



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

## **Zinātniski pamatots mācību programmas pārstrādes pamatojums**

Kursa „Programmēšana I” satura aktualizācija balstīta uz empīriskiem datiem, pedagogu aptaujām un pedagoģijas teorijām, kas apstiprina nepieciešamību pāriet no tehniski izolēta satura uz integrētu, praksē balstītu pieeju. Aptaujāto 46 pedagogu dati un Hotspots analīze parāda satura nelīdzsvarotību tēmās par datu apstrādi, risinājuma plānošanu un testēšanu, kas rada mācību pārslodzi.

Satura pārstrukturējums pamatots ar Ž. Piažē (J. Piaget) un L. Vigotska (L. Vygotsky) kognitīvās attīstības teorijām, kas apliecina, ka vidusskolas posmā skolēni spēj apgūt sarežģītus algoritmus, ja mācīšanās tiek sasaistīta ar praktisku darbību un sadarbību. Atbilstoši Eiropas digitālās kompetences ietvaram DigComp 2.2 un Latvijas Izglītības attīstības pamatnostādnēm 2021.-2027. gadam, kursa saturs ir sasaistīts ar DigComp 2.2 digitālās kompetences jomām, īpaši akcentējot datu pratību, digitālā satura radīšanu, problēmrisināšanu un drošu tehnoloģiju lietošanu, tādējādi nodrošinot saskaņu ar Eiropas un Latvijas izglītības attīstības virzieniem.

Ieviestā kompetenču pieeja stiprina pāreju no zināšanu reproducēšanas uz prasmi analizēt, pielietot un pamatot risinājumus praksē, savukārt programmēšanas procesa cikliska organizācija (plānošana - izstrāde - testēšana - optimizācija) nodrošina mācīšanos ar atgriezenisko saiti. Šie empīriskie un teorētiskie pamatojumi apliecina, ka izstrādātais satura pārkārtojums ir pedagoģiski, metodiski un zinātniski pamatots.

## 1 Esošās situācijas analīze

Programmēšanas mācību satura izmaiņu nepieciešamība balstīta uz situācijas izvērtējumu Latvijas vidējās izglītības praksē un pedagogu aptaujas rezultātiem. Analīze veikta, ņemot vērā pedagoģiskos un kognitīvos aspektus, kā arī digitālās kompetences ietvarus.

Veiktajā aptaujā piedalījās 46 vidusskolu pedagogi, kuri māca programmēšanas kursu vai integrētus datorzinātņu priekšmetus. Datu analīze parādīja vairākas būtiskas tendences:

- 73 % pedagogu norādīja, ka programmēšanas kursā trūkst laika praktiskiem uzdevumiem un projektu darbam, īpaši algoritmu veidošanā;
- 57 % uzsvēra nepieciešamību iekļaut mākslīgā intelekta rīku izmantošanas principus un automatizācijas piemērus programmēšanas apgūvē;
- 52 % izcēla, ka testēšanas un atklādošanas prasmes tiek apgūtas nepietiekami dziļi;
- 48 % ieteica paplašināt drošības un datu aizsardzības saturu, iekļaujot autentifikāciju un kodu drošības aspektus;
- 39 % uzsvēra vajadzību mācīt sadarbības un versiju kontroles rīkus.

Papildu kvalitatīvie komentāri liecina, ka:

- skolēni bieži izjūt grūtības pārejā no vizuālās programmēšanas uz programmēšanu, izmantojot programmēšanas valodas;
- teorētiskie temati tiek apgūti izolēti, bez saiknes ar praktisku programmatūras izstrādes kontekstu;
- daļa pedagogu joprojām izmanto mācību vides vai programmēšanas valodas, kas vidusskolas posmā ne vienmēr nodrošina pietiekamu pāreju uz tekstuālu programmēšanu un mūsdienu izstrādes praksi;
- kopējais satura sarežģītības līmenis daļai skolēnu ir atbilstošs, bet atsevišķiem skolēniem - nepietiekami izaicinošs; vienlaikus tematu sadalījums un secība vairākās jomās rada pārslodzes risku.

Balstoties uz šiem rezultātiem, satura pilnveide veikta, ievērojot šādus zinātniski un pedagoģiski pamatotus principus:

- pedagoģiskie faktori - mācīšanās motivācija, piemērotība domāšanas līmenim, strukturēta prasmju apguve;

- kognitīvie faktori - pāreja no konkrētās uz abstrakto domāšanu, algoritmiskās loģikas veidošana;
- sociāli pedagoģiskie faktori - sadarbība, komunikācija un digitālā ētika;
- didaktiskie faktori - secīguma un vertikālās progresijas ievērošana;
- digitālās kompetences faktori - droša digitālo rīku lietošana, datu aizsardzības izpratne, automatizācijas piemēru izmantošana un iepazīšanās ar MI atbalsta rīkiem.

