



**RĪGAS TEHNISKĀ
UNIVERSITĀTE**

Rīgas Tehniskā universitāte, Reģ. Nr. 90000068977, Ķīpsalas iela 6A, Rīga, LV- 1048, Latvija
Tālr. 67089999, fakss 67089710, e-pasts: rtu@rtu.lv, www.rtu.lv

Aktualizācijas nepieciešamības pamatojuma apraksts

Mācību priekšmeta “Programmēšana I” programma tika izstrādāta projekta Skola2030 ietvaros, balstoties uz Ministru kabineta 27.11.2018. noteikumiem Nr. 416 *“Noteikumi par valsts vispārējās vidējās izglītības standartu un vispārējās vidējās izglītības programmu paraugiem”*.

Šī programma ir kalpojusi par nozīmīgu pamatu skolēnu algoritmiskās un loģiskās domāšanas attīstīšanai, taču kopš tās izstrādes ir notikušas būtiskas izmaiņas tehnoloģiju vidē, darba tirgus pieprasījumā un mākslīgā intelekta attīstībā.

Pedagogu un ekspertu izvērtējums rāda, ka esošā programma vairs pilnībā neatbilst mūsdienu tehnoloģiskajām un izglītības vajadzībām. Programmas saturs bija orientēts galvenokārt uz kodēšanas sintaksi un atsevišķiem algoritmiem, taču skolēniem trūka izpratnes par programmēšanas integrāciju ar datiem, drošību, mākoņpakalpojumiem un MI.

Tāpēc tika veikta pedagogu aptauja un karsto punktu (*Hotspots*) analīze, lai identificētu satura pārslogotās un vājāk pārstāvētās jomas un izstrādātu zinātniski pamatotu programmas aktualizācijas modeli, saglabājot kopējo 210 stundu apjomu.

1 Pedagogu aptaujas rezultāti

Lai nodrošinātu programmas aktualizācijas atbilstību izglītības procesa vajadzībām, tika veikta pedagogu aptauja, kurā piedalījās skolotāji no Latvijas skolām.

Aptaujas mērķis bija noteikt, kuras programmēšanas tēmas skolotāji uzskata par visproblemātiskākajām, un kādas satura pārmaiņas nepieciešamas, lai mācību process kļūtu mūsdienīgāks, efektīvāks un jēgpilnāks.

Aptaujā iekļautie jautājumi galvenokārt bija vērsti uz to, kurās programmēšanas satura jomās pedagogi saskata lielāko pilnveides nepieciešamību un kuras prasmes mācību procesā būtu jāstiprina.

Pedagogi vērtēja katru jomu skalā no 1 līdz 5 (no “nav nepieciešams” līdz “ļoti nepieciešams”).

Aptaujas kvantitatīvie rezultāti

Aptaujas tēma / jautājums	% skolotāju, kas norādīja nepieciešamību	Galvenās piezīmes no respondentiem
Testēšanas un atklādošanas prasmes jāpaplašina	61 %	“Testēšana ir formāla, skolēni nezina, kā pārbaudīt kļūdas vai izmantot automatizētus testus.”
Datu apstrāde un vizualizācija jāpadara praktiskāka	57 %	“Nepieciešams reāls darbs ar datiem, CSV un API, ne tikai teorētiska apstrāde.”
Risinājuma plānošana un dizaina domāšana jāiekļauj programmā	54 %	“Skolēni nesaprot, kā pārvērst ideju par produktu; trūkst projekta plānošanas posma.”
Mākslīgā intelekta (MI) rīki jāintegrē mācību procesā	48 %	“MI palīdz izskaidrot kodu un radīt idejas, bet skolēniem jāmaāca, kā to lietot droši.”
Datu drošība un ētika jāstiprina	45 %	“Koda drošums, datu aizsardzība un autortiesības nav pietiekami uzsvērtas.”
Sadarbības rīki un versiju kontrole jāiekļauj programmā	39 %	“Nepieciešams ieviest GitHub, komandas darbu un koda recenzēšanu.”
Integrācija ar citām jomām (matemātika, dizains, ekonomika) jāattīsta	37 %	“Programmēšana ir pārāk noslēgta, trūkst saistes ar citiem priekšmetiem.”

