“Programmēšana (augstākais mācību satura apguves līmenis)”

VPD vērtēšanas kritēriji

2022./2023. mācību gads

1. DAĻA – DATORTĪKLS UN DROŠA DATUBĀZE (20 punkti)

1. uzdevuma vērtēšanas kritēriji

Kritēriji	Punkti
Korekti izveidots vismaz viens lauks katrā no tabulām *SAIMNIEKI* un *PRODUKTI*.	1
Korekti izveidoti visi lauki katrā no tabulām *SAIMNIEKI* un *PRODUKTI*.	1
Korekti izveidoti visi lauki visās 3 tabulās.	1
Ievadīts vismaz viens korekts datu tips katrā no tabulām *SAIMNIEKI* un *PRODUKTI*.	1
Cenas un summas lauku datu tipi atļauj saglabāt decimālos skaitļus.	1
Visiem laukiem visās 3 tabulās ir iestatīti pareizie datu tipi.	1
Jebkurā vienā tabulā ir izveidots ID-lauks.	2
Visās tabulās ir izveidoti ID-lauki.	2
Visiem ID-laukiem ir teksta virknes vai skaitliskais datu tips.	1
Dots ID-lauka izvēlētā datu tipa pamatojums.	1
Izveidota relācija starp tabulām *SAIMNIEKI* un *PRODUKTI*.	1
Izveidotas 2 relācijas, tādējādi sasaistot savā starpā visas 3 tabulas.	1
Relācijai starp tabulām *SAIMNIEKI* un *PRODUKTI* ir lietots atbilstošs apzīmējums.	2
Visām relācijām ir lietoti atbilstoši apzīmējumi.	2
Vismaz 3 laukiem ir ievadīti korekti testa datu piemēri.	1
Visiem laukiem ir ievadīti korekti testa datu piemēri.	1
Kopā:	**20**

2. DAĻA – PROGRAMMATŪRAS DZĪVES CIKLS (20 punkti)

1. uzdevuma vērtēšanas kritēriji

Kritēriji	Punkti
Nosaukta precīza izpētes metode.	1
Ir sniegts nosauktās izpētes metodes pamatojums, kurā ir skaidri redzama saistība ar doto problēmsituāciju.	1
Definēta precīza mērķauditorija, kas būs iesaistīta izpētes procesā un izmantos risinājumu.	1
Ir sniegts definētās mērķauditorijas pamatojums, kurā ir skaidri redzama saistība ar doto problēmsituāciju.	1
Nosaukti un īsi paskaidroti **vismaz divi** izpētes procesa **soļi**, tiem ir saistība ar 1. un 2. uzdevumā sniegto informāciju.	1
Nosaukti un īsi paskaidroti pieci izpētes procesa soļi, tiem ir saistība ar 1. un 2. uzdevumā sniegto informāciju.	1
Ir minēti ievaddatu nosaukumu piemēri.	1
Ir minēti ievaddatu tipi un mērvienības.	1
Ir minētas ievaddatu iespējamās vērtības.	1
Ir minēts ievaddatu iesniegšanas veids.	1
Ir minēts izvaddatu tips.	1
Ir minētas izvaddatu mērvienības.	1
Ir minēti izvaddatu iespējamie saglabāšanas veidi.	1
Ir piedāvāta kāda papildfunkcija.	1
Ir minēti 2 piemēri ar konkrētiem ievaddatiem un sagaidāmajiem rezultātiem, kā arī katram piemēram ir minēti konkrēti veidi, kā šo rezultātu var sasniegt.	3
Tiek ievērota labā prakse mainīgo un lauku nosaukumu veidošanā.	1
Ir nosaukts piemērotākais programmatūras izstrādes modelis.	1
Ir dots pamatojums, kura argumenti ir saistīti ar doto problēmsituāciju (“darbinieki labprāt sadarbosies ar izstrādātājiem”), mērķauditoriju u. tml.	1
Kopā:	**20**

3. DAĻA – OOP UN ĀRĒJĀS BIBLIOTĒKAS (38 punkti)

1. uzdevuma vērtēšanas kritēriji

Kritēriji	Punkti
Definēta bāzes klase un tās konstruktors.	1
Konstruktors satur parametrus: **zimols, modelis, reg_datums, pilna_masa un degvielas_veids**.	1
Katram parametram tiek piešķirts attiecīgs atribūts (īpašības).	1
Izveidota metode, kas atgriež formatēta teksta veidā, objekta vērtības.	1
Izveidot objektu ar testa datiem: **Audi, A4, 22.10.2019, 1800, BG**.	1
Izdrukāti visi atribūti, kas definēti objektā.	1
Kopā:	**6**

1. uzdevuma risinājumi dažādās programmēšanas valodās

Python	JS	C#	C++
`class Transportlidzeklis:` `  def __init__`	`function Transportlidzeklis` konstruktors ir ķermenis	`class Transportlidzeklis` `{  public Transportlidzeklis`	`class Transportlidzeklis` `{  public Transportlidzeklis`

`self.zimols = zimols`
`self.modelis = modelis`
`self.reg_datums = reg_datums`
`self.pilna_masa = pilna_masa`
`self.degvielas_veids = degvielas_veids`
	`this.zimols = zimols;`
	`this.modelis = modelis;`
	`this.reg_datums = reg_datums;`
	`this.pilna_masa = pilna_masa;`
	`this.degvielas_veids = degvielas_veids;`
		`private string zimols;`
		`private string modelis;`
		`private string reg_datums;`
		`private int pilna_masa;`
		`private string degvielas_veids;`
			`private:`
			`std::string zimols;`
			`std::string modelis;`
			`std::string reg_datums;`
			`int pilna_masa;`
			`std::string degvielas_veids;`

		`this.zimols = zimols;`
		`this.modelis = modelis;`
		`this.reg_datums = reg_datums;`
		`this.pilna_masa = pilna_masa;`
		`this.degvielas_veids = degvielas_veids;`
			`this->zimols = zimols;`
			`this->modelis = modelis;`
			`this->reg_datums = reg_datums;`
			`this->pilna_masa = pilna_masa;`
			`this->degvielas_veids = degvielas_veids;`

