

::: Lista de Exercícios #2 :::

- 1) Quais são os princípios da Orientação a Objetos? Comente brevemente sobre cada um deles.
- 2) Qual é o significado de cada um dos termos a seguir: *classe*, *objeto*, *instância*, *atributo* e *método*?
- 3) Quais as vantagens e desvantagens em se utilizar a linguagem Java para implementar algoritmos?
- 4) Qual a diferença entre o j2sdk e o j2re?
- 5) Quais são os passos necessários para se instalar o ambiente Java em uma máquina de desenvolvimento?
- 6) Quais são as linhas de comando para compilar e para executar uma aplicação Java?
- 7) Quais são as regras para nomear:
 - a) Classes
 - b) Atributos e métodos
- 8) Faça uma aplicação em Java que escreva seu nome na console.
- 9) Qual a diferença entre os modificadores *private* e *public* quando aplicados a membros de uma classe?
- 10) Faça uma classe que represente uma conta bancária básica, devendo ela definir como atributos: um campo para o nome, outro para o número da conta e, por fim, um para armazenamento do saldo. É claro que será necessário especificar os métodos adequados para a manipulação da conta, ou seja, os métodos que compõe a interface do objeto com os elementos externos a ele.
- 11) Construa uma classe para representar um elemento do acervo de uma biblioteca, devendo obrigatoriamente conter atributos para armazenar o nome do livro, o seu código, a data do empréstimo e data prevista para a devolução.
- 12) Crie um método para cadastrar as informações de um livro e outro para exibir suas propriedades.
- 13) Crie um método chamado *emprestar* que atribua a data informada como parâmetro para o atributo *dataDoEmprestimo* e também, calcule e atribua para a *dataPrevistaDeDevolução*, assumindo 7 dias corridos.
- 14) Defina uma classe chamada *Ponto*, para representar as coordenadas de um ponto no plano cartesiano (*Faça a definição completa da classe, ou seja, os atributos e métodos pertinentes*).
- 15) Defina uma classe chamada *Triangulo* que apresente como atributo um vetor de três objetos do tipo *Ponto*, bem como, seus métodos construtores, de acesso e modificadores.

- 16) Defina uma classe chamada *Grau* e outra chamada *Radiano*, ambas para o armazenamento de valores do arco trigonométrico.
- Cada classe deverá conter os métodos *set* (*métodos modificadores*) e métodos *get* (*métodos de acesso*) que sirvam de comunicação com o objeto
 - Um mecanismo que permita a conversão automática entre objetos destas classes (*casting*)
 - Um programa que instancie um objeto da classe *Graus*, leia um valor do teclado e armazene nele, instancie um objeto da classe *Radianos*, atribua um para o outro e escreva os dois valores na tela.

- 17) Desenvolva uma classe *Fração* que permita realizar os seguintes cálculos:

- Simplificação de fração:
 $a/b = ((a/mdc(a,b)) / (b/mdc(a,b)))$, onde o *mdc* retorna o máximo divisor comum de *a* e *b*;
- Soma de fração:
 $(a/b) + (c/d) = ((a*d + c*b) / (b*d))$
- Multiplicação de fração:
 $(a/b) * (c/d) = ((a*c) / (b*d))$
- Igualdade:
 $(a/b) == (c/d)$ se $(a*d) == (b*c)$
- Não igualdade:
 $(a/b) != (c/d)$ se $(a*d) != (b*c)$
- Outros métodos que julgar necessário para o bom funcionamento e melhor aproveitamento da classe por outros usuários.

- 18) Usando seus conhecimentos em programação orientada a objetos e na linguagem Java, modele e implemente sua solução para o problema abaixo utilizando os conceitos de classes, sobrecarga de funções (polimorfismo) e herança.

“Uma multinacional chamada Geral Motors Co. (GMC), instalada na cidade de São Paulo, monta veículos de passeio e para o transporte de cargas. Sua linha de montagem é imensa e conta com a produção de uma grande variedade de modelos, sendo atualmente, três modelos diferentes de veículos de passeio e quatro modelos de veículos para o transporte de cargas.

A maior parte dos componentes é produzido pela própria GMC e o restante é terceirizado. Recentemente a GMC fechou um contrato para o fornecimento dos pneus de seus veículos com a Goodweek e das rodas esportivas com a Nino. Existe ainda um único componente importado, no caso dos motores, que é fabricado em sua matriz na Alemanha e enviado para a montadora em São Paulo.

Para controlar todo este processo foi criada uma tabela para cada tipo de componente¹, onde são anotadas todas as suas características e enviadas para o almoxarifado geral. Todos os veículos produzidos são encaminhados a uma filial responsável pela distribuição para todo o Brasil.

Cada um dos veículos, antes de ser enviado para a distribuição, deve ter suas características anotadas em uma guia de transporte ², de forma que possam ser unicamente identificados.”

- (1) Tabela de Motores: número de cilindros e tipo de combustível

Tabela de Rodas: tamanho do aro e modelo

Tabela de Pneus: tamanho do aro e largura

- (2) Guia de transporte de veículo de passeio: chassi, cor, dados do motor, número de portas, dados das rodas, dados dos pneus.
- (3) Guia de transporte de veículo de transporte: chassi, cor, dados do motor, capacidade de carga em kilos, dados das rodas, dados dos pneus.

- 19) Pensando em um sistema de controle para uma lanchonete, faça uma classe Lanche que apresente um atributo para armazenar o tipo de carne (hambúrguer, frango, lombo ou filé) e mais oito atributos para dizer se há ou não outros itens como queijo, ovo, presunto, salsicha, bacon, alface, tomate e catupiry. Esta classe deverá também apresentar um atributo para armazenar o valor do lanche, que tem um preço básico de acordo com o tipo de carne, e depois tem seu montante acrescido de acordo com os itens adicionais (vide tabela abaixo). Deseja-se que a classe tenha um mecanismo automático para a contagem do número de lanches feitos, bem como, os métodos para acrescentar cada um dos itens opcionais, um construtor cujo parâmetro é o tipo de carne, um construtor sem parâmetros que produz hambúrguer básico com queijo (para isso invoque o construtor com parâmetros informando que o lanche é um hambúrguer).

Ítem	Valor
Hambúrguer	R\$ 2,00
Frango	R\$ 2,10
Lombo	R\$ 2,50
Filé	R\$ 2,70
Queijo	R\$ 0,10
Ovo	R\$ 0,20
Presunto	R\$ 0,15
Bacon	R\$ 0,40
Salsicha	R\$ 0,30
Alface	R\$ 0,10
Tomate	R\$ 0,10
Catupiry	R\$ 0,70

- 20) Crie uma classe Contador que possui um atributo inteiro e três métodos (incremento, decremento e retorna o valor do atributo). Faça o método *main* que chame mais de uma vez o contador, mostrando o valor do contador na tela.
- 21) Crie uma classe chamada ContadorEstatico similar a classe anterior, porém que contenha um atributo estático inteiro e três métodos (incremento, decremento e retorna o valor do atributo). Utilize o mesmo método *main* da classe do exercício anterior para chamar os métodos desta classe.
- 22) Quais foram os resultados obtidos nos dois exercícios anteriores? Compare-os descrevendo qual a diferença da utilização ou não do modificador *static*