

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Andri Soone
Programmeerimise kursusele lisamaterjalide
koostamine
Bakalaureusetöö (9 EAP)

Juhendaja: Reimo Palm

Tartu 2020

Programmeerimise kursusele lisamaterjalide koostamine

Lühikokkuvõte:

Käesoleva bakalaureusetöö raames loodi Tartu Ülikooli informaatika eriala sissejuhatavale programmeerimiskursusele lisamaterjalide komplekt, mille eesmärk on suurendada tudengite silmaringi programmeerimise valdkonnas, õpetada võimalusi programmeerimisalase projekti loomiseks ja tutvustada õppekavas järgnevate ainete temaatikat. Materjalide teemade valimiseks uuriti teisi Pythoni algkursuseid ja järgnevaid kursuseid informaatika erialal. Kaheksa valitud teema hulka kuuluvad regulaaravaldised, andmebaasid, veebirakenduste loomine, graafiliste mängude loomine, objektorienteeritud programmeerimine ja muud. Bakalaureusetöös kirjeldatakse teemade valikut, valitud teemade põhjendusi ning materjalide koostamise põhimõtteid.

Võtmesõnad:

„Programmeerimine”, õppematerjalide koostamine, lisamaterjalid

CERCS: P175 Informaatika, süsteemiteooria; S270 Pedagoogika ja didaktika

Creating Additional Materials for the Introductory Programming Course

Abstract:

The purpose of this thesis is to create additional learning materials for the introductory programming course in the University of Tartu Computer Science programme. The goal of the materials is to introduce different programming fields, teach opportunities for developing a programming project and to introduce upcoming courses in the curriculum. Other introductory programming courses and upcoming courses in the curriculum were researched in order to select topics for the materials. The eight chosen topics include regular expressions, databases, web application development, graphical game development, object-oriented programming and more. The thesis describes the selection of the topics, reasons for selecting the topics and the principles of creating the materials.

Keywords:

Introductory programming course, learning materials creation, additional learning materials

CERCS: P175 Informatics, system theory; S270 Pedagogy and didactics

Sisukord

1. Eesmärgi püstitus	5
1.1. Sissejuhatav programmeerimiskursus	5
1.2. Lisamaterjalide olulisus	5
2. Lisamaterjalide teemad	7
2.1. Teised Pythoni algkursused	7
2.2. Järgnevad kursused informaatika õppekavas	8
2.3. Valitud teemad	8
2.3.1. Standardteek ja moodulid	9
2.3.2. Rakendusliidesed	9
2.3.3. Regulaaravaldised	10
2.3.4. Andmebaasid	10
2.3.5. Veebirakenduste loomine	11
2.3.6. Objektorienteeritud programmeerimine	12
2.3.7. Graafiliste mängude loomine	13
2.3.8. Keerulisemad Pythoni võimalused	14
2.4. Valikust välja jäänud teemad	15
3. Lisamaterjalide koostamine	16
3.1. Materjalide koostamise põhimõtted	16
3.2. Materjalide struktuur	17
3.2.1. Põhiosa	17
3.2.2. Enesekontrolliküsimused	20
3.2.3. Ülesanded	21
3.3. Materjalide tulevik	22
4. Viidatud kirjandus	24
Lisad	26
I. Kaasa pandud materjalid	26
II. Litsents	27

1. Eesmärgi püstitus

1.1. Sissejuhatav programmeerimiskursus

Tartu Ülikooli informaatika eriala esimene programmeerimiskursus on aine „Programmeerimine“ (LTAT.03.001) [1], edaspidi Programmeerimiskursus. Selles aines tehakse sissejuhatus programmeerimisse, kasutades programmeerimiskeelt Python. Kursusel käsitletakse põhilisi programmeerimise konstruktsioone, andmetüüpe ja -struktuure, õpetatakse analüüsima programmi töö käiku ja looma lihtsamatele ülesannetele algoritmilisi lahendusi.

Kursus algab sissejuhatusel, kus paigaldatakse arenduskeskkond, luuakse esimesed programmid kasutaja sisendiga ja lihtlausetega ning joonistatakse kilpkonnagraafikaga. Seejärel vaadatakse tingimus- ja korduslauseid, funktsioone ning erindeid. Kursuse teises osas tutvustatakse keerulisemaid algoritme plokk skeemidega ning keerulisemaid andmestruktuure: järjendid, maatriksid, sõnastikud, ennikud ja hulgad. Kursuse lõpu poole käsitletakse kahekordseid tsükleid ja rekursiooni.

1.2. Lisamaterjalide olulisus

Üks Programmeerimiskursuse osadest on koostöös rühmakaaslasega programmeerimisalase projekti realiseerimine. Projekti teema valivad tudengid ise ning neil on selleks palju võimalusi, sh graafilise kasutajaliidese, mängu või veebirakenduse loomine. Selliseid oskuseid aines ei käsitleta ning kui tudeng tahab neid rakendada, peab need iseseisvalt omandama.

Paljudel ainet läbivatel tudengitel on varasem Pythoni kogemus olemas, olgu see omandatud iseseisvalt õppides, kooli kaudu või läbides mõnda e-kursust. Sellised tudengid saavad proovida eeleksamit, mille läbides saavad nad aine läbitud ilma kursusel osalemata. Eeleksam hõlmab kogu aine materjali ja selle läbivad edukalt vähesed. 2018. aastal läbis eeleksami 8 tudengit 376-st ja 2019. aastal 18 tudengit 362-st. Sellest tulenevalt peavad paljud tudengid läbima ainet, millest arvestatav osa on neil selge. Sellistele tudengitele ei pruugi aine olla huvitav.

Selleks, et teha kogenud esmakursuslasele aine huvitavamaks ja pakkuda projektide realiseerimiseks vajalikke oskusi, tulevad kasuks lisamaterjalid, mille järgi saavad tudengid kursuse kõrval omandada lisateadmisi. Lisamaterjalid pakuvad lisaks võimalusi õppida teemasid, mis põhikursuse sisse ei mahu ja saada sissejuhatusi informaatika õppekavas programmeerimisele järgnevatele kursustele.

