

Classes Abstratas e Interfaces

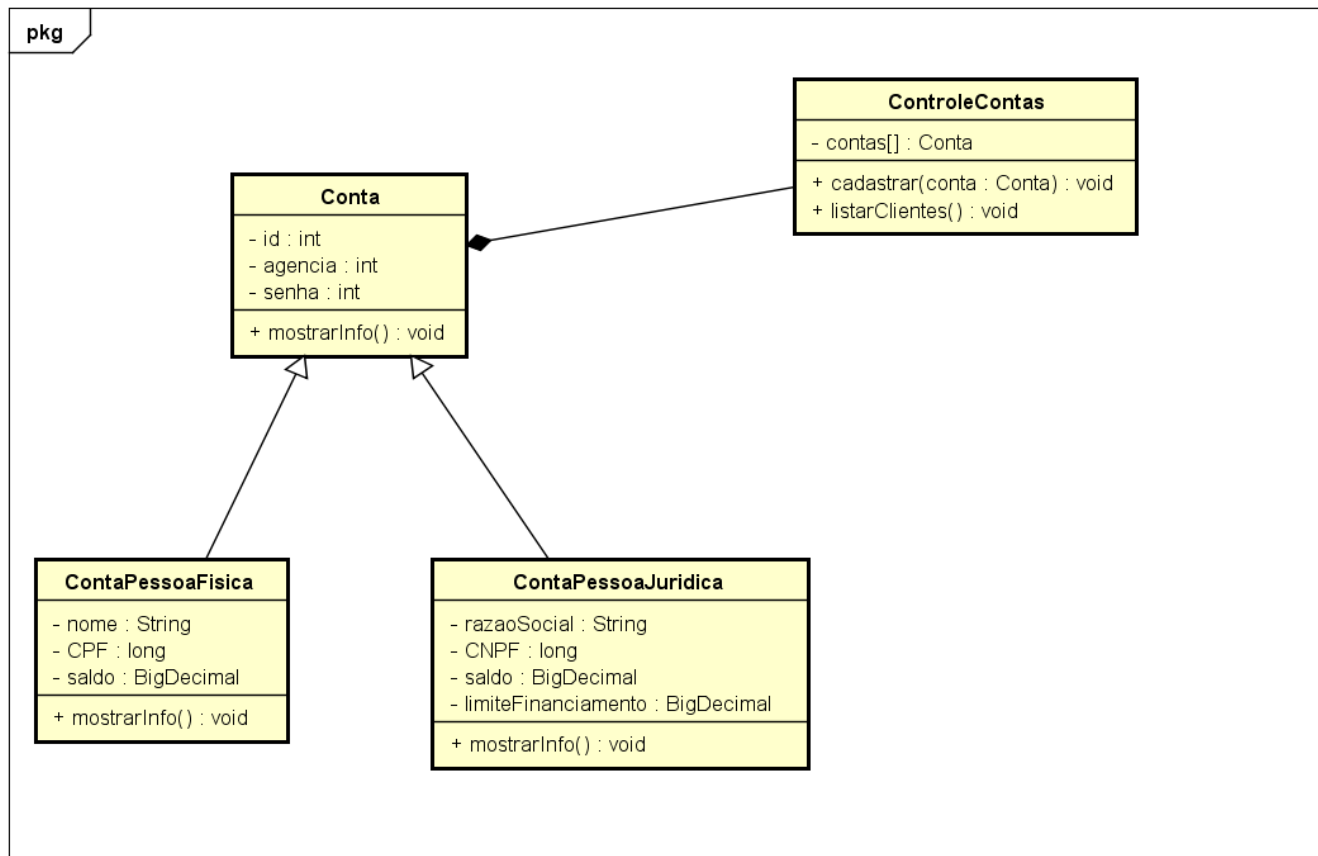
Capítulo VII e VIII

Sistema de conta bancária

1. Cria um sistema para controle de contas bancárias, sabendo que toda conta tem um número de identificação e uma agência. Existem vários tipos de conta, porém neste momento estamos interessados somente nos tipos abaixo:
 - i. Pessoa física: possui informações do proprietário (nome e CPF) além do saldo
 - ii. Pessoa jurídica: possui as informações da empresa (nome e CNPJ), além do saldo e limite de financiamento



Sistema de conta bancária



Ou uma conta é de Pessoa Física ou de Pessoa Jurídica. Como garantir que não haverá uma conta sem "dono"?

