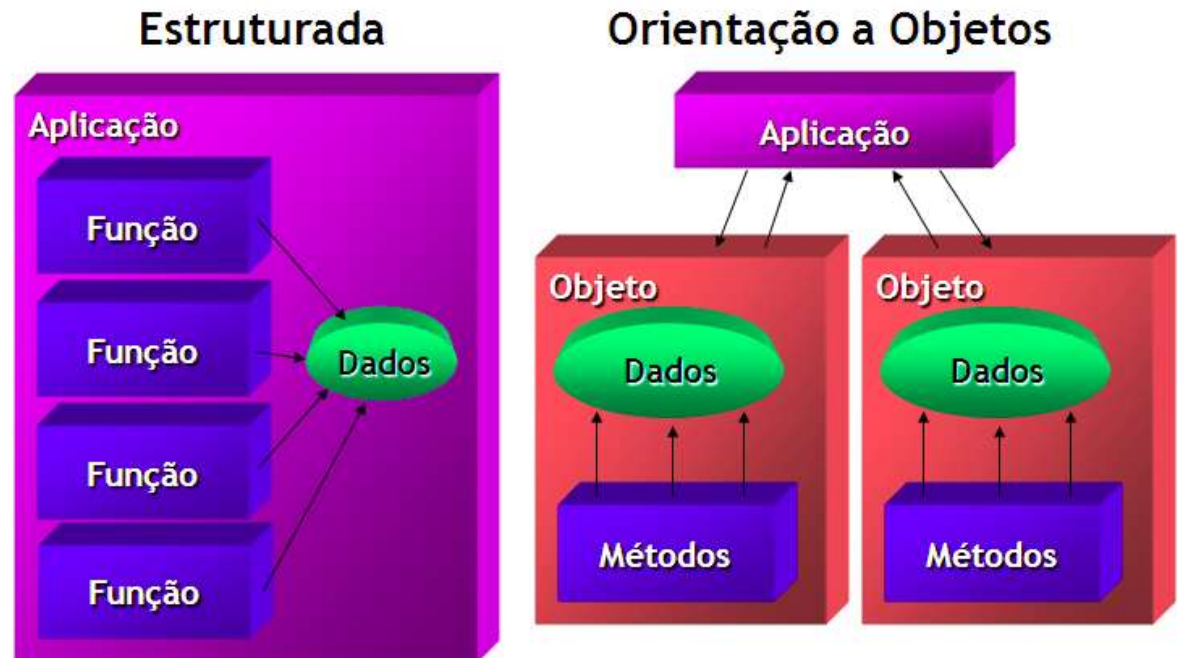


Modelagem de Software

Prof. Esp. Erinaldo Sanches Nascimento
erinaldonascimento@yahoo.com.br

- UML (*Unified Modeling Language*).
- Adotada em 1997 pela OMG (*Object Management Group*).
- Linguagem de modelagem orientada a objetos.
- Apoia e incentiva as boas práticas para os projetos de desenvolvimento de software.
- Representação gráfica parcial de um sistema.

- Utiliza objetos que colaboram entre si para a construção de um software.
- A colaboração é feita por meio da troca de mensagens.