Kursuse õppematerjalina kasutataval programmeerimise õpikul [2] on olemas lisamaterjalide osa. Sealsed lisamaterjalid eeldavad aga, et lugejal on neist arusaamiseks õpik juba läbitud. Materjalides on 11 erinevat teemat ning 5 neist on bakalaureusetöö kirjutamise hetkel poolikud.

Lõputöö eesmärk on kursusele Programmeerimiskursusele valida välja sobivad Pythonis programmeerimisega seotud lisamaterjalide teemad ja luua nendele vastavad materjalid, mida saab läbida Programmeerimiskursusega paralleelselt. Lõputöö kirjalikus osas kirjeldatakse lisamaterjalide teemade valimisprotsessi, valitud teemasid koos põhjendustega ning lisamaterjalide koostamise põhimõtteid. Loodud materjalid on tööga kaasa pandud ja kirjeldatud lisa 1.

2. Lisamaterjalide teemad

Käesolevas peatükis tehakse ülevaade lisamaterjalide teemade valimisprotsessist ning valitud ja valimata jäetud teemadest. Teemade valimisel uuris lõputöö autor teisi Pythoni algkursuseid. Võeti ka arvesse informaatika õppekavas järgnevate kursuste teemasid, millele saab teha sissejuhatusi Pythoniga.

2.1. Teised Pythoni algkursused

Siin on välja toodud ülevaade teistest Pythoni kursustest ja nende teemadest, mida Programmeerimiskursus ei käsitle. Kui kursuse sisu ei ole avalikult kättesaadav, on järeldused tehtud kursuste sisukordadest.

Uuritavate kursuste hulka kuulusid Tartu Ülikooli MOOC-id (*massive open online course*) „Programmeerimisest maalähedaselt“ [3], „Programmeerimise alused“ [4] ja „Programmeerimise alused II“ [5]. Kursuste põhiosa on sarnane Programmeerimiskursuse materjalidega, aga mõne erinevusega. Iga peatükk sisaldab osa, mille eesmärk on avardada kursuse läbija silmaringi. Paljud sellised silmaringiteemad on näiteks seotud programmeerimise ajaloo ja erinevate IT valdkondadega. Mõned teemad on ka kasulikest Pythoni võimalustest, näiteks regulaaravaldistest, andmebaasidest, objektorienteeritud programmeerimisest ja andmeteadusest.

Coursera platvormi MOOC „Python for Everybody“ [6] keskendub süvitsi andmeteadusele. Õpetatakse andmete kogumist rakendusliidestega, nende salvestamist andmebaasidesse ning töötlemist ja visualiseerimist vastavate Pythoni moodulitega. Andmete salvestamisel tutvustatakse ka objektorienteeritud programmeerimist.

Massachusettsi Tehnoloogiainstituudi (MIT) sissejuhatav Pythoni kursus „Introduction to Computer Science and Programming in Python“ on avalikult kättesaadav MIT OpenCourseWare keskkonnas [7]. Kursuses pööratakse tähelepanu objektorienteeritud programmeerimisele, programmi testimisele, silumisele ja programmi efektiivsuse hindamisele.

Veebikursuse „Learn Python“ [8] põhiosas käsitletakse erinevalt Programmeerimiskursusest lisaks objektorienteeritud programmeerimist ning standardteeki ja selle mooduleid. Kursusel

on ka andmeteaduse osa ja edasijõudnute osa. Edasijõudnute osa keskendub Pythoni funktsionaalprogrammeerimise võimalustele.

Codecademy keskkonna tasuta veebikursus „Learn Python 3“ [9] tutvustab lisaks Programmeerimiskursuse teemadele mõnda standardteegi moodulit, objektorienteeritud programmeerimist ja funktsioonide lisavõimalusi.

Allen Downey raamat „Think Python“ [10], käsitleb lisaks Programmeerimiskursuse teemadele objektorienteeritud programmeerimist, keeletehnoloogiat, ajalist keerukust, silumist, moodulite kirjutamist ja graafilisi liideseid.

2.2. Järgnevad kursused informaatika õppekavas

Selleks, et säilitada tudengi huvi programmeerimise vastu ja tutvustada erinevaid programmeerimise võimalusi, saab lisamaterjalides teha sissejuhatusi mõnele Programmeerimiskursusele järgnevale ainele. Varasem kokkupuude järgnevate kursustega teeb nendele ainetele ülemineku lihtsamaks.

Aines „Objektorienteeritud programmeerimine“ (LTAT.03.003) [11] tutvustatakse objektorienteeritud programmeerimise põhimõisteid programmeerimiskeeles Java. Aine „Andmebaasid“ (LTAT.03.004) [12] suur osa on SQL keele õppimine. Aine „Veebirakenduste loomine“ (LTAT.05.004) [13] käsitleb veebirakenduste loomist erinevate keelte ja raamistikuga. Ained „Arvutigraafika“ (MTAT.03.015) [14] ning „Arvutimängude loomine ja disain“ (MTAT.03.263) [15] käsitlevad mingil määral kahemõõtmelisi mängu. Aine „Automaadid, keeled ja translaatorid“ (LTAT.03.006) [16] algab regulaaravaldiste õppimisega. Aines „Programmeerimiskeeled“ (MTAT.03.006) [17] käsitletakse funktsionaalprogrammeerimist. Kõiki loetletud teemasid on võimalik tutvustada programmeerimiskeelega Python, kasutades vajadusel vastavaid teeke.

