perioodi hindeks $(5*4+4*7)/11 = \sim 4,4$.

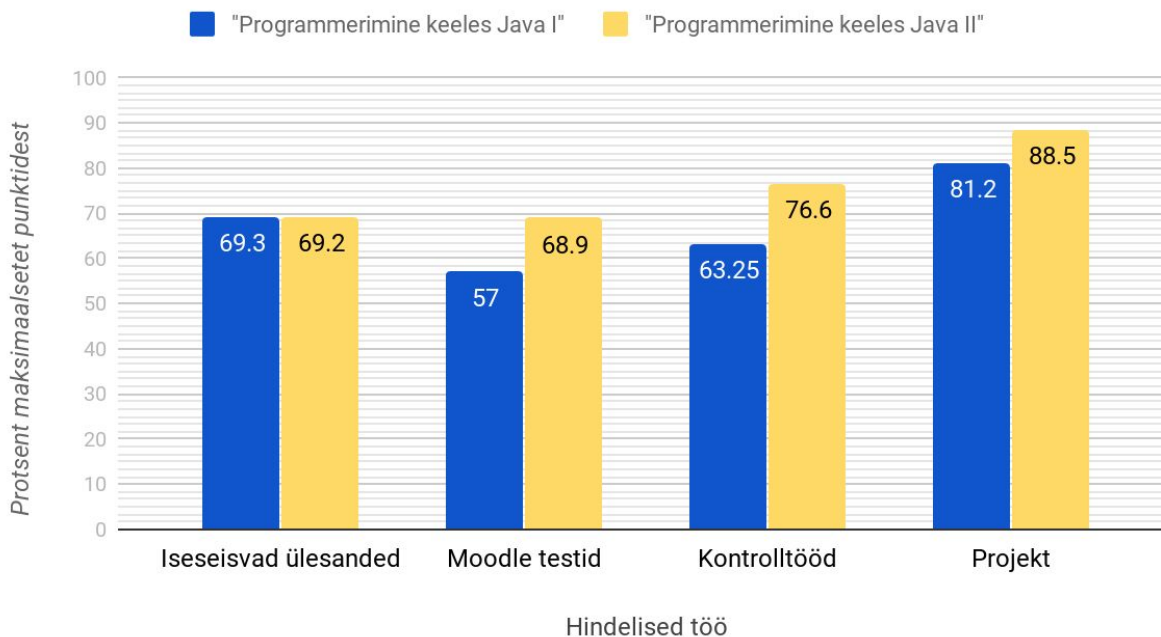


Joonis 1. TJPG õpilaste perioodi hinnete jaotus aines “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”.

Jooniselt 2 on näha, et mõlemas aines läks õpilastel kõige edukamalt projektidega. “Programmeerimine keeles Java I” kursusel teenisid õpilased Moodle testide eest keskmiselt 11,9% vähem punkte kui “Programmeerimine keeles Java II” kursuse õpilased. “Programmeerimine keeles Java I” kursuse kontrolltööde keskmine tulemus oli 13,35% madalam kui “Programmeerimine keeles Java II” kursusel. “Programmeerimine keeles Java II”

kontrolltööde tulemus on arvestatud kontrolltöö absoluutsest maksimaalsest tulemusest, milleks oli 17 punkti, mitte hinde arvutamiseks kasutatud maksimumist, milleks oli 15 punkti.

Maksimaalne tulemus 100%



Joonis 2. TJPG õpilaste keskmised tulemused kogu klassi peale hindelise töö kaupa aines “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”.

Kõigi õpilaste, kes läbisid 2018/2019 õppeaastal “Programmeerimine keeles Java I” ning “Programmeerimine keeles Java II” ainet perioodi hinne oli “Programmeerimine keeles Java II” aines kõrgem kui “Programmeerimine keeles Java I” aines. Üks õpilane teenis “Programmeerimine keeles Java I” aines hinde C, kuid “Programmeerimine keeles Java II” aines saavutas ta hinde B. Nelja õpilase perioodihinne aines “Programmeerimine keeles Java I” oli D, kuid “Programmeerimine keeles Java II” aines oli C. Ükski õpilane, kes saavutas “Programmeerimine keeles Java I” aines 2018/2019. õppeaastal hindeks A ei osalenud “Programmeerimine keeles Java II” kursusel.

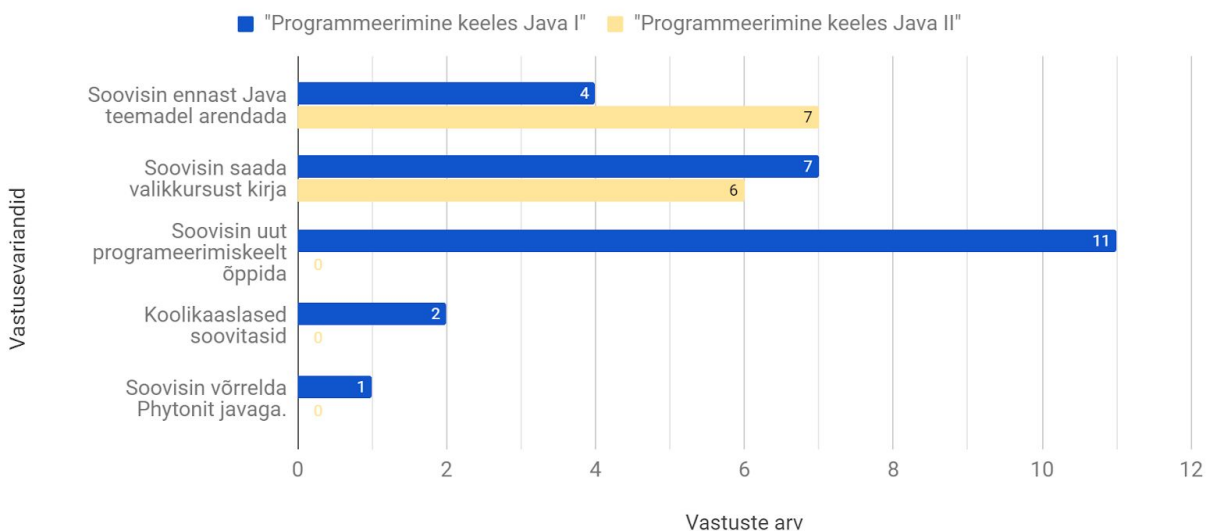
4.2. Tartu Jaan Poska gümnaasiumi õpilaste tagasiside analüüs kursustel “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II”

“Programmeerimine keeles Java I” valikkursuse läbinud 18 õpilasest vastas tagasiside küsimustikule 14 õpilast (~78% kõigist kursusel osalenutest) ning “Programmeerimine keeles Java II” kursust läbinud 11 õpilasest vastas tagasiside küsimustikule 7 õpilast (~64%).

Joonisel 3 on toodud õpilaste vastused küsimusele, miks nad soovisid programmeerimise valikkursuseid läbida. Tulemustest selgub, et enamik (~79%) “Programmeerimine keeles Java I” kursuse läbinud vastajatest soovis õppida uut programmeerimiskeelt ning kõik “Programmeerimine keeles Java II” kursusel osalenud vastajatest soovisid ennast Java teemadel arendada.

"Programmeerimine keeles Java I" kursuse läbinud vastajate arv 14

"Programmeerimine keeles Java II" kursuse läbinud vastajate arv 7



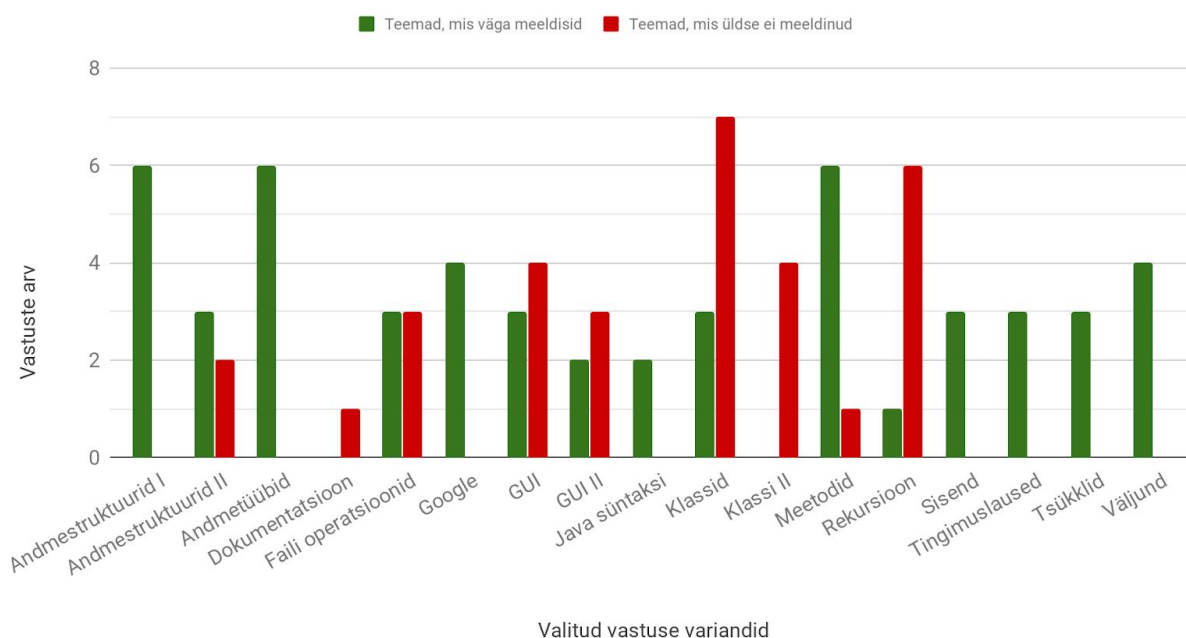
Joonis 3. TJPG õpilaste vastused küsimusele “Miks soovisid valikaines osaleda?”.

Õppekava täitmine oli oluline seitsmele (50%) “Programmeerimine keeles Java I” kursuse läbinule ning kuuetele (~86%) “Programmeerimine keeles Java II” kursuse läbinud vastajale. Üks õpilane kirjutas vabas tekstilises vastuses, et ta soovis võrrelda programmeerimiskeelt Python keelega Java, mis on samuti sarnane uue keele õppimise soovile. Küsimuse vastuste põhjal saab öelda, et ~86% “Programmeerimine keeles Java I” läbinud vastajatest ning kõigil

“Programmeerimine keeles Java II” kursust läbinud vastajatest oli sisemine motivatsioon kursusel osaleda.

“Programmeerimine keeles Java I” aine õpilastele meeldisid kõige rohkem “Andmestruktuurid I”, “Andmestruktuurid II” ning “Meetodid” teemad. Kõige vähem meeldisid õpilastele mõlemad Java klasside õppimise teemad ning rekursiooni õppimine. Rekursiooni oskamine ei ole “Programmeerimine keeles Java I” õpilastele kohustuslik, vaid on sisse toodud lisa teemana, et õpilastel oleks soovi korral võimalik rekursiooniga tutvuda. Täpsemaid tulemusi on näha joonisel 4. Igal õpilasel oli võimalik valida kuni kolm vastusevarianti kummagi küsimuse kohta.

Vastajate arv 14

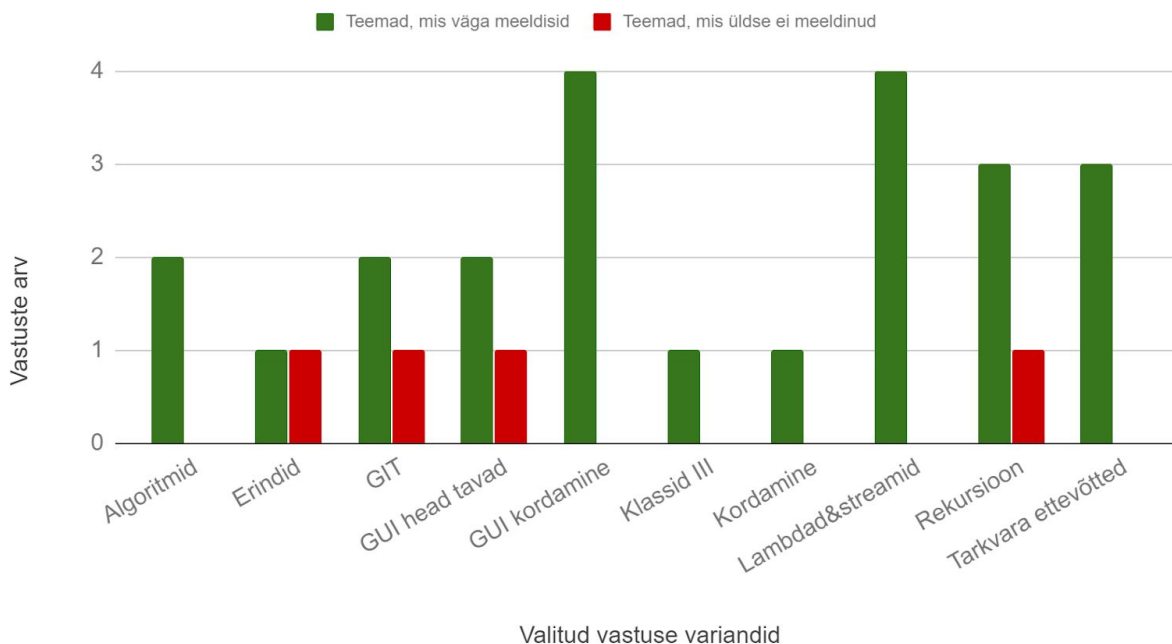


Joonis 4. TJPG õpilaste vastused küsimustele ”Millised olid lemmikud Programmeerimine keeles Java I läbitud teemad?” ja “Millised teemad Programmeerimine keeles Java I kursuses üldse ei meeldinud?”.

“Programmeerimine keeles Java II” aines meeldisid õpilastele kõige rohkem graafilise kasutajaliidese kordamine ja lambdate ning andmevoogude õppimine. Ligikaudu 57% küsitlusele vastanud õpilastest pidas neid teemasid parimateks. Üllatuslikult on kolm õpilast seitsmest valinud lemmik teemaks rekursiooni. Kui “Programmeerimine keeles Java I” aines ei olnud rekursiooni õppimine kohustuslik, siis “Programmeerimine keeles Java II” teemas tuli

õpilastel endil koostada üks lihtne rekursiivne meetod. Rekursiooni ülesande esitas edukalt kaheksa õpilast üheteistkümnest. “Programmeerimine keeles Java II” kursusel ei olnud ühtegi kindlat teemat, mis õpilastele ei meeldinud. Õpilaste vastused on toodud joonisel 5. Igal õpilasel oli võimalik valida kuni kolm vastusevarianti kummagi küsimuse kohta.

Vastajate arv 7



Joonis 5. TJPG õpilaste vastused küsimustele ”Millised olid lemmikud Programmeerimine keeles Java II läbitud teemad?” ja “Millised teemad Programmeerimine keeles Java II kursus üldse ei meeldinud?”

