



Programmēšana II

**Padziļinātā kursa programmas paraugs
vispārējai vidējai izglītībai**

Valsts izglītības satura centrs | ESF projekts Nr. 8.3.1.1/16/1/002
Kompetenču pieeja mācību saturā

Programmēšana II

Padziļinātā kursa programmas paraugs vispārējai vidējai izglītībai

Padziļinātā kursa programmas paraugs ir izstrādāts Eiropas Sociālā fonda projektā "Kompetenču pieeja mācību saturā" (turpmāk – Projekts).

Mācību satura izstrādi pirmsskolas, pamatizglītības un vispārējās vidējās izglītības pakāpē Projektā vadija **Dace Namsone** un **Zane Oliņa**.

Padziļinātā kursa programmas parauga izstrādi un sagatavošanu publicēšanai Projektā vadija **Madara Kosolapova** un **Pāvels Pestovs**.

Padziļinātā kursa programmas paraugu izstrādāja **Kaspars Antonevičs, Ance Kancere, Edgars Kupčs, Kristaps Muižnieks**.

Padziļinātā kursa programmas paraugu izvērtēja ārējie eksperti: mācību satura recenzente **Iveta Milta** un zinātniskais recenzents **Kaspars Kiris**.

Projekts izsaka pateicību visām Latvijas izglītības iestādēm, kas piedalījās mācību satura aprobācijā.

ISBN 978-9934-24-009-6

Saturs

Ievads	4	5. Programmatūras izstrāde	23
Mērķis un uzdevumi	5	6. Produkta izveide grupā, izmantojot visus	
Mācību saturs	5	programmatūras modeļa izstrādes posmus	25
Vērtēšanas saturs, mācību sasniegumu vērtēšanas			
formas un metodiskie paņēmieni	7	Pielikumi	27
Sasniedzamo rezultātu veidi, vērtēšanas formas		1. Kursu programmu paraugos lietotie kodi	27
un metodiskie paņēmieni	7	2. Programmēšana II padziļinātajā kursā	
Kursa vērtēšanas saturs	7	plānotie skolēnam sasniedzamie rezultāti	28
Kursa apguves prasības	10	3. Plānotie skolēnam sasniedzamie rezultāti caurviju	
Ieteikumi mācību darba organizācijai	11	prasmēs, beidzot vispārējās vidējās izglītības pakāpi	30
Mācību satura apguves norise	12	4. Vērtēšanas uzdevumu piemēri	32
Kursa satura pārskats	12	1. uzdevums	32
Temata ietvara struktūras paraugs	13	2. uzdevums	34
Izvērstis kursa saturs	14	3. uzdevums	35
1. Objektorientēta programmēšana un		4. uzdevums	36
ārējās bibliotēkas	14	5. uzdevums	37
2. Datortīkla, servera un drošas datubāzes		6. uzdevums	38
izveide un konfigurācija	16	7. uzdevums	39
3. Datu struktūras, programsaskarne (API),		8. uzdevums	40
mašīnmācīšanās	19	9. uzdevums	45
4. Problēmas analīze, programmatūras		10. uzdevums	47
specifikācija un darba plānošana	21	5. Mācību satura apguvei izmantojamie mācību līdzekļi un resursi	48

levars

Kursa programmas struktūra

Programmēšana II padziļinātā kursa programmas (turpmāk – programma) paraugs ir veidots, lai palīdzētu skolotājiem īstenot Ministru kabineta 2019. gada 3. septembra noteikumos Nr. 416 “Noteikumi par valsts vispārējās vidējās izglītības standartu un vispārējās vidējās izglītības programmu paraugiem” (turpmāk – standarts) noteiktos plānotos skolēnam sasniedzamos rezultātus tehnoloģiju mācību jomā augstākajā mācību satura apguves līmenī.

Programmā iekļauti:

- kursa mērķis un uzdevumi;
- mācību saturs;
- vērtēšanas saturs, mācību sasniegumu vērtēšanas formas un metodiskie paņēmieni;
- ieteikumi mācību darba organizācijai;
- mācību satura apguves norise;
- vērtēšanas uzdevumu piemēri.

Mācību satura apguvei izmantojamo mācību līdzekļu un resursu apkopojošs uzskaitījums pievienots 5. pielikumā. Mācību saturs programmā ir veidots atbilstoši standartā noteiktajiem tehnoloģiju mācību jomas plānotajiem skolēnam sasniedzamajiem rezultātiem augstākajā mācību satura apguves līmenī.

Programma veidota, paredzot, ka kursa apguvei vidējās izglītības pakāpē tiks atvēlētas 210 mācību stundas. Taču skolai ir iespējams mainīt mācību stundu skaitu kursā, to nesamazinot vairāk par 15 %. Ieteikumus mācību darba organizācijai skatīt programmas sadaļā “Ieteikumi mācību darba organizācijai”. Programmas paraugam ir ieteikuma raksturs. Skolotāji var izmantot šo programmu vai arī izstrādāt savu programmu.

Mācību satura un pieejas akcenti

Vispārējās vidējās izglītības satura īstenošanas mērķis ir lietpratīgs skolēns, kurš apzinās savas personiskās spējas un intereses mērķtiecīgai personiskās un profesionālās nākotnes veidošanai, kurš ciena sevi un citus, padziļina zināšanas, izpratni, prasmes un turpina nostiprināt vērtības un tikumus atbilstoši saviem nākotnes mērķiem, atbildīgi, inovatīvi un produktīvi darbojas paša, ģimenes, labklājīgas un ilgtspējīgas Latvijas valsts un pasaules veidošanā.

Vidējās izglītības pakāpes loma ir dot iespēju jauniešiem mācīties, iedziļinoties atbilstoši viņu interesēm un nākotnes mērķiem, padziļinot un vispārinot pamatzglītībā apgūto (10./11. klase) un mācoties dziļāk, šaurākā mācību jomu lokā (11./12. klase).

Mācību jomā plānotos rezultātus augstākajā mācību satura apguves līmenī skolēns apgūst padziļinātajā kursā. Tā mērķis ir sniegt zināšanas, izpratni un veidot prasmes, apziņāti, atbildīgi, radoši un patstāvīgi pārraugot savu izzināto darbību, risinot problēmas nepazīstamās, sarežģītās situācijās, veidojot dziļu konceptuālu izpratni mācību jomā, saskatot starpdisciplināras likumsakarības un mācoties patstāvīgi plānot, īstenot, uzraudzīt un izvērtēt produkta radīšanas procesu.

Apgūstot Programmēšana II kursa mācību saturu augstākajā mācību satura apguves līmenī, skolēns apgūst objektorientētas programmēšanas valodas pamatprincipus un bibliotēku pielietojumu un spēj izstrādāt konkrētu programmatūras risinājumu. Skolēns spēj izveidot un konfigurēt atvērtu vai aizsargātu daudzlietotāju lokālu tīklu, izveidot vienkāršu serveri programmatūras izstrādātāja vajadzībām, izplānot un realizēt datubāzes pielietojumu programmaproduktā. Skolēns pašvadītas izpētes rezultātā prot saskatīt automatizācijas iespējas dažādos ikdienas darba procesos un spēj izstrādāt programmvadāmu risinājumu grupā (t. sk. sadalot darba pienākumus un iekārtojot sadarbības vidi), īstenojot visus programmatūras izstrādes dzīves cikla posmus. Atbilstoši sevis izvēlētajai problēmsituācijai projekta izstrādes laikā skolēns izvēlas un izmanto atvērtā koda bibliotēkas un dažādas datu struktūras, pielieto un izstrādā API risinājumu, kā arī prot pielietot mašīnmācīšanās algoritmu darbības principus programmvadāmu risinājumu izstrādē.

Mērķis un uzdevumi

Programmēšana II kursa apguves mērķis un uzdevumi skolēnam ir:

- 1) izmantojot objektorientētas programmēšanas valodas pamatprincipus un bibliotēkas, spēt izstrādāt konkrētu programmatūras risinājumu;
- 2) spēt izveidot un konfigurēt atvērtu vai aizsargātu daudzlietotāju lokālu tīklu, izveidot vienkāršu serveri programmatūras izstrādātāja vajadzībām, izplānot un realizēt datubāzes pielietojumu programmaproduktā;
- 3) izmantojot atvērtā koda bibliotēkas un dažādas datu struktūras, spēt izstrādāt API risinājumu, kā arī prast pielietot mašīnmācīšanās algoritmu darbības principus programmvadāmu risinājumu izstrādē;
- 4) izpētes rezultātā prast saskatīt automatizācijas iespējas dažādos ikdienas darba procesos, definēt darba uzdevumu (lai izstrādātu programmatūras risinājumu) un veidot programmatūras prasību specifikāciju, pēc kuras ir iespējams izstrādāt atbilstošu programmatūru;
- 5) strukturējot programmas kodu, spēt izstrādāt programmatūru, izmantojot programmatūras izstrādes labās prakses principus;
- 6) prast izstrādāt programmvadāmu risinājumu grupā (t. sk. sadalot darba pienākumus un iekārtojot sadarbības vidi), īstenojot visus programmatūras izstrādes dzīves cikla posmus.

Mācību saturs

Vidējās izglītības pakāpē mācību saturs ir izstrādāts, fokusējoties uz skolēnam būtiskāko, lai veidotos lietpratība (kompetence) kā komplekss skolēna mācīšanās rezultāts ilgākā periodā. Mācību saturs ir organizēts saskaņā ar mācību satura būtiskākajiem pamatjēdzieniem jeb lielajām idejām, kas skolēnam jāapgūst, lai veidotos vienota izpratne par apkārtējo pasauli un sevi tajā. Lielās idejas veido obligātā mācību satura strukturālo ietvaru. Tām atbilstoši aprakstītas prasības mācību satura apguvei jeb plānotie skolēnam sasniedzamie rezultāti, pabeidzot noteiktu izglītības pakāpi.

Tehnoloģiju mācību jomas lielās idejas, par kurām skolēns veido izpratni arī mācību kursā Programmēšana II, atbild uz jautājumiem “Kā?”, “Ar ko?” un “Kāpēc?” un tiek aktualizētas katrā tematā, kā arī viena otru papildina.

- **Kā?** – 1. lielā ideja “Dizaina risinājumi (produktu un informācijas dizains, vides risinājumi) tiek radīti dizaina procesā” parāda, ka risinājumi tiek izstrādāti, sekojot konkrētiem dizaina procesa/risinājuma izstrādes soļiem. Pamatojoties uz to, ka kursa Programmēšana II apguves priekšnosacījums ir apgūts pamatkurss Programmēšana I, padziļinātajā kursā tiek izmantotas iepriekš apgūtās prasmes dizaina procesa vadībā, lietotāju izpētē, vajadzību apzināšanā un ideju radīšanā, prototipēšanā. Risinājuma radīšanā ņem vērā izmantoto tehnoloģiju ietekmi uz to izmantotāju, kā arī radīto risinājumu ietekmi uz sabiedrību un vidi.
- **Ar ko?** – 2. lielā ideja “Atbilstošu un drošu materiālu un tehnoloģiju izvēle, to prasmīga izmantošana dod iespēju radīt labākus dizaina risinājumus (produktu un informācijas dizaina, vides risinājumus)” aptver to, ka katra risinājuma izstrādē tiek izmantoti dažādi programmēšanas paņēmieni un dažādi rīki, kas ļauj panākt konkrētu rezultātu. Skolēns nostiprina un apgūst jaunas zināšanas un prasmes objektorientētajā programmēšanas vidē, datortīklu izveidē un konfigurēšanā, programsaskarnes un bibliotēku pielietojumā, iepazīstas ar mašīnmācīšanās pamatprincipiem. Padziļinātajā kursā būtisks uzsvars tiek likts uz to, ka skolēns tiek aicināts izsvērt, kādas ir iespējamās alternatīvas risinājuma izstrādei, un analizēt pieejamo informāciju par jau esošiem risinājumiem, paņēmieniem un radīt idejas, kā esošo izmantot uzlabotu un jaunu risinājumu izveidei.
- **Kāpēc?** – 3. lielā ideja “Dizaina risinājumus (produktu un informācijas dizaina, vides risinājumus) un programmatūru lieto un rada atbilstoši konkrēta lietotāja un sabiedrības

vajadzībām, vēlmēm un iespējām” dod kontekstu veiktajām darbībām un apskata to, kādos nolūkos un no kā iedvesmojoties tiek veidoti jauni risinājumi, vai ir jēgpilni tādus radīt. Kursā Programmēšana II zem 3. lielās idejas ir apkopoti tādi būtiski aspekti kā kriptogrāfijas izmantošana un atvērto kodu licences un to atšķirības. Jāņem vērā arī pamatkursā apgūtais – risinājuma radīšanā ņemt vērā darba drošības un ergonomikas prasības un pārdomāti radīt risinājumus, kas nerada kaitējumu sabiedrībai un videi.

Standartā plānotie skolēnam sasniedzamie rezultāti mācību jomā un no tiem atvasinātie sasniedzamie rezultāti padziļinātā kursa programmā ir kompleksi – galarezultāts veidojas darbībā, kura ietver gan mācību jomas zināšanas, izpratni un prasmes, gan vispārīgās jeb caurviju prasmes (turpmāk – caurviju prasmes), gan vērtībās balstītus ieradumus. Katra kursa skolotāja viens no uzdevumiem ir tos attīstīt.

Caurviju prasņu apguve un izmantošana ikdienā ir nozīmīgs priekšnoteikums dziļākas izpratnes veidošanai kursā. Vingrinoties izmantot caurviju prasmes kursam specifiskos veidos un situācijās, skolēns vienlaikus ir ieguvis vispārīgas prasmes, kuras varēs izmantot visu dzīvi. Standartā plānotie skolēnam sasniedzamie rezultāti caurviju prasmēs, beidzot vispārējās vidējās izglītības pakāpi, iekļauti programmas 3. pielikumā.

Programmēšana II kursā īpaši tiek attīstītas tādas caurviju prasmes kā:

- **Digitālā pratība** – izstrādājot programmēšanas risinājumus, to dokumentāciju un nodrošinot komunikāciju ar atbilstošiem līdzekļiem.
- **Kritiskā domāšana un problēmrisināšana** – plānojot un apzinot lietotāju vajadzības, ieviešot inovācijas, plānojot un vadot programmēšanas risinājumu projektu izstrādes procesu.
- **Pašvadīta mācīšanās** – pastāvīgi organizējot darba procesu, izstrādi, ieviešanu, apzinoties vajadzību iesaistīt papildu resursus (izstrādei, konsultācijām).
- **Jaunrade un uzņēmējspēja** – radot jaunus programmēšanas risinājumus atbilstoši iepriekš identificētām lietotāju vajadzībām.

