



Programação orientada a objetos

Maximilian Jaderson de Melo

Aula 8



Introdução a Orientação a objetos



- Herança.



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



- Um dos conceitos mais importantes da POO.
- Hierarquia.
- Inspiração na biologia.
- Herança familiar.



- Imagine que você tenha uma classe C e uma classe D.
- Ambas iguais, com exceção que D tem 2 atributos a mais que C. O que fazer?
 - Escrever duas vezes o mesmo código? E o reaproveitamento?
- Herança permite compartilhar os atributos entre classes **relacionadas**.



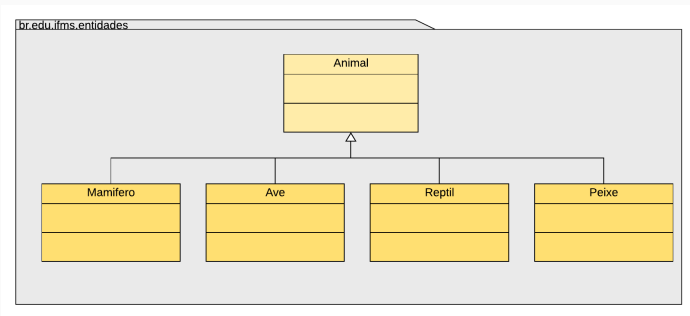
- Animais.
- Pessoas.
- Pessoas (cargos).
- Veículos.
- Obras (artes).



- A notação gráfica é feita pela linha ligando duas classes, com uma seta não preenchida.
- A direção da seta indica o sentido da Herança.
 - A ponta da seta indica o lado da super classe.
- As subclasses compartilham todos os atributos e métodos marcados como herdáveis.
 - Em java é feito pelo modificador “protected”.

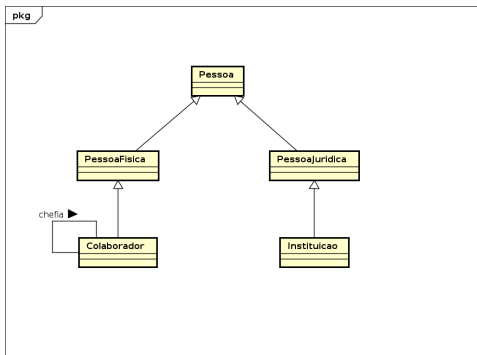


- Animais.





- Pessoas (cargos).





- Especificação vs generalização.
 - Sentido da leitura.
- Sinônimos:
 - super classe e subclasse.



- Palavra reservada **extends**.
- Especifica que uma subclasse estende o comportamento e atributos de uma super classe.



- Arquivo SuperClasse.java.

```
public class SuperClasse{  
    //atributo privado,  
    //exceto para subclasses  
    protected Integer atributo1;  
}
```

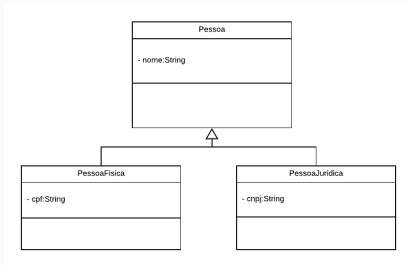


- Arquivo SubClasse.java.

```
public class SubClasse extends SuperClasse{  
    public static void main(String args[]){  
        SubClasse objeto = new SubClasse();  
        System.out.println( objeto.atributo1 );  
    }  
}
```



- Pessoa, Pessoa Física e Pessoa Jurídica





- Arquivo Pessoa.java.

```
public class Pessoa{  
    private String nome;  
  
    public String getNome(){  
        return nome;  
    }  
}
```




- Arquivo PessoaFisica.java.

```
public class PessoaFisica extends Pessoa{  
    public String cpf;  
}
```



- Arquivo PessoaJuridica.java.

```
public class PessoaJuridica extends Pessoa{  
    public String cnpj;  
  
    public static void main(String args[]){  
        Pessoa p1 = new Pessoa();  
        PessoaFisica p2 = new PessoaFisica();  
        PessoaJuridica p3 =  
            new PessoaJuridica();  
    }  
}
```



- Complete o quadro abaixo com S/N conforme o acesso para a classe/atributo.

Classe/atributo	nome	cpf	cnpj
Pessoa			
PessoaFisica			
PessoaJuridica			



- Mova a classe Pessoa para um pacote diferente de PessoaFisica e PessoaJuridica.
- Responda:
 - Quais atributos/métodos pertencem a p1?
 - E se o qualificador de acesso fosse *default* (sem qualificador)?
 - Se o qualificador de acesso fosse *protected* o quadro anterior mudaria?
 - Quais atributos/métodos pertencem a p2?
 - Quais atributos/métodos pertencem a p3?



- Como saber se há a relação de herança?
 - Quando é possível relacionar as classes pelo conectivo: “É um”.
 - Quando a frase fizer sentido, haverá herança.
 - “SubClasse é uma SuperClasse”.
 - PessoaFisica é uma Pessoa: fez sentido, há herança.
 - PessoaFisica é uma Escola: não fez sentido, não há herança.



- Quando uma subclasse herda características de mais de uma superclasse.
 - Um filho com dois pais.
- Java **não** implementa herança múltipla.
- C++ implementa herança múltipla.



- Para os exercícios 1,2,3, faça a leitura dos atributos por meio da classe Scanner e escreva os dados lidos na tela por meio de `System.out.println()`.
-
1. Crie as classes do exemplo de animais (Animal, Mamífero e Aves). Instancie 2 objetos (um de cada tipo).
 2. Crie as classes do exemplo de veículos (Veículo, Automovel e Motocicleta). Instancie 2 objetos (um de cada tipo).



3. Melhore o exemplo de pessoas, adicionando pelo menos um atributo em PessoaFisica e um em PessoaJuridica (um para cada).
4. Pesquise e disserte sobre classes abstratas. Descreva em quais classes dos exercícios 1,2,3 seria aplicável. Explique a relação entre ser solicitado duas instâncias quando em cada exercício há 3 três classes.



maximilian.melo@ifms.edu.br

max.mjm.melo@gmail.com