Latvijas Republikas izglītības un zinātnes ministrija Daugavpils Tehnoloģiju un tūrisma tehnikums

PROJEKTA DARBS

Programmēšanas tehniķis

specialitāte

1. projekta darbs

temats

Testēšanas sistēmas izveidošana

PASKAIDROJOŠAIS RAKSTS

**DTTT.P.D.0009.009PR**

Profesijas kods 33 481 031 Programmēšanas nodaļa

PR-21 grupa

|  |  |  |
| --- | --- | --- |
| Izveidoja | paraksts | R.Giruckis  V.,Uzvārds |
| Pārbaudīja | paraksts | I. Dortiņa  V.,Uzvārds |

2024./2025. m.g.

**Saturs**

[Ievads 3](#_Toc193962522)

[1. Uzdevuma nostādne 3](#_Toc193962523)

[1.1. Sistēmas modelis 3](#_Toc193962524)

[1.1.1. Priekšmetiskās jomas informācijas modelis 3](#_Toc193962525)

[1.1.2. Sistēmas apkārtnes shēma 4](#_Toc193962526)

[1.1.3. Viedokļu analīze 4](#_Toc193962527)

[1.1.4. Datu modelēšana 6](#_Toc193962528)

[1.1.5. Datu vārdnīca 7](#_Toc193962529)

[1.2. Sistēmas evolūcija 8](#_Toc193962530)

[1.3. Sistēmas funkcionālās prasības 8](#_Toc193962531)

[1.4. Sistēmas nefunkcionālās prasības 12](#_Toc193962532)

[2. Priekšmetiskās jomas klašu sistēmas izstrāde 15](#_Toc193962533)

[2.1. Klašu diagramma 16](#_Toc193962534)

[2.2. Klašu realizēšana Java valodā 17](#_Toc193962535)

[3. Testa programmas izstrāde 19](#_Toc193962536)

[3.1. Testēšanas metodikas 19](#_Toc193962537)

[3.2. Testēšanas programmas struktūra 19](#_Toc193962538)

[3.3. Testēšanas rezultāti 20](#_Toc193962539)

[4. Lietotāja rokasgrāmata 21](#_Toc193962540)

[4.1. Lietotāja instrukcija 21](#_Toc193962541)

[4.2. Palīdzības sistēma (Help) 21](#_Toc193962542)

[4.3. Sistēmas ziņojumi 21](#_Toc193962543)

[Secinājumi 21](#_Toc193962544)

[Izmantotās literatūras (informācijas avotu) saraksts 23](#_Toc193962545)

[Pielikums A. Klašu sistēma 24](#_Toc193962546)

[Pielikums B. testa programmas kods 25](#_Toc193962547)

Ievads

Projekta darbā ir aprakstīta testēšanas programma, kura izpilda sekojošas funkcijas:

* lietotāju reģistrēšana
* datu ievade
* datu saglabāšana
* datu rediģēšana
* testa pildīšana
* rezultātu aprēķināšana

Sistēma nodrošina audzēkņa testēšanu par noteiktu tēmu/vielu, tas ir domāts, lai skolotājs varētu viegli izlikt atzīmi skolēnam digitālā veidā attālināti vai klātienē. Tāpat šī programma novērš cilvēcisko faktoru, piemēram, noapaļošanu noteiktā virzienā.

# Uzdevuma nostādne

## Sistēmas modelis

### Priekšmetiskās jomas informācijas modelis

Apskatāmā sistēmā tika izdalīti objekti: Datu bāze (Fails ar atbildēm), Tests, Rezultāti, Administrators / Skolotājs, Lietotājs. Šie objekti un saites starp tiem ir paradīti ([1.1. att.](#att1_1)).

Datu bāze

Administrātors/Skolotājs

Tests

Rezultāti

Lietotājs

Treniņš

1.1 att. Vienkāršots testēšanas sistēmas modelis

Saišu nozīme vienkāršotā nodaļas modelī:

* Skolotājs ir testa veidotājs un pārbaudītājs, kuram ir pieeja pie atbildēm, testam un lietotāja rezultātiem;
* Datu bāze satur failus ar jautājumiem un ar pariezām atbildēm;
* Lietotājs ir testa pildītājs, kas beigās var apskatīt savus rezultātus, nepieciešamības gadījumā viņš varēs sazināties ar skolotāju;
* Tests ir instruments, kas palīdzēs pārbaudīt lietotāju zināšanas un novērtēt to ar atzīmi;
* Rezultāti parāda atzīmi, pareizās un nepareizās atbildes skaitliski un procentuāli.