2.3. Valitud teemad

Lõputöö autor valis välja 8 teemat, millest kirjutada lisamaterjalid, lähtudes teemade esinemisest teistes kursustes, teemade kasulikkusest programmeerimisalase projekti realiseerimisel ning võimalusest valmistada ette tudeng Programmeerimiskursusele järgnevate ainete võtmisele. Autor võttis arvesse seda, et teemasid peab saama läbida paralleelselt Programmeerimiskursusega ning et valitud teemade komplekt on terviklik.

2.3.1. Standardteek ja moodulid

Lisamaterjalide esimeses peatükis tutvustatakse erinevaid Pythoni standardteegis olevaid mooduleid:

- moodul *math* matemaatiliste funktsioonide rakendamiseks
- moodul *random* juhuslikkuse kasutamiseks programmides
- moodul *datetime* kuupäevade kasutamiseks ja nendega arvutamiseks programmides
- moodul *os* operatsioonisüsteemi funktsionaalsusele ligi pääsemiseks
- moodul *sys* käsurea argumentidele ligi pääsemiseks ja programmi töö lõpetamiseks

Tutvustatakse enda mooduli loomist ning selgitatakse, kuidas Pythoni moodulid töötavad ja kuidas nende sisu saab uurida.

See teema esineb paljudes teistes kursustes ning selle õpetamine on kasulik, et tutvustada tudengitele ideed, et paljud moodulid on juba kirjutatud ja nende kasutamine on lihtne.

Selle peatüki materjale saab läbida vahetult pärast õpiku esimese peatüki läbimist, sest see ei nõua palju eelteadmisi, mis teeb selle heaks esimeseks lisamaterjali temaks. Ühes osas kasutatakse funktsiooni ja järjendi mõistet, mida peab õpiku peatükkidest 4 ja 7 lugema juhul, kui need ei ole selged.

2.3.2. Rakendusliidesed

Teises peatükis tutvustatakse internetilehekülgedelt JSON-vormingus andmete pärimist mooduliga Requests [18]. Käsitletakse erinevaid päringuid ja JSON-vormingu teisendamist Pythoni sõnastikeks. Ühe näitena ehitatakse valuutakalkulaator, mis võtab valuuta väärtused Internetist. Teise näitega valmib internetiaadressi lühendamise programm, mis kasutab vastavat rakendusliidest. Tutvustatakse teisi rakendusliideseid, mis tagastavad JSON-vormingus andmeid: ilmaennustus, nimepäevade andmed, sõnastikud, tõlkimine ja muud. Lühidalt mainitakse autentimist nõudvaid rakendusliideseid. Tuuakse välja neli avalikku nimekirja rakendusliidestest.

Rakendusliideste teema esineb Coursera platvormi Pythoni kursuses [6] ning selle omandamine annab võimaluse paljude projektiideede teostamiseks.

Sellest peatükist arusaamiseks piisab Pythoni lihtlausetate ja moodulite kasutamise oskamisest. Neid selgitatakse õpiku esimeses ja teises peatükis. Vajalik on ka sõnastiku andmestruktuurist

väärtuste kättesaamise oskus. Kui sõnastiku mõiste on võõras, tuleb läbi lugeda õpiku 10. peatüki vastav lõik.

2.3.3. Regulaaravaldised

Kolmandas peatükis tutvustatakse regulaaravaldisi. Tehakse sissejuhatus regulaaravaldiste mustritesse, käsitletakse regulaaravaldistes kasutatavaid sümboleid, grupeerimist, gruppide kordusi ning Pythonis regulaaravaldistega sõnade kontrollimist, nende seest otsimist ja nende osade asendamist. Näited tuuakse e-posti aadressi korrektsuse kontrollimisest, telefoninumbrite otsimisest ning Vikipeedia artiklist viidete eemaldamisest. Selleks, et vähendada peatüki mahtu, ei käsitleta vastetest gruppide väärtuste kättesaamist. Seda mainitakse küll peatüki lõpus.

Regulaaravaldiste teema esineb mõnes teises Pythoni kursuses ning aine „Automaadid, keeled ja translaatorid“ [16] algab süvisi regulaaravaldiste uurimisega. Peatükk on kasulik, sest paljud programmeerijatel ette tulevad ülesanded on seotud sõnade ja teksti analüüsiga.

Sellest peatükist arusaamiseks peab oskama avaldise, lihtlauseid ja tingimuslauseid ehk läbima peab õpiku esimesed 3 peatükki. Kasutatakse järjendi mõistet. Kui see ei ole selge, peab järjenditega tutvuma õpiku 7. peatükis vastavas lõigus.

2.3.4. Andmebaasid

Neljandas peatükis tutvustatakse andmebaase. Tehakse sissejuhatus SQL keelde: tabelite loomine, andmete lisamine tabelisse ja andmete pärimine tabelist. Kasutatakse andmebaasi SQLite [19] ja selle Pythoni liidest [20], et rakendada õpitud SQL-käsked Pythonis. Näitena luuakse kasutajate andmebaas kasutajanimede ja paroolidega. Rõhutatakse, et paroole ei tohi tegelikult andmebaasis tekstina hoida ning tuuakse näide turvariskist, mis võib juhtuda, kui anda kasutajale võimalus SQL-käsked mõjutada.

Andmebaaside teema esineb mõnes Pythoni kursuses ning see on kasulik, sest see teeb sissejuhatus Programmeerimiskursusele vahetult järgnevale ainele „Andmebaasid“. [12] Andmebaaside kasutamise oskus väldib olukorda, kus algaja programmeerija salvestab palju andmeid tekstifailidesse.

Sellest peatükist arusaamiseks peab oskama avaldise ja lihtlauseid ning tingimuslauseid ehk läbima peab õpiku esimesed 3 peatükki. Kasutatakse järjendi ja enniku mõisteid. Kui need ei ole selged, peab nendega tutvuma õpiku 7. peatüki vastavates osades.

