



Valsts izglītības  
satura centrs

**Programmēšana  
(augstākais mācību satura  
apgaves līmenis)**

**Ieteikumi, gatavojoties  
eksāmenam  
2022./2023. mācību gadā**

Anita Līva  
Marina Juzova

## Satura rādītājs

Ievads .....	3
1. Valsts pārbaudes darba struktūra .....	4
2. Valsts pārbaudes darba piekļuves nosacījumi .....	6
Dažas idejas projektu veidošanai: .....	6
Dokumentācijai jāsaturs visi programmatūras dzīves cikla etapi: .....	6
Minimālās prasības programmvadāmajam risinājumam: .....	6
Augšupielāde .....	7
3. Katras daļas specifika, uzdevuma paraugs ar risinājumu un vērtējumu .....	8
1. daļa. Datortīkls un droša datubāze (DB) .....	8
2. daļa. Programmatūras dzīves cikls .....	13
3. daļa. Objektorientētā programmēšana (OOP) un ārējās bibliotēka .....	17
4. daļa. Dažādas datu struktūras, programmsaskarne (API) un mašīnmācīšanās principi ...	22
4. Ieteikumi, kā gatavoties eksāmenam .....	26
5. Palīg līdzekļi, kurus atļauts izmantot eksāmena laikā .....	26
Ieteicamās vietnes .....	26
Pielikums .....	27
Šogad VPD izmantos SR, kuri norādīti eksāmena paraugā: .....	27
VPD piekļuves darba dokumentācija .....	28

## **Ievads**

Metodiskais materiāls par gatavošanos 2022./ 2023. m.g. centralizētajam eksāmenam programmēšanā (augstākajā mācību satura apguves līmenī) paredzēts gan skolēniem, gan programmēšanas skolotājiem, kuri gatavo skolēnus.

Metodiskā materiāla pirmajā daļā ir izskaidrota eksāmena struktūra, otrajā daļā izskaidrots, ko nozīmē eksāmena pielāides darbs un kādas ir prasības tā iesniegšanai. Šī materiāla trešajā daļā ar piemēriem ir raksturoti eksāmena uzdevumu veidi, to veidošanas un vērtēšanas pamatprincipi. Ceturtajā daļā apkopots vielas atkārtēšanas plāns. Piektajā daļā norādīti palīglīdzekļi, kurus atļauts izmantot eksāmena laikā. Patstāvīgajam darbam norādīti ieteicamie tīmekļa resursi.

Ceram, ka šis metodiskais materiāls palīdzēs veiksmīgi sagatavoties eksāmenam un mazinās bažas par ieviesto centralizēto eksāmenu.

Paldies par sadarbību metodiskā materiāla sagatavošanā vērtēšanas ekspertiem, ārštata metodiķiem – Anitai Līvai un Marinai Juzovai.

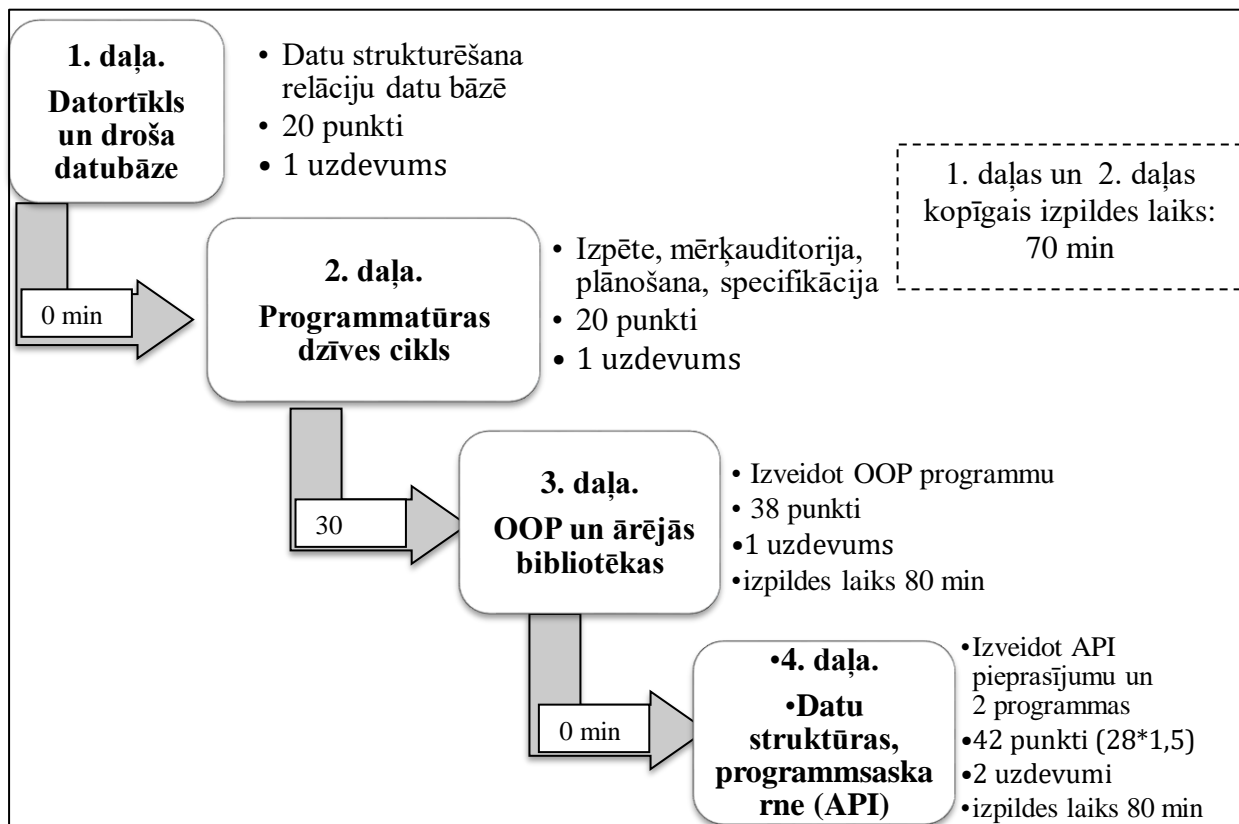
Kaspars Špūle

VISC Vispārējās izglītības pārbaudījumu nodaļas vadītājs

Rīgā, 2022. gada novembrī

## 1. Valsts pārbaudes darba struktūra

Programmēšanas augstākā mācību satura apguves līmeņa valsts pārbaudes darba (turpmāk – VPD) adresāts – skolēni, kuri ir apguvuši tehnoloģiju jomas sasniedzamos rezultātus (turpmāk – SR) optimālajā un augstākajā mācību satura apguves līmenī atbilstoši mācību priekšmetu kursiem “Programmēšana I” un “Programmēšana II”. VPD norise plānota 17. maijā atbilstoši 1. attēla struktūrai.



1. attēls. Valsts pārbaudes darba struktūra

1. daļā dots situācijas apraksts, kuram jāmodelē datu bāze. Uzmanīgi izanalizē un izplāno relāciju (tabulu savienošana) datu bāzes struktūrai. Atver doto WORD datni un ieraksti tajā datu bāzes struktūru norādītajās tabulās. Nemaini kolonu nosaukumus, bet zem tiem ieraksti atbilstošās vērtības. Divu tabulu nosaukumi ir doti, bet pārējām tabulām nosaukums tev jāpiešķir pašam. Piemēru katram datu bāzes laukam raksti vienu, nesajauc vienskaitli ar daudzskaitli. Savieno pareizos laukus, lai veidotu savstarpēji saistītas tabulas (relāciju). Uzmanību: dotie melnraksti programmēšanā netiek laboti! Saglabā Word datnē visu darbu un tad to iesniedz!

2. daļā dota problēma, kurai tev jānorāda: piemērotākās izpētes metodes, mērķauditorija, izpētes procesa plāns un programmatūras prasību specifikācijas programmatūras produkta izstrādei [1]. Testos ir jautājumi, kuriem nav dotas atbildes, bet tev tās pašam jāieraksta brīvā formā.

3. daļā doti 2 uzdevumi, kuriem jāiesniedz programmas, izmantojot objektu orientētās programmēšanas (turpmāk OOP) paradigmu.

4. daļā doti 2 uzdevumi. Pirmajā uzdevumā nosaukts API domēns, kuram jāizveido norādītais pieprasījums. Pieprasījums var būt arī tukšs, tiks pārbaudīti veidotie pieprasījumi un programma, kura apstrādā iegūto informāciju.

Otrajā uzdevumā jāveido programma, kurā dotajai teksta datnei izpilda norādītos uzdevumus, atkārtoti darbu ar teksta datņu nolasīšanu, apstrādāšanu. Ja tev nesanāk nolasīt doto tekstu, tad izveido virknes mainīgo ar tekstu, kuru turpini apstrādāt, lai iegūtu pārējos punktus.

Visām programmām noteikti jābūt norādītai programmēšanas valodai. Darbu saglabā norādītajā datnes formātā un iesniedz to. Ievēro labās prakses principus koda pierakstā, noteikti raksti komentārus un ievēro atkāpes.

## 2. Valsts pārbaudes darba piekļuves nosacījumi

Lai piekļūtu VPD, jāizstrādā projekta darbs un jāiesniedz ne vēlāk kā astoņas nedēļas pirms eksāmena norises (2023. gadā – ne vēlāk par 22.martu) programmvadāmo risinājumu un tā dokumentāciju. VPD piekļuves darbu izglītības iestāde novērtē ne vēlāk kā divas nedēļas pirms eksāmena norises (2023. gadā – līdz 5.aprīlim).

### Dažas idejas projektu veidošanai:

- Uzrakstīt klonu failiem.lv.
- Izstrādāt GIS risinājumu, kas izmanto Google API.
- Uzrakstīt šaha vai dambretes programmu u.t.t.

### Dokumentācijai jāsaturs visi programmatūras dzīves cikla etapi:

Ieteicamie, bet ne obligātie vērtēšanas kritēriji:

Programmatūras dzīves cikla etapi	Punkti
1. Problēmas izpēte un analīze - izpētes metodes izvēle un pamatojums, izpētes procesa apraksts, izpētes datu procesa apraksts, izpētes datu apkopojums.	3
2. Programmatūras prasību specifikācija - risinājuma mērķauditorijas izvēle un tās raksturojums, programmvadāmā programmatūras izstrādes projekta risinājuma un tā Ievaddatu /Izvaddatu funkciju apraksts.	14
3. Programmatūras izstrādes plāns ar pamatojumu.	3
4. Programmatūras kods, kas veidots ievērojot labās prakses principus – jāievieto pielikumā. 4.1. Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam. (1p) 4.2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā (1p) 4.3. Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām. (6p) 4.4. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai (4p) 4.5. Ievērojot labās prakses OOP principus izveidota vismaz viena bāzes klase ar 2 atvasinātajām klasēm (4p) 4.6. Datu ievadīšana / izvadīšana / rediģēšana (4p)	20
5. Atklūdošanas un akcepttestēšanas pārskats.	6
6. Lietotāja ceļvedis.	2
7. Piemērotās licences pamatojums	1

Piekļuves materiālus izglītības iestādes skolotājs vērtē ballēs atbilstoši izglītības iestādes saistošai izglītojamo mācību sasniegumu vērtēšanas kārtībai un ne vēlāk kā sešas nedēļas pirms eksāmena norises dienas vērtējumu ievada VPS. Izglītojamais eksāmenu drīkst kārtot, ja vērtējums par piekļuves materiālu nav zemāks par četrām ballēm.

