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 № 5**

**«Сохранение и загрузка данных из файла»**

**по дисциплине «Объектно - ориентированное программирование на языке Java»**

Выполнил Баймухамедов Р. Р.

Факультет КТИ

Группа № 3311

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024 г

**Цель работы**

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

**Репозиторий**

[https://github.com/brick1ng5654/course-2/tree/main/OOP/lab\_0](https://github.com/brick1ng5654/course-2/tree/main/OOP/lab_02)5

В этом репозитории находятся исходные файлы проекта:

В /src/edu/java/lab05 находится файл CinemaList.java

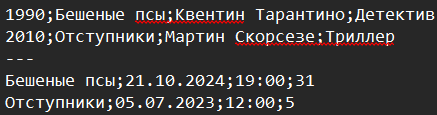
В /doc находится документация, сгенерированная JavaDoc

Ссылка на видеоотчёт

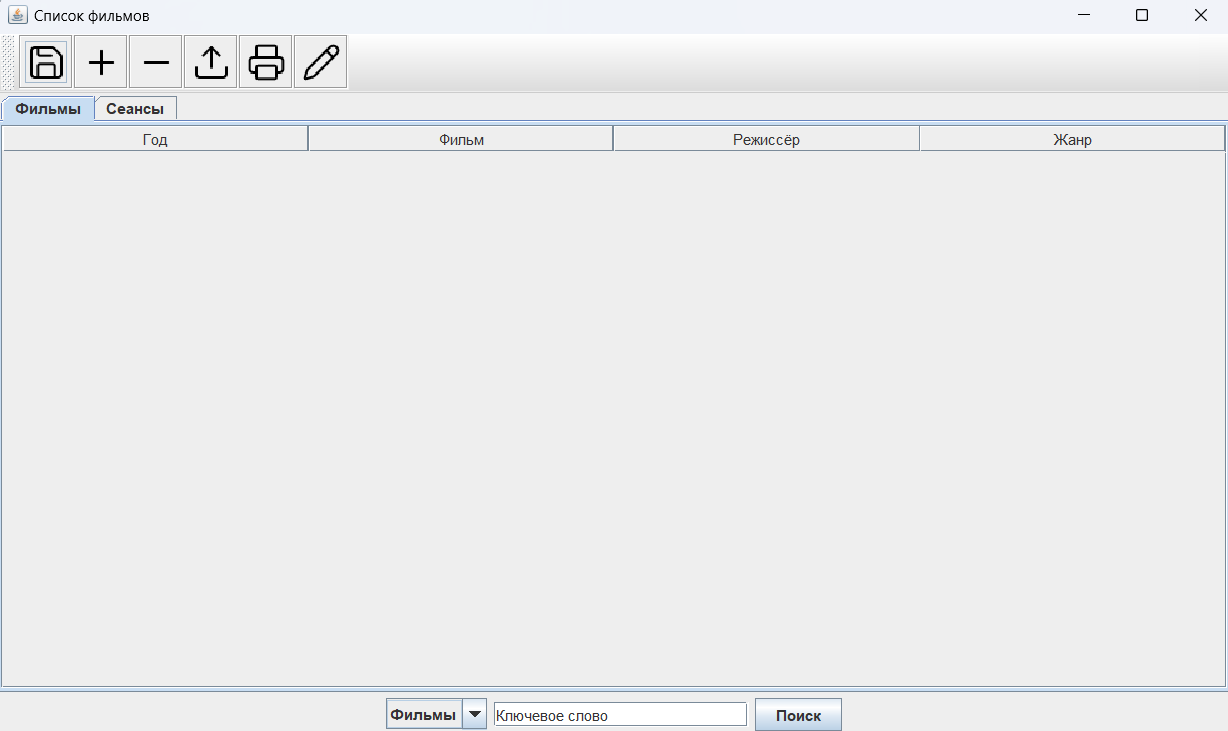
ht

**Распечатки содержимого файлов с данными до и после внесения изменений**

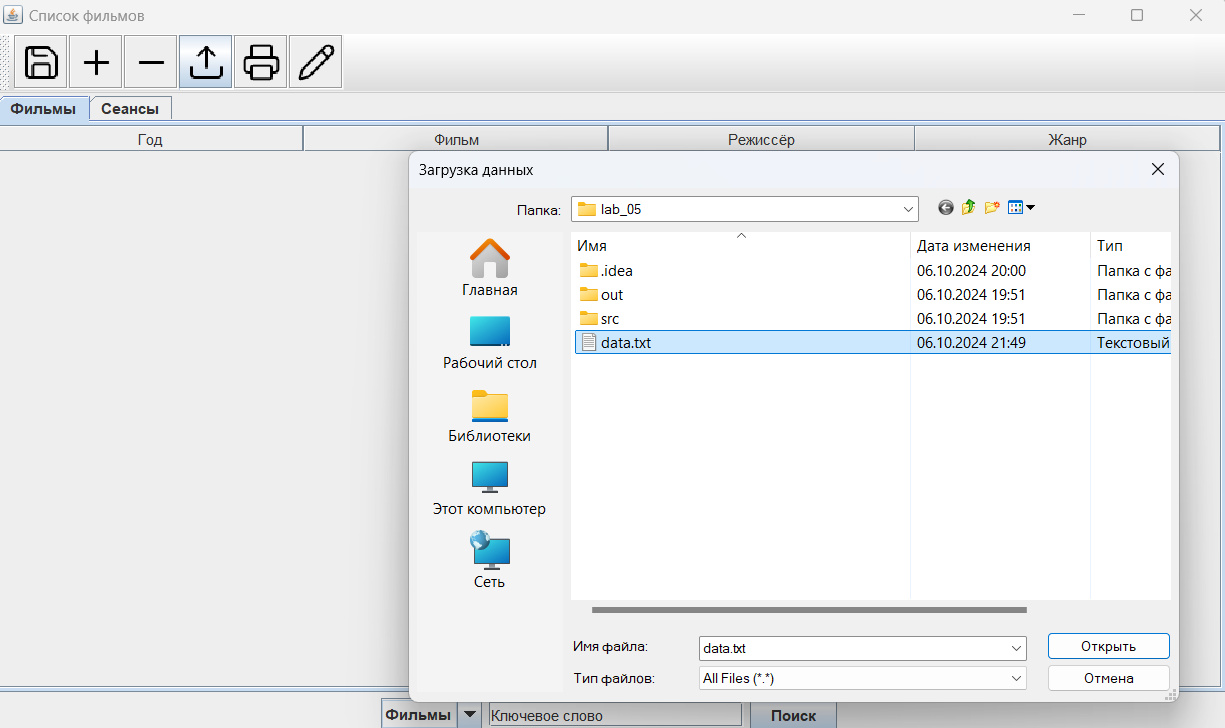
Процесс загрузки файла, внесения в него изменения и сохранения файла показан ниже. Также продемонстрирован исходный файл с данными до и после работы программы.



