**MĀCĪBU PRIEKŠMETA TEMATISKAIS PLĀNS 2020./2021.m.g**

**Mācību priekšmets: Klase: 10a Skolotāja: Marina Juzova Stundu skaits: 4**

Saturs

[Kā uzrakstīt kodu pēc dotas specifikācijas? (30. st.) 1](#_Toc82367028)

[• Ergonomiska darba vide var stimulēt produktīvu darbu. (T.Li.3.) 1](#_Toc82367029)

[• Programmēšanas valoda jāizvēlas atbilstoši risināmajai problēmai. (T.Li.2.) 1](#_Toc82367030)

[• Programmatūras dzīves ciklā ir septiņi galvenie posmi: plānošana, analīze, projektēšana, attīstība, testēšana, ieviešana, uzturēšana. (T.Li.1.) 2](#_Toc82367031)

[• Programmatūras prasību specifikācijā ir izstrādātājam saprotamā veidā definēts problēmas risinājums. (T.Li.2.) 2](#_Toc82367032)

[• Dažādās programmēšanas valodās ir līdzīgi programmēšanas pamatprincipi, bet atšķiras valodu sintakse. (T.Li.2.) 2](#_Toc82367033)

[• Rakstot kodu, jāievēro koda pieraksta stils, lai kods būtu vieglāk lasāms un saprotams.(T.Li.2.) 3](#_Toc82367034)

[• Programmējot kodu nepieciešams komentēt, lai padarītu to sev un citiem vieglāk uztveramu. (T.Li.2.) 3](#_Toc82367035)

[• Lai nedublētu pirmkodu, izmanto cikla konstrukcijas vai funkcijas. (T.Li.2.) 3](#_Toc82367036)

[2\_“Kā izplānot un izveidot lietotājam atbilstošu programmētu risinājumu un uzraudzīt tā īstenošanu?” (30 mācību stundas) uzsvērts programmatūras izstrādes process un tā vadība 4](#_Toc82367037)

[I Planning and requirement analysis (Plānošana un prasību analīze) Kādām problēmām nepieciešams risinājums! 4](#_Toc82367038)

[II Defining requirements Prasību nodrošināšana, saskaņošana. Darbības plāns problēmu atrisināšanai.. 4](#_Toc82367039)

[III Design the product architecture (Projekta arhitektūras izstrāde) 5](#_Toc82367040)

[IV Building or developing the product (Produkta izstrāde vai attīstīšana) 5](#_Toc82367041)

[V Testing the product (Produkta testēšana) 6](#_Toc82367042)

[VI Deployment in the Market and Maintenance (Izvietošana tirgū un uzturēšana) 6](#_Toc82367043)

[3. Kā automātiski apstrādāt un attēlot datus? 40h 8](#_Toc82367044)

[Kvalitatīvs informācijas avots ir viens no priekšnosacījumiem augstas kvalitātes datu iegūšanā. (T.Li.2.) 8](#_Toc82367045)

[Datubāzes pareiza izveide atvieglo darbu ar datiem. (T.Li.2.) 8](#_Toc82367046)

[Automatizācijas rīki atvieglo darbu ar apjomīgiem datiem. (T.Li.2.) 8](#_Toc82367047)

[Datu validācija aizsargā datubāzi. (T.Li.2.) 9](#_Toc82367048)

[Datus var apstrādāt un veidot kvalitatīvu to vizuālo noformējumu, lai tie būtu uzskatāmāki. (T.Li.2.) 9](#_Toc82367049)

[Programmas darbībai ir jābūt pietiekoši ātrai, lai lietotājam būtu ērti to lietot. (T.Li.2.) 9](#_Toc82367050)

[Kārtošanas un meklēšanas algoritmi lietotājam atvieglo programmatūras lietojamību. (T.Li.2.) 10](#_Toc82367051)

[Zinot algoritma sarežģītību, iespējams izvērtēt programmatūras izveides iespējamību un lietojamību atkarībā no izpildāmo darbību apjoma. (T.Li.2.) 10](#_Toc82367052)

[4. Kā sistemātiski pārbaudīt un atkļūdot risinājumu? 20h 11](#_Toc82367053)

[Atkļūdo kļūdaini uzrakstītas programmas Atkļūdošana ir būtiska daļa no programmatūras izstrādes. (T.Li.2.) Rakstot kodu, iespējams pieļaut dažādu veidu kļūdas, kuras var atkļūdot, zinot kļūdas paziņojuma veidu. (T.Li.2.) Kļūdas var rasties, izmantojot nepareizus datu tipus. (T.Li.2.) 11](#_Toc82367054)