### Minimālās prasības programmvadāmajam risinājumam:

- Dokumentācijā līdz 15 A4 lapām, neskaitot titullapu, satura rādītāju un pielikumu.

- Datu bāze ar vairākām tabulām.
- Pielieto vismaz vienu bibliotēku un/vai API.
- Pielieto dažādas datu struktūras.
- Ir realizēta lietotāja piekļuves datu aizsardzība.

### **Augšupielāde**

- Līdz 22.03.2023 , t.i. ne vēlāk kā 8 nedēļas pirms eksāmena dienas, informācijas sistēmā jāpievieno izstrādātais projekta darbs arhivētā formātā.
- Izglītojamie, kuri eksāmenu kārtos augstskolā, piekļuves materiālus neiesniedz.
- Pedagoģs materiālus izvērtē ne vēlāk kā sešas nedēļas pirms eksāmena norises dienas un vērtējumu ievada informācijas sistēmā.

### 3. Katras daļas specifika, uzdevuma paraugs ar risinājumu un vērtējumu

#### 1. daļa. Datortīkls un droša datubāze (DB)

Skolēniem jābūt informētiem par iespējamajiem apdraudējumiem, bet 2022./2023. m.g. konfigurēšanu nepārbaudīs praktiskajā līmenī.

Eksāmenā jāmodelē DB dotajai situācijai: jāprot strukturēt datus, veidot savstarpēji saistītās tabulas. Tabulas būs uzzīmētas, doti savienojošie elementi, no kuriem būs jāizvēlas pareizie un jāpievieno pareizā virzienā. Jāprot atpazīt un lietot DB komandas, bet 2022./2023. m.g. nebūs jāizveido DB konkrētā programmēšanas valodā vai ar kādu no DBVS.

#### Uzdevums (20 punkti)

Analizē pieejamo informāciju un zemāk dotajās tabulās izplāno relāciju datubāzes struktūru (aizpildi tikai tos tabulu laukus, kuri, tavuprāt, nepieciešami):

- izplāno atbilstošus laukus, kas atvieglo datu apstrādi,
- norādi atbilstošus datu tipus,
- ieraksti vienu datu piemēru katrā no laukiem,
- sasaisti tabulas ar atbilstošu relāciju tipu.

**Situācijas apraksts:** Pēterim nepieciešama datu bāze, kura saturēs savstarpēji saistītas tabulas. Šoreiz datu bāzei jānorāda, kurš skolēns pie kura skolotāja kādu priekšmetu mācās, bet ar laiku Pēteris uzlabos to, lai tā ietvertu visas skolas skolēnus un skolotājus.



Atrisinājums:

Tabula Skolotajs

Lauka nosaukums	Datu tips	Datu piemērs
skolotaja_id	TEXT	kset45q-23gfd-kvc72-laxm79-bz
varda	TEXT	Sandra
uzvards	TEXT	Skudra
mob_nr	TEXT	29010101

Tabula Prieksmets

Lauka nosaukums	Datu tips	Datu piemērs
skolotaja_id	TEXT	kset45q-23gfd-kvc72-laxm79-bz
skolena_id	TEXT	Tdfr73q-28qld-xzc12-lhd9-cmh
prieksmets	TEXT	Matemātika

Tabula Skolens

Lauka nosaukums	Datu tips	Datu piemērs
skolena_id	TEXT	Tdfr73q-28qld-xzc12-lhd9-cmh
varda	TEXT	Alīna
uzvards	TEXT	Krauze
mob_nr	TEXT	29010102
klase	TEXT	11.a

Tabula

Lauka nosaukums	Datu tips	Datu piemērs

<p>Risinājums</p> <table border="1" data-bbox="209 342 667 510"> <thead> <tr> <th colspan="2">Prieksmeti</th> </tr> <tr> <th>uzvards</th> <th>prieksmets</th> </tr> </thead> <tbody> <tr> <td>Lapiņš</td> <td>Latviešu valoda, literatūra</td> </tr> <tr> <td>Čakāne</td> <td>Matemātika, datorika, programmēšana</td> </tr> </tbody> </table>	Prieksmeti		uzvards	prieksmets	Lapiņš	Latviešu valoda, literatūra	Čakāne	Matemātika, datorika, programmēšana	<p>Risinājuma analīze</p> <p>Laukā 'Prieksmets' nedrīkst būt masīvs, <i>laukā</i> jābūt tikai <i>vienai vērtībai</i>, pēc kuras var atlasīt informāciju. <i>Lauka nosaukumā nedrīkst būt tukšumi un diakritiskās zīmes.</i></p>													
Prieksmeti																						
uzvards	prieksmets																					
Lapiņš	Latviešu valoda, literatūra																					
Čakāne	Matemātika, datorika, programmēšana																					
<table border="1" data-bbox="209 544 775 779"> <thead> <tr> <th colspan="2">Prieksmeti</th> </tr> <tr> <th>uzvards</th> <th>prieksmets</th> </tr> </thead> <tbody> <tr> <td>Lapiņš</td> <td>latviešu valoda</td> </tr> <tr> <td>Lapiņš</td> <td>literatūra</td> </tr> <tr> <td>Čakāne</td> <td>matemātika</td> </tr> <tr> <td>Čakāne</td> <td>datorika</td> </tr> <tr> <td>Čakāne</td> <td>programmēšana</td> </tr> </tbody> </table>	Prieksmeti		uzvards	prieksmets	Lapiņš	latviešu valoda	Lapiņš	literatūra	Čakāne	matemātika	Čakāne	datorika	Čakāne	programmēšana	<p>Pārkāpta 1.normālforma: “Katrā laukā tikai viena vērtība un ieraksti neatkārtojas”.</p>							
Prieksmeti																						
uzvards	prieksmets																					
Lapiņš	latviešu valoda																					
Lapiņš	literatūra																					
Čakāne	matemātika																					
Čakāne	datorika																					
Čakāne	programmēšana																					
<table border="1" data-bbox="209 801 624 958"> <thead> <tr> <th colspan="2">Skolotāji</th> </tr> <tr> <th>skolotāja_id</th> <th>uzvards</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Lapiņš</td> </tr> <tr> <td>2</td> <td>Čakāne</td> </tr> </tbody> </table>	Skolotāji		skolotāja_id	uzvards	1	Lapiņš	2	Čakāne	<p>Skolotājam vēl jānorāda vārds, personas kods, konts un pārējie dati nākamajos laukos (nākamajās kolonnās).</p>													
Skolotāji																						
skolotāja_id	uzvards																					
1	Lapiņš																					
2	Čakāne																					
<table border="1" data-bbox="209 992 624 1126"> <thead> <tr> <th colspan="2">Skolēni</th> </tr> <tr> <th>skolēni_id</th> <th>uzvards</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Lapiņš</td> </tr> <tr> <td>2</td> <td>Čakāne</td> </tr> </tbody> </table>	Skolēni		skolēni_id	uzvards	1	Lapiņš	2	Čakāne	<p>Skolēniem vēl jānorāda vārds, personas konts, klase un pārējie dati nākamajos laukos (nākamajās kolonnās).</p>													
Skolēni																						
skolēni_id	uzvards																					
1	Lapiņš																					
2	Čakāne																					
<table border="1" data-bbox="209 1160 775 1429"> <thead> <tr> <th colspan="3">Prieksmeti</th> </tr> <tr> <th>prieksmeta_id</th> <th>skolotāja_id</th> <th>prieksmets</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>latviešu valoda</td> </tr> <tr> <td>2</td> <td>1</td> <td>literatūra</td> </tr> <tr> <td>3</td> <td>2</td> <td>matemātika</td> </tr> <tr> <td>4</td> <td>2</td> <td>datorika</td> </tr> <tr> <td>5</td> <td>2</td> <td>programmēšana</td> </tr> </tbody> </table>	Prieksmeti			prieksmeta_id	skolotāja_id	prieksmets	1	1	latviešu valoda	2	1	literatūra	3	2	matemātika	4	2	datorika	5	2	programmēšana	<p>Priekšmetos redzam, kurš skolotājs kuram skolēnam māca kādu priekšmetu, var izmantot klases, grupas.</p>
Prieksmeti																						
prieksmeta_id	skolotāja_id	prieksmets																				
1	1	latviešu valoda																				
2	1	literatūra																				
3	2	matemātika																				
4	2	datorika																				
5	2	programmēšana																				
<table border="1" data-bbox="209 1552 762 1720"> <thead> <tr> <th colspan="3">Tabula Skolotājs</th> </tr> <tr> <th>Lauka nosaukums</th> <th>Datu tips</th> <th>Datu piemērs</th> </tr> </thead> <tbody> <tr> <td>Skolotāja ID</td> <td>TEXT</td> <td>1,2,3,4,5</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> </tbody> </table>	Tabula Skolotājs			Lauka nosaukums	Datu tips	Datu piemērs	Skolotāja ID	TEXT	1,2,3,4,5	...	...	...	<p>1) Lauka nosaukumā nedrīkst būt tukšumi un diakritiskās zīmes.  2) Ja datu bāzē ir sensitīvi dati, piemēram, bankas konts, tad id jāveido unikāls kā TEXT tips, piemēram, gfdhj45df-ghkti-39kg97-fhso94vnmnl, bet, ja datu bāze tiek veidota kā izmēģinājuma vai lokālā, tad drīkst palikt skaitlisks datu tips autoNumber  3) “Datu piemērs” ir vienskaitlī, tāpēc 1,2,3,4,5 atbilde ir nepareiza.</p>									
Tabula Skolotājs																						
Lauka nosaukums	Datu tips	Datu piemērs																				
Skolotāja ID	TEXT	1,2,3,4,5																				
...	...	...																				
<table border="1" data-bbox="209 1933 762 2042"> <thead> <tr> <th colspan="3">Tabula Skolotājs</th> </tr> <tr> <th>Lauka nosaukums</th> <th>Datu tips</th> <th>Datu piemērs</th> </tr> </thead> <tbody> <tr> <td>skolotāja_id</td> <td>TEXT</td> <td>1</td> </tr> </tbody> </table>	Tabula Skolotājs			Lauka nosaukums	Datu tips	Datu piemērs	skolotāja_id	TEXT	1	<p>Ja piemērā ir skaitlis, tad datu tipam jābūt skaitliskam AutoNumber.</p>												
Tabula Skolotājs																						
Lauka nosaukums	Datu tips	Datu piemērs																				
skolotāja_id	TEXT	1																				

...	...	...	
Tabula Prieksmeti			Pareizi.
Lauka nosaukums	Datu tips	Datu piemērs	
skolotaja_id	AutoNumber	1	
...	...	...	
Tabula Skolotajs			Pareizi. Dati sākotnēji tiek replicēti uz ārējo tīklu un tur publicēti mājas lapā. Tā rīkojas Valsts Zemes dienests, tādā gadījumā drīkst būt tikai drošs ID.
Lauka nosaukums	Datu tips	Datu piemērs	
skolotaja_id	TEXT	gfdhj45df-ghkti-39kg97-fhso94vnmdl	
...	...	...	
Tabula Skolotajs			1) Lauka nosaukumā nedrīkst būt punkts, tukšums, pieturzīmes, drīkst būt pasvītrojuma zīme. 2) Telefona numuru bieži veido kā TEXT tipa lauku. Telefona nr. ir personas dati, un uz to attiecas datu aizsardzības regula, tāpēc to nelieto kā ID.
Lauka nosaukums	Datu tips	Datu piemērs	
skolotaja_id	TEXT	gfdhj45df-ghkti-39kg97-fhso94vnmdl	
Tel. numurs	number	12345678	
...	...	...	
Tabula Prieksmeti			klases_id un skolotaja_id ir lauki, caur kuriem veido relācijas saiti ar priekšmetu, tāpēc AutoNumb <i>tikai</i> prieksmeta_id laukam, pretējā gadījumā AutoNumber automātiski palielinās par 1 katrā ierakstā, kurus nevarēs jēgpilni piesaistīt citām tabulām.
Lauka nosaukums	Datu tips	Datu piemērs	
prieksmeta_id	AutoNumb	1	
skolotaja_id	AutoNumb	1	
klases_id	AutoNumb	1	
...	...	...	
			Lauka nosaukumā nedrīkst lietot diakritisko zīmi 'ā'.
Lauka nosaukums	Datu tips	Datu piemērs	
skolotaja_id	TEXT	gfdhj45df-ghkti-39kg97-fhso94vnmdl	
Vārds	TEXT	Anna	
...	...	...	