Vērtēšanas saturs, mācību sasniegumu vērtēšanas formas un metodiskie paņēmieni

Sasniedzamo rezultātu veidi, vērtēšanas formas un metodiskie paņēmieni

Programmas ietvaros paredzēti četru veidu plānotie skolēnam sasniedzamie rezultāti: zināšanas un izpratne, prasmes, vērtībās balstīti ieradumi un zināšanu, izpratnes, prasmju un ieradumu kombinācijas. Katram sasniedzamo rezultātu veidam ir norādītas būtiskas sasniedzamo rezultātu grupas, kuras apkopo standartā noteikto mācību saturu.

Zināšanu un izpratnes apguve attiecas uz standartā plānotajiem skolēnam sasniedzamajiem rezultātiem, kuri parasti sākas ar darbības vārdiem “skaidro”, “pamato” u. c. Plānoto skolēnam sasniedzamo rezultātu apguvi skolēns parāda, piemēram, skaidrojot jēdzienus, algoritmus, piedaloties sarunās un diskusijās.

Prasmju grupas atspoguļo būtiskas priekšmeta specifiskās prasmes, domāšanas un caurviju prasmes. Prasmju apguvi skolēns demonstrē darbībā, piemēram, modelē, aprēķina, analītiski spriež, lieto priekšmeta specifisko valodu.

Ieradumus, kas balstīti vērtībās, skolēns demonstrē darbībā; tos vērtē, novērojot skolēna darbību ilgākā laikposmā, īpaši situācijās, kuras ietver izvēles iespējas.

Zināšanu, izpratnes, prasmju un ieradumu kombinācijas, kuras ir raksturīgas un būtiskas šī kursa mācību satura apgūvē, skolēns demonstrē darbībā, jaunajās, to skaitā reālajās, dzīves situācijās, risinot problēmas. Katra padziļinātā kursa ietvaros ir norādītas raksturīgas problēmas, piemēram, pētniecība, mākslinieciskā jaunrade, kurās skolēns definē problēmu vai iespēju, formulē un izvēlas risinājumu, plāno un rīkojas, pārbauda un izvērtē risinājumu.

Skolotājs atbilstoši sasniedzamajam rezultātam izvēlas uzdevumu un vērtēšanas formu (mutiski, rakstiski, praktiski vai kombinēti). Būtiska uzdevumu daļa ir vērtēšanas kritēriji, saskaņā ar kuriem iespējams izvērtēt snieguma kvalitāti. Ja skolēns var demonstrēt sniegumu dažādās kvalitātes gradācijās, tad ir svarīgi veidot snieguma aprakstu attiecībā pret būtiskiem kritērijiem. Kritēriju izstrādē un vērtēšanā var iesaistīt skolēnus, lai pilnveidotu viņu pašvadītas mācīšanās prasmes.

Kursa vērtēšanas saturs

Programmēšana II kursa detalizēts sasniedzamo rezultātu grupu uzskaitījums ir apkopots tabulā. Programmas ietvaros atbilstoši sasniedzamo rezultātu grupām ir izstrādāti uzdevumu piemēri ar vērtēšanas kritērijiem. Atsevišķi uzdevumi ietver vairākas sasniedzamo rezultātu grupas. Šo uzdevumu mērķis ir atspoguļot zināšanas un izpratni, prasmes, ieradumus un zināšanu, izpratnes, prasmju un ieradumu kombinācijas, kuras skolēns ir apguvis kursa ietvaros.

Sasniedzamo rezultātu veids	Sasniedzamo rezultātu grupa	Vērtēšanas uzdevuma piemērs*
Zināšanas un izpratne	Skaidro objektorientētās programmēšanas, mašīnmācīšanās un datortīkla darbības pamatprincipus, API pielietojumu un nepieciešamību, dizaina procesa norisi un programmatūras izstrādes dzīves cikla posmus. Skaidro atvērtā koda bibliotēku pielietojumu un iespējamās problēmas. Salīdzina atvērto kodu licences un pamato piemērotību savam programmaproduktam.	2., 3. uzdevums * Skolēns demonstrē sniegumu šajās SR grupās integrēti ar citām SR grupām.
	Pamato servera uzturēšanas drošības tehnikas noteikumus, izvēlētās programmēšanas valodas, vides un projekta izstrādes pieejas piemērotību izvirzītajam mērķim.	9. uzdevums * Skolēns demonstrē sniegumu šajās SR grupās integrēti ar citām SR grupām.
Prasmju grupa	Analizē un plāno: <ul style="list-style-type: none"> • analizē un vērtē dažādus ikdienas procesus; • plāno atbilstošus programmaprodukta risinājumus, t. sk. izmantojot datubāzes, bibliotēku piesaisti u. tml.; • izvēlas piemērotu programmatūras izstrādes modeli, programmēšanas valodu un programmatūras izstrādes vidi. 	4., 6., 7., 8. uzdevums
	Informācijas pratība: izvērtējot avotus, iegūst kvantitatīvus un kvalitatīvus datus. Tos analizē, izvērtē un pielieto programmrisinājumā, piemēram, lielo datu apstrādē, bibliotēku piesaistē u. tml. Izmanto programmēšanas valodu un to bibliotēku dokumentāciju un palīdzības sistēmu, lai patstāvīgi apgūtu citas to piedāvātās iespējas.	* Skolēns demonstrē sniegumu šajās SR grupās integrēti ar citām SR grupām.
	Pašvadītā mācīšanās: plāno, uzrauga, izvērtē savu mācīšanos.	* Skolēns demonstrē sniegumu šajās SR grupās integrēti ar citām SR grupām.
	Veido, izstrādā: <ul style="list-style-type: none"> • veido programmas kādā no programmēšanas valodām, t. sk. objektorientētajā valodā; • izveido un lieto datubāzi. Šifrē datus; • izveido un konfigurē atvērtu vai aizsargātu daudzlietotāju lokālu tīklu; • izveido tīmekļa serveri programmaprodukta izstrādātāja vajadzībām; • izmanto dažādas datu struktūras un ar tiem saistītos pamatalgoritmus; • izstrādā programmsaskarni; • lieto mašīnmācīšanos programmvadāmā risinājumā; • veido programmatūras prasību specifikāciju; • izstrādā programmatūras izvēršanas un uzturēšanas plānu, lietotāja ceļvedi. 	1., 3., 4., 5., 6., 7., 9., 10. uzdevums

Sasniedzamo rezultātu veids	Sasniedzamo rezultātu grupa	Vērtēšanas uzdevuma piemērs*
Prasmju grupa	Apvieno, piesaista, pievieno: <ul style="list-style-type: none"> piesaista standartbibliotēku funkcionalitāti; meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai. 	6., 7., 10. uzdevums
	Salīdzina programmatūras prasību specifiku ar produkta funkcionalitāti (akcepttestēšana) un veic atklāšanu.	10. uzdevums
	Demonstrē izstrādāto programmatūras risinājumu (piemēram, klasē).	10. uzdevums * Skolēns demonstrē sniegumu šajās SR grupās integrēti ar citām SR grupām
Ieradumi	Izmanto labās prakses principus programmas koda pierakstā un strukturēšanā. Ievēro ergonomikas principus darba procesā. Izvērtē un ņem vērā izmantojamo resursu un paša radīto produktu drošības riskus. Koplieto un izmanto savu un citu radītu saturu, prasmīgi pārvalda savu digitālo identitāti, efektīvi un droši komunicē un sadarbojas ar citiem digitālajā vidē.	* Skolēns demonstrē sniegumu šajās SR grupās integrēti ar citām SR grupām
Zināšanu, izpratnes, prasmju un ieradumu kombinācijas	Problēmrisināšana: izstrādā izvēlētajai mērķauditorijai nepieciešamu programmaproduktu, veicot izpēti un definējot risināmo problēmu, izvēloties piemērotāko risinājumu, plānojot resursus un veicot akcepttestēšanu. Projekta vadība: sadala darba pienākumus grupā, plāno nepieciešamos resursus un secīgi veicamos darba soļus, efektīvi komunicē un sadarbojas grupas ietvaros, pārbauda programmaprodukta izstrādes procesu.	8., 9., 10. uzdevums

* Vērtēšanas uzdevumu piemērus skatīt 4. pielikumā.

Kursa apguves prasības

Programmā piedāvāts piemērs Programmēšana II kursa beigšanas prasībām, kurā ir norādīti uzdevumi ar attiecīgo īpatsvaru kursa vērtējumā. Norādītais īpatsvars ir saistīts ar sasniedzamo rezultātu nozīmīgumu un mācību laiku, kas ir paredzēts to apguvei.

Prasības skolēnam kursa apguvei		Īpatsvars kursa vērtējumā (%)	
1. Programmprodukta izveide objektorientētas programmēšanas vidē, izmantojot ārējās bibliotēkas. Programmproduktā lieto objektorientētas programmēšanas pamatprincipus – klasi (īpašības, metodes) un objektu. Pielieto konstruktorus. Izmantojot standarta klašu bibliotēkas, aplūko grafiskās lietotāja saskarnes programmēšanas principus. Pielieto standarta bibliotēkas un, pēc nepieciešamības, citu bibliotēku piedāvātās risinājuma iespējas.		15	
2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija, iesniedzot paveiktā apliecināšu dokumentāciju. Konfigurē maršrutētāju. Izveido tīmekļa serveri programmprodukta izstrādātāja vajadzībām. Plāno un izveido datubāzi ar vairākām tabulām, pārbauda tās darbību. Izmanto datubāzi datu glabāšanai.		15	
3. Programmprodukta izveide, izmantojot dažādas datu struktūras, programmsaskarni (API) un mašīnmācīšanās principus. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai programmproduktā, tajā skaitā – sava programmēšanas projekta īstenošanai. Pielieto dažādus datu struktūru veidus (t. sk. masīvus, kopas, ierakstus, stekus, rindas, sarakstus, kokus, grafus, datnes) un ar tiem saistītos pamatalgoritmus. Prot skaidrot mašīnmācīšanās principus. Lieto mašīnmācīšanos programmvadāmā risinājumā.		15	
4. Programmprodukta izveide, izmantojot visus programmatūras modeļa izstrādes posmus. Izstrādā programmproduktu, izmantojot iepriekš epizodiski apgūtās prasmes un zināšanas par atsevišķiem izstrādes posmiem, tās ir papildinājis un nostiprinājis. Izprot katra izstrādes posma vietu kopējā izstrādes procesā. Izstrādes posmu var iedalīt divās daļās:	Problēmas analīze, programmatūras specifikācija un darba plānošana. Veic izpēti, analizē dažādus ikdienas procesus, lai saskatītu to automatizācijas iespējas. Definē problēmu, formulē darba uzdevumu. Veido programmatūras prasību specifikāciju, izvēlas programmatūras izstrādes modeli, programmēšanas valodu un programmatūras izstrādes vidi, pamato savu izvēli.	10	30
	Programmatūras izstrāde Izstrādā programmatūru izvēlētajā programmēšanas valodā, izmantojot labās prakses principus programmas koda pierakstā un strukturēšanā. Veic programmas akcepttestēšanu un atklāšanu.	20	
5. Programmprodukta izveide, sadarbojoties grupā un, izmantojot visus programmatūras modeļa izstrādes posmus. Izstrādā programmproduktu, izmantojot iepriekš apgūtās prasmes, zināšanas. Uzsvars tiek likts uz darbu dalīšanas un sadarbības prasmēm strādājot grupā, simulējot projektu izstrādes procesus uzņēmumos – skolēns to izprot un kvalitatīvi piedalās.		25	

Ieteikumi mācību darba organizācijai

Satura starpdisciplināritāte

Plānojot kursa satura apguvi, jāņem vērā, ka starp kursiem pastāv vairāki starpdisciplināritātes līmeņi, kas izpaužas gan kursu tematos, gan apgūstamajās prasmēs, kuru starpdisciplināritātes pakāpe mainās no formāla apvienojuma (var mācīt kopā vai atsevišķi) līdz pat pilnīgai integrācijai. Svarīgi nostiprināt skolotāju sadarbību – lai skolēnam veidotos izpratne par datorikas virzienu pielietojumu dažādu nozaru problēmu risināšanā. Tāpat, lai pētnieciskās u. c. prasmes tiktu apgūtas pēctecīgi un sistēmiski, kā arī pārnestas no viena padziļinātā kursa uz citu, izšķiroša loma ir konstruktīvai skolotāju sadarbībai. Skolotāju sadarbība nepieciešama gan mācību satura plānošanai, gan īstenošanai.

Programmēšana II kursa ietvaros priekšmeta skolotājam, lai nodrošinātu starpdisciplināritāti, ir aktuāli sadarboties ar dizaina un tehnoloģiju, robotikas, digitālā dizaina priekšmetu skolotājiem, lai skolēni saskatītu programmēšanas pielietojumu arī inženierzinātnēs (piemēram, mehatronikā) un produktu dizainā.

Stundu sadalījums/grafiks

Programmēšana II padziļinātā kursa apguve plānota 210 mācību stundām. Kursā sasniegtie rezultāti ir sadalīti pa atsevišķiem tematiem, kuru apguves secību pēc nepieciešamības var mainīt. Skola var paredzēt dažādu laika plānojumu, taču apjomīgus darbus ieteicams veikt blokstundās.

Kursa sasniegtā rezultātu apguvei skolotājam svarīgi izmantot daudzveidīgas mācību organizācijas formas, tajā skaitā nozīmīgu daļu laika mācību procesā atvēlot mērķtiecīgi atbalstītam skolēna patstāvīgajam – pētnieciskajam vai jaunrades – darbam. Tādēļ skolotājam sabalansēti jāplāno mācību darbs stundā, atvēlot atbilstošu laiku mērķtiecīgi virzītam un atbalstītam skolēnu patstāvīgajam darbam klasē, arī izmantojot skolā vai kopienā pieejamos informatīvos resursus (komunikatīvās platformas, infogrammas u. c.). Tikpat svarīgi skolotājiem ir izsvērti plānot un savstarpēji koordinēt skolēnu patstāvīgā darba apjomu un saturu ārpus mācību stundām, paredzot, ka tam skolēni nedēļā papildus veltīs aptuveni trešdaļu no kursa mācību stundu skaita.

