**MĀCĪBU PRIEKŠMETA TEMATISKAIS PLĀNS 2021./2022.m.g**

**Mācību priekšmets: Programmēšāna I Klase: 11a Skolotāja: Marina Juzova Stundu skaits: 4**

Saturs

[5. Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu? 40h 1](#_Toc83414279)

[Atpazīst mašīnmācīšanās pamatprincipus. Mašīnmācīšanās principus iespējams izmantot ikdienas situācijās un dažādos risinājumos. (T.Li.2.) 1](#_Toc83414280)

[Mašīnmācīšanās palīdz kompleksās jomās, kurās nav vienkārši izmantojamu algoritmu. (T.Li.2.) Zinātniskie pētījumi un atklājumi palīdz attīstīties programmēšanas nozarei un palielina tās pielietošanas iespējas. (T.Li.3.) 1](#_Toc83414281)

[Programmē, izmantojot objektorientēto pieeju 2](#_Toc83414282)

[Objektorientētā programmēšana atvieglo darbu ar kompleksām datu struktūrām un apkārtējās pasaules modelēšanu. (T.Li.2.) 2](#_Toc83414283)

[Izvēlas un izmanto papildu bibliotēku resursus 2](#_Toc83414284)

[Standarta bibliotēkas izmantošana atvieglo programmēšanas risinājumu izveidi. (T.Li.2.) 2](#_Toc83414285)

[Konfigurē drošības parametrus tīklā 3](#_Toc83414286)

[API (lietojumprogrammu) izmantošana var samazināt programmas izstrādes laiku. (T.Li.2.) Pastāv dažādi datortīkli. (T.Li.2.) Darba un programmatūras drošība ir svarīga. (T.Li.3.) 3](#_Toc83414287)

[Katrai datortīklā esošajai ierīcei ir noteikta funkcionalitāte. (T.Li.2.) 3](#_Toc83414288)

[Šifrē datus Šifrētai un nešifrētai datu plūsmai ir dažādi pielietojumi. (T.Li.3.) 3](#_Toc83414289)

[6. Kā īstenot risinājumu? 50h 5](#_Toc83414290)

[Eksistē vairāki programmatūras izstrādes modeļi. (T.Li.2.) Ir daudzi programmatūras izstrādes moduļi – ūdenskrituma, V­modelis, pakāpeniskais, iteratīvais un citi. (T.Li.2.) 5](#_Toc83414291)

[1. Analīze (tirgus izpēte, apraksts par risinājuma lietderību un efektivitāti) 5](#_Toc83414292)

[2. Idejas radīšana (idejas koncepcija). Idejas radīšana (idejas koncepcija) 5](#_Toc83414293)

[3. Plānošana (darba plāns, lomu sadale u. c.) Plānojot risinājumu, jāparedz tā izmantošanas veids un ietekme uz lietotāja veselību un labklājību. (T.Li.3.) 6](#_Toc83414294)

[4. Projektēšana (programmēšanas uzdevumu izvirzīšana) Programmatūras prasību specifikācija palīdz veikt programmatūras izstrādi. (T.Li.2.) 6](#_Toc83414295)

[Izstrādā prototipu un plāno savu risinājumu Izstrādājot programmatūru, ir noteikta veicamo darbību secība. (T.Li.2.) 6](#_Toc83414296)

[5. Attīstība (programmētā risinājuma izstrāde) Izstrādā un testē savu risinājumu 7](#_Toc83414297)

[6. Testēšana un pilnveide (pilns testēšanas cikls, datu iegūšana un analīze, uzlabojumu veikšana) 7](#_Toc83414298)

[7. Ieviešana (dokumentācija lietotājam, apmācība) Vizuāli patīkama un ērta lietotnes saskarne motivē lietotāju izmantot lietotni. (T.Li.2.) 7](#_Toc83414299)

[Lai aizsargātu savu risinājumu, ir jāparedz intelektuālā īpašuma aizsardzības veids. (T.Li.3.) 8](#_Toc83414300)

[8. Uzturēšana (uzturēšanas plāns). 8](#_Toc83414301)

[Projekta noslēgumā skolēni prezentē savu risinājumu un stāstījumā iekļauj būtiskākos izstrādes posmus un paveikto 8](#_Toc83414302)

