

# Herança

Prof. Hugo de Paula



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Curso de Engenharia de Software

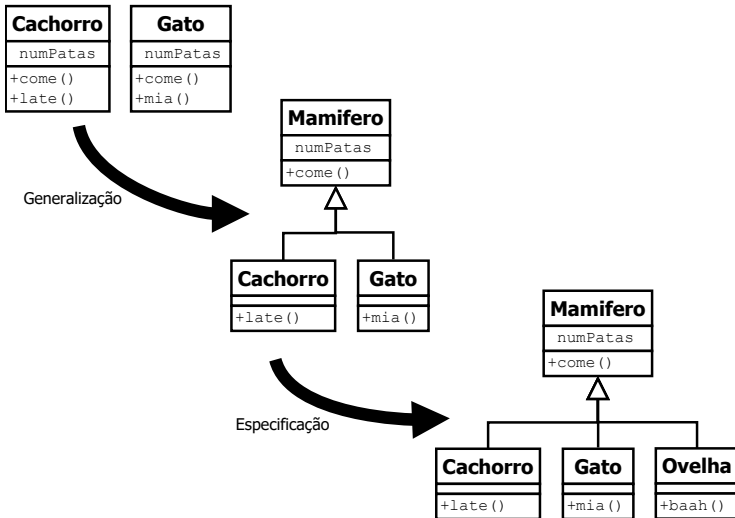
# Sumário

- 1 Herança
- 2 Herança simples
  - Construtores em classes estendidas
- 3 Exemplo: Herança de Produto

# Herança

- Mecanismo para definição de uma classe em termos de outra classe existente.
- Relação: *é um tipo de* / *é um*.
- Herança permite o reuso do comportamento de uma classe na definição de outra.
- A classe derivada herda todas as características de sua classe base, podendo adicionar novas características.
- Baseada em dois princípios fundamentais do projeto de software:
  - especificação (*top-down*)
  - generalização / abstração (*bottom-up*)

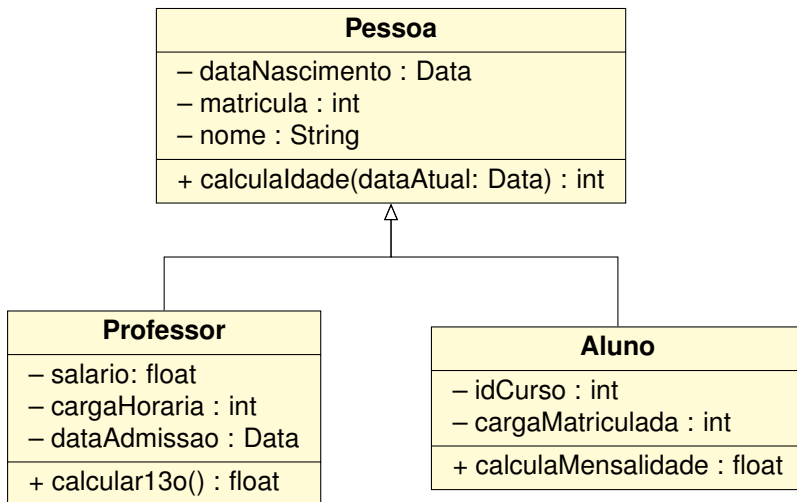
# Generalização / Especialização



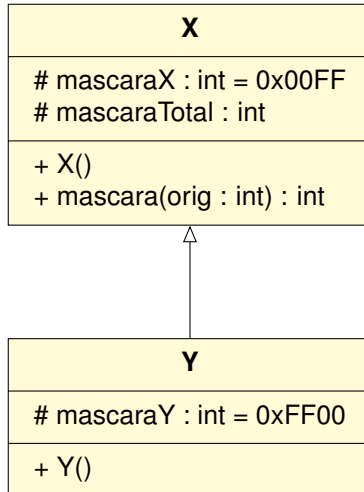
# Herança Simples

- Novas classes, chamadas de classes derivadas (ou subclasses), são definidas a partir de apenas uma classe base (ou superclasse).
- Exemplos:
  - Um professor é uma pessoa.
  - Um ônibus é um veículo.
  - Um automóvel é um veículo.
- Membros da classe base podem ser redefinidos na classe derivada.
- Em Java, qualquer classe herda da classe `Object`.
- Em Java, usa-se a palavra chave **`extends`**, para indicar herança.

