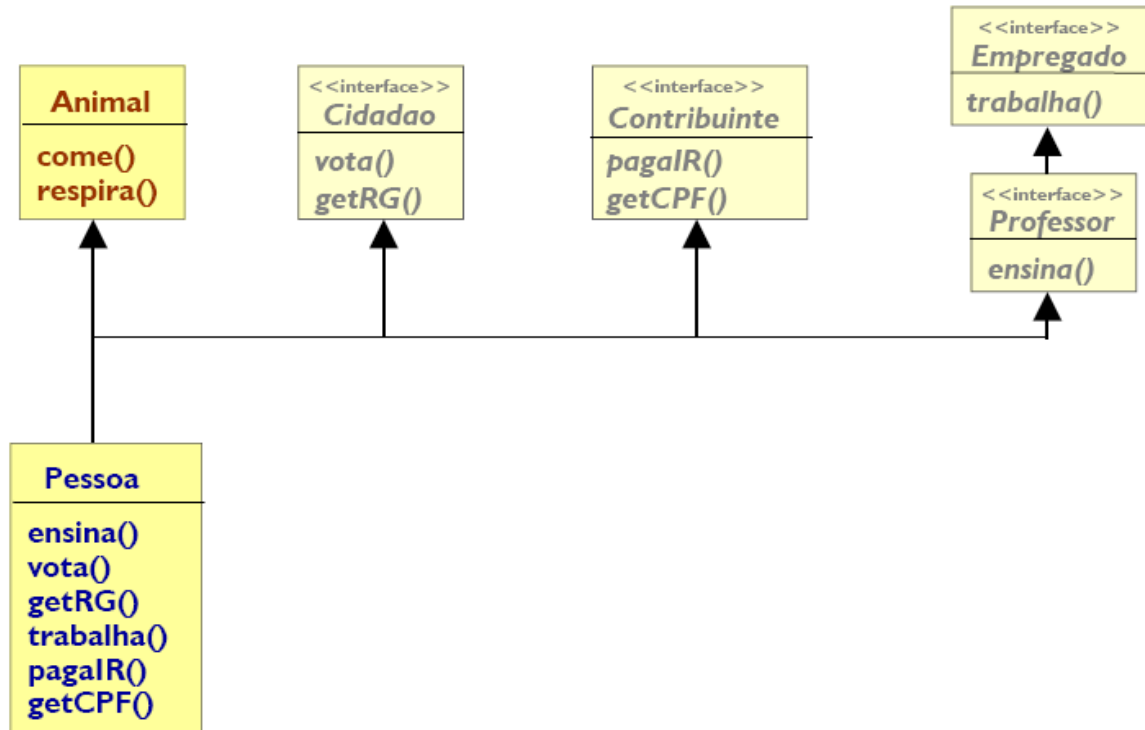


**Lista de Exercício - Classe Interface**  
**Prof. Dr. Daniel Pires**

**Exercício 1)**

De acordo com o Diagrama a seguir, responda:



- Crie uma classe chamada **Animal** contendo os métodos `come()` e `respira()`;
- Crie uma interface chamada **Cidadao** contendo os métodos abstratos `vota()` e `getRG()`;
- Crie uma interface chamada **Contribuinte** contendo os métodos abstratos `pagaIR()` e `getCPF()`;
- Crie uma interface chamada **Empregado** contendo o método abstrato `trabalha()`;
- Crie uma interface chamada **Professor** contendo o métodos abstrato `ensina()` e que herda da interface **Empregado**;
- Crie uma classe **Pessoa** que herda da classe **Animal**, e implementa as interfaces **Cidadao**, **Contribuinte**, e **Professor**. Coloque um texto qualquer em cada método com o `System.out.println()` para mostrar que o método foi chamado;
- Faça testes deixando de implementar certos métodos na classe **Pessoa** para ver as mensagens de erro obtidas

h) Faça uma classe chamada TestaPessoa. Crie um objeto do tipo Pessoa e faça-o chamar todos os métodos possíveis. Veja quais são os possíveis.

i) Faça uma classe chamada Cidade. Crie os seguintes métodos:

```
public void contrata(Professor p){  
    p.ensina();  
    p.trabalha();  
}
```

```
public void contrata(Empregado e){  
    e.trabalha();  
}
```

```
public void cobraDe(Contribuinte c) {  
    c.pagaIR();  
}
```

```
public void registra(Cidadao c) {  
    c.getRg();  
}
```

```
public void alimenta(Animal a ){  
    a.come();  
}
```

j) Faça uma classe chamada TestaPessoaCidade. Crie um objeto do tipo Pessoa e um objeto do tipo Cidadao. Faça chamadas aos métodos como o código a seguir. Analise e anote os resultados.

```
a) Pessoa fulano = new Pessoa();  
b) Cidade rp = new Cidade();  
c) rp.contrata(fulano);  
d) rp.contrata((Empregado) fulano);  
e) rp.cobraDe(fulano);  
f) rp.registra(fulano);  
g) rp.alimenta(fulano);
```

## **Exercício 2)**

Quais as diferenças entre uma classe interface e uma classe abstrata?

## **Exercício 3)**

Apresente uma implementação de uma classe adaptadora. Faça classes e interfaces se forem necessárias. Não vale fazer idêntico ao exercício feito em sala de aula