Dažādas mācību darba organizācijas formas

Programmēšana II kursa programmā skolēnam izpratnes padziļināšanai par informācijas apstrādes procesiem un problēmrisināšanas prasmju nostiprināšanai ieteicams izmantot dažādas mācību darba organizācijas formas: diskusijas, gadījuma izpēti, pētniecības projekti, praktiskie darbi, vieslekcijas, semināri, mācību braucieni u. c. Tikpat svarīgi skolotājiem izsvērti plānot un savstarpēji koordinēt skolēnu patstāvīgā darba apjomu un saturu ārpus mācību stundām.

Kursa apguves pamatā ir seši temati. Jāņem vērā, ka sestā temata "Programmaprodukta izveide, sadarbojoties grupā un izmantojot visus programmatūras modeļa izstrādes posmus" darba organizācijas forma ir patstāvīga projekta izstrāde grupā.

Mācību satura apguves norise

Kursa satura pārskats

Programmēšana II saturs ir grupēts sešos tematos. Pirmajos trijos skolēns padziļināti turpina apgūt Programmēšana I iekļautos tematus, kā arī apgūst jaunas zināšanas un prasmes. Visu triju tematu saturs ir savstarpēji integrējams. Piemēram, objektorientētas programmēšanas apgūvē izmantotos uzdevumus (programmproduktus) vēlāk ir iespējams papildināt vai modificēt ar datu uzglabāšanu datubāzēs. Arī dažādu datu struktūru pielietojumu iespējams apgūt, izmantojot objektorientēto programmēšanu. Vēlāk šīs zināšanas un prasmes skolēns pielieto projektu izstrādē 4.–6. tematā.

4.–5. tematā skolēns sistematizē līdzšinējās epizodiski apgūtās iemaņas programmprodukta dzīves cikla izstrādē. Skolēns individuāli izstrādā programmproduktu, sākot no izpētes un problēmas definēšanas līdz pat produkta testēšanai un uzlabošanai. Tādējādi praktiski tiek iziets programmprodukta dzīves cikls, kāds tas ir arī uzņēmumos.

6. temats atkārto 4.–5. temata saturu ar būtisku papildinājumu, kas raksturīgs arī IT un citos uzņēmumos, kas saistīti ar informācijas apstrādi – programmprodukts tiek izstrādāts grupā, t. sk. sadalot darba pienākumus un iekārtojot sadarbības vidi.

Šī kursa saturs ir strukturēts šādos tematos:

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programmsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
---	--	---	--	-----------------------------------	---

Nākamajā sadaļā izklāstīts izvērsts kursa saturs, katru tematu aprakstot pēc turpmāk norādītā temata ietvara struktūras parauga.

Programmā lietoto kodu skaidrojums pievienots 1. pielikumā.

Temata ietvara struktūras paraugs

1. Temata numurs un nosaukums	2. Temata numurs un nosaukums	3. Temata numurs un nosaukums	4. Temata numurs un nosaukums	5. Temata numurs un nosaukums	6. Temata numurs un nosaukums
-------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------

Temata apguvei ieteicamais laiks

Temata apguves mērķis: tematā plānoto skolēnam sasniedzamo rezultātu kopums un apguves pamatojums

Temata izpētes jautājumi – dažādu veidu jautājumu piemēri, kas atspoguļo mācību satura dziļumu un plašumu. Šī sadaļa ietver jautājumu piemērus, kas:

- aktualizē nepieciešamās zināšanas un veido dziļāku konceptuālu izpratni par temata saturu/jēdzieniem;
- rosina diskusiju par tematu, ir apskatāmi no vairākiem aspektiem un nav viennozīmīgi atbildami.

Sasniedzamie rezultāti – skolēna spēja koordinēti lietot zināšanas, prasmes un ieradumus jaunās, neierastās situācijās. Iekavās norādīts kods no standarta attiecīgās mācību jomas plānoto skolēnam sasniedzamo rezultātu tabulas.

Temata apguves norise

Temata vienuma nosaukums	Tematā plānoto skolēnam sasniedzamo rezultātu apguvei nepieciešamās skolēna darbības
--------------------------	--

Izvērstis kursa saturs

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
---	--	--	--	-----------------------------------	---

1. Objektorientēta programmēšana un ārējās bibliotēkas

Temata apguvei ieteicamais laiks: 30 mācību stundas

Temata apguves mērķis: izmantojot objektorientētas programmēšanas valodas pamatprincipus un bibliotēkas, spēt izstrādāt konkrētu programmatūras risinājumu.

Temata izpētes jautājumi

- Kādi ir objektorientētas programmēšanas valodas pamatprincipi?
- Kur tiek deklarēti iekšējie mainīgie un funkcijas?
- Kas ir klase un objekts?
- Kādas iespējas piedāvā standarta un ārējā objektorientētas programmēšanas valodas bibliotēka?
- Kādam nolūkam lieto konstruktorus?
- Kādam nolūkam nepieciešama abstrakcija, iekapsulēšana, mantošana, polimorfisms?

Sasniedzamie rezultāti

Skaidro objektorientētās programmēšanas pamatprincipus, veido programmas vienā no objektorientētajām programmēšanas valodām. (T.A. 2.4.15.)

Izmanto programmēšanas valodas un tās bibliotēku dokumentāciju un palīdzības sistēmu, lai patstāvīgi apgūtu citas to piedāvātās iespējas, kas nepieciešamas konkrētās programmatūras izstrādei. (T.A. 2.4.10.)

Temata apguves norise

<p>Objektorientētas programmēšanas pamatprincipi</p>	<p>Skaidro galvenos objektorientētās programmēšanas pamatprincipus kādā no OOP valodām (piemēram, <i>Python</i>, <i>Java</i>, <i>C#</i> u. c.): Klase (<i>class</i>) Objekts (<i>object</i>)</p> <p>Raksturo objektorientētas programmēšanas valodas izmantošanas iespējas, tās priekšrocības un trūkumus.</p> <p>Veic vienāda koda rindas salīdzinājumu divās dažādās objektorientētās programmēšanas valodās (piemēram, <i>Python</i> un <i>C#</i>).</p> <p>Izveido savu klasi (<i>class</i>), iekļaujot tajā attiecīgas īpašības (<i>property/field</i>) un metodes (<i>method</i>), piemēram, klase "prece" ar īpašībām "preces nosaukums", "cena" un metode "cena ar PVN".</p> <p>Pārbauda klases funkcionalitāti, izveidojot programmā klases objektu un veicot ar to attiecīgas darbības, piemēram, izsauc metodi un izmaina īpašību.</p> <p>Izveido savā klasē (angl. <i>class</i>) vismaz trīs dažādus konstruktorus (angl. <i>constructor</i>), ievērojot to uzbūves pamatprincipus. Veidojot konstruktorus pēc nepieciešamības, pielieto arī īpašību noklusējuma vērtības (piemēram, daudzums noliktavā = "nav informācijas").</p> <p>Pārbauda konstruktoru darbību, izveidojot savā programmā dažādus objektus pēc to konstruktoru uzbūves.</p> <p>Skaidro atslēgvārda "<i>this</i>"/"<i>self</i>" pielietojuma piemērus objektorientētā programmēšanā un izvērtē to nepieciešamību un pielietojuma iespējas programmas kodā, piemēram, veidojot konstruktorus.</p> <p>Skaidro objektorientētās programmēšanas pamatjēdzienus, izmantojot skolotāja sagatavotos praktiskos piemērus: abstrakcija; iekapsulēšana; mantošana; polimorfisms.</p> <p>Veido nelielu programmas risinājumu vienā no objektorientētās programmēšanas valodām, ievērojot tās pamatprincipus un labās prakses principus programmas koda pierakstā un strukturēšanā, piemēram, preču informācijas apstrādes programma ar ievades, izvades laukiem un pogu. Programmā tiek lietota paša veidota klase ar tās īpašībām un metodēm informācijas reģistrēšanai un izvadišanai, izmantojot grafisko lietotāja saskarni.</p>
<p>Bibliotēku piesaiste programmproduktā</p>	<p>Lietojot izvēlētās programmēšanas valodas un tās bibliotēku dokumentāciju un palīdzības sistēmu, raksturo standarta bibliotēku piedāvātās iespējas.</p> <p>Pēc nepieciešamības pielieto savam programmatūras risinājumam nepieciešamo funkcionalitāti no standarta bibliotēkas, piemēram, <i>C#</i> programmēšanas valodā bibliotēkas "<i>System.IO</i>" klasē "<i>StreamWriter</i>" izveido objektu, ar kura palīdzību izveido teksta datni ar paplašinājumu <i>.txt</i> un saglabā tajā nepieciešamo tekstuālo informāciju, kuru savukārt varēs nolasīt ar citas klases palīdzību "<i>StreamReader</i>".</p> <p>Izvērtē citu bibliotēku piedāvātās risinājuma iespējas. Pēc nepieciešamības pielieto tās savā programmatūras izstrādes risinājumā, ievērojot bibliotēkas licences nosacījumus.</p>

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programmsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
--	---	--	---	----------------------------	--

2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija

Temata apguvei ieteicamais laiks: 30 mācību stundas

Temata apguves mērķis: spēt izveidot un konfigurēt atvērtu vai aizsargātu daudzlietotāju lokālu tīklu, izveidot vienkāršu serveri programmatūras izstrādātāja vajadzībām, izplānot un realizēt datubāzes pielietojumu programmaproduktā.

Temata izpētes jautājumi

- Kādam nolūkam lieto datubāzi?
- Kāds ir relāciju pielietojums datubāzē?
- Kādam nolūkam kalpo SQL komandas *INSERT*, *SELECT*, *DELETE*?
- Kas ir kriptēšana, un kāds ir tās pielietojums?
- Ko ietver maršrutētāja noklusētā konfigurācija?
- Kas ir dinamiskā un statiskā IP adrese?
- Kas ir MAC adreses, un kā tās var izmantot tīkla lietotāja piekļuves iespēju kontrolei maršrutētājā?
- Kā maršrutētājā var bloķēt nevēlamas saites?
- Kāda ir tīmekļa servera problemātika, to uzturot savā uzņēmumā?
- Kāda ir tīmekļa servera problemātika, to nomājot?

Sasniedzamie rezultāti

Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam. (T.A. 2.3.2.)

Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām. (T.A. 2.4.17.)

Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā. (T.A. 3.1.2.)

Izveido un konfigurē atvērtu vai aizsargātu daudzlietotāju lokālu tīklu, t. sk. veidojot atvērtus vai aizsargātus bezvadu piekļuves punktus. Izveido vienkāršu serveri un konfigurē piekļuvi tam no interneta. (T.A. 2.3.1.)

Temata apguves norise

Datubāzes projektēšana un izveide	<p>Nepieciešams datubāzes serveris ar uzstādītu <i>MySQL</i> vai līdzīgu datubāzes programmatūru.</p> <p>Uzzīmē uz papīra (izplāno) datubāzes shēmu ar vairākām tabulām un relācijām.</p> <p>Izvēlas klienta programmatūru, lai pieslēgtos datubāzei (<i>MySQL Workbench</i>, <i>PHPMyAdmin</i> vai līdzīgu).</p> <p>Izveido datubāzi.</p> <p>Izveido datubāzes tabulas, konfigurē laukus.</p> <p>Konfigurē relācijas.</p> <p>Pārbauda datubāzes darbību, izmantojot <i>SQL</i> komandas <i>INSERT</i>, <i>SELECT</i>, <i>DELETE</i>.</p>
Kriptogrāfija	<p>Pielieto kriptēšanas funkcijas, piemēram, ievietojot ierakstu datubāzē ar kriptētu paroli <i>MySQL MD5</i>.</p> <p>Skaidro jēdzienus “privātā” un “publiskā” atslēga. Nosauc to pielietojumu.</p>
Tīkla un servera konfigurācija	<p>Izveido tīmekļa serveri (piemēram, <i>Linux Ubuntu</i>, <i>Apache</i>, <i>MySQL</i> vai <i>Windows</i> un <i>XAMPP/WAMP/Laragon</i>). /Apgūst kopā ar 2.3.2. punktu./</p> <p>Standarta tīkla izveide – tīkls, kurš iekļauj vismaz vienu maršrutētāju, komutatoru un nodrošina pieslēgumu internetam.</p> <p>Skaidro datu pārraides teorijas pamatprincipus.</p> <p>Sagatavo/pārliecinās, ka ir sagatavota nepieciešamā datortehnika:</p> <ul style="list-style-type: none"> • interneta pieslēgums; • kabeli; • komutators, maršrutētājs, vismaz 2 datori, 1 <i>web</i> serveris. <p>Konfigurē maršrutētāju:</p> <ul style="list-style-type: none"> • iepazīstas ar noklusēto konfigurāciju; • skaidro atšķirību starp dinamisko un statisko publisko IP un pielieto atbilstošo IP maršrutētāja konfigurācijā; • konfigurē piekļuvi, izmantojot MAC adreses (ACL – piekļuves kontrole); • pielieto <i>port forwarding</i>; • iestata <i>WiFi</i>; • skaidro <i>HTTPS</i> nepieciešamību no drošības viedokļa (angl. <i>letsencrypt</i>); /skolotāja demonstrācija, sasaistot ar kriptogrāfijas tēmu/; • bloķē nevēlamās saites. <p>Pamato drošības tehnikas noteikumu ievērošanu servera uzturēšanā (piemēram, izvietojums savās telpās, tā drošība, mikroklimats) un ārpakalpojumā (servera nomas iespējas un piekļuves tiesības).</p>

Metodiskais komentārs

Datubāzu apguvē var izmantot:

<https://www.w3schools.com/sql/default.asp>

https://www.w3schools.com/sql/sql_join.asp

<https://medium.com/@tushar0618/how-to-create-er-diagram-of-a-database-in-mysql-workbench-209fbf63fd03>

Kriptogrāfijas apguvē var izmantot:

<https://dev.mysql.com/doc/refman/5.6/en/encryption-functions.html>

Datorīkla apguvē var izmantot:

<https://help.mikrotik.com/docs/display/ROS/First+Time+Configuration>

Papildu iespējas maršrutētāja konfigurācijā darbā ar spējīgākajiem, uzcītīgākajiem:

- IP piekļuve;
- interneta savienojumi;
- maršrutētāja aizsardzība;
- lietotāju piekļuves tiesības;
- piekļuve, izmantojot IP adreses;
- administratīvo u. c. servisu konfigurācija;
- NAT konfigurācija.

