

O trabalho deve ser realizado individualmente. Os arquivos devem ser compactados em um arquivo `.zip` ou `.tar` contendo o nome do aluno. O arquivo compactado deverá conter apenas os códigos-fontes (pasta `src` do Eclipse) e os diagramas UML em formato `.jpg` ou `.png`. **Não serão aceitos projetos com códigos-fonte no formato `.class`!**

Trabalho de POO

Tema 4: Aplicação de gerenciamento de notas

Crie uma aplicação que permita aos alunos cadastrar as disciplinas aos quais ele cursa e as avaliações das disciplinas. A aplicação deve ser capaz de calcular a média semestral do aluno (parcial se o aluno ainda não completou todas as avaliações) e exibir quanto o aluno necessita tirar no exame, de acordo com a tabela da UDESC (ver Fórmula 1). A aplicação deve possuir as seguintes funcionalidades:

- Permitir o cadastro de novos alunos e a realização de login de um aluno;
- Permitir o cadastro, edição e exclusão de novos semestres;
- Permitir o cadastro, edição e exclusão de novas disciplinas;
- Permitir a inserção de novas avaliações em cada disciplina. Onde cada avaliação possui nome, peso e data;
- Permitir a inserção de notas para as avaliações dos alunos nas disciplinas. Onde cada nota de ser atribuída a um aluno em uma avaliação de uma disciplina;
- Calcular, para cada aluno, para as disciplinas que estão sendo cursadas, a nota necessária no exame de acordo com a Fórmula 1;
- Gerar um pdf utilizando a biblioteca `iText` contendo um relatório com todos os semestres, disciplinas e médias finais do aluno.

$$Exame = -1.5 * M_f + 12.5, \quad 1.7 \leq M_f < 7 \quad (1)$$

Para utilizar a biblioteca é necessário fazer download das bibliotecas `slf4j-api` (ver Figura 1) e `itext7` (ver Figura 2).

É necessário adicionar todos os arquivos `.jar` ao **build path**. No código abaixo está um exemplo de criação de tabelas usando a biblioteca:

org/slf4j/slf4j-api/1.7.9



...		
slf4j-api-1.7.9-javadoc.jar	2014-12-16 21:57	206578
slf4j-api-1.7.9-javadoc.jar.asc	2014-12-16 21:57	473
slf4j-api-1.7.9-javadoc.jar.asc.md5	2014-12-16 21:57	32
slf4j-api-1.7.9-javadoc.jar.asc.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9-javadoc.jar.md5	2014-12-16 21:57	32
slf4j-api-1.7.9-javadoc.jar.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9-sources.jar	2014-12-16 21:57	50642
slf4j-api-1.7.9-sources.jar.asc	2014-12-16 21:57	473
slf4j-api-1.7.9-sources.jar.asc.md5	2014-12-16 21:57	32
slf4j-api-1.7.9-sources.jar.asc.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9-sources.jar.md5	2014-12-16 21:57	32
slf4j-api-1.7.9-sources.jar.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9-tests.jar	2014-12-16 21:57	21271
slf4j-api-1.7.9-tests.jar.asc	2014-12-16 21:57	473
slf4j-api-1.7.9-tests.jar.asc.md5	2014-12-16 21:57	32
slf4j-api-1.7.9-tests.jar.asc.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9-tests.jar.md5	2014-12-16 21:57	32
slf4j-api-1.7.9-tests.jar.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9.jar	2014-12-16 21:57	32121
slf4j-api-1.7.9.jar.asc	2014-12-16 21:57	473
slf4j-api-1.7.9.jar.asc.md5	2014-12-16 21:57	32
slf4j-api-1.7.9.jar.asc.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9.jar.md5	2014-12-16 21:57	32
slf4j-api-1.7.9.jar.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9.pom	2014-12-16 21:57	2660
slf4j-api-1.7.9.pom.asc	2014-12-16 21:57	473
slf4j-api-1.7.9.pom.asc.md5	2014-12-16 21:57	32
slf4j-api-1.7.9.pom.asc.sha1	2014-12-16 21:57	40
slf4j-api-1.7.9.pom.md5	2014-12-16 21:57	32
slf4j-api-1.7.9.pom.sha1	2014-12-16 21:57	40

Figure 1: Download da biblioteca slf4j-api.