Galvenās tendences:

- Lielākā daļa pedagogu norāda uz programmēšanas priekšmeta “slēgto ciklu” — skolēni apgūst koda rakstīšanu, bet nezina, kā to pielietot plašākā kontekstā (dati, dizains, automatizācija).
- Grūtības rada pāreja no teorijas uz praksi, jo trūkst pietiekami daudz uzdevumu ar reāliem datiem un projektveida darbu.
- Testēšanas un atklādošanas tēmas uzskatāmas par kritiski vāji pārstāvētām (Hotspots intensitāte = 2).
- MI integrācija un datu drošība kļuvušas par mācību prioritātēm, kas atspoguļo tehnoloģiju attīstības tendences pēdējos 3-5 gados.
- Pedagogu noskaņojums kopumā pozitīvs: 82 % skolotāju atbalsta priekšmeta programmas modernizāciju, saglabājot kopējo 210 stundu apjomu, bet pārdalot laiku starp tēmām.

2 Programmas stundu sadalījuma izmaiņu pamatojums

Balstoties uz veiktās pedagogu aptaujas rezultātiem, tika pārskatīts mācību programmas tematisko vienību stundu sadalījums. Aptaujā iegūtie dati parāda, ka pedagogi saskaras ar vairākām problēmām programmēšanas mācību procesā: nepietiekamu praktisko uzdevumu skaitu, grūtībām pāriet no teorijas uz praktisku risinājumu izstrādi, kā arī ierobežotu programmēšanas sasaisti ar datu analīzi un reāliem pielietojuma scenārijiem.

Vienlaikus 82 % pedagogu norādīja, ka atbalsta programmas modernizāciju, saglabājot kopējo 210 stundu apjomu, bet pārskatot stundu sadalījumu starp tematiskajām vienībām. Ņemot vērā šos rezultātus, programmā tika veikta stundu pārdale, lai vairāk laika veltītu praktiskai programmēšanai un darbam ar datiem, vienlaikus integrējot citas prasmes programmas kopējā struktūrā.

Programmēšanas pamatu apguves paplašināšana

Tēma: Kā rakstīt programmu, izmantojot izvēlēto programmēšanas valodu?

Stundu skaits: no 30 uz 60 stundām

Stundu skaita palielinājums šajā tematā saistīts ar pedagogu aptaujā identificēto problēmu, ka skolēniem bieži trūkst pietiekamu praktisku iemaņu programmēšanas pamatos. Aptaujas rezultāti parāda, ka skolotāji norāda uz grūtībām pārejā no teorētiskas izpratnes uz praktisku programmas izstrādi.

Papildu stundas ļauj:

- nodrošināt vairāk praktisku uzdevumu;
- pakāpeniski attīstīt algoritmisko domāšanu;
- veicināt skolēnu spēju patstāvīgi izstrādāt vienkāršus programmatūras risinājumus.

Šāda pieeja palīdz novērst pedagogu aptaujā identificēto problēmu, ka programmēšanas mācību process bieži veido "slēgtu ciklu", kurā skolēni apgūst koda rakstīšanu, bet nepietiekami attīsta spēju izmantot šīs prasmes plašākā kontekstā.

Risinājuma plānošanas tēmas optimizēšana

Tēma: Kā izplānot un izveidot lietotājam atbilstošu programmētu risinājumu un uzraudzīt tā īstenošanu?

Stundu skaits: no 30 uz 25 stundām

Pedagogu aptaujas rezultāti liecina, ka 54 % respondentu uzskata, ka programmā nepieciešams stiprināt risinājumu plānošanas un dizaina domāšanas aspektus. Respondenti norādīja, ka skolēniem bieži trūkst izpratnes par to, kā no idejas nonākt līdz strukturētam digitālam risinājumam.

Programmas aktualizācijas procesā tika secināts, ka šīs prasmes efektīvāk attīstāmas integrējot plānošanas darbības programmēšanas un projektu izstrādes procesā, nevis koncentrējot tās vienā atsevišķā tematā.

