Centro Federal de Educação Tecnológica –

**Disciplina: Programação Orientada a Objetos**

Prof. Gustavo Guedes

****

**Aluno(a): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Turma:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Esta avaliação deve ser respondida preferencialmente usando caneta esferográfica azul. Não utilize cor vermelha ou alguma tonalidade próxima.**

**Faça distinção clara entre maiúsculas e minúsculas.**

**Seja claro, formal e sucinto.**

**Códigos desnecessários e que reduzam o desempenho do sistema serão penalizados.**

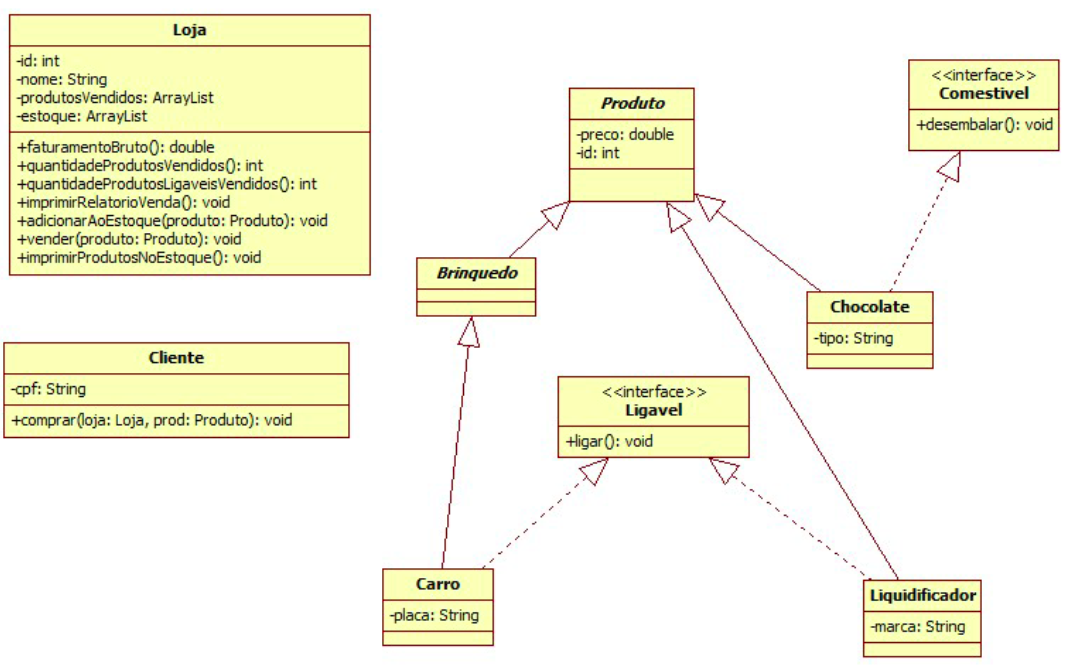
**Utilize as boas práticas de programação.**

**Atributos de intância devem sempre ser declarados como privados.**

**LEIA AS QUESTÕES ATÉ O FINAL ANTES DE COMEÇAR.**

**Questão 1 (1) –** Crie uma classe chamada Utils. Em seguida, crie um método que receba um array de boolean e retorne um novo array com zeros e uns. Caso a posição do array de boolean seja falsa, preencha a mesma posição do novo array com zero. Caso seja true, preencha com 1.

**Questão 2 (5,0) -**

****

1 - Crie todas as classes e interfaces acima.

2- Crie os getters e setters que julgar necessário para o funcionamento do sistema.

3- Observe que Produto e Brinquedo são classes abstratas.

4- Implemente todos os métodos das classes do diagrama acima, de forma que

funcionem de acordo com o nome que apresentam.

5- Em produto, adicione um contrutor que receba o preco e o id como argumentos. Não adicione outro construtor.

6- Respeite as assinaturas dos métodos.

7- Os métodos das entidades com retorno void devem seguir o padrão de aula,

imprimindo no console “XXX fazendo alguma coisa”.

8- O método comprar da classe cliente deve chamar o método vender na loja específica

em que quer comprar determinado produto.

9- O método faturamentoBruto deve retornar a soma dos preços dos produtos vendidos.