Uzstāda, konfigurē VPN.

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programmsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
--	---	--	---	----------------------------	--

3. Datu struktūras, programmsaskarne (API), mašīnmācīšanās

Temata apguvei ieteicamais laiks: 36 mācību stundas

Temata apguves mērķis: izmantojot atvērtā koda bibliotēkas un dažādas datu struktūras, spēt izstrādāt API risinājumu, kā arī prast pielietot mašīnmācīšanās algoritmu darbības principus programmvadāmu risinājumu izstrādē.

Temata izpētes jautājumi

- Kā pielieto atvērta koda bibliotēkas, un kādi ir riski?
- Kā pielieto API risinājumus, kādi ir drošības riski, izmantojot API?
- Kā izveidot API atbildi?
- Kādam mērķim paredzētas šādas datu struktūras: masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne?
- Kā pielietot dažādas datu struktūras?
- Kādi ir mašīnmācīšanās principi?
- Kā pielietot mašīnmācīšanās algoritmus programmvadāmā risinājumā?

Sasniedzamie rezultāti

Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai, tajā skaitā – sava programmēšanas projekta īstenošanai. (T.A. 2.4.11.)

Izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus. (T.A. 2.4.14.)

Mašīnmācīšanās algoritmu darbības principi un to pielietojums programmvadāmu risinājumu izstrādē. Izmanto gatavu mašīnmācīšanās algoritmu mācību uzdevuma risinājuma izveidē. (T.A. 2.4.18.)

Temata apguves norise

<p>Programmsaskarne (API) un atvērtā koda bibliotēkas</p>	<p>Pamato atvērtā koda bibliotēku pielietojumu un to pielietojuma iespējamās problēmas, t. sk. riskus un koda daļas, kuras programma neizmanto.</p> <p>Izmanto atvērtā koda bibliotēkas savā programmrisinājumā, atvieglojot savas programmas darbu.</p> <p>Pamato API lietošanas jēgu un nepieciešamos drošības apsvērumus (JSON).</p> <p>Skaidro API pielietojumu un nepieciešamību. Pamato efektīvas API iespējas.</p> <p>Skaidro API atslēgas pielietojumu no drošības aspekta.</p> <p>Izmanto API atvērtos risinājumus, pielieto API atslēgas.</p> <p>Izveido API risinājumu – uzbūvē savu API galapunktu, kurš kādā no formātiem atbild uz pieprasījumu (piemēram, izveido čatu).</p>
<p>Datu struktūras</p>	<p>Izmēģina dažādus datu struktūru veidus (piemēram, darbojoties ar API).</p> <p>Jēgpilni lieto dažādus datu struktūru veidus (piemēram, nelielu uzdevumu veidā), izstrādā jaunas vai papildina esošās programmas.</p>
<p>Mašīnmācīšanās algoritmu darbības principi</p>	<p>Skaidro mašīnmācīšanās principus (piemēram, patstāvīgi darbojoties ar interneta avotiem, skolēns sagatavo prezentāciju par mašīnmācīšanās principiem, to pielietojumu).</p> <p>Lieto mašīnmācīšanos programmā vadāmā risinājumā (piemēram, izmantojot gatavu algoritmu, izveido programmu, kas atpazīst dažādu objektu attēlus, piemēram, mājas attēlus).</p>

Metodiskais komentārs

API apgūvē var izmantot:

https://www.w3schools.com/whatis/whatis_json.asp

Veidojot API izsaukumus, var izmantot bezmaksas API servīsus, kuriem nav vajadzīga atslēga. Ja skolēni ir gana droši, tad var izmantot API servīsus ar atslēgām un svarīgi skaidrot API atslēgu vajadzību drošības nolūkos.

Tēmu iespējams sasaistīt ar dabubāzu apguvi.

Mašīnmācīšanās algoritmu apgūvē var izmantot:

https://www.w3schools.com/python/python_ml_getting_started.asp

Var papildināt jau esošos darbus jeb izveidotās programmas.

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programmsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
--	---	--	---	----------------------------	--

4. Problēmas analīze, programmatūras specifikācija un darba plānošana

Temata apguvei ieteicamais laiks: 24 mācību stundas

Temata apguves mērķis: izpētes rezultātā prast saskatīt automatizācijas iespējas dažādos ikdienas darba procesos, definēt darba uzdevumu (lai izstrādātu programmatūras risinājumu) un veidot programmatūras prasību specifikāciju, pēc kuras ir iespējams izstrādāt atbilstošu programmatūru.

Temata izpētes jautājumi

- Kādas metodes var pielietot dažādu ikdienas procesu izpētē un analīzē?
- Kā izvēlēties piemērotāko izpētes metodi, lai dažādos ikdienas procesos saskatītu automatizācijas iespējas?
- Kādas metodes var pielietot problēmas definēšanas procesā?
- Kā izvēlēties piemērotāko problēmas definēšanas metodi, lai precīzi definētu risināmo problēmu?
- Kā precīzi definēt darba uzdevumu programmētājam?
- Kādu informāciju nepieciešams iekļaut programmatūras prasību specifikācijā, un kā to noformēt?
- Kādas metodes ir iespējams izmantot programmvadāma risinājuma prototipa veidošanā, un kā izvēlēties piemērotāko?
- Kā izvēlēties programmvadāma risinājuma izveidei piemērotāko programmatūras izstrādes modeli, programmēšanas valodu un programmatūras izstrādes vidi?

Sasniedzamie rezultāti

Veic izpēti, analizē dažādus ikdienas procesus. (T.A. 2.4.1.)

Veic problēmas definēšanu, formulē darba uzdevumu. (T.A. 2.4.1.)

Veido programmatūras prasību specifikāciju, izvēlas programmatūras izstrādes modeli, programmēšanas valodu un programmatūras izstrādes vidi, pamato savu izvēli. (T.A. 2.4.2., T.A.2.4.4., T.A.2.4.12.)

Temata apguves norise

Dažādu ikdienas procesu izpēte un analīze	<p>Salīdzina dažādas izpētes metodes (piemēram, balstoties uz dizaina domāšanas metodoloģiju – startdesign.lv). Izvēlas piemērotāko izpētes metodi, lai veiktu dažādu ikdienas procesu izpēti un analīzi un saskatītu tajos automatizācijas iespējas. Pamato izpētes metodes piemērotību, izveido izpētes plānu.</p> <p>Analizē dažādus ikdienas darba procesus (Kas ir konkrētais process? Kurus procesa elementus ir iespējams automatizēt? Kāpēc procesa automatizācija ir nepieciešama? Kam un kādi būs ieguvumi no procesa automatizācijas?), veic izpētes procesa dokumentēšanu.</p>
Problēmas un darba uzdevuma definēšana	<p>Balstoties uz izpētes procesa rezultātiem, definē risināmo problēmu (piemēram, balstoties uz dizaina domāšanas metodoloģiju – startdesign.lv).</p> <p>Definē un raksturo izstrādājamā programmatūras risinājuma mērķauditoriju, analizē un dokumentē mērķauditorijas specifiku un vajadzības (piemēram, veidojot lietotāja profilu vai lietotāja pieredzes karti), lai varētu pēc iespējas precīzāk formulēt darba uzdevumu un veidot programmatūras prasību specifikāciju.</p> <p>Formulē darba uzdevumu programmatūras izstrādātājam, kas turpmāk tiks detalizēti izklāstīts programmatūras prasību specifikācijā.</p>
Programmatūras prasību specifikācijas izveide	<p>Veido vienkāršotu programmatūras prasību specifikāciju atbilstoši konkrētajam uzdevumam. Programmatūras prasību specifikācijā iekļauj:</p> <ol style="list-style-type: none"> 1) risināmo problēmu un īsu tās raksturojumu, 2) izstrādājamā automatizācijas risinājuma galveno mērķi un uzdevumus, 3) risinājuma lietotāju lomas un tiem paredzētās darbības, 4) datu klases, 5) programmatūras darbības aprakstu (funkcijas), 6) citu informāciju pēc nepieciešamības. <p>Veido vienkāršotu programmatūras risinājuma prototipu, piemēram, lietotāja saskarnes struktūrskices (<i>wireframe</i>), blokshēmas, iekļauj to programmatūras prasību specifikācijā.</p> <p>Balstoties uz definēto darba uzdevumu un tā izklāstu programmatūras prasību specifikācijā, izvēlas programmvadāma risinājuma izveidei piemērotāko programmatūras izstrādes modeli, programmēšanas valodu un programmatūras izstrādes vidi.</p> <p>Pamato programmatūras izstrādes modeļa, programmēšanas valodas un programmatūras izstrādes vides piemērotību konkrētajam darba uzdevumam, dokumentē to.</p>

Metodiskais komentārs

Projekta izstrādes posmu apskatē (izpētes, problēmas definēšanas, ideju ģenerēšanas) iespējams izmantot bezmaksas tiešsaistes kursu par dizaina domāšanu: <https://www.start-design.lv/>

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
--	---	---	---	----------------------------	--

5. Programmatūras izstrāde

Temata apguvei ieteicamais laiks: 42 mācību stundas

Temata apguves mērķis: strukturējot programmas kodu, spēt izstrādāt programmatūru, izmantojot programmatūras izstrādes labās prakses principus.

Temata izpētes jautājumi

- Kā izstrādāt programmatūru, balstoties uz izveidotu programmatūras prasību specifikāciju?
- Kādi ir labās prakses principi programmas koda pierakstā un strukturēšanā, un kā tos pielietot programmvadāma risinājuma izstrādē?
- Kā un kādus rīkus ir iespējams pielietot projektu un versiju pārvaldībai programmatūras izstrādes laikā?
- Kas ir akcepttestēšana, un kā to veic?
- Kāpēc programmatūras izstrādes un testēšanas procesus nepieciešams dokumentēt?
- Kā veikt dokumentēšanu?

Sasniedzamie rezultāti

Programmatūras izstrāde, izmantojot labās prakses principus programmas koda pierakstā un strukturēšanā. (T.A. 2.4.13., T.A.2.4.19., T.A.2.4.6., T.A.2.4.9.)

Programmatūras testēšana. (T.A. 2.4.6., T.A.2.4.9.)

Temata apguves norise

Programmatūras izstrāde, izmantojot labās prakses principus programmas koda pierakstā un strukturēšanā	<p>Izstrādā programmatūru izvēlētajā programmēšanas valodā (individuāli vai grupā):</p> <ol style="list-style-type: none"> 1) lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā (piemēram, https://github.com/); 2) izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumu atbilstoši izstrādātajai programmatūras prasību specifikācijai; 3) izmanto gatavus algoritmus, pielāgojot un/vai kombinējot tos, un/vai izstrādājot jaunus; 4) apvieno un integrē programmatūras vienības.
Programmatūras testēšana	<p>Veic programmatūras vienībtestēšanu:</p> <ol style="list-style-type: none"> 1) lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā (piemēram, https://github.com/); 2) veic akcepttestēšanu atbilstoši izstrādātajai programmatūras prasību specifikācijai; 3) veic programmatūras atklādošanu; 4) pielāgo programmatūru atbilstoši specifikācijai; 5) dokumentē testēšanas un pielāgošanas procesu; 6) demonstrē programmatūras risinājumu klientam. Ja nepieciešams, papildina vai maina specifikācijas prasības un pielāgo programmatūras risinājumu.

1. Objektorientēta programmēšana un ārējās bibliotēkas	2. Datortīkla, servera un drošas datubāzes izveide un konfigurācija	3. Datu struktūras, programmsaskarne (API), mašīnmācīšanās	4. Problēmas analīze, programmatūras specifikācija un darba plānošana	5. Programmatūras izstrāde	6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus
--	---	--	---	----------------------------	--

6. Produkta izveide grupā, izmantojot visus programmatūras modeļa izstrādes posmus

Temata apguvei ieteicamais laiks: 54 mācību stundas

Temata apguves mērķis: prast izstrādāt programmvadāmu risinājumu grupā (t. sk. sadalot darba pienākumus un iekārtojot sadarbības vidi), īstenojot visus programmatūras izstrādes dzīves cikla posmus.

Temata izpētes jautājumi

- Kā definēt programmrisinājuma problēmu?
- Kā izveidot specifikāciju?
- Kā vadīt programmatūras izstrādes procesu?
- Kā testēt izstrādāto programmatūras izstrādājumu?
- Kādu licences veidu izmantot savam programmatūras risinājumam?
- Kā ieviest programmatūras risinājumu?
- Kā prezentēt ieviesto programmatūras risinājumu?

Sasniedzamie rezultāti

Veic vienkāršotu programmatūras projektēšanu (t. sk. lietotāju saskarnes un vienkāršotu datu modeļa izveidi) atbilstoši programmatūras prasību specifikācijā izvirzītajām funkcionālajām un nefunkcionālajām prasībām. Veic vienkāršotu programmatūras izstrādes plānošanu (bez darbietilpības novērtējuma) un darba pienākumu sadali grupā. (T.A. 2.4.5.)

Izstrādā programmatūru grupā, ievērojot izvēlēto programmatūras izstrādes modeļa posmus, veic katra posma vienkāršotu dokumentēšanu. (T.A. 2.4.3)

Veido programmu, ievērojot labās prakses pieredzi tās pieraksta strukturēšanā un komentāru veidošanā, vienojoties par vienotu stilu ar visiem grupas dalībniekiem. (T.A. 2.4.8.)

Veic programmatūras vienību izstrādi un vienībtestēšanu, izstrādājamās programmatūras vienību apvienošanu, integrācijas un akcepttestēšanu atbilstoši izstrādātajai programmatūras prasību specifikācijai un projektējuma aprakstam. (T.A. 2.4.6)

Lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā. (T.A. 2.4.9.)

Izstrādā programmatūru grupā, ievērojot izvēlēto programmatūras izstrādes modeļa posmus, veic katra posma vienkāršotu dokumentēšanu. (T.A. 2.4.3.)