Tādēļ aktualizētajā programmā:

- risinājuma plānošana tiek izmantota programmēšanas uzdevumos, kur skolēni vispirms definē problēmu un izstrādā algoritmisko risinājumu;
- projektveida darbos skolēni veido vienkāršus sistēmas plānus, datu struktūras un lietotāja saskarnes uzmetumus;
- plānošanas un dizaina elementi tiek integrēti arī datu apstrādes un risinājuma izstrādes posmos.

Ņemot vērā, ka plānošanas prasmes tiek attīstītas vairākās programmas tēmās, nepieciešamais laiks šai tematiskajai vienībai tika optimizēts līdz 25 stundām, saglabājot būtiskāko teorētisko pamatu un vienlaikus nodrošinot šo prasmju praktisku pielietojšanu citos mācību posmos.

Šāda pieeja ļauj plānošanas prasmes attīstīt kā nepārtrauktu programmatūras izstrādes procesa sastāvdaļu, kas labāk atbilst reālajai digitālo risinājumu izstrādes praksei.

Datu apstrādes un vizualizācijas tēmas paplašināšana

Tēma: Kā automātiski apstrādāt un attēlot datus?

Stundu skaits: no 40 uz 45 stundām

Pedagogu aptaujas rezultāti parāda, ka 57 % respondentu uzskata, ka datu apstrādes un vizualizācijas tēma jāpadara praktiskāka. Respondenti norādīja, ka mācību procesā bieži tiek izmantoti vienkārši vai mākslīgi dati, kas ierobežo skolēnu izpratni par datu analīzes praktisko nozīmi. Tāpēc pedagogi uzsvēra nepieciešamību vairāk strādāt ar reāliem datu avotiem, tīmekļa datiem vai citiem strukturētiem datu formātiem.

Ņemot vērā šos rezultātus, programmā šai tematiskajai vienībai paredzētais laiks tika palielināts līdz 45 stundām, lai nodrošinātu vairāk praktisku mācību aktivitāšu. Papildu stundas paredzētas, lai skolēni varētu:

- apgūt dažādu datu formātu nolasīšanu un apstrādi;
- veikt datu analīzes darbības, izmantojot programmēšanas valodas rīkus;
- attēlot datu analīzes rezultātus grafiskā vai strukturētā veidā;
- strādāt ar vienkāršiem datu kopumiem no reāliem avotiem.

Šāda pieeja veicina datu pratības attīstību un programmēšanas sasaisti ar praktiskiem pielietojuma scenārijiem, kas atbilst arī pedagogu aptaujā identificētajai nepieciešamībai stiprināt programmēšanas izmantošanu plašākā kontekstā, piemēram, datu analīzē, ekonomikā vai citās mācību jomās.

Stundu skaita palielinājums līdz 45 stundām ļauj nodrošināt līdzsvaru starp teorētisko izpratni un praktisku datu analīzes uzdevumu izpildi, vienlaikus saglabājot kopējo programmas apjomu 210 stundas.

Testēšanas un atklādošanas tēmas integrācija

Tēma: Kā sistemātiski pārbaudīt un atklādot risinājumu?

Stundu skaits: no 20 uz 15 stundām

Pedagogu aptaujas rezultāti liecina, ka 61 % respondentu norādīja uz nepieciešamību stiprināt testēšanas un atklādošanas prasmes, uzsverot, ka skolēniem bieži trūkst sistemātiskas pieejas kļūdu

identificēšanai un programmas darbības pārbaudei. Respondenti arī norādīja, ka testēšana mācību procesā nereti tiek īstenota formāli un netiek pietiekami integrēta programmas izstrādes procesā.

Ņemot vērā šos rezultātus, programmas aktualizācijas laikā tika mainīta pieeja testēšanas prasmju apguvei. Aktualizētajā programmā testēšanas un atklūdošanas principi netiek aplūkoti tikai vienā tematā, bet tiek integrēti visos programmatūras izstrādes posmos, piemēram:

programmēšanas uzdevumu izpildē, kur skolēni regulāri pārbauda izstrādātā koda darbību;
projektveida darbos, kuros tiek analizēti iespējamie kļūdu scenāriji un testēti izveidotie risinājumi;
datu apstrādes un automatizācijas uzdevumos, kuros tiek pārbaudīta programmas darbība ar dažādiem datu ievades gadījumiem.