2.3.5. Veebirakenduste loomine

Viiendas peatükis tehakse sissejuhatus veebirakenduste arendamisse, kasutades Flaski [21]. Flask on Pythoni raamistik, mis lubab veebirakenduse püsti saada vaid mõne reaga ilma, et arendaja peab teadma, kuidas veebirakendused täpsemalt töötavad. Peatükis alustatakse tervitusrakendusega ning seejärel luuakse mitu lehekülge, küsitakse kasutajalt sisendit ja väljastatakse HTML koodi. Peatükki läbides luuakse juhusliku arvu generaator, alustades väga lihtsast variandist ja lisades sellele aina rohkem võimalusi juurde (Joonis 1). Peatüki lõpus on juhend, kuidas loodud veebirakendust üles laadida keskkonda PythonAnywhere [22].

Veebirakenduste peatükk on kasulik, sest see teeb sissejuhatusse väga tähtsale programmeerimise valdkonnale. Omandatud oskusi saab kasutada Programmeerimiskursuse projektis. Aine „Veebirakenduste loomine“ [13] süveneb sügavamalt temaatikasse ning tudengile on kasulik, kui enne selle aine võtmist on mingi kogemus olemas.

Sellest peatükist arusaamiseks peab oskama avaldise, lihtlauseid ja funktsioone ehk läbima peab õpiku esimese osa. HTML-i oskus on peatüki läbimisel kasulik, kuid pole vajalik: kõik kasutatud HTML ja CSS koodid on ette antud.



Joonis 1. Veebirakenduste loomise peatükis valmiv projekt

2.3.6. Objektorienteeritud programmeerimine

Kuuendas peatükis tehakse sissejuhatus objektorienteeritud programmeerimise paradigmasse. Peatüki alguses uuritakse mooduli *datetime* objekti *datetime* ning selle võimalusi. Seejärel luuakse enda kuupäeva klass, alustades konstruktoritest ja isendiväljadest ning lisatakse meetodeid, mille abil õpetatakse Pythoni klasside omadusi. Peatüki lõpuks valmib kuupäeva klass, millel on võimalused kuupäeva väljastamiseks sõnena, päevade juurde lisamiseks ja hetkese kuupäeva saamiseks.

Peatükk on kasulik, sest Programmeerimiskursusele vahetult järgnev kursus on „Objektorienteeritud programmeerimine“ [11]. Kuigi selles aines kasutatakse programmeerimiskeelt Java, teeb objektorienteeritud programmeerimise tundmine Pythoniga ülemineku järgmisele kursusele lihtsamaks. Kui tudengil on plaanis Pythoni kasutamisega jätkata pärast Programmeerimiskursust, on objektide kasutamisoskus vajalik, sest paljud

teegid ja raamistikud kasutavad neid. Objektorienteeritud programmeerimist tutvustavad kõik teised Pythoni algkursused, mida lõputöö autor uuris.

Sellest peatükist arusaamiseks peab lugejal olema läbitud õpiku esimene osa.

2.3.7. Graafiliste mängude loomine

Seitsmendas peatükis tehakse sissejuhatus graafiliste mängude loomisse, kasutades teeki Pygame [23]. Peatükki alustatakse kõige lihtsama Pygame'i akna kuvamisega ning jätkatakse selle täitmisega: aken täidetakse, joonistatakse kujundeid, aknasse paigutatakse pildifaile ning kuvatakse teksti. Seejärel vaadatakse, kuidas joonistatud esemeid liikuma panna ning seejärel liigutatakse neid kasutajasisendiga. Lõpus vaadatakse, kuidas kontrollida kahe eseme kokkupuutumist.

Peatüki jooksul valmib mäng, kus mängija juhib rott, kes peab liikuma juustu juurde. Kui rott juustu puudutab, siis juust hüppab uuele asukohale. Ekraanitõmmist valminud mängust on näha joonisel 2.

Graafiliste mängude loomise peatükk on kasulik, sest see tutvustab tudengeid arvutigraafika valdkonda ning annab võimaluse paljude projektiideede teostamiseks. Arvutigraafikaga tegelemine teeb lihtsama ülemineku ainetesse „Arvutigraafika“ [14] ning „Arvutimängude loomine ja disain“ [15].

Sellest peatükist arusaamiseks peab olema läbitud õpiku esimene osa. Tuleb kasuks, kui lugejal on läbitud lisamaterjalide peatükk objektorienteeritud programmeerimisest, sest peatükis käsitletakse palju objekte, aga see ei ole vajalik, et aru saada graafiliste mängude loomise peatükist.



Joonis 2. Graafiliste mängude loomise peatükis valmiv mäng.

2.3.8. Keerulisemad Pythoni võimalused

Kaheksandas peatükis tehakse tutvustus Pythoni funktsionaalprogrammeerimise võimalustesse. Peatükis käsitletakse lühemaid if-lauseid, järjendi hõlmamist (*list comprehension*), lambda-funktsioone ning funktsioone *map*, *filter* ja *zip*.

See peatükk on kasulik, sest see tutvustab lugejale Pythoni võimalusi, mis teevad paljudele programmeerijatel ettetulevatele probleemidele lahendused lihtsamaks. Kuna peatükk käsitleb Pythoni funktsionaalprogrammeerimise võimalusi, siis on peatükk sissejuhatuseks ainele „Programmeerimiskeeled“ [17].

Sellest peatükist arusaamiseks peab lugejal olema väga hea ülevaade Pythonist ehk läbitud peavad olema õpiku mõlemad osad. Selle tõttu on see teema hea valik kõige viimaseks lisamaterjalide peatükiks.

2.4. Valikust välja jäänud teemad

Mõned lisamaterjalide teemad olid veel valikus, aga jäid valimata, et vähendada autori töömahtu. Lõputöö autor leidis, et ilma nende teemadeta on loodud lisamaterjalide komplekt piisav, kuid tulevikus saab need juurde lisada.