Daži ieteikumi:

- Parasti DB glabā pamatdatus, bet dažreiz ir specifiskas vajadzības glabāt arī starprezultātus, tomēr VPD starprezultātiem neveido lauku, jo tad tiek pārkāpta 3.NF.
- Katrai tabulai izveido lauku unikālai ierakstu identifikācijai.
- SQLITE ir NULL, INTEGER, REAL, TEXT un BLOB tipa lauki, ja norādīta valoda PYTHON, tad datumam ir TEXT tipa lauks.
- Daļskaitlim standarts programmēšanā ir punkts.
- Svaru var glabāt gramos un naudu centos, ja pamato ar pareizu piemēru.
- Matemātiskajām operācijām tikai skaitliskās vērtības.
- Par primāro atslēgu nelikt telefona numurus, e-pastus, kuri samazina DB drošību.
- Telefona numura lauku parasti lieto kā TEXT tipa, tāpēc ka ar tiem neveic matemātiskās darbības; ja izvēlēties skaitlisko tipu, lai aizņemtu mazāk atmiņas, tad tas arī ar piemēru jāpamato. Telefona numurus profesionāļi veido kā telefona tipu, bet pārbaudes darbā ar piemēru nevarēsi ātri pierādīt, tāpēc šis tips nav ieteicams.
- Ar piemēru pamatot – pareizi uzrakstīt piemēru, nevis piemērus.

## 2. daļa. Programmatūras dzīves cikls

### Uzdevums (20 punkti)

#### Situācijas apraksts:

Kosmetoloģijas salona īpašniecei Dainai ir pastāvīgi apmeklētāji, kuri vēlas ierasties vienā un tajā pašā laikā pie viena un tā paša darbinieka, bet ir apmeklētāji, kuriem plāni bieži mainās.

Dainai būtu noderīgs digitālais rīks ar dažādām datu atlasēs iespējām un vizualizācijām, kas konkrētos datus attēlo viņai saprotamā un ērtā formātā, kurš laicīgi brīdinātu par katra klienta ierašanās laikiem, lai nepierakstītu 2 klientus vienā laikā, lai darbiniekiem izmaksātu laicīgi nopelnīto algu.

Uzdevums par doto situācijas aprakstu	Risinājums	Risinājuma analīze
Uzraksti vienu piemērotāko izpētes metodi dotajai situācijai	Aptauja/intervija	Šajā gadījumā var aptaujāt pasūtītāju, darbiniekus
	Dokumentu apstrāde	Šeit nav dotas *.csv datnes, tāpēc nepareizi, bet, ja tās ir dotas, tad ir jābūt dokumentu apstrādei
Pamatojums.	Aptaujāšu lietotāju, lai saprastu kādu veidot lietotni	Ja darbinieki nelietos, tad viņus nav jāapjautā.
Definē precīzu mērķauditoriju, kuru iesaistīs izpētes metodē	Kosmetoloģijas salona īpašniece	Mērķauditorija ir digitālā rīka lietotājs. Ja darbinieki nelietos, tad viņi nav mērķauditorija, šajā gadījumā mērķauditorija ir tikai īpašniece.
	Kosmetoloģijas salona apmeklētāji	Šajā gadījumā nepareizi. Šāda atbilde būtu pareiza, ja pasūtītu mājas lapas vietni vai reklāmu apmeklētāju piesaistei.
Sastādi dotās situācijas izpētes procesa plānu: 1. 2. 3. 4. 5. Izdaru secinājumu 6.	1. Apskatu datus 2. Izdomāju, ko gribu redzēt 3. Sāku rakstīt vaicājumu 4. Veidoju vaicājumus, atlasot datus, kurus gribu redzēt 5. Izdaru secinājumu	Šeit veido vaicājumus datu bāzei. 6. punkts neaizpildīts.
	1. Izveidoju prototipu 2. Iedodu lietotājiem aprobācijai 3. Apskatos informācijas izmantošanas tendenci 4. Uztaisu nākamo prototipu ar izmantotāko informāciju 5. Testēju 6. Secinu	Šeit aprakstīti gandrīz viss <b>produkta</b> dzīves posms, nevis izpētes procesa <b>plāns</b> . 5. punkts bija dots, tāpēc tas nav jālabo.
	1. Analizēju iegūtos datus 2. Projektēju digitālo līdzekli 3. Izstrādāju digitālo līdzekli 4. Skenēju un testēju izstrādāto 5. Dokumentēju izstrādāto līdzekli 6. Atbalstīšu ekspluatāciju	Aprakstīts produkta <b>izstrādes</b> process, nevis tas, kā <b>plāns</b> situācijas aprakstu.

	<p>1. <b>Sagatavošanās posms</b> – izveidoju aptaujas</p> <p>2. <b>Norise</b> - intervēju īpašnieci, digitālā rīka lietotāju, par informācijas lietošanas vēlamajiem veidiem.</p> <p>3. <b>Rezultātu apkopošana</b> - no aptaujas rezultātiem noskaidrošu lietotāja nepieciešamības.</p> <p>4. <b>Izvērtēšana</b> – apkopojū iegūto informāciju, piedāvāju Dainai informācijas galējo rezultātu.</p> <p>5. <b>Secināšana</b> - Izdaru secinājumus, lai varētu veikt turpmākās darbības.</p> <p>6. <b>Secinājumu formulēšana.</b> Formulēju secinājumu un piedāvāju vajadzības saistībā ar programmatūras produkta izstrādi.</p>	Dotās situācijas izpētes plāns.
Ņemot vērā uzdevumā doto problēmsituāciju un tajā pieejamo informāciju, īsi apraksti, kādu programmvadāmo risinājumu ir nepieciešams izstrādāt	<p>Apmeklētāju uzskaites dažādība apakšdatnes risinājuma *.csv</p> <p>Salona viesus raksturojoša informācija un statistika.</p> <p>Datubāze - jo tajā var glabāt daudz datus.</p> <p>Izstrādāšu aplikāciju ņemot vērā lietotāju specifiku.</p> <p>Tiks izveidota viegli pārskatāma mājaslapa, kurā varēs nolasīt no datu bāzes datus, apskatīt, labot un saglabāt. Ievadīt jaunus datus</p>	<p>Lietotājam nebūs ērti atvērt/izlabot/saglabāt *.csv datnes – tas nav programmrisinājums.</p> <p>Tas nav programmrisinājums.</p> <p>Tas nav programmrisinājums.</p> <p>Ir programmvadāms risinājums.</p> <p>Ir programmvadāms risinājums.</p>
Īsi apraksti programmvadāmā risinājuma datu uzglabāšanas veidu un datu fizisko izvietojumu	<p><b>MySQL datubāzē skaisti viss sakārtots tabulā.</b></p> <p>Oracle datubāze, kas izvietota uzņēmuma serverī.</p> <p>Lokālajā failu serverī ar datubāzi.</p> <p>Datus uzglabā MySQL datubāzē, kas izvietota nomas hostinga serverī.</p>	<p>Datu bāze nosaukta pareizi, bet jāatzīmē arī, kur tā atrodas</p> <p>Pareizās atbildes</p>
Kādi ievaddati tiks doti?	Klientu dati, apmeklēšanas datumu un laiku.	Šeit drīkst vienkārši, pat latviski aprakstīt ievaddatus

	Darbinieku dati, darba dienas, nostrādātās stundas.	
Ievaddatu tips	<pre> Persona{   personas_id – skaitlisks,   veidots-generēts   vards (TEXT)   uzvards (TEXT)   mob_talr (TEXT) } Alga{   darbinieka_id (TEXT)   klienta_id (TEXT)   ddiena (TEXT/Date)   cena (Float)   kvalit(INT/Bool)   plkst(INT) } </pre>	<p>Persona ir bāzes klase, no kuras jāatvasina Darbinieka un Klienta klase. Darbiniekam vēl jābūt personas kodam un konta numuram.</p> <p>Nepieciešama arī klase Alga kvalit(INT/Bool) – ja klients nav apmierināts, tad darbu neieskaita plkst(INT) – tikai precīzi laiki.</p>
Ievaddatu mērvienības, iespējamās vērtības	<p><b>klients1</b>=Klients(Kate, Krūmiņa, 28765431)</p> <p><b>darbinieks1</b>=Darbinieks(Dina, Dobe, 020200-12345, 21345678)</p> <p><b>alga</b> = Alga(22.12.12, 15:00, 35.00 eiro, 1,)</p>	Pareizi.
Kādā veidā tiks iesniegti ievaddati (izsaucot funkciju, no standarta ievades, no datnes, no DB)	informāciju nolasīs no datu bāzes, esošos telefonu numurus piedāvās izvēlēties saskarnē no saraksta, ievadīs jaunās personas, datus un labos tos arī saskarnē.	Pareizi.
Metodes nosaukums un deklarācija	<p>Metodes nosaukums un deklarācija?</p> <p>aprekina_algu(/Real/Float/)</p> <p>aprekina_pelnu(/Real/Float/)</p>	Izmanto ‘camelCase’ vai ‘snake_case’ stilu, bez diakritiskajām zīmēm, bez pieturzīmēm, bez tukšumiem
Kā iegūt (aprēķināt, kombinēt, formatēt) rezultātu?	<p>Katram darbiniekam aprēķināt nopelnīto uzņēmumam un pareizināt ar 20%</p> <p>Nopelnīto mēnesī aprēķināt, saskaitot visu darbinieku nopelnīto – darbiniekiem izmaksāto</p>	Pareizi.
Kā jāsniedz atbilde (atgriežot vērtību no funkcijas, izvadot atbildi uz ekrāna, datnē, DB)?	<p>Uz ekrāna izvadīt, kurš klients uz kuru datumu pierakstīts, kurš darbinieks kurā datumā cikos strādā,</p> <p>Ievadot darbinieka personas kodu, izdrukāt nopelnīto.</p>	

