```
public class Parser {
    private Parser successor;

    public void parse(String fileName) {
        if (getSuccessor() != null)
            getSuccessor().parse(fileName);
        else
            System.out.println("unable to find the correct parser
                                   for the filedesign_ : " + fileName);
    }

    protected boolean canHandleFile(String fileName,String format) {
        return (fileName == null) || (fileName.endsWith(format));
    }

    public Parser getSuccessor() {
        return successor;
    }

    public void setSuccessor(Parser successor) {
        this.successor = successor;
    }
}

public class CSVParser extends Parser {
    public CSVParser(Parser successor){
        this.setSuccessor(successor);
    }

    public void parse(String fileName) {
        if (canHandleFile(fileName,".csv"))
            System.out.println("CSV parser is handling the file : " +
                                   fileName);
        else
            super.parse(fileName);
    }
}

public class TextParser extends Parser {
    public TextParser(Parser successor) {
        this.setSuccessor(successor);
    }

    public void parse(String fileName) {
        if (canHandleFile(fileName,".txt"))
            System.out.println("text parser is handling the file : " +
                                   fileName);
        else
            super.parse(fileName);
    }
}
```

```
public class XMLParser extends Parser {
    public void parse(String fileName) {
        if (canHandleFile(fileName,".xml"))
            System.out.println("XML parser is handling the file : " +
                                   fileName);
        else
            super.parse(fileName);
    }
}


import java.util.List;
import java.util.ArrayList;

public class Application {
    private static List<String> populateFiles(){
        List<String> fileList = new ArrayList<>();

        fileList.add("someFile.txt");
        fileList.add("xmlFile.xml");
        fileList.add("csvFile.csv");
        fileList.add("csvFile.doc");

        return fileList;
    }

    public static void main(String... args) {
        List<String> fileList = populateFiles();
        Parser xmlParser = new XMLParser();
        Parser csvParser = new CSVParser(xmlParser);
        Parser textParser = new TextParser(csvParser);

        for (String fileName : fileList)
            textParser.parse(fileName);
    }
}
```