



# Java 核心技术(进阶)

第四章 高级文件处理

第八节 PDF简介及解析

华东师范大学 陈良育

# PDF



- PDF

- Portable Document Format的简称，意为“便携式文档格式”。
- Adobe公司发明的。
- PostScript，用以生成和输出图形，在任何打印机上都可保证精确的颜色和准确的打印效果。
- 字型嵌入系统，可使字型随文件一起传输。
- 结构化的存储系统，绑定元素和任何相关内容到单个文件，带有适当的数据压缩系统。

- PDF的版本，一般是1.4+。



# PDF 处理和第三方包

- 常见功能处理
  - 解析PDF
  - 生成PDF(转化)
- 第三方包
  - Apache PDFBox (免费)
  - iText (收费)
  - XDocReport (将docx转化为pdf)



# PDFBox

- Apache PDFBox
  - 纯Java类库
  - 主要功能：创建，提取文本，分割/合并/删除，...
  - 主要类
    - PDDocument pdf文档对象
    - PDFTextStripper pdf文本对象
    - PDFMergerUtility 合并工具

# XDocReport



- XDocReport
  - 将docx文档合并输出为其他数据格式(pdf/html/...)
  - PdfConverter
  - 基于poi和iText完成



# pdf总结



- pdf操作，可读取解析、合并、删除页面
- 产生pdf和修改pdf，建议先生成docx，再进行转化
- API很多，需要多查询、多练习



# 代码(1) NewMergePdfs.java

```
public class NewMergePdfs
{
    private static final Log LOG = LoggerFactory.getLog(NewMergePdfs.class);

    public static void main(String[] a) throws Exception
    {
        merge();
    }

    public static void merge() throws Exception
    {
        FileOutputStream fos = new FileOutputStream(new File("merge.pdf"));

        ByteArrayOutputStream mergedPDFOutputStream = null;
        File file1 = new File("sample1.pdf");
        File file2 = new File("sample2.pdf");

        List<InputStream> sources = new ArrayList<InputStream>();
```

# 代码(2) NewMergePdfs.java



```
try
{
    sources.add(new FileInputStream(file1));
    sources.add(new FileInputStream(file2));

    mergedPDFOutputStream = new ByteArrayOutputStream();

    // 设定来源和目标
    PDFMergerUtility pdfMerger = new PDFMergerUtility();
    pdfMerger.addSources(sources);
    pdfMerger.setDestinationStream(mergedPDFOutputStream);

    // 设置合并选项
    PDDocumentInformation pdfDocumentInfo = new PDDocumentInformation();
    pdfMerger.setDestinationDocumentInformation(pdfDocumentInfo);

    // 合并
    pdfMerger.mergeDocuments(MemoryUsageSetting.setupMainMemoryOnly());

    fos.write(mergedPDFOutputStream.toByteArray());
    fos.close();
}
```



## 代码(3) NewMergePdfs.java



```
,  
catch (Exception e)  
{  
    throw new IOException("PDF merge problem", e);  
}  
finally  
{  
    for (InputStream source : sources)  
    {  
        IOUtils.closeQuietly(source);  
    }  
    IOUtils.closeQuietly(mergedPDFOutputStream);  
    IOUtils.closeQuietly(fos);  
}  
}  
}
```

# 代码(4) MergePdfs.java



```
public class MergePdfs {  
    public static void main(String[] args) throws IOException {  
  
        //Loading an existing PDF document  
        File file1 = new File("sample1.pdf");  
        PDDocument doc1 = PDDocument.Load(file1);  
  
        File file2 = new File("sample2.pdf");  
        PDDocument doc2 = PDDocument.Load(file2);  
  
        //Instantiating PDFMergerUtility class  
        PDFMergerUtility PDFmerger = new PDFMergerUtility();  
  
        //Setting the destination file  
        PDFmerger.setDestinationFileName("merge2.pdf");  
  
        //adding the source files  
        PDFmerger.addSource(file1);  
        PDFmerger.addSource(file2);  
    }  
}
```