<p>Izvaddatu tips, mērvienības, iespējamās vērtības?</p>	<p>Saskarnē /TEXT/:</p> <p>izvaddati: Dora Bīlava 2022.gada decembrī alga 450.00 eiro</p> <p>izvaddati: Dora Bīlava 24.12.2022 plkst 10 12 14</p> <p>izvaddati: Kristīne Kārkle 13.septembrī 15:00</p> <p><b>Datu bāzē</b> Ievadītos datus par klientu, darbinieku, pakalpojuma cenu tādās pašās mērvienībās.</p>	<p>Ja ievaddati precīzi aprakstīti, tad var norādīt uz tiem, ja tādi tie tiks saglabāti.</p>
<p>Kāds programmatūras izstrādes modelis būs piemērotākais konkrētā programmvadāmā risinājuma izstrādē? Pamato, kāpēc!</p>	<p>Ūdenskrituma, t.i. kāda no lineārām modeļiem</p> <hr/> <p>Agile (Spējais), t.i. iteratīvais modelis</p>	<p>Nepareizi, jo Daina nav precīzi definējusi uzdevumu un nav brīdinājusi, ka nesadarbosies. Lineāros modeļus izmanto, kad pasūtītājs iedod apstiprinātu projektu un vairs nesadarbosies.</p> <p>Pareizi. Iteratīvos modeļus izmanto, kad pasūtītājs vēlas sadarboties, lai veidotu mūsdienīgu, pieprasītu produktu, regulāri sadarboties, atļaujot uzlabot produktu</p>



### 3.daļa. Objektorientētā programmēšana (OOP) un ārējās bibliotēka

#### Darbs ar datnēm

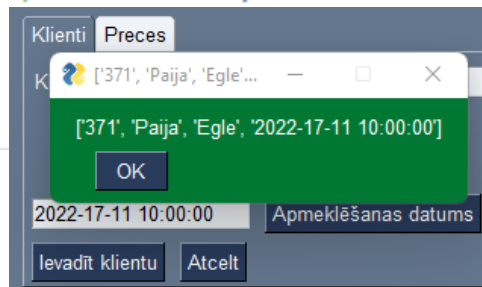
Atrod atšķirību kodā, lai saprastu, kāpēc tā drukā

```
def saglaba(self, dati): ##### saglaba(self,dati)
    self.dati=dati #delimiter=" ",
    with open('kontakti2.csv','w',newline='') as csv_datne:
        ierakstit = csv.writer(csv_datne, delimiter=',')
        for rinda in self.dati:
            ierakstit.writerow(self.dati)
```

```
1 ['numurs','varda','uzvards','datums'],['372','Maija','Liepa','2022-17-11 11:00:00'],['372','Maija','Liepa','2022-17-11 11:00:00']
2 ['numurs','varda','uzvards','datums'],['372','Maija','Liepa','2022-17-11 11:00:00'],['372','Maija','Liepa','2022-17-11 11:00:00']
3 ['numurs','varda','uzvards','datums'],['372','Maija','Liepa','2022-17-11 11:00:00'],['372','Maija','Liepa','2022-17-11 11:00:00']
```

```
def saglaba(self, dati): ##### saglaba(s
    self.dati=dati #delimiter=" ",
    with open('kontakti2.csv','w',newline='') as csv_datne:
        ierakstit = csv.writer(csv_datne, delimiter=',')
        for rinda in self.dati:
            ierakstit.writerow(rinda)
```

numurs	varda	uzvards	datums
371	Paija	Egle	2022-17-11 10:00:00
372	Maija	Liepa	2022-17-11 11:00:00



#### Uzdevums (38 punkti):

Izmantojot bāzes klasi un no tās atvasinātās klases, izveido programmu, kura aprēķina kvadrāta, taisnstūra un trijstūra perimetrus. Izdrukā katras figūras nosaukumu un perimetru, ja tas ir iespējams.

#### Piemēram:

Ievaddati:

'Trijstūra malu garumi : ',3,4,5

'Taisnstūra malu garumi : ', 6,7

'Kvadrāta malas garums : ', 8

#### Izvaddati:

Trijstūra perimetrs ir 12

Taisnstūra perimetrs ir 26

Kvadrāta perimetrs ir 32

#### Risinājums, ievērojot labo praksi:

```
class Figura: # Define bāzes klasi

    def __init__(self, vards): # konstruktors
        self.vards = vards # figūras nosaukums

    def perimetrs(self): # metode perimetra aprēķināšanai
        pass #tuksha metode
```

```

def druka(self): # drukaas figuras nosaukumu un perimetru
    self.rez = self.perimetr() # iegust izsauktaa objekta perimetru
    return f'{self.vars} {self.rez}' # atgriez izsauktas figuras
nosaukumu un perimetru

class Trijsturis(Figura): #klase Trijsturis atvasinata no klases Figura

    def __init__(self, vars, mala1, mala2, mala3): #inicialize klases
objektu
        self.vars = vars # objekta vars
        self.mala1 = mala1 # trijstura mala
        self.mala2 = mala2 # trijstura mala
        self.mala3 = mala3 # trijstura mala

    def perimetr(self): #metode aprekinā trijstura perimetru
        if self.mala1 + self.mala2 <= self.mala3 or self.mala1 + self.mala3
<= self.mala2 or self.mala2 + self.mala3 <= self.mala1:
            return ", ja divu malu summa > par trešo malu"
        else:
            return (self.mala1 + self.mala2 + self.mala3)

class Taisnsturis(Figura): #klase Taisnsturis atvasinata no klases
Figura

    def __init__(self, vars, mala1, mala2): #inicialize klases objektu
        self.vars = vars #objekta vars
        self.mala1 = mala1 # taisnstura mala
        self.mala2 = mala2 # taisnstura mala

    def perimetr(self): # metode aprekinā taisnstura perimetru
        self.rez = 2 * (self.mala1 + self.mala2) # konkrēta taisnstura
perimetr
        return self.rez # atgriez taisnstura perimetru

class Kvadrats(Figura): #klase Kvadrats atvasinata no klases Figura

    def __init__(self, vars, mala1): #inicialize klases objektu
        self.vars = vars #objekta vars
        self.mala1 = mala1 # kvadrata mala

    def perimetr(self): # metode kvadrata perimetru aprekinā
        self.rez = 4 * self.mala1 # konkrēta kvadrata perimetr
        return self.rez # atgriez kvadrata perimetru

figuras = [] # notira jauno sarakstu
# Ievada datus
trij1, trij2, trij3 = (int(input('Trijstūra malu garumi : ')))
for i in range(3)
print('\n')
taisnst1, taisnst2 = (int(input('Taisnstūra malu garumi : ')))
for i in range(2)
kvadr1 = int(input('\nKvadrāta malu garumi : '))

```

```

# izveido objektu trijsturis
trijsturis = Trijsturis('\nTrijstūra perimetrs ir ', trij1, trij2, trij3)
figuras.append(trijsturis) # sarakstam figuras pieliek objektu
trijsturis
#print(' (=====)', trijsturis)

# izveido objektu taisnsturis
taisnsturis = Taisnsturis('Taisnstūra perimetrs ir ', taisnst1, taisnst2)
figuras.append(taisnsturis) # sarakstam figuras pieliek taisnsturis

# izveido objektu kvadrats
kvadrats = Kvadrats('Kvadrāta perimetrs ir ', kvadr1)
figuras.append(kvadrats) # sarakstam figuras pieliek objektu kvadrats

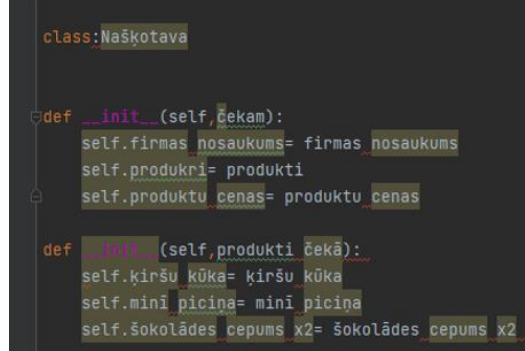

# izdrukā doto objektu nosaukumus un perimetrus
for fig in figuras: # apstrada visus saraksta elementus
    print(fig.druka()) # drukā

#raise NotImplementedError("kļūda")

```

Risinājuma detaļas	Risinājuma analīze
<b>class Figura:</b> # Define bāzes klasi	Bāzes klasei nav iekavu.
<b>def __init__(self, vards):</b> # konstruktors self.vards = vards # figuras nosaukums	Bāzes klases konstruktors.
<b>def __int__(self, vards, a, b, c):</b> # konstruktors self.vards = vards # figuras nosaukums self.a = a # figuras mala self.b = b # figuras mala self.c = c # figuras mala	Grūti saskatāma kļūda, ka nav konstruktora.
<b>def perimetrs(self):</b> # metode perimetra aprekināšanai pass # tukša metode	Ja atvasinātajā klasē ir metode, tad bāzes klasē tā vairs netiek meklēta.
<b>def druka(self):</b> # drukaas figuras nosaukumu un perimetru self.rez = self.perimetrs() # iegust izsauktaa objekta perimetru return f'{self.vards} {self.rez}' # atgriez izsauktas figuras nosaukumu un perimetru	Izdrukās šādi: Trijstūra perimetrs ir 12 Taisnstūra perimetrs ir 26 Kvadrāta perimetrs ir 32
<b>def druka(self):</b> # drukaas figuras nosaukumu un perimetru self.rez = self.perimetrs() # iegust izsauktaa objekta perimetru return f'{self.vards, self.rez}' # atgriez izsauktas figuras nosaukumu un perimetru	Izdrukās šādi: (Trijstūra perimetrs ir ', 12) (Taisnstūra perimetrs ir ', 26) (Kvadrāta perimetrs ir ', 32)

<pre>class Kvadrats(Figura): #klase Kvadrats atvasinata no klases Figura  def __init__(self, vards, a): #initalize klases objektu self.vards = vards #objekta vards self.a = a # kvadrata mala  def perimetr(self): # metode kvadrata perimetru aprekina self.rez = 4 * self.a # konkrēta kvadrata perimetr return self.rez # atgriez kvadrata perimetru</pre>	<p>Klasei</p> <pre>class Kvadrats(Figura):</pre> <p>iekavās ieraksta bāzes klasi, no kuras tā mantos atribūtus un metodes.</p> <p>Konstruktorā</p> <pre>def __init__(self, vards, a):</pre> <p>ir formālie argumenti: 'vards' un 'a', bet, ja metodē</p> <p>'def perimetr(self):' ierakstot tos pašus formālos argumentus nestrādās polimorfisms, jo visām 3 atvasinātajām klasēm ir dažāds argumentu skaits.</p>
<p><b>Vai trijstūri var izveidot no jebkuriem malu garumiem?</b></p>	<p>Izveido pārbaudi, vai no dotajiem lielumiem sanāks trijstūris!</p>

Kļūdainie risinājumi	Analīze
 <pre>class Naškotava: def __init__(self, čiekam): self.firmas_nosaukums= firmas_nosaukums self.produkti= produkti self.produktu_cenas= produktu_cenas  def __str__(self, produkti_čiekā): self.ķiršu_kūka= ķiršu_kūka self.mini_piciņa= mini_piciņa self.šokolādes_cepums_x2= šokolādes_cepums_x2</pre>	<p>Klases, parametru, atribūtu, metožu nosaukumos nedrīkst izmantot latviešu diakritiskās zīmes, punktus, komatus, tukšumus.</p>
<pre>class Cucamber(LabakaKompanija): def __init__(self): self.name = "Gurķis" self.price = 1.5</pre>	<p>Klase ir kā šablons, kurš sastāv no atribūtiem un metodēm, ar kuriem konstruē objektus, nedrīkst šādi norādīt konkrētus datus.</p>
 <pre>16 class Kvadrats(Daudzsturis): 17     def drukaVardu(self): 18         print("Kvadrats " + f" {self.figura}" + " izveidots") 19 20 class Trijsturis(Daudzsturis): 21     def drukaVardu(self): 22         print("Trijsturis " + f" {self.figura}" + " izveidots") 23 24</pre> <p>File "main.py", line 20 class Trijsturis(Daudzsturis):</p>	<p>Pitonā par atkāpju neievērošanu soda uzreiz, neraksta 'indent', bet parādās sarkans komats kā brīdinājums.</p>
<pre>class Kvadrats: def __int__(self, figura, malas): self.figura = figura self.malas = malas  def drukaFiguru(self): assert isinstance(self.figura, object) print("Es esmu " + self.figura + "\nMan ir " + self.malas)</pre>	<p>Šeit nav konstruktora, izlaists viens burts 'i'.</p>

Daži ieteikumi:

- Ja csv datne ir atvērta, tad replit.com vietnē viss strādās, bet ar PyCharm redaktoru datus atvērtā datnē nesaglabā, tātad nestrādās un būs grūti atrodama kļūda.
- Bez Python, C#, C++, Java ir pieļaujams arī webisks risinājums, kas balstīts uz aspx, php, jsp, vai pat ar html, css un javascriptu drīkst.
- Vispirms izpildi visus vienkāršos darbus, pirms koda uzlabošanas izveido kopiju, lai nav jāskatās vēsture.