[Pārbauda (testē) programmas darbību. Testējot var pārbaudīt, vai programmatūra strādā korekti un izpilda uzstādītās prasības. (T.Li.2.) 12](#_Toc82367055)

[Testējot programmu ar lietotājiem, iespējams atklāt nepieciešamos uzlabojumus programmas lietojamībā. (T.Li.1., T.Li.3.) 12](#_Toc82367056)

[5. Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu? 40h 13](#_Toc82367057)

[Atpazīst mašīnmācīšanās pamatprincipus 13](#_Toc82367058)

[Mašīnmācīšanās principus iespējams izmantot ikdienas situācijās un dažādos risinājumos. (T.Li.2.) 13](#_Toc82367059)

[Mašīnmācīšanās palīdz kompleksās jomās, kurās nav vienkārši izmantojamu algoritmu. (T.Li.2.) 13](#_Toc82367060)

[11.klase 14](#_Toc82367061)

[6. Kā īstenot risinājumu? 50h 15](#_Toc82367062)

[Izmantojamā literatūra: 16](#_Toc82367063)

| 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 |
| --- | --- | --- | --- | --- |
| Kā uzrakstīt kodu pēc dotas specifikācijas? (30. st.) | | | | | |
| • Ergonomiska darba vide var stimulēt produktīvu darbu. (T.Li.3.) | | | | | |
| 1/1,2 | 2 | • Izvēlas un lieto tehnoloģijas, izvērtējot to radītos riska faktorus cilvēka veselībai. (T.V.2.1.1., T.V.3.1.1., T.V.3.1.4.)  • Pielāgo savu darba vietu, ievērojot atbilstību ergonomikas prasībām. (T.V.3.1.1., T.O.3.1.1.) / sports, bioloģija, angļu valoda | Reģistrēties mācību vidēs macibas.e-skola.lv MS Office 365, pasts.edu.riga.lv | [1.1], [1.2], [1.3] |
| 1/3,4 | 2 | • Versiju vadības rīki atvieglo koda izmaiņu saglabāšanu un atjaunošanu, kā arī ērtu veidu kā tos lietderīgi izmantot, ja projektā sadarbojas vairāki izstrādātāji. (T.Li.2.) / teams | Apskatīt atvērtā koda licences ( GPLv3, BSD, MIT, Apache), Atšķirt public no private. izveido kontu tiešsaistes GitHub sistēmā, pieslēgties un izveidot repozitoriju, izvēloties tādu atvērtā koda licenci, kura atbilstoši situācijai definē koda izmantošanas tiesības. Klonēt (git clone) izveidoto repozitoriju datorā, lai izmantotu versiju vadības iespējas sava koda saglabāšanai un izmaiņu reģistrēšanai. |
| 2/5,6 | 2 | • Programmatūras izstrādes procesā lieto versiju vadības rīku. (T.O.2.4.9.) / sadarbība | Izmantojot git vai koda redaktora iebūvēto funkcionalitāti, saglabāt savu kodu klonētajā repozitorijā (git commit) un iesūtīt izmaiņas (git push) tiešsaistes GitHub sistēmā |
| • Programmēšanas valoda jāizvēlas atbilstoši risināmajai problēmai. (T.Li.2.) | | | | | |
| 2/7,8 | 2 | • Izvēlas konkrētam mērķim piemērotāko programmēšanas valodu, apsverot programmēšanas valodu iespējas un ierobežojumus. (T.O.2.4.12.) / angļu valoda | macibas.e-skola.lv [01\_Ergonomika\_Programmēšanas valoda](https://macibas.e-skola.lv/mod/assign/view.php?id=32804) | [1.5] |
| • Programmatūras dzīves ciklā ir septiņi galvenie posmi: plānošana, analīze, projektēšana, attīstība, testēšana, ieviešana, uzturēšana. (T.Li.1.) | | | | | |
| 3/9,10 | 2 | Analizē vienkāršus ikdienas darba procesus, saskata automatizācijas iespējas tajos vai to daļās. (T.O.2.4.1.) | Aprakstīt vēlamo lietotāja pieredzi, plānot, veikt un dokumentēt projekta norisi, izmantojot šādus soļus: 1. uzzīmēt risinājuma blokshēmu vai uzrakstīt pseidokodu. | [1.6] [1.7]  macibas.e-skola.lv  [02\_Programmatūras dzīves cikli](https://macibas.e-skola.lv/mod/assign/view.php?id=32808) |
