



TeaMe



Rakenduste loomise ja programmeerimise alused

Valikkursus gümnaasiumitele

Üldosa

Jüri Vilipõld, Kersti Antoi, Irina Amitan

Õppematerjalid on valminud Eesti Teadusagentuuri programm TeaMe raames ning rahastatud Euroopa Sotsiaalfondist. TeaMe on teadust ja tehnoloogiat populariseeriv programm, mille üheks eesmärgiks on uute loodus-, täppis- ja tehnikateaduste valdkonna õppematerjalide väljatöötamine üldhariduskoolile.

Lisainfo: <http://www.etag.ee/teame>.

Kursuse kontseptuaalne kirjeldus

Kursuse olulisus ühiskonnas

Käesoleval ajal on kõik inimtegevuse valdkonnad, alates teadusest ja tehnoloogiast ning lõpetades kunsti, spordi, olme ja meelelahutusega, väga tihedalt seotud infotehnoloogiliste vahendite kasutamisega. Võiks öelda, et käesoleva ajastu ja infoühiskonna keskseteks mõisteteks on informatsiooni töötlemine ja programmjuhtimisega süsteemid. Üheks tähelepanuväärseks aspektiks on infotehnoloogia ja kommunikatsioonitehnoloogia ühtesulamine. Piltlikult öeldes on tänapäeval telefonist saanud arvuti ja arvutist telefon.

Kõik see eeldab selliste töötajate hulga kasvu, kes suudavad projekteerida ja ehitada erinevaid programmjuhtimisega seadmeid ning luua nende jaoks tarkvara ja rakendusi. Tarkvara loomisega seotud tegevused on saanud mitmete uute erialade põhisisuks. Üha enam kasvab vajadus spetsialistide järele, kelle ametiks on arvuti tarkvara loomine, haldamine ja hooldamine. Tänapäeval peab praktiliselt kõikidel erialadel mitte ainult oskama kasutada IKT vahendeid vaid suutma püstitada ülesandeid oma valdkonna probleemide lahendamiseks ja osalema vastavates projektides. Sellepärast peab juba koolidest alates pöörama oluliselt suuremat tähelepanu arendustöö aluste õpetamisele.

Infotehnoloogia õpetamise seisukohast võib eristada kolme omavahel seotud valdkonda:

- **Infotöö:** informatsiooni kasutamine ja loomine programmjuhtimisega süsteemide abil. Siia kuuluvad sellised tegevused nagu tekstitöötlus, arvutused ja analüüs, arvutigraafika, informatsiooni hankimine ja korrastamine, kommunikatsioon jmt.
- **Arendustöö:** rakenduste, infosüsteemide ja tarkvara loomine, arendamine ja haldamine. Põhitegevused: modelleerimine, süsteemianalüüs, disain, algoritmimine ja programmeerimine.
- **Programmjuhtimisega süsteemide (PJS) arhitektuur:** PJS töökorraldus, riist- ja süsteemi-tarkvara, informatsiooni esitusviisid jmt.

Infotöö ja programmjuhtimisega süsteemide arhitektuuri alaseid oskusi on tänapäeval vaja kõikidel gümnaasiumi lõpetajatel tööle või edasi õppima asumisel. Käesoleval ajal õpetatakse kutse-, kõrg- ja ülikoolides sageli mitmesuguseid informaatikaaineid, mis sisuliselt kujutavad endast nõ taseandus-õpet, kuna paljude õppima asujate oskused nimetatud valdkondades ei ole piisavad. Ehk oleks otstarbekas alustada teatud korrastamist ja standardiseerimist juba koolide informaatikast, vähemalt gümnaasiumide lõpetajate jaoks: määratleda vajalikud valdkonnad ja ka nõuded tulemustele.

Üleeuroopaline ja rahvusvaheline arvutikasutaja oskustunnistuse süsteemi näol (ECDL – *The European Computer Driving Licence* <http://www.ecdl.ee/AO.html>) on standardiseerimise alused täiesti olemas. Kahjuks ei rakendata seda praegu eriti meie koolides. Seitsmest moodulist: Info- ja sidetehnoloogia mõisted, Arvuti kasutamine ja failihaldus, Tekstitöötlus, Arvutustabelid, Andmebaaside kasutamine, Esitlus, Veebisirvimine ja suhtlus vastavad gümnaasiumi lõpetajate minimaalsete oskuste nõuetele. ECDL dokumentides on määratletud mõiste „digitaalne kirjaoskus“.

Infotöö aluste õpetamise korrastamine ja viimine vajalikule tasemele oleks väga oluliseks aluseks ka arendustöö, sh vaadeldava aine, õpetamisel.

Arendustöö õpetamine meie koolides seisneb praktiliselt programmeerimise (sageli lihtsalt programmeerimiskeele) õpetamises, mida tehakse käesoleval ajal ainult IT-õppesuunaga koolides.