## 4.daļa. Dažādas datu struktūras, programmsaskarne (API) un mašīnmācīšanās principi

### Uzdevumi

#### 4.1. uzdevums (14 punkti)

Dots API domēns: <https://www.boredapi.com/api>

Lai meklētu aktivitātes, tiek veidots šāds pieprasījums: <https://www.boredapi.com/api/activity>

Izveido API izsaukumu, kurš piedāvā gadījuma aktivitātes, lai nebūtu garlaicīgi:

- iegūsti nejaušas atbildes no servera ar iespējamām brīvā laika pavadīšanas aktivitātēm,
- apstrādā atbildes,
- izvadi iegūtās atbildes pārskatāmā veidā,
- izvadi aktivitātes nosaukumus vienu zem otra.

#### 4.2. uzdevums (28 punkti)

Dota teksta datne *test.txt*.

Lai veidotu atslēgas vārdus tekstam, nepieciešams noteikt vārdu biežumu tekstā. Izveido programmu, kas analizē tekstu no datnes, izvadot atslēgas vārdus, t.i., vārdus, kas ir atkārtotojušies visbiežāk, atslēgas vārda garums nedrīkst būt īsāks par 2 burtiem. Programma izvada līdz 8 atslēgas vārdiem.

4.1. Atvērtā koda bibliotēkas un API (programmsaskarne)		
Uzdevums par doto situācijas aprakstu	Risinājums	Risinājuma analīze
Sameklē un pievieno atvērtā koda bibliotēku pieprasījumiem, angļiski – requests.	<code>import requests</code>	Lai darbotos ar API Python valodā, nepieciešama atvērtā koda HTTP requests library, (pieprasījumu bibliotēka), kas ļauj izmantot pieprasījumu GET/POST/PUT/PATCH/DELETE
Lai kontrolētu datu izvadi konsolē, datu rediģēšanai importē moduli JSON.	<code>import json</code>	Pievieno JSON bibliotēku ar nosaukumu json. Tas palīdz darboties ar JSON objektiem.
Pieprasi datus no API servera, šos datus saglabā mainīgajā ar nosaukumu atbilde.  No tīmekļa vietnes <a href="https://www.boredapi.com/api/activity">https://www.boredapi.com/api/activity</a> iegūsti padomu, angļu valodā, JSON formātā, ko darīt, ja ir garlaicīgi.	<code>atbilde = requests.get("https://www.boredapi.com/api/activity")</code>	Mainīgajam 'atbilde' tiek piešķirts pieprasījuma datu saturs.  <a href="https://www.boredapi.com/api/activity">https://www.boredapi.com/api/activity</a> - šajā vietnē tiks ģenerētas dažādas aktivitātes
Izdrukā padomu ko darīt, ja ir garlaicīgi, angļu valodā, JSON formātā.	<code>print(atbilde.status_code)</code> <hr/> <code>200</code>	JSON formāts – populārs datu pārsūtīšanai  Iespējamās atbildes JSON formātā: <b>200</b> – Sekmīga pārsūtīšana. Dati tiek atsūtīti.

		<p><b>204</b> – Nav satura, sekmīgi izpildīja, bet neko neatgrieza.</p> <p><b>301</b> – API galapunkts pārvietots uz citu adresi.</p> <p><b>400</b> – Nepareizi izveidots pieprasījums.</p> <p><b>401</b> – Nenotiek autorizācija, bet vajag, lai pārsūtītu datus.</p> <p><b>403</b> – Neatļauta piekļuve datiem.</p> <p><b>404</b> – Resursi nav atrasti uz servera.</p> <p><b>500</b> – Servera kļūda.</p>
<p>Izvadi konsoļē padomu ko darīt, ja ir garlaicīgi, angļu valodā, JSON formātā</p> <p><a href="https://www.boredapi.com/api/activity">https://www.boredapi.com/api/activity</a></p> <p>Izvadi iegūtos datus uz konsoles.</p>	<pre>print(atbilde.json())</pre>	<p>Ja tika pareizi pieprasīti dati, kas tika saglabāti mainīgajā ar nosaukumu atbilde, šie dati tiks izdrukāti neformatētā veidā.</p> <p>Izvada iegūtos datus JSON formātā.</p> <pre>{'activity': 'Learn how to french braid hair', 'type': 'education', 'participants': 1, 'price': 0, 'link': '', 'key': '8926492', 'accessibility': 0.1}</pre>
<p>Izvadi JSON datus pārskatāmā veidā.</p>	<pre>def jsonprint(obj):     izvade = json.dumps(obj,                         sort_keys=True,                         indent=5)     print(izvade)</pre>	<p>Veido funkciju jsonprint, kas saturēs un sakārtos objektu.</p> <p><b>json.dumps()</b> – pārvērtīs Python objektus par json simbolu virkni.</p> <p>Parametri:</p> <p><b>allow_nan</b> – (noklusējuma vērtība True), ja patiess, var pielietot NaN (nav skaitlis), infinity (bezgalību) un -Infinity (- bezgalību)</p> <p><b>check_circular</b> – (noklusējuma vērtība True), ja nepatiess, izlaidīs pārbaudi vai norādes cirkulāras</p> <p><b>default</b> – atgriež objekta JSON versiju vai TypeError kļūdu</p> <p><b>ensure_ascii</b> – (noklusējuma vērtība True), ja patiess, neparāda simbolus, kas nav no ascii tabulas</p> <p><b>indent</b> – atkāpe no kreisās malas kā skaitlis</p> <p><b>obj</b> – objektu pārvērš par JSON formāta simbolu virkni</p> <p><b>separators</b> – noklusējums (', ', ': '), ja lieto citus, pirmajās iekavās vērtību pāru atdalītājsimbols, tad pāra iekšpuses atdalītājsimbols (item_separator, key_separator)</p> <p><b>skipkeys</b> – ja patiess, tad atslēgas, kas nav str, int, float, bool vai None, tiks izlaistas, nebūs kļūdas</p>

		<b>sort_keys</b> – (noklusējuma vērtība False) vai sakārtot iegūtos datus pēc atslēgas nosaukuma, true – sakārtot, false – nesakārtot
Iegūsti un izvadi atbildi uz konsoles.	<pre>jsonprint(atbilde.json())</pre>	Izvada formatēto atbildi JSON formātā, lai ērtāk lasīt.  atbilde.json() – atbildes dati, kas pārvērsti json formā.  { "accessibility": 0.1, "activity": "Learn how to french braid hair", "key": "8926492", "link": "", "participants": 1, "price": 0, "type": "education" }
Izvadi vienu ieteikumu	<pre>ieteikums = atbilde.json()['activity'] jsonprint(ieteikums)</pre>	Ieteikums būs atslēgas ar nosaukumu 'activity' vērtība, jo ar jsonprint() funkciju izdrukās konsolē.  "Learn how to french braid hair"
Izveido ieteikumu masīvu ar nosaukumu ieteikumi.	<pre>ieteikumi = []</pre>	Tukšs masīvs.
Dokumentācija	<a href="https://www.boredap i.com/documentation">https://www.boredap i.com/documentation</a>	Izdrukā vismaz 3 aktivitātes

#### 4.2. Datu struktūras un algoritmi

Populārie algoritmi: meklēšanas, kārtošanas, koku apceļošanas, biežuma noteikšana

**Dota datne 'test.txt'** (*Trenējies ar savu datni*)

Pie mājas staigāja abi mājdzīvnieki, suns Reksis un kaķis Muris. Kaimiņu mājā dzīvoja kaķis Ņuris. Kopā visā ciematā dzīvoja desmit suņu un pieci kaķi.

Uzdevums par doto situācijas aprakstu	Risinājums	Risinājuma analīze
Ievieto bibliotēku	<pre>import re</pre>	Pievieno bibliotēku, lai rakstītu regulārās izteiksmes (regular expressions).
Ievieto bibliotēku	<pre>import string</pre>	Pievieno simbolu virknes bibliotēku.
Izveido vārdnīcu	<pre>biezhums = { }</pre>	Izveido tukšu vārdnīcu, kur glabās biežumus.
Atver teksta datni ar nosaukumu test.txt	<pre>teksts = open('test.txt', 'r', encoding='utf-8')</pre>	Atver teksta datni ar vārdiem, kuru biežumu skaitīs, encoding = 'utf-8', lai skatītos arī latviešu valodas rakstzīmes.



Pārveido tekstu uz mazajiem burtiem, lai salīdzinātu vārdus.	teksta_simb_virkne = teksts.read().lower()	Nolasa tekstu, visu pārveido par mazajiem burtiem.
Izmanto regulāro izteiksmi un funkciju findall(), lai atrastu visus vienādos vārdus	meklet_vardu = re.findall(r"(\w+)", teksta_simb_virkne, re.UNICODE,)	Meklē vārdus garumā no 3 līdz 15 simboliem.  findall(pattern, string, flags=0) – pattern – neatkārtojošies paraugi simbolu virknē, r"(\w+)", r" – salīdzina simbolus, \w++ salīdzina simbolus [a-zA-Z0-9_] tik reizes cik nepieciešams ( <a href="https://regex101.com">https://regex101.com</a> – regulāro izteiksmju veidošanai) string - simbolu virkne, skatās no kreisās uz labo pusi, atgriež atrasto tādā pašā secībā, kā atrasti paraugi, re.UNICODE - unikods
Meklē vārdu biežumu.	for vards in meklet_vardu: skaits = biezhums.get(vards, 0) biezhums[vards] = skaits + 1	Ja atrod vienādus vārdus, tad šim vārdam biežumu palielina par 1.
Veido biežumu sarakstu.	biezhumu_saraksts = biezhums.keys()	Biežumu saraksts ir biežumu vārdnīcas atslēgu skaits.
Izvada biežumu sarakstu ar vārdiem un to biežumu.	for vardi in biezhumu_saraksts: print(vardi, biezhums[vardi])	Drukā biežumu vārdnīcās veidā.  pie 1 abi 1 suns 1 reksis 1 un 2 muris 1 desmit 1 pieci 1

## 4. Ieteikumi, kā gatavoties eksāmenam

Vidusskolas programmēšanas temati pēc satura ir atšķirīgi un daudzveidīgi, tāpēc nepieciešama nopietna atkārtošana.