Tādējādi testēšana kļūst par nepārtrauktu programmas izstrādes procesa sastāvdaļu, nevis izolētu mācību tematu. Ņemot vērā, ka būtiska daļa testēšanas aktivitāšu tiek pārcelta uz citām tematiskajām vienībām, šai tēmai paredzētais laiks tika optimizēts līdz 15 stundām, saglabājot nepieciešamo teorētisko pamatu un vienlaikus nodrošinot šo prasmju praktisku pielietojanu visā mācību procesā.

Šāda pieeja veicina ieradumu sistemātiski pārbaudīt programmas darbību katrā izstrādes posmā, kas atbilst gan pedagogu aptaujā identificētajām vajadzībām, gan mūsdienu programmatūras izstrādes praksēm.

Esošu risinājumu izmantošanas tēmas optimizēšana

Tēma: Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu?

Stundu skaits: no 40 uz 25 stundām

Programmas aktualizācijas procesā tika veikta tematiskā satura analīze, izvērtējot sākotnējās programmas interpretāciju Skola2030 kontekstā. Tika konstatēts, ka jēdziens “efektīva risinājumu izmantošana” sākotnējā programmas versijā tika interpretēts pārāk plaši, kā rezultātā šajā tematā tika iekļauti sasniedzamie rezultāti, kas pēc būtības vairāk attiecas uz datu apstrādi un automatizāciju.

Piemēram, tematā tika iekļauti rezultāti, kas saistīti ar:

- standarta bibliotēku izmantošanu,
- API izmantošanu,
- mašīnmācīšanās principu izmantošanu.

Šie aspekti konceptuāli ir ciešāk saistīti ar datu apstrādes un automatizācijas uzdevumiem, tāpēc programmas aktualizācijas laikā tie tika pārcelti uz tematisko vienību “Kā automātiski apstrādāt un attēlot datus?”, kur šīs prasmes tiek apgūtas praktiskā datu analīzes kontekstā. Šāds risinājums atbilst arī aptaujā identificētajai nepieciešamībai padarīt datu apstrādi praktiskāku (57 %).

Vienlaikus programmas aktualizācijā tika iekļautas tēmas, kas stiprina skolēnu algoritmisko domāšanu un programmēšanas pamatu izpratni, piemēram:

- objektorientētās programmēšanas pamati,
- algoritmu efektivitātes novērtēšana (Big O notācija),

- meklēšanas algoritmi,
- kārtošanas algoritmi.

Šo tēmu iekļaušana programmā palīdz attīstīt skolēnu spēju analizēt un izvērtēt programmēšanas risinājumus, kas ir būtiska digitālo risinājumu izstrādes kompetence.

Ņemot vērā tematiskā satura precizēšanu, daļas sasniedzamo rezultātu pārcelšanu uz citām tematiskajām vienībām un jaunu algoritmikas tēmu iekļaušanu programmā, šai tematiskajai vienībai nepieciešamais stundu skaits tika optimizēts līdz 25 stundām, saglabājot tās būtisko saturu un integrējot daļu prasmju projektveida uzdevumos.

Šāda pieeja nodrošina skaidrāku tematisko struktūru programmā un palīdz attīstīt skolēnu spēju analizēt, izvērtēt un izmantot programmēšanas risinājumus dažādos kontekstos, kas atbilst arī pedagogu aptaujā identificētajām mācību procesa pilnveides vajadzībām.

Risinājuma īstenošanas tēmas optimizēšana

Tēma: Kā īstenot risinājumu?

Stundu skaits: no 50 uz 40 stundām

Programmas aktualizācijas procesā tika pārskatīta arī tematiskās vienības “Kā īstenot risinājumu?” vieta un saturs programmas kopējā struktūrā. Pedagogu aptaujas rezultāti liecina, ka viena no būtiskākajām problēmām programmēšanas mācību procesā ir pāreja no teorijas uz praktisku risinājumu izstrādi. Respondenti norādīja, ka skolēni bieži apgūst programmēšanas jēdzienus izolēti, bet nepietiekami attīsta spēju tos izmantot reālu problēmu risināšanā.

Šo problēmu atspoguļo arī aptaujas rezultāti, kuros:

- 54 % pedagogu norādīja uz nepieciešamību stiprināt risinājuma plānošanas un dizaina domāšanas prasmes,
- 57 % uzsvēra nepieciešamību padarīt datu apstrādi praktiskāku,
- kā arī respondenti kopumā norādīja uz grūtībām pāriet no teorētiskas izpratnes uz praktisku risinājumu izstrādi.

Ņemot vērā šos rezultātus, programmas aktualizācijas laikā tika pieņemts lēmums agrāk integrēt praktiskās izstrādes aktivitātes programmēšanas pamatu apguves procesā. Tas nozīmē, ka skolēni jau programmēšanas pamatu apguves posmā veido vienkāršus funkcionējošus risinājumus, izmantojot pakāpeniski sarežģītākus programmēšanas paņēmienus.

Papildus tam metodisko materiālu aprobācijas procesā pedagogi ieteica mainīt tematisko vienību apguves secību, novietojot programmēšanas pamatus un testēšanas prasmes programmas sākumā. Šāda secība nodrošina, ka skolēni vispirms apgūst programmēšanas un testēšanas pamatus, pēc tam datu apstrādes principus un tikai pēc tam pāriet uz kompleksāku projektu plānošanu un īstenošanu.



**RĪGAS TEHNISKĀ
UNIVERSITĀTE**

Rīgas Tehniskā universitāte, Reģ. Nr. 90000068977, Ķīpsalas iela 6A, Rīga, LV- 1048, Latvija
Tālr. 67089999, fakss 67089710, e-pasts: rtu@rtu.lv, www.rtu.lv

Rezultātā praktiskā risinājumu izstrāde programmā notiek visā mācību procesā, nevis koncentrējas tikai vienā tematiskajā vienībā. Ņemot vērā šo pieeju un praktisko aktivitāšu pārdali starp vairākām tematiskajām vienībām, tematam “Kā īstenot risinājumu?” paredzētais stundu skaits tika optimizēts līdz 40 stundām, vienlaikus saglabājot iespēju skolēniem izstrādāt pilnvērtīgus programmēšanas projektus.

Šāda pieeja nodrošina pakāpenisku prasmju attīstību un palīdz novērst pedagogu aptaujā identificēto problēmu – programmēšanas mācību procesa “slēgto ciklu”, kurā skolēni apgūst atsevišķas tehniskas prasmes, bet nepietiekami attīsta spēju tās izmantot kompleksu digitālu risinājumu izstrādē.

3 Tematu secības aktualizācija programmas īstenošanā

Metodisko materiālu izstrādes un aprobācijas procesā pedagogi sniedza ieteikumu mainīt vairāku tematisko vienību apguves secību. Aprobācijas laikā tika secināts, ka sākotnēji paredzētā tematu secība ne vienmēr nodrošina pakāpenisku programmēšanas prasmju attīstību, jo daļa skolēnu sāk darbu ar projektu plānošanas vai risinājuma īstenošanas aktivitātēm pirms pietiekami nostiprinātas programmēšanas un testēšanas pamatu izpratnes.

Balstoties uz aprobācijas laikā iegūtajiem novērojumiem un pedagogu rekomendācijām, tika ieteikts programmas apguvē izmantot secību, kas nodrošina pakāpenisku pāreju no programmēšanas pamatu apguves uz sarežģītāku risinājumu izstrādi un projektu darbu.

Aprobācijas rezultātā ieteiktā tematu apguves secība ir šāda:

- Kā rakstīt programmu, izmantojot izvēlēto programmēšanas valodu?
- Kā sistemātiski pārbaudīt un atklājot risinājumu?
- Kā automātiski apstrādāt un attēlot datus?
- Kā izplānot un izveidot lietotājam atbilstošu programmētu risinājumu un uzraudzīt tā īstenošanu?
- Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu?
- Kā īstenot risinājumu?

Šāda tematu secība nodrošina loģisku prasmju attīstības progresiju: vispirms tiek nostiprinātas programmēšanas pamatiemaņas, pēc tam skolēni apgūst programmas testēšanu un datu apstrādes principus, un tikai pēc tam pāriet uz kompleksāku digitālo risinājumu plānošanu, izstrādi un īstenošanu.

