

Herança

Prof. Pedro Pongelupe



PUC Minas



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Curso de Engenharia de Software

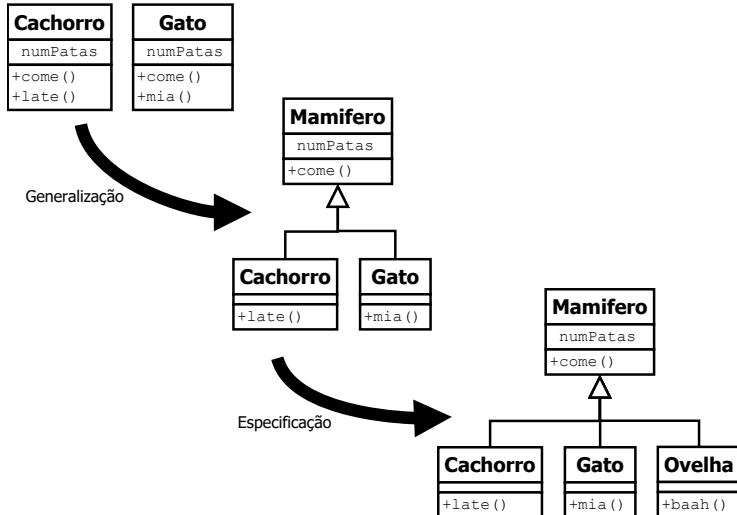
Sumário

- 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*)

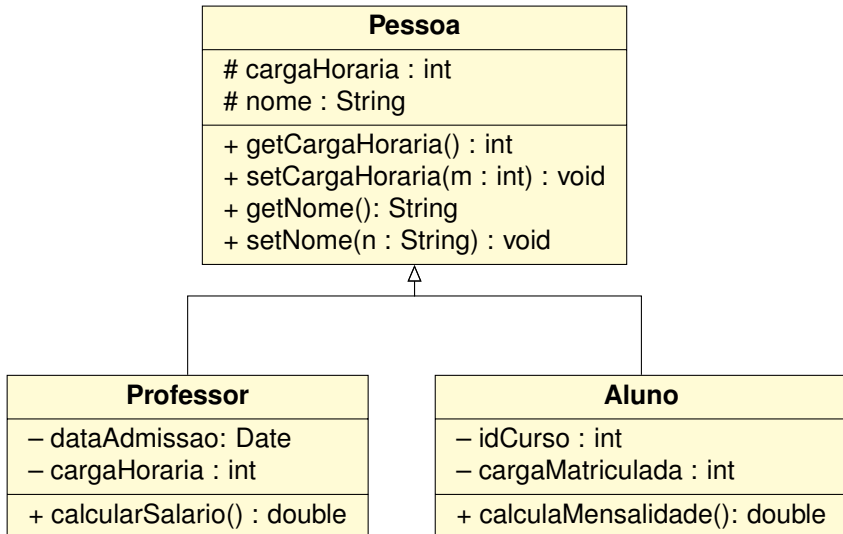
Generalização / Especialização



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



Exemplo: Construtores em classes estendidas

```
class Pessoa {  
    protected int cargaHoraria;  
    protected String nome;  
    public Pessoa(int cargaHoraria, String nome) {  
        this.cargaHoraria = cargaHoraria;  
        this.nome = nome;  
    }  
    /** getters e setters */  
}  
  
class Professor extends Pessoa {  
    protected Date dataAdmissao;  
    public Professor(int cargaHoraria, String nome, Date  
        dataAdmissao, int cargaHoraria) {  
        super(cargaHoraria, nome);  
        this.dataAdmissao = dataAdmissao;  
    }  
    public double calculaSalario() {  
        return (double) this.cargaHoraria * 1000000;  
    }  
}
```

Exemplo: Construtores em classes estendidas

```
class Aluno extends Pessoa {  
    protected int idCurso;  
    public Professor(int cargaHoraria, String nome, int idCurso,  
        int cargaMatriculada) {  
        super(cargaHoraria, nome);  
        this.idCurso = idCurso;  
    }  
    public double calculaMensalidade() {  
        return (double) this.cargahoraria * 0;  
    }  
}
```

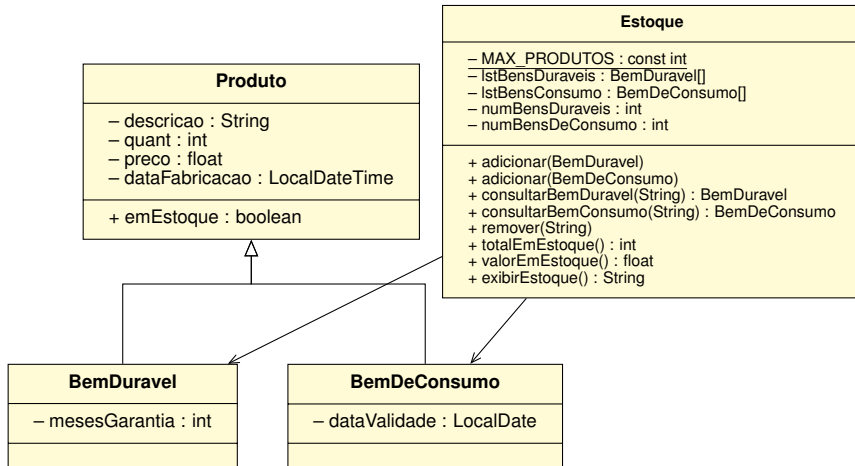

Exemplo: Construtores em classes estendidas

- Fases de cada construtor:
 - Chamar construtor da superclasse.
 - Inicializar os campos utilizando suas instruções de inicialização.
 - Executar corpo do construtor.

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);  
    }  
}
```

Obrigado!!

Muito obrigado pela atenção! Alguma dúvida? Bora praticar!!!

"Temos ódio à ditadura. Ódio e nojo." - Ulysses Guimarães