**PROGRAMMĒŠANA**

**Augstākais mācību satura apguves līmenis Metodiskie ieteikumi,**

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

2022

## Satura rādītājs

[Ievads 3](#_bookmark0)

1. [Valsts pārbaudes darba struktūra 4](#_bookmark1)
2. [Valsts pārbaudes darba piekļuves nosacījumi 6](#_bookmark2)

[Dažas idejas projektu veidošanai 6](#_bookmark3)

[Dokumentācijai jāsatur visi programmatūras dzīves cikla etapi 6](#_bookmark4)

[Minimālās prasības programmvadāmajam risinājumam 6](#_bookmark5)

[Augšupielāde 7](#_bookmark6)

1. [Katras daļas specifika, uzdevuma paraugs ar risinājumu un vērtējumu 8](#_bookmark7)
   1. [daļa. Datortīkls un droša datubāze (DB) 8](#_bookmark8)
   2. [daļa. Programmatūras dzīves cikls 13](#_bookmark9)
   3. [daļa. Objektorientētā programmēšana (OOP) un ārējās bibliotēka 17](#_bookmark10)
   4. [daļa. Dažādas datu struktūras, programmsaskarne (API) un mašīnmācīšanās principi 22](#_bookmark11)
2. [Ieteikumi, kā gatavoties eksāmenam 26](#_bookmark12)
3. [Palīglīdzekļi, kurus atļauts izmantot eksāmena laikā 26](#_bookmark13)

[Ieteicamās vietnes 26](#_bookmark14)

[Pielikums 27](#_bookmark15)

[Šogad VPD izmantos SR, kuri norādīti eksāmena paraugā 27](#_bookmark16)

[VPD piekļuves darba dokumentācija 28](#_bookmark17)

## 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 pielaides 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 atkārtošanas 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.

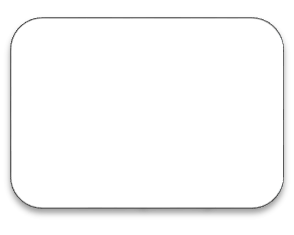
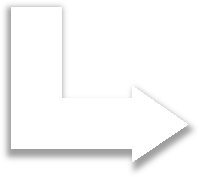
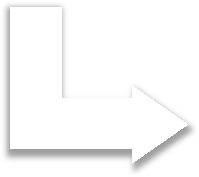
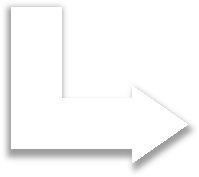
Kaspars Špūle

VISC Vispārējās izglītības pārbaudījumu nodaļas vadītājs Rīgā, 2022. gada novembrī

## 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. daļa.**

**Datortīkls un droša datubāze**

* Datu strukturēšana relāciju datu bāzē
* 20 punkti
* 1 uzdevums

**2. daļa.**

**Programmatūras dzīves cikls**

* Izpēte, mērķauditorija, plānošana, specifikācija
* 20 punkti
* 1 uzdevums

**3. daļa.**

**OOP un ārējās**

**bibliotēkas**

* Izveidot OOP programmu
* 38 punkti
* 1 uzdevums
* izpildes laiks 80 min
* **4. daļa.**
  + **Datu**
* Izveidot API pieprasījumu un 2 programmas

**struktūras,** •42 punkti (28\*1,5)

**programmsaska** •2 uzdevumi

**rne (API)** •izpildes laiks 80 min

0 min

30

0 min

1. daļas un 2. daļas kopīgais izpildes laiks:

70 min

1. attēls. Valsts pārbaudes darba struktūra

1. daļā dots situācijas apraksts, kuram jāmodelē datu bāze. Uzmanīgi izanalizē un izplāno relāciju (tabulu savienošanu) 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, atkārto 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.