Tematu secības aktualizācija ļauj efektīvāk organizēt mācību procesu un veicina skolēnu spēju pakāpeniski attīstīt prasmes digitālu risinājumu izstrādē, kas tika identificēts arī pedagogu aptaujā kā būtisks uzlabojuma virziens.

4 Metodisko materiālu izstrādes pamatojums

Esošo mācību līdzekļu un metodisko materiālu izvērtējums

Izvērtējot esošos mācību līdzekļus un metodiskos materiālus mācību priekšmetam “Programmēšana I”, secināms, ka lielākā daļa no tiem tika izstrādāti ESF projekta Nr. 8.3.1.1/16/I/002 “Kompetenču pieeja mācību saturā” (Skola2030) īstenošanas laikā 2018.–2020. gadā.

Šie materiāli veicināja programmēšanas mācību satura ieviešanu skolās un nodrošināja skolēniem iespēju apgūt programmēšanas pamatus, attīstot algoritmisko un loģisko domāšanu.

Papildus tika izvērtēti arī publiski pieejamie mācību resursi platformā mape.gov.lv, kurā apkopoti Skola2030 ietvaros izstrādātie mācību satura paraugi un tematu apraksti. Analīze liecina, ka platformā pieejamie materiāli nodrošina mācību satura konceptuālo ietvaru, ietverot tematu aprakstus, sasniedzamos rezultātus un atsevišķus mācību aktivitāšu piemērus.

Vienlaikus konstatēts, ka pieejamie resursi galvenokārt fokusējas uz mācību satura aprakstu, bet ierobežotā apjomā nodrošina detalizētus metodiskos norādījumus mācību procesa plānošanai un īstenošanai.

Materiālā nav iekļauti:

- konkrēts stundu sadalījums tematiskajās vienībās;
- strukturēti stundu plāni un mācību aktivitāšu secība;
- praktisko uzdevumu vai projektu piemēri;
- metodiskie ieteikumi diferencētai mācību organizācijai.

Tādējādi secināms, ka pieejamie materiāli galvenokārt nodrošina mācību satura struktūru, bet nepilnīgi atbalsta pedagogu darbu mācību procesa praktiskajā plānošanā un organizēšanā.

Papildus analizējot esošos metodiskos materiālus, konstatēti arī vairāki metodiskie ierobežojumi:

- izmantotie piemēri un uzdevumi bieži balstās tehniskos vai abstraktos scenārijos, kas mazāk veicina skolēnu motivāciju un spēju risināt problēmas reālā kontekstā;
- nav integrēti datu analīzes, sadarbības rīku, versiju kontroles un digitālās drošības aspekti, kas mūsdienu programmēšanas praksē ir būtiska sastāvdaļa;
- mācību materiāli neatbalsta diferencētu un adaptīvu mācīšanos, kas nepieciešama skolēniem ar dažādām iepriekšējām zināšanām un mācīšanās tempu.

Vienlaikus izvērtēti arī programmēšanas apgūvē paredzētie sasniedzamie rezultāti Skola2030 mācību satura paraugā, kas paredz, ka skolēnam jāprot:

- izstrādāt algoritmu un pārbaudīt tā pareizību;
- izstrādāt, testēt un pilnveidot programmu, izmantojot programmēšanas valodas un digitālos rīkus;
- analizēt datus un izmantot digitālās tehnoloģijas problēmu risināšanai;
- sadarboties digitālā vidē un ievērot datu drošības principus.



**RĪGAS TEHNISKĀ
UNIVERSITĀTE**

Rīgas Tehniskā universitāte, Reģ. Nr. 90000068977, Ķīpsalas iela 6A, Rīga, LV- 1048, Latvija
Tālr. 67089999, fakss 67089710, e-pasts: rtu@rtu.lv, www.rtu.lv

Ņemot vērā konstatēto, tiks pilnveidoti metodiskie materiāli, saglabājot Skola2030 kompetenču pieejas principus un papildinot tos ar mūsdienu programmēšanas prakses un digitālās prasmes elementiem. Pilnveidotajā materiālu kopumā tiks iekļauts detalizēts stundu sadalījums un pilns metodisko materiālu komplekts, kas ietvers mērķu un uzdevumu definēšanu, mācību satura aprakstu, mācību sasniegumu vērtēšanas formas un metodiskos paņēmienus, ieteikumus mācību darba organizācijai, mācību satura apguves norisi, kā arī mācību satura apguves secību, norādot tematu, temata apguves mērķi, ieteicamo laiku mācību stundās un sasniedzamos rezultātus, skaidri definējot apgūstamās zināšanas un prasmes.

5 Ārējās vides ietekmes pamatojums

Priekšmeta “Programmēšana I” programmas aktualizācija veikta, ņemot vērā tehnoloģiju attīstību, darba tirgus pieprasījumu un pārmaiņas digitālajā sabiedrībā. Pēdējos gados programmēšanas nozīme ir būtiski paplašinājusies – tā vairs nav tikai informācijas tehnoloģiju nozares kompetence, bet kļūst par nozīmīgu prasmi dažādās profesionālajās jomās un izglītības nozarēs.

Tehnoloģiju un digitālās vides attīstība

Strauja attīstība novērojama mākslīgā intelekta, datu apstrādes un automatizācijas tehnoloģijās. Ņemot vērā šīs tendences, programmēšanas saturs tiek papildināts ar tēmām, kas atspoguļo aktuālās digitālo tehnoloģiju attīstības virzienus.

Tādēļ mācību saturā:

- tēmā “Datu apstrāde un attēlošana” iekļauti praktiski uzdevumi, kas saistīti ar datu analīzi, vizualizāciju un mākslīgā intelekta rīku izmantošanu;
- tēmā “Testēšana un atklādošana” paredzēta programmu testēšana un kļūdu analīze, izmantojot mūsdienīgus izstrādes rīkus;
- tēmā “Risinājuma plānošana” uzsvērtā lietotāja vajadzībām orientēta risinājumu izstrāde un dizaina domāšanas principi.

Šādas izmaiņas palīdz skolēniem izprast, kā programmēšana tiek izmantota dažādās digitālajās sistēmās, piemēram, datu apstrādes, analīzes un automatizācijas risinājumos.

Darba tirgus vajadzības

Darba tirgū pieaug pieprasījums pēc speciālistiem ar programmēšanas, datu apstrādes un digitālo risinājumu izstrādes prasmēm. Šīs prasmes kļūst nozīmīgas ne tikai programmētājiem, bet arī citu nozaru speciālistiem, piemēram, inženieriem, dizaineriem, mārketinga un finanšu jomas darbiniekiem.

Tāpēc programmā:

- lielāka uzmanība pievērsta programmēšanas praktiskajam pielietojumam, strādājot ar reāliem datiem;
- iekļauta programmatūras izstrādes procesa izpratne – no idejas formulēšanas līdz testēšanai un risinājuma pilnveidei;
- nostiprinātas prasmes, kas saistītas ar datu drošību, sadarbību un digitālo rīku izmantošanu.

Šāda pieeja palīdz skolēniem uztvert programmēšanu kā praktisku problēmu risināšanas instrumentu un veido pamatu turpmākai profesionālai attīstībai digitālajās jomās.

Sabiedrības un izglītības vides pārmaiņas

Digitālās tehnoloģijas kļuvušas par būtisku ikdienas sastāvdaļu gan darbā, gan izglītībā. Tādēļ skolēniem nepieciešams attīstīt prasmes droši, atbildīgi un kritiski izmantot digitālās tehnoloģijas.

Programmā:



**RĪGAS TEHNISKĀ
UNIVERSITĀTE**

Rīgas Tehniskā universitāte, Reģ. Nr. 90000068977, Ķīpsalas iela 6A, Rīga, LV- 1048, Latvija
Tālrunis: 67089999, fakss 67089710, e-pasts: rtu@rtu.lv, www.rtu.lv

- programmēšanas tematos integrēti digitālās drošības, autortiesību un digitālās ētikas jautājumi;
- veicināta pašvadītas mācīšanās un sadarbības prasmju attīstība, izmantojot tiešsaistes izstrādes un sadarbības platformas;
- paredzēta mākslīgā intelekta rīku apzināta izmantošana mācību procesā, veidojot izpratni par šo rīku iespējām un ierobežojumiem.