Conta

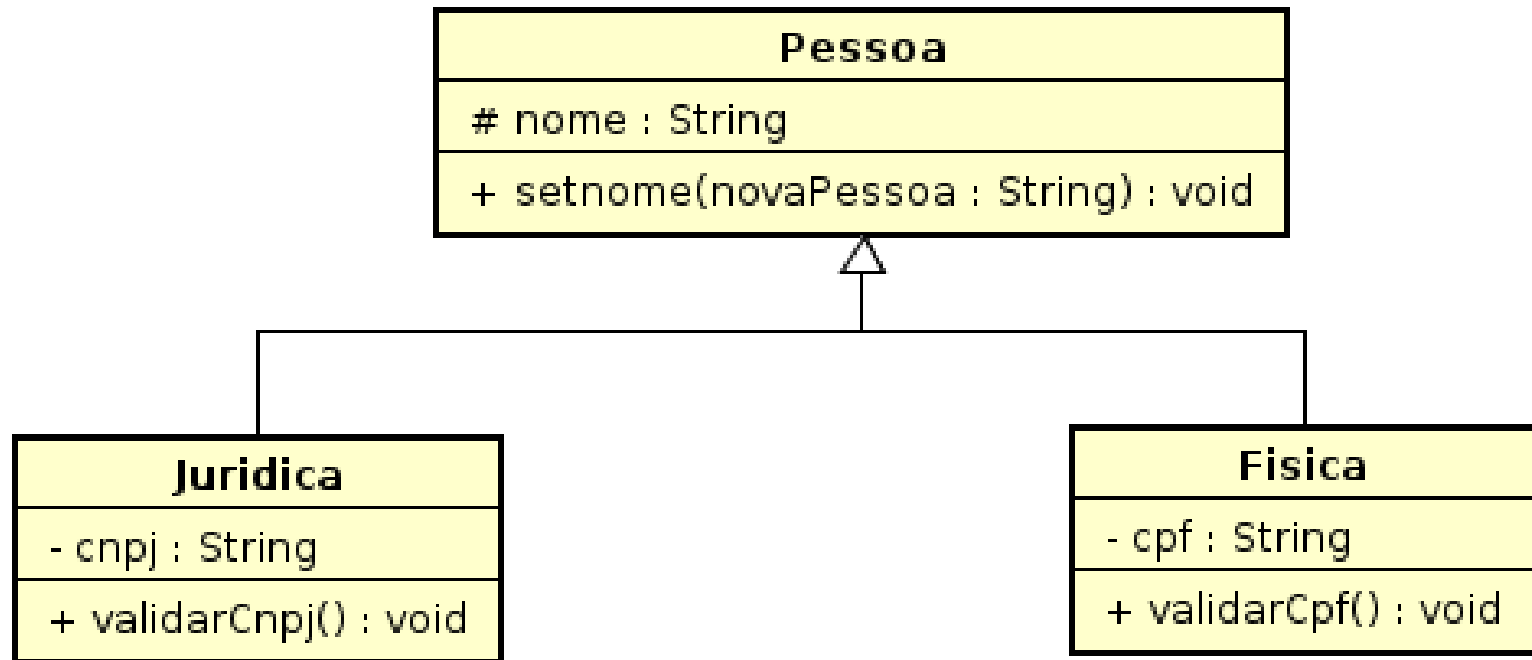
- tipoConta : int
- cliente : String
- dataAbertura : Date
- saldo : float

- + sacar(valor : float) : void
- + depositar(valor : float) : void
- + consultar() : void

```
1 import java.util.Date;
2
3 public class Conta {
4
5     private int tipoConta;
6
7     private String cliente;
8
9     private Date dataAbertura;
10
11     private float saldo;
12
13     public void sacar(float valor) {
14
15     }
16
17     public void depositar(float valor) {
18
19     }
20
21     public void consultar() {
22
23     }
24
25 }
```



```
26     public int getTipoConta() {
27         return tipoConta;
28     }
29
30     public void setTipoConta(int tipoConta) {
31         this.tipoConta = tipoConta;
32     }
33
34     public String getCliente() {
35         return cliente;
36     }
37
38     public void setCliente(String cliente) {
39         this.cliente = cliente;
40     }
41
42     public Date getDataAbertura() {
43         return dataAbertura;
44     }
45
46     public void setDataAbertura(Date dataAbertura) {
47         this.dataAbertura = dataAbertura;
48     }
49
50     public float getSaldo() {
51         return saldo;
52     }
53
54     public void setSaldo(float saldo) {
55         this.saldo = saldo;
56     }
57
58 }
```