Taolist tendentsi on täheldatud ka mujal maailmas. Eriti tõsist tähelepanu pööratakse sellele viimasel ajal infotehnoloogia juhtriigis – USA-s. Sealne arvutiõpetajate assotsiatsioon (CSTA – <http://csta.acm.org>), kes tegutseb ACM (<http://www.acm.org>) haldusalas, on viimastel aastatel koostanud koolide (K-12) arvutiõpetuse tüüpõppekavasid, mis üha rohkem on orienteeritud arendustööle. Viimases versioonis (2011) on ette nähtud järgmised ained, mille kohta on toodud detailed kirjeldused:

- *Human Computer Interaction* (20 tundi – üldised põhimõtted, riistvara jmt)
- *Problem Solving* (25 tundi – ülesannete lahendamine arvutil, andmed ja algoritmid)
- *Web Design* (30 tundi – veebidisain, JavaScripti kasutamine)
- *Introduction to Programming* (35 tundi – programmeerimine keeles Scratch)
- *Robotics* (40 tundi – robotika, Lego Mindstorms jmt)
- *Computing Applications* (30 tundi – rakenduste loomine erinevate vahendite abil)

Seoses koolide arvutiõpetusega levib viimasel ajal USAs kiiresti mõiste *Computational Thinking*, (eestikeelne vaste veel puudub), vt näiteks Google'i saiti *Exploring Computational Thinking*, kus on mitmeid linke ka teistele materjalidele. Selle mõiste, mille olemuse üle veel arutletakse, alla kuuluvad ka sellised arendustöö elemendid nagu modelleerimine, algoritmimine ja programmeerimine. Uues koolide arvutiõpetuse rahvusliku standardite projektis (2011): „*CSTA K-12 Computer Science Standards*“ on mõiste *Computational Thinking* kesksel kohal.

Üheks tähelepanuväärseks nähtuseks seoses programmeerimise ja arendustööga on see, et viimastel aastatel on tekkinud kiiresti arenevad ja levivad uue põlvkonna spetsiaalsed programmeerimise õpetamise keeled: Scratch, Alice, AgentSheets, Greenfoot jm, renoveeritakse vanu programmeerimiskeeli nagu Logo, Basic (StarLogo, MS Small Basic) jms. Tegemist on suhteliselt lihtsa süntaksiga keeltega ning mugava kasutajaliidesega keskkondadega, mis on eeskätt mõeldud programmeerimise õpetamiseks algajatele, sh lastele ja noortele. Taolistes süsteemides on osutatud suurt tähelepanu atraktiivsusele ja multimeedia elementide kasutamisele, võimalusele lihtsalt ja kiirelt luua mängu, animatsioone. See võimaldab omandada programmeerimise olemuse, põhimõisted ja meetodid märgatavalt kiiremini kui traditsiooniliste keelte abil.

Ülaltoodud asjaoludega on arvestatud ka valikkursuses „Rakenduste loomise ja programmeerimise alused“.

Kursuse maht ja õppe eesmärgid

Kursuse mahuks on käesoleval ajal 35 tundi. Olgu märgitud, et see maht ei ole piisav arvestades arengutendentse ja võrreldes sellega, millist suurt tähelepanu osutatakse antud probleemide teiste riikide koolisüsteemides. Valikkursusega taotletakse, et õpilane:

- arendab loovust, loogilist, analüütilist ja algoritmilist mõtlemist ning süsteemset käsitusviisi probleeme ja ülesandeid lahendades
- teadvustab ja tunnetab programmjuhtimisega süsteemide tööpõhimõtet ning informatsiooni esitamise ja töötlemise põhiprotsesside olemust
- tunneb rakenduste ja programmide loomise vahendeid ning põhimeetodeid
- omandab programmide ja algoritmide koostamise ning probleemide lahendamise baasoskused programmjuhtimisega süsteemide abil

- tutvub objektorienteeritud modelleerimise, analüüsi ja disaini põhimõtetega
- saab aru objektide ja andmete olemusest, nende omadustest ning nendega täidetavatest tegevustest algoritmides ja programmides
- omandab algoritmimise ja programmeerimise põhikontseptsioonid ja mõisted ning oskused kavandada, koostada, siluda ja testida programme, mis koosnevad mitmest koostoimivast üksusest (protseduurist)
- kasutab objekte, skalaarandmeid ja massiive ning kirjeldab eri liiki protsesse

Kursusel omandatavad pädevused

Valikkursuse „Rakenduste loomise ja programmeerimise alused“ läbinud õpilane omandab järgmised pädevused ja õpitulemused:

