



## 1. Introdução

Uma empresa de análise de dados te contratou como desenvolvedor freelancer para um projeto em Java de conversão de arquivos. Eles trabalham com vários tipos de arquivos para armazenar os dados dos clientes, mas gostariam de ter a flexibilidade de convertê-los entre si, seguindo os padrões da empresa.

A sua tarefa é criar os métodos para converter os arquivos enviados pelo usuário em outro arquivo com o formato requisitado. Os formatos de arquivos a serem convertidos são:

Formato	Converter para	
CSV	XML	Tabela HTML
XML	CSV	Tabela HTML
Tabela HTML	CSV	XML

A empresa segue os seguintes padrões para os arquivos:

CSV:

```
First Column,Second Column,Third Column
a,123,0.33
b,234,0.34
c,345,0.35
```

XML (note que o nome das colunas devem ser convertidas para camelCase):

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <row>
    <firstColumn>a</firstColumn>
    <secondColumn>123</secondColumn>
    <thirdColumn>0.33</thirdColumn>
  </row>
  <row>
    <firstColumn>b</firstColumn>
```

```

        <secondColumn>234</secondColumn>
        <thirdColumn>0.34</thirdColumn>
    </row>
    <row>
        <firstColumn>c</firstColumn>
        <secondColumn>345</secondColumn>
        <thirdColumn>0.35</thirdColumn>
    </row>
</root>

```

Tabela HTML:

```

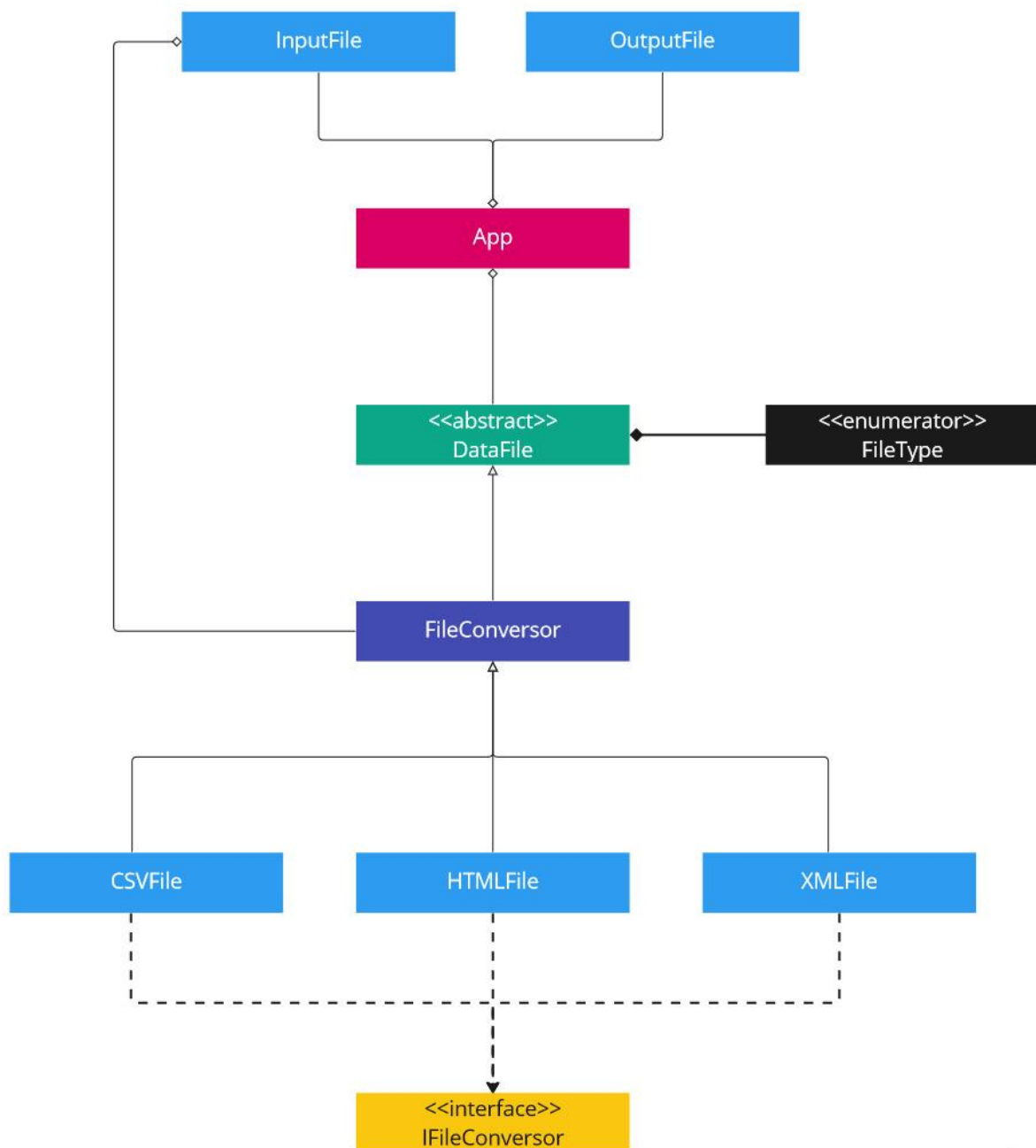
<html>
  <body>
    <table>
      <tr>
        <th>First Column</th>
        <th>Second Column</th>
        <th>Third Column</th>
      </tr>
      <tr>
        <td>a</td>
        <td>123</td>
        <td>0.33</td>
      </tr>
      <tr>
        <td>b</td>
        <td>234</td>
        <td>0.34</td>
      </tr>
      <tr>
        <td>c</td>
        <td>345</td>
        <td>0.35</td>
      </tr>
    </table>
  </body>
</html>

```

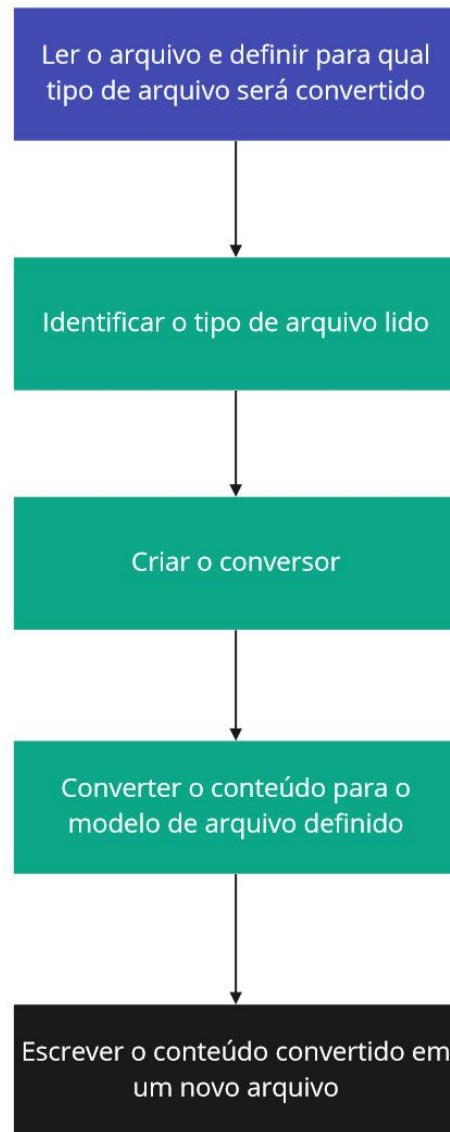
Esse é o diagrama de classe implementado na aplicação. A classe principal (App) agrega as classes InputFile e OutputFile para ler e escrever os arquivos. DataFile é a classe referente aos dados dos arquivos. FileConvensor herda DataFile, e cada tipo de arquivo (CSVFile, HTMLFile e XMLFile) são polimorfos de FileConvensor, e implementam os métodos da interface IFileConvensor. Note que todo tipo de arquivo é DataFile ao mesmo tempo que é FileConvensor, ou seja, eles são arquivos e conversores para outros arquivos.

Mais informações sobre o diagrama nesse link:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial>



O diagrama a seguir representa o fluxo de funcionamento da aplicação. O tipo de arquivo lido é convertido com os métodos estáticos de FileConversor, que por sua vez chama o método de conversão correspondente ao tipo de arquivo identificado para o tipo de arquivo que o usuário deseja converter.



miro

É requisitado que você desenvolva os métodos:

1. Em FileConversor:

```
public static DataFile createFile(String fileName, String  
fileContent, String fileExtension)
```

2. Em cada model dos conversores (CSVFile, HTMLFile e XMLFile):

```
public static DataFile createFile(String fileName, String  
fileContent, String fileExtension)
```

3. Os métodos de conversão em cada model dos conversores (CSVFile, HTMLFile e XMLFile):

```
public CSVFile convertToCSV()  
public HTMLFile convertToHTML()  
public XMLFile convertToXML()
```

### 3. Critério de Correção

- 60% para implementação do conteúdo:
  - o programa funciona e faz o que foi pedido;
  - os métodos acessam adequadamente os objetos definidos.
- 25% para eficiência:
  - não há código duplicado, redundante ou não atingível;
  - o acesso e manipulação dos dados foram otimizados.
- 15% para estilo geral de programação:
  - layout limpo do código Java;
  - comentários claros e úteis (caso algum código não seja entendido ou não possua comentário, este fato será descontado na nota);
  - definições eficientes e concisas dos métodos.

### 4. Submissão

Antes de submeter, coloque seu nome completo e número usp no código-fonte Java, na forma de comentário JavaDoc no início de cada arquivo, utilizando a tag @author:

```
@author <nome> <número usp>
```

Ou seja, substitua <nome> pelo seu nome completo e <número usp> pelo seu número usp. Por exemplo, o aluno de nome Pedro Marcondes cujo número usp é 11111 deve substituir a linha acima por:

```
@author Pedro Marcondes 11111
```

Para documentação de código novo, siga o mesmo padrão existente no código já existente em JavaDoc. Sinta-se livre para adicionar arquivos, código e documentar o código, mas NÃO altere código ou documentação existentes, bem como a estrutura de pastas e arquivos.

- Seu sistema deverá ser submetido na atividade correspondente no TIDIA-AE, dentro do prazo limite;
- Submeta apenas o código-fonte do seu sistema como arquivo anexo na atividade no TIDIA-AE;
- Este trabalho deve ser entregue de forma individual.