`def __str__(self):`	`prototype.toString`	`public override string ToString()`	`std::string toString()`
`auto = Transportlidzeklis(“Audi”, “A4”, “22.10.2019”, 1800, “BG”)`	`var auto = new Transportlidzeklis(“Audi”, “A4”, “22.10.2019”, 1800, “BG”);`	`Transportlidzeklis(“Audi”, “A4”, “22.10.2019”, 1800, “BG”);`	`Transportlidzeklis auto1(“Audi”, “A4”, “22.10.2019”, 1800, “BG”);`
`print(auto)`	`document.write(auto.toString());`	`Console.WriteLine(auto);`	`std::cout << auto1.toString() << std::endl;`

PHP	Java	GoLang
class Transportlidzeklis { public function __construct(	class Transportlidzeklis { public Transportlidzeklis	type Transportlidzeklis struct { func izveidotTransportlidzekli(
private $zimols; private $modelis; private $reg_datums; private $pilna_masa; private $degvielas_veids;	private string zimols; private string modelis; private string reg_datums; private int pilna_masa; private string degvielas_veids;	zimols string modelis string registracijasDatums string pilnaMasa int degvielasVeids DegvielasVeids

$this->zimols = $zimols;
$this->modelis = $modelis;
$this->reg_datums = $reg_datums;
$this->pilna_masa = $pilna_masa;
$this->degvielas_veids = $degvielas_
veids;
	this.zimols = zimols;
	this.modelis = modelis;
	this.reg_datums = reg_datums;
	this.pilna_masa = pilna_masa;
	this.degvielas_veids = degvielas_veids;
		return &Transportlidzeklis{
		zimols: zimols,
		modelis: modelis,
		registracijasDatums:
		registracijasDatums,
		pilnaMasa: pilnaMasa,
		degvielasVeids: degvielasVeids}

public function __toString()	public String toString()	func (t Transportlidzeklis) izvadit()
$auto = new Transportlidzeklis(“Audi”, “A4”, “22.10.2019”, 1800, “BG”);	Transportlidzeklis auto = new Transportlidzeklis(“Audi”, “A4”, “22.10.2019”, 1800, “BG”);	transportlidzeklis := izveidotTransportlidzekli(“Audi”, “A4”, “22.10.2019”, 1800, BG)
echo $auto;	System.out.println(auto);	transportlidzeklis.izvadit()

2. uzdevuma vērtēšanas kritēriji

Kritēriji	Punkti
Izveidota bāzes klase **kubs**.	1
Aprakstīti klases **kubs** atribūti (īpašības): **malas garums** un **krāsa**.	1
Izveidots klases **kubs** konstruktors, visi nepieciešamie atribūti tiek inicializēti.	1
Pārbaude vai malas garums ir **no 2 līdz 10 ieskaitot**.	2
Ja mala ir **<2**, tiek iestatīta min vērtība, **malas garums = 2**.	
Izveidota metode **objekta dzēšanai**, kur tiek izdrukāts paziņojums, norādot dzēstā **objekta krāsu**.	1
Izveidota metode **aprekinat_tilpumu**, kas aprēķina un atgriež kuba tilpumu, kā veselu skaitli.	1
Izveidota kubs apakšklase **bloks**.	1
Apakšklasē **bloks** tiek nodrošināta atribūtu (īpašību) mantošana no klases **kubs**.	2
Izveidots apakšklases klases **bloks** konstruktors.	
Izveidots publisks atribūts (īpašība): **nosaukums**, kurš veidots no **kuba krāsas un skaita**, piemēram, orange4.	1
Izveidoti publiski atribūti (īpašības) **derīgums, forma**, vesels divciparu skaitlis ar iespējamām vērtībām **11, 12, 13, 14** un **22**.	2
Atribūta **forma** atbilstības pārbaude, atbilst **derīgums=0**, citādi **derīgums = 1**.	1
Izveidots privāts atribūts (īpašība) **kubu skaits**, kas ir vesels skaitlis **no 1 līdz 4 ieskaitot**.	1
Izvadīt paziņojumu: “**Neatbilst nosacījumiem**”, ja kubu skaits blokā nav no 1 līdz 4 ieskaitot.	1
Izveidota metode **tilpums**, kura aprēķina un atgriež veselu skaitli, **cm^3**.	1
Izveidots objekts **kubg**, kura krāsa ir **zaļa** un malas garums **10 cm**.	1
Izveidots objekts **kubr**, kura krāsa ir **sarkana** un malas garums **1 cm**.	1
Izdrukāt objekta **kubg krāsu** un **tilpumu**.	1
Izdrukāt objekta **kubr malas garumu**.	1
Tiek dzēsts objekts **kubr**.	2
Pārbaude, par objekta **kubr** pieejamību, un izvadīt ekrānā atbilstošu paziņojumu.	
Izdrukāt objekta **kubg** malas garumu.	1
Izveidots objekts ar nosaukumu **oranzs3** vai orange3 ar **oranžu krāsu 3**, **malas garumu 5 cm**, **formu 13**.	2
Izdrukāt izveidotā objekta **nosaukumu** un **tilpumu**.	1
Izveidots objekts ar nosaukumu **zils5** vai blue5 ar **zilu krāsu 5**, **malas garumu 7 cm**, **formu 23**.	2
Izdrukāt izveidotā objekta nosaukumu un derīgumu.	1
Nomainīt objektam formas numuru uz **12**.	1
Izdrukāt izveidotā objekta nosaukumu un derīgumu.	1
Kopā:	**32**

2. uzdevuma risinājumi dažādās programmēšanas valodās

Python	JS	C#
class Kubs:	class Kubs { vai function Kubs{	public class Kubs {
		private int malas_garums; private string krasa;

def __init__ (self, malas_garums, krasa):
self.malas_garums = malas_garums
self.krasa = krasa
	constructor(malas_garums, krasa) {
	this.malas_garums = malas_garums;
	this.krasa = krasa;
		public Kubs(int malas_garums, string krasa)
		{
		this.malas_garums = malas_garums;
		this.krasa = krasa;

if self.malas_garums not in range(2,11): self.malas_garums = 2	if (malas_garums < 2 || malas_garums > 10) { this.malas_garums = 2; }	if (malas_garums < 2 || malas_garums > 10) {this.malas_garums = 2;}
def __del__(self): print(f"Izdzēsts {self.krasa} objekts")	dzest() { document.write("Izdzēsts " + this.krasa + " objekts "); }	public void dzest() { Console.WriteLine("Izdzēsts "+krasa+" objekts"); }

