

SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE VIÇOSA · UFV

CAMPUS FLORESTAL

PROFESSOR: PHILIPPE DE FREITAS MELO MELO

SEMESTRE: 2025-2

QUESTÕES ABERTAS

Programação Orientada a Objetos

Érick Vinícius Issa Silva [5936]

Florestal - MG

2025

Lista 4

1) O polimorfismo é a capacidade que um método tem em, quando chamado, produzir um comportamento diferente, dependendo do objeto que o utiliza.

2) Polimorfismo de escrita: Override. Acontece quando uma subclasse redefine um método da superclasse. Ele é executado em tempo de execução e usado em herança.

Executado **em tempo de execução**.

Polimorfismo de Sobrecarga: Overload. Acontece quando vários métodos têm o mesmo nome porém assinaturas diferentes (parâmetros diferentes).

E ocorre **em tempo de compilação**.

Ex:

```
void somar(int a, int b);
```

```
void somar(double a, double b);
```

```
void somar(int a, int b, int c);
```

Polimorfismo de Inclusão. É o polimorfismo de referência: `Animal a = new Leao();`

3-a) Polimorfismo de sobrecarga. Overload.

b) as funções somar possuem mesmo nome, só que com parâmetros diferentes.

4) Não. Esse código não compila pois, eles possuem o mesmo nome e os mesmos parâmetros, apenas a troca de tipo no retorno da função mas isso não é permitido em java.

Dá erro de compilação pois o compilador vê os dois métodos assim:

```
Media(int, int)
```

```
Media(int, int)
```

5) Similarmente a questão 4, esse código também não compila, isso se deve ao nome da função e seus parâmetros serem iguais.

Ambas são::

```
calculaSalario(int, String)
```

Ou seja, mesmo que tenham retorno diferente e nomes de parâmetros recebidos diferentes ainda sim, são iguais a nível de compilador.

6) `byte → short → int → long → float → double`

`b,s - int - m1`

`i,s - int - m1`

`i, i - int m1`

`l,b - double - m2`

`f,s - double - m2`

d,b - double - m2

b,d - double - m3

i,l - double - m3

l, l - double - m3

d,f - double - m3

i,d - double - m3