Šī analīze ļauj secināt, ka esošais programmēšanas kursa saturs ir veidots kā noslēgta sistēma jeb “slēgts cikls”, kurā skolēni pārsvarā apgūst programmēšanas valodas sintaksi un atsevišķu algoritmu rakstīšanu, bet trūkst izpratnes par programmēšanas sasaisti ar citiem mācību priekšmetiem un reālām lietojuma vidēm.

Latvijas Universitātes pētījums par programmēšanas izglītības programmu izvēli ietekmējošiem motivācijas faktoriem vidusskolas skolēniem (Milka, 2025) apstiprina šo problēmu, norādot, ka viens no galvenajiem demotivējošajiem faktoriem ir skolu programmēšanas satura neatbilstība augstskolas prasībām un reālajai praksei [1]. Tāpēc programmēšanas kursa satura pilnveides mērķis ir pāriet no izolētas tematu apguves uz sasaistītu un ciklisku programmēšanas procesa izpratni.

### **Aptaujas rezultātu datu ticamības un pietiekamības analīze**

Lai izvērtētu, cik lielā mērā aptaujas rezultāti var kalpot par pamatu programmēšanas kursa satura aktualizācijai, tika analizēta iegūto datu ticamība un pietiekamība. Aptauja tika vērsta uz pedagogu pieredzi, izvērtējot esošā standarta atbilstību, īstenošanas iespējas un tematu secīgumu mācību procesā.

Respondentu skaits: 46 pedagogi no dažādiem Latvijas reģioniem (valsts ģimnāzijas, tehnikumi un vispārizglītojošās skolas).

Mērķauditorija: programmēšanas priekšmetu un datorzinātņu pedagogi vidusskolas posmā (10.-12. klase).

Jautājumu skaits: 16 tematiskie jautājumi, kas aptvēra četras galvenās jomas:

- Mācību standarta atbilstība - cik lielā mērā noteiktais saturs atbilst skolēnu prasmju līmenim un pieejamajiem resursiem;
- tematu loģiskā struktūra un secība - vai priekšmeta temati ir sabalansēti un secīgi apgūstami;

- reālā izpilde - vai noteiktās prasmes ir iespējams kvalitatīvi apgūt noteiktajā stundu skaitā;
- tematu aktualitāte un nepieciešamība - vai esošais saturs atspoguļo mūsdienu tehnoloģiju un darba tirgus prasības.

Atbilžu forma: Likerta skala (1 - "neatbilst" līdz 5 - "pilnībā atbilst").

Dati analizēti, izmantojot aprakstošās statistikas un ticamības testus. Iegūti šādi rezultāti:

- vidējais vērtējums par standarta atbilstību: 3,6 no 5 - tas norāda, ka lielākā daļa pedagogu uzskata saturu par daļēji atbilstošu, tomēr nepieciešami precizējumi noteiktās tēmās (īpaši testēšana, MI un drošība);
- vidējais vērtējums par tematu izpildāmību: 3,2 no 5, kas nozīmē, ka daļa pedagogu saskata grūtības kvalitatīvi īstenot visas prasmes noteiktajā laika posmā;
- vidējais vērtējums par tematu secīgumu un loģisko struktūru: 3,8 no 5 - kopumā secība ir vērtēta pozitīvi, tomēr ir uzsvērtā nepieciešamība labāk sasaistīt programmēšanu ar datu apstrādi un testēšanu;
- vidējais vērtējums par satura aktualitāti: 4,0 no 5 - pedagogi atzīst, ka programmēšana ir nozīmīga un nepieciešama, tomēr saturs jāpapildina ar modernām tehnoloģijām.

*Cronbach's Alpha* (iekšējās konsekvences rādītājs) = 0,81, kas pārsniedz minimālo robežu (0,7), apliecinot datu uzticamību.

Ņemot vērā respondentu skaitu un mērķauditorijas specifiku, aptaujas rezultāti sniedz pietiekamu indikatīvu pamatu biežāko problēmzonu identificēšanai un satura pilnveides virzienu noteikšanai.

### **Aptaujas rezultātu interpretācija**

Aptaujas dati liecina, ka esošais programmēšanas standarts ir konceptuāli piemērots, bet daļēji pārslogots un ne vienmēr pielāgots skolēnu apgūšanas tempam. Pedagogi atzīmēja, ka tematu apguve bieži aprobežojas ar sintakses apguvi un algoritmu rakstīšanu, nepietiekami attīstot testēšanas, sadarbības un integrācijas aspektus.