def aprekinat_tilpumu(self):
tilpums = self.malas_garums ** 3
return tilpums
	aprekinat_tilpumu() {
	var tilpums = Math.pow(this.malas_garums, 3);
	return tilpums; }
		public int AprekinatTilpumu()
		{ int tilpums = (int)Math.Pow(malas_garums, 3);
		return tilpums; }

class Bloks(Kubs):
def __init__(self, malas_garums, krasa, kubu_skaits, forma):
super().__init__(malas_garums, krasa)
self.__kubu_skaits = kubu_skaits
	class Bloks extends Kubs {
	constructor(malas_garums, krasa, kubu_skaits, forma) {
	super(malas_garums, krasa);
	this.kubu_skaits = kubu_skaits;
	this.forma = forma;
		public class Bloks : Kubs
		{ private int kubu_skaits;
		private string forma;
		private int derigums;
		private string nosaukums;
		public Bloks(int malas_garums, string krasa, int kubu_skaits, string forma) : base(malas_garums, krasa)
		{ this.kubu_skaits = kubu_skaits;
		this.forma = forma;

self.nosaukums = f"{self.krasa}{self.__kubu_skaits}"	this.nosaukums = krasa + kubu_skaits;	this.nosaukums = krasa + kubu_skaits;

if forma in [11, 12, 13, 14, 22]:
self.derigums = 0
else:
print(“Bloks nederīgs”)
self.derigums = 1
	if (forma === “11” || forma === “12” || forma === “13” || forma === “14” || forma === “22”)
	{ this.derigums = 0;
	} else {
	document.write(“Bloks nederīgs
	”);
	this.derigums = 1; }
		if (forma == “11” || forma == “12” || forma == “13” || forma == “14” || forma == “22”)
		{ this.derigums = 0; }
		else {
		Console.WriteLine(“Bloks nederīgs”);
		this.derigums = 1; }

if self.__kubu_skaits not in range(1, 5): print(“Neatbilst nosacījumiem, kubu skaits blokā nav no 1 līdz 4”)	if (kubu_skaits < 1 || kubu_skaits > 4) { document.write(“Neatbilst nosacījumiem, kubu skaits blokā nav no 1 līdz 4”); }	if (kubu_skaits < 1 || kubu_skaits > 4) { Console.WriteLine(“Neatbilst nosacījumiem, kubu skaits blokā nav no 1 līdz 4”); }
def tilpums(self): return ((self.__kubu_skaits * self.malas_garums)**3)	tilpums() { return Math.pow(this.kubu_skaits * this.getMalas_garums(), 3); }	public int Tilpums() {return (int)Math.Pow(kubu_skaits * GetMalas_garums(), 3); }
kubg = Kubs(10, “zaļš”) kubr = Kubs(1, “sarkans”)	var kubg = new Kubs(10, “zaļš”); var kubr = new Kubs(1, “sarkans”);	Kubs kubg = new Kubs(10, “zaļš”); Kubs kubr = new Kubs(1, “sarkans”);
print(kubg.krasa, kubg.aprekinat_tilpumu()) print(“kubr malas garums”, kubr.malas_garums)	document.write(“kubg krāsa: “ + kubg.getKrasa() + “ un tilpums: “ + kubg.aprekinat_tilpumu() + “ ”); document.write(“kubr malas garums: “ + kubr.getMalas_garums() + “ ”);	Console.WriteLine(“kubg krāsa: “ + kubg.GetKrasa() + “ un tilpums: “ + kubg.AprekinatTilpumu()); Console.WriteLine(“kubr malas garums: “ + kubr.GetMalas_garums());
del(kubr)	kubr.dzest();	kubr.dzest();
print(“kubg malas garums”,kubg.malas_garums)	document.write(“kubg malas garums: “ + kubg.getMalas_garums() + “ ”);	Console.WriteLine(“kubg malas garums: “ + kubg.GetMalas_garums());
bloks1 = Bloks(5, “oranžs”, 3, 13) print(bloks1.nosaukums, bloks1.tilpums())	var bloks1 = new Bloks(5, “oranžs”, 3, “13”); document.write(bloks1.getNosaukums() + “ tilpums: “ + bloks1.tilpums() + “ cm^3 ”);	Bloks bloks1 = new Bloks(5, “oranžs”, 3, “13”); Console.WriteLine(bloks1.GetNosaukums() + “ tilpums: “ + bloks1.Tilpums() + “ cm^3”);
bloks2 = Bloks(7, “zils”, 5, 23) print(bloks2.nosaukums, bloks2.derigums)	var bloks2 = new Bloks(7, “zils”, 5, “23”); document.write(“bloks2 nosaukums: “ + bloks2.getNosaukums() + “ “ + bloks2.getDerigums());	Bloks bloks2 = new Bloks(7, “zils”, 5, “23”); Console.WriteLine(“bloks2 nosaukums: “ + bloks2.GetNosaukums() + “ “ + bloks2.GetDerigums());
bloks2.forma = 12 print(bloks2.nosaukums, bloks2.derigums)	bloks2.setForma(“12”); document.write(“bloks2 nosaukums: “ + bloks2.getNosaukums() + “ “ + bloks2.getDerigums());	bloks2.SetForma(“12”); Console.WriteLine(“bloks2 nosaukums: “ + bloks2.GetNosaukums() + “ “ + bloks2.GetDerigums());

C++	PHP	Java	GoLang
class Kubs {	class Kubs {	class Kubs {	type Kubs struct {
private: int malas_garums; string krasa;	private $malas_garums; private $krasa;	private int malas_garums; private String krasa;	malas_garums int krasa string

public:
Kubs(int malas_garums, string krasa) {
this->malas_garums = malas_garums;
this->krasa = krasa;
	public function __ construct($malas_garums, $krasa) {
	$this->malas_garums = $malas_garums;
	$this->krasa = $krasa;
		public Kubs(int malas_garums, String krasa) {
		this.malas_garums = malas_garums;
		this.krasa = krasa;
			func izveidotKubu(malas_garums int, krasa string) *Kubs {
			…
			return &Kubs{malas_garums: malas_garums, krasa: krasa}