[Izmantojamā literatūra: 9](#_Toc83414303)

| Nedēļa/ stunda | Stundu skaits | Tēma / Kompleksais sasniedzamais rezultāts /Caurviju prasmes un metodiskie paņēmieni | Uzdots /Formatīvā vērtēšana mācību procesa vadīšanai/ Summatīvā vērtēšana sasniedzamā rezultāta novērtēšanai | Mācību palīglīdzekļi |
| --- | --- | --- | --- | --- |
| 5. Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu? 40h | | | | | |
| Atpazīst mašīnmācīšanās pamatprincipus. Mašīnmācīšanās principus iespējams izmantot ikdienas situācijās un dažādos risinājumos. (T.Li.2.)Mašīnmācīšanās palīdz kompleksās jomās, kurās nav vienkārši izmantojamu algoritmu. (T.Li.2.) Zinātniskie pētījumi un atklājumi palīdz attīstīties programmēšanas nozarei un palielina tās pielietošanas iespējas. (T.Li.3.) | | | | | |
| 1/1,2 | 2 | Bezmaksas tiešsaistes mašīnmācīšanās kursu sērija | Piereģistrēties course.elementsofai.com/lv, izpildīt līdz Taksonomijai, ieskaitot | [https://course.elementsofai.com/lv](https://course.elementsofai.com/lv/) |
| 1/3,4 | 2 | Secina, kā zinātniskie pētījumi ietekmē sasniegumus tehnoloģiju jomā un citur. (T.O.3.2.5.) | Salīdzina to iespējas un piedāvā idejas, kā šādus produktus var izmantot savos projektos.  <https://deepdreamgenerator.com> un [https://deepart.io](https://deepart.io/) attēlu veidošanai, [https://www.autodraw.com](https://www.autodraw.com/)  https://artsandculture.google.com/experiment/YAGuJyDB-XbbWg | https://artsandculture.google.com/experiment/YAGuJyDB-XbbWg |
| 2/5,6 | 2 | Vispārīgi skaidro, kā mašīnmācīšanās strādā, kas tai ir nepieciešams un kā tā ietekmē mūsdienu sabiedrību. (T.O.2.4.18., T.O.3.2.5.) | Izvēlas konkrētu piemēru, kā šobrīd tiek izmantota mašīnmācīšanās (piemēram, *IBM Watson*, *OpenAI* *GPT-3*, spēļu spēlētāji – *AlphaGo*, *OpenAI Five*, [https://experiments.withgoogle.com/ai­duet](https://experiments.withgoogle.com/ai-duet) klavieru spēlētājs), izpēta to – nosaka pielietojuma mērķi, principus, iespējamas problēmsituācijas u. tml. – | https://experiments.withgoogle.com/collection/arts-culture |
| 2/7,8 | 2 | Skaidro atšķirības starp mākslīgo intelektu un vienkāršu algoritmu izmantošanu dažādās dzīves situācijās. (T.O.2.4.18., T.O.3.2.5.)   * Sastāda nepieciešamo programmvadāmo ierīču, programmatūras un citu resursu sarakstu atbilstoši lietotāja vajadzībām konkrētā uzdevuma veikšanai. (T.V.2.3.9.) | Aplūko piemērus mašīnmācīšanās lietošanai sociālajos tīklos un tiešsaistes veikalos, piemēram, “raksti, kuri jums varētu patikt” (*stories you might like*), “produkti, kuri jums varētu patikt” (*products you might like*). Diskutē par ieguvumiem, trūkumiem un ētikas problēmām, pielietojot mašīnmācīšanos šajās jomās. un **prezentē** apkopoto informāciju par mašīnmācīšanās pielietojumu kādā sevi interesējošā jomā. | https://experiments.withgoogle.com/ai%C2%ADduet |
| Programmē, izmantojot objektorientēto pieejuObjektorientētā programmēšana atvieglo darbu ar kompleksām datu struktūrām un apkārtējās pasaules modelēšanu. (T.Li.2.) | | | | |
| 3/9,10 | 2 | Veidojot savu risinājumu un izmantojot objektorientēto programmēšanas pieeju, izmanto klases, objektus, metodes un laukus (atribūtus). (T.O.2.4.15.) | Saviem vārdiem skaidro jēdzienus klase (*class*), objekts (*object*), īpašība (*property*/*field*) un metode (*method*). |  |
| Izvēlas un izmanto papildu bibliotēku resursusStandarta bibliotēkas izmantošana atvieglo programmēšanas risinājumu izveidi. (T.Li.2.) | | | | |