Ņemot vērā respondentu skaitu un mērķauditorijas specifiku, aptaujas rezultāti sniedz pietiekamu indikatīvu pamatu biežāko problēmzonu identificēšanai un satura

pilnveides virzienu noteikšanai. Šī izolētība mazina motivāciju un ierobežo programmēšanas uztveri kā radošu un starpdisciplināru darbību.

Līdz ar to aptaujas rezultāti sniedz skaidru empīrisku pamatojumu programmēšanas satura pārkārtošanai, lai nodrošinātu atvērtu, integrētu un praktiski orientētu mācību pieeju, kas balstās uz reāli īstenojamām, secīgām un mūsdienām atbilstošām tēmām.

## Karsto punktu (Hotspots) analīze un vizualizācija

### *Metodes izvēles pamatojums*

Lai identificētu programmēšanas mācību satura nelīdzsvarotību un noteiktu jomas, kur nepieciešama precizēšana vai paplašināšana, tika izmantota karsto punktu “Hotspots” analīzes metode.

Šī metode ir piemērota izglītības satura pētniecībai, jo ļauj kvantitatīvi novērtēt satura intensitāti un kvalitāti pēc noteiktiem kritērijiem un vizuāli attēlot problēmzonas.

Metode apvieno aptaujas rezultātus, pedagogu kvalitatīvos komentārus un stundu sadalījuma analīzi, identificējot:

- kuras tēmas programmā ir pārslogotas vai pārāk sarežģītas;
- kurās jomās trūkst secīguma un sasaistes ar citām prasmēm;
- kurās tēmās skolēniem un pedagogiem visbiežāk rodas grūtības vai laika deficīts.

Šāda pieeja ir noderīga izglītības satura analīzē, jo tā palīdz vizualizēt problēmzonas un sasaistīt empīriskos novērojumus ar didaktiskiem secinājumiem.

### *Analīzes gaita*

Pamatojoties uz 46 pedagogu aptaujas datiem un kursa “Programmēšana I” esošo satura struktūru (tā ietver 6 tēmas - koda rakstīšana, risinājuma plānošana, datu apstrāde, testēšana, esošu risinājumu izmantošana, īstenošana), katrai tēmai tika piešķirta intensitātes vērtība no 0 līdz 2, kur:

| Vērtība | Nozīme             | Kritērijs                                                               |
|---------|--------------------|-------------------------------------------------------------------------|
| 0       | Nav problēmas      | Tēma ir sabalansēta, laiks pietiekams, saturs īstenojams                |
| 1       | Vidēja intensitāte | Tēma iekļauta, bet trūkst dziļuma vai sasaistes ar citām jomām          |
| 2       | Augsta intensitāte | Tēma nepietiekami pārstāvēta, pārslogota, vai neatbilst skolēnu līmenim |

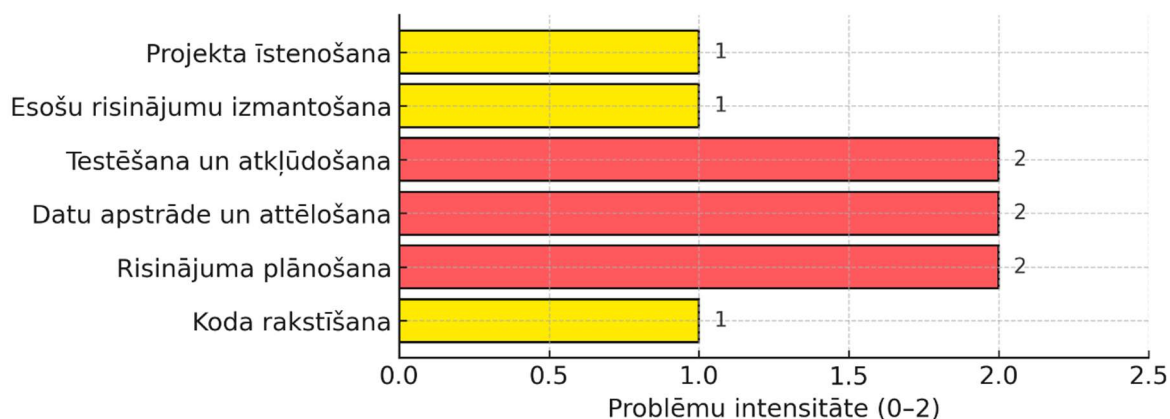
Datu vērtējumi balstīti uz trīs avotiem:

- pedagogu vērtējumi par tematu izpildāmību un atbilstību standartam;
- komentāru kvalitatīva analīze, īpaši par laika sadali un MI integrācijas trūkumu;

- salīdzinājums starp standarta saturu un reālo situāciju praksē (stundu īpatsvars katrai tēmai).