if (malas_garums < 2 || malas_garums > 10) { this->malas_garums = 2; }	if ($malas_garums < 2 || $malas_garums > 10) { $this->malas_garums = 2; }	if (malas_garums < 2 || malas_garums > 10) { this.malas_garums = 2; }	if malas_garums < 2 || malas_garums > 10 { fmt.Println(“Nepareizs malas garums!”) malas_garums = 2 }

void dzest() { cout << “Izdzēsts “ << krasa << “ objekts” << endl;
delete this; }
	public function dzest() {
	echo “Izdzēsts “ . $this->krasa . “ objekts” . PHP_EOL;
	}
		public void dzest() { System.out.println(“Izdzēsts “+krasa+” objekts”); }	func iznicinatKubu(kubs *Kubs) {
			fmt.Println(kubs.krasa)
			kubs = nil }

int aprekinat_tilpumu() {
int tilpums = pow(malas_garums, 3);
return tilpums;}
	public function aprekinat_tilpumu() {
	$tilpums = pow($this->malas_garums, 3); return $tilpums; }
		public int aprekinat_tilpumu() {
		int tilpums = (int) Math.pow(malas_garums, 3);
		return tilpums;}
			func (k Kubs) aprekinat_tilpumu() int {
			return k.malas_garums * k.malas_garums * k.malas_garums }

class Bloks : public Kubs {
private: int kubu_skaits;
string forma;
int derigums;
string nosaukums;
public: Bloks(int malas_garums, string krasa, int kubu_skaits, string forma)
: Kubs(malas_garums, krasa) {
this->kubu_skaits = kubu_skaits;
this->forma = forma;
	public function __construct($malas_garums, $krasa, $kubu_skaits, $forma) {
	parent::__construct($malas_garums, $krasa);
	$this->kubu_skaits = $kubu_skaits;
	$this->forma = $forma;
		class Bloks extends Kubs {
		private int kubu_skaits;
		private String forma;
		private int derigums;
		private String nosaukums;
		public Bloks(int malas_garums, String krasa, int kubu_skaits, String forma) {
		super(malas_garums, krasa);
		this.kubu_skaits = kubu_skaits;
		this.forma = forma;
			type Bloks struct {
			Kubs
			kubu_skaits int
			Nosaukums string
			Forma BlokaForma
			Derigums int }
			...
			func izveidotBloku(malas_garums int, krasa string, kubu_skaits int, forma BlokaForma) *Bloks {

this->nosaukums = krasa + to_string(kubu_skaits);	$this->nosaukums = $krasa . $kubu_skaits;	this.nosaukums = krasa + kubu_skaits;	nosaukums := fmt.Sprintf(“%s%d”, krasa, kubu_skaits)

if (forma == “11” || forma == “12” || forma == “13” || forma == “14” || forma == “22”) {
this->derigums = 0; }
else {cout << “Bloks nederīgs” << endl;
this->derigums = 1;}
	if ($forma === “11” || $forma === “12” || $forma === “13” || $forma === “14” || $forma === “22”) {
	$this->derigums = 0; } else {
	echo “Bloks nederīgs” . PHP_EOL;
	$this->derigums = 1;}
		if (forma.equals(“11”) || forma.equals(“12”) || forma.equals(“13”) || forma.equals(“14”) || forma.equals(“22”)) {
		this.derigums = 0; } else {
		System.out.println(“Bloks nederīgs”);
		this.derigums = 1;}
			derigums := 0
			if !(forma == 11 || forma == 12 || forma == 13 || forma == 14 || forma == 22) {
			fmt.Println(“Nepareiza bloka forma!”)
			derigums = 1 }

if (kubu_skaits < 1 || kubu_skaits > 4) { cout << “Neatbilst nosacījumiem, kubu skaits blokā nav no 1 līdz 4” << endl; }	if ($kubu_skaits < 1 || $kubu_skaits > 4) { echo “Neatbilst nosacījumiem, kubu skaits blokā nav no 1 līdz 4” . PHP_EOL; }	if (kubu_skaits < 1 || kubu_skaits > 4) { System.out.println(“Neatbilst nosacījumiem, kubu skaits blokā nav no 1 līdz 4”); }	if kubu_skaits < 1 || kubu_skaits > 4 { fmt.Println(“Nepareizs kubu skaits!”) kubu_skaits = 1 }

int tilpums() { return pow(kubu_skaits * getMalas_garums(), 3); }	public function tilpums() { return pow($this->kubu_skaits * $this->getMalas_garums(), 3);}	public int tilpums() { return (int) Math.pow(kubu_skaits * getMalas_garums(), 3); }	func (b Bloks) tilpums() int { return ((b.kubu_skaits * b.malas_ garums) * (b.kubu_skaits * b.malas_garums) * (b.kubu_skaits * b.malas_garums)) }
Kubs* kubg = new Kubs(10, “zaļš”); Kubs* kubr = new Kubs(1, “sarkans”);	$kubg = new Kubs(10, “zaļš”); $kubr = new Kubs(1, “sarkans”);	Kubs kubg = new Kubs(10, “zaļš”); Kubs kubr = new Kubs(1, “sarkans”);	kubg := izveidotKubu(10, “zaļa”) kubr := izveidotKubu(1, “sarkana”)
cout << “kubg krāsa: “ << kubg- >getKrasa() << “ un tilpums: “ << kubg->aprekinat_tilpumu() << endl; cout << “kubr malas garums: “ << kubr->getMalas_garums() << endl;	echo “kubg krāsa: “ . $kubg- >getKrasa() . “ un tilpums: “ . $kubg->aprekinat_tilpumu() . PHP_EOL; echo “kubr malas garums: “ . $kubr->getMalas_garums() . PHP_ EOL;	System.out.println(“kubg krāsa: “+kubg.getKrasa() + “ un tilpums: “ + kubg.aprekinat_tilpumu()); System.out.println(“kubr malas garums: “+kubr.getMalas_ garums());	fmt.Println(kubg.krasa, kubg. aprekinat_tilpumu()) fmt.Println(kubr.malas_garums)
kubr->dzest();	$kubr->dzest();	kubr.dzest();	iznicinatKubu(kubr)
cout << “kubg malas garums: “ << kubg->getMalas_garums() << endl;	echo “kubg malas garums: “ . $kubg->getMalas_garums() . PHP_ EOL;	System.out.println(“kubg malas garums: “+kubg.getMalas_ garums());	fmt.Println(kubg.malas_garums)
Bloks* bloks1 = new Bloks(5, “oranžs”, 3, “13”); cout << bloks1->getNosaukums() << “ tilpums: “ << bloks1->tilpums() << “ cm^3” << endl;	$bloks1 = new Bloks(5, “oranžs”, 3, “13”); echo $bloks1->getNosaukums() . “ tilpums: “ . $bloks1->tilpums() . “ cm^3” . PHP_EOL;	Bloks bloks1 = new Bloks(5, “oranžs”, 3, “13”); System.out.println(bloks1. getNosaukums()+” tilpums: “+bloks1.tilpums()+” cm^3”);	bloks1 := izveidotBloku(5, “oranža”, 3, 13) fmt.Println(bloks1.Nosaukums, bloks1.tilpums())
Bloks* bloks2 = new Bloks(7, “zils”, 5, “23”); cout << “bloks2 nosaukums: “ << bloks2->getNosaukums() << “ “ << bloks2->getDerigums() << endl;	$bloks2 = new Bloks(7, “zils”, 5, “23”); echo “bloks2 nosaukums: “ . $bloks2->getNosaukums() . “ “ . $bloks2->getDerigums() . PHP_EOL;	Bloks bloks2 = new Bloks(7, “zils”, 5, “23”); System.out.println(“bloks2 nosaukums: “+bloks2. getNosaukums()+” “+bloks2. getDerigums());	bloks2 := izveidotBloku(7, “zila”, 5, 23) fmt.Println(bloks2.Nosaukums, bloks2.Derigums)
bloks2->setForma(“12”); cout << “nosaukums: “ << bloks2-> getNosaukums() << “ “ <<bloks2- >getDerigums();	$bloks2->setForma(“12”); echo “nosaukums: “ . $bloks2-> getNosaukums() . “ “ . $bloks2- >getDerigums();	bloks2.setForma(“12”); System.out.println(“bloks2 nosaukums: “+bloks2. getNosaukums()+” “+bloks2. getDerigums());	bloks2.nomainitFormu(12) fmt.Println(bloks2.Nosaukums, bloks2.Derigums)