Šādas pieejas mērķis ir nodrošināt, ka mācību saturs ir saistīts ar mūsdienu digitālās sabiedrības vajadzībām un palīdz skolēniem attīstīt prasmes, kas nepieciešamas turpmākajai izglītībai un profesionālajai darbībai.

6 Programmas aktualizācijas kopsavilkums

Mācību priekšmeta “Programmēšana I” programmas aktualizācija veikta, lai nodrošinātu tās atbilstību mūsdienu tehnoloģiju attīstības tendencēm, izglītības standartam un darba tirgus vajadzībām, vienlaikus saglabājot programmas kopējo apjomu 210 akadēmisko stundu ietvaros. Aktualizācijas process balstīts uz empīriskiem datiem, kas iegūti pedagogu aptaujā un karsto punktu (Hotspots) analizē, kā arī esošo metodisko materiālu izvērtējumā.

Projekta Skola2030 ietvaros izstrādātā programmas sākotnējā versija nodrošināja būtisku pamatu skolēnu algoritmiskās un loģiskās domāšanas attīstīšanai. Tomēr kopš tās ieviešanas ir notikušas būtiskas pārmaiņas tehnoloģiju attīstībā un digitālajā vidē. Izvērtējums parādīja, ka programmā lielāks uzsvars tika likts uz programmēšanas sintakses apguvi un atsevišķu algoritmisku paņēmieni izmantošanu, bet mazāk – uz programmēšanas praktisko pielietojumu, darbu ar datiem un sistemātisku risinājumu izstrādi.

Pedagogu aptaujas rezultāti norādīja uz vairākām pilnveides nepieciešamībām: stiprināt programmēšanas pamatu praktisko apguvi, paplašināt darbu ar datu apstrādi un vizualizāciju, sistemātiskāk attīstīt testēšanas un atklādošanas prasmes, kā arī veidot skaidrāku saikni starp programmēšanas uzdevumiem un reāliem pielietojuma scenārijiem. Vienlaikus metodisko materiālu izvērtējums parādīja, ka pedagogiem nepieciešams detalizētāks atbalsts mācību procesa plānošanā, tostarp konkrēts stundu sadalījums, strukturēti stundu plāni, praktisko uzdevumu piemēri un metodiskie ieteikumi mācību darba organizācijai.

Pamatojoties uz iegūtajiem rezultātiem, programmā tika precizēts tematisko vienību saturs, pārskatīts stundu sadalījums un aktualizēta tematu secība. Aktualizētajā programmā lielāka uzmanība pievērsta programmēšanas pamatu nostiprināšanai, datu apstrādes un vizualizācijas praktiskam pielietojumam, kā arī algoritmiskās domāšanas attīstīšanai. Programmā papildus iekļauti temati, kas saistīti ar objektorientētās programmēšanas pamatiem, algoritmu efektivitātes izvērtēšanu, meklēšanas un kārtošanas algoritmiem, kā arī mākslīgā intelekta rīku izmantošanu programmatūras izstrādes procesos. Programmā ietverti arī temati par datortīkliem, mašīnmācīšanās pamatprincipiem un vienkāršiem mašīnmācīšanās algoritmiem, kas ļauj skolēniem veidot izpratni par datu analīzes un mākslīgā intelekta pielietojumu programmēšanas uzdevumu risināšanā. Vienlaikus testēšanas un atklādošanas prasmes tiek attīstītas visā mācību procesā, integrējot tās dažādos programmēšanas uzdevumos un projektos.

Rezultātā aktualizētā programma nodrošina skaidrāku prasmju attīstības secību un stiprina sasaisti starp algoritmisko domāšanu, datu apstrādi un programmēšanas risinājumu praktisku izstrādi. Tā veicina skolēnu spēju analizēt problēmas, izstrādāt programmēšanas risinājumus un izmantot programmēšanas prasmes dažādu uzdevumu risināšanā, vienlaikus nodrošinot pedagogiem skaidrāku metodisko ietvaru programmas īstenošanai.