



# Projeto de Sistemas II

Faculdade Prof. Miguel Ângelo da Silva Santos

## Material 2- Classes abstratas, concretas e interfaces

Professor: Isac Mendes Lacerda, M.Sc., PMP, CSM  
e-mail: [isac.curso@gmail.com](mailto:isac.curso@gmail.com)

# Métodos concretos e abstratos

- **Método abstrato** → composto apenas por assinatura
  - Uma declaração de responsabilidade, mas sem a capacidade de cumpri-la, em função da ausência de algoritmo
  - Grafado em itálico

# Métodos concretos e abstratos

- **Método concreto** → composto por assinatura e corpo
  - Corpo é composto pelo algoritmo de interesse
  - Grafado **sem** itálico

# Classes concretas e abstratas

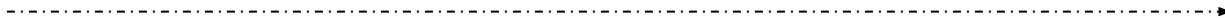
- **Classe concreta** → possui exclusivamente métodos concretos

# Classes concretas e abstratas

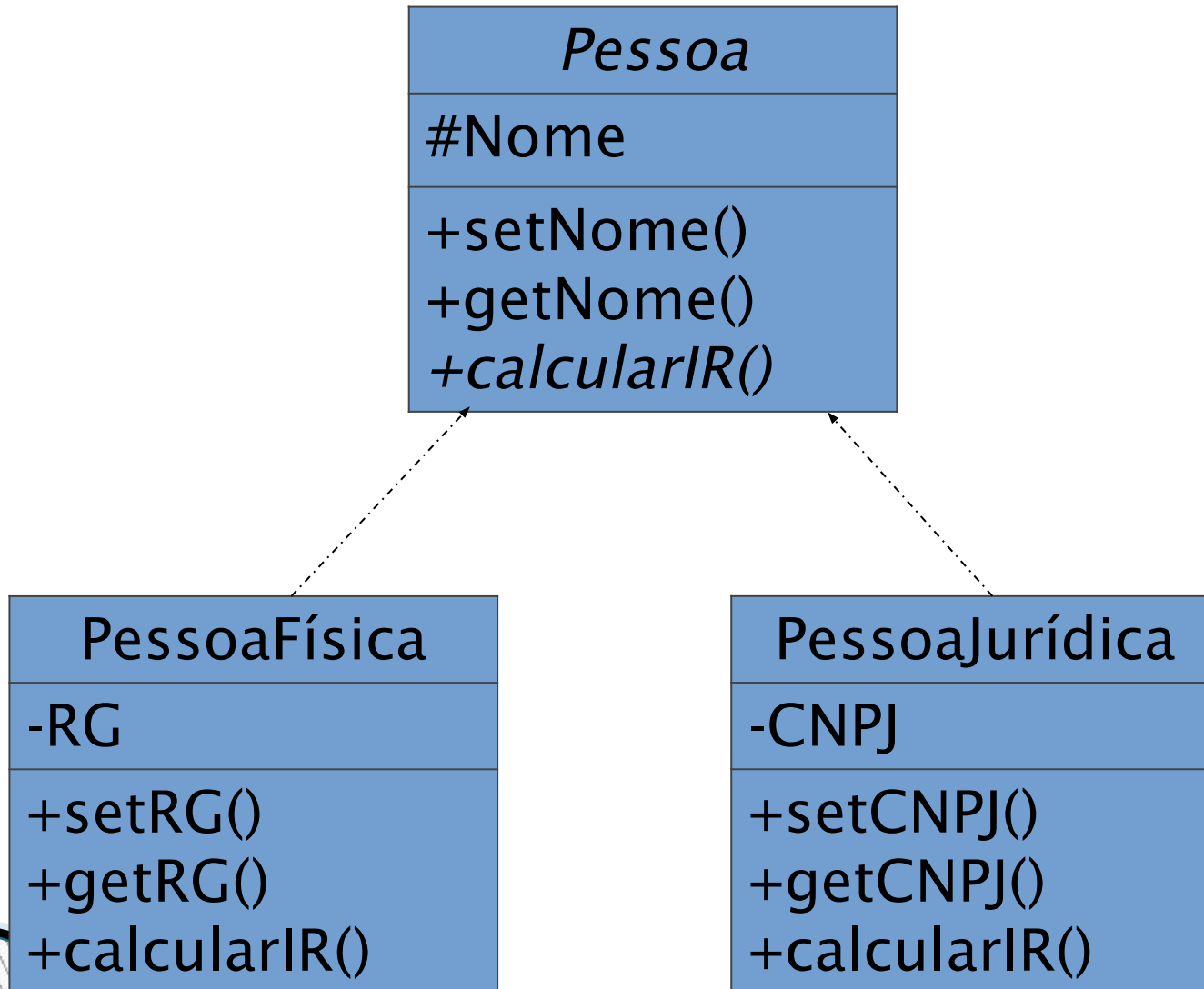
- **Classe abstrata** → possui pelo menos um método abstrato
  - Identificador da classe grafado em itálico
  - Nem todas as responsabilidades da classe materializadas em capacidades
    - Nem todos os métodos possuem algoritmo definido
    - Não pode originar instâncias em tempo de execução

# Classes concretas e abstratas

- **Realização** → associação entre dois elementos em que um deles especifica uma responsabilidade a ser implementada e o outro incorpora a obrigação de implementá-la
  - Usado para associar uma classe concreta a uma classe abstrata ou interface.