# Exemplo de herança simples



# Construtores em classes estendidas



# Exemplo: Construtores em classes estendidas

```
class X {  
    protected int mascaraX = 0x00ff;  
    protected int mascaraTotal;  
    public X() {  
        mascaraTotal = mascaraX;  
    }  
    public int mascara(int orig) {  
        return (orig & mascaraTotal);  
    }  
}
```

```
class Y extends X {  
    protected int mascaraY = 0xff00;  
    public Y() {  
        mascaraTotal |= mascaraY;  
    }  
}
```



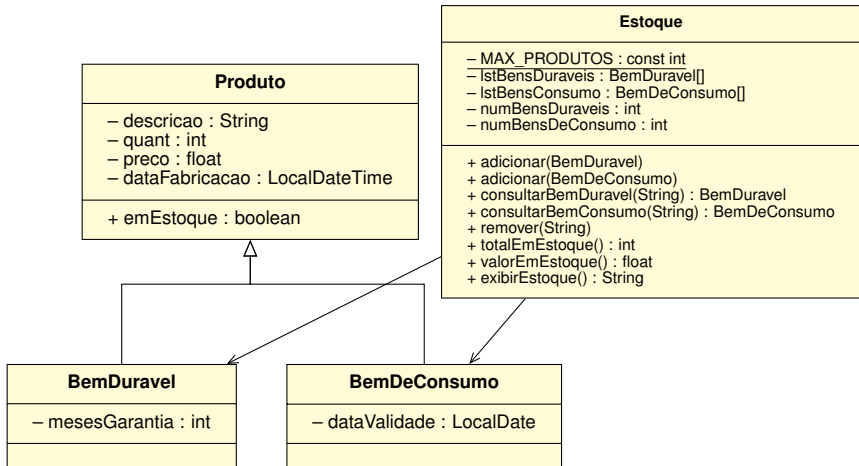
# Exemplo: Construtores em classes estendidas

- Fases de cada construtor:
  - Chamar construtor da superclasse.
  - Inicializar os campos utilizando suas instruções de inicialização.
  - Executar corpo do construtor.
- Digite as classes X e Y e acompanhe os passos de criação com comandos de impressão (use uma classe Aplicacao para isso).

# Construtores em classes estendidas

- Nova classe deve escolher qual construtor da superclasse a chamar.
- Em um construtor da subclasse pode-se chamar diretamente o construtor da superclasse: `super()`.
- A assinatura do método `super()` deve ser a mesma assinatura do construtor que se deseja chamar.
- Se não especificar construtor executa-se chamada ao construtor padrão da classe base (se houver).

# Exemplo: Herança de Produto



# Exemplo: Bem de Consumo

```
public class BemDeConsumo extends Produto {  
    private LocalDate dataValidade;  
  
    public LocalDate getDataValidade() { return dataValidade; }  
    public void setDataValidade(LocalDate dataValidade) {  
        // a data de fabricação deve ser anterior à data de validade.  
        if (getDataFabricacao().isBefore(dataValidade.atStartOfDay()))  
            this.dataValidade = dataValidade;  
    }  
    public BemDeConsumo() {  
        super();  
        // o default é uma validade de 6 meses.  
        dataValidade = LocalDate.now().plusMonths(6);  
    }  
    public BemDeConsumo(String d, float p, int q,  
                        LocalDateTime f, LocalDate v) {  
        super(d, p, q, f);  
        setDataValidade(v);  
    }  
}
```

# Exemplo: Bem Durável

```
public class BemDuravel extends Produto {  
    private int mesesGarantia;  
  
    public int getMesesGarantia() { return mesesGarantia; }  
    public void setMesesGarantia(int mesesGarantia) {  
        if (mesesGarantia > 0)  
            this.mesesGarantia = mesesGarantia;  
    }  
    public BemDuravel() {  
        super();  
        // o valor default é garantia de 6 meses.  
        mesesGarantia = 6;  
    }  
    public BemDuravel(String d, float p, int q,  
                        LocalDateTime f, int g) {  
        super(d, p, q, f);  
        setMesesGarantia(g);  
    }  
}
```