

TEMATISKAIS PLĀNS

6. TEMATS

Kā īstenot risinājumu? (40 stundas)

Pilnveidot prasmi plānot, izstrādāt un realizēt programmēšanas risinājumu, izmantojot programmēšanas valodas rīkus, dokumentāciju un programmatūras izstrādes procesa pamatprincipus, kā arī veidot lietotājam saprotamu saskarni un ievērot drošības, dizaina un intelektuālā īpašuma principus.

Tematā realizētie valsts pamatizglītības standarta sasniedzamie rezultāti

- T.O.2.4.2. Izstrādā programmatūru, izmantojot iteratīvu izstrādes modeli un ievērojot tā galvenos posmus.
- T.O.2.4.3. Izstrādā programmatūras risinājumu, ievērojot programmatūras izstrādes procesa galvenos posmus un pakāpeniski pilnveidojot risinājumu.
- T.O.2.4.4. Izveido vienkāršotu programmatūras prasību specifikāciju, aprakstot izstrādājamā risinājuma mērķi, funkcionalitāti un lietotāju vajadzības.
- T.O.2.4.7. Izveido lietotājam saprotamu programmatūras saskarni, ievērojot lietojamības principus.
- T.O.2.4.10. Izmanto programmēšanas valodas dokumentāciju programmatūras izstrādes laikā.
- TV.2.2.3. Veic datu apstrādi, veidojot programmatūrai nepieciešamos grafiskos elementus.
- TV.3.1.2. Ievēro drošības prasības programmatūras risinājuma izstrādes laikā un paredz nepieciešamos pasākumus darba drošības prasību nodrošināšanai visā risinājuma izstrādes procesā.
- T.O.3.1.5. Ievēro godīga dizaina principus un paredz sava risinājuma intelektuālā īpašuma aizsardzības veidu.
- T.O.3.2.1. Risinājuma izveidē ņem vērā kultūras mantojuma un kultūras izpausmju mijiedarbību ar inovatīviem tehnoloģiskiem risinājumiem un pamato savu dizaina koncepciju un tajā ievērotos principus.
- T.O.3.2.2. Dizaina risinājuma izveidē saskata un raksturo dizaina nozīmi, lietojumu un ietekmi uz pilsētvidi, sadzīvi, sabiedrības veselību un sabiedriskajām norisēm.
- T.O.3.2.3. Salīdzina un analizē dažādu laikmetu, tostarp mūsdienu, inovāciju piemērus zinātnē un dizainā un to aktualitāti ieviešanas brīdī, apskata starpdisciplinārus izgudrojumus un modelē starpdisciplināru risinājumu, izmantojot dizaina procesā iegūtās zināšanas un prasmes.
- T.O.3.2.4. Risinājuma izstrādi un modelēšanu balsta dažādu uzņēmumu produktu ražošanas ciklu un inovāciju ieviešanas procesa izpētē.
- T.O.3.2.5. Analizē, kā zinātniskie pētījumi ietekmē tehnoloģiju attīstību un sasniegumus dažādās dzīves jomās, un izmanto iegūto informāciju sava risinājuma koncepcijas izstrādē.
- T.O.3.2.6. Skaidro dažādu ražošanas nozaru produktu iegūšanas tehnoloģiskos procesus, analizē dažādu faktoru ietekmi uz tiem un, izmantojot iegūtās zināšanas, modelē jaunus risinājumus.
- T.O.3.2.7. Izvērtē dažādu zinātnisko apakšnozaru izmantošanas iespējas un riskus sabiedrības un zinātnes attīstībā un modelē jaunus risinājumus sabiedrības vajadzību nodrošināšanai.

6. TEMATS. Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu?		40 stundas
1	Konsoles lietotnes izstrāde	8
2	Datu apstrādes risinājuma izstrāde	8
3	Apzina, izpēta un analizē problēmu/vajadzību	4
4	Izstrādā prototipu un plāno savu risinājumu	6
5	Izstrādā un testē savu risinājumu	8
6	Ievieš savu risinājumu	6