- Izpildi skolēniem izveidotos interaktīvos uzdevumus Digitālo Mācību Līdzeklī (DML) skolo.lv vietnē, kur pieejama teorijas daļa, gan digitālie, interaktīvie uzdevumi, gan pašvērtējums temata beigās.
- Atkārti “Programmēšana I” tematu “Kā uzrakstīt kodu pēc dotas specifikācijas?”
- Patstāvīgi izrēķini dotos uzdevumus valsts pārbaudes darba paraugā un tikai pēc tam pārbaudi savus risinājumus, kuri ir doti paraugā.
- Izpēti dotā VPD piekļuves parauga dokumentācijas posmus 2.pielikumā.
- Doto datņu vietā strādā ar savām datnēm.

## 5. Palīg līdzekļi, kurus atļauts izmantot eksāmena laikā

Skolēnam tiek atļauts izmantot izvēlētās programmēšanas valodas dokumentāciju, kura ir iekļauta datorā instalētajā izstrādes vidē, pašā programmēšanas valodā vai lejupielādēta no programmēšanas valodas izstrādātāju mājas lapas, tīmekļa resurss “w3schools”.

### Ieteicamās vietnes

1. Vidusskola. 1. Kā uzrakstīt kodu pēc dotas specifikācijas? Skatīts 11.11.2022. Pieejams: <https://mape.skola2030.lv/materials/on6GnMJQEoGxTyr9Rq3jXS>
2. Padziļinātais kurss Programmēšana II augstākajā mācību satura apguves līmenī. Skatīts 11.11.2022. Pieejams: [https://www.youtube.com/watch?v=nZLrNDAqLF8&list=PLLLs\\_n2ybYppgO-tV7n\\_r8OGnOboj17So&index=17](https://www.youtube.com/watch?v=nZLrNDAqLF8&list=PLLLs_n2ybYppgO-tV7n_r8OGnOboj17So&index=17)
3. S. Kodors. Ievads prasību inženierijā. Skatīts 11.11.2022. Pieejams: <http://books.rta.lv/index.php/RTA/catalog/download/9/8/29-3?inline=1>
4. SQL UNIQUEIDENTIFIER Data Typ. Skatīts 11.11.2022. Pieejams: <https://www.dofactory.com/sql/uniqueidentifier>
5. MySQL Data Types. Skatīts 11.11.2022. Pieejams: <https://www.mysqltutorial.org/mysql-data-types.aspx>
6. Padziļinātais kurss Programmēšana II augstākajā mācību satura apguves līmenī. Valsts pārbaudes darba paraugs. Skatīts 11.11.2022. Pieejams: <https://mape.skola2030.lv/resources/14206>
7. Padziļinātais kurss Programmēšana II augstākajā mācību satura apguves līmenī. Valsts pārbaudes darba programma. Skatīts 11.11.2022. Pieejams: <https://mape.skola2030.lv/resources/13865>
8. Programmēšana II Padziļinātā kursa programmas paraugs vispārējai vidējai izglītībai. Skatīts 11.11.2022. Pieejams: <https://mape.skola2030.lv/resources/5290>

## **Pielikums**

### 1. pielikums

**Šogad VPD izmantos SR, kuri norādīti eksāmena paraugā:**

#### **1. daļas SR - Datortīkls un droša datubāze (DB)**

2.3.2. Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam.

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

#### **2. daļas SR - Programmatūras Dzīves Cikls**

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.

2.4.4. Sastāda vienkāršotu programmatūras prasību specifikāciju atbilstoši konkrētajam uzdevumam, izvērtējot mērķauditorijas specifiku un vajadzības.

#### **3. daļas SR - Objektorientētā programmēšana (OOP) un ārējās bibliotēka**

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.

2.4.15. Skaidro objektorientētās programmēšanas pamatprincipus, veido programmas vienā no objektorientētajām programmēšanas valodām.

#### **4. daļas SR - Dažādas datu struktūras, programmsaskarne (API) un mašīnmācīšanās principi**

2.4.11. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai.

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

## 2. pielikums

### VPD piekļuves darba dokumentācija

*Temata apguves mērķis: Prast noformulēt problēmu un tai izveidot specifiskāciju pēc kuras spēt izstrādāt programmatūru, dokumentējot to.*

#### 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.)*

**Uzdevums:** Izveidot situācijas aprakstu pēc parauga savā nozarē

#### **DOTS PROBLĒMSITUĀCIJAS APRAKSTS:**

Kosmetoloģijas kabineta īpašniecei Dainai regulāri jāatjauno krēmu, attīrošo pienu, vatītes plāksnišu krājumi, kuriem dažādi uzglabāšanas termiņi, ja krēms tiek atvērts, tad uzglabāšanas termiņš mainās. Kabineta īpašniecei ir patstāvīgi apmeklētāji, kuri vēlas ierasties vienā un tajā pašā laikā, bet ir apmeklētāji, kuriem plāni bieži mainās.

Dainai būtu noderīgs digitālais rīks ar dažādām datu atlasēs iespējām un vizualizācijām, kas konkrētos datus attēlo viņai saprotamā un ērtā formātā, kurš laicīgi brīdinātu par vajadzīgo izejvielu iegādi, par katra klienta ierašanās laikiem, lai nepierakstītu 2 klientus vienā laikā.

Dažu nozaru piemēri:

Nr.	Nozare	Nr.	Nozare
1	Operācijas ar nekustamo īpašumu	9	Obligātā sociālā apdrošināšana
2	Zinātniskie un tehniskie pakalpojumi	10	Finanšu un apdrošināšanas darbība
3	Apkalpojošo dienestu darbība	11	Mājsaimniecību kā darba devēju darbība
4	Transports un uzglabāšana	12	Sociālā aprūpe
5	IKT pakalpojumi	13	Ārstniecības pakalpojumi
6	Futbola privātskola	14	Izmitināšana pakalpojumi
7	Automobiļu remonts	15	Ēdināšanas pakalpojumi
8	IKT pakalpojumi	16	Būvniecība vai Dzīvokļu remonts.

Izmantoju populārākās lietotāju izpētes metodes:

- interviju,
- aptauju,
- novērojumu,
- eksperimentus,

lai noskaidrotu lietotāja paradumus, analizēju iegūtos datus un secinu, kādas ir turpmākas darbības risinājuma izstrādē.

#### 2. Programmatūras prasību specifiskācija - risinājuma mērķauditorijas izvēle un tās raksturojums, programmvadāmā risinājuma un tā funkciju apraksts, programmatūras izstrādes projekts

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

PYTHON ir augsta līmeņa objektorientētā valoda, automātiski dzēš norādes uz izlieto kodu, atbrīvojot atmiņu.

Bezmaksas un atvērta pirmkoda programmatūra, kurā var brīvi izmantot bibliotēkas, kuras nemītīgi pieaug. Nodrošina ērtu datu bāzes apstrādi, piemēram, ar MySQL uz WEB serveri var strādāt tikai importējot bibliotēku MySQLdb.

Pārnesamība – strādā daudzās platformās. Mērogojama, piemēram, uzlabot ātrdarbību var iegulstot C/C++ kodus. Izplatīt pat komerciālai lietošanai.

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

Informāciju aprakstīt, nevis pateikt:

- Konkrēti;
- Strukturēti;
- Jēgpilni;
- Atbilstoši tēmai.

Uzdevuma **nosaukuma** formulēšana :

- Nosaukums – īss tikai jēgpilns, mazāk darbības vārdu, apzīmētāju, aprakstīta būtība, uzsvērts svarīgākais, atslēgas vārdi, no nosaukuma var secināt tā uzdevumu.

Nepareizi	Pareizi
Burvīgā, jaukā kosmetoloģijas kabineta krēmu, pienu izmantošanas termiņa noteikšana un klientu pierakstīšana uz procedūrām.	Kosmetoloģijas salona apmeklētāju un izejvielu uzskaitē.

Uzdevuma mērķa formulēšana:

- Iekļauj jēgpilnu informāciju, kāpēc šo darbu svarīgi veikt.
- Ko iegūst lietotājs vai mūsu kompānija?
- Norāda, uz kurieni mēs ejam, nevis kā ejam.

Nepareizi	Pareizi
Burvīgā, jaukā kosmetoloģijas kabineta krēmu, pienu izmantošanas termiņa noteikšana un klientu pierakstīšana uz procedūrām.	Apmeklētāju grafika izveide un izejvielu automatizēta derīguma termiņa uzskaitē.

Uzdevuma formulēšana:

- Raksti īsi, atbilstoši nosaukumam, uz detaļām uzliec norādi.
- Raksti strukturēti, sadali informāciju loģiskajās daļās, izceļot nodaļu nosaukumus.
- Secīgi apraksti visu uzdevumu no sākuma līdz beigām, nepieciešamības gadījumā paskaidro detalizētāk.
- Apraksti visus nepieciešamos ievaddatus.
- Informāciju var strukturēt tabulās vai čekboksos, kur lietotājam ērti atzīmēt paveikto. Var strukturēt ar atkāpēm.

Ievadīt datus par izejvielām un klientiem.	Ievadīt klientu, izejvielu datus, apmeklēšanas datumu un laiku. Detalizētāk skatīt specifikāciju.
--	---

	<p>Datus uzglabāt datu bāzē.</p> <p>Ja kāda izejviela tiek atvērta, tad tai nomainīt derīguma termiņu.</p> <ul style="list-style-type: none"> <li>• Izvēloties vielas, izvadīt visu vielu nosaukumus, skaitu un derīgumu termiņus.</li> <li>• Izvēloties klientu, izvadīt klienta pierakstīšanās datumu, ja tāda nav, iespējamās pierakstīšanās datumus, atļaujot mainīt esošo datumu.</li> </ul>
--	---

Tehniskās detaļas:

- API, dizaini;
- Datu bāzes, datnes;
- Platforma, kurā realizēs;
- Saistītie uzdevumi;
- Nepieciešamās norādes.

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

**Mērķauditorija** – kosmetoloģijas kabineta īpašniece, jo viņa lietos šo aplikāciju

Specifikācijā jābūt:

**1. Kādi ievaddati tiks doti?**

Klientu, izejvielu dati, apmeklēšanas datumu un laiku.

**2. Ievaddatu tips, mērvienības, iespējamās vērtības?**

**Persona**{

vards - TEXT,  
uzvards - TEXT,  
mob\_talr - Number  
klienta\_kods – skaitlisks/veidots-ģenerēts  
datums – TEXT/Date

}

**Darbinieks(Persona)**{

pk - TEXT,

}

**Klients(Persona)**{

}

**3. Kādā veidā tiks iesniegti ievaddati (izsaucot funkciju, no standarta ievades, no datnes, no DB)**

Saskarnē ievadīs jaunās personas, pakalpojumu, izejvielu datus, esošos datus piedāvās izvēlēties no saraksta.

**4. Metodes nosaukums un deklarācija?**

izejvielu\_skaits(nosaukums, derīgums) - skaitliska

**5. Kā iegūt (aprēķināt, kombinēt, formatēt) rezultātu?**

Atlasīt darbadienu skaitu norādītajā gadā un mēnesī.

Saskaitīt nostrādāto laiku pareizinātu ar kvalitāti (1-darbs ieskaitīts/ 0-darbs neieskaitīts).

Pareizināt ar algu stundā.

