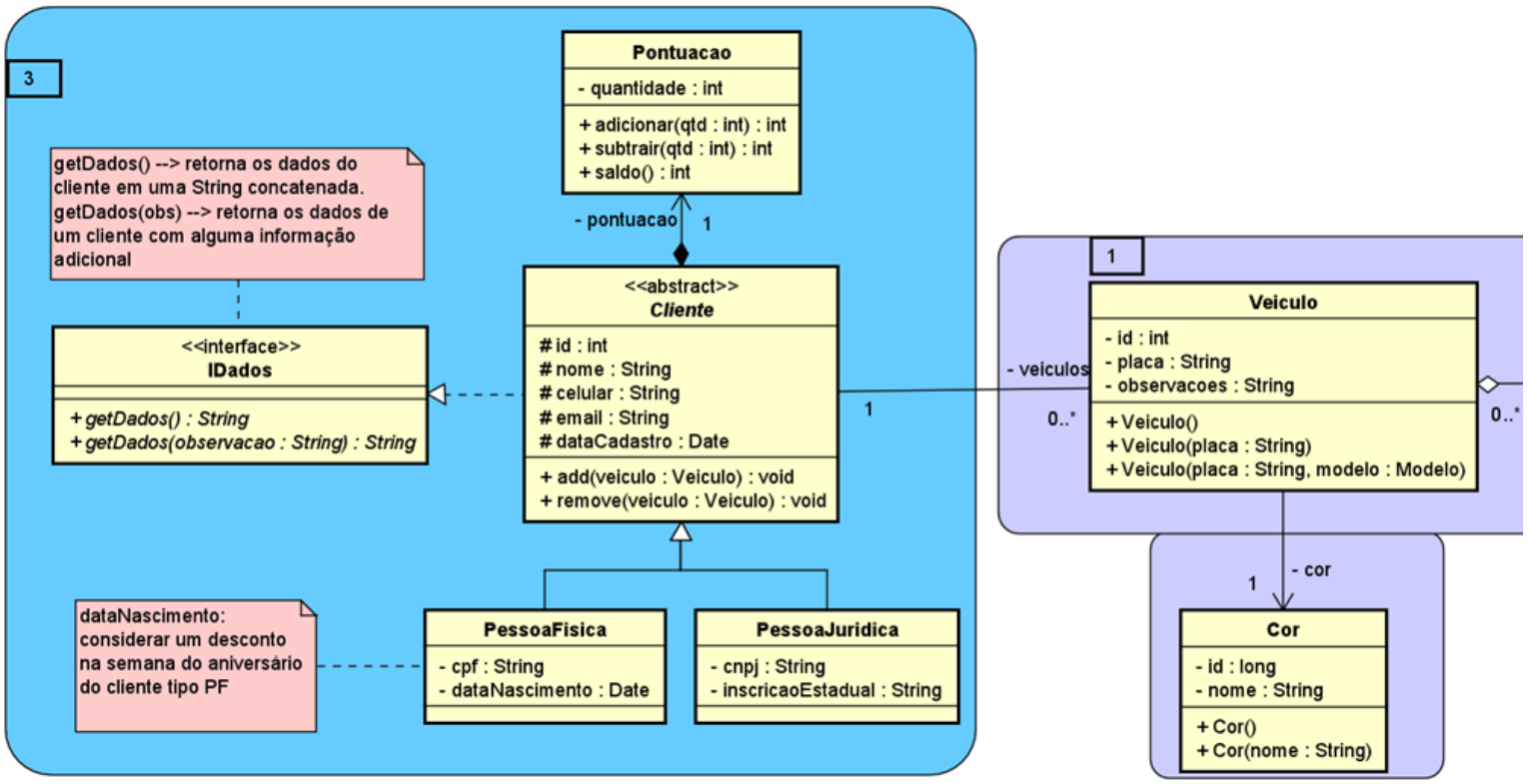
Atividade mão na massa 2

Observe o diagrama de classes mostrado na figura a seguir (a imagem do diagrama completo, considerando uma continuidade da atividade Mão na Massa 1, pode ser visualizada melhor na Figura 2, no final do documento).



A atividade Mão na Massa 2 é continuação da atividade Mão na Massa 1. O desenvolvimento da atividade é incremental e gradativo, isto é, de acordo com o avançar dos estudos, você pode aplicar os conceitos no projeto conforme orientação do professor até o prazo final de entrega. Esta atividade deverá atender aos seguintes requisitos:

1. Implementação da superclasse Cliente e das subclasses PessoaFisica e PessoaJuridica.
2. Implementação da classe abstrata Cliente.
3. Implementação de atributos protegidos (protected - "#").
4. Implementação da Interface IDados e a declaração dos métodos abstratos sobrecarregados ***getDados***.
5. Declarar a Interface ***IDados*** na classe Cliente.
6. Implementar os métodos ***getDados*** concretos nas classes Cliente, PessoaFisica e PessoaJuridica.
    - A classe Cliente retorna uma String contendo nome, celular, email e dataCadastro.

- A classe PessoaFisica retorna os dados da superclasse mais os detalhes de PessoaFisica, isto é, cpf e dataNascimento.
- A classe PessoaJuridica retorna os dados da superclasse mais os detalhes de PessoaJuridica, isto é, cnpj e inscricaoEstadual.
  Observação: o método getDados(observacao: String) deve receber uma mensagem de observação para ser impressa além dos atributos do cliente. Exemplo:

```
Nome..............: Fulano de Tal
Celular...........: 48998987766
E-mail............: fulano@ifsc.edu.br
Data de cadastro...: 12/12/2024
CPF...............: 123456321-99
Data de nascimento.: 05/01/2000
Observação........: Hoje é seu aniversário, parabéns!!!
Você terá um desconto de 20% na próxima Lavação.
```

7. Implementar polimorfismo na classe MainApp. Para isso, implemente um método print que recebe como parâmetro uma variável IDados e imprima os dados de Cliente, seja ele PessoaFisica ou PessoaJuridica.
8. Implemente a classe Pontuacao e associe ao Cliente.
9. Considere a implementação bidirecional (de multiplicidade um para muitos) entre Cliente e Veiculo.
10. Faça um método print sobrecarregado (overloading) na classe MainApp e que tenha um comportamento polimórfico recebendo como parâmetro um Cliente. O método deverá usar o operador instanceof para imprimir:

- Nome e CPF (quando PessoaFisica)
- Nome e CNPJ (quando PessoaJuridica)
- A lista de Veiculo do Cliente
- A quantidade de pontos do Cliente

Desafio:

Declare a interface IDados, também na classe Veiculo, e implemente os métodos concretos getDados para imprimir os dados de um veículo (placa, modelo, marca, categoria e potência do motor).