### Sistēmas apkārtnes shēma

Dati, kā atbildes, ko ievada audzēknis tiek saglabāti failā. Tomēr dažreiz atbildes netiek automātiski novērtētas un skolotājam ir iespēja manuāli novērtēt iesniegto atbildi. Saskarni ar lietotājiem nodrošinās interfeisa modulis ([1.2. att.](#att1_2)).

Lietotājs

Lietotāja saskarne

Sistēma

Datu bāze

Lietotāja dati un rezultāti

1.2. att. Sistēmas apkārtnes modeļa piemērs

### Viedokļu analīze

Savāktie un identificētie viedokļi par sistēmu ir attēloti ar burbuļdiagrammas palīdzību ([1.3. att.](#att1_3)), tas faktiski ir viedokļu kopums, kur katra viedokļa nosaukums ir ierakstīts atsevišķa elipsē.

1.3. att. Viedokļu burbuļu diagramma

Nefunkcionalie viedokli

Funkcionalie viedokli

Viedokli par lietotaju

Datu viedokļi

Otrais etaps viedokļu analīzē ir viedokļu klasifikācija, kad viedokļi ir sagrupēti grupās ([1.4. att.](#att1_4)).

Testēšanas sistēma (V0)

Datu bāze (V12)

Lietotājs (V11)

Tests (V13)

Lietotāju registrt (V121)

Skolotājs (V111)

Testa jautajumu registrs (V122)

Audzēknis (V112)

Audzekņa funkcijas (V132)

Skolotāja funkcijas (V131)

Testa pildīšana (V1321)

Testa izveide (V1311)

Testa rezultātu saņemšana (V1323)

Testa manuāla novērtēšana (V1312)

Treniņa manuāla izveidošana

(V1313)

Treniņā

pildīšana (V1322)

Treniņa jautajumus registrs (V123)

1.4. att. Viedokļu hierarhiskā struktūra

### Datu modelēšana

Datu modelēšanai ir tiek izmantots ER modelis. Testēšanas sistēmas ER modelis ([1.5. att.](#att1_5)) ietver datu entītiju kopumu un relāciju kopumu starp datu entītijām, pie kam ar šo modeli var attēlot dažāda tipa relācijas starp datu elementiem.

Audzēknis

Pilda

Tests

Skolotājs

Izveido

Jautājums

Satur

Tests

1.5. att. Vienkārsots testēšanas sistēmas ER modelis

### 

### Datu vārdnīca

|  |  |  |
| --- | --- | --- |
| **Datu tips** | **Nosaukums** | **Apraksts** |
| **String** | **name** | Audzēknis ievada savu vārdu un uzvārdu |
| **String** | **questionText** | Neliels teksta apgabals, kas apraksta jautājumu |
| **String[]** | **answersText** | Atbilžu variantu masīvs. Tiek piedāvāti vairāki atbilžu  varianti, kur ir jāizvēlas viens pareizs |
| **int** | **correctAnswer** | Aile, kas nosaka, vai tika izvēlēts pareizs atbildes variants |
| **float** | **result** | Audzēkņa saņemto punktu skaits (0-100) |
| **int** | **mark** | Audzēkņa atzīme (0-10) |

## Sistēmas evolūcija

Testēšanas sistēma ir neatkarīga un pašpietiekama programma. Produktam nav paredzētas vairākas sastāvdaļas.

Programmai ir trīs ārējās saskarnes, kuras var apskatīt 1.4. nodaļā – viena paredzēta reģistrēšanai sistēmā, otrā – ielogošanai sistēmā un trešā – testa pildīšanai. Grafisko lietotāja saskarni var pilnveidot, papildinot to ar dažādām krasu un noformējuma shēmām.