- eristab ja oskab kirjeldada rakenduste loomise põhifaase: ülesande püstitus, analüüs, disain, realisatsioon
- teab ning oskab võrrelda ja hinnata rakenduste loomise erinevaid vahendeid ning meetodeid
- nimetab ja kirjeldab objektorienteeritud modelleerimise, analüüsi ja disaini põhimõisteid ning saab aru unifitseeritud modelleerimiskeeles (UML) esitatud klassi- ja tegevusdiagrammidest
- teab programmide loomise, töötlemise, täitmise, silumise ning testimise põhimõtteid
- eristab ning oskab programmides ja algoritmides kasutada eri liiki andmeid (arve, tekste, tõeväärtusi, graafika- ja heliandmeid) ning on omandanud ettekujutuse nende esitusviisidest arvutis; oskab neid kasutada Scratchi projektides ning VBA või Pythoni protseduurides
- teab ja mõistab konstantide, muutujate, massiivide ja objektide olemust ning kasutab neid otstarbekalt algoritmides ja programmides
- eristab ning oskab kasutada eri liiki operatsioone, avaldisi ja funktsioone väärtuste leidmiseks
- teab ja tunnetab omistamise operatsiooni olemust ning oskab seda kasutada algoritmides ja Scratchi ning VBA või Pythoni programmides
- teab ja oskab kasutada vahendeid andmete lugemiseks väliskeskonnast ning kirjutamiseks väliskeskonda
- teab graafikaandmete peamisi vorminguid, oskab määrata tegevusi graafikaobjektidega ning tunneb ja oskab kasutada vahendeid ja meetodeid graafiliste kujundite joonestamiseks
- tunneb ning oskab algoritmides ja programmides kasutada protsesside juhtimise vahendeid ning kirjeldada eri liiki protsesse: järjestikused protsessid (jada), tsüklilised protsessid (kordused), hargnevad protsessid (valikud) ja paralleelprotsessid
- oskab programmides luua ja kasutada loendeid ja massiive ning kirjeldada tüüpalgoritme nendega: summade ja keskmiste arvutamine, ekstreemumite leidmine, otsimine ja sorteerimine;
- oskab korraldada programmide jaotamist üksusteks (protseduurideks, funktsioonideks või skriptideks) ning korraldada nende vahel koostööd ja andmevahetust.

Seos karjäärivalikutega

Kursus on kõige otsesemalt seotud karjäärivalikuga, tutvustades informaatika ja infotehnoloogia erialasid. **Programmjuhtimisega süsteemid** (PJS) leiavad ühe laiemat kasutamist kõikides eluvaldkondades. Taoliste süsteemide valik ja nende võimalused kasvavad kiiresti. Nimetame ainult mõnda: arvutid ja arvutivõrgud, robotid, nutitelefonid, programmjuhtimisega seadmed ja tööpingid, automaatjuhtimisega tehnoloogilised liinid ja terved ettevõtted, sisseehitatud protsessoritega seadmed, alates autodest ja lõpetades külmkappide ja pesumasinatega. On välja kujunenud eraldi majandusharu, mida nimetatakse sageli info- ja kommunikatsioonitehnoloogia (IKT) sektoriks, mille asutused tegelevad seadmete ja tarkvara tootmisega, pakuvad erinevaid teenuseid jm. Tegemist on kõige kiiremini areneva majandusharuga ja siin leiab järjest rohkem rakendamist erineva eriala ja ametiga IT-spetsialiste.

Teiseks ja mitte väiksemaks IT-spetsialistide rakendusvaldkonnaks on infotehnoloogia vahendeid kasutavad ettevõtted ja seda teevad tänapäeval praktiliselt kõik: pangad, äri-, tootmis- ja transpordiettevõtted, meditsiini- ja haridusasutused jm. Enamiku tänapäeva firmade üheks tähtsaks osaks on arvutivõrkudele ja andmebaasidele tuginevad **infosüsteemid**, ilma milleta pole firma töö praktiliselt peaaegu mõeldav. Asutuste infosüsteemid on tihedalt seotud riiklike ja ülemaailmsete infosüsteemide ja võrgustikega. Asutuste infosüsteemide loomiseks, arendamiseks ja hooldamiseks läheb vaja samuti suurel hulgal IT-spetsialiste.

Eestis pööratakse suurt tähelepanu IKT-vahendite arendamisele ja kasutamisele. Mitmed saavutused, näiteks e-teenuste valdkonnas, on laialt tuntud kogu maailmas. Vabariigi juhtkond, IKT eksperdid ja visionäärid soovivad laiendada ja kiirendada IT-alast tegevust. Näiteks 2013 aastal toimunud konverentsil „IT kui Eesti majandusmootor 2020“ rõhutas Eesti Infotehnoloogia ja Telekommunikatsiooni Liidu (ITL) president Enn Saar, et aastaks 2020 peaks Eesti IKT sektor andma vähemalt 10%-15% meie SKT-st (praegu umbes 5%). Selleks on vaja lähiaastatel IKT oskustega spetsialistide arvu vähemalt kahekordistada. Märgatavalt suurendatakse vastuvõttu IT erialadele ülikoolides, kolledžites ja kutseharidussüsteemis. Koolide õppekavadesse lülitakse IT-ga seotud aineid.

Väga üldiselt ja tinglikult võib IT-spetsialistide ettevalmistamisel eristada kahte omavahel tihedalt seotud ja põimunud valdkonda:

- riistvara
- tarkvara.

Riistvaraspetsialistide näol on tegemist arvutiinseneridega, elektroonikutega, mehhatroonikutega jm, kes tegelevad eeskätt arvutite, arvutivõrkude, sidevahendite (sh ka telefonide), robotite, automaatjuhtimisega seadmete riistvara loomisega ja/või hooldamisega. Samal ajal peavad nad suuremal või väiksemal määral tundma, kasutama ja arendama ka tarkvara, eriti nn süsteemitarkvara: seadmete tööd juhivad ja korraldavad programmid (näit. operatsioonisüsteemid).

