2024./2025. m. g.

centralizētā eksāmena

“PROGRAMMĒŠANA

(AUGSTĀKAIS MĀCĪBU SATURA

APGUVES LĪMENIS)”

programma

SATURS

1. EKSĀMENA MĒRĶIS UN ADRESĀTS

Centralizētā eksāmena (turpmāk – eksāmens) mērķis ir novērtēt skolēnu sniegumu mācību priekšmetā atbilstoši Ministru kabineta 2019. gada 3. septembra noteikumiem Nr. 416 “Noteikumi par valsts vispārējās vidējās izglītības standartu un vispārējās vidējās izglītības programmu paraugiem” (turpmāk – standarts) un standarta 7. pielikumam “Plānotie skolēnam sasniedzamie rezultāti tehnoloģiju mācību jomā” optimālajā un augstākajā mācību satura apguves līmenī, identificēt un izvērtēt, cik lielā mērā ir apgūti plānotie sasniedzamie rezultāti (turpmāk – SR).

Eksāmena adresāts – skolēni, kuri ir apguvuši tehnoloģiju mācību jomas SR optimālajā un augstākajā mācību satura apguves līmenī atbilstoši mācību priekšmetu kursiem “Programmēšana I” un “Programmēšana II” (standarta 9. pielikums).

2. VĒRTĒŠANAS SATURS

Eksāmena vērtēšanas saturu raksturo trīs kategorijas:

1. SR veids un grupa;
2. Satura modulis;
3. Izziņas darbības līmenis.

Tas nozīmē, ka katru eksāmena testelementu[1] raksturo noteikts SR veids un grupa, satura modulis un izziņas darbības līmenis.

2.1. SR veidi un grupas

Skolēnam plānoti triju veidu SR:

1. Zināšanas un izpratne;
2. Prasmju grupas;
3. Zināšanu, izpratnes, prasmju un ieradumu kombinācijas.

Katram SR veidam ir norādītas SR grupas (sk. 1. tabulu), kuras apkopo standartā noteikto mācību saturu un tiek pārbaudītas un/vai mērītas eksāmenā.

Zināšanu un izpratnes pārbaudei eksāmenā iekļauti uzdevumi, kuros skolēni:

1. Atpazīst pamatalgoritmus un izprot to darbības principus;
2. Salīdzina programmatūras izstrādes modeļus un skaidro programmēšanas jēdzienus.

Prasmju grupu apguvi raksturo trīs SR grupas:

1. Izstrādā programmatūru atbilstoši tās prasību specifikācijai un/vai citiem norādījumiem;
2. Lieto prasmes darbā ar informāciju, piemēram, projektējot un veidojot datubāzi, izstrādājot programmatūru, jēgpilni izmantojot esošo bibliotēku piedāvātās iespējas un programmēšanas valodas dokumentāciju;
3. Ievēro programmēšanas labās prakses principus.

[1] Testelements ir uzdevums vai uzdevuma daļa, kas veidota, lai pēc kritērijiem vērtētu kādu konkrētu skolēnu darbības aspektu.

Zināšanu, izpratnes, prasmju un ieradumu kombināciju pārbaudei izmanto uzdevumus, kuri skolēniem liek atrisināt problēmu jaunā situācijā, turklāt to saturs var būt no jebkura satura moduļa (sk. 2. tabulu).

1. tabula

SR veidi, grupas un to īpatsvari eksāmenā

<table>
<tr><th colspan="2">SR veids un grupa</th><th colspan="2">Īpatsvars, %</th></tr>
<tr><td rowspan="2">Zināšanas un izpratne</td><td>Atpazīst pamatalgoritmus un izprot to darbības principus.</td><td colspan="2" rowspan="2">10 ± 5</td></tr>
<tr><td>Salīdzina programmatūras izstrādes modeļus un skaidro programmēšanas jēdzienus.</td></tr>
<tr><td rowspan="3">Prasmju grupas</td><td>Izstrādā programmatūru atbilstoši tās prasību specifikācijai un/vai citiem norādījumiem.</td><td>57 ± 5</td><td rowspan="3">68 ± 4</td></tr>
<tr><td>Lieto prasmes darbā ar informāciju, piemēram, projektējot un veidojot datubāzi, izstrādājot programmatūru, jēgpilni izmantojot esošo bibliotēku piedāvātās iespējas un programmēšanas valodas dokumentāciju.</td><td>8 ± 3</td></tr>
<tr><td>Ievēro programmēšanas labās prakses principus.</td><td>3 ± 1</td></tr>
<tr><td>Zināšanu, izpratnes, prasmju un ieradumu kombinācijas</td><td>Kombinē savā darbā vairākus programmatūras dzīvescikla posmus.</td><td colspan="2">16 ± 4</td></tr>
</table>

2.2. Satura moduļi

Satura moduļus un to īpatsvarus eksāmenā var aplūkot 2. tabulā.

2. tabula

Satura moduļi un to īpatsvari eksāmenā

Satura modulis	Īpatsvars, %
Programmatūras dzīvescikls	17 ± 5
Datubāzes izstrāde un izmantošana	17 ± 5
Objektorientētā programmēšana un ārējās bibliotēkas	33 ± 5
Datu struktūras un programmsaskarnes	33 ± 5

Eksāmena izstrādes procesā tiek saskaņots un nodrošināts sadaļu procentuālais sadalījums gan SR veidiem un grupām, gan satura moduļiem.

