

Projeto de Sistemas I

Faculdade Prof. Miguel Ângelo da Silva Santos

Material 5 – Orientação a Objetos em Java (Herança)

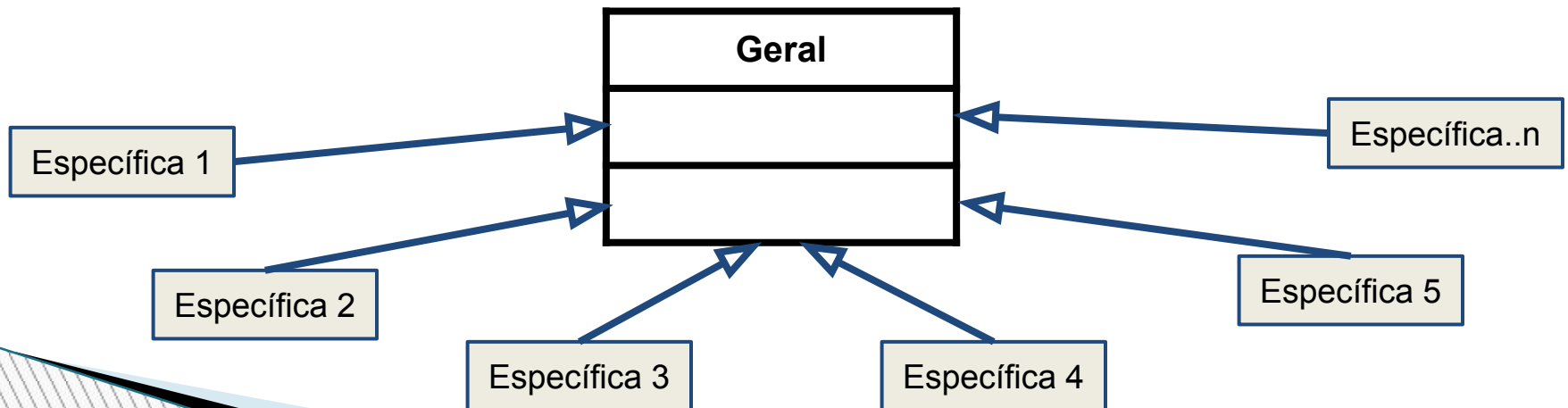
Professor: Isac Mendes Lacerda, M.Sc., PMP, CSM
e-mail: isac.curso@gmail.com