## 代码(5) MergePdfs.java




```
//Merging the two documents
PDFmerger.mergeDocuments();

System.out.println("Documents merged");
//Closing the documents
doc1.close();
doc2.close();
    }
}
```



## 代码(6) PdfReader.java

```
public class PdfReader {  
    public static void main(String[] args){  
        File pdfFile = new File("simple.pdf");  
        PDDocument document = null;  
        try  
        {  
            document=PDDocument.Load(pdfFile);  
  
             AccessPermission ap = document.getCurrentAccessPermission();  
            if (!ap.canExtractContent())  
            {  
                throw new IOException("你没有权限抽取文本");  
            }  
            // 获取页码  
            int pages = document.getNumberOfPages();  
        }  
    }  
}
```



# 代码(7) PdfReader.java



```
// 读文本内容
PDFTextStripper stripper=new PDFTextStripper();
// 设置按顺序输出
stripper.setSortByPosition(true);
stripper.setStartPage(1); //起始页
stripper.setEndPage(pages); //结束页
String content = stripper.getText(document);
System.out.println(content);
}
catch(Exception e)
{
    System.out.println(e);
}
}
```



# 代码(8) PdfWriter.java



```
public class PdfWriter {  
  
    public static void main(String[] args) {  
        createHelloPDF();  
    }  
    public static void createHelloPDF() {  
        PDDocument doc = null;  
        PDPage page = null;  
  
        try {  
            doc = new PDDocument();  
            page = new PDPage();  
            doc.addPage(page);  
            PDFont font = PDType1Font.HELVETICA_BOLD;
```

# 代码(9) PdfWriter.java



```
PDPageContentStream content = new PDPageContentStream(doc, page);
content.beginText();
content.setFont(font, 12);
content.moveTextPositionByAmount(100, 700);
content.showText("hello world");

content.endText();
content.close();
doc.save("test.pdf");
doc.close();
} catch (Exception e) {
    System.out.println(e);
}
}
```

# 代码(10) RemovePdf.java



```
public class RemovePdf {  
    public static void main(String[] args) throws Exception {  
        File file = new File("merge.pdf");  
        PDDocument document = PDDocument.Load(file);  
  
        int noOfPages = document.getNumberOfPages();  
        System.out.println("total pages: " + noOfPages);  
  
        // 删除第1页  
        document.removePage(1); // 页码索引从0开始算  
  
        System.out.println("page removed");  
  
        // 另存为新文档  
        document.save("merge2.pdf");  
  
        document.close();  
    }  
}
```



# 代码(11) XDocReportTest.java

```
public class XDocReportTest {  
  
    public static void main(String[] args) throws Exception {  
        XWPFDocument doc = new XWPFDocument(new FileInputStream("template.docx")); // docx  
        PdfOptions options = PdfOptions.create();  
        options.fontProvider(new IFontProvider() {  
            // 设置中文字体  
            public Font getFont(String familyName, String encoding, float size, int style, Color color) {  
                try {  
                    BaseFont bfChinese = BaseFont.createFont(  
                        "C:\\Program Files (x86)\\Microsoft Office\\root\\VFS\\Fonts\\private\\STSONG.TTF",  
                        BaseFont.IDENTITY_H, BaseFont.EMBEDDED);  
                }  
            }  
        });  
    }  
}
```



# 代码(12) XDocReportTest.java



```
Font fontChinese = new Font(bfChinese, size, style, color);
if (familyName != null)
    fontChinese.setFamily(familyName);
return fontChinese;
} catch (Throwable e) {
    e.printStackTrace();
    return ITextFontRegistry.getRegistry().getFont(familyName, encoding, size, style, color);
}
});
PdfConverter.getInstance().convert(doc, new FileOutputStream("template.pdf"), options); // pdf
}
```





谢谢!