
Trabalho Prático 05 (TP05)

Instruções:

- i - O arquivo deve ser entregue em formato .ZIP ou .RAR seguindo a nomenclatura: “XXXX.KKK” onde XXXX é o número de sua matrícula e KKK a extensão do arquivo.
 - ii - Cada um dos exercícios deve criado em um diretório com o seguinte nome: Exercicio_XX onde XX é o número da questão solucionada.
 - iii - Para cada programa desenvolvido deverão ser entregues **SOMENTE** os arquivos de projeto e classes Java em seus respectivos pacotes.
 - iv - O arquivo deve ser enviado via moodle limitado a data e hora de entrega definida no Plano de Ensino. Não serão aceitos trabalhos enviados por e-mail.
-

Questão 1. Imagine uma lâmpada e implemente o que se pede:

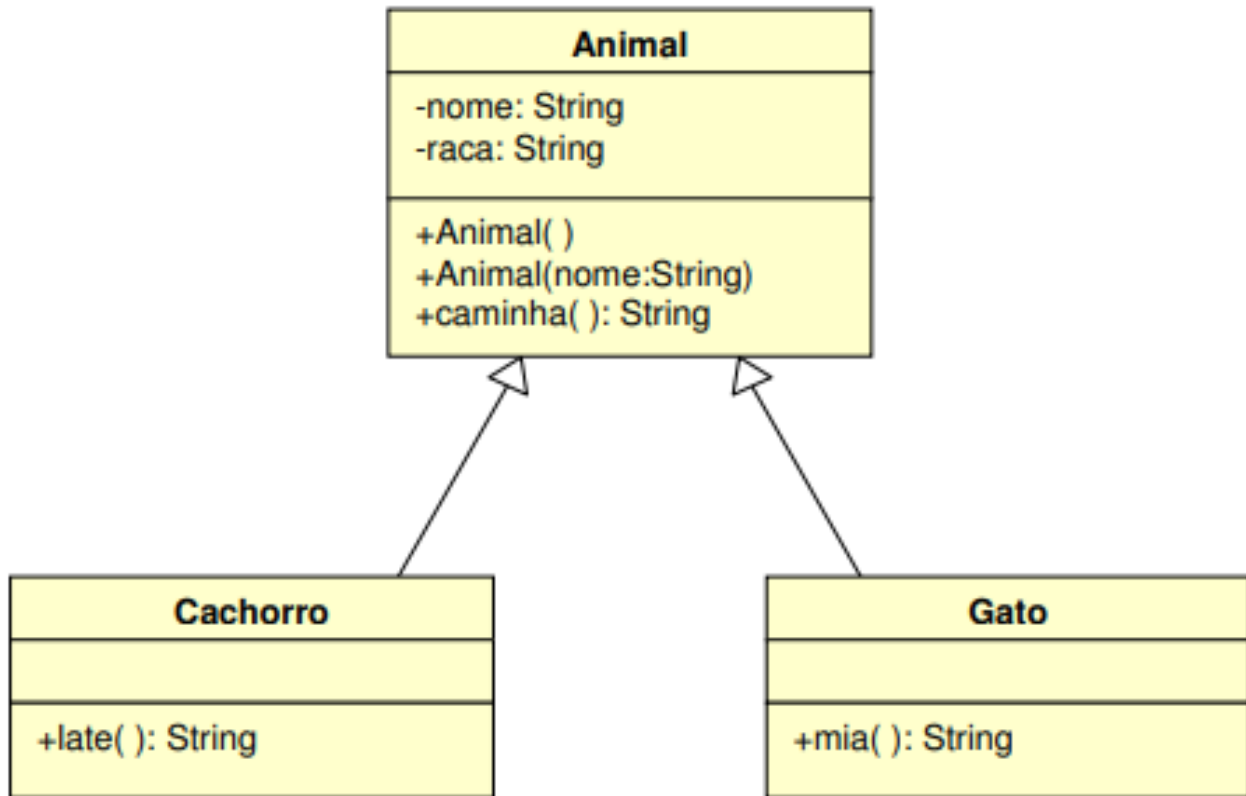
- (a) a lâmpada poderá ter três estados: apagada, acesa e meia-luz. Torne possível a representação de uma lâmpada onde a luminosidade pode ser ajustada com qualquer valor entre 0% (apagada) e 100% (acesa). Dica: em vez de operações para possibilitar o ajuste para cada um dos estados, descreva uma operação que receba um valor de ajuste.
- (b) Inclua, no modelo Lampada, uma operação estáLigada que retorne verdadeiro se a lâmpada estiver ligada e falso, caso contrário.
- (c) A classe deve representar também o número de watts da lâmpada. Escreva um método éEconômica que retorne o valor booleano true se a lâmpada consumir menos de 40 watts e false caso contrário.
- (d) Inclua em sua classe Lampada uma funcionalidade que indique quantas vezes a lâmpada foi acesa.
- (e) Escreva dois construtores para a classe: um que recebe como argumentos o número de watts da lâmpada, e outro, sem argumentos, que considera que a lâmpada tem 60 watts por default.
- (f) Crie uma estratégia que permita contar o número de instâncias construídas (static).