Tópicos

- Herança

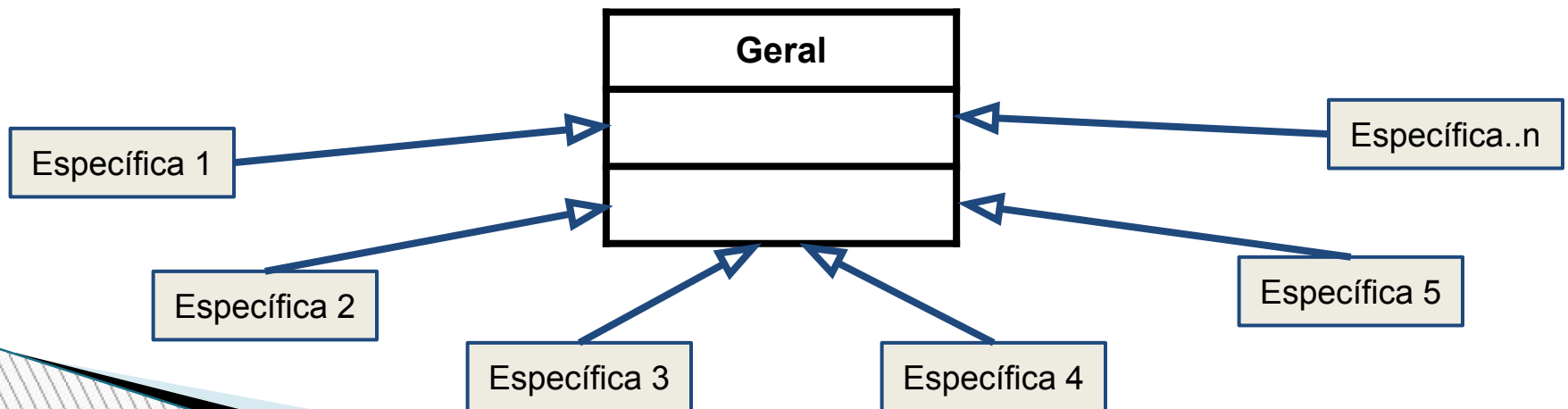
Herança

- Como você já sabe herança é uma maneira de reaproveitar código, generalizando o que é comum e especializando o que é particular entre classes.



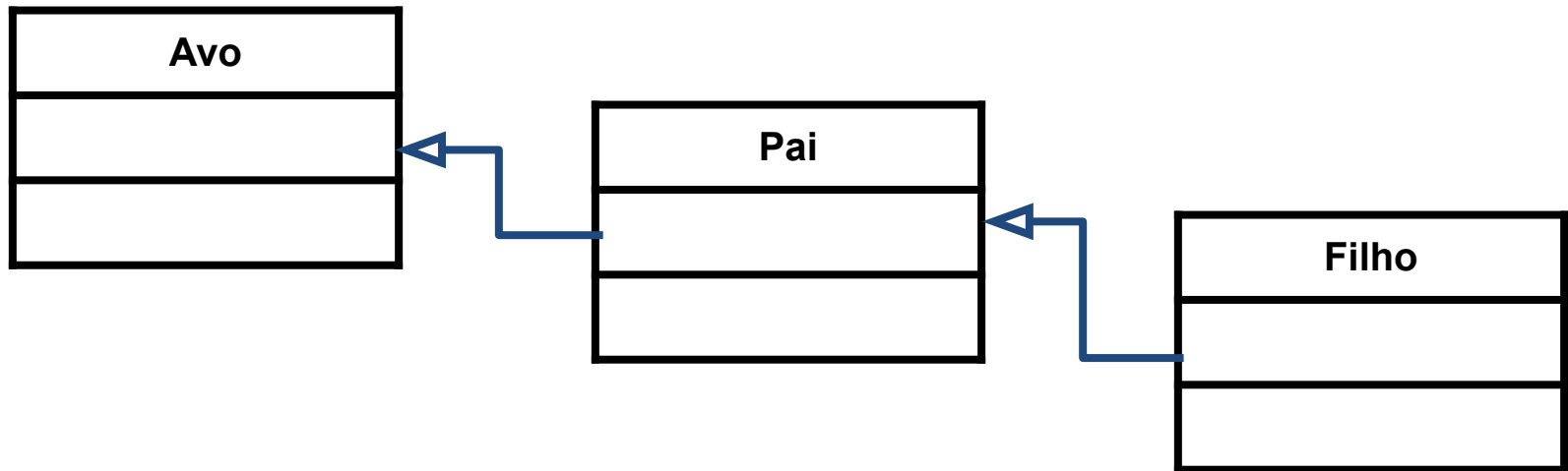
Herança

- Em Java, o tipo de herança praticado é o simples, ou seja, uma superclasse pode ter várias subclasses, mas um subclasse só tem uma superclasse.
- O número de níveis pode ser na medida da necessidade.



Herança

- Em Java, a herança ocorre através da palavra reservada ***extends***, indicada na linha de definição da subclasse.



