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 № 7, 8**

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

Выполнил Баймухамедов Р. Р.

Преподаватель Павловский М.Г

Факультет КТИ

Группа № 3311

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

Санкт-Петербург

2024 г

**Цель работы**

Знакомство со способами формирования отчетов с использованием конструктора Jaspersoft iReport Designert и правилами построения параллельных приложений в языке Java.

**Описание задания**

* Распечатка исходного XML – файла.
* Исходный и отредактированный XML-файл.
* Скриншоты, иллюстрирующие построение шаблона в дизайнере iReport.
* Распечатка сгенерированных файлов.
* Текст документации, сгенерированный Javadoc.
* Исходный текст построения отчета.
* Фрагмент кода, отвечающие за организацию параллельной работы трех потоков.
* Фрагмент кода, отвечающие за сохранения данных в pdf и html форматах.

**Репозиторий**

<https://github.com/brick1ng5654/course-2/tree/main/OOP/lab_09>

В этом репозитории находятся исходные файлы проекта:

В /src/edu/java/lab09 находится файл CinemaList.java

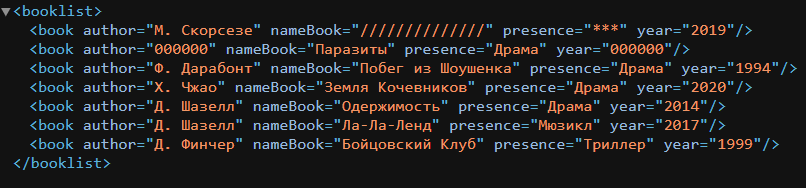
В /doc находится документация, сгенерированная JavaDoc

Ссылка на видеоотчёт

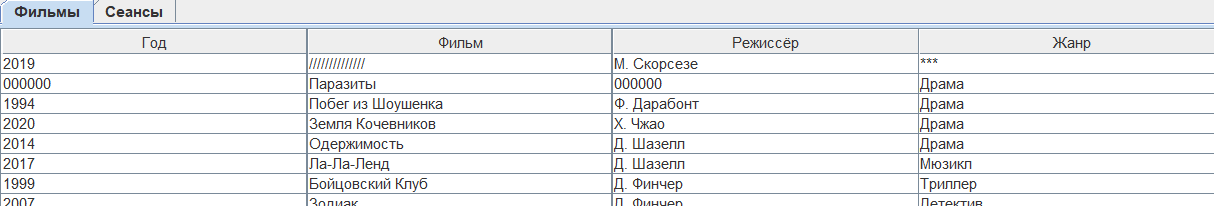
https://youtu.be/

**Описание и проверка работоспособности приложения**

Исходные данные:

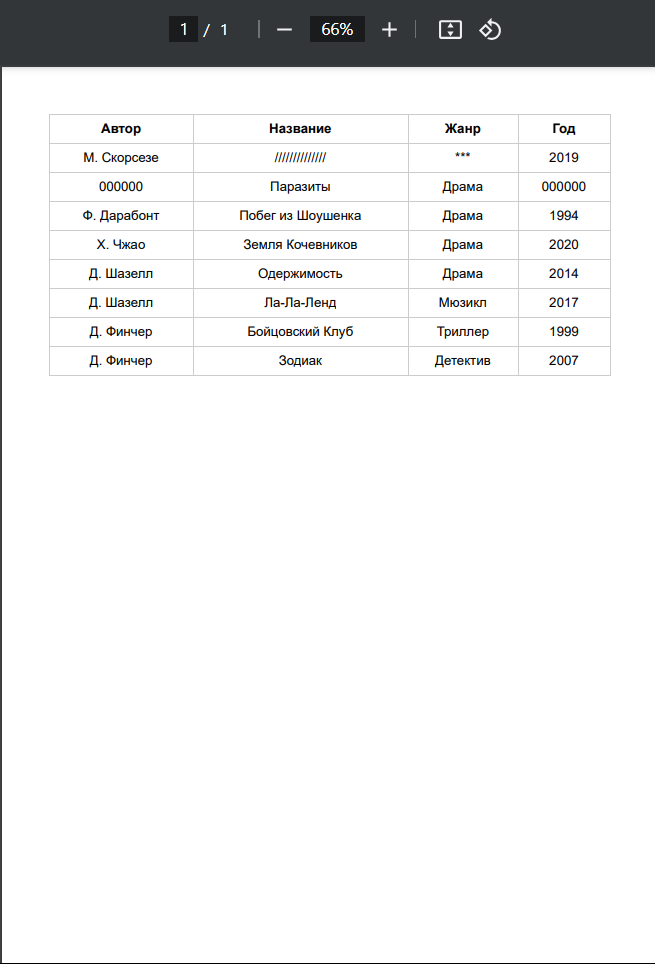


К данным, которые загрузились будет добавлена строка (“2007”, “Зодиак”, “Д. Финчер”, “Детектив”)



В исходном текстовом файле данной строки нет

Все данные сохраняются в PDF и HTML файл movienew.pdf и movienew.html соответственно



