**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ВТ**

отчет

**по лабораторной работе №7**

**по дисциплине «Объектно-ориентированное программирование»**

Тема: **Построение отчетов в PDF и HTML форматах**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3315 |  | Красавцев Н.В. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

2024

**Цель работы.**

Знакомство со способами формирования отчетов с использованием конструктора Jaspersoft iReport Designert.

**Описание задания.**

Разработать и включить в проект метод построения отчета в PDF-формате для первого варианта шаблона и в HTML-формате для второго.

**Код класса.**

private void saveToPDF() {

String filePath = "C:/lab7/Emlpoyee.pdf";

try {

// Создаем объект для записи в PDF

PdfWriter writer = new PdfWriter(new FileOutputStream(filePath));

PdfDocument pdf = new PdfDocument(writer);

Document document = new Document(pdf);

// Заголовок документа

document.add(new Paragraph("Employee list"));

document.add(new Paragraph("\n"));

// Создаем таблицу с 4 столбцами

Table table = new Table(3);

// Добавляем заголовки столбцов в таблицу

table.addCell("Surname");

table.addCell("Name");

table.addCell("Surname2");

// Получаем данные из модели таблицы и заполняем таблицу

DefaultTableModel model = (DefaultTableModel) books.getModel();

for (int i = 0; i < model.getRowCount(); i++) {

// Добавляем данные по каждому сотруднику (по строкам)

table.addCell((String) model.getValueAt(i, 0));

table.addCell((String) model.getValueAt(i, 1));

table.addCell((String) model.getValueAt(i, 2));

}

// Добавляем таблицу в документ

document.add(table);

// Закрываем документ

document.close();

// Уведомление о успешном сохранении

JOptionPane.showMessageDialog(ServiceList, "PDF saved successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);

} catch (Exception e) {

// Обработка ошибки

e.printStackTrace();

}

}

private void saveToHTML() {

String filePath = "C:/lab7/emloyee.html";

// Создаем HTML контент

StringBuilder htmlContent = new StringBuilder();

htmlContent.append("<html>");

htmlContent.append("<head><title>Employee list</title>");

htmlContent.append("<style>");

htmlContent.append("body { font-family: Arial, sans-serif; padding: 20px; }");

htmlContent.append("h1 { color: #333; text-align: center; }");

htmlContent.append("table { width: 100%; border-collapse: collapse; margin-top: 20px; }");

htmlContent.append("th, td { padding: 10px; text-align: left; border: 1px solid #ddd; }");

htmlContent.append("th { background-color: #f2f2f2; }");

htmlContent.append("tr:nth-child(even) { background-color: #f9f9f9; }");

htmlContent.append("tr:hover { background-color: #f1f1f1; }");

htmlContent.append("</style></head>");

htmlContent.append("<body><h1>Employee list</h1><table>");

htmlContent.append("<tr><th>Surname</th><th>Name</th><th>Surname2</th>");

// Получаем данные из модели таблицы (DefaultTableModel)

DefaultTableModel model = (DefaultTableModel) books.getModel();

// Заполняем таблицу HTML данными

for (int i = 0; i < model.getRowCount(); i++) {

String Surname = (String) model.getValueAt(i, 0);

String Name = (String) model.getValueAt(i, 1);

String Surname2 = (String) model.getValueAt(i, 2);

char[] inspectionDate = new char[0];

htmlContent.append("<tr><td>").append(Surname).append("</td><td>")

.append(Name).append("</td><td>")

.append(Surname2).append("</td><td>");

}

htmlContent.append("</table></body></html>");

// Пишем отформатированный HTML в файл

try (FileWriter writer = new FileWriter(filePath)) {

writer.write(htmlContent.toString());

JOptionPane.showMessageDialog(ServiceList, "HTML saved successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);

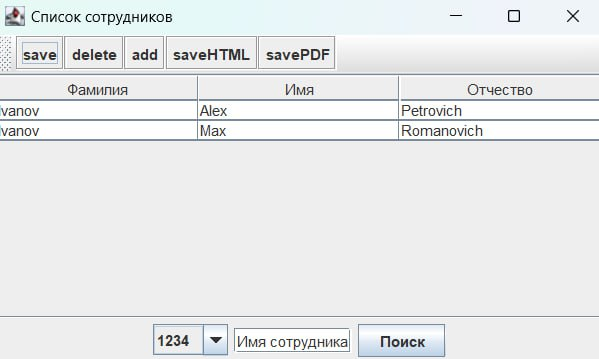
} catch (IOException e) {

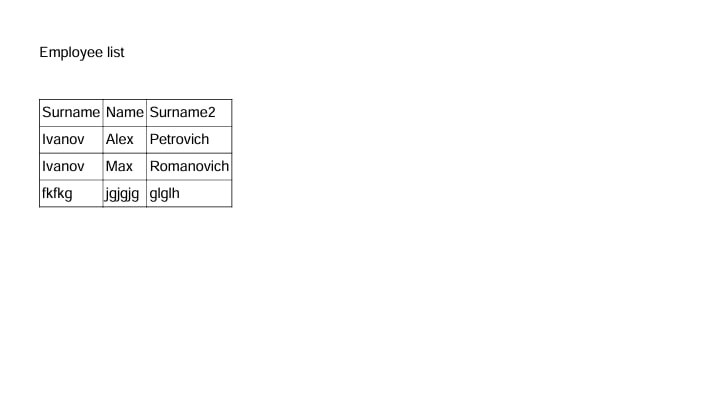
e.printStackTrace();