## 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 jāsatur 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 |
| 1. Programmatūras kods, kas veidots ievērojot labās prakses principus – jāievieto pielikumā.    1. Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam. (1p)    2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā (1p)    3. Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām. (6p)    4. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai (4p)    5. Ievērojot labās prakses OOP principus izveidota vismaz viena bāzes klase ar 2 atvasinātajām klasēm (4p)    6. Datu ievadīšana / izvadīšana / rediģēšana (4p) | 20 |
| 5. Atkļūdošanas 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 kārto augstskolā, piekļuves materiālus neiesniedz.
    - Pedagogs materiālus izvērtē ne vēlāk kā sešas nedēļas pirms eksāmena norises dienas un vērtējumu ievada informācijas sistēmā.

# Katras daļas specifika, uzdevuma paraugs ar risinājumu un vērtējumu

## 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** | |  |  |  | Tabula **Prieksmets** | |
| **Lauka nosaukums** | **Datu tips** | **Datu piemērs** |  | **Lauka nosaukums** | **Datu tips** | **Datu piemērs** |
| **skolotaja\_id** | **TEXT** | **kset45q-23gfd-**  **kvc72-laxm79-bz** | **skolotaja\_id** | **TEXT** | **kset45q-23gfd-**  **kvc72-laxm79-bz** |
| **vards** | **TEXT** | **Sandra** | **skolena\_id** | **TEXT** | **Tdfr73q-28qld-**  **xzc12-lhd9-cmh** |
| **uzvards** | **TEXT** | **Skudra** | **priekšmets** | **TEXT** | **Matemātika** |
| **mob\_nr** | **TEXT** | **29010101** |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabula **Skolens** | |  |  |  | Tabula | |
| **Lauka nosaukums** | **Datu tips** | **Datu piemērs** |  | **Lauka nosaukums** | **Datu tips** | **Datu piemērs** |
| **skolena\_id** | **TEXT** | **Tdfr73q-28qld- xzc12-lhd9-cmh** |  |  |  |
| **vards** | **TEXT** | **Alīna** |  |  |  |
| **uzvards** | **TEXT** | **Krauze** |  |  |  |
| **mob\_nr** | **TEXT** | **29010102** |  |  |  |
| **klase** | **TEXT** | **11.a** |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Skolotaji** | |
| **skolotaja\_id** | **uzvards** |
| 1 | Lapiņš |
| 2 | Čakāne |

|  |  |  |
| --- | --- | --- |
| Tabula Skolotajs | | |
| Lauka  nosaukums | Datu tips | Datu piemērs |
| Skolotaja ID | TEXT | 1,2,3,4,5 |
| ... | ... | ... |