4. DAĻA – DATU STRUKTŪRAS, PROGRAMMSASKARNE (API) (40 punkti (28*1,5))

1. uzdevuma vērtēšanas kritēriji

Kritēriji	Punkti
Izveidot izsaukumu uz doto resursu: 1 punkts – izveidots pieprasījums vai importēta bibliotēka, vai nosūtīts pieprasījums; 2 punkti – importēta pieprasījumu bibliotēka, izveidots un nosūtīts pieprasījums (atkarīgs no valodas).	2
Analizēt ārējā resursa atbildi, vai tas atbild ar korektu atbildi: 1 punkts – pārbaudīts, ka saņemta derīga vai nederīga atbilde; 2 punkti – pārbaudīti abi varianti - gan derīga, gan nederīga atbilde.	2
Izvadīt atbilstošu tekstu, ja serveris neatbild: 1 punkts – ir izvadīts kaut kas (piemēram, “Kļūda”), nepaskaidrojot, kāda kļūda; 2 punkti – ir izvadīts teksts, ka serveris neatbild (vai izmantots resursā pieejamais kļūdas ziņojums).	2
Izvadīt atbilstošu tekstu, ja serveris atbild ar tukšu (tukšu masīvu) atbildi.	1
Izvadīt attiecīgā atkritumu nodošanas punkta adresi, kurā var nodot baterijas un akumulatorus.	1
Izvadīt attiecīgā atkritumu nodošanas punkta novada nosaukumu, kurā var nodot baterijas un akumulatorus.	1
Izvadīt attiecīgā atkritumu nodošanas punkta adresi, kurā var nodot nolietotās riepas.	1
Izvadīt attiecīgā atkritumu nodošanas punkta novada nosaukumu, kurā var nodot nolietotās riepas.	1
Izvadīt attiecīgā atkritumu nodošanas punkta adresi, kurā var nodot metālu.	1
Izvadīt attiecīgā atkritumu nodošanas punkta novada nosaukumu, kurā var nodot metālu.	1
Kopā:	**13**

1. uzdevuma risinājumi dažādās programmēšanas valodās

JavaScript	Python
request.open('GET', 'https://data.gov.lv/dati/lv/api/3/action/datastore_search?resource_id=92ac6e57-c5a5-444e-aaca-ae90c120cc3d', true)	pieprasijums = requests.get("https://data.gov.lv/dati/lv/api/3/action/datastore_search?resource_id=92ac6e57-c5a5-444e-aaca-ae90c120cc3d")

if (request.status < 200 || request.status >= 400) { // Ja serveris neatbild
...
} else { // Ja serveris atbild
...
}
	if (pieprasijums.status_code >= 400): # Ja serveris neatbild
	...
	else: # Ja serveris atbild ar derīgu atbildi

console.log("Serveris neatbild!")	print(teksts["error"]["message"]) VAI print("Serveris neatbild")

if (data == null) {
console.log("Nav datu")
}
	if len(teksts) == 0: # Ja atbild ar tukšu atbildi
	print("Nav datu")

if (punkts["8 : Baterijas un akumulatori"] == "x") {
console.log(punkts.adrese)
}
	if (record["8 : Baterijas un akumulatori"] == "x"):
	print(record["adrese"])

if (punkts["8 : Baterijas un akumulatori"] == "x") {
console.log(punkts.pilsetanovads)
}
	if (record["8 : Baterijas un akumulatori"] == "x"):
	print(record["pilsetanovads"])

if (punkts["10 : Nolietotās riepas"] == "x") {
console.log(punkts.adrese)
}
	if (record["10 : Nolietotās riepas"] == "x"):
	print(record["adrese"])

if (punkts["10 : Nolietotās riepas"] == "x"] == "x") {
console.log(punkts.pilsetanovads)
}
	if (record["10 : Nolietotās riepas"] == "x"):
	print(record["pilsetanovads"])

if (punkts["3 : Metāls"] == "x"] == "x") {
console.log(punkts.adrese)
}
	if (record["3 : Metāls"] == "x"] == "x"):
	print(record["adrese"])

if (punkts["3 : Metāls"] == "x"] == "x") {
console.log(punkts.pilsetanovads)
}
	if (record["3 : Metāls"] == "x"] == "x"):
	print(record["pilsetanovads"])

