

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Санкт-Петербургский государственный электротехнический университет
“ЛЭТИ” им.В.И.Ульянова (Ленина) »

Кафедра МОЭВМ

ОТЧЕТ
по лабораторно-практической работе № 2
«Разработка интерфейса пользователя»
по дисциплине «Объектно - ориентированное
программирование на языке Java»

Выполнил: Мохно Д. А.

Факультет КТИ

Группа № 3312

Преподаватель: Павловский М.Г.

Подпись преподавателя _____

Санкт-Петербург

2024 г

Цель работы

Знакомство с правилами построения экранной формы.

Описание назначения экранной формы

Экранная форма предназначена для отображения информации о больных, врачах и их расписании работы. Она может менять свой размер на экране. Форма должна реализовывать следующие функции: загрузку списка врачей и больных из файла, сохранение информации в файле, добавление данных о враче или больном в список, редактирования данных о враче или больном, удаление информации из списка, поиск врача по имени или специализации, а также больного по имени. Так же, форма должна обеспечивать вывод на печать справки о болезни больного.

Макет экранной формы

Поликлиника

Сохранить Искать Поиск

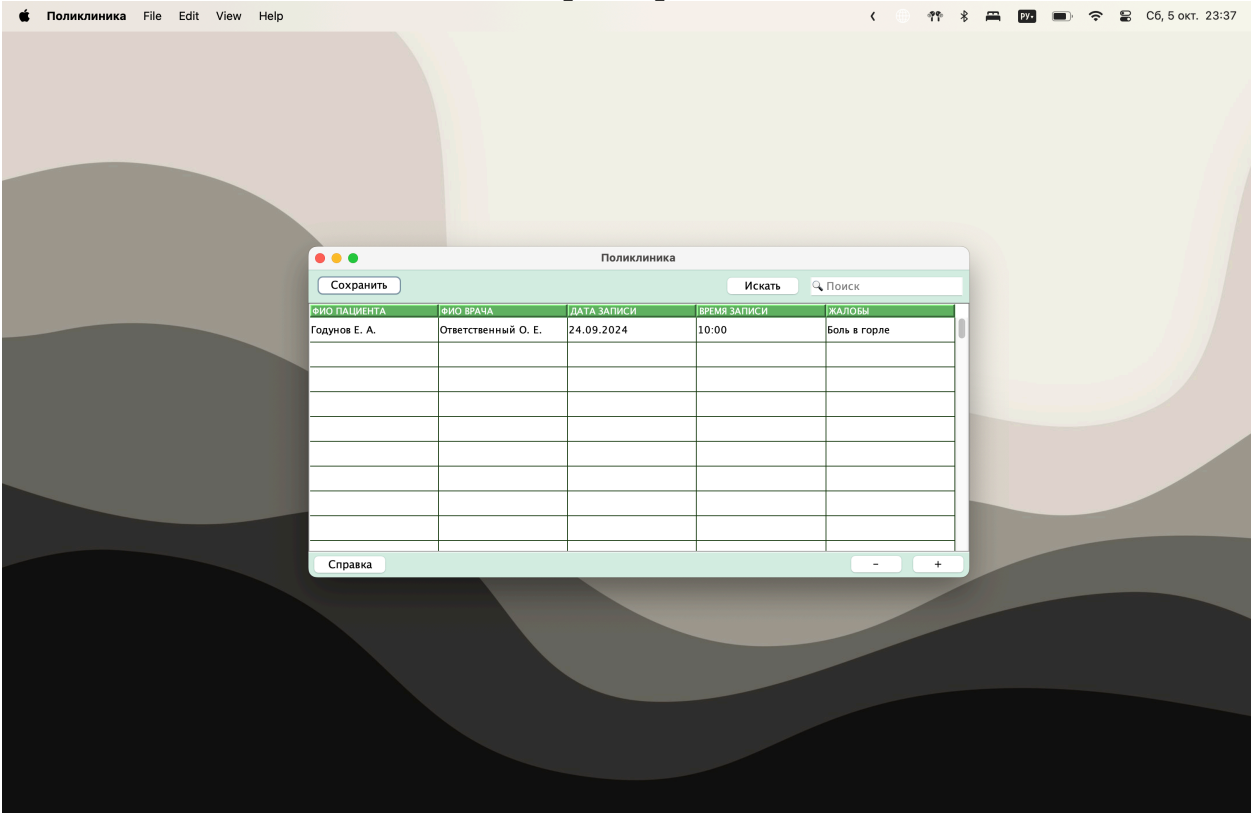
ФИО ПАЦИЕНТА	ФИО ВРАЧА	ДАТА ЗАПИСИ	ВРЕМЯ ЗАПИСИ	ЖАЛОБЫ	

Справка - +

Mon Jun 5 9:41 AM

Контрольные примеры

Пример 1:



Текст документации, сгенерированный Javadoc.