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Risinājums | | | | | | | | | | | | Risinājuma analīze |
|  | | | | | | | | | | | | Laukā ‘Prieksmets’ nedrīkst būt masīvs, ***laukā*** jābūt tikai ***vienai vērtībai***, pēc kuras var atlasīt informāciju. ***Lauka nosaukumā nedrīkst būt tukšumi un diakritiskās zīmes.*** |
|  | **Prieksmeti** | | | | | | | | |  | |
|  | **uzvards** | **prieksmets** | | | | | | | |
|  | Lapiņš | Latviešu valoda, literatūra | | | | | | | |
|  | Čakāne | Matemātiķa, datorika,  programmēšana | | | | | | | |
|  | | | | | | | | | | | | Pārkāpta 1.normālforma: “Katrā laukā tikai viena **vērtība un ieraksti neatkārtojas”.** |
|  | **Prieksmeti** | | | | | | | | | | |
|  | **uzvards** | | | | | **prieksmets** | | | | | |
|  | Lapiņš | | | | | latviešu valoda | | | | | |
|  | Lapiņš | | | | | literatūra | | | | | |
|  | Čakāne | | | | | matemātiķa | | | | | |
|  | Čakāne | | | | | datorika | | | | | |
|  | Čakāne | | | | | programmēšana | | | | | |
|  | | | | | | | | | | | | Skolotājam vēl jānorāda vārds, personas kods, konts un pārējie dati nākamajos laukos (nākamajās kolonnās). |
|  | | | | | | | | | | | | Skolēniem vēl jānorāda vārds, personas konts, klase un pārējie dati nākamajos laukos (nākamajās kolonnās). |
|  | **Skoleni** | | | | | | | |  | | |
|  | **skoleni\_id** | | | | **uzvards** | | | |
|  | 1 | | | | Lapiņš | | | |
|  | 2 | | | | Čakāne | | | |
|  | | | | | | | | | | | | Priekšmetos redzam, kurš skolotājs kuram skolēnam māca kādu priekšmetu, var izmantot klases, grupas. |
|  | **Prieksmeti** | | | | | | | | | | |
|  | **prieksmeta\_id** | | | **skolotaja\_ id** | | | | **prieksmets** | | | |
|  | 1 | | | **1** | | | | latviešu valoda | | | |
|  | 2 | | | **1** | | | | literatūra | | | |
|  | 3 | | | **2** | | | | matemātiķa | | | |
|  | 4 | | | **2** | | | | datorika | | | |
|  | 5 | | | **2** | | | | programmēšana | | | |
|  | | | | | | | | | | | |
|  | | | | | | | | | | | | 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-fhso94vnmdl, 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. |
|  | Tabula Skolotajs | | | | | | | | | |  | Ja piemērā ir skaitlis, tad datu tipam jābūt skaitliskam AutoNumber. |
|  | Lauka  nosaukums | | Datu tips | | | | Datu piemērs | | | |
|  | skolotaja\_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 | | |
|  | ... | ... | | ... | | |
|  | | | | | | | | 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. |
|  | Tabula Skolotajs | | | | | |  |
|  | 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 ***tikai*** 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 izvēlēsies 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.

## 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 atlases 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 iesaistīsi 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 lietotajiem aprobācijai 3. Apskatos informācijas izmantošanas tendenci 4. Uztaisu nākamo prototipu ar izmantotāko informāciju 5. Testēju 6. Secinu | Šeit aprakstīti gandrīz viss **produkta** dzīves posms, nevis izpētes procesa **plāns.**  5. punkts bija dots, tātad tas nav jālabo. |
| 1. Analizēšu iegūtos datus 2. Projektēšu digitālo līdzekli 3. Izstrādāšu digitālo līdzekli 4. Skenēšu un testēšu izstrādāto 5. Dokumentēšu izstrādāto līdzekli 6. Atbalstīšu ekspluatāciju | Aprakstīts produkta **izstrādes** process, nevis tas, kā **plānos** situācijas aprakstu. |