Testēšanas sistēmā paredzēti divu lietotāju veidi: administrators un lietotājs, kurš pilda testu. Nākotnē plānots pievienot vēl vienu lietotāju grupu ar funkciju veidot jaunus testus un pārbaudīt izpildītos testus, līdz ar to lietotājiem būs divas lomas: audzēknis, kurš pilda testu un skolotājs, kurš veido un pārbauda testus. Administratoram plānots realizēt funkciju mainīt lietotāja vārdu, uzvārdu, lietotājvārdu vai paroli, ka arī pievienot un dzēst lietotājus un mainīt vīnu lomas.

Programma satur vienu testu, kurā ir 10 jautājumi, tomēr nākotnē sistēmu var pilnveidot, pievienojot jaunas testa tēmas, kļūdu uzrādīšanu un kļūdu skaidrojumu pēc testa izpildes, lietotāju reitingu, lietotāju datu izvadi lietotāja izvēlnē vai administratora izvēlnē.

Sistēmas dažiem datu tipiem tiks izmantoti dati, kuri tiek saglabāti teksta failā, nākotnē programmu var pilnveidot darbam ar datu bāzi (datu bāzes savienojums, testa jautājumi un atbildes uz tiem u.c.)

## Sistēmas funkcionālās prasības

1. **Funkcija “Reģistrēties sistēmā”**

***ID: T1***

**Ievads**: Ļauj ievadīt nepieciešamus datus, lai izveidotu savu kontu. Visi ievadīti dati tiek saglabāti failā (datu bāzē).

Ievade:

1. Lietotāja vārds
2. Lietotāja uzvārds
3. Lietotāja lietotājvārds
4. Lietotāja parole
5. Ievadītās paroles apstiprināšana.

**Apstrāde**:

1. Lietotāja vārda saglabāšana failā.
2. Lietotāja uzvārda saglabāšana failā.
3. Lietotāja lietotājvārda saglabāšana failā.
4. Lietotāja paroles saglabāšana failā.

Ja ievadītā parole ir nepareiza, tad tiek atvērts kļūdas dialoglodziņš.

**Izvade**: Dialoglodziņš, kurā uzrakstīts “Reģistrēšana ir veiksmīgi pabeigta!”.

1. **Funkcija “Ielogoties sistēmā”**

***ID: T2***

**Ievads**: Ļauj ievadīt nepieciešamus datus, lai ieiet sava kontā.

**Ievade**:

1. Lietotāja lietotājvārds
2. Lietotāja parole

**Apstrāde**:

1. Lietotāja lietotājvārda pārbaude un atrašana DB.
2. Lietotāja paroles pārbaude un atrašana DB.

Ja ievadītā parole vai/un lietotājvārds ir nepareiza, tad tiek atvērts kļūdas dialoglodziņš.

**Izvade**: Dialoglodziņš, kurā uzrakstīts “Ieeja ir veiksmīgi pabeigta!”.

1. **Funkcija “Sākt testu”**

***ID: T3***

**Ievads**: Ļauj administratoram sākt testu.

**Ievade**: Administrators noklikšķina uz pogu “Sākt testu”. Pēc tam sistēma uzdod jautājumu: "Vai Jūs vēlaties sākt testu?" Administratoram ir divi varianti atbildēm "Jā" vai "Nē".

**Apstrāde**: Sistēma pārbauda atbildi uz uzdoto jautājumu un pēc tam sāk testu, ja uz uzdoto jautājumu atbilde "Jā". Tad tests ir pieejams lietotājiem to veikšanai.

**Izvade**: Administratora ekrānā tiek parādīts dialoglodziņš ar tekstu "Tests ir atļauts!"

1. **Funkcija “Pildīt testu”**

**I*D: T4***

**Ievads**: Ļauj lietotājam sākt testa izpildi.

**Ievade**: Lietotājs noklikšķina uz pogu “Pildīt testu”.

**Apstrāde**: Ja tests ir pieejams izpildei, tad pēc testa izvēles tiek parādīts dialoglodziņš ar tekstu "Tests ir pieejams! Vai vēlaties izpildīt testu?". Lietotājam ir divi varianti atbildēm "Jā" vai "Nē". Ja tests nav pieejams izpildei, tad pēc testa izvēles tiek parādīts dialoglodziņš ar tekstu "Tests nav pieejams!"