Lisas 11 joonisel 44 on näha, et enamikule õpilastele meeldis nii “Programmeerimine keeles Java I” kui ka “Programmeerimine keeles Java II” aines vabatahtlik tundidest osavõtmise kord. Mõlema aine küsimustikule vastanute hulgas ei leidunud ühtegi õpilast, kellel ei oleks meeldinud iseseisvad või kontrolltöoks ettevalmistavad Moodle testid. Kõige enam vastanduvad õpilaste arvamused projekti osas. Viis õpilast 14st “Programmeerimine keeles Java I” osalenud vastanuist arvas, et neile projekt meeldis, samas 2 “Programmeerimine keeles Java I” kursuse läbinud õpilast pidas seda üheks halvemaks osaks aimest. Projekt ei meeldinud ka ühele “Programmeerimine keeles Java II” aine läbinud õpilasele. Kõige enam olid “Programmeerimine keeles Java I” aine õpilased rahulolematud esimese kontrolltööga. Küsimusele, mis kursuse

juures eriti meeldis, kirjutasid õpilased vabasse tekstilisse vastusesse järgmist: “Toetav õpetaja”, “Et anti aega juurde iseseisvate ülesannete lõpetamiseks.” ning “Iseseisvate tööde esitamise keskkond”. Küsimusele, mis kursuse juures üldse ei meeldinud, õpilased vastuseid ei andnud.

Õpilastel oli võimalus anda tagasiside kursustele üheksa väite kaudu. Hinnangud toodud väidetele on esitatud lisas 11, joonistel 42 ja 43. Nii “Programmeerimine keeles Java I” kui ka “Programmeerimine keeles Java II” aine läbinud küsimustikule vastajad arvasid, et ainete iseseisvad ülesanded olid pigem huvitavad ning ei olnud liialt keerulised. Mõlema kursuse õpilased tõdesid, et Moodle testid aitasid kontrolltöödeks õppimisele kaasa. Kuigi õpilased leidsid, et “Programmeerimine keeles Java II” kursuse kontrolltöö ei olnud nende jaoks liiga raske, siis “Programmeerimine keeles Java I” kursuse kontrolltöid peetakse pigem keerulisteks. Ligikaudu 21% õpilastest arvas, et “Programmeerimine keeles Java I” kursuse kontrolltööd olid pigem liialt keerulised ning 43% õpilastest olid kahtleval seisukohal. Vaatamata keeruliste kontrolltöödele meeldisid õpilastele siiski kursuse materjalid ning ülesehitus. Ligikaudu 79% õpilastest arvas, et “Programmeerimine keeles Java I” kursus pigem või täiesti täitis nende ootusi ning ligikaudu 86% õpilastest jäi kursusega pigem või täiesti rahule. Üks õpilane tunnistas, et ta pigem ei jäänud kursusega rahule, kuid oma vastust ta täpsemalt ei selgitanud. Suurema enamuse (~86%) arvates kursus “Programmeerimine keeles Java II” pigem või täiesti täitis nende ootusi ning õpilased jäid kursusega pigem või täiesti rahule. Lisas 11, tabelis 44 on välja toodud õpilaste soovitusid aine korralduse ning materjalide muutmiseks.

4.3. TJPG õpilaste õpitulemuste ning tagasiside küsitluse analüüsi järeldused

Õpilaste perioodi hinnete jaotuse ning keskmise hinde põhjal võib öelda, et kummagi valikkursuse hinded ei olnud liiga madalad ega kõrged. “Programmeerimine keeles java II” kursuse hinded on kõrgemad kui “Programmeerimine keeles Java I” kursuse hinded, kuna seda kursust valisid tõenäoliselt õpilased, kellel oli sisemine motivatsioon ja huvi Javat süvitsi õppida. Mõlema kursuse ajal oli õpilastel vaheaeg, kuid “Programmeerimine keeles Java I” kursuse ajal toimunud vaheaeg oli nädala võrra pikem kui detsembri–jaanuari kuu vaheaeg ning see võis õpilaste õppe edukusele negatiivselt mõjuda, kuna peale vaheaega kooli tulles oli raske meenutada varem õpitut.

Kõige edukamalt läks õpilastel mõlemas aines projektidega. Samuti hindasid õpilased projekti kõrgelt tagasiside küsitluses. See on ka arusaadav, kuna projekt on õpilaste enda valitud teemal programmi kirjutamine ning seetõttu oli ilmselt õpilastel selle vastu suurem huvi kui teiste hindeliste tööde vastu.

Kuna Moodle testid andsid kursuse perioodi hindesse vähe punkte (kokku 10% kursuse hindest), siis ei pööranud õpilased nendele suurt tähelepanu. Moodle testide eesmärk on aidata valmistuda kontrolltöödeks. “Programmeerimine keeles Java I” aines teenisid õpilased vähem punkte nii Moodle testide kui ka kontrolltööde eest kui “Programmeerimine keeles Java II” aines. Sellest ning õpilaste tagasisidest johtuvalt võib väita, et Moodle testide järjepidev lahendamine aitab õpilastel parema kontrolltöö tulemuse saavutamisele kaasa. Õpilaste tagasiside analüüsi põhjal saab öelda, et nad jäid iseseisvate ülesannetega rahule ning pidasid neid huvitavateks.

Õpitulemuste ning tagasiside analüüsi põhjal soovib autor ainet edaspidi läbi viies vähendada “Programmeerimine keeles Java I” kursuse kontrolltööde mahtu või muuta kontrolltööde mõni osa vabatahtlikuks. Eriti oluline on see koolides, kus ühe tunni pikkus on 45 minutit. Kindlasti tuleks õpilasi julgustada lahendama Moodle teste, kuna nii õpitulemustest kui ka tagasiside küistlusest selgus, et Moodle testide lahendamine aitas kontrolltöödeks õppimisele kaasa. Muudes aspektides olid õpilased aines edukad ning jäid kursustega rahule ning bakalaureusetöö autor soovib edaspidiseid aineid korraldada samadel põhimõtetel, muutes ainekavasid sobivaks konkreetse kooli, õpetaja ning õpilaste vajadustele.

Kokkuvõte

Bakalaureusetöö eesmärk on luua gümnaasiumiastmes kasutatavad programmeerimiskeele Java õpetamise ainekavad ja materjalid, mille põhjal läbi viidud kursusel osalemise tulemusena on õpilane paremini teadlik IT erialade väljakutsetest ja võimalustest ning ta oskab koostada lihtsamal tasemel Java programme. Bakalaureusetöö raames sai loodud gümnaasiumiastme valikkursustele “Programmeerimine keeles Java I” ja “Programmeerimine keeles Java II” materjalid ning ainekavad.

Bakalaureusetöö autor viis loodud materjalide ning ainekavadega läbi kaks valikkursust Tartu Jaan Poska gümnaasiumis ajavahemikus 18. detsember 2018 kuni 12. aprill 2019. Valikkursuste õpilased demonstreerisid programmeerimisoskust projektide ja iseseisvate ülesannete esitamisega. IT erialade väljakutsetest said õpilased aimu jooksvalt iseseisvate ülesannete ja projekti küsimustele vastuseid otsides ning “Tarkvara ettevõtted” teemat läbides. Omandatud teadmisi demonstreerisid õpilased Moodle testides.

Kontrollimaks, kas loodud materjalid ja ainekavad sobivad gümnaasiumiastme valikkursuste korraldamiseks, viidi peale valikkursuste lõppu õpilaste seas läbi kursusega seotud rahuloluküsitlus ning paluti anda soovitusi ainekavade ning materjalide parendamiseks. Õpilaste õpitulemusi ning tagasiside tulemusi analüüsiti ning jõuti järeldusele, et loodud ainekavad ning materjalid sobivad TJPGga sarnase õppekorraldusega gümnaasiumides kasutamiseks ning õpilased jäid kursustega rahule.

“Programmeerimine keeles Java I” kursuse jaoks loodi 12 teema jagu slaidi esitlusi, 54 iseseisvat ülesannet kolmes erinevas raskusastmes, neli kontrolltööde õppimist toetavat Moodle testi ning kaks kontrolltöö näidist. “Programmeerimine keeles Java II” kursuse jaoks töötati välja 9 teema jagu slaidiesitlusi, 10 iseseisvat ülesannet, kolm kontrolltööde õppimist toetavat Moodle testi ning üks õpilase eneseanalüüsi Moodle test. “Programmeerimine keeles Java II” aine projekti hindamiseks töötati välja detailne hindamisjuhend, mis lihtsustab nii õpilaste kui ka õpetajate tööd. Lisaks loodi mõlema ainega seotud ülesannete ning teemade jaoks näidis Java ning tekstifaile.

Bakalaureusetöö autor soovib töö käigus valminud materjale ning ainekavasid kasutades pidada silmas kooli ning õpilaste eripärasid. “Programmeerimine keeles Java I” aine esimene kontrolltöö võib jääda paljudele õpilastele keeruliseks ning autor soovib seda vastavalt kooli tasemele kergemaks muuta. “Programmeerimine keeles Java I” teise kontrolltöö maht võib ühes tunnis lahendamiseks olla liiga suur, eriti kui kooli tunni pikkus on 45 minutit mitte 75 minutit nagu TJPGs.

Kindlasti tuleb Eesti erinevate üldhariduskoolide tarbeks luua veel erinevaid programmeerimise teemalisi õppematerjale. Bakalaureusetöö autor soovib luua gümnaasiumiastmele valikkursus “Arvutimängude loomine”, milles kasutatakse arenduskeskkonda Unity. Taoline kursus võiks mängustada õppeprotsessi ning seeläbi õppimist lihtsustada. Samuti soovib töö autor luua kursuse informaatika olümpiaadi ettevalmistuseks. Nimetatud kursus võiks olla läbitav veebipõhiselt ning see võiks põhineda informaatika olümpiaadidele sarnaselt püstitatud probleemide lahendamises. Informaatika olümpiaadi kursusel peaksid olema lubatud kõik need programmeerimiskeeled, mille kasutamine on lubatud olümpiaadil. Selline kursus võiks tekitada rohkemates noortes, sealhulgas tüdrukutes, huvi informaatika olümpiaadi vastu.

Tööl on lisad, mis on esitatud eraldi failidena.

Tänuavaldused

Ma soovin tänada kõiki, kes mu lõputöö valmimisele kaasa aitasid.

Esmajärjekorras soovin ma tänada Tartu Jaan Poska gümnaasiumi selle eest, et nad lubasid mul enda loodud materjalide ja ainekavadega läbi viia kaks reaalselt valikkursust. Ilma Tartu Jaan Poska gümnaasiumi toetuseta ei oleks saanud see lõputöö valmida.

Ma soovin tänada oma elukaaslast ja isa, kes kuulasid mu muret ning jõudsid vaatamata oma enda kohustustele lugeda mu tööd.

Ning viimaseks soovin ma tänada oma juhendajaid Anne Villemisit ning Merike Heina!

Aitäh!



Allikad

- [1] G. Cattaneo, T. Hüsing, M. Kolding, W. B. Korte, R. Lifonti, N. Fonstad (2013). E-Leadership. E-skills for Competitiveness and Innovation Vision, Roadmap and Foresight Scenarios Final Report.
<https://publications.europa.eu/en/publication-detail/-/publication/84e8d8fa-e123-4610-a207-61ab1d1627a>. (Vaadanud 17.05.2019)
- [2] Mis saab Eesti IT haridusest?, 2015,
https://sisu.ut.ee/sites/default/files/ikt/files/iktraport_31.08.2015.pdf (Vaadatud 17.05.2019)
- [3] TIOBE veebilehekül <https://www.tiobe.com/tiobe-index/> (Vaadatud 02.12.2018)
- [4] Programeerimiskeele Java veebilehekül
<https://go.java/index.html?intcmp=gojava-banner-java-com> (Vaadatud 27.10.2018)
- [5] Oracle'i Java alamleht <https://www.oracle.com/java/> (Vaadatud 03.02.2019)
- [6] Ettevõtte Playtech Estonia OÜ veebilehekül <http://playtech.ee> (Vaadatud 29.10.2018)
- [7] Ettevõtte Nortal veebilehekül <https://nortal.com/> (Vaadatud 29.10.2018)
- [8] Ettevõtte Helmes veebilehekül <https://helses.com> (Vaadatud 29.10.2018)
- [9] Konveretsi Geekout 2018 veebilehekül <https://2018.geekout.ee/> (Vaadatud 28.10.2018)
- [10] R.Juurak, Vaja on 300 uut informaatikaõpetajat, 2018
<http://opleht.ee/2018/09/vaja-on-300-uut-informaatikaopetajat/> (Vaadatud 18.05.2019)
- [11] S. Puniste, 2015, "Eesti gümnaasiumides õpetatavad programmeerimiskursused"
<https://dspace.ut.ee/bitstream/handle/10062/56092/thesis.pdf?sequence=1&isAllowed=y>
- [12] Progetiigri õppematerjalide kogumik <http://www.progetiiger.ee/?q=> (Vaadatud 18.05.2019)
- [13] Eesti Informaatikaolümpiaadi veebilehekül <http://eio.ee/> (Vaadatud 23.03.2019)
- [14] Neil C.C. Brown, Greg Wilson, 2018, „Ten quick tips for teaching programming“ in PLoS Computational Biology, Vol 14, Iss 4.
<https://doi.org/article/047b207f4b8648d194f50f28f7ff47dc> (Vaadatud 29.10.2018)

- [15] D. Clark, C MacNish, G. F. Royle, 1998, Java as a teaching language — opportunities, pitfalls and solutions
https://www.researchgate.net/profile/Cara_Macnish/publication/221222923_Java_as_a_teaching_language-opportunities_pitfalls_and_solutions/links/5649af5308ae54697fbef121/Java-as-a-teaching-language-opportunities-pitfalls-and-solutions.pdf (Vaadatud 18.05.2019)
- [16] Tartu Jaan Poska gümnaasiumi veebilehekülg <http://jpg.tartu.ee> (Vaadatud 23.03.2019)
- [17] Tartu Jaan Poska gümnaasiumi hindamisjuhend
<http://sisu.jpg.tartu.ee/files/hindamisjuhend.2018.pdf> (Vaadatud 15.05.2019)
- [18] Giti veebisait. Getting Started - About Version Control
<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control> (Vaadatud 18.05.2019)

Lisa 1 “Programmeerimine keeles Java I” kursuse ainekava

Kursuse nimi: Programmeerimine keeles Java I

Osalejate piirarv: 25

Toimumised: III periood 2018/2019. Hommikune valikaine (T, K, N 8.20–9.35)

Eeldusained: Programmeerimine keeles Python I

Kursuse sisu: “Programmeerimine keeles Java I” aines tehakse tutvust programmeerimiskeelega Java, kasutades selleks arenduskeskkonda IntelliJ. Kursuse käigus õpitakse tundma Java süntaksit, eripärasid ning võimalusi. Õpitakse kirjutama jätkusuutliku, arusaadavat ning kvaliteetset koodi. Tehakse tutvust objekt-orienteerituse mõistega. Kursuse lõpus on kõik õpilased võimelised looma ise reaalselt kasutatava rakenduse/programmi.