Üks teema võimalus on tutvustada Pythoni koodi stiilipõhimõtteid, mida soovitab Pythoni stiiliõpetus PEP 8 [24]. Peatükis saab veel mainida Pythoni muutujatüüpide vihjamise võimalusi, dokumenteerimise tavaid ja muid hea koodi kirjutamise tavaid. See teema ei nõua palju eelteadmisi, aga on kasulik, kui lugejal on juba mõningaid kogemusi Pythoniga, seega sobib see kuskile materjalide keskele. Lõputöö autor leidis, et selle teema informatsiooni on keeruline huvitavalt õpetada ning ülesandeid kirjutada.

Teine teema võimalus on tutvustada andmeteadust, kasutades Pythoni teeki NumPy [25] ja pandas [26]. Peatükis saab teha ülevaate erinevatest andmestruktuuridest, milles andmeid hoida ning andmete visualiseerimise võimalustest. Kui materjalidesse mahub, saab ka tutvustada lihtsamaid masinõppe omadusi. Selle teemaga saab teha sissejuhatuse informaatika õppekava andmeteaduse ainetesse, näiteks „Sissejuhatus andmeteadusesse“ (LTAT.02.002) [27]. Teema on üpris keeruline ja seda sobib paremini paigutada kursuse lõpupoole. Kursusel "Programmeerimise alused II" [5] on sellest teemast mahukad materjalid olemas.

Kolmas teema võimalus on tutvustada graafiliste kasutajaliideste loomist teegiga tkinter [28] või mõne sarnasega. Programmeerimise õpiku lisade all on sellest teemast juba mahukad materjalid olemas. Teema sobib pigem paigutamiseks kursuse lõpupoole.

Veel võimalikud lisamaterjalide teemad on programmide testimine, Pythoni virtuaalkeskonnad (*virtualenv*), veebilehtede parsimine ning erinevate arendustarkvarade tutvustamine.

3. Lisamaterjalide koostamine

Käesolevas peatükis tehakse ülevaade lisamaterjalide koostamise põhimõtetest ja nende struktuurist. Peatüki lõpus tehakse ülevaade materjalide kasutamisest tulevikus.

3.1. Materjalide koostamise põhimõtted

Lisamaterjalide koostamisel tugines lõputöö autor Wikibooks'i õpikute kirjutamise põhimõtetele [29]. Need koosnevad viiest reeglist: raamistike reegel, tähendusrikaste nimede reegel, hallatavate arvude reegel, hierarhia reegel ja kordamise reegel.

Raamistike reegel nõuab, et õppematerjalid järgivad ühtlast struktuuri eesmärgiga, et lugeja saab esimese mõne peatükiga selgeks, kuidas materjale lugeda ning ei pea seda mitu korda ümber õppima. Lõputöö raames loodud lisamaterjalide peatükid põhinevad kindlal struktuuril, mida kirjeldatakse peatükis 3.2.

Tähendusrikaste nimede reegel nõuab, et igal käsitletud mõistel, meetodil ja faktil peab olema üheselt kasutatav tähendusrikas nimetus, et vähendada mõistete arvu, mida lugeja peab õppima. Need nimetused peavad olema juba kasutusel selles valdkonnas, et edasi õppides ei tohiks olla vajadust neid nimetusi ümber õppida. Lisamaterjalides on pööratud sellele reeglile tähelepanu ning igale ideele viidatakse ainult ühe mõistega, mis on kasutusel ka mujal selle temaga seotud materjalides.

Hallatavate arvude reegel nõuab, et ühe korraga tutvustatavat informatsiooni peab piirama 4 kuni 6 uue ideega, et lugeja ei peaks enne peatüki lõpuni jõudmist materjali kordama. Lisamaterjalid on kirjutatud eesmärgiga edasi anda võimalikult vähe informatsiooni nii, et kõik vajalik on mainitud ning lõpptulemus oleks kasulik ja huvitav. Näiteks jäi regulaaravaldiste peatükis näitamata püütud gruppidest informatsiooni kättesaamine. See on tähtis osa, aga selle lisamine teeks peatüki liiga pikaks ja peatükk on ilma selleta terviklik.

Hierarhia reegel nõuab, et uued materjalid on märgatavalt seotud läbitud materjalidega, et lugeja saaks uue materjali omandamisel tugineda varasematele teadmistele. Lisamaterjale ei saa omavahel nii siduda, sest kõik teemad on eraldiseisvad ja igat peatükki peaks saama läbida ilma eelmiste läbimiseta. Näiteks kordub internetiaadresside parameetrite teema nii rakendusliideste kui veebirakenduste peatükis ja seda selgitatakse mõlemas peatükis eraldi.

Kuigi lisamaterjalide peatükid pole omavahel seotud, on need seotud õpiku materjaliga, sest õpiku materjalide tundmine on vajalik lisamaterjalidest arusaamiseks. Selle reegli järgimiseks korratakse lisamaterjalides üle õpikumaterjalid, mida lisamaterjalides kasutatakse.

Kordamise reegel nõuab, et kõige tähtsamaid ideid peab kordama, sest enamik inimestest õpib kõige paremini läbi kordamise. Lisamaterjalides korratakse kõige tähtsamaid mõisteid mitmetes kontekstides. Peatükkide lõpus ja materjalide keskel on enesekontrolliküsimused, mille abil saab lugeja teadmisi korrata. Nendest kirjutatakse lähemalt peatükis 3.2.2.

3.2. Materjalide struktuur

Selleks, et järgida raamistike reeglit ja teha lugejale materjalide omandamine lihtsamaks, järgivad lisamaterjalid kindlat struktuuri. Iga peatükk algab teema sissejuhatusega, kus selgitatakse teema olulisust. Sissejuhatuses tuuakse välja huvitavaid näiteid probleemidest, mida lugeja oskab lahendada pärast peatüki läbimist. Sissejuhatusese järgneb ettevalmistuse osa, kus käsitletakse vajalike tööriistade paigaldamist ning seda, mis peatükid õpikust peavad olema loetud. Järgnevad materjalide põhiosa, enesekontrolliküsimused ja ülesanded, millest kirjutatakse lähemalt järgmistes alapeatükkides.

