

Lista 2 - Programação Orientada a Objetos com Java

Prof. Ms. Gregory Acacio Seibert Oliveira

2º Semestre de 2021

- 1) Crie uma classe Agenda que pode armazenar 10 pessoas e que seja capaz de realizar as seguintes operações:
 - void armazenaPessoa(String nome, int idade, float altura);
 - void removePessoa(String nome);
 - int buscaPessoa(String nome); // informa em que posição da agenda está a pessoa.
 - void imprimeAgenda(); // imprime os dados de todas as pessoas da agenda
 - void imprimePessoa(int index); // imprime os dados da pessoa que está na posição “i” da agenda.
- 2) Crie uma classe Televisao e uma classe ControleRemoto que pode controlar o volume e trocar os canais da televisão. O controle de volume permite: aumentar ou diminuir a potência do volume de som em uma unidade de cada vez; aumentar e diminuir o número do canal em uma unidade para trocar para um canal indicado; consultar o valor do volume de som e o canal selecionado.
- 3) Identifique as classes e implemente um programa para a seguinte especificação:

“O supermercado vende diferentes tipos de produtos. Cada produto tem um preço e uma quantidade em estoque. Um pedido de um cliente é composto de itens, onde cada item especifica o produto que o cliente deseja e a respectiva quantidade. Esse pedido pode ser pago em dinheiro, cheque ou cartão.”
- 4) A classe RoboSimples representa um robô que pode se movimentar apenas nas quatro direções cardeais, mas não impõe restrições no valor do ângulo que pode ser passado para o método mudaDireção (herdado da classe RoboAbstrato). Escreva, para a classe RoboSimples, o método mudaDireção, sobrepondo o método herdado, de forma que a direção atual seja calculada como: • Se valores abaixo de 45 graus ou acima de 315 graus forem passados como argumento para o método,

a direção do robô será 0 (zero) grau; • Se valores entre 45 e 135 graus forem passados como argumento para o método, a direção do robô será 90 graus; • Se valores entre 135 e 225 graus forem passados como argumento para o método, a direção do robô será 180 graus; • Se valores entre 225 e 315 graus forem passados como argumento para o método, a direção do robô será 270 graus.

- 5) Escreva um programa para armazenar dados de veículos. Primeiramente, crie a classe Motor que contém NumCilindro (int) e Potenci(int). Inclua um construtor sem argumentos que inicialize os dados com zeros e um que inicialize os dados com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de dados, Get(), e uma que imprima os dados, Print().

Escreva a classe Veiculo contendo Peso em quilos (int), VelocMax em Km/h (int) e Preco em R\$ (oat). Inclua um construtor sem argumentos que inicialize os dados com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de dados, Get(), e uma que imprima os dados, Print(). Crie uma classe CarroPasseio derivada das classes Motor e Veículo como base. Inclua Cor (string) e Modelo (string). Inclua um construtor sem argumentos que inicialize os dados com zeros e uma que inicialize os dados com os valores recebidos como argumentos. Acrescente duas funções, uma para a entrada de dados, Get(), e uma que imprima os dados, Print().

Crie uma classe Caminhao derivada das classes Motor e Veiculo. Inclua Toneladas (carga máxima), AlturaMax (int) e Comprimento (int). Inclua um construtor sem argumentos que inicialize os dados com zeros e um que inicialize com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de dados, Get(), e uma que imprima os dados, Print().

- 6) Implemente uma classe abstracta de nome Forma onde são declarados dois métodos abstractos:

- float calcularArea();

- float calcularPerimetro();

Crie, como subclasse de Forma, uma classe de nome Rectângulo cujas instâncias são caracterizadas pelos atributos lado e altura ambos do tipo oat. Implemente na classe Retangulo os métodos herdados de Forma e outros que ache necessários. Crie, como subclasse de Forma, uma classe de nome Círculo cujas instâncias são caracterizadas pelo atributo raio do tipo oat. Implemente na classe Circulo os métodos herdados de Forma e outros que ache necessários. Nota: poderá aceder ao valor de Pi fazendo Math.Pi.

Crie, como subclasse de Rectângulo, uma classe de nome Quadrado cujas instâncias são caracterizadas por terem os atributos lado e altura com o mesmo valor. Elabore um programa de teste onde é declarado um array, de dimensão 5, do tipo estático Forma. Nesse array devem ser guardadas instâncias de Retângulo, Círculo e Quadrado seguindo uma ordem aleatória. Nota: para gerar números aleatórios crie primeiro uma instância da classe Random (presente na biblioteca java.util) e para extrair um inteiro entre 0 e n efetue a evocação nextInt(n). Depois implemente um ciclo que percorre o array evocando, relativamente a cada um dos objectos guardados, os métodos calcularArea e calcularPerimetro

- 7) Crie uma classe chamada Empresa capaz de armazenar os dados de uma empresa (Nome, Endereço, Cidade, Estado, CEP e Fone). Inclua um construtor sem argumentos e um que receba os dados como argumentos e os inicialize. Escreva duas funções, uma para fazer a interface com o usuário da entrada de dados, Get(), e outra para imprimir os dados, Print(). Use a classe Empresa como base para criar a classe Restaurante. Inclua o tipo de comida, o preço médio de um prato, duas funções construtoras, a interface de entrada de dados, Get(), e a função que imprima os dados, Print(). Construa um programa para testar a classe Restaurante.
- 8) Crie uma classe em Java chamada fatura para uma loja de suprimentos de informática. A classe deve conter quatro variáveis: 3 • o número (String), • a

descrição (String), • a quantidade comprada de um item (int) • e o preço por item (double). A classe deve ter um construtor e um método get e set para cada variável de instância. Além disso, forneça um método chamado getTotalFatura que calcula o valor da fatura e depois retorna o valor como um double. Se o valor não for positivo, ele deve ser configurado como 0. Se o preço por item não for positivo, ele deve ser configurado como 0.0. Escreva um aplicativo de teste chamado FaturaTeste (em outro arquivo) que demonstra as capacidades da classe Fatura.