Kursusel läbitavad teemad on järgmised:

- Muutujate tüübid *String*, *int*, *double* ja *boolean*.
- Programmi konsooli sisend ja -väljund.
- Tingimuslaused
- Tsüklid
- Andmestruktuurid - massiiv, järjend, *HashSet* ja *HashMap*
- Meetodid ja rekursioon
- Failist lugemine ja faili kirjutamine
- Java klassid
- Graafiline kasutajaliides
- Dokumentatsioon
- Internetist informatsiooni otsimine

Õpiväljundid: Lisaks programmeerimiskeele Java tutvustamisele on kursuse eesmärgiks arendada õpilaste loovust ning loogilist, analüütilist ja algoritmilist mõtlemist ning süsteemset probleemide lahendamise oskust. Aine põhiliseks eesmärgiks on laiendada õpilaste silmaringi, mis aitab kaasa teadlike eriala valikute tegemisele.

Kursuse läbinud õpilane:

- tunneb rakenduste ja programmide loomise vahendeid ning põhimeetodeid,
- teab ja kasutab hea koodi kirjutamise põhilisi tavasid,
- õpilane orienteerub programmeerimisega seonduvas sõnavaras (muutuja, väärtus, funktsioon, tingimuslause, tsükkel, käsuvoog, objekt-orienteeritud programmeerimine),
- saab aru lihtsamast võõrast koodist, oskab sellest leida vigu ja selgitada selle tööd,
- oskab iseseisvalt koostada mõistliku raskusastmega Java programme,
- oskab Google'ist tekkinud programmeerimise alastele küsimustele vastuseid leida ning leitusse kriitiliselt suhtuda,
- on kogenud programmeerimisega seotud väljakutseid ning oskab selle põhjal teha läbimõeldud otsuseid edasiste õpingute suhtes.

Õpilased rakendavad aine jooksul omandatud oskuseid projekti loomiseks.

Hinde kujunemine:

- Iseseisvad ülesanded 25% perioodihindest
 - Õpilane peab edukalt esitama vähemalt seitsme ploki jagu iseseisvaid ülesandeid
- Moodle testid 10%
- Esimene kontrolltöö 20%
 - Kontrolltöö peab olema esitatud vähemalt positiivsele tulemusele (>50%)
- Teine kontrolltöö 20%
- Projekt 25%
- Lisapunktid maksimaalselt +10%
 - Ühest iseseisvate ülesannete plokist kõik ülesanded ära lahendades +1%
 - Kontrolltööde lisaküsimused või -ülesanded

Lisa 2 “Programmeerimine keeles Java I” aine slaidid

“Programmeerimine keeles Java I” aine sissejuhatava teema näidis slaidid on nähtavad joonisel 6 ja 7. Kõiki tunni slide saab näha ja alla laadida siit:

https://drive.google.com/open?id=1Yq-6UU27Y5JUVwyuruYan9FADv3Av_yaJsS1awVbI40.

Kui kasutate tunnis kooli arvuteid

Kasutage mõnda turvalist pilveteenust

- Dropbox <https://www.dropbox.com/>
- Google Drive <https://drive.google.com/>
- OneDrive

Või vanamoodsalt mälupulka

Saate oma tunnitööd vajadusel kodus jätkata ja kontrolltöö eel ülevaadata

Joonis 6. Näidis slaid 1 “Programmeerimine keeles Java I” sissejuhatusest

int

- Täisarv (integer)
- Ei ole lõpmatu
 - -2 147 483 648 ... 2 147 483 648 (-2^{31} ... 2^{31})
- Tehted
 - Liitmine $a+b$ • Korrutamine $a*b$
 - Lahtumine $a-b$ • Jagamine a/b
 - NB $9/2=4$
- Tehete järjekorda saame mõjutada sulgudega $(a+b)*c$
- *String*-ile saab arve juurde liita, $1+$ ja $”+3$ annab $”1$ ja $3”$

Joonis 7. Näidis slaid 2 “Programmeerimine keeles Java I” sissejuhatusest

“Programmeerimine keeles Java I” aine muutujad, loogika ja sisend teema näidis slaidid on nähtavad joonisel 8 ja 9. Kõiki tunni slide saab näha ja alla laadida siit: https://drive.google.com/open?id=1r4RIPsRXVxi2D7s_9nYuW9PPFmeAeuqLVPWRovufpZE.

Muutujad

- Nagu Pythonis
 - String str= “Pluto”;
 - Int arv=20;
- Tehted
 - Väärtustamine arv=4; str=“X”; x=y;
 - Kokku liitmine arv+b; str+b;
 - Juurde liitmine arv+=b; str+=b;
 - Lahutamine arv-=b;

Joonis 8. Näidis slaid 1 “Programmeerimine keeles Java I” “Muutujad” teemast

Loogika - Tehted

Tõeväärtustega		Arvudega		Sõnedega	
Võrdne	==	Võrdne	==	Võrdus	a.equals(b)
Ja	&&	Väiksem	<	a.equalsIgnoreCase(b)	
Või		Väiksem=:	<=		
Vastand	!	Suurem	>	Sisaldus	a.contains(b)
		Suurem=:	>=		

Joonis 9. Näidis slaid 2 “Programmeerimine keeles Java I” “Loogika tehted” teemast

“Programmeerimine keeles Java I” aine tsüklid ja Google teema näidis slaidid on nähtavad joonisel 10 ja 11. Kõiki tunni slide saab näha ja alla laadida siit: https://drive.google.com/open?id=17izPiGjR_S3vy1AFH-t31leTp0Oe7ujJgKmPSN-4IMo

Tsükel - while

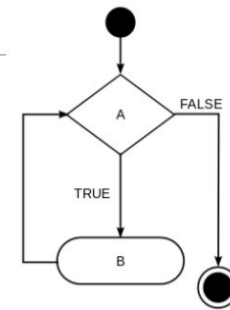
- Kordab kuni mingi tingimus on täidetud.

```
while(A){  
    B; // korratav tegevus  
}
```

- Saab teha lõpmatu tsüklit

```
while (true) {  
    System.out.println("Tere");  
}
```

•Kuna väärtus „true“ on alati tõene, siis printib lõpmatu arv kordi „Tere“ (kuni programm kinni pannakse/ arvuti suletakse vms)



While tsükel

Joonis 10. Näidis slaid 1 “Programmeerimine keeles Java I” “Tsüklid ja Google” teemast

Google

- Google on sinu sõber number 1
 - 99.9% juhtudest on juba keegi kunagi olnud sama mure otsas nagu sina
 - Eelistage inglise keelseid materjale (eesti keelseid on vähe)
 - Vahel on vaja küsimus jaotada osadeks ja guugeldada osade kaupa
 - Kui te leiate vastuse, siis ärge unustage seda kontrollimast – neid vastused on samuti kirjutanud inimesed ja inimesed võivad eksida.
- Kasulikud allikad Javale
 - stackoverflow.com (foorum, isiklikud probleemid, parimad lahendused)
 - docs.oracle.com (kirjeldused, dokumentatsioon)

Joonis 11. Näidis slaid 2 “Programmeerimine keeles Java I” “Tsüklid ja Google” teemast

“Programmeerimine keeles Java I” aine andmestruktuurid I teema näidis slaidid on nähtavad joonisel 12 ja 13. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1-SPkn3rBWncDntaFiMIYbLYnIhNKI3yaBHJur9PZH54>.

Array (Massiiv)

- Järjend
 - Kindla arvu elementidega nimekiri (esimene element asub kohal 0)
 - Primitiivne = kiire, saab luuga kõigi andmetüüpidega
- Loomine
 - `int[] numbers=new int[n];` „Tühi“ (kõiki kohti näidab „null“, 0, false...), suurus on n.
 - `int[] numbers={1,2,3,4,5};` Kindlate määratud väärtustega

Joonis 12. Näidis slaid 1 “Programmeerimine keeles Java II” “Andmestruktuurid I” teemast

List (List) – ArrayList<>

- Järjend
 - Muudetavate elementide list (0-indexed)
 - Aeglasem kui massiiv, primitiivseid andmetüüpe ei saa kasutada
- Loomine
 - `List<Integer> numbers= new ArrayList<>();` Tühi (0 elementi)
 - `List<Integer> numbers=Arrays.asList(0, 1, 2, 3, 4, 5);` Kindlate määratud väärtustega
- Listi kasutamine vajab importi (nagu *Scanner*-gi)

- `int` -> Integer
- `double` -> Double
- `boolean` -> Boolean
- jne

Joonis 13. Näidis slaid 2 “Programmeerimine keeles Java I” “Andmestruktuurid I” teemast

“Programmeerimine keeles Java I” aine meetodid ja rekursioon teema näidis slaidid on nähtavad joonisel 14 ja 15. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1eDMOARfwpPvI8KTEUNGVFtRonPgMvJzydVuxoJVIJuE>.

Meetod – Kuidas kasutada?

- Korralik näide

```
public static void main(String[] args) {
    System.out.println("Peetrile mõjuv raskusjõud on: " + leiaRaskusjõud( mass: 90) + "N");
    System.out.println("Marile mõjuv raskusjõud on: " + leiaRaskusjõud( mass: 60) + "N");
    System.out.println("Mukile mõjuv raskusjõud on: " + leiaRaskusjõud( mass: 25) + "N");
}

public static double leiaRaskusjõud(double mass) {
    return mass * 9.8;
}
```

ja tagastab double tuupi vaartusi (...static double leiaRaskusjõud...)

Joonis 14. Näidis slaid 1 “Programmeerimine keeles Java I” “Meetodid” teemast

Meetod - rekursioon

- Rekursioon – meetodi seest iseenda välja kutsumine (nagu tsükkel)
- TÄHTIS!
 - Lõpu tingimus
 - Erandjuhud, kui on (võivad kattuda lõpu tingimustega)
 - Argumentide muutus kordustel
- Tihti saab asendada tsükliga

```
public static int summaAstBni(int a, int b) {
    if (a == b) {
        return a;
    }
    return a + summaAstBni( a: a + 1, b);
}


public static int summaAstBni(int a, int b) {
    int summa = 0;
    for (int arv = a; arv <= b; arv++) {
        summa += arv;
    }
    return summa;
}
```

Joonis 15. Näidis slaid 2 “Programmeerimine keeles Java I” “Rekursioon” teemast

“Programmeerimine keeles Java I” aine failist lugemine ja faili kirjutamine teema näidis slaidid on nähtavad joonisel 16 ja 17. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1J2dGjKw7pdFgnU0a1Nm8sWql7JnZ5iAT7eQ8ReJjqPc>.

Failist lugemine - Scanner


- Vana-tuttav Scanner
- Scanner s= new Scanner (<failiasukoht.txt>);
s.nextLine(); // loeb failist terve rea
 - s.next(); // loeb failist järgmise sõna
 - s.nextInt();
 - s.nextDouble();
 - s.hasNextLine();
 - s.close(); // Scanneri sulgemiseks kui kasutamine on lõpetatud



Joonis 16. Näidis slaid 1 “Programmeerimine keeles Java I” faili operatsioonide slaididest

Faili kirjutamine- PrintWriter

- Peaaegu nagu vastand Scannerile
- Vajab importi
- PrintWriter writer = new PrintWriter (<failiasukoht.txt>, “UTF-8”);
 - writer.println(<Sisestatav tekst>);
 - writer.print (<Sisestatav tekst>);
 - writer.close(); // PrintWriteri sulgemiseks kui kasutamine on lõpetatud



Joonis 17. Näidis slaid 2 “Programmeerimine keeles Java I” faili operatsioonide slaididest

“Programmeerimine keeles Java I” aine klasside teema näidis slaidid on nähtavad joonisel 18 ja 19. Kõiki tunni slaide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1RFoYINQ5NCARogwmLVuJTJ38WstM2btuGC7U1fPQDo>

Klassid – uus klass

- Loome uue Java faili Isik ja sellega koos uue klassi Isik. Olgu sellel kaks välja

```
public class Koer {  
    String nimi; //Isendimuutuja nimi jaoks  
    double kaal; //Isendimuutuja kaal jaoks  
}
```

- Uus andmetüüp Koer. Igal loodud Koer andmetüüpi objektil saab olla nimi ja kaal.

Joonis 18. Näidis slaid 1 “Programmeerimine keeles Java I” “Klassid” temast

Klassid – uus väärtustatud objekt Isik

- Nüüd saab peaklassis isendeid luues nende isendimuutjad väärtustada

```
public class Peaklass {  
    public static void main(String[] args) {  
        Koer koerake=new Koer( nimi: "Muki", kaal: 5);  
        Koer peni=new Koer( nimi: "Triksi", kaal: 12);  
        System.out.println(koerake.nimi);  
        System.out.println(koerake.kaal);  
        System.out.println(peni.nimi);  
        System.out.println(peni.kaal);  
    }  
}
```

Joonis 19. Näidis slaid 2 “Programmeerimine keeles Java I” “Klassid” temast

“Programmeerimine keeles Java I” aine andmestruktuurid II teema näidis slaidid on nähtavad joonisel 20 ja 21. Kõiki tunni slaide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1XiHbPEBNfr6GcNEZt3e96yLBlcGSYASKVqaaPWi4DOY>.

Set (hulk) – HashSet<>

•int ->Integer
•double -> Double
•boolean -> Boolean
•jne

- Hulk
 - Järjekord ei ole oluline, kuid iga elementi saab olla vaid üks
 - Sarnane Listile
- Loomine
 - Set<Integer> numbers=new HashSet<>(); Tühi
- Seti kasutamine vajab importi (nagu Scannergi)

Joonis 20. Näidis slaid 1 “Programmeerimine keeles Java I” “Andmestruktuurid II” teemast

Teisendamine

- Array -> List** – Arrays.asList(arrayName);
- Array -> Set** – new HashSet<>(Arrays.asList(arrayName));
- List-> Array** - listName.toArray(new T[listName.size()]); //T on tüüp
- List->Set** – new HashSet<>(listName);
- Set -> Array** – setName.toArray(new T[setName.size()]); //T on tüüp
- Set -> List** – new ArrayList<>(setName);

Joonis 21. Näidis slaid 2 “Programmeerimine keeles Java I” “Andmestruktuurid II” teemast

“Programmeerimine keeles Java I” aine GUI teema näidis slaidid on nähtavad joonisel 22 ja 23. Kõiki tunni slaide saab näha ja alla laadida siit: http://drive.google.com/open?id=1HjSASY3i3iYJq-ApsQL_TEKFeosN2NvsjhIqFaIXoNc.