| 3/11,12 | 2 | Lieto programmēšanas valodas dokumentāciju un palīdzības sistēmu. (T.O.2.4.10.) / Sadarbības prasmes. | 2. Izveidot risinājumu. 3. Pārbaudīt risinājumu. 4. Demonstrēt risinājumu |
| • Programmatūras prasību specifikācijā ir izstrādātājam saprotamā veidā definēts problēmas risinājums. (T.Li.2.) | | | | | |
| 4/13,14 | 2 | • Salīdzina jau esošu programmatūru ar tās specifikāciju, raksturo specifikācijā izmantotos paņēmienus un nosauc, kādas darbības veiks, lai to izstrādātu (T.V.1.1.1.) | *Zināt priekšstata līmenī datorbāzētas sistēmas dzīves cikla galvenos posmus, pēc kuriem* Demonstrēt gatavo risinājumu, stāstīt par projekta gaitā pārvarētajām grūtībām. |  |
| 4/15,16 |  | Izstrādā vienkāršu programmatūru atbilstoši programmatūras prasību specifikācijai un uzdevumam. (T.O.2.4.5.) |  |
| • Dažādās programmēšanas valodās ir līdzīgi programmēšanas pamatprincipi, bet atšķiras valodu sintakse. (T.Li.2.) | | | | | |
| 5/17 | 1 | • Rakstot kodu, izmanto izstrādes vidē iebūvētās iespējas lietotāja darba produktivitātes paaugstināšanai. (T.O.2.3.2.) | Prast izmantot IDLE, repl.it, python pip. | MG - 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  python pip [skatīts 17.09.2020.]. Pieejams:  https://www.w3schools.com/python/python\_pip.asp |
| • Rakstot kodu, jāievēro koda pieraksta stils, lai kods būtu vieglāk lasāms un saprotams.(T.Li.2.)• Programmējot kodu nepieciešams komentēt, lai padarītu to sev un citiem vieglāk uztveramu. (T.Li.2.) | | | | | |
| 5/18 |  | • Veido jēgpilnus mainīgo un funkciju nosaukumus. (T.O.2.4.8.) | Atšķirt atkāpes nozīmi pyton valodā un pārējās! Veidot mainīgo, visā pirmkodā lietot vienotu stilu, camelCase vai snake\_case, bet ne abus. Konstantēm jāizmanto LIELIE\_BURTI. | MG 5.lpp – 37.lpp  Python Comments . [skatīts 17.09.2020.]. Pieejams: https://www.w3schools.com/python/default.asp |
| • Lai nedublētu pirmkodu, izmanto cikla konstrukcijas vai funkcijas. (T.Li.2.) | | | | | |
| 5/19,20 | 2 | • Lieto standarta ievadi un izvadi. (T.O.2.4.13.) | Prast lietot input(), print(), mainīgo tipus, matemātiskos operatorus. Izrēķināt savas istabas tilpumu. | MG 38.lpp – 41.lpp  Python Variables, Python Data Types  [skatīts 17.09.2020.]. Pieejams: https://www.w3schools.com/python/default.asp |
| 6/21-24 | 4 | Lieto zarošanās konstrukcijas un loģiskās izteiksmes. (T.O.2.4.13.) | Izrēķināt savas istabas tilpumu, nepieciešamās tapetes! Izrēķināt izdevīgako risinājumu savam projektam | 42.lpp – 48.lpp  Python If ... Else, Python Booleans [skatīts 17.09.2020.]. Pieejams: https://www.w3schools.com/python/default.asp |
| 7/25-28 | 4 | • Lieto cikla konstrukcijas. (T.O.2.4.13.) | for(), while(), break(), continue,  Izrēķināt vai ievadītā parple ir pareiza | MG 49.lpp – 55.lpp.  Python While Loops [skatīts 17.09.2020.]. Pieejams: https://www.w3schools.com/python/default.asp |
| 8/29,30 | 2 | • Veidojot programmu, izmanto funkcijas. (T.O.2.4.13.) | prot def(), atšķirt formālos parametrus no faktiskajiem parametriem . Izveidot n! | MG 81.lpp – 93.lpp.  [skatīts 17.09.2020.]. Pieejams: https://www.w3schools.com/python/default.asp |

