

# Programação Modular - 11/03

---

## polimorfismo

---

- metodos iguais -> mesma assinatura
- comportamentos diferentes -> ações diferentes
- mesmo resultado semantico é interceção, significa fazer a mesma coisa
- Java utiliza referencias e por isso as comparações são mais facéis

## Dica INPORTANTE!

---

Se o meu código tiver algum if que dependa da classe, está errado

## Classificação de Cardeli

---

### Polimorfismo ad-hoc

Ad-Hoc significa Livre, independencia, sem viés. Livre de vinculo  
Em programação modular, significa que o programador não precisa se preocupar.  
Compilador quem faz o polimorfismo. Mas o compilador não é um genio

- Sobrecarga ( Overloading )
  - Permite definir funções com o mesmo nome, com o mesmo escopo, mas que se diferenciam em assinatura. Nesse caso, a assinatura é a diferença entre os parametros de uma função. EM classes abstratas declaramos sem corpo
  - Bloco de chaves é a definição de uma função.
  - Declarar, avisar que existe
    - Assinatura acontece na declaração, ao definir os parametros da função.
    - A função não depende do tipo de retorno da função
  - Definir, Especificar o comportamento
- Coerção
  - Ocorre quando você passa um objeto inesperado, e a função trata o valor, mesmo sem ser do tipo do parametro declarado.
  - "Antes de dar erro, será que eu consigo converter esse parametro sem assinatura"

```
// Sobrecarga de operador  
7 / 2 == 3  
7.0 / 2.0 == 3.5
```

---

## Polimorfismo universal

Polimorfismo programado a mão.

- Polimorfismo Universal
- Polimorfismo universal de inclusão
  - É o `@Override`
  - Funções de mesmo nome, mesma assinatura, mas com escopo diferente ou código diferente
  - Amarração tardia ( não faz do java uma linguagem interpretada )
  - Java toma decisão durante a execução, ou *runtime*
  - Perda de performance na hora de tomar decisão, MAS... o código poder ser mudado, se torna dinâmico.
  - **Requisitos:**
    - Herança
      - Dentro da classe filho existe a classe pai
    - Sobreposição de Métodos;
    - Referência de uma superclasse apontando para um objeto de uma subclasse ( Referências Polimórficas );

Livro bom para UML: Bezerra

## Proxima Aula: Dedicada a sobreposição de métodos;

---