10- O método quantidadeProdutosVendidos deve retornar o número total de produtos

vendidos.

11- O método quantidadeProdutosLigaveisVendidos deve retornar o número total de

produtos Ligaveis vendidos.

*12- O método imprimirRelatorioVenda deve imprimir o “id” e “preco” dos produtos*

*vendidos junto com (se for carro, imprimir a placa – se for Liquidificador, imprimir a*

*marca, se for chocolate, imprimir o tipo).*

*13- O método imprimirProdutosNoEstoque deve imprimir o “id” e “preco” dos produtos*

*existentes no estoque junto com (se for carro, imprimir a placa – se for Liquidificador,*

*imprimir a marca, se for chocolate, imprimir o tipo).*

*\*Cuide para que todas as classes acima compilem sem problemas.*

14- O método vender da classe loja deve remover o produto do estoque e adicionar a lista

aos produtos vendidos. Considere que ArrayList possui o método remove(int i) que remove o elemento do índice i e retorna o elemento removido.

16- Crie uma classe com o método main.

\*Sempre atribua valores aos atributos de instância de todos os objetos que criar.

a. Crie uma loja nova.

b. Crie um carro, um liquidificador e um chocolate. Adicione ao estoque da loja.

c. Crie um objeto do tipo cliente.

d. Faça esse cliente comprar o carro e o chocolate da loja criada.

e. Imprima o faturamento bruto da loja.

f. Imprima a quantidade de produtos vendidos.

g. Imprima a quantidade de produtos ligaveis vendidos.

h. Imprima o relatório de venda.

i. Imprima os produtos existentes no estoque.

**Questão 3 (1,0) -** Crie uma classe chamada Carro que possua dois atributos privados: id (do tipo int) e motor (do tipo Motor). Crie os getters e setters.

- Crie a classe Motor. Crie um atributo privado chamado cor (escolha o tipo apropriado). Crie os getters e setters.

- Crie um programa (classe chamada Programa1) que deve instanciar um objeto do tipo Carro, um objeto do tipo Motor e atribuir o Motor ao Carro. Receba o id do Carro e a cor do Motor do console (entrada do teclado). Em seguida, insira nessa classe o método abaixo. Dentro desse método, escreva o código para imprimir os dados do Motor associado ao Carro referenciado por “n”.

public static void imprimeDadosCarro (Carro n) {

//escreva o código para imprimir os dados do motor associado ao Carro referenciado por “n”.

}

**Questão 4 (1,0) –** Escreva o nome de todos os tipos primitivos do Java.

**Questão 5 (2) – Observe as classes abaixo. O que sai no console?**

**public** **class** Arvore {

**private** String nome;

**public** **int** a;

**public** **static** **int** *b*;

**public** String getNome() {

**return** nome;

}

**public** **void** setNome(String nome) {

**this**.nome = nome;

}

}

**public** **class** TesteArvore {

**public** **static** **void** main(String[] args) {

Arvore v1 = **new** Arvore();

System.***out***.println(v1.getNome());

v1.setNome("Acacia");

Arvore v2 = **new** Arvore();

v2.setNome("Ipe");

*teste1*(v1, v2);

System.***out***.println(v1.getNome());

System.***out***.println(v2.getNome());

v1.a = 10;

**int** x = v1.a;

v1.a=5;

v2.a = v1.a;

System.***out***.println(x);

*teste2*(v1.a, x, v1);

System.***out***.println(v1.getNome());

System.***out***.println(x);

System.***out***.println(v1.a);

System.***out***.println(v1.*b*);

System.***out***.println(v2.a);

System.***out***.println(v2.*b*);

*teste3*(v2);

System.***out***.println(v2.getNome());

System.***out***.println(v2.a);

}

**public** **static** **void** teste1(Arvore v2, Arvore v1) {

v1.setNome("Abacateiro");

v1 = v2;

v1.setNome("Eucalipto");

}

**public** **static** **void** teste2(**int** a, **int** x, Arvore arv) {

a = 20;

x=15;

arv.*b* = 99;

arv.setNome("Macieira");

arv = **new** Arvore();

}

**public** **static** Arvore teste3(Arvore v2) {

v2 = **new** Arvore();

v2.setNome("Mogno");

**return** v2;

}

}