GUI klassid

- Kohe olemas
 - sample/controller.java ning sample/main.java
 - GUI kontrolleri ja main klass
 - Saate soovi korral lisada oma klassi täpselt nii, nagu eile klasse lõime
 - Esimene rida „package sample;“ ütleb, et fail on sample kaustas
- Lisaks on olemas sample.fxml



Joonis 22 Näidis slaid 1 “Programmeerimine keeles Java II” “GUI” teemast

FXML

- Määrab, milline näeb kasutajaliides välja
- Meie kasutame Scene Builderit – lihtne ja mugav
- GUI-d saab ka luua käsitsi koodi kirjutades, aga see on tüütu ja keeruline
- **FXML-is peab igal GUI elemendil, millel on mingi tegevus (näiteks nupul vajutus) olema määratud, millise meetodiga see kokku käib.**
- **Igal elemendil peab olema id ja see peab kattuma muutuva nimega Controlleris**
- Peab olema määratud Controller



Joonis 23. Näidis slaid 2 “Programmeerimine keeles Java II” “GUI” teemast

“Programmeerimine keeles Java I” aine dokumentatsioon teema näidis slaid on nähtav joonisel 24. Kõiki tunni slaide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1gg6kTAX6m3rFfH-KT5LZS40GAFCY16HukxZtahQuV8E>.

Schort-cuts

`sout+tab` – `System.out.println()`;

`psvm+tab` – `public static void main (String args[]){}`

Ctrl+Alt+L – formaadib koodi ilusaks

Alt+Enter – Pakub lahendusi koodi parendamiseks/parandamiseks

Ctrl+Shift+ üles/alla nooleklahv – Liigutab ridu

Ctrl+D – dubleerib rea

<https://www.jetbrains.com/help/idea/mastering-keyboard-shortcuts.html>

Joonis 24. Näidis slaid 1 “Programmeerimine keeles Java II” “Dokumentatsioon” teemast “Programmeerimine keeles Java I” aine klassid II teema näidis slaidid on nähtavad joonisel 25. Kõiki tunni slaide saab näha ja alla laadida siit: <https://drive.google.com/open?id=11CqldQY2R9eYMm8itUNoA9HjCRQ4k-LiKrMbVFSXuxo>

Klassid – Getter ja Setter

- Et saada ligi klasside private muutujatele
- Muutuja sisu kätte saamiseks kasutame getter meetodit
- Muutuja sisu määramiseks kasutame setter meetodit

ProTip: Alt+Insert

- Miks peaks üldse muutujad/meetodid private olema?
 - Et piirata ligipääsu muutujate sisu muutmisele/saamisele. Näiteks saab nii olla muutuja, mille sisu saab pärida, kuid muuta ei saa.
 - Teisendused

Getter Jaani



Setter

Joonis 25. Näidis slaid 1 “Programmeerimine keeles Java II” “Klassid II” teemast

“Programmeerimine keeles Java I” aine GUI II teema näidis slaidid on nähtavad joonisel 26. Kõiki tunni slaide saab näha ja alla laadida siit: https://drive.google.com/open?id=1NVfDE5Rwt14LfYW_Yie76HYRfi20bbkGIqYn2Ai-5tk

Joonis 26. Näidis slaid 2 “Programmeerimine keeles Java II” “GUI II” teemast

Lisa 3. “Programmeerimine keeles Java I” aine ülesanded

Kõik vajalikud näidisfailid (nii Java kui ka tekstifailid) on leitavad kaustast <https://drive.google.com/open?id=1Ct6dHjjop5oFHfr6R7-aaPGbMa1Aojbi>.

Tabel 2. “Programmeerimine keeles Java I” iseseisvad ülesanded 1

Kood	Ülesanne
a1	Koosta Java programm, mis prindib välja "Tere!" ja järgmisel real "Kuidas läheb?"
a2	Koosta Java programm, mis prindib välja 234 ja 654 summa (vastuse arvutab programm).
b1	Koosta Java programm, mis prindib välja järgmise teksti: Taandrida<taandrida>Tänases tunnis õppisin ma printima järgmist sümbolit: ”\”.
b2	Koosta programm, mis prindib välja “234+654=<vastus>, 234*654=<vastus>” (vastused arvutab programm).
c1	Koosta programm, mis prindib lause "Red fox jumps over lazy dog" ühe reana, kasutades iga sõna jaoks eraldi print käsku ning seejärel sama lause selliselt, et iga sõna on eraldi real, kuid kasutades ühte print käsku.
c2	Koosta programm, mis prindib punaselt (errorina) välja “234+654=<vastus>, 234*654=<vastus>” (vastused arvutab programm). Guugelda. Märksõnad: <i>Java, system error message, print</i> .

Tabel 3. “Programmeerimine keeles Java I” iseseisvad ülesanded 2

Kood	Ülesanne
a3	Koosta Java programm, mis küsib kasutajalt tema lemmik värvi ning seejärel tagastab “Sinu lemmikvärv on <sisestatud värv>”.
a4	Koosta Java programm, mis küsib kolmnurga külgede pikkused ning väljastab selle übermõõt (ainult arvuline tulemus).
a5	Koosta programm, mis küsib kasutajalt kontrolltöös saadud punkte (täisarvudena) ja seejärel kontrolltöö maksimum tulemust (täisarvudena). Tagasta “A”, kui õpilane saab selle kontrolltöö tulemusega hindeks A ($A \geq 91\%$) ning “Muu”, kui tulemus jääb alla A piiri (Muu < 91).
b3	Sa tahad pidada oma 3 sõbraga sünnipäeva lasketiirus, kuid selle jaoks peavad su sõbrad

	olema vähemalt 21 aastat vanad. Koosta Java programm, mis küsib kasutajalt eraldi iga sõbra vanust (täisarvuna) ning tagastab “Saab”, kui see külaline saab sünnipäevale tulla või “Ei saa”, kui see pole võimalik.
b4	Koosta Java programm, mis küsib kolmnurga kõigi kolme sisenurga suurust. Tagasta selle kolmnurga tüüp nurkade järgi (“Teravnurkne”, “Nürinurkne”, “Täisnurkne”) või “Ei ole kolmnurk”, kui sisestatud nurgad ei vasta kolmnurga sisenurkade summa reeglile.
b5	Koosta programm, mis küsib kasutajalt kontrolltöös saadud punkte ja seejärel kontrolltöö maksimum tulemust. Tagasta õpilase kontrolltöö eest saadud hinne (A $\geq 91\%$, B $\geq 81\%$, C $\geq 71\%$, D $\geq 61\%$, E $\geq 50\%$, F $< 50\%$).
c3	Koosta programm, mis arvutab sinu kaalu järgi välja minimaalse ratsahobuse soovitusliku kaalu, kellega sa sõita võiksid. Guugelda vajaliku valemi saamiseks.
c4	Koosta Java programm, mis küsib kasutajalt täisnurkse kolmnurga kahe kaateti pikkust ning tagastab kolmnurga pindala kolme komakoha täpsusega. Guugelda. Märksõnad: <i>rounding</i> .
c5	Koosta programm, mis küsib kasutajalt kuu numbrit ja tagastab vastava kuu nime. Näiteks sisestades 1 on väljund "Jaanuar", sisestades 11 on väljund "November". Programm peab kasutama if-tingimuslausete asemel switch'i. Guugelda. Märksõnad: <i>Java switch</i> .

Tabel 4. “Programmeerimine keeles Java I” iseseisvad ülesanded 3

Kood	Ülesanne
a6	Koosta Java programm, mis prindib välja arvude 1 kuni 100 summa.
a7	Leia guugeldades, kuidas lõpmatutest tsüklitest vajadusel väljuda. Loo programm, mis imiteerib Eeslit filmist “Shrek 2” (video: https://youtu.be/basofea2UEs) ja küsib “Kas me oleme juba kohal?” kuni kasutaja lõpuks vastab “jah”. Viimase asjana prindib programm välja "LÕPUKS!". Seotud märksõnad: <i>infinite, break</i> .
b6	Koosta Java programm, mis prindib välja arvud vahemikus 1...100, mis jaguvad 6-ga.
b7	Leia guugeldades kuidas lõpmatutest tsüklitest vajadusel väljuda. Loo Java programm, mis imiteerib Eeslit filmist “Shrek 2” ja küsib “Kas me oleme juba kohal?” kuni kasutaja lõpuks vastab “jah” – seejärel küsi “kas tõesti?” ning “jah” korral prindib programm välja "LÕPUKS!". Kui vastatakse "ei", siis tuleb taas küsida "Kas me oleme juba kohal?". Kui kasutaja sisestab midagi muud, kui "jah" või "ei", siis peaks programm ütlema "Palun vasta jah või ei" ning ootama uut vastust. Seotud märksõnad:

	<i>infinite, break.</i>
c6	Koostada Java programm, mis leiab järgmise summa: $\frac{1}{3} + \frac{3}{5} + \frac{5}{7} + \frac{7}{9} + \dots + \frac{95}{97} + \frac{97}{99}$.
c7	Leia guugeldades, kuidas tsüklitest väljuda. Kirjuta programm, mis prindib välja kõik algarvud kuni kasutaja poolt antud algarvuni. Eelda, et kasutaja annab algarvu = pole vaja programmeerida juhu, kus kasutaja antud arv ei ole algarv. Seotud märksõna: <i>break</i> .

Tabel 5. “Programmeerimine keeles Java I” iseseisvad ülesanded 4

Kood	Ülesanne
a8	Loo massiiv (<i>array</i>), mille pikkus on 5. Seejärel lisa sinna nimed Ahto, Ahti, Aksel, Anna, Arabella. Prindi tulemus välja.
a9	Loo ArrayList. Seejärel lisa sinna nimed Bianka, Boris, Beatrice, Bella, Ben. Prindi tulemus välja.
b8	Sulle on antud massiiv {89,18,7,-5,22,5,21,56,-78,99}. Prindi välja 5 kõige suuremat arvu suuruse järjekorras (alustades väiksemast).
b9	Loo ArrayList. Lisa sinna kõik arvud vahemikus 1..100. Seejärel eemalda need arvud, mis jaguvad seitsmega või kolmega. Prindi tulemus välja.
c8	Sul on pikk String loomanimedega “Kass,Koer,Lammas,Hobune,Elevant,Ahv,Tiiger,Lehm,Okassiga,Rott”. Kirjuta programm, mis võtab selle sõne, eraldab loomad massiivi.Prindi välja massiv ise ja seejärel sellest massiivist vaid need loomad, mille nimes on a-täht.
c9	Loo ArrayList. Küsi kasutajalt inimeste nimesid. Lõpeta nimede küsimine siis, kui kasutaja sisestab "Lõpp". Lisa kõik nimed järjendisse (<i>List</i>). Sorteeri saadud järjend tähestikulises järjekorras (alustades A-st) ning prindi tulemus välja.

Tabel 6. “Programmeerimine keeles Java I” iseseisvad ülesanded 5

Kood	Ülesanne
a10	Kirjuta programm, mis küsib kasutajalt eesnime ja perenime ning kasutades meetodit prindib inimesele tervitussõnumi: “Tere <eesnimi> <perenimi>!” (kõik String) meetod: <code>looTervitus(eesnimi, perenimi)</code> .
a11	Kirjuta programm, mis küsib kasutajalt kuu numbri ja prindib arvule vastava kuu (nt

	5→Mai) kasutades selleks meetodit: kuuNumbriJärgi(kuunumber). Väljasta “-”, kui kuud ei leidu. Meetodi töö tulemus printitakse peameetodis välja.
b10	Kirjuta programm, mis küsib kasutajalt kera raadiust ning prindib übermõõdu, pindala ja ruumala, kasutades selleks kolme meetodit (kõik double; tagastavad väärtuse): übermõõt(raadius), pindala(raadius), ruumala(raadius). Ümardada kolme komakohani.
b11	Kirjuta programm, mis küsib kasutajalt sõna ning seejärel prindib selle tagurpidi välja, kasutades sõna tagurpidi pööramiseks meetodit (kõik String; tagastab String): tagurpidi(sõna). (Näiteks puu -> uup, kruus -> suurk, kirik -> kirik). Lisaks prindib see välja "True" kui tegu on palindroomiga ja "False", kui ei ole.
c10	Kirjuta programm, mis küsib kasutajalt arvu n ning prindib fibonacci jada n-inda elemendi, kasutades selleks rekursiivset meetodit: fibonacci(n).
c11	Kirjuta programm, mis küsib kasutajalt inimeste nimesid ja pikkuseid (cm) ning kasutades eraldi meetodit lisab HashMapi info, kas inimene saab sõita ameerika mägedel, mille pikkuse piirang on 160 cm. Pikkuste küsimine lõppeb, kui sisestatakse nimeks "Lõpp". Lõpuks prindi HashMapi sisu ükshaaval välja (Kujul "<Nimi>=<väärtus>"). (HashMapis peaks olema umbes sellised paarid (Peeter=Jah, Anni=Ei), kui sisestati Peeter 180 ja Anni 156.

Tabel 7. “Programmeerimine keeles Java I” iseseisvad ülesanded 6

Kood	Ülesanne
a12	Kirjuta Java programm, mis loeb failist loomadeNimed.txt kõik loomade nimed ning prindib need ühekaupa lugemise järjekorras välja. Kokkukäivad näidisfailid: loomadeNimed.txt.
a13	Kirjuta Java programm, mis küsib kasutajalt loomade nimesid kuni kasutaja sisestab "Lõpp". Salvesta nimed sisestamise järjekorras faili väljund.txt.
b12	Kirjuta Java programm, mis loeb failist aastaarvud.txt aastaarve ning salvestab need kasvavas järjekorras uude faili korrastatudAastaarvud.txt. Kokkukäivad näidisfailid: korrastatudAastaarvud.txt.
b13	Failis loomad.txt on järjest sisestatud loomade andmed (faili näidis ülesande lõpus). Kõigi loomade andmed on failis samas järjekorras ja samal kujul. Kahe looma andmete vahel on rida, mille sisuks on "---". Loe loomade andmed sisse ning loo igale loomale tema nimeline fail <loom>.txt, mille sisuks on loomad.txt failist saadud andmed. Kokkukäivad näidisfailid: loomad.txt, merisiga.txt, halljänes.txt.
c12	Sulle on antud fail sündmused.txt. Selles failis on ajaloo sündmused juhuslikus

	järjekorras. Loe andmed sisse ning salvesta need ajalises kasvavas järjekorras faili järjekorrasSündmused.txt. Enne Kristust aasta arvud on märgitud miinusemärgiga. Kokkukäivad näidisfailid: järjekorrasSündmused.txt.
c13	Eeldame, et meil on kaust "loomad", kuhu on salvestatud loomade andmetega tekstifailid. Loe kaustast "loomad", kõikidest tekstifailidest loomade andmed. Salvesta saadud loomade info uude tekstifaili loomad.txt. Faili ülesehitus on samasugune nagu ülesandes b13. Kokkukäivad näidisfailid: loomad.txt, merisiga.txt, halljänes.txt.

