

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 № 4
«Обработка исключений»
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 механизмом обработки исключений в языке Java.

Перечень ситуаций, которые контролируются с помощью исключений

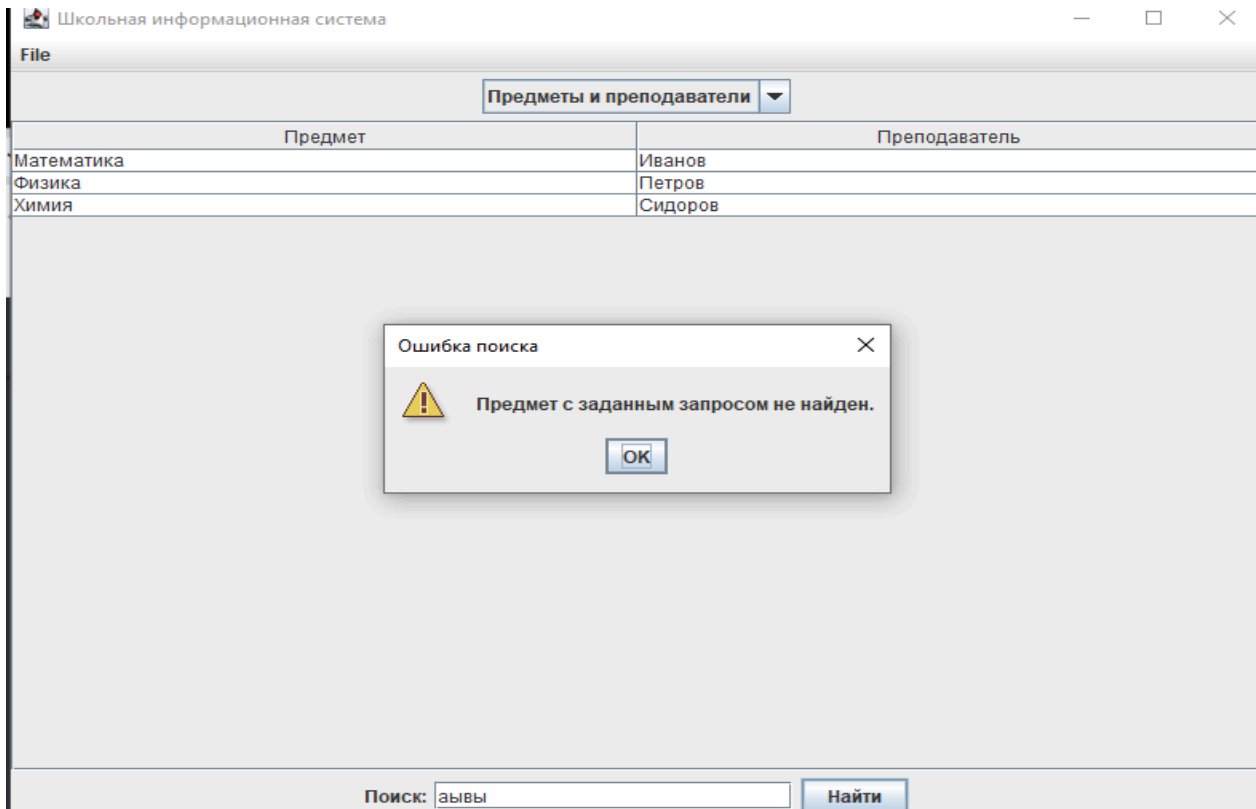
- Ошибка поиска данных:

Пользователь нажимает кнопку поиска, вводя запрос, который не соответствует ни одному студенту в таблице. Пользователю выводится уведомление о том, что данные по запросу не найдены. Это предупреждает пользователя и позволяет ему изменить критерии поиска