PACKAGE

CLASS

USE

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

SEARCH

Class `UserInterface`

`java.lang.Object`
`UserInterface`

```
public class UserInterface
extends Object
```

Field Summary

Modifier and Type	Field	Description
<code>JBButton</code>	<code>add</code>	
<code>JBButton</code>	<code>del</code>	
<code>JMenuBar</code>	<code>menuBar</code>	
<code>String</code>	<code>placeholder</code>	
<code>JBButton</code>	<code>save</code>	
<code>JScrollPane</code>	<code>scrollPane</code>	
<code>JBButton</code>	<code>searchButton</code>	
<code>JTextField</code>	<code>searchField</code>	
<code>JBButton</code>	<code>showReference</code>	
<code>JTable</code>	<code>table</code>	
<code>DefaultTableModel</code>	<code>tableModel</code>	
<code>JFrame</code>	<code>window</code>	

Constructor Summary

Constructors	Description
<code>UserInterface()</code>	

PACKAGE

CLASS

USE

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

SEARCH

Method Summary

All Methods

Static Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	
void	show()	Метод для отображения окна приложения

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Details

window

public JFrame window

save

public JButton save

menuBar

public JMenuBar menuBar

searchField

public JTextField searchField

searchButton

public JButton searchButton

placeholder

public String placeholder

showReference

public JButton showReference

add

public JButton add

del

public JButton del

tableModel

public DefaultTableModel tableModel

table

public JTable table

scrollPane

public JScrollPane scrollPane

Constructor Details

UserInterface

public UserInterface()

Method Details

show

public void show()

Метод для отображения окна приложения

main

public static void main(String[] args)

Текст программы

```

/**
 * Лабораторная работа №2
 *
 * @author Даниил Мохно 3312
 * @version 1.0
 */
import javax.swing.*;
import java.awt.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.FocusAdapter;
import java.awt.event.FocusEvent;

public class UserInterface {
    public JFrame window;
    private Image icon;

    private JPanel panel;
    private JPanel header;
    private JPanel saveArea;
    private JPanel searchArea;
    public JButton save;
    public JMenuBar menuBar;
    public JTextField searchField;
    public JButton searchButton;
    public String placeholder;
    private JPanel footer;
    public JButton showReference;
    public JButton add;
    public JButton del;
    private JPanel editPanel;

    public DefaultTableModel tableModel;
    public JTable table;
    public JScrollPane scrollPane;

    /**
     * Метод для отображения окна приложения
     */
    public void show(){
        if (System.getProperty("os.name").toLowerCase().contains("mac"))
            System.setProperty("apple.awt.application.name", "Поликлиника");
    };

    // Создание основного окна
    window = new JFrame("Поликлиника");
    window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    window.setSize(800, 400);
    window.setLocationRelativeTo(null);

    // Добавление иконки
    icon =
window.getToolkit().getImage(getClass().getResource("/img/icon.png"));
    if (System.getProperty("os.name").toLowerCase().contains("mac")) {
        Taskbar.getTaskbar().setIconImage(icon);
        System.setProperty("apple.laf.useScreenMenuBar", "true");
    } else {
        window.setIconImage(icon);
    }

    // Устанавливаем панель
    panel = new JPanel();
    panel.setBackground(new Color(216, 240, 211));
    panel.setLayout(new BorderLayout());