Tabel 8. "Programmeerimine keeles Java I" iseseisvad ülesanded 7

Kood	Ülesanne
a14	Loo klass Lemmikloom. Klassis Lemmikloom on kolm isendimuutujad String nimi, List<String> mänguasjad ja int vanus. Isendimuutujad väärtustatakse konstruktoris. Loo ka testimise jaoks peaklass, kus lood kaks lemmiklooma. Ühe nimi on "Muki". Muki on 7 aastane ning tema mänguasjadeks on pall, lendav taldrik ja närimiskont. Teise lemmiklooma andmed mõtle ise välja.
a15	Loo klass Raamat. Klassis Raamat on kolm isendimuutujad String pealkiri, String autor ja double hind. Isendimuutujad väärtustatakse konstruktoris. Loo meetod allahindlus(int protsent), mis võtab sisendina protsendi täisarvudes (nt 50% on 50) ja tagastab raamatu hinna peale allahindlust antud protsendiga (nt hind on 20, allahindlus protsent on 40, siis programm tagastab 12). Loo ka testimise jaoks peaklass, kus lood kaks raamatut (nimi ja autor pole olulised, hinnaks on ühel 20 ja teisel 26) ning seejärel printid välja mõlema hinna peale allahindlust protsendiga 35.
b14	Failis "elevandid.txt" on järjest sisestatud loomaaias elavate elevantide andmed. Kõigi elevantide andmed on failis samas järjekorras ja samal kujul. Kahe looma andmete vahel on rida, mille sisuks on "---". <ul style="list-style-type: none"> • Loo klass Elevant. Elevantil peaks olema isendimuutujad nimi, sünniaasta, kaal ja londipikkuse jaoks. Klassis peaks lisaks muule vajalikule olema ka meetod vanus. See meetod ei võta sisse ühtegi argumenti ning tagastab Elevanti vanuse. • Loo peaklass. Peaklassis loe failist elevandid.txt sisse elevantide andmed, loo iga elevanti kohta eraldi Objekt ning lisa need järjendisse (<i>List</i>) elevandid. Lõpuks printi välja iga elevanti nimi.
b15	Loo klass Kallur. Kallur klassi loo isendimuutujad juhiNimi (andmetüüp String) ja laadungiKaal (andmetüüp double) ning meetod tühjenda(), mis tagastab isendimuutujast laadungi kaalu ning tühjendab selle (laadung = 0). Muutujate väärtused sea konstruktoris. Loo ka peaklass, milles demonstreerid põhjalikult klassi Kallur kasutamist. (automaattest puudub).
c14	Loo klass KeraMatemaatika. Klassis pole ühtegi isendimuutujat, kuid see-eest on viis

	<p>staatilist meetodit: diameeter(int r), übermõõt(int r), läbilõikePindala(int r), täisPindala(int r), ruumala(int r). Läbilõike pindala tuleks arvutada läbi keskpunkti ($r^2 \cdot \pi$).</p> <p>Loo peaklass c16, mis küsib kasutajalt raadiuse väärtust ning seejärel kutsub ükshaaval välja kõik eelnevalt loodud meetodid klassist KeraMatemaatika ning prindib välja nende tulemused. Peaklassis ümarda saadud väärtused 3 komakohani.</p>
c15	<p>Loo klassid Kallur, Isik ja Laadung. Kallur klassi loo isendimuutujad juht (andmetüüp Isik) ja laadung (andmetüüp Laadung) ning meetod tühjenda(), mis tagastab isendimuutujast laadungi ning tühjendab selle isendimuutuja (laadung = null). Muutujate väärtused sea konstruktoris. Klass isik peaks kirjeldama isiku eesnime, perenime ja vanuse. Klass laadung peaks kirjeldama laadungi nimetuse ja koguse. Vali tingimuste täitmiseks sobivad vahendid. Loo ka peaklass, milles demonstreerid põhjalikult klassi Kallur kasutamist.</p>

Tabel 9. “Programmeerimine keeles Java I” iseseisvad ülesanded 8

Kood	Ülesanne
a16	<p>Küsi kasutajalt viie(5) inimese kohta nende 1. nime; 2. pikkust. Lisa iga saadud tulemus HashMapi nii, et inimese nimi on võti ja sellele tema pikkus. Seejärel küsi kasutajalt, millise inimese andmeid ta soovib kätte saada ning seejärel prindi need välja formaadis "Nimi: <nimi>, pikkus: <kehamassiindeks>".</p>
a17	<p>On antud järjend nimed, mille sisu on {"Bianka", "Andrus", "Robert", "Lisanne", "Jelena"} ja massiiv kaalud, mille sisu on {55,85,70,48,59}. Ühenda need kaks andmestruktuuri kokku üheks HashMap'iks nii, et nimele nimed.get(i) vastab kaal kaalud[i] (Ehk näiteks esimesele nimele järjendis vastab esimene kaal massiivis) ning nimi on HashMapi võtmeks. Tulemiseks on inimeste nimed, millele vastavad nende kaalud. Prindi välja Bianka ja Roberti kaal.</p>
b16	<p>Küsi kasutajalt 1. inimese nime; 2. tema pikkust; 3. tema kehakaalu kuni kasutaja sisestab nimeks "Lõpp". Lisa iga saadud tulemus HashMapi nii, et inimese nimi on võti ja sellele tema kehamassiindeks (ümarda 3 komakohani). Kui kasutaja on sisestanud "Lõpp", küsi kasutajalt, millise inimese andmeid ta soovib kätte saada ning seejärel prindi need välja formaadis "Nimi: <nimi>, kehamassiindeks: <kehamassiindeks>". Kui küsitud inimese andmeid mapis ei ole, prindi "Sellist inimest ei leitud". Programm lõpetab töö, kui inimese andmete kättesaamisel sisestatakse nimena "Lõpp".</p>
b17	<p>Küsi kasutajalt isikukoodi, kuni kasutaja sisestab 0. Lisa isikukoodid mõnda eile ja täna õpitud neljast andmestruktuurist (massiiv (array), ArrayList, HashSet, HashMap) selliselt, et igat isikukoodi oleks ainult üks kord. Prindi andmestruktuuri elemendid ükshaaval välja. Et saada ülesanne arvestatud, pead sa valima andmestruktuuri, mille</p>

	omadused kattuvad kõige enam antud ülesandega.
c16	Küsi kasutajalt 1. inimese nime; 2. tema pikkust ; 3. tema kehakaalu kuni kasutaja sisestab nimeks "Lõpp". Lisa iga saadud tulemus HashMapi nii, et inimese nimi on võti ja sellele vastab kolm väärtust (tema pikkus, kaal ja kehamassiindeks (ümarda 3 komakohani)). Kui kasutaja on sisestanud "Lõpp", küsi kasutajalt, millise inimese andmeid ta soovib kätte saada ning seejärel prindi need välja formaadis "Nimi: <nimi>, pikkus: <pikkus>cm, kaal: <kaal>kg, kehamassiindeks: <kehamassiindeks>". Kui küsitud inimese andmeid mapis ei ole, prindi "Sellist inimest ei leitud". Programm lõpetab töö, kui inimese andmete kättesaamisel sisestatakse nimena "Lõpp".
c17	Küsi kasutajalt URL aadresse, kuni kasutaja sisestab "Lõpp". Lisa URL aadressid HashSeti nii, et igat külastatud lehekülge oleks Setis ainult üks kord, sõltumata sellest, mis on URL-i algus (näiteks www.postimees.ee, https://www.postimees.ee/ ja http://www.postimees.ee/ oleksid samad, ehk siis ei korduks). Prindi kogu saadud Set välja.

Tabel 10. “Programmeerimine keeles Java I” iseseisvad ülesanded 9

Kood	Ülesanne
a18	Loo kasutajaliidesega programm, kus kasutaja sisestab oma eesnime, oma perenimi ning valib radio-button tüüpi elemendiga, kas ta on mees või naine. Peale nupu "Kinnita" vajutamist kuvatakse GUI alumisse serva "Tere tulemast <eesnimi> <perenimi>! (<N/M>)".
b18	Lahenda mõni varasem kasutajalt sisendi küsimise b (keskmise) või c (raskem) taseme ülesanne kasutades graafilist kasutajaliidest. Ülesande ja selle ülesehituse võid ise valida.
c18	Looge kaks-kolm halba moodust, kuidas kasutajaliideses oma telefoninumbrit sisestada. Inspiratsiooniks https://qz.com/679782/programmers-imagine-the-most-ridiculous-ways-to-input-a-phone-number/ . Võite kasutada lingist saadud ideid või mõelda välja mõne enda jabura lahenduse telefoninumbri sisestamiseks.

Tabel 11. “Programmeerimine keeles Java I” teine kontrolltöö

Kood	Ülesanne
kt1	On antud failid autod.txt ja omanikud.txt. Failis autod.txt on kirjas iga auto kohta selle numbrimärk, mark ning väljalaske aasta. Failis omanikud.txt on kirjas omaniku nimi

	<p>ning tema auto number. Sõidukeid eraldab omavahel "---" ja omanike eraldab omavahel "---". Ülesandeks on kirjutada Java programm, mis loob faili output.txt, kus on kirjas omaniku nimi ja temale vastava auto mark. Fail on järjestatud samamoodi nagu omanikud.txt fail (ehk omanik failis omanikud kohal n on output.txt failis kohal n).</p>
kt2	<p>Jätk kt1 ülesandele. Loo klass Auto. Klassis auto peab olema isikuväli auto numbrile (String), auto margile (String), väljalaske aasta (int) ning omaniku nimi (String). Loo klassile auto konstruktor, mis sisaldab kõiki isendi väljasid. Loo meetod vanus(), mis ei võta sisse ühtegi argumenti ning tagastab täisarvuna (int) auto vanuse 2018 aasta seisuga. Lisaks loo lihtne testKlass, kus näitad klassi Auto toimimist (võib ka kasutada kt1 ülesandes loodud klassi).</p> <p>Lisaülesanded</p> <ol style="list-style-type: none"> 1. Loo meetod vanusAastal, mis võtab argumendina sisse aastaarvu (int) ning tagastav auto vanuse sellel antud aastal. 2. Loo lisaks teine konstruktor, mis sisaldab auto numbrit, auto margi, väljalaske aastat (EI SISALDA omaniku nime). Loo juurde <i>set</i> meetod omaniku nime määramise jaoks. 3. Loo meetod ülevaatus, mis auto numbrimärgi järgi tagastab ülevaatusse kuu (soovitus: kasuta Switchi). Vihje: http://www.uemoisaulevaatus.ee/pdf-id/ylevaatus_teade.pdf

Lisa 4 “Programmeerimine keeles Java I” aine Moodle testid

“Programmeerimine keeles Java I” moodle testid on kättesaadavad sellelt lingilt:

<https://drive.google.com/open?id=1rnPf3ynrBXLilxOExQd5ugx9v7WgJN6u>.

Lisa 5. “Programmeerimine keeles Java I” aine kirjalik kontrolltöö

Kontrolltöö 1 A

Kontrolltöö koosneb kahest kohustuslikust osast ja ühest (vabatahtlikust) lisaküsimuste osast. Materjale, sõbra abi, (tahvel-)arvuteid, telefone, nutikellasid jne kasutada ei tohi.

PALUN KIRJUTA LOETAVA KÄEKIRJAGA. Loe ülesande kirjeldus rahulikult läbi.

Esimeses osas kontrollitakse sinu elementaarseid teadmisi Java keelest. Küsimused on sarnased Moodle ja klikkeri küsimustele. Vastama pead konkreetselt ja pigem lühidalt. **Kui ei saa aru küsimuse püstitusest, siis palun küsige kohe.**

Teine osa jaguneb pooleks. Siin kontrollitakse teie arusaamist Java koodist. Esimeses pooles näidatakse programmi ja on vaja märgistada vigu koodis (nagu esimeses Moodle testis), mis muidu koodi funktsionaalsust rikuksid (kui näed viga, aga ei ole kindel, kas mõtlesin seda, siis mõtlesin küll). Lisaks tuleb kirjutada käsitsi väike jupp Java koodi. **Kui ei saa aru, mida kood suures plaanis tegema peab, siis küsige.**

Kolmas osa koosneb vaid lisaküsimustest ehk seda ei ole vaja lahendada.

Esimene osa

1. (0p) Kirjuta töö igale lehele oma nimi.
2. (1p) Millal on hea kasutada ujukomaarve (double) ja millal on hea kasutada int arve Javas? Too välja nende kahe andmetüübi peamine vahe ja näide nende kasutusest.
3. (1p) Mida teeb ‘%’ operaator (tehe) Java koodis ja millal on eriti kasulik seda kasutada (too näide)?
4. (1p) Miks kasutatakse Scanneri meetodit nimega hasNextInt()? Too näide, kuidas.
5. (2p) Kirjelda, mis on erinevat (2) ja mis on sarnast (2) järjendil (List) ja massiivil (array).
6. (1p) Mida sa peaksid kirjutama järgneva meetodisse T1, T2 ja T3 asemel, et meetod tagastaks mingi sõna kui argumentideks on kaks arvu?

```
public static T1 methodName(T2 argument1, T3 argument2) {  
    /* Some code */  
    return something;  
}
```
7. (1p) Mida kirjutad meetodi sulgude sisse (ehk eelmise ülesande näitel *T2 argument1*, *T3 argument2* asemel) kui tahad, et meetod töötaks ilma argumentideta?
8. (2p) Kirjuta kaks varianti for-tsüklist millega saaks välja printida mingi massiivi kõik elemendid. Lisa näide while-tsükli kasutusest. Miks kasutatakse vahel üht, vahel teist ning vahel kolmandat varianti?
9. (1p) Nimeta kaks peamist meetodit, kuidas luua mingit faili-tüüpi objekti (faili asukohta kirjeldavat objekti) Javas, kus saad kasutada tekstifaili nime “tekst.txt”.