**Izvade**: Lietotājā ekrānā tiek parādīts dialoglodziņš ar uzrakstu "Veiksmi testā!"

1. **Funkcija “Atbildēt uz testa jautājumu”**

***ID: T5***

**Ievads**: Ļauj ievadīt atbildi uz testa jautājumus.

**Ievade**: Lietotājs izvēlas 1 no 4 piedāvātajām atbildēm uz uzdoto jautājumu.

**Apstrāde**: Sistēma pārbauda atbildi uz jautājumu. Pēc katras pareizās atbildes sistēma pievieno 1.

**Izvade**: Pēc izvēlētās atbildes lietotājs noklikšķina uz pogas "Nākamais", pēc kura tiek parādīts nākamā izvēlne ar testa jautājumiem.

1. **Funkcija “Parādīt rezultātu”**

***ID: T6***

**Ievads**: Ļauj uzzināt nokārtotā testa rezultātus.

**Ievade**: Lietotājs noklikšķina uz pogas "Parādīt rezultātu".

**Apstrāde**: Sistēma saskaita pareizo atbilžu skaitu, nepareizo atbilžu skaitu un neatbildēto jautājuma skaitu, tad aprēķina pareizo atbilžu procentuālo vērtību, pēc tam nolasa vērtējumu par 10 balles skalu un atrod komentāru par saņemto atzīmi.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Neatbildēto jautājuma skaits.
2. Pareizo atbilžu skaits.
3. Nepareizo atbilžu skaits.
4. Pareizo atbilžu procentuālā daļa.
5. Atzīme.
6. Sistēmas komentārs par saņemto atzīmi.
7. **Funkcija “Salidzini Rezultatie ar Videjie”**

***ID: T7*** **Ievads**: Ļauj salīdzināt skolēna rezultātus ar vidējiem klase

**Ievade**: Lietotājs noklikšķina uz pogas " Salīdzini rezultatie ar vidējie klase".

**Apstrāde**: Sistēma nosaka vidējo atzīme un salīdzini tos ar skolēnu atzīme.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Vidēja atzīme/punktus/procenti klase.
2. Skolēnu atzīme/punktus/procenti.
3. Vai skolēna atzīme/punktus/procenti ir mazāka vai lielāka nekā vidēja.
4. **Funkcija “Iespēja apskatīt pareizās atbildes”**

***ID: T8*** **Ievads**: Iespēja redzēt pareizās atbildes tikai pēc tam, kad skolotājs ir devis atļauju to darīt.

**Ievade**:Skolotajs noklikšķina uz pogas pirms testa uzsākšana "Atļaut skatit pareizās atbildes" un pec tam Lietotājs noklikšķina uz pogas " Skatīt pareizās atbildes" pec testa nobeigums.

**Apstrāde**: Sistēmā jau ir pareizās un skolēna atbildes uz jautājumiem.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Skolēna atbildes.
2. Pareizās atbildes.
3. **Funkcija “Apskatīt visu skolēnu vidējās atzīmes”**

***ID: T9*** **Ievads**: Iespēja skatīt visu skolēnu var redzēt vidējo atzīmi par visu darbus (varat redzet tikai tā skolēna vārdu, kuram pieder konts visi citi atzimi bus bez skolenus vardiem).

**Ievade**: Lietotājs noklikšķina uz pogas " Apskatīt visu skolēnu vidējās atzīmes ".

**Apstrāde**: Sistēma aprēķina katra skolēna vidējo atzīmi par visus darbus.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Skolēna vidējo atzīmi par visus darbus.
2. Citi skolēni un viņu vidējo atzīmi par visus darbus.
3. Kāds ir katra skolēna rangs no augstākās vidējās atzīmes līdz zemākajai videjai atzīmei
4. **Funkcija “Iespēja Trenieties”**

***ID: T10*** **Ievads**: Iespēja nokārtot nelielu testu, kas būs saistīts ar testa tēmu, bet tikai tad, ja administrators dod iespēju praktizēties.

**Ievade**: Lietotājs noklikšķina uz pogas " Treniņš".

**Apstrāde**: Administrators var izvēlēties, vai veikt vai neveikt papildu testu, lai pārbaudītu zināšanas,.Ja treniņš nav pieejams izpildei, tad pēc treniņa izvēles tiek parādīts dialoglodziņš ar tekstu "Treniņš nav pieejams!".