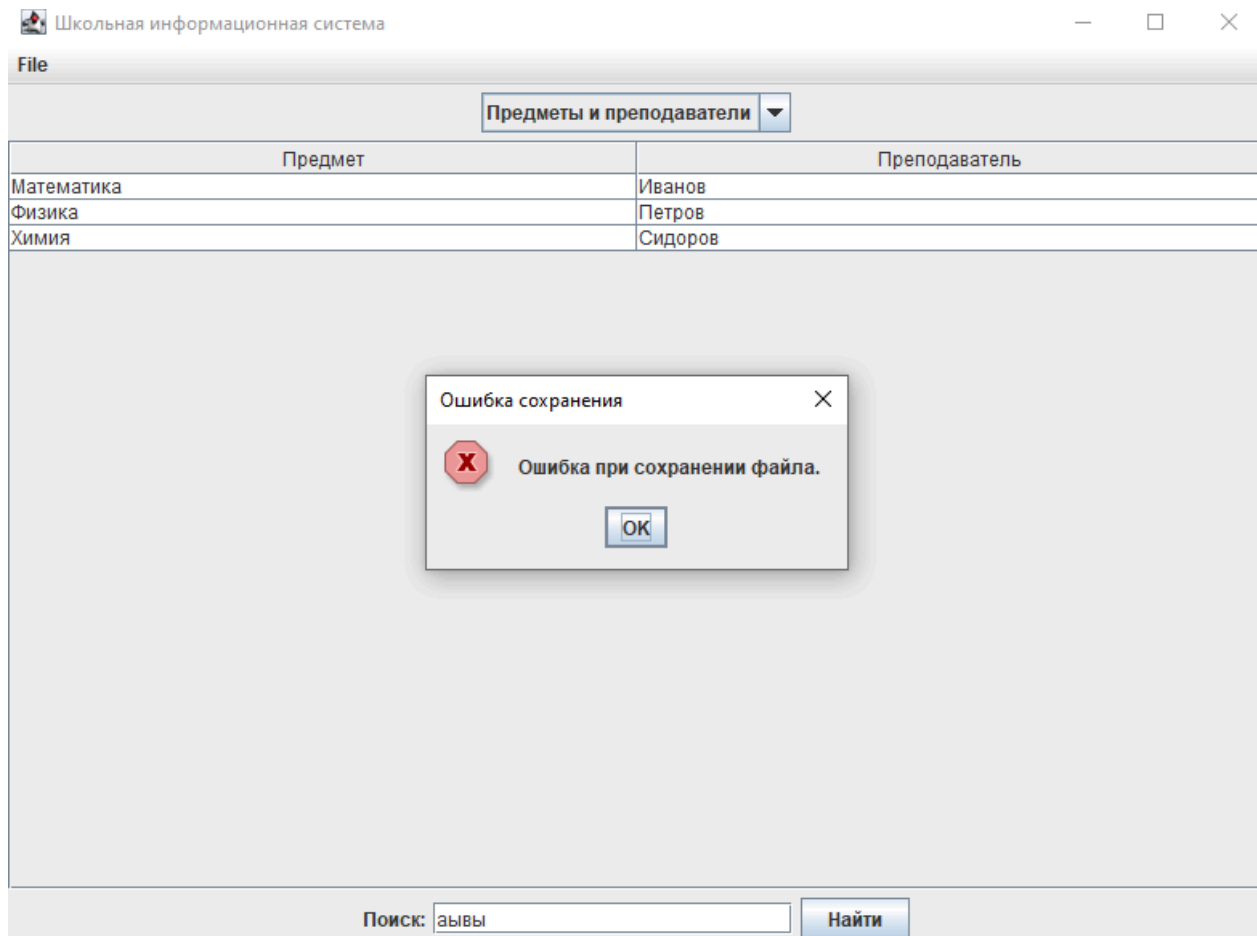
- Ошибка сохранения файла :

Отображается диалоговое окно с сообщением "Ошибка при сохранении файла". Это уведомляет пользователя о возникшей проблеме и предлагает повторить операцию или выбрать другой путь для сохранения.

Пример работы обработчиков ситуаций



Пример 1 - ошибка поиска



Пример 2 - ошибка сохранения файла

Текст программы

```
package lab4;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
/**
 * Класс создает экранную форму
 * @version 1.1
 */
public class SchoolApp extends JFrame {
    private JTable subjectTable, classTable, studentTable;
    private JPanel tablePanel;
    private JComboBox<String> tableSelector, classSelector;
    private CardLayout cardLayout;
```

```

private JTextField searchField;
private JButton searchButton;
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"};
private String[][] originalStudentData;
private String[][] subjectData = {
    {"Математика", "Иванов"},
    {"Физика", "Петров"},
    {"Химия", "Сидоров"}
};

private String[] subjectColumns = {"Предмет", "Преподаватель"};

private String[][] classData = {
    {"Иванов", "5А, 6Б"},
    {"Петров", "6Б, 7В"},
    {"Сидоров", "7В, 8А"}
};

private String[] classColumns = {"Преподаватель", "Классы"};
public SchoolApp() {
    super("Школьная информационная система");
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setSize(800, 600);
    setLayout(new BorderLayout());
    originalStudentData = allStudentData.clone();
    subjectTable = new JTable(subjectData, subjectColumns);
    classTable = new JTable(classData, classColumns);
    studentTable = new JTable(allStudentData, studentColumns);
    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");
    tableSelector = new JComboBox<>(new String[]{"Предметы и преподаватели",
"Преподаватели и классы", "Ученики"});
    tableSelector.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String selectedTable = (String) tableSelector.getSelectedItem();
            cardLayout.show(tablePanel, selectedTable);
            classSelector.setVisible(selectedTable.equals("Ученики"));
        }
    });
    classSelector = new JComboBox<>(new String[]{"Все классы", "5А", "6Б",
"7В"});
    classSelector.setVisible(false);
    JMenuBar menuBar = new JMenuBar();
    JMenu fileMenu = new JMenu("File");
    JMenuItem newItem = new JMenuItem("New");