```

```

// Создаём шапку
header = new JPanel();
header.setBackground(new Color(211, 240, 228));
header.setLayout(new BorderLayout());

// Кнопка сохранения
saveArea = new JPanel();
saveArea.setBackground(new Color(211, 240, 228));
save = new JButton("Сохранить");
saveArea.add(save);
header.add(saveArea, BorderLayout.WEST);

// Создание меню
menuBar = new JMenuBar();
String []menus = {"File", "Edit", "View", "Help"};
String [][]menuItems = {"Open file", "Save", "Save as", "Print"},
{"Add", "Delete"}};
for (int i = 0; i < menus.length; i++) {
    JMenu menu = new JMenu(menus[i]);
    for (int j = 0; menuItems.length > i && j < menuItems[i].length;
j++) {
        item.addActionListener(e -> {})
        menu.add(new JMenuItem(menuItems[i][j]));
    }
    menuBar.add(menu);
}

// Добавление меню
window.setJMenuBar(menuBar);

// Поле поиска
searchField = getSearchField();

// Кнопка поиска
searchButton = new JButton("Искать");
searchButton.setPreferredSize(new Dimension(100, 30));

// Добавляем поиск в шапку
searchArea = new JPanel();
searchArea.add(searchButton, BorderLayout.EAST);
searchArea.setBackground(new Color(211, 240, 228));
searchArea.add(searchField, BorderLayout.EAST);
header.add(searchArea, BorderLayout.EAST);

// Добавляем шапку на панель
panel.add(header, BorderLayout.NORTH);

// создаём футер
footer = new JPanel();
footer.setBackground(new Color(211, 240, 228));
footer.setLayout(new BorderLayout());

// Кнопка показа справки
showReference = new JButton("Справка");
showReference.setPreferredSize(new Dimension(100, 30));
footer.add(showReference, BorderLayout.WEST);

// Кнопка добавления строки в таблицу
add = new JButton("+");
del = new JButton("-");
editPanel = new JPanel();

```

```

editPanel.setBackground(new Color(211, 240, 228));
editPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 0, 0));
editPanel.add(del);
editPanel.add(add);

footer.add(editPanel, BorderLayout.EAST);
panel.add(footer, BorderLayout.SOUTH);

// Создаём табличку
String[] columnNames = {"ФИО ПАЦИЕНТА", "ФИО ВРАЧА", "ДАТА ЗАПИСИ",
"ВРЕМЯ ЗАПИСИ", "ЖАЛОБЫ"};
String[][] data = {"Годунов Е. А.", "Ответственный О. Е.",
"24.09.2024", "10:00", "Боль в горле"};
tableModel = new DefaultTableModel(data, columnNames);
table = new JTable(tableModel);

// Заполняем таблицу
for (int i = 0; i < 100; i++) {
    tableModel.addRow(new Object[]{"", "", "", "", ""});
}

// Оформляем таблицу
table.getTableHeader().setBackground(new Color(100, 180, 100)); //
зеленый заголовок
table.getTableHeader().setForeground(Color.WHITE);
table.setRowHeight(30);
table.getTableHeader().setReorderingAllowed(false); // Запрет менять
местами
table.setGridColor(new Color(19, 54, 11));
table.setSelectionBackground(new Color(216, 240, 211));
table.setSelectionForeground(new Color(4, 17, 3));

// Добавляем скролл
scrollPane = new JScrollPane(table);
panel.add(scrollPane, BorderLayout.CENTER);

// Добавление панели в основное окно
window.add(panel);

// Отображение окна
window.setVisible(true);
}

/**
 * Создаёт поле поиска с автоматически убирающимся текстом внутри
 * @return поле поиска
 */
private JTextField getSearchField() {
    searchField = new JTextField(15);
    searchField.setPreferredSize(new Dimension(100, 30));

    // Устанавливаем начальный текст как плейсхолдер
    placeholder = "🔍Поиск";
    searchField.setText(placeholder);
    searchField.setForeground(Color.GRAY);

    // Добавляем FocusListener для обработки placeholder
    searchField.addFocusListener(new FocusAdapter() {
        /**

```

```

        * Убирает плейсхолдер при фокусировании на поле
        * @param e событие для обработки
        */
@Override
public void focusGained(FocusEvent e) {
    if (searchField.getText().equals(placeholder)) {
        searchField.setText("");
        searchField.setForeground(Color.BLACK); // Обычный цвет
    }
}

/**
 * Добавляет плейсхолдер при расфокусировании из поля
 * @param e событие для обработки
 */
@Override
public void focusLost(FocusEvent e) {
    if (searchField.getText().isEmpty()) {
        searchField.setText(placeholder);
        searchField.setForeground(Color.GRAY); // Цвет
    }
}

});
return searchField;
}

public static void main(String[] args) {
    UserInterface userInterface = new UserInterface();
    userInterface.show();
}
}

```

Вывод

В результате выполнения работы были изучены правила построения экранной формы и спроектирован, и разработан графический интерфейс программы. Также были изучены библиотеки для создания GUI на языке java и получены практические навыки в программировании на этом языке.

Ссылки

https://github.com/DanyaMokhno/OOP_Labs/tree/main/com.study_oop.Laba_2