| 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 |
| --- | --- | --- | --- | --- | --- | --- |
| 2\_“Kā izplānot un izveidot lietotājam atbilstošu programmētu risinājumu un uzraudzīt tā īstenošanu?” (30 mācību stundas) uzsvērts programmatūras izstrādes process un tā vadība | | | | | | |
| **Lai izveidotu lietotājam vajadzīgu un ērti izmantojamu risinājumu, ir jāveic mērķauditorijas izpēte. (T.Li.3.)** I Planning and requirement analysis (Plānošana un prasību analīze) Kādām problēmām nepieciešams risinājums! | | | | | | |
| 8/31,32 | | 2 | Programmatūras izstrādes paņēmieni | | Zināt Agile filozofijas 4 manifestus, Waterfall, SCRUM paņēmienus programmatūras izstrādē. | [2.7.], [2.8.], [2.9.] |
| 9/33,34 | | 2 | Analizē vienkāršus ikdienas darba procesus un saskata iespējas to automatizācijai. (T.V.2.4.1., T.O.2.4.1.) | | 1.2.Izpētīt un analizēt lietotāju vajadzības, un veidot savu ideju problēmu atrisināšanai. | [2.1.] google.com, skola, mājas,... |
| 9/35,36 | | 2 | Pēc dotā parauga izveido lietotāju vajadzību noskaidrošanai atbilstošus jautājumus. (T.V.1.2.1.) un kādā veidā tie ikdienā izmanto, lietojot dažādas pētniecības metodes. | | šodien katrai grupai - 1.2.1.Noskaidrot lietotāju paradumus, intereses un to, kādus risinājumus piedāvāt Jūsu idejas realizēšanai. Sagatavot vismaz 9 jautājumus vienā no elektroniskajām aptaujām.  - iesniegt  9/35,36 - pieejams: https://macibas.e-skola.lv/mod/assign/view.php?id=33397 | [2.1], [2.2], [2.3], [2.4] |
| Izmanto populārākās lietotāju izpētes metodes (intervija, aptauja, novērojums, eksperiments), lai noskaidrotu lietotāja paradumus, analizē iegūtos datus un secina, kādas ir turpmākas darbības risinājuma izstrādē. (T.V.1.2.1.) | | Apskatīt dotos vērtēšanas kritērijus 9/35,36 - pieejams: <https://macibas.e-skola.lv/mod/assign/view.php?id=33397> 2.3.4. Sagatavot mērķauditorijas aptaujas un anketēšanas formas, organizēt aptaujas un anketēšanu, veikt iegūto datu manuālu un automatizētu apkopošanu. |
| Programmatūras prasību specifikācija apraksta vēlamo programmatūras funkcionalitāti no lietotāja viedokļa. (T.Li.2.) II [Defining requirements Prasību nodrošināšana, saskaņošana.](https://www.softwaretestingclass.com/software-requirement-specification-srs/#:~:text=A%20software%20requirements%20specification%20(SRS,functional%20and%20non%2Dfunctional%20requirements.&text=The%20software%20requirement%20specification%20document,requirements%20required%20for%20project%20development.) Darbības plāns problēmu atrisināšanai.. | | | | | | | |
| 10/37,38 | | 2 | Koplietotā dokumentā dažādiem lietotājiem iestata atšķirīgas piekļuves, rediģēšanas un komentēšanas tiesības. (T.V.2.3.2.) | | Apgūt [Software Requirement Specification (SRS) Programmu nodrošinājuma prasību specifikāciju.](https://www.softwaretestingclass.com/software-requirement-specification-srs/#:~:text=A%20software%20requirements%20specification%20(SRS,functional%20and%20non%2Dfunctional%20requirements.&text=The%20software%20requirement%20specification%20document,requirements%20required%20for%20project%20development.) | [2.5], [1.1], [1.2], [1.3] |
| 10/39,40 | | 2 | Izveido vienkāršotu programmatūras prasību specifikāciju, ievērojot mērķauditorijas vajadzības. (T.O.2.4.4.) | | Savas ideja realizēšanai - plānot izmaksas un ieguvumus, apzināt riskus. Apzināt kontrolējamo, nekontrolējamo atstāt nākamajam etapam.  2.4.4. Pēc parauga sastādīt vienkāršotu programmatūras prasību specifikāciju atbilstoši uzdevumam, ievērojot mērķauditorijas vajadzības. | google.lv [1.6], [1.7],  [2.5], [2.6]  macibas.e-skola.lv |
| Atbilstoši uzdevumam izstrādā vienkāršotu programmatūras prasību specifikāciju. (T.O.2.4.4.) | |
