



Programação Orientada a Objetos (POO)

Parte II - Herança



Implementar vs utilizar classes

Implementar

- Estabelecer **o que é a classe**.
 - Há atributos de dados que caracterizam a classe.
- Estabelecer **o que a classe faz**.
 - Há métodos que caracterizam as funcionalidades da classe

Utilizar

- Criar instâncias de um objeto.
 - Define-se um novo tipo de objeto a ser manipulado.
- Manipular tais objetos
 - Executar operações com tais objetos.



Implementar vs utilizar classes

- Contém a definição de dados e métodos comuns a todas as instâncias da classe.
- O nome da classe define um novo tipo

```
class Ponto(object)
```

- `self` é um parâmetro para acessar atributos relativos a uma instância da classe (objeto).

```
self.x=x
```



Implementar vs utilizar classes

- A instância de uma classe é um objeto específico

```
p = Ponto(3, 4)
```

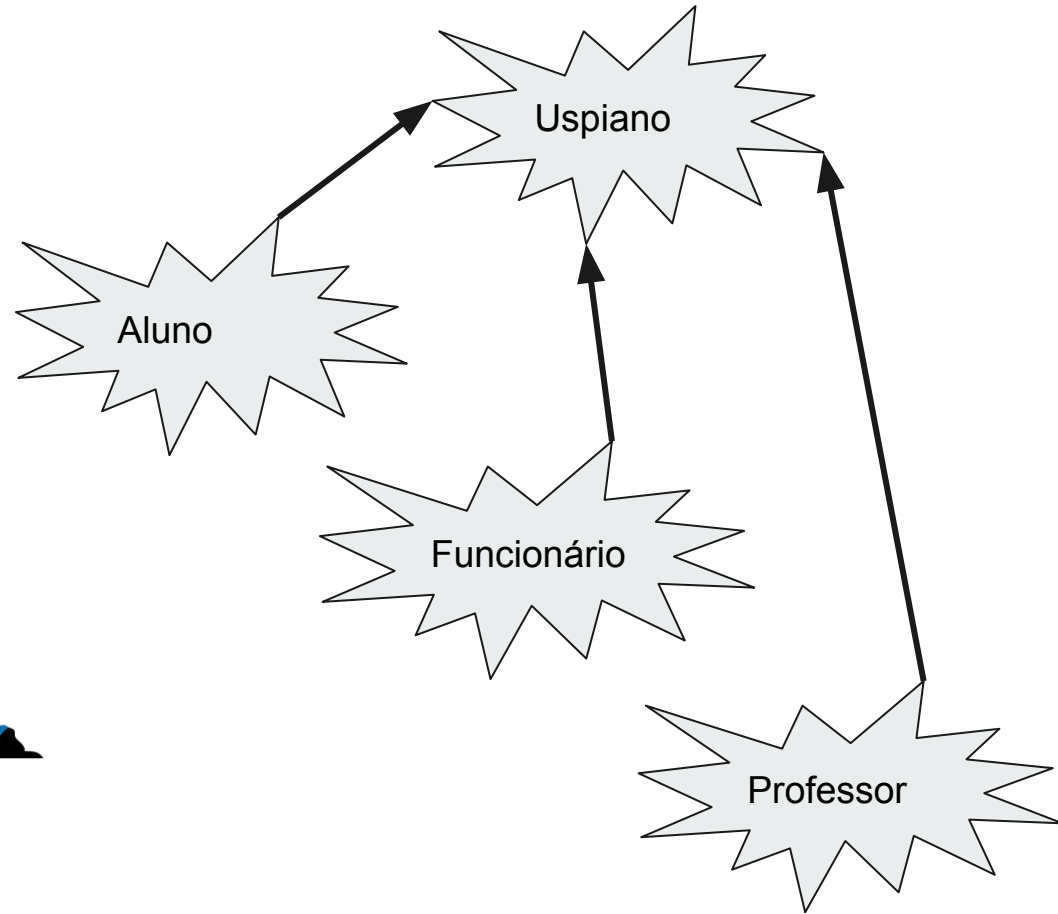
- Duas instâncias de uma mesma classe podem apresentar valores diferentes

```
p = Ponto(3, 4) q = Ponto(2, 1)
```

- A estrutura de duas instâncias de uma mesma classe é a mesma.