Atbilstība valsts vispārējās vidējās izglītības standartam

Apakštēmas Nr.	Apakštēmas nosaukums	Standarta punkti
1	Konsoles lietotnes izstrāde	T.O.2.4.2., T.O.2.4.3., T.O.2.4.7., T.O.2.4.10.
2	Datu apstrādes risinājuma izstrāde	T.O.2.4.2., T.O.2.4.3., TV.2.2.3., T.O.2.4.10.
3	Apzina, izpēta un analizē problēmu/vajadzību	T.O.2.4.4., T.O.3.2.3., T.O.3.2.5.
4	Izstrādā prototipu un plāno savu risinājumu	T.O.2.4.4., T.O.2.4.7., T.O.3.1.5., T.O.3.2.1.
5	Izstrādā un testē savu risinājumu	T.O.2.4.2., T.O.2.4.3., T.O.2.4.10., TV.3.1.2.
6	Ievieš savu risinājumu	T.O.2.4.3., T.O.3.1.5., TV.3.1.1., T.O.3.2.2.

Sasniedzamie rezultāti

Zināšanas

- Izstrādājot programmatūru, tiek ievērota noteikta darbību secība (T.Li.2.)
- Programmatūras izstrādē tiek izmantoti dažādi izstrādes modeļi. (T.Li.2.)
- Programmatūras prasību specifikācija palīdz skaidri definēt risinājuma mērķi, funkcionalitāti un lietotāju vajadzības. (T.Li.2.)
- Izvēloties vai izmantojot jau esošus risinājumus projektā, tiek izvērtēta to funkcionalitāte, pielāgojamība un piemērotība konkrētās problēmas risināšanai. (T.Li.2.)
- Vizuāli saprotama, estētiska un lietotājam ērta saskarne veicina lietotnes lietojamību un motivē lietotāju izmantot risinājumu. (T.Li.2.)
- Lai izveidotu lietotājiem nepieciešamu risinājumu, tiek noteikta mērķauditorija un analizētas tās vajadzības un lietošanas scenāriji. (T.Li.1., T.Li.3.)
- Izstrādājot risinājumu, tiek ievēroti godīga dizaina principi un paredzēts intelektuālā īpašuma aizsardzības veids. (T.Li.3.)
- Plānojot un ieviešot tehnoloģisku risinājumu, tiek izvērtēta tā ietekme uz lietotāja veselību, drošību un labklājību. (T.Li.3.)
- Izstrādājot programmatūru, tiek izmantota programmēšanas valodas dokumentācija un jau esoši programmatūras komponenti vai bibliotēkas. (T.Li.2.)
- Programmatūras risinājuma izstrādes laikā tiek veikta testēšana un risinājuma pilnveidošana, lai nodrošinātu tā stabilu darbību. (T.Li.2.)

Prasmes

- Izstrādā programmatūru, izmantojot iteratīvu izstrādes modeli un ievērojot tā galvenos posmus. (T.O.2.4.2.)
- Programmatūras izstrādē ievēro galvenos izstrādes posmus. (T.O.2.4.3.)
- Izveido vienkāršotu un lietotājam saprotamu programmatūras saskarni un risinājuma darbības aprakstu. (T.O.2.4.7.)
- Izmanto programmēšanas valodas dokumentāciju un jau esošus programmēšanas risinājumus problēmas atrisināšanai. (T.O.2.4.10.)
- Izveido vienkāršotu programmatūras prasību specifikāciju, nosakot risinājuma mērķi un funkcionalitāti. (T.O.2.4.4.)
- Analizē problēmu vai lietotāja vajadzību un izvērtē piemērotākos tehnoloģiskos risinājumus tās īstenošanai. (T.O.3.2.3., T.O.3.2.5.)
- Izstrādā un testē programmatūras risinājumu, identificē kļūdas un pilnveido programmas darbību. (T.O.2.4.2., T.O.2.4.3.)
- Risinājuma izstrādes laikā ievēro darba drošības prasības un paredz nepieciešamos pasākumus drošai darba videi. (TV.3.1.2.)
- Veic datu apstrādi, veidojot programmatūrai nepieciešamos elementus. (TV.2.2.3.)
- Ievēro godīga dizaina principus un paredz sava risinājuma intelektuālā īpašuma aizsardzības veidu. (T.O.3.1.5.)
- Paredz risinājuma izmantošanas ietekmi uz cilvēka veselību un labklājību un rīkojas, lai samazinātu iespējamās negatīvos riskus. (TV.3.1.1.)

Komplekss sniedzamais rezultāts

Ievērojot dizaina procesa soļus, izstrādā projektu, kas ar programmēšanas un programmvadāmo ierīču palīdzību atrisina kādu lokālu problēmu. (T.O.1.1.2., T.O.2.4.1, T.O.2.4.8, T.O.2.4.13.)