Java	C#	GoLang
String url = “https://data.gov.lv/dati/lv/api/3/action/datastore_search?resource_id=92ac6e57-c5a5-444e-aaca-ae90c120cc3d”; URL apiUrl = new URL(url); HttpURLConnection connection = (HttpURLConnection) apiUrl.openConnection(); connection.setRequestMethod(“GET”);	string url = “https://data.gov.lv/dati/lv/api/3/action/datastore_search?resource_id=92ac6e57-c5a5-444e-aaca-ae90c120cc3d”; using (HttpClient client = new HttpClient()) { HttpResponseMessage response = await client.GetAsync(url);	apiURL := “https://data.gov.lv/dati/lv/api/3/action/datastore_search?resource_id=92ac6e57-c5a5-444e-aaca-ae90c120cc3d” request, err := http.NewRequest(“GET”, apiURL, nil)

int responseCode = connection.getResponseCode();
if (responseCode == 408) { // Ja serveris neatbild
...
} else if (responseCode == 200) { // Ja serveris atbild
...
}
	if (statusCode == 408) // Ja serveris neatbild
	{
	...
	}
	else if (statusCode == 200) // Ja serveris atbild ar derīgu atbildi
	{
	...
	}
		if response.StatusCode == http.StatusOK { // Ja serveris atbild ar derīgu atbildi
		...
		} else if response.StatusCode == http.StatusNotFound {
		// 404 (Not Found) koda pārbaude
		...
		} else { // Citi kļūdu kodi
		...
		}

System.out.println(“Savienojuma noildze”);	Console.WriteLine(“Savienojuma noildze”);	fmt.Println(“Resurss neeksistē”)

if (teksts.length()<1) {
System.out.println(“Nav datu”);
}
	if (teksts.length()<1)
	{
	Console.WriteLine(“Nav datu”);
	}
		if len(records) == 0 {
		fmt.Println(“Tika atgriezts tukšs masīvs”)
		}

if (record.getString(“8 : Baterijas un akumulatori”).equals(“x”)) {
filteredRecord.put(“adrese”, record.getString(“adrese”));
}
	if (record[“8 : Baterijas un akumulatori”].ToString() == “x”)
	{
	filteredRecord[“adrese”] = record[“adrese”].ToString();
	}
		if len(punkti5un6) == 0 {
		fmt.Println(“Nekas netika atrasts”)
		} else {
		fmt.Println(“Tika atrastas”, len(punkti5un6), “vietas:”)
		for _, item := range punkti5un6 {
		fmt.Println(“Adrese:”, item.Adrese)
		}
		}

if (record.getString(“8 : Baterijas un akumulatori”).equals(“x”)) {
filteredRecord.put(“pilsetnovads”, record.getString(“pilsetanovads”));
}
	if (record[“8 : Baterijas un akumulatori”].ToString() == “x”)
	{
	filteredRecord[“pilsetnovads”] = record[“pilsetanovads”].ToString();
	}
		if len(punkti5un6) == 0 {
		fmt.Println(“Nekas netika atrasts”)
		} else {
		fmt.Println(“Tika atrastas”, len(punkti5un6), “vietas:”)
		for _, item := range punkti5un6 {
		fmt.Println(“Novada nosaukums:”, item.PilsetaNovads)
		}
		}

if (record.getString(“10 : Nolietotās riepas”).equals(“x”)) {
filteredRecord.put(“adrese”, record.getString(“adrese”));
}
	if (record[“10 : Nolietotās riepas”].ToString() == “x”)
	{
	filteredRecord[“adrese”] = record[“adrese”].ToString();
	}
		if len(punkti7un8) == 0 {
		fmt.Println(“Nekas netika atrasts”)
		} else {
		fmt.Println(“Tika atrastas”, len(punkti5un6), “vietas:”)
		for _, item := range punkti5un6 {
		fmt.Println(“Adrese:”, item.Adrese)
		}
		}

if (record.getString(“10 : Nolietotās riepas”).equals(“x”)) {
    filteredRecord.put(“pilsetnovads”, record.getString(“pilsetanovads”));
}
	if (record[“10 : Nolietotās riepas”].ToString() == “x”)
	{
	      filteredRecord[“pilsetnovads”] = record[“pilsetanovads”].ToString();
	}
		if len(punkti7un8) == 0 {
		   fmt.Println(“Nekas netika atrasts”)
		} else {
		   fmt.Println(“Tika atrastas”, len(punkti5un6), “vietas:”)
		   for _, item := range punkti5un6 {
		      fmt.Println(“Novada nosaukums:”, item.PilsetaNovads)
		   }
		}

if (record.getString(“3 : Metāls”).equals(“x”)) {
    filteredRecord.put(“adrese”, record.getString(“adrese”));
}
	if (record[“3 : Metāls”].ToString() == “x”)
	{
	     filteredRecord[“adrese”] = record[“adrese”].ToString();
	}
		if len(punkti9un10) == 0 {
		   fmt.Println(“Nekas netika atrasts”)
		} else {
		   fmt.Println(“Tika atrastas”, len(punkti5un6), “vietas:”)
		   for _, item := range punkti5un6 {
		      fmt.Println(“Adrese:”, item.Adrese)
		   }
		}

if (record.getString(“3 : Metāls”).equals(“x”)) {
    filteredRecord.put(“pilsetnovads”, record.getString(“pilsetanovads”));
}
	if (record[“3 : Metāls”].ToString() == “x”)
	{
	      filteredRecord[“pilsetnovads”] = record[“pilsetanovads”].ToString();
	}
		if len(punkti5un6) == 0 {
		   fmt.Println(“Nekas netika atrasts”)
		} else {
		   fmt.Println(“Tika atrastas”, len(punkti5un6), “vietas:”)
		   for _, item := range punkti5un6 {
		      fmt.Println(“Novada nosaukums:”, item.PilsetaNovads)
		   }
		}