# Classes concretas e abstratas: um exemplo



# Classes concretas e abstratas: um exemplo

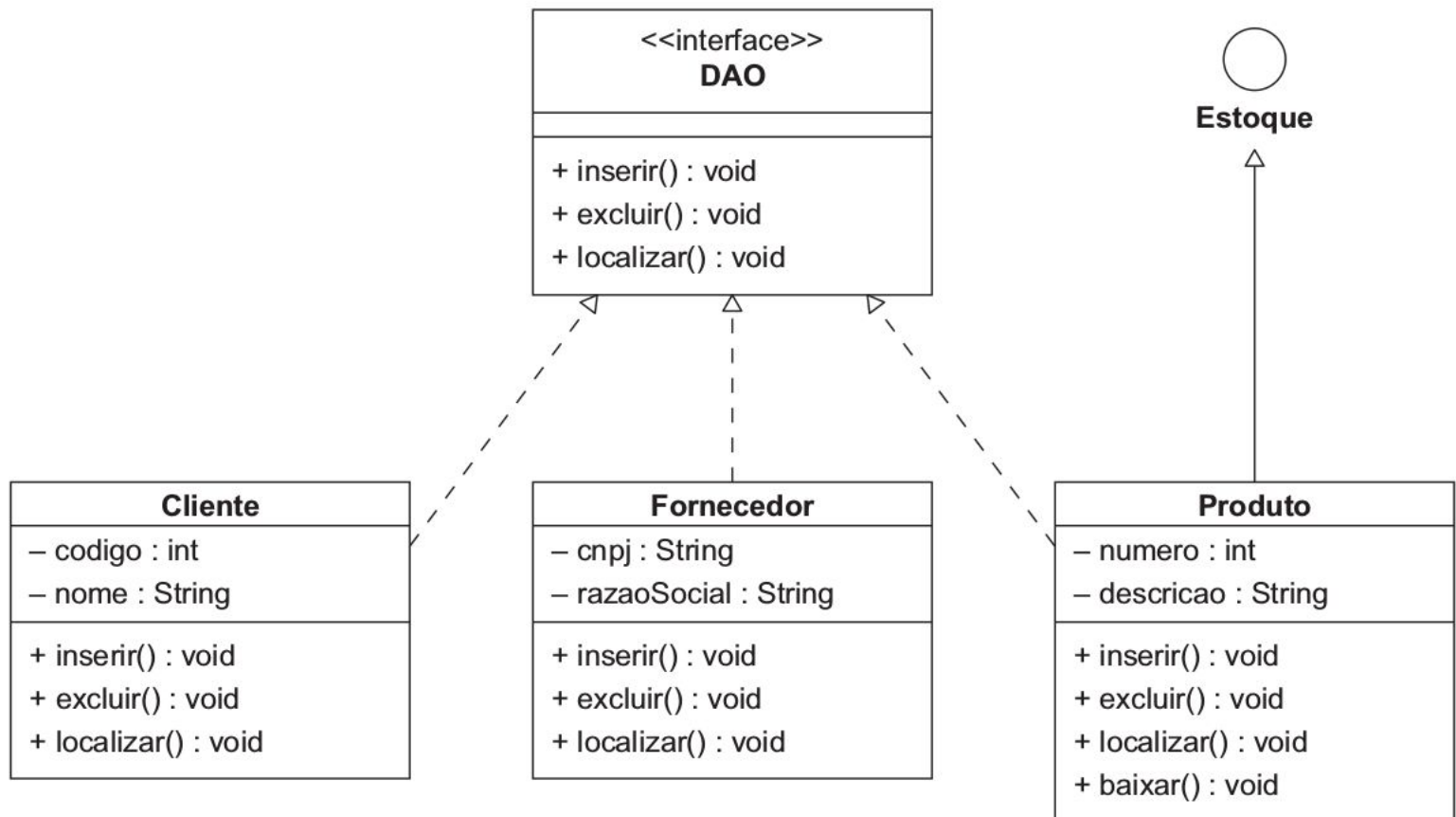
```
14 public abstract class ClasseAbstrata {  
15     public abstract void método1();  
16     public void método2(){  
17         System.out.println("Olá, sou o método 2");  
18     }  
19     public abstract void método3();  
20 }  
  
21 public abstract class ClasseAbstrata2 extends ClasseAbstrata{  
22     //public abstract void método1();  
23     public void método3(){  
24         System.out.println("Olá, eu sou o método 3!");  
25     }  
26 }  
  
27 public class ClasseConcreta extends ClasseAbstrata2{  
28     public void método1(){  
29         System.out.println("Olá, eu sou o método1!");  
30     }  
31 }  
32 }
```



# Interfaces

- Em algumas linguagens podemos separar totalmente as fachadas dos métodos de suas implementações.
- Quando todos os métodos são abstratos podemos chamar esse conjunto de INTERFACE.

# Interfaces: um exemplo



# Interfaces: um exemplo

```
15 public interface IT {  
    public void método1();  
    public int método2();  
}
```

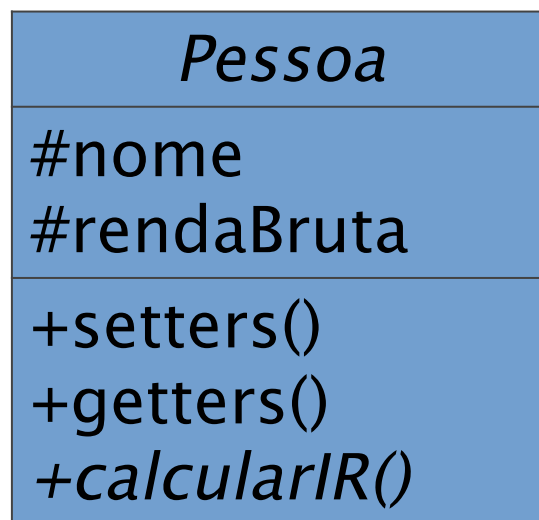
```
12 public class ClasseConcreta implements IT{  
14     public void método1(){  
15         System.out.println("Olá, eu sou o método um!");  
16     }  
17     public int método2(){  
18         int número = 10;  
19         return (número);  
20     }  
}
```

# Exercício 1



- ❑ Crie uma interface, usando java, chamada Controle que especifique os métodos “andar”, “virar” e “falar”. A seguir, crie uma classe chamada Robô que implemente esses métodos. Dentro de cada método imprima uma mensagem em tela contendo a ação correspondente. Para testar, use a classe chamada UsaRobo como principal.

# Exercício 2: Implemente

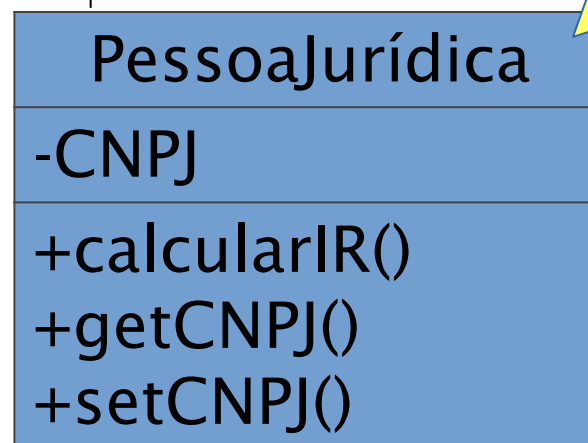
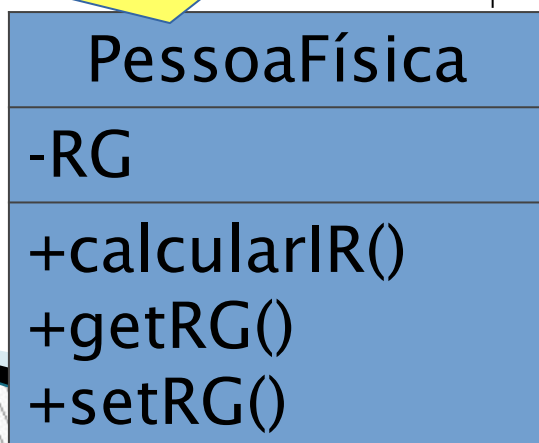


## CalcularIR():

Representa o % da renda bruta:  
.0% se for até 1500;  
.10% se for entre 1501-3000;  
.20% se for entre 3001-10000;  
.27 se for acima de 10000.

## CalcularIR():

Representa o % da renda bruta:  
.0% se for até 30000;  
.10% se for entre 30000-50000;  
.20% se for entre 50000-90000;  
.27% se for acima de 90000.



# Exercício 3: Implemente



Implemente a estrutura apresentada e crie uma instância de cada classe:

