

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

Лабораторная работа по ООП №5 по теме “Сохранение и загрузка данных из файла”

Выполнил: Цацкис Артём гр. 3311

Проверил: Павловский М. Г.

Подпись преподавателя: _____

Санкт-Петербург

2024

Цель работы:

Знакомство с организацией обмена данными между объектами экранной формы и файлом.

Описание задания:

1. Создать проект, который дублирует код лабораторной работы №4
2. Напишите и добавьте в проект обработчики кнопок загрузки текста в файл и выгрузки из него.
3. Загрузите данные в экранную форму приложения.
4. Внесите изменения в загруженные данные и сохраните их в файле.

Макет созданный в Balsamiq Mockups:

Информация о книгах			
Название книги	Автор	Шрифт	закреплена?
Война и мир	Лев Толстой	Arial	Да
1984	Джордж Оруэлл	Calibri	Нет
Прощай оружие!	Эрнест Хемингуэй	Garamond	Да
Убить пересмешника	Харпер Ли	Fraktur	Да
На дороге	Джек Керуак	Papyrus	Нет

СохранитьДобавитьРедактироватьУдалить

Пример выполнения программы:

1.Содержание файла

```
Война и мир,Лев Толстой,Arial,Нет
1984,Джордж Оруэлл,Calibri,Да
Прощай оружие!,Эрнест Хемингуэй,Garamond,Нет
Убить пересмешника,Харпер Ли,Fraktur,Да
На дороге,Джек Керуак,Papyrus,Нет
Преступление и наказание, Достоевский, Calibri, Да|
```

2.После загрузки файла

Информация о книгах			
Название книги	Автор	Шрифт	Закреплена?
Война и мир	Лев Толстой	Arial	Нет
1984	Джордж Оруэлл	Calibri	Да
Прощай оружие!	Эрнест Хемингуэй	Garamond	Нет
Убить пересмешника	Харпер Ли	Fraktur	Да
На дороге	Джек Керуак	Papyrus	Нет
Преступление и наказан...	Достоевский	Calibri	Да

Сохранить Загрузить Добавить Редактировать Удалить

Код программы:

```
package lab5;

import java.io.*;
import java.awt.BorderLayout; // Добавлен импорт
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class OOPlab5 {
    private JFrame bookList;
    private DefaultTableModel model;
    private JButton save, add, edit, delete, load;
    private JScrollPane scroll;
    private JTable books;
```

```

public void show() {
    bookList = new JFrame("Информация о книгах");
    bookList.setSize(600, 400);
    bookList.setLocation(100, 100);
    bookList.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    String[] columns = {"Название книги", "Автор", "Шрифт", "Закреплена?"};
    String[][] data = {
        {"Война и мир", "Лев Толстой", "Arial", "Нет"},
        {"1984", "Джордж Оруэлл", "Calibri", "Да"},
        {"Прощай оружие!", "Эрнест Хемингуэй", "Garamond", "Нет"},
        {"Убить пересмешника", "Харпер Ли", "Fraktur", "Да"},
        {"На дороге", "Джек Керуак", "Papyrus", "Нет"}
    };
    model = new DefaultTableModel(data, columns);
    books = new JTable(model);
    books.setAutoCreateRowSorter(true);
    scroll = new JScrollPane(books);
    bookList.getContentPane().add(scroll, BorderLayout.CENTER);

    JPanel buttonPanel = new JPanel();
    save = new JButton("Сохранить");
    load = new JButton("Загрузить");
    add = new JButton("Добавить");
    edit = new JButton("Редактировать");
    delete = new JButton("Удалить");

    buttonPanel.add(save);
    buttonPanel.add(load);
    buttonPanel.add(add);
    buttonPanel.add(edit);
    buttonPanel.add(delete);
    bookList.getContentPane().add(buttonPanel, BorderLayout.SOUTH);

    save.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            saveToFile();
            JOptionPane.showMessageDialog(bookList, "Данные сохранены.");
        }
    });

    load.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            loadFromFile();
            JOptionPane.showMessageDialog(bookList, "Данные загружены.");
        }
    });

    add.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(bookList, "Добавлена новая книга.");
        }
    });

    edit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                int selectedRow = books.getSelectedRow();
                editBook(selectedRow, "Новое название", "Новый автор", "Новый
шрифт", "Нет");
            }
        }
    });
}

```

```

        JOptionPane.showMessageDialog(bookList, "Книга
отредактирована.");
    } catch (InvalidBookOperationException ex) {
        JOptionPane.showMessageDialog(bookList, ex.getMessage(), "Ошибка
редактирования", JOptionPane.ERROR_MESSAGE);
    }
}

delete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            int selectedRow = books.getSelectedRow();
            deleteBook(selectedRow);
            JOptionPane.showMessageDialog(bookList, "Книга удалена.");
        } catch (BookDeletionException ex) {
            JOptionPane.showMessageDialog(bookList, ex.getMessage(), "Ошибка
удаления", JOptionPane.ERROR_MESSAGE);
        }
    }
});

bookList.setVisible(true);
}

private void saveToFile() {
    try (BufferedWriter writer = new BufferedWriter(new
FileWriter("books_data.csv"))) {
        for (int i = 0; i < model.getRowCount(); i++) {
            for (int j = 0; j < model.getColumnCount(); j++) {
                writer.write(model.getValueAt(i, j).toString()); // Используем
toString() для приведения к строке
                if (j < model.getColumnCount() - 1) {writer.write(",");
                }
            }
            writer.newLine();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private void loadFromFile() {
    try (BufferedReader reader = new BufferedReader(new
FileReader("books_data.csv"))) {
        String line;
        model.setRowCount(0); // Очистка текущих данных
        while ((line = reader.readLine()) != null) {
            String[] rowData = line.split(",");
            model.addRow(rowData);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public void editBook(int rowIndex, String title, String author, String font,
String pinned) throws InvalidBookOperationException {
    if (rowIndex < 0 || rowIndex >= model.getRowCount()) {
        throw new InvalidBookOperationException("Не выбрана книга");
    }
    model.setValueAt(title, rowIndex, 0);
}

```

```

        model.setValueAt(author, rowIndex, 1);
        model.setValueAt(font, rowIndex, 2);
        model.setValueAt(pinned, rowIndex, 3);
    }

    public void deleteBook(int rowIndex) throws BookDeletionException {
        if (rowIndex < 0 || rowIndex >= model.getRowCount()) {
            throw new BookDeletionException("Не выбрана книга");
        }
        model.removeRow(rowIndex);
    }

    public static void main(String[] args) {
        new OOPlab5().show();
    }
}

class InvalidBookOperationException extends Exception {
    public InvalidBookOperationException(String message) {
        super(message);
    }
}

class BookDeletionException extends Exception {
    public BookDeletionException(String message) {
        super(message);
    }
}

```

Ссылка на репозиторий Git-hub:

[TsatskisArtem/For-OOP5](#)

OOPlab5.java-код программы

Dos- папка с документацией сгенерированная javadocs

Books_data.csv- файл с данными

Tsatskis-lab5.pdf-отчет по лабораторной работе