Centro Federal de Educação Tecnológica –

**Disciplina: Programação e Desenvolvimento de Sist.**

Prof. Gustavo Guedes

****

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

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

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

**Utilize as boas práticas de programação, sempre que possível.**

**Vale lembrar que, quando possível, declarar atributos de instância como privados são uma boa prática de programação.**

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

**Questão 1 – (6) – Não use generics nessa questão. Se não sabe o que é, não se preocupe.**

A - Crie uma classe abstrata chamada Veículo. Crie um atributo chamado modelo (String). Crie apenas UM construtor nessa classe, que recebe o modelo do Veículo. Crie os getters e setters para modelo. Se a referencia de um objeto Veiculo for impressa no console, deverá ser impresso o modelo do veículo. Syso (x), em que x é do tipo Veiculo -> saída XYZ1243.

B - Crie a interface Abastecivel. Abastecivel possui um método chamado abastecer (double litros), que retorna um boolean que indica se o tanque desse Abastecivel recebeu aquela quantidade ou esta cheio.

C - Crie a classe Bicicleta como subclasse de Veiculo. Crie a classe Carro e Moto como subclasse de Veiculo e, além disso, as duas devem implementar Abastecivel. Crie nas classes Carro e Moto uma constante privada que represente o tamanho total do tanque, denominada tamanhoTanque (double). Essa constante deve ser inicializada sempre pelos respectivos construtores. Crie apenas o getter para tamanhoTanque. Crie o atributo privado quantidadeAtual, que representa quantos litros há no tanque de gasolina. Crie os getters e setters para quantidadeAtual. Implemente a lógica do abastecer tanto de Moto como de Carro. Considere que o abastecer de carro permite encher o tanque até o tamanho total do tanque (tamanhoTanque) e o de moto, completa até 99% da capacidade (Repare que é o método abastecer das duas classes que permite ou não que o combustível seja inserido). Considere que as bombas sempre abastecem de 0.1 em 0.1 litros. Considere que o método abastecer sempre vai receber 0.1.

D - Crie a classe BombaCombustivel. Crie os atributos quantidadeLitros (que representa a quantidade de litros de gasolina que existe na bomba) e o preço por litro. Crie o método completarTanque que receba um abastecivel como argumento. O método deve controlar o abastecimento dos Abasteciveis, ou seja, deve colocar 0.1 litros de gasolina até que o tanque seja completado. Ao final, deve imprimir no console o total de litros para encher o tanque e o preço que o dono do Abastecivel deve pagar. Observe que, se acabar a gasolina da bomba, o sistema deve parar de funcionar, não pode continuar. Mesmo assim, o preço a ser pago pelo dono deve ser impresso no console.

E - Crie, ainda em BombaCombustivel, um novo método chamado abastecerFila (ArrayList x), que deve abastecer (completar o tanque) os Abasteciveis da fila (do ArrayList) até que a fila acabe ou a gasolina acabe. Percorra a lista do inicio ao fim (não remova os carros da fila). Antes de abastecer o Abastecivel efetivamente, imprima no console a referencia ao Abastecivel.

F - Crie a classe principal. Crie uma BombaCombustivel. Coloque 5000 litros de gasolina nessa bomba. Crie um objeto do tipo Carro. Mande abastecer (completar). Crie um ArrayList (representando uma fila) contendo 1 objeto do tipo Carro e dois do tipo Moto. Utilize o método abastecerFila para abastecer. Considere o litro da gasolina custando 4.50 reais.

G- Você pode passar um objeto da classe bicicleta para o método completarTanque da classe BombaCombustivel? Desenvolva sua resposta.

H – Considerando uma classe denominada GarrafaPet que implementa Abastecivel e não implementa Veiculo, podemos passar um objeto dessa classe para o método completarTanque da classe BombaCombustivel? Desenvolva sua resposta.

**Questão 2 (2) – Crie uma classe chamada Substitui com um método estático, que retorna uma String, chamado processa (String str). Esse método recebe uma String e retorna uma nova String, toda em caixa alta (letras maiúsculas), invertida e com as letras A, E, I, O substituídas por 4, 3, 1, 0 respectivamente. Dessa forma, se a String passada for “paralelismo”, o método exibirá no console a String “0MS1L3L4R4P”. Crie, ainda nessa classe, o método main, receba do console (Utilizando o Scanner) uma palavra e, em seguida, chame o método processa e imprima no console o retorno.**

**Questão 3 (2.0) – 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("Macieira");

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

v2.setNome("Laranjeira");

teste1(v1, v2);

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

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

v1.a = 81;

**int** x = v1.a;

v1.a=11;

v1.a = v2.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 v1, Arvore v2) {

v1.setNome("Acacia ");

v1 = v2;

v1.setNome("Eucalipto");

v2=v1;

}

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

a = 12;

x=18;

arv.setNome("Coqueiro");

arv = **new** Arvore();

arv.a = 46;

}

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

v2 = **new** Arvore();

v2.setNome("Mamoeiro");

**return** v2;

}

}