## Rezultāti

| Tēma                                                                | Problēmu intensitāte (0-2) | Pamatots skaidrojums                                                                                                                                 |
|---------------------------------------------------------------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kā uzrakstīt kodu pēc dotas specifikācijas?                         | 1                          | Temats īstenojams, bet trūkst pārejas no vizuālās uz programmēšanu, izmantojot programmēšanas valodas un reālu piemēru.                              |
| Kā izplānot un izveidot lietotājam atbilstošu programmu risinājumu? | 2                          | Pedagogi atzīmē, ka skolēniem grūti izprast lietotāja vajadzību analīzi un risinājuma plānošanu, nepieciešama strukturēta pieeja (dizaina domāšana). |
| Kā automātiski apstrādāt un attēlot datus?                          | 2                          | Augsta intensitāte - trūkst sasaistes ar citiem priekšmetiem un praktiskas pieredzes darbā ar datiem.                                                |
| Kā sistemātiski pārbaudīt un atklāt risinājumu?                     | 2                          | Lielākā problēmu zona - testēšana notiek virspusēji, laiks nepietiekams, trūkst strukturētas pieejas atklādošanai un automatizētai testēšanai.       |
| Kā izvēlēties un efektīvi izmantot jau esošu risinājumu?            | 1                          | Vidējās intensitātes problēma - skolēni reti prot analizēt citu kodu un droši izmantot bibliotēkas.                                                  |
| Kā īstenot risinājumu?                                              | 1                          | Projekta daļa tiek īstenota, bet bieži bez drošības un dokumentācijas elementiem, nepieciešams lielāks uzsvars uz gala produkta kvalitāti.           |



## Vizualizācija

Attēlā vizuāli atspoguļotas tēmas pēc problēmu intensitātes līmeņa (0 = zaļš, 1 = dzeltens, 2 = sarkans):

- sarkana josla (augsta intensitāte) - "Risinājuma plānošana", "Datu apstrāde un attēlošana", "Testēšana un atklādošana";
- dzeltena josla (vidēja intensitāte) - "Koda rakstīšana", "Esošu risinājumu izmantošana", "Projekta īstenošana";
- zaļa josla (zema intensitāte) - netika konstatētas, tātad neviena tēma nav bez nepieciešamības pēc uzlabojumiem.

### *Analīzes interpretācija*

"Karsto punktu" analīze parāda, ka kritiskākās jomas ir saistītas ar praktisko prasmju pielietojumu un satura integrāciju.

Visaugstākā problēmu intensitāte (2) konstatēta datu apstrādes, risinājuma plānošanas un testēšanas tēmās, kur skolēniem trūkst sistemātiskas pieejas un pietiekama praktiskā nostiprinājuma.

Vidēja intensitāte (1) ir novērota projekta īstenošanā un koda rakstīšanā, kur skolēni strādā tehniski, bet bieži bez konteksta.

Šie rezultāti atklāj, ka programmēšanas saturs pašlaik neveido pietiekamu sasaisti starp plānošanu, īstenošanu un pārbaudi, tādējādi radot risku, ka skolēns redz programmēšanu kā izolētu darbību, nevis ciklisku procesu ar atgriezenisko saiti.

Lai mazinātu šo nelīdzsvarotību, nepieciešams:

- nostiprināt praktiskās datu apstrādes un testēšanas;
- paplašināt risinājumu plānošanas daļu (dizaina domāšana, lietotāja vajadzību analīze);
- uzsvērt integrāciju ar citām jomām un projekta cikla pilnīgu izpratni.

Analīze apliecina, ka kursa "Programmēšana I" saturs pašlaik nav līdzsvarots - dominē koda rakstīšana un gala rezultāta iegūšana, bet trūkst starpposmu: plānošanas, testēšanas un datu interpretācijas. Šī nelīdzsvarotība rada "karstos punktus", kas ietekmē skolēnu izpratni par programmēšanas procesu kā atvērtu, iteratīvu un integrētu ciklu. Satura aktualizācija būtu jāvirza uz šīs cikliskās struktūras atjaunošanu, kur programmēšana nav noslēgta darbība, bet saistīts, sistemātisks process no idejas līdz pārbaudītam risinājumam.