Herança

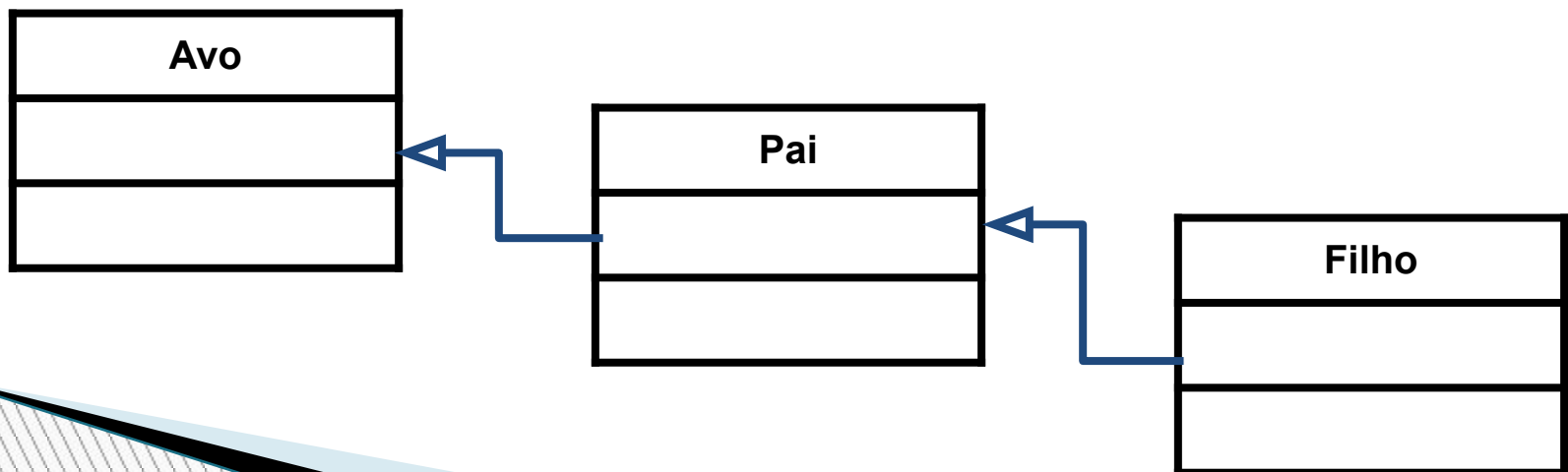
```
13 public class Avo {  
14     String nome;  
15     String cor_olhos;  
16  
17  
18  
19  
20     public void ler(){  
21         System.out.println("Lendo...");  
22     }  
23 }
```

```
13 public class Pai extends Avo{  
14     String cor_pele;  
15  
16     public void trabalhar(){  
17         System.out.println("Trabalhando...");  
18     }  
19 }
```

```
12 public class Filho extends Pai{  
13     String cor_cabelo;  
14  
15 }
```

Herança (constructores com *super*)

- Quando existe herança e alguma superclasse recebe parâmetros explicitamente em seu construtor, suas subclasses precisam fazer uso do comando **super**.
- Este comando ativa o construtor da superclasse na subclasse.



Herança

```
13 public class Avo {  
14     String nome;  
15     String cor_olhos;  
16     public Avo(String n, String co) {  
17         this.nome = n;  
18         this.cor_olhos = co;  
19     }  
20     public Avo() {  
21     }  
22     public void ler() {  
23         System.out.println("Lendo");  
24     }  
25 }
```

```
13 public class Pai extends Avo {  
14     String cor_pele;  
15     public Pai(String n, String co, String cp) {  
16         super(n, co);  
17         this.cor_pele = cp;  
18     }  
19     public Pai() {  
20     }  
21     public void trabalhar() {  
22         System.out.println("Trabalhando...");  
23     }  
24 }
```

```
12 public class Filho extends Pai {  
13     String cor_cabelo;  
14     public Filho (String n, String co, String cp, String cc) {  
15         super(n, co, cp);  
16         this.cor_cabelo = cc;  
17     }  
18     public Filho() {  
19     }  
20 }  
21  
22 }
```


Herança

Quais atributos e métodos estão disponíveis às instâncias dessas classes?