Escreva um programa que utilize instâncias da classe Lampada e faça os testes dos métodos propostos.

Questão 2. Escreva a classe LampadaFluorescente como sendo herdeira da classe Lampada. A classe LampadaFluorescente deve ter um campo que represente o comprimento da lâmpada em centímetros. Crie nessa classe um construtor para inicializar os seus atributos.

Questão 3. Crie uma classe Empresa capaz de armazenar os dados de uma empresa (Nome, Endereço, Cidade, Estado, CEP e Telefone). Inclua um construtor sem argumentos e um com argumentos para inicialização dos atributos. Crie métodos que funcionem como getter, setter e print. Utilize a classe Empresa como base para criar a classe Restaurante. Esta classe filha deve conter atributos que representam o tipo de comida e o preço médio de um prato. Crie um construtor para esta classe que chame explicitamente o construtor da classe Empresa, um getter e um setter, além de um método print, que utiliza o método print da classe mãe. Crie uma Main para testar sua aplicação.

Questão 4. Implemente as classes indicadas pelo diagrama abaixo:



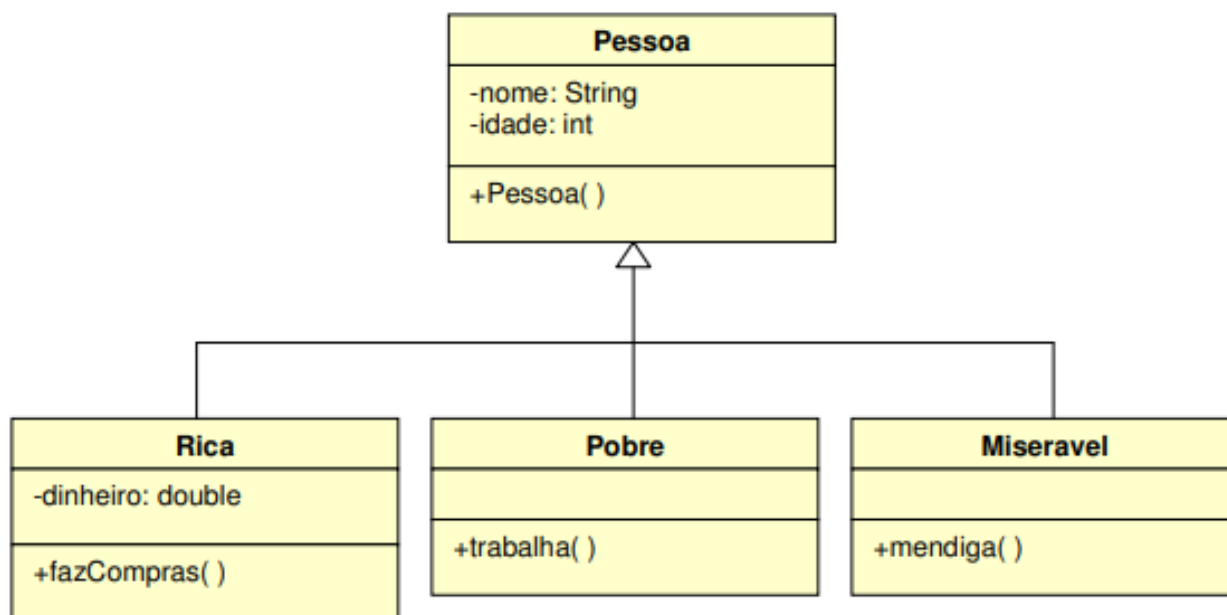
Crie um animal do tipo cachorro e faça-o latir. Crie um gato e faça-o miar. Faça os dois animais caminharem.

Questão 5. Crie uma classe chamada Ingresso que possui um valor em reais e um método `imprimeValor()`.

- i - 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).
- ii - crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal".
- iii - 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.

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.

Questão 6. Implemente as classes indicadas pelo diagrama abaixo:



Crie uma classe para testar as implementações propostas no exercício.