Ierādumi

- Attīsta ieradumu veidot cieņpilnas attiecības, komunicējot ar lietotāju. (Tikums – laipnība; vērtība – cilvēka cieņa)
- Attīsta ieradumu uzdrošināties pašizpausties un gūt jaunu pieredzi, lai realizētu programmēšanas projektu. (Tikums – drosme; vērtība – darbs)

- Attīsta ieradumu izprast un risināt kompleksas problēmas, lai izstrādātu programmēšanas projektu. (Tikums – gudrība; vērtība – darbs)
- Attīsta ieradumu sekmīgi darboties digitālajā vidē, digitāli veicot lietotāju aptauju un datu apstrādi. (Tikums – gudrība; vērtība – darbs).
- Attīsta ieradumu lietpratīgi un atbildīgi izmantot informācijas un komunikācijas tehnoloģiju iespējas, veidojot projekta prezentāciju un dokumentāciju. (Tikums – gudrība; vērtība – darbs).
- Attīsta ieradumu atbildīgi, ētiski un droši izmantot digitālos rīkus, datu avotus un automatizācijas risinājumus. (*Tikums – atbildība; vērtība – drošība*)

Temata saturs un tā realizācija

Temata apakšdaļa	Saturs
Konsoles lietotnes izstrāde	Individuāli vai pāros izvēlas vienu no piedāvātajiem uzdevumiem un izstrādā vienkāršu konsoles tipa programmatūras risinājumu, izmantojot iepriekš apgūtās programmēšanas pamatprasmes. Skolēni analizē uzdevuma aprakstu, nosaka risināmo problēmu un formulē programmas mērķi. Precizē paredzēto lietotāju un definē risinājuma robežas, skaidri norādot, kādas darbības programma veiks un kādi uzdevuma aspekti netiks iekļauti risinājumā. Plāno konsoles lietotnes izstrādi, formulējot vienkāršotu prasību aprakstu. Definē programmas funkcionālās prasības (piemēram, kādu ievadi lietotājs veic un kādi rezultāti tiek aprēķināti vai attēloti) un nefunkcionālās prasības, piemēram, saprotamu izvēlnes struktūru, kļūdu paziņojumus un ievades validāciju. Nosaka minimālo funkcionalitātes komplektu un definē kritērijus, pēc kuriem būs iespējams izvērtēt, vai izstrādātā programma izpilda uzdevumā noteikto mērķi. Izstrādā programmas algoritmu un plāno tās darbības plūsmu, veidojot konsoles lietotnei raksturīgu struktūru ar izvēlni, galveno ciklu un funkcijām, kas nodrošina lietotāja ievadi, datu apstrādi un rezultātu izvadi. Apraksta galvenos lietošanas scenārijus un identificē iespējamās problēmsituācijas, piemēram, nekorektu ievadi, tukšas vērtības vai neatbilstošus datu tipus. Pakāpeniski realizē programmu, izveidojot koda karkasu un sadalot programmas loģiku funkcijās. Skolēni izmanto nosacījumus, ciklus un funkcijas, lai nodrošinātu programmas darbību, testē atsevišķas funkcijas un pārbauda programmas reakciju uz dažādiem ievades datiem. Nepieciešamības gadījumā pilnveido kodu, uzlabo izvades struktūru un ievieš ievades validāciju, lai programma korekti apstrādātu arī kļūdainu lietotāja ievadi. Testē izstrādāto konsoles lietotni, izmantojot dažādus testu gadījumus, piemēram, korektus ievades datus, nekorektu ievadi un robežvērtības. Fiksē testēšanas rezultātus, analizē konstatētās kļūdas un veic nepieciešamos labojumus programmā, izvērtējot, vai programma darbojas atbilstoši iepriekš definētajām prasībām. Noslēgumā dokumentē izstrādāto risinājumu, sagatavojot īsu lietotāja instrukciju un tehnisku aprakstu par programmas struktūru, funkcijām un darbības principiem. Demonstrē izstrādāto konsoles lietotni, skaidro programmas darbību, prezentē testēšanas rezultātus un pamato, kā risinājums atbilst uzdevumā definētajām prasībām. Tādējādi skolēni praktiski īsteno pilnu neliela programmatūras risinājuma izstrādes ciklu, izstrādājot konkrētu konsoles tipa programmatūras risinājumu.
Datu apstrādes risinājuma izstrāde	Individuāli vai pāros izvēlas vienu no piedāvātajiem uzdevumiem un izstrādā datu apstrādes risinājumu, izmantojot sarakstus un divdimensiju datu struktūras. Skolēni analizē uzdevuma aprakstu, nosaka risināmo problēmu, formulē programmas mērķi un paredzēto lietotāju, kā arī definē risinājuma robežas, skaidri norādot, ko programma darīs un ko noteiktajā izstrādes laikā neīstēnos. Aplūko un salīdzina vismaz vienu līdzīgu risinājumu, piemēram, citu programmu, lietotni vai <i>Excel</i> veidni, analizējot tā funkcionalitāti, lietojamību un ierobežojumus. Plāno datu apstrādes risinājuma izstrādi, formulējot vienkāršotu prasību aprakstu. Definē programmas funkcionālās prasības, piemēram, datu ievadi, glabāšanu, filtrēšanu, aprēķinus vai kopsavilkuma veidošanu, kā arī nefunkcionālās prasības, piemēram, saprotamu izvēlni, kļūdu paziņojumus un ievades validāciju. Nosaka minimālo funkcionalitātes komplektu un definē kritērijus, pēc kuriem būs iespējams izvērtēt, vai