Veic izstrādātajā programmatūrā izmantoto algoritmu sarežģītības novērtēšanu pēc laika un izmantotās atmiņas un, ja nepieciešams, veic algoritmu optimizāciju. (T.A. 2.4.16)

Izstrādā un prezentē izveidotās programmatūras vienkāršotu izvēršanas (t. sk. ieviešanas) plānu, lietotāja ceļvedi un uzturēšanas plānu, ievērojot tās lietotāju mērķauditorijas specifiku. (T.A. 2.4.7.)

Izstrādā programmatūru grupā, ievērojot izvēlēto programmatūras izstrādes modeļa posmus, veic katra posma vienkāršotu dokumentēšanu. (T.A. 2.4.3)

Salīdzina atvērto kodu licences un to atšķirības, izmanto un piemēro atbilstošāko no licencēm savam programmatūras projektam. (T.A. 3.1.4)

Temata apguves norise

<p>Izpēte, problēmas definēšana un specifikācijas izveide</p>	<p>Izpētes rezultātā apzina vajadzības un risinājuma iespējas, definē risināmo problēmu.</p> <p>Grupā ģenerē idejas, izvēlas risinājumu un stratēģiju.</p> <p>Plāno sava risinājuma izstrādei nepieciešamos resursus un secīgi veicamos darba soļus.</p> <p>Veido programmatūras prasību specifikāciju.</p> <p>Sadala darba pienākumus grupā.</p>
<p>Programmatūras izstrāde un testēšana</p>	<p>Izstrādā programmatūru izvēlētajā programmēšanas valodā:</p> <ul style="list-style-type: none"> • lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā (piemēram, https://github.com/); • izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam atbilstoši izstrādātajai programmatūras prasību specifikācijai; • izmanto gatavus algoritmus, pielāgojot un/vai kombinējot tos, un/vai izstrādājot jaunus; • apvieno un integrē programmatūras vienības. <p>Veic programmatūras vienībtestēšanu:</p> <ul style="list-style-type: none"> • veic akcepttestēšanu atbilstoši izstrādātajām programmatūras prasībām; • veic programmatūras atklūdošanu; • pielāgo programmatūru atbilstoši specifikācijai; • dokumentē testēšanas un pielāgošanas procesu; • demonstrē programmatūras risinājumu klientam. Ja nepieciešams, papildina vai maina specifikācijas prasības un pielāgo programmatūras risinājumu; • veic izstrādātajā programmatūrā izmantoto algoritmu sarežģītības novērtēšanu pēc laika un izmantotās atmiņas un, ja nepieciešams, veic algoritmu optimizāciju.
<p>Ieviešanas plāns, lietotāja ceļvedis un uzturēšanas plāns</p>	<p>Salīdzina atvērto kodu licences, izmanto un piemēro atbilstošāko no licencēm savam programmatūras projektam.</p> <p>Izstrādā un prezentē izveidotās programmatūras vienkāršotu izvēršanas (t. sk. ieviešanas) plānu, lietotāja ceļvedi un uzturēšanas plānu, ievērojot tās lietotāju mērķauditorijas specifiku.</p> <p>Prezentē izstrādāto programmatūras risinājumu (piemēram, klasē).</p>

Metodiskais komentārs

Dažādu licenču noteikumu salīdzināšanai var izmantot:

<https://choosealicense.com/licenses/>

https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licences

Pielikumi

1. pielikums

Kursu programmu paraugos lietotie kodi

Atsaucei uz standartu* mācību priekšmetu programmās izmantoti šādi plānoto skolēnam sasniedzamo rezultātu (SR) un lielo ideju (Li) kodi. (Standarta pielikumi, kuros lietoti šie kodi, atrodami *Skola2030* tīmekļa vietnē.)

SR kodi

Piemērs:

VLM.3.2.1.9.			
VLM.	3.	2.1.9.	
Mācību joma (visu mācību jomu apzīmējumus sk. tabulā)	Izglītības posma pēdējās klases numurs	Mācību jomas SR kārtas numurs standartā	
		2.1.9. Saprāt, ka tekstveide ir process, kura laikā tekstu vairākkārt var uzlabot. Pēc parauga un pedagoga ieteikumiem labo un pilnveido tekstu	2.1.9. Labo un pilnveido savu tekstu, sniedz un iegūst atgriezenisko saiti par teksta saturu un noformējumu. Prot strādāt individuāli un sadarboties teksta pilnveides laikā
			2.1.9. Redīgē savu tekstu. Sniedz un saņem konstruktīvu atgriezenisko saiti. Izmanto dažādus paņēmienus teksta uzlabošanai, piemēram, jautājumu formulēšanu, diskusijas, nepieciešamo avotu un resursu izmantošanu, laika plānojumu

Li kodi

Piemērs:

S.Li.8.		
S.	Li.	8.
Mācību joma	Lielā ideja	Mācību jomas Li kārtas numurs standartā
		8. Informācijas avoti, kas atspoguļo norises sabiedrībā pagātnē un mūsdienās, ir izvērtējami kritiski
		8.1. Raksturo dažādu plašsaziņas līdzekļu sniegtās informācijas izmantošanas iespējas, atrod un atlasa faktus
		8.1. Kritiski izvērtē un izmanto dažādu plašsaziņas līdzekļu un vēstures avotu sniegto informāciju. Salīdzina dažādos informācijas avotos atrodamos faktus, meklē līdzības un atšķirības
		8.1. Analizē un skaidro plašsaziņas līdzekļu iespējas atspoguļot un ietekmēt cilvēku politiskos, sabiedriskos, estētiskos priekšstatus un uzskatus, manipulēt ar personisko un kultūras identitātes izpratni, priekšstatiem par kultūras mantojumu un vērtībām

Mācību jomu apzīmējumi

V	Valodu mācību joma	
	VL	Latviešu valoda
	VLM	Latviešu valoda un literatūra izglītības iestādēs, kas īsteno mazākumtautību izglītības programmas
	VS	Svešvaloda
	VM	Mazākumtautības valoda
K	Kultūras izpratnes un pašizpaušmes mākslā mācību joma	
S	Sociālā un pilsoniskā mācību joma	
D	Dabaszinātņu mācību joma	
M	Matemātikas mācību joma	
T	Tehnoloģiju mācību joma	
F	Veselības un fiziskās aktivitātes mācību joma	

* Ministru kabineta 2018. gada 27. novembra noteikumi Nr. 747 "Noteikumi par valsts pamatzglītības standartu un pamatzglītības programmu paraugiem".

2. pielikums

Programmēšana II kursā plānotie skolēnam sasniedzamie rezultāti

Tehnoloģiju mācību jomas plānotie skolēnam sasniedzamie rezultāti augstākajā mācību satura apguves līmenī

2.3.1. Izveido un konfigurē atvērtu vai aizsargātu daudzlietotāju lokālu tīklu, t. sk. veidojot atvērtus vai aizsargātus bezvadu piekļuves punktus. Izveido vienkāršu serveri un konfigurē piekļuvi tam no interneta.
2.3.2. Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam.
2.4.1. Analizē dažādus ikdienas darba procesus, saskata tajos vai to daļās automatizācijas iespējas un to, kā pasūtītājs formulē darba uzdevumu programmatūras izstrādātājam.
2.4.2. Salīdzina un izvēlas piemērotāko programmatūras izstrādes modeli konkrētā uzdevuma atrisināšanai, pamato izvēli.
2.4.3. Izstrādā programmatūru grupā, ievērojot izvēlētajā programmatūras izstrādes modeļa posmus, veic katra posma vienkāršotu dokumentēšanu.
2.4.4. Sastāda vienkāršotu programmatūras prasību specifikāciju atbilstoši konkrētajam uzdevumam, izvērtējot mērķauditorijas specifiku un vajadzības.
2.4.5. Veic vienkāršotu programmatūras projektēšanu (t. sk. lietotāju saskarnes un vienkāršotu datu modeļa izveidi) atbilstoši programmatūras prasību specifikācijā izvirzītajām funkcionālajām un nefunkcionālajām prasībām. Veic vienkāršotu programmatūras izstrādes plānošanu (bez darbietilpības novērtējuma) un darba pienākumu sadali grupā.
2.4.6. Veic programmatūras vienību izstrādi un vienībtestēšanu, izstrādājamās programmatūras vienību apvienošanu, integrācijas un akcepttestēšanu atbilstoši izstrādātajai programmatūras prasību specifikācijai un projektējuma aprakstam.
2.4.7. Izstrādā un prezentē izveidotās programmatūras vienkāršotu izvēršanas (t. sk. ieviešanas) plānu, lietotāja ceļvedi un uzturēšanas plānu, ievērojot tās lietotāju mērķauditorijas specifiku.
2.4.8. Veido programmu, ievērojot labās prakses pieredzi tās pieraksta strukturēšanā un komentāru veidošanā, vienojoties par vienotu stilu ar visiem grupas dalībniekiem.
2.4.9. Lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā.

2.4.10. Izmanto programmēšanas valodas un tās bibliotēku dokumentāciju un palīdzības sistēmu, lai patstāvīgi apgūtu citas to piedāvātās iespējas, kas nepieciešamas konkrētās programmatūras izstrādei.
2.4.11. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai.
2.4.12. Izvēlas programmēšanas valodu un programmatūras izstrādes vidi programmatūras izstrādē atbilstoši uzdevuma specifikai, pamato savu izvēli.
2.4.13. Izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam.
2.4.14. Izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus.
2.4.15. Skaidro objektorientētās programmēšanas pamatprincipus, veido programmas vienā no objektorientētajām programmēšanas valodām.
2.4.16. Veic izstrādātajā programmatūrā izmantoto algoritmu sarežģītības novērtēšanu pēc laika un izmantotās atmiņas un, ja nepieciešams, veic algoritmu optimizāciju.
2.4.17. Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām.
2.4.18. Salīdzina mašīnmācīšanās algoritmus (t. sk. attēlu atpazīšanai) un raksturo to izmantošanas iespējas jaunas programmatūras izstrādē. Izmanto bibliotēku, kurā realizēti mašīnmācīšanās algoritmi, izstrādājot programmatūru mācību uzdevuma realizācijai.
2.4.19. Veido dotā uzdevuma (problēmas) risinājumu, izmantojot gatavus algoritmus un/vai pielāgojot vai kombinējot tos, un/vai izstrādājot jaunus algoritmus. Izprot un skaidro dažādu algoritmu darbību, pielāgo tos dažādām nestandarta situācijām, ja nepieciešams, veidojot jaunas datu struktūras.
3.1.2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā.
3.1.4. Salīdzina atvērto kodu licences un to atšķirības, izmanto un piemēro atbilstošāko no licencēm savam programmatūras projektam.

3. pielikums

Plānotie skolēnam sasniedzamie rezultāti caurviju prasmēs, beidzot vispārējās vidējās izglītības pakāpi

1. Kritiskā domāšana un problēmrisināšana:

- 1.1. mērķtiecīgi formulē precīzus jautājumus, lai kritiski analizētu kompleksas situācijas un abstraktas idejas. Izzina kontekstu, to analizē, kritiski izvērtē, kā arī sintezē un interpretē informāciju, lai sasniegtu konkrētu mērķi. Gūst vispusīgu, precīzu informāciju par kompleksiem jautājumiem, izvērtē tās ticamību un analizē, kādēļ atsevišķās situācijās iegūt ticamu informāciju ir grūti;
- 1.2. kompleksās situācijās spriež no konkrētā uz vispārīgo un no vispārīgā uz konkrēto. Pamana loģiskās argumentācijas kļūdas savos un citu izteikumos, novērš tās. Argumentē, pierādot izteiktā apgalvojuma ticamību un veidojot pamatotus secinājumus;
- 1.3. nosaka aktuālas vajadzības, precīzi formulē kompleksu problēmu un pamato nepieciešamību to risināt, izvirza mērķi, piedāvā vairākus risinājumus, izvērtē tos attiecībā pret mērķi, izvēlas īstenot labāko;
- 1.4. kompleksās, neskaidrās situācijās patstāvīgi izstrādā problēmas risinājuma plānu un īsteno to, izvēloties, lietojot un pielāgojot piemērotas problēmrisināšanas stratēģijas, elastīgi reaģē uz neparedzētām pārmaiņām, izvērtē paveikto un gūtos secinājumus izmanto arī citā kontekstā.

2. Jaunrade un uzņēmējspēja:

- 2.1. interesējas par atklājumiem un inovācijām; proaktīvi meklē jaunas iespējas, kā efektīvi uzlabot savu un citu dzīves kvalitāti; rosina uzlabot esošo situāciju, pieņem nepieredzētus, kompleksus izaicinājumus, saglabā emocionālu līdzsvaru un atvērtību nenoiektības apstākļos;
- 2.2. raugoties uz situāciju no dažādiem skatpunktiem, pamana jaunas iespējas, mērķtiecīgi un elastīgi izmanto vai attīsta pats savas ideju radīšanas stratēģijas, lai nonāktu pie jauniem un noderīgiem risinājumiem; efektīvi organizē resursus (cilvēku, zināšanu, kapitāla, infrastruktūras), lai īstenotu savu ieceri; patstāvīgi meklē, izvērtē un atbildīgi izmanto citu idejas, kā arī piedāvā savas, lai iedvesmotu citus;
- 2.3. gan patstāvīgi, gan grupā attīsta ideju ilgtspējīgā piedāvājumā, kļūdas un grūtības izmanto kā iespēju izaugsmei.

3. Pašvadīta mācīšanās:

- 3.1. regulāri un atbilstoši savām vajadzībām izvirza īstermiņa un ilgtermiņa mērķus; formulē kritērijus, pēc kuriem izvērtēt, vai mērķis ir sasniegts; plāno mērķa īstenošanas soļus, uzņemas atbildību par savu lomu soļu īstenošanā un mērķu sasniegšanā;
- 3.2. patstāvīgi un regulāri analizē un reflektē par savas darbības saistību ar emocijām, personiskajām īpašībām un uzvedību; rod veidus, kā attīstīt spējas pārvaldīt savu domāšanu, emocijas un uzvedību;
- 3.3. patstāvīgi izvēlas, pielāgo un rada savas domāšanas stratēģijas kompleksās situācijās;
- 3.4. pieņemot atbildīgus lēmumus, vada emocijas sociāli pieņemamā veidā un orientējas uz iespējām, ieguvumiem un pozitīviem risinājumiem;
- 3.5. patstāvīgi izmanto kritērijus, kas palīdz īstenot darba uzraudzīšanu un pilnveidošanu; izvērtē, apkopo un turpmākajā darba procesā mērķtiecīgi izmanto gūto pieredzi.