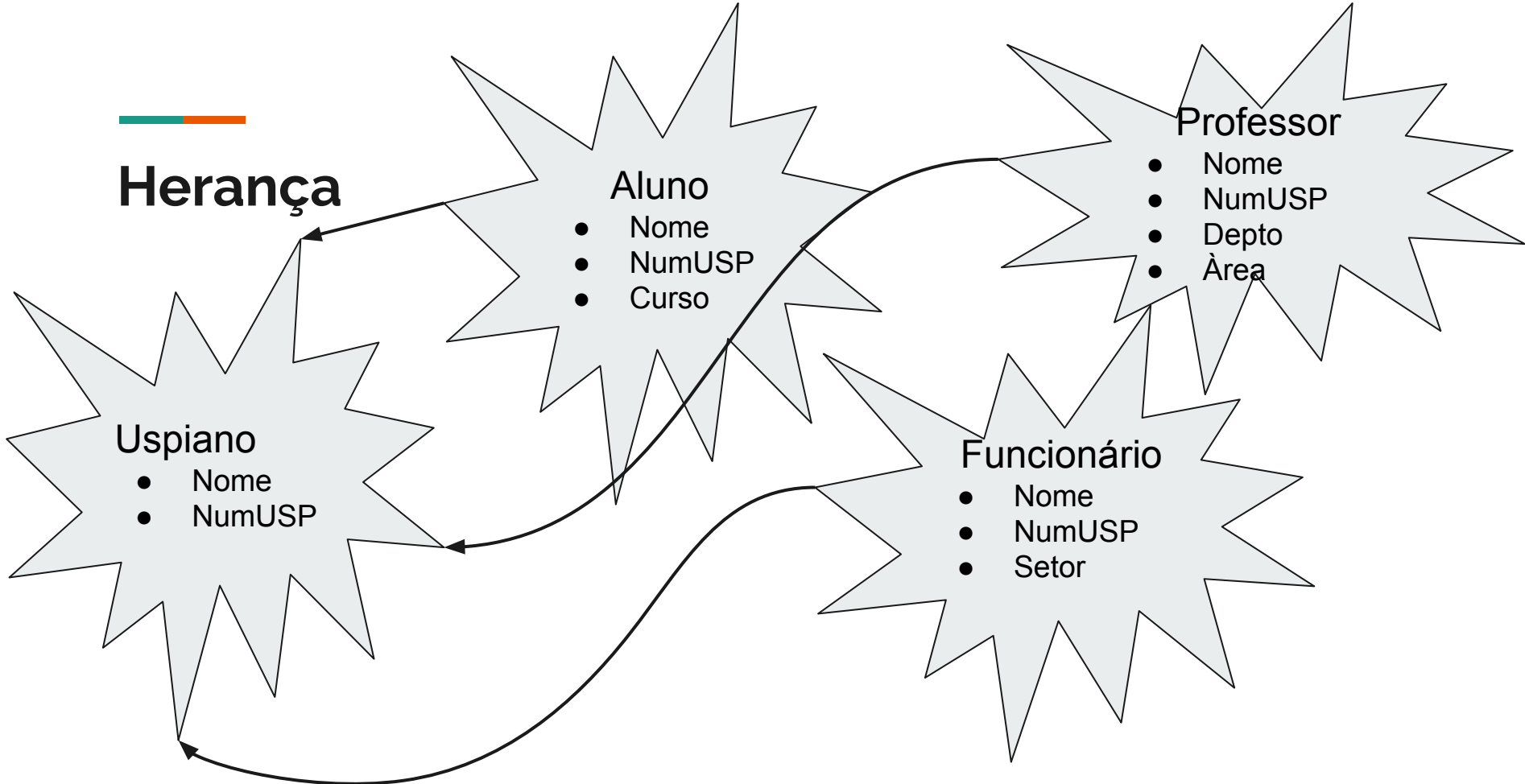


Herança





Herança





Herança

- Trata-se de uma forma bastante conveniente de relacionar grupos de abstrações.
- Torna-se possível estabelecer uma hierarquia onde classes filha herdam atributos das classes mãe.