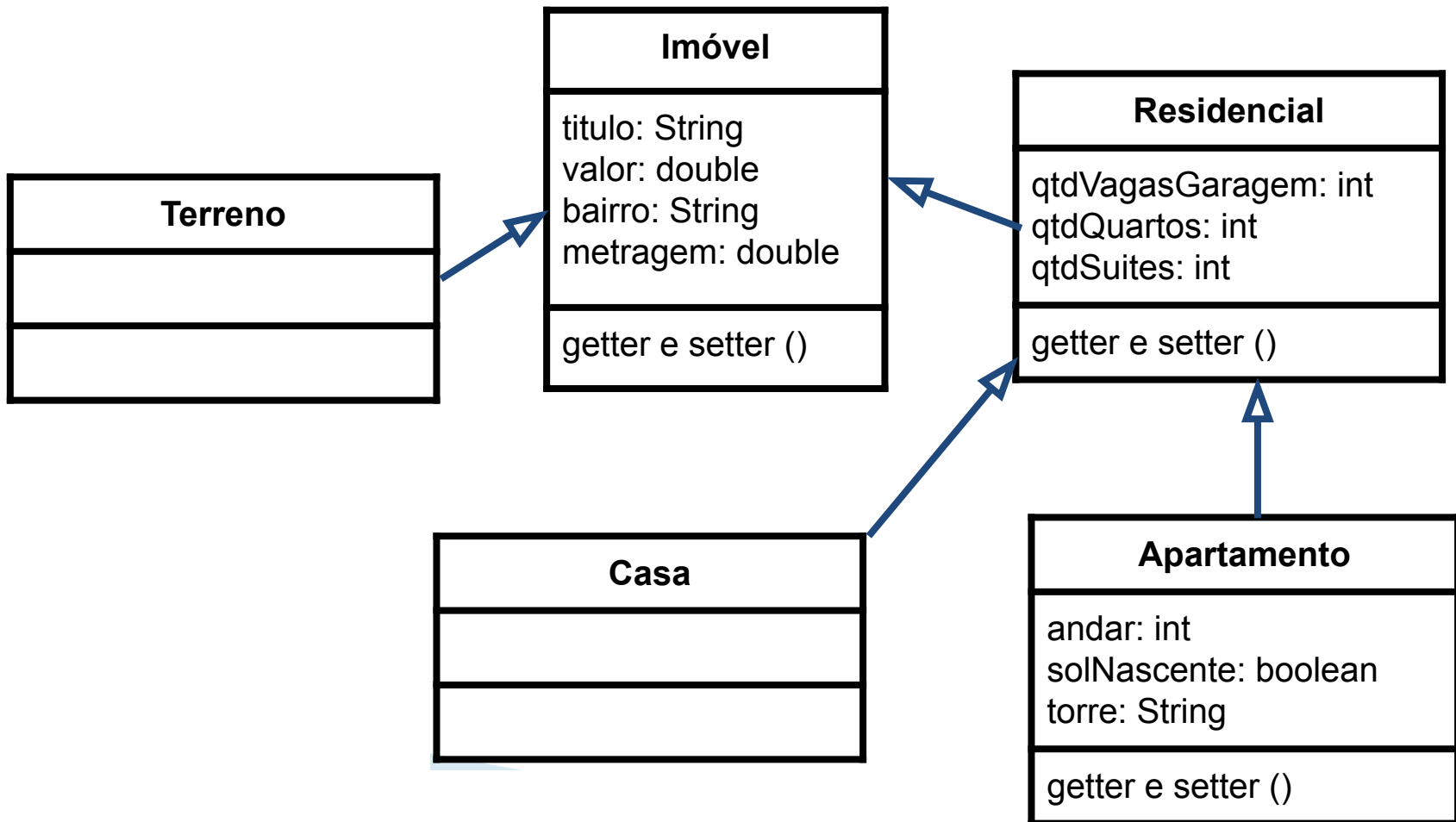
```
13 public class Avo {
14     String nome;
15     String cor_olhos;
16     public Avo(String n, String co) {
17         this.nome = n;
18         this.cor_olhos = co;
19     }
20     public Avo() {
21     }
22     public void ler() {
23         System.out.println("Lendo");
24     }
25 }
```

```
13 public class Pai extends Avo {
14     String cor_pele;
15     public Pai(String n, String co, String cp) {
16         super(n, co);
17         this.cor_pele = cp;
18     }
19     public Pai() {
20     }
21     public void trabalhar() {
22         System.out.println("Trabalhando...");
23     }
24 }
```

```
12 public class Filho extends Pai {
13     String cor_cabelo;
14     public Filho (String n, String co, String cp, String cc) {
15         super(n, co, cp);
16         this.cor_cabelo = cc;
17     }
18     public Filho() {
19     }
20 }
21
22 }
```