**Izvade**: Tiek atvērts dialoglodziņš, kurā kur tiks uzdots neliels tests.

## Sistēmas nefunkcionālās prasības

1. Prasības produktam:
   1. Lietotāju saskarne ar sistēmu notiek latviešu valodā.
   2. Produktam nav paredzētas vairākas sastāvdaļas. Produktam ir trīs ārējās saskarnes: reģistrēšanai sistēmā, ielogošanai sistēmā, testa pildīšanai.
   3. Programma ir paredzēta vienam lietotājam, kurš saprot latviešu valodu. Lietotāja vecuma vai izglītības ierobežojumi nav paredzēti.
2. Ārējās saskarnes prasības
   1. Lietotāja saskarne: produkts paredz vienotu lietotāja saskarni.
   2. Ekrāna formāti: minimālais ekrāna izmērs ir 10” ar minimālo izšķirtspēju –800x600 punkti.
   3. Lietotāja saskarne “Lietotāja reģistrēšanas saskarne”. Saskarnes struktūra: skat. [1.6. att.](#att1_6)

Reģistreties

Uzvards:

Lietotajvards:

Parole:

Parole (atkartoti):

Vards:

Ielogoties

1.6. att. Lietotāja reģistrēšanas saskarne

**Saskarnes elementi:**

1. teksta lauks “Vārds”
2. teksta lauks “Uzvārds”
3. teksta lauks “Lietotājvārds”
4. teksta lauks ‘Parole”
5. teksta lauks “Parole (atkārtoti)”
6. poga “Reģistrēties”
7. poga “ Ielogoties”
   1. Lietotāja saskarne “Lietotāja ielogošanas saskarne”. Saskarnes struktūra: skat. [1.7. att.](#att1_7)

Reģistreties

Lietotājvārds:

Parole:

Ielogoties

**Lietotāja pieslēgšanā**

1.7. att. Lietotāja ielogošanas saskarne

**Saskarnes elementi:**

1. teksta lauks “Lietotājvārds”
2. teksta lauks ‘Parole”
3. poga “Ielogoties”
4. poga “Reģistrēties”
   1. Lietotāja saskarne “Testa izpilde”. Saskarnes struktūra: skat. [1.8. att.](#att1_8)

a) b)

Talak

1. Jautajums “Jautajuma teksts”

1. atbilde

2. atbilde

3. atbilde

Sakt testu

Tests “Nosaukums”

Sakt treniņu

c)

Beigt

Rezultati

Procenti: X%

Atzime: A

Skatīt pareizās atbildes

1.8. att. Testa izpildes saskarne: testa izvēle/sakums; b) atbilde uz jautājumu/-iem ; c) rezultāti

Saskarnes elementi:

a)

1. Teksts”Tests Nosaukums”
2. Poga “Sakt testu”
3. Poga “Sakt treniņu”

b)

1. Teksts” Jautajums/Jautajuma teksts”
2. Atbilžu izvēle
3. Poga “Talak”

c)

1. Teksts”Rezultati”
2. Teksts”Procenti”
3. Teksts”Atzime”
4. Poga “Beigt”
5. Poga “Skatīt pareizās atbildes”
   1. Lietotāja saskarne “Treniņa izpilde”. Saskarnes struktūra: skat. [1.9. att.](#att1_8)

Talak

1. Jautajums “Jautajuma teksts”

1. atbilde

2. atbilde

3. atbilde

a)

b)

Beigt

Treniņa Rezultati

Procenti: X%

Atzime: A

Skatīt pareizās atbildes

1.9. att. Testa izpildes saskarne: treniņa sakums; b) atbilde uz jautājumu/-iem ; c) rezultāti

Saskarnes elementi:

a)