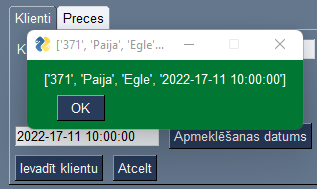
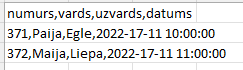
|  |  |  |
| --- | --- | --- |
|  | 1. **Sagatavošanās posms** – izveidoju aptaujas 2. **Norise** - intervēju īpašnieci, digitālā rīka lietotāju, par informācijas lietošanas vēlamajiem veidiem. 3. **Rezultātu apkopošana** - no aptaujas rezultātiem noskaidrošu lietotāja nepieciešamības. 4. **Izvērtēšana** – apkopoju iegūto informāciju, piedāvāju Dainai informācijas galējo rezultātu. 5. **Secināšana** - Izdaru secinājumus, lai varētu veikt turpmākās darbības. 6. **Secinājumu formulēšana**. Formulēju secinājumu un piedāvāju vajadzības saistībā ar programmatūras produkta izstrādi. | Dotās situācijas izpētes plāns. |
|  |  |  |
| Ņemot vērā uzdevumā doto problēmsituāciju un tajā pieejamo informāciju, īsi apraksti, kādu programmvadāmo risinājumu ir nepieciešams izstrādāt | Apmeklētāju uzskaites dažādība apakšdatnes risinājuma \*.csv | Lietotājam nebūs ērti atvērt/izlabot/saglabāt \*.csv datnes – tas nav  programmrisinājums. |
| Salona viesus raksturojoša  informācija un statistika. | Tas nav programmrisinājums. |
| Datubāze - jo tajā var glabāt  daudz datus. | Tas nav programmrisinājums. |
| Izstrādāšu aplikāciju ņemot vērā  lietotāju specifiku. | Ir programmvadāms risinājums. |
| Tiks izveidota viegli pārskatāma mājaslapa, kurā varēs nolasīt no datu bāzes datus, apskatīt, labot  un saglabāt. Ievadīt jaunus datus | Ir programmvadāms risinājums. |
| Īsi apraksti programmvadāmā risinājuma datu uzglabāšanas veidu un datu fizisko izvietojumu | **MySQL datubāzē skaisti viss sakārtots tabulā.** | Datu bāze nosaukta pareizi, bet  jāatzīmē arī, kur tā atrodas |
| Oracle datubāze, kas izvietota uzņēmuma serverī. | Pareizās atbildes |
| Lokālajā failu serverī ar datubāzi. |
| Datus uzglabā MySQL datubāzē,  kas izvietota nomas hostinga serverī. |
| 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 | Persona{  personas\_id – skaitlisks, veidots-ģenerēts  vards (TEXT) uzvards (TEXT) mob\_talr (TEXT)  }  Alga{  darbinieka\_id (TEXT) klienta\_id (TEXT) ddiena (TEXT/Date) cena (Float) kvalit(INT/Bool) plkst(INT)  } | Persona ir bāzes klase, no kuras jāatvasina Darbinieka un Klienta klase. Darbiniekam vēl jābūt personas kodam un konta numuram.  Nepieciešama arī klase Alga kvalit(INT/Bool) – ja klients nav apmierināts, tad darbu neieskaita plkst(INT) – tikai precīzi laiki. |
| Ievaddatu mērvienības, iespējamās vērtības | **klients1**=Klients(Kate, Krūmiņa, 28765431)  **darbinieks1**=Darbinieks(Dina, Dobe,  020200-12345, 21345678)  **alga** = Alga(22.12.12, 15:00,  35.00 eiro, 1,) | 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 jaunos personas, datus un labos tos arī saskarnē. | Pareizi. |
| Metodes nosaukums un deklarācija | Metodes nosaukums un deklarācija?  aprekina\_algu(/Real/Float/) aprekina\_pelnu( /Real/Float/) | Izmanto ‘camelCase’ vai ‘snake\_case’ stilu, bez  diakritiskajām zīmēm, bez pieturzīmēm, bez tukšumiem |
| Kā iegūt (aprēķināt, kombinēt, formatēt) rezultātu? | Katram darbiniekam aprēķināt nopelnīto uzņēmumam un pareizināt ar 20%  Nopelnīto mēnesī aprēķināt, saskaitot visu darbinieku nopelnīto – darbiniekiem izmaksāto | Pareizi. |
| Kā jāsniedz atbilde (atgriežot vērtību no funkcijas, izvadot atbildi uz ekrāna, datnē, DB)? | Uz ekrāna izvadīt, kurš klients uz kuru datumu pierakstīts,  kurš darbinieks kurā datumā cikos strādā,  Ievadot darbinieka personas kodu, izdrukāt nopelnīto. |  |

