# Uue põlvkonna vahendid programmeerimise õppimiseks

## Lühiülevaade programmeerimise õppimise keeltest

Viimastel aastatel on tekkinud ja kiiresi arenenud ning levinud uue põlvkonna programmeerimise õppimise/õpetamise keeled ja keskkonnad. Traditsiooniliselt nimetatakse neid programmeerimiskeelteks, kuid pigem võiks taolisi vahendeid nimetada **programmeerimise keskkondadeks** või ka süsteemideks, sest neis on palju enamat, kui lihtsalt keel. Olgu märgitud, et ka programmeerimise olemus taoliste keskkondade ja keelte kasutamisel on muutumas. Rohkem on tegemist süsteemide ja rakenduste modelleerimisega ja disainiga, kui programmeerimisega traditsioonilises tähenduses. Konkureerimiseks uute vahendiega, arendatakse ja modifiseeritakse ka mitmeid olemasolevaid keeli ja süsteeme. Wikipeedias, jaotises „*Educational programming language*“ [1], on nimetatud juba ligi 60 uut ja vanat taolist keelt.

Tegemist on suhteliselt lihtsa süntaksiga keeltega ning mugavate kasutajaliidestega keskkonda­dega, mis on eeskätt mõeldud programmeerimise õppimi­seks/õpetamiseks algajatele, sh lastele ja noortele. Taolistes süsteemides on osutatud suurt tähelepanu atraktiivsusele ja multimeedia kasutamisele, võimalusele lihtsalt ja kiirelt luua mänge, animatsioone jmt. See võimaldab omandada programmeerimise olemuse, põhimõisted ja meetodid märgatavalt kiiremini sellest, kui alustada kohe mingi "suure", proffidele mõeldud, keelega (Pascal, C, PHP, Java jmt). Ka autoga sõitmise õppimist ei pea alustama Vormel I-ga, võib alustada ka kardist või vanast Fordist ning järjest liikuda võimsamate ja uuemate masinate poole.

Vanadest õpetamise keeltest peab kõigepealt, muidugi, nimetama juba 1964. aastal loodud, **Basic**’ut, mille jaoks on loodud üle 200 variandi ja dialekti: Turbo Basic, Quick Basic, Power Basic, Visual Basic, VBA, Basic‑256, Liberty Basic, SiMPLE, MS Small Basic jt. Üks haru: **Visual Basic**, on jõunud ka laialt kasutatavate tööstusliku tarkvara loomise keelte hulka. On tähelepanuväärne, et 2008. aasta lõpus lõi Microsoft keele MS Small Basic, mida levitatakse tasuta ja mis on läbinud kiire arengu. See on loodud just arvestades uusi suundi programmeerimise õpetamise keelte arengus ning nö. vastukaaluks uutele vahenditele. MS Small Basic’us on küll üsna piiratud lausete hulk, keel ise on aga lihtne ja mugava kasutajaliidesega, ning seda võib lugeda objektipõhiseks. Olulisel kohal on graafika, heli ja Interneti kasutamise võimalused. Olgu märgitud, et enamikus Basic’u versioonides on multimeedia vahendite (sh. ka graafikaobjektide) kasutamine võrdlemisi kohmakas. Kuna keele variandid on paisunud üsna suurteks, on ka nende kasutajaliidesed ja kasutamisjuhendid võrdlemisi mahukad ja keerulised.

Vanadest tegijatest peab nimetama ka **Logo**’t (1967), mille jaoks on aegade jooksul loodud samuti üsna palju variante. Viimased dialektid (Lego Logo, StartLogo, NetLogo, Etoys jm) võib juba arvata ka uue põlvkonna õpetamise keelte hulka. Meil kasutatakse Logo’t suhteliselt vähe.

Rääkides programmeerimise õpetamise keeltes, ei saa minna mööda **Pascal**’ist, mis tänavu sai 40 aastaseks. Pikemat aega oli see, Basic’u kõrval, üheks peamiseks õpetamise keeleks, eriti infotehnoloogia erialasid õppivate või sellele orienteeritud õpilaste jaoks. Arvatavasti, inertsi tõttu kasutatakse seda õpetamisel veel üsna palju, sh. meie koolides. Meie arvates, on käesolevaks ajaks Pascal oma töö selles osas teinud ning enam ei vasta õppimise keelte nõutele oma keerulise süntaksi, vähese atraktiivsuse, suletuse jm tõttu (igatahes ei sobi ta esimeseks keeleks ja mitteinformaatikute jaoks). Meie loobusime Pascal’ist mitteinformaatikute jaoks juba pea 10 aastat tagasi, kasutades seda ainult paar kolm aastat.