```

```

JMenuItem saveItem = new JMenuItem("Save file");
JMenuItem importItem = new JMenuItem("Import");
JMenuItem exportItem = new JMenuItem("Export");
fileMenu.add(newItem);
fileMenu.add(saveItem);
fileMenu.add(importItem);
fileMenu.add(exportItem);
menuBar.add(fileMenu);
setJMenuBar(menuBar);
JPanel searchPanel = new JPanel();
searchPanel.setLayout(new FlowLayout());
JLabel searchLabel = new JLabel("Поиск:");
searchField = new JTextField(20);
searchButton = new JButton("Найти");
searchPanel.add(searchLabel);
searchPanel.add(searchField);
searchPanel.add(searchButton);
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);
searchButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        handleSearch();
    }
});
saveItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        handleFileSave();
    }
});
}

private void handleSearch() {
    String query = searchField.getText().toLowerCase();
    String selectedTable = (String) tableSelector.getSelectedItem();

    try {
        if (selectedTable.equals("Ученики")) {
            searchInStudentTable(query);
        } else if (selectedTable.equals("Предметы и преподаватели")) {
            searchInSubjectTable(query);
        } else if (selectedTable.equals("Преподаватели и классы")) {
            searchInClassTable(query);
        }
    } catch (SearchException e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Ошибка поиска",
JOptionPane.WARNING_MESSAGE);
    }
}
}

```

```

private void searchInStudentTable(String query) throws SearchException {
    ArrayList<String[]> filteredData = new ArrayList<>();
    for (String[] student : originalStudentData) {
        for (String field : student) {
            if (field.toLowerCase().contains(query)) {
                filteredData.add(student);
                break;
            }
        }
    }
    if (filteredData.isEmpty()) {
        throw new SearchException("Ученик с заданным запросом не найден.");
    }
    String[][] newStudentData = filteredData.toArray(new String[0][]);
    studentTable.setModel(new javax.swing.table.DefaultTableModel(newStudentData,
studentColumns));
}

private void searchInSubjectTable(String query) throws SearchException {
    ArrayList<String[]> filteredData = new ArrayList<>();
    for (String[] subject : subjectData) {
        for (String field : subject) {
            if (field.toLowerCase().contains(query)) {
                filteredData.add(subject);
                break;
            }
        }
    }
    if (filteredData.isEmpty()) {
        throw new SearchException("Предмет с заданным запросом не найден.");
    }
    String[][] newSubjectData = filteredData.toArray(new String[0][]);
    subjectTable.setModel(new javax.swing.table.DefaultTableModel(newSubjectData,
subjectColumns));
}

private void searchInClassTable(String query) throws SearchException {
    ArrayList<String[]> filteredData = new ArrayList<>();
    for (String[] classRow : classData) {
        for (String field : classRow) {
            if (field.toLowerCase().contains(query)) {
                filteredData.add(classRow);
                break;
            }
        }
    }
    if (filteredData.isEmpty()) {
        throw new SearchException("Класс с заданным запросом не найден.");
    }
    String[][] newClassData = filteredData.toArray(new String[0][]);
    classTable.setModel(new javax.swing.table.DefaultTableModel(newClassData,
classColumns));
}

private void handleFileSave() {
    JFileChooser fileChooser = new JFileChooser();
    int option = fileChooser.showSaveDialog(SchoolApp.this);
}

```

```

        if (option == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            try {
                saveToFile(file);
            } catch (FileSaveException e) {
                JOptionPane.showMessageDialog(this, e.getMessage(), "Ошибка
сохранения", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private void saveToFile(File file) throws FileSaveException {
        try (FileWriter writer = new FileWriter(file)) {
            for (String column : studentColumns) {
                writer.write(column + "\t");
            }
            writer.write("\n");
            for (String[] row : allStudentData) {
                for (String data : row) {
                    writer.write(data + "\t");
                }
                writer.write("\n");
            }
            JOptionPane.showMessageDialog(this, "Данные успешно сохранены.",
"Сохранение", JOptionPane.INFORMATION_MESSAGE);
        } catch (IOException ex) {
            throw new FileSaveException("Ошибка при сохранении файла.");
        }
    }

    public static void main(String[] args) {
        SchoolApp app = new SchoolApp();
        app.setVisible(true);
    }
}

```

<https://github.com/lovushker/JavaLabs/tree/main/lab4>