Note: Java 7 support [was deprecated](#) as of the 7.1.13 release, and so from this release we're targeting Java 8

New Features

- SVG: Support `<pattern>` element
- Support of form fields border styles (Beveled, Underline and Inset)





Improvements

- SVG: add support of font-relative units for the font-size property in a text element
- Allow OTF language records to be fetched by tag
- Support word-wrap and word-break properties

Bugs

- Expand colored background on the overflowed part of span
- SVG stroke-width doesn't work in `<use>` tags
- Signed short number overflows in `readCoverageFormat()`
- Page-break-inside: avoid not being taken into account for floating DIVs

Assets 4

 iText7-Core-7.1.14-only-jars.zip	6.24
 iText7-Core-7.1.14.zip	17.6
 Source code (zip)	
 Source code (tar.gz)	


15 Privacy Security Status Docs  Contact GitHub Pricing API Training Blog

Figure 2: Download da biblioteca itext7.

```
import java.io.FileNotFoundException;

import com.itextpdf.kernel.pdf.PdfDocument;
import com.itextpdf.kernel.pdf.PdfWriter;
import com.itextpdf.layout.Document;
import com.itextpdf.layout.element.Cell;
import com.itextpdf.layout.element.Paragraph;
import com.itextpdf.layout.element.Table;

public class Main {
    public static void main(String[] args) {

        try {
            // Creating a PdfDocument object
            String dest = "addingTable.pdf";
            PdfWriter writer;
            writer = new PdfWriter(dest);
            // Creating a PdfDocument object
            PdfDocument pdf = new PdfDocument(writer);

            // Creating a Document object
            Document doc = new Document(pdf);

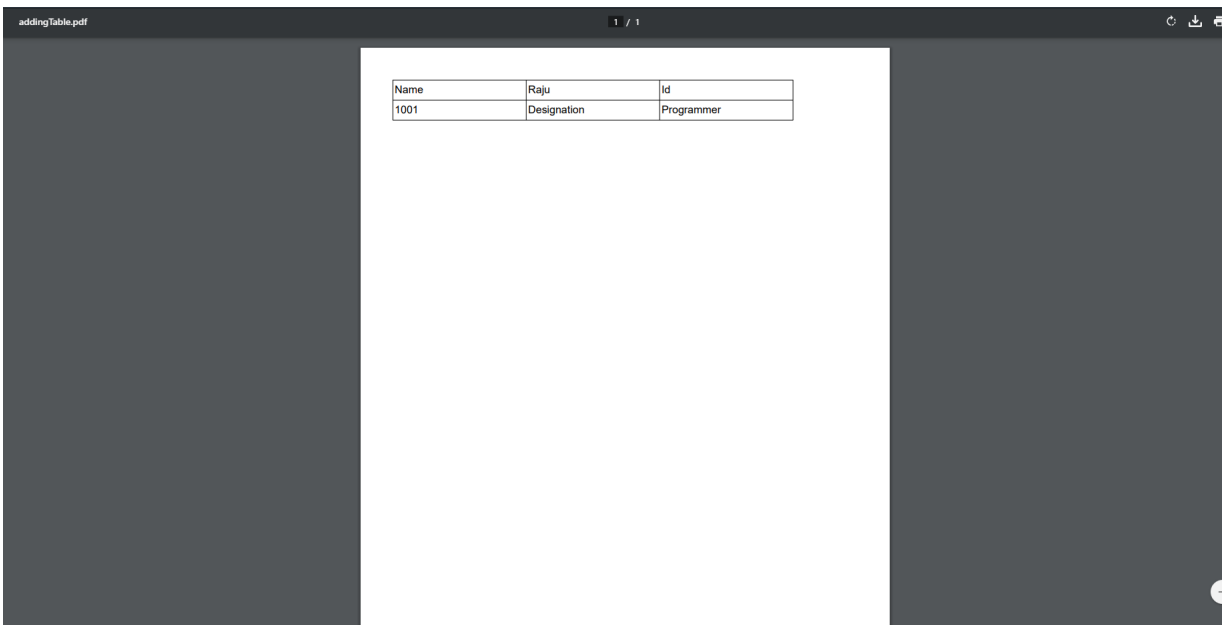
            // Creating a table
            float[] pointColumnWidths = { 150F, 150F, 150F };
            Table table = new Table(pointColumnWidths);

            // Adding cells to the table
            table.addCell(new Cell().add(new Paragraph("Name")));
            table.addCell(new Cell().add(new Paragraph("Raju")));
            table.addCell(new Cell().add(new Paragraph("Id")));
            table.addCell(new Cell().add(new Paragraph("1001")));
            table.addCell(new Cell().add(new Paragraph("Designation")));
            table.addCell(new Cell().add(new Paragraph("Programmer")));

            // Adding Table to document
            doc.add(table);

            // Closing the document
            doc.close();
            System.out.println("Table_created_successfully..");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

O resultado da execução do código pode ser visto na Figura 3.



The image shows a PDF viewer window titled 'addingTable.pdf'. The PDF content consists of a table with 2 rows and 3 columns. The first row contains the headers 'Name', 'Raju', and 'Id'. The second row contains the values '1001', 'Designation', and 'Programmer'.

Name	Raju	Id
1001	Designation	Programmer

Figure 3: PDF gerado pelas bibliotecas.