## 2 Pedagoģiskie un attīstības faktori, kas noteica satura pārstrukturēšanu

Programmēšanas mācību satura pārstrukturēšana veikta, balstoties uz skolēnu kognitīvās attīstības īpatnībām, motivācijas veicināšanas principiem un nepieciešamību nodrošināt jēgpilnu pāreju no algoritmiskās domāšanas uz radošu problēmu risināšanu. Satura izmaiņu mērķis ir panākt, lai programmēšana tiktu uztverta nevis kā izolēta tehniska darbība, bet kā radoša un starpdisciplināra domāšanas un darbības sistēma.

### Kognitīvā attīstība un konstruktīvisma pieeja

Vidusskolas vecuma skolēni sasniedz formāli operacionālo domāšanas līmeni (pēc Ž. Piažē teorijas), kad viņi spēj veikt loģiskus secinājumus, izprast hipotētiskus procesus un apzināti strukturēt darbības soļus. Šajā vecuma posmā programmēšana kļūst par efektīvu instrumentu abstraktās un sistēmiskās domāšanas attīstībai.

Satura izmaiņas balstās uz konstruktīvisma pedagoģiju, īpaši L. Vigotska "tuvākās attīstības zonas" konceptu. Pētījumi par programmēšanas izglītību norāda, ka efektīva mācīšanās notiek tad, kad skolēniem tiek nodrošinātas atbilstošas "sastatnes" - strukturēts atbalsts, kas pakāpeniski tiek samazināts, ļaujot pāriet no vadītas koda rakstīšanas uz patstāvīgu problēmu risināšanu [2].

### Kombinētu mācību pieeja un motivācija

Motivācija programmēšanas apgūvē cieši saistīta ar praktiskuma izjūtu un redzamu rezultātu. Latvijas Universitātes pētījums par kombinētu mācību pieeju vidusskolas kursā "*Programmēšana I*" (Irbe, 2024) secina, ka mācību procesa organizācijai jābūt elastīgākai, savukārt mācību saturam pielāgojamam un papildināmam atbilstoši skolēnu vajadzībām. Pētījums pierāda, ka starptautiski atzītu datorzinātņu kursu (piemēram, Hārvarda universitātes) digitālie mācību līdzekļi un tiešsaistes rīki ir veiksmīgi adaptējami Latvijas izglītības standartam, izmantojot kombinēto mācību pieeju [3].

Jaunajā pieejā:

- katrs temats ietver praktisku uzdevumu vai projektu, kas demonstrē programmēšanas nozīmi reālajā dzīvē;
- tiek izmantoti mākoņvides izstrādes rīki un MI atbalsta rīki kā mācību procesa atbalsts;

- formatīvā vērtēšana balstīta uz procesa novērtējumu, nevis tikai gala rezultātu.

Šāda pieeja veicina aktīvu skolēnu iesaisti mācību procesā, ļaujot programmēšanu uztvert nevis kā abstraktu teoriju, bet kā praktisku problēmu risināšanas instrumentu. Vienlaikus kombinētā mācību pieeja nodrošina lielāku elastību mācību organizācijā, ļaujot integrēt digitālos rīkus, pašvadītu mācīšanos un projektveida darbu, kas atbilst mūsdienu programmēšanas izglītības tendencēm un palīdz attīstīt skolēnu spēju patstāvīgi apgūt jaunus tehnoloģiskos risinājumus.

### **Sociāli pedagoģiskie faktori un sadarbība**

Balstoties uz sociālās mācīšanās teoriju, programmēšanas kursa saturs pārveidots, lai veicinātu kolektīvo mācīšanos un savstarpējo atbalstu.

Skolēni tiek iesaistīti:

- pāru programmēšanā, kas attīsta komunikāciju un sadarbību;
- komandas projektu darbā, izmantojot GitHub vai citas versiju kontroles sistēmas;
- koda recenzēšanā, veicinot kritisko domāšanu un spēju pamatot risinājuma izvēli.

Šāda pieeja palīdz attīstīt arī sociāli emocionālās prasmes - pacietību, argumentāciju un cieņpilnu attieksmi pret citu darbu.

### **Starppriekšmetu un digitālo kompetenču integrācija**

Kursa programmas pārstrukturēšanā īpaši uzsvērtā sasaiste ar citām jomām:

- matemātiku - algoritmu, funkciju un datu struktūru izpratnei;
- fiziku - mērījumu datu apstrādei un simulācijām;
- ekonomiku un uzņēmējdarbību - automatizācijas un procesu analīzei;
- datoriku un drošību - koda drošības un datu aizsardzības aspektiem.

Šāda integrācija veicina caurviju kompetenču attīstību, kas saskan ar DigComp 2.2 ietvaru - informācijas prasību, problēmrisināšanu, sadarbību un drošu tehnoloģiju lietošanu.