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

## daļa. Objektorientētā programmēšana (OOP) un ārējās bibliotēka Darbs ar datnēm

|  |
| --- |
| Atrod atšķirību kodā, lai saprastu, kāpēc tā drukā |
| Attēls, kurā ir teksts  Apraksts ģenerēts automātiski  Attēls, kurā ir teksts  Apraksts ģenerēts automātiski |
|  |



|  |
| --- |
| **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 bazes klasi  **def**  **init** (self, vards): # konstruktors self.vards = vards # figuras nosaukums  **def perimetrs**(self): # metode perimetra aprekinashanai  **pass** #tuksha metode |

|  |
| --- |
| **def druka**(self): # 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  **class Trijsturis**(Figura): #klase Trijsturis atvasinata no klases Figura  **def**  **init** (self, vards, mala1, mala2, mala3): #inicialize klases objektu  self.vards = vards # objekta vards self.mala1 = mala1 # trijstura mala self.mala2 = mala2 # trijstura mala self.mala3 = mala3 # trijstura mala  **def perimetrs**(self): #metode aprekina 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, vards, mala1, mala2): #inicialize klases objektu self.vards = vards #objekta vards  self.mala1 = mala1 # taisnstura mala self.mala2 = mala2 # taisnstura mala  **def perimetrs**(self): # metode aprekina taisnstura permetru self.rez = **2** \* (self.mala1 + self.mala2) # konkreta taisnstura  perimetrs  **return** self.rez # atgriez taisnstura perimetru  **class Kvadrats**(Figura): #klase Kvadrats atvasinata no klases Figura  **def**  **init** (self, vards, mala1): #inicialize klases objektu self.vards = vards #objekta vards  self.mala1 = mala1 # kvadrata mala  **def perimetrs**(self): # metode kvadrata perimetru aprekina self.rez = **4** \* self.mala1 # konkreta kvadrata perimetrs **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('**\n**Kvadrāta malu garumi : ')) |

|  |
| --- |
| # izveido objektu trijsturis  trijsturis = Trijsturis('**\n**Trijstūra 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  # izdruka doto objektu nosaukumus un perimetrus  **for** fig **in** figuras: # apstrada visus saraksta elementus  **print**(fig.druka()) # drukaa  #raise NotImplementedError("kļūda") |
|  |

|  |  |
| --- | --- |
| **Risinājuma detaļas** | **Risinājuma analīze** |
| **class Figura**: # Define bazes klasi | Bāzes klasei nav iekavu. |
| **def**  **init** (self, vards): # konstruktors self.vards = vards # figuras nosaukums | Bāzes klases konstruktors. |
| **def int** (self, vards, a, b, c): # 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. |
| **def perimetrs**(self): # metode perimetra aprekinashanai  **pass** #tuksha metode | Ja atvasinātajā klasē ir metode, tad bāzes klasē tā vairs netiek meklēta. |
| **def druka**(self): # 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 |
| **def druka**(self): # 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) |

|  |  |
| --- | --- |
| **class Kvadrats**(Figura): #klase Kvadrats atvasinata no klases Figura  **def init** (self, vards, a): #inicialize klases objektu self.vards = vards #objekta vards  self.a = a # kvadrata mala  **def perimetrs**(self): # metode kvadrata perimetru aprekina  self.rez = **4** \* self.a # konkreta kvadrata perimetrs  **return** self.rez # atgriez kvadrata perimetru | Klasei  **class Kvadrats**(Figura): iekavās ieraksta bāzes klasi, no kuras tā mantos atribūtus un metodes.  Konstruktorā  **def init** (self,vards,a):  ir formālie argumenti: ‘vards’ un ‘a’ ,  bet, ja metodē  ‘**def perimetrs**(self): ’ ierakstot tos pašus formālos argumentus nestrādās polimorfisms, jo visām 3 atvasinātajām klasēm ir dažāds  argumentu skaits. |
| **Vai trijstūri var izveidot no jebkuriem malu**  **garumiem?** | Izveido pārbaudi, vai no dotajiem  lielumiem sanāks trijstūris! |

|  |  |
| --- | --- |
| Ķļūdainie risinājumi | Analīze |
|  | Klases, parametru, atribūtu, metožu nosaukumos nedrīkst izmantot latviešu diakritiskās zīmes, punktus, komatus, tukšumus. |
|  | Klase ir kā šablons, kurš sastāv no atribūtiem un metodēm, ar kuriem konstruē objektus, nedrīkst šādi norādīt konkrētus datus. |
|  | Pitonā par atkāpju neievērošanu soda uzreiz, neraksta ‘indent’, bet parādās sarkans komats kā brīdinājums. |
|  | Šeit nav konstruktora, izlaists viens burts ‘i’. |

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.

