mr. A.B.C. Achternaam
JE MAINTIENDRAI
Notaris te Plaatsnaam

```
package nl.novadoc.tools;

import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.pdmodel.PDPageContentStream;
import org.apache.pdfbox.pdmodel.font.PDFont;
import org.apache.pdfbox.pdmodel.font.PDType1Font;
import org.apache.pdfbox.pdmodel.graphics.image.PDImageXObject;

import lombok.extern.slf4j.Slf4j;
import nl.novadoc.utils.FileUtils;
import nl.novadoc.utils.WrapperException;

@Slf4j
public class GenPDF {

    private static class PDFInfo implements AutoCloseable {

        private static final float OFFSET = 100.0f;
        private static final float XMAX = 612.0f;
        private static final float YMAX = 792.0f;
        private static final int LINE = 15;

        private final PDDocument doc;
        private PDPage page;
        private PDPageContentStream stream;
private final PDFont font;
private final int fontSize;
private float x;
private float y;

public PDFInfo() {
    this(PDType1Font.HELVETICA, 10);
}

public PDFInfo(PDFont font, int fontSize) {
    this.font = font;
    this.fontSize = fontSize;
    doc = new PDDocument();
}

void drawImage(String name, boolean atEnd) {
    try {
        PDImageXObject img = PDImageXObject.createFromFile(name, doc);
        if(atEnd) {
            stream.drawImage(img, XMAX - (OFFSET/2) - img.getWidth(), y - 15);
        } else {
            stream.drawImage(img, x, y);
            y += img.getHeight();
        }
    } catch(Exception e) {
        // doesn't matter
    }
}

void newPage() {
    try {
        page = new PDPage();
        doc.addPage(page);
        stream = new PDPageContentStream(doc, page);
        stream.beginText();
        stream.setFont(font, fontSize);
        x = OFFSET;
        y = YMAX - (OFFSET/2);
        stream.newLineAtOffset(x, y);
    } catch (Exception e) {
        throw new WrapperException(e);
    }
}

void newLine() {
    try {
        y -= LINE;
```

```
            if (y < OFFSET / 2) {
                stream.endText();
                stream.close();
                newPage();
            }
            stream.newLineAtOffset(0, -LINE);
            x = OFFSET;
        } catch (Exception e) {
            throw new WrapperException(e);
        }
    }

    public void print(String word) {
        try {
            float w = font.getStringWidth(word)  / 1000 * fontSize;
            if((w + x) > (XMAX - OFFSET/2)) {
                newLine();
            }
            stream.showText(word);
            x += w;
        } catch (Exception e) {
            throw new WrapperException(e);
        }
    }

    public void save(String name) {
        try {
            stream.endText();
            stream.close();
            doc.save(new File(name));
        } catch (Exception e) {
            log.error("{}", e, e);
            throw new WrapperException(e);
        }
    }

    @Override
    public void close() throws Exception {
        doc.close();
    }
};

public static File createPDF(String name, List<String> lines) {

    try (PDFInfo pdf = new PDFInfo()) {
        pdf.newPage();
        pdf.drawImage("Stempel.jpg", false);
```

```
            for (String line : lines) {
                line = line.replace("\t", "    ");
                String[] words = line.split(" ");
                for (String word : words) {
                    pdf.print(word + " ");
                }
                pdf.newLine();
            }
            pdf.drawImage("handtekening.png", true);
            pdf.save(name + ".pdf");
            File f = new File(name + ".pdf");
            if (f.exists()) {
                return f;
            } else {
                throw new IOException("PDF not created");
            }
        } catch (Exception e) {
            log.error("{}", e, e);
            throw new WrapperException(e);
        }
    }

    public static void main(String... args) {
        String[] lines = FileUtils.readLines(new File("marien.txt"));
        createPDF("marien", Arrays.asList(lines));
    }
}
```

[illegible]