| Informācijas strukturēšana atvieglo tās uztveramību. (T.Li.2.) **1.1. Dizaina procesa īstenošana.** III Design the product architecture (Projekta arhitektūras izstrāde) | | | | | | | |
| 11/41,42 | | 2 | Analizē doto informācijas dizaina risinājumu priekšrocības un trūkumus. (T.V.2.2.2.)  Design Document Specification (DDS) -risku produkta drošība laika ierobežošanai, | | 2.2.2. Salīdzinaāt dažādos medijos izmantotos informācijas dizaina risinājumus, kuri līdzīgi Jūsu idejai, analizēt konkrēto piemēru priekšrocības un trūkumus. Nosaukt vismaz 3 priekšrocības un vienu trūkumu. | google.lv |
| 11/43,44 | | 2 | Atpazīst un analizē informācijas dizaina pamatprincipu izmantošanu daudzveidīgos piemēros. (T.V.2.2.1.) | | Piedāvāt vismaz 3 dizainu veidus (2.2.1 Atpazīt un analizēt informācijas dizaina pamatprincipu izmantošanu daudzveidīgos piemēros) 1.3.1. Plānojot risinājumu, veido tā modeļus un variantus, lai veiktu nepieciešamās izmaiņas un uzlabojumus gala risinājuma idejā. |
| **Ar programmēšanas un programmvadāmo ierīču palīdzību iespējams atvieglot automatizējamus ikdienas procesus. (T.Li.2.)** IV Building or developing the product (Produkta izstrāde vai attīstīšana) | | | | | | | |
| 12/45,46 | | 2 | Lieto projekta vadības rīkus informācijas apmaiņai un laika plānošanai. (T.V.2.3.8.) | | Satādīt komandu ievērojot scrum projekta vadīšanas metodes: product owner, scrum master, teams, backlogs stickers, sprint review meeting. | [1.1], [1.2], [1.3] |
| 12/47,48 | | 2 | Plāno risinājumu, veidojot versiju pārvaldību un uzlabojumus risinājuma gaitā. (T.V.1.3.1, T.O.2.4.9.) | | 2.1.2. Salīdzina dažādas programmvadāmas ierīces un to izmantojamību sadzīves un ražošanā, tai skaitā datorvadāmās iekārtas datorizētu telpisku modeļu, digitālu rasējumu un attēlu izveidē. | [1.1], [1.2], [1.3] |
| 13/49,50 | | 2 | Savā risinājumā izmanto atbilstošus publiski pieejamus materiālus un krātuves, datus un informācijas avotus, lai papildinātu savas zināšanas programmatūras funkcionalitātes nodrošināšanā un uzlabotu risinājumu. (T.O.2.3.3.) | | Katra komandai iesniegt backlog sarakstu, (uzd izpildes ilgumam izmantot fibonači skaitļu)  3.2. Izmanto dažādu lietotņu piedāvātās dokumentu koplietošanas iespējas, nosakot atšķirīgiem lietotājiem atšķirīgas piekļuves, rediģēšanas un komentēšanas tiesības un iespējas. | [1.1], [1.2], [1.3] |
| 13/51,52  14/53,54 | | 2 | Sadarbojas programmatūras izstrādes procesā, izmantojot versiju vadības rīku. (T.O.2.4.9.) | | 2.3.2. Izmanto dažādās lietotnēs pieejamās programmēšanas iespējas, izstrādājot vienkāršas programmas (skriptus) piedāvāto datu apstrādes iespēju papildināšanai un lietotāja darba produktivitātes paaugstināšanai. | [1.1] |
| Labas komunikācijas prasmes un empātija palīdz iegūt kvalitatīvu informāciju no lietotājiem. (T.Li.1.) V Testing the product (Produkta testēšana) | | | | | | |
| 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 |
|  | | | | | | | |
| 14/55 | 1 | Izvēlas un izveido aptaujas anketas formas, organizē aptaujas un anketēšanu, veic iegūto datu manuālu un automatizētu apkopošanu. (T.V.2.3.4.) | | Iesniegt vismaz 3 ierosinājumus produkta uzlabošanai  1.3.2. Testēt un lietot radīto risinājumu iterācijas, analizēt iegūtos datus un formulēt pamatotus ierosinājumus, lai pilnveidotu risinājuma izstrādes darba plānu. | | [1.1] |
| 14/56 | 1 | Citu izstrādātāju risinājumos atpazīst informācijas atlases, attēlošanas un strukturēšanas pamatprincipus un šos pamatprincipus ievēro savos risinājumos. (T.V.2.3.10.) | | [1.1] |
