

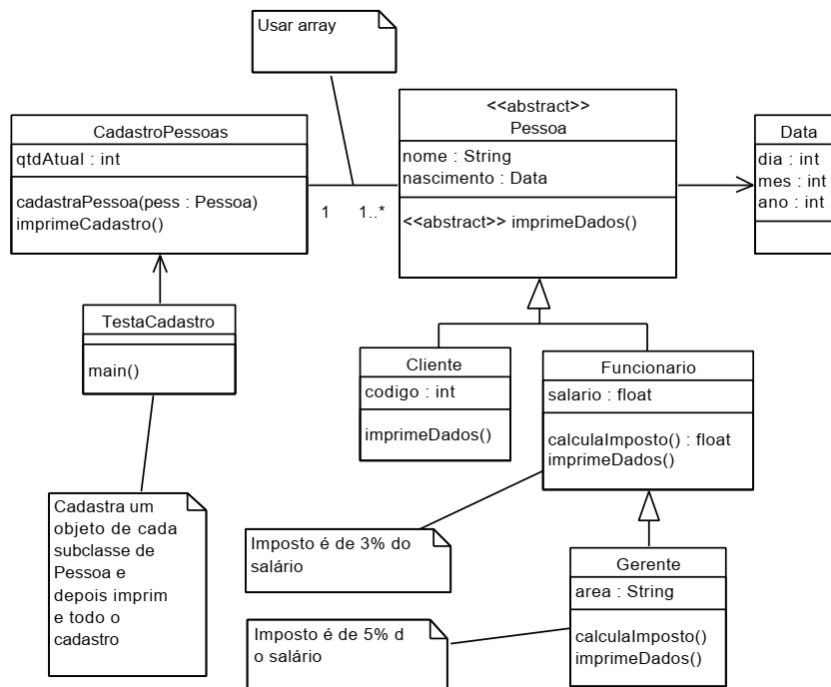
5COP011 - Laboratório de Programação

Prof. Bruno B. Zarpelão

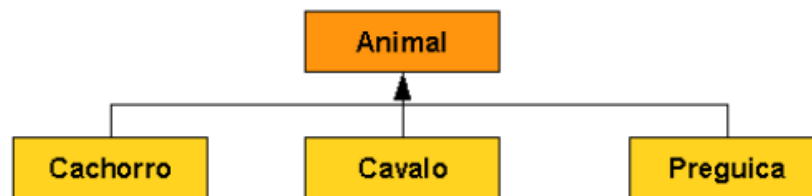
Exercícios de revisão: herança, classes abstratas, polimorfismo, e threads

08/08/2019

1. Construa um programa em Java de acordo com as classes apresentadas no diagrama abaixo:



2. Crie uma hierarquia de classes conforme o diagrama abaixo com os seguintes atributos e comportamentos (observe a tabela). Utilize os seus conhecimentos e distribua as características de forma que tudo o que for comum a todos os animais fique na classe Animal:



Após executar esse primeiro passo, desenvolva o restante do programa conforme os itens a seguir:

Cachorro	Cavalo	Preguiça
Possui nome	Possui nome	Possui nome
Possui idade	Possui idade	Possui idade
Deve emitir som	Deve emitir som	Deve emitir som
Deve correr	Deve correr	Deve subir em árvores

- (a) Implemente um programa que crie os 3 tipos de animais definidos no exercício anterior e invoque o método que emite o som de cada um de forma polimórfica, isto é, independente do tipo de animal.
- (b) Implemente uma classe `Veterinario` que contenha um método `examinar()` cujo parâmetro de entrada é um `Animal`. Quando o animal for examinado ele deve emitir um som (o método deve estar preparado para lidar com os três tipos de animal).
- (c) Crie uma classe `Zoologico`, com 10 jaulas (utilize um array) e coloque em cada jaula um animal diferente. Percorra cada jaula e emita o som e, se o tipo de animal possuir este comportamento, faça-o correr.
3. Nesse exercício, vamos criar um programa para uma loja de livros, CDs e DVDs:
- (a) Crie uma hierarquia de classes de domínio para uma loja que venda livros, CDs e DVDs. Sobrescreva o método `toString()` para que imprima: (i) Para livros: nome, preço e autor; (ii) Para CDs: nome, preço e número de faixas; (iii) Para DVDs: nome, preço e duração. Evite ao máximo repetição de código utilizando `super` no construtor e no método sobrescrito. Em seguida, crie uma classe `Loja` com o método `main()` que adicione 5 produtos diferentes (a sua escolha) a um vetor e, por fim, imprima o conteúdo do vetor.
- (b) Modifique o código do programa anterior, da seguinte forma:
- Adicione um atributo que represente o código de barras do produto (é um valor obrigatório e, portanto, deve ser pedido no construtor);
  - Sobrescreva o método `equals()` retornando `true` se dois produtos possuem o mesmo código de barras;
  - Na classe `Loja`, implemente um simples procedimento de busca que, dado um produto e um vetor de produtos, indique em que posição do vetor se encontra o produto especificado ou imprima que o mesmo não foi encontrado;
  - No método `Loja.main()`, após a impressão do vetor (construído no primeiro item da questão), escolha um dos 5 produtos e crie duas novas instâncias idênticas a ele: uma com o mesmo código de barras e outra com o código diferente. Efetue a busca deste produto no vetor utilizando as duas instân-

cias e verifique o resultado.

4. Nesse exercício, vamos trabalhar com a distribuição de vagas para um curso. Suponha que o curso tem 15 vagas e 20 interessados. O interessado pode executar duas ações: requerer a inscrição no curso e desistir do curso. Quando ele requer a inscrição no curso, há uma análise de disponibilidade de vagas e, caso todas estejam ocupadas, a inscrição é negada. Nesse caso, o interessado aguarda um período e tenta nova inscrição. Caso haja vagas disponíveis, o interessado é inscrito e a vaga é ocupada. Quando o interessado desiste do curso, a vaga que ele ocupava é imediatamente aberta. Construa um programa em Java que simule o período de inscrições para esse curso. Cada interessado deve ser uma thread. A cada mudança no quadro de vagas, o programa deve imprimir quantas vagas estão disponíveis e quantos candidatos estão inscritos. Não pode haver inconsistências nesses valores.