Sistema de conta bancária

```
public abstract class Conta {  
  
    private final int ID;  
    private final int AGENCIA;  
    private int senha;  
  
    public Conta(int id, int agencia){  
        this.ID = id;  
        this.AGENCIA = agencia;  
    }  
  
    public int getId() {  
        return ID;  
    }  
  
    public int getAgencia() {  
        return AGENCIA;  
    }  
  
    public int getSenha() {  
        return senha;  
    }  
  
    public void setSenha(int senha) {  
        this.senha = senha;  
    }  
  
    protected void mostrarInfo() {  
        System.out.println("ID da conta: "+ID);  
        System.out.println("Agencia da conta: "+AGENCIA);  
    }  
}
```

Tornamos a classe Conta abstrata através da utilização da palavra reservada **abstract**. Isso nos garante que nenhum objeto do tipo Conta seja instanciado

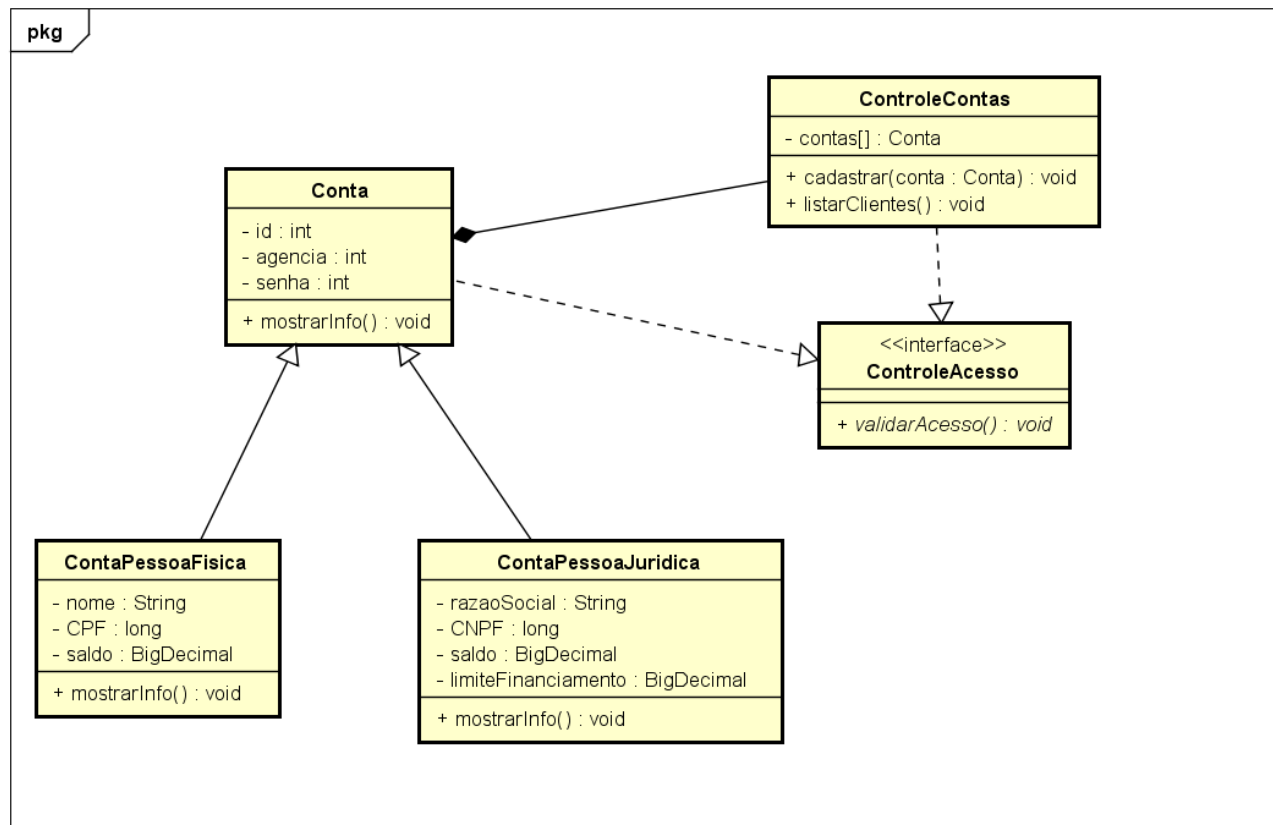
```
public class Main {  
    public static void main(String[] args) {  
  
        Conta c1 = new Conta(0, 0);  
  
    }  
}
```

Conta is abstract; cannot be instantiated

(Alt-Enter shows hints)

Sistema de conta bancária

E se fosse necessário implementar um controle de acesso no cadastro de contas?