3.2.1. Põhiosa

Kuna lisamaterjalide teemad on väga erinevad, siis on iga peatüki põhiosa erinev, kuid neil leidub ühiseid jooni.

Koodi kasutamist tutvustatakse läbi näidete koodiplokkidega, mille järel on ülesanded nende proovimiseks. Näiteks veebirakenduste peatükis näidatakse ette muutuva sisu kuvamist veebilehel koodinäitega, pärast mida on ülesanne siduda see eelmises alapeatükis õpetatud dünaamiliste aadresside kasutamisega (Joonis 3). Regulaaravaldiste peatükis tuuakse näiteid erinevate funktsioonide kasutusest ja pärast igat näidet antakse näitest erinev ülesanne seda funktsiooni rakendada. Sellised näited esinevad kõikides peatükkides.

Huvitavam sisu

Lehekülje sisu ei pea samuti staatiline olema. Oletame, et lehekülg `random.org` on maas ja kirjutame sellest enda versiooni. Algul tagastame lihtsalt juhusliku arvu 1-st 10-ni:

```
from flask import Flask
import random

app = Flask(__name__)

@app.route("/")
def index():
    juhuslik_arv = random.randint(1, 10)
    return "Sinu juhuslik arv on {}".format(juhuslik_arv)

if __name__ == "__main__":
    app.run(debug=True)
```

Proovi täiendada seda rakendust nii, et kasutaja saab ise määrata juhusliku arvu ülem- ja alampiiri.

Kui külastada lehekülje juurt ehk `/`, on piirid `1` ja `10`.

Kui külastada aadressi `/<maksimum>`, on piirid `1` ja `<maksimum>`.

Kui külastada aadressi `/<miinimum>/<maksimum>`, on piirid `<miinimum>` ja `<maksimum>`. Seda peaks lahendama ainult ühe funktsiooniga.

Vihje: tuleta meelde, kuidas määrata funktsiooni parameetritele vaikeväärtused.

Joonis 3. Veebirakenduste loomise koodiploki näide

Paljudel peatükkidel on läbiv narratiiv: alustatakse lihtsamatest teemadest ja jõutakse välja keeruliseni. Näiteks veebirakenduste peatükis alustatakse lihtsa tervitusprogrammiga, näidatakse ette väga algeline juhusliku arvu generaator ja seejärel lisatakse sinna aina rohkem detaile. Mängude loomise peatükis alustatakse tegelase joonistamisega, pannakse see liikuma, seejärel liigutatakse seda kasutaja sisendiga ja lõpuks lisatakse mängule eesmärk.

Kui peatükk käsitleb teemat, mille rakendamiseks Pythoniga on vaja kõrvaloskusi, koosneb põhiosa kahest osast: kõrvaloskuste tutvustus ning Pythonis nende rakendamise tutvustus. Näiteks regulaaravaldiste peatükis tutvustatakse enne regulaaravaldisi ja seejärel rakendatakse neid Pythoniga. Andmebaaside peatükis tutvustatakse enne SQL-i ja seejärel kasutatakse seda Pythonis.

Põhiosa lõpuks leitakse lahendused peatüki sissejuhatuses püstitatud probleemidele. Näiteks regulaaravaldiste peatükis tuuakse sissejuhatuses näited kolmest ülesandest, mille lahendamisel on regulaaravaldised kasulikud (Joonis 4) ning peatüki jooksul leitakse neist kahele lahendused (Joonised 5 ja 6) ja kolmandale tuuakse analoogne näide, jättes mainitud ülesanne endale lahendamiseks. Standardteegi ja moodulite peatüki sissejuhatuses

maintatakse, kuidas homse kuupäeva kätte saamine võib olla keeruline probleem ning peatüki jooksul näidatakse, kuidas seda mooduli abil lihtsasti lahendada.

3. Regulaaravaldised

Paljud programmeerijatel ette tulevad ülesanded on seotud sõnade ja teksti analüüsiga. Sellised ülesanded võivad osutada üpris keeruliseks. Näiteks, kuidas teha kindlaks, et kasutaja sisestas ikka korrektse e-posti aadressi? Kuidas leida üles tekstist kõik telefoninumbrid? Kuidas [1] eemaldada [5] Vikipeedia [15] artiklist [43] kõik [119] viited [327]? Neid ülesandeid võib ju proovida puhta Pythoniga lahendada hakata, aga selliseid ja keerukamaidki ülesandeid on palju lihtsam ja kiirem lahendada regulaaravaldistega.

Joonis 4. Regulaaravaldiste peatüki sissejuhatus

```
import re

kasutajanimi = input("Sisesta kasutajanimi: ")
kasutajanimi_regex = r"^[a-z0-9_-]{3,20}$"

if re.match(kasutajanimi_regex, kasutajanimi):
    print("See kasutajanimi sobib!")
else:
    print("See kasutajanimi ei sobi.")
```

Millised kasutajanimed programmile meeldivad? Proovi erinevaid variante.

Sarnast kontrolli saab teha e-posti aadressidega, aga muster on keerulisem:

```
# https://emailregex.com/
email_regex = r"^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+$"
```

Proovi muuta programmi nii, et kontrollitakse telefoninumbri korrektsust.

Joonis 5. Lahendus esimesele regulaaravaldiste peatüki sissejuhatuses esitatud probleemile

Mustrite asendamine sõnedes

Funktsiooniga `re.sub()` (sõnast *substitute*) saab asendada tekstis kõik mustrite vastavused:

```
>>> tekst = "koer kass kaer koor koger k03r k??r"
>>> muster = r"k..r"
>>> asendus = "kass"
>>> re.sub(muster, asendus, tekst)
'kass kass kass kass koger kass kass'
```

Selle koodiga asendasime kõik muustrile `"k..r"` vastavad osasõned sõnega `"kass"`.

Kopeerisid Vikipeediast pika artikli, aga teksti sisse jäävad nurksulgudes viited.

```
>>> tekst = "Python was conceived in the late 1980s[34] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL),[35] capable of exception handling and interfacing with the Amoeba operating system.[8] Its implementation began in December 1989.[36]"
```

Nende eemaldamine manuaalselt on tüütu ja neid koodiga eemaldada võib esmapilgul tunduda keeruline, aga regulaaravaldised teevad selle väga lihtsaks:

```
>>> muster = r"\[\d+\]"
>>> re.sub(muster, "", tekst)
'Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.'
```

Joonis 6. Lahendus teisele regulaaravaldiste peatüki sissejuhatuses esitatud probleemile

3.2.2. Enesekontrolliküsimused

Lisamaterjalides on teemade kordamiseks ja enese kontrollimiseks enesekontrolliküsimused. Iga peatüki lõpus on 3 küsimust terve peatüki kohta ning mõned küsimused on peatüki põhiosa sees – enamasti pärast mõnda terviklikku või keerulisemat teemat. Küsimused on valikvastustega ning igal küsimusel on 5 vastusevarianti. Osadel küsimustel saab valida mitu vastusevarianti, osadel saab valida ainult ühe. Joonisel 7 on näide sama peatüki põhiosa sees olevatest küsimustest ning joonisel 8 on näide regulaaravaldiste peatüki lõpus olevatest küsimustest.

Millised sõned sobivad mustrile "ha{3,}"?

- hahahaha
- hahaha
- haaaa
- haahaahaa
- haha

Kontrolli!

Algväärtusta

Millised sõned sobivad mustrile "(no)*(ni)+"?

- nonii
- noniinonii
- niinii
- nononiiinii
- nonono

Kontrolli!

Algväärtusta

Joonis 7. Regulaaravaldiste peatüki põhiosa sees olevad enesekontrolliküsimused

Enesekontrolliküsimused

Mis on vahet sõne meetodil `replace()` ja regulaaravaldiste funktsioonil `sub()`?

- Neil ei ole mingit vahet
- `sub()` töötab kiiremini kui `replace()`
- `sub()` kasutab otsimiseks regulaaravaldisi, `replace()` ainult sõnesid
- `replace()` asendab ainult esimese leitud osa
- `sub()` asendab ainult esimese leitud osa

Kontrolli!

Algväärtusta

Mis funktsioonidega saab regulaaravaldistega kontrollida vastavust, leida kõik vastavused ja asendada kõik vastavused millegi muuga?

- `match()`, `findall()`, `sub()`
- `compare()`, `search()`, `replace()`
- `match()`, `find()`, `replace()`
- `compare()`, `findall()`, `replace()`
- `match()`, `find()`, `sub()`

Kontrolli!

Algväärtusta

Millised sõned sobivad mustriksile `^#?[a-f0-9]{6}|[a-f0-9]{3}$`?

- #2196f3
- bada55
- #cabbage
- #fb1
- 3f51p5

Kontrolli!

Algväärtusta

Joonis 8. Regulaaravaldiste peatüki lõpus olevad enesekontrolliküsimused

3.2.3. Ülesanded

Peatüki lõpus antakse lugejale teemakohased ülesanded, et õpitud rakendada ja seeläbi materjal selgemaks saada.

Tihti on osa peatüki ülesannetest seotud materjalide põhiosa projekti või koodinäite täiendamisega. See aitab lugejal ülesannete lahendamist alustada, sest pool ülesannet on juba lahendatud. Näiteks rakendusliideste peatüki jooksul tuuakse näide valuutakursside kättesaamisest ning üks ülesannetest on kirjutada valuutakalkulaator. Andmebaaside peatükis luuakse tabel kasutajanimede ja paroolidega ning ülesannetes peab looma sarnase tabeli ühe lisatud väljaga ja seda kasutama. Veebirakenduste peatükis luuakse juhusliku arvu generaator ning ülesannetes tehakse selle kasutamine mugavamaks ning lisatakse võimalus genereerida mitu arvu korraga (Joonis 9).

Ülesanded

1. Juhusliku arvu generaatoril lähevad alam- ja ülempiirid nupu vajutamisel tagasi vaikeväärtustele. Täienda rakendust nii, et need säilitavad oma väärtused vastavalt parameetritele.

2. Täienda juhusliku arvu generaatorit nii, et genereerida saab mitu juhuslikku arvu. Selleks tuleb lisada uus aadressiparameeter, mis määrab ära, mitu arvu peab genereerima. Tuleb ka lisada HTML-koodi sisendikast, mis lubab seda parameetrit muuta. Näide:

```
<label for="kogus">Kogus:</label>  
<input id="kogus" name="kogus" type="number" value="1"><br>
```

Joonis 9. Veebirakenduste loomise peatüki esimene ja teine ülesanne

Mõned ülesanded nõuavad loomingulisust: lugeja peab programmi idee ise välja mõtlema ja teemaga seotud nõudeid täitma. Näiteks standardteegi ja moodulite peatükis peab ühe ülesande lahendusena esitama ükskõik millise programmi, mis kasutab asjalikult ära kolme erinevat moodulit. Rakendusliideste peatükis on üks ülesanne kirjutada programm, mis kasutab ühte rakendusliidest. Veebirakenduste peatüki ühe ülesande raames peab looma uue lihtsa veebirakenduse lugeja valitud ideega (Joonis 10).