| 3/13,14 | 2 | Salīdzina programmēšanas valodā biežāk lietotās standarta bibliotēkas funkcijas, procedūras un metodes un izvēlas piemērotāko no tām sava risinājuma izstrādē. (T.O.2.4.13.) | Veido “Atbilstoši kases aparāta lietotāja vajadzībām izveido skici vienkāršam tā vizuālajam izskatam (logs ar dažiem teksta laukiem un pogām).” |  |
| 4/15,16 | 2 | Salīdzina dažādas programmvadāmas ierīces un to lietojamību sadzīvē un ražošanā. (T.V.2.3.9., T.V.2.1.2.) | Grupas demonstrē veikto darbu, diskutē par darba gaitu, novērtē kolēģu darbu **Formatīvais** vērtējums “Atbilstoši kases aparāta lietotāja vajadzībām izveido skici vienkāršam tā vizuālajam izskatam (logs ar dažiem teksta laukiem un pogām).” |  |
| Konfigurē drošības parametrus tīklāAPI (lietojumprogrammu) izmantošana var samazināt programmas izstrādes laiku. (T.Li.2.) Pastāv dažādi datortīkli. (T.Li.2.) Darba un programmatūras drošība ir svarīga. (T.Li.3.) | | | | | |
| 5/17,18 | 2 | Izvēlas konkrētai situācijai atbilstošāko datortīklu, izvērtējot dažādu datortīklu veidu iespējas un drošības riskus. (T.V.2.3.1., T.O.3.1.3.) Pielāgo operētājsistēmas un lietotnes iestatījumus, veicinot darba drošību. (T.V.2.3.11.) Izvēlas piemērotāko datortīklu, balstoties uz uzdevuma prasībām. (T.V.2.3.1.) | Izmēra un salīdzina datu pārraides ātrumu dažādiem pieslēguma veidiem. Konfigurē savu Wi-Fi pieejas punktu ar paroli  Pārrunā iespējamos risinājumus, izmantojot tikai esošās zināšanas un spriež, kas būtu jāapgūst papildus. | *Random User Generator* [tiešsaiste]. *Arron J. Hunt* un *Keith Armstrong*.  [skatīts 24.07.2021.]. Pieejams: [https://randomuser.me](https://randomuser.me/) |
| Katrai datortīklā esošajai ierīcei ir noteikta funkcionalitāte. (T.Li.2.) | | | | |
| 5/19,20 | 2 | Pēc nepieciešamības izmanto *API* (lietojumprogrammu) projekta izstrādē. (T.O.2.4.11.) | Pievieno *Python* *Flask* (vai *Python Bottle*) bibliotēku vienkārša tīmekļa servera izveidošanai. Izmantojot piemērus, veido vienkāršu tīmekļa serveri, kurš atgriež tekstu “Sveika, pasaule”. Papildina to ar iespējām atgriezt no ārējās *API* iegūtos datus. |  |
| 6/21,22 | 2 | Ievēro drošības nosacījumus, lietojot atvērto datu apmaiņu. (T.V.3.1.3.) | Pēc nepieciešamības izmanto esošu programmēšanas valodas dokumentāciju sava risinājuma izveidē. (T.O.2.4.10.) |  |
| 6/23,24 | 2 | Veic maršrutētāja konfigurāciju. (T.O.2.3.1.) | **Formatīvais** vērtējums. Katrai grupai izveidot prezentāciju par maršrutētājiem. |  |
| Šifrē datus Šifrētai un nešifrētai datu plūsmai ir dažādi pielietojumi. (T.Li.3.) | | | | | |
| 7/25,26 | 2 | Atbilstoši situācijai izvēlas atvērto vai šifrēto datu apmaiņu. (T.V.3.1.3.) | Iepazīstas ar datu šifrēšanas lietošanu un nepieciešamību šifrēt datus dažādos digitālos un tiešsaistes produktos. Diskutē par drošības riskiem, izmantojot tīmekli pa atvērta teksta pieslēgumu (*HTTP*). Argumentē, kādēļ nepieciešams izmantot šifrētu pieslēgumu (*HTTPS*). |  |
| 7/27,28 | 2 | Novērtē drošības riskus, lietojot programmvadāmās ierīces, un rīkojas, lai tos samazinātu. (T.O.3.1.3.) | Iepazīstas ar simetrisko un asimetrisko šifrēšanu un jaucējfunkcijām, apskata to pielietojumu dažādos pazīstamos kontekstos. Salīdzina dažas kriptogrāfijas bibliotēkas (piemēram, *Python bcrypt, Cryptography*), to pielietojumu un atbilstību konkrētām situācijām.  Veido datorprogrammu, kura, izmantojot kriptogrāfijas bibliotēku, šifrē un atšifrē tekstu vai simbolu virkni. |  |
| 8/29,30 | 2 | **Summatīvais** vērtējums. | **Demonstrē darbus, pārrunas par to, kas sanāca, nesanāca, ko varētu uzlabot savā darbā, kolēģa darbā.** |  |