	izstrādātais risinājums sasniedz uzdevumā noteikto mērķi. Vienlaikus plāno arī datu modeli, nosakot, kā viens ieraksts tiks glabāts sarakstā un kā visi dati kopā tiks organizēti divdimensiju struktūrā. Izstrādā programmas algoritmu un plāno tās darbības plūsmu, veidojot konsoles lietotnei raksturīgu struktūru ar izvēlni, galveno ciklu un funkcijām, kas nodrošina datu ievadi, glabāšanu, apstrādi un rezultātu izvadi. Apraksta galvenos lietošanas scenārijus un identificē iespējamās problēmsituācijas, piemēram, nekorektu ievadi, tukšas vērtības, nepareizu datu tipu vai robežvērtību gadījumus. Pakāpeniski realizē programmu, izveidojot koda karkasu un sadalot programmas loģiku funkcijās. Skolēni izmanto sarakstus un divdimensiju datu struktūras, lai uzkrātu un apstrādātu datus, realizē minimālās funkcionālās prasības, testē atsevišķas funkcijas un pārbauda programmas reakciju uz dažādiem ievades datiem. Nepieciešamības gadījumā pilnveido kodu, uzlabo izvades struktūru, samazina koda dublēšanos un ievieš ievades validāciju, lai programma korekti apstrādātu arī kļūdainu lietotāja ievadi. Testē izstrādāto datu apstrādes risinājumu, izmantojot dažādus testu gadījumus, piemēram, korektus ievades datus, nekorektu ievadi un robežvērtības. Fiksē testēšanas rezultātus, analizē konstatētās kļūdas, veic nepieciešamos labojumus un atkārtoti pārbauda programmu, izvērtējot, vai tā darbojas atbilstoši iepriekš definētajām prasībām. Noslēgumā dokumentē izstrādāto risinājumu, sagatavojot īsu lietotāja instrukciju un tehnisku aprakstu par datu struktūru, funkcijām un programmas darbības principiem. Publicē pirmkodu un risinājuma aprakstu tīmeklī, sagatavo īsu produkta raksturojumu un prezentē izstrādāto risinājumu, skaidrojot programmas darbību, datu struktūru izvēli, testēšanas rezultātus un salīdzinājumu ar līdzīgu risinājumu. Tādējādi skolēni praktiski īsteno pilnu neliela datu apstrādes risinājuma izstrādes ciklu, izmantojot sarakstus un divdimensiju datu struktūras.
Apzina, izpēta un analizē problēmu/vajadzību	Individuāli vai grupās identificē kādu lokālu problēmu vai lietotāju vajadzību, kuru varētu risināt, izstrādājot digitālu programmatūras risinājumu. Piemēram, tā var būt klases informācijas vai paziņojumu apmaiņas sistēma, pasākumu vai konsultāciju pieraksta sistēma, skolas bibliotēkas grāmatu rezervēšanas sistēma, sporta vai interešu pulciņu dalībnieku reģistrācijas sistēma, mājasdarbu vai projektu uzdevumu pārvaldības rīks vai inventāra rezervēšanas sistēma. Aplūko un salīdzina dažādu līdzīgu risinājumu – programmu, vietņu vai lietotņu – funkcionalitāti, analizējot to priekšrocības, ierobežojumus un lietojamību. Izvēlas un lieto mērķim atbilstošu lietotāju izpētes metodi, lai iegūtu datus par risinājuma nepieciešamību, potenciālo pieprasījumu, vēlamu izskatu un funkcionalitāti. Var izmantot, piemēram, aptauju, interviju, novērojumu vai lietotāju scenāriju analīzi. Apkopo iegūtos rezultātus un, vadoties pēc pieejamajām tehniskajām iespējām, sagatavo argumentētu izvērtējumu par pieņemamām un atmetamām idejām vai prasībām. Veido programmas vai digitālā risinājuma aprakstu, apkopo lietotāju prasības un sagatavo vienkāršotu tehnisko specifikāciju, kā arī nosaka galvenos kritērijus, pēc kuriem būs iespējams novērtēt, vai izstrādātais risinājums sasniedz paredzēto mērķi. Aktualizē apgūtos programmatūras izstrādes pamatprincipus un aplūko piemērus, izveidojot vienkāršu prezentāciju vai tīmekļa lapu ar informāciju par topošo projektu. Analizē drošības riskus, kas var rasties, izstrādājot vairāklietotāju tīmekļa programmu, piemēram, saistībā ar lietotāju datu aizsardzību, piekļuves tiesību pārvaldību vai lietotāju autentifikāciju, un diskutē par drošības jautājumu nozīmi projekta plānošanā. Vienojas par minimālajām drošības prasībām projekta veidošanas un ieviešanas gaitā.

