



# Programação orientada a objetos

---

Maximilian Jaderson de Melo

Aula 9



# Introdução a Orientação a objetos



- Polimorfismo.



- Abstração (alicerce).
- Encapsulamento.
- Herança.
- **Polimorfismo.**
- *Composição e agregação*



- Do grego: *polýs* (muito) + *morphé* (forma)+ismo.
- Várias formas para a mesma funcionalidade.



- Sobrecarga.
- Sobreposição.
- Paramétrico.



- Quantos métodos/funções/sub-rotinas podem ser implementadas em um mesmo arquivo com o mesmo nome?
  - Somente uma.
- Exceto quando polimorfismo é suportado!



- É a possibilidade de ter em uma mesma classe vários métodos com o mesmo nome.
  - Com assinatura diferente.
  - Mas o que é a assinatura de um método?
  - Lembram-se dos protótipos das funções/subrotinas?





```
public float metodo(int p1, float p2){  
    //implementação  
}
```

- No exemplo acima, a assinatura (identificação do método) consiste de:
  - O nome (metodo).
  - O tipo de retorno (float).
  - O conjunto de parâmetros (int p1, float p2).



- Imagine a operação de soma (+).
- Ela serve apenas para um tipo de dados específico?
- A operação deveria somar qualquer comparação.
  - Inteiros com inteiros.
  - Inteiros com decimais e vice-versa.
  - Decimais com decimais.
- Ainda que seja algo óbvio para um ser humano, um computador faz somente aquilo que é programado a fazer.
  - Alguém precisa programar todas as possibilidades



- Pense no seguinte enunciado:
- Faça um programa que receba duas notas e imprima aprovado ou reprovado, conforme algum critério de avaliação.
  - O tipo de dados não está especificado. Qual tipo de dados escolher?



- Algunas posibilidades:

```
public float media(int n1, int n2){  
    return (n1+n2)/2.0;  
}
```

```
public float media(int n1, float n2){  
    return ((float)n1+n2)/2;  
}
```



- Algunas posibilidades:

```
public float media(float n1, int n2){  
    return (n1+(float)n2)/2.0;  
}
```

```
public float media(float n1, float n2){  
    return (n1+n2)/2;  
}
```



- É a conversão de um valor de um tipo de dados original para um segundo tipo.
- Pode ser implícito (promoção) ou explícito (coerção).
- Algumas linguagens fazem mais ou menos promoções de tipos de dados que outras.



- $n1 + n2$
- $n1$  é do tipo inteiro,  $n2$  é double.
- Somente podem ser operadas variáveis do mesmo tipo (soma no exemplo).
- É necessário decidir qual tipo é mais representativo (representa mais valores).
  - Entre inteiros e reais, qual conjunto é maior?
- $n1$  é promovida para double e em seguida a soma é realizada.
  - Caso seja possível, converta.



- No exemplo abaixo, como funciona o cast aplicado?

```
public float media(int n1, float n2){  
    return ((float)n1+n2)/2;  
}
```





- $(float)n1 + n2$
- $n1$  é do tipo inteiro,  $n2$  é float.
- Aqui,  $(float)n1$  indica a conversão forçada de  $n1$  para float.
  - O(a) programador(a) assume o risco.
  - CONVERTA!

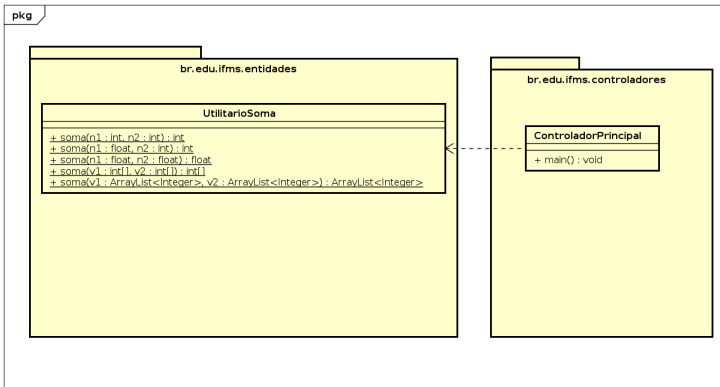


- Para haver a conversão os tipos precisam ser compatíveis.
  - String sempre é convertível para Integer?
- A Conversão de dados pode ser tanto explícita quanto implícita.
- Exemplo de cast implícito?
  - Herança.

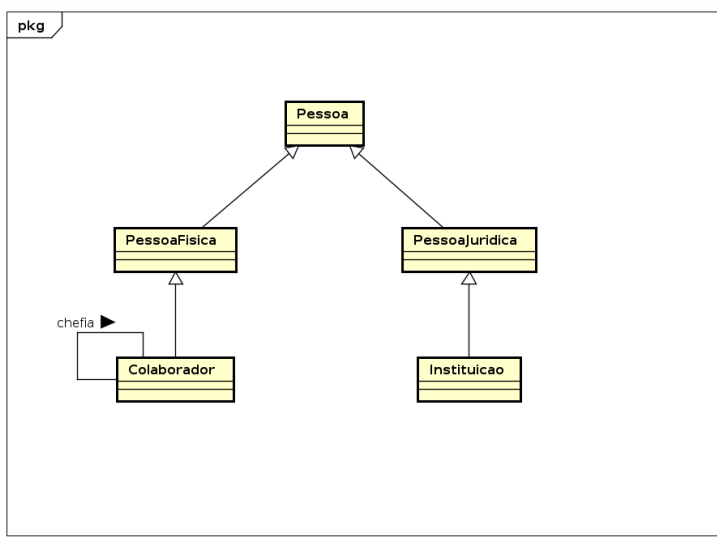


1. Faça o código do diagrama de classe sobre sobrecarga de soma (próximo slide).

# Exercício



# Exercício





- Aqui existe cast?

```
public static void main(String args[]){  
    Pessoa c1 = new Colaborador();  
}
```

- Sempre será possível instanciar na forma:
  - Superclasse objeto = new Subclasse();
  - A declaração é suficiente para o tipo da classe pai, mas na instância somente os elementos do filho são “inicializados”.



- É um operador para realizar cast explícito.

```
public void setChefe(Pessoa chefe){  
    if( p instanceof Colaborador ){  
        this.chefe = (Colaborador)chefe;  
    }  
    else{  
        System.out.println  
            ("O tipo informado é incompatível!!");  
    }  
}
```



- Dentro do método, é impossível saber o tipo da classe é Colaborador ou Instituição, caso o objeto chefe seja do segundo tipo, essa atribuição causaria um erro.





- Outro lugar que a sobrecarga é utilizada com frequência é nos construtores:
- Construtores também são métodos!
  - Apesar de não serem membros.



```
public Colaborador(){  
    nome = "";  
    cpf = "";  
}  
  
public Colaborador(String nome, String cpf){  
    setNome(nome);  
    setCpf(cpf);  
}
```



- Em C++ é possível adicionar comportamentos a operadores, como a soma (+).
  - Pesquise por sobrecarga de operadores [2].



- Polimorfismo por sobreposição.
- Polimorfismo paramétrico.



- [1] COELHO, ALEX. “JAVA-com orientação a objetos.” Editora Ciência Moderna (2012).
- [2] wikibooks C++.

# Dúvidas, críticas ou sugestões?



**maximilian.melo@ifms.edu.br**

**max.mjm.melo@gmail.com**