|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования**«МИРЭА – Российский технологический университет»****РТУ МИРЭА** |
| **Институт информационных технологий** |
| **Кафедра корпоративных информационных систем****Лабораторная работа №5**по дисциплинеОбъектно-ориентированное программирование**Тема лабораторной работы:** Графический пользовательский интерфейс с использованием библиотеки swing

|  |  |
| --- | --- |
| Студент группы ИКБО-08-18 | Валяев Данила Андреевич  |
|  |  |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись студента) |
|  |  |
|  |  |
| Работа представлена к защите |  «07» декабря 2019 г. |
|  |  |
|  |  |

 |
|  |
| **Содержание**[**Задание** 3](#_Toc27908433)[**Заключение** 7](#_Toc27908434)[**Список используемых источников** 8](#_Toc27908435)[**Приложение** 9](#_Toc27908436) |

# **Задание:** Реализовать классы в соответствии с предложенной диаграммой.



Рисунок 1.1 – Диаграмма варианта

Интерфейсы:

- Интерфейс для входа в систему.

- Интерфейс для регистрации пользователей (преподавателей).

- Интерфейс администратора (для указания числа аудиторий и др.)

- Интерфейс для приема заявок.

- Интерфейс для составления расписания.

Таблицы данных:

- Пользователи

- Заявки

- Информация об университете (число аудиторий и др.)

Реализация класса пользователя:



Рисунок 1.2 – Класс пользователя

Реализация класса преподавателя:



Рисунок 1.3 – Класс преподавателя

**UML** – унифицированный язык моделирования (Unified Modeling Language) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем.

Реализация класса запроса:



Рисунок 1.4 – Класс запроса

Реализация функции входа пользователя:

private boolean loginUser() {

 Scanner in = new Scanner(System.in);

 print("Введите логин: ");

 String login = in.nextLine();

 print("Введите пароль: ");

 String password = in.nextLine();

 Timetable.currentUser = this.findUser(login, password);

 return Timetable.currentUser.enter(login, password);

 }

Реализация функции построения расписания:



Рисунок 1.5 – Функция построения расписания

Swing — библиотека для создания графического интерфейса для программ на языке Java. Swing был разработан компанией Sun Microsystems. Он содержит ряд графических компонентов (англ. Swing widgets), таких как кнопки, поля ввода, таблицы и т. д.

Для создания графического интерфейса приложения необходимо использовать специальные компоненты библиотеки Swing, называемые контейнерами высшего уровня (top level containers). Они представляют собой окна операционной системы, в которых размещаются компоненты пользовательского интерфейса. К контейнерам высшего уровня относятся окна JFrame и JWindow, диалоговое окно JDialog, а также апплет JApplet (который не является окном, но тоже предназначен для вывода интерфейса в браузере, запускающем этот апплет). Контейнеры высшего уровня Swing представляют собой тяжеловесные компоненты и являются исключением из общего правила. Все остальные компоненты Swing являются легковесными.

# **Заключение**

В процессе выполнения лабораторной работы мы познакомились с UML диаграммами и научились реализовывать графический интерфейс с помощью библиотеки Swing.

# **Список используемых источников**

1. UML[Электронный ресурс]. – URL: <https://prog-cpp.ru/uml-classes/>
2. Swing [Электронный ресурс]. – URL: http://java-online.ru/libs-swing.xhtml

# **Приложение**

TimeTable

package com.company;

import java.util.ArrayList;

public class Timetable

{

 public static ArrayList<Professor> users;

 public static ArrayList<Request> requests;

 public static Pairs[][] pairs;

 public static Professor currentUser;

 public static int rooms;

 public static int groups;

 public static void main(String[] args) {

 LoginWindow loginWindow = new LoginWindow();

 loginWindow.setVisible(true);

 for (int i = 0; i < 6; ++i) {

 for (int j = 0; j < 6; ++j) {

 Timetable.pairs[i][j] = new Pairs();

 }

 }

 }

}

User

package com.company;

class User {

 private String name;

 private String login;

 private String password;