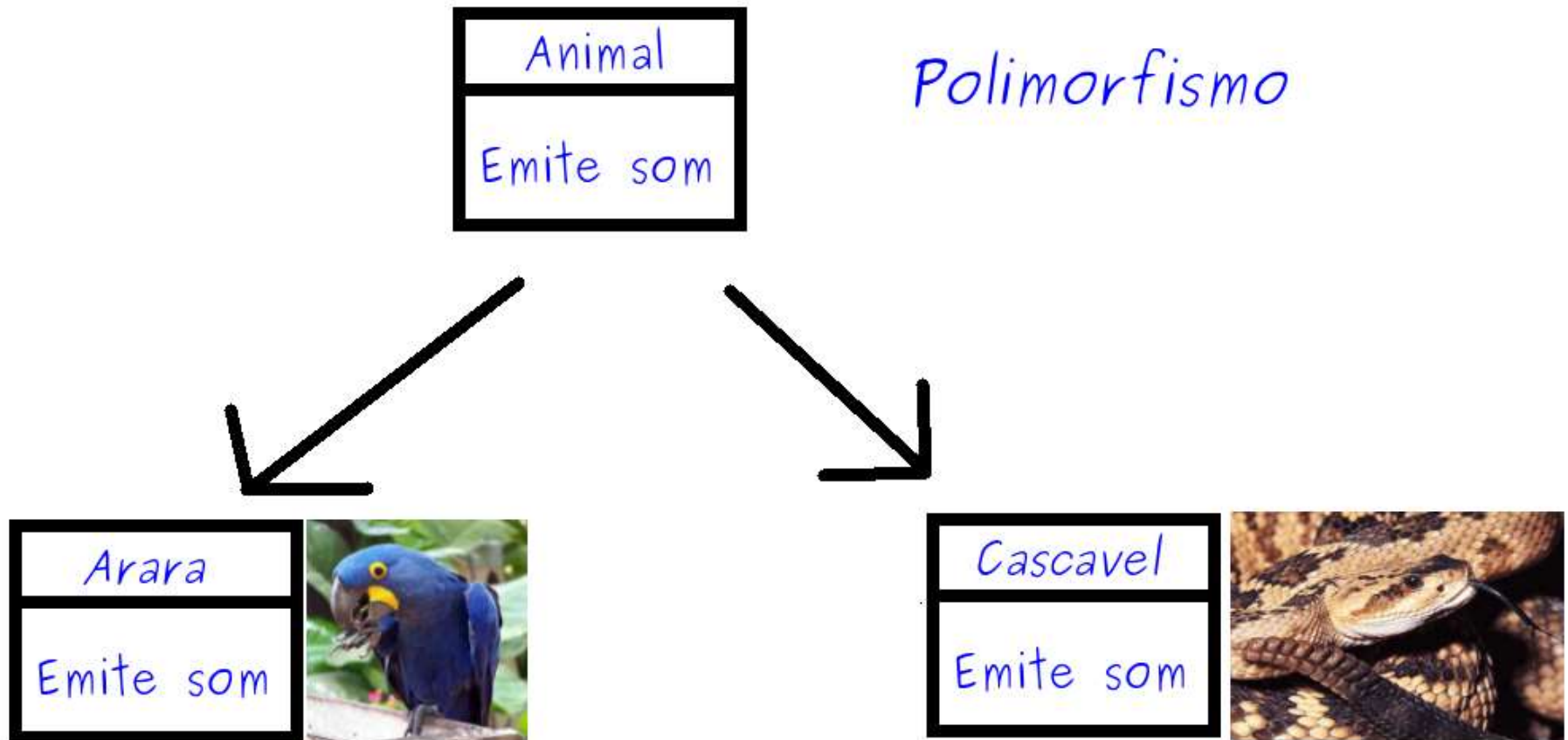



```
1 public class Pessoa {  
2  
3     protected String nome;  
4  
5     public void setnome(String novaPessoa) {  
6  
7     }  
8 |  
9 }
```

```
1 public class Fisica extends Pessoa {  
2  
3     private String cpf;  
4  
5     public void validarCpf() {  
6  
7     }  
8  
9 }
```

```
1 public class Juridica extends Pessoa {  
2  
3     private String cnpj;  
4  
5     public void validarCnpj() {  
6  
7     }  
8  
9 }
```

- Capacidade de assumir diferentes formas.
- Possibilita que métodos herdados de uma superclasse sejam reescritos.
 - Uma mensagem pode ser interpretada de maneiras diferentes.
- Não tem como modelar o polimorfismo.



- Orientação a objetos é um conjunto de encapsulamento com abstração e polimorfismo.
- A abstração permite estudar uma entidade complexa abstraindo dela somente os detalhes que interessam em um determinado momento.
- Facilita a manutenção e reuso de pacotes.
- Princípio de dependência entre objetos.