| Autortiesības un licences nosaka programmatūras pieejamību, kā arī aizsargā izstrādātāja tiesības. (T.Li.2.) VI Deployment in the Market and Maintenance (Izvietošana tirgū un uzturēšana) | | | | | | | |
| 15/57 | 1 | Veidojot prezentāciju, izmanto vizuālus, audiālus un audiovizuālus elementus, kuri pielāgoti tās mērķim un mērķauditorijai. (T.V.2.3.10.) | | Veidojot prezentāciju, izmanto vizuālus, audiālus un audiovizuālus elementus, kuri pielāgoti tās mērķim un mērķauditorijai. (T.V.2.3.10.) | |  |
| 15/58 | 1 | Savā prezentācijā ietver galvenos programmatūras izstrādes posmus. (T.O.2.4.7.) | | Savā prezentācijā ietver galvenos programmatūras izstrādes posmus. (T.O.2.4.7.) | |  |
| 15/59,60 | 2 | Risinājuma izstrādē un lietošanā ievēro programmatūras licences nosacījumus, intelektuālā īpašuma un personas datu aizsardzības principus. (T.V.3.1.5.) | | Prezentēt savu darbu konkurenta grupai. Novērtēt katra komandas biedra ieguldījumu 10 ballu skalā. Novērtēt pretinieka komandas darbu 10 ballu skalā | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3. Kā automātiski apstrādāt un attēlot datus? 40h | | | | | | |
| 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 |
| Kvalitatīvs informācijas avots ir viens no priekšnosacījumiem augstas kvalitātes datu iegūšanā. (T.Li.2.) | | | | | | |
| 16/ 61-64 | 4 | Izmanto dažādas datu struktūras. (T.O.2.4.14.) | | 1.uzrakstīt savas datu [bāzes](https://macibas.e-skola.lv/mod/glossary/showentry.php?courseid=1454&eid=2902&displayformat=dictionary) struktūru uz papīra ( skatīt 10.slaidu) | |  |
| Sagatavo un rediģē lielus strukturētus dokumentus un izklājlapas. (T.V.2.3.3, T.V.2.3.5.) | | 2.ievadīt Excelī vismaz 7 rindas (ierakstus) par objektu   iesniegt šeit vienu datni, kurā ielikts datu bāzes struktūras papīra attēls  3.ieinstalēt ‘**DB Browser for SQLite**’ | | <https://youtu.be/r0xVcRroHJU>  [*www.youtube.com/watch?v=Ic6etzJZF-M&ab\_channel=selfedu*](http://www.youtube.com/watch?v=Ic6etzJZF-M&ab_channel=selfedu)  [*sqlitebrowser.org/dl/*](https://sqlitebrowser.org/dl/) |
| Datubāzes pareiza izveide atvieglo darbu ar datiem. (T.Li.2.) | | | | | | |
| 17/ 65-68 | 2 | Veido datubāzi. (T.O.2.3.4.) | | saprast import sqlite3 , connect()  cursor() , close() , execute(Create()) | | https://youtu.be/\_CWbiIF6vpI  repl.it  <https://www.python.org/dev/peps/pep-0249/>  <https://www.severcart.ru/blog/all/python_sqlite3/>  <https://docs.python.org/3/library/sqlite3.html>  SQLite\_01.pptx |
|  | 2 | Prast savienot pyton ar DB  izveidot tabulu savai DB | | [youtube.com/watch?v=TwnCXdCa8qg&t=3s&ab\_channel=selfedu](https://www.youtube.com/watch?v=TwnCXdCa8qg&t=3s&ab_channel=selfedu)  <https://proproprogs.ru/python> |
| Automatizācijas rīki atvieglo darbu ar apjomīgiem datiem. (T.Li.2.) | | | | | | |
| 18/ 69-72 | 4 | Organizē datu ievadi, veicot to validāciju. (T.O.2.4.13.) Piesaista programmai vienkāršu datubāzi. (T.O.2.4.17.) | | Saprast Relācijas tabulas, Ievadīt savu tabulu savā datubāzē, katram jābūt vismaz 7 ierakstiem https://youtu.be/mtKKCz\_MaoQ | | https://youtu.be/mtKKCz\_MaoQ  <https://pythonru.com/osnovy/sqlite-v-python> |
| 19/ 73-76 | 4 | Sagatavo un rediģē lielus strukturētus dokumentus un izklājlapas. (T.V.2.3.3, T.V.2.3.5.) | | iesniegt savas datu bāzes 2 tabulas https://macibas.e-skola.lv/mod/assign/view.php?id=34702 | |  |
| 20/ 77-80 | 2 | Darbam ar lieliem dokumentiem izmanto lietotnēs iebūvētos efektivitātes un automatizācijas rīkus. (T.V.2.3.3.) | | WORD dokumentā iekopēt 2 attēlus no DB Browser un 3 INSERT komadas iekopētas tā, lai varu iekopēt | |  |
| 2 | Veic datu apstrādi gatavā datubāzē, izmantojot vaicājumus, formas un pārskatus.  (T.V.2.3.7., T.O.2.3.4.) | | * Izveidot EXCELī datni ar   + id,   + 13 lietotāju vārdiem,   + parolēm, vecumu, * importēt datu bāzē | |  |