Programmeerimise õpetamiseks mõeldud uutest keeltest ja keskkondadest võiks nimetada järgmisi: Alice, AgentSheets, Baltie, Greenfoot, Karel++, Phrogram ja Scratch. Siin peatume veidi pikemalt kahel tüüpilisel ja enim levinud selle grupi keelel: **Alice** ja **Scratch**. Viimase aga võtame, võib olla, edaspidi veel eraldi vaatluse alla. Mõlema juhul on tegemist atraktiivsete, multimeedia laialdasele kasutusele orienteeritud lihtsate keskkondadega ja keeltega. Programm pannakse neis kokku hiirega ja erilisi probleeme süntaksiga nende puhul ei ole. Eriti on see iseloomulik Scratch’i jaoks, mis on, võiks öelda, täielikult graafiline programmeerimiskeel.

Uue põlvkonna vahendite tekke põhjuseid ja arendamise eesmärke on analüüsitud näiteks Scratch’i arendusmeeskonna (MIT, Media Lab) publikatsioonides [2]. Üheks peamiseks ajendiks oli neile see, et märgatavalt hakkas vähenema noorte huvi programmeerimise vastu. Traditsioonilised programmeerimisvahendid on vähe atraktiivsed ja laiemale kasutajate ringile üldiselt rasked (keeruline süntaks, andmetüübid jm), samal ajal Internet pakub noortele hulgaliselt huvitavat tegevust.

Ka meie oleme pannud tähele, et õppima asujate seas on oluliselt langenud nende arv, kes on koolis puutunud kokku programmeerimisega, sh ka informaatika erialadele astujate sees. Kui veel mõned aastat tagasi olid varem programmeerimisega tegelenud 70‑80% informaatika erialasid õppima asujatest, siis praegu on see protsent 20‑30%. Väheneb ka infotehnoloogia erialasid õppima asujate arv.

Uued vahendid ja keeled aga sobivad igas vanuses algajatele programmeerimise õpetamiseks. Neid kasutatakse koolide algklassidest kuni kolledžide ja ülikoolide prog­ram­­­mee­ri­mise sissejuhatavate kursusteni välja, vt näieks [3] ja [4]. Käsitletavad teemad ja lahen­data­vad ülesanded saab valida, arvestades õppijate vanust, taset, õpetamise eesmärke jm, Kusjuures valiku võimalusi ja variatsioone on siin tunduvalt rohkem, kui traditsiooniliste keelte kasutamise korral.

Kui eesmärgis on näiteks sihipärane programmeerimise õpetamine, võimaldavad uued keskkonnad kontsentreerida õppimise algfaasis põhitähelepanu programmeerimise ja arendustöö üldistele põhimõtetele: rakenduste disain, algoritmimine ja modelleerimise alused, objektorienteeritud lähenemisviisi olemus, erinevat liiki objektide ja andmete kasutamine, protsesside liigid: jada, valik, kordus, paralleelprotsessid, ja nende juhtimine, sündmused jmt. Ülesannete valik võib olla alatest lihtsatest animatsioonidest ja mängudest kuni klassikaliste programmeerimise algoritmideni: otsimine, sorteerimine, operatsioonid massiividega jm. Kusjuures taoliste algoritmide tööpõhimõtteid saab ka õppija iase suhteliselt lihtsalt visualiseerida. Programmeerimise õpetamine traditsiooniliste vahendite ja meetoditega kipub sageli taanduma kasutatava programmeerimiskeele süntaksi õpetamisele ja omandamisele.

## Alice’st

**Alice**'i on loodud Virginia Ülikoolis juba 10 aastat tagasi, praegu arendatakse seda Garnegie Mellon Ülikoolis (<http://www.alice.org>). Laiema leviku on ta saanud viimase 3‑5 aasta jooksul loodud versioonidega.

