3ª. Avaliação de Programação Orientada a Objetos

❑ Objetivos da Tarefa

- Implementar as classes do domínio em Java, conforme descrição dos requisitos abaixo;
- Implementar as classes da camada de visão (interface gráfica) e da camada de controle da aplicação;
- Implementar a persistência dos objetos da aplicação em arquivo binário.

❑ Estudo de Caso: Escritório de Advocacia

Um Escritório de Advocacia deseja que seja desenvolvido um sistema para automatizar o controle de processos de seus clientes. Para isso, o sistema deverá atender os seguintes requisitos:

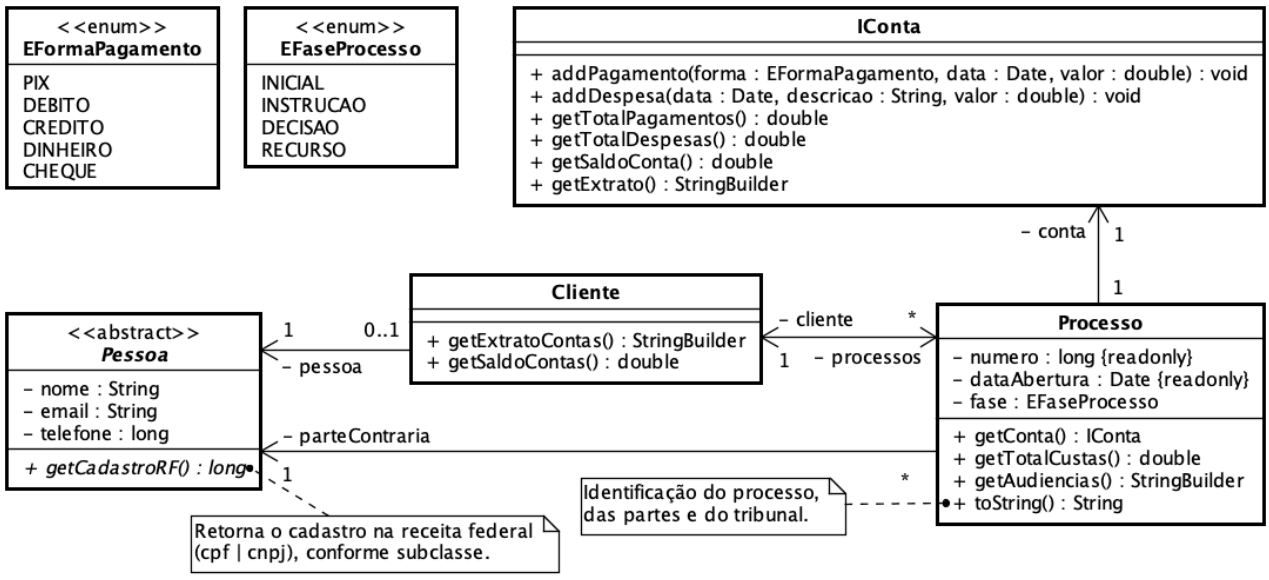
O escritório possui um cadastro de diversas pessoas que participam de processos como clientes ou como partes contrárias. Uma pessoa pode ser tanto física como jurídica e pode ter sido cliente do escritório em uma determinada época e parte contrária em outra;

Existe uma grande quantidade de processos cadastrados, alguns concluídos outros em andamento. Cada processo deve armazenar informações como o número do processo, o tribunal em que ele tramita, o cliente ao qual o processo se refere, a parte contrária envolvida e a data de abertura do processo;

Cada processo pode possuir diversas audiências. Cada audiência é relativa a um determinado processo. Para fins de histórico do processo, cada audiência deve ser registrada, devendo-se cadastrar a data, o advogado que acompanhou a audiência e a recomendação do tribunal relativa a cada audiências.

Um processo pode gerar custas (despesas com xerox, viagens etc.). O registro de uma custa deve conter a data em que ela foi gerada, sua descrição e o valor gasto. Os honorários dos advogados que atuem no processo também devem gerar lançamentos como despesas. Os pagamentos realizados pelo cliente devem ser registrados, como também o saldo das contas de cada cliente deve ser acompanhado.

❑ Diagrama de Classes (parcial)



❑ Implementar uma aplicação em Java com uso de Interfaces Gráficas conforme as seguintes convenções:

- ***MenuView***: janela principal da aplicação com menu de opções para as acesso às demais janelas do sistema;
- *Classe**View***: as classes da camada de apresentação devem ter o sufixo ***View*** acrescido ao nome do domínio;
- *Classe**Controller***: as classes ***View*** devem utilizar um controlador ***Controller*** para instanciar os objetos do domínio (a partir dos dados fornecidos pela **View**) e armazenar a lista dos objetos de domínio instanciados;
- Distribuir as classes da aplicação nos pacotes ***main*** (inicialização), ***view***, ***controller***, ***model*** e ***persistence***;
- Pacotes adicionais independentes podem ser criados para melhor organização do projeto, por exemplo: pacote para classes de exceção – ***exception*** – criadas especificamente para a aplicação; pacote para implementação de classes de utilidades – ***util*** – para tratamento de dados e serviços auxiliares ao modelo.