4. Sadarbība:

- 4.1. plāno un īsteno personisko un grupas mērķu sasniegšanai nozīmīgu, cieņpilnu verbālu, neverbālu un digitālu komunikāciju;
- 4.2. piedalās gan viendabīgas, gan neviendabīgas grupas darba procesā, pieņem viedokļu atšķirības, dalībnieku dažādo pieredzi un spējas, prognozē, novērš un risina domstarpības un konfliktus, tostarp digitālajā vidē;
- 4.3. mācību procesā un sabiedriskajā dzīvē apzināti orientējas uz kopīgo labumu un grupai nozīmīgu mērķu sasniegšanu, spēj pārstāvēt savas un respektēt citu intereses, ja grupas un paša vajadzības atšķiras.

5. Pilsoniskā līdzdalība:

- 5.1. skaidro un pamato savu skatījumu par kopsakarībām gan vietējā, gan globālā mērogā, izvērtē individu, sabiedrības un vides mijiedarbību;
- 5.2. balstoties savās vērtībās un cienot citu vērtības, izsvērti izvēlas pasākumus un ikdienas situācijas, kurās iesaistīties un iesaistīt citus, cieņpilni pamatojot savu nostāju; prot atteikties, ja pasākums neatbilst vērtībām, un spēj nepakļauties grupas spiedienam, paliekot saistīts ar tiem, kuriem nepiekrīt;
- 5.3. skaidro savas rīcības sekas un uzņemas par tām atbildību ikdienas situācijās, lokālos un globālos procesos;
- 5.4. patstāvīgi un kopā ar citiem gūst pieredzi, iesaistoties risinājumu meklēšanā un īstenošanā, kas palīdz uzlabot dzīves kvalitāti.

6. Digitālā prasme:

- 6.1. lai īstenotu daudzveidīgas ieceres, mērķtiecīgi izvēlas vai pielāgo un efektīvi izmanto atbilstošas digitālās tehnoloģijas;
- 6.2. analizē digitālās komunikācijas ieguvumus un riskus, atbildīgi uzvedas un komunicē digitālajā vidē atbilstoši savām un citu interesēm;
- 6.3. kritiski analizē mediju radīto realitāti un informācijas ticamību; uzņemas atbildību rīkoties, lai novērstu nekvalitatīva mediju satura radīto ietekmi, un, radot savu mediju saturu, ievēro privātuma, ētiskos un tiesiskos nosacījumus;
- 6.4. analizē un novērtē tehnoloģiju lomu dažādos kontekstos, izvērtē veselīgus un drošus tehnoloģiju lietošanas paradumus, ievēro un pielāgo tos savām vajadzībām, reflektē par savu digitālo identitāti un tās atbilstību savām un sabiedrības interesēm.

4. pielikums

Vērtēšanas uzdevumu piemēri

1. uzdevums

Instrumentu un tehnikas uzskaites sistēma

(1. temats. Objektorientēta programmēšana un ārējās bibliotēkas)

Sasniedzamais rezultāts: veido programmu vienā no objektorientētajām programmēšanas valodām; izmanto programmēšanas valodas un tās bibliotēku dokumentāciju un palīdzības sistēmu, lai patstāvīgi apgūtu citas to piedāvātās iespējas, kas nepieciešamas konkrētās programmatūras izstrādei.

Uzdevums

Izlasī problēmas aprakstu un specifikāciju un, ievērojot objektorientētās programmēšanas valodas pamatprincipus, uzraksti programmu, kas risina doto problēmu!

Problēmas apraksts: Kārlim ir savs uzņēmums, kas piedāvā nomāt dažādus celtniecības instrumentus un tehniku. Līdz šim iznomāto produktu uzskaiti Kārlis veica pierakstu kladē.

Pieaugot klientu un nomas pieprasījumu skaitam, Kārlim kļuvis grūtāk veikt instrumentu un tehnikas uzskaitījumu kladē. Informācija ir kļuvusi nepārskatāma.

Kāds paziņoja Kārlim ieteica automatizēt nomas instrumentu un tehnikas uzskaiti datorā. Kārlis vērsās pēc palīdzības pie programmētājiem, un, vadoties pēc viņa uzskaites principiem pierakstos, tika sastādīta aptuvena specifikācija vēlamo programmatūras risinājumam.

Specifikācija par nepieciešamajām klasēm, konstruktoriem, metodēm, lietotāja saskarni, datu glabāšanu:

Programma sevī ietver klasi (*class*) **Noma**, kurā ir īpašības (*property/field*):

- **Produkta_kategorija** (piemēram, urbjmašīna, zāģis)
- **Produkta_nosaukums** (piemēram, Bosch GSR 18)

- **Tehniskie_raksturojumi** (piemēram, akumulators: 2 gab. / 18 V / 5,0 Ah / Li-ion)
- **Nomas_cena_dienā** (piemēram, 15 EUR)
- **Produkts_pieejams** (piemēram, jā/nē)
- **Nomnieks:_vārds** (piemēram, Andrejs)
- **Nomnieks:_uzvārds** (piemēram, Priede)
- **Nomnieks:_p.k.** (piemēram, 123456-78901)
- **Nomnieks:_tel._numurs** (piemēram, 26543218)
- **Nomas_sākuma_datums** (piemēram, 07.12.2020.)
- **Nomas_beigu_datums** (piemēram, 10.12.2020.)

Konstruktori:

Programma ietver sevī divus konstruktorus:

- **Public Noma()** {**Produkts_pieejams** = **true**} – konstruktors nesaņem nekādus parametrus, bet pēc noklusējuma iestata, ka produkts ir pieejams.
- **Public Noma (string prod._kategorija, string prod._nosaukums, string tehn._raksturojumi)** { **Produkta_kategorija** = **prod._kategorija**; **Produkta_nosaukums** = **prod._nosaukums**; **Tehniskie_raksturojumi** = **tehn._raksturojumi**; **Nomas_cena_dienā** = **nomas_cena**; **Produkts_pieejams** = **true**; } – konstruktors piedāvā savādīt visu nepieciešamo informāciju par produktu un iestata, ka produkts ir pieejams.

Metodes:

- **Nomas_atlikušais_laiks** (metode atgriež atlikušo laiku līdz nomas beigām)
- **Cena_kopā** (metode atgriež kopējo summu par nomu, ņemot vērā, uz cik dienām ir iznomāts produkts)

- **Produkta_info** (metode atgriež kopēju informāciju par produktu)
 - **Produkta_info_print** (metode izprintē datus par produktu, izmantojot standarta bibliotēku teksta datnē, ekrānā vai kādā citā datnē, piemēram, .pdf, izmantojot papildus bibliotēku)
 - **Nomnieks_info** (metode atgriež kopējo informāciju par nomnieku)
 - **Nomnieks_info_print** (metode izprintē datus par nomnieku, izmantojot standarta bibliotēku teksta datnē, ekrānā vai kādā citā datnē, piemēram, .pdf, izmantojot papildus bibliotēku)

Lietotāja saskarne:

Lietotāja saskarne sastāv no trim galvenajām sadaļām (var lietot *MenuStrip*, *Pogas*, *Paneļus*, *ToolStrip*, *TabControl* u. c.):

- Datu ievade (sadaļa, kurā ievada datus par produktu un nomnieku)
- Datu izvade (sadaļa, kurā var izvadīt informāciju par produktu, nomnieku)
- Datu printēšana (atskaīšu izveide) (sadaļa, no kuras var izprintēt informāciju par produktu vai nomnieku atsevišķā teksta vai cita formāta datnē)

Programma piedāvā:

- ievadīt datus par nomas produktu;
- ievadīt datus par nomnieku;
- izvadīt informāciju uz ekrāna, izmantojot izveidotās metodes (atlikušais laiks, cena par nomu kopā, produkta *info*, nomnieka *info*);
- saglabāt datus atskaītes veidā kādā no datnēm (piemēram, .txt standarta bibliotēka vai .pdf ārējā bibliotēka).

Padziļināti:

Programma glabā un ielasa datus no kādas datnes (piemēram, .txt), līdz ar to dati nepazūd pēc programmas aizvēršanas.

2. uzdevums

Tests – objektorientētās programmēšanas pamatprincipi (1. temats. Objektorientēta programmēšana un ārējās bibliotēkas)

Sasniedzamais rezultāts: skaidro objektorientētās programmēšanas pamatprincipus.

Uzdevums

Lasi jautājumus un atbildi vai atzīmē pareizo atbildi (drīkst būt viena pareizā atbilde)!

1. Kāds ir viens no iespējamajiem veidiem, kā uzstādīt objekta mainīgajiem sākuma vērtības?

2. Kurš no konstruktoriem ir pareizi izveidots, ja klases nosaukums ir "Klients"?

- a. `public Klients(string x) { vards = x};`
- b. `public Klienti(string y) { vards = y};`
- c. `public Pircēji(string v) { };`
- d. `public Pirkums(string prece) { prece = prece};`

3. Kurā piemērā ir pareizi definēta dotā klase ar doto konstruktoru?

```
public class Pilsonība
{
    string ? Pilsonis;
    string Vārds;
    string Uzvārds;
    public Pilsonība (string x) { pilsonis = x};
}
```

- a. `Pilsonība pilsonis1 = new Pilsonība ("jā");`
- b. `Pilsonība pilsonis1 = new Pilsonība ();`
- c. `new Pilsonība ("jā");`
- d. `Pilsonība pilsonis1 = "jā";`

4. Ko dara atslēgvārds "this" ("self") programmēšanā?

- a. Atslēgvārds izveido jaunu konstruktoru.
- b. Atslēgvārds izveido jaunu mainīgo.
- c. Atslēgvārds atgriež mainīgā vērtību.
- d. Atslēgvārds norāda saiti uz pašreizējo klasi.

3. uzdevums

Dinamiskā un statiskā publiskā IP

(2. temats. Datortīkls, serveris un droša datubāze)

Sasniedzamais rezultāts: izveido un konfigurē atvērtu vai aizsargātu daudzlietotāju lokālu tīklu, t. sk. veidojot atvērtus vai aizsargātus bezvadu piekļuves punktus. Izveido vienkāršu serveri un konfigurē piekļuvi tam no interneta.

Uzdevums

Atzīmē atbildes, kas raksturo statistikas IP adreses:

- Iespēja precīzāk noteikt fizisko atrašanās vietu.
- Atvieglo DNS servera darbu.
- Atvieglo attālināti nodarbināto darbinieku pieslēgšanos caur VPN vai tml.
- Atvieglo balss un video komunikāciju telekonferencēm, izmantojot VoIP vai tml.
- Aprūrina hakeru uzbrukumus.
- Viegli, automātiski piešķiramas, izmantojot DHCP serveri.
- Likvidējot tīkla iekārtu, tai piešķirtā IP adrese nav manuāli jāatbrīvo (jāpiešķir) citai iekārtai.
- Statiskā publiskā IP adrese ir lētāka par dinamisko.
- Attēlā redzams statistikas adreses piešķiršanas mehānisms.

DHCP Server:	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Start IP Address:	<input type="text" value="192.168.0.100"/>
End IP Address:	<input type="text" value="192.168.1.199"/>
Address Lease Time:	<input type="text" value="120"/> minutes (1~2880 minutes, the default value is 120)
Default Gateway:	<input type="text" value="192.168.0.1"/> (Optional)
Default Domain:	<input type="text"/> (Optional)
Primary DNS:	<input type="text" value="0.0.0.0"/> (Optional)
Secondary DNS:	<input type="text" value="0.0.0.0"/> (Optional)

Iepazīstas ar noklusēto konfigurāciju.

Atver maršrutētāja konfigurācijas paneli.

- Nosaki joslas platumu.
- Nosaki, kādus protokola IEEE 802.11 standartus maršrutētājs atbalsta.
- Nosaki maršrutētāja MAC adresi.
- Nomaini ražotāja iestatīto maršrutētāja lietotājvārdu un paroli uz jaunu un drošu (ieliec ekrānuuzņēmumu).
- Nomaini ražotāja iestatīto WiFi paroli uz jaunu un drošu (ieliec ekrānuuzņēmumu).
- Nosaki, vai ir pieejams maršrutētāja programmatūras atjauninājums (ieliec atbildes ekrānuuzņēmumu).
- Pārliecinies, ka ir ieslēgts WiFi un DHCP (ieliec atbildes ekrānuuzņēmumus).

Iestata WiFi

- Pārliecināties, ka ir ieslēgts DHCP (ieliec atbildes ekrānuuzņēmumus).
- Palielināt izsniedzamo IP adrešu daudzumu līdz 600 (ieliec atbildes ekrānuuzņēmumu).
- Pārliecināties, ka ir ieslēgts WiFi un tas darbojas (ieliec atbildes ekrānuuzņēmumus).

Nevēlamo saišu bloķēšana

- Konfigurēt maršrutētāju, lai lietotājam ar MAC adresi 10-65-30-64-A3-C1 tiktu atļautas tikai tās saites, kas satur atslēgvārdus "Google", "Facebook", "Wikipedia" (ieliec atbildes ekrānuuzņēmumu).

4. uzdevums

Izveido datubāzi un tabulas. Konfigurē laukus un relācijas. (2. temats. Datortīkls, serveris un droša datubāze)

Sasniedzamais rezultāts: plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam.

Uzdevums

Papildini 1.uzdevumu sadaļā “Objektorientēta programmēšana un ārējās bibliotēkas”, izveidojot datubāzi un tabulas informācijas glabāšanai. Konfigurē datubāzes laukus atbilstoši datu tipam un relācijas, lai informācija tabulās būtu savstarpēji saistīta.

Problēmas apraksts: Kārlim ir savs uzņēmums, kas piedāvā nomāt dažādus celtniecības instrumentus un tehniku. Līdz šim iznomāto produktu uzskaiti Kārlis veica pierakstu kladē.

Pieaugot klientu un nomas pieprasījumu skaitam, Kārlim kļuvis grūtāk veikt instrumentu un tehnikas uzskaitījumu kladē. Informācija ir kļuvusi nepārskatāma.

Kāds paziņā Kārlim ieteica automatizēt nomas instrumentu un tehnikas uzskaiti datorā. Kārlis vērsās pēc palīdzības pie programmētājiem, un, vadoties pēc viņa uzskaites principiem pierakstos, tika sastādīta aptuvena specifikācija vēlamajam programmatūras risinājumam.