| Nedēļa/ stunda | Stundu skaits | Tēma / Kompleksais sasniedzamais rezultāts /Caurviju prasmes un metodiskie paņēmieni | Uzdots /Formatīvā vērtēšana mācību procesa vadīšanai/ Summatīvā vērtēšana sasniedzamā rezultāta novērtēšanai | Mācību palīglīdzekļi |
| --- | --- | --- | --- | --- |
| 6. Kā īstenot risinājumu? 50h | | | | |
| Eksistē vairāki programmatūras izstrādes modeļi. (T.Li.2.) Ir daudzi programmatūras izstrādes moduļi – ūdenskrituma, V­modelis, pakāpeniskais, iteratīvais un citi. (T.Li.2.) | | | | |
| 8/31,32 | 2 | Atkārto programmatūras izstrādes moduļus – ūdenskrituma, V­modelis, pakāpeniskais, iteratīvais un citi.  Izstrādā programmatūru, izmantojot iteratīvo izstrādes modeli. (T.O.2.4.2.) | Vienojas par izstrādes modeli, minimālām drošības prasībām projekta veidošanas un ieviešanas gaitā, piemēram, pieeja čatam tikai ar paroli. Izvēlas un lieto mērķim atbilstošu lietotāju izpētes metodi, lai iegūtu datus par risinājuma nepieciešamību, potenciālo pieprasījumu, vēlamo izskatu un funkcionalitāti. Analizē piemērus ar dažādiem programmatūras izstrādes modeļiem (piemēram, ar ūdenskrituma, V­modeli, iteratīvo un pakāpenisko), salīdzina to darbības principus un efektivitāti dažādās situācijās. |  |
| 1. Analīze (tirgus izpēte, apraksts par risinājuma lietderību un efektivitāti) Lai izveidotu pieprasītu risinājumu, ir jānosaka risinājuma mērķauditorija un jāizzina tās vajadzības. (T.Li.1., T.Li.3.) | | | | |
| 9/33,34 | 2 | Paredz risinājuma izmantošanas ietekmi uz cilvēka veselību un labklājību, rīkojas, lai samazinātu negatīvo ietekmi. (T.V.3.1.1.) | Analizē piemērus tam, kā līdzīgu risinājumu izmantošana ietekmē lietotājus, to fizisko un garīgo veselību un drošību. Analizē drošības riskus, izstrādājot vairāklietotāju tīmekļa programmu. |  |
| 2. Idejas radīšana (idejas koncepcija). Idejas radīšana (idejas koncepcija) | | | | |
| 9/35,36 | 2 | Ideju radīšanai tiek lietotas dažādas metodes (T.Li.1.). Diskutē par drošības jautājumu nozīmi projekta plānošanā. | Grupās identificē kādu lokālu problēmu, aplūko un salīdzina dažādu līdzīgu risinājumu – programmu un vietņu – funkcionalitāti (piemēram, *WhatsApp, Skype, Discord, Slack)*. |  |
| 3. Plānošana (darba plāns, lomu sadale u. c.) Plānojot risinājumu, jāparedz tā izmantošanas veids un ietekme uz lietotāja veselību un labklājību. (T.Li.3.) | | | | |
| 10/37,38 | 2 | Plānošana (darba plāns, lomu sadale u. c.) Programmatūras izstrādes modeļa izveidē ievēro tā galvenos izstrādes posmus. (T.O.2.4.3.) | Apkopo rezultātus un, vadoties pēc tehniskām iespējām, sagatavo argumentētu invertējumu par pieņemamām un atmetamām vēlmēm.  Diskutē par ētisku risinājumu izveidi. Sava risinājuma izstrādē veido aprakstu par tā ietekmi uz lietotājiem un vidi. Balstoties uz izstrādāto projekta koncepciju, grupas dalībnieki vienojas par darbu sadali un izplāno izstrādi, nosakot katra atbildību. |  |
| 4. Projektēšana (programmēšanas uzdevumu izvirzīšana) Programmatūras prasību specifikācija palīdz veikt programmatūras izstrādi. (T.Li.2.) | | | | |
|  |  |  |  |  |
| 10/39,40 | 2 | Izveido vienkāršotu programmatūras prasību specifikāciju. (T.O.2.4.4.) |  |  |
| Izstrādā prototipu un plāno savu risinājumu Izstrādājot programmatūru, ir noteikta veicamo darbību secība. (T.Li.2.) | | | | |
| 11/41-44 | 2 | Aktualizē apgūtos pamatprincipus un aplūko piemērus *HTML* un *CSS* lietošanā. Izveido vienkāršu tīmekļa lapu ar informāciju par topošo projektu | Plāno projekta uzdevumus un izstrādes soļus, piemēram, čata tīmekļa lietotnei,   * pāros zīmē risinājuma blokshēmu; * vienojas par servera un klienta komunikācijas parametriem ziņu lasīšanai (*URL* shēmu, datu formātu); |  |