2.3. Izziņas darbības līmeņi

Eksāmenā iekļautie uzdevumi grupēti četros izziņas darbības līmeņos, un to līmeņa noteikšanai izmanto SOLO jeb novēroto mācīšanās rezultātu taksonomiju. SOLO taksonomijā skolēna sniegums tiek raksturots, analizējot ideju jeb struktūrelementu skaitu un saišu kvalitāti starp šiem struktūrelementiem. Vispārīgs izziņas darbības līmeņu apraksts, kas piemērots eksāmenā, apkopots 3. tabulā.

3. tabula

Izziņas darbības līmeņu raksturojumi un īpatsvari eksāmenā

Izziņas darbības līmenis un tā apraksts		**Īpatsvars, %**
I	Atceras, nosauc atsevišķas idejas, izpilda īsas procedūras	25 ± 5
II	Saista, skaidro, lieto zināšanas vai prasmes pazīstamās situācijās	25 ± 5
III	Saista, skaidro, lieto zināšanas vai prasmes jaunās situācijās, demonstrējot izpratni	30 ± 5
IV	Lieto zināšanas un prasmes situācijās ar augstu kompleksuma pakāpi	20 ± 5

Katram līmenim atbilstošo uzdevumu īpatsvars noteikts, ievērojot eksāmena mērķi un šādus galvenos vērtēšanas principus:

1. Skolēnu grupai ar zemu un vidēju snieguma līmeni programmēšanā dota iespēja apliecināt savas zināšanas un prasmes pietiekami plašā satura jautājumu lokā, t. sk. uzdevumos, kas mēra zināšanu, izpratnes, prasmju un ieradumu kompleksu lietojumu;
2. SR veidu “Zināšanas un izpratne” un “Prasmju grupas” vērtēšanai iekļauti testelementi, kas atbilst II un III izziņas darbības līmenim, tādējādi akcentējot izpratnes vērtēšanu;
3. Visi SR veida “Zināšanu, izpratnes, prasmju un ieradumu kombinācijas” vērtēšanai iekļautie uzdevumi ietver vismaz III izziņas darbības līmenim atbilstošus testelementus.

3. EKSĀMENA DARBA UZBŪVE

Eksāmena darba uzbūvi, daļu īpatsvarus un izpildei paredzētos laikus var aplūkot 4. tabulā.

4. tabula

Eksāmena daļu īpatsvari un izpildei paredzētie laiki

Daļa	Uzdevumu skaits	Maksimālais punktu skaits	Daļas īpatsvars, %	Izpildes laiks, min.
1. Programmatūras dzīvescikls	8	20	17	30
2. Datubāzes izstrāde un izmantošana	2	20	17	40
3. Objektorientētā programmēšana un ārējās bibliotēkas	1	38	31	80
4. Datu struktūras un programmsaskarnes	4	42 = 28 * 1,5	35	80
Kopā	15	120	100	230

1. daļā ir iekļauti uzdevumi, kuros skolēnam atbilstoši dotajam aprakstam jāveic problēmsituācijas izpēte un topošās programmatūras izstrādes plānošana. Tie var būt dažāda veida uzdevumi, piemēram, izvērsto atbilžu uzdevumi, informācijas apkopošana un apstrāde, problēmas analīze. 2. daļā ir iekļauti uzdevumi, kuros skolēnam atbilstoši dotajam aprakstam jāveic datubāzes izstrāde un vaicājumu izpilde. Savukārt 3. un 4. daļā ir iekļauti uzdevumi, kuros skolēnam atbilstoši dotajiem nosacījumiem jāizstrādā dažādas programmatūras.

2., 3. un 4. daļā ir uzdevumi, kuros netieši tiek vērtēta programmēšanas labās prakses principu ievērošana (sk. 1. pielikumu). Izpildot uzdevumus šajās daļās, skolēns drīkst izmantot jebkuru lokāli pieejamo datubāzu vadības sistēmu, kā arī programmēšanas valodas un izstrādes vides, kuras ir apguvis un ar kurām ir darbojies mācību procesā.

Katra no eksāmena daļām var saturēt visu iepriekš minēto veidu uzdevumus. Katra veida uzdevumu skaits un īpatsvars daļā un eksāmenā kopumā nav noteikts. Uzdevuma veida izvēli nosaka atbilstība SR, ko tas pārbauda.

4. EKSĀMENA PIEKĻUVES NOSACĪJUMI

Eksāmenu var kārtot jebkurš skolēns, kas izstrādājis programmatūras produktu un tā dokumentāciju, īstenojot visus programmatūras izstrādes dzīvescikla posmus. Izstrādātās programmatūras dokumentācijas jeb piekļuves materiālu apjomam jābūt līdz 15 (piecpadsmit) A4 formāta lapām, neskaitot titullapu, satura rādītāju un pielikumu(-us). Piekļuves nosacījumi var tikt izpildīti, apgūstot “Programmēšana II. Padziļinātā kursa programmas paraugs vispārējai vidējai izglītībai” 4. tematu “Problēmas analīze, programmatūras specifikācija un darba plānošana” un 5. tematu “Programmatūras izstrāde”. Dokumentācijai ir jāsatur visas tālāk norādītās nodaļas. Katrā nodaļā dokumentācijas saturam jāatbilst atbilstošā snieguma līmeņu apraksta (turpmāk – SLA) vismaz 2. līmenim, t. i., “turpina apgūt” (sk. 3. pielikumu), un programmatūras produktam jāatbilst tālāk norādītajām minimālajām prasībām.