	Analizē piemērus tam, kā līdzīgu digitālo risinājumu izmantošana ietekmē lietotājus, tostarp to fizisko un garīgo labsajūtu, drošību un privātumu. Diskutē par ētisku risinājumu izveidi un sava risinājuma izstrādē sagatavo īsu aprakstu par tā iespējamo ietekmi uz lietotājiem un vidi.
Izstrādā prototipu un plāno savu risinājumu	Balstoties uz izstrādāto projekta koncepciju, grupas dalībnieki vienojas par darba sadali un izplāno risinājuma izstrādes procesu, nosakot katra dalībnieka atbildību un veicamos uzdevumus. Skolēni definē galvenos projekta posmus, nosaka prioritātes un vienojas par darba organizāciju, lai nodrošinātu pakāpenisku risinājuma izstrādi un testēšanu. Plānojot projekta īstenošanu, skolēni strukturē izstrādes soļus un veido vienkāršu risinājuma prototipu. Pāros vai nelielās grupās skolēni izstrādā risinājuma darbības aprakstu vai blokshēmu, kas attēlo galvenos sistēmas komponentus un to savstarpējo darbību. Tiek apspriesta datu plūsma starp sistēmas daļām, lietotāja ievade, datu apstrāde un rezultātu attēlošana. Skolēni vienojas par sistēmas komponentu savstarpējās komunikācijas principiem, piemēram, kā tiks nodoti dati starp klienta un servera daļu vai starp dažādām programmas funkcijām, kā arī kādā datu formātā informācija tiks saglabāta vai pārsūtīta. Tiek definēta ievades datu struktūra, rezultātu attēlošanas veids un iespējamās kļūdu apstrādes situācijas. Praktiskajā darbībā skolēni izveido vienkāršu funkcionālu prototipu, kurā tiek realizētas projekta pamatfunkcijas. Piemēram, viens grupas dalībnieks izstrādā datu apstrādes vai servera daļu, savukārt otrs izstrādā lietotāja saskarni vai klienta daļu. Skolēni pārbauda, vai programmas komponenti savstarpēji korekti sadarbojas, vai dati tiek pareizi nosūtīti, apstrādāti un attēloti lietotājam. Izstrādes gaitā skolēni testē risinājuma darbību dažādās situācijās, piemēram, pārbaudot, kā programma reaģē uz nepareiziem ievades datiem vai sistēmas kļūdām. Tiek analizēti iespējamie uzlabojumi un veikti nepieciešamie labojumi prototipā. Darba noslēgumā skolēni izvērtē izveidoto prototipu un plānoto risinājumu, analizējot, vai izstrādātais modelis atbilst sākotnēji definētajai problēmai un lietotāju vajadzībām, kā arī nosaka turpmākos soļus risinājuma pilnveidei.
Izstrādā un testē savu risinājumu	Skolēni iepazīstas ar dažādiem programmatūras izstrādes modeļiem, piemēram, ūdenskrituma modeli, V-modeļi, iteratīvo un pakāpenisko izstrādes pieeju. Analizē šo modeļu darbības principus un salīdzina to piemērotību dažādās situācijās, diskutējot par to, kā izstrādes pieeja ietekmē projekta plānošanu, testēšanu un rezultāta kvalitāti. Balstoties uz izvēlēto pieeju, skolēni savu projektu izstrādā iteratīvi, pakāpeniski pilnveidojot risinājumu un dokumentējot izstrādes procesu. Atbilstoši izstrādātajam plānam skolēni veido programmatūras risinājumu, nepieciešamības gadījumā izmantojot programmēšanas valodas dokumentāciju, piemērus un citus pieejamos resursus. Izstrādes gaitā skolēni regulāri pārbauda izveidotā risinājuma darbību, testējot programmu dažādās situācijās un pārlicinoties, ka tā darbojas atbilstoši definētajām prasībām. Skolēni demonstrē starprezultātus citiem skolēniem vai skolotājam, iegūst atgriezenisko saiti un analizē, kā risinājums atbilst sākotnējam plānam un lietotāju vajadzībām. Pamatojoties uz saņemtajiem ieteikumiem un testēšanas rezultātiem, skolēni pilnveido izstrādāto risinājumu, nepieciešamības gadījumā precizē projekta plānu, uzlabo funkcionalitāti vai novērš konstatētās kļūdas. Ja tiek identificētas jaunas prasības vai uzlabojumu iespējas, skolēni atkārtoti veic izstrādes, testēšanas un izvērtēšanas ciklu. Darba noslēgumā skolēni demonstrē gatavo risinājumu, prezentē tā funkcionalitāti un skaidro projekta izstrādes gaitu, tostarp analizējot izstrādes laikā sastaptās grūtības, pieņemtos risinājumus un iegūto pieredzi programmatūras izstrādes procesā.