Specifikācija par nepieciešamajām klasēm, konstruktoriem, metodēm, lietotāja saskarni, datu glabāšanu:

Programma sevī ietver klasi (*class*) **Noma**, kurā ir īpašības (*property/field*):

- **Produkta_kategorija** (piemēram, urbjmašīna, zāģis)
- **Produkta_nosaukums** (piemēram, *Bosch GSR 18*)
- **Tehniskie_raksturojumi** (piemēram, akumulators: 2 gab. / 18 V / 5,0 Ah / *Li-ion*)

- **Nomas_cena_dienā** (piemēram, 15 EUR)
- **Produkts_pieejams** (piemēram, jā/nē)
- **Nomnieks:_vārds** (piemēram, Andrejs)
- **Nomnieks:_s_uzvārds** (piemēram, Priede)
- **Nomnieks:_p.k.** (piemēram, 123456-78901)
- **Nomnieks_tel._numurs** (piemēram, 26543218)
- **Nomas_sākuma_datums** (piemēram, 07.12.2020.)
- **Nomas_beigu_datums** (piemēram, 10.12.2020.)

Pārbauda datubāzes darbību, izmantojot SQL komandas *INSERT, SELECT, DELETE*.

Pārbaudi datubāzes darbību, izmantojot SQL komandas *INSERT, SELECT, DELETE*. Ieliec ekrānuzņēmumus, kas apliecina paveikto!

5. uzdevums

Šifrēšana

(2. temats. Datortīkls, serveris un droša datubāze)

Sasniedzamais rezultāts: izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā.

Uzdevums

Pievieno laukā "Produkta nosaukums" vērtību "Zāģis" un to šifrē ar *MySQL MD5*.

Papildu nosacījums – atlasī šifrēto vērtību ar *SQL* komandu *SELECT*!

Ieliec ekrānuzņēmumus, kas apliecina paveikto!

6. uzdevums

(3. temats. Datu struktūras, programsaskarne (API))

Sasniedzamais rezultāts: meklē un pievieno atvērtā koda bibliotēkas un lieto API (programsaskarni) specializētu funkciju veikšanai, tajā skaitā – sava programmēšanas projekta īstenošana; izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus.

Uzdevums

Uzņēmumam “*Help Me*” ir nepieciešams izstrādāt programsaskarni ar savu ārpakalpojuma serveri. Problēma ir pareiza laika noteikšanā. Serveris atrodas Amerikas Savienotajās Valstīs. Amerikas Savienotās Valstis atrodas citā laika zonā. Jūsu uzdevums ir izveidot vienkāršu programsaskarni jeb API, kurš pieprasījumā atgriež tā brīža pulksteņa laiku un laika zonu, tāpat arī laika zonas starpību. Atbilde jāatgriež *JSON* formātā, lai abās pusēs komunikācija notiktu droši.

Vērtēšanas kritēriji:

1. Izveidots vienkāršs servera puses pieprasījums.
2. Izveidota tukša atbilde no servera.
3. Atbilde no servera ir izveidots kā tukšs *JSON* objekts tā formātā.
4. Atbilde no servera tiek izveidota *JSON* formātā, iekļaujot reāllaika izdruku.
5. Tiek iestrādāti laika zonas starpības parametri pieprasījumā.
6. Serveris atbild ar kļūdu, izmantojot dažādas datu struktūras kļūdas definēšanā, ja līdzī nav padoti laika zonas starpības parametri.
7. Serveris atbild ar izmainītu laika un datuma izdruku atbilstoši iedotajai stundu nobīdei.

7. uzdevums

(3. temats. Datu struktūras, programmsaskarne (API))

Sasniedzamais rezultāts: meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai, tajā skaitā – sava programmēšanas projekta īstenošana; izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus.

Uzdevums

Uzņēmumam “Gudrelis” jāsadarbojas ar ārvalstu universitātēm, bet rodas problēmas ar mājaslapu atcerēšanos un uzskaiti. Tāpēc nepieciešams programmisks risinājums, kas, ievadot valsts nosaukumu angļu valodā, ļauj lietotājam iegūt konkrētu skaitu universitāšu vajadzīgajā valstī, izvadot universitātes šī brīža nosaukumu un mājaslapu.

Programmas izveidē jāparedz, ka datu avots datus var neatgriezt un programmai uz to atbilstoši jāreaģē. Kad dati iegūti, tos saglabā teksta failā ar nosaukumu “Izvads.txt”.

Vērtēšanas kritēriji:

1. Izveidots ievades lauks, kurā notiek teksta ievade.
2. Teksta ievades lauka rezultāts tiek pārbaudīts uz iespējamām kļūdām (tukšumi).
3. Izveidots servera puses pieprasījums uz ārējo universitāšu resursu ar ievadītā lauka datiem.
4. Tiek apstrādāta servera atbilde sev vēlamā formātā.
5. Apstrādātie dati tiek izvadīti datnē Izvads.txt atbilstoši uzdevumā minētajiem datu laukiem.

8. uzdevums

Problēmas izpēte un analīze, darba uzdevuma formulēšana

(4. temats. Problēmas analīze, programmatūras specifikācija un darba plānošana)

Sasniedzamais rezultāts: analizē dažādus ikdienas darba procesus, saskata tajos vai to daļās automatizācijas iespējas un to, kā pasūtītājs formulē darba uzdevumu programmatūras izstrādātājam.

Uzdevums

Uzdevums tiek īstenots trīs posmos:

1. posms – izpēte un dažādu ikdienas procesu analīze ar mērķi saskatīt automatizācijas iespējas;
2. posms – problēmas definēšana un mērķauditorijas noteikšana;
3. posms – darba uzdevuma formulēšana.

Izmantojamā literatūra: Dizaina domāšanas tiešsaistes kurss startdesign.lv

Ikdienā sastopamo situāciju un tajās iespējamo automatizācijas risinājumu piemēri (atbilstoši skolēnu spējām un standartā definētajiem sasniedzamajiem rezultātiem):

1. Automatizācijas risinājuma izveide, izmantojot sensoru datus.

Skolas klašu telpu gaisa kvalitātes novērtēšanas laikā vairākkārt ir secināts, ka fizikas kabinetā lielāko dienas daļu ir paaugstināts CO₂ līmenis. Tas samazina skolēnu spēju koncentrēties un kvalitatīvi uztvert informāciju. Tādējādi telpu var novērtēt kā nepiemērotu mācībām. Lai risinātu radušos situāciju, ir vēlme izveidot automatizētu risinājumu klases vēdināšanas sistēmai, kas ar sensoru palīdzību spētu noteikt skābekļa un CO₂ līmeni klases telpā un nodrošināt optimālus apstākļus mācību procesam. Risinājums nodrošina grafisku lietotāja saskarni skolotāja datorā, paredzot skolotājam atvērt/aizvērt logus, arī izmantojot saskarnes komandpogas. Dati par veikto vēdināšanu tiek uzkrāti datubāzē un ir izgūstami.

2. Lielo datu apstrāde (piemēram, izmantojot portālā <https://data.gov.lv/lv> pieejamos datus).

Cēsu Tūrisma informācijas centrs regulāri atjauno statistiku, kas satur dažādu Cēsu viesu raksturojošu informāciju, piemēram, par viesu izraudzītajiem galamērķiem; valstīm, no kurienes viesi ir ieradusies; ierašanās veidu, viesiem sniegto informācijas tematiku, kanāliem u. tml. (<https://data.gov.lv/dati/lv/dataset/cesu-turisma-centra-sniegtas-informacijas-tematika-2015-2018>). Statistika par vairākiem gadiem ir pieejama lielās Excel tabulās, līdz ar to informāciju ir grūti pārskatīt un analizēt. Tūrisma informācijas centram būtu noderīgi izveidot pārskatu ar dažādām datu atlases iespējām un vizualizācijām, kas konkrētos datus attēlo lietotājam saprotamā un ērtā formātā. Papildus Tūrisma informācijas centra pārstāvji būtu priecīgi, ja varētu izveidot "Cēsu viesu" profilu, kas palīdzētu regulāri vērtēt centra darbību un uzlabot vispieprasītākos pakalpojumus.

8. uzdevums. Problēmas izpēte un analīze, darba uzdevuma formulēšana**1. posms****Uzdevums**

Izvēlies uzdevumam piemērotāko izpētes metodi, veic dažādu ikdienas procesu izpēti un analīzi un izveido apkopojumu par dažādām automatizācijas iespējām tajos!

Izpētes metode: ... (piemēram, aptauja, intervija, novērojumi)

Izpētes metodes piemērotības pamatojums: ...

Izpētes procesa plāns, lai atrastu iespējamās problēmsituācijas:

1. ... (piemēram, veikšu aptauju)
2. ... (piemēram, izmantošu *Google Forms*)
3. ... (piemēram, kā izplatīšu/aizsniegšu respondentus)

...

Izpētes procesa rezultāts:

Nr. p. k.	Ikdienā sastopama situācija	Kādu procesu konkrētajā situācijā ir iespējams automatizēt?	Kāpēc konkrētā procesa automatizācija ir nepieciešama?	Kam un kādi būs ieguvumi no procesa automatizācijas?	Kāpēc process līdz šim nav automatizēts?
1.	Situācija Nr. 1				
2.	Situācija Nr. 2				
3.	Situācija Nr. 3				
...					

8. uzdevums. Problēmas izpēte un analīze, darba uzdevuma formulēšana**2. posms****Uzdevums**

Balstoties uz izpētes procesa rezultātiem, definē vienu risināmo problēmu un raksturo plānotā automatizācijas risinājuma mērķauditoriju!

Problēmas definēšanas metode: ...

Problēmas definēšanas metodes piemērotības pamatojums: ...

Problēmas definēšanas procesa raksturojums:

1. ...

2. ...

3. ...

...

Risināmā problēma: ...

Mērķauditorijas raksturojums:

Kas ir plānotā automatizācijas risinājuma galvenā mērķauditorija?	Kādas specifiskas izvēlētās mērķauditorijas vajadzības jāņem vērā, veidojot konkrēto automatizācijas risinājumu?	Nosauktās vajadzības detalizēts skaidrojums un pamatojums

8. uzdevums. Problēmas izpēte un analīze, darba uzdevuma formulēšana**3. posms****Uzdevums**

Ņemot vērā risināmo problēmu, formulē veicamo darba uzdevumu, kas vēlāk tiks izvērsti programmatūras prasību specifikācijā!

Plānotā automatizācijas risinājuma raksturojums (veicamais darba uzdevums):

Nr. p. k.	Raksturojuma sadaļa	Raksturojums
1.	Kādu problēmu plānotais automatizācijas risinājums risina?	
2.	Kas ir sagaidāmais rezultāts? (automatizācijas risinājuma raksturojums, galveno funkciju uzskaitījums, vienkārša risinājuma darbības skice, kāds tehniskais nodrošinājums būs nepieciešams risinājuma lietotājam)	
3.	Kādas tehnoloģijas potenciāli varētu izmantot automatizācijas risinājuma izstrādē?	

Vērtēšanas kritēriji (8. uzdevums)

1. Izpētes procesa plānošana:
 - a. Izvēlēta atbilstoša izpētes metode.
 - b. Ir saprotami pamatota izvēlētais izpētes metodes piemērotība veicamajam uzdevumam.
 - c. Ir izveidots izpētes plāns, nosaucot galvenos izpētes procesa soļus.
 - d. Katrs no izpētes procesa soļiem ir raksturots, minot, kāds ir tā mērķis un kāds ir sagaidāmais rezultāts.
2. Izpētes procesa rezultātu dokumentēšana:
 - a. Ir identificētas vairākas dažādu jomu ikdienas situācijas, kurās esošos procesus būtu iespējams automatizēt.
 - b. Situācijas raksturotas, demonstrējot iedziļināšanos un izpratni par automatizācijas procesu nepieciešamību un īstenošanu.
3. Problēmas definēšanas process:
 - a. Izvēlēta atbilstoša problēmas definēšanas metode.
 - b. Ir saprotami pamatota izvēlētais problēmas definēšanas metodes piemērotība veicamajam uzdevumam.
 - c. Ir izveidots problēmas definēšanas procesa plāns, nosaucot galvenos procesa soļus.
 - d. Katrs no problēmas definēšanas procesa soļiem ir raksturots, minot, kāds ir tā mērķis un kāds ir sagaidāmais rezultāts.
4. Risināmā problēma:
 - a. Saprotami definēta risināmā problēma (tā iekļauj informāciju par situāciju, risināmo problēmu un dod priekšstatu par iespējamo automatizācijas risinājumu).
5. Mērķauditorijas definēšana un raksturojums:
 - a. Definēta automatizācijas risinājuma galvenā mērķauditorija.
 - b. Uzskaitītas risinājuma mērķauditorijas specifiskās vajadzības.
 - c. Pamatotas un detalizēti skaidrotas uzskaitītās mērķauditorijas specifiskās vajadzības.
6. Darba uzdevuma formulēšana:
 - a. Darba uzdevumā ir pateikts, kādu problēmu plānotais automatizācijas risinājums risina.
 - b. Ir īsi raksturots, kāds ir sagaidāmais rezultāts (automatizācijas risinājuma raksturojums, galveno funkciju uzskaitījums, vienkārša risinājuma darbības skice, kāds tehniskais nodrošinājums būs nepieciešams risinājuma lietotājam);
 - c. Uzskaitītas iespējamās tehnoloģijas, kādas būs nepieciešams pielietot automatizācijas risinājuma izstrādē.

9. uzdevums

Programmatūras specifikācijas izveide

(4. temats. Problēmas analīze, programmatūras specifikācija un darba plānošana)

Sasniedzamais rezultāts: balstoties uz veikto izpēti un definēto darba uzdevumu, izveido programmatūras prasību specifikāciju, izvēlas programmatūras risinājuma izveidei piemērotāko programmatūras izstrādes modeli, programmēšanas valodu un programmatūras izstrādes vidi, pamato veiktās izvēles.

Uzdevums

Izveido programmatūras prasību specifikāciju, balstoties uz iepriekš veikto izpēti un definēto darba uzdevumu!

