

# Herança

**Para cada exercício:**

**a) Considere a implementação de construtores com argumentos e construtores sem argumentos, toString e equals. Crie objetos e teste os métodos.**

**b) Considere a implementação utilizando herança de interface (modificador de acesso private) e herança de implementação (modificador de acesso protected)**

**c) Considere a construção do diagrama de classes da UML**

1. Crie uma classe base chamada Animal com métodos como fazerSom(). Crie subclasses como Cachorro, Gato e Vaca que herdam de Animal e implementam o método fazerSom() com sons específicos para cada animal.

2. Crie uma classe Veículo com atributos como marca, modelo e ano. Crie subclasses como Carro e Caminhao que adicionam atributos específicos, como numeroDePortas para Carro e capacidadeDeCarga para Caminhao.

3. Crie uma classe Pessoa com atributos como nome e data de nascimento. Crie subclasses como Aluno e Professor que adicionam atributos específicos, como matrícula para Aluno e titulação para Professor.

4. Crie uma classe Conta com atributos como número e saldo e métodos como depositar() e sacar(). Crie subclasses ContaEspecial (tem atributo limite) e ContaPoupanca (tem atributo data de aniversário) que implementam suas próprias regras para sacar().

5. Crie uma classe abstrata Forma com um método abstrato calcularArea(). Crie subclasses como Circulo, Retangulo e Triangulo que implementam o método calcularArea().

6. Crie uma classe Funcionario com atributos como nome, salario e métodos como calcularSalario(). Crie subclasses `Gerente`, `Desenvolvedor` e `Analista` que adicionam atributos específicos e redefinem o método `calcularSalario()` para incluir bônus ou comissões.

7. Crie uma classe Livro com atributos como titulo e autor. Crie subclasses LivroFisico e Ebook que adicionam atributos específicos como numeroDePaginas e tamanhoArquivo.