

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 №7 по теме “Построение отчетов в PDF- и HTML-форматах ”**

Выполнил: Цацкис Артём гр. 3311

Проверил: Павловский М. Г.

Подпись преподавателя: \_\_\_\_\_

Санкт-Петербург

2024

## **Цель работы:**

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

## **Описание задания:**

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

## **Скриншот созданного макета:**

Title	Author	Font	Pinned
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
qwеrtye	qwеrtye	qwеrtye	Да

## Фрагмент кода, отвечающие за сохранения данных в pdf и html форматах

### Сохранения данных в pdf формате

```
import java.io.FileNotFoundException; import
java.io.FileOutputStream; import
java.io.IOException;

import javax.swing.table.DefaultTableModel; import

com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException; import
com.itextpdf.text.Font;
import com.itextpdf.text.PageSize; import
com.itextpdf.text.Phrase; import
com.itextpdf.text.pdf.BaseFont; import
com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable; import
com.itextpdf.text.pdf.PdfWriter;

/**
 * Класс {@code HtmlSave} генерирует pdf файл
 */

public class Report { Report(DefaultTableModel tableModel) {

    Document document = new Document(PageSize.A4, 50, 50, 50, 50);
    PdfPTable t = new PdfPTable(3);

    try {
        PdfWriter writer = PdfWriter.getInstance(document, new FileOutputStream("./data/PdfData.pdf"));
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (DocumentException e) {
        e.printStackTrace();
    }

    BaseFont bfComic = null; try {
        bfComic = BaseFont.createFont("C:/Windows/Fonts/Arial.ttf",BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
    } catch (DocumentException e1) {
        e1.printStackTrace();
    } catch (IOException e1) {
        e1.printStackTrace();
    }

    Font font1 = new Font(bfComic, 12);

    t.addCell(new PdfPCell(new Phrase("Товар",font1))); t.addCell(new PdfPCell(new
Phrase("Производитель",font1))); t.addCell(new PdfPCell(new
Phrase("Наличие",font1)));

    for(int i = 0; i < tableModel.getRowCount(); i++){
        t.addCell(new Phrase((String) tableModel.getValueAt(i,0),font1)); t.addCell(new
Phrase((String) tableModel.getValueAt(i,1),font1)); t.addCell(new
Phrase((String)
tableModel.getValueAt(i,2),font1));
    }
}
```

```

        document.open(); try {
            document.add(t);
        } catch (DocumentException e) {
            e.printStackTrace();
        }

        document.close();
    }
}

```

## Сохранения данных в html формате

```

import java.io.FileWriter; import
java.io.IOException; import
java.io.PrintWriter;

import javax.swing.table.DefaultTableModel;

/**
 * Класс { @code HtmlSave } генерирует html файл
 */

public class HtmlSave { HtmlSave(DefaultTableModel

    tableModel){

        PrintWriter pw = null;
        try {
            pw = new PrintWriter(new FileWriter("./data/HtmlData.html"));
        } catch (IOException e) {
            e.printStackTrace();
        }

        pw.println("<TABLE
BORDER><TR><TH>Товар<TH>Производитель<TH>Наличие</TR>");
        for(int i = 0; i < tableModel.getRowCount(); i++) { int square = i * i;
            pw.println("<TR><TD>" + (String) tableModel.getValueAt(i,0)
                + "<TD>" + (String) tableModel.getValueAt(i,1)
                + "<TD>" + (String) tableModel.getValueAt(i,2));
            }

        pw.println("</TABLE>"); pw.close();
    }
}

```

## **Вывод**

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

**Ссылка на репозиторий github:**

[TsatskisArtem/For-OOP7](https://github.com/TsatskisArtem/For-OOP7)

### **Ссылка на репозиторий Git-hub:**

ООPlab7.java-код программы

Дос- папка с документацией сгенерированная javadocs

Tsatskis-lab7.pdf-отчет по лабораторной работе