

18. Valikkursus „Rakenduste loomise ja programmeerimise alused“

18.1. Õppe- ja kasvatuseesmärgid

Valikkursusega taotletakse, et õpilane:

- 1) arendab loovust, loogilist, analüütilist ja algoritmilist mõtlemist ning süsteemset käsitusviisi probleeme ja ülesandeid lahendades;
- 2) teadvustab ja tunnetab programmjuhtimisega süsteemide tööpõhimõtet ning informatsiooni esitamise ja töötlemise põhiprotsesside olemust;
- 3) tunneb rakenduste ja programmide loomise vahendeid ning põhimeetodeid;
- 4) omandab programmide ja algoritmide koostamise ning probleemide lahendamise baasoskused programmjuhtimisega süsteemide abil;
- 5) tutvub objektorienteeritud modelleerimise, analüüsi ja disaini põhimõtetega;
- 6) saab aru objektide ja andmete olemusest, nende omadustest ning nendega täidetavatest tegevustest algoritmides ja programmides;
- 7) 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.

18.2. Kursuse lühikirjeldus

Kursuse põhiosad:

- 1) rakenduste loomise põhimõtted;
- 2) mudelid ja modelleerimise alused;
- 3) algoritmimise ja programmeerimise alused.

Osi ega nende 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 realiseeritavate programmide jaoks. 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 Tehnoloogia Instituudis (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 eesmäärke 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 põhisas olema orienteeritud praktilisele tegevusele ja e-õppe materjalide kasutamisele.

18.3. Õpitulemused

Gümnaasiumi õpitulemused kajastavad õpilase rahuldavat saavutust.

Kursuse lõpul õpilane:

- 1) eristab ja oskab kirjeldada rakenduste loomise põhifaase: ülesande püstitus, analüüs, disain, realisatsioon;
- 2) teab ning oskab võrrelda ja hinnata rakenduste loomise erinevaid vahendeid ning meetodeid;
- 3) nimetab ja kirjeldab objektorienteeritud modelleerimise, analüüsi ja disaini põhimõisteid ning saab aru unifitseeritud modelleerimiskeeles (UML) esitatud klassi- ja tegevusdiagrammidest;
- 4) teab programmide loomise, töötlemise, täitmise, silumise ning testimise põhimõtteid;
- 5) eristab ning oskab programmides ja algoritmides kasutada eri liiki andmeid (arve, tekste, tõeväärtusi, graafika- ja heliandmeid) ning on omandanud ettekujutuse nende esitusviisidest arvutis;
- 6) teab ja mõistab konstantide, muutujate, massiivide ja objektide olemust ning kasutab neid otstarbekalt algoritmides ja programmides;
- 7) eristab ning oskab kasutada eri liiki operatsioone, avaldisi ja funktsioone väärtuste leidmiseks;
- 8) teab ja tunnetab omistamise operatsiooni olemust ning oskab seda kasutada algoritmides ja programmides;
- 9) teab ja oskab kasutada vahendeid andmete lugemiseks väliskeskkonnast ning kirjutamiseks väliskeskkonda;
- 10) teab graafikaandmete peamisi vorminguid, oskab määrata tegevusi graafikaobjektidega ning tunneb ja oskab kasutada vahendeid ja meetodeid graafiliste kujundite joonestamiseks;
- 11) 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;
- 12) oskab programmides luua ja kasutada massiive ning kirjeldada tüüpalgoritme nendega: summade ja keskmiste arvutamine, ekstreemumite leidmine, otsimine ja sorteerimine;
- 13) oskab korraldada programmide jaotamist üksusteks (protseduurideks, funktsioonideks või skriptideks) ning korraldada nende vahel koostööd ja andmevahetust.

18.4. 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. Tüüpgevused 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 määratlemine ja loomine. Viitamine massiivide elementidele. Tüüpalgoritmid 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.

18.5. Õppetegevus

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

18.6. Füüsiline õpikeskkond

Programmeerimise õpikeskkond.

Tõnis Lukas

Haridus- ja teadusminister