Tarkvaraspetsialiste nimetakse meil viimasel ajal sageli informaatikuteks aga näiteks ka tarkvarainsenerideks. Tegemist on laia profiiliga tarkvaraspetsialistidega, kes võivad tegeleda tarkvara, infosüsteemide ja rakenduse loomise ja arendamisega ning infosüsteemide haldamise ja hooldamisega väga erinevates valdkondades: üldotstarbeline ja spetsialiseeritud tarkvara, e-teenuste ja e-äri süsteemid, veebirakendused, tööprotsesside juhtimine jm. Informaatikuid õpetatakse kõigis kolmes peamises Eesti ülikoolis: TÜ, TTÜ ja TLÜ.

IKT-spetsialistid võivad töötada väga erinevatel ametitel: alates programmeerijast ja insenerist kuni IT-firma juhatajani. Vahepeale mahuvad taolised ametid nagu tarkvara ja/või infosüsteemi arendaja, juhtiv insener, konstruktor, disainer (projekteerija), süsteemianalüütik, IT juht (infojuht), andmebaasiadministraator, infosüsteemi või arvutivõrgu administraator, projektijuht jm. Ülevaate IKT-spetsialistide ametitest võib saada Euroopa informaatika professionaalide sertifikaatide kirjeldustest – [EUCIP](#).

Töö iseloomu järgi võib eristada kahte liiki tegevusi:

- süsteemide loomine ja arendamine,
- süsteemide hooldamine ja administreerimine.

Esimest liiki tegevused on iseloomulikumat IKT-sektori ettevõtetes töötavate spetsialistide jaoks, kuid ei ole välistatud ka teistes valdkondades tegutsevate IT-spetsialistide puhul. Mitmetes firmades (näiteks pangad) on IT-osakonnad, kes tegelevad nii süsteemide administreerimisega kui ka arendustööga.

IKT valdkonnas on oluline koht loomulikult teadusuuringutel. Tegemist on mitmete uudsete ja kiiresti arenevate valdkondadega: nanotehnoloogia, kvantarvutid ja kvantrobotid, tehisintellekt ja palju muud.

Kuigi IT-spetsialistid tegelevad erinevates valdkondades (riistvara ja tarkvara), on nende ettevalmistuses mõndagi ühist ja on ka mitmeid ühiseid õppeaineid. Erialavahetus või spetsialiseerumine teisele valdkonnale ei ole haruldane.

Üheks ühiseks osaks praktiliselt kõikide erialade jaoks on **programmeerimine**. Kursuse maht ja kasutatavad vahendid (programmeerimiskeeled) võivad olla üsna erinevad, kuid programmeerimine on üheks esmaseks asjaks, mille alused peab kohe alguses omandama iga tulevane IT-spetsialist. See on vajalik programmijuhtimisega seadmete (eeskätt arvuti) töö põhimõtete mõistmiseks ja teadvustamiseks ning mitmete teiste õppeainete (algoritmimine, modelleerimine jm) paremaks omandamiseks.

Programmeerimine ei seisne ainult programmide kirjutamises. Sellega on seotud väga erinevad tegevused: algoritmimine, andmete, protsesside ja süsteemide modelleerimine, süsteemianalüüs, infosüsteemide, rakenduste ja dokumentide projekteerimine ja disain, kasutajaliideste loomine, testimine jm. IT-spetsialist peab oskama kasutada vähemalt 4-5 erinevat programmeerimiskeelt, töötada mitme spetsiaalse arendusvahendiga ja modelleerimissüsteemiga.

Seoses infotehnoloogia vahendite laialdase kasutamisega on arendustöö ning modelleerimise ja programmeerimise aluste tundmine vajalik (või vähemalt kasulik) ka teiste erialade spetsialistidele. Tänapäeval peavad paljud mitteinformaatika erialade spetsialistid mitte ainult oskama kasutada IKT vahendeid vaid suutma püstitada ülesandeid oma valdkonna probleemide lahendamiseks, määratleda andmeid ja tegevusi nendega, luua algoritme, vajadusel osaleda infotehnoloogiaalastes projektides.

Õppe alustamise ning kursuse läbimise tingimused

Valikkursuse „Rakenduste loomise ja programmeerimise alused“ alustamiseks spetsiaalsed eelteadmised ei nõuta. Kuid oleks väga kasulik, kui oleks läbitud sisukas üldinformaatikakursus, mis arvestaks ECDL nõudeid ja soovitusi digitaalse kirjaoskuse osas.

Kursuse läbimise tingimused määrab kool, arvestades aine struktuuri, osade sisu ja mahtu

Praktiliste tegevuste osakaal kursusel

Põhirõhk kursuses on just praktilisel tööl. Võiks isegi öelda, et tegemist on praktikumidega juhitava probleemorienteeritud kursusega.