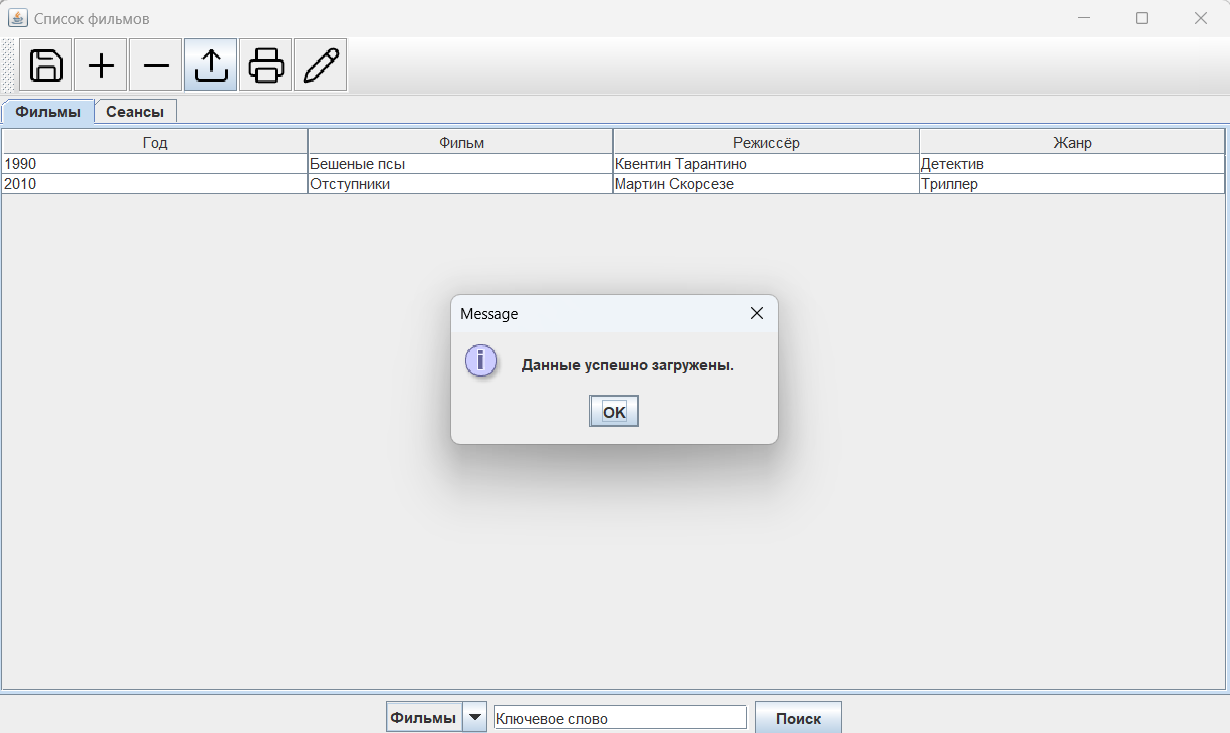
Файл data.txt до работы программы



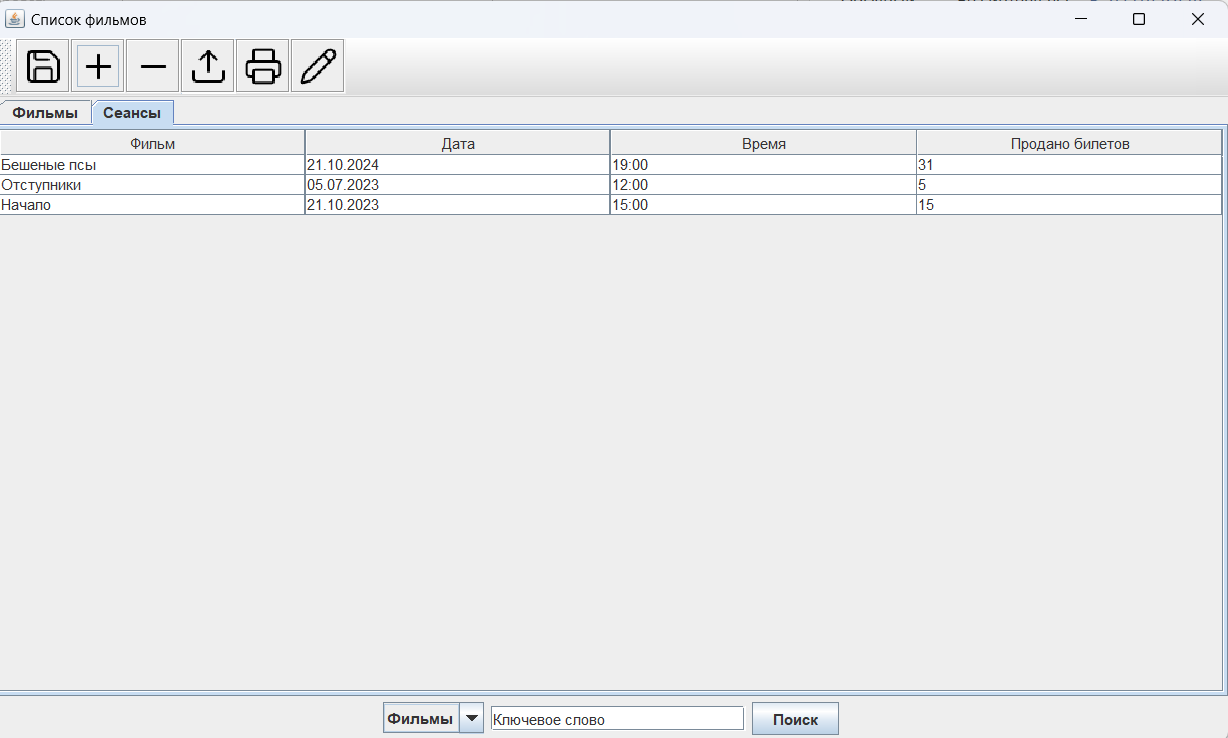
Программа без загрузки данных



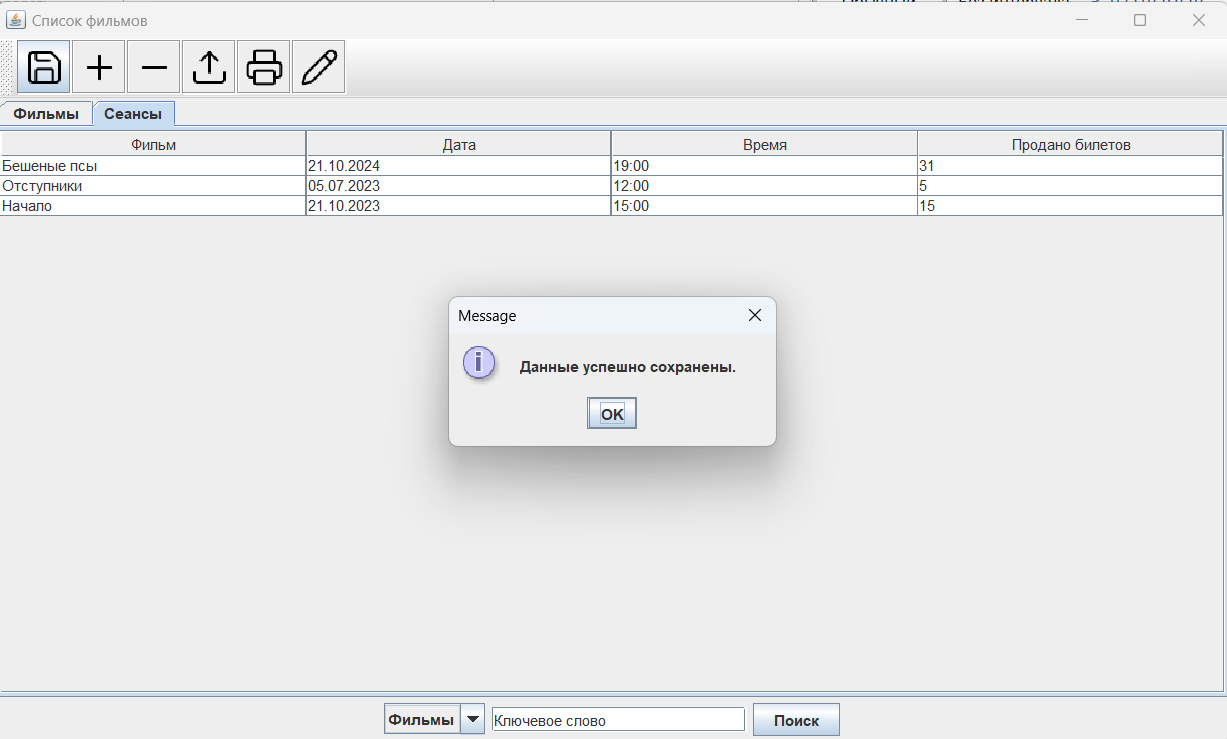
Выбор файла с данными



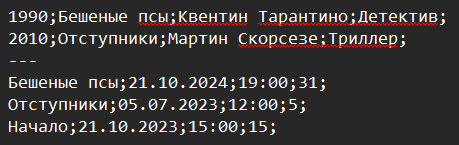
Данные успешно загружены



Внесены новые данные



Сохранение новых данных



Файл data.txt после работы программы

**Фрагменты кода, отвечающие за сохранение и чтение данных из файла**

save.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 saveToFile();  
 }  
});

upload.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 loadFromFile();  
 }  
});

private void saveToFile() {  
 // Открытие диалогового окна для выбора файла  
 FileDialog saveDialog = new FileDialog(cinemaList, "Сохранение данных", FileDialog.*SAVE*);  
 saveDialog.setFile("\*.txt");  
 saveDialog.setVisible(true);  
  
 String directory = saveDialog.getDirectory();  
 String filename = saveDialog.getFile();  
 if (directory == null || filename == null) return; // Пользователь нажал "Отмена"  
  
 String filePath = directory + filename;  
  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {  
 // Сохранение данных из таблицы фильмов  
 for (int i = 0; i < modelMovies.getRowCount(); i++) {  
 for (int j = 0; j < modelMovies.getColumnCount(); j++) {  
 writer.write(modelMovies.getValueAt(i, j).toString());  
 writer.write(";"); // Используем табуляцию как разделитель  
 }  
 writer.newLine(); // Переход на новую строку  
 }  
 // Добавляем разделитель между таблицами  
 writer.write("---");  
 writer.newLine();  
 // Сохранение данных из таблицы сеансов  
 for (int i = 0; i < modelSessions.getRowCount(); i++) {  
 for (int j = 0; j < modelSessions.getColumnCount(); j++) {  
 writer.write(modelSessions.getValueAt(i, j).toString());  
 writer.write(";");  
 }  
 writer.newLine();  
 }  
 JOptionPane.*showMessageDialog*(cinemaList, "Данные успешно сохранены.");  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(cinemaList, "Ошибка при сохранении файла.");  
 }  
}  
  
private void loadFromFile() {  
 FileDialog loadDialog = new FileDialog(cinemaList, "Загрузка данных", FileDialog.*LOAD*);  
 loadDialog.setFile("\*.txt");  
 loadDialog.setVisible(true);  
  
 String directory = loadDialog.getDirectory();  
 String filename = loadDialog.getFile();  
 if (directory == null || filename == null) return; // Пользователь нажал "Отмена"  
  
 String filePath = directory + filename;  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {  
 // Очистка таблиц перед загрузкой  
 modelMovies.setRowCount(0);  
 modelSessions.setRowCount(0);  
  
 String line;  
 boolean isSessionsTable = false;  
 while ((line = reader.readLine()) != null) {  
 if (line.equals("---")) {  
 isSessionsTable = true; // Переходим к таблице сеансов  
 continue;  
 }  
 String[] data = line.split(";");  
 if (!isSessionsTable) {  
 modelMovies.addRow(data);  
 } else {  
 modelSessions.addRow(data);  
 }  
 }  
 JOptionPane.*showMessageDialog*(cinemaList, "Данные успешно загружены.");  
 } catch (FileNotFoundException ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(cinemaList, "Файл не найден.");  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(cinemaList, "Ошибка при чтении файла.");  
 }  
}