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

 this.name = name;

 this.login = login;

 this.password = password;

 }

 boolean enter(String login, String password) {

 return this.login.equals(login) && this.password.equals(password);

 }

 String getName() {

 return this.name;

 }

 String getLogin() {

 return this.login;

 }

 String getPassword() {

 return this.password;

 }

}

Professor

package com.company;

class Professor extends User {

 Professor(String name, String login, String password) {

 super(name, login, password);

 }

}

Request

package com.company;

class Request {

 private Professor requester;

 private String discipline;

 private int group;

 private int pairsInWeek;

 Request(Professor requester, String disc, int group, int hours) {

 this.requester = requester;

 this.discipline = disc;

 this.group = group;

 this.pairsInWeek = hours;

 }

 String getRequester() {

 return this.requester.getName();

 }

 String getDiscipline() {

 return this.discipline;

 }

 int getGroup() {

 return this.group;

 }

 int getPairs() {

 return this.pairsInWeek;

 }

}

Window

package com.company;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class Window extends JFrame {

 private JLabel labelDict = new JLabel("Дисциплина: ");

 private JLabel labelGroup = new JLabel("Группа: ");

 private JLabel labelCount = new JLabel("Кол-во пар: ");

 private JTextField inputDict = new JTextField("");

 private JTextField inputGroup = new JTextField("");

 private JTextField inputCount = new JTextField("");

 private JButton Rec = new JButton("Сделать запрос");

 private JButton Pairs = new JButton("Узнать расписание");

 private JButton Menu = new JButton("Главное меню");

 Window() {

 super("Основное меню");

 Dimension dim = getToolkit().getScreenSize();

 this.setBounds(dim.width/2, dim.height/2, 500, 300);

 this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

 Container container = this.getContentPane();

 container.setLayout(new GridLayout(7, 7, 7, 7));

 container.add(labelDict);

 container.add(inputDict);

 container.add(labelGroup);

 container.add(inputGroup);

 container.add(labelCount);

 container.add(inputCount);

 ButtonGroup group = new ButtonGroup();

 group.add(Rec);

 group.add(Pairs);

 Rec.addActionListener(new ButtonRecListener());

 container.add(Rec);

 Pairs.addActionListener(new ButtonPairsListener());

 container.add(Pairs);

 Menu.addActionListener(new ButtonMenuListener());

 container.add(Menu);

 }

 class ButtonMenuListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 Window.this.dispose();

 LoginWindow loginWindow = new LoginWindow();

 loginWindow.setVisible(true);

 }

 }

 class ButtonPairsListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 printPairs(Timetable.currentUser);

 }

 }

 class ButtonRecListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 if (inputDict.getText().trim().length() > 0

 && inputGroup.getText().trim().length() > 0

 && inputCount.getText().trim().length() > 0) {

 addRequest(inputDict.getText(), Integer.parseInt(inputGroup.getText()), Integer.parseInt(inputCount.getText()));

 JOptionPane.showMessageDialog(null, "Запрос сделан.", "ШТАЩ", JOptionPane.PLAIN\_MESSAGE);

 }

 else

 JOptionPane.showMessageDialog(null, "Поля ввода должны быть заполнены.", "Error", JOptionPane.PLAIN\_MESSAGE);

 }

 }

 private void printPairs(Professor user) {

 for (int i = 0; i < 6; ++i) {

 for (int j = 0; j < 6; ++j) {

 Timetable.pairs[i][j].printPair(user.getName());

 }

 }

 }

 private void addRequest(String disc, int group, int pairs) {

 Request r = new Request(Timetable.currentUser, disc, group, pairs);

 Timetable.requests.add(r);

 }

}

LoginWindow

package com.company;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class LoginWindow extends JFrame{

 private JLabel labelLogin = new JLabel("Логин: ");

 private JLabel labelPassword = new JLabel("Пароль: ");

 private JTextField inputLogin = new JTextField("");

 private JTextField inputPassword = new JTextField("");

 private JButton buttonEnter = new JButton("Войти");

 private JButton buttonRegistration = new JButton("Регистрация");

