INSTITUTO MAUÁ DE TECNOLOGIA



Linguagens I

Polimorfismo

Prof°. Tiago Sanches da Silva

Polimorfismo

Definição

Permite que um **mesmo nome** represente vários **comportamentos** diferentes.



Assinatura de uma função

A assinatura de uma função é definida pela **quantidade** e **tipos** dos parâmetros de entrada.

Quais funções possuem a mesma assinatura?

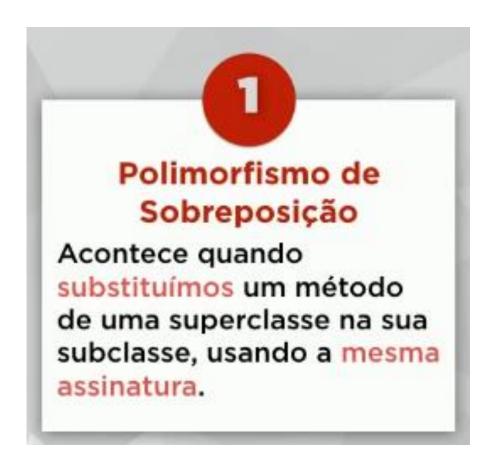
Quantidade e tipos dos parâmetros.

```
publico metodo calcMedia(n1: Real,
                         n2: Real): Real
publico metodo calcMedia(v1: Real,
                         v2: Real):Inteiro
publico metodo calcMedia(bim: Inteiro,
                         n1: Real,
                         n2: Real): Real
publico metodo calcMedia(n1: Real,
                         n2: Real,
                         n3: Real,
                         n4: Real): Real
publico metodo calcMedia(medMin: Real,
                         medMax: Real.
                         sit: Caractere,
                         bim: Inteiro)
                         :Caractere
```

Tipos de polimorfismo

Vamos focar nos dois principais tipos de polimorfismo:

- 1. Sobreposição
- Sobrecarga



Por exemplo, no exercício da concessionária a classe Funcionario possui um método para cálculo da bonificação, Gerente e Vendedor possuem outro método com a mesma assinatura, gerando assim uma sobrescrita no método da Superclasse Funcionário.

Ao chamar o método de um objeto Gerente que é uma subclasse de funcionário.

Qual será executado?

- O da Superclasse?
- O da Subclasse?
- Ambos?

Será executado o método que pertence a própria classe, no caso de um objeto gerente, será executado o método da classe Gerente.

a anotação @Override

Há como deixar explícito no seu código que determinador método é a reescrita de um método da sua classe mãe. Fazemos isso colocando @Override em cima do método. Isso é chamado anotação. Existem diversas anotações e cada uma vai ter um efeito diferente sobre seu código.

```
@Override
public double getBonificacao() {
  return this.salario * 0.15;
}
```

Repare que, por questões de compatibilidade, isso não é obrigatório. Mas caso um método esteja anotado com @Override, ele necessariamente precisa estar reescrevendo um método da classe mãe.

Obs: Em Linguagens 1 é obrigatório.

E se não quero que um determinado método seja sobrescrito pela classe herdeira?

Existe algum jeito de fazer isso?

Modificador: final

Ocorre em uma mesma classe quando existe mais de uma implementação diferente para um mesmo método. Para tento a assinatura da função deve ser diferente!



falar frase

agradável: abanar e latir agressiva: rosnar

horário do dia

manhã: abanar tarde: abanar e latir noite: ignorar

dono

é dono: abanar não é: rosnar e latir

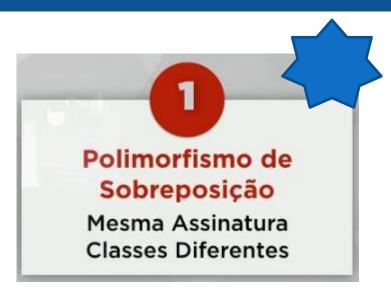
idade e peso

novo e leve: abanar novo e pesado: latir velho e leve: rosnar velho e pesado: ignorar

```
classe Cachorro estende Lobo
 publico metodo reagir(frase: Caractere)
 fimMetodo
 publico metodo reagir(hora, min: Inteiro)
 fimMetodo
 publico metodo reagir(dono: Logico)
 fimMetodo
 publico metodo reagir(idade: Inteiro,
                       peso: Real)
 fimMetodo
FimClasse
```

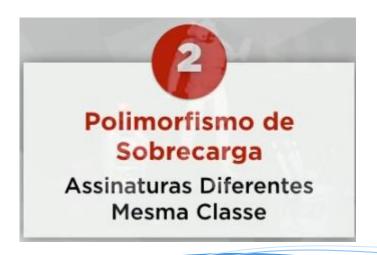
Assinaturas diferentes!

Polimorfismo



@Override

- Ligação dinâmica;
 - Cada vez que se aplica um método a um objeto, o compilador gera o código para calcular qual método chamar;
- O compilador não gera o código em tempo de compilação;



- Estática;
- Códigos diferentes gerados para assinaturas diferentes;

Perguntas?