Programmatūras prasību specifikācijā iekļauj:

- risināmo problēmu un īsu tās raksturojumu;

Piemērs:

Nepieciešams izveidot risinājumu, kas ļautu Tūrisma informācijas centra darbiniekiem apjomīgus datus, kas līdz šim tika glabāti *Excel* tabulās, attēlot ikvienam Tūrisma informācijas centra darbiniekam ērtā un pārskatāmā formātā. Svarīgi ir tas, ka risinājums paredz iespēju veidot vizuāli pievilcīgus un uztveramus pārskatus un atskaides, kā arī regulāri atjaunot aktuālo informāciju.

- izstrādājamā automatizācijas risinājuma galveno mērķi un uzdevumus;

Piemērs:

Automatizācijas risinājuma mērķis ir vienuviet ērtā un pārskatāmā veidā attēlot datus par pilsētas viesiem un viņus raksturojošu informāciju tā, lai Tūrisma informācijas centra darbinieki varētu veidot pārskatus un atskaides dažādos formātos un iegūto informāciju izmantot Tūrisma informācijas centra darbības izvērtēšanai un uzlabošanai.

- risinājuma lietotāju lomas un tiem paredzētās darbības;

Piemērs:

Tūrisma centra vadītājs

Tūrisma centra vadītāja loma ietver visu sistēmā esošo datu apskati dažādos griezumos.

Tūrisma centra vadītājs sagatavo nepieciešamās atskaides *PDF* formātā, vai arī veido cita veida dokumentus (piemēram, prezentācijas ar attēliem un diagrammām), attēlojot no sistēmas iegūto informāciju.

- datu klases;

Piemēri:

- Lietotāji
- Pilsētas viesu veidi
- Pilsētas viesu sniegtās informācijas uzskaites
- Atskaides
- u. c.

- programmatūras darbības aprakstu (funkcijas);

Piemērs:

Tūrisma centra administrators var pievienot un saglabāt jaunus ierakstus par pilsētas viesiem – galvenais galamērķis; valsts, no kuras viesis ir ieradies; veids, kā viesis ir ieradies konkrētajā pilsētā; citi apskates objekti vai pieredzes, ko viesis vēlas iegūt pilsētas apciemojuma laikā. Ērti būtu, ja nepieciešamo informāciju administrators varētu izvēlēties no saraksta un/vai ierakstīt pats, ja tas ir nepieciešams.

- izvēlēto programmatūras izstrādes modeli, tā raksturojumu un izvēles pamatojumu;

- izvēlēto programmēšanas valodu un programmatūras izstrādes vidi, to raksturojumu un izvēles pamatojumu;

- citu informāciju pēc nepieciešamības.

* Izveidoto programmatūras prasību specifikāciju izmantosi turpmākajā darbā, izstrādājot programmatūru.

Vērtēšanas kritēriji

1. Ir saprotami definēta risināmā problēma un dots īss tās raksturojums.
2. Ir definēts izstrādājamā automatizācijas risinājuma galvenais mērķis un uzdevumi.
3. Ir nosauktas visas risinājuma lietotāju lomas un raksturotas tiem paredzētās darbības.
4. Ir dotas visas programmatūras risinājumā paredzētās datu klases.
5. Ir identificētas programmatūras risinājumā nepieciešamās funkcijas un izveidots katras funkcijas apraksts.
6. Dota informācija par izvēlēto programmatūras izstrādes modeli, tā raksturojums un izvēles pamatojums.
7. Dota informācija par izvēlēto programmēšanas valodu un programmatūras izstrādes vidi, to raksturojums un izvēles pamatojums.
8. Programmatūras prasību specifikācijas dokumentam ir izveidots vienots noformējums, kas atbilst dokumentu noformēšanas labās prakses principiem.

10. uzdevums

(5. temats. Programmatūras izstrāde)

Sasniedzamais rezultāts: izstrādā programmatūru izvēlētajā programmēšanas valodā (individuāli vai grupā), izmantojot 9.uzdevumā izveidoto programmatūras prasību specifikāciju. Veic programmatūras vienībtestēšanu.

Uzdevums

Izstrādā programmatūru izvēlētajā programmēšanas valodā (individuāli vai grupā), izmantojot 9. uzdevumā izveidoto programmatūras prasību specifikāciju:

- lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā (piemēram <https://github.com/>):
 - iekārtota vide (visiem projekta dalībniekiem ir piekļuve, izveidotas projekta pamatmapes, pievienots projekta īss apraksts, piemēram, “Automatizācijas risinājums klases telpas vēdināšanai”);
 - izsekojams grupas darba process – redzamas regulāras izmaiņas darbā pie projekta (piemēram, koda uzlabošana, komentāri u. tml.);
 - programmatūras vienības ir apvienotas/integrētas;
- izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam atbilstoši izstrādātajai programmatūras prasību specifikācijai:
 - ja nepieciešams, tiek papildināta vai labota specifikācija;
 - tiek lietoti optimāli datu tipi;
 - kods ir pārskatāms un viegli uztverams – atbilstoši labās prakses principiem;
 - piesaistīta vismaz viena bibliotēka;
- izmanto gatavus algoritmus, pielāgojot un/vai kombinējot tos, un/vai izstrādājot jaunus:
 - dokumentē izmantotos gatavos algoritmus (saglabā saites, pievieno īsu aprakstu par izmantošanas mērķi, konstatē izmantošanas tiesības).

Veic programmatūras vienībtestēšanu:

- lieto projektu un versiju pārvaldības rīkus sadarbībai ar citiem programmatūras izstrādes procesā un tā vadībā (piemēram <https://github.com/>);
- veic akcepttestēšanu atbilstoši izstrādātajai programmatūras prasību specifikācijai:
 - izstrādātājs veic produkta stresstestu (slodzes, nepareizu vērtību ievades pārbaudi u. tml.);
 - klients veic formālu testēšanu, kurā uzsvars tiek likts uz lietotāja vajadzībām un biznesa procesiem (pārbauda, vai sistēma apmierina vai neapmierina akceptēšanas kritērijus, un dod iespēju lietotājiem, klientiem un citiem, kam ir autorizēta pieeja sistēmai, izlemt, vai sistēma ir akceptējama);
- dokumentē testēšanas un pielāgošanas procesu;
- pielāgo programmatūru atbilstoši specifikācijai. Papildina specifikāciju atbilstoši akcepttestēšanas rezultātiem;
- veic programmatūras atklūdošanu un uzlabošanu;
 - novērš testēšanas rezultātā atklātās kļūdas un nepilnības;
 - papildina programmatūru ar nepieciešamajiem uzlabojumiem.

5. pielikums

Mācību satura apguvei izmantojamie mācību līdzekļi un resursi

Izmantošanas nolūks	Mācību līdzekļu veids	Mācību līdzekļu nosaukums
Darbam nepieciešamie tehniskie resursi	IT un ierīces, kuras ir savietojamas ar IT	Dators, multimediju projektors, interneta pieslēgums, maršrutētājs, ierīces demonstrēšanai, printeris
Materiālu atbalsts skolotājiem un skolēniem	Mācību materiāli	Skola2030 mācību līdzeklis
	Programmēšanas valodas	JavaScript, Python, SQL, JAVA, C# un citas
	Materiāli darbam	<p>Datubāzu apguvē var izmantot: w3schools.com [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://www.w3schools.com/sql/default.asp https://www.w3schools.com/sql/sql_join.asp medium.com [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://medium.com/@tushar0618/how-to-create-er-diagram-of-a-database-in-mysql-workbench-209fbf63fd03</p> <p>Kriptogrāfijas apguvē var izmantot: mysql.com [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://dev.mysql.com/doc/refman/5.6/en/encryption-functions.html Kriptogrāfijas bibliotēkas, piemēram, Python <i>bcrypt</i>, <i>Cryptography</i></p> <p>Datortīkla apguvē var izmantot: mikrotik.com [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://help.mikrotik.com/docs/display/ROS/First+Time+Configuration</p> <p>API apguvē var izmantot: w3schools.com [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://www.w3schools.com/whatis/whatis_json.asp Brīvpieejas bibliotēka, piemēram, Python <i>Flask</i> vai Python <i>Bottle</i> Veidojot API izsaukumus, var izmantot bezmaksas API servisu, kuriem nav vajadzīga atslēga. Ja skolēni ir gana droši, tad var izmantot API servisu ar atslēgām un svarīgi skaidrot API atslēgu vajadzību drošības nolūkos.</p>

Izmantošanas nolūks	Mācību līdzekļu veids	Mācību līdzekļu nosaukums
Materiālu atbalsts skolotājiem un skolēniem	Materiāli darbam	<p>Publiskas API, piemēram: <i>Random User Generator</i> [tiešsaiste]. Arron J. Hunt un Keith Armstrong. [skatīts 31.12.2020.]. Pieejams: https://randomuser.me <i>chucknorris.io</i> [tiešsaiste]. Mathias Schilling. [skatīts 31.12.2020.]. Pieejams: https://api.chucknorris.io</p> <p>Mašīnmācīšanās algoritmu apgūvē var izmantot: <i>w3schools.com</i> [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://www.w3schools.com/python/python_ml_getting_started.asp <i>Deep-Image</i> [tiešsaiste]. Teonite. [skatīts 31.12.2020.]. Pieejams: https://deep-image.ai <i>Deep Dream Generator</i> [tiešsaiste]. Deep Dream Generator. [skatīts 31.12.2020.]. Pieejams: https://deepdreamgenerator.com <i>DeepArt</i> [tiešsaiste]. DeepArt UG. [skatīts 31.12.2020.]. Pieejams: https://deepart.io <i>AutoDraw</i> [tiešsaiste]. Dan Motzenbecker un Kyle Phillips. [skatīts 31.12.2020.]. Pieejams: https://www.autodraw.com <i>AI Duet</i> [tiešsaiste]. Yotam Mann. [skatīts 31.12.2020.]. Pieejams: https://experiments.withgoogle.com/ai-duet IBM Watson OpenAI GPT-3 AlphaGo OpenAI Five</p> <p>Bezmaksas tiešsaistes kurss par mākslīgo intelektu: <i>Elements of AI</i> [tiešsaiste]. Reaktor un Helsinku Universitāte. [skatīts 31.12.2020.]. Pieejams: https://course.elementsofai.com/lv</p> <p>Projekta izstrādes posmu apskatē (izpētes, problēmas definēšanas, ideju ģenerēšanas) iespējams izmantot bezmaksas tiešsaistes kursu par dizaina domāšanu: <i>Startdesign.lv</i> [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://www.startdesign.lv/</p>

Izmantošanas nolūks	Mācību līdzekļu veids	Mācību līdzekļu nosaukums
Materiālu atbalsts skolotājiem un skolēniem	Materiāli darbam	<p>Dažādu licenču noteikumu salīdzināšanai var izmantot: choosealicense.com [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://choosealicense.com/licenses/ wikipedia.org [tiešsaiste]. [skatīts 31.12.2020.]. Pieejams: https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licences Choose an open source license [tiešsaiste]. GitHub, Inc. [skatīts 31.12.2020.]. Pieejams: https://choosealicense.com/licenses Software Licenses in Plain English [tiešsaiste]. FOSSA, Inc. [skatīts 31.12.2020.]. Pieejams: https://tldrlegal.com Brīvpieejas bibliotēkas, piemēram, <i>Python Turtle</i>, <i>Python Requests</i>, <i>Tinytag</i>, <i>Pagan</i>, <i>Postman</i>, <i>Chrome Developer Tools</i></p> <p>Citi materiāli: Piemēri lietotāju izpētes metodēm, datu vizualizēšanas piemēri Materiāli (attēli, prezentācijas, videomateriāli u. c.) demonstrēšanai un analizēšanai Programmatūras prasību specifikācijas, lietotāja ceļvežu piemēri Situāciju pētījumi (<i>case study</i>). Atvērto kodu (piemēram, <i>GPLv3</i>, <i>BSD</i>, <i>MIT</i>, <i>Apache</i>) licences Fizisks <i>Kanban</i> sistēmas dēlis Dati uzdevumu veikšanai brīvpiekļuves datubāzēs, piemēram: csdd.lv [tiešsaiste]. Rīga: Ceļu satiksmes drošības direkcija. [skatīts 31.12.2020.]. Pieejams: https://www.csdd.lv Centrālās statistikas pārvaldes datu kopas [tiešsaiste]. Rīga: Centrālās statistikas pārvalde. [skatīts 24.07.2020.]. Pieejams: https://www.csb.gov.lv SQL Tutorial [tiešsaiste]. Andrew Cumming. [skatīts 31.12.2020.]. Pieejams: https://sqlzoo.net Online Interactive SQL Course [tiešsaiste]. SQL Climber. [skatīts 31.12.2020.]. Pieejams: https://www.sqlclimber.com Programmēšanas mācību materiāli [tiešsaiste]. Rīga: LU Starpnozaru izglītības inovāciju centrs. [skatīts 31.12.2020.]. Pieejams: https://siic-lv.github.io/programmesana/vidusskola Krātuvei iespējams pievienot savus materiālus un ieteikumus. Start (IT) mācību līdzekļi [tiešsaiste]. Rīga: IT izglītības fonds Start(IT). [skatīts 31.12.2020.]. Pieejams: https://datorika.startit.lv/datorika https://www.youtube.com/startitlatvia https://kursi.startit.lv/</p>

Izmantošanas nolūks	Mācību līdzekļu veids	Mācību līdzekļu nosaukums
Materiālu atbalsts skolotājiem un skolēniem	Lietotnes	Komunikācijas rīki, piemēram, <i>WhatsApp, Skype, Discord, Slack</i> . Projektu pārvaldības rīki, piemēram, <i>Asana, kalendārs, GitHub, Trello</i> .
Mācību vide		Datorikas kabinets

DOMĀT. DARĪT. ZINĀT.

Valsts izglītības satura centra īstenotā projekta "Kompetenču pieeja mācību saturā" mērķis ir izstrādāt, aprobēt un pēctecīgi ieviest Latvijā tādu vispārējās izglītības saturu un pieeju mācīšanai, lai skolēni gūtu dzīvei 21. gadsimtā nepieciešamās zināšanas, prasmes un attieksmes.

Projekts Nr. 8.3.1.1/16/I/002 Kompetenču pieeja mācību saturā



NACIONĀLAIS
ATTĪSTĪBAS
PLĀNS 2020



EIROPAS SAVIENĪBA
Eiropas Sociālais
fonds

I E G U L D Ī J U M S T A V Ā N Ā K O T N Ē