Teine osa

1. (2p) Leia koodist (pildil on kogu java faili sisu) vähemalt 3 viga. Tõmba neile ring ümber ning kirjuta lühidalt, mis on valesti.

```
public class TestiNäide {  
    public static main(String[] args) {  
        System.out.println("Mis on sinu nimi?")  
        Scanner scanner1= new Scanner(System.in);  
        String nimi=scanner.nextLine();  
        System.out.println("Mis on sinu vanus?");  
        int vanus=scanner.nextInt();  
        System.out.println("Tere "+nimi+" . Sinu vanus on "+vanus+".");  
    }  
}
```

1. (3p) Koodist arusaamine (kood järgmisel leheküljel)
 - a. Leia koodist vähemalt 5 viga. Tõmba neile ring ümber ning kirjuta lühidalt juurde, mis on valesti. Vigadena lähevad arvesse nii süntaksivead (ehk mis ei laseks koodi kompileerida) kui ka programmi kirjutamise loogikavead.
 - b. Kirjelda lühidalt, mida selline programm tegema peaks.

```
public class Näidis {  
    public static void main(String[] args) {  
        Scanner scanner1 = new Scanner(System.out);  
        List<double> vanused = new ArrayList<>();  
        while (true) {  
            System.out.println("Kas soovid sisestada veel ühe vanuse?");  
            if (scanner1.hasNext()) {  
                String vastus = scanner1.next()  
                if (vastus=="jah") {  
                    vanused.add(küsiVanus(scanner1));  
                } else if (vastus=="ei") {  
                    break;  
                } else  
                System.in.println("Palun sisesta \"Jah\" või \"Ei\".");  
            }  
            System.out.println(vanused);  
        }  
        System.out.println(LeiaKeskmine(vanused));  
    }  
}
```

```
public static String küsiVanus(Scanner scanner) {  
    double vanus = 0;  
    System.out.println('Sisesta vanus');  
    if (scanner.hasNext()) {
```

```

        vanus = scanner.nextDouble();
    }
    return vanus;
}

private static double leiaKeskmine(List<Double> järjend) {
    double sum = 0;
    for (double arv : järjend) {
        sum += arv;
    }
    double keskmine = sum / järjend.size();
    return keskmine;
}
}

```

3. (5p) Loo ArrayList, kuhu saab lisada täisarve. Lisa sinna arvud vahemikust 1...100 (kasvavalt), mis jaguvad 6-ga. (Tulemuse alguse näide: 6, 12, 18, 24,).

```

public class Kontrolltöö {
    public static void main(String[] args){

```

Kolmas osa

1. (+1p) Kirjuta rekursiivne meetod fibonacci jada arvutamiseks. (jada algab 0,1,1,2,3,5,...)
2. (+0.5p) Mida teeb alltoodud kood?

```

import java.io.*;
import java.util.*;

public class KontrolltööNäide {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner scanner = new Scanner(new File( pathname: "loomad.txt"));
        List<String> list = new ArrayList<>();
        while (scanner.hasNextLine()) {
            list.add(scanner.nextLine().split( regex ":" )[1].trim().toLowerCase() + ".txt");
        }
        System.out.println(list);
    }
}

```

(+0.5p) Kui lisada peale while tsüklit Mida see väljastab, kui faili loomad.txt sisu on:

Loom: Jänes
 Loom: koer
 Loom: REBANE
 Loom: rabakull
 Loom: lEhm

3. (+1p) Milleks kasutatakse Switchi?

Lisa 6. “Programmeerimine keeles Java II” ainekava

Kursuse nimi: Programmeerimine keeles Java II

Osalejate piirarv: 25

Toimumised: IV periood 2018/2019. Hommikune valikaine (T, K, N 8.20–9.35)

Eeldusained: Programmeerimine keeles Java I

Kursuse sisu: “Programmeerimine keeles Java II” kursus jätkab sealt, kus “Programmeerimine keeles Java I” lõppes. Selle kursuse põhirõhk on omandatud teadmiste kinnistamisel ning eriolukordades rakendamisel meeskonnatööna loodud projektis. Räägitakse rohkem sellest, mis on objekt-orienteeritus.

Kursuse jooksul käsitletavat teemasid:

- Versioonihaldussüsteem Git
- Erindid
- Abstraktsed klassid
- Java 8 andmevood ja lambdad
- Algoritmid
- Graafilise kasutajaliidese head tavad
- Tarkvara ettevõtted ja töökogemus

Õpiväljundid: Kursuse eesmärk on laiendada õpilaste silmaringi, et aidata neil teha läbimõeldud eriala valik, ning arendada nende teadmisi ja oskusi programmeerimiskeeles Java. Õpilased arendavad aine käigus pidevalt digipädevusi, kuna aine läbivaks teemaks on internetist vajaliku informatsiooni leidmine ning leitusse kriitiliselt suhtumine. Kursuse tulemusena loob iga õpilane meeskonnas neid huvitavaid teemasid rakenduse/programmi (projekt).

Kursuse läbinud õpilane:

- tunneb erinevaid rakenduste ja programmide loomise vahendeid ning põhimeetodeid;
- teab ja kasutab hea koodi kirjutamise enamlevinud tavasid;

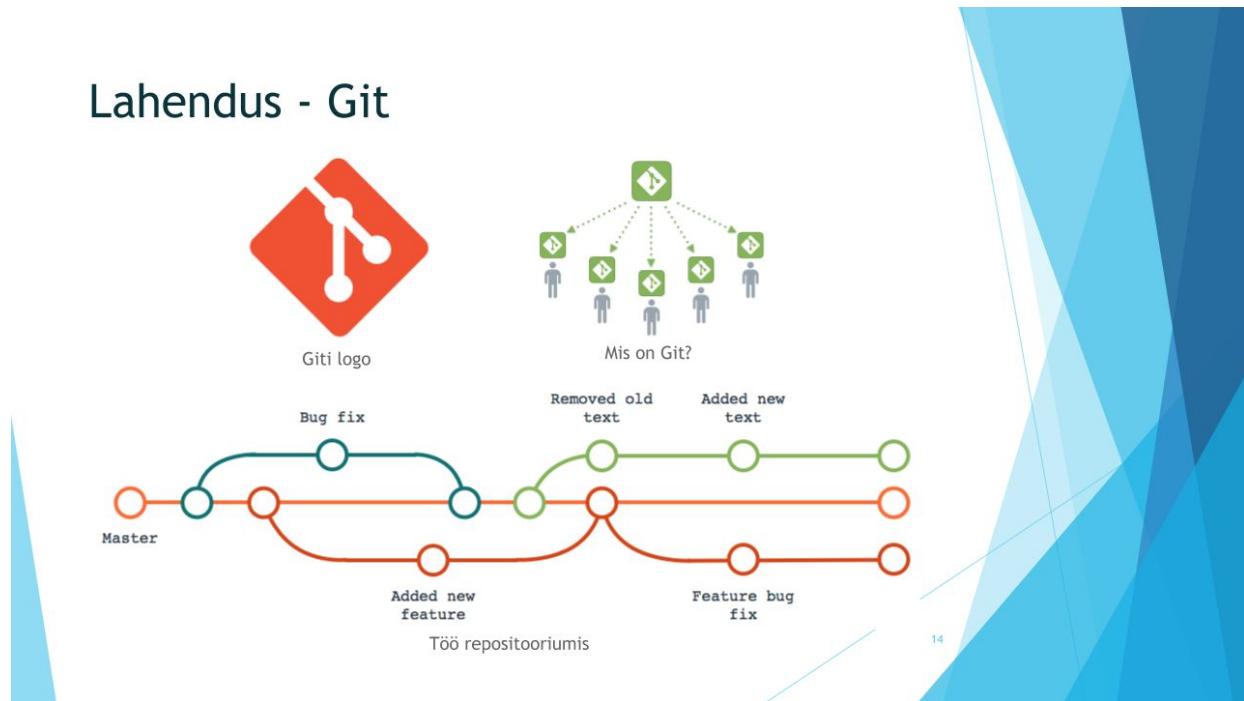
- õpilane orienteerub programmeerimisega seonduvas sõnavaras (muutuja, väärtus, funktsioon, tingimuslause, tsükkel, andmevoog, objektorienteeritud programmeerimine);
- saab aru võõrast koodist, oskab sellest leida vigu, selgitada selle tööd ning täiendada seda vastavalt vajadusele;
- oskab iseseisvalt koostada pigem keerulisema raskusastmega Java programme;
- oskab leida internetist vastuseid tekkinud teemakohastele küsimustele ning suudab leitud kriitiliselt suhtuda.

Hinde kujunemine:

- Iseseisvad ülesanded 20% perioodihindest
- Moodle testid 10%
- Esimene kontrolltöö 20%
 - Kontrolltöö peab olema esitatud vähemalt positiivsele tulemusele (>50%)
- Projekt 55%
- Lisapunktid maksimaalselt +10%

Lisa 7 “Programmeerimine keeles Java II” aine slaidid

“Programmeerimine keeles Java II” aine sissejuhatava ning Giti tutvustava tunni näidis slaidid on nähtavad joonisel 27. Kõiki tunni slide saab näha ja alla laadida siit: https://drive.google.com/open?id=1K_Fj6uM1JaOBY3gE4L5Of3fLjvlfQEZH7AW7TKuJyNM.



Joonis 27. Näidis slaid 1 “Programmeerimine keeles Java II” “Sissejuhatus ja Git” teemast

“Programmeerimine keeles Java II” aine kordamise teema näidis slaidid on nähtavad joonisel 28 ja 29. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1heV4UrVmaNUm0Q3URCUUOpPXYiQA1SWwDggaGDnBiGI>.

Meetodid

meetodi ligipääsu piiritleja - kust saab meetodile ligi
tagastustüüp (return type) - andmetüüp, mis meetodi lõpetamisel tagastatakse
meetodinimi - nimi, millega meetodit välja kutsuda
parameetrid - meetodi sisendid
meetodi keha - meetodi kood

```
public double max(double arv1, double arv2) {  
    if (arv1 >= arv2) {  
        return arv1;  
    } else {  
        return arv2;  
    }  
}
```

Joonis 28. Näidis slaid 1 “Programmeerimine keeles Java II” kordamise temast

Klassid

Klassi ligipääsu piiritleja
Klassinimi
Isendimuutujad
Konstruktor
Setter
Getterid
Klassi meetod

```
public class Kass {  
    private String nimi;  
    private int vanus ;  
  
    public Kass(String nimi, int vanus) {  
        this.nimi = nimi;  
        this.vanus = vanus;  
    }  
    public void setNimi(String nimi) {  
        this.nimi = nimi;  
    }  
    public String getNimi() {  
        return nimi;  
    }  
    public int getVanus() {  
        return vanus;  
    }  
  
    public void räagi() {  
        System.out.println("Mjäu!");  
    }  
}
```

Joonis 29. Näidis slaid 2 “Programmeerimine keeles Java II” kordamise temast

“Programmeerimine keeles Java II” aine Streamide ja Lambdade teema näidis slaidid on nähtavad joonisel 30 ja 31. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=11Yt-andxw-AMJjnADcH2bi8DE0OHS0BeV4tovVXuDc>.

(Java 8+) Streamid

- ▶ “Laisad vood”
- ▶ Asendavad tsükleid (tüüpjuhte)
- ▶ Mis mõttes asendava tsükleid?
 - ▶ `List<Isik> isikud = ...;`
`isikud.stream()`
 - `.filter(isik -> isik.vanus >= 18)`
 - `.map(isik -> isik.nimi)`
 - `.forEach(System.out::println);` ← (method reference)
- ▶ Mida see teeb?
 - ▶ Trükime välja täisealiste isikute nimed.

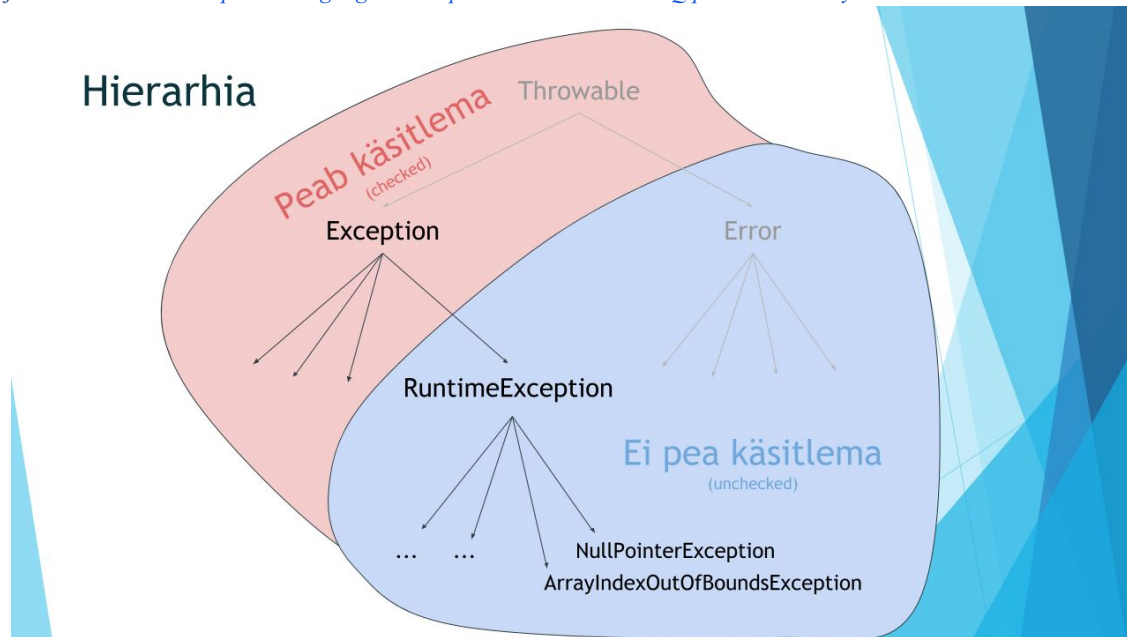
Joonis 30. Näidis slaid 1 “Programmeerimine keeles Java II” “Streamid” teemast

Method Reference

- ▶ Kuhu sobib Lambda, sobib ka Method Reference asemele:
- ▶ Viide meetodile:
 - ▶ **Meetod klassis *Mate*:**
`public static int neljandasAstmes(int arv) {`
 - `int ruudus = arv * arv;`
 - `return ruudus * ruudus;``}`
 - ▶ **Viide sellele**
`Mate::neljandasAstmes`
 - ▶ eesti keeles: meetod `neljandasAstmes` klassis `Mate`

Joonis 31. Näidis slaid 2 “Programmeerimine keeles Java II” “Lambdad” teemast

“Programmeerimine keeles Java II” aine erindite teema näidis slaidid nähtavad joonisel 32 ja 33. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1uT5z1XtYTYiQqEA0J6kkzxoSAyE41E4SEz5L1bXaGCc>.