Piekļuves materiālus **no 2025. gada 3. marta, bet ne vēlāk kā 8 (astoņas) nedēļas pirms eksāmena norises dienas, t. i., līdz 2025. gada 22. aprīlim, skolēnam jāaugšupielādē Valsts pārbaudījumu informācijas sistēmā** (turpmāk – VPS). Kārtība, kādā ir augšupielādējami piekļuves materiāli, atrodama VISC Lietotāju atbalsta dienesta tīmekļvietnē.

Piekļuves materiālus izglītības iestādes skolotājs vērtē ballēs atbilstoši izglītības iestādes saistošai izglītojamo mācību sasniegumu vērtēšanas kārtībai un **ne vēlāk kā 6 (sešas) nedēļas pirms eksāmena norises dienas, t. i., līdz 2025. gada 6. maijam, vērtējumu ievada VPS**.

Izglītojamais eksāmenu drīkst kārtot, ja vērtējums par piekļuves materiālu nav zemāks par 4 (četrām) ballēm. Atbilstoši Ministru kabineta 2022. gada 5. jūlija noteikumu Nr. 398 “Noteikumi par centralizēto eksāmenu saturu un norises kārtību” 35. punktam izglītojamie, kuri eksāmenu kārto augstskolā, piekļuves materiālus neiesniedz.

Iesniedzamajai dokumentācijai ir jāsatur šādas nodaļas (sk. 3. un 4. pielikumu):

1. **Problēmas izpēte un analīze – izpētes metodes izvēle un pamatojums, izpētes procesa apraksts, izpētes datu apkopojums;**
   SR: Analizē dažādus ikdienas darba procesus, saskata tajos vai to daļās automatizācijas iespējas. (T.A.2.4.1.)

2. **Programmatūras prasību specifikācija – risinājuma mērķauditorijas izvēle un tās raksturojums, programmatūras un tās funkciju apraksts, programmatūras skice;**

   SR: Ikdienas darba procesos vai to daļās saskata automatizācijas iespējas un to, kā pasūtītājs formulē darba uzdevumu programmatūras izstrādātājam. (T.A.2.4.1.)

   SR: Sastāda vienkāršotu programmatūras prasību specifikāciju atbilstoši konkrētajam uzdevumam, izvērtējot mērķauditorijas specifiku un vajadzības. (T.A.2.4.4.)

   SR: Veic vienkāršotu programmatūras projektēšanu (t. sk. lietotāju saskarnes un vienkāršotu datu modeļa izveidi) atbilstoši programmatūras prasību specifikācijā izvirzītajām funkcionālajām un nefunkcionālajām prasībām. (T.A.2.4.5.)

   SR: Izvēlas programmēšanas valodu un programmatūras izstrādes vidi programmatūras izstrādē atbilstoši uzdevuma specifikai, pamato savu izvēli. (T.A.2.4.12.)

3. **Programmatūras izstrādes plāns;**

   SR: Salīdzina un izvēlas piemērotāko programmatūras izstrādes modeli konkrētā uzdevuma atrisināšanai, pamato savu izvēli. (T.A.2.4.2.)

   SR: Veic vienkāršotu (bez darbietilpības novērtējuma) programmatūras izstrādes plānošanu. (T.A.2.4.5.)

4. **Atkļūdošanas un akcepttestēšanas pārskats;**

   SR: 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.)

5. **Lietotāja ceļvedis;**

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

6. **Programmatūras kods (pievienojams dokumentācijas pielikumā), kas veidots, ievērojot labās prakses principus.**

   SR: Veido programmu, ievērojot labās prakses pieredzi tās pieraksta strukturēšanā un komentāru veidošanā. (T.A.2.4.8.)

Iesniedzamajam programmatūras produktam ir jāapmierina šādas minimālās prasības (sk. arī 3. pielikumu):

1. **Ir izmantota datubāze ar vairākām saistītām tabulām;**

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

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

2. **Ir izmantota vismaz viena bibliotēka un/vai programmsaskarne (turpmāk – API);**

   SR: 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.)

   SR: Meklē un pievieno atvērtā koda bibliotēkas, kā arī izmanto API specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai. (T.A.2.4.11.)

   SR: Izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam. (T.A.2.4.13.)

3. **Ir izmantota vismaz viena no datu struktūrām;**

   SR: Izmanto dažādas datu struktūras un ar tām saistītos pamatalgoritmus. (T.A.2.4.14.)

   SR: 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. (T.A.2.4.19.)

4. **Ir realizēta lietotāja piekļuves un datu aizsardzība.**

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

5. NEPIECIEŠAMO RESURSU NODROŠINĀJUMS

1. Darbs veicams, izmantojot izglītības iestādes datoru.
2. Pildot visas eksāmena daļas, kārtotāji drīkst izmantot tikai operētājsistēmas pamatprogrammas un biroja programmatūru (piemēram, kalkulatoru, datņu pārlūku, izklājlapu redaktoru), kā arī interneta pārlūkprogrammu (tikai 7. nodaļā uzskaitītajām atļautajām darbībām).
3. Pildot eksāmena 2., 3. un 4. daļu, papildus 2. punktā uzskaitītajam, kārtotāji drīkst izmantot:
    a. Izvēlēto lokāli pieejamo datubāzu vadības sistēmu (piemēram, *DB Browser for SQLite*, *Microsoft SQL Server* u. c.);
    b. Izvēlētos programmatūras izstrādes rīkus: integrētās izstrādes vidi (piemēram, *Visual Studio*, *PyCharm*, *Eclipse*, *NetBeans*, *IntelliJ* u. c.), koda redaktoru (piemēram, *Visual Studio Code*, *Notepad*++ u. c.) un/vai to atsevišķos komponentus.

6. VĒRTĒŠANAS KĀRTĪBA UN KRITĒRIJI

6.1. Vērtēšanas kārtība

Katrā uzdevumā ir norādīts maksimālais iegūstamo punktu skaits. Eksāmena vērtētājam ir pieejami kritēriji, pēc kuriem nosaka punktu skaitu, ko skolēns ieguvis. Skolēna rezultātus eksāmenā – iegūto punktu summu visā darbā un iegūto punktu summu katrā daļā – izsaka procentuālā novērtējumā. Vidēji 20 % eksāmenā iekļauto testelementu reprezentē minimālo prasību kopumu, kas:

- Atbilst I un II izziņas darbības līmeņu uzdevumu apjomam – atpazīst pamatalgoritmus, nosauc piemērotāko izpētes metodi, izpilda vienkāršas darbības, nodrošina datu ievadizvadi, kombinē zināšanas un prasmes standarta situācijās utt.;
- Ir kursa pamatprasmes – problēmsituācijas izpēte un analizēšana, programmatūras izstrādes plānošana, datu modelēšana un datubāzes izstrāde, dažādu datu struktūru un ar tiem saistīto pamatalgoritmu izmantošana, programmu veidošana kādā no objektorientētajām programmēšanas valodām, programmsaskarņu un ārējo bibliotēku pielietošana specializētu funkciju veikšanai, – kuru sniegums ir “sācis apgūt” līmenī.

Skolēnu sniegums un atbildes saskaņā ar izstrādātajiem un pēc norises standartizētajiem vērtēšanas kritērijiem tiek vērtētas izvērsto atbilžu uzdevumos. Skolēni aiz katra uzdevuma formulējuma sniedz atbildi (raksta tiešsaistes teksta lodziņā un/vai augšupielādē datnes) tam paredzētajā vietā.

Atbilstoši standarta 21.[1] punktam **eksāmenā vērtējums nav iegūts, ja darba kopvērtējums 2024./2025. mācību gadā ir mazāks nekā 20 %**.

6.2. Vērtēšanas kritēriji

Skolēnu sniegumu eksāmenā vērtē pēc kritērijiem, kas var būt izteikti kā katram punktam atbilstošu atbilžu, darbību, rezultāta apraksts. Programmēšanas labās prakses principu ievērošana tiek vērtēta pēc SLA, katram līmenim piešķirot noteiktu punktu skaitu (sk. 1. pielikumu).

Eksāmenā biežāk lietoto rīcības vārdu skaidrojums (2. pielikums) ir iekļauts, lai veidotu vienotu skolotāju un skolēnu izpratni par šo vārdu nozīmi un tai atbilstošu skolēna sniegumu mācību procesā un eksāmenā.

7. PALĪGLĪDZEKĻI, KURUS ATĻAUTS IZMANTOT EKSĀMENA LAIKĀ

1. Eksāmena laikā kārtotājiem ir iespēja izmantot izvēlēto programmēšanas valodu un programmatūras izstrādes rīku dokumentācijas.
2. Tāpat eksāmena kārtotājiem ir iespēja bez autorizācijas izmantot tīmekļvietni https://www.w3schools.com papildinformācijas meklēšanai un iespējamo kļūdu labojumiem.
3. Pie izglītojamajiem un personām, kuras piedalās eksāmena nodrošināšanā, no brīža, kad viņiem ir pieejams eksāmena materiāls, līdz eksāmena norises beigām nedrīkst atrasties ierīces (planšetdators, piezīmjdators, viedtālrunis, viedpulkstenis u. c. saziņas un informācijas apmaiņas līdzekļi), kuras nav paredzētas eksāmena norises darbību laikos.

PIELIKUMI

1. pielikums

Programmēšanas labās prakses principu piemēri

Daži programmēšanas labās prakses principu piemēri ir šādi:

1. Katru atsevišķu priekšrakstu raksta jaunā rindā;
2. Koda loģiskās daļas (piemēram, zarošanos, ciklu, masīvu) savstarpēji atdala ar tukšu rindu;
3. Izmanto atkāpes, lai vizualizētu priekšrakstu vai struktūru iekļaušanu citās struktūrās;
4. Izvairās no garām koda rindām, izmanto pārnesi jaunā rindā, atvieglojot koda lasīšanu;
5. Mainīgo, funkciju u. c. nosaukumus veido jēgpilnus, atvieglojot koda uztveri (piemēram, perimetru apzīmējot nevis ar “p”, bet “perim” vai “perimetrs”);
6. Ar komentāriem skaidro programmatūras koda loģiskās daļas (piemēram, zarošanos, ciklu, masīvu), to lomu programmatūrā (piemēram, “datu izvade”).

Līmenis / **Kritērijs**	**Sācis apgūt**	**Turpina apgūt**	**Apguvis**	**Apguvis padziļināti**
Veido programmatūru, ievērojot labās prakses principus tās pieraksta strukturēšanā un komentāru veidošanā	Lielākajā daļā programmatūras koda lieto labās prakses principus, bet dara to nekonsekventi vai daļēji korekti.	Labās prakses principus programmatūras kodā lieto kopumā korekti un konsekventi, pieļaujot dažas neprecizitātes.	Labās prakses principus programmatūras kodā lieto korekti un konsekventi.	Labās prakses principus programmatūras kodā lieto vienmēr korekti un konsekventi, patstāvīgi piedāvā, ievieš un izmanto līdzīgos principus.

2. pielikums

Rīcības vārdu skaidrojums

Eksāmenā biežāk lietoto vai mācību procesā nereti dažādi interpretēto rīcības vārdu skaidrojums (sk. 5. tabulu) ir iekļauts, lai veidotu vienotu skolotāju un skolēnu izpratni par šo vārdu nozīmi un tai atbilstošu skolēna sniegumu mācību procesā un eksāmenā.

5. tabula

Biežāk lietotie rīcības vārdi un to skaidrojumi

Rīcības vārds	Skaidrojums
Analizē	Sīki, pamatīgi pēta, raksturo, piemēram, algoritmu, problēmas aprakstu.
Apvieno	Izveido vienu veselu (no atsevišķām vienībām, daļām, piemēram, apvieno programmatūras daļas).
Atpazīst	Pazīst, identificē (kādu/ko pēc noteiktām pazīmēm, parasti starp līdzīgiem, piemēram, OOP darbības principus).
Formulē	Izsaka (domu, atzinumu, lēmumu, jautājumu), piemēram, kādu programmatūras produktu nepieciešams izstrādāt.
Identificē	Nosaka, pazīst; konstatē (ko/kādu pēc kādām raksturīgām pazīmēm, piemēram, lomu sadalījumu projekta komandā).
Ievēro	Rīkojas saskaņā (ar ieteikumiem, norādījumiem u. tml.); iegaumē un izpilda (noteikumus, prasības u. tml.), piemēram, labās prakses principus, rakstot programmatūras kodu.
Integrē	Iekļauj kādā vienībā; papildina, pilnveido, izveido par veselu vienību (piemēram, programmatūras daļas vienā veselumā).
Izmanto	Lieto (ko) par pamatu, materiālu (piemēram, kā noteikta gatavošanai, izveidošanai), piemēram, programmēšanas valodas dokumentāciju.
Izprot	Pilnīgi izzina un saprot, piemēram, algoritma darbību.
Izveido/Izstrādā	Ar mērķtiecīgu darbību panāk vai organizē, ka, piemēram, datubāze (vai kas cits) iegūst vēlamo veidu un atbilst noteiktām prasībām; rada prasīto.

Rīcības vārds	Skaidrojums
Izvēlas	Iegūst lietošanai, atrod, izraugās izmantošanai no kāda kopuma (ko piemērotu, atbilstošu, piemēram, programmatūras izstrādes modeli savam projektam atbilstoši problēmai, nosacījumiem u. tml.).
Izvērtē	Novērtē kādu kopumu, parasti pa tā sastāvdaļām, detaļām; katru faktu, notikumu u. tml. noteiktā kopsakarībā (piemēram, mērķauditorijas specifiku un vajadzības).
Kombinē	Veido noteiktā veselumā (no atsevišķām sastāvdaļām), piemēram, vairākas algoritma daļas, savienojot vienā veselumā.
Konfigurē	Pielāgo sistēmu (piemēram, datortīklu), uzstādot aparatūru un programmatūru (piemēram, maršrutētāja); noteiktā veidā uzstāda.
Lieto	Rīkojas (ar materiālu, koda fragmentu) mērķtiecīgi, pārdomāti – tā, lai gūtu labumu; izmanto (piemēram, lieto API).
Meklē	Cenšas (skatoties u. tml.) dabūt (ko vajadzīgu, nepieciešamu u. tml., piemēram, atvērtā koda bibliotēkas).
Pamato	Min faktus, cēloņus, loģiskus slēdzienus u. tml., kas pierāda (kā) patiesumu, nepieciešamību (piemēram, izvēli lietot konkrētu izpētes metodi lietotāju vajadzību noskaidrošanai).
Pārbauda	Pārliecinās, vai (kas) atrodas noteiktā stāvoklī, kārtībā (piemēram, programmatūras veiktspēju, programmatūras kodu), arī – vai kas ir noticis, izdarīts (piemēram, iegūts pareizais rezultāts, izpildot programmu).
Pielāgo	Veido, izveido (ko, piemēram, algoritmu) atbilstoši izmantošanas mērķim, apstākļiem.
Piemēro	Padara atbilstošu (piemēram, apstākļiem, iespējām programmatūras licenci).
Pievieno	Pieliek (pie kā, kam klāt, piemēram, atvērtā koda bibliotēkas).
Plāno	Veido (kāda objekta, piemēram, datubāzes) plānu, skici, shēmu. Iepriekš sadala (laiku) pa posmiem (kā veikšanai, piemēram, projekta īstenošanai).
Raksturo	Nosaka, apraksta, vērtē (kā) raksturīgās īpašības, pazīmes (piemēram, mašīnmācīšanās algoritmus).

Rīcības vārds	Skaidrojums
Salīdzina	Pēta, analizē, novēro u. tml. (divus vai vairākus priekšmetus, blokshēmas u. tml.), lai noteiktu (to) kopīgās vai atšķirīgās īpašības, pazīmes.
Saskata	Skatoties, pilnīgi precīzi uztver (ko, piemēram, automatizācijas iespējas), atšķir, pazīst (ko); skatoties atklāj, konstatē (ko), secina (ko).
Sastāda	Izveido (piemēram, programmatūras prasību specifikāciju), apvienojot materiālus; uzraksta noteiktā formā, izmantojot kādus materiālus.
Skaidro	Stāstot, rakstot, arī rādot panāk, ka kādam (kas) kļūst zināms, saprotams (piemēram, programmatūras koda darbības principi).
Veic	Ar savu darbību, rīcību panāk, ka īstenojas kāda norise vai tiek sasniegts kāds mērķis (piemēram, veic akcepttestēšanu).

Rīcības vārdu skaidrojumi ar uzdevumu piemēriem ir minēti 6. tabulā. Šīs tabulas virsrakstu skaidrojums:

- SR jeb sasniedzamais rezultāts – atbilstoši standartam;
- SOLO jeb novēroto mācīšanās rezultātu taksonomija – maksimālais SR pārbaudes dziļums atbilstoši SOLO taksonomijai, eksāmenā SR var pārbaudīt zemākā, bet ne augstākā SOLO līmenī;
- Rīcības vārdi – SR iekļautie darbības (rīcības) vārdi;
- Uzdevumu piemēri – rīcības vārdu interpretācija dažādu veidu uzdevumu piemēros (piemēru saraksts nav pilnīgs).

6. tabula

Rīcības vārdu skaidrojumi ar uzdevumu piemēriem

SR	SOLO	Rīcības vārdi	Uzdevumu piemēri
T.A.2.3.2.	III	Plāno	• Izveido datubāzi ar vairākām saistītām tabulām atbilstoši dotajam aprakstam.
T.A.2.4.1.	III	Analizē un saskata	• Iepazīstas ar problēmas aprakstu teksta formātā un uzskaita dažādas procesu automatizācijas iespējas. • Ir dots problēmas apraksts un varianti automatizācijas risinājumiem – identificē, kuri no tiem atbilst dotajai situācijai un kuri neatbilst.
		Formulē	• Balstoties uz dotu problēmas aprakstu, formulē, kādu programmatūras produktu nepieciešams izstrādāt un/vai kādus uzlabojumus esošajā produktā nepieciešams ieviest. • Uzskaita un apraksta tehniskās prasības programmatūras produkta izstrādei.
T.A.2.4.2.	III	Salīdzina	• Atpazīst un raksturo vienu vai vairākus programmatūras izstrādes modeļus. • Uzskaita priekšrocības un trūkumus konkrētu programmatūras izstrādes modeļu izmantošanā un salīdzina divus vai vairākus programmatūras izstrādes modeļus.

SR	SOLO	Rīcības vārdi	Uzdevumu piemēri
		Izvēlas un pamato	• Ņemot vērā problēmas aprakstu, izvēlas un pamato, kuru programmatūras izstrādes modeli izmantot. • Plāno projekta izstrādes procesu, iekļaujot tajā programmatūras izstrādes modeli un paskaidrojumu par tā izvēli.
T.A.2.4.4.	IV	Izvērtē	• Definē un raksturo mērķauditoriju atbilstoši dotajam uzdevumam. • Atbilstoši dotajai problēmai identificē un apkopo informāciju par mērķauditorijas specifiku un vajadzībām. • Pielāgo programmatūras produkta ideju dažādām mērķauditorijām.
		Izveido [programmatūras prasību specifikāciju]	• Balstoties uz doto problēmas aprakstu, izstrādā programmatūras prasību specifikāciju vai kādas tās daļas. • Papildina iesākto programmatūras prasību specifikāciju.
		Veic [akcepttestēšanu]	• Veic akcepttestēšanu atbilstoši izstrādātajām programmatūras prasībām. • Fiksē atbilstības un neatbilstības.
T.A.2.4.8.	IV	Ievēro [labās prakses principus]	• Ievēro labās prakses principus, izstrādājot programmatūras kodu.
T.A.2.4.10.	IV	Analizē un izmanto [piedāvātās iespējas]	• Izstrādājot programmatūru, jēgpilni izmanto esošo bibliotēku piedāvātās iespējas. • Lieto esošo klašu konstruktorus un to metodes, tādējādi saīsinot esošo programmatūras kodu un uzlabojot izstrādājamās programmatūras darba efektivitāti. • Analizē doto bibliotēku dokumentāciju un palīdzības sistēmu informāciju, lai atrastu un pielietotu atbilstošu bibliotēku problēmas risināšanā. • Izmanto programmēšanas valodas dokumentāciju, lai apgūtu programmēšanas valodas iespējas dotās problēmas risināšanā.

SR	SOLO	Rīcības vārdi	Uzdevumu piemēri
T.A.2.4.11.	IV	Meklē	• Atrod atvērtā koda bibliotēkas, kas veic nepieciešamās funkcijas, un pamato savu izvēli.
		Pievieno	• Pievieno paša atrastās un/vai dotās bibliotēkas.
		Lieto	• Pielieto bibliotēkas un/vai API dotā uzdevuma risinājumam.
T.A.2.4.14.	III	Izmanto	• Izveido risinājumu dotajam uzdevumam, izmantojot dažādas datu struktūras.
T.A.2.4.15.	IV	Veido [programmatūras produktu]	• Veido programmatūras produktu, balstoties uz OOP pamatprincipiem – izveido un pielieto klases un to īpašības, metodes un konstruktorus.
T.A.2.4.17.	IV	Izveido	• Izveido programmatūras produktu.

3. pielikums

Snieguma līmeņu apraksts projekta izstrādes dokumentācijai

Šis SLA var tikt izmantots, izstrādājot projektus vai to daļas optimālajā un padziļinātajā līmenī, veicot pašnovērtējumu, summatīvo vai formatīvo vērtēšanu mācību procesā, kā arī eksāmena piekļuves nosacījumu izpildes vērtēšanai. SLA programmatūras koda veidošanas posmam sk. atsevišķi – 4. pielikumā.

Līmenis / Kritērijs	Sācis apgūt	Turpina apgūt	Apguvis	Apguvis padziļināti
1. Problēmas izpēte un analīze – izpētes metodes izvēle un pamatojums, izpētes procesa apraksts, izpētes datu apkopojums	Izpētes procesa soļi ir sajauktā secībā un/vai neveido loģisku procesu (iztrūkst kāds būtisks posms, nav sākuma vai noslēguma u. tml.).	Aprakstā ir mazāk par pieciem procesa soļiem, tie ir loģiskā secībā un atspoguļo izpētes procesu no sākuma līdz galam, piemēram: sagatavošanās, norise un secinājumu izdarīšana.	Aprakstā iekļautā izpētes procesa gaita ir loģiska, detalizēta un sastāv no vismaz pieciem soļiem, piemēram: sagatavošanās, norise, rezultātu apkopošana, izvērtēšana un secinājumu izdarīšana.	Precīzi aprakstīta problēmas tēma sasaistē ar problēmas izpētes mērķi, kas formulēts konkrēti un saprotami. Izmantoto problēmas izpētes metožu pielietošana ir pamatota un jēgpilna, izpētes dati ir daudzveidīgi un sniedz atbildes uz izpētes jautājumiem.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***

Līmenis / Kritērijs	Sācis apgūt	Turpina apgūt	Apguvis	Apguvis padziļināti
2. Programmatūras prasību specifikācija – risinājuma mērķauditorijas izvēle un tās raksturojums, programmatūras un tās funkciju apraksts, programmatūras skice	Pielikumā pievienotie ievadizvades dati daļēji atbilst pētāmajai problēmai.	Pielikumā pievienotie ievadizvades dati atbilst pētāmajai problēmai. Norādīts ievadizvades datu avots.	Pielikumā pievienotie ievadizvades dati atbilst pētāmajai problēmai. Norādīts ievadizvades datu avots un metožu nosaukumi ar korekto deklarāciju.	Pielikumā pievienotie ievadizvades dati atbilst pētāmajai problēmai. Norādīts ievadizvades datu avots un metožu nosaukumi ar korekto deklarāciju, datu tipi, iespējamās vērtības. Norādīts, kā tiek iegūts (aprēķināts, izvadīts, formatēts utt.) rezultāts.
	1 – 2 punkti	***3 – 4 punkti***	***5 – 6 punkti***	***7 – 8 punkti***
3. Programmatūras izstrādes plāns	Iesniegts vispārīgs sagatavošanās posma, intervēšanas norises vai datu apstrādes apraksts.	Iesniegts sagatavošanās posma, intervēšanas norises vai datu apstrādes apraksts. Visu posmu rezultāti ir apkopoti.	Iesniegts sagatavošanās posma, intervēšanas norises vai datu apstrādes apraksts. Visu posmu rezultāti ir apkopoti un kritiski izvērtēti.	Iesniegts sagatavošanās posma, intervēšanas norises vai datu apstrādes apraksts. Visu posmu rezultāti ir apkopoti un kritiski izvērtēti. No rezultātiem ir izdarīti un noformulēti secinājumi.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***

Līmenis / Kritērijs	Sācis apgūt	Turpina apgūt	Apguvis	Apguvis padziļināti
4. Atkļūdošanas un akcepttestēšanas pārskats	Testēšanas plāns paredz pārbaudīt tikai atsevišķas sistēmas daļas. Akcepttestēšanas pārskats ir vispārīgs un satur virspusēju programmatūras prasību specifikācijā esošo prasību pārbaudi no lietotāja skatupunkta.	Testēšanas plāns ir nepilnīgs, tajā trūkst informācijas par atsevišķu sistēmas daļu testēšanu. Ne visas specifikācijā minētās programmatūras produkta funkcijas tiek pārbaudītas. Akcepttestēšanas pārskats ir vispārīgs un satur lielāko daļu programmatūras prasību specifikācijā esošo prasību pārbaudi no lietotāja skatupunkta.	Testēšanas plānā noteikts, kuras sistēmas daļas un kādā veidā jātestē. Nosaka nepieciešamo rezultātu atbilstoši specifikācijā esošajam programmatūras produkta funkciju aprakstam. Akcepttestēšanas pārskatā atspoguļota sistēmas darbības pārbaude no lietotāja skatupunkta atbilstoši specifikācijai.	Testēšanas plānā noteikts, kuras sistēmas daļas un kādā veidā jātestē. Nosaka nepieciešamo rezultātu atbilstoši specifikācijā esošajam programmatūras produkta funkciju aprakstam. Akcepttestēšanas pārskatā atspoguļota sistēmas darbības pārbaude no vairākiem lietotāju grupu skatupunktiem atbilstoši specifikācijai. Testēšanai ir izmantots digitāls rīks un/vai bibliotēka.
	1 – 2 punkti	***3 – 4 punkti***	***5 – 6 punkti***	***7 – 8 punkti***

Līmenis / Kritērijs	Sācis apgūt	Turpina apgūt	Apguvis	Apguvis padziļināti
5. Lietotāja ceļvedis	Izmantota iespējami vienkārša un saprotama valoda. Pievienota vizuāla informācija.	Izmantota iespējami vienkārša un saprotama valoda. Pievienota vizuāla informācija. Loģiska dokumenta struktūra. Viegli saprotamas sadaļas.	Izmantota iespējami vienkārša un saprotama valoda. Pievienota vizuāla informācija. Loģiska dokumenta struktūra. Viegli saprotamas sadaļas. Iespēja viegli atrast nepieciešamo informāciju. Iekļauta tikai nepieciešamā informācija.	Izmantota iespējami vienkārša un saprotama valoda. Pievienota vizuāla informācija. Loģiska dokumenta struktūra. Viegli saprotamas sadaļas. Iespēja viegli atrast nepieciešamo informāciju. Iekļauta tikai nepieciešamā informācija. Ceļveža izstrādei un publicēšanas tiešsaistē ir izmantots digitāls rīks.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***

Snieguma līmeņu apraksts programmatūras koda veidošanas posmam

Līmenis / Kritērijs	Sācis apgūt	Turpina apgūt	Apguvis	Apguvis padziļināti
1. Datubāzes izstrāde un izmantošana	Plāno datubāzi, t. sk. izveido ER modeli, konkrētā uzdevuma datu apstrādes risinājumam.	Plāno datubāzi, t. sk. izveido ER modeli, konkrētā uzdevuma datu apstrādes risinājumam. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā.	Izveido vienkāršu datu apstrādes programmatūru datu nolasīšanai, izmantojot paša veidoto datubāzi ar vienu tabulu un izmantojot kriptogrāfijas metodes datu aizsardzībai.	Izveido vienkāršu datu apstrādes programmatūru datu nolasīšanai un saglabāšanai, izmantojot paša veidoto datubāzi ar vairākām saistītām tabulām un izmantojot kriptogrāfijas metodes datu aizsardzībai.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***
2. API izmantošana	Ir kaut viena ideja par iespējamiem API izmantošanas variantiem, un ir sākta viena API integrēšana programmatūrā.	Ir vairākas idejas par iespējamiem API izmantošanas variantiem, no kuriem ir izvēlēts visatbilstošākais, un ir sākta tā integrēšana programmatūrā.	Ir kritiski izvēlēts un prasmīgi integrēts pedagoga ieteiktais API, kas korekti darbojas, bet tas pilnībā neatbilst programmatūras uzdevumiem un mērķiem.	Ir kritiski izvēlēts un prasmīgi integrēts patstāvīgi atrastais API, kas korekti darbojas un pilnīgi atbilst programmatūras uzdevumiem un mērķiem.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***

Līmenis / Kritērijs	Sācis apgūt	Turpina apgūt	Apguvis	Apguvis padziļināti
3. Datu ievadizvades organizēšana	Visi dati tiek tikai ievadīti.	Dati tiek gan ievadīti, gan izvadīti.	Dati vairākumā tiek ne tikai ievadīti un izvadīti, bet arī saglabāti un apstrādāti.	Visi dati tiek ievadīti, saglabāti, apstrādāti un izvadīti. Datu tiek ielādēti un saglabāti ārējos datu avotos, piemēram, datnēs, tīmekļa serveros utt.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***
4. OOP principu izmantošana	Ir definēta viena vai vairākas klases, bet nav realizēta mantošana: nav bāzes un atvasināto klašu.	Ir definēta viena bāzes klase un viena atvasinātā klase.	Ir definēta vismaz viena bāzes klase un 2 atvasinātās klases.	Ir definēta vismaz viena bāzes klase un 2 atvasinātās klases. Ir ievēroti OOP labās prakses principi.
	1 punkts	***2 punkti***	***3 punkti***	***4 punkti***