3. Kirjuta uus kasulik veebirakendus, mis kasutab kasutajasisendit. Mõned ideed:

- Kuupäeva kalkulaator (peatükk "Standardteek ja moodulid")
- Mingist rakendusliidest saadud andmete kuvamine
 - Pokémoni andmed ja pilt - <https://pokeapi.co/>
 - Koerte pildid - <https://dog.ceo/dog-api/documentation/>
 - Kasside faktid - <https://cat-fact.herokuapp.com/facts/random>
- Vikipeedia artiklist viidete eemaldaja (peatükk "Regulaaravaldised")
 - Siin tuleks kasutada `<textarea>` HTML elementi
- Anonüümne foorum
 - Sisestatud tekst lisatakse tekstifaili ja faili sisu kuvatakse veebilehel
 - Veel parem, kui kasutada andmebaasi
 - Siin tuleks kasutada POST-päringut

Soovitav on taaskasutada juhusliku arvu generaatori HTML- ja CSS-koodi.

Joonis 10. Veebirakenduste loomise peatüki kolmas ülesanne

3.3. Materjalide tulevik

Lõputööga valmis lisamaterjalide komplekt, mis on kasulik Programmeerimiskursust läbivatele tudengitele programmeerimisalase projekti realiseerimisel, silmaringi laiendamisel ja informaatika õppekavas järgnevatele ainetele üleminekul. Loodud materjalid lisatakse 2020. aasta sügissemestri Programmeerimiskursuse materjalide hulka keskkonnas

courses.cs.ut.ee. Programmeerimiskursust läbivatel tudengitel saab olema võimalus asendada mõned Programmeerimiskursuse osad lisamaterjalide peatüki läbimisega ja ülesannete esitamisega.

4. Viidatud kirjandus

- [1] Programmeerimine | ÕIS II. <https://ois2.ut.ee/#/courses/LTAT.03.001/details>
(09.04.2020)
- [2] Programmeerimise õpik. <https://progeopik.cs.ut.ee/> (09.04.2020)
- [3] Programmeerimisest maalähedaselt. <https://courses.cs.ut.ee/2019/progmaa/spring>
(11.04.2020)
- [4] MOOC Programmeerimise alused. <https://courses.cs.ut.ee/2018/eprogalused/fall>
(11.04.2020)
- [5] MOOC Programmeerimise alused II. <https://courses.cs.ut.ee/2019/eprogalused2/spring>
(11.04.2020)
- [6] Python for Everybody | Coursera. <https://www.coursera.org/specializations/python>
(11.04.2020)
- [7] Bell A., Grimson E., Guttag J. 6.0001 Introduction to Computer Science and Programming in Python. Fall 2016. Massachusetts Institute of Technology: MIT OpenCourseWare.
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/index.htm> (11.04.2020)
- [8] Learn Python. <https://www.learnpython.org/> (06.12.2019)
- [9] Learn Python 3 | Codecademy. <https://www.codecademy.com/learn/learn-python-3>
(11.04.2020)
- [10] Downey A. Think Python. 2nd Edition. Green Tea Press. 2015.
<http://greenteapress.com/thinkpython2/thinkpython2.pdf>
- [11] Objektorienteeritud programmeerimine | ÕIS II.
<https://ois2.ut.ee/#/courses/LTAT.03.003/details> (12.04.2020)
- [12] Andmebaasid | ÕIS II. <https://ois2.ut.ee/#/courses/LTAT.03.004/details> (12.04.2020)
- [13] Veebirakenduste loomine | ÕIS II. <https://ois2.ut.ee/#/courses/LTAT.05.004/details>
(12.04.2020)

- [14] Arvutigraafika | ÕIS II. <https://ois2.ut.ee/#/courses/MTAT.03.015/details> (12.04.2020)
- [15] Arvutimängude loomine ja disain | ÕIS II.
<https://ois2.ut.ee/#/courses/MTAT.03.263/details> (12.04.2020)
- [16] Automaadid, keeled ja translaatorid | ÕIS II.
<https://ois2.ut.ee/#/courses/LTAT.03.006/details> (12.04.2020)
- [17] Programmeerimiskeeled | ÕIS II. <https://ois2.ut.ee/#/courses/MTAT.03.006/details>
(08.05.2020)
- [18] Requests: HTTP for Humans™. <https://2.python-requests.org/en/master/> (12.04.2020)
- [19] SQLite. <https://sqlite.org/> (12.04.2020)
- [20] sqlite3 - DB-API 2.0 interface for SQLite databases.
<https://docs.python.org/3/library/sqlite3.html> (12.04.2020)
- [21] Flask Documentation (1.1.x). <https://flask.palletsprojects.com/en/1.1.x/> (19.04.2020)
- [22] PythonAnywhere. <https://www.pythonanywhere.com/> (19.04.2020)
- [23] Pygame. <https://www.pygame.org/> (06.05.2020)
- [24] PEP 8 -- Style Guide for Python Code. <https://www.python.org/dev/peps/pep-0008/>
(06.05.2020)
- [25] NumPy. <https://numpy.org/> (06.05.2020)
- [26] pandas - Python Data Analysis Library. <https://pandas.pydata.org/> (06.05.2020)
- [27] Sissejuhatus andmeteadusesse | ÕIS II. <https://ois2.ut.ee/#/courses/LTAT.02.002/details>
(08.05.2020)
- [28] tkinter — Python interface to Tcl/Tk. <https://docs.python.org/3/library/tkinter.html>
(08.05.2020)
- [29] Wikibooks: Textbook considerations.
https://en.wikibooks.org/wiki/Wikibooks:Textbook_considerations (19.04.2020)

Lisad

I. Kaasa pandud materjalid

Kaasa pandud zip-konteiner sisaldab materjale kolmes formaadis:

1. Google Docs keskkonnast eksporditud PDF-dokument
2. Google Docs keskkonnast eksporditud HTML-dokument koos pildifailidega
3. Courses.cs.ut.ee keskkonnale ettevalmistatud PmWiki-vormingus tekstifailid koos pildifailidega ja interaktiivsete enesekontrolliküsimustega

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Andri Soone**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Programmeerimise kursusele lisamaterjalide koostamine**, mille juhendaja on **Reimo Palm**, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 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.

Andri Soone

08.05.2020