| Datu validācija aizsargā datubāzi. (T.Li.2.) | | | | | | |
| 21/ 81-84 | 2 | Novērš datu dublēšanos datubāzē. (T.O.2.3.4.) | | WORD dokumentā ievietot attēlus no pyton:   * + Izdrukāta datu bāze ar vairākiem vienādiem lietotāju vārdiem - repl.it vidē   + Izdrukāta datu bāze ar ievadītu esošo lietotāja vārdu un brīdinājumu, ka tāds jau eksistē un nedublēto to datu bāzē - repl.it vidē | |  |
| 2 | Risina vienkāršu datu apstrādes uzdevumu. (T.O.2.4.17.) Veic datu aprēķinus atbilstoši dotajiem kritērijiem. (T.V.2.3.5.) | |  |
| Formatē datu izvadi. (T.O.2.4.13.) | |  |
| Datus var apstrādāt un veidot kvalitatīvu to vizuālo noformējumu, lai tie būtu uzskatāmāki. (T.Li.2.) | | | | | | |
| 22/ 85-88 | 2 | Veic formu validāciju. (T.V.2.3.5.) | | Programmā pārbaudīt ievaddatus norādot kritērijus, kuriem jāatbilst vismaz viena lauka validācijai. Piemēram, datuma laukam vai lietotāja vecumam. | |  |
| 2 | Veic datu pārstrukturizāciju un to attēlošanu, izmantojot vizualizācijas. (T.V.2.3.6.) | |  |
| Programmas darbībai ir jābūt pietiekoši ātrai, lai lietotājam būtu ērti to lietot. (T.Li.2.) | | | | | | |
| 23/ 89,90 | 2 | Risina uzdevumu, izmantojot gatavu algoritmu. (T.O.2.4.16.) | | Pēc blokshēmas aprēķināt vidējo. | |  |
| 23/ 91,92 | 2 | Risina vienkāršu datu apstrādes uzdevumu. (T.O.2.4.17.) | | Pārbaudot piekļuves tiesības (*man tabulā lietotaaji*)- diferencēt tabulas (man staadiitDobees) lietošanas atļauju vai tikai apskatīt vai labot. | |  |
| 24/ 93,94 | 2 | Veic datu aprēķinus atbilstoši dotajiem kritērijiem. (T.V.2.3.5.) | | Jūsu piedāvājums | |  |
| Kārtošanas un meklēšanas algoritmi lietotājam atvieglo programmatūras lietojamību. (T.Li.2.) | | | | | | |
| 24/ 95,96 | 2 | Izvēlas uzdevuma izpildei atbilstošu datu kārtošanas un meklēšanas algoritmu. (T.O.2.4.19.) | |  | |  |
| Zinot algoritma sarežģītību, iespējams izvērtēt programmatūras izveides iespējamību un lietojamību atkarībā no izpildāmo darbību apjoma. (T.Li.2.) | | | | | | |
| 25/ 97,98 |  | Veic vienkāršotu programmatūras ātrdarbības novērtējumu. (T.O.2.4.16.) | |  | |  |
| 25/ 99,100 |  | Tests, kolēģu darba novērtēšana | | Summatīvais pārbaudes darbs. | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4. Kā sistemātiski pārbaudīt un atkļūdot risinājumu? 20h | | | | |
| 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 |
| Atkļūdo kļūdaini uzrakstītas programmas Atkļūdošana ir būtiska daļa no programmatūras izstrādes. (T.Li.2.) Rakstot kodu, iespējams pieļaut dažādu veidu kļūdas, kuras var atkļūdot, zinot kļūdas paziņojuma veidu. (T.Li.2.) Kļūdas var rasties, izmantojot nepareizus datu tipus. (T.Li.2.) | | | | |
| 26/ 101-104 | 4 | * Veic testēšanu un lieto versiju vadības rīkus, lai pilnveidotu programmatūras risinājumu. (T.V.1.3.2.) | Izveidot LKD algoritmu  Aplūko dotu programmu ar sintakses kļūdu koda redaktorā un pievērš uzmanību redaktora iezīmētajām vietām kodā un ieteikumiem. Izpilda dotu programmu un pievērš uzmanību pēdējai kļūdas paziņojuma rindai vai rindai pirms tās. Izmanto kļūdas paziņojumu, programmēšanas valodas dokumentāciju un koda redaktora ieteikumus sintakses kļūdas atrašanai, lai izlabotu kļūdas savā programmā.  Izpilda dotu programmu, kurā ir izpildlaika kļūda (*runtime error*), un pievērš uzmanību pēdējai kļūdas paziņojuma rindai un komandai. Lasa, kāds ir kļūdas veids, un salīdzina komandas izsaukumu, tās parametrus ar plānotajiem. Izmanto kļūdas paziņojumu un programmēšanas valodas dokumentāciju kļūdas atrašanai un izlabo to. |  |