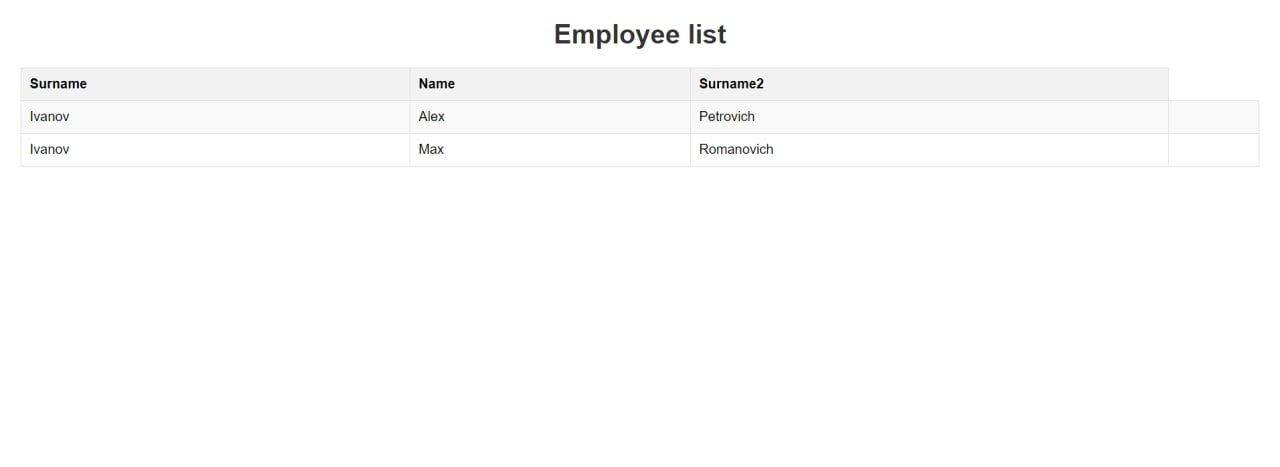
}

}

**Работа класса.**







Javadoc.

/\*\*

\*

\* It is a constructor.

\*

\*/

private void saveToPDF() {

String filePath = "C:/lab7/Emlpoyee.pdf";

try {

// Создаем объект для записи в PDF

PdfWriter writer = new PdfWriter(new FileOutputStream(filePath));

PdfDocument pdf = new PdfDocument(writer);

Document document = new Document(pdf);

// Заголовок документа

document.add(new Paragraph("Employee list"));

document.add(new Paragraph("\n"));

// Создаем таблицу с 4 столбцами

Table table = new Table(3);

// Добавляем заголовки столбцов в таблицу

table.addCell("Surname");

table.addCell("Name");

table.addCell("Surname2");

// Получаем данные из модели таблицы и заполняем таблицу

DefaultTableModel model = (DefaultTableModel) books.getModel();

for (int i = 0; i < model.getRowCount(); i++) {

// Добавляем данные по каждому сотруднику (по строкам)

table.addCell((String) model.getValueAt(i, 0));

table.addCell((String) model.getValueAt(i, 1));

table.addCell((String) model.getValueAt(i, 2));

}

// Добавляем таблицу в документ

document.add(table);

// Закрываем документ

document.close();

// Уведомление о успешном сохранении

JOptionPane.showMessageDialog(ServiceList, "PDF saved successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);

} catch (Exception e) {

// Обработка ошибки

e.printStackTrace();

}

}

/\*\*

\*

\* It is a constructor.

\*

\*/

private void saveToHTML() {

String filePath = "C:/lab7/emloyee.html";

// Создаем HTML контент

StringBuilder htmlContent = new StringBuilder();

htmlContent.append("<html>");

htmlContent.append("<head><title>Employee list</title>");

htmlContent.append("<style>");

htmlContent.append("body { font-family: Arial, sans-serif; padding: 20px; }");

htmlContent.append("h1 { color: #333; text-align: center; }");

htmlContent.append("table { width: 100%; border-collapse: collapse; margin-top: 20px; }");

htmlContent.append("th, td { padding: 10px; text-align: left; border: 1px solid #ddd; }");

htmlContent.append("th { background-color: #f2f2f2; }");

htmlContent.append("tr:nth-child(even) { background-color: #f9f9f9; }");

htmlContent.append("tr:hover { background-color: #f1f1f1; }");

htmlContent.append("</style></head>");

htmlContent.append("<body><h1>Employee list</h1><table>");

htmlContent.append("<tr><th>Surname</th><th>Name</th><th>Surname2</th>");

// Получаем данные из модели таблицы (DefaultTableModel)

DefaultTableModel model = (DefaultTableModel) books.getModel();

// Заполняем таблицу HTML данными

for (int i = 0; i < model.getRowCount(); i++) {

String Surname = (String) model.getValueAt(i, 0);

String Name = (String) model.getValueAt(i, 1);

String Surname2 = (String) model.getValueAt(i, 2);

char[] inspectionDate = new char[0];

htmlContent.append("<tr><td>").append(Surname).append("</td><td>")

.append(Name).append("</td><td>")

.append(Surname2).append("</td><td>");

}

htmlContent.append("</table></body></html>");

// Пишем отформатированный HTML в файл

try (FileWriter writer = new FileWriter(filePath)) {

writer.write(htmlContent.toString());

JOptionPane.showMessageDialog(ServiceList, "HTML saved successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);

} catch (IOException e) {

e.printStackTrace();

}

}

**Вывод.**

В процессе выполнения задания мы изучили способы работы с Jaspersoft iReport Designert в Java и создали собственные функции для создания и сохранения отчетов в PDF и HTML файл.