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.**

**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.**

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

**Questão 1 (4,0) – Observe a classe abaixo e a saída no console. Escreva todas as classes para que o programa compile e rode sem problemas. A saída deve ser a saída apresentada no console. Algumas observações:**

1)As classes não devem ter métodos set.

2)Você deve mudar o comportamento de suas classes para que ao imprimir uma coleção de veículos, apareçam as características dos veículos no console, ao invés de xxx@12312.

3)O uso de más práticas de POO resultarão em perda de pontos ou até mesmo em anulação da questão.

4)Observe os comentários antes de começar a escrever suas classes.

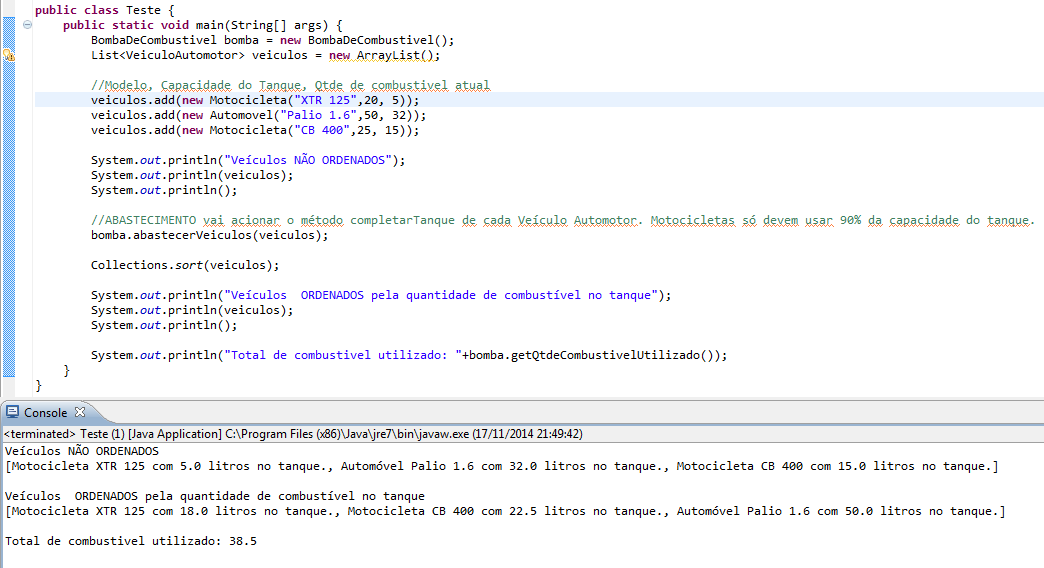
5)O método abastecerVeículos deve chamar o método completarTanque de cada veículo distinto. O tanque das Motocicletas deve ser utilizado em apenas 90% de sua capacidade.

6) Considere o segundo parâmetro do construtor como a capacidade máxima do veículo. O terceiro é a quantidade de combustível inicial.

7) Cuide para que nossa coleção possa ser ordenada pela quantidade de combustível no tanque.

8) Faça com que a classe VeiculoAutomotor não seja instanciável.

9) Faça com que a classe Motocicleta não possa ser estendida, ou seja, nenhuma classe poderá estender Motocicleta.



Questão 3 (4,5) –

A – Escreva uma classe Pessoa com 3 atributos: cpf, nome e idade (defina os tipos). Crie os getters e setters e o que mais for necessário. Implemente um método em uma classe chamada Utils com a seguinte assinatura: public boolean existe (List<Pessoa> x, Pessoa y). Escreva esse método de forma que seja verificada a existência do objeto Pessoa representado por y na lista representada por x, retorne verdadeiro se existir e falso se não existir. Considere que dois objetos Pessoa são iguais se possuem o mesmo CPF. Não é permitida qualquer iteração para realizar esse item, ou seja, não use *for*, *iterator*, etc.

B - Dada a classe Utils, crie o método public Map<String, Pessoa> retornaDados(Set<String> conjuntoPessoas) throws FormatoIncorretoException.

Considere que o conjunto recebido como argumento (conjuntoPessoas) contém Strings no seguinte formato: CPF#nome#idade. Por exemplo, considere os elementos desse conjunto como (080949343-23#Arthur Novaes Moura#34, 310949321-44#Erika da Silva Souza#30, etc.).

Dessa maneira, implemente o método *retornaDados* de forma que seja retornado um mapa da seguinte forma: os elementos de *conjuntoPessoas* devem ser percorridos e o CPF de cada elemento é a chave do Mapa e os valores do mapa são objetos do tipo PessoaResumindo, você irá criar um objeto Pessoa representando cada elemento em *conjuntoPessoas* e adicionar ao mapa. Caso algum elemento em *conjuntoPessoas* possua mais de dois caracteres # ou menos de dois caracteres #, lance a exceção checked FormatoIncorretoException. O formato de saída da exceção deve ser:FormatoIncorretoException: O formato da String [XXX] esta incorreto. [XXX] representa a String em conjuntoPessoas que gerou o erro.

**Questão 3 (1,5) – Observe a questão abaixo. Compila? Se não, qual o erro? Se sim, O que sai no console?**

**public** **class** TT2 {

**private** **static** **int** *k*;

**private** **int** j;

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

TT2 x = **new** TT2();

x.*k* = 55;

x.j = 33;

x = **new** TT2();

x.j = 25;

x.*k* = 42;

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

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

*teste*(x.j, x);

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

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

teste2(x.j, x);

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

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

}

**public** **static** **void** teste(**int** x, TT2 y) {

y.*k* = 111;

x = 91;

}

**public** **static** **void** teste2(**int** x, TT2 y) {

y = **new** TT2();

y.*k* = 37;

y.j = 39;

}

}

**BOA SORTE!**