

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 Васильев Т.В.

Факультет КТИ

Группа № 3312

Подпись преподавателя \_\_\_\_\_

Санкт-Петербург

2024 г

# Цель работы

знакомство с правилами построения экранной формы.

## Макет экранной формы

Меню			X
<div>Выбор таблицы</div>			
Фамилия	Класс	Успеваемость	
Поиск		<div></div>	

Пример 1

Описание проверки работоспособности приложения и экранные формы, которые отображаются при запуске контрольного примера

Школьная информационная система

File

Ученики

Все классы

Фамилия	Класс	Все классы	Успеваемость
Иванов	5А	5А	Отличник
Петров	6Б	6Б	Двоечник
Сидоров	7В	7В	Хорошист
Кузнецов	5А		Хорошист
Смирнов	6Б		Отличник
Попов	7В		Двоечник

Поиск:  Найти

Пример 1

Школьная информационная система

File

New

Save file

Import

Export

Преподаватели и классы

Преподаватель	Классы
Сидоров	5А, 6Б
	6Б, 7В
	7В, 8А

Поиск:  Найти

Пример 2

## **Описание назначения экранной формы**

### **1. Основное окно (JFrame)**

Основное окно приложения, с заголовком “Школьная информационная система”, представляет форму, в которой располагаются все элементы интерфейса.

### **2. Выпадающий список JComboBox(выбор таблицы)**

Предназначен для выбора, какую таблицу данных нужно отобразить:

"Предметы и преподаватели"

"Преподаватели и классы"

"Ученики"

При выборе "Ученики" дополнительно отображается выпадающий список для выбора класса.

### **3. Таблицы данных Jtable**

Таблица "Предметы и преподаватели"

Отображает список предметов и соответствующих преподавателей, которые ведут эти предметы.

Столбцы: "Предмет" и "Преподаватель".

Таблица "Преподаватели и классы"

Показывает преподавателей и классы, которые они курируют.

Столбцы: "Преподаватель" и "Классы".

Таблица "Ученики"

Показывает список всех учеников, их классы и успеваемость.

Столбцы: "Фамилия", "Класс", "Успеваемость".

### **4. Панель меню JMenuBar и JMenu**

Панель меню содержит стандартные пункты, которые позволяют работать с файлами.

### **5. Панель поиска JTextField (текстовое поле) и JButton (кнопка "Найти")**

Визуальная панель, где пользователь может ввести текст для поиска информации по текущей таблице. Кнопка "Найти" предназначена для выполнения поиска по введённому запросу.

## Текст программы

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
/**
 * Класс создает экранную форму
 * @author Vasilyev Timofey 3312
 * @version 1.0
 */
public class SchoolApp extends JFrame {
    // Объявление графических компонентов
    private JTable subjectTable, classTable, studentTable;
    private JPanel tablePanel;
    private JComboBox<String> tableSelector, classSelector;
    private CardLayout cardLayout;
    private JTextField searchField;
    private JButton searchButton;

    // Данные для таблицы учеников и успеваемости
    private String[][] allStudentData = {
        {"Иванов", "5А", "Отличник"},
        {"Петров", "6Б", "Двоечник"},
        {"Сидоров", "7В", "Хорошист"},
        {"Кузнецов", "5А", "Хорошист"},
        {"Смирнов", "6Б", "Отличник"},
        {"Попов", "7В", "Двоечник"}
    };

    private String[] studentColumns = {"Фамилия", "Класс", "Успеваемость"};
    // Конструктор приложения
    public SchoolApp() {
        // Настройка заголовка окна
        super("Школьная информационная система");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(800, 600);
        setLayout(new BorderLayout());
        // Данные для таблицы предметов и преподавателей
        String[][] subjectData = {
            {"Математика", "Иванов"},
            {"Физика", "Петров"},
            {"Химия", "Сидоров"}
        };

        String[] subjectColumns = {"Предмет", "Преподаватель"};
        // Данные для таблицы классов и преподавателей
        String[][] classData = {
            {"Иванов", "5А, 6Б"},
            {"Петров", "6Б, 7В"},
            {"Сидоров", "7В, 8А"}
        };

        String[] classColumns = {"Преподаватель", "Классы"};
```

```

// Создание таблицы
subjectTable = new JTable(subjectData, subjectColumns);
classTable = new JTable(classData, classColumns);
studentTable = new JTable(allStudentData, studentColumns);
// Создание панели с карточной компоновкой
cardLayout = new CardLayout();
tablePanel = new JPanel(cardLayout);
tablePanel.add(new JScrollPane(subjectTable), "Предметы и преподаватели");
tablePanel.add(new JScrollPane(classTable), "Преподаватели и классы");
tablePanel.add(new JScrollPane(studentTable), "Ученики");
// Выпадающий список для выбора таблицы
tableSelector = new JComboBox<>(new String[]{"Предметы и преподаватели",
"Преподаватели и классы", "Ученики"});
tableSelector.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Переключение видимой таблицы на основе выбора пользователя
        String selectedTable = (String) tableSelector.getSelectedItem();
        cardLayout.show(tablePanel, selectedTable);
        if (selectedTable.equals("Ученики"))
        {
            classSelector.setVisible(true);
        }
        else
        {
            classSelector.setVisible(false);
        }
    }
});
// Выпадающий список для выбора класса
classSelector = new JComboBox<>(new String[]{"Все классы", "5А", "6Б",
"7В"});
classSelector.setVisible(false);
// Добавление панели с выпадающим меню File
JMenuBar menuBar = new JMenuBar();
JMenu fileMenu = new JMenu("File");

// Пункты меню
JMenuItem newItem = new JMenuItem("New");
JMenuItem saveItem = new JMenuItem("Save file");
JMenuItem importItem = new JMenuItem("Import");
JMenuItem exportItem = new JMenuItem("Export");
// Добавление пунктов меню в "File"
fileMenu.add(newItem);
fileMenu.add(saveItem);
fileMenu.add(importItem);
fileMenu.add(exportItem);
// Добавление меню в панель меню
menuBar.add(fileMenu);
// Установка меню в окно
setJMenuBar(menuBar);
// Панель для поиска
JPanel searchPanel = new JPanel();
searchPanel.setLayout(new FlowLayout());
JLabel searchLabel = new JLabel("Поиск:");

```

```
searchField = new JTextField(20);
searchButton = new JButton("Найти");
// Добавление компонентов панели поиска
searchPanel.add(searchLabel);
searchPanel.add(searchField);
searchPanel.add(searchButton);
// Добавление компонентов в основное окно
JPanel topPanel = new JPanel();
topPanel.setLayout(new FlowLayout());
topPanel.add(tableSelector);
topPanel.add(classSelector);
add(topPanel, BorderLayout.NORTH);
add(tablePanel, BorderLayout.CENTER);
add(searchPanel, BorderLayout.SOUTH);
}
// Метод для запуска приложения
public static void main(String[] args)
{
    SchoolApp app = new SchoolApp();
    app.setVisible(true);
}
}
```