Utilizando de uma **Interface** temos a garantia de que toda conta irá implementar o método `validarAcesso()`

Sistema de conta bancária

```
public interface ControleAcesso {  
    public abstract void validarAcesso(int senha);  
}
```

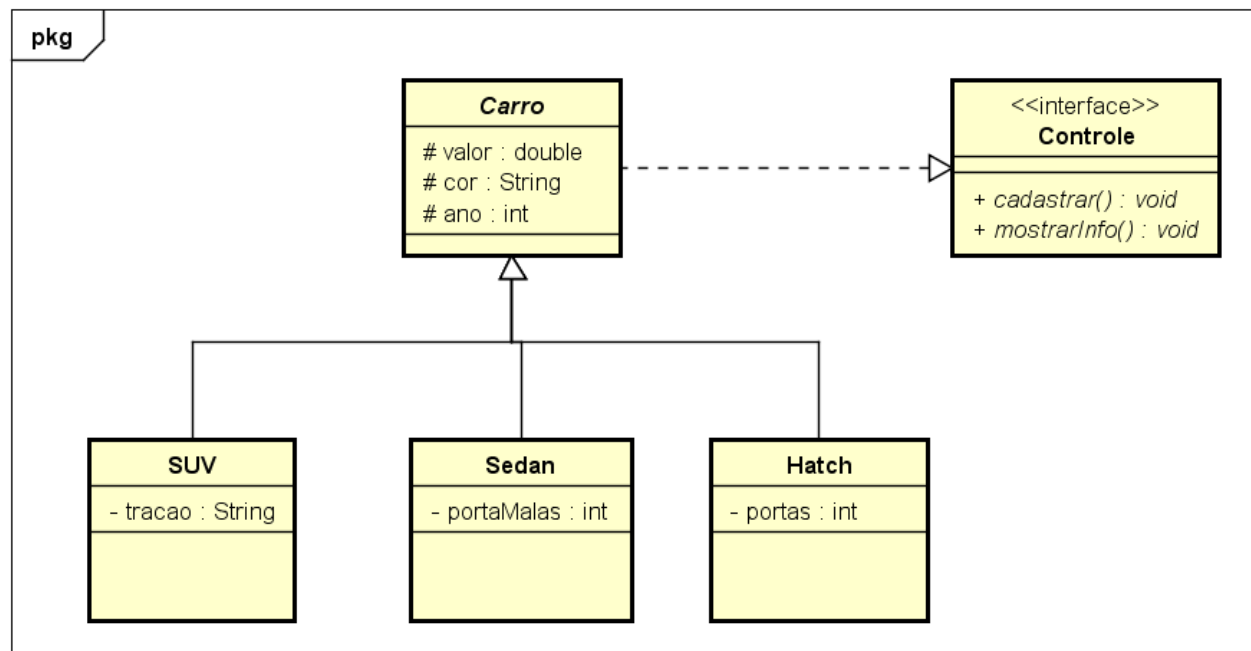
Declaramos uma interface utilizando a palavra reservada **interface**.

Para uma classe implementar uma interface, ela deve declarar sua necessidade através da palavra reservada **implements**

```
public class CadastroContas implements ControleAcesso{  
    private Conta[] contas;  
  
    public CadastroContas(){  
        contas = new Conta[0];  
    }  
  
    public void cadastrar(Conta conta){  
        Conta[] aux = new Conta[contas.length+1];  
        for(int i = 0; i < contas.length; i++){  
            aux[i] = contas[i];  
        }  
        aux[contas.length] = conta;  
        contas = aux;  
    }  
  
    public Conta[] listarClientes(){  
        return contas;  
    }  
  
    @Override  
    public void validarAcesso(int senha) {  
        System.out.println("Usuário logado!!");  
    }  
}
```

Exercícios

1. Desenvolva um sistema para a concessionária CarroFeliz



2. Você trabalha em um grande varejista, e foi designado para criar um sistema de cadastro das mercadorias da empresa. Neste primeiro momento serão inseridas as informações dos eletrodomésticos abaixo:
 1. Mercadoria: código, fabricante, descrição e valor
 2. Fogão: quantidade de bocas, tipo de acendimento (manual ou automático) e tipo de gás
 3. Geladeira: quantidade de portas, tamanho em litros e se ela é inox ou não

Obs.: faça um menu simples para inclusão e consulta em ambos exercícios

Obrigado!