2. uzdevuma vērtēšanas kritēriji

Kritēriji	Punkti
Nodrošināt vārda ievadīšanu.	1
Pārbaudīt lietotāja ievadīto vērtību, vai tas ir vārds.	1
Katru reizi ievadīt tikai vienu vārdu, nevis vārdu virkni.	1
Pārbaudīt, vai vārda pirmais burts ir lielais burts.	1
Atkārtoti ievadīt vērtību, ja ievadītā vērtība bijusi kļūdaina.	1
Izveidot datu struktūru, kurā uzglabāt latviešu alfabētu un tam atbilstošā sākumburta pozīcijas kārtas numuru.	1
Izveidot datu struktūru, kurā glabāt jauno izveidoto sarakstu.	1
Noteikt ievadītā vārda pirmo burtu.	1
Ja vārda pozīcija sarakstā jau ir aizņemta ar kādu vārdu, tad aizvietot ar ievadīto vārdu.	1
Ja vārda pozīcijas sarakstā vārda nav, tad tajā tiek ievietots ievadītais vārds.	1
Pārbaudītais vārds tiek ievietots alfabēta secības atbilstošajā pozīcijā, ņemot vērā tā pirmo burtu.	1
Izdrukāt katru programmas darbību kā lasāmu teikumu.	1
Izvadīt kļūdas paziņojumu, ja ievadītais vārds neatbilst noteikumiem.	1
Datu struktūrā nedrīkst būt nulltais elements.	1
Programmas darbojas, līdz viss saraksts ir aizpildīts.	1
Kopā:	**15**

2. uzdevuma risinājumi dažādās programmēšanas valodās

JavaScript	Python
`let vards = prompt("Ievadi vārdu latviešu valodā, kas sākas ar lielo burtu:")`	`vards = input("Ievadiet vārdu: ")`
`if (/^[\p{L}]+$/u.test(vards) == True) // Bez Regex nedarbojas ar unikoda simboliem`	`if not vards.isalpha(): // Darbojas arī ar unikoda simboliem`

`// Uzdevumā pēc būtības nav teikts, ka tas ir jāpārbauda, bet šādai vajadzētu būt pārbaudei (ja satur atstarpi, ir vairāki vārdi)`
`if (vards.includes(" ")) {`
`  console.log("Jāievada tikai 1 vārds!")`
`}`
	`if not vards.isalpha(): // Pārbaudīs arī uz vārdu skaitu, jo atstarpe nav alfabēta simbols`

`if (vards.charAt(0) == vards.charAt(0).toUpperCase())`	`if not vards[0].isupper()`
Nodrošina cikls, kas izpildās, kamēr viss saraksts ir aizpildīts - ja ir kāda kļūda, pārlec uz nākamo iterāciju.	Nodrošina cikls, kas izpildās, kamēr viss saraksts ir aizpildīts - ja ir kāda kļūda, pārlec uz nākamo iterāciju.
`let alfabetsTXT = "AĀBCČDEĒFGĢHIĪJKĶLĻMNŅOPRSŠTUŪVZŽ"` `const alfabets = alfabetsTXT.split("")`	`alfabets = {'A': 1, 'Ā': 2, 'B': 3, 'C': 4, 'Č': 5, 'D': 6, 'E': 7, 'Ē': 8, 'F': 9, 'G': 10, 'Ģ': 11, 'H': 12, 'I': 13, 'Ī': 14, 'J': 15, 'K': 16, 'Ķ': 17, 'L': 18, 'Ļ': 19, 'M': 20, 'N': 21, 'Ņ': 22, 'O': 23, 'P': 24, 'R': 25, 'S': 26, 'Š': 27, 'T': 28, 'U': 29, 'Ū': 30, 'V': 31, 'Z': 32, 'Ž': 33}`
	`alfabets = "AĀBCČDEĒFGĢHIĪJKĶLĻMNŅOPRSŠTUŪVZŽ"` `alfabets = list(alfabets)`
`let vardi = ["-"]; // Nodrošina, ka 0. elements "neeksistēs"`	`saraksts = [[] for i in range(33)]`
	`saraksts = []`
`burts = vards.charAt(0)`	`burts = vards[0]`

`if (vardi[poz] != undefined) {`
`  console.log("Vārds " + vardi[poz] + " tiks aizvietots ar " + vards)`
`  vardi[poz] = vards`
`}`
	`if saraksts[pozicija]:`
	`  vecais_vards = saraksts[pozicija][0]`
	`  saraksts[pozicija] = [vards]`

	`if saraksts[pozicija] == “”:`
	`  vecais_vards = saraksts[pozicija]`
	`  saraksts[pozicija] = vards`

`poz = alfabets.indexOf(pirmais_burts)+1` `vardi[poz] = vards`	`else:` `  saraksts[pozicija].append(vards)`
	`else:` `  saraksts[pozicija] = vards`
`poz = alfabets.indexOf(pirmais_burts)+1` `vardi[poz] = vards`	`saraksts[pozicija].append(vards)`
`console.log(“Vārds “ + vardi[poz] + “ tiks aizvietots ar “ + vards + “ “ + poz + “. pozīcijā”)`	`print(f”Vārds ‘{vecais_vards}’ tika aizstāts ar ‘{vards}’”)`
`console.log(“Pievienots vārds “ + vards + “, pozīcija: “ + poz.toString())`	`print(f”Pievienoju vārdu ‘{vards}’ {pozicija}. vietā”)`
`console.log(“Ievadītais vārds nesākas ar lielo burtu vai satur vēl citus simbolus”)`	`print(“Kļūda: Ievadītajam vārdam jāsatur tikai burti!”)`
JavaScript masīvam 0. elementu nevar atstāt tukšu (tehniski būs undefined, bet elements eksistēs).	Atkarīgs no datu struktūras - dictionary var izdarīt, list nevar izveidot bez 0. elementa (var neaizpildīt, bet elements kā tāds eksistēs).

`while ((vardi.length < 34) || (vardi.includes(undefined)))`	`if all(saraksts):`
	`  print(“Visi vārdi ir ievietoti atbilstoši latviešu alfabētam!”)`
	`  break`

	`if not “” in saraksts:`
	`  print(“Visi vārdi ir ievietoti atbilstoši latviešu alfabētam!”)`
	`  break`

Java	C#	GoLang

`String input = scanner.nextLine();`	`string input = Console.ReadLine();`	`fmt.Println(“Ievadiet vārdu:”)`
		`scanner := bufio.NewScanner(os.Stdin)`
		`if scanner.Scan()`
		`{`
		`vards = scanner.Text()`
		`fmt.Printf(“Jūs ievadījāt: %q\n”, vards)`
		`...`