**6. Kā jāsniedz atbilde (atgriežot vērtību no funkcijas, izvadot atbildi uz ekrāna, datnē, DB)?**  
Datu bāzē saglabāt katra darbinieka nostrādātās stundas attiecīgajā darba laikā un kvalitāti, **uz ekrāna** izvadīt, kurš klients uz kuru datumu pierakstīts, kuram darbiniekam kurā dienā jāstrādā, cik kuram darbiniekam jāmaksā.

**7. Izvaddatu tips, mērvienības, iespējamās vērtības?**  
Izvadīt saskarnē, datu bāzē teksta datus, algu eiro.

**8. Doti piemēri ar konkrētiem ievaddatiem un atbilstošu sagaidāmo rezultātu**

Klientam

ievaddati: 372

izvaddati: Maija Liepa 2022 gada 17.novembrī 11:00

ievvaddati: **telefona numurs**

izvaddati: **vārds, uzvārds, datums, apmeklēšanas laiks**

var ievadīt jaunu klientu



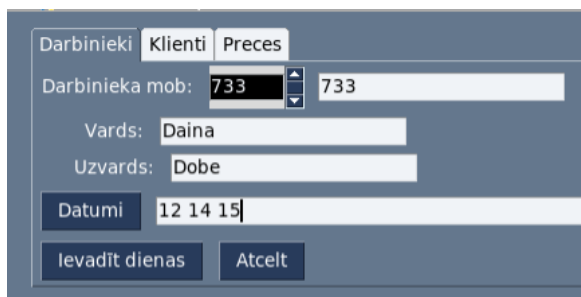
Darbiniekam

ievaddati: 733 ,gads un mēnesis

izvaddati: Daina Dobe 12 13 14

ievvaddati: **telefona numurs**

izvaddati: **vārds, uzvārds, datums mēneša datumā**



<https://mape.skola2030.lv/materials/on6GnMJQEOGxTyr9Rq3jXS>

*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.)*

### 3. Programmatūras izstrādes plāns

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

Ūdenskrituma modelis

Spirāles modelis

Spējās izstrādes modelis

SR: Veic vienkāršotu programmatūras izstrādes plānošanu (bez darbietilpības novērtējuma) un darba pienākumu sadali grupā. (T.A. 2.4.5.)

#### 4. Programmatūras kods, kas veidots ievērojot labās prakses principus

Labās prakses principi ir:

- katru atsevišķu komandu raksta jaunā rindā;
- koda loģiskās daļas (piemēram, zarošanās, cikls, masīvs) savstarpēji atdala ar atstarpī;
- lieto atkāpes, lai vizualizētu komandu vai struktūru iekļaušanu citās struktūrās;
- izvairās no garām koda rindām, lieto pārnesi jaunā rindā, atvieglojot koda lasīšanu;
- mainīgo, funkciju u.c. nosaukumus veido īsus, bet jēgpilnus, atvieglojot koda uztveri (piemēram, perimetru apzīmējot nevis ar "a", bet "perim" vai "perimetrs");
- ar komentāriem skaidro programmas koda loģiskās daļas (piemēram, zarošanās, cikls, masīvs), pamatojot to nepieciešamību, lomu programmā (piemēram "Datu izvade").

Dotajam kodam pieliec darbu ar datu bāzi un komentārus.

```
1 import PySimpleGUI as sg # 28.nov 10:11
2 import csv
3
4
5 # ===== Pakalpojums
6 class Pakalpojums:
7     def __init__(self, uzvards, cena, datums, laiks):
8         self.uzvards = uzvards
9         self.cena = cena
10        self.datums = datums
11        self.laiks = laiks
12
13
14 # ===== Izejvielas
15 class Izejvielas:
16     def __init__(self, uzvards, derigums, derigums_atvertam, laiks):
17         self.uzvards = uzvards
18         self.derigums = derigums
19         self.derigums_atvertam = derigums_atvertam
20         self.laiks = laiks
21
22
23 # ===== Persona
24 class Persona:
25     def __init__(self, kods, numurs, vards, uzvards, datums, galvene, key_li, datne):
26         self.datne = datne
27         self.key_li = key_li
28         self.vards = vards
29         self.uzvards = uzvards
30         self.mob_nr = numurs
31         self.datums = datums
32         self.galvene = galvene
33
34     def saglaba(self, dati): # ===== saglaba(self,dati)
35         self.dati = dati # delimiter="",
36         with open(self.datne, 'w', newline='') as csv_datne:
37             ierakstit = csv.writer(csv_datne, delimiter=',')
38             for rinda in self.dati:
39                 ierakstit.writerow(rinda)
40
41     def nolasa(self): # ===== nolasa(self)
42         try:
43             with open(self.datne, 'r', newline='') as csv_datne:
44                 dati = csv.reader(csv_datne, delimiter=',', quoting=csv.QUOTE_NONE)
45                 dati = list(dati)
46             return dati
47         except:
48             print('Lūdzu datni,\''', self.datne, '\'' ievietojiet')
49             exit()
```



```

50 def kodu(self):
51     dat = self.nolasa()
52     kods = []
53     for rinda in dat:
54         kods.append(rinda[0])
55     return (kods)
56
57 def layout_pers(self, kods, key_li): # ===== izkaartojums #format = '%Y-%m-%d'
58     izkarto_pers = [[sg.Text('Personas mob:'), sg.ListBox(kods, enable_events=True, key='-NRP'),
59                     sg.Input(key='P-NRP-', enable_events=True, size=(20, 1))],
60                    [sg.Text('  Vards:'), sg.Input(key='P-VARDS-', enable_events=True, size=(20, 1))],
61                    [sg.Text('  Uzvars:'), sg.Input(key='P-UZVARDS-', enable_events=True, size=(20, 1))],
62                    [sg.Button('Ievadīt klientu'), sg.Exit('Atcelt')]]
63     return izkarto_pers
64
65 class Darbinieks(Persona):
66     def layout_pers(self, kods, key_li):
67         izkarto_pers = [[sg.Text('Darbinieka mob:'), sg.ListBox(kods, enable_events=True, key='-NRD'),
68                         sg.Input(key='D-NR-', enable_events=True, size=(20, 1))],
69                        [sg.Text('  Vards:'), sg.Input(key='D-VARDS-', enable_events=True, size=(20, 1))],
70                        [sg.Text('  Uzvars:'), sg.Input(key='D-UZVARDS-', enable_events=True, size=(20, 1))],
71                        [sg.CalendarButton('Datumi', target='D-DATUMS-', format='%Y-%m-%d'),
72                         sg.Input(key='D-DATUMS-', enable_events=True, size=(70, 1)), ],
73                        [sg.Button('Ievadīt dienas'), sg.Exit('Atcelt')]]
74         return list(izkarto_pers)
75
76
77 class Klienti(Persona):
78     def layout_pers(self, kods, key_li):
79         izkarto_pers = [[sg.Text('Klienta mob:'), sg.ListBox(kods, enable_events=True, key='-NRK'),
80                         sg.Input(key='K-NR-', enable_events=True, size=(20, 1))],
81                        [sg.Text('  Vards:'), sg.Input(key='K-VARDS-', enable_events=True, size=(20, 1))],
82                        [sg.Text('  Uzvars:'), sg.Input(key='K-UZVARDS-', enable_events=True, size=(20, 1))],
83                        [sg.CalendarButton('Apmeklēšanas datums', target='K-DATUMS-', format='%Y-%m-%d'),
84                         sg.Input(key='K-DATUMS-', enable_events=True, size=(20, 1))],
85                        [sg.Button('Ievadīt klientu'), sg.Exit('Atcelt')]]
86         return list(izkarto_pers)
87
88
89 # ===== main
90 key_ievad = ('-NR-', '-VARDS-', '-UZVARDS-', '-DATUMS-')
91 key_li = ('-NRD', '-NRK', '-NRP')
92 galvene = ['numurs', 'vards', 'uzvars', 'datums']
93 # ===== dati saskarnei
94 darbinieks = Darbinieks(" ", " ", " ", galvene, key_li, 'darbinieki.csv')
95 darbinieki = darbinieks.nolasa()
96 darbinieku_kod = darbinieks.kodu()
97
98 klienti = Klienti(" ", " ", " ", galvene, key_li, 'klienti.csv')
99 klienti = klienti.nolasa()
100 klientu_kod = klienti.kodu()
101
102 preces = [['krēms', 'pieniņš', 'špātulas']]
103 precu_kodi = [['P001', 'P002', 'P003']]
104 # ===== izkaartojums
105 layout_preces = [[sg.Text('Preces:'), sg.ListBox(precu_kodi, enable_events=True, key='-NRP'),
106                 sg.Input(key='P-NR-', size=(20, 1))],
107                [sg.Input(key='P-DATUMS-', size=(20, 1)), sg.CalendarButton('Termiņš', target='P-DATUMS-',
108                                default_date_m_d_y=(1, None, 2022)),
109                [sg.Button('Ievadīt preci'), sg.Exit('Atcelt')]]
110
111 # ===== tabulaacijas
112
113 tab_group = sg.TabGroup([
114     sg.Tab("Darbinieki", darbinieks.layout_pers(darbinieku_kod, key_li), key='-TABD-'),
115     sg.Tab("Klienti", klienti.layout_pers(klientu_kod, key_li), key='-TABK-'),
116     sg.Tab("Preces", layout_preces, key='-TABP-')
117 ])
118 col = [[tab_group]]
119 layout = [[sg.Column(col)]]
120 window = sg.Window('Skaistumkopšanas kabinets', layout)
121
122 # ===== Notikumi:
123 event = "
124 key_ievadi= ('D-NR-', 'D-VARDS-', 'D-UZVARDS-', 'D-DATUMS-', 'K-NR-', 'K-VARDS-', 'K-UZVARDS-', 'K-DATUMS-', 'P-NR-',
125 'P-DATUMS-')

```

```

126 vardnic_dati = {
127     '-NRD': darbinieki,
128     '-NRK': klienti,
129     '-NRP': preces
130 }
131 vardnic_kodi = {
132     '-NRD': darbinieku_kod,
133     '-NRK': klientu_kod,
134     '-NRP': precu_kodi
135 }
136 vardnic_key_ievad = {
137     '-NRD': ('D-NR-', 'D-VARDS-', 'D-UZVARDS-', 'D-DATUMS-'),
138     '-NRK': ('K-NR-', 'K-VARDS-', 'K-UZVARDS-', 'K-DATUMS-'),
139     '-NRP': ('P-NR-', 'P-DATUMS-')
140 }
141 while event != sg.WIN_CLOSED and event != 'Atcelt':
142     event, values = window.read()
143
144     if event in key_li: # meklee mob nr vai kodu
145         dati = vardnic_dati[event]
146         kodi = vardnic_kodi[event]
147         key_ievad = vardnic_key_ievad[event]
148         jauns = []
149         text_event = event # viens no ('-NRD', '-NRK', '-NRP')
150         nr = kodi.index(values[event][0]) # atrod rindu ar izveleto numuru
151         i=0
152         for text_event in key_ievad: # klientam ('K-NR-', 'K-VARDS-', 'K-UZVARDS-', 'K-DATUMS-')
153             values[text_event] = dati[nr][i] #nr rindaa kolonas i
154             jauns.append(values[text_event])
155             window[text_event].update(values[text_event]) # atjauno, lai redzetu teksta lauka
156             i = i + 1
157
158         if event == '-NRD': # darbin mob.nr
159             darbinieki=dati # darbinieku dati
160         elif event == '-NRK': # klientu mob.nr
161             klienti=dati # klientu dati
162         else:
163             preces=dati
164     elif event in key_ievadi: # kad ievadiits teksta laukaa
165         dati = vardnic_dati['-NR'+event[0]] #'-NRD', '-NRK', '-NRP')
166         kodi = vardnic_kodi['-NR'+event[0]]
167         key_ievad = vardnic_key_ievad['-NR'+event[0]]
168         i = 0 #key_ievad=('D-NR-', 'D-VARDS-', 'D-UZVARDS-', 'D-DATUMS-')
169         jauns = []
170         for text_event in key_ievad:
171             jauns.append(values[text_event])
172             window[text_event].update(values[text_event]) # atjauno, lai redzetu teksta lauka
173             i = i + 1
174     elif event == "Ievadīt klientu": # nospiesta poga "Ievadīt"
175         if len(jauns) >= 4 and " not in jauns:
176             sg.popup(jauns, background_color='#007733')
177             flag = False
178             for r in dati:
179                 try:
180                     nr = r.index(jauns[0]) ### meklee numuru sarakstaa
181                     flag = True
182                 except:
183                     pass
184             if flag:
185                 for k in range(4):
186                     dati[nr + 1][k] = jauns[k]
187             else:
188                 dati.append(jauns)
189                 klienti.saglabi(dati)
190         else:
191             sg.popup('Kļūda', 'Aizpildīt visus laukus', background_color='#FF0000')
192     elif event == "Ievadīt dienas": # nospiesta poga "Ievadīt"
193         if len(jauns) >= 4 and " not in jauns:
194             sg.popup(jauns, background_color='#007733')
195             flag = False
196             for r in dati:
197                 try:
198                     nr = r.index(jauns[0]) ### meklee numuru sarakstaa
199                     flag = True
200                 except:
201                     pass