 LoginWindow() {

 super("Меню авторизации");

 Dimension dim = getToolkit().getScreenSize();

 this.setBounds(dim.width/2, dim.height/2, 275, 125);

 this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

 Container container = this.getContentPane();

 container.setLayout(new GridLayout(3, 2, 2, 2));

 container.add(labelLogin);

 container.add(inputLogin);

 container.add(labelPassword);

 container.add(inputPassword);

 buttonRegistration.addActionListener(new ButtonRegisterEventListener());

 buttonEnter.addActionListener(new ButtonEnterEventListener());

 container.add(buttonRegistration);

 container.add(buttonEnter);

 }

 static class ButtonRegisterEventListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 RegistrationWindow registrationWindow = new RegistrationWindow();

 registrationWindow.setVisible(true);

 }

 }

 class ButtonEnterEventListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 if (inputLogin.getText().trim().length() > 0 && inputPassword.getText().trim().length() > 0) {

 try{

 if(inputLogin.getText().equals("admin") && inputPassword.getText().equals("admin")){

 AdminWindow adminWindow = new AdminWindow();

 adminWindow.setVisible(true);

 return;

 }

 boolean userNum = Timetable.currentUser.enter(inputLogin.getText(), inputPassword.getText());

 if (userNum) {

 LoginWindow.this.dispose();

 Window window = new Window();

 window.setVisible(true);

 }

 }

 catch (Exception e1){

 JOptionPane.showMessageDialog(null, "Проверьте введённые данные.", "Error", JOptionPane.PLAIN\_MESSAGE);

 }

 }

 else

 JOptionPane.showMessageDialog(null, "Поля ввода должны быть заполнены.", "Error", JOptionPane.PLAIN\_MESSAGE);

 }

 }

}

Pairs

package com.company;

import javax.swing.\*;

import java.util.ArrayList;

class Pairs

{

 private String[] days;

 private ArrayList<Integer> room;

 private ArrayList<Integer> group;

 private ArrayList<String> professor;

 private ArrayList<String> pair;

 Pairs() {

 this.days = new String[] { "Понедельник", "Вторник", "Среда", "Четверг", "Пятница", "Суббота" };

 this.room = new ArrayList<Integer>();

 this.group = new ArrayList<Integer>();

 this.professor = new ArrayList<String>();

 this.pair = new ArrayList<String>();

 }

 void Room\_(int room) {

 this.room.add(room);

 }

 void Group\_(int group) {

 this.group.add(group);

 }

 void Prof\_(String professor) {

 this.professor.add(professor);

 }

 boolean checkRoom(int room) {

 return this.room.contains(room);

 }

 boolean checkGroup(int group) {

 return this.group.contains(group);

 }

 boolean checkProfessor(String professor) {

 return this.professor.contains(professor);

 }

 int getRooms() {

 return this.room.size();

 }

 void setPair(String professor, String disc, int room, int group, int hour, int day) {

 this.pair.add("группа: " + group + ", профессор: " + professor + ", дисциплина: " + disc + ", аудитория: " +

 room + ", пара:" + (hour + 1) + ", день: " + this.days[day]);

 }

 void printPair(String prof) {

 for (String i : this.pair) {

 if (i.contains(prof)) {

 System.out.println(i);

 }

 }

 }

}

RegistrationWindow

package com.company;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class RegistrationWindow extends JFrame {

 private JLabel labelName = new JLabel("Имя: ");

 private JLabel labelLogin = new JLabel("Логин: ");

 private JLabel labelPassword = new JLabel("Пароль: ");

 private JTextField inputName = new JTextField("");

 private JTextField inputLogin = new JTextField("");

 private JTextField inputPassword = new JTextField("");

 private JButton buttonRegistration = new JButton("Регистрация");

 RegistrationWindow() {

 super("Меню регистрации");