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

* 1. **uzdevums (14 punkti)**

Dots API domēns: https:/[/www](http://www.boredapi.com/api).[boredapi.com/api](http://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.

## 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 atkārtojušies 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, angliski – requests. | import requests | 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. | import json | Pievieno JSON bibliotēku ar nosaukumu json. Tas palīdz darboties ar JSON objektiem. |
| Pieprasi datus no API servera, šos datus saglabā mainīgajā ar nosaukumu atbilde. | atbilde = requests.get("https://dog.ceo/api/breeds/image/random") | Mainīgajam 'atbilde' tiek piešķirts pieprasījuma datu saturs. https://dog.ceo/api/breeds/image/random - šajā vietnē tiks ģenerēti nejauši suņu attēli. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| No tīmekļa vietnes iegūsti nejaušu suņa attēlu JSON formātā. | print(atbilde.status\_code) | JSON formāts – populārs datu pārsūtīšanai. Iespējamās atbildes JSON formātā: 200 – Sekmīga pārsūtīšana. Dati tiek atsūtīti. 404 – Resursi nav atrasti uz servera. |

|  |  |  |
| --- | --- | --- |
| Izvadi saiti uz nejaušu suņa attēlu JSON formātā. | print(atbilde.json()) | Ja tika pareizi pieprasīti dati, kas tika saglabāti mainīgajā ar nosaukumu 'atbilde', šie dati tiks izdrukāti neformatētā veidā. |
| Izvadi JSON datus pārskatāmā veidā. | def jsonprint(obj): izvade = json.dumps(obj, sort\_keys=True, indent=4) print(izvade) jsonprint(atbilde.json()) | Veido funkciju jsonprint, kas saturēs un sakārtos objektu. json.dumps() – pārvērtīs Python objektus par json simbolu virkni. |
|  |
| Izvadi JSON datus pārskatāmā veidā. | dati = atbilde.json() suņa\_attēls = dati['message'] print(f"Nejauša suņa attēls: {suņa\_attēls}") | No atbildes JSON iegūst vajadzīgo informāciju un izdrukā to konsolē. |

|  |  |  |
| --- | --- | --- |
|  | | |
| **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 | import re | Pievieno bibliotēku, lai rakstītu regulārās  izteiksmes (regular expressions). |
| Ievieto bibliotēku | import string | Pievieno simbolu virknes bibliotēku. |
| Izveido vārdnīcu | biezhums = {} | Izveido tukšu vārdnīcu, kur glabās biežumus. |
| Atver teksta datni ar nosaukumu test.txt | teksts =  open('test.txt', 'r', encoding='utf-8') | 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 (https://regex101.com – 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 |

# 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.
  + Atkārto “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.

# Palīglīdzekļi, 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-](https://www.youtube.com/watch?v=nZLrNDAqLF8&list=PLLls_n2ybYppgO-tV7n_r8OGnOboj17So&index=17) [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.

## daļas SR - Programmatūras Dzīves Cikls

* + 1. Analizē dažādus ikdienas darba procesus, saskata tajos vai to daļās automatizācijas iespējas un to, kā pasūtītājs formulē darba uzdevumu programmatūras izstrādātājam.
    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.

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

* + 1. 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.

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

* + 1. 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.