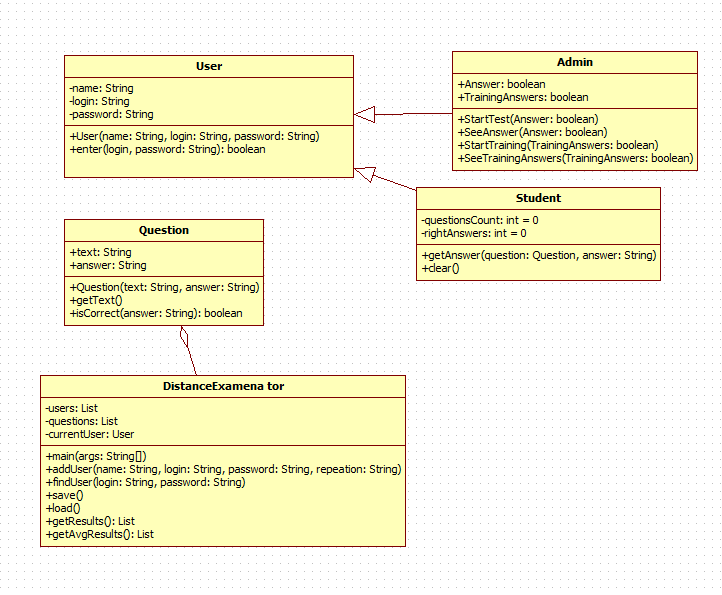
1. Teksts” Jautajums/Jautajuma teksts”
2. Atbilžu izvēle
3. Poga “Talak”

b)

1. Teksts”Treniņa Rezultati”
2. Teksts”Procenti”
3. Teksts”Atzime”
4. Poga “Beigt”
5. Poga “Skatīt pareizās atbildes”

# Priekšmetiskās jomas klašu sistēmas izstrāde

## Klašu diagramma



2.1. att. Testēšanas sistēmas klašu diagramma UML valodā

Klase User – ir atbildīga par darbu ar lietotājiem.

* User (name: String, login: String, password: String) – konstruktors, kurš izveido

jaunu lietotāju ar norādītiem datiem.

* enter (login: String, password: String) – metode, kura nodrošina lietotāju pieslēgšanu

sistēmai.

Klase Student – nodrošina audzēkņa testa pildīšanas darbu.

## getAnswer(question: Question, answer: String) –medote,kura izveido ievadītas atbildes pārbaudi.

## clear();

Klase Admin – nodrošina darbu ar lietotajiem un testiem.

startTest(answer: boolean) - metode, kas atbildiga par testu uzsakšanu.

seeAnswer(answer: boolean) – metode, kas dot iespeju lietotājam apskatīt savus rezultatus.

StartTraining(TrainingAnswers: boolean) - metode, kas atbildiga par treniņu uzsakšanu.

SeeTrainingAnswers(TrainingAnswers: boolean) – metode, kas dot iespeju lietotājam apskatīt savus treniņu rezultatus.

Klase Question – ir atbildīga par darbu ar testa jautājumiem, to izvadi un ievadītas atbildes pārbaudi.

· Question(text: String, answer: String) - metode, kas padara to iespējamu atbildet uz jautajumi.

· getText()

· isCorrect(answer: String) - metode, kas parbaude vai atbilde ir preizi vai ne pareizi.

Klase DistanceExamenator – programmas darbību koordinējošā klase. Tā nodrošina programmas galvenās metodes izsaukumu un grafiskās saskarnes izveidi, tā arī.

main(args: string[])

addUser(name: String, login: String, password: String, repeation: String) - metode, kas izveidot lietotāju pievenošanu sistemai;

findUser(login: String, password: String) - metode, kas izveidot lietotāju atrašanu vieglak;

save() – metode, kura fikse rezultātus;

load() – metode, sgalaba rezultātus datu bazē;

getResults() – metode, kas iedot iespeju lietotāju redzet savus rezultatus;

getAvgResults() – metode, kas kas iedot iespeju lietotāju redzet videjo rezultatus klasse;

## Klašu realizēšana Java valodā

Izstrādātās klašu diagrammas ([2.1. att.](#att2_1)) realizēšanai ***NetBeans*** vidē tika izveidots projekts ar nosaukumu ***TestingSystem***, kurā izstrādātās klases tika aprakstītas ***Java*** valodā. Šeit tiek piedāvāts klašu vispārīgā apraksta (klašu deklarācijas) programmas kods un metožu apraksti.

Pilns avota programmas kods ar klašu metožu definīcijām ir ievietots pielikumā “[Pielikums A.](#_bookmark33) [Klašu sistēma](#_bookmark33)”, bet programmas kods, kas realizē šīs klašu sistēmas testēšanu (klase ar grafisko lietotāja saskarni) ir ievietots pielikumā “[Pielikums B. testa programmas kods](#_bookmark34)”.

class User{

private String name; private String login; private String password;

private void User(String name, String login, String password){

}

private boolean enter(String login, String password){ return true;

}

}

class Admin

# Testa programmas izstrāde

## Testēšanas metodikas

Pirms priekšmetiskās jomas klašu sistēmas testēšanas programmas izstrādes, ir jānosaka testēšanas metodika.