| 5. Attīstība (programmētā risinājuma izstrāde) Izstrādā un testē savu risinājumu | | | | |
| 12/45,46 |  | Darbs ar klientu – serveri izmantojot servera un klienta komunikācijas parametrius ziņu sūtīšanai (*URL* shēmu, datu formātu); | vienojas par servera un klienta komunikācijas parametriem ziņu sūtīšanai (*URL* shēmu, datu formātu. Ziņu pārsūtīšana no klientu uz serveri un atpakaļ |  |
| 12/47,48 |  | Izmanto programmēšanas valodas dokumentāciju. | Izstrādā risinājumam nepieciešamo funkcionalitāti. |  |
| 6. Testēšana un pilnveide (pilns testēšanas cikls, datu iegūšana un analīze, uzlabojumu veikšana) | | | | |
|  |  |  |  |  |
| 13/49,50 |  | * Veic attēlu, audio un video datņu apstrādi, veidojot programmatūrai nepieciešamos grafiskos elementus. (T.V.2.2.3.) * Testē, sastādot testa plānu, un pēc nepieciešamības atkļūdo risinājumu un demonstrē to citiem. | **Projektu veic iteratīvi un dokumentē tā norisi:**   1. Atbilstoši plānam veido risinājumu un nepieciešamības gadījumā izmanto programmēšanas valodas dokumentāciju. |  |
| 13/51,52 | 2 | 1. Pārbauda risinājumu. Testē paša un citu izstrādātos projektus, vērtē to atbilstību iepriekš definētajiem kritērijiem. Labo citu norādītās kļūdas savā projektā. |  |
| 14/53,54 | 2 | 1. Demonstrē risinājumu citiem skolēniem vai skolotājam un iegūst atgriezenisko saiti. |  |
| 14/55,56 | 2 | 1. Salīdzina risinājumu ar plānu, kuru izlabo atbilstoši realitātei un ieteikumiem. |  |
| 15/57,58 | 2 | 1. Ja nepieciešami uzlabojumi, atgriežas pie 1. punkta. |  |
| 7. Ieviešana (dokumentācija lietotājam, apmācība) Vizuāli patīkama un ērta lietotnes saskarne motivē lietotāju izmantot lietotni. (T.Li.2.) | | | | |
| 15/59,60 | 2 | Izveido vienkāršotu un lietotājam saprotamu programmatūras ceļvedi. (T.O.2.4.7.) | Analizē citus risinājuma publicēšanas veidus. Veido programmas aprakstu, apkopo lietotāju prasības un tehnisko specifikāciju. Nosaka noteicošos kritērijus, lai novērtētu, vai vēlamais rezultāts ir sasniegts. |  |
| 16/61,62 | 2 | Izstrādā sava risinājuma dokumentāciju gala lietotājam, piemēram, lietotāja ceļvedi, reklāmas bukletu vai atgādni, kura izveidē pēc nepieciešamības veic attēlu, audio un video datņu apstrādi. |  |
| Lai aizsargātu savu risinājumu, ir jāparedz intelektuālā īpašuma aizsardzības veids. (T.Li.3.) | | | | |
| 16/63,64 |  | Ievēro godīga dizaina principus un paredz sava risinājuma intelektuālā īpašuma aizsardzības veidu. (T.O.3.1.5.)  Paredz nepieciešamos pasākumus darba aizsardzības prasību nodrošināšanai visā risinājuma izstrādes procesā. (T.V.3.1.2.). | Izvēlas un pamato sava risinājuma intelektuālā īpašuma aizsardzības veidu. |  |
| 8. Uzturēšana (uzturēšanas plāns). | | | | |
| 17/65,66 |  | Risinājuma izstrādes laikā ievēro darba drošības prasības. (T.V.3.1.2.) | Publicē projektu publiski pieejamā tīmekļa vietnē, izmantojot *Repl.it*, *Heroku*, *PythonAnywhere* vai citus mākoņpakalpojumus. |  |
| Projekta noslēgumā skolēni prezentē savu risinājumu un stāstījumā iekļauj būtiskākos izstrādes posmus un paveikto | | | | |
| 17/67,68 | 4 | **Summatīvais** vērtējums. | **Demonstrē gatavo risinājumu, stāsta par projekta gaitā pārvarētajām grūtībām. kas sanāca, nesanāca, ko varētu uzlabot savā darbā, kolēģa darbā.** |  |
| 18/69,70 |  |
| Par katru dokumentēto starpposmu skolēns saņem atgriezenisko saiti no skolotāja un/vai citiem skolēniem. | | | | |