Rezultātā skolēni programmēšanu sāk uztvert kā rīku starpdisciplināru risinājumu veidošanai, nevis tikai kā tehnisku darbību.

### 3 Saistība ar DigComp un kompetenču pieeju

Programmēšanas priekšmeta satura pārkārtojums un aktualizācija ir tieši saistīta ar Eiropas digitālās kompetences ietvaru DigComp 2.2, kas definē piecas pamatkompetenču jomas, un ar kompetenču pieejas principiem, kas noteikti Latvijas vispārējās izglītības standartā. Šīs saiknes mērķis ir nodrošināt, ka mācību saturs attīsta skolēnu spēju lietot digitālās tehnoloģijas droši, radoši un jēgpilni, ne tikai apgūstot programmēšanas valodu, bet arī izprotot tās lietojuma nozīmi reālos procesos.

Saskaņā ar Eiropas Komisijas "Digital Decade" ziņojumu (2024), Latvijā tikai 45,3% iedzīvotāju ir vismaz pamatlīmeņa digitālās prasmes, kas ir zem ES vidējā rādītāja (55,6%) un būtiski atpaliek no 2030. gada mērķa (80%) [4]. Tomēr Latvija uzrāda labus rezultātus IKT absolventu īpatsvarā (5% pret ES vidējo 4,2%) [4].

Starptautiskais datorprasmju un informācijpratības pētījums ICILS 2023 uzsver nepieciešamību pāriet no informācijas patērēšanas uz tās radīšanu, analīzi un jēgpilnu izmantošanu [5]. Pētījums īpaši izceļ algoritmiskās domāšanas un programmēšanas nozīmi reālās pasaules problēmu risināšanā [5].

Kursa aktualizācija pilnībā atbilst "Izglītības attīstības pamatnostādņem 2021.-2027. gadam", kas kā vienu no prioritātēm izvirza digitālo prasmju attīstību un STEM jomu stiprināšanu [6]. Eurydice ziņojums par informātikas izglītību Eiropā (2022) norāda, ka Latvija ir viena no valstīm, kas ieviesusi programmēšanu kā atsevišķu priekšmetu vidusskolas posmā, taču izaicinājums paliek satura integrācija un skolotāju metodiskais atbalsts [7].

#### DigComp 2.2 kompetenču ietvars un tā atspoguļojums kursā "Programmēšana I"

Kursa saturs ir sasaistīts ar DigComp 2.2 kompetenču jomām, īpaši akcentējot datu pratību, digitālā satura radīšanu, drošību, sadarbību un problēmrisināšanu.

| DigComp joma                    | Kompetence                                                                     | Programmēšanas I piemērs mācību saturā                                                                                                                                        |
|---------------------------------|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Informācijas un datu pratība | Spēja atrast, analizēt, interpretēt un novērtēt digitālo informāciju un datus. | Skolēni apgūst datu lasīšanu no dažādiem avotiem, novērtē datu ticamību un pielieto tos programmās.                                                                           |
| 2. Digitālā satura radīšana     | Jauna satura veidošana un modificēšana, izmantojot digitālos rīkus.            | Skolēni raksta programmas, kas rada vai pārveido digitālu saturu (tekstu, grafiku, datus), veido prototipus un dokumentāciju, izmantojot versiju kontroles un kopdarba rīkus. |



**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](mailto:rtu@rtu.lv), [www.rtu.lv](http://www.rtu.lv)

|                                             |                                                                          |                                                                                                                                                              |
|---------------------------------------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. Drošība                                  | Datu aizsardzība, kiberdrošības un digitālās labklājības principi.       | Skolēni saņem zināšanas par droša koda rakstīšanas principiem, paroles un autentifikācijas mehānismiem, kā arī par atbildīgu mākslīgā intelekta izmantošanu. |
| 4. Digitālā komunikācija un sadarbība       | Sadarbība tiešsaistē un kopīgi digitāli projekti.                        | Programmēšanas uzdevumi tiek veikti komandās, izmantojot digitālus sadarbības rīkus, koplietojot koda rediģēšanu un savstarpēju darba novērtējuma principus. |
| 5. Problēmu risināšana ar digitāliem rīkiem | Tehnoloģiju izmantošana jaunu risinājumu radīšanai un esošo uzlabošanai. | Skolēni izmanto MI atbalsta rīkus koda analīzei, kļūdu noteikšanai un risinājuma uzlabošanai..                                                               |

Šāda strukturēta sasaiste ļauj attīstīt ne tikai programmēšanas zināšanas, bet arī plašākas digitālās prasmes, kas ir būtiskas mūsdienu darba tirgū un sabiedrībā.