Ir jāatšķir koda testēšana un atkļūdošana. Atkļūdošanu veic programmētājs, izmantojot iebūvētos izstrādes vides rīkus un balstoties uz pieredzi programmas koda rakstīšanā. Būtībā runa iet par sintaktisko un semantisko kļūdu identificēšanu programmas tekstā.

Testēšana ir process, kurā nepieciešams plānot un īstenot vairākas sākotnējās procedūras, no kurām galvenā ir testa piemēru kopas izstrāde, kas veido testa plānu. Testpiemēri vairumā gadījumu ir balstīti uz sistēmas funkcionālajām prasībām un var ietekmēt dažādus attīstības līmeņus (vienības testēšana, integrācijas testēšana, sistēmas testēšana).

Projekta darba laikā nepieciešams veikt vienkāršotu vienības testēšanas versiju, kas realizē visu izstrādāto klašu metožu testēšanu. Ar testēšanas metodiku projekta darbā pieņem dažādu klases metožu izsaukšanas iespēju sarakstu ar gaidāmajiem rezultātiem un šo testēšanas izsaukumu izpildes kārtību.

## Testēšanas programmas struktūra

Testēšanas programmas projektēšana un programmatūras izstrāde ir ļoti līdzīga iepriekš aprakstītajam klašu sistēmas izstrādes procesam.

Jāatzīmē tie momenti un prasības, kas pastāv, izstrādājot testēšanas programmu.

* + Testēšanas programmai ir loga interfeiss, kas ietver sevī izvēlnes un interaktīvā interfeisa rīkus, ar kuru palīdzību var pārbaudīt izveidoto klašu sistēmu.
  + Interfeisa elementu (pogas, saraksti, ievades lauki, dialoglodziņi, grafiskie elementi utt.) komplekts tiek noteikts, pamatojoties uz iepriekšējā sadaļā aprakstīto testēšanas metodiku.
  + Testēšanas programmas interfeisa kvalitātei jābūt vienkāršai lietošanā un interaktīvai. Galvenā prasība ir spēja realizēt testēšanas metodes.

1. Ielogošanas procesa apraksts:
   * Ielogošanas process paredzēts reģistrēto lietotāju pievienošanai sistēmā. Lai lietotājs varētu piekļūt testam viņam pēc ir nepieciešams autorizēties. Ielogošanas procesā lietotājs ievada iepriekš reģistrētos datus – lietotājvārdu un paroli ([3.1. att.](#att3_1) ). Ja lietotāja dati sakrīt ar validācijas datiem, tad lietotājam tiek atvērts programmas galvenais logs ([3.x. att.](#att3_2)).
   * Ielogošanas procesā ir iespējamas vairākas kļūdas – neaizpildīti lauki, neprecīzi ievadīti lietotāja dati.
   1. att. Testēšanas sistēmas ielogošanas forma

Reģistrācijas procesa apraksts

## Testēšanas rezultāti

# Lietotāja rokasgrāmata

## Lietotāja instrukcija

1. Lietotājs palaiž programmu.
2. Piesakās sistēmā, izmantojot savu lietotājvārdu un paroli, ja lietotājam nav sava konta, tad viņš var reģistrēt to.
   1. Reģistrācijas logā lietotājs aizpilda visus nepieciešamus datus un noklikšķina uz pogas Reģistrēties.
3. Lietotāja izvēlnē ir 2 pogas.
   1. Poga «Pildīt testu» ļauj sākt pildīt testu.
      1. Pēc testa izpildīšanas tiek paradīts testa izpildes rezultāti.
   2. Poga «Atteikties no konta» ļauj iziet no sava konta.
4. Administratora izvēlnē ir 3 pogas.
   1. Poga «Sākt testu» ļauj atļaut lietotājam piekļuvi testam.
   2. Poga «Testa statuss» ļauj paradīt testa statusu – «Tests ir atļauts» vai «Tests nav atļauts».
   3. Poga «Atteikties no konta» ļauj iziet no administratora konta.