Alice keskkonnas saab luua mänge, animatsioone ja ka muid rakendusi, kasutades kolmemõõtmelisi graafikaobjekte. Süsteemiga on kaasas suur valik graafikaobjektide klasse: inimesed, loomad, hooned, autod jne, mille alusel saab luua konkreetseid objekte. Viimases versioonis (3.0, momendil veel beta) saab kasutada ka tuntud süsteemi SIMS objekte (klasse). Objektid saab programmi toimel panna tegutsema ekraani piirkonnas (aknas), mida nimetatakse (virtuaalseks) **maailmaks** (*world*). Selle jaoks saab valida erinevaid taustu. Rakenduse loomisel ajal võetakse objektid hoidlast ja paigutatakse hiire abil maailma aknasse, valides nende jaoks asukoha, orientatsiooni, suuruse ja muud algomadused. Programm saab panna objektid täitma mitmesugusid tegevusi: kõndima, tantsima, lendama jm.

Joonisel 1 on maailma „Onu ja pingviinid võimlemas“ näide. Sellel on neli objekti: onu (süsteemi klass Coach), kaks pingviini (klass Penguin) ja hoone (Farmhouse).



Joonis 1. Alice maailm „Onu ja pingviinid võimlemas“

Paljude klasside o**bjektid koosnevad alamobjektidest.** Klassi Coach objekti onu struk­tuur on toodud joonisel 2. Onul on kaks jalga, mis koosnevad omakorda osadest (skeemil peidetud), ülakehast, mis koosneb samuti alamobjektides jne. Seega, klassikaline, hierarhilise struktuuriga objekt. Õppijale selgelt nähtav ja arusaadav.

  

Joonis 2. Objekti onu struktuur ja meetodite näited.

Objektide (klasside) jaoks on süsteemis olemas suhteliselt suur valik nn elementaar- ehk baasmeetodeid: teisalda (liigu), pööra, muuda suurust, ütle jm Väikene valik baasmeetodeid onu parema jala jaoks on toodud joonisel 2.

Kasutades baasmeetodeid, juhtimiskorraldusi, funktsioone jm., saab luua uusi, üldise iseloomuga, meetodeid, lohistades komponendid hiirega redaktori alasse ja valides (sisestades) sobivad argumentide väärused. Programm üldjuhul koosneb mitmest meetodist ja funktsioonist.



Joonis 3. Alice meetodi fragment

Joonisel 3 on toodud meetodi fragment. Iga täppidega algav riba (plokk) kujutab endast korraldust ehk lauset. Need moodustatakse baasmeetodite ja juhkäskude (**Loop**, **If**, **While** jm) mallides, sisestades avanevate menüüde abil vajalikud argumendid ja väärtused. Nagu teistes keeltes, on ka siin olemas **liht**‑ ja **liitlaused**. Näiteks praegu liitlause **Loop** määrab, et kahte tema sees olevat lihtlaused (baasmeetodit) täidetakse kolm korda: onu tõstab ja paneb maha jalga järjest kolm korda. Liitlause **Do together** täidab kaks käsku paralleelselt: onu toob mõlemad käed koraga ette.

Alice näol on tegemist objektipõhise aga mitte objektorienteeritud süsteemiga (keelega): saab olemasolevatest klassides luua uusi objekte, uusi klasse aga luua ei saa. Aga algajale, meie meelest, seda vaja polegi. Kasutades olemasolevaid objekte, saab õppija kiiresti selgeks objektorienteeritud süsteemide (mitte ainult programmeerimis­keelte) põhimõistete (objektid, klassid, meetodid, omadused ja sündmuste) olemuse.

## Scratch’ist

Scratch on **graafiline** programmeerimise keskkond, mis teeb lihtsaks lugude, mängude, animatsioonide ja ka tõsisemate asjade loomise ning loodu jagamise teistega veebis. Scratch on loodud Massachusetts'i Tehnoloogia Instituudi (Massachusetts Institute of Technology - MIT) Meedia Laboratooriumis. Muide, samas on aastaid tagasi loodud ka Logo. Olgu aga kohe öeldud, et Logo’ga on Scratch’il võrdlemisi vähe ühist. Scratchi loomist on rahaliselt toetanud sellised firmad nagu USA Rahvuslik Teadusfond, Intel, Microsoft, Nokia jt. Scratch'i viimane versioon: Scratch 1.4, on ilmunud juulis 2009. Alates versioonist 1.3 (august 2008) on võimalik kasutada ka eestikeelset keskkonda, juhendeid ja õppematerjale. Scratch’i kasutajaliides on olemas ca 50-es keeles.