Herança



Classe mãe -Superclasse

Pessoal

- Nome
- NumUSP

Classes filha -Subclasses

Aluno

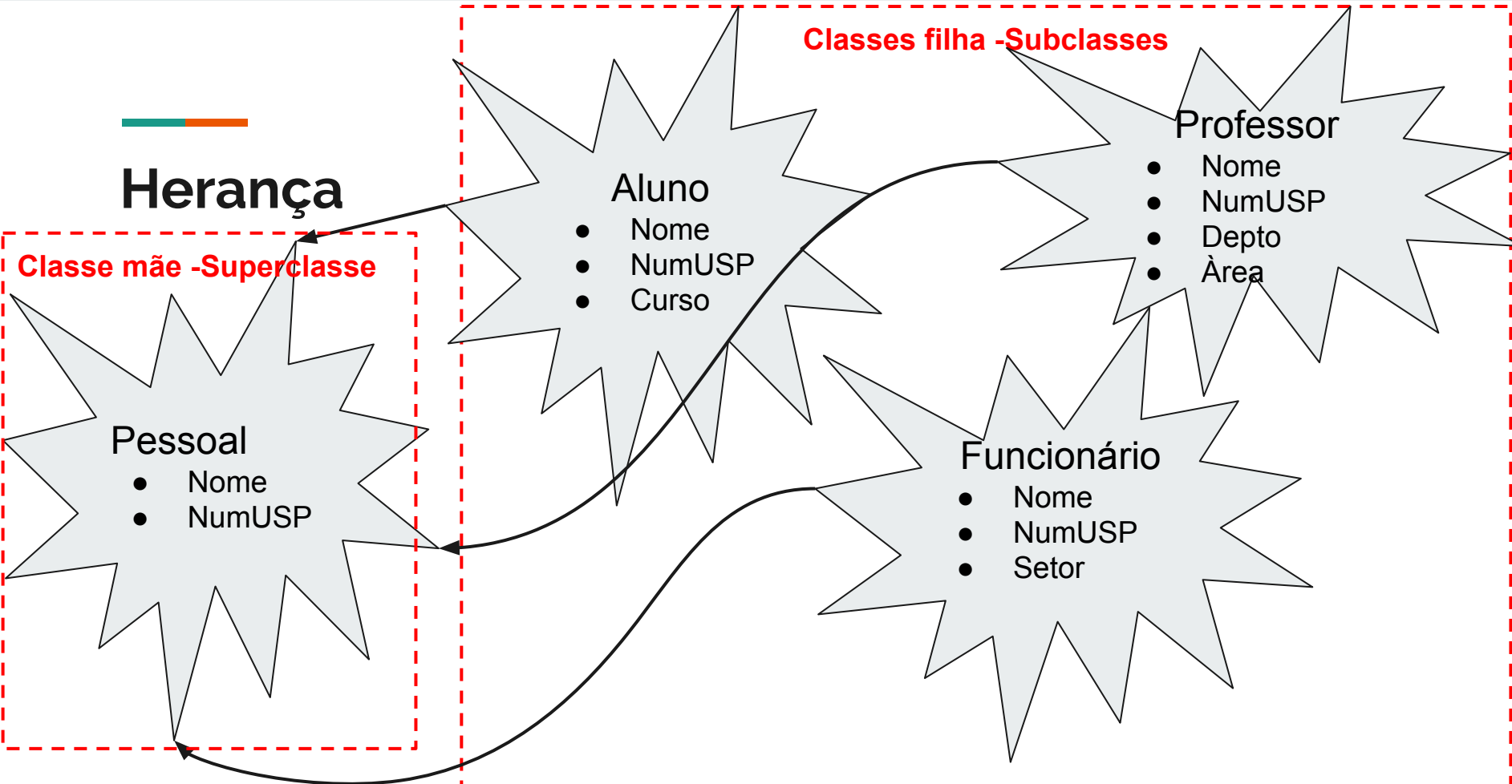
- Nome
- NumUSP
- Curso

Professor

- Nome
- NumUSP
- Depto
- Área

Funcionário

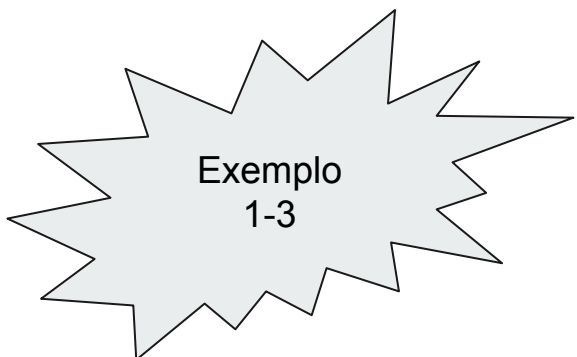
- Nome
- NumUSP
- Setor





Herança

- Subclasse herda os atributos da superclasse: dados e métodos.
- Na subclasse é possível:
 - Adicionar novos dados e métodos
 - Sobrescrever métodos da superclasse.



Exemplo
1-3