## 4 Pāreja no zināšanu apguves uz kompetenču attīstību

Tradicionāli programmēšanas kursa mācību saturs bija orientēts uz zināšanu nodošanu un tehnisku prasmju apguvi - skolēni iemācījās programmēšanas valodas sintaksi un algoritmus, bet bieži nespēja sasaistīt šīs zināšanas ar praktiskiem risinājumiem. Šāda pieeja veidoja “slēgtu zināšanu ciklu”, kur programmēšana tika uztverta kā noteiktu darbību izpilde, nevis domāšanas un problēmu risināšanas process.

Jaunajā pieejā programmēšanas kurss tiek pārveidots par kompetenču attīstības instrumentu, kurā galvenais uzsvars vairs nav uz faktu atcerēšanos, bet gan uz zināšanu lietojumu praksē, analīzi un pielāgošanu dažādām situācijām. Šī pāreja tiek īstenota četros saistītos virzienos.

### 1. No reproduktīvas darbības uz radošu problēmu risināšanu

Skolēni ne tikai atkārto algoritmus, bet paši konstruē risinājumus un izvēlas piemērotākās pieejas dažādām situācijām. Piemēram, tēmas “*Risinājuma plānošana*” un “*Esošu risinājumu izmantošana*”, veicina spēju analizēt, salīdzināt un optimizēt risinājumus.

Kompetence: spēja radoši pielietot zināšanas, analizējot un uzlabojot programmas darbību.

### 2. No teorētiskām zināšanām uz praktisku pieredzi un pielietojumu

Mācību process orientēts uz praktiskiem uzdevumiem, projektiem un datu apstrādi, kas ļauj skolēnam redzēt rezultātu reālā laikā. Tematos tiek paredzēti praktiski uzdevumi, kuros skolēni izmanto datus, pārbauda risinājuma darbību un pilnveido programmu.

Attīstāmā prasme: spēja plānot, izstrādāt, pārbaudīt un pilnveidot programmētu risinājumu.

### 3. No individuālas uz sadarbībā balstītu mācīšanos

Kompetenču pieeja paredz mācīšanos sadarbībā, kur skolēni strādā pāros vai grupās, izmantojot koplietošanas rīkus. Skolēni apgūst versiju kontroli, koda dokumentēšanu un recenzēšanu, tādējādi attīstot ne tikai tehniskās, bet arī komunikācijas un sociālās prasmes.

Kompetence: spēja sadarboties digitālajā vidē, plānot komandas darbu un dalīties ar zināšanām.

#### **4. No pedagogu vadītas uz pašvadītu un adaptīvu mācīšanos**

Skolēns kļūst par aktīvu mācību procesa dalībnieku, kurš pats plāno, testē un novērtē savu darbu. Pedagoga loma mainās - no informācijas sniedzēja uz mentoru un atbalstītāju, kas palīdz skolēnam izmantot digitālos rīkus pašattīstībai.

Kompetence: pašvadīta mācīšanās, metakognitīvās prasmes, spēja analizēt un uzlabot savu sniegumu.

#### **Pārejas rezultāts un ietekme uz mācību kvalitāti**

Šī pāreja no zināšanu apguves uz kompetenču attīstību maina gan mācību satura struktūru, gan mācīšanās pieredzi.

Programmēšana kļūst par atvērtu ciklisku procesu, kurā skolēns:

- plāno risinājumu;
- izstrādā un testē kodu;
- analizē rezultātus;
- pielāgo un optimizē risinājumu jaunām situācijām.

Šāds cikls atbilst kompetenču pieejas būtībai, skolēns ne tikai zina, ko darīt, bet saprot, kāpēc un kā to darīt efektīvāk. Tas nodrošina jēgpilnu un ilgtspējīgu mācīšanos, kas veicina pašvadītu, analītisku un radošu domāšanu, kas ir būtiskas prasmes turpmākai mācībām, profesionālajai attīstībai un jēgpilnai darbībai digitālā vidē.

## 5 Zinātniskie secinājumi

### Empīriskais pamatojums

Kursa “Programmēšana I” satura aktualizācija balstīta empīriskos datos, kas iegūti, analizējot 46 pedagogu aptaujas rezultātus un Hotspots analīzi. Iegūtie dati norāda uz nepieciešamību pārstrukturēt mācību saturu, pārejot no tehniski izolētu tematu apguves uz integrētu programmēšanas procesa izpratni, īpaši stiprinot prasmes datu apstrādē, programmatūras testēšanā un risinājumu plānošanā.