`if (vards.matches(“[A-ZĀ-Ž].*”) &&`
`new String(alfabets).contains(String.`
`valueOf(vards.charAt(0))))`
	`if (string.IsNullOrWhiteSpace(word))`
	`{`
	`  ...`
	`}`
		`func parbauditVardu(vards string) bool {`
		`for _, burts := range vards {`
		`if !unicode.IsLetter(burts) {`
		`return false`
		`}`
		`}`
		`return true`
		`}`
		`...`
		`if !parbauditVardu(vards) {`
		`fmt.Println(“Kļūda: Ievadītajam vārdam`
		`jāsatur tikai burti!”)`
		`}`
		`...`

`if (input.trim().isEmpty()) {`
`  System.out.println(“Ievadīta tukša`
`rinda. Ievade apstādināta.”);`
`  return;`
`}`
	`if (string.IsNullOrWhiteSpace(word))`
	`{`
	`  Console.WriteLine(“Ievadīta tukša`
	`rinda. Ievade apstādināta.”);`
	`  return;`
	`}`
		Nodrošināts, ka ievadē tiek ņemta vērā vesela rinda, kas var būt ar vairākiem vārdiem, bet programmas darbības neturpināsies - tiks izvadīts paziņojums, ka ir neatļautie simboli, jo ir atstarpe starp vārdiem

`if (vards.matches(“[A-ZĀ-Ž].*”) &&`
`new String(alfabets).contains(String.`
`valueOf(vards.charAt(0))))`
	`if (word.Length > 0 && alphabet.`
	`Contains(word[0]))`
	`{`
	`  ...`
	`}`
		`else if !unicode.IsUpper(pirmaisBurts)`
		`{`
		`fmt.Println(“Kļūda: Vārdam jāsākas ar`
		`lielo burtu!”)`
		`}`
		`...`

Nodrošina cikls, kas izpildās, kamēr viss saraksts ir aizpildīts - ja ir kāda kļūda, pārlec uz nākamo iterāciju.	Nodrošina cikls, kas izpildās, kamēr viss saraksts ir aizpildīts - ja ir kāda kļūda, pārlec uz nākamo iterāciju.	Nodrošina cikls, kas izpildās, kamēr viss saraksts ir aizpildīts - ja ir kāda kļūda, pārlec uz nākamo iterāciju.
String alfabetsTXT = "AĀBCČDEĒFGĢHIĪJKĶLĻMNŅOPRSŠTUŪVZŽ"; char[] alfabets = alfabetsTXT. toCharArray();	string alphabet = "ABCDEFGHIJKLMNOPRSTUVZ";	var ALFABETS = map[rune]int{'A': 1, 'Ā': 2, 'B': 3, 'C': 4, 'Č': 5, 'D': 6, 'E': 7, 'Ē': 8, 'F': 9, 'G': 10, 'Ģ': 11, 'H': 12, 'I': 13, 'Ī': 14, 'J': 15, 'K': 16, 'Ķ': 17, 'L': 18, 'Ļ': 19, 'M': 20, 'N': 21, 'Ņ': 22, 'O': 23, 'P': 24, 'R': 25, 'S': 26, 'Š': 27, 'T': 28, 'U': 29, 'Ū': 30, 'V': 31, 'Z': 32, 'Ž': 33}
Map<Integer, String> vardi = new HashMap<>();	Dictionary<char, List<string>> words = new Dictionary<char, List<string>>();	var saraksts [34]string
char pirmais_burts = vards.charAt(0);	char firstLetter = word. Normalize(NormalizationForm.FormC)[0];	pirmaisBurts, _ := utf8. DecodeRuneInString(vards) fmt.Printf("Ievadītā vārda pirmais burts: %#U\n", pirmaisBurts)
int nr = new String(alfabets). indexOf(pirmais_burts) + 1; vardi.put(nr, vards);	if (words.ContainsKey(alphabet[index - 1])) { List<string> wordList = words[alphabet[index - 1]]; wordList.RemoveAll(w => w.StartsWith(firstLetter.ToString())); } words[alphabet[index - 1]].Add(word);	if saraksts[pozicija] != "" { vardsVecais := saraksts[pozicija] saraksts[pozicija] = vardsVecais fmt.Printf("Vārds %q tika aizvietots ar vārdu %q\n", vardsVecais, vards) } ...
int nr = new String(alfabets). indexOf(pirmais_burts) + 1; vardi.put(nr, vards);	words[alphabet[index - 1]].Add(word);	... else { saraksts[pozicija] = vards fmt.Printf("Vārds %q tika ievietots %d. pozīcijā\n", vards, pozicija) }

`vardi.put(nr, vards);`	`words[alphabet[index - 1]].Add(word);`	`saraksts[pozicija] = vards`
`System.out.print("Raksti latviešu vārdu, sāc ar lielo burtu: ");`	`Console.Write("Ievadiet latviešu vārdu, sāciet ar lielo burtu (lai pārtrauktu, atstājiet tukšu ievadi): ");`	Ir vairāki piemēri augstāk.
`System.out.println("Pievienoju " + vards + " ar kārtas numuru " + nr);`	`Console.WriteLine("Pievienoju vārdu \"" + word + "\" " + index + ". pozīcijā.");`	
`System.out.println("Ievadītais vārds nesākas ar lielo burtu vai satur vēl citus simbolus");`	`Console.WriteLine("Ievadītais vārds nesākas ar lielo burtu vai satur vēl citus simbolus.");`	Ir vairāki piemēri augstāk.

`for (int sk = 1; sk <= 33; sk++) {`
`  vardi.put(sk, "");`
`}`
	Izmantojot vārdnīcu (dictionary), 0. elementa nebūs.	Masīvu nevar izveidot bez 0. elementa (var neaizpildīt vai aizpildīt ar kādu noklusēto *dummy* vērtību, bet elements kā tāds vienmēr eksistēs).

`int cik_pilns = 0;`
`while (cik_pilns < alfabetsTXT.length())`
`{`
`  ...`
`}`
	`int filledCount = 0;`
	`while (filledCount < alphabet.Length)`
	`{`
	`  ...`
	`}`
		`func parbauditSarakstu() bool {`
		`for _, vards := range saraksts {`
		`if vards == "" {`
		`return false`
		`}`
		`}`
		`return true`
		`}`
		`...`
		`if parbauditSarakstu() {`
		`fmt.Println("Saraksts ir aizpildīts, un visi ievadītie vārdi ir ievietoti atbilstoši latviešu valodas alfabētam!")`
		`break`
		`}`