Valikuvõimalused kursuse läbimiseks

Valikkursus „Rakenduste loomise ja programmeerimise alused“ koosneb üldjuhul kahest osast: põhimoodulist ja lisamoodulist.

Põhimooduli programmeerimise osa on orienteeritud programmeerimissüsteemi Scratch kasutamisele.

Lisamoodulis võib kooli valikul praegu olla kasutusel kas arendussüsteem *Visual Basic for Application* (VBA) või programmeerimiskeel Python. Käesoleval ajal viivad erinevaid programmeerimise kursuseid läbi ka mitmed koolitusettevõtted. Kokkuleppel kooliga (õpetajaga) võiks mõni taoline kursus minna arvesse ka lisamoodulina.

Võimalus on läbida valikkursus või osa sellest e-õppena kasutades e-õppekeskkonda Moodle.

Valikkursuse ainekava, vajalikud eelteadmised, seos teiste ainetega

Ainekava

Õppe- ja kasvatuseesmärgid

Valikkursusega taotletakse, et õpilane:

- arendab loovust, loogilist, analüütilist ja algoritmilist mõtlemist ning süsteemset käsitlusviisi probleeme ja ülesandeid lahendades;
- teadvustab ja tunnetab programmjuhtimisega süsteemide tööpõhimõtet ning informatsiooni esitamise ja töötlemise põhiprotsesside olemust;
- tunneb rakenduste ja programmide loomise vahendeid ning põhimeetodeid;
- omandab programmide ja algoritmide koostamise ning probleemide lahendamise
- baasoskused programmjuhtimisega süsteemide abil;
- tutvub objektorienteeritud modelleerimise, analüüsi ja disaini põhimõtetega;
- saab aru objektide ja andmete olemusest, nende omadustest ning nendega täidetavatest
- tegevustest algoritmides ja programmides;
- omandab algoritmimise ja programmeerimise põhikontseptsioonid ja mõisted ning oskused kavandada, koostada, siluda ja testida programme, mis koosnevad mitmest koostoimivast üksusest (protseduurist); kasutab objekte, skalaarandmeid ja massiive ning kirjeldab eri liiki protsesse.

Kursuse lühikirjeldus

Kursuse põhiosad:

- rakenduste loomise põhimõtted
- mudelid ja modelleerimise alused
- algoritmimise ja programmeerimise alused.

Neid teemasid ei läbita järjestikku, vaid sobiva käsitluse valib õpetaja, arvestades kasutatavat programmeerimisvahendit, kursuse suunitlust, oma metoodilisi kogemusi ja eelistusi jmt.

Rakenduste loomise põhimõtteid ja põhifaase tutvustatakse praktilise töö kaudu, koostades mõned projektid, mis sisaldavad modelleerimise, analüüsi ning disaini elemente. Modelleerimises on peamine koostada ja esitada algoritme unifitseeritud modelleerimiskeele UML abil. Klassimudelite olemust võib tutvustada võrdlemisi lühidalt ja üldiselt, arvestades, et õppija oskaks lugeda nende abil esitatud süsteemide ja andmete struktuure ning saada neist aru. Põhiosa ajast kulub programmeerimise ning algoritmimise aluste omandamisele praktilise töö kaudu.

Kursuse eesmärkide saavutamiseks on tähtis teadvustada ning tunnetada programmide ja programmjuhtimise olemust, koostades praktilises töös programme ning realiseerides neid arvutil. Et seda protsessi lihtsustada ja kiirendada ning mitte tekitada õpilastes vastumeelsust aine vastu, peaks esimeseks või ainsaks keeleks olema võimalikult lihtsa süntaksiga, atraktiivne ning multimeedia vahendite kasutamist võimaldav programmeerimiskeel või -keskkond. Viimasel ajal on kiiresti arenenud ning levinud spetsiaalsed nn programmeerimise õppimise keeled ja keskkonnad (Scratch, Alice, Game Maker, uue põlvkonna Basic- ning Logo-põhised keeled jm). Need on mõeldud algajatele, eeskätt lastele ja noortele programmeerimise õppimiseks ning õpetamiseks. Võrreldes traditsiooniliste vahenditega (nt Pascal) võimaldavad need õppijal omandada programmeerimise ning algoritmimise põhitõed ja -oskused 2–3 korda kiiremini.

Praeguse seisuga võiks esimeseks keskkonnaks (keeleks) olla kiiresti leviv Massachusettsi Tehnoloogiainstituudis (MIT) loodud ja arendatav Scratch (<http://scratch.mit.edu>). Tegemist on vabavaraga, mille kohta on internetis üsna palju õppematerjale, sh eesti ja vene keeles. Arvestades kursuse eesmärke ja mahtu, võiks lisaks Scratchile lühidalt tutvustada mõnda sellist vahendit nagu Alice, MS Small Basic, rakenduste arendussüsteem Visual Basic (VBA), Python vm. Valiku võiks teha kool.

Kursus peaks põhiosas olema orienteeritud praktilisele tegevusele ja e-õppe materjalide kasutamisele.

Õpitulemused

Gümnaasiumi õpitulemused kajastavad õpilase rahuldavat saavutust. Kursuse lõpul õpilane:

- eristab ja oskab kirjeldada rakenduste loomise põhifaase: ülesande püstitus, analüüs, disain, realisatsioon
- teab ning oskab võrrelda ja hinnata rakenduste loomise erinevaid vahendeid ning meetodeid
- nimetab ja kirjeldab objektorienteeritud modelleerimise, analüüsi ja disaini põhimõisteid ning saab aru unifitseeritud modelleerimiskeeles (UML) esitatud klassi- ja tegevusdiagrammidest
- teab programmide loomise, töötlemise, täitmise, silumise ning testimise põhimõtteid
- eristab ning oskab programmides ja algoritmides kasutada eri liiki andmeid (arve, tekste, tõeväärtusi, graafika- ja heliandmeid) ning on omandanud ettekujutuse nende esitusviisidest arvutis
- teab ja mõistab konstantide, muutujate, massiivide ja objektide olemust ning kasutab neid otstarbekalt algoritmides ja programmides
- eristab ning oskab kasutada eri liiki operatsioone, avaldisi ja funktsioone väärtuste leidmiseks
- teab ja tunnetab omistamise operatsiooni olemust ning oskab seda kasutada algoritmides ja programmides

- teab ja oskab kasutada vahendeid andmete lugemiseks väliskeskkonnast ning kirjutamiseks väliskeskkonda
- teab graafikaandmete peamisi vorminguid, oskab määrata tegevusi graafikaobjektidega ning tunneb ja oskab kasutada vahendeid ja meetodeid graafiliste kujundite joonestamiseks
- tunneb ning oskab algoritmides ja programmides kasutada protsesside juhtimise vahendeid ning kirjeldada eri liiki protsesse: järjestikused protsessid (jada), tsüklilised protsessid (kordused), hargnevad protsessid (valikud) ja paralleelprotsessid
- oskab programmides luua ja kasutada massiive ning kirjeldada tüüpiliste algoritmidega: summade ja keskmiste arvutamine, ekstreemumite leidmine, otsimine ja sorteerimine;
- oskab korraldada programmide jaotamist üksusteks (protseduurideks, funktsioonideks või skriptideks) ning korraldada nende vahel koostööd ja andmevahetust.

Õppesisu

Rakenduste loomise põhimõtted. Rakenduse olemus. Rakenduste loomise meetodid ja vahendid. Üldotstarbelised programmeerimissüsteemid ja -keeled. Rakendusprogrammid ja nende arendusvahendid. Võrgurakenduse loomise vahendid. Rakenduste loomise põhifaasid: ülesande püstitus, analüüs, disain, realisatsioon.

Mudelid ja modelleerimise alused. Mudeli olemus. Mudelite liigid. Geomeetrilised, matemaatilised ja füüsikalised mudelid. Simulatsioonimudelid. Objektorienteeritud modelleerimine ja modelleerimiskeel UML. Objektid ja klassid. Objektide omadused ja tegevused. Sündmused. Seosed objektide ja klasside vahel, klassidiagrammid. Tegevused ja tegevusdiagrammid. Tegevusdiagrammide kasutamine algoritmide esitamiseks.

Algoritmimise ja programmeerimise alused. Programmi olemus. Programmide loomise, töötlemise ja täitmise põhimõtted. Programmeerimiskeeled ja -süsteemid. Laused ning programmiüksused: protseduurid, funktsioonid ja skriptid. Translaatorid: interpretaatorid ja kompilaatorid. Algoritmi olemus. Algoritmide esitusviisid: tegevusskeemid, algoritmikeeled, programmeerimiskeeled jm.

Objektide ja andmete käsitlemine programmides. Objektide omaduste, meetodite ja sündmuste kasutamine. Andmete liigid ja organisatsioon. Konstandid ja muutujad. Massiivid. Operatsioonid andmetega. Avaldised ja funktsioonid. Omistamine. Andmete lugemine väliskeskkonnast ja kirjutamine väliskeskkonda.

Graafikaandmete kasutamine programmides. Graafikaobjektide importimine. Graafiliste kujundite loomine (joonistamine) programmi poolt. Graafikaobjektide põhiomadused ja nende muutmine. Tüüpilised tegevused graafikaobjektidega. Animatsioon.

Protsesside liigid algoritmides ja programmides: järjestikused protsessid (jada), tsüklilised protsessid (kordus), hargnevad protsessid (valik), paralleelprotsessid. Eri liiki protsesside kirjeldamise ja täitmise põhimõtted. Korduste liigid: lõpmatu kordus, etteantud kordamisarvuga kordus, eel- ja järelkontrolliga iteratiivsed kordused. Valik ühest, kahendvalik ja mitmene valik.

Massiivid. Massiivide põhiomadused: nimi, dimensioon, mõõtmed, dünaamilisus. Massiivide määramine ja loomine. Viitamine massiivide elementidele. Tüüpilised algoritmid massiividega: summade ja keskmiste arvutamine, ekstreemumite leidmine, otsimine, sorteerimine jmt. Koostöö ja andmevahetus programmiüksuste vahel. Pöördumised ja teadete saatmine. Globaalsed ja lokaalsed andmed.

Kasutajaliidese loomine. Programmide silumine ja testimine. Programmide vormistamine.

Õppetegevus

- Õppetegevus toimub arvutiklassis praktikumide ja seminaride vormis.
- Õppetööd peab toetama elektroonsete õppematerjalide komplekt: teooria põhiosad, harjutused, näited, lingid, elektroonsed töövihikud jm.
- Õpilased teevad iseseisvalt (kas klassis või kodus) 3–4 tööd, mille alusel määratakse hinne.

Vajalikud eelteadmised

Valikkursuse „Rakenduste loomise ja programmeerimise alused“ alustamiseks spetsiaalsed eelteadmised ei nõuta. Kuid oleks väga kasulik, kui oleks läbitud sisukas Informaatika kursus, mis arvestaks ECDL nõudeid ja soovitusi digitaalse kirjaoskuse osas.

Seos teiste ainetega

Kõige tihedamini on seotud ainega „Robootika“, kus otseselt kasutatakse programmeerimist ja muid arendusvahendeid.

Aine on teatud määral seotud põhikooli informaatikaalaste ainetega. Kahjuks on seos üsna hägune ja nõrk, kuna nende sisu on määratletud üsna üldiselt ja on suurel määral orienteeritud Interneti kasutamisele.

Sama võiks öelda ka seoste kohta teiste ainetega, eriti reaalainetega, millega võiks seos olla tunduvalt tihedam. Neis on IKT vahendite rakendamisest räägitud väga vähe ja formaalselt. Kusjuures peamiselt on nimetatud kasutamist: näiteks arvuti simulatsioonid, e-õpe vahendite kasutamine jms.

Kursuse õppekomplekti lühitutvustus õpilastele, koolidele ja avalikkusele

Kursuse tutvustus eesti keeles

Valikkursuse „Rakenduste loomise ja programmeerimise alused“ eesmärkide ja sisu kavandamisel on arvestatud maailmas viimasel ajal toimuvate protsesside ja arengutega informaatika ja arvutiteaduse õpetamisel. Maailma ja eriti infotehnoloogia juhtriigi USA koolide õppekavades pannakse viimasel ajal suurt rõhku arendustöö (modelleerimine, analüüs, disain, algoritmimine, programmeerimine) aluste õpetamisele.

Valikkursuses arvestatakse gümnaasiumite ainekavas pakutud kolme osaga: rakenduste loomise meetodid ja vahendid, modelleerimise ja algoritmimise põhimõtted, programmeerimise alused.

Rakenduste loomise meetodeid, vahendeid ja põhifaase (ülesande püstitus, analüüs, disain ja programmeerimine) käsitletakse ülesannete lahendamisel läbi terve kursuse. Läbivalt käsitletakse ka modelleerimist ja algoritmimist. **Modelleerimises** tutvustatakse objektorienteeritud lähenemisviisi ja unifitseeritud modelleerimiskeelt UML. Põhjalikumalt vaadeldakse **protsesside** ja **algoritmide** esitamist **tegevusdiagrammide** ja **algoritmikeele** (pseudokoodi) abil.

Programmeerimise osa sisu jaguneb **põhi-** ja **lisamooduliks**.

Põhimoodulis on programmeerimiskeeleks uue põlvkonna graafiline keel Scratch, mis on vabavara ja saab tasuta laadida alla saidilt <http://scratch.mit.edu>. Sealt võib leida ka suurel hulgal näiteid ja demosid ja mitmesuguseid õppematerjale. Programmeerimiskeel on loodud Massachusettsi Tehnoloogiainstituudis (MIT) spetsiaalselt programmeerimise õpetamiseks algajatele ning leiab laialdast kasutamist koolides (eriti USAs) ja ka mitmetes ülikoolides (Harvard, Berkeley, TTÜ jt). Scratchi keskkonda saab kasutada paljudes keeltes, sh ka eesti ja vene keeles. Tegemist on lihtsa ja atraktiivse, multimeedia elementide kasutamisele orienteeritud, keele ja keskkonnaga, mis toetab olulisemaid programmeerimise kontseptsioone.

Valikkursuse „Rakenduste loomise ja programmeerimise alused“ **lisamoodulid** kujutavad endast skaleeritavaid ja arenevaid e-õppekursuseid rakenduste loomiseks erinevate programmeerimiskeelte abil. Programmeerimiskeeli võiks tulevikus olla 3–4. Esialgse visioonina tuleks kõne alla **Python** (vabavara), mõni **Basicu versioon**: Microsoft Small Basic (vabavara), Visual Basic Express (tasuta haridusasutustele), **Visual Basic for Application** (VBA). Viimane kuulub paljudes koolides kasutatava MS Office'i juurde, pakkudes lihtsaid vahendeid õppimiseks ja dokumendipõhiste rakenduste loomiseks.

Praegu on kasutusel **kaks lisamooduli** varianti: **Pythoni-** ja **VBA-põhine**. Lisamooduli keele ja moodulite mahud (kokku koos põhimooduliga 35 tundi) valib kool sõltuvalt õppesuunast – reaal-, majandus-, humanitaar- jm. Lisamoodul võib ka puududa. Spetsialiseerumist saab määrata ka ülesannete valikuga.

Kursuse tutvustus vene keeles

Краткое описание курса

Этот курс является курсом по выбору. При разработке (планировании) целей и содержания курса „Rakenduste loomise ja programmeerimise alused“ («Создание приложений и основы программирования») учитывались процессы и направления развития, происходящие в мире в последнее время в обучении информатике. В школьных программах ведущих государств мира и особенно в США особое внимание уделяется обучению основам разработки приложений (моделирование, анализ, дизайн, алгоритмизация, программирование).

В курсе по выбору учитываются три части предлагаемой программы для гимназий: методы и средства разработки приложений, основы моделирования и алгоритмизации, основы программирования.

Основные этапы, методы и средства создания **приложений** (постановка задачи, анализ, дизайн и программирование) рассматриваются при решении заданий на протяжении всего курса. Аналогично рассматриваются моделирование и алгоритмизация. Представляется объектно-ориентированный подход моделирования и унифицированный язык моделирования UML. Подробнее рассматривается представление **процессов** и **алгоритмов** с помощью диаграмм действий и алгоритмического языка (псевдокода).

Программирование представлено двумя модулями: **основным** и **дополнительным**.

В основном модуле в качестве языка программирования выбран графический язык нового поколения Scratch, который можно бесплатно загрузить с сайта <http://scratch.mit.edu>. На этом сайте можно также найти большое количество примеров и учебных материалов. Язык программирования Scratch создан в Медия лаборатории Массачусетского технологического института (Massachusetts Institute of Technology - MIT). Этот язык предназначен специально для обучения программированию начинающих и находит широкое применение в школах (особенно в США), а также и в университетах (Гарвард, Беркли, ТТУ и т.д.). Scratch можно использовать на многих языках, в том числе на эстонском и русском языках.

Scratch - это новая графическая среда программирования, ориентированная на использование мультимедийных элементов и поддерживающая основные концепции программирования.

В предмете по выбору „Rakenduste loomise ja programmeerimise alused“ **дополнительные модули** представляют собой развивающиеся э-курсы о различных языках программирования. В будущем рассматриваемых языков программирования может быть 3 или 4. В первоначальной концепции речь может идти о языке Python (бесплатное программное обеспечение), некоторые версии Basic: Microsoft Small Basic (бесплатное программное обеспечение), Visual Basic Express (бесплатно для образовательных учреждений), Visual Basic for Application (VBA). Последний входит в состав MS Office, используемый во многих школах, и предлагает простые средства для обучения и создания приложений.

Используются два варианта дополнительного модуля: на основе языка **Python** и **VBA**. Язык дополнительного модуля и его объем выбирает школа в зависимости от специализации обучения: реальное, экономическое, гуманитарное и т.п. Дополнительный модуль может и отсутствовать. Специализация может задаваться также выбором задач.

Kursuse tutvustus inglise keeles

Course Overview

Course "Creation of Applications and the Basics of Programming" objectives and the content planning processes developments is taken into account in informatics and computer science teaching. Head of the IT world, and especially in the U.S. school curriculum recently placed great emphasis on development (modeling, analysis, design, algorithms, programming), teaching bases.

In Course considered high school program offered by the three components: applications, methods and tools for creating, modeling, and algorithmic principles of programming bases.

Applications creation of methods, tools and phases (task set, analysis, design and programming) on solving problems through the entire course. Throughout the modeling and treatment shall be algorithmic. Modeling introduces object-oriented approach and the unified modeling language UML. The study focuses on the processes and algorithms for the submission of activity diagrams and using algorithmic language (pseudo code).

Programming content is divided into **primary** and **secondary modules**.

In basic module the programming language is the new generation of graphical programming language Scratch, which is freeware and available for free download from the site <http://scratch.mit.edu>. There are also a large number of examples and demos, and other materials. A programming language itself is created by the Massachusetts Institute of Technology (MIT) program specifically to teach beginners, and is widely used in schools (especially in the U.S.) and in a number of universities (Harvard, Berkeley, TUT, etc.). Scratch environment can be used in many languages, including Estonian and Russian. This is a simple and attractive, with multimedia-oriented elements language and environment that supports the key concepts of computer programming.

Course "Creation of Applications and the Basics of Programming" modules represent scalable and evolving e-learning courses in different programming language. Programming languages in the future could be 3–4. The initial vision should be **Python** (freeware), a **Basic version**: Microsoft Small Basic (freeware), Visual Basic Express (free for educational institutions), **Visual Basic for Application** (VBA). It is used in many schools with MS Office, providing a simple means to study and document-based applications.

There are two versions of secondary modules: Python-based, and VBA-based. A school can choose a language and duration depending on the direction of learning (real, economic, humanitarian and other). A secondary module is not mandatory. Specialization can also set the range of tasks.