1. pielikums

## VPD piekļuves darba dokumentācija

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

## 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 pieniņu, vatītes plāksnīšu 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 atlases 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**.**

## Programmatūras prasību specifikācija - 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 atvērtā pirmkoda programmatūra, kurā var brīvi izmantot bibliotēkas, kuras nemitīgi 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, pieniņu izmantošanas termiņa noteikšana un klientu pierakstīšana uz procedūrām. | Kosmetoloģijas salona apmeklētāju un izejvielu uzskaite. |

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, pieniņu izmantošanas termiņa noteikšana un klientu pierakstīšana uz procedūrām. | Apmeklētāju grafika izveide un izejvielu automatizēta derīguma termiņa uzskaite. |

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. |

Tehniskās detaļas:

Datus uzglabāt datu bāzē.

Ja kāda izejviela tiek atvērta, tad tai nomainīt derīguma termiņu.

* Izvēloties vielas, izvadīt visu vielu nosaukumus, skaitu un derīgumu termiņus.
* Izvēloties klientu, izvadīt klienta pierakstīšanās datumu, ja tāda nav, iespējamos pierakstīšanās datumus, atļaujot mainīt esošo datumu.
  + 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:

### Kādi ievaddati tiks doti?

Klientu, izejvielu dati, apmeklēšanas datumu un laiku.

1. ***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){

}

### Kādā veidā tiks iesniegti ievaddati (izsaucot funkciju, no standarta ievades, no datnes, no DB)

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

### Metodes nosaukums un deklarācija?

izejvielu\_skaits(nosaukums, derigums) - skaitliska

### 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.

### 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.

### Izvaddatu tips, mērvienības, iespējamās vērtības?

Izvadīt saskarnē, datu bāzē teksta datus, algu eiro.

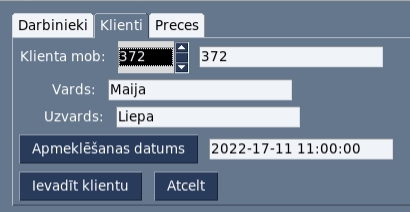
### Doti piemēri ar konkrētiem ievaddatiem un atbilstošu sagaidāmo rezultātu

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.)*

## 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***.)**

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

1. **def kodu**(self):
2. dat = self.nolasa()
3. kods = []
4. **for** rinda **in** dat:
5. kods.append(rinda[**0**])
6. **return** (kods)

56

1. **def layout\_pers**(self, kods, key\_li): # ================ izkaartojums #format = '%Y-%m-%d'
2. izkarto\_pers = [[sg.Text('Personas mob:'), sg.Listbox(kods, enable\_events=True, key='-NRP'),
3. sg.Input(key='P-NRP-', enable\_events=True, size=(**20**, **1**))],
4. [sg.Text(' Vards:'), sg.Input(key='P-VARDS-', enable\_events=True, size=(**20**, **1**))],
5. [sg.Text(' Uzvards:'), sg.Input(key='P-UZVARDS-', enable\_events=True, size=(**20**, **1**))],
6. [sg.Button('Ievadīt klientu'), sg.Exit('Atcelt')]]
7. **return** izkarto\_pers

64

1. **class Darbinieks**(Persona):
2. **def layout\_pers**(self, kods,key\_li):
3. izkarto\_pers = [[sg.Text('Darbinieka mob:'), sg.Listbox(kods, enable\_events=True, key='-NRD'),
4. sg.Input(key='D-NR-', enable\_events=True, size=(**20**, **1**))],
5. [sg.Text(' Vards:'), sg.Input(key='D-VARDS-', enable\_events=True, size=(**20**, **1**))],
6. [sg.Text(' Uzvards:'), sg.Input(key='D-UZVARDS-', enable\_events=True, size=(**20**, **1**))],
7. [sg.CalendarButton('Datumi',target='D-DATUMS-', format='%Y-%m-%d'),
8. sg.Input(key='D-DATUMS-', enable\_events=True, size=(**70**, **1**)), ],
9. [sg.Button('Ievadīt dienas'), sg.Exit('Atcelt')]]
10. **return** list(izkarto\_pers)