# Izmantojamā literatūra:

1. daļa
   1. Programmēšanas pamati ar valodu Python © Jānis Zuters,Latvijas Universitāte, 2019-2020 [skatīts 01.09.2020.]. Pieejams: http://home.lu.lv/~janiszu/courses/python/python3.pdf - tiek apzīmēta **MG**
   2. Hello World [skatīts 24.07.2020.]. Pieejams: <https://guides.github.com/activities/hello-world/>
   3. Choose an open source license [tiešsaiste]. GitHub, Inc. [skatīts 24.07.2020.]. Pieejams: https://choosealicense.com/licenses○Software Licenses in Plain English [tiešsaiste]. FOSSA, Inc. [skatīts 24.07.2020.]. Pieejams: <https://tldrlegal.com>
   4. Built for developers [skatīts 24.07.2020.]. Pieejams: <https://github.com/> Atom, cmd
   5. 10 populārākās un labāk apmaksātās programmēšanas valodas, kuras vērts iemācīties. [skatīts 01.09.2020.]. Pieejams: <https://kursors.lv/2019/06/25/10-popularakas-un-labak-apmaksatas-programmesanas-valodas-kuras-verts-iemacities/>
   6. 9.28 Programmatūras izstrāde I [skatīts 01.09.2020.]. Pieejams: <https://datorika.startit.lv/datorika/klase/9/stunda/531>
   7. 6.27 Programmatūras izstrāde I [skatīts 01.09.2020.]. Pieejams: https://datorika.startit.lv/datorika/klase/6/stunda/512
2. daļa
   1. Citāti no J.Teilora grāmatas Managing Information Technology Projects. [skatīts 17.10.2020.]. Pieejams: https://habr.com/ru/company/dcmiran/blog/521718/
   2. SDLC Tutorial [skatīts 17.10.2020.]. Pieejams: https://www.tutorialspoint.com/sdlc/index.htm
   3. Aptauju paraugi [skatīts 17.10.2020.]. Pieejams: www.visidati.lv/surveys\_examples/
   4. Izveidojiet lieliskas veidlapas www.google.com/intl/lv/forms/about/
   5. Valsts un pašvaldību iestāžu tīmekļvietņu vienotā platforma[skatīts 17.10.2020.]. Pieejams: https://www.mk.gov.lv/sites/default/files/editor/Iepirkumi/2018/mkvk2018\_9\_1\_pielikums\_tehniska\_specifikacija.docx
   6. SDLC — Краткое руководство [skatīts 17.10.2020.]. Pieejams: coderlessons.com/tutorials
   7. Методология разработки Waterfall: что это, как работает и чем отличается от Agile [skatīts 17.10.2020.]. Pieejams: <https://skillbox.ru/media/management/waterfall/>
   8. Agile programmatūras izstrādes manifests [skatīts 17.10.2020.]. Pieejams: https://agilemanifesto.org/iso/lv/manifesto.html
   9. Waterfall Methodology [skatīts 17.10.2020.]. Pieejams: https://www.projectmanager.com/waterfall-methodology