Scratch'i saab tasuta alla laadida selle kodusaidilt: <http://scratch.mit.edu>. Saidil on mitmesuguseid õppe- ja abimaterjale [erinevates keeltes](http://info.scratch.mit.edu/Languages), sh ka eesti keeles [5] ning tohutul hulgal näiteid (praegu ca 1 700 000!), mida saab vabalt alla laadida, uurida ja remiksida. Igaüks võib ühe hiireklõpsuga laadida oma toote (projekti) üles Scratch'i kodusaidile, kus teised sellega tutvuda ja seda kommenteerida saavad. Saidil on eraldi osa, kus õpetajad tutvustavad oma õppe‑ metoodikamaterjale [6].

Scratch leiab üha laiemat kasutamist programmeerimise õpetamisel koolides, eriti muidugi USA’s. Näiteks California osariigi koolide (3 aste, 9-10 kl) jaoks CSTA (Computer Science Teachers Assosation) poolt koostatud arvutiteaduste värskes (2009) ainekavas põhineb osa „Sissejuhatus programmeerimisse“ (35 t) täielikult Scratch’il [7]. Scratch’iga alustatakse programmeerimise kursusi näiteks sellistes tuntud ülikoolides nagu Harvardi Ülikool (kõrgeima reitinguga ülikool maailmas) ja Berkeley Ülikool.

Scratch’i programmide (projektide) keskseteks elementideks on graafikaobjektid, mida nimetame spraitideks (ingl. sprite – haldjas või vaim). Program­mide (skriptide) abil saab nendega määrata mitmesuguseid tegevusi: muuta asukohta, suurust, värvust jmt, panna nad kõndima, tantsima, tekitama helisid ja muusikat. Süsteemiga tuleb kaasa üsna suur valik spraite, mis asuvad liikide kaupa (inimesed, loomad, asjad jm) erinevates kaustades. Spraitidena võib kasutada ka oma graafikaobjekte. Lihtsamaid spraite saab ka joonistada Scratch’i koosseisu kuuluva joonistamis­redak­to­riga. Oma olemuselt vastavad Scratch’i spraidid Alice objektidele ja klassidele, ainult nad on tasapinnalised (2D) ja nende käsitlemine on lihtsam. Spraidiga saab siduda heliklippe (MP3 või WAV failid), muutujaid ja massiive.

Allpool on toodud mõned spraitide näited. Esimene sprait: jalutav kassipoeg (eestikeelse nimega Kraps), on Scratch’i logoks. Välja arvatud keskel asuv kriipspois Juku, on need kõik võetud Scratch’i spraitide kaustadest.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cat1-a | duck1 | Juku | ballerina-a | beachball1 soccer1car1 |

Joonis 4. Spraitide näiteid

Spraidil võib olla mitu **kostüümi** (***costume***): tüüpiliselt spraidi erinevad variandid (teisikud). Krapsul on näiteks 2 kostüümi, baleriinil 4. Vahetades neid programmselt teatud sagedusega, saab tekitada vastavaid animatsiooni efekte: kõndimine, tantsimine vm. Alice’s peab taolisi animatsioone programmeerima üksikute alamobjektide (näiteks käed, jalad jane) lõikes, mis on üsna keeruline ja aeganõudev tegevus, eriti arvestades, et tegemist on ruumilise geomeetriaga.

Joonis 5. Kostüümide näiteid

Scratch’i projektis võib olla kasutusel mitu programmiüksust, mida nime­ta­kse skriptideks. Skript (script) koosneb käskudest ehk lausetest, mis kujutavad endast ekraanil graafilisi plokke. Plokid ühendatakse omavahel hiire abil. Skript kuulub alati ühele kindlale spraidile või lavale. Ühel spraidil võib olla suvaline hulk skripte. Skriptid saavad teha omavahel koostööd teadete abil.

Joonisel 6 on toodud Scratch’i kasutajaliides. See koosneb mitmest piirkonnast: lava, skriptide ala, käsuplokkide ala, spraitide ikoonide ala, menüü ja tööriistariba.

**Lava** (*Stage*) on ekraani (kasutajaliidese) piirkond, kus asuvad ja tegutsevad kõik spraidid. Laval võib olla mitu **tausta**, neid saab lisada, eemal­da­da ja redigeerida. Süsteemiga on kaasas suur valik taustu, kuid võib kasutada ka oma graafikafaile.



Joonis 6. Scratch’i kasutajaliides

Skripti loomiseks lohistatakse vajalikud plokid käskude alast skriptide alasse ja ühendatakse need pinuks (stack). Spetsiaalsed plokkide liidese elemendid võimaldavad ühendada plokke ainult ühel viisil. Skriptides võib kasutada muutujaid ja ühemõõtmelisi massiive (loendeid). Enamiksu plokkkides on väljad, kuhu saab sisestada argumente: konstantidena, muutujte või avaldistena. Plokid on funktsioonide alusel jagatud gruppidesse: Liikumine, Välimus, Juhtimine jne.

Skriptide käivitamiseks saab kasutada erinevaid vahendeid. Kui skript algab rohelise lipuga plokiga, saab selle käivitada rohelise lipuga nupuga, mis asub lava ülemises parempoolses nurgas. Samas on ka punane nupp programmi töö katkestamiseks suvalisel hetkel. Korraga võib käivitada mitu skripti, mis hakkavad tööle paralleelselt. Scratch’i programme saab täita otse kasutajaliideses. Neid saab väga lihtsalt paigutada apletidena ka veebi dokumentidesse.

## Scratch’i programmi näide

Allpool toodud skriptide näited projektist, kus tegutsevad kolm spraiti: Kraps (2 kostüümi, 2 skripti), baleriin (4 kostüümi, 1 skript), breikar (4 kostüümi, 1 skript). Üks skript on ka laval. Kasutusel on ka üks muutja: ***n***, mida kasutakse korduste arvu määramiseks Krapsu skriptis ***Hüppa***. Parameetreid ja argumente Scratch’is ei ole. Kui klõpsatakse rohelist lippu hakkavad korraga tööle kolm skripti, mis algavad vastava plokiga.

Krapsu skript tekitab kõigepealt heli „näu“, kasutades olemasolevat heliklippi. Edasi võetakse muutuja ***n*** väärtuseks **3** ning käivitakse skript ***Hüppa***. Käsuga korda imitee­ritakse Krapsu hüppamist, korrates sisemisi käske kolm korda. Paralleelselt sellega töötab ka baleriini skript. Peale algseadeid ja tervitust, tekitakse kaheksakordse kostüümide vahetusega tantsu imitatsioon: kostüü­mi­dele (neid on neli) tehakse nö kaks ringi peale: kui jõutakse viimase kostüümini, jätkatakse algusest. Seejäärel käivitakse käsuga teavita breikari skript ja antud skript jääb ooteseisu. Peale skripti ***Tantsi*** täitmist tuleb täitmisjärg tagasi. Kuvatakse teade „Nägemist!“ ja käsk peata kõik lõpetab kogu programmi töö.

|  |  |  |
| --- | --- | --- |
| 0_Demo_Kraps_3Spraidi Kraps skriptid | 0_Demo_Baleriin_3Spraidi baleriin skript | 0_Demo_Pois_3Spraidi breikar skriptid |

Breikari skript ***Tantsi*** kuvab küsimuse, loeb kuvatavast sisendväljast kasutaja vastuse ning omistab selle sisemuutujale ***vastus***. Edasi tuleb nö klassikaline kahendvalik. Kui kasutaja sisestas väärtuse **jah**, käivitakse taustamuusika (klipp nimega **hiphop**), võetakse muutuja ***n*** näärtuseks **7** ja käivitakse Krapsu skript ***Hüppa***. Vastupidisel juhul kuvatakse teade „Kahju!“. Sõltumata eelnevast, jätkatakse käsuga korda, mille abil imiteeritakse kostüümide vahetamisega breiktantsu. **NB!** Kui käivitadi skript ***Hüppa***, siis selle töö lõppu ei oodata, vaid jätkatakse kohe antud skripti täitmist. Peale skripti ***Tantsi*** töö lõppu, täitmisjärg aga läheb tagasi baleriini skriptile, kust toimus pöördumine käsuga teavita ... ja oota.

|  |  |
| --- | --- |
| 0_Demo_Lava_3**Lava** skript | Lava skript töötab kogu aeg paralleelselt teistega. Käsk lõputult pidevalt kordab kahte sisemist käsku: muudab lava värvi ja teeb vahele pausi. Viimase pikkus valitakse juhusliku arvu abil. Korduse ja skripti töö katkestab käsk peata kõik baleriini skriptis. |

## Kokkuvõtteks

Oleme kasutanud Scratch’i programmeerimise kursuse sisejuhatavas osas alatest aastast 2008 mitteinformaatikute jaoks, altes aastast ka 2009 informaatikute jaoks. Umbes 35 tunnilise mahuga kursusest Scratch’i kasutame alguses 6‑10 tundi, sõltuvalt erialast ja õppevormist. Osa lõppeb omal individuaalse tööga, mille teema ja realisatsiooni valivad õppijad ise. Kursus jätkub mitteinformaatikutele VBA’ga Exceli keskkonnas, informaatikutele - VB .NET’iga Visual Studio keskkonnas. Tulemused, eriti mitteinfor­maa­tikute osas, on nö silmaga nähtavad. On süvenenud huvi aine vastu ning on kiirenenud ja paranenud programmeerimise aluste omandamine., mis edaspidi lihtsustab tööd VBA’ga.

Oleme õpetanud ja tutvustanud Scratch’i mitmetel kursustel nii õpetajatele kui ka õpilastele (alates 8. klassist). Kahes Tallinna gümnaasiumis kasutatakse Scratch’i programmeerimise valikkursuses. Saadud kogemused lubavad teha järelduse, et taoline lähenemisviis õigustab ennast. Uutes gümnaasiumide õppekavadesse on lülitatud ka meie poolt pakutud valikaine „Rakenduste loomise ja programmeerimise alused“. Selle kursuse läbiviimisel, soovitame kasutada praktikumides mõnda uue põlvkonna keelt, eesätt me soovitame Scratch’i. Olgu märgitud et nn „Tiobe indeksis“, kus püütakse võrrelda erinevate keelte populaarsust ja kasutamise ulatust ning, mis algab Java, C-de ja Basicuga, oli Scratch veebruaris 2010 juba 22. kohal (Alice oli 31).

**Kui võrrelda** Scratch'i **Alice’ga,** peab märkima, et viimane sisaldab nö võimsamaid vahendeid, võimaldades kasutada 3D graafikat ja otseslt objektipõhist lähenemisviisi (klassid, objektid, meetodid jmt). Kuid selle Alice kasutamine on siiski märgatavalt keerulisem, eriti ruumilise geeomeetria tõttu, mis iseenesest on mõnedele üsna raske valdkond. **Scratch** on lihtsam, mõnevõrra üldisema iseloomuga ja on leidnud viimasel ajal ka laiemat kasutamist. Kuid üldiselt võttes nad täiendavad teineteist ning, arvestades nende kasutamise lihtsust ja seda et tegemist on vabavaraliste toodetega, võib kasutada ka mõlemat koos või järjestiku. Taolist käsitlus on näiteks vaadeldud töös [4].

## Viidatud allikad

1. Programmeerimise õppimise/õpetamise keeltest Wikipedias. [http://en.wikipedia.org/wiki/Category:Educational\_programming\_languages](http://en.wikipedia.org/wiki/Category%3AEducational_programming_languages)
2. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, Yasmin Kafai. Scratch: Programming for All. [Communications](http://cacm.acm.org/about-communications) of the ACM, november 2009, Vol. 52, NO. 11
3. David J. Malan and Henry H. Leitner. S[ratch for Budding Computer Scientists](http://www.cs.harvard.edu/malan/publications/fp079-malan.pdf). 38th ACM Technical Symposium on Computer Science Education. Covington, Kentucky. March 2007.
4. Programmeerimise õpetamisest: Scratch => Alice => Java. <http://www.dickbaldwin.com/tocHomeSchool.htm>
5. Scratch’i abi‑ ja õppematerjalid erinevates keeltes. <http://info.scratch.mit.edu/Languages>
6. Scratch’i õppematerjalide sait. <http://scratched.media.mit.edu/>
7. Arvutiteaduste ainekava. <http://csta.acm.org/Curriculum/sub/ExploringCS.html>

Autorid:

Tehnikaülikooli Informaatikainstituudi õppejjõud:

Kaarel Allik, Irina Amitan, Kersti Antoi, Jaan Übi, Jüri Vilipõld