| 27/ 105-108 | 4 | * Testējot dokumentē rezultātus un atbilstoši iegūtajiem rezultātiem veic programmatūras atkļūdošanu vai uzlabošanu. (T.V.1.3.2., T.O.2.4.6.) | Apskata atkļūdošanas rīkus, kuri ir pieejami izvēlētajā koda redaktorā (*IDE*): mainīgo vērtību monitors (*watch* *section*), pārtraukumpunkti (*breakpoint*), izpilde pa soļiem (*step in*, *step over*). Izvērtē šo rīku izmantošanas priekšrocības, salīdzinot ar informācijas izvadi standarta izvadē. Izmanto atkļūdošanas rīkus, lai atrastu un izlabotu kļūdas dotā programmā. |  |
| Pārbauda (testē) programmas darbību. Testējot var pārbaudīt, vai programmatūra strādā korekti un izpilda uzstādītās prasības. (T.Li.2.)Testējot programmu ar lietotājiem, iespējams atklāt nepieciešamos uzlabojumus programmas lietojamībā. (T.Li.1., T.Li.3.) | | | | |
| 28/ 109-112 | 4 | * Izstrādā testa komplektus atbilstoši konkrētam uzdevumam. (T.V.1.3.2.) | Ar dekorātoru pārbauda LKD algoritma darbību. Svarīgi testēt vismaz ar   * parametriem, kuri ir pareizi un kuriem ir zināma pareizā atbilde, – testēt rezultāta pareizību (piemēram, plkst. 10.00 atgriež tekstu “Labrīt!”); * parametriem, kuri nav pareizi, – testēt, vai tiek atgriezts pareizais kļūdas paziņojums (piemēram, plkst. 25.00 atgriež tekstu “Nepareizs laiks”); * parametriem, kuri ir tieši uz kādas izvēles robežas, – pārbaudīt, vai izvēle tiek veikta pareizi (piemēram, plkst. 12.00 atgriež tekstu “Labdien!”, nevis “Labrīt!”). |  |
| 29/ 113-116 | 4 | * Veic programmatūras izstrādi un testēšanu atbilstoši izstrādātai programmatūras prasību specifikācijai. (T.O.2.4.6.) | Izstrādā testa piemērus (ievades un izvades pārus) dotajam uzdevuma aprakstam, patstāvīgi aplūko doto programmu, veic testēšanu atbilstoši sastādītajiem testa piemēriem atrod un dokumentē kļūdas. |  |
| 30/ 117-120 | 4 | Analizē testēšanā iegūtos datus, lai identificētu pamatotus un nepieciešamus uzlabojumus, un ievieš uzlabojumus. (T.V.1.3.2.) | Salīdzina savus testa piemērus ar klases biedru testiem |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5. Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu? 40h | | | | |
|  |  |  |  |  |
| 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 |
| Atpazīst mašīnmācīšanās pamatprincipusMašīnmācīšanās principus iespējams izmantot ikdienas situācijās un dažādos risinājumos. (T.Li.2.)Mašīnmācīšanās palīdz kompleksās jomās, kurās nav vienkārši izmantojamu algoritmu. (T.Li.2.) | | | | |
| 31/ 121-124 | 4 |  | Bezmaksas tiešsaistes mašīnmācīšanās kursu sērija. Piereģistrēties, izpildīt līdz Taksonomijai, ieskaitot | [https://course.elementsofai.com/lv](https://course.elementsofai.com/lv/) |
| 32/ 125-128 | 4 | 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://deep­image.ai](https://deep-image.ai/) attēlu uzlabošanai, [https://deepdreamgenerator.com](https://deepdreamgenerator.com/) un [https://deepart.io](https://deepart.io/) attēlu veidošanai, [https://www.autodraw.com](https://www.autodraw.com/) zīmēšanai |
| 33/ 129-132 | 4 | 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. – un **prezentē** apkopoto informāciju par mašīnmācīšanās pielietojumu kādā sevi interesējošā jomā. | https://experiments.withgoogle.com/ai%C2%ADduet |
| 34/ 133-136 | 4 | Skaidro atšķirības starp mākslīgo intelektu un vienkāršu algoritmu izmantošanu dažādās dzīves situācijās. (T.O.2.4.18., T.O.3.2.5.) | 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. |  |
| 35/ 137-140 | 4 | Jaunumi MI |  |  |
| 11.klase | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6. Kā īstenot risinājumu? 50h | | | | |
|  |  |  |  |  |
| 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 |
|  |  |  |  |  |
|  |  |  |  |  |

# 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