

Tartu Ülikool
Sotsiaalteaduste valdkond
Narva kolledž
Õppekava Infotehnoloogiliste süsteemide arendus

Ekaterina Sidorova

**MÄNGULISTE ÜLESANNETEGA PYTHONI
EESTIKEELNE ALGKURSUS**

Bakalaurusetöö

Juhendaja: Deniss Ruder

NARVA 2023

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, _____ Ekaterina Sidorova _____,
(*autori nimi*)

annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

_____ Mänguliste ülesannetega Pythoni eestikeelne algkursus _____,
(*lõputöö pealkiri*)

mille juhendaja on _____ Deniss Ruder _____,
(*juhendaja nimi*)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

autori nimi Ekaterina Sidorova

pp.kk.aaaa 12.05.2023

RESÜMEE

This work is a diploma thesis titled "Mänguliste ülesannetega Pythoni eestikeelne algkursus". The aim of the work is to create a programming course in Python for beginner high school students, which would focus on introducing students to various industries and companies that use Python in their work, teach them the basics of programming, and provide practical interactive tasks to complete.

Within the scope of the work, was developed a course of 12 lessons, 10 of which are practical. For each lesson was prepared a lesson plan, topics to be learnt at home, lesson structure (detailed lesson plan, and the introduction of a company in each), and practical tasks. For each practical task was written a detailed guide that would be understandable for students and other teachers. For each task was provided a solution. The created practical tasks are interactive, where user interacts with the program. Tasks are created in different ways (games, chatbots, robots, travel planners, flight visualizers, etc.). Each task demonstrates the work of a company and practices the Python topics covered in the lesson.

The use of interactive and non-standard teaching methods, the introduction to different fields where Python is used, its use cases, and learning Python through interactive and playful tasks should increase students' motivation to learn IT. The created course differs from existing solutions, and a school course in Estonian is relevant due to schools transitioning to Estonian-language education.

SISUKORD

| | |
|---|----|
| SISUKORD | 5 |
| SISSEJUHATUS | 7 |
| 1. PROBLEEM..... | 8 |
| 2. ÕPITAVA PROGRAMMEERIMISKEELE VALIK | 10 |
| 3. OLEMASOLEVAD ÕPPEKAVAD | 11 |
| 4. METOODIKA | 13 |
| 4.1 Interaktiivsed meetodid | 13 |
| 5. KOOLITUSE FORMAAT KOOLILE..... | 16 |
| 6. LAHENDUSE VALIK..... | 17 |
| 7. PYTHONI KASUTATAVAD VALDKONNAD | 19 |
| 8. TEEMAD PYTHONI ÕPPIMISEKS..... | 22 |
| 9. KURSUSE STRUKTUUR..... | 23 |
| 9.1 Üldinfo kursuse kohta..... | 23 |
| 9.1.2 Eesmärgid | 23 |
| 9.1.3 Õpiväljundid | 23 |
| 9.2 Tund 1. Sissejuhatus | 24 |
| 9.3 Tund 2. Sissejuhatus Pythoni ja muutujatesse. Google | 24 |
| 9.4 Tund 3. Loogilised avaldised, tingimuslause ja tsüklid. Game Dev | 27 |
| 9.5 Tund 4. Sissejuhatus funktsioonidesse ja nende kasutusviisidesse. Tesla | 30 |
| 9.6 Tund 5. Faili sisend ja väljund Pythonis. JPMorgan Chase | 33 |
| 9.7 Tund 6. Sissejuhatus täiustatud andmetüüpidesse. Järjendid. Spotify | 37 |
| 9.8 Tund 7. Sõnastikud ja ennikud. Amazon..... | 40 |
| 9.9 Tund 8. Sissejuhatus objektorienteeritud programmeerimisse. Robotika | 43 |
| 9.10 Tund 9. Objektorienteeritud programmeerimine. Klassid ja objektid. Instagram... .. | 47 |
| 9.11 Tund 10. Graafilise kasutajaliidese (GUI) programmeerimine Tkinteriga. | |
| Tervishoid - Nortal | 50 |

| | |
|---|----|
| 9.12 Tund 11. Sissejuhatus PyGame'i. Boeing..... | 52 |
| 9.13 Tund 12. Hinnete parandamine. Tagasiside | 55 |
| KOKKUVÕTE | 56 |
| KURSUSE TULEVIK..... | 59 |
| KASUTATUD KIRJANDUS | 60 |

SISSEJUHATUS

IT valdkond on praegu aktuaalne ja perspektiivne valik, spetsialistidele on avatud paljud tööpakkumised, mis võimaldavad huvitavaid projekte luua, kaugtöö teha, paremat palka teenida, jne. IT on rohkem ja rohkem seotud teiste valdkondadega. Näiteks inseneeria (toodete ja seadmete modelleerimisel), disain (UX-disainis, UI-disainis), arhitektuur (ehitiste ja rajatiste kavandamisel, modelleerimisel), tervishoiu valdkond (masinõppes, tehisintellekti arendamisel, ravimite tootmisel, andmetega töös jm.), haridus (õppetarkvara ja –materjalide luues, kaugõppes), turundus (erinevatel viisidel, kodulehtedel, reklaamis jm.) ning finantsid (analüüsis, panganduse valdkonnas, krüptoraha tehnoloogiates jm.) on mõned valdkonnad, mille töö on tihedalt seotud IT-ga. Spetsialist, kes omab IT-teadmisi ja oskab neid mõnes teises valdkonnas rakendada, on tööturul kahekordselt hinnatud.

IT-valdkonnas suurema nõudluse tekkimiseks on vaja teha otsust koostööd õppevaldkonna valijatega. Eriala esmane valik tehakse juba põhikoolis või gümnaasiumis. Väga oluline on teha seda põhjalikult ja arusaadavalt, kuhu edasi liikuda, ning kuidas see elus aitab. Erinevate stereotüüpide, ebapiisava informatsiooni, motivatsiooni ja eriala tundmise tõttu võivad tekkida sellised probleemid, et koolilõpetajad ei vali IT-valdkonda ning naisspetsialiste jääb vähemusse (de Wit, 2021).

On erinevaid haridusprogramme, mis soodustavad gümnaasiumiõpilaste karjäärivalikut ja julgustavad neid tegema üht või teist valikut (Rocker et al., 2022). Praktika näitab, et mõne protsent lõpetajatest ei näe ikka veel helget eluteed, ei kujuta ette oma tööalast karjääri, ei oska adekvaatselt hinnata oma erialaseid väljavaateid (Polishchuk, 2020). Hea lahendus oleks, sissejuhatavad kursused, mis toimuksid koolides valikainetena ja motiveeriksid õpilasi valima sellist suunda nagu IT, samuti näitaksid praktilisi reaalseid ülesandeid, mis paneksid paremini valdkonna konteksti ja näitaksid erinevaid võimalusi tulevikus.

1. PROBLEEM

Tulevase elukutse valik tuleb enne koolilõpetamist ja siis on oluline aidata õpilast tema valikul, tutvustada talle erinevaid valdkondi ja võimalusi, tööülesandeid. Peamine koht, mis saab sellele kaasa aidata, on kool.

Programmis “Infotehnoloogiliste süsteemide arendus” õppivate Narva kolledži üliõpilaste sõnul ajendas enamikku seda õppekava valima enda huvi IT-valdkonna ja selle arengu vastu. Kooli sissejuhatavad kursused motiveerisid vaid väga väikest osa õpilastest.

Ligikaudu 80% vastajatest said IT-valdkond valides aru, mis suunas nad liiguvad, omasid ettekujutust tööülesannetest, võimalikest tööstusharudest jne. Osa üliõpilasi õppis palju uut alles ülikoolis. Nagu uuringust selgus, kujunes õpilastel IT-valdkonnast ja programmeerimisest terviklikum arusaam tänu enda huvile ja ülikoolile. Ükski küsitletud õpilastest ei valinud selliseid valikuid nagu kool, sõbrad, perekond ja lisakursused.

Programmeerimise või muid IT-alaseid kursusi koolis ei olnud 44% õpilastest, mis on ligi pooled vastanutest. Koolis programmeerimiskursused läbinutest vastas 50%, et need kursused ei motiveerinud neid selles valdkonnas edasi õppima, nende huvi ei suurenenud. 33% vastanutest arvas, et kursused motiveerisid vaid osaliselt. Seega oli vaid 17% küsitletud õpilastest motiveeritud pärast kooliprogrammi läbimist IT-valdkonnas edasi õppima. See näitab, et koolides kas puuduvad ained, mis on mõeldud õpilasele tarkvara arendaja eriala tutvustamiseks või pole olemasolevad programmid õpilastele piisavalt huvitavad.

50% koolis programmeerimiskursustel käinud õpilastest vastas, et kursused toimusid eesti keeles, teine pool vastas, et kursused olid vene keeles.

2024. aastal plaanitakse Eestis üleminek eestikeelsele haridusele, mis algab lasteaedadest ja 1. ja 4. klassidest ning plaani järgi kogu õppesüsteem on üle läinud eesti keelele 2030. aastaks. 2022. aastal võttis Riigikogu selle alusel seadusemuudatused vastu. Riik on huvitatud sellest, et suurendada kõigil õppetasemetel eestikeelse õppe mahtu (Haridus- ja Teadusministeerium veebileht). Koolid püüdlavad ka selle poole, et haridusasutustes mindaks eesti keelele üle.

Seega suureneb vajadus eestikeelse programmeerimise koolituse järele. Praegu ei suuda kõik koolid eesti keele kursust pakkuda.

Kõik küsitatud üliõpilased üksmeelselt vastasid, et koolis on vajalik programmeerimise sissejuhatav kursus, mis tutvustab praktilisi ülesandeid ja annab parema ülevaate selle rakendusvaldkondadest.

Läbiviidud küsitluse põhjal võib öelda, et koolinoored on huvitatud tulevase elukutse valikust ja IT valdkond tundub neile atraktiivne, kuid neil puudub piisav teadlikkus teadmiste praktilisest rakendamisest tulevikus, samuti praktiline kogemus, mis võiks motiveerida IT-valdkonda valima ja äratada huvi selle valdkonna vastu.

Sellest tulenev probleem on: koolidel ja gümnaasiumidel puuduvad sissejuhatavad programmeerimis kursused, kus antakse huvitavaid ülesandeid praktikas lahendamiseks, mis motiveeriks üliõpilasi IT-spetsialisti karjääri valima.

2. ÕPITAVA PROGRAMMEERIMISKEELE VALIK

Algajatele mõeldud kooli algkursuse läbiviimiseks valiti Python. Python on avatud lähtekoodiga skriptimiskeel, mis on muutunud USA tippülikoolide populaarseimaks sissejuhatavaks õppekeeleks. Võrreldes teiste programmeerimiskeeltega võtab Pythoni programmide arendamine oluliselt vähem aega ja need on tavaliselt lühemad kui samaväärsed Javas kirjutatud programmid. Pythonil on väga suur hulk väga kasulikke teeke ja neid on algajale lihtsam mõista (Shein, 2015). Lihtsam õppekeel võib meelitada huvitatud teismelisi kodeerimise õppimise väga varases staadiumis.

Teised laialdaselt kasutatavad programmeerimiskeeled on Java ja C++. Üks peamisi erinevusi nende ja Pythoni vahel on see, et Pythoni programmid on aeglasemad, kuid Python Software Foundationi andmetel kulub nende arendamiseks palju vähem aega ja koodiridu. Võrreldes Javaga, mis kasutab keerulisi süntaksi ja keerulist staatilist semantikat, on Python mugavam neile, kes kirjutavad väikesi programme (Shein, 2015).

Guttag leiab, et teine põhjus, miks Pythonit algajate õpetamiseks kasutatakse, on kasulike sisseehitatud teekide suur arv (Guttag, 2013).

Samas kõik õpetajad ei arva, et Python on hea programmeerimiskeel algajatele. Tööriistade puudumise tõttu mõned hästi struktureeritud andmeid nõudvad ülesanded muutuvad keerulisemaks ja võtavad õpilastel rohkem aega ning testimiseks ei pruugi tööriistu jätkuda. Mõnikord on Pythonis kirjutatud koodi süntaksi suurte erinevuste tõttu raske teistesse keeltesse portida.

Mõned teadlased ei pea Pythonit tõsiselt võetavaks programmeerimiskeeleks (Java on suurtes korporatsioonides endiselt levinud), kuid on hea valik neile, kellel pole varasemat programmeerimiskogemust. Python on alustuseks hea keel (Shein, 2015).

Python annab hea ülevaate automatiseerimise toimimisest, koodi kasutamisest tööülesannete täitmiseks. Just seda on vaja sissejuhatavaks kursuseks neile, kellel pole praktilist programmeerimise kogemust. Tulevikus saab Pythonit kasutada ka koos teiste programmeerimiskeeltega.

3. OLEMASOLEVAD ÕPPEKAVAD

Koolitusprogrammi paremaks koostamiseks on vaja võrrelda olemasolevaid Pythoni koolituskursusi algajatele.

Üks peamisi Pythoni programmeerimise kursusi on Tartu Ülikooli kursus "Programmeerimine" (SVNC.00.054). Kursuse aluseks on TÜ Arvutiteaduse instituudi programmeerimise algkursuse õpik. Selle kursuse eesmärk on õpetada Pythoni keele abil programmeerimist algtasemel. Kursuse läbinud üliõpilane peaks olema kursis põhiliste programmeerimiskonstruktsioonide, põhistruktuuride ja andmetüüpidega. Õpilane oskab omandatud teadmisi rakendada, programme kirjutada ja meeskonnas läbi viia õppeprojekte. See koolituskursus sisaldab 156 tundi koolitust (6 EAP), millest 32 tundi on praktikume ja seminare (Tartu Ülikooli ÕISI veebileht). Kursusel kasutatava elektroonilise õpikuga tutvudes võib öelda, et kursus on üsna mahukas ja hõlmab nii lihtsamaid kui ka keerulisemaid teemasid. See on jagatud kolmeks osaks, mis hõlmavad Pythoni keele tunnuseid (avaldised, funktsioonid, lihtlauseid jne), keerukamaid tehnilisi struktuure (algoritmid, andmestruktuurid, rekursioon jne) ja lisafunktsioone (moodulid, teegid, lisapaketid jne) (Tartu Ülikooli arvutiteaduse instituut veebileht).

See kursus on küllaltki mahukas ja võrreldes nõutava lahendusega hõlmab keerulisemaid teemasid, mida koolikursusesse panna ei saa. Samuti on see koolitus suunatud neile, kes on juba süvitsi otsustanud programmeerimisega tegeleda, ja annab hea baasi, kuid ei ole keskendunud huvi tekitamisele ja programmeerimise valdkonna tutvustamisele ebatavaliste ülesannete kaudu.

Teine Tartu Ülikooli pakutav kursus on "Programmeerimine koolis (Python)", (MTAT.03.302). Kursuse eesmärk on õpetada õpilasi infot otsima ja õpetama. Kursusel selgub, kuidas Pythonit koolides programmeerimise õpetamisel kõige paremini kasutada. Kursuse läbinud õpilane teab, kuidas programmi õigesti kirjutada. Kursuse programm eeldab, et töös kooliõpilastega kaasatakse koolitusse sellised teemad nagu Thonny, Kilpkonnagraafika, PyGame (Tartu Ülikooli ÕISI veebileht). Neid andmeid saab kasutada algajatele õpilastele mõeldud programmi koostamisel.

Tartu Ülikool teeb koostööd Coursera platvormiga, mis pakub erinevaid õppekursusi. Üheks selliseks kursuseks on partner Meta "Programmeerimine Pythonis". Kursus vastab

algtasemele ja selle eesmärk on anda põhilised programmeerimisoskused. Kursuse programm on kõrgelt hinnatud ja sisaldab põhimõisteid, põhilisi süntakseid, funktsioone, veakäsitlust, OOP-i, algoritme, aga ka pakette ja teeke, mida saab oma töös kasutada. Kursusel on palju positiivset tagasisidet, mis näitab, et see on väga kasulik, rikkalik ja informatiivne (Coursera veebileht).

Sellel põhjal on see kursus suunatud neile, kes kavatsevad Pythonit täielikult õppida, see sobib isegi neile, kes on programmeerimisega tegelema pikka aega. Vastupidiselt kooli vajadusele näidata õpilastele programmeerimise ja valdkonna huvitavat poolt, kasutades Pythoni keelt.

Teine Pythoni kursus, mida pakub Ida-Virumaa Kutsehariduskeskus pakub, on "Programmeerimine koolis (Python)", mis on mõeldud Pythonit õppida soovijatele. See kursus on huvitav, kuna see on kooliformaadile lähedasem. Selle kursuse programm sisaldab põhifunktsioonide, operatsioonide, andmetüüpide, testimise, OOP-i põhimõistete ja Pygame mooduli õppimist (Kutsehariduskeskuse veebileht).

See kursus näitab hästi, kuidas programmeerimise põhiprintsiipe õppekavas tutvustada ilma keele süvaõppeta, sobitades selle 60-tunnise koolikursuse mahuks.

Õppides erinevaid kursusi Coursera platvormil ja Eestis pakutavatel kursustel, võib öelda, et need on suunatud neile, kes soovivad õppida Pythonit, kuid puudub kursus, mis kasutaks Pythonit IT-valdkonna ülesannete lahendamiseks, mis on suunatud kooliõpilastele ning mille eesmärgiks on tutvuda IT-valdkonna võimalusi, anda ülevaade erinevatest valikutest ja motiveerida õppima selles valdkonnas.

4. METOODIKA

Eesmärkide saavutamiseks on vaja valida koolituse läbiviimise meetod. Erinevad meetodid aitavad kaasa tulemuste saavutamisele, bakalaaurusetöö on eesmärgiks õpilaste kaasamine ja huvi tekitamine, samuti ebastandardsete interaktiivsete ülesannete esitamine, mis erinevad tavapärasest õppemeetodist. Kursus on keskendunud praktilistele ülesannetele, mis on parim programmeerimise õpetamise meetod, sisaldades ainult vajalikku algteoreetilise osa.

4.1 Interaktiivsed meetodid

21. sajandi lapsed kasvavad üles keskkonnas, mis on täis erinevat tehnoloogiat, nad on harjunud saama vajalikku teavet info- ja kommunikatsioonitehnoloogia abil, õpilastena on neil teatud ootused, et selline olukord jätkub haridusasutuses (Topalska & Mladenova, 2020). Bakhreddinova kirjutab, et interaktiivsed õppemeetodid sisaldavad palju tõhusaid tegevusi ja lähenemisviise, mis on suunatud õpetamisprotsessi kiirendamisele ja edu saavutamisele (Bakhreddinova, 2020, lk 77). Positiivseteks külgedeks, mida õpetajad interaktiivsete meetodite lisamisel õppetöösse näevad, on: õpilaste tunnetusliku tegevuse aktiveerimine, õpilaste huvi ja motivatsiooni tõstmine ning õpilaste enesekindluse arendamine (Topalska & Mladenova, 2020).

Interaktiivsed meetodid eeldavad õpilaste aktiivset kaasamist ja osalemist, mis suurendab nende huvi. Programmeerimise õpetamisel võivad interaktiivsed meetodid olla (ChatGPT veebileht):

- Paarisprogrammeerimine.

Paarisprogrammeerimisel jagavad kaks arendajat ühte arvutit, et kirjutada kokku üks koodijupp. Suurtes ettevõtetes toimub töö meeskonnas ning oskus üheskoos probleemi lahendada võib töös palju kasuks tulla. Paarisprogrammeerimise olemus seisneb selles, et üks meeskonnast prindib koodi ja teine jälgib toimuvat, kus peamine on sage rollide vahetamine. Paarisprogrammeerimise eelised on parem koodikvaliteet, parem meeskonnatöö, paremad oskused iga meeskonnaliikme jaoks, vähem vigu ja lihtsam kood. Nawahdah ja teised leiavad, et eriti oluline on esmakursuslaste ja algajate jaoks saada abi õpingutes ning integreeruda keskkonda, mida hõlbustab paarisprogrammeerimine. Nad viisid läbi uuringu, milles osa õpilasi õppis aasta

jooksul standardisel viisil, osa aga paaris. Uuringu tulemused näitavad, et paarides kirjutatud kood oli lühem, paaris laboritöö tulemused ja kursuse üldised tulemused paremad, ning õpilastega suheldes selgus, et paaris töötavad õpilased olid motiveeritumad, tundsid end enesekindlamalt ja said laboritööst rohkem naudingut (Nawahdah et al., 2015). Paarisprogrammeerimine võib olla hea õpetamismeetod programmeerimise sissejuhatava kursuse jaoks, kuna see võib tõsta õpilaste enesekindlust oma töös, suurendada nende efektiivsust ja huvi, mis tuleb hiljem kasuks kursuse jooksul. Paarisprogrammeerimine hariduses võib aidata saavutada töö eesmärgi.

- Koodiarutelu.

Õpilased arutavad ühiselt üksteise koodi, annavad tagasisidet ja aitavad erinevate lähenemisviiside kaudu lahendust täiustada. See võib aidata parandada nii koostöö- kui ka kodeerimisoskusi.

- Mängustamine.

Arvutimängud moodustavad olulise osa laste ja noorukite jaoks harjunud tehnoloogiast ning neid saab kasutada ka hariduses, et aidata arendada uusi oskusi. Need äratavad noortes huvi, arvutist saab vastane, kes reageerib kirjutatud käsule. Treeningu käigus luuakse kindlat programmeerimiskeelt kasutades mäng või sooritatakse mänguväljakutseid.

Õppetöös saab paluda õpilastel ellu viia selliseid projekte nagu kaardimängud, klassikalised arvutis realiseeritud lauamängud, mängulised ülesanded (Stoffova & Horvath, 2021).

- Projektipõhine lähenemine.

Projektipõhine õpe hõlmab koolituse käigus keskendumist konkreetsele projektile ning elluviimiseks kasutatakse valitud tööriista/programmeerimiskeelt. Seda õpetehnikat kasutades lahendavad õpilased probleemi praktikas, keskendudes konkreetsele projektile (Stoffova & Horvath, 2021).

- Ümberpööratud klassiruum

Õpilastele pakutakse mitmesuguseid materjale, õppevideoid ja harjutusi, samas kui klassiruumis toimuvaid tunde kasutatakse projektide ja harjutusülesannete täitmiseks.

- Interaktiivsed programmeerimiskeskonnad

Õpetaja kasutab õpetamisel interaktiivseid õpikeskkondi ja veebiplatvorme, mis pakuvad juhendeid ja interaktiivseid harjutusi. Pythoni õppimiseks on need Jupyter Notebook, PyCharm Edu, Thonny, Codecademy, Trinket, Repl.it.

5. KOOLITUSE FORMAAT KOOLILE

Kuna käesolevas lõputöös koostatav koolituskursus on mõeldud kooliõpilastele ning sellise kursuse läbiviimise koht ja formaat on kool, siis on mitmeid piiravaid tegureid, mida tuleb arvesse võtta.

Seda kursust saab pakkuda valikainena gümnaasiumiõpilastele (10.-12. klass), kuna ülikooli bakalaureuseõppesse astumine eeldab 12. klassi läbimist. See kursus võib pakkuda huvi neile, kes valivad ülikoolis täiendõppe.

Gümnaasiumi õppekava koostatakse 35 tundi kestvate kursuste vormis (Riigiteataja veebileht). See programmeerimiskursus on 35 akadeemilist tundi, mis sisaldab nii tunde koolis kui ka iseseisvat tööd/kodutööd.

Tavaline tunni kestus koolis on 45 minutit, millest ei pruugi piisata täisväärtusliku programmeerimistunni läbiviimiseks. Tõhusamateks tundideks on vaja pikemaid tunde, näiteks 90 minutit kestvaid topelt tunde. Samuti on üheks lahenduseks sellise õpetamismetoodika valik, mis võimaldab kõige tõhusamalt läbida materjali koolis, jagades iseseisva töö ja loengute mahud.

6. LAHENDUSE VALIK

Töö eesmärk on pakkuda gümnaasiumiõpilastele alternatiivset sissejuhatavat kursust, mille teoreetiline osa põhineb olemasolevatel kursustel. Kuna kõiki teemasid ühes koolikursuses ei ole võimalik läbida, keskendutakse kursusel erinevate IT-spetsialistide ülesandeid kajastavate teemade selgitamisele interaktiivsete ülesannete kaudu. Esialgseks õppeks valitud programmeerimiskeel on Python.

Kursus on mõeldud algajatele, kes pole programmeerimisega veel kursis või on selle õppimise alguses, kuid edaspidi saavad nad selle suuna valida ja saada selle ala spetsialistiks. Lõputöö on keskendunud praktilistele interaktiivsete ülesannete loomisele ja teoreetilistele teemadele viisil, mis õpilasi rohkem kaasaks.

Sissejuhatava kursuse läbiviimise meetodiks valiti ümberpööratud klassiruum. See annab õpilastele võimaluse kodutööna teoreetilise osaga iseseisvalt tutvuda. Koolitundide aja saab pühendada praktiliste ülesannete täitmisele, mis kinnistavad kodus õpitavat materjali. Praktilised ülesanded nõuavad õpetaja abi ja neid on raske iseseisvalt täita. Koolitunnid on suunatud teema kordamisele ja kinnistamisele, õpitud teadmiste rakendamisele.

Tundides antakse õpilastele täita erinevaid ülesandeid, mis demonstreerivad Pythoni programmeerimiskeelt kasutavate reaalsete tööstusharude ülesandeid (või nende osi). Nende ülesannete käigus käsitletakse põhiteemasid, mida pakutakse sarnastel olemasolevatel programmeerimise algkursustel.

Tundide ülesanded tehakse paarisprogrammeerimise abil. Paarisprogrammeerimine on hea tehnika õpilaste huvi hoidmiseks, suurendades nende enesekindlust ja motivatsiooni.

Alguses sooritatakse ülesanne paaris ning õpetaja pakub välja omapoolse lahenduse. Pärast seda arutatakse lahendus kogu klassiga läbi ja koos õpetajaga viiakse läbi, et mõista parimat lahendusalgoritmi. See aitab õpilastel vaadata lahendust erinevate nurkade alt.

Projektipõhine meetod sellise kursuse läbiviimiseks ei sobi, kuna igas tunnis esitatakse õpilastele uus ülesanne. Rõhk on erinevate tööstusharude ülesannetel, milles programmeerimist saab kasutada.

Kontroll toimub nii, et paarisülesannet lahendades õpetaja abistab, ärgitab ja vastab küsimustele. Pärast antud aja lõppemist toimub üldine arutelu, selle aja jooksul õpilased saavad muudatusi teha. Iga tunni lõpus vaatab õpetaja lahendust, esitab küsimusi programmi toimimise kohta, võib paluda teha mõningaid muudatusi, et kontrollida koodist arusaamist. Küsimustele vastamise ja lahenduse leidmise eest saab õpilane punkti. Kursuse hinne sõltub kogutud punktidest. Seega on õpilased rohkem motiveeritud tundides osalema, kuna tundides osalemine mõjutab hindamist. Kursuse lõpus toimub tund, kus saab oma hinnet parandada lisapunkte kogudes - iseseisvalt ülesandeid täites.

Materjali valdamiseks sobivad kõige paremini interaktiivsed mängulised ülesanded, mis hõlmavad graafilisi teekes, mänguelemente, kasutajaga suhtlemist, mänguülesandeid, kindla teemaga seotud lihtsate klassikaliste mängude loomist jne.

7. PYTHONI KASUTATAVAD VALDKONNAD

Kursuse koostamisel tuleb välja selgitada valdkonnad või tuntud ettevõtted, kes oma töös Pythonit kasutavad. Neid valdkondi tutvustatakse tundides ning nendega seostatakse praktilisi ülesandeid. See annab õpilastele arusaama sellest, millised ettevõtted võivad olla nende potentsiaalsed tööandjad ning kus ja kuidas programmeerimist rakendatakse.

Selle kursuse jaoks on esitluseks valitud järgmised teemad:

- **Google**
Google on üks suurimaid tehnoloogiaettevõtteid, kes kasutab oma töös teiste keelte hulgas aktiivselt ka Pythonit. Google'il on erinevaid teenuseid, seega tuleb lahendada otsingumootori, tehisintellekti, masinõppe ja isegi robotikaprojektid (Towards Data Science veebileht).
- **Instagram**
Instagram on tuntud sotsiaalvõrgustik fotode ja videote jagamiseks, mis kasutab oma töös aktiivselt Pythoni raamistikku – Gjangot. Pythoniga tehtavad ülesanded on pildi- ja videotöötlus, soovitusüsteemid, sisu modereerimine, rämpsposti konto tuvastamine jne. (Towards Data Science veebileht).
- **Spotify**
Spotify on muusika voogedastusrakendus, kus Pythonit kasutatakse andmete analüüsiks, siseteenuste suhtluseks, samuti põhiliseks ja loominguks signaalitöötluseks, helituvastuseks ja muuks. Selleks kasutatakse teeki nagu Pyo, pyAudioAnalysis, Dejavu jt (Towardsdatascience veebileht). Soovitusüsteemis kasutatakse masinõppe algoritmide jaoks ka Pythoni teeki, nagu Scikit-learn ja TensorFlow (LinkedIn veebileht).
- **Tesla**
Pythonit kasutatakse ka autotööstuses. Siin võivad ülesanded olla: autonoomne sõit, käitumise ennustamine, kvaliteedikontroll. Maailmakuulus ettevõtte Tesla kasutab autopiloodisüsteemis Pythonit auto andurite ja kaamerate juhtimiseks, kasutades oma töös PyTorch'i teeki (HashDork veebileht).
- **JPMorgan Chase**
Pythonit kasutatakse rahanduses, algoritmilises kauplemises, kvantitatiivses analüüsis, riskijuhtimises, majandusproгноosides ja mujal. Üks suurimaid Pythonit rahanduses kasutavaid ettevõtteid on JPMorgan Chase. Näiteks kasutatakse andmete

analüüsimiseks ja kauplemissignaali genereerimiseks teke nagu NumPy ja Pandas, aga ka masinõpet (Chen & Hsu, 2019).

- Boeing

Pythonit kasutatakse lennunduses selliste ülesannete täitmiseks nagu lennujuhtimistarkvara, mis sisaldab andmete analüüsi, modelleerimist, visualiseerimist ja palju muud. Tööriistad, mida ülesannete täitmiseks kasutatakse, on NumPy, SciPy, Matplotlib. Pythonit kasutatakse lennukite hoolduses, näiteks pardaandurites ja ennustavas hoolduses. Boeing kasutab Pythonit ka lennusiimulatsioonide, lennumudelite ja stsenaariumide jaoks (Bissell & Fox, 2017).

- Amazon

Suur veebiturg Amazon kasutab Pythonit oma platvormi mitmes valdkonnas, nagu soovitusüsteem, tehisintellekt, masinõpe. Süsteem analüüsib kasutajate oste ja harjumusi, mille põhjal soovib neile tooteid. Nii suure andmehulga töötlemiseks kasutatakse Pythonit, nimelt Hadoopi tööriista, Jupyteri märkmikke jne. (Towards Data Science veebileht).

- Mänguarendus

Pythonit kasutatakse sageli 2D- ja 3D-mängude arendamisel, selleks on keel varustatud kasulike teekidega PyGame, Pycap, Construct, Panda3D, PySoy, PyOpenGL. Pythonit kasutatakse aktiivselt paljudele tuntud moodsate mängude arendamiseks, nagu Sims 4, World of Tanks, Civilization IV ja Eve Online (Towardsdatascience veebileht).

- Tervishoid - Nortal

Pythonit kasutatakse ka tervishoius selliste ülesannete täitmiseks nagu prognoositavate patsientide tulemuste mudelite väljatöötamine, mis aitavad arstidel teha paremaid otsuseid patsiendihoolduse kohta, või patsiendiandmete ja haiguslugude analüüsimine, et saada teavet, mis võib parandada patsiendi ravi. Eestis on üheks tervishoiusektorile spetsialiseerunud ettevõtteks Nortal (Nortal veebileht).

- Robotika

Üks tuntud Pythonit robotikas kasutatav Eesti ettevõtte on Starship Technologies, mis toodab autonoomseid kohaletoimetamisrobotiteid. Starship Technologies kasutab Pythonit oma robotite, sealhulgas kaamerate, ultraheliandurite ja GPS-i andurite andmete töötlemiseks. Samuti kasutavad nad Pythonit arvutinägemissüsteemide ehitamiseks, mis võimaldavad nende robotitel linnatänavatel navigeerimisel ära

tunda ja vältida takistusi, nagu jalakäijad ja sõidukid (Starship Technologies veebileht).

8. TEEMAD PYTHONI ÕPPIMISEKS

Kursuse läbiviimiseks tuleb valida teemad, mis õpetavad põhilisi Pythoni programmeerimise oskusi. Järgnevad teemad on valitud olemasolevate algajatele mõeldud kursuste põhjal. Kursuse struktuur on kombinatsioon erinevate allikate materjalidest, mis oli isiklikult koostatud. Tehnilise sisu ja Pythoni teemade avamiseks õppetundides kasutati õpikut Programmeerimise õpik (<https://progeopik.cs.ut.ee/index.html>), Pythoni ametlikku dokumentatsiooni (<https://docs.python.org/3/contents.html>) ning TTÜ Küberneetikainstituudi materjale (<https://www.ioc.ee/~dima/YFX0500/>).

Loodud kursus katab järgmised teemad:

- Muutujad ja andmetüübid (sõned (stringid), täisarvud, ujukomaarvud)
- Põhilised operatsioonid (aritmeetika, võrdlus, loogilised operatsioonid)
- Tingimus- ja korduslaused
- Võrdlusoperaatorid (>, <, >=, <=, ==, !=)
- Boole'i loogika (ja, või, mitte)
- While-lause ja for-tsükkel
- range() funktsioon
- Break- ja continue-avaldused
- Järjendite ja ennikute (tuple'd) loomine ja neile juurdepääs
- Järjendite meetodid (append, insert, remove, pop)
- Sõnastike ja kogumite loomine ja neile juurdepääs
- Sõnastiku meetodid (keys, values, items)
- Set-operatsioonid (union, intersection, difference)
- Funktsioonide defineerimine ja kutsumine
- Funktsiooni argumendid ja tagastusväärtused
- Moodulite importimine ja kasutamine
- Klassid ja objektid
- Meetodid ja atribuudid
- Pärilikkus, polümorfism ja inkapsulatsioon
- Faili sisestamine ja väljastamine
- Tekstifailide lugemine ja kirjutamine
- Graafilise kasutajaliidese (GUI) programmeerimine Tkinteriga
- Sissejuhatus Pygame'i

9. KURSUSE STRUKTUUR

See kursus koosneb 12 kahekordsest õppetunnist, igaüks 90 minutit, kus esimene tund on sissejuhataja ja viimane on mõeldud hinnete parandamiseks neile, kes seda vajavad. Praktiliste ülesannete täitmiseks on aega 10 õppetundi. Teoreetilise materjali kodus õppimiseks on ette nähtud 1 akadeemiline tund, mis võib õpilase puhul individuaalselt erineda.

1. Sissejuhatus.
2. Sissejuhatus Pythoni ja muutujatesse. Google.
3. Loogilised avaldised, tingimuslause ja tsüklid. Game Dev.
4. Sissejuhatus funktsioonidesse ja nende kasutamisse. Tesla.
5. Faili sisend ja väljund Pythonis. JPMorgan Chase.
6. Sissejuhatus täiustatud andmetüüpidesse. Järjendid. Spotify.
7. Sõnastikud ja ennikud. Amazon.
8. Sissejuhatus objektorienteeritud programmeerimisse. Robotika.
9. Objektorienteeritud programmeerimine. Klassid ja objektid. Instagram.
10. Graafilise kasutajaliidese (GUI) programmeerimine Tkinteriga. Tervishoid - Nortal.
11. Sissejuhatus PyGame'i. Boeing.
12. Hinnete parandamine. Tagasiside.

9.1 Üldinfo kursuse kohta

9.1.2 Eesmärgid

- Tutvustada õpilastele Pythoni programmeerimiskeelt põhitõdesid.
- Tutvustada õpilastele erinevaid tööstusharusid, kus Pythonit kasutatakse.
- Pakkuda õpilastele mängulisi ülesandeid harjutamiseks.
- Suurendada õpilaste huvi programmeerimise vastu läbi mängutegevuste ja interaktiivsete õppemeetodite.

9.1.3 Õpiväljundid

Kursuse lõpus õpilane:

- Mõistab Pythoni programmeerimiskeele põhitõdesid.
- Kirjutab lihtsaid Pythoni programme.
- Tunneb erinevaid Pythoni teekondi ja mooduleid.

- Mõistab objektorienteeritud programmeerimise põhimõtteid.
- Tunneb erinevaid valdkondi ja tuntud ettevõtteid, kus Pythonit kasutatakse.
- Saab aru, milliseid ülesandeid Pythoniga lahendatakse.
- Oskab paariprogrammeerimise abil koodi kirjutada.

9.2 Tund 1. Sissejuhatus

Tunni plaan:

- Tutvustada õpilastele kursuse eesmärgi ja struktuuri.
- Tutvustada Pythoni keelt ja paigaldada see kooli arvutitesse.
- Tutvustada õpilastele erinevaid programmeerimiskeskonnasid.
- Paigaldada kooli arvutitesse vajalik programmeerimiskeskond nagu Visual Studio Code.

Tunni struktuur:

- Selgitada õpilastele kursuse ülesehitust, eesmärgi, tööde formaate, hindamismeetodeid, kodutöid.
- Tutvustada end õpilastele, lasta igapähele rääkida oma kogemusest ja ootustest kursusele.
- Tutvustada õpilastele paarisprogrammeerimise mõistet, jaguneda paaridesse.
- Arutleda programmeerimise ja Pythoni keele tähtsuse üle.
- Tutvustada õpilastega Pythoni keele ajalugu, selle iseärasusi.
- Paigaldada kooli arvutitesse Pythoni uusim versioon ametlikult veebilehelt <https://www.python.org/downloads/>.
- Arutada erinevate programmeerimiskeskondade, nagu Visual Studio Code, Thonny, PyCharm, eeliste ja puuduste üle.
- Paigaldada kooli arvutitesse Visual Studio Code.
- Tutvustada õpilastele Visual Studio Code'i põhifunktsioone ning luua esimene Python-faili.
- Selgitada õpilastele kodutööd ning millist materjali tuleb järgmiseks tunniks ette valmistada.

9.3 Tund 2. Sissejuhatus Pythoni ja muutujatesse. Google

Tunni plaan:

- Tutvustada õpilastele Pythoni programmeerimise põhimõisteid.
- Tutvustada õpilastele muutujaid, põhilisi muutujaoperatsioone, andmetüüpe.
- Tutvustada õpilastele sõneoperatsioone, meetodeid, sõne vormindamist ja väljundit.
- Tutvustada tingimusoperaatoreid ja nende kasutamist.
- Tutvustada õpilastele Google'i ettevõtet ja selle tööstusvaldkonda.
- Anda õpilastele praktiline ülesanne, mis hõlmab kõiki käsitletud teemasid ning näitab, kuidas erinevad Google'i teenused töötavad.

Kodutöö teemad tunniks:

- Pythoni süntaks: taane, tühikud, üherealised ja mitmerealised kommentaarid.
- Muutujad Pythonis, nende kasutamine, andmetüübid (int, float, str, bool).
- Aritmeetilised operatsioonid muutujatega- liitmine, lahutamine, korrutamine, jagamine, moodul, astendamine.
- Funktsioon print() ja sõnede väljastamine konsooli. Funktsioon input() ja kasutaja sisendi saamine.
- Sõnede ühendamine.
- Andmetüüpide teisendused: int(), float(), str().
- Sõnede vormindamine, sõnede meetodid (.lower(), .upper(), .split(), .strip()).
- Sissejuhatus tingimuslausetesse: if, elif ja else ning nende kasutamine.

Tunni struktuur:

- Teha kiire ülevaade materjalist "Sissejuhatus Pythoni ja muutujatesse", millega õpilased on kodus kokku puutunud, vastates küsimustele materjali selgitamiseks.
- Tutvustada õpilastele Google'i, kuidas Pythoni kasutatakse Google'is, ettevõtte tööülesandeid, Google'i kui tehnoloogiaettevõtte rolli ja selle mõju maailmale.
- Ülevaade Google'i teenustest ja tehnoloogiatest, mida kasutavad Pythoni, näiteks, Google Cloud Platform, TensorFlow ja masinõpe, Google API, Google Sheets API, Google Colab, Google Disk jne (LinkedIn veebileht).
- Ülevaade Google'i poolt kasutatavatest Pythoni moodulitest (nt google-cloud-storage, google-cloud-bigquery, google-api-python-client) (Google Cloud veebileht).
- Pakkuda õpilastele praktilise ülesande lahendamist, mis näitab Google'i lihtsamaid ülesandeid ja hõlmab läbitud teemasid.
- Viia läbi arutelu lahenduste üle, näidates oma versiooni ülesandest.

- Selgitada õpilastele kodutööd, millist materjali tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Google Travel Planner – loo programm, mis planeerib reisi Google'i teenuseid kasutades.

1. Arvutage "Google Mapi" abil kahe linna vaheline reisiaeg.
 - Paluge kasutajal sisestada linnadevaheline kaugus kilomeetrites.
 - Paluge kasutajal sisestada keskmine kiirus kilomeetrites tunnis.
 - Arvutage sõiduaeg tundides ja minutites ning kuvage tulemus.
2. Arvutage reisi maksumus, kasutades "Google Flights".
 - Paluge kasutajal sisestada kahe linna vahelise üksikpileti hind.
 - Küsige kasutajalt, kas ta soovib edasi-tagasi piletit (jah või ei).
 - Arvutage lennu(de) kogumaksumus ja kuvage tulemus.
3. Planeerige reisi kestus.
 - Paluge kasutajal sisestada päevade arv, mille jooksul ta soovib vaatamisväärsustega tutvuda.
 - Paluge kasutajal sisestada päevade arv, mida ta soovib veeta puhkusel.
 - Arvutage reisi kogukestus päevades ja kuvage tulemus.
4. Planeeri igapäevaseid tegevusi Google'i kalendri abil.
 - Paluge kasutajal sisestada tema ärkamisaeg (tunnid ja minutid).
 - Paluge kasutajal sisestada tundide arv, mille jooksul ta soovib vaatamisväärsusi külastada.
 - Paluge kasutajal sisestada tundide arv, mida ta soovib veeta puhkamiseks.
 - Loendage aeg (tund ja minutid), millal kasutaja oma päeva lõpetab, ja kuvage tulemus.
5. Arvutage reisi kogumaksumus, kasutades „Google Sheets“i.
 - Paluge kasutajal sisestada söögi päevane maksumus.
 - Paluge kasutajal sisestada majutuse maksumus öö kohta.
 - Arvutage reisi kogumaksumus, summeerides toidu-, majutus- ja lennukulud (sammust 2) kogu reisi kohta (sammust 3).
 - Kuvage reisi kogumaksumus.

9.4 Tund 3. Loogilised avaldised, tingimuslaused ja tsüklid. Game Dev

Tunni plaan:

- Korrata õpilastega ja süvendatult uurida Pythoni loogiliste väljendeid ja nende kasutuste kohta lisateavet.
- Tutvustada õpilastele tsükleid ja nende kasutamist. Tutvustada õpilastele break käsku.
- Selgitada õpilastele, kuidas kasutatakse koodis võrdlusoperaatoreid ja tõeväärtusloogikat.
- Anda õpilastele põhiteadmised mängude arendamisest Pythoniga.
- Anda õpilastele praktilist ülesannet tunnis käsitletavate teemade harjutamiseks, mis on lihtne mäng.

Kodutöö teemad tunniks:

- Operaatorid if, elif, else.
- Loogilised väljendid ja nende kasutusala (tingimuste kombineerimine sõnadega "ja" ja "või", tingimuste inverteerimine "mitte"-ga).
- While ja for tsüklid ning nende süntaks.
- Tsüklite kasutusjuhud: andmete iteratsioon, korduvad ülesanded ja kasutaja sisendi valideerimine.
- While tsükli ja break käsku süntaks, nende kasutamine.
- Võrdlusoperaatorid: >, <, >=, <=, ==, != ja kuidas neid Pythonis kasutada.
- Boole'i loogika põhimõtted (ja, või, mitte). Boole'i loogika kasutamise võimalused tingimuslausetes ja tsüklites.
- Moodul random ja randint funktsioon.

Tunni struktuur:

- Viia läbi kodus õpitud teemal "Loogilised väljendid, tingimused ja tsüklid" materjali kiirülevaade, vastata materjali täpsustavatele küsimustele.
- Rääkida õpilastele Game Dev'i valdkonnast, anda lühiülevaade mängude arenduskontseptsioonidest, milliseid tehnoloogiaid seal kõige sagedamini kasutatakse.
- Tuua õpilastele näiteid Pythoni kasutamisest mänguarenduses: skriptimine (mängureeglite ja mehaanika rakendamine, NPC ja AI käitumise haldamine, mängusündmuste ja päästikute haldamine), mängude loomise tööriistad ja utiliidid,

mänguloogika (liikumise juhtimine, tegelaste interaktsioon, mängu rakendamine, olekud, mänguandmete haldamine) ja muud (animatsiooni- ja füüsikasüsteemid, protseduuriline sisu genereerimine, võrgu- ja mitme mängijaga funktsioonid) (Tutorialspoint veebileht).

- Näited stuudiotest ja mänguarendusmeeskondadest, kes kasutavad Pythonit (World of Tanks (Python skriptimiseks ja mänguloogika jaoks), Sims 4 (Python mängude skriptimiseks ja modimiseks), Eve Online (Python serveripoolseks skriptimiseks ja mänguloogikaks)) (Game Rant veebileht).
- Tutvustada populaarseid Pythoni mängude arendusteeke (Pygame: tutvustus ja rakendused, Panda3D: Pythoni toega 3D-mängumootor, Godot Engine: Pythoni skriptikeele (GDScript) mängumootor, jm) (Tutorialspoint veebileht).
- Pakkuda õpilastele lahendamiseks praktilise ülesande, mis on lihtne mäng ja sisaldab käsitletavaid teemasid.
- Arutada lahendusi, näidata harjutusele oma lahendust.
- Selgitada õpilastele kodutööd, milline materjal tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Seiklusmäng – tekstipõhine seiklusmäng, milles mängija peab piiratud arvu käikude jooksul jõudma aardeni.

1. Näidake tervitussõnumit, selgitage mängu eesmärki ja teatage lubatud käikude arvu (7 käiku).
2. Seadistage mängukeskkond - initsialiseerige mängumuutujad: määrake *käiku_jäänud* väärtuseks 7, *aare_leitud* väärtuseks False ja *etapp* väärtuseks 1.
3. Kasuta while-tsüklit, mis töötab seni, kuni *käiku_jäänud* on suurem kui 0 ja *aare_leitud* on False.
4. Näita mängijale tsükli sees enne iga käigu algust järelejäänud käikude arvu.
5. Rakendage etappe ja valikuid.
 - 1. etapp: mängija alustab metsast ja peab valima suuna (põhja, lõuna, ida või lääne). Õige suund on ida.
 - 2. etapp: mängija jõuab jõe äärde ja peab valima ujumise, parve ehitamise või silla leidmise vahel. Õige valik on parv või sild.
 - 3. etapp: mängija ületab jõe ja kohtab teelahkmet. Ta peab valima, kas liikuda vasakule või paremale. Õige valik on vasakule.

- 4. etapp: Mängija leiab end ukse ees, kus on peidetud aare. Ukse avamiseks peab koodi ära arvama, sellele antakse 5 katset. Kood on juhuslik arv vahemikus 1 kuni 10.
6. Kasutage tsükli sees iga etapi puhul if-elif lauseid, et kontrollida, millise etapiga on tegu (1, 2 või 3), ning esitage mängijale stsenaarium ja valik olenevalt etapist. Mängu alguses panime juba algväärtuse, muutuja *etapp* =1.
 7. Kasutage input(), et saada mängija valik ja uuendada *käiku_jäänud*, vähendades seda 1 võrra pärast iga valikut.
 8. Kasutage kasutaja sisestuse kontrollimiseks sisseehitatud if-else avaldusi: kui see on õige, siis uuendage etapimuutujat, et minna järgmisesse etappi; kui see on vale, siis kuvage veateade ja laske mängijal uuesti proovida.
Vihje: kasutage kasutaja sisendi kontrollimisel lower() või upper() sõne meetodeid.
 9. Loo salakoodi ära arvamise 4. samm. 4. sammu elif-avaldukes genereeri salakoodiks juhuslik täisarv vahemikus 1-10 (importige selleks random mooduli) (loo muutuja *salakood* ja anna sellele väärtus) ning anna mängijale 5 katset selle ära arvamiseks (loo muutuja *katset_jäänud* ja anna sellele väärtus 5).
 10. Kasuta sisseehitatud while-tsükli katsete käsitlemiseks: kui *katset_jäänud* on suurem kui 0, mis tähendab, et mängijal on veel katseid, hankige mängija sisend funktsiooniga input() ja võrrelge seda salakoodiga.
 11. Kui mängija arvab koodi õigesti ära, määra *aare_leitud* True, näita teade, et uks on avatud ja aare on leitud, ning katkesta while-tsükkel break'iga.
 12. Kui mängija oletus on vale, vähenda katsete arvu *katset_jäänud* ja kuva teade ülejäänud katsetega.
 13. Väljaspool mängutsükli lisage else-lause (while-tsükli sees), mis käivitatakse, kui tsükkel lõpeb katkestusteta (kasutage break). See tähendab, et mängijal on käigud otsa saanud.
 14. Pärast mängutsükli kasutage lauset if-elif-else, et kuvada sobiv teade sõltuvalt sellest, kas mängija võitis, kaotas või tegi liiga palju käike. Kui *aare_leitud* = True, on mängija võitnud. Kui *käiku_jäänud* <=0 – mängija on teinud liiga palju käike, on mäng läbi. Kui mängija ei arvanud ära salakoodi, mängija kaotas.
 15. Testige mängu, veendumaks, et see töötab korralikult ja kas mängija suudab võita, kaotada või teha liiga palju pöördeid.

9.5 Tund 4. Sissejuhatus funktsioonidesse ja nende kasutusviisidesse. Tesla

Tunni plaan:

- Tutvustada õpilastele funktsioone ja nende kasutamist.
- Selgitada õpilastele parameetreid ja tagastusväärtusi.
- Läbi viia Pythoni sisseehitatud teekide ülevaade.
- Selgitada, kuidas importida teeke Pythonis.
- Tutvustada õpilastele Teslat.
- Anda õpilastele praktiline ülesanne, mis hõlmab kõiki käsitletud teemasid ja näitab, kuidas Tesla töötab.

Kodutöö teemad tunniks:

- Funktsioonide määratlemine ja määramine (taaskasutatavad koodi jupid, koodi modulaarsus ja korraldus, koodi taaskasutamine ja korduste vältimine).
- Funktsioonide süntaks ja struktuur (def funktsioonide määratlemiseks; funktsioonide nimed, sulg- ja kooloni süntaks, taane ja funktsiooni keha).
- Funktsioonide loomine ja kutsumine Pythonis. Funktsiooni kutsumine selle nime järgi, millele järgneb sulgud. Funktsioonide näited Pythonis.
- Funktsiooni argumendid ja parameetrid.
- Vaikimisi väärtused parameetrite jaoks. Vaikimisi väärtuste määramine määramisoperaatori (=) abil. Argumentidega ja ilma funktsioonide kutsumine vaikimisi parameetrite jaoks.
- Tagastusväärtused return ja nende kasutamine.
- Tuntud sisseehitatud teegid ja nende otstarve (math – matemaatilised funktsioonid ja konstandid; random – juhuslike arvude genereerimine ja randomiseerimine; datetime – kuupäeva ja kellaaja manipuleerimine; sys – juurdepääs süsteemi parameetritele ja funktsioonidele; os – operatsioonisüsteemi funktsionaalsus).
- Erinevad teekide ja moodulite importimise viisid (import, import from, import..as..).

Tunni struktuur:

- Viia läbi kodus õpitud teemal “Sissejuhatus funktsioonidesse ja nende kasutusviisidesse” materjali kiirülevaade, vastata materjali täpsustavatele küsimustele.

- Tutvustage õpilastele Teslat, Pythoni kasutamist Teslas, ettevõtte tööülesandeid, ettevõtte kasutatavaid uuendusi (elektrisõidukid, energiasalvestid ja päikeseenergia tooted, autonoomne sõidutehnoloogia).
- Tutvustada õpilastele, milleks Tesla Pythonit kasutab (automaatika ja testimine, andmete analüüs ja töötlemine, masinõpe ja tehisintellekti arendamine, simuleerimine ja modelleerimine) (Tesla veebileht).
- Tuua näiteid spetsiifilisematest Tesla ülesannetest Pythoni abil (ennustuslik hooldus ja rikete tuvastamine: sõiduki andurite andmete analüüs, mustrite ja kõrvalekallete tuvastamine; autopiloodisüsteemi arendamine ja täiustamine: masinõppemudelite koolitamine objektide tuvastamiseks ja sõiduraja hoidmiseks, modelleerimine ja testimisalgoritmid autonoomse sõidu analüüs ja energiatarbimise optimeerimine: aku töövõime modelleerimine ja simuleerimine, laadimisskeemi analüüs ja laadimisstrateegiate optimeerimine, taustasüsteemide arendamine: sõidukite ja mobiilirakenduste ühendamiseks API-de loomine) (Tesla veebileht).
- Ülevaade Pythoni kasutatavatest Tesla teenustest ja tehnoloogiatest, nagu TensorFlow masinõppe ja tehisintellekti rakenduste jaoks, NumPy ja pandas andmete analüüsiks ja töötlemiseks, Pytest ja Selenium testimiseks ja automatiseerimiseks, Matplotlib ja Seaborn andmete visualiseerimiseks, SciPy teaduslikeks arvutusteks ja optimeerimiseks (HashDork veebileht).
- Pakkuda õpilastele lahendada praktiline ülesanne, mis simuleerib mõningaid elektrisõiduki funktsioone ja sisaldab käsitletavaid teemasid.
- Arutada lahendusi, näidata harjutusele oma lahendust.
- Selgitada õpilastele kodutööd, milline materjal tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Tesla elektriauto - lihtne elektriauto simulaator, mis kasutab funktsioone ja teekondi reisiparameetrite ja laadimisvajaduste arvutamiseks.

1. Määrake oma auto parameetrid muutujatena.
 - Määrake oma sõiduki maksimaalne aku laeng (nt 100 kWh) muutujaks nimega *maks_aku_laeng*.
 - Määrake oma auto maksimaalne vahemaa täislaadimisel (nt 500 km) muutujana nimega *maks_vahemaa*.

- Määrake oma auto laadimiskulu kilomeetri kohta (nt 0,2 kWh/km) muutujaks, mille nimi on *laadimiskulu*.
 - Määrake kiirus, millega teie auto laeb (nt 22 kW) muutujaks *laadimiskiirus*.
2. Määrake funktsioon auto järelejäänud aku protsendi arvutamiseks.
 - Looge funktsioon nimega *aku_protsent*, millel on kaks argumenti: praegune aku laeng ja maksimaalne aku laeng.
 - Arvutage järelejäänud aku protsent, jagades praeguse laengu maksimaalse laenguga ja korrutades tulemuse 100-ga.
 - Kasutage aku protsendi tagastamiseks return-lauset.
 3. Määrake funktsioon, mis arvutab vahemaa, mille auto saab järelejäänud laadimisega läbida.
 - Looge funktsioon nimega *vahemaa_jäänud*, millel on kaks argumenti: aku järelejäänud protsent ja maksimaalne vahemaa täislaadimisel.
 - Arvutage järelejäänud vahemaa, korrutades järelejäänud aku protsendi maksimaalse vahemaaga.
 - Kasutage järelejäänud vahemaa tagastamiseks return-lauset.
 4. Määrake funktsioon aja arvutamiseks, mis kulub teatud vahemaa läbimiseks teatud kiirusega.
 - Looge funktsioon *reisi_aeg*, millel on kaks argumenti: vahemaa ja kiirus.
 - Arvutage aeg tundides, jagades vahemaa kiirusega.
 - Tagastage reisiaeg return-lausega.
 5. Määrake funktsioon teatud vahemaa läbimiseks nõutava aku laengu arvutamiseks.
 - Looge funktsioon *nõutav_aku_laeng*, millel on kaks argumenti: vahemaa ja laadimiskulu km kohta.
 - Arvutage nõutav aku laeng, korrutades vahemaa laadimiskuluga.
 - Nõutud aku laengu tagastamiseks kasutage return-lauset.
 6. Määrake funktsioon laadimisaja arvutamiseks.
 - Looge funktsioon nimega *laadimisaeg*, millel on kaks argumenti: nõutav aku laeng ja laadimiskiirus.
 - Arvutage auto laadimiseks kuluv aeg, jagades nõutava aku laengu laadimiskiirusega.
 - Tagastage laadimisaeg return-lausega.

7. Tervitage kasutajat ja näidake tema auto maksimaalne laetavus ja mitu kilomeetrit tema auto saab sõita täislaadimisega.
8. Simulatsiooni käivitamiseks looge põhifunktsioon parameetriga *praegune_aku_laeng*.
 - Hankige järelejäänud aku protsent funktsiooni *aku_protsent* abil.
 - Arvutage *vahemaa_jäänud* funktsiooni abil vahemaa, mille saab antud järelejäänud laenguga läbida.
 - Näidake kasutajale tema auto järelejäänud laengut ja seda, mitu kilomeetrit ta veel sõita suudab.
 - Genereeri juhuslik kiirus ja arvuta järelejäänud vahemaa läbimiseks kuluv aeg, kasutades funktsiooni *reisi_aeg*.
 - Importige moodul `random`. Juhusliku kiiruse (km/h) genereerimiseks vahemikus 50–100 kasutage käsku `random.randint()`.
 - Kuvage kasutajale välja tema auto juhuslik kiirus ja see, kui kaua kulub sellel kiirusel järelejäänud vahemaa läbimiseks.
 - Küsige kasutajalt soovitud läbisõidukaugust ja arvuta *nõutav_aku_laeng* funktsiooni abil nõutav aku laengu.
 - Veenduge, et nõutav laeng on praegusest laengust väiksem või sellega võrdne. Kui jah, printige teade, et auto suudab soovitud vahemaa läbida. Muul juhul arvuta *laadimisaeg* funktsiooni abil laadimisaeg ja prindi välja teade, mis näitab vajalikku laadimisaega.
9. Simulatsiooni käivitamiseks kutsuge üles põhifunktsioon, sisestades parameetrina aku praeguse laeng (kWh). Proovige katsetada auto parameetreid ja laengu taset.

9.6 Tund 5. Faili sisend ja väljund Pythonis. JPMorgan Chase

Tunni plaan:

- Anda ülevaade failitüüpidest ja nende kasutusalaadest.
- Õpetada õpilastele, kuidas Pythonis faile avada ja sulgeda.
- Õpetada õpilastele, kuidas lugeda andmeid failidest ja kirjutada andmeid faili.
- Tutvustada õpilastele failide haldamist ja manipuleerimist.
- Tutvustada õpilastele CSV-failide struktuuri ja seda, kuidas Pythoni abil andmeid lugeda, kirjutada.

- Tutvustada õpilastele JSON-failide struktuuri ning andmete lugemist ja kirjutamist Pythoni abil.
- Tutvustada õpilastele vigade ja erandite käsitlemist.
- Tutvustada õpilastele JPMorgan Chase'i.
- Anda õpilastele praktiline ülesanne, mis hõlmab kõiki käsitletud teemasid ja näitab, kuidas JPMorgan Chase töötab.

Kodutöö teemad tunniks:

- Failide sisestamine ja väljastamine Pythonis.
- Failitüübid ja nende kasutamine. Tekstifailid. Binaarfailid.
- Failide avamine ja sulgemine Pythonis. Funktsioon `open()`. Meetod `close()`.
- Andmete lugemine failidest. Kogu faili lugemine, meetod `read()`. Faili rea kaupa lugemine, meetod `readline()`, failiobjekti läbikäimine.
- Andmete kirjutamine failidesse. Meetod `writelines()`. Meetod `write()`. Andmete lisamine faili.
- Faili avamise režiimid ('r': lugemisrežiim, 'w': kirjutamisrežiim, 'a': lisamisrežiim, 'x': eksklusivne loomisrežiim, 'b': binaarne režiim).
- Failide ümbernimetamine ja kustutamine. Funktsioon `os.rename()`. Funktsioon `os.remove()`.
- Töö kataloogidega. Funktsioon `os.mkdir()`. Funktsioon `os.chdir()`. Funktsioon `os.rmdir()`. Funktsioon `os.listdir()`.
- Failide olemasolu ja atribuutide kontrollimine. Funktsioon `os.path.exists()`. Funktsioon `os.path.isfile()`. Funktsioon `os.path.isdir()`.
- Sissejuhatus CSV-failidesse ja nende struktuuri. CSV-failide lugemine Pythoni sisseehitatud `csv`-mooduli abil. Funktsioon `csv.reader()`. Andmete kirjutamine CSV-faili kasutades Pythoni `csv`-moodulit. Funktsioon `csv.writer()`.
- Sissejuhatus JSON-failidesse ja nende struktuuri. JSON-failide lugemine Pythoni sisseehitatud `json`-mooduli abil. Funktsioon `json.load()`. Andmete kirjutamine JSON-faili kasutades Pythoni `json`-moodulit. Funktsioon `json.dump()`.
- Tavalised vead ja erandid faili sisse- ja väljundtegevustes (nt. `FileNotFoundError`, `IOError`, `PermissionError`).
- Try-except plokkide kasutamine vigade ja erandite käsitlemiseks.

Tunni struktuur:

- Viia läbi kiire kordamine teemal "Faili sisend ja väljund Pythonis", millega õpilased on kodus õppides tutvustanud, vastates küsimustele materjali selgitamiseks.
- Tutvustada õpilastele JPMorgan Chase'i, kuidas Python'i kasutatakse JPMorgan Chase'is, ettevõtte ärieesmärgid ja tegevusvaldkondi, peamisi ärisegmente, investeeringuid tehnoloogiasse ja innovatsiooni ning koostööd FinTech startup'idega.
- Tutvustada õpilastele, milleks Python'i kasutatakse JPMorgan Chase'is (kvantitatiivsed uuringud ja analüüs, automatiseerimine ja testimine, andmeanalüüs ja -töötlus, riskijuhtimine ja pettuste tuvastamine, algoritmiline kauplemine ja finantsinsenerlus, loomuliku keele töötlemine tundmusanalüüsiks ja dokumentide klassifitseerimiseks) (eFinancialCareers veebileht).
- Tuua näiteid spetsiifilisematest JPMorgan Chase'i ülesannetest, mille puhul kasutatakse Python'i (finantsandmete analüüsimine trendide ja muustrite leidmiseks, et anda teavet kauplemisotsuste kohta, riskijuhtimine ja finantsportfellide stressitestimine, kauplemis- ja investeerimisstrateegiate algoritmide väljatöötamine, sisevahendite ja veebirakenduste loomine töötajatele, riskihindamine ja pettuste tuvastamine masinõppe mudelite abil) (eFinancialCareers veebileht).
- Ülevaade JPMorgan Chase'i teenustest ja tehnoloogiast, mida kasutatakse Pythoni abil, näiteks: Athena: JPMorgan Chase'i riskijuhtimise ja varadega kauplemise platvorm (ehitatud Pythoni abil); Bquant: JPMorgan Chase'i poolt välja töötatud Pythonil põhinev kvantitatiivse rahanduse teek; Quantlib: JPMorgan Chase'i poolt välja töötatud kvantitatiivse rahanduse teek; Quantlib: Pythoni teek kvantitatiivse rahanduse ja riskijuhtimise jaoks; Pyfolio: Pythoni teek portfelli- ja riskianalüüsi jaoks; Gensim: Pythoni teek loomuliku keele töötlemiseks ja dokumentide sarnasuse analüüsiks; Zipline: Pythoni teek algoritmilise kauplemise jaoks (JP Morgan Chase & Co veebileht).
- Pakkuda õpilastele lahendada praktiline ülesanne, mis on seotud finantsandmete töötlemise ja analüüsiga ja hõlmab õpitud teemasid.
- Viia läbi arutelu lahenduste üle, näidates oma versiooni ülesanne lahendusest.
- Selgitada õpilastele kodutööd, millist materjali tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Finantstehingute analüsaator - Finantstehingute analüüsi tarkvara, mis võimaldab kasutajatel analüüsida sisestatud andmeid, koostada aruandeid ja salvestada need failidesse.

1. Importige csv-, json- ja os-moodulid.
2. Looge funktsioon *sisendtehing*, mis küsib kasutajalt tehingu kuupäeva (YYYY-MM-DD), summat ja kategooriat (tulud, kulud, investeringud). Funktsioon tagastab saadud andmed.
3. Looge funktsioon *salvesta_tehing_csv_faili*, mis võtab parameetriteks tehingu ja csv-faili nime. Funktsioon peab kirjutama tehingu määratud faili "transactions.csv" lõppu, kasutades csv-moodulit, kui faili ei ole olemas, siis luua see.
4. Looge funktsioon *analüüsi_tehinguid*, mis loeb finantsandmeid CSV-failist ja teostab põhianalüüsi. Funktsioon peab vastu võtma kolm sisendparameetrit: analüüsitava tehingute alguskuupäev ja lõppkuupäev ning faili nimi, millest andmeid loetakse.

Funktsioon peaks arvutama kogusissetulek, kogukulu ja koguinvesteeringu ning tagastama need väärtused.

5. Looge funktsioon *salvesta_analüüsitud_andmed_json_faili*, mis võtab sisendparameetritena kogusissetulek, kogukulu, investeeringute kogusumma, kuupäevavahemiku ja faili nime, kuhu andmed kirjutatakse, ning loob analüüsi põhjal finantsaruande (selles ülesannes analüüsitud andmed võib kirjutada tekstisõnnesse).

Aruanne tuleks salvestada JSON-faili nimega "analysed_data.json".

6. Looge funktsioon *json_fail_on_olemas*, mis kontrollib, kas JSON-fail on praeguses kataloogis olemas, kasutades os-moodulit.
7. Rakendage funktsioonis *main()* tsükkel, mis võimaldab kasutajatel valida järgmiste valikute vahel: 1. Lisa tehing; 2. Analüüsida tehinguid; 3. Salvesta analüüsitud tehingud json-faili; 4. Välju. Kirjutage kasutaja sisestus muutujasse *valik*.
8. 1. valiku puhul kutsuge funktsiooni *sisendtehing*, et saada kasutajalt tehinguandmed ja salvestada need CSV-faili, kasutades funktsiooni *salvesta_tehing_csv_faili*.
9. 2. valiku puhul paluge kasutajal sisestada analüüsitava tehingute algus- ja lõppkuupäevad, kutsuge *analüüsi_tehinguid* funktsiooni analüüsi teostamiseks ja väljastage tulemused konsooli.
10. 3. valiku puhul kutsuge funktsiooni *salvesta_analüüsitud_andmed_json_faili*, et koostada finantsaruanne ja salvestada see JSON-faili. Kasutage funktsiooni

`json_fail_on_olemas`, et kontrollida, kas fail loodi, ja kui fail loodi edukalt, kuvage kasutajale selle kohta konsooli sõnum.

11. 4. valiku puhul väljuge programmist.

12. Käivitage põhifunktsioon.

9.7 Tund 6. Sissejuhatus täiustatud andmetüüpidesse. Järjendid. Spotify

Tunni plaan:

- Tutvustada õpilastele laiendatud andmetüpe, nagu järjendid, ennikud ja sõnastikud, nende sarnasusi ja erinevusi.
- Anda ülevaade andmete struktuurist ja selle kasutusvõimalustest.
- Selgitada õpilastele, kuidas järjendeid luua ja elementidele ligi pääseda.
- Anda ülevaade järjendite meetoditest ja operatsioonidest.
- Tutvustada õpilastele järjendite sorteerimist.
- Tutvustada õpilastele sisseehitatud järjendeid ja mitmemõõtmelisi massiive.
- Tutvustada õpilastele Spotify't.
- Tutvustada õpilastele praktilist ülesannet, mis hõlmab kõiki käsitletud teemasid ja näitab, kuidas Spotify töötab.

Kodutöö teemad tunniks:

- Sissejuhatus järjenditesse, ennikutesse ja sõnastikesse. Sisseehitatud andmestruktuuride kasutamise eelised.
- Järjendite, ennikute ja sõnastike võrdlus (ühised omadused: andmete säilitamine ja korraldus, sisseehitatud meetodid ja funktsioonid; erinevused: muutlikkus, korrastatus ja korrastamatus, andmekorraldus).
- Ennikud (määratlus ja kasutusosalad, muutuv ja korrastatud elementide kogum, sobivad fikseeritud andmete jaoks, mida ei ole vaja muuta).
- Sõnastikud (määratlus ja kasutusosalad, muutuv ja korrastamata võtme-väärtuse paaride kogum, sobib võtme-väärtuse kaardistamiseks ja andmete kiireks otsimiseks).
- Järjendid (määratlus ja kasutusviisid, muutuv ja järjestatud elementide kogum, sobivad sageli muutuvate andmete käsitlemiseks).
- Järjendite loomine ja juurdepääs elementidele. Tühjade ja eeltäidetud järjendite loomine. Juurdepääs elementidele indeksiga. Negatiivne indekseerimine.
- Järjendi lõikamine ja indekseerimine. Järjendite lõikamine alamjärjendite loomiseks.

- Järjendite ja elementide muutmine. Elementidele uute väärtuste määramine. Järjendite ühendamine. Operaatori * kasutamine järjendite kordamiseks.
- Üldised järjendimeetodid ja operatsioonid (append(): elementide lisamine järjendi lõppu, insert(): elementide lisamine konkreetsele positsioonile, remove(): elementide eemaldamine väärtuse järgi, pop(): elementide eemaldamine indeksi järgi ja kustutatud väärtuse tagastamine).
- Järjendite sorteerimine, viitamine ja pikkuse määramine. (sort(): järjendi sorteerimine kasvavas või kahanevas järjekorras, reverse(): järjendi elementide järjekorra ümberpööramine, len(): järjendi elementide arvu leidmine).
- Sisestatud järjendid ja mitmemõõtmelised massiivid. Sisestatud järjendite loomine ja elementidele juurdepääsu võimaldamine.
- Järjendite hõlmamised (List comprehensions) - põhilised, pesastatud ja tingimuslikud.

Tunni struktuur:

- Viia läbi kiire ülevaade "Sissejuhatus täiustatud andmetüüpidesse. Järjendid", mida õpilased on kodus õppinud, vastates küsimustele materjali selgitamiseks.
- Tutvustada õpilastele Spotify't, selle ajalugu, peamisi teenuseid (muusika voogedastus, podcastid ja otseülekanded) ning seda, kuidas see keskendub tehnoloogiale ja innovatsioonile.
- Tutvustada õpilastele, kuidas Spotify's kasutatakse Pythoni keelt: taustateenused ja API-d, andmetöötlus ja analüüs, masinõpe ja soovitusalgoritmid, loomuliku keele töötlemine laulusõnade ja podcastide transkriptsioonide analüüsimiseks (Spotify Engineering veebileht).
- Tutvustada õpilastele Pythoni teeki, tehnoloogiaid ja vahendeid, mida Spotify kasutab (Luigi: Spotify poolt välja töötatud Pythoni platvorm andmepiiprite loomiseks, Annoy: Spotify poolt välja töötatud Pythoni teek lähima naabri ligikaudseks otsinguks, TensorFlow: masinõppe teek, Pandas: andmeanalüüsi ja -töötlemise teek, Scikit-learn: masinõppe teek andmekavandamiseks ja -analüüsiks, NLTK ja spaCy: loomuliku keele töötlemise teegid, Flask ja Django: veebiraamistikud APIde ja veebirakenduste loomiseks) (Spotify Engineering veebileht).
- Anda reaalseid näiteid Pythoni kasutamise kohta Spotify's (soovitusalgoritmide väljatöötamine personaliseeritud esitusloendite loomiseks, kasutajate käitumise ja

kuulamisharjumuste analüüsimine kasutajakogemuse parandamiseks, loomuliku keele töötlemine meeleolude analüüsiks ja sisu kategoriseerimiseks, sisemiste vahendite ja APIde loomine ja haldamine andmetele juurdepääsuks ja nendega manipuleerimiseks) (Spotify Engineering veebileht).

- Pakkuda õpilastele lahendada praktiline ülesanne, milles kasutaja loob ja redigeerib oma esitusloendi ning kaasab selles käsitletud teemad.
- Viia läbi arutelu lahenduste üle, näidates oma versiooni ülesannetest.
- Selgitada õpilastele nende kodutööd, millist materjali tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Esitusloend peo jaoks – peo esitusloend, kus saab hallata oma laulude nimekirja - lisada ja kustutada laule, sorteerida erinevate kriteeriumide järgi, otsida konkreetset laulu ja salvestada saadud esitusloendi eraldi faili.

1. Looge põhifunktsioon, selle sees looge tühi nimekiri. Funktsiooni sees looge while True tsüklil, et programm töötaks seni, kuni kasutaja selle peatab. Tsükli sees väljastage kasutaja tegevusvalikud (Valikud: Lisada laul; Eemalda laul; Sorteerida esitusloend; Kuva esitusloend; Salvesta esitusloend; Segada esitusloend; Tühjenda esitusloend; Otsida laul; Väljuda). Looge iga tegevuse jaoks eraldi funktsioon ja käivitage need sõltuvalt kasutaja valikust.
2. Lisada laul: see funktsioon peaks võimaldama kasutajal lisada parameetrina sisestatud esitusloendisse laulu. Paluge kasutajat üles sisestama laulu pealkirja, artisti ja kestust (sekundites). Lisage saadud andmed järjendisse alamjärjendina ja kuvage ekraanile teade lisatud laulu kohta.
3. Eemalda laul: See funktsioon peaks võimaldama kasutajal laulu kustutada parameetrina sisestatud esitusloendist. Paluge kasutajal sisestada kustutatava laulu nimi. Võrreldage iga järjendis olevat laulu sisestatud pealkirjaga. Kuvage teade, et laul on kustutatud.
4. Sorteerida esitusloend: see funktsioon peaks võimaldama kasutajal sorteerida esitusloendi (mis on sisestatud funktsiooni parameetrisse) laulu pealkirja, artisti või kestuse järgi. Kutsutage kasutajat üles valima sorteerimiskriteeriume (pealkiri, artist ja kestus). Kasutage sorteerimiseks meetodit sort() ja konstruktsiooni key=lambda. Väljastage teade kriteeriumide kohta, mille järgi esitusloend on sorteeritud.
5. Kuva esitusloend: see funktsioon peaks kuvama praeguse esitusloendi (mis on sisestatud funktsiooni parameetrisse) koos iga laulu nime, artisti ja kestusega. Samuti

- peaks see näitama esitusloendis olevate laulude arvu, kasutage meetodit `len()`. Kui esitusloendis ei ole ühtegi laulu, andke kasutajale sellest teada.
6. Salvesta esitusloend faili: see funktsioon peaks salvestama lõpliku esitusloendi (mis on sisestatud funktsiooni parameetrisse) tekstifaili, mille nime küsige kasutajalt, kus iga laul on eraldi real ja selle andmed on eraldatud komadega. Väljastage teade, et esitusloend on salvestatud.
 7. Segada esitusloend: see funktsioon peaks segada esitusloendi (funktsiooniparameetrisse sisestatud) laule juhuslikult. Kasutage `random` moodulit ja `shuffle` meetodit. Väljastage teade, et esitusloend on segatud.
 8. Tühjenda esitusloend: see funktsioon peaks eemaldama kõik laulud esitusloendist (sisestatud funktsiooni parameetrina). Kasutage meetodit `clear()`. Väljastage kasutajale teade, et esitusloend on tühjendatud.
 9. Otsida laul: see funktsioon peab võimaldama kasutajal otsida esitusloendist (mis on sisestatud funktsiooni parameetrisse) laulu pealkirja, artisti või kestuse järgi. Paluge kasutajal sisestada otsingukriteerium ja seejärel peaks tarkvara kuvama kõik esitusloendis olevad laulud, mis vastavad sellele otsingukriteeriumile. Looge eraldi tühi nimekiri, kui otsingukriteerium vastab laulu pealkirjale, esitajale või kestusele, siis lisage see laul tühja nimekirja. Kontrollige kõiki laule esitusloendis. Kuvage sobivate laulude nimekiri ja kui ühtegi laule ei leita, teavitage sellest kasutajat.
 10. Väljuda – lõpetage programm.
 11. Käivitage põhifunktsioon ja proovige luua oma esitusloend.

9.8 Tund 7. Sõnastikud ja ennikud. Amazon

Tunni plaan:

- Selgitada õpilastele, kuidas luua sõnastikke ja töötada nendega, ning tutvustage sõnastike üldisi meetodeid ja operatsioone.
- Selgitada õpilastele, kuidas luua ja töötada ennikutega, ning tutvustada ennikute üldisi meetodeid ja operatsioone.
- Tutvustada õpilastele Amazoni.
- Anda õpilastele praktiline ülesanne, mis hõlmab kõiki käsitletud teemasid ja näitab, kuidas Amazon töötab.

Kodutöö teemad tunniks:

- Sõnastike loomine ja objektidele juurdepääs. Tühjade ja eeltäidetud sõnastike loomine. Juurdepääs elementidele võtmete abil. Meetodi `get()` kasutamine turvaliseks juurdepääsuks.
- Võtmeväärtuspaaride lisamine, uuendamine ja eemaldamine. Uute võtmeväärtuspaaride lisamine. Olemasolevate võti-väärtuspaaride uuendamine. Võtmeväärtuspaaride kustutamine, kasutades `del` ja `pop()`.
- Üldised sõnastiku meetodid ja operatsioonid (`keys()`: saada kõik võtmed sõnastikus; `values()`: saada kõik väärtused sõnastikus; `items()`: saada kõik võtmeväärtuse paarid ennikutena; `update()`: ühendada kaks sõnastikku; `clear()`: eemaldada kõik võtmeväärtuse paarid sõnastikust).
- Põhilised ja tingimuslikud sõnastiku hõlmamised (dictionary comprehensions).
- Ennikute loomine ja juurdepääs elementidele. Tühjade ja eeltäidetud ennikute loomine. Juurdepääs objektidele indeksiga.
- Ennikute muutumatus ja selle eelised. Enniku muutmise katse põhjustab erandi.
- Ennikute pakkimine ja lahtipakkimine. Mitme väärtuse pakkimine ennikusse. Enniku lahtipakkimine eraldi muutujateks.
- Üldised ennikuoperatsioonid ja meetodid (`len()`: enniku pikkuse määramine, `count()`: elemendi esinemiste loendamine ennikus, `index()`: elemendi indeksi otsimine ennikus).
- Põhilised ja tingimuslikud generaatoriavaldised.

Tunni struktuur:

- Viia läbi kiire ülevaade teemast "Sõnastikud ja ennikud", millega õpilased on kodus tuttavad, vastates küsimustele materjali selgitamiseks.
- Tutvustada õpilastele Amazoni, selle ajalugu ja peamisi teenuseid: e-kaubandus, AWS, Prime Video, Kindle jne.
- Tutvustada, kuidas Amazon kasutab Pythoni (backend-teenused ja API'd, andmetöötlus ja analüüs, masinõppe ja soovituselise algoritmid, loomuliku keele töötlemine ja juturobotite arendamine).
- Tutvustada õpilastele Amazon Web Services (AWS) (populaarsed teenused: EC2, S3, Lambda, RDS jne), andes lühiülevaate AWSist kui pilveplatvormist (Amazon Web Services veebileht).
- Näidata õpilastele Pythoni teeki, vahendeid ja tehnoloogiaid, mida Amazon kasutab (Boto3: Python SDK AWS-i teenuste jaoks, TensorFlow, Pandas, Scikit-learn,

NLTK ja spaCy, Flask ja Django, AWS Glue: serverita ETL teenus, mis kasutab andmete töötlemiseks Pythonit, Amazon SageMaker: masinõppe platvorm Python SDK-ga, AWS Chalice: serverita Python mikroraamistik AWSi jaoks, AWS Lambda: Pythonit toetav serverita arvutusteenus) (Amazon Web Services veebileht).

- Anda reaalseid näiteid Pythoni kasutamise kohta Amazonis (soovitusalgoritmide väljatöötamine personaliseeritud ostude jaoks, klientide käitumise ja eelistuste analüüs, loomuliku keele töötlemine Alexa ja juturobotite arendamiseks, sisemiste vahendite ja APIde loomine ja haldamine andmetele juurdepääsuks ja nendega manipuleerimiseks, infrastruktuuri haldamise ja kasutuselevõtu automatiseerimine Pythoni ja AWSi teenuste abil) (Amazon Web Services veebileht).
- Pakkuda õpilastele lahendada praktiline ülesanne, mis simuleerib lihtsat assistenti, mis aitab teha veebioste Amazonist ja sisaldab õpitud teemasid.
- Viia läbi arutelu lahenduste üle, näidates oma versiooni lahendusest.
- Selgitada õpilastele, millist kodutööd nad peavad järgmiseks tunniks ette valmistama.

Praktiline ülesanne:

Isiklik ostuassistent Amazonis – Lihtne assistent, mis võimaldab kasutajal leida soovitud toote, lisada selle ostukorvi, eemaldada selle ostukorvist, vaadata hinda või esitada tellimus.

1. Looge põhifunktsioon *main()*. Looge funktsiooni sees olemasolevate toodete (nt puuviljad ja marjad) sõnastik, milles on iga toote kohta järgmised andmed: nimi, hind ja olemasolev kogus. Looge funktsiooni sees tühi järjend ostukorv, kuhu lisatakse kasutaja ostud.
2. Põhifunktsiooni sees looge while True tsükkel, selle tsükli sees tervitage kasutajat isikliku assistendi nimel ja kuvage menüü valikuvõimalustega (Otsida toode; Lisada toode ostukorvi; Vaadata oma ostukorvi; Eemaldada toode; Vaadata hind kokku; Vormistada tellimus; Väljuda). Läbige kasutaja sisendi, kasutades if-elif-else konstruktorit, ning looge ja kutsuge iga valiku jaoks eraldi funktsioon.
3. Otsida toode - looge *otsi_toode(tooted)* funktsioon. Paluge kasutajal sisestada toote otsingusõna ja salvestage sisend muutujas. Looge otsingutulemuste salvestamiseks tühi sõnastik *sobib*. Kontrollige iga toote puhul, kas otsingusõna on elementi võtme (st toote nime) alamsõne. Kasutage for-tsükli. Kui on, lisage element sõnastikku. Kasutage veel ühte for-tsükli sõnastikust leitud elementide läbimiseks. Iga toote

- puhul trükkige välja võti (st toote nimi), hind ja olemasolev kogus. Kui *sobib*-sõnastik on tühi, trükkige teade, et tulemusi ei leitud.
4. Lisada toode ostukorvi – looge *lisada_ostukorvi(tooted,ostukorv)* funktsioon. Paluge kasutajal sisestada toote nimi ja salvestage sisend muutujas. Kui toode ei ole toodete sõnastikus, andke kasutajale sellest teada (kasutage *for-not* in konstruktsiooni). Paluge kasutajal sisestada toote kogus ja salvestage sisend muutujas. Kui sisestatud kogus on suurem, kui olemasolev kogus, andke kasutajale teada, et kahjuks laos puudub piisav kogus. Uuendage algses sõnastikus oleva toodete kogust, lahutades sellest kasutaja poolt sisestatud koguse. Lisage sisestatud toode ostukorvi ja andke sellest kasutajale teada.
 5. Vaadata oma ostukorvi – looge *vaadata_ostukorvi(ostukorv)* funktsioon. Kui ostukorv on tühi, kuvage sellekohane teade. Näidake iga ostukorvis olev toode koos hinna ja kogusega. Kasutage *for-in* konstruktsiooni.
 6. Eemaldada toode – looge *eemaldada_ostukorvist(tooted,ostukorv)* funktsioon. Paluge kasutajal sisestada toote nimi ja salvestage sisend muutujas. Iga ostukorvis asuva toode eemaldage ostukorvist, kasutades meetodit *remove()*, ning andke sellest teada. Uuendage algses sõnastikus oleva toodete kogust, liitudes ostukorvis asuva toote kogus. Kui sisestatud toodet pole ostukorvis, andke sellest teada.
 7. Vaadata hind kokku – looge *vaadata_hind_kokku(ostukorv)* funktsioon. Näidake kõigi ostukorvis olevate kaupade hindade summat.
 8. Vormistada tellimus – looge *checkout(tooted,ostukorv)* funktsioon. Kontrollige, kas ostukorv on tühi. Kui see on, trükkige kasutajat teavitav teade. Arvutage ostukorvis olevate toodete koguhind, korrutades iga toote hinna ja koguse ning summeerides need kokku. Küsige kasutajalt, kas ta soovib ostu kindlasti lõpule viia. Kui kasutaja kinnitab ostu, vähendage sõnastikus olevast saadaolevast kogusest iga ostukorvis oleva toote kogus ja tühjendage ostukorv. Trükkige kasutajale tänusõnum. Kui kasutaja tühistab ostu, trükkige tühistamise teade.
 9. Kui kasutaja tühistab ostu, trükkige tühistamise teade.
 10. Väljuda – lõpetage programm ja tänage kasutajat Amazoni valimise eest.
 11. Käivitage põhifunktsioon ja proovige vormistada oma tellimuse.

9.9 Tund 8. Sissejuhatus objektorienteeritud programmeerimisse. Robotika

Tunni plaan:

- Tutvustada objektorienteeritud programmeerimise põhimõisteid ja põhiprintsiipe.
- Tutvustada õpilastele pärilikkuse, inkapsulatsiooni ja polümorfismi põhimõtteid, teema sissejuhatus.
- Rääkida õpilastele klassidest ja objektidest.
- Õpetada õpilastele atribuute ja meetodeid.
- Õpetada õpilastele robootika põhitõdesid.
- Anda õpilastele praktiline ülesanne, mis hõlmab käsitletud teemasid ja täidab robootikaülesannet.

Kodutöö teemad tunniks:

- Sissejuhatus objektorienteeritud programmeerimisse. Ülevaade OOP mõistetest. OOP määratlus ja eesmärk. Peamised põhimõtted: pärilikkus, inkapsulatsioon ja polümorfism.
- Klassid ja objektid. Klasside ja objektide määratlemine. Klasside loomise näited.
- Atribuudid ja meetodid. Atribuutide ja meetodite määratlemine. Eksemplari atribuudid ja klassi atribuudid.
- Konstruktorid ja destruktorid, nende määratlus ja otstarve. `__init__` ja `__del__` meetodid.
- Pärilikkus. Pärilikkuse määratlus ja eesmärk. Alamklasside loomine ja omaduste pärimine.
- Polümorfism. Polümorfismi määratlus ja eesmärk. Meetodite üledefineerimine alamklassides.
- Inkapsulatsioon. Inkapsulatsiooni määratlus ja eesmärk. Privaatsed ja avalikud atribuudid ja meetodid. Omaduste kasutamine privaatsetele atribuutidele juurdepääsuks ja nende muutmiseks.

Tunni struktuur:

- Viia läbi kiire ülevaade materjalist "Sissejuhatus objektorienteeritud programmeerimisse", millega õpilased on kodus tutvunud, vastata küsimustele materjali selgitamiseks.
- Teha sissejuhatus robootikasse, kirjeldades robotite tüüpe (tööstusrobotid, teenindusrobotid, humanoidid jne) ja nende peamisi komponente (andurid, ajamid ja juhtimissüsteemid).
- Rääkida robootika valdkondadest (tootmine ja kokkupanek, tervishoid ja kirurgia, põllumajandus, uurimis- ja teadustegevus, meelelahutus ja mänguasjad).

- Tutvustada, milleks Python robotikas kasutatakse (Python algoritmide arendamiseks, andmete analüüsiks ja prototüüpimiseks; Python andurite, täiturmehhanismide ja sidesüsteemidega ühendamiseks) (Franjee et al., 2016).
- Rääkida õpilastele Pythoni moodulitest ja vahenditest robotika jaoks (ROSPy: Pythoni teekond robotite operatsioonisüsteemi (ROS) jaoks, Pygame: simulatsioonide ja visualiseerimiste loomiseks, OpenCV: arvutinägemise teek pilditöötlemiseks ja objektide tuvastamiseks, NumPy ja SciPy, Pypot: teek robotite ajamite ja sensorite juhtimiseks) (ROS Wiki veebileht).
- Anda reaalseid näiteid Pythoni kasutamise kohta robotikas (robotkäte juhtimiseks vajalike algoritmide väljatöötamine, arvutinägemise rakendamine objektide tuvastamiseks ja jälgimiseks, andurite andmete analüüs ja töötlemine robotite navigeerimiseks; robotsüsteemide prototüüpimine ja testimine simulatsioonikeskkondades) (Franjee et al., 2016).
- Pakkuda õpilastele praktilise ülesande lahendamist, mis simuleerib erinevate robotite toimimist ja sisaldab käsitletud teemasid.
- Viia läbi arutelu lahenduste üle, näidata oma versiooni ülesandest.
- Selgitada õpilastele nende kodutööd ja seda, millist materjali tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Roboti juhtimine - rakendus, mis võimaldab suhelda robotitega, valida puhastus- või tarneroboti ja täita erinevaid funktsioone sõltuvalt valikust.

1. Määrake vanemklass nimega *Robot*, millel on järgmised atribuudid:
 - *nimi*: roboti nime tähistav sõne.
 - *aku_tase*: täisarv, mis näitab roboti aku taset, vaikumisi väärtus 100.
 - Määrake konstruktor (meetod *init*), mis initsialiseerib need atribuudid.
 - Defineerige meetod nimega *liikuda*, mis võtab parameetri *sihtkoht* (sõne) ja väljastab teate, mis näitab, et robot liigub sihtkohta. See meetod peaks ka vähendada aku taset 10 võrra iga liikumise korral.
 - Defineerige meetod nimega *laadida*, mis seab aku taseme tagasi 100-le ja trükkib teate, mis näitab, et robot on üle laetud.
2. Määrake lapsroboti klass nimega *TarneRobot*, millel on järgmised atribuudid:
 - Määrake konstruktor, mis kutsub vanemkonstruktorit ja initsialiseerib atribuudid *nimi* ja *aku_tase*.

- Defineerige meetod nimega *tarnida_tellimus*, mis võtab vastu kaks parameetrit: *saaja*: sõne, mis kujutab pakendi sihtkoha nime; *sihtkoht*: sõne, mis kujutab tarne sihtkohta.
 - See meetod peab kõigepealt kontrollima, kas aku tase on tarne lõpetamiseks piisavalt kõrge (vähemalt 25%). Kui aku tase on liiga madal, trükitakse teade, et saatmist ei saa lõpule viia. Kui aku tase on piisav, peaks robot liikuma sihtkohta, andma paki kätte ja vähendama aku taset 25 võrra.
 - Määratlege meetod nimega *skaneerida_paki*, mis võtab parameetri *pakend* (sõne). See meetod peaks kontrollima, kas aku tase on piisavalt kõrge, et skaneerimine lõpule viia (vähemalt 20%). Kui aku tase on liiga madal, trükkida teade, et skaneerimist ei saa lõpule viia. Kui aku tase on piisav, peaks robot skaneerima pakki ja vähendama aku taset 20 võrra.
3. Määrake veel üks lapsroboti klass nimega *PuhastusRobot*, millel on järgmised atribuudid:
- Määrake konstruktor, mis kutsub vanemkonstruktorit ja initsialiseerib atribuudid *nimi* ja *aku_tase*.
 - Määrake täiendav atribuut nimega *puhastusrežiim*, mis tähistab robotite praegust režiimi (näiteks, pühkima, pesema, tolmuimeja).
 - Määrake meetod nimega *vali_puhastusrežiim*, mis võtab parameetri *režiim* (sõne), et esindada puhastusrežiimi. See meetod peaks seadma atribuudi *puhastusrežiim* ja vähendama aku taset 10, 20 või 25 võrra sõltuvalt valitud režiimi energiatarbimisest.
 - Määrake meetod *puhastada_tuba*, mis võtab vastu parameetri *tuba* (sõne), mis tähistab puhastatavat ruumi. See meetod peab kõigepealt kontrollima, kas aku tase on piisavalt kõrge, et puhastamine lõpule viia (vähemalt 10%). Kui aku tase on liiga madal, kuvatakse teade, et puhastamist ei saa lõpule viia. Kui aku tase on piisav, peaks robot liikuma ruumi ja puhastama, kasutades parajasti valitud puhastusrežiimi. See meetod peaks samuti vähendama aku taset 10 võrra.
4. Looge ja käivitage funktsioon *main()*. Selle funktsiooni sees::
- Looge while-tsüklil, mis hoiab programmi käimas, kuni kasutaja otsustab lõpetada. Näita while-tsükli sees menüü järgmiste parameetritega: *TarneRobot*, *PuhastusRobot*, *Välju*. Paluge kasutajal sisestada oma valik.

- Kui kasutaja valib TarneRobot, looge uus TarneRobot klassi eksemplar ja paluge kasutajal sisestada roboti nimi. Kuvage menüü, milles on järgmised parameetrid tarneroboti jaoks: Tarnida pakend, Skaneerida pakend, Laadida aku, Tagasi peamenüüsse. Paluge kasutajal sisestada oma valik. Sõltuvalt kasutaja valikust kutsuge vastav TarneRobot klassi meetod (tarnida_tellimus, skaneerida_paki, laadida) või naaske peamenüüsse.
- Kui kasutaja valib PuhastusRoboti, looge uus PuhastusRoboti klassi eksemplar ja paluge kasutajal sisestada roboti nimi. Kuvatakse menüü järgmiste puhastusroboti valikutega: Puhastada tuba, Vali puhastusrežiim, Laadida aku, Tagasi peamenüüsse. Paluge kasutajal sisestada oma valik. sõltuvalt kasutaja valikust kutsuge PuhastusRobot klassi vastavat meetodit (puhastada_tuba, vali_puhastusrežiim, laadida) või naaske peamenüüsse.
- Kui kasutaja valib "Välju", kuvage teade ja väljutakse programmist.
- Kui kasutaja sisestab vale valiku, kuvage veateade ja naaske peamenüüsse.

9.10 Tund 9. Objektorienteeritud programmeerimine. Klassid ja objektid. Instagram

Tunni plaan:

- OOP põhimõistete kordamine. Lihtsate näidete abil rääkida abstraktsetest klassidest ja liidestest.
- Rääkida õpilastele mitmekordsest pärilikkusest ja segamisest.
- Tutvustada õpilastele Instagrami tööd.
- Anda õpilastele praktiline ülesanne, mis hõlmab käsitletud teemasid ja näitab, kuidas Instagram töötab.

Kodutöö teemad tunniks:

- Täiustatud OOP põhimõisted. Abstraktsed klassid ja liidesed. Abstraktsete klasside ja meetodite loomine Pythonis. Abstraktsete klasside abil liideste rakendamine.
- Mitmekordne pärilikkus ja mixinid. Mitmekordse pärilikkuse ja mixinide määratlus ja eesmärk. Kuidas rakendada Pythonis mitmekordset pärilikkust. Eelisjärjekord ja meetodite lahendamise järjekord (MRO).

Tunni struktuur:

- Viia läbi kiire ülevaade teemal "Objektorienteeritud programmeerimine. Klassid ja objektid", millega õpilased on kodus tutvunud, vastates küsimustele materjali selgitamiseks.

- Korrata koos õpilastega klasside ja objektide teemasid OOPis.
- Tutvustada Instagrami, selle rolli tehnoloogiasektoris ja selle mõju ühiskonnale.
- Tutvustada õpilastele, kuidas Instagram kasutab Pythoni keelt serveripoolseks arenduseks (Django: Pythoni veebiraamistik, mida Instagram kasutab) ning andmeanalüüsiks ja masinõppeks (kasutajate käitumise ja kaasamise meetrikate analüüs, soovitus- ja personaliseerimisalgoritmid, pilditunnustamine ja sisu modereerimine) (RegInsights veebileht).
- Tutvustada Instagrami seotud Pythoni teeke ja vahendeid (InstaPy: Instagrami robotite automatiseerimine ja loomine, meeldimiste, kommentaaride ja tellimuste automatiseerimine, postituste ajastamine ja mitme konto haldamine; Instagram-API-python: Instagrami mitteametlik API Pythonile, tarkvara juurdepääs Instagrami andmetele ja funktsioonidele, kohandatud rakenduste ja vahendite loomine Instagrami jaoks; Instagram Scraper: teek Instagrami andmete puhastamiseks, avalike kasutajate profiilide, postituste ja kommentaaride väljavõtmiseks, Instagrami andmete analüüsimiseks ja visualiseerimiseks Pythoni abil) (RegInsights veebileht).
- Rääkida reaalsetest näidetest Pythoni kasutamisest Instagramis (Instagrami serveripoolse osa loomine ja skaleerimine Django abil - miljonite kasutajate ja miljardite interaktsioonide töötlemine, jõudluse optimeerimine ja usaldusväärsuse tagamine; Instagrami andmeanalüüsi ja visualiseerimise tööriistade arendamine - kasutajate kaasatuse ja kasvu jälgimine, kasutajate käitumise suundumuste ja mustrite väljaselgitamine; Instagrami botide loomine ja automatiseerimistööriistade kasutamine Pythoniga) (RegInsights veebileht).
- Pakkuda õpilastele praktilist ülesannet, mis võimaldab kasutajal luua Instagrami profiile, neid redigeerida ja kustutada ning hõlmab läbitud teemasid.
- Viia läbi arutelu lahenduste üle, näidates oma versiooni ülesandest.
- Selgitage õpilastele nende kodutööd, millist materjali tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Instagrami profiil - lihtne näide Instagrami profiilide haldamisest, uue profiili lisamisest, mitme profiili kuvamisest ja muutmisest.

1. Määrake klass nimega *InstagramProfiil*. Lisage meetod *init*, mis initsialiseerib järgmised instantsmuutujad: *kasutajanimi*, *täisnimi*, *elulugu*, *jälgijad*, *following* ja

- postitused*. Rakendage meetod *str*, mis tagastab InstagramiProfiili objekti vormindatud sõne-esituse.
2. Määrake abstraktne klass nimega *AbstractKuvamine*. Lisage abstraktne meetod nimega *kuva_profiilid*, mis võtab parameetrina InstagramiProfiil objektide järjendi.
 3. Määrake klass nimega *LihtneKuvamine*, mis pärib klassi *AbstractKuvamine*. Rakendage meetod *kuva_profiilid*, mis iteratsiiteratiivselt läbib InstagramiProfiil objektide järjendi ja kuvab iga profiili, kasutades meetodit *str*.
 4. Määrake klass nimega *FancyKuvamine*, mis pärib klassi *AbstractKuvamine*. Rakendage meetod *kuva_profiilid*, mis itereerib läbi InstagramiProfiil objektide järjendi ja kuvab iga profiili ebatavalises formaadis emotikonidega.
 5. Klass *InstagramRakendus*: määrake klass nimega *InstagramRakendus*. Lisage meetod *init*, mis algväärtustab profiilide järjendi ja määrab vaikimisi kuvamisstiiliks *LihtneKuvamiseks*.
 6. Rakendage meetod *lisa_profiil*, mis võtab InstagramProfiil objekti ja lisab selle profiilid järjendisse.
 7. Rakendage meetod *muuda_profiil*, mis võtab parameetrina kasutajanimi, leiab sobiva kasutajanimiga profiili ja võimaldab kasutajal muuta kasutajanimi, täisnimi ja elulugu.
 8. Rakendage *kustuta_profiil* meetod, mis võtab parameetrina kasutajanimi ja kustutab profiilid järjendist sobiva kasutajanimiga profiili.
 9. Rakendage meetod *kuva_profiil*, mis kontrollib, kas profiilide nimekirjas on mingeid profiile, ja kui on, siis kutsub *kuva_profiil* meetodit.
 10. Rakendage *seadista_kuvamise_stiil* meetod, mis võtab parameetri stiil ja seab kuvamine kas *LihtneKuvamine* või *FancyKuvamine* vastavalt kasutaja sisestusele.
 11. Looge peafunktsioon *main()*, mis loob *InstagramRakendus* objekti. Looge tsükkel, mis kuvab menüü, milles on kasutaja jaoks valikud profiilide lisamiseks, muutmiseks, kustutamiseks, kuvamiseks, kuvamisstiili muutmiseks või rakendusest väljumiseks. Kutsuge kasutaja sisendi põhjal *InstagramRakendus* objektilt asjakohased meetodid ning töödelge vastavalt vajadusele kasutaja sisendit ja väljundit.
 12. Kutsuge *main()* funktsioon ja proovige lisada oma Instagrami profiil.

9.11 Tund 10. Graafilise kasutajaliidese (GUI) programmeerimine Tkinteriga.

Tervishoid - Nortal

Tunni plaan:

- Tutvustada õpilastele graafilise kasutajaliidese moodulit Tkinter ja selle kasutamise võimalusi.
- Tutvustada õpilastele Tkinteri põhiakna loomist, rakendustsükli, üldisi vidinaid ja paigutusjuhte.
- Õpetada õpilastele sündmusepõhist programmeerimist.
- Rääkida tervishoiutööstusest ja Pythoni kasutamisest selles.
- Rääkida õpilastele Nortalist.
- Anda õpilastele praktiline ülesanne, mis hõlmab käsitletud teemasid ja demonstreerib tervishoiusektori probleemi lahendamist.

Kodutöö teemad tunniks:

- Sissejuhatus Tkinterisse. Tkinteri paigaldamine. Lühike ülevaade Tkinterist. Miks kasutada Tkinterit graafiliste kasutajaliideste loomiseks. Võrdlus teiste Pythoni GUI moodulitega.
- Tkinteri põhiaken ja rakendustsükkel. Põhiakna loomine. Rakendustsükli käivitamine.
- Tkinteri vidinad ja paigutushaldurid. Üldised vidinad: Button, Label, Entry, Text ja Canvas. Paigutusjuhid: Pack, Grid ja Place.
- Sündmusepõhine programmeerimine Tkinteris. Sündmuste sidumine vidinatega. Kohandatud sündmuste käsitlemine ja suhtlemine vidinatega.

Tunni struktuur:

- Viia läbi kiire ülevaade teemast "Graafilised kasutajaliidesead Tkinteriga", mida õpilased on kodus lugenud, vastates küsimustele materjali selgitamiseks.
- Selgitada õpilastele tervishoiutööstust, tehnoloogia rolli tervishoius ja Pythoni kasutamist tervishoius.
- Anda näiteid ülesannetest, milleks Pythoni kasutatakse tervishoius (andmeanalüüs ja visualiseerimine: meditsiiniandmete, patsiendiandmete ja ravitulemuste analüüs, trendide, muustrite ja ideede visualiseerimine; masinõpe ja tehisintellekt: prognoosiv modelleerimine ja diagnoosimine, loomuliku keele töötlus ja meditsiiniline tekstianalüüs; tervishoiuprotsesside ja ülesannete automatiseerimine:

vastuvõtuaegade ja patsiendiandmete haldamine, kliiniliste töövoogude ja haldusülesannete optimeerimine) (DataCamp veebileht).

- Rääkida õpilastele ettevõttest Nortali, selle rollist tervishoiuvaldkonnas, Nortali poolt tervishoiuvaldkonnas pakutavatest tehnoloogiatest ja teenustest.
- Anda näiteid Nortali kasutatavate Pythoni moodulite ja vahendite kohta: Pandas ja NumPy, Scikit-learn ja TensorFlow, Flask ja Django, APLUS (Wornow et al., 2023).
- Tutvustada reaalseid näiteid Pythoni kasutamise kohta Nortali tervishoiuprojektides (meditsiiniliste andmete analüüsiplatvormide arendamine, masinõppe mudelite rakendamine diagnoosimiseks ja raviks, tervishoiu jaoks kohandatud rakenduste ja tööriistade loomine) (DataCamp veebileht).
- Kutsuda õpilasi lahendama praktilist ülesannet, mis esitleb haiglakohtade reguleerimise vahendit ja sisaldab käsitletud teemasid.
- Viia läbi arutelu lahenduste üle, näidates oma versiooni ülesandest.
- Selgitada õpilastele nende kodutööd, millist materjali tuleb järgmiseks tunniks ette valmistada.

Praktiline ülesanne:

Haigla voodikohtade haldus - lihtne haigla voodikohahaldussüsteem, mis võimaldab haigla personalil hallata voodite kättesaadavust, patsientide vastuvõtmist ja väljakirjutamist.

13. Importige vajalikud moodulid (importige tkinter kui tk, importige messagebox-i klass moodulist tkinter).
14. Defineeri põhiklass *HaiglaVoodikohtadeHaldamine*: lisage meetod `__init__`, mis initsialiseerib põhiakna, määrab pealkirja, initsialiseerib voodikohtade arvu ja patsientide nimekirja ning kutsub peamenüü meetodit.
15. Rakendage *peamine_menüü* meetod, mis loob nupud patsientide registreerimiseks, patsientide väljasaatmiseks, voodikohasaadavuse, vabade voodikohtade graafiku ja väljumise jaoks ning määrab igale nupule vastavad meetodid.
16. Rakendage *patsiendi_vastuvõtuaken* meetod: looge uus aken, millel on sildid ja kanded patsiendi nime, vanuse ja seisundi jaoks ning nupp teabe esitamiseks. Lisage esitamisnupule käsk, mis kutsub *võtta_patsient_vastu* meetodit koos sisestatud teabega.
17. Rakendage *võtta_patsient_vastu* meetod: kontrollige, kas on vabad voodikohad, kui on, vähendage vabade voodikohtade arvu ja lisage patsient patsientide nimekirja. Kui voodeid ei ole saadaval, näidake veateadet, kasutades `tk.messagebox.showerror`.

18. Rakendage *tühjendada_patsiendi_aken* meetod: kontrollige, kas patsiendid on olemas, kui ei ole, näidake veateadet. Looge uus aken kasutades OptionMenu patsiendi valimiseks ja Button patsiendi vabastamiseks. Lisage nupule käsk, mis kutsub *tühjendada_patsiendi* meetodit valitud patsiendi nimega.
19. Rakendage *tühjendada_patsiendi* meetod: eemaldage patsient patsientide nimekirjast, suurendage vabade voodikohtade arvu ja sulgege aken.
20. Rakendage *vaata_voodi_saadavus_aken* meetod: looge uus aken, mille silt näitab vabade voodikohtade arvu ning kasutage Canvas vidin, mis kuvab vastuvõetud patsientide nimekirja. Kui patsiendid on olemas, kuvatakse nende andmed Canvas vidin, vastasel juhul kuvatakse teade, et ühtegi patsienti ei ole vastu võetud.
21. Rakendage meetod *voodi_saadavus_graafik_aken*: Looge uus aken koos Canvas vidinaga, et kuvada hõivatud ja vabade voodikohtade tulpdiaagrammi. Joonistage Canvas vidinale riskülikud ja tekst, et kujutada hõivatud ja vabade voodikohtade arvu.
22. Blokis `__main__` initialiseeri `tk.Tk()` põhiaken `root`'ina. Instantseerige klass `HaiglaVoodikohtadeHaldamine`, mille argumentiks on `root`. Käivitage põhitsüklil `root.mainloop()` abil.

9.12 Tund 11. Sissejuhatus PyGame'i. Boeing

Tunni plaan:

- Tutvustada õpilastele PyGame'i moodulit ja selle kasutusvõimalusi.
- Näidata õpilastele PyGame'i põhitõdesid.
- Tutvustada õpilastele PyGame'i graafikat ja sündmuste käsitlemist.
- Tutvustada õpilastele Boeingi ettevõtet.
- Anda õpilastele praktiline ülesanne, mis hõlmab käsitletud teemasid ja demonstreerib Boeingi väljakutset graafilise näite abil.

Kodutöö teemad tunniks:

- Sissejuhatus PyGame'ile. PyGame'i paigaldamine. Miks kasutada PyGame'i mängude arendamiseks. Võrdlus teiste Pythoni mängumoodulitega.
- PyGame'i põhitõed. PyGame'i seadistamine. Mänguakna loomine. Mängutsüklil ja sündmuste käsitlemine. PyGame'i pinna ja ekraani haldamine. Piltide laadimine ja renderdamine. PyGame'i koordinaatsüsteemide mõistmine.

- Kujude ja teksti joonistamine. Ristkülikud, ringid ja hulknurgad. Teksti kuvamine PyGame'i fontidega.
- Kasutaja sisendi ja sündmuste töötlemine. Klaviatuuri ja hiire sündmused. Kokkupõrgete tuvastamine. Mänguobjektide animatsioon.

Tunni struktuur:

- Viia läbi kiire ülevaade materjalist "Sissejuhatus PyGame'i", millega õpilased olid kodus tuttavad, vastates küsimustele materjali selgitamiseks.
- Tutvustada õpilastele Boeingi ning selle peamisi tooteid ja teenuseid.
- Tutvustada õpilastele, kuidas Python'i kasutatakse lennunduses (andmeanalüüs ja visualiseerimine, simulatsioon ja modelleerimine, automatiseerimine ja testimine, masinõpe ja tehisintellekt) (Miao, 2022).
- Tutvustada õpilastele Boeingis kasutatavaid Pythoni teeke ja vahendeid (NumPy ja SciPy; Matplotlib ja Seaborn andmete visualiseerimiseks; pandas; scikit-learn masinõppe jaoks; TensorFlow ja Keras süvaõppe jaoks; Jupyter Notebooks interaktiivseks arvutamiseks ja esitlusteks) (Pundir 2020).
- Pythoni konkreetsete kasutusvõimaluste tutvustamine Boeingis (lennuandmete ja jõudlusnäitajate analüüs, suundumuste ja muustrite visualiseerimine andmetes, õhusõidukite süsteemide ja jõudluse simuleerimine, lennutrajektooride modelleerimine ja optimeerimine, tarkvara ja riistvara testimise protsesside automatiseerimine, inseneridele kohandatud tööriistade väljatöötamine, hooldusnõuete analüüs ja prognoosimine ning tootmisprotsesside parandamine).
- Kutsuda õpilasi üles lahendama praktilist ülesannet, mis visualiseerib õhusõiduki lendu seatud punktidel ja sisaldab käsitletud teemasid.
- Viia läbi arutelu lahenduste üle, näidake nende versiooni probleemi lahendusest.

Praktiline ülesanne:

Lihtne lennutee visualiseerimine - lihtne lennutee visualiseerimine PyGame'i abil.

1. Importige vajalikud moodulid (importige pygame, sys ja random).
2. Initsialiseerige Pygame funktsiooniga pygame.init(). Määrake ekraani mõõtmed, ruudustiku suurus ja värvid visualiseerimiseks.
3. Looge peamine ekraan. Looge Pygame'i kuvapind määratud ekraani mõõtmetega. Määrake Pygame'i aknale pealkiri.

4. Laadige ja muutke lennukite, mägede ja lennujaamade kujutiste suurust, kasutades funktsioone `pygame.image.load()` ja `pygame.transform.scale()`. Hankige lennuki pildi riskülik edasiseks manipuleerimiseks.
5. Määrake ruudustiku koordinaadid ja takistused: looge tühi nimekiri, et salvestada lennu punktid. Genereerige määratud arv takistusi ja lennuvälju juhuslikel positsioonidel.
6. Määrake joonistusfunktsioonid. Looge funktsioonid ruudustiku, lennu, lennuki, punktide, takistuste ja lennujaamade joonistamiseks. Kasutage igas funktsioonis sobivaid Pygame'i joonistusfunktsioone, nagu `pygame.draw.line()`, `pygame.draw.circle()` ja `screen.blit()`.
7. Initsialiseerige kellavõti objekti, mis aitab kontrollida mängutsükli kaadrisagedust. Selleks kasutatakse funktsiooni `pygame.time.Clock()`. Initsialiseerige muutuja *edenemine* väärtuseks 0. Seda muutujat kasutatakse lennuki liikumise kontrollimiseks mööda rada. Initsialiseeri muutuja *praegune_segment* väärtuseks 0. Seda muutujat kasutatakse selleks, et jälgida, millist raja segmenti mööda lennuk parajasti lendab.
8. Looge Pygame'i akna põhitsükkel kasutades `while True` avaldust. Tühjendage ekraan valge taustaga ja joonistage ruudustik, takistused ja lennujaamad.
9. Käsitlege kasutaja sündmus, iteratsiooniga läbi sündmuste funktsioonis `pygame.event.get()`, kasutades `for`-tsükli. Kui sündmuse tüüp on `pygame.QUIT`, peaks mäng lõpetama. Kui sündmuse tüüp on `pygame.MOUSEBUTTONDOWN`, tuleb lisada uus punkt *tee_punktid* järjendisse.
10. Joonistage lennupunktid määratud värviga.
11. Kontrollige, kas *tee_punktid* järjend on vähemalt kaks punkti. Kui jah, siis tehke järgmised toimingud: joonistage rada, kasutades funktsiooni *joonista_tee(tee_punktid)*; suurendage muutujat *edenemine* 0,01 võrra; kontrollige, kas *edenemine* on suurem kui 1. Kui jah, siis nullige *edenemine* väärtuseks 0 ja suurendage *praegune_segment* 1 võrra. Seejärel kontrollige, kas *praegune_segment* on suurem või võrdne *tee* segmentide arvuga (st *tee_punktid* järjendi pikkus miinus 1). Kui jah, siis nullige *praegune_segment* väärtuseks 0. Arvutage lennuki asukoht mööda praegust segmenti, kasutades muutujaid *algus_punkt*, *lõpp_punkt* ja *edenemine*. Joonistage lennuk arvutatud positsioonile.
12. Uuendage ekraani `pygame.display.flip()` abil ja kontrollige kaadrisagedust `clock.tick(30)` abil.

9.13 Tund 12. Hinnete parandamine. Tagasiside

Tunni plaan:

- Paku õpilastele täiendavaid individuaalseid ülesandeid, et parandada nende hindeid.
- Koguda tagasisidet kursuse kohta, anda õpilastele tagasisidet.

Tunni struktuur:

- Anda lühiülevaade kursusel käsitletavatest teemadest.
- Koguda üliõpilastelt tagasisidet kursuse korralduse kohta, nende ootusi enne ja pärast kursust. Rääkige õpilastele, millised püstitatud eesmärgid saavutati ja millised mitte.
- Arutada õpilaste järgmisi samme, näiteks edasist õppimist või kursusel õpitud oskuste praktilist rakendamist.
- Anda õpilastele iseseisvat ülesannet, mis on juhuslikult antud ülesanne läbitud kursusest ja mis hõlmab mis tahes kursusel käsitletud teemasid.
- Vaadata täidetud ülesanded läbi, esitada täpsustavaid küsimusi teadmisi kontrollimiseks, parandada kursuse hindeid.

KOKKUVÕTE

Lõputöös püstitati selline probleem - puudub eestikeelne programmeerimise sissejuhatav koolikursus, mis näitaks IT-valdkonda erinevatest vaatenurkadest, annaks gümnaasiumiõpilastele alustadmised programmeerimisest, pakuks huvitavaid praktilisi ülesandeid ja motiveeriks neid valdkonda edasi õppima. Tartu Ülikooli Narva kolledži IT-üliõpilaste seas viidi läbi küsitlus, mis näitas, et õpilaste motivatsiooni ja huvi programmeerimise vastu ei toeta kooli õppekava, kuid nad pidasid selliseid kursusi koolides vajalikuks.

Algajatele õpilastele programmeerimise õpetamise keeleks valiti Python, kuna see on populaarne ja algajatele kergesti mõistetav keel, mida kasutatakse mitmesugustes valdkondades. Just see võib aidata saavutada häid tulemusi pärast kursuse läbimist.

Vaadeldi olemasolevaid algajatele mõeldud Pythoni programmeerimiskursuseid, nagu Tartu Ülikooli kursused, kutsehariduskeskuse kursus ja tuntud õppeplatvormi Coursera kursus. Kõik leitud olemasolevad lahendused keskenduvad programmeerimise baastadmiste omandamisele, samal ajal kui on vajadus kursuse järele, mis keskenduks valdkonna tutvustamisele, õpilaste huvi suurendamisele ja motivatsioonile edasiseks õppimiseks.

Koolikursuse loomiseks valiti erinevaid interaktiivse õppe meetodeid ja ebatavalisi lahendusi. Näiteks toimuvad tunnid pööratud õppe formaadis, kus õpilased õpivad põhimaterjali kodus kodutööna ja tunniaeg on praktikale pühendatud. Tunnis töötavad õpilased paarides, õpivad paariprogrammeerimist, lahendavad interaktiivseid ülesandeid ja pärast toimub ühine arutelu võimalike lahenduste üle. Interaktiivsed õppemeetodid suurendavad õpilaste huvi õppimise vastu, muudavad õppimise vähem formaalseks, tõstavad õpilaste enesekindlust ja motivatsiooni.

Bakalaureusetöö raames loodi koolikursuse programm, mis on planeeritud 36 tunniks ja koosneb 12 õppetunnist, 1,5 tundi iga, millest 10 on praktilised. Kursuse esimene tund on sissejuhatav, viimane on tagasiside kogumiseks ja hindamise korrigeerimiseks neile, kes seda vajavad.

Kursuse programm seisneb selles, et olemasolevate lahenduste põhjal valiti Pythoni õppimiseks teemad, mis hõlmavad nii algtaseme teemasid, objektorienteeritud

programmeerimisse sissejuhatust kui ka tööd graafiliste teekidega. Lisaks valiti 10 valdkonda/tuntud ettevõtet, mis kasutavad oma töös Pythonit, et mõista, kus ja kuidas programmeerimist kasutatakse. Pythoni teemad ja töövaldkonnad ühendati autori poolt nii, et kursusel on 10 praktilist tundi, millest igaüks õpetab ühte programmeerimisteemat, räägib ettevõttest ja materjal kinnitatakse praktilise ülesandega, mis näitab ettevõtte tööd, keskendub tunnis õpitud teemadele ja on interaktiivses vormis, kus kasutaja (õpilane) suhtleb programmiga.

Iga õppetunni jaoks kirjutati eesmärgid, mis näitavad, mida antud õppetunniga peaks läbima. Iga õppetunni jaoks kirjeldati üksikasjalikult Pythoni teemad, millega õpilased peaksid iseseisvalt tutvuma. Iga õppetunni jaoks koostati õppetunni kava, kus õpilased tutvuvad valdkonna/ettevõttega ja lahendavad praktilise ülesande. Iga õppetunni jaoks koostati praktiline ülesanne, üksikasjalik juhend selle täitmiseks ja pakuti lahenduse variant. Kuuenda tundi esitluses on kombineeritud erinevad õppemeetodid ja allikad. Esitluse koostamiseks kasutati Pythoni dokumentatsiooni, programmeerimisõpikut "Progeõpik", videoloenguid, teemalisi artikleid ja interaktiivseid harjutusi, mis on saadaval erinevatel platvormidel. Ülesanne on esitatud etappide kaupa koos lahendusega. Antud esitlus on erinevate meetodite ühendamine, sest selline lähenemine suudab kõige paremini kõita ja säilitada õpilaste tähelepanu ja huvi. Kõik esitlused on näited sellest, kuidas tund saab erinevalt käsitleda, ja neid saab vastavalt õpetaja eelistustele muuta. Ülesannete lahendustega saab tutvuda Git repositooriumis lingil https://github.com/SKatjaa/Loputoo_2023.

Interaktiivsed ja mittestandardsete õppemeetodite kasutamine, tutvumine erinevate valdkondadega, kus Pythonit kasutatakse, selle kasutusviisidega ning Pythoni õppimine interaktiivsete ja mänguliste ülesannete kaudu peaks tõstma õpilaste motivatsiooni IT-valdkonna õppimiseks. Selle kursuse eesmärk on näidata gümnaasiumiõpilastele, et programmeerimist kasutatakse praktiliselt igas eluvaldkonnas ning valides programmeerimise, saavad nad töötada täiesti erinevates valdkondades ja valida just neile huvipakkuvad erialad. Interaktiivsed ülesanded praktikaks peaksid andma õpilastele võimaluse suhelda nende poolt loodud programmiga, proovida erinevaid probleemilahendusi ja tunda end uuritava ettevõtte töötajana, mis suurendab nende huvi ja motivatsiooni.

Kokkuvõttes on eesmärgid saavutatud: koostatud kursus erineb olemasolevatest lahendustest ja keskendub seatud eesmärkide saavutamisele ning tuvastatud probleemi lahendamisele. Eestikeelne koolikursus on aktuaalne seoses koolide üleminekuga eestikeelsele õppele.

KURSUSE TULEVIK

Antud kursust pole praktikas testitud ega seni koolis läbi viidud, kuid selle läbiviimiseks on olemas kõik vajalik.

Kursuse eesmärgid on selgelt määratletud, iga tunni eesmärgid on kirjas, õppetöö teemad on nimetatud, iga tunni plaan, üksikasjalikud juhised praktiliste ülesannete täitmiseks ja nende lahendamise võimalused.

Kuna iga õpetaja eelistab oma õpetamismeetodeid, näiteks klassikalisi esitlusi, isiklike kogemusi, videoõpet, suulist jutustamist jne, on loodud mõned näited sellest, kuidas tundide esitus võib välja näha. Esitlused koostati kolme kursuse praktilise tunni jaoks (valiti tunnid 2, 3 ja 6), et näidata, kui erinevalt võib kursust esitada. 2. tunni jaoks oli tehtud klassikaline esitus, kus kasutati Programmeerimise õpiku materjali (<https://progeopik.cs.ut.ee/index.html>). Olid toodud koodi näited ning praktiline ülesanne oli jagatud osadeks ning järk-järgult näidatud koos lahendusega. 3. tunni jaoks oli esitluses kasutatud erinevaid videoid Youtube-ist, mille kestus on 5-15 minutit. Praktiline ülesanne oli antud tervena pdf-failina ning selle lahendus on antud etappide kaupa. Videotega tekkis probleem, kuna eesti keeles materjalid puudusid, seega enamik neist on inglise keeles. Mõned videod olid asendatud artiklitega, sest vajalikku materjali oli raske video formaadis leida. Esitlustega saab tutvuda lingil https://github.com/SKatjaa/Loputoo_2023/tree/master/esitlused.

Kursuse praktilisel läbiviimisel on oluline koguda õpilastelt tagasisidet. Tuleb pöörata tähelepanu sellistele aspektidele nagu, kui palju on õpilaste arusaam IT-valdkonnast, selle võimalustest ja programmeerimisest enne ja pärast kursuse läbimist muutunud; kui arusaadav oli programmeerimisega seotud teemad, kas neid oli keeruline või lihtne mõista. Tuleb tähelepanu pöörata sellele, kui palju aega võtab iga õppetunni osa ja kas see vastab plaanile.

Sõltuvalt saadud tagasisidest võivad kursuse läbiviimise mõned aspektid muutuda, säilitades samas algse eesmärgi ja struktuuri.

KASUTATUD KIRJANDUS

Haridus- ja Teadusministeerium. (2023, 17. veebruar). *Eestikeelsele õppele üleminek*. <https://www.hm.ee/uleminek> , viimati vaadatud 20.02.23

Rocker Yoel, S., & Dori, Y. J. (2022). FIRST High-School Students and FIRST Graduates: STEM Exposure and Career Choices. *IEEE Transactions on Education, Education, IEEE Transactions on, IEEE Trans. Educ*, 65(2), 167–176. <https://doi-org.ezproxy.utlib.ut.ee/10.1109/TE.2021.3104268>

Polishchuk Olena. (2020). The problem of high school students' profession choice: theoretical discourse. *Науковий Вісник Південноукраїнського Національного Педагогічного Університету Імені К. Д. Ушинського*, 134(4), 117–127. <https://doi-org.ezproxy.utlib.ut.ee/10.24195/2617-6688-2020-4-15>

Topalska, R., & Mladenova, M. (2020). Interactive Methods in Information Technology and Computer Modeling Training. *Knowledge: International Journal*, 42(2), 433–438.

Bakhreddinova, M. (2020). The interactive methods and principles of foreign language teaching, *International Journal on Integrated Education*, Volume 3, Issue I

Shein, E. (2015). Python for Beginners. *Communications of the ACM*, 58(3), 19–21. <https://doi-org.ezproxy.utlib.ut.ee/10.1145/2716560>

de Wit, S. de, Hermans, F., & Aivaloglou, E. (2021). Children's Implicit and Explicit Stereotypes on the Gender, Social Skills, and Interests of a Computer Scientist. *Proceedings of the 17th ACM Conference on International Computing Education Research*, 239–251. <https://doi-org.ezproxy.utlib.ut.ee/10.1145/3446871.3469753>

Gutttag, J.V. (2013) *Introduction to Computation and Programming in Python*, MIT Press

Tartu Ülikool arvutiteaduse instituut (2011). *TÜ Arvutiteaduse instituudi programmeerimise algkursuse õpik*. <https://progeopik.cs.ut.ee/> , viimati vaadatud 15.03.23

Coursera. *Programming in Python*. <https://www.coursera.org/learn/programming-in-python>, viimati vaadatud 15.03.23

Kutsehariduskeskus. *Kursuse Õppekava*. https://www.kutsehariduskeskus.ee/sites/kutsehariduskeskus.ee/files/oppekava_programmerimine.pdf, viimati vaadatud 16.03.23

Nawahdah, M., Taji, D., & Inoue, T. (2015). Collaboration leads to success: A study of the effects of using pair-programming teaching technique on student performance in a Middle Eastern society. *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Teaching, Assessment, and Learning for Engineering (TALE), 2015 IEEE International Conference On*, 16–22. <https://doi.org.ezproxy.utlib.ut.ee/10.1109/TALE.2015.7386009>

STOFFOVÁ, V., & HORVÁTH, R. (2021). Project-Based and Problem-Based Teaching Programming. *ELearning & Software for Education*, 1, 545–552. <https://doi.org.ezproxy.utlib.ut.ee/10.12753/2066-026X-21-068>

ChatGPT. (2023). *How to use interactive teaching methods in teaching programming?* <https://github.com/meriamchabir/NLP-QA-Chatbot>, viimati vaadatud 19.03.2023

Riigiteataja. (2002). *Põhikooli ja gümnaasiumi riiklik õppekava*. <https://www.riigiteataja.ee/akt/12888846>, viimati vaadatud 19.03.2023

Tartu Ülikooli ÕIS. (2020). *Programmeerimine koolis (Python) (3EAP) MTAT.03.302*. <https://ois2.ut.ee/#/courses/MTAT.03.302/version/41436d6e-9e3e-3cdc-3967-096a4dc304ab/details>, viimati vaadatud 27.03.2023

Tartu Ülikooli ÕIS. (2020). *Programmeerimine (6 EAP) SVNC.00.054*. <https://ois2.ut.ee/#/courses/SVNC.00.054/version/cbb4948b-0d90-ef62-652a-37c49c579929/details>, viimati vaadatud 27.03.2023

Towardsdatascience. (2021). *Top 16 Python Applications in Real World. Towards Data Science.* <https://towardsdatascience.com/top-16-python-applications-in-real-world-a0404111ac23>, viimati vaadatud 27.03.2023

LinkedIn. (2022). *How Industries are using Python.* <https://www.linkedin.com/pulse/how-industries-using-python-sonal-jaiswal/>, viimati vaadatud 29.03.2023

HashDork. (2022). *Tesla's AI team talks about their favorite programming languages.* <https://hashdork.com/tesla-ai-programming-languages/>, viimati vaadatud 29.03.2023

Towards Data Science. (2021). *10 World-Class Companies Using Python.* <https://towardsdatascience.com/10-world-class-companies-using-python-26cde24919a8>, viimati vaadatud 29.03.2023

Chen, L., & Hsu, Y. (2019). The use of Python in finance: A review of the literature. *Journal of Applied Finance & Banking*, 9(4), 1-11.

Bissell, C., & Fox, D. (2017). Python for aerospace engineering: Implementing and testing a UAV controller. *Journal of Aerospace Information Systems*, 14(12), 680-689.

Nortal. (2021). *The Future Delivered Seamlessly.* <https://nortal.com/>, viimati vaadatud 30.03.2023

Starship Technologies. (2021). *The global leader in autonomous delivery.* <https://www.starship.xyz/company/>, viimati vaadatud 30.03.2023

Python Software Foundation. *The Python Standard Library. Python 3.10.0 documentation.* <https://docs.python.org/3/contents.html>, viimati vaadatud 20.04.2023

Institute of Cybernetics at Tallinn University of Technology. *YFX0500 Sissejuhatus programmeerimisse Pythoni baasil.* <https://www.ioc.ee/~dima/YFX0500/>, viimati vaadatud 20.04.2023

LinkedIn. (2022, January 14). *How does Google use Python in its technology stack?*
LinkedIn.<https://www.linkedin.com/pulse/how-does-google-use-python-its-technology-stack-muqadas-rasheed-/>, viimati vaadatud 20.04.2023

Google Cloud. *Google Cloud Python Libraries*.
<https://cloud.google.com/python/references/libraries>, viimati vaadatud 20.04.2023

Tutorialspoint. (2021). *Is Python Good for Developing Games? Why or Why Not?*
<https://www.tutorialspoint.com/is-python-good-for-developing-games-why-or-why-not#:~:text=Python%20does%20allow%20game%20creation,with%20Python%20is%20not%20difficult>, viimati vaadatud 20.04.2023

Game Rant. (2021). *10 best games that use Python*. <https://gamerant.com/best-games-that-use-python/#the-sims-4>, viimati vaadatud 20.04.2023

Tesla, Inc. *Senior Full Stack Engineer, Autopilot AI - Python, React, ML (181633)*.
<https://www.tesla.com/careers/search/job/senior-full-stack-engineer-autopilot-ai-python-react-ml-181633>, viimati vaadatud 20.04.2023

JP Morgan Chase & Co. *Projects*. <https://jpmorganchase.github.io/projects>, viimati vaadatud 20.04.2023

eFinancialCareers. (2020). *JPMorgan providing Python training to traders and analysts*.
<https://www.efinancialcareers.co.uk/news/2020/06/jpmorgan-python-training-analysts-and-traders>, viimati vaadatud 20.04.2023

Spotify Engineering. (2013). *How we use Python at Spotify*.
<https://engineering.atspotify.com/2013/03/how-we-use-python-at-spotify/>, viimati vaadatud 20.04.2023

Amazon Web Services. *What is Python? Amazon Web Services*.
<https://aws.amazon.com/ru/what-is/python/>, viimati vaadatud 20.04.2023

ROS Wiki. rospy - *ROS Python Client Library*. <http://wiki.ros.org/rospy>, viimati vaadatud 20.04.2023

Fraanje, R., Koreneef, T., Le Mair, A., & de Jong, S. (2016). Python in robotics and mechatronics education. *2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM), Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM), 2016 11th France-Japan & 9th Europe-Asia Congress On*, 014–019. <https://doi-org.ezproxy.utlib.ut.ee/10.1109/MECATRONICS.2016.7547108>

RegInsights. (2021). *The Role of Python in the Making of Instagram*. <https://www.regenesys.net/reginsights/the-role-of-python-in-the-making-of-instagram/#:~:text=To%20help%20the%20scaling%20and,beyond%2Dthe%2Dborder%20capabilities,> , viimati vaadatud 20.04.2023

Wornow, M. (1), Gyang Ross, E. (2,3), Callahan, A. (2), & Shah, N. H. (2,4,5,6). (2023). APLUS: A Python library for usefulness simulations of machine learning models in healthcare. *Journal of Biomedical Informatics*, 139. <https://doi-org.ezproxy.utlib.ut.ee/10.1016/j.jbi.2023.104319>

DataCamp. (2019). *Python in Healthcare: AI Applications in Hospitals*. McGowan, L. <https://www.datacamp.com/blog/python-in-healthcare-ai-applications-in-hospitals>, viimati vaadatud 20.04.2023

Miao, X. (2022). Punctuality Analysis of Aviation Industry Based on Python Data Mining Algorithm. *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Electrical Engineering, Big Data and Algorithms (EEBDA), 2022 IEEE International Conference On*, 976–978. <https://doi-org.ezproxy.utlib.ut.ee/10.1109/EEBDA53927.2022.9744933>

Pundir, A. (2020). *Python for Mechanical and Aerospace Engineering*. CRC Press.