```

```

202     if flag:
203         for k in range(4):
204             dati[nr + 1][k] = jauns[k]
205     else:
206         dati.append(jauns)
207     darbinieks.saglaba(dati)
208     else:
209         sg.popup('Kļūda', 'Aizpildīt visus laukus', background_color='#FF0000')
window.Close()

```

## 5. Atklūdošanas 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.)*

### 5.1. Vienībtestēšana

Ir sastopams testēšanas veids, kurā kodu sadala vienībās un testē katru vienību atsevišķi, lai pārbaudītu un apstiprinātu, ka vienība strādā, kā paredzēts. Šo testēšanu parasti veic paši izstrādātāji, veidojot kodu. Vienības var būt: metode, funkcija, objekts.

Vienībtestēšana ir baltās kastes testēšanas paveids. Baltās kastes programmēšanā testētājs zina iekšējo programmas kodu un struktūru, pārbauda lielākoties datu plūsmu no ievaddatiem līdz izvaddatiem, vai pareizi norit nelielo vienību darbība, pārbaudot iespējamās ievades un izejas datus.

Vienībtesti ir automātiskie un pašrocīgie, kuros pārbauda programmas darbības sazarojumus, ciklus, funkcijas, kopējo datu pārskatu.

Labā prakse vienībtestēšanā:

- testēšanas gadījumiem jābūt neatkarīgiem;
- testēšanas gadījumi nav pārāk sarežģīti un lieli, citādi jāpatērē pārāk daudz resursu pārbaudei;
- veido skaidrus un testu izskaidrojošus nosaukumus testam;
- pirms veidot koda nākamo daļu izlabo testā notvertās kļūdas;
- testus veido, jau kodējot.

Testēšanas (piemērs) telefona numuram :

Testa numurs	Ieejas dati	Izejas dati
1	0234567	Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo mazāk par 8 cipariem!
2	-013	Tālruņa numuram jābūt pozitīvam!
3	012345678	Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo vairāk par 8 cipariem!
4	""	Izvada kļūdas paziņojumu. Nekas nav ievadīts, ievadi tālruņa numuru!
5	21234567	Ievadīts derīgs tālruņa numurs!

Testēšanas rezultāti:

Testa	Ieejas dati	Izejas dati	Iegūtais	Piezīmes

numurs			rezultāts	
1	2234567	Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo mazāk par 8 cipariem!	atbilst	Kļūdas paziņojums. Programma turpina darbu, līdz ievada 8 ciparu tālruņa numuru.
2	-123	Tālrūņa numuram jābūt pozitīvam!	atbilst	Kļūdas paziņojums. Programma turpina darbu, līdz ievada pozitīvu skaitli, kas atbilst tālruņa numuram.
3	123456789	Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo vairāk par 8 cipariem!	atbilst	Kļūdas paziņojums. Programma turpina darbu, līdz ievada 8 ciparu tālruņa numuru.
4	“”	Izvada kļūdas paziņojumu. Nekas nav ievadīts, ievadi tālruņa numuru!	neatbilst	Programma neizvada kļūdas paziņojumu
5	21234567	Ievadīts derīgs tālruņa numurs!	atbilst	Programma izvada atbildi un beidz darbu.

## 5.2. Akcepttestēšana

Akcepttestēšana notiek programmas izstrādes beigās, kad pārbauda visu sistēmu, vai tā atbilst specifikācijai, vai tā ir pietiekama, vai darbojas korekti un ir gatava lietošanai. Tas ir melnās kastes testēšanas paveids, kad nevar izpētīt visu koda iekšpusē notiekošo, bet ievada datus un redz izvades rezultātu. Melnās kastes programmēšanā testētājs nezina detaļas par to, kas notiek vienības iekšpusē, vai neprot programmēt, bet zina ievaddatus un izvaddatus.

Pirms testēšanas jāizveido tabula ar visām programmas darbībām, kuras programmatūrai jāpiedāvā atbilstoši specifikācijai. Lietotāji pārbauda produktu un novērtē, vai tas strādā, kā aprakstīts.

Programmatūras iespējamās darbības, funkcijas	Jā	Daļēji	Nē	Piezīmes
Ievada klientu apmeklēšanas datumu un laiku.		x		
Mainīt klienta apmeklēšanas datumu			x	
Klientu datus uzglabāt datu bāzē.	x			
Maina klienta datus datu bāzē	x			

## 6. Lietotāja ceļvedis

*SR: 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.)*

Jāapraksta:

1. Ievads. Kam paredzēta programmatūra?
2. Programmatūras apraksts:
  - 2.1. Ko sistēma atbalsta, kādus pārlūkus?
  - 2.2. Kā var pieslēgties sistēmai?
  - 2.3. Kas notiks, ja pieslēdzas un kļūdās paroles vai lietotājvārda ievadē?
  - 2.4. Ko redz lietotājs savā lietotāja klasē?
3. Ieviešanas plānu, uzturēšanas plānu, lietotāju mērķauditorijas specifiku

Piemērs:

Šīs sistēmas lietotāja ceļvedis ir paredzēts visām lietotāju klasēm – bet apskatītajā piemērā ir tikai viena klase, kosmētikas kabineta īpašniece.

Sistēma atbalsta Mozilla Firefox, Opera.

Lai pieslēgtos sistēmai, ir jāatver tīmekļa pārlūks un jāievada vietnes adrese. Tad jāievada lietotājvārds, parole un jānospiež uz pogu *pieslēgties*.

Ja pieslēdzas un kļūdās, lietotājvārda vai paroles ievadē parāda kļūdas ziņojumu “Nepareiza parole/lietotājvārds!”.

Kosmētikas kabineta īpašnieces funkcijas:

var aplūkot, ievadīt un mainīt klienta datus,

var aplūkot, ievadīt un mainīt pakalpojuma datus,

var aplūkot, ievadīt un mainīt izejvielu datus,

var aplūkot, ievadīt un mainīt darījuma datus.

Klienta dati:

Pakalpojuma dati:

Izejvielu dati:

Darījuma dati:

Ieviešanas plāns – plāno, kā ieviest programmatūru dzīvē, nosakot konkrētus termiņus un mērķus, kā arī aprakstot konkrētās lietošanas grupas.

## 7. Piemērotās licences pamatojums







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

### Atvērtā koda licences:

- Licences bez autortiesību ierobežojumiem (Copyleft) licences - jāizplata tikpat brīvi
  - Aparatūrai
    - CERN OHL ( open hardware licence) līdzīgas atļaujas kā atvērtā pirmkoda programmatūrai)
    - TAPR OHL līdzīgas atļaujas kā atvērtā pirmkoda programmatūrai, bet atvērtā koda aparatūrai
  - Vispārējās
    - GNU GPL (General Public License) – nedrīkst piemērot papildu nosacījumus, kas slēptu kodu
    - CC Share Alike - izveidoto darba atvasinājumus nedrīkst licencēt stingrāk
- Atļauju dodošas (permissive) licences
  - MIT licence jānorāda atsauce uz autoru (*attribution*)
  - Free BSD - Berkley Software Distribution – modificētās licences
  - Creative Commons (CC) Attribution - jānorāda atsauce uz autoru (*attribution*)

Atvērtā koda bibliotēka parasti nozīmē, ka kodu var mainīt, kopēt, publicēt, bet dažreiz autors ierobežo:

- Neaizsargātie darbi (public domain):
  - lietot var visi bez ierobežojumiem,
  - nav jāmaksā,
  - nav autortiesību.
- Brīvprogrammatūras(freeware):
  - lietot var visi, bet var neļaut kopēt, izplatīt un nepieejams kods,
  - nav jāmaksā,
  - ir autortiesības.
- Brīvā programmatūra (free software):
  - lietot var visi, pat kopēt un izplatīt kodu,
  - var prasīt ziedojumus, licence var aizliegt ierobežot atvasinātos darbus,
  - ir autortiesības.
- Atvērtā pirmkoda programatūra (open-source software):
  - lietot var visi, pat kopēt, mainīt tiešsaistēs grupās, bet var būt licence,
  - kodu nav jāmaksā, bet par papildpakalpojumiem lietotājiem var prasīt apmaksu,
  - ir autortiesības.

Saīsinājumi	Skaidrojumi
	CC - Creative Commons, atvērtā licence. Tā aizsargā tikai daļu no autortiesībām. Autors var izvēlēties dažādus atvērto licenču paveidus, lai precīzāk nodēfinētu, ko vēlas un ko nevēlas atļaut darīt ar savu darbu.
	BY - attribution, var lietot, kopēt, pavairot, veidot atvasinātus darbus, miksēt, bet jāuzrāda autora vārds uzvārds, kas veidojis darbu, jābūt atsaucei.
	SA - share alike, tad, kad cits autors izveido darba atvasinājumus, tos nedrīkst licenzēt ar stingrāku licenci nekā sākotnējā licence. Savukārt, ja nav SA, tad, izmantojot kāda darbu un remiksējot, veidojot atvasinājumus rezultātu, drīkst licenzēt stingrāk.
	NC - non-commercial, nelietot komerciāliem nolūkiem, nedrīkst gūt peļņu no darba izmantošanas.
	ND - no derivatives, nedrīkst veidot darba atvasinājumus, izmainīt vai pārveidot darbu, piemēram, lietot daļu no attēla kopā ar citiem attēliem.
	Public domain - visbrīvākā licence, darbu var lietot, veidojot kopijas, darbu var mainīt, var izmantot komerciālos nolūkos, var mainīt licences veidu

### Darbā noteikti jābūt:

#### 1. Datu bāzei ar vairākām tabulām:

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

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

#### 2. Vismaz viena bibliotēka un/vai API:

SR: 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.

SR: 2.4.11. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai.

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

#### 3. Vismaz divām dažādām datu struktūrām:

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

SR: 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.

#### 4. Lietotāja piekļuvei ar datu aizsardzību.

SR: 3.1.2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā