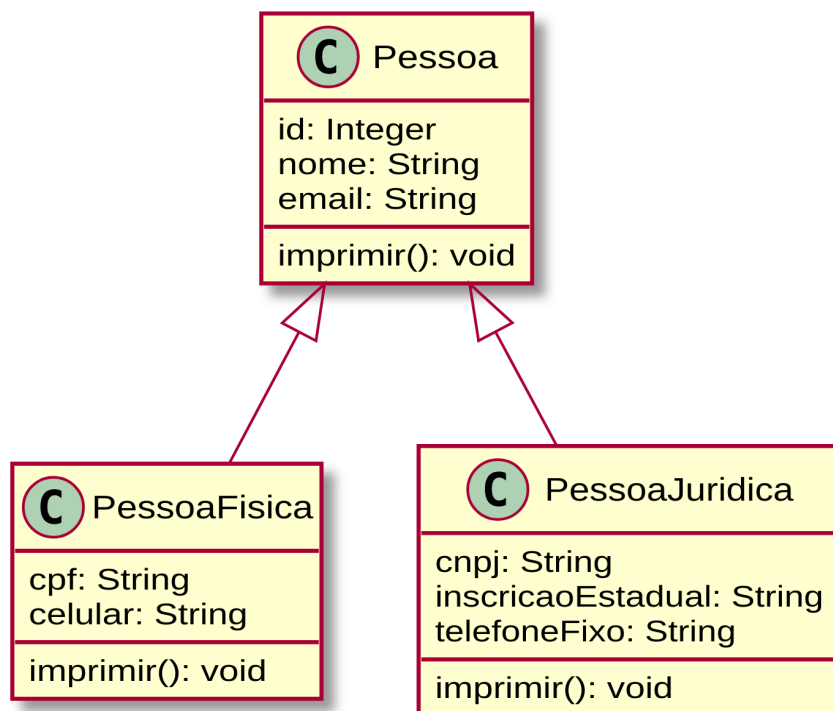


Trabalho - Polimorfismo e Herança

1. Desenvolva uma classe chamada **Matemática** e implemente as operações (somar, subtrair, dividir e multiplicar). Utilizando polimorfismo (Sobrecarga), implemente 3 métodos para cada operação matemática. Construa uma classe principal e faça os testes de todos os métodos.
2. Selecione uma das operações do exercício anterior (exceto o somar) e implemente uma nova classe que, através do mecanismo de conversão **cast**, permita a execução de métodos do mesmo nome (Sobrecarga) com diferentes tipos de parâmetros. O retorno do método deve ser **int**. Os métodos devem realizar a operação apenas sobre os valores inteiros. Deve-se utilizar os tipos: int, float, double, long, Float, e Double nos parâmetros. Implemente uma classe principal e faça os teste de todos os métodos. Exemplos:
Assinatura: `int nomedometodo(int, float);`
Assinatura: `int nomedometodo(float, double);`
Assinatura: `int nomedometodo(int, Double);`
Obs.: Pesquisem na internet como fazer a conversão de tipos por referência (Float, Double, Integer, etc) para tipos primitivos (float, double, int, etc).
3. De acordo com o modelo UML abaixo, implemente a sobrescrita do método imprimir das classes **PessoaFisica** e **PessoaJuridica**. O método deve imprimir todos os atributos (inclusive da superclasse). Desenvolva um método **main** para teste das impressões.



4. Utilizando o modelo UML do exercício anterior. Implemente um método que imprima todas as informações de um determinado objeto. A assinatura do método é:
public static void imprimir(Pessoa pessoa).
Desenvolva um método **main** que utilize esse método **imprimir** passando como parâmetro objetos de: uma **Pessoa**, uma **PessoaFisica** e uma **PessoaJuridica**.
Dica 01: Use o comando **instanceof**.
Dica 02: Para usar o método de uma subclasse a partir de uma superclasse faça o uso do **cast** como mostra o exemplo abaixo:
- a. ((PessoaFisica)pessoa).getCpf()
 - b. ((PessoaJuridica)pessoa).getCnpj()
5. Utilizando o modelo UML da questão 3. Implemente as classes em Java e utilize a técnica de encapsulamento. Crie: os métodos **get** e **set** para todos os atributos, os construtores das classe, defina os modificadores de acesso (adequado) para cada atributo e método, e sobrescreva os métodos **toString** e **equals**.