Ievieš savu risinājumu	Skolēni izstrādā sava izveidotā risinājuma dokumentāciju gala lietotājam, sagatavojot saprotamu aprakstu par programmas izmantošanu. Atkarībā no projekta veida skolēni var izveidot lietotāja ceļvedi, īsu lietošanas instrukciju, reklāmas bukletu vai informatīvu atgādni, kurā skaidrots risinājuma mērķis, galvenās funkcijas un lietošanas soļi. Dokumentācijas izstrādes procesā skolēni pēc nepieciešamības apstrādā attēlu, audio vai video datnes, lai uzskatāmi demonstrētu programmas darbību. Pēc risinājuma sagatavošanas skolēni publicē projektu publiski pieejamā tīmekļa vidē, izmantojot kādu no mākoņpakalpojumiem vai izstrādes platformām (piemēram, <i>Replit</i>, <i>GitHub</i> vai līdzīgus risinājumus). Skolēni iepazīstas arī ar citiem programmatūras publicēšanas veidiem un analizē to priekšrocības un ierobežojumus. Skolēni testē gan savu, gan citu skolēnu izstrādātos projektus, pārbaudot to darbību un vērtējot atbilstību iepriekš definētajiem projekta kritērijiem. Testēšanas procesā tiek identificētas iespējamās kļūdas vai nepilnības, kuras skolēni analizē un labo savā projektā, balstoties uz citu lietotāju vai skolotāja sniegto atgriezenisko saiti. Skolēni iepazīstas ar intelektuālā īpašuma aizsardzības principiem digitālu risinājumu izstrādē un izvēlas piemērotu veidu sava projekta aizsardzībai, piemēram, izvēloties atbilstošu licenci vai nosakot projekta izmantošanas nosacījumus. Skolēni pamato savu izvēli un izprot autortiesību nozīmi programmatūras izstrādē. Noslēgumā skolēni prezentē savu izstrādāto risinājumu potenciālajiem lietotājiem vai klasesbiedriem, demonstrējot programmas funkcionalitāti, skaidrojot tās izmantošanas iespējas un raksturojot projekta izstrādes rezultātus.
--------------------------------------	--