## Palīdzības sistēma (Help)

Lai sākt pildīt testu, Jums ir nepieciešams ieiet savā kontā, ievadot lietotājvārdu un paroli.

## Sistēmas ziņojumi

# Secinājumi

Tika izstrādāts projekts «Testēšanas sistēma» NetBeans vidē. Tika izpildītas visas programmas bāzes prasības. Projekta izpildīšanas laikā radās dažādas problēmas, kas neļāva veikt sākotnējo vēlamu rezultātu, bet tajā pašā laikā bija interesanti uzzināt jaunu informāciju, kas noderēs turpmāk. Piemēram, darbs ar JDBC (the Java Database Connectivity) un jaunas darba izpildes metodes. Tā bija mana pirmā pieredze, izstrādājot projektu. Diemžēl ir palikušas nerealizētas idejas, kas varētu izveidot manu projektu labāk, bet galvenais, ka ir realizētas galvenās prasības un viss strādā. Ceru, jo tālāk, jo labāk!

# Izmantotās literatūras (informācijas avotu) saraksts

1. **Zaiceva, L.** *Programmatūras izstrādes tehnoloģija.* Rīga : RTU, 2002.

2. **Kodors, Sergejs.** *IEVADS PRASĪBU INŽENIERIJĀ. Mācību līdzeklis.* Rēzekne : Rēzeknes Tehnoloģiju akadēmija, 2019. gada. 978-9984-44-226-6.

3. **Dortiņa, I. un Dortiņš, A.** 1.1.1. Ievads Java. Java pamati. [Tiešsaiste] 2024. gada 13. 2. [Citēts: 2025. gada 20. 3.] https://skolo.lv/mod/resource/view.php?id=82126301.

4. **—.** 1.1.2. Java sazarotas struktūras operātori. [Tiešsaiste] 2024. gada 14. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82257905.

5. —. 1.1.3. Java cikliskās struktūras operatori. [Tiešsaiste] 2022. gada 22. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82383944.

6. —. 1.2.1.1. Java masīvi. [Tiešsaiste] 2024. gada 29. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82503734.

7. —. 1.2.1.2. Java rakstzīmju masīvi un rakstzīmju virknes. [Tiešsaiste] 2024. gada 7. 3. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82608403.

8. —. 1.2.1.3. Java dinamiskie masīvi. [Tiešsaiste] 2024. gada 13. 3. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82703677.

9. —. 1.2.2. Java ieraksti. [Tiešsaiste] 2024. gada 19. 3. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82785148.

10. —. 1.2.3. Java Ievades/izvades sistēma. [Tiešsaiste] 2024. gada 3. 4. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82944472&redirect=1.

11. —. 2.1.1.1. Java. OOP koncepcija. [Tiešsaiste] 2024. gada 16. 4. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=83119822.

12. **Dortiņa, I. un Stašanova, L.** 2.1.1.2. Java. Metodes un konstruktori. [Tiešsaiste] 2024. gada 2. 9. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=84426472.

13. **—.** 2.1.2. Java. Mantošana un metožu pārdefinēšana. [Tiešsaiste] 2024. gada 30. 9. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=85417326.

14. —. 2.1.3. Java pakotnes un interfeisi. [Tiešsaiste] 2024. gada 14. 10. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=86185154.

15. —. 2.2. Java izņēmumi un to apstrāde. [Tiešsaiste] 2022. gada 23. 10. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=86272136&redirect=1.

16. —. 4.1.1. Java. Grafiskais lietotāja interfeiss. [Tiešsaiste] 2025. gada 23. 1. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=87282004.

17. —. 4.1.2. Java GUI izveidošana NetBeans vidē. [Tiešsaiste] 2025. gada 5. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=87334441.

18. Latvijas Nacionālais terminoloģijas portāls. *termini.gov.lv.* [Tiešsaiste] Izstrādātājs: Tilde., 2005-2023. gada. [Citēts: 2025. gada 20. 2.] https://termini.gov.lv/.

19. **Taylor Johnson, Dung X. Nguyen.** UML and more JAVA Syntax. [Tiešsaiste] COMP 212 LAB 1.5, 2007. gada 16. 1. [Citēts: 2025. gada 20. 2.] https://www.clear.rice.edu/comp212/07-spring/labs/01.5/.

# Pielikums A. Klašu sistēma

# Pielikums B. testa programmas kods