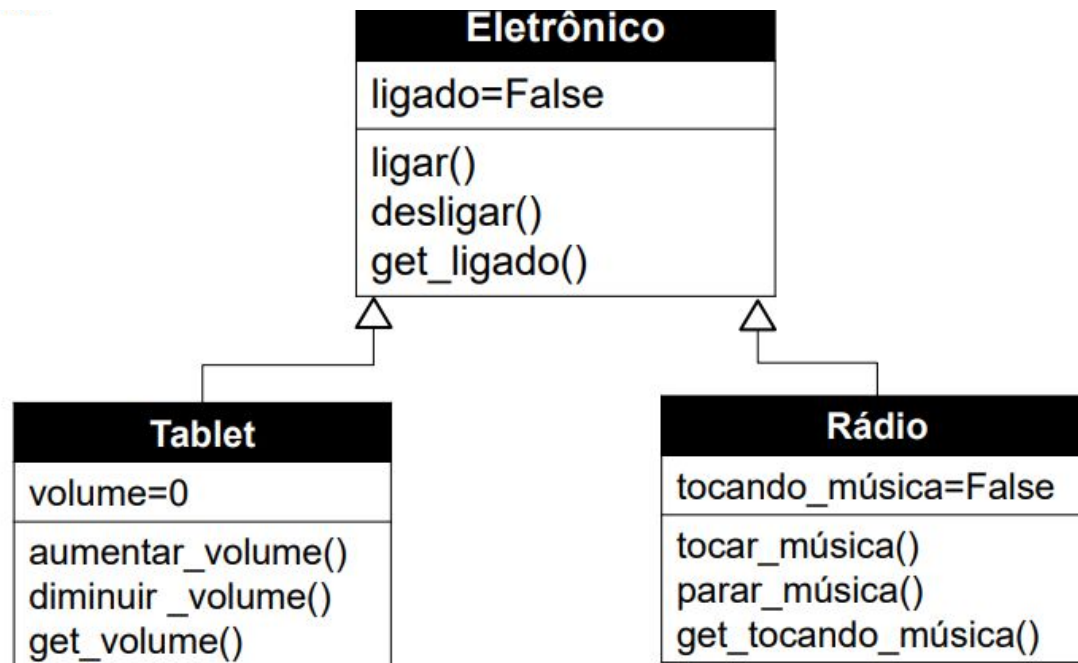
Exercício 1

Implemente as classes e instancie um objeto de cada uma e teste todos os métodos. Use a classe principal para testar.



Exercício 2

Implemente a estrutura proposta, crie uma instância de cada classe e teste todos os métodos. Os testes devem ser feitos na classe principal.



Exercício 3

Escreva uma classe que represente país. Armazene as seguintes informações dos países: nome, capital, dimensão, lista de países que faz fronteira. Represente a classe e forneça os seguintes métodos:

- Construtor que inicialize o nome, capital e a dimensão.
- Métodos de acesso (get e set) para as propriedades.
- Um método que permita verificar se dois países são iguais. Dois países são iguais se tiverem o mesmo nome e a mesma capital.
- Um método que define quais outros países fazem fronteira (note que um país não pode fazer fronteira com ele mesmo).
- Um método que retorne uma lista de países que fazem fronteira
- Um método que receba um outro país como parâmetro e retorne uma lista de vizinhos comuns aos dois países.