Joonis 32. Näidis slaid 1 “Programmeerimine keeles Java II” “Erindi” teemast

Käsitlemine - try-catch

► try-catch võimaldab käsitleda

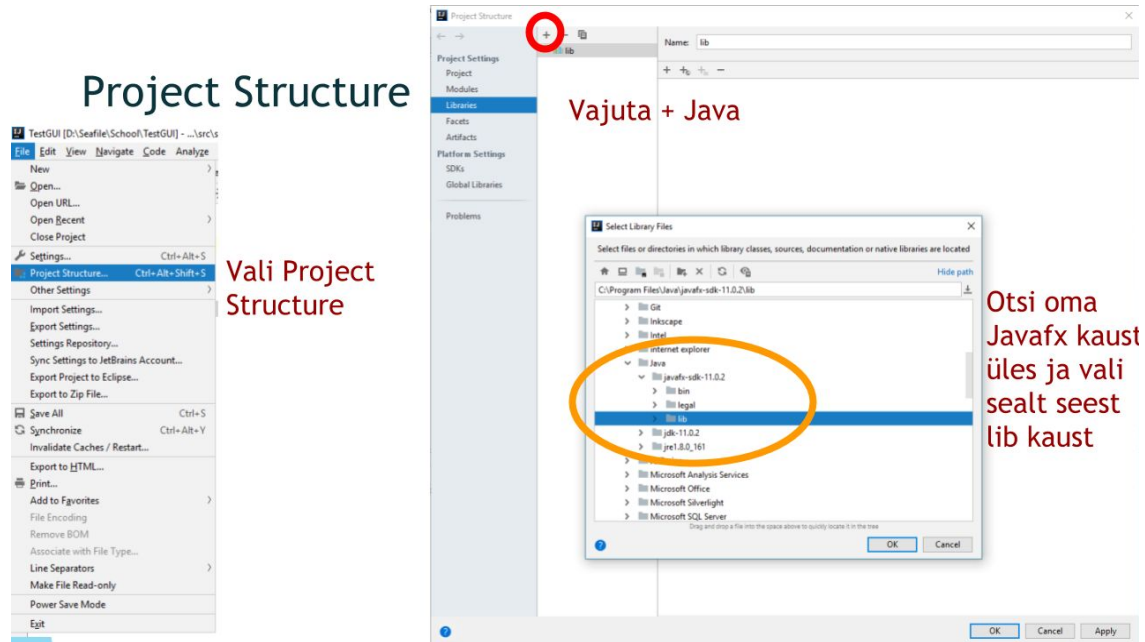
```
String str = "10";
try {
    int arv = Integer.parseInt(str);
    int tulemus = 100 / arv;
    System.out.println("100 jagatud " + arv + "-ga on " + tulemus);
} catch (NumberFormatException e) {
    System.out.println("String ei ole legit arv");
} catch (ArithmeticException e) {
    System.out.println("Jagad nulliga?");
    e.printStackTrace();
}
```

Mis juhtub kui str = “kümme”? str = “0”?

```
str = "10" → 100 jagatud 10-ga on 10
str = "kümme" → String ei ole legit arv
str = "0" → Jagad nulliga?
java.lang.ArithmeticException: / by zero
at näide.main(näide.java:7)
```

Joonis 33. Näidis slaid 2 “Programmeerimine keeles Java II” “Erind” teemast

“Programmeerimine keeles Java II” aine graafilise kasutajaliidese teema näidis slaidid nähtavad joonisel 34 ja 35. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=19zVdl8HtqU0bvaCUL4haC63I5QjfNJBbNR7K56jyc>.



Joonis 34. Näidis slaid 1 “Programmeerimine keeles Java II” “GUI head tavad” teemast



Joonis 35. Näidis slaid 2 “Programmeerimine keeles Java II” “GUI head tavad” teemast

“Programmeerimine keeles Java II” aine algoritmide teema näidis slaidid nähtavad joonisel 36 ja 37. Kõiki tunni slide saab näha ja alla laadida siit: https://drive.google.com/open?id=1yVnxsyKLuLeAHfNobGB_Rr1hUyIytu-gpDPWp7_MGLY.

Parem programmi töökiirus

- ▶ Lineaarne paremus - nt loeb failist alati 2x kiiremini.
 - ▶ Tehted paremini optimeeritud
 - ▶ Ei tee üleliigseid operatsioone (nt ei logi)
 - ▶ Aegluse kompenseerimiseks aitab kiirem arvuti
- ▶ Eksponentsiaalne paremus - nt 100kB töötleb sama kiiresti, aga 1MB 10x kiiremini.
 - ▶ Parema algoritmi
 - ▶ Mingi nutikas nipp
 - ▶ Võib kasutada rohkem mälu
 - ▶ Aegluse kompenseerimiseks ei piisa üksnes kiiremast arvutist - tuleb algoritmi vahetada

Joonis 36. Näidis slaid 1 “Programmeerimine keeles Java II” “Algoritmid” teemast

Kahendotsing

```
int[] algarvud = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47};
```

Leida, mitmes algarv on 29.

- ▶ 2? ei. 3? ei. 5? ei... 23? ei. 29? Jah! - Selge, üheteistkümnend. (10 võrdlust) - Naiivne otsimisalgoritm
- ▶ 19? ei, aga suurem. 37? ei, aga väiksem. 29? Jah! - Selge, üheteistkümnend. (3 võrdlust) - Binaarne otsimisalgoritm

Joonis 37. Näidis slaid 2 “Programmeerimine keeles Java II” “Algoritmid” teemast

“Programmeerimine keeles Java II” aine tarkvaraettevõtte teema tunni näidis slaidid on nähtavad joonisel 38 ja 39. Kõiki tunni slide saab näha ja alla laadida siit: <https://drive.google.com/open?id=1vUnk1UXeddlI-mIJ4v4xWh4bLn2CbOlq94MZr8AKEM8>

Rollid IT ettevõttes

- | Tehniline personal | Äriline personal | Tugiüksused |
|---|---|--|
| <ul style="list-style-type: none">▶ Arendaja▶ Analüütik▶ QA ehk testija▶ DevOps▶ Süsteemiadmin▶ Team lead, arhitekt▶ Projektijuht▶ Scrum master▶ UI & UX disainer | <ul style="list-style-type: none">▶ Analüütik▶ Tootejuht▶ Delivery manager▶ Ärijuht▶ Account manager▶ Jurist | <ul style="list-style-type: none">▶ HR (Human Resources)▶ Värbajad▶ Kontorihaldur▶ Korraldajad▶ jne▶ Helpdesk▶ Küberturve▶ Raamatupidajad |

Joonis 38 Näidis slaid 1 “Programmeerimine keeles Java II” tarkvara ettevõtete teemast

Suvepraktikatest

- ▶ Enamasti tasustatud
- ▶ Kandideerimiseks tuleb:
 - ▶ Esitada oma CV ja motivatsioonikiri
 - ▶ Enamasti teha test
 - ▶ Intervjuu
 - ▶ (Mõned muud ettevõtte spetsiifilised tegevused)

Mõned lingid suve praktikatele:

- ▶ CGI <http://cgisuvepraktika.ee/>
- ▶ Playtech <http://playtech.ee/?nav=Internships>
- ▶ Nortal <https://nortal.com/careers/summeruniversity/>

Joonis 39. Näidis slaid 2 “Programmeerimine keeles Java II” tarkvara ettevõtete teemast

“Programmeerimine keeles Java II” aine “Klassid ja rekursioon” teema näidis slaidid on nähtavad joonisel 40 ja 41. Kõiki tunni slaide saab näha ja alla laadida siit: https://drive.google.com/open?id=1IxxxyjIMiqqBM5s7PHINEvb_oiuTBOWB6-XaeIG7lk.

Mis on rekursioon?

- ▶ Rekursioon - meetodi seest iseenda välja kutsumine (kolmas viis tsüklite loomiseks!)

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println(arvudeSummaAkuniB(scanner.nextInt(), scanner.nextInt()));
    scanner.close();
}

static int arvudeSummaAkuniB(int A, int B) {
    if (A == B) {
        return A;
    } else {
        return A + arvudeSummaAkuniB(A + 1, B);
    }
}
```

Joonis 40. Näidis slaid 1 “Programmeerimine keeles Java II” “Klassid III” teemast

Abstraktsed klassid

- ▶ Ülemklass ei pea kõike ära defineerima
 - ▶ Saab kohustada klasse kasutama teatud meetodeid
 - ▶ Ülemiselt määrateltud konstruktorid
- ▶ Sarnane asi on veel Interface
 - ▶ An interface can extend any number of interfaces at a time
 - ▶ An interface can only extend another interface
 - ▶ An interface can have only abstract methods

Joonis 41. Näidis slaid 2 “Programmeerimine keeles Java II” “Klassid III” teemast

Lisa 8. “Programmeerimine keeles Java II” aine iseseisvad ülesanded

Kõik vajalikud näidisfailid (nii Java kui ka tekstifailid) on leitavad kaustast <https://drive.google.com/open?id=1TEBq8uaY5ncutCa8O3aqdFaxdtyrEbSj>.

Tabel 12. “Programmeerimine keeles Java II” aine iseseisvad ülesanded

Kood	Teema	Ülesanne
j1	Git	<p>Teha läbi Git-it juhend Giti tundma õppimiseks. Juhendit on kohustuslik täita kuni sammuni “Remote control” (k.a.). Õpetajale tuleb esitada link loodud avalikule repositooriumile Githubis.</p> <p>Git-it juhend:</p> <ul style="list-style-type: none">• http://jlord.us/git-it/index.html• https://github.com/jlord/git-it-electron
j2	Kordamine	<p>Ülesande eesmärk: Loe standardsisendist (System.in) sisse fraasid, nt: "practice makes perfect", "sacred site", "small seagulls flying", "flying practice site". Väljasta fraasi keeleline struktuur (näidetel vastavalt: "noun verb adjective", "adjective noun", "adjective noun verb", "verb noun noun"). Kasuta nimisõnu (noun): "practice", "site", "seagulls", "glasses"; tegusõnu (verb): "makes", "flying", "are"; omadussõnu (adjective): "perfect", "sacred", "small", "clear", "fast".</p> <p>Lahendamiseks abistavad sammud:</p> <ol style="list-style-type: none">1. Loo (programmi jooksutamiseks) klass a21 main meetodiga.2. Loo klass Word (omadused: <i>String text</i>, <i>String type</i>) ja sellele konstruktor sisenditega: <i>text</i>, <i>type</i>.3. Loo klassile Word alamklassid Verb (<i>type="verb"</i>), Noun (<i>type="noun"</i>) ja Adjective (<i>type="adjective"</i>) ja neile kõigile konstruktorid sisendiga tekst.4. Loo etteantud sõnadega Word-tüüpi järjend (<i>List</i>) või massiiv (<i>Array</i>).

		<p>Selles peaksid olema sõnad oma tüübile vastavate klassidega (<code>new Noun("practice")</code>, <code>new Verb("makes")</code>) jne).</p> <p>5. Loe Scanneriga standardsisendist sisse sõnu, kuni neid on (<code>s.hasNext()</code> ja <code>s.next()</code>)</p> <p>6. Iga sõna jaoks leia loodud järjendist/massivist üles vastav objekt (text järgi, näiteks tsükliga üle käies)</p> <p>7. Trüki leitud sõna tüüp (võib ka trükkida mitmele reale).</p> <p>NB! Tegelikuses võib sõnal olla erinevaid vorme ning erinevas kontekstis võib sama kirjapildiga sõna olla erineva tähendusega (sh ka tüübiga). Kõige sellega me ei arvesta.</p> <p>Näide:</p> <p>Noun klass:</p> <pre>public class Noun extends Word{ public Noun(String text) { super(text, "noun"); } }</pre>
j3	Kordamine	<p>Eesmärk: Matemaatilise statistika tehete lahendamise programm (permutatsioonid, kombinatsioonid, variatsioonid).</p> <p>Loe failist <code>tehe.txt</code> (kus on ainult üks) käsk kujul "Permutatsioonid X", "Kombinatsioonid X Y" või "Variatsioonid X Y" ja salvesta tulemus (täpselt 1 täisarv) faili <code>vastus.txt</code>. X ja Y on mingid täisarvud. Arvutamiseks tee eraldi meetodid permutatsioonide, variatsioonide ja kombinatsioonide arvutamiseks sisendi(te) põhjal. Faktoriaali arvutamise koodi mitte dubleerida.</p>

		<p>Spikriks käskudele vastavate tehete matemaatilised kirjeldused</p> <p>Permutatsioonid: $X!$ (X faktoriaal) ehk $1*2*3*...*(X-1)*X$</p> <p>Variatsioonid: $X! / (X-Y)!$</p> <p>Kombinatsioonid: $X! / (Y! * (X-Y)!)$</p> <p>NB! Võib eeldada, et sisendid on piisavalt väikesed, et arvud (ka tehete vahetulemused) mahuvad int-tüüpi muutujasse.</p> <p>Lisaülesanne (1p): Kasuta kombinatsioonide arvutamiseks variatsioonide meetodit ja variatsioonide arvutamiseks permutatsioonide meetodit.</p>
j4	Lambdad & Streamid	<p>Loe failist "linnad.txt" sisse linnanimed ja trüki need konsooli suurte tähtedega (<i>uppercase</i>). Lahendus peab põhinema Streamidel ja Lambdadel.</p>
j5	Lambdad & Streamid	<p>Trüki konsooli suurim paarisarv failist "arvud.txt". Võib eeldada, et kõik arvud failis on täisarvud. Lahendus peab põhinema Streamidel ja Lambdadel.</p> <p>Lisaülesanne (1p): trüki ka kõigi arvude summa.</p>
j6	Erindid	<p><i>Stacktrace</i> lugemine ja koodi parandamine.</p> <p>Taustalugu:</p> <p>Õpetaja Tamm üritab leida väga tublide õpilaste (aka superstaaride) keskmist vanust, aga kiiruga kirjutatud programm ei tööta. Et temal aega vähe, siis palub ta sinul selle ära parandada.</p> <p>Mida teha?</p> <p>Lae alla kokkupakitud katkine programm j6.zip.</p> <p>Paki see lahti ja käivita programm. Uuri <i>stacktrace</i>'i ja erindile eelnevates tegevustes esinevaid vigu.</p>

		<p>Paranda Java programmis kõik vead ning esita parandatud fail õpetajale. Eesmärk on saada lõpuks etteantud hinded.txt failiga tulemuseks 17.5. Programmis on 3 viga. Peale iga vea parandamist käivita programm uuesti.</p> <p>Mida programm teeb?</p> <p>Failis hinded.txt on kirjas iga õpilase kohta tema nimi, vanus ning (keskmise) hinne. Programm leiab failist selliste õpilaste keskmise vanuse, kes on saanud maksimaalse võimaliku (keskmise) hinde.</p>
j7	Erindid	<p>Taustalugu:</p> <p>Käes on Farmimaa valimispäeva õhtu. Hääli loetakse, aga e-valimiste kesksüsteemi häälte teisendaja (kandidaadi number -> kandidaadi nimi) tõrgub pidevalt - testimisel selliseid probleeme ette ei tulnud. Avastad peagi, et mõned valijad on oma digitaalse hääletusedeli ära rikkunud. Olemasolev programm on kirjutatud keeles Penguin, mille kohta kuskil mingit infot pole. Ülesanne on luua uus programm, mis saaks hakkama ka ebakorreksete valikutega.</p> <p>Kuidas?</p> <p>Programm peab lugema failist "kandidaadid.txt" (alati kindlasti olemas) sisse kõikide kandidaatide nimed ning lisama need järjendisse (<i>List</i>) (võib teha nii tsükliliga kui ka streamiga, kuidas soovid). Failis kandidaadid.txt on iga kandidaadi nimi uuel real.</p> <p>Seejärel loeb programm failist "valik.txt" sisse numbri, mis tähistab kandidaadi järjekorranumbrit varasemalt loodud järjendis (faili järjekorranumbri loendus algab 1-st).</p> <p>Töötle kõiki tekkivaid erindeid <i>try-catch</i>-iga. Prindi iga erinevat moodi rikitud valiku põhjus ning vastav stacktrace. Kui valik pole rikitud, siis trüki kandidaadi nimi.</p>

		<p>Võimalikud rikitud valikute põhjused:</p> <p>"valiku faili ei leitud" - kui faili "valik.txt" pole või sellest lugemine ebaõnnestub</p> <p>"valik pole täisarv" - kui failis "valik.txt" on mingi jama, mis ei teisendu täisarvuks näiteks "vahet pole" või "ainult mitte Veskerakonda!"</p> <p>"valik pole sobiv järjekorranumber" - kui failis "valik.txt" on täisarv, aga see on näiteks liiga suur; näiteks 3 kandidaati, aga valik on 9001.</p> <p>Boonus 1p: Väldi erindi käsitlemise duplitseerimist - viska erindid <i>Exception</i>-ina (koos sõnumi ja põhjusega) edasi ja käsitle neid ühes kohas (terve koodi ümber, kõige välimiselt).</p>
j8	GUI	<p>Looge kaks-kolm ERINEVAT halba moodust, kuidas kasutajaliideses oma telefoninumbrit sisestada. Inspiratsiooniks https://qz.com/679782/programmers-imagine-the-most-ridiculous-ways-to-input-a-phone-number/. Võite kasutada lingist saadud ideid või ise mõelda välja mõne halva lahenduse telefoninumbri sisestamiseks.</p> <p>Kui olete seda ülesannet “Programmeerimine keeles Java I” juba teinud, siis looge kaks-kolm halba moodust sünnikuupäeva sisestamiseks.</p> <p>GUI disain 1p</p> <p>Koodi loogika 1p</p> <p>Esitage Controller, Main ja fxml fail. Kui loote lisaklasse, siis esitage ka need.</p>
j9	Internetist andmete laadimine	<p>Vabas vormis demonstreerige internetist andmete laadimist – kuidas? guugeldage. Lahendus on sobiv, kui teete vähemalt ühe päringu teete selle vastusega midagi (nt trükite konsooli). Kui ideed puuduvad, trükkige välja TÜ ilmajaama päringu vastuse sisu: http://meteo.physic.ut.ee/xml/data3.php.</p>

		Lisaülesanne (1p): Tehke interneti andmetega midagi enamat (ilmajaama näite puhul nt eraldage arvvaartused muust tekstist).
j10	Rekursioon	<p>Kirjutage programm, mis küsib kasutajalt ühte täisarvu (võib eeldada, et kasutaja sisestab alati positiivse täisarvu).</p> <p>Printige välja selle arvu faktoriaal, kasutades selleks rekursiivset meetodit. Meetod võib võtta argumendina sisendiks ainult ühe täisarvu.</p> <p>Meeldetuletus: Faktoriaal on n esimese positiivse täisarvu korrutis. Näiteks $6! = 6 * 5 * 4 * 3 * 2 * 1 = 720$.</p>
kt1	Kontrolltöö	<p>Paranda etteantud koodis kaks viga ning täienda koodi nii, et see arvutaks sisse loetud uudiste pealkirjade keskmise pikkuse (viimased kaks rida koodis. Otsi TODO märksõna). Esita sama fail koos paranduste/täiendustega. Abiks <i>stacktrace</i>-id programmi käivitamisel.</p> <p>NB! Uudiste sisselugemine on korrektne – seal vigu pole ning midagi parandama ei pea.</p> <p>Kontrolltöö etteantud kood on leitav Google Drive kaustast “Programmeerimine keeles java II” failid”: https://drive.google.com/open?id=1TEBq8uaY5ncutCa8O3aqdFaxdyrEbSj.</p> <p>Kontrolltöö fail on EestiUudised.java.</p>

Lisa 9. “Programmeerimine keeles Java II” aine moodle testid

“Programmeerimine keeles Java I” moodle testid on kättesaadavad sellelt lingilt:

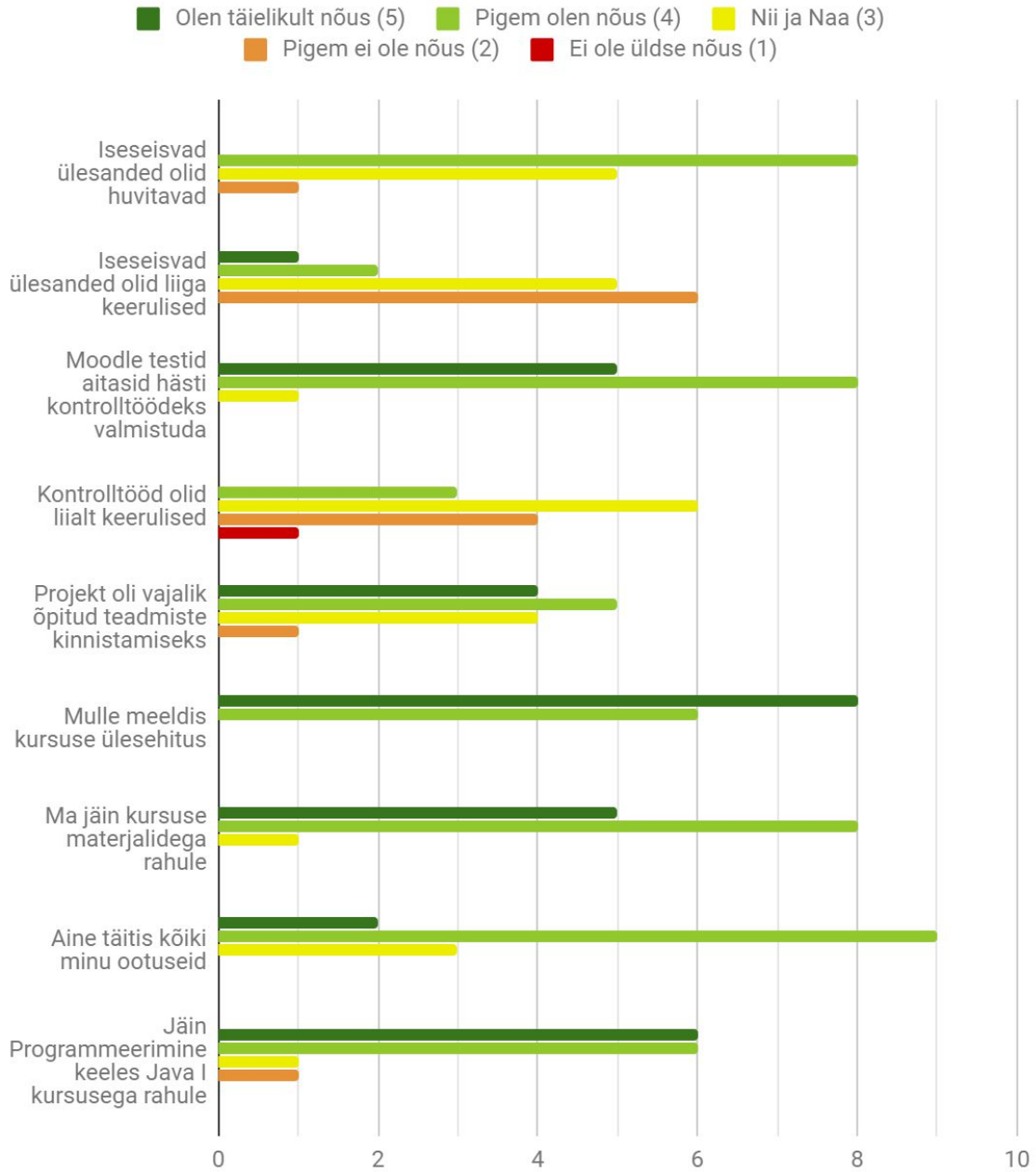
<https://drive.google.com/open?id=1QsTnKrPS-JDGRcEGDtobQ2gi0clJiblH>.

Lisa 10. Kursuste tagasiside küsitluste link

Link tagasiside küsimustikule: <https://forms.gle/aPR8cUGTPBnUYYzz7>.

Lisa 11. Kursuste tagasiside küsitluste suured graafikud

Vastajate arv 14



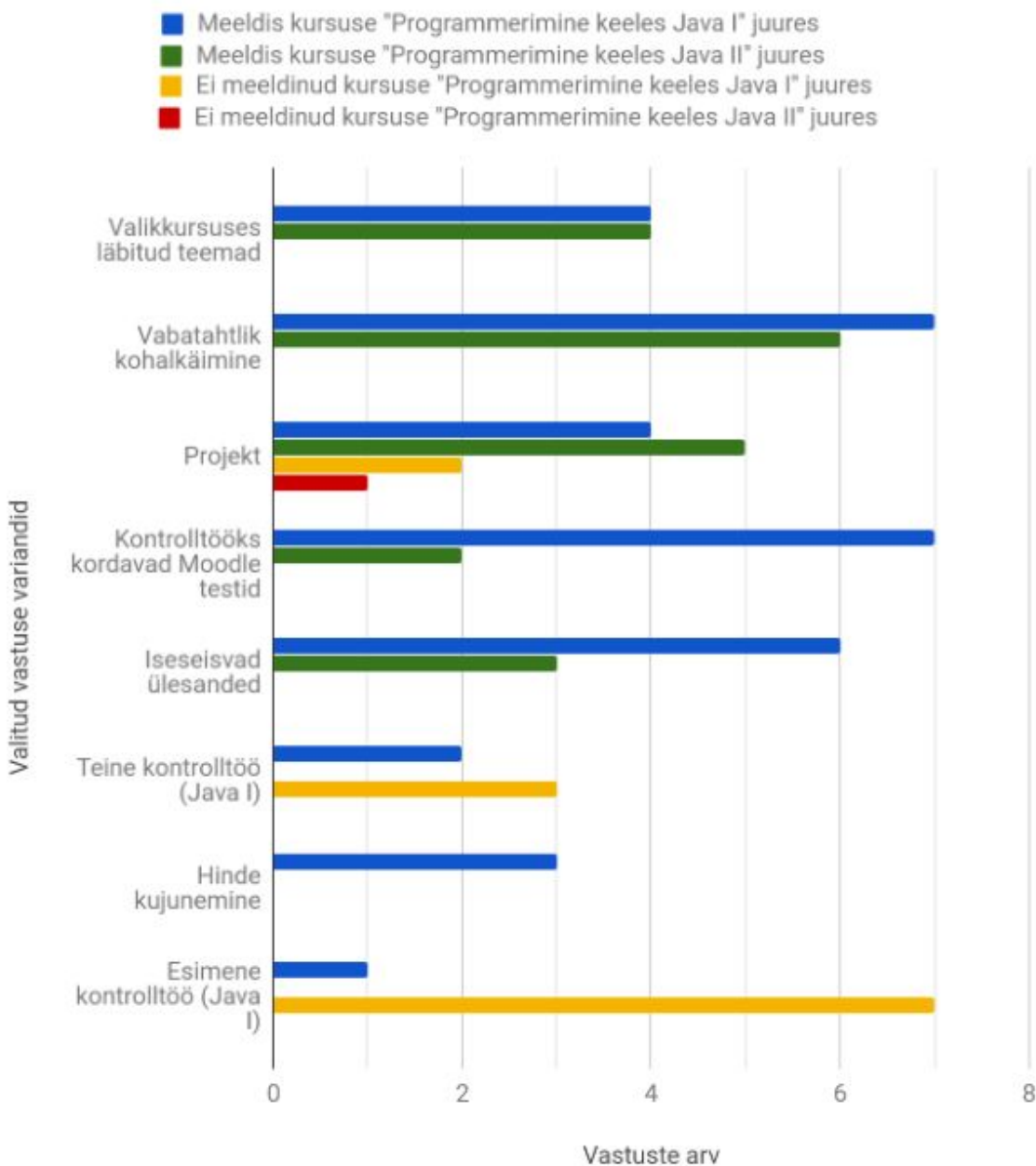
Joonis 42. TJPG õpilaste hinnangud “Programmeerimine keeles Java I” kursusega rahulolu kohta esitatud väidete põhjal

Vastajate arv 7



Joonis 43. TJPG õpilaste hinnangud “Programmeerimine keeles Java II” kursusega rahulolu kohta esitatud väidete põhjal

"Programmerimine keeles Java I" kursuse läbinud vastajate arv 14
 "Programmerimine keeles Java II" kursuse läbinud vastajate arv 7



Joonis 44. TJPG õpilaste vastused küsimusele “Mis üldse ei meeldinud kursuse korralduse juures?” ja “Mis eriti meeldis kursuse korralduse juures”

Lisa 12. Õpilaste soovitused “Programmeerimine keeles Java I” ning “Programmeerimine keeles Java II” korraldamiseks

Vaba tekstilisele küsimusele “Mida oleks võinud kursuse korralduses teisiti teha?” ning “Mida oleks võinud kursuse materjalides teisiti teha?” vastasid enamik õpilasi, et nad ei oska midagi soovitada või, et nad jäid kõigega rahule.

Õpilaste ettepanekud “Programmeerimine keeles Java I” kursuse korralduse parendamiseks olid järgmised:

- “Võiks olla kaks juhendajat, selleks et oleks võimalik rohkem õpilasi aidata.”;
- “Kursus ise oli väga hea aga siiski mahukas. Mulle oleks meeldinud, kui kiirus oleks natu aeglasem olnud, et oleks rohkem aega tegeleda harjutuste ja teemadega.”;
- “Oluline on käskude ja koodi meelde jätmine/ pähe õppimine. Seetõttu võiks mingeid koodijuppe teha lihtsamaid, aga rohkem kordi, et jääks meelde ja ei läheks segamini.”;
- “Kergemad kontrolltööd ja vähem iseseisvad ülesandeid”
- “Teha koos ülesandeid rohkem.”.

Õpilaste ettepanekud “Programmeerimine keeles Java II” kursuse korralduse parendamiseks olid järgmised:

- “Rohkem projektiga tegeleda, et kõik oleks jõudnud selle valmis teha”
- “Sujuvam info levimine, nendele, kes tunnis kohal ei ole.”

Õpilaste ettepanekud “Programmeerimine keeles Java I” kursuse materjalide parendamiseks olid järgmised:

- “Hea mitmekesine materjal, võibolla siiski lühemaid selgemaid harjutusi algul.”;
- “Võib-olla uue teema juures ette antud näite ülesanne (ülesande juhend, lahendus ning kommentaarid miks midagi tehti)”;
- “Materjalid olid head aga kasutasin enamus ajast hoopis googli abi võimalik et mõned kohad vajasisid rohkem selgitust aga see on pisidetail.”;

- “Rääkida või rohkem kirjutada näite ülesannet puhul, et mida mingi käsk või osa teeb.”
- “Näidis videoid oleks võinud olla”

Õpilaste ettepanekud “Programmerimine keeles Java II” kursuse materjalide parendamiseks olid järgmised:

- “Oleks võinud käsitleda teemasid mida sooviti kasutada projektides”
- “Materjale oleks võinud rohkem olla”
- “Hindajas [autor: aines kasutatud automaatse hindamise süsteem] oleks võinud näidata perfektset lahendust kommentaaridega.”

Lihtlitsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Helen Õunapuu,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Ainekavade ja materjalide koostamine gümnaasiumi valikainetele „Programmeerimine keeles Java I“ ja „Programmeerimine keeles Java II“, mille juhendajad on Anne Villems ja Merike Hein, 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 litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Helen Õunapuu

19.05.2019