 Dimension dim = getToolkit().getScreenSize();

 this.setBounds(dim.width/2, dim.height/2, 275, 125);

 this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

 Container container = this.getContentPane();

 container.setLayout(new GridLayout(4, 2, 2, 2));

 container.add(labelName);

 container.add(inputName);

 container.add(labelLogin);

 container.add(inputLogin);

 container.add(labelPassword);

 container.add(inputPassword);

 buttonRegistration.addActionListener(new ButtonRegisterEventListener());

 container.add(buttonRegistration);

 }

 class ButtonRegisterEventListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 if (inputName.getText().trim().length() > 0

 && inputLogin.getText().trim().length() > 0

 && inputPassword.getText().trim().length() > 0) {

 if (findUser(inputLogin.getText(), inputPassword.getText()) == null) {

 JOptionPane.showMessageDialog(null, "Регистрация прошла успешно, " + inputName.getText() + "!", "INFO", JOptionPane.PLAIN\_MESSAGE);

 Timetable.currentUser = new Professor(inputName.getText(), inputLogin.getText(), inputPassword.getText());

 Timetable.users.add(Timetable.currentUser);

 RegistrationWindow.this.dispose();

 }

 else {

 JOptionPane.showMessageDialog(null, "Пользователь уже зарегистрирован.", "Error", JOptionPane.PLAIN\_MESSAGE);

 }

 }

 }

 private Professor findUser(String login, String password) {

 for (Professor i : Timetable.users) {

 if (i.getLogin().equals(login) && i.getPassword().equals(password)) {

 return i;

 }

 }

 return null;

 }

 }

}

AdminWindow

package com.company;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class AdminWindow extends JFrame{

 private JLabel countGroups = new JLabel("Количество групп: ");

 private JLabel countRooms = new JLabel("Количество аудиторий: ");

 private JTextField inputGroups = new JTextField("");

 private JTextField inputRooms = new JTextField("");

 private JButton buttonSet = new JButton("Установить");

 AdminWindow() {

 super("Меню администратора");

 Dimension dim = getToolkit().getScreenSize();

 this.setBounds(dim.width/2, dim.height/2, 350, 125);

 this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

 Container container = this.getContentPane();

 container.setLayout(new GridLayout(3, 2, 2, 2));

 container.add(countGroups);

 container.add(inputGroups);

 container.add(countRooms);

 container.add(inputRooms);

 buttonSet.addActionListener(new ButtonRegisterEventListener());

 container.add(buttonSet);

 }

 class ButtonRegisterEventListener implements ActionListener {

 public void actionPerformed (ActionEvent e) {

 if (inputGroups.getText().trim().length() > 0

 && inputRooms.getText().trim().length() > 0){

 Timetable.groups = Integer.parseInt(inputGroups.getText());

 Timetable.rooms = Integer.parseInt(inputRooms.getText());

 processRequests();

 JOptionPane.showMessageDialog(null, "Кол-во групп и аудиторий установлено.", "INFO", JOptionPane.PLAIN\_MESSAGE);

 }

 AdminWindow.this.dispose();

 }

 }

 private static void processRequests() {

 for (Request r : Timetable.requests) {

 int number = r.getPairs();

 for (int i = 0; i < 6; ++i) {

 for (int j = 0; j < 6; ++j) {

 int room = 1;

 if (Timetable.pairs[i][j].getRooms() < Timetable.rooms && number > 0) {

 while (Timetable.pairs[i][j].checkRoom(room)) {

 ++room;

 }

 if (!Timetable.pairs[i][j].checkGroup(r.getGroup()) && !Timetable.pairs[i][j].checkProfessor(r.getRequester())) {

 Timetable.pairs[i][j].setPair(r.getRequester(), r.getDiscipline(), room, r.getGroup(), j, i);

 Timetable.pairs[i][j].Group\_(r.getGroup());

 Timetable.pairs[i][j].Prof\_(r.getRequester());

 Timetable.pairs[i][j].Room\_(room);

 --number;

 }

 }

 }

 }

 }

 }

}