

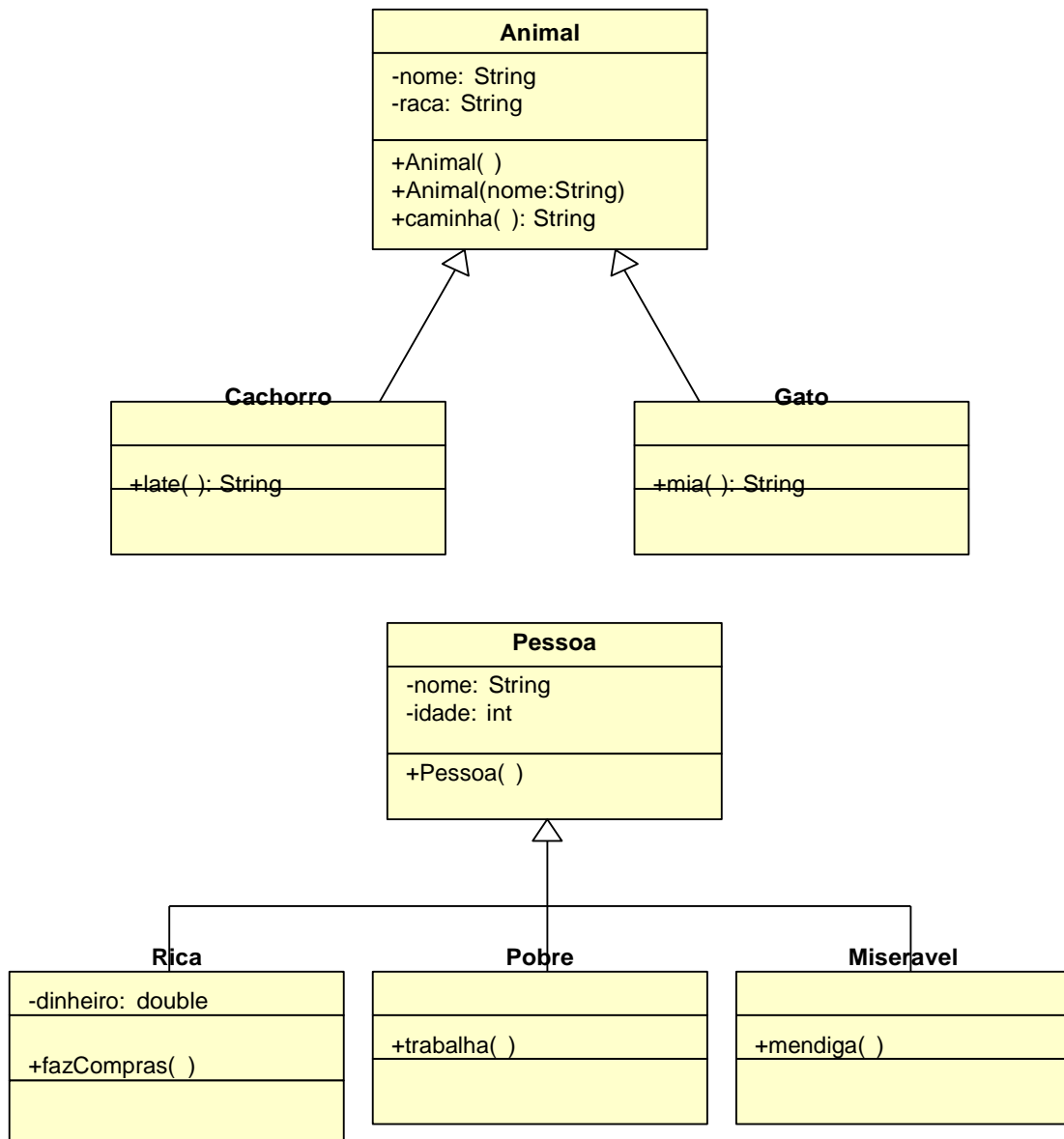
## Exercícios - Herança e Polimorfismo

---

**Exercício 1:** Implemente a classe Funcionario e a classe Gerente.

- crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula (faça o método GET). Sobrescreva o método `exibeDados()`.
- sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes `Tecnico` e `Administrativo`.

**Exercício 2:** Implemente os diagramas de classe abaixo:



**Exercício 3:** Crie uma classe chamada Ingresso que possui um valor em meticais e um método `imprimeValor()`.

- crie uma classe VIP, que herda Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído).
- crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal".
- crie uma classe CamaroteInferior (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe CamaroteSuperior, que é mais cara (possui valor adicional). Esta última possui um método para retornar o valor do ingresso. Ambas as classes herdam a classe VIP.

**Exercício 4:** Crie a classe Imovel, que possui um endereço e um preço.

- crie uma classe Novo, que herda Imovel e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.
- crie uma classe Velho, que herda Imovel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

**Exercício 5:** Crie uma classe de Teste com o método *main*. Neste método:

- crie um assistente administrativo e um técnico. Imprima o número de matrícula e o nome de cada um deles.
- crie um animal do tipo cachorro e faça-o latir. Crie um gato e faça-o miar. Faça os dois animais caminharem.
- teste (como quiser) as classes Rica, Pobre e Miseravel.
- crie um ingresso. Peça para o usuário digitar 1 para normal e 2 para VIP. Conforme a escolha do usuário, diga se o ingresso é do tipo normal ou VIP. Se for VIP, peça para ele digitar 1 para camarote superior e 2 para camarote inferior. Conforme a escolha do usuário, diga se que o VIP é camarote superior ou inferior. Imprima o valor do ingresso.
- crie um imóvel. Peça para o usuário digitar 1 para novo e 2 para velho. Conforme a definição do usuário, imprima o valor final do imóvel.

**Exercício 6:** Criar uma estrutura hierárquica que contenha as seguintes classes: Veiculo (classe abstracta), Bicicleta e Automóvel.

Os métodos da classe Veiculo são todos abstractos e possuem a seguinte assinatura:

- `listarVerificacoes()`
- `ajustar()`
- `limpar()`

Estes métodos são implementados nas subclasses Automóvel e Bicicleta.

Acrescentar na classe Automóvel o método `mudarOleo()`

**Exercício 7:** Para desenvolver a classe Teste que é apresentada a seguir é necessário criar também

a classe **Oficina** que terá dois métodos:

- `proximo()` que retorna aleatoriamente um objecto do tipo bicicleta ou automóvel
- `manutencao(Veiculo v)` que recebe como parâmetro um objecto do tipo veiculo e chama os métodos definidos na classe veiculo:
  - `listarVerificacoes()`
  - `ajustar()`
  - `limpar()`
  - se o veiculo for Automóvel deve também chamar o método `mudarOleo()`