### Kognitīvā atbilstība

Satura struktūra un mācību aktivitātes veidotas, ņemot vērā Ž. Piažē un L. Vigotska kognitīvās attīstības teorijas, kas norāda, ka vidusskolēnu formāli operacionālā domāšana ļauj apgūt sarežģītākas algoritmiskas struktūras, ja mācību process tiek organizēts konstruktīvisma pieejā, nodrošinot pakāpenisku atbalstu un mācību sastatnes (scaffolding) [2].

### Kombinētā mācīšanās pieeja

Latvijā veiktie pētījumi (Irbe, 2024) apliecina, ka kombinētā mācīšanās pieeja, kurā tiek apvienoti klātienē mācību procesi ar digitāliem resursiem un tiešsaistes rīkiem, ir efektīva kursa “Programmēšana I” īstenošanai. Šāda pieeja ļauj izmantot arī starptautiskus digitālos mācību resursus, tos adaptējot Latvijas izglītības standartam [3].

### Motivācijas faktori

Mācību satura aktualizācijā īpaša uzmanība pievērsta praktisku, reālos datos balstītu uzdevumu izmantošanai, kas palielina skolēnu iesaisti un motivāciju programmēšanas apgūvē. Pētījumi Latvijā norāda, ka tieši praktiskā pielietojamība un redzams rezultāts būtiski ietekmē skolēnu motivāciju apgūt programmēšanu (Milka, 2025) [1].

### Atbilstība Eiropas Savienības un Latvijas izglītības mērķiem

Programmas saturs ir saskaņots ar DigComp 2.2 digitālās kompetences ietvaru, Latvijas Izglītības attīstības pamatnostādņēm 2021.-2027. gadam, kā arī ņem vērā ICILS 2023 pētījuma secinājumus, kas uzsver nepieciešamību attīstīt skolēnu spēju ne tikai patērēt digitālo informāciju, bet arī to radīt un analizēt [4][5][6].



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

## **Programmēšanas procesa pieeja mācību organizācijā**

Programmas aktualizācijā programmēšana tiek skatīta kā pilns risinājuma izstrādes process, kas ietver secīgas darbības: problēmas analīzi, risinājuma plānošanu, programmas izstrādi, testēšanu un optimizāciju. Šāda pieeja nodrošina sistemātisku mācīšanos ar regulāru atgriezenisko saiti, attīstot skolēnu analītisko domāšanu, problēmrisināšanas prasmes un pašvadītu mācīšanos, kas atbilst Skola2030 kompetenču pieejai.

## Atsauces

- [1] Milka, Ņ. (2025). Programmēšanas izglītības programmu izvēli ietekmējošie motivācijas faktori vidusskolas skolēniem [Maģistra darbs, Latvijas Universitāte]. DSpace. <https://dspace.lu.lv/items/789c1273-b957-4f66-be56-495eac4ea837>
- [2] Belland, B. R., Glazewski, K. D., & Richardson, J. C. (2013). A Framework for Designing Scaffolds That Improve Motivation and Cognition. *Educational Psychologist*, 48(4), 343-370. <https://pmc.ncbi.nlm.nih.gov/articles/PMC3827669/>
- [3] Irbe, K. (2024). Kombinētu mācību pieeja vidusskolas kursā "Programmēšana I" [Maģistra darbs, Latvijas Universitāte]. DSpace. <https://dspace.lu.lv/items/a257df89-3a26-4f5b-b8ca-9ce0ef274bb7>
- [4] European Commission. (2024). Latvia: a snapshot of digital skills. Digital Skills and Jobs Platform. <https://digital-skills-jobs.europa.eu/en/latest/briefs/latvia-snapshot-digital-skills>
- [5] Izglītības un zinātnes ministrija. (2024, 15. oktobris). Skolēnu datorprasmju un informācijpratības pētījums ļaus novērtēt, cik veiksmīgi Latvijas pamatskolēni apguvuši digitālās prasmes. <https://www.izm.gov.lv/lv/jaunums/skolenu-datorprasmju-un-informacijpratibas-petijums-laush-novertet-cik-veiksmigi-latvijas-pamatskoleni-apguvusi-digitalas-prasmes>
- [6] Izglītības un zinātnes ministrija. (2021). Izglītības attīstības pamatnostādnes 2021.-2027. gadam "Nākotnes prasmes nākotnes sabiedrībai". <https://likumi.lv/ta/id/324332-par-izglitibas-attistibas-pamatnostadnem-2021-2027-gadam>
- [7] European Education and Culture Executive Agency. (2022). Informatics education at school in Europe. Publications Office of the European Union. [https://www.researchgate.net/publication/399713974\\_Informatics\\_education\\_at\\_school\\_in\\_Europe](https://www.researchgate.net/publication/399713974_Informatics_education_at_school_in_Europe)