*Фрагмент кода, отвечающий за работу трёх потоков*

// Запуск потоков  
new tthread(1, modelMovies, tableMovies).start();  
new tthread(2, modelMovies, tableMovies).start();  
new tthread(3, modelMovies, tableMovies).start();

class tthread extends Thread {  
 private int type;  
 private static final Object *shared* = new Object();  
 private DefaultTableModel tableModel;  
 private JTable table;  
  
 public tthread(int i, DefaultTableModel tableModel, JTable table) {  
 type = i;  
 this.tableModel = tableModel;  
 this.table = table;  
 }  
  
 public void run() {  
 if (type == 1) {  
 synchronized (*shared*) {  
 try {  
 *shared*.wait();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 try {  
 xmlOpen q = new xmlOpen("Открыть данные XML", tableModel, table);  
 } catch (Exception e1) {  
 e1.printStackTrace();  
 }  
 }  
 }  
  
 if (type == 2) {  
 synchronized (*shared*) {  
 *shared*.notifyAll();  
 try {  
 *shared*.wait();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 tableModel.addRow(new String[]{"2007", "Зодиак", "Д. Финчер", "Детектив"});  
  
 *shared*.notifyAll();  
 }  
 }  
  
 if (type == 3) {  
 synchronized (*shared*) {  
 *shared*.notifyAll();  
 try {  
 *shared*.wait();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 pdfSave qwe = new pdfSave(tableModel);  
 htmlSave qwe2 = new htmlSave(tableModel);  
 }  
 }  
 }

}

**Фрагмент кода, отвечающий за сохранение данных в PDF и HTML форматах**

*Сохранение в PDF*

private void saveToPDF() {  
 FileDialog savePDFDialog = new FileDialog(cinemaList, "Сохранение данных в PDF", FileDialog.*SAVE*);  
 savePDFDialog.setFile("\*.pdf");  
 savePDFDialog.setVisible(true);  
  
 String directory = savePDFDialog.getDirectory();  
 String filename = savePDFDialog.getFile();  
 if (directory == null || filename == null) return; // Пользователь нажал "Отмена"  
  
 String filePath = directory + filename;  
  
 Document document = new Document(PageSize.A4, 50, 50, 50, 50);  
 PdfPTable t = new PdfPTable(4);  
  
 try {  
 PdfWriter writer = PdfWriter.getInstance(document, new FileOutputStream(filePath));  
 } catch (FileNotFoundException | DocumentException e) {  
 e.printStackTrace();  
 }  
  
 BaseFont bfComic = null;  
 try {  
 bfComic = BaseFont.createFont("src/fonts/Arial.ttf", BaseFont.IDENTITY\_H, BaseFont.EMBEDDED);  
 } catch (DocumentException | IOException e1) {  
 e1.printStackTrace();  
 }  
  
 Font font1 = new Font(bfComic, 12);  
  
 t.addCell(new PdfPCell(new Phrase("Год", font1)));  
 t.addCell(new PdfPCell(new Phrase("Фильм", font1)));  
 t.addCell(new PdfPCell(new Phrase("Режиссёр", font1)));  
 t.addCell(new PdfPCell(new Phrase("Жанр", font1)));  
  
 for (int i = 0; i < modelMovies.getRowCount(); i++) {  
 t.addCell(new Phrase((String) modelMovies.getValueAt(i, 0), font1));  
 t.addCell(new Phrase((String) modelMovies.getValueAt(i, 1), font1));  
 t.addCell(new Phrase((String) modelMovies.getValueAt(i, 2), font1));  
 t.addCell(new Phrase((String) modelMovies.getValueAt(i, 3), font1));  
 }  
  
 document.open();  
  
 try {  
 document.add(t);  
 } catch (DocumentException e) {  
 e.printStackTrace();  
 }  
  
 document.close();  
 JOptionPane.*showMessageDialog*(cinemaList, "Данные успешно сохранены в PDF.");  
}

*Сохранение в HTML*

private void saveToHTML() {  
 FileDialog saveHTMLDialog = new FileDialog(cinemaList, "Сохранение данных в HTML", FileDialog.*SAVE*);  
 saveHTMLDialog.setFile("\*.html");  
 saveHTMLDialog.setVisible(true);  
  
 String directory = saveHTMLDialog.getDirectory();  
 String filename = saveHTMLDialog.getFile();  
 if (directory == null || filename == null) return; // Пользователь нажал "Отмена"  
  
 String filePath = directory + filename;  
  
 PrintWriter pw = null;  
 try {  
 pw = new PrintWriter(new FileWriter(filePath));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 pw.println("<TABLE BORDER><TR><TH>Год<TH>Фильм<TH>Режиссёр<TH>Жанр</TR>");  
 for (int i = 0; i < modelMovies.getRowCount(); i++) {  
 pw.println("<TR><TD>" + (String) modelMovies.getValueAt(i, 0)  
 + "<TD>" + (String) modelMovies.getValueAt(i, 1)  
 + "<TD>" + (String) modelMovies.getValueAt(i, 2)  
 + "<TD>" + (String) modelMovies.getValueAt(i, 3));  
 }  
 pw.println("</TABLE>");  
 pw.close();  
 JOptionPane.*showMessageDialog*(cinemaList, "Данные успешно сохранены в HTML.");  
}

Вывод

Во время выполнения данной лабораторной работы, были получены навыки работы с технологией формирования отчетов с использованием конструктора Jaspersoft iReport Designer и правилами построения параллельных приложений в языке Java. В пункте «Описание и проверки работоспособности приложения» ошибок выявлено не было.