75

76

1. **class Klients**(Persona):
2. **def layout\_pers**(self, kods, key\_li):
3. izkarto\_pers = [[sg.Text('Klienta mob:'), sg.Listbox(kods, enable\_events=True, key='-NRK'),
4. sg.Input(key='K-NR-', enable\_events=True, size=(**20**, **1**))],
5. [sg.Text(' Vards:'), sg.Input(key='K-VARDS-', enable\_events=True, size=(**20**, **1**))],
6. [sg.Text(' Uzvards:'), sg.Input(key='K-UZVARDS-', enable\_events=True, size=(**20**, **1**))],
7. [sg.CalendarButton('Apmeklēšanas datums',target='K-DATUMS-', format='%Y-%m-%d'),
8. sg.Input(key='K-DATUMS-', enable\_events=True, size=(**20**, **1**))],
9. [sg.Button('Ievadīt klientu'), sg.Exit('Atcelt')]]
10. **return** list(izkarto\_pers)

87

88

1. # ==================================================== main
2. key\_ievad = ('-NR-', '-VARDS-', '-UZVARDS-', '-DATUMS-')
3. key\_li = ('-NRD', '-NRK', '-NRP')
4. galvene = ['numurs', 'vards', 'uzvards', 'datums']
5. # ==================================================== dati saskarnei
6. darbinieks = Darbinieks('', '', '', '', '', galvene, key\_li, 'darbinieki.csv')
7. darbinieki = darbinieks.nolasa()
8. darbinieku\_kod = darbinieks.kodu() 97
9. klients = Klients('', '', '', '', '', galvene, key\_li, 'klienti.csv')
10. klienti = klients.nolasa()
11. klientu\_kod = klients.kodu() 101
12. preces = [['krēms', 'pieniņš', 'špātulas']]
13. precu\_kodi = [['P001', 'P002', 'P003']]
14. # ==================================================== izkaartojums
15. layout\_preces = [[sg.Text('Preces:'), sg.Listbox(precu\_kodi, enable\_events=True, key='-NRP'),
16. sg.Input(key='P-NR-', size=(**20**, **1**))],
17. [sg.Input(key='P-DATUMS-', size=(**20**, **1**)), sg.CalendarButton('Termiņš', target='P-DATUMS-',
18. default\_date\_m\_d\_y=(**1**, None, **2022**))],
19. [sg.Button('Ievadīt preci'), sg.Exit('Atcelt')]] 110

111 # ==================================================== tabulaacijas

112

1. tab\_group = sg.TabGroup([[
2. sg.Tab("Darbinieki", darbinieks.layout\_pers(darbinieku\_kod, key\_li), key='-TABD-'),
3. sg.Tab("Klienti", klients.layout\_pers(klientu\_kod, key\_li), key='-TABK-'),
4. sg.Tab("Preces", layout\_preces, key='-TABP-')

117 ]])

1. col = [[tab\_group]]
2. layout = [[sg.Column(col)]]
3. window = sg.Window('Skaistumkopšanas kabinets', layout) 121
4. # ==================================================== Notikumi:
5. event = ''
6. key\_ievadi= ('D-NR-', 'D-VARDS-', 'D-UZVARDS-', 'D-DATUMS-','K-NR-', 'K-VARDS-', 'K-UZVARDS-', 'K-DATUMS-','P-NR-',
7. 'P-DATUMS-')

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

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

## Atkļūdošanas un akcepttestēšanas pārskats

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

## 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 | Tālruņa 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. |

## 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 |  |  |  |

## 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:
   1. Ko sistēma atbalsta, kādus pārlūkus?
   2. Kā var pieslēgties sistēmai?
   3. Kas notiks, ja pieslēdzas un kļūdās paroles vai lietotājvārda ievadē?
   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.

## 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
  + Vispārējas
    - 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 nodefinētu, 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  licencēt 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.

## 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.

## 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.